

中断控制器的设计与实现

王凯祺

2018 年 1 月 2 日

总览

1. 选题背景
2. 技术路线
3. 设计过程
4. 效果评价

选题背景

为什么选择中断控制器，而不选其他？

选题背景

为什么选择中断控制器，而不选其他？

我们的课本仅用了 6 页（Section 4.9 Exceptions, Page 314 - 319）简单地介绍了系统中断，并没有把系统中断的过程、原理讲得十分透彻。

我希望自己设计一个中断控制器，填补课本上在系统中断的这块缺失。

技术路线

使用 Vivado 作为设计、仿真软件。

使用 Verilog 语言编写源程序。

如何测试中断控制器？

技术路线

使用 Vivado 作为设计、仿真软件。

使用 Verilog 语言编写源程序。

如何测试中断控制器？

使用之前实验课设计的多周期 CPU ， 配合我设计的中断指令， 对中断控制器进行测试。

测试通过后， 将中断控制器集成到 CPU ， 那么就变成了能处理中断的多周期 CPU 。

设计过程

中断处理流程

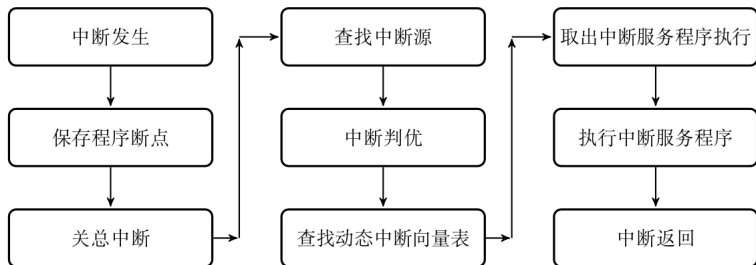


图 1: 中断处理流程示意图

设计过程

中断控制器设计

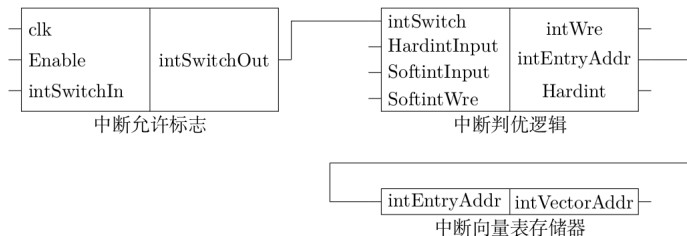


图 2: 中断控制器元器件及连线图

中断控制器是一个中断判优的组合电路，根据输入的中断源，选择优先级最高的响应，并把对应的中断入口地址接到中断向量表存储器中，由中断向量表存储器查询中断向量地址，接到 PC 输入端。

设计过程

中断判优

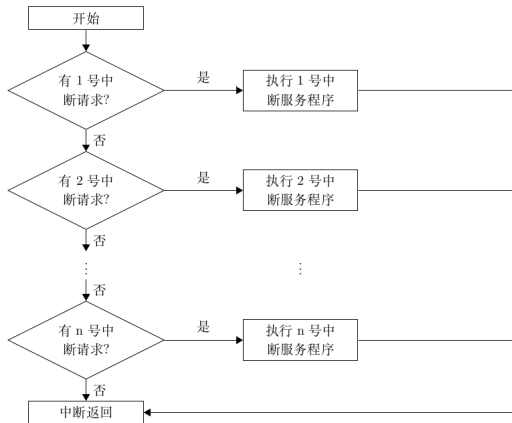


图 3: 软件查询法流程图

设计过程

堆栈设计

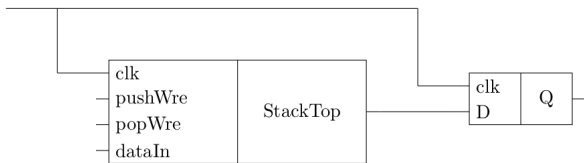


图 4: 硬件堆栈设计图

中断服务程序结束后，必须能正确地返回到被中断的断点处继续原来程序的执行。为了保存断点位置，同时为了支持中断嵌套，我们需要实现一个硬件堆栈。

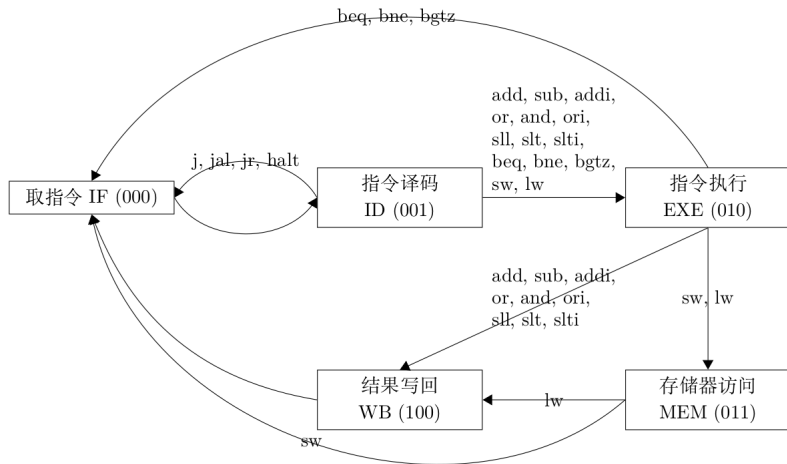
设计过程

重新设计 CPU 状态

为了测试中断控制器，我们还需要修改多周期 CPU，使该 CPU 可以处理中断。

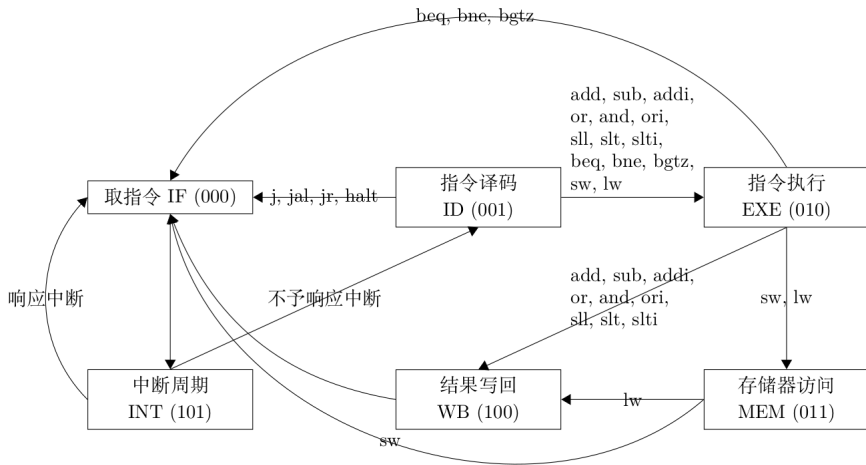
设计过程

重新设计 CPU 状态 (不含中断控制器的多周期 CPU 状态转移图)

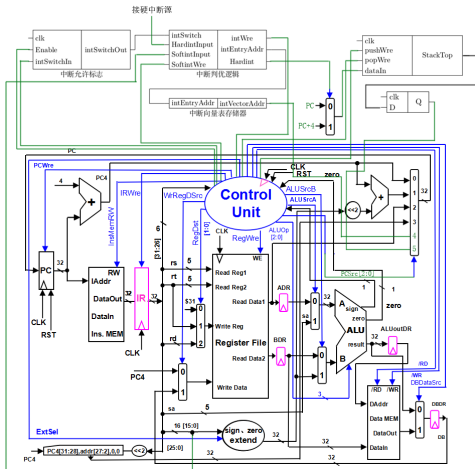


设计过程

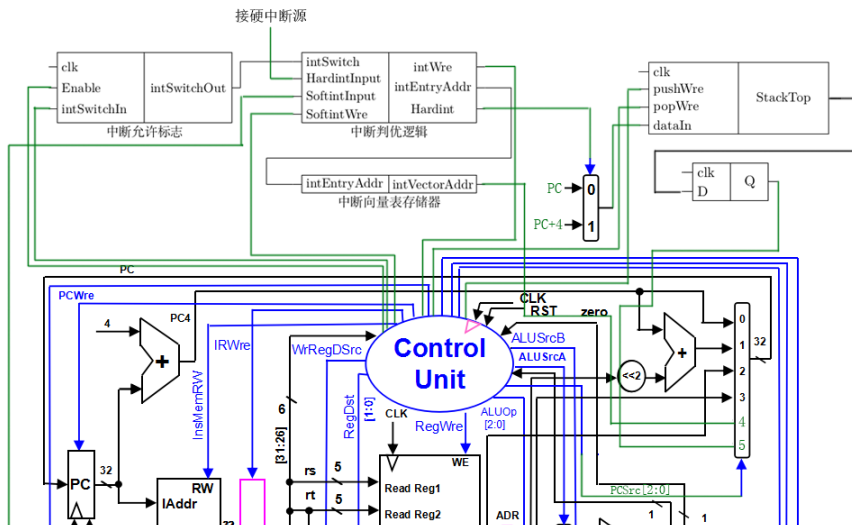
重新设计 CPU 状态 (含中断控制器的多周期 CPU 状态转移图)



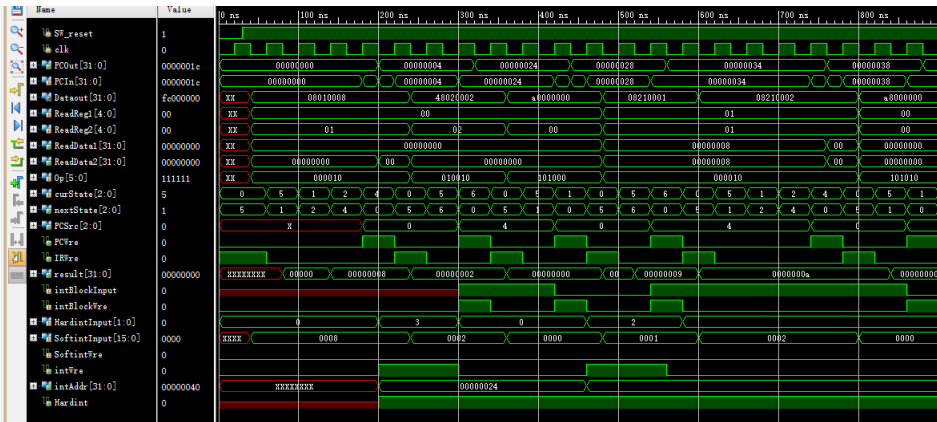
数据通路



数据通路



模拟仿真



效果评价

自主设计

能处理软硬中断

能处理中断嵌套

能处理多中断判优

填补了课本上在中断控制方面的空白

致谢

感谢李国桢教授提供课堂交流平台！

感谢何朝东老师提供多周期 CPU 数据通路图！

感谢同学们的聆听！