序，该指令将适当的状态弹回到处理器的控制和数据寄存器中，如果异常中断的是一个用户程序，就将状态恢复为用户模式(见 8.2.4 节)，然后将控制返回给被中断的程序。

8.1.2 异常的类别

异常可以分为四类：中断(interrupt)、陷阱(trap)、故障(fault)和终止(abort)。图 8-4 中的表对这些类别的属性做了小结。

类别	原因	异步 / 同步	返回行为
中断	来自 I/O 设备的信号	异步	总是返回到下一条指令
陷阱	有意的异常	同步	总是返回到下一条指令
故障	潜在可恢复的错误	同步	可能返回到当前指令
终止	不可恢复的错误	同步	不会返回

图 8-4　异常的类别。异步异常是由处理器外部的 I/O 设备中的事件产生的。同步异常是执行一条指令的直接产物

1. 中断

中断是异步发生的，是来自处理器外部的 I/O 设备的信号的结果。硬件中断不是由任何一条专门的指令造成的，从这个意义上来说它是异步的。硬件中断的异常处理程序常常称为中断处理程序(interrupt handler)。

图 8-5 概述了一个中断的处理。I/O 设备，例如网络适配器、磁盘控制器和定时器芯片，通过向处理器芯片上的一个引脚发信号，并将异常号放到系统总线上，来触发中断，这个异常号标识了引起中断的设备。

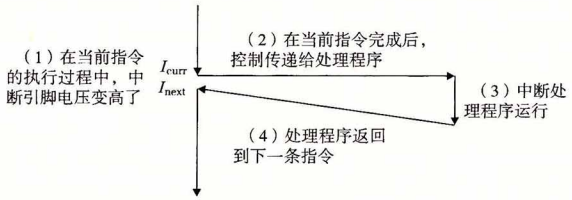

图 8-5　中断处理。中断处理程序将控制返回给应用程序控制流中的下一条指令

在当前指令完成执行之后，处理器注意到中断引脚的电压变高了，就从系统总线读取异常号，然后调用适当的中断处理程序。当处理程序返回时，它就将控制返回给下一条指令(也即如果没有发生中断，在控制流中会在当前指令之后的那条指令)。结果是程序继续执行，就好像没有发生过中断一样。

剩下的异常类型(陷阱、故障和终止)是同步发生的，是执行当前指令的结果。我们把这类指令叫做故障指令(faulting instruction)。

2. 陷阱和系统调用

陷阱是有意的异常，是执行一条指令的结果。就像中断处理程序一样，陷阱处理程序将控制返回到下一条指令。陷阱最重要的用途是在用户程序和内核之间提供一个像过程一样的接口，叫做系统调用。

用户程序经常需要向内核请求服务，比如读一个文件(read)、创建一个新的进程(fork)、加载一个新的程序(execve)，或者终止当前进程(exit)。为了允许对这些内核服务的受控的访问，处理器提供了一条特殊的“syscall n”指令，当用户程序想要请求