

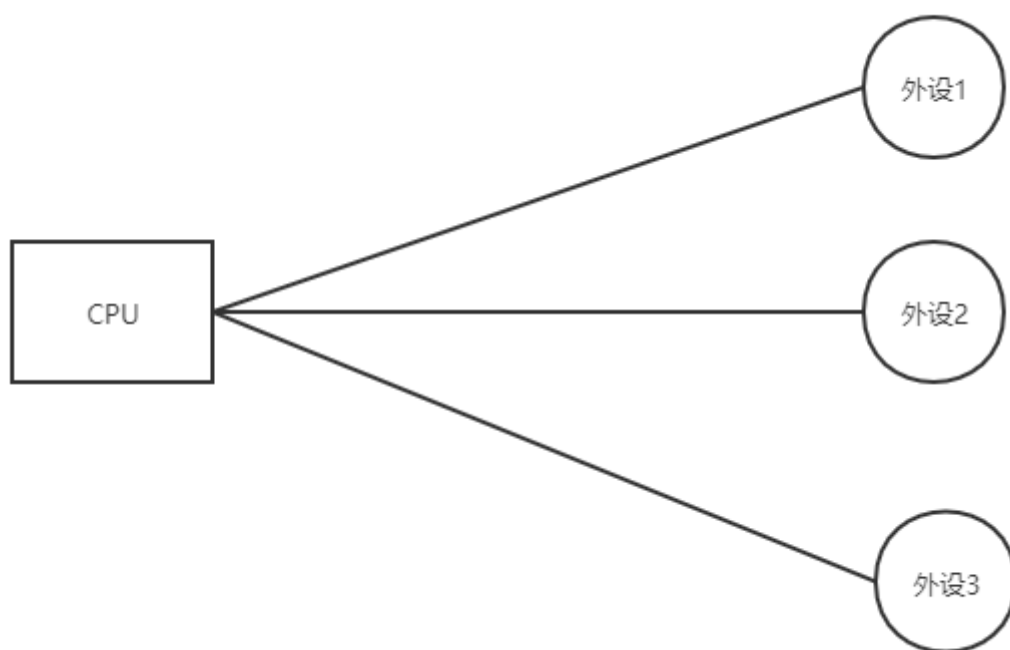
中断系统和程序中断方式

笔记本： 计算机组成原理

创建时间： 2021/6/19 8:46

更新时间： 2021/6/19 17:20

作者： 134exetj717



中断系统和程序中断方式

首先，我们可以把外设看作源头，请求方式看作连线，CPU看作对象。一整套流程，就是我们的中断。

我们可以从三大部分出发，阐述中断，分别是：基本概念、请求过程、应对方案。

故事篇章：

CPU是就像是一位老父亲，一直在努力工作。有一天，外设音箱需要向CPU发出中断请求，此时，音箱称作中断源。

音箱很懂事，因为他们在向CPU发出中断请求以前，都会自己先把准备工作做好，当请求成功以后就可以让CPU直接开始给他们工作，这样一来就能节省CPU的时间，减少了等待时间。

CPU通过子程序调用音箱和音箱发出中断请求的过程是不同的，比如，CPU调用音箱是程序员事先安排好的，但是音箱发出中断请求却是随机的，因为音箱此时可能出现了错误，而这个错误是我们意想不到的；同时，CPU在调用音箱时，主程序会接着调用其他和音箱相关的内容，但是音箱发出中断请求却是单独的，和主程序无关；另外，多个外设可以同时发出中断请求，但是CPU一次只能调用一个子程序。

外设发出中断请求以后，会随着线路送往CPU。此时，线路共分为三种，分别是独立请求线、公用请求线和二维结构。独立请求线的速度是最快的，因为它直接把请求送到CPU，但是当发出请求的外设数量增多时需要占用非常多线；公用请求线相对来说能节省很多线，但是存在一个问题，那就是不能快速找到中断源，此时小伙伴们可能有疑惑了，为什么要确定中断源呢？该内容就涉及到了中断隐指令的内容，其实CPU在从现执行程序切换到中断服务程序的过程中，硬件会隐式帮助我们完成切换，此时在切换的过程中，系统硬件需要识别中断源，其目的在于使CPU转入为该中断源专门设置的中断服务程序中。咱们继续说，如何找到中断源呢？系统设置了中断请

求触发器 (INTR)，专门用于存储中断源，以便我们能够快速找到中断源；二维结构是什么？二维结构就是不同优先级的中断请求处于不同的请求线上，该方法相当于折中了以上两种方法，一方面提高我们CPU接收请求的速度，另一方面便于我们寻找中断源。当外设的中断请求发送到CPU以后，我们如何采取措施准备接收呢？首先，我们需要判断是现行程序重要还是接收的中断请求重要，此时就引出了中断优先级的问题，现行程序和中断请求都含有优先级。既然每个请求都有对应的优先级，那么我们又该如何判断优先级然后选择中断与否呢？此时，我们可以通过软件判优和硬件判优的方式进行选择。那么软件判优和硬件判优的区别是什么呢？打个比方，前方十字路口有一辆公交车，一辆私家车，一辆救护车，车上分别有3人、2人、0人，我们可能会觉得公交车先走效率能够最高，但是规则告诉我们救护车的优先级最高，此时，我们的现场判断称为软件判优，规则称为硬件判优，为什么这么说呢？因为硬件判优是已经设定好的规则，由不同硬件组合实现特定功能接口，但是软件判优我们可以通过修改代码从而修改优先级，做到实时检测。显然，硬件判优的效率是最快的，但是灵活性也相对低下。当CPU已经接收到请求并且判优以后，就会进行中断响应，那么此时我们问自己，CPU中断响应的条件是什么呢？首先，CPU需要接收到请求；其次，CPU需要允许中断请求；最后，需要等待一条指令执行结束以后。响应以后，CPU就会自动帮助我们引出中断服务程序，该类行为就称为中断隐指令。CPU接收请求的时候，可能会有多个请求中断，那么嵌套的过程中，我们如何处理呢？答案是遵循栈的先进后出原则，先进的中断点，后恢复。就好像括号一样，一层套一层。

那么，如果孩子太调皮，老是要打扰老父亲怎么办呢？此时，CPU可以设置应对方案，分别有：允许中断、中断屏蔽、中断升级。第一个是允许和不允许，即01问题；第二个是根据优先级判断该中断是否允许，就好像是你学习的时候，其他小伙伴叫你也不搭理，但是家长叫的话就中断学习，去找父母；第三个是通过设置屏蔽字，给不同优先级设置不同的编码，就好像赫夫曼编码一样，从而决定不同优先级。但我们需要改变优先级时，只需要设定不同的编码的新优先级即可。

故事就到这里了。老父亲很辛苦，外设们应该要尽量少打扰老父亲哦~因此我们引进了DMA，直接内存访问~

- 基本概念
 - 程序中断的思想
 - 外设完成数据传送的准备工作以后，向CPU发出中断请求，CPU中止现行程序，转去执行中断服务程序，完成以后继续执行现行程序。
 - 中断处理过程
 - 从现行程序切换到中断服务程序，再从中断服务程序切换到现行程序
 - 中断系统
 - 计算机实现中断功能的软、硬件的总称
 - 程序中断与调用子程序的区别
 - 子程序的执行是程序员安排好的，而中断服务程序的执行是随机的
 - 子程序的执行受到主程序或上层子程序的控制，而中断服务程序一般与中断的现行程序毫无关系
 - 不存在同时调用多个子程序的情况，而有可能发生多个外设同时请求CPU为自己服务的情况
 - 中断的基本类型
 - 自愿中断和强迫中断
 - 自愿中断又称自中断，是程序中安排的有关指令
 - 强迫中断是随机产生的中断
 - 程序中断和简单中断
 - 程序中断需要占用CPU
 - 简单中断是外设和主存之间直接进行信息交换的方法，即DMA方式
 - 内中断和外中断
 - 内中断：由于CPU内部硬件或者软件原因引起的中断
 - 外中断：CPU以外的部件引起的中断
 - 向量中断和非向量中断
 - 向量中断是指中断服务程序的入口地址由中断事件自己提供
 - 非向量中断则不能提供
 - 单重中断和多重中断
 - 单重中断在CPU执行的过程中不能被打断

- 多重终端在执行某个中断服务程序的过程中，CPU可取相应级别更高的中断请求，又称为中断嵌套

- 请求过程

- 中断源
 - 中断请求的来源，如各个可以发出中断请求的外设
- 中断请求触发器 (INTR)
 - 负责记录中断源 = 具有存储功能的触发器
- 中断请求信号的传送
 - 独立请求线
 - 每个中断源单独设置中断请求线，将中断信号直接送往CPU。
 - 公共请求线
 - 多个中断源共有一条公共请求线。
 - 二维结构
 - 同一优先级的中断源，采用一根公共请求线；不同请求线上的中断源优先级不同。
- 中断优先级
 - 发出请求的外设具有优先级
 - 现行程序也有优先级
- 中断判优
 - 软件判优
 - 用程序来判别优先级
 - 硬件判优
 - 采取硬件判优电路实现判别
- CPU中断响应的条件
 - CPU接收到中断请求信号
 - CPU允许中断
 - 一条指令执行完毕
- 中断隐指令
 - 定义：CPU响应中断以后，经过某些操作，转去执行中断服务程序。这些操作由硬件直接实现的，把它成为中断隐指令。
 - 指令有：
 - 保存断点
 - 暂不允许中断
 - 引出中断服务程序
- 中断嵌套
 - 为了保护多个断点，采取栈的先进后出原则，如：先发生的中断请求的断点，先保护后恢复。

- 应对方案

- 允许和禁止中断
 - 中断允许触发器 (EINT) 被置0时，允许中断；置1时，禁止中断。
- 中断屏蔽
 - 有选择的封锁部分中断
- 中断升级
 - 作用是：改变中断优先级，将原则级别较低的中断源变成较高的级别，称为中断升级。

