

单片机中断系统

第5章 单片机的中断系统

中断的概念

生活中的中断实例：对看报过程中的电话铃响和水开了两个突发事件的处理

中断定义

计算机在运行当前程序的过程中，若遇紧急或突发事件，可以暂停当前程序的运行，转向处理该突发事件，处理完成后再从当前程序的间断处接着运行。

- 中断管理系统能够处理的突发事件称为中断源；
- 中断源向CPU提出的处理请求称为中断请求；
- 针对中断源和中断请求提供的服务函数称为中断服务函数（或中断函数）；
- 在中断服务过程中执行更高级别的中断服务称为中断嵌套；
- 具有中断嵌套功能的系统称为多级中断系统，反之称为单级中断系统。

一般函数与中断函数的区别：
• 调用一般函数过程是程序设计者事先安排的，而调用中断函数过程却是系统根据工作环境随机决定的；
• 主函数与调用函数之间具有主从关系，而主函数与中断函数之间则是平行关系；
• 一般函数调用是纯粹的软件处理过程，而中断函数调用却是需要软硬件配合才能完成的过程。

中断控制系统

中断系统结构

中断源

- 80C51共有5个中断源，即/INT0、/INT1、T0、T1、TX/RX；
- 引导汇编语言的中断服务程序调用的5个特殊ROM单元地址称为中断向量；
- 引导C51中断函数调用的是5个中断号0-4。

当中断源的突发事件出现时，单片机中某些特殊功能寄存器的特殊标志位将被硬件方式自动修改，CPU只要定期查看中断请求标志是否为1，便知道有无中断事件发生

0~3号中断源中各有1个中断请求标志，而4号中断源对应有两个中断请求标志，即IE0、IF0、TIE1、TF1、TI、RI

中断请求标志

外部中断源 (/INT0和/INT1)
• 通过P3.2和P3.3引脚输入，输入的信号有电平和脉冲两种形式，通过IT0和IT1逻辑开关切换；
• 若INT0或INT1为低电平，经非门到达IE0或IE1，产生硬件置1的中断请求；
• 若INT0或INT1为负脉冲，经施密特触发器到达IE0或IE1，产生硬件置1的中断请求。

内部中断源 (T0和T1)
• 在T0或T1中装入初值并闭合逻辑开关后，T0或T1中便会自动累加输入的脉冲信号；
• 充满溢出后，可向寄存器TF0或TF1“进位”，产生硬件置1的中断请求。

内部中断源 (TX/RX)
• 来自RXD引脚的一帧数据经过移位寄存器被送入“接收SBUF”单元后，接收控制器将使寄存器RI硬件置1的中断请求；
• 来自“发送SBUF”单元的一帧数据经过输出门发送出去后，发送控制器将使寄存器TI硬件置1的中断请求

中断的控制

用户对单片机中断系统的操作是通过4个控制寄存器实现的，即定时控制寄存器TCON、串行口控制寄存器SCON、中断优先级控制寄存器IP及中断允许控制寄存器IE。

TCON为定时控制寄存器
6个位寄存器与中断有关
INT0的中断请求标志位IE0、T0的中断请求标志位TF0、INT1的中断请求标志位IE1、T1的中断请求标志位TF1、INT0的中断触发方式选择位IT0和INT1的中断触发方式选择位IT1

SCON为串行口控制寄存器
只有两位与中断有关
接收中断请求标志RI和发送中断请求标志TI；
TI和RI虽然是两个中断请求标志位，但在SCON之后经或门电路合成为一个信息，统一接受中断管理

IE为中断允许控制寄存器
5个源允许和1个总允许
• 单片机复位后，IE的初值为0，因此默认是整体中断屏蔽；
• 每个源允许位ES、ET1、EX1、ET0、EX0都可以根据需要分别使其处于开放 (=1) 或屏蔽 (=0) 状态

IP寄存器为中断优先级控制寄存器
• 每个中断源的优先级PS、PT1、PX1、PT0、PX0都可被设置为高优先级中断 (=1) 或低优先级中断 (=0)；
• 当多个同级中断源同时提出中断请求时，CPU将依据自然优先级查询中断请求，自然优先级高的中断请求优先得到响应；
• 中断系统有两个优先级状态触发器，一个指出CPU是否正在执行高优先级中断服务程序，另一个指出CPU是否正在执行低优先级中断服务程序

中断控制过程

中断响应

- CPU响应中断的基本条件
- ① 有中断源发出中断请求；
 - ② 中断允许总控制位EA=1，即CPU开中断；
 - ③ 申请中断的中断源的中断允许位为1，即没有被屏蔽。

- CPU响应中断后，由硬件自动执行的操作
- ① 中断优先级查询，对后来的同级或低级中断请求不予响应；
 - ② 保护断点，即把程序计数器PC的内容压入堆栈保存；
 - ③ 清除可清除的中断请求标志位（见中断撤销）；
 - ④ 调用中断函数并开始运行；
 - ⑤ 返回断点继续运行。

响应时间

- 以外部中断为例，最短的响应时间为3个机器周期；
- 如果中断响应受阻，则需要更长的响应时间，最长响应时间为8个机器周期；
- 提高单片机的时钟频率（缩短机器周期），可减少中断响应时间；
- 如果中断响应受阻，则需要更长的响应时间，最长响应时间为8个机器周期；提高单片机的时钟频率（缩短机器周期），可减少中断响应时间；
- 提高单片机的时钟频率（缩短机器周期），可减少中断响应时间。

中断撤销

- 中断响应后，TCON和SCON中的中断请求标志位应及时清0，否则中断请求将仍然存在，并可能引起中断误响应。
- 定时/计数器中断——可由硬件自动对中断标志位TF0和TF1清0；
- 脉冲触发的外部中断请求——可由硬件自动对中断请求标志位IE0和IE1清0；
- 电平触发的外部中断请求——硬件不能自动对中断请求标志位IE0和IE1清0；
- 串行口中断——硬件不能自动对中断请求标志位TI和RI清0。

中断函数

C51中断函数的定义格式：void 函数名 (void) interrupt n [using m]

- C51中断函数的特点
- ① 中断函数是没有返回值的void型函数；
 - ② 中断函数是没有形参的无参函数；
 - ③ 中断函数采用系统默认的编译模式；
 - ④ 中断函数不是可重入的函数。

- C51中断函数注意事项
- ① 允许在中断函数中使用return语句，但不能使用带有表达式的return语句，如return(z)。
 - ② 可以通过使用全局变量，将变量值传入或传出中断函数。
 - ③ 中断函数只能被系统调用，不能被其他任意函数调用。
 - ④ 为提高中断响应的实时性，中断函数应尽量简短，并尽量使用简单变量类型及简单算术运算。

中断编程和应用实例

中断应用实例

实例5.2 中断扫描法行列式键盘

- 电路分析：在实例4.8的基础上增加了一个型号为4082的4与门集成元件。4个与门输入端分别与4条行线并联，与门输出端则与 /INT0 (P3.2) 引脚相连。无论按下哪个按键，与门的输出端都可形成的中断请求信号。
- 编程要点：将按键的扫描查询工作放在中断函数中进行，从而达到既快速响应按键动作，又提高CPU工作效率的目的。

实例5.3 中断方式的键控流水灯

- 电路分析：在实例4.3的基础上加装一个4输入与门电路（输入端与P0口并联），这样就能将按键闭合电平转化为/INT0中断信号
- 编程要点：利用外部中断监测按键的状态，一旦有按键动作发生，系统可立即更新标志位。主函数只负责流水灯循环运行，中断函数则负责按键检测与标志位刷新

实例5.4 中断嵌套演示

- 编程要点
- ① 3个数码管分别循环显示字符1~9，主函数采用无限计数显示，K0和K1的中断函数则采用单圈计数显示。
 - ② 由于K0的自然优先级高于K1，故需要将K1的中断级别设为高优先级，即PX1=1，PX0=0。
 - ③ 由于IE0的撤销过程发生在K0响应中断的瞬间，故在K0中断函数中将IE0值送P3.0输出可验证这一过程。

扩充外部中断源

- 扩展方式
1. 利用定时/计数器扩展外部中断源（将没用到的定时/计数器中断源作为外部中断源使用）
 2. 利用查询法扩展外部中断源（每条中断输入线通过“线或”关系连接多个外部中断源，利用输入端口线作为各个中断源的识别线。）
 3. 采用优先权解码芯片74LS148进行中断扩展

实例5.5 扩展8路外部中断源

- 电路分析：P2中包含了解码芯片74LS148的3位编码输出值7~0和5位悬空引脚的高电平
- 编程要点：P2进行取反后再屏蔽高5位的运算，即~P2&0x07。然后将变换结果作为分支跳转条件和数码管字模数组指针，实现模拟扩展外部中断函数的效果。