

第5章 中断系统

(课时：6学时)

教学目的

- 了解中断的基本概念和功能。
- 掌握中断系统的结构和控制方式。
- 掌握中断系统的中断处理过程。
- 用中断系统设计一个手控流水灯控制器。
- 用中断系统设计一个交通信号灯控制器。

学习重点和难点

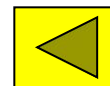
- 中断响应的条件和中断优先级的应用。
- 外部中断源的扩展方法。

第5章 中断系统

- 5.1 中断系统结构与控制
- 5.2 中断处理过程
- 5.3 外部中断扩展
- 本章小结
- 习题

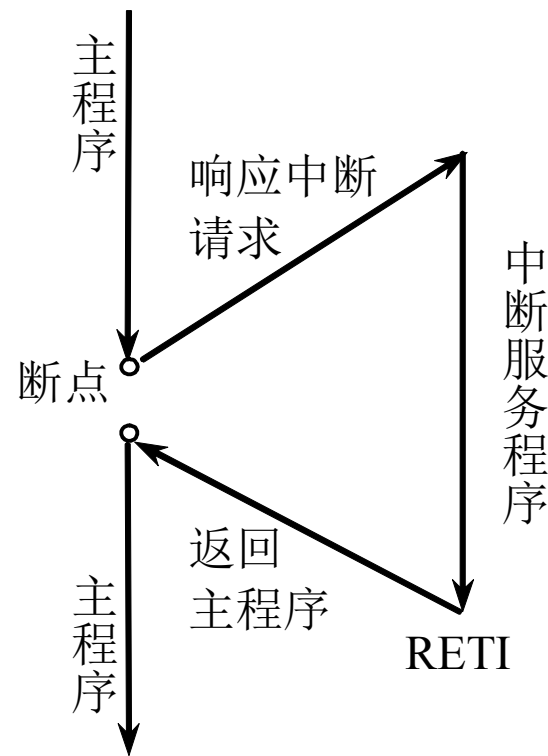
5.1 中断系统结构与控制

- 5.1.1 中断的概念
- 5.1.2 中断的功能
- 5.1.3 中断系统结构
- 5.1.4 中断系统控制



5.1.1 中断的概念

- 在单片机中，当**CPU**在执行程序时，由单片机内部或外部的原因引起的随机事件要求**CPU**暂时停止正在执行的程序，而转向执行一个用于处理该随机事件的程序，处理完后又返回被中止的程序断点处继续执行，这一过程就称为中断。
- 单片机处理中断的4个步骤：中断请求、中断响应、中断处理和中断返回。
- 向**CPU**发出中断请求的来源，或引起中断的原因称为中断源。中断源要求服务的请求称为中断请求。中断源可分为两大类：一类来自单片机内部，称之为内部中断源；另一类来自单片机外部，称之为外部中断源。



5.1.2 中断的功能

- 中断系统是指能实现中断功能的硬件和软件。
- 中断系统的功能一般包括以下几个方面。
 - 进行中断优先级排队
 - 实现中断嵌套
 - 自动响应中断
 - 实现中断返回
- 中断的特点。
 - 可以提高**CPU**的工作效率
 - 实现实时处理
 - 处理故障

5.1.2 中断的功能

(1) 进行中断优先级排队

- 通常，单片机中有多个中断源，设计人员能按轻重缓急给每个中断源的中断请求赋予一定的中断优先级。
- 当两个或两个以上的中断源同时请求中断时，**CPU**可通过中断优先级排队电路首先响应中断优先级高的中断请求，等到处理完优先级高的中断请求后，再来响应优先级低的中断请求。

3、中断优先级

当系统中有多个设备提出中断请求时，就有一个该响应谁的问题，也就是一个优先级的問題。解决优先级的问题一般可有三种方法：软件查询法、简单硬件方法及专用硬件方法。

1) 软件查询法

只需有简单的硬件电路，如将A、B、C三台设备的中断请求信号“或”后作为系统INTR，这时，A、B、C三台设备中只要至少有一台设备提出中断请求，都可以向CPU发中断请求。进入中断服务子程序后，再用软件查询的方式分别对不同的设备的服务，查询程序的设计思想同查询式，查询的前后顺序就给出了设备的优先级，如图6所示。

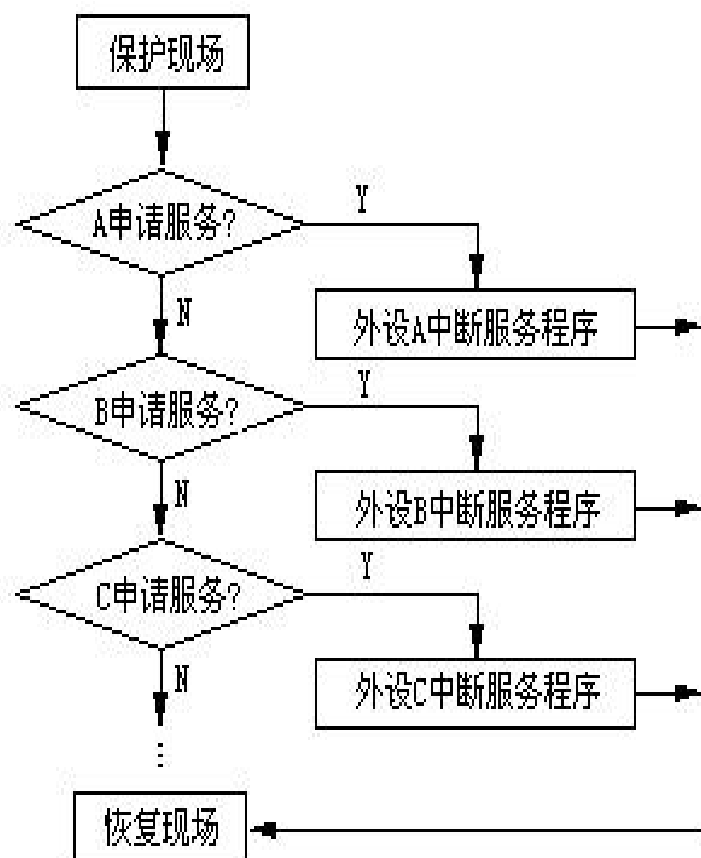


图6 中断优先级软件查询法

件查询法

2) 简单硬件方法

以链式中断优先权排队电路为例

基本设计思想: 将所有设备连成一条链，靠近CPU的设备优先级最高，越远的设备优先级越低，则发出中断响应信号，若级别高的设备发出了中断请求，在它接到中断响应信号的同时，封锁其后的较低级设备使得它们的中断请求不能响应，只有等它的中断服务结束以后才开放，允许为低级的设备服务。如图7所示。

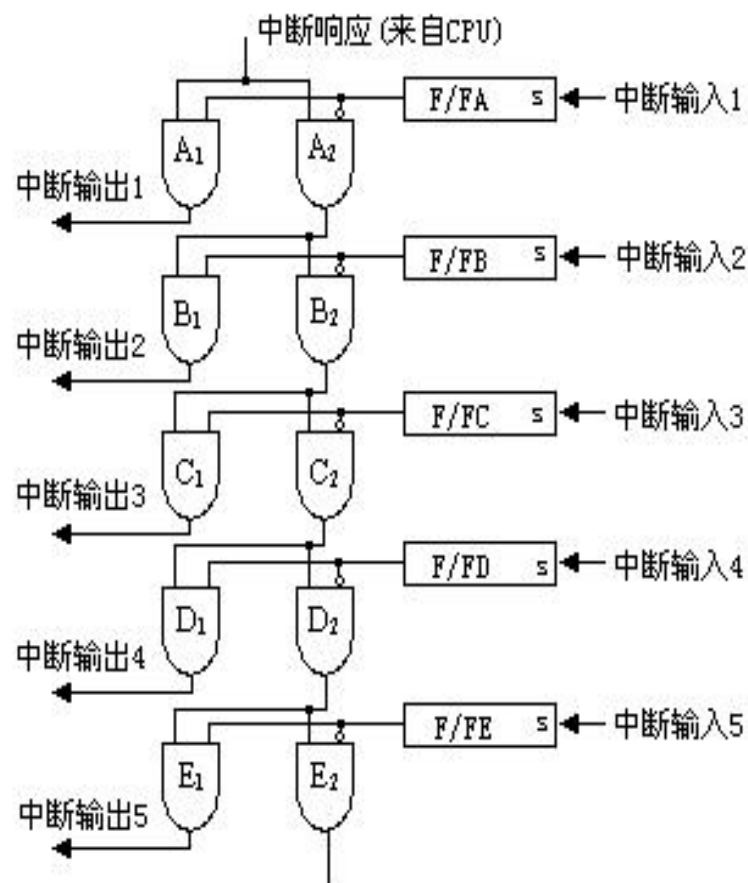


图7 链式中断优先

级排队电路

3) 专用硬件方式

采用可编程的中断控制器芯片，如Intel8259A。

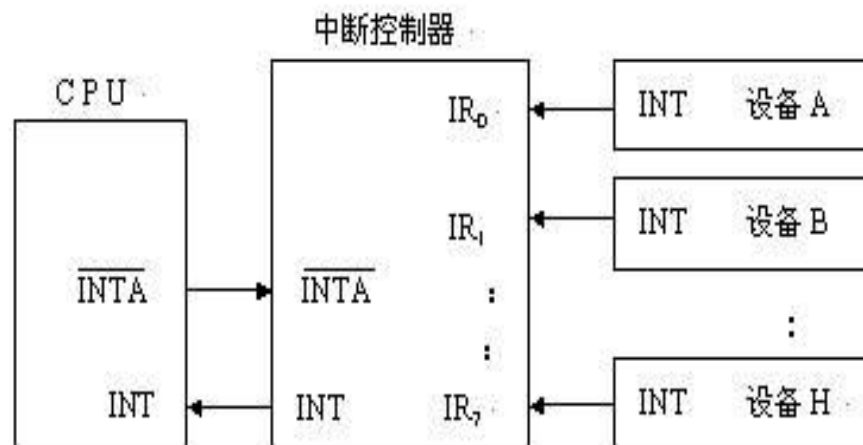


图8 中断控制器的系统连接

5.1.2 中断的功能

(2) 实现中断嵌套

- **CPU**在响应某一中断源中断请求而进行中断处理时，若有中断优先级更高的中断源发出中断请求，**CPU**会暂停正在执行的中断服务程序，转向执行中断优先级更高的中断源的中断服务程序，等处理完这个高优先级的中断请求后，再返回来继续执行被暂停的中断服务程序。这个过程称为中断嵌套。

5.1.2 中断的功能

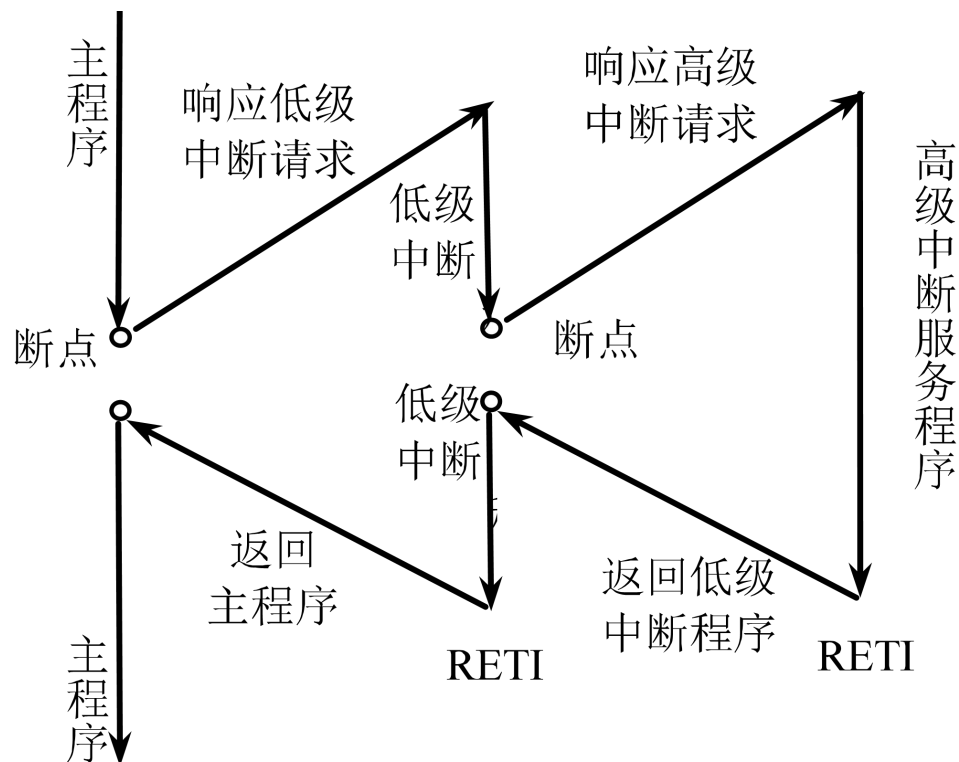
(3) 自动响应中断

- 中断源向**CPU**发出的中断请求是随机的。通常，**CPU**总是在每条指令的最后状态对中断请求信号进行检测；当某一中断源发出中断请求时，**CPU**能根据相关条件(如中断优先级、是否允许中断)进行判断，决定是否响应这个中断请求。若允许响应这个中断请求，**CPU**在执行完相关指令后，会自动完成断点地址压入堆栈、中断矢量地址送入程序计数器**PC**、撤除本次中断请求标志，转入执行相应中断服务程序。

5.1.2 中断的功能

(4) 实现中断返回

- **CPU**响应某一中断源中断请求，转入执行相应中断服务程序，在执行中断服务程序最后的中断返回指令时，会自动弹出堆栈区中保存的断点地址，返回到中断前的原程序中。



2. 中断的特点

((1))可以提高**CPU**的工作效率

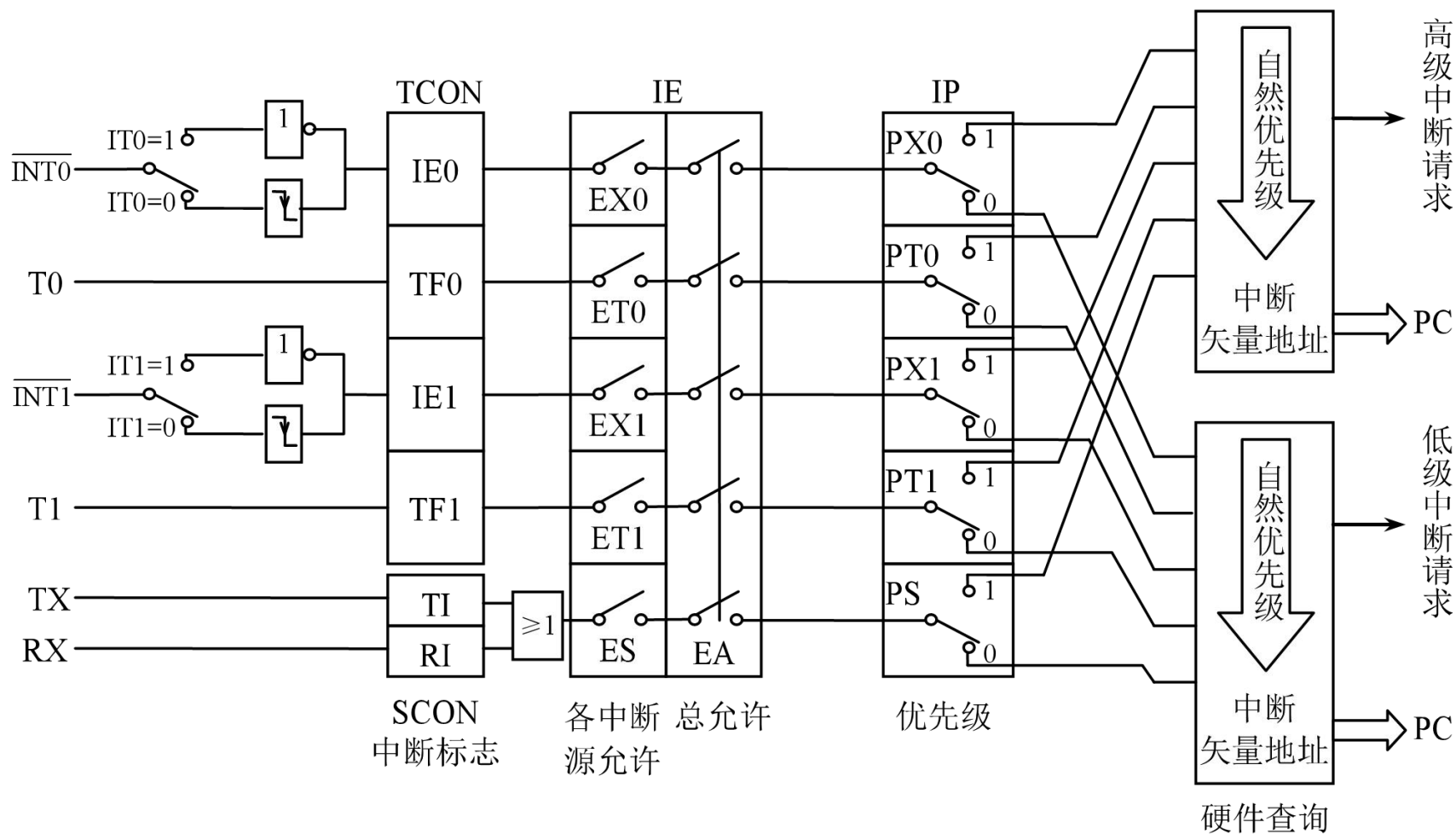
((2))实现实时处理

((3))处理故障

5.1.3 中断系统结构

- **8051**单片机的中断系统主要由与中断有关的**4**个特殊功能寄存器和硬件查询电路等组成。
 - 定时器控制寄存器**TCON** ——主要用于保存中断信息。
 - 串行口控制寄存器**SCON** ——主要用于保存中断信息。
 - 中断允许寄存器**IE** ——主要用于控制中断的开放和关闭。
 - 中断优先级寄存器**IP**——主要用于设定优先级别。
 - 硬件查询电路——主要用于判定**5**个中断源的自然优先级别。
- **8051**单片机的中断源有五个，可分为三类。
 - 外部中断
 - 定时中断
 - 串行口中断

5.1.3 中断系统结构



5.1.3 中断系统结构

1. 外部中断类

- 外部中断是由外部原因(如打印机、键盘、控制开关、外部故障)引起的, 可以通过两个固定引脚来输入到单片机内的信号, 即外部中断0(**INT0**)和外部中断1(**INT1**)。
- **INT0**——外部中断0中断请求信号输入端, **P3.2**的第二功能。
由定时器控制寄存器**TCON**中的**IT0**位决定中断请求信号是低电平有效还是下降沿有效。一旦输入信号有效, 即向**CPU**申请中断, 并且硬件自动使**IE0**置1。
- **INT1**——外部中断1中断请求信号输入端, **P3.3**的第二功能。
由定时器控制寄存器**TCON**中的**IT1**位决定采用电平触发方式还是边沿触发方式。一旦输入信号有效, 即向**CPU**申请中断, 并且硬件自动使**IE1**置1。

5.1.3 中断系统结构

2. 定时中断类

- 定时中断是由内部定时(或计数)溢出或外部定时(或计数)溢出引起的，即定时器0(T0) 中断和定时器1(T1) 中断。
- 当定时器对单片机内部定时脉冲进行计数而发生计数溢出时，即表明定时时间到，由硬件自动使TF0(TF1)置1，并申请中断。当定时器对单片机外部计数脉冲进行计数而发生计数溢出时，即表明计数次数到，由硬件自动使TF0(TF1)置1，并申请中断。外部计数脉冲是通过两个固定引脚来输入到单片机内的。
- **T0外部计数输入端——P3.4的第二功能。**
当定时器0工作于计数方式时，外部计数脉冲下降沿有效，定时器0进行加1计数。
- **T1外部计数输入端——P3.5的第二功能。**
当定时器1工作于计数方式时，外部计数脉冲下降沿有效，定时器1进行加1计数。

- 定时器 / 计数器0、1、2溢出中断
- **MCS—52**有**3**个定时器 / 计数器，分别称为定时器 / 计数器0（**T 0**）、定时器 / 计数器1（**T 1**）和定时器 / 计数器2（**T 2**）。
- 三个定时器 / 计数器既可作为定时器用，又可作为计数器用，可编程设定。当作为定时器使用时，其中断信号取自内部定时时钟，当作为计数器使用时，其中断请求信号取自**T 0（P3.4）、T1（P3.5）和T2（P1.0）、T2EX（P1.1）**引脚。启动**T0、T1或T2**后，每来一个时钟脉冲或在引脚上每检测到一个脉冲信号，计数器就加**1**一次，当计数器的值由全**0**变为全**1**时或当**T2EX**引脚有负跳变产生时（必须**EXEN2=1**）就会向**CPU**申请中断。

5.1.3 中断系统结构

3. 串行口中断类

串行口中断是为接收或发送串行数据而设置的。串行中断请求是在单片机芯片内部发生的。

- RXD——串行口输入端，P3.0的第二功能。
当接收完一帧数据时，硬件自动使RI置1，并申请中断。
- TXD——串行口输出端，P3.1的第二功能。

当发送完一帧数据时，硬件自动使TI置1，并申请中断。

定时器 / 计数器0、1、2溢出中断与串行口中断属于内部中断。
定时器 / 计数器2的中断请求，其中断源有溢出中断和定时器 / 计数器2外部中断两种方式。

定时器 / 计数器0、1、2溢出中断与串行口中断属于内部中断。

5.1.4 中断系统控制

1. 定时器控制寄存器TCON

定时器控制寄存器**TCON**的作用是控制定时器的启动与停止，并保存**T0**、**T1**的溢出中断标志和外部中断的中断标志。

TCON的格式：

	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TCON	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H

(88H)

各位的功能说明：

(1) **TF1(TCON.7)**：定时器1溢出标志位。定时器1被启动计数后，从初值开始进行加1计数，当定时器1计满溢出时，由硬件自动使**TF1**置1，并申请中断。该标志一直保持到**CPU**响应中断后，才由硬件自动清0。也可用软件查询该标志，并由软件清0。

(2) **TR1(TCON.6)**：定时器1启停控制位。

5.1.4 中断系统控制

(3) IT1(TCON.2): 外部中断1触发方式选择位。

当**IT1=0**时，外部中断1为电平触发方式。在这种方式下，**CPU**在每个机器周期的**S5P2**期间对 **INT1(P3.3)**引脚采样，若采到低电平，则认为有中断申请，硬件自动使**IEI**置**1**；若为高电平，认为无中断申请或中断申请已撤除，硬件自动使**IEI**清**0**。在电平触发方式中，**CPU**响应中断后硬件不能自动使**IEI**清**0**，也不能由软件使**IEI**清**0**，所以在中断返回前必须撤消**INT1**引脚上的低电平，否则将再次响应中断造成出错。

当**IT1=1**时，外部中断1为边沿触发方式。**CPU**在每个机器周期的**S5P2**期间采样 **INT1 (P3.3)**引脚。若在连续两个机器周期采样到先高电平后低电平，则认为有中断申请，硬件自动使**IEI**置**1**，此标志一直保持到**CPU**响应中断时，才由硬件自动清**0**。在边沿触发方式下，为保证**CPU**在两个机器周期内检测到先高后低的负跳变，输入高低电平的持续时间至少要保持**12**个时钟周期。

边沿触发和电平触发的区别

- 如果是下降沿触发，当从高至低电平转变时，触发产生；低电平保持多久都只产生一次。所以，边沿触发适合于负脉冲形式输入的外部中断请求。
- 如果是低电平触发，那么在低电平时间内中断一直有效；因此如果在电平没有恢复之前中断程序就已经执行完成从而退出，那么会在退出后又再次进入中断；但只要中断没有退出是不会重复触发的。所以，电平触发方式适合于外部中断以低电平输入而且中断服务能清除外部中断源的情况（实际中通常采用这样的做法：在中断退出前关闭中断，等后面恰当时机再打开）。

5.1.4 中断系统控制

各位的功能说明：

- (4) **IE1(TCON.3)**: 外部中断1请求标志位。**IE1=1**表示外部中断1向CPU申请中断。当CPU响应外部中断1的中断请求时，由硬件自动使**IE1清0**(边沿触发方式)。
- (5) **TF0(TCON.5)**: 定时器0溢出标志位。其功能同**TF1**。
- (6) **TR0(TCON.4)**: 定时器0启、停控制位。其功能同**TR1**。
- (7) **IE0(TCON.1)**: 外部中断0请求标志位。其功能同**IE1**。
- (8) **IT0(TCON.0)**: 外部中断0触发方式选择位。其功能同**IT1**。

5.1.4 中断系统控制

2. 串行口控制寄存器SCON

串行口控制寄存器**SCON**的低2位**TI**和**RI**保存串行口的接收中断和发送中断标志。

SCON的格式:

	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
SCON	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H

(98H)

各位的功能说明:

- (1) **TI (SCON.1)**: 串行发送中断请求标志。**CPU**将一个字节数据写入发送缓冲器**SBUF**后启动发送, 每发送完一帧数据, 硬件自动使**TI**置**1**。但**CPU**响应中断后, 硬件并不能自动使**TI**清**0**, 必须由软件使**TI**清**0**。
- (2) **RI (SCON.0)**: 串行接收中断请求标志。在串行口允许接收时, 每接收完一帧数据, 硬件自动使**RI**置**1**。但**CPU**响应中断后, 硬件并不能自动使**RI**清**0**, 必须由软件使**RI**清**0**。

5.1.4 中断系统控制

3. 中断允许寄存器IE

中断允许寄存器**IE**的作用是控制**CPU**对中断的开放或屏蔽以及每个中断源是否允许中断。

IE的格式：

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

IE AFH ACH ABH AAH A9H A8H

(A8H)

各位的功能说明：

- (1) **EA(IE.7)**: **CPU**中断总允许位。**EA=1**, **CPU**开放中断。每个中断源是被允许还是被禁止, 分别由各中断源的中断允许位确定; **EA=0**, **CPU**屏蔽所有的中断要求, 称为关中断。
- (2) **ES(IE.4)**: 串行口中断允许位。**ES=1**, 允许串行口中断; **ES=0**, 禁止串行口中断。

5.1.4 中断系统控制

各位的功能说明：

- (3) **ET1(IE.3)**: 定时器1中断允许位。**ET1=1**, 允许定时器1中断;
ET1=0, 禁止定时器1中断。
- (4) **EX1(IE.2)**: 外部中断1中断允许位。**EX1=1**, 允许外部中断1
中断; **EX1=0**, 禁止外部中断1中断。
- (5) **ET0(IE.1)**: 定时器0中断允许位。**ET0=1**, 允许定时器0中断;
ET0=0, 禁止定时器0中断。
- (6) **EX0(IE.0)**: 外部中断0中断允许位。**EX0=1**, 允许外部中断0
中断; **EX0=0**, 禁止外部中断0中断。

5.1.4 中断系统控制

4. 中断优先级寄存器IP

中断优先级寄存器**IP**的作用是设定各中断源的优先级别。

IP的格式：

IP

—	—	—	PS	PT1	PX1	PT0	PX0
BCH			BBH	BAH	B9H	B8H	

(B8H)

各位的功能说明：

- (1) **PS(IP.4)**：串行口中断优先级控制位。**PS=1**，串行口为高优先级中断；**PS=0**，串行口为低优先级中断。
- (2) **PT1(IP.3)**：定时器1中断优先级控制位。**PT1=1**，定时器1为高优先级中断；**PT1=0**，定时器1为低优先级中断。

5.1.4 中断系统控制

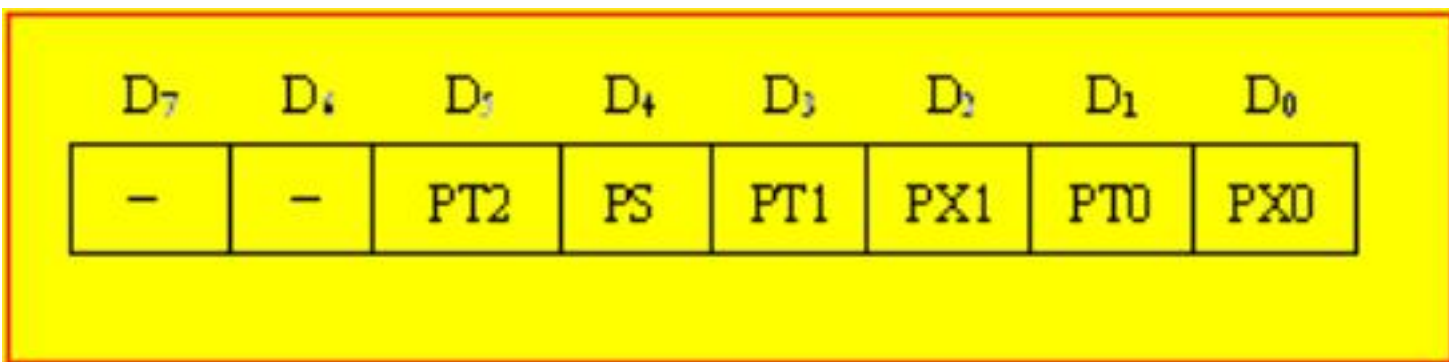
各位的功能说明：

- (3) **PX1(IP.2)**: 外部中断1中断优先级控制位。**PX1=1**, 外部中断1为高优先级中断; **PX1=0**, 外部中断1为低优先级中断。
- (4) **PT0(IP.1)**: 定时器0中断优先级控制位。**PT0=1**, 定时器T0为高优先级中断; **PT0=0**, 定时器0为低优先级中断。
- (5) **PX0(IP.0)**: 外部中断0中断优先级控制位。**PX0=1**, 外部中断0为高优先级中断; **PX0=0**, 外部中断0为低优先级中断。

MCS-51单片机有两个中断优先级，每个中断源都可以通过编程确定为高优先级中断或低优先级中断，高优先级的优先权高。同一优先级别中的中断源不止一个，所以也有中断优先权排队问题。

(1) 软件排队

中断优先级软件排队由中断优先级寄存器**IP**控制。**IP**的格式如下：



IP中的每一位都可以由软件来置1或清零。
置 1——高优先级， 0——低优先级。

(2) 硬件排队

对于同一优先级中的中断源优先权排队由中断系统的硬件确定，用户无法自行安排。如下表。

中断源	中断入口地址	同级内优先权排列
外部中断0	0003H	最高 ↓ 最低
定时器/计数器0 中断	000BH	
外部中断1	0013H	
定时器/计数器1 中断	001BH	
串行接口中断	0023H	
定时器/计数器2 中断	002BH	

MCS-51系列单片机的中断优先权有三条原则：

- (1) 正在进行的 interrupt 过程不能被新的同级或低优先级的 interrupt 请求所 interrupt，一直到该 interrupt 服务程序结束，返回了主程序且执行了主程序中的一条指令后，CPU才响应新的 interrupt 请求。**
- (2) 正在进行的低优先级 interrupt 服务程序能被高优先级 interrupt 请求所 interrupt，实现二级 interrupt 嵌套。**
- (3) CPU同时接收到几个 interrupt 请求时，首先响应优先权最高的 interrupt 请求。**

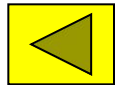
二、中断的控制

寄存器名称		D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
定时器控制 寄存器	TCON(88H)	TF1		TF0		IE1	IT1	IE0	IT0
	位地址	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H
串行口控制 寄存器	SCON(98H)							TI	RI
	位地址	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H
中断允许 寄存器	IE(A8H)	EA			ES	ET1	EX1	ET0	EX0
	位地址	AFH			ACH	ABH	AAH	A9H	A8H
定时器2 控制寄存器	T2CON	TF2	EXF2						
	位地址	CF	CE						

除了外部中断电平触发方式外，其他各个中断实际上是由标志位IE0、IE1、TF0、TF1、TI、RI、TF2、EXF2 置位引起的。这些标志位除了由相应的硬件置位外，还可以由软件置位，因此，如果有需要，可以用程序安排产生中断。

5.2 中断处理过程

- 5.2.1 中断响应
- 5.2.2 中断处理
- 5.2.3 中断返回
- 5.2.4 中断的应用实例



5.2.1 中断响应

1. CPU的中断响应条件

CPU响应中断必须首先满足以下三个基本条件。

- (1) 有中断源发出中断请求。
- (2) 中断总允许位**EA=1**。
- (3) 请求中断的中断源的中断允许位为**1**。

在满足以上条件的基础上，若有下列任何一种情况存在，中断响应都会受到阻断。

- (1) **CPU**正在执行一个同级或高优先级的中断服务程序。
- (2) 正在执行的指令尚未执行完。
- (3) 正在执行中断返回指令 **RETI**或者对专用寄存器**IE**、**IP**进行读/写的指令。**CPU**在执行完上述指令之后，要再执行一条指令，才能响应中断请求。

例 8000H MOV A, #06H

① 8002H MOV B, #05H

8005H MUL AB

② 8006H MOV IE, #05H

8009H MOV R0, #30H

800BH

中断信号发生在①处时,压入堆栈保存的是下条指令地址**8005H**

中断信号发生在②处时,压入堆栈保存的是再下条指令地址**800BH**

5.2.1 中断响应

2. 中断优先级的判定

- 中断源的优先级别分为高级和低级，通过由软件设置中断优先级寄存器**IP**相关位来设定每个中断源的级别。
- 如果几个同一优先级级别的中断源同时向**CPU**请求中断，**CPU**通过硬件查询电路首先响应自然优先级较高的中断源的中断请求。
- 中断可实现两级中断嵌套。高优先级中断源可中断正在执行的低优先级中断服务程序，除非执行了低优先级中断服务程序的**CPU**关中断指令。同级或低优先级的中断不能中断正在执行的中断服务程序。



5.2.1 中断响应

3. 中断响应过程

- 在满足中断响应条件时，**CPU**响应中断。首先，将相应的优先级状态触发器置**1**，以屏蔽同级别中断源的中断请求。其次，硬件自动生成长调用指令(**LCALL**)，把断点地址压入堆栈保护(但不保护状态寄存器**PSW**及其他寄存器内容)，然后将中断源对应的中断入口地址装入程序计数器**PC**中，使程序转向该中断入口地址，并执行中断服务程序。
- 8051**单片机的中断入口地址(称为中断矢量)由单片机硬件电路决定。

中 断 源	中断入口地址
外部中断0	0003H
定时器T0中断	000BH
外部中断1	0013H
定时器T1中断	001BH
串行口中断	0023H

1. 中断响应的过程

- **CPU响应中断后**，由硬件执行如下功能：
- （1）根据中断请求源的优先级高低，使相应的优先级状态触发器置1。
- （2）保留断点，即把程序计数器PC的内容(即断点地址)推入堆栈保存。
- （3）清相应的中断请求标志位IE0、IE1、TF0、TF1。
- （4）把被响应的中断入口地址送入PC，从而转入相应的中断服务程序入口。

2. 中断响应时间

中断返回指令：RETI= RET指令 + 通知CPU中断服务已结束。

中断响应时间：正常中断响应时间至少为3~8个机器周期，如果有同级或高级中断服务，将延长中断响应时间。

5.2.2 中断处理

- 中断处理就是执行中断服务程序，从中断入口地址开始执行，直到返回指令(**RETI**)为止。此过程一般包括三部分内容，一是保护现场，二是处理中断源的请求，三是恢复现场。
- 通常，主程序和中断服务程序都会用到累加器**A**、状态寄存器**PSW**及其他一些寄存器。在执行中断服务程序时，**CPU**若用到上述寄存器，就会破坏原先存在这些寄存器中的内容，中断返回，将会造成主程序的混乱。因此，在进入中断服务程序后，一般要先保护现场，然后再执行中断处理程序，在返回主程序以前，再恢复现场。

5.2.2 中断处理

- 在编写中断服务程序时要注意以下几个方面。
- 一般在这些中断入口地址区存放一条无条件转移指令，转向中断服务程序的起始地址。
- 若要求禁止更高优先级中断源的中断请求，应先用软件关闭**CPU**中断或屏蔽更高级中断源的中断，在中断返回前再开放被关闭或被屏蔽的中断。
- 在保护现场和恢复现场时，为了不使现场数据受到破坏而造成混乱，在保护现场之前要关中断，在保护现场之后再开中断；在恢复现场之前关中断，在恢复现场之后再开中断。

5.2.3 中断返回

1. 中断返回

- 中断返回是指中断服务完成后，**CPU**返回到原程序的断点(即原来断开的位置)，继续执行原来的程序。
- 中断返回通过执行中断返回指令**RETI**来实现，该指令的功能是首先将相应的优先级状态触发器置**0**，以开放同级别中断源的中断请求；其次，从堆栈区把断点地址取出，送回到程序计数器**PC**中。因此，不能用**RET**指令代替**RETI**指令。

5.2.3 中断返回

2. 中断请求的撤除

CPU响应某中断请求后，在中断返回前，应该撤消该中断请求，否则会引起另一次中断。不同中断源中断请求的撤除方法是不一样的。

(1) 定时器溢出中断请求的撤除

CPU在响应中断后，硬件会自动清除中断请求标志**TF0**或**TF1**。

(2) 串行口中断的撤除

在**CPU**响应中断后，硬件不能清除中断请求标志**TI**和**RI**，而由软件来清除相应的标志。

5.2.3 中断返回

(3) 外部中断的撤除

- 外部中断为边沿触发方式时，**CPU**响应中断后，硬件会自动清除中断请求标志**IE0**或**IEI**。
- 外部中断为电平触发方式时，**CPU**响应中断后，硬件会自动清除中断请求标志**IE0**或**IEI**，但由于加到**INT0**或**INT1**引脚的外部中断请求信号并未撤除，中断请求标志**IE0**或**IEI**会再次被置**1**，所以在**CPU**响应中断后应立即撤除**INT0**或**INT1**引脚上的低电平。一般采用加一个**D**触发器和几条指令的方法来解决这个问题。

复位状态

CPU响应中断请求后，在中断返回（执行**RETI**）前，必须撤除请求，将中断标志位清除，回复到原始的状态，否则会错误地再一次引起中断响应。

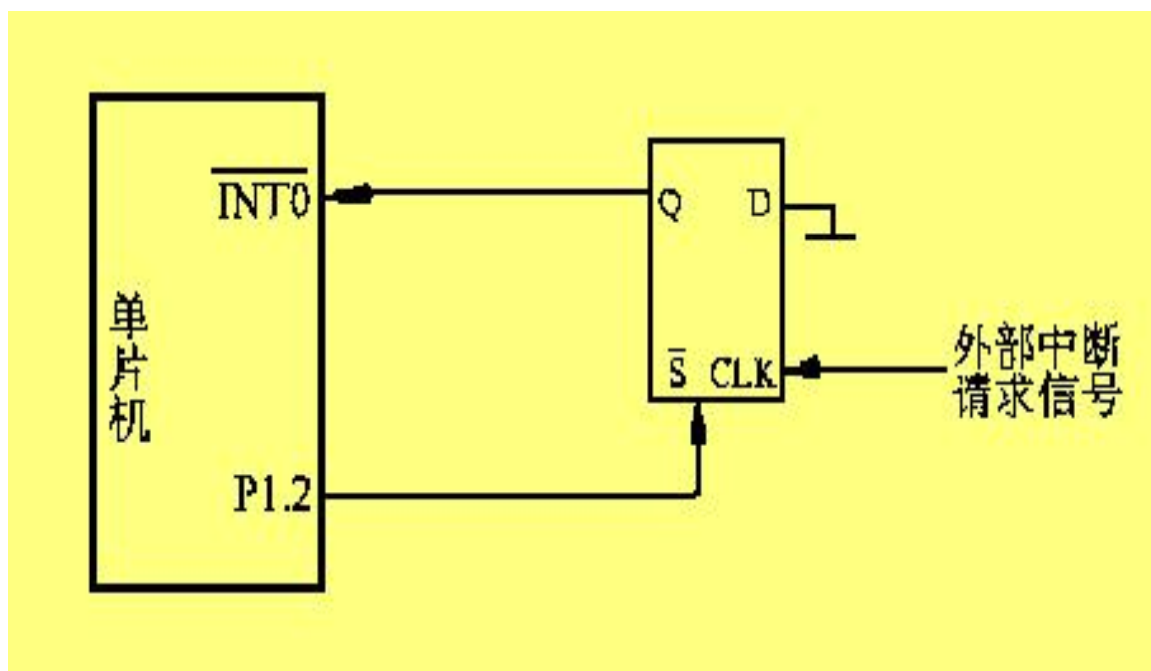
对于定时器 / 计数器0、1的中断请求及边沿触发方式的外部中断0、1，CPU在响应中断后用硬件清除了相应的中断请求标志**TF0**、**TF1**、**IE0**、**IE1**，即自动撤除了中断请求。

对于串行接口中断及定时器 / 计数器2中断，CPU响应中断后没有用硬件清除中断标志位，必须由用户编制的中断服务程序来清除相应的中断标志。如用指令**CLR TF2**清除**TF2**，用指令**CLR EXF2**清除**EXF2**等。

对于电平触发的外部中断，由于CPU对、引脚没有控制作用，也没有相应的中断请求标志位，因此需要外接电路来撤除中断请求信号。

外部中断请求信号通过D触发器加到单片机、引脚上。当外部中断信号使D触发器的CLK端发生正跳变时，由于D端接地，Q端输出0，向单片机发出中断请求。CPU响应中断后，利用一根口线如P1.2作应答线，在中断服务程序中用两条指令：

如 **CLR P1.2**
SETB P1.2
来撤销中断请求。



四、程序的初始化及中断服务程序

1. 程序初始化的概念及步骤

所谓程序中断系统初始化，就是指用户对中断控制的相关特殊寄存器中的各有关控制位进行赋值。其步骤如下：

- 1) 置位相应中断的中断允许标志及EA。
- 2) 设定所用中断源的中断优先级。
- 3) 对外部中断应设定中断请求信号形式（电平触发 / 边沿触发）。对于定时 / 计数中断应设置工作方式（定时 / 计数）

比如：编程设定为低电平触发的高优先级中断源。

CLR IT1 或 MOV IP , #04H

SETB EX1 MOV TCON, #04H

SETB PX1 MOV IE , #84H

SETB EA

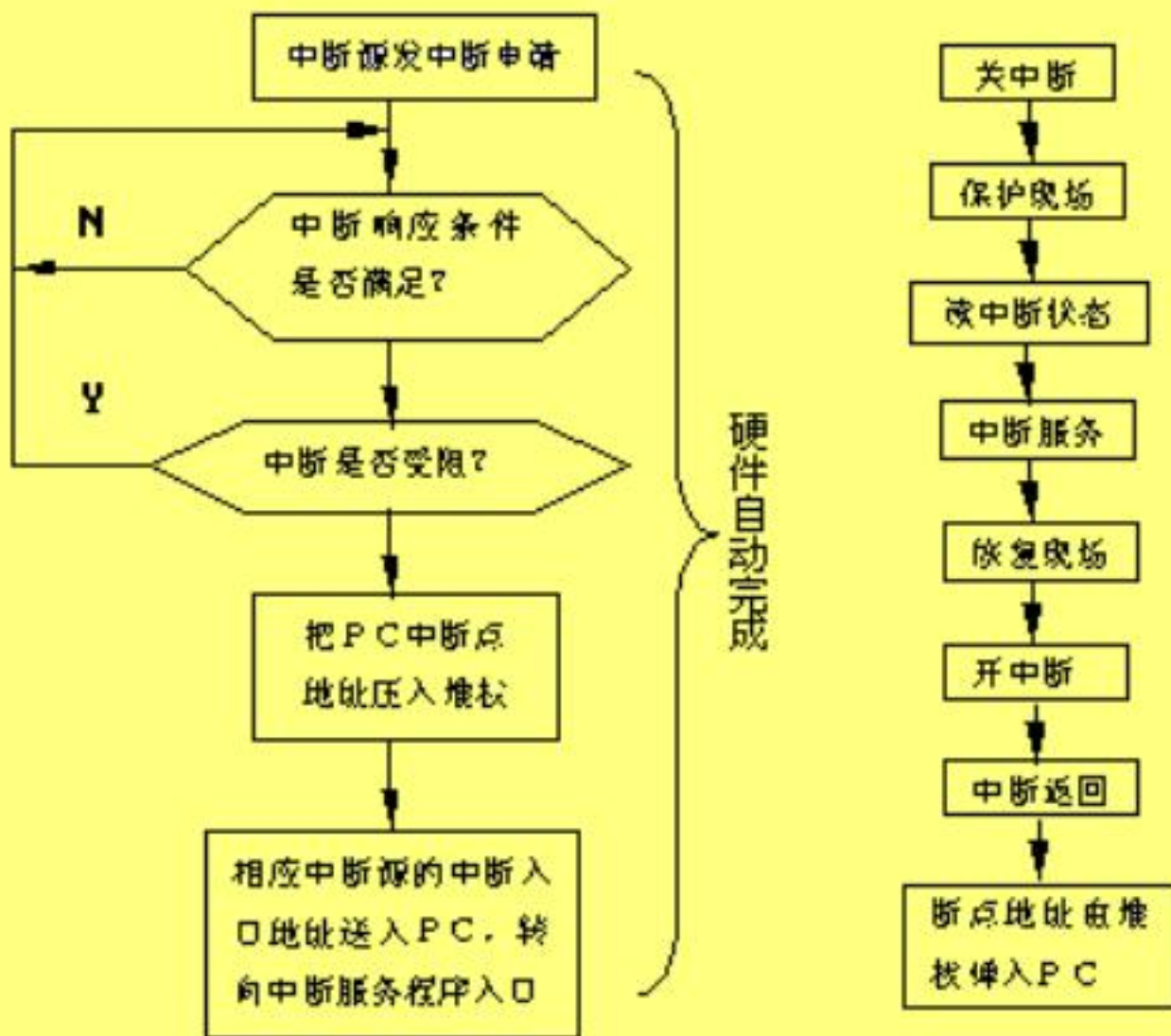
2.中断服务程序

CPU响应中断后即转至中断服务程序的入口。从中断服务和程序的第一条指令开始到返回指令，这个过程称为中断处理或称中断服务。不同的中断源服务的内容及要求各不相同，其处理过程也就有所区别。一般情况下，中断处理包括两部分内容：一是保护现场，二是为中断源服务。

用户在编写中断服务程序时应注意的问题：

- 1) 在中断入口矢量地址单元处存放一条无条件转移指令，而转至程序设计者安排的其它地址单元。
- 2) 若要在执行当前中断程序时禁止更高优先级中断，可在中断服务程序中应用软件关闭CPU中断，或屏蔽更高级中断源的中断，在中断返回前再开放中断。
- 3) 在保护现场和恢复现场时，为了不使现场信息受到破坏，一般在此情况下应关闭CPU中断，使CPU暂不响应新的中断请求。这样就要在编写中断服务程序时，应注意在保护现场之前要关中断，在保护现场之后若允许高优先级中断，则应开中断。同样在恢复现场之前应关中断，恢复之后再开中断。
- 4) 中断服务程序的最后一条指令必须是中断返回指令**RETI**。CPU执行该指令时，先将相应的优先级状态触发器清零，然后从堆栈中弹出栈顶的两个字节到PC，从而返回到断点处。

中断处理流程图



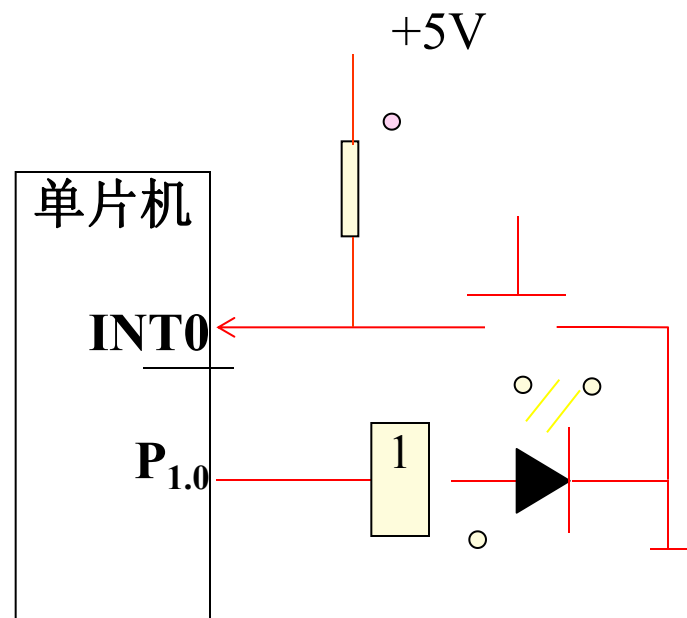
中断处理举例

要求每次按动按键，使外接发光二极管LED改变一次亮灭状态。

解：INT0输入按键信号，P_{1.0}输出改变LED状态。

1) 使用跳变触发：每次跳变引起一次中断请求

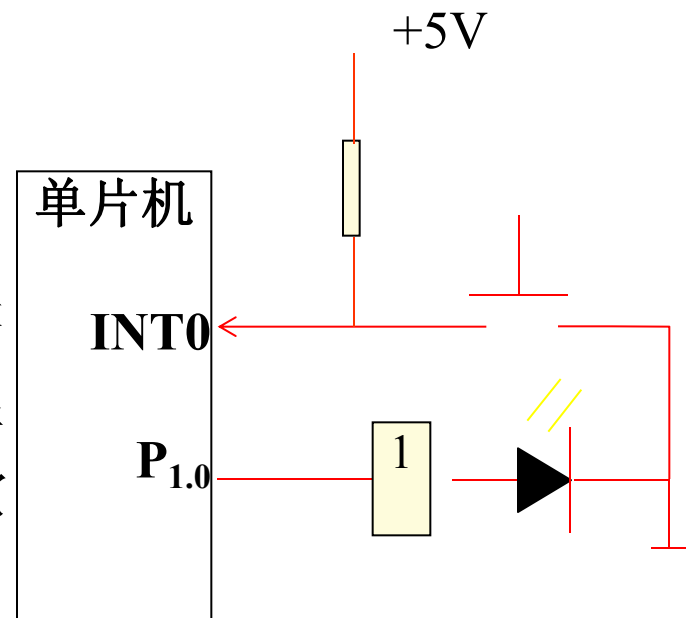
```
ORG    0000H
AJMP   MAIN
      ORG    0003H  ; 中断入口
AJMP   PINT0
      ORG    0100H  ; 主程序
MAIN:  SETB   EA    ; 开总允许开关
      SETB   EX0    ; 开INT0中断
      SETB   IT0    ; 负跳变触发中断
      Here: SJMP   Here ; 相当于执行其它任务
      ORG    0200H  ; 中断服务程序
PINT0: CPL   P1.0   ; 改变LED
      RETI          ; 返回主程序
```



2)使用电平触发：避免一次按键引起多次中断响应。

软件等待按键释放；硬件清除中断信号。

```
ORG    0000H;
AJMP   MAIN
ORG    0003H  ; 中断入口
AJMP   PINT0
ORG    0100H  ; 主程序
MAIN:  SETB   EA      ; 开总允许开关
       SETB   EX0     ; 开INT0中断
       CLR    IT0     ; 低电平触发中断
       Here:  SJMP  Here ; 相当于执行其它任务
                   ;
       ORG    0200H; 中断服务程序
PINT0:  CPL    P1.0    ; 改变LED
WAIT:  JNB    P3.2, WAIT; 等按键释放
       RETI           ; 返回主程序
```



5.2.4 中断的应用实例

- 中断系统的初始化步骤。

- (1) 开放**CPU**中断和有关中断源的中断允许，设置中断允许寄存器**IE**中相应的位。
- (2) 根据需要确定各中断源的优先级别，设置中断优先级寄存器**IP**中相应的位。
- (3) 根据需要确定外部中断的触发方式，设置定时器控制寄存器**TCON**中相应的位。

5.2.4 中断的应用实例

【例1】 P1口做输出口，控制八只灯(P1口输出低电平时灯被点亮)，利用手控单脉冲信号作为外部中断信号，控制八只灯按一定的规律循环点亮。

解：手控单脉冲信号作为外部中断信号由INT0(P3.2)管脚输入，设置中断允许寄存器IE中的EA、EX0位为1；只有一个中断源可不设置优先级别；中断触发方式设为边沿触发，控制位IT0应设置为1。

				ORG 0000H	;程序入口
0000H	02	00	50	LJMP MAIN	;转向主程序
				ORG 0003H	;外部中断0的入口地址
0003H	02	01	00	LJMP INT	;转向中断服务程序
				ORG 0050H	
0050H	D2	AF		MAIN: SETB EA	
0052H	D2	A8		SETB EX0	
0054H	D2	88		SETB IT0	;中断触发方式为边沿触发
0056H	74	FE		MOV A,#0FEH	
0058H	F5	90		MOV P1,A	
005AH	80	FE		SJMP \$	
				ORG 0100H	
0100H	23			INT: RL A	;中断服务程序
0101H	F5	90		MOV P1,A	
0103H	32			RETI	
				END	

5.2.4 中断的应用实例

【例2】 P1口做输出口，正常时控制八只灯(P1口输出低电平时灯被点亮)每隔0.5秒全亮全灭一次；按下开关1八只灯从右向左依次点亮，按下开关2八只灯从左向右依次点亮。

解：开关1的低电平脉冲信号作为外部中断信号由INT0(P3.2)管脚输入，开关2的低电平信号作为外部中断信号由INT1(P3.3)管脚输入。中断允许寄存器IE中相应的EA、EX1、EX0位设置为1。

外部中断0为低优先级，IP中的PX0位设置为0；外部中断1为高优先级，IP中的PX1位设置为1。

外部中断0的中断触发方式设为边沿触发，控制位IT0应设置为1；外部中断1的中断触发方式设为电平触发，控制位IT1应设置为0。

5.2.4 中断的应用实例

				ORG	0000H		;程序入口
0000H	02	00	30	LJMP	MAIN		;转向主程序
				ORG	0003H		;外部中断0的入口地址
0003H	02	01	00	LJMP	INT0		;转向外部中断0中断服务程序
				ORG	0013H		;外部中断1的入口地址
0013H	02	02	00	LJMP	INT1		;转向外部中断1中断服务程序
				ORG	0030H		
0030H	75	81	30	MAIN:	MOV	SP,#30H	
0033H	75	A8	85		MOV	IE,#85H	;允许外部中断0、外部中断1中断
0036H	75	B8	04		MOV	IP,#04H	;外部中断1为高优先级
0039H	75	88	01		MOV	TCON,#01H	;外部中断0为边沿触发
003CH	74	00			MOV	A,#00H	
003EH	F5	90		LP1:	MOV	P1,A	
0040H	12	03	00		LCALL	DELAY	
0043H	F4				CPL	A	
0044H	80	F8			SJMP	LP1	

				ORG 0100H	
0100H	C0	E0	INT0:	PUSH A	;外部中断0中断服务程序
0102H	C0	D0		PUSH PSW	
0104H	C2	D4		CLR RS1	;选择第1组工作寄存器
0106H	D2	D3		SETB RS0	
0108H	7A	07		MOV R2,#07H	
010AH	74	FE		MOV A,#0FEH	;灯点亮的初始状态
010CH	F5	90	NEXT0:	MOV P1,A	
010EH	12	03	00	LCALL DELAY	
0111H	23			RL A	;点亮左边一盏灯
0112H	DA	F8		DJNZ R2,NEXT0	
0114H	D0	D0		POP PSW	
0116H	D0	E0		POP A	
0118H	32			RETI	

外部中断1中断服务程序：

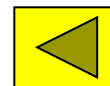
				ORG	0200H	
0200H	C0	E0	INT1:	PUSH	A	;外部中断1中断服务程序
0202H	C0	D0		PUSH	PSW	
0204H	D2	D4		SETB	RS1	;选择第2组工作寄存器
0206H	C2	D3		CLR	RS0	
0208H	7A	07		MOV	R2,#07H	
020AH	74	7F		MOV	A,#7FH	;灯点亮的初始状态
020CH	F5	90	NEXT1:	MOV	P1,A	
020EH	12	03	00	LCALL	DELAY	
0211H	03			RR	A	;点亮右边一盏灯
0212H	DA	F8		DJNZ	R2,NEXT1	
0214H	D0	D0		POP	PSW	
0216H	D0	E0		POP	A	
0218H	32			RETI		

延时子程序：

			ORG	0300H	
0300H	7B	FA	DELAY:	MOV	R3,#250 ;延时子程序
0302H	7A	F8	DEL2:	MOV	R2,#248
0304H	00			NOP	
0305H	DA	FD	DEL1:	DJNZ	R2,DEL1
0307H	DB	F9		DJNZ	R3,DEL2
0309H	22			RET	;子程序返回
				END	

5.3 外部中断扩展

- 5.3.1 定时器扩展法
- 5.3.2 中断加查询扩展法
- 5.3.3 外部中断扩展的应用实例



5.3.1 定时器扩展法

思路：8051单片机内部设有两个16位可编程定时器/计数器。定时器/计数器工作于计数方式时，定时器0外部输入端(P3.4)或定时器1外部输入端(P3.5)作为计数脉冲输入端，在计数输入脉冲的下降沿到来时进行加1计数。当定时器/计数器计满后，若再有脉冲到来，定时器/计数器将产生溢出中断。

方法：利用上述特点，预先将定时器/计数器设置为满量程，通过外部输入端P3.4或P3.5输入下降沿脉冲，将立即向CPU发出中断请求，在满足CPU的中断响应条件的情况下，CPU将执行相应的中断服务程序。

应用：在外部中断源个数不太多并且定时器有空余的情况下，一般可采用定时器扩展法进行外部中断源的扩展。

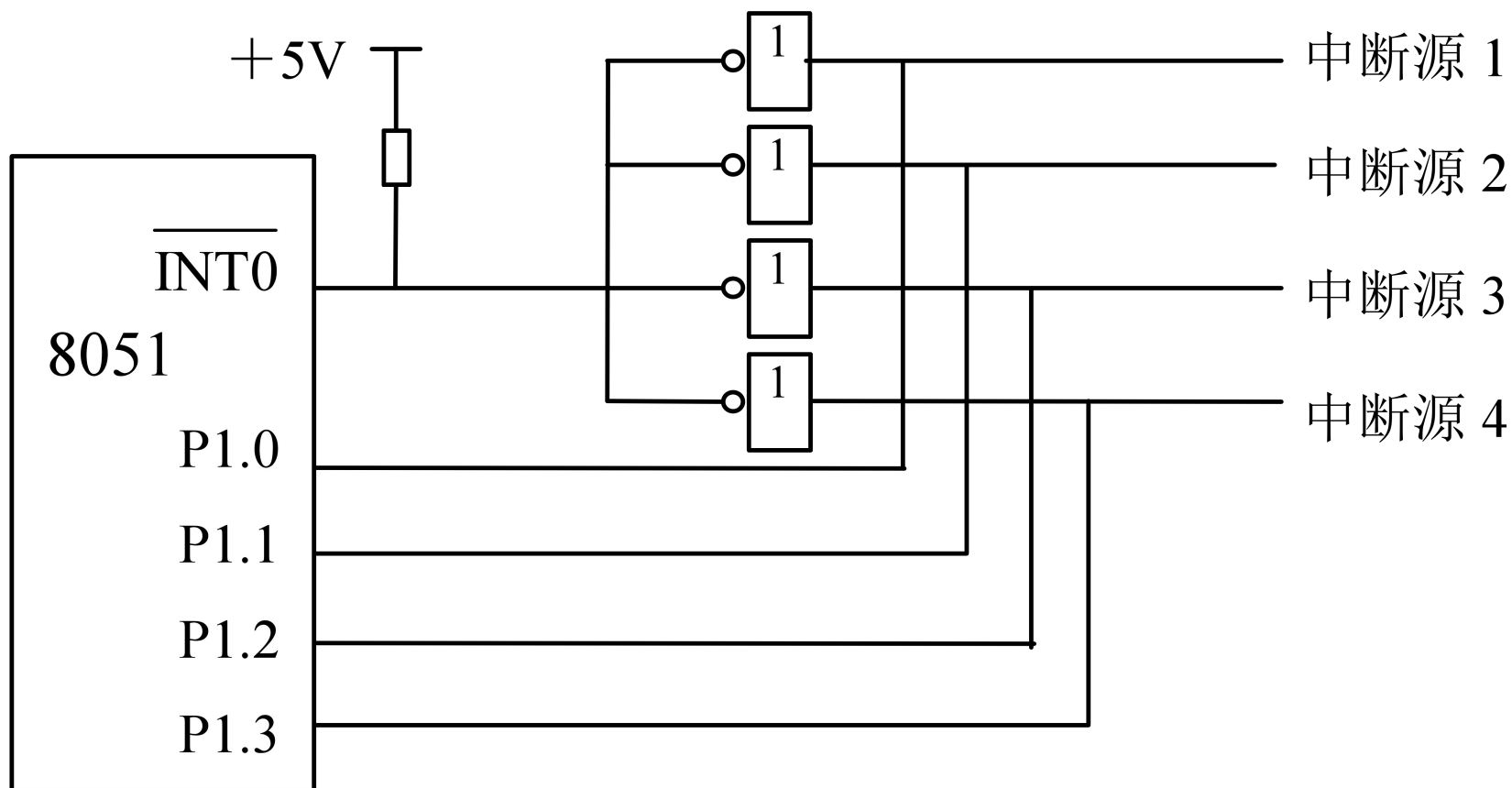
5.3.2 中断加查询扩展法

思路：对多个外部中断源，采用中断加查询相结合的方法响应中断。扩展电路原理如下图所示。多个外部中断源通过多个OC门电路组成线或电路后与P3.2(P3.3)相连，同时，每一个外部中断源将并行I/O口(如P1口)作为多个外部中断源的识别线。

方法：在多个外部中断源中若有一个或几个为高电平则输出为0，则P3.2(P3.3)为低电平，向CPU发出中断请求；CPU在执行中断服务程序时，先依次查询P1口的中断源输入状态，然后转入到相应的中断服务程序。

应用：中断加查询扩展法比较简单，但当外部中断源的个数较多时，因查询时间较长，不能满足实时控制的要求。

外部中断源扩展电路原理



5.3.3 外部中断扩展的应用实例

【例3】 用发光二极管模拟交通信号灯，用逻辑电平开关模拟控制开关，设计一个交通信号灯控制系统。设计要求如下：

A车道与**B**车道交叉组成十字路口，**A**是主道，**B**是支道；正常情况下，**A**、**B**两车道轮流放行。具体放行时间和要求如下：

- (1) A**车道放行**50s**，其中绿灯常亮**44s**，绿灯闪烁**3s**(用于警告)，黄灯常亮**3s**(用于警告)。
- (2) B**车道放行**30s**，其中绿灯常亮**24s**，绿灯闪烁**3s**(用于警告)，黄灯常亮**3s**(用于警告)。

在交通繁忙时，交通信号灯控制系统应有手控开关，可人为地改变信号灯的状态，以缓解交通拥挤状况。控制要求如下：

- (1)** 在**B**车道放行期间，若**A**车道有车而**B**车道无车，按下开关使**A**车道放行**15s**。
- (2)** 在**A**车道放行期间，若**B**车道有车而**A**车道无车，按下开关使**B**车道放行**15s**。
- (3)** 有紧急车辆通过时，按下开关使**A**、**B**车道均为红灯，禁行**15s**。

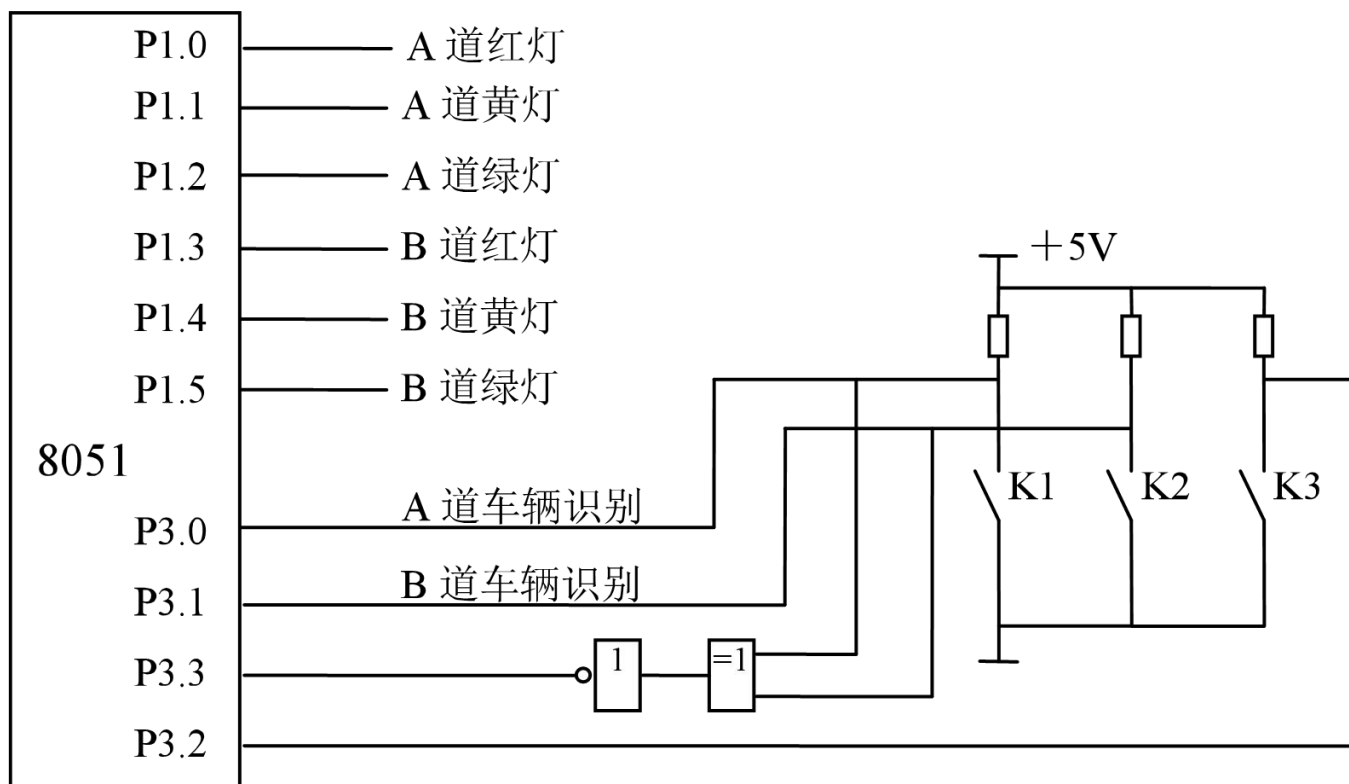
解：用发光二极管模拟交通信号灯，用**P1**口输出控制信号，输出低电平时，信号灯亮；反之，则灯灭。根据题意分析，在正常情况和交通繁忙时，**A、B**两车道的六只信号灯的控制状态有五种形式。

交通信号灯与控制状态的对应关系

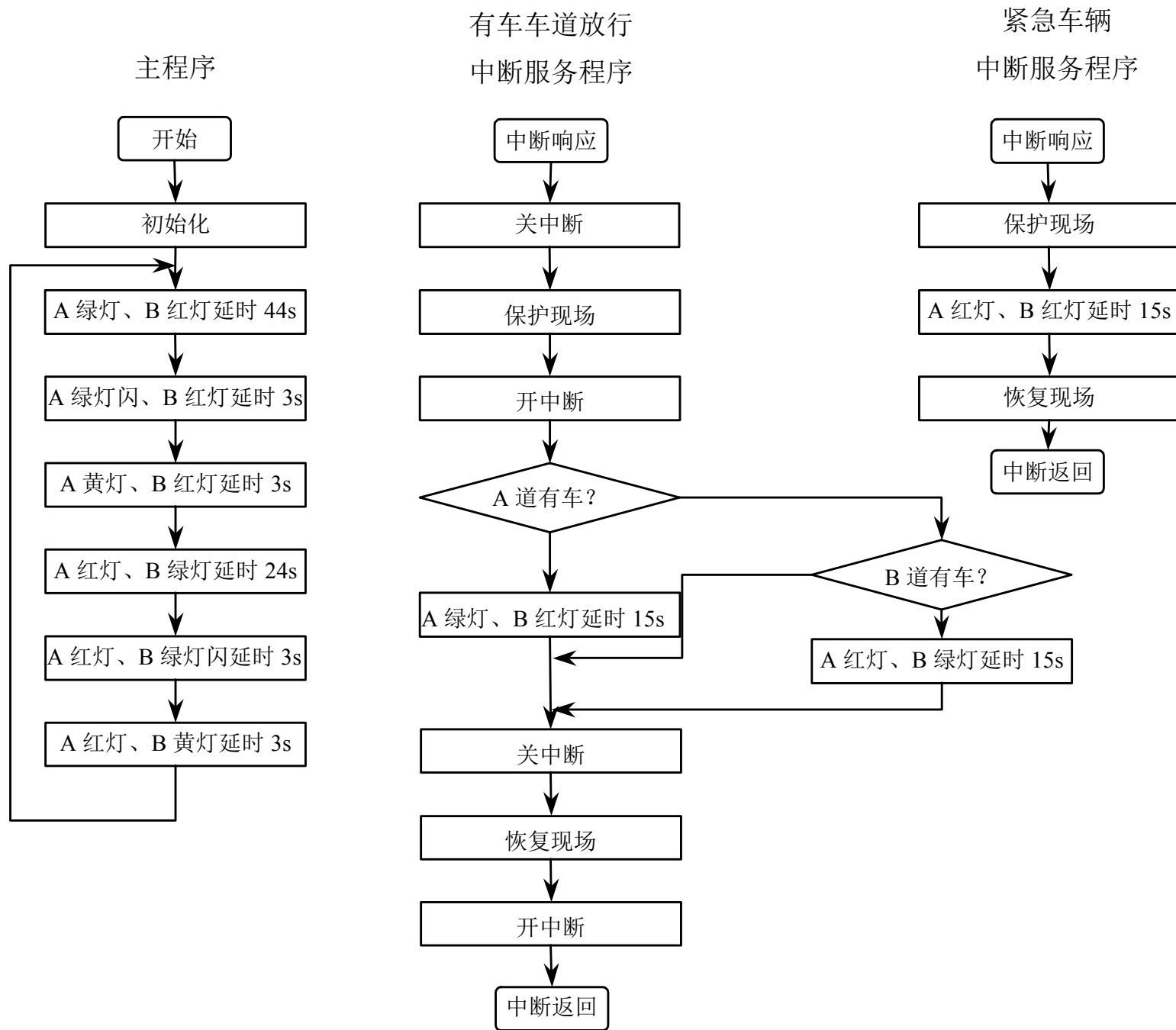
控制状态	P1口控制码	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
		未用	未用	B道绿灯	B道黄灯	B道红灯	A道绿灯	A道黄灯	A道红灯
A绿灯亮 B红灯亮	F3H	1	1	1	1	0	0	1	1
A黄灯亮 B红灯亮	F5H	1	1	1	1	0	1	0	1
A红灯亮 B绿灯亮	DEH	1	1	0	1	1	1	1	0
A红灯亮 B黄灯亮	EEH	1	1	1	0	1	1	1	0
A红灯亮 B红灯亮	F6H	1	1	1	1	0	1	1	0

用逻辑电平开关模拟控制开关，开关**K1**按下时，**B**车道放行；开关**K2**按下时，**A**车道放行；开关**K3**按下时，**A、B**车道禁行。开关**K1**和**K2**的控制信号经异或取反后，产生中断请求信号(低电平有效)，通过外部中断1向**CPU**发出中断请求；采用中断加查询扩展法，可以判断出要求放行的是**A**车道(按下开关**K1**)还是**B**车道(按下开关**K2**)。

控制系统电路示意图



交通信号灯控制系统程序流程图



主程序：

				ORG	0000H	
0000H	12	00	30	LJMP	MAIN	;转向主程序
				ORG	0003H	
0003H	12	02	00	LJMP	INT0	;转向紧急车辆中断服务程序
				ORG	0013H	
0013H	12	03	00	LJMP	INT1	;转向有车车道放行中断服务程序
				ORG	0030H	
0030H	D2	B8		MAIN:	SETB PX0	;置外部中断0为高优先级中断
0032H	75	88	00	MOV	TCON,#00H	;置外部中断0、1为电平触发
0035H	75	A8	85	MOV	IE,#85H	;开CPU中断、外部中断0、1中断
0038H	75	90	F3	LOOP:	MOV P1,#0F3H	;A道绿灯亮,B道红灯亮
003BH	79	58		MOV	R1,#88	;44s延时的循环次数
003DH	12	03	50	AP1:	LCALL DELAY	;调用0.5s延时子程序
0040H	D9	FB		DJNZ	R1,AP1	;44s不到,继续循环
0042H	79	06		MOV	R1,#6	;3s延时的循环次数
0044H	B2	92		AP2:	CPL P1.2	;A道绿灯闪烁
0046H	12	03	50	LCALL	DELAY	

主程序：

0049H	D9	F9		DJNZ	R1,AP2	;3s未到,继续循环
004BH	75	90	F5	MOV	P1,#0F5H	;A道黄灯亮,B道红灯亮
004EH	79	06		MOV	R1,#6	
0050H	12	03	50	AP3: LCALL	DELAY	
0053H	D9	FB		DJNZ	R1,AP3	;3s未到,继续循环
0055H	75	90	DE	MOV	P1,#0DEH	;A道红灯亮,B道绿灯亮
0058H	79	30		MOV	R1,#48	
005AH	12	03	50	BP1: LCALL	DELAY	
005DH	D9	FB		DJNZ	R1,BP1	;24s未到,继续循环
005FH	79	06		MOV	R1,#6	
0061H	B2	95		BP2: CPL	P1.5	;B道绿灯闪烁
0063H	12	03	50	LCALL	DELAY	
0066H	D9	F9		DJNZ	R1,BP2	;3s未到,继续循环
0068H	75	90	EE	MOV	P1,#0EEH	;A道红灯亮,B道黄灯亮
006BH	79	06		MOV	R1,#6	
006DH	12	03	50	BP3: LCALL	DELAY	
0070H	D9	FB		DJNZ	R1,BP3	;3s未到,继续循环
0072H	80	C4		SJMP	LOOP	;循环

外部中断0中断服务程序：

				ORG	0200H	
0200H	C0	90		INT0: PUSH	P1	;保护P1口数据
0202H	75	90	F6	MOV	P1,#0F6H	;A道红灯亮,B道红灯亮
0205H	7A	1E		MOV	R2,#30	;15s延时的循环次数
0207H	12	03	50	DELAY0: LCALL	DELAY	
020AH	DA	FB		DJNZ	R2,DELAY0	;15s未到,继续循环
020CH	D0	90		POP	P1	;恢复P1口数据
020EH	32			RETI		;返回主程序

外部中断1中断服务程序：

地址	机器码			程序	注释
				ORG 0300H	
0300H	C2	AF		INT1: CLR EA	;关中断
0302H	C0	90		PUSH P1	;保护现场
0304H	C0	04		PUSH 04H	
0306H	C0	05		PUSH 05H	
0308H	C0	06		PUSH 06H	
030AH	D2	AF		SETB EA	;开中断
030CH	30	B0	05	JNB P3.0,AP0	;A道无车,转向判断B道
030FH	75	90	F3	MOV P1,#0F3H	;A道绿灯亮,B道红灯亮
0312H	80	06		SJMP DEL1	;转向15s延时程序
0314H	30	B1	0A	AP0: JNB P3.1,EXIT	;B道无车,退出中断
0317H	75	90	DE	MOV P1,#0DEH	;A道红灯亮,B道绿灯亮
031AH	7B	1E		DEL1: MOV R3,#30	;15s延时的循环次数
031CH	12	03	50	NEXT: LCALL DELAY	
031FH	DB	FB		DJNZ R3,NEXT	;15s未到继续循环

外部中断1中断服务程序：

0321H	C2	AF	EXIT: CLR	EA	
0323H	D0	06	POP	06H	;恢复现场
0325H	D0	05	POP	05H	
0327H	D0	04	POP	04H	
0329H	D0	90	POP	P1	
032BH	D2	AF	SETB	EA	
032DH	32		RETI		;返回原程序

延时子程序:

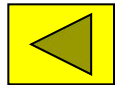
		ORG	0350H	
0350H	7C	0A	DELAY: MOV R4,#10	;0.5s延时子程序
0352H	7D	32	LP1: MOV R5,#50	
0354H	7E	F8	LP2: MOV R6,#248	
0356H	00		NOP	
0357H	DE	FE	LP3: DJNZ R6,LP3	
0359H	DD	F9	DJNZ R5,LP2	
035BH	DC	F5	DJNZ R4,LP1	
035DH	22		RET	
			END	

本章小结

- 单片机处理中断有中断请求、中断响应、中断处理和中断返回四个步骤。中断源有的来自单片机内部，称为内部中断源；也有的来自于单片机外部，称为外部中断源。中断系统的功能包括进行中断优先级排队、实现中断嵌套、自动响应中断和实现中断返回。中断的特点是可以提高**CPU**的工作效率、实现实时处理和故障处理。
- **8051**中断系统主要由定时器控制寄存器**TCON**、串行口控制寄存器**SCON**、中断允许寄存器**IE**、中断优先级寄存器**IP**和硬件查询电路等组成。
- 定时器控制寄存器**TCON**用于控制定时器的启动与停止，并保存**T0**、**T1**的溢出中断标志和外部中断、的中断标志。串行口控制寄存器**SCON**的低2位**TI**和**RI**用于保存串行口的接收中断和发送中断标志。中断允许寄存器**IE**用于控制**CPU**对中断的开放或屏蔽以及每个中断源是否允许中断。中断优先级寄存器**IP**用于设定各中断源的优先级别。

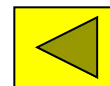
本章小结

- 中断处理过程包括中断响应、中断处理和中断返回三个阶段。中断响应是在满足**CPU**的中断响应条件之后，**CPU**对中断源中断请求的回答。由于设置了优先级，中断可实现两级中断嵌套。中断处理就是执行中断服务程序，包括保护现场、处理中断源的请求和恢复现场三部分内容。中断返回是指中断服务完成后，返回到原程序的断点，继续执行原来的程序；在返回前，要撤消中断请求，不同中断源中断请求的撤消方法不一样。
- 中断系统初始化的内容包括开放中断允许、确定中断源的优先级别和外部中断的触发方式。
- 扩展外部中断源的方法有定时器扩展法和中断加查询扩展法两种。定时器扩展法用于外部中断源个数不太多并且定时器有空余的场合。中断加查询扩展法用于外部中断源个数较多的场合，但因查询时间较长，在实时控制中要注意能否满足实时控制要求。



习 题

1. 什么是中断？中断系统的功能和特点有哪些？
2. 8051单片机的中断源有几个？自然优先级是如何排列的？
3. 外部中断触发方式有几种？它们的特点是什么？
4. 中断处理过程包括几个阶段？
5. 请简述中断响应的过程。
6. 外部中断请求撤销时要注意哪些事项？
7. 中断系统的初始化一般包括哪些内容？
8. 扩展外部中断源的方法有几种？



Q & A?

Thanks!

