



中国科学技术大学
University of Science and Technology of China

计算机组成原理

Lab6 综合设计

计算机实验教学中心

2023/5/29

实验目标

- 理解计算机硬件系统的组成结构和工作原理
- 掌握软硬件综合系统的设计和调试方法

实验原理

方向 1 指令集扩展

“工欲善其事，必先利其器”

实验原理

方向 1：指令集扩展

□ RISC-V 指令集结构：开源、简单、模块化

□ 基础整数指令集 RV-32I

算逻指令：整数之间的算数与逻辑运算

分支指令：条件分支、无条件跳转

访存指令：读取存储器、写入存储器

系统指令：状态寄存器、中断、存储一致性等（不要求实现）

□ 扩展指令集：在基础指令集之外增加更多功能支持

M 扩展：整数乘法运算、整数除法运算

F 扩展：单精度浮点数运算与存储

A 扩展：原子操作指令

C 扩展：压缩指令（用于减小内存消耗）

实验原理

方向 1：指令集扩展

□ 指令扩展的内容包括：

1. 实现 RV-32I 所有非系统指令。具体包括：

必做部分 10 条：add、addi、auipc、lui、lw、sw、beq、blt、jal、jalr [Lab5 已实现]

算数与逻辑指令 13 条：sll、slli、srl、srli、sra、srai、sub、xor、xori、or、ori、and、andi

分支与条件指令 8 条：bne、bge、bltu、bgeu、slt、slti、slti、sltiu

访存指令 6 条：lb、lh、lbu、lhu、sb、sh

2. 实现 RV-32M 乘除法指令。具体包括：

mul、mulh、mulhu、mulhsu、div、divu、rem、remu

3. 实现 RV-32F 部分单精度浮点指令。具体包括：

fadd.s、fsub.s、fcvt.s.w、fcvt.w.s

实验要求

方向 1：指令集扩展

- 需要自行设计并修改流水线数据通路，支持指令的正确执行。其中乘除法指令需要设计相应的乘法器、除法器模块并接入流水线，浮点指令需要支持浮点寄存器与浮点运算。
- 所实现的指令均需要在流水线中正确处理其可能的冒险。
- 注意： **不允许**使用乘法运算符、除法运算符实现乘除法指令。

实验原理

方向 2

异常处理：算术溢出

“谋无主则困，事无备则废”

实验原理

方向 2: 异常处理_算术溢出

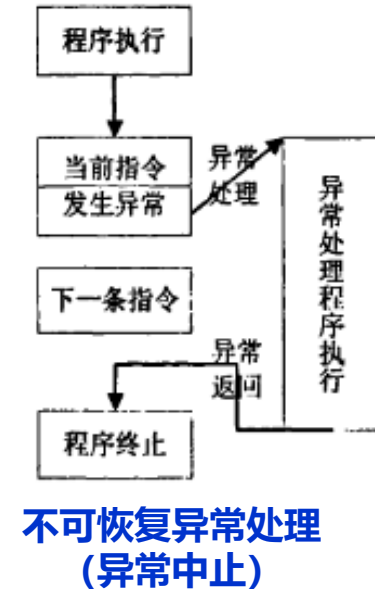
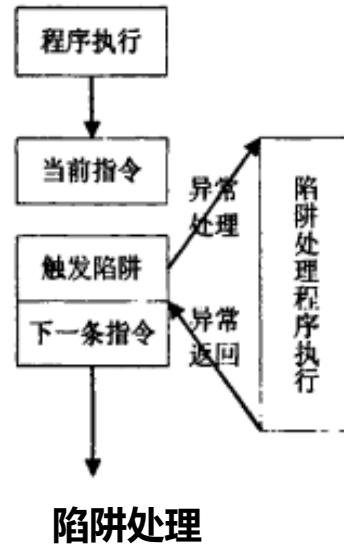
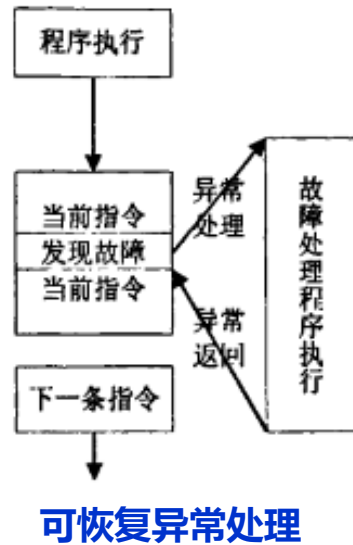
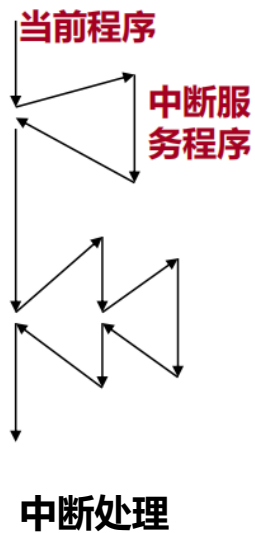
- 异常和中断，有时也可统称为异常，是处理器中十分常见的情况。
- 异常
 - ✓ 发生在CPU内部的非预期事件，如：未定义指令、系统调用、访存地址不对齐、ALU 结果溢出、除法除数为 0 等
- 中断
 - ✓ 来自外部的非预期事件，如：键鼠操作、U盘插入等

注意：Lab5 的流水线 CPU 并没有支持中断异常的结构。

实验原理

方向 2: 异常处理_算术溢出

□ 非预期的突发事件发生时，需要改变程序的控制流



实验要求

方向 2：异常处理_算术溢出

□ 异常处理的内容包括：

1. 实现异常状态保存

□ 程序控制与状态寄存器 CSR (Control and Status Register)

- ✓ CSR 是独立的寄存器单元，用于记录出现中断或异常时处理器内部的情况。
- ✓ 本次实验需要实现 mepc (编号 0x341, 可自己指定其编号)，用于存储异常指令的 PC。
- ✓ 你也可以根据需要自行设计其他的 CSR 寄存器。

□ 发生异常时，流水线自动将有关信息写入 CSR 寄存器（硬件实现）

实验要求

方向 2: 异常处理_算术溢出

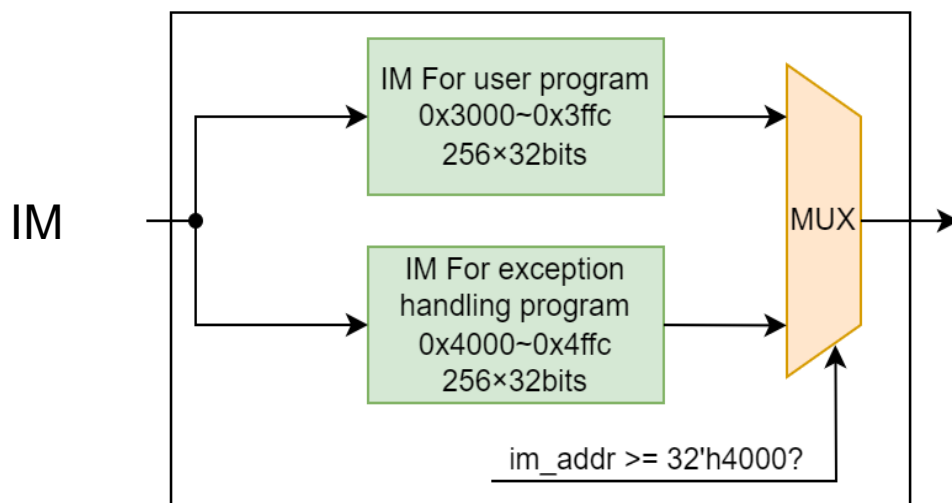
□ 异常处理的内容包括:

1. 实现异常状态保存

2. 实现异常处理程序与用户程序的切换

□ 根据处理器内部的异常情况, 执行异常处理程序

□ 异常处理程序的起始地址为 0x4000, 存储在独立的存储器中



实验要求

方向 2: 异常处理_算术溢出

□ 异常处理的内容包括:

1. 实现异常状态保存
2. 实现异常处理程序与用户程序的切换
3. 编写异常处理程序

□ RISC-V 汇编程序实现

- 需要在数码管上显示 0x8880F888 以告知用户此时出现算数溢出。这里需要使用 Lab4 中介绍的外设输出交互流程: 异常处理程序轮询等待用户按下按钮, 从而确保数码管显示的内容被正确接收。MMIO 规则参考 Lab4 实验手册。
- 处理完成后, 回到正常的程序执行, 即 $mepc + 4$ 对应的指令。需要使用 `csrrs` 等指令 (注意处理其带来的冒险) 将 $mepc+4$ 加载到 `x1` 寄存器, 并用 `ret (jalr, x0, 0(x1))` 指令作为异常处理程序结束的标志。

实验原理

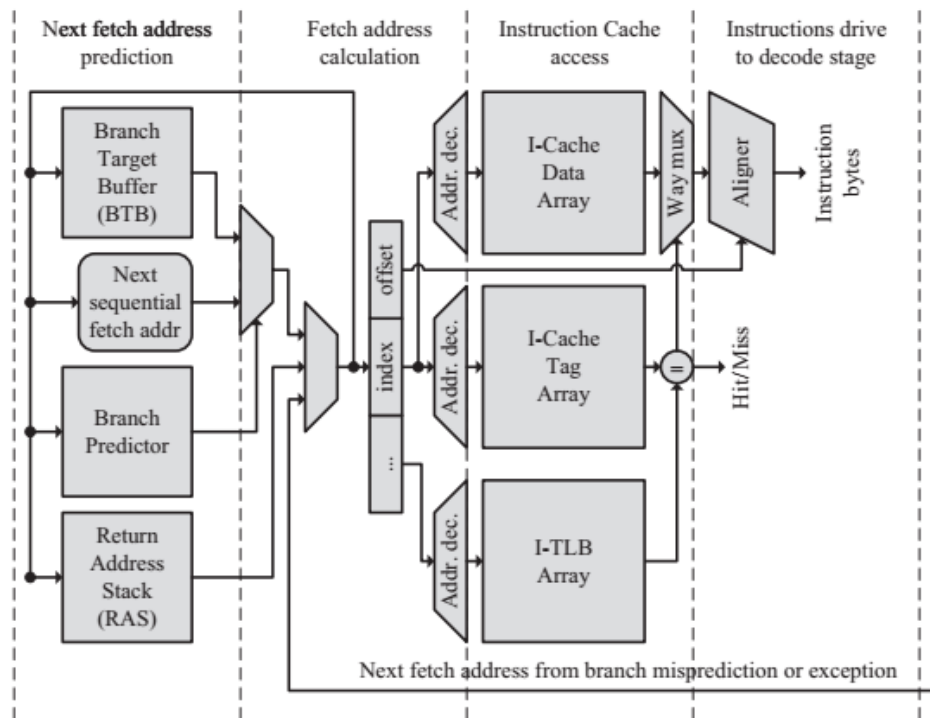
方向 3 分支预测

“纸上得来终觉浅，绝知此事要躬行”

实验原理

方向 3：分支预测

- Lab5 的流水线 CPU 在遇到需要跳转的分支指令时，需要冲刷 IF、ID 段的两条指令，对处理器的性能产生很大影响。
- 分支预测技术是指处理器在 IF 阶段预测分支的跳转与否以及跳转目标地址，根据预测结果来实现不间断的指令流，从而让处理器的 CPI 接近理想情况中的 1。

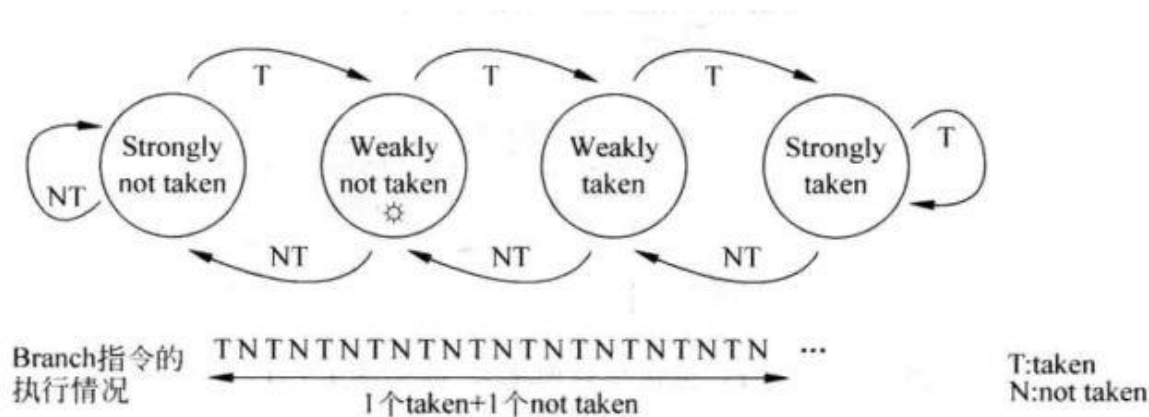


带有分支预测和 Cache 的流水线示意图（局部）

实验原理

方向 3：分支预测

- 分支预测包括静态预测、基于两位饱和计数器的预测、基于局部历史的预测、基于全局历史的分支预测等。这些预测方法可以处理不同的跳转情况，且各有所长。



- 本方向需要实现基于局部历史的分支预测、基于全局历史的分支预测，并正确处理二者之间的竞争关系。

实验要求

方向 3：分支预测

□ 分支预测的内容包括：

1. 基于局部历史的分支预测

设计 BHT 与 PHT 的结构，并接入流水线 CPU。正确修改预测失败后流水线与分支预测器的相关内容。

2. 基于全局历史的分支预测以及竞争处理

设计全局预测器的结构，并设计算法处理全局预测与局部预测结果不一致时的竞争关系，以及对于全局预测器的实时修正。

注：可以通过提供的矩阵乘法程序检验分支预测的效果。

实验原理

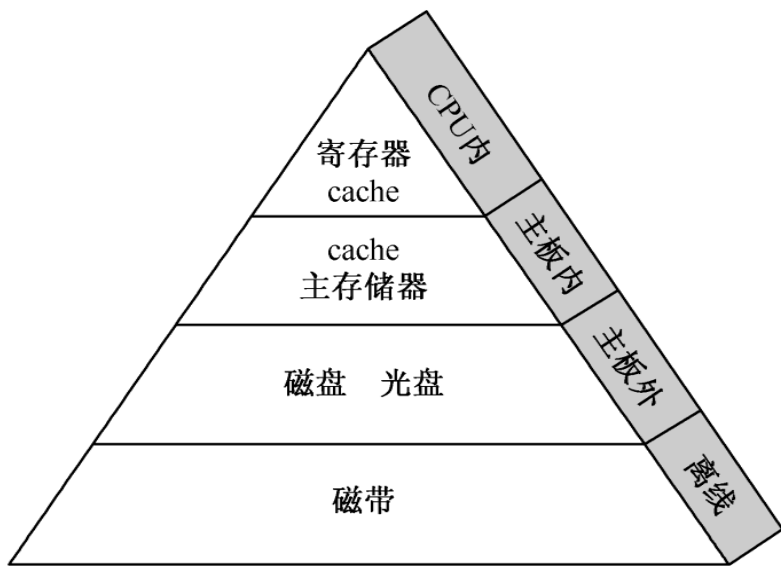
方向 4 存储系统与高速缓存

“成也萧何，败也萧何”

实验原理

方向 4：存储系统与高速缓存

- 真实的存储系统往往是多级存储体系。计算机系统对存储器的容量、速度和价格这三个基本性能指标都有一定的要求，只有多级存储体系才能较好地平衡三者之间的矛盾。



速度	容量	价格/位
快	小	高
↓	↓	↓
慢	大	低

实验原理

方向 4：存储系统与高速缓存

- 真实的存储系统往往是多级存储体系。计算机系统对存储器的容量、速度和价格这三个基本性能指标都有一定的要求，只有多级存储体系才能较好地平衡三者之间的矛盾。
- 引入 Cache 的原因：
 - 解决 CPU 和主存之间速度不匹配的问题
 - 据统计，CPU 的速度平均每年改进 60%，而组成主存的 DRAM 的速度平均每年只改进 7%
 - 避免 CPU 与 I/O 设备的访存冲突
 - 一旦主存需与 I/O 设备交换信息时，CPU 可以访问 Cache 获取信息
- Cache 与主存储器之间采用 AXI4 总线协议进行交互。

实验要求

方向 4：存储系统与高速缓存

□ 高速缓存的内容包括：

1. 指令缓存 Icache

补全框架代码，实现基于 LRU 替换算法的 2 路组相连指令高速缓存，并能够通过参数来调节 Cache 的行宽度和组数。

2. 数据缓存 Dcache

补全框架代码，实现基于 LRU 替换算法的 2 路组相连数据高速缓存，并能够通过参数来调节 Cache 的行宽度和组数。

注：可以通过提供的随机访问测试检测 ICache 和 DCache 的正确性。

3. 流水线适配

将 ICache 和 DCache 接入自己的流水线，正确处理流水线停顿和冒险，并成功运行快速排序程序。

实验说明

实验内容自选说明

自选项目 (任选其一)	实验方向	具体项目
自选1 (必做+任1选做)	RV32-I 指令子集扩展 [必做]	支持所有非系统 RV32-I 指令(共 27 条)
	RV32-MF 指令子集扩展 [选做1]	流水乘法器
		移位除法器
		实现浮点数基本运算
	异常处理：算术溢出 [选做2]	实现异常状态保存
		编写异常处理程序
		实现用户程序与异常处理程序的切换
	分支预测 [选做3]	2bits 感知机局部历史分支预测
		2bits 感知机全局历史分支预测
自选2	存储系统与高速缓存 [独立项目]	根据框架实现 LRU 替换算法的 Icache
		根据框架实现 LRU 替换算法的写回写分配Dcache
		将实现的 Cache 接入流水线中

实验说明

实验评分说明

- 综合实验所选项目必须在**同一个流水线 CPU 中**完成。你需要谨慎处理不同项目之间可能的冲突与依赖关系
- 选择**存储系统与高速缓存方向**的同学**可以不做 RV32-I 指令子集扩展**，其他方向的同学均需要先完成 RV32-I 指令子集扩展
- 每个实验方向具体赋分见“评分细则”，**每人得分上限为10分**
- 由于汇编程序与显示器显示的多样性，这里不设计统一的评分标准，请与助教联系以确定最终的评分规则



中国科学技术大学
University of Science and Technology of China

The End