

软硬协同的用户态中断机制研究

综合论文训练开题报告

尤予阳

系内导师: 马洪兵 交叉导师: 陈渝

清华大学电子工程系

2021 年 1 月 6 日

① 课题背景

② 研究现状

③ 项目目标

④ 进度计划

① 课题背景

中断与特权级架构
RISC-V 指令集

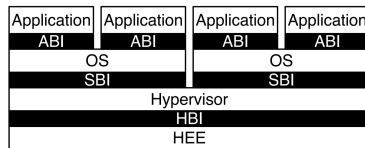
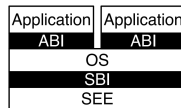
② 研究现状

③ 项目目标

④ 进度计划

中断与特权级架构

- 处理器通过划分特权级限制软件行为，提供安全保护和隔离
- 中断提示处理器某个特殊事件的产生，通常由较高特权的软件处理，如操作系统内核
- 内核可以通过软件方式为用户程序模拟中断，但效率较低



RISC-V 指令集

- 一个开放、免费、易实现的精简指令集
- 可以灵活地进行修改和扩展
- 中断设计与指令集架构紧密相关



① 课题背景

② 研究现状

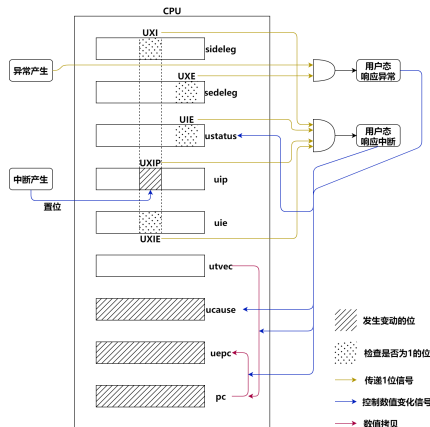
RISC-V 用户态中断扩展
x86 用户态中断扩展

③ 项目目标

④ 进度计划

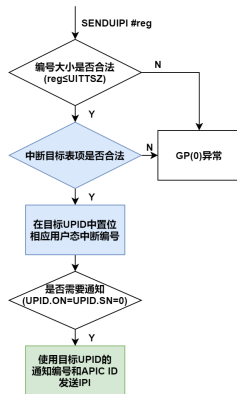
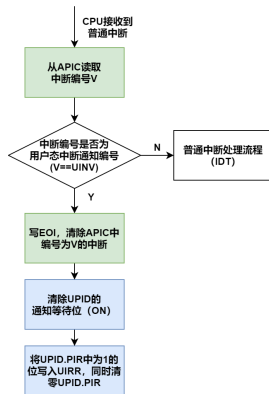
RISC-V 用户态中断扩展

- 也被称作“N 扩展”，仍处于草案阶段
- 中断控制寄存器和指令规范
- 未定义软件的跨核中断和外设的中断行为
- 已知有 shakti-c 处理器实现 N 扩展，但未见软件支持



x86 用户态中断扩展

- 在英特尔即将发布的 Sapphire Rapids 处理器中支持
- 已在 Linux 内核中实现了软件接口
- 目前只能用于线程间通信, 尚未实现外设到软件的中断



① 课题背景

② 研究现状

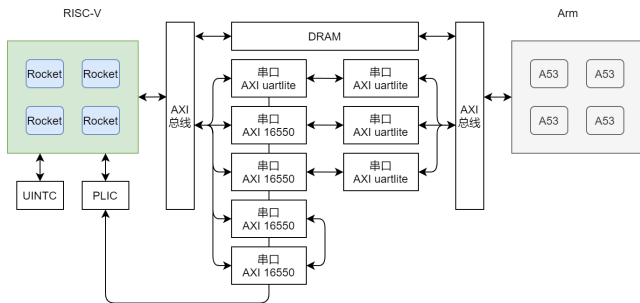
③ 项目目标

用户态中断机制实现
内核软件支持
应用程序优化

④ 进度计划

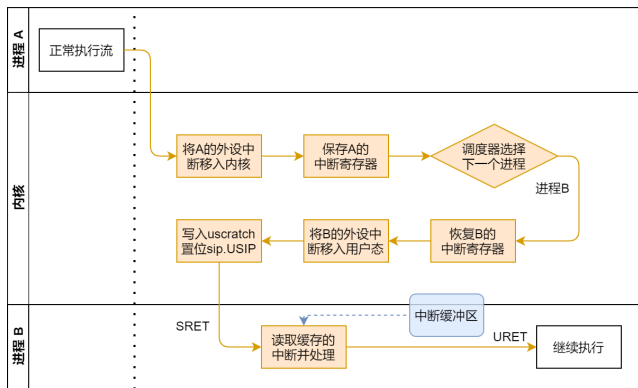
在 RISC-V 指令集上实现用户态中断机制

- 基础 N 扩展
- 软件间中断
- 外设到软件的中断
- 在模拟器与 FPGA 上实现



内核软件支持

- 在保持隔离性和并发性的同时，允许用户程序高效运用用户态中断机制
- 在 zCore 内核中实现支持

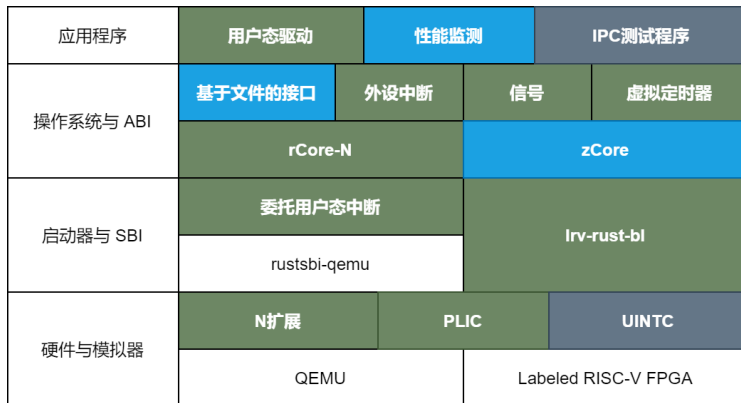


进入调度
抢占或主动让出

应用程序优化

- 在跨进程通信、用户态驱动等应用上获得一倍以上的性能提升
- 编写微测例，测试理想条件下的性能
- 针对现有的应用程序进行改造优化

项目架构



已实现的模块或功能



部分实现的模块



未来要完善的模块或功能

① 课题背景

② 研究现状

③ 项目目标

④ 进度计划

2021 年夏季学期

2021 年秋季学期

2022 年春季学期

2021 年夏季学期

- RISC-V 用户态中断调研及 N 扩展规范草案完善
- 模拟器和 FPGA 上的 N 扩展实现
- rCore-N 系统中的用户态中断支持
- 外设的用户态中断

The screenshot displays a terminal window with three main sections of output:

- 调试器输出窗口 (Debugger Output Window):** Shows the execution of a program named 'hell_world'. It includes details about the program's size (image size = 137936, payload num = 4228, kernel image size = 2358072) and the execution of 'user_trap_demo'. It also shows the execution of 'user_trap_demo' and the handling of a user-mode interrupt.
- 内核输出窗口 (Kernel Output Window):** Shows the interrupt handling process. It includes the execution of 'user_trap_demo' and the handling of a user-mode interrupt. It also shows the execution of 'user_trap_demo' and the handling of a user-mode interrupt.
- 用户态输出窗口 (User-mode Output Window):** Shows the execution of 'hell_world' and the handling of a user-mode interrupt. It includes the execution of 'user_trap_demo' and the handling of a user-mode interrupt.

Annotations on the screenshot include:

- 1. 用户程序申请中断并设置中断 (User program applies for interrupt and sets interrupt)
- 2. 内核通过直接置位 UPI 产生的时钟中断 (Kernel generates clock interrupt by directly setting UPI)
- 3. 发送信号 (Send signal)
- 4. 内核转发信号 (Kernel forwards signal)
- 5. 目标进程接收到信号并输出 (Target process receives signal and outputs)

2021 年秋季学期

- x86 用户态中断调研
- 串口驱动优化
- 多核下的用户态中断裸机测试程序
- 用户态软件中断控制器设计及模拟器实现

2022 年春季学期

- 一月、二月：完成用户中断控制器的 FPGA 实现
- 三月：在 zCore 内核中实现对用户态中断的支持
- 四月：编写测试程序，评估性能
- 五月：论文撰写

Thanks!