Argo：面向教学的乱序双发射处理器

王华强
计算技术研究所 中国科学院大学
2021.6

中国科学院大学
University of Chinese Academy of Sciences

在简单顺序核的基础上实现乱序核

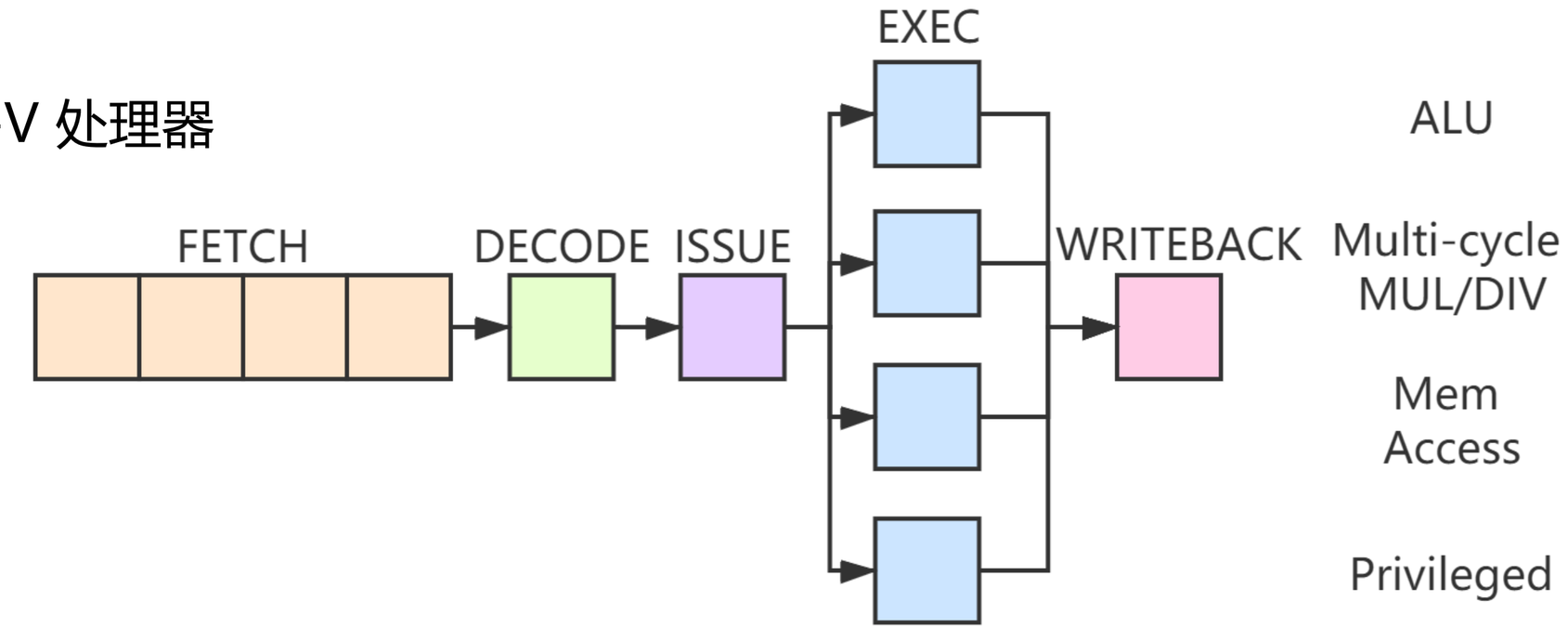
- 五级流水线顺序核是常见的本科生课程设计要求
- 在此基础上实现一个简单乱序核，需要做那些更改？

取指前端	执行后端	访存子系统
提高前端取指宽度	寄存器重命名	乱序访存支持
提高分支预测宽度	乱序指令调度	推测 store 支持
	分支预测恢复	

在简单顺序核的基础上实现乱序核

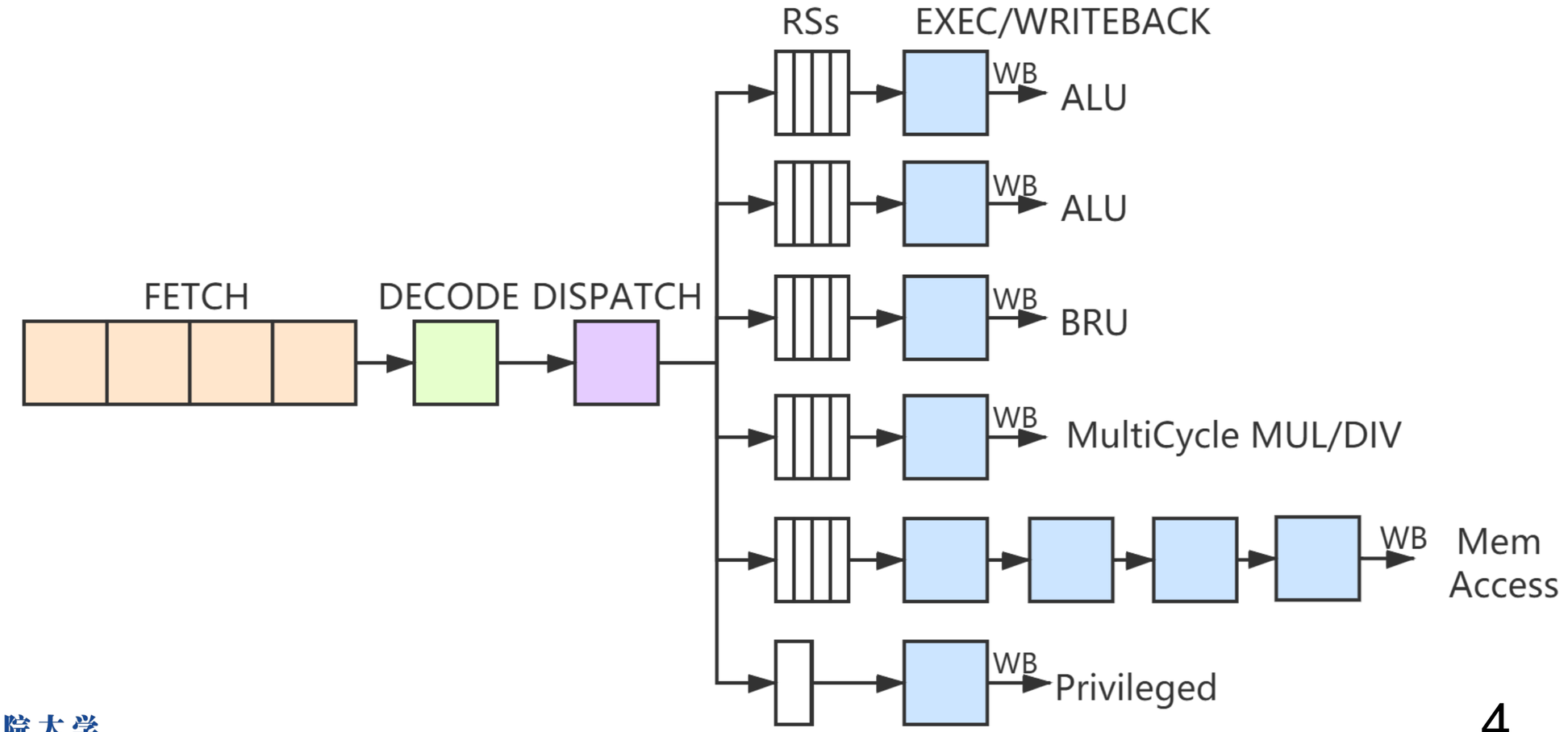
- 基于开源 RISC-V 处理器 NutShell 进行修改
- 以尽可能少的改动将其修改为支持乱序超标量执行的架构

- NutShell
  - 单发射 RISC-V 处理器



在简单顺序核的基础上实现乱序核

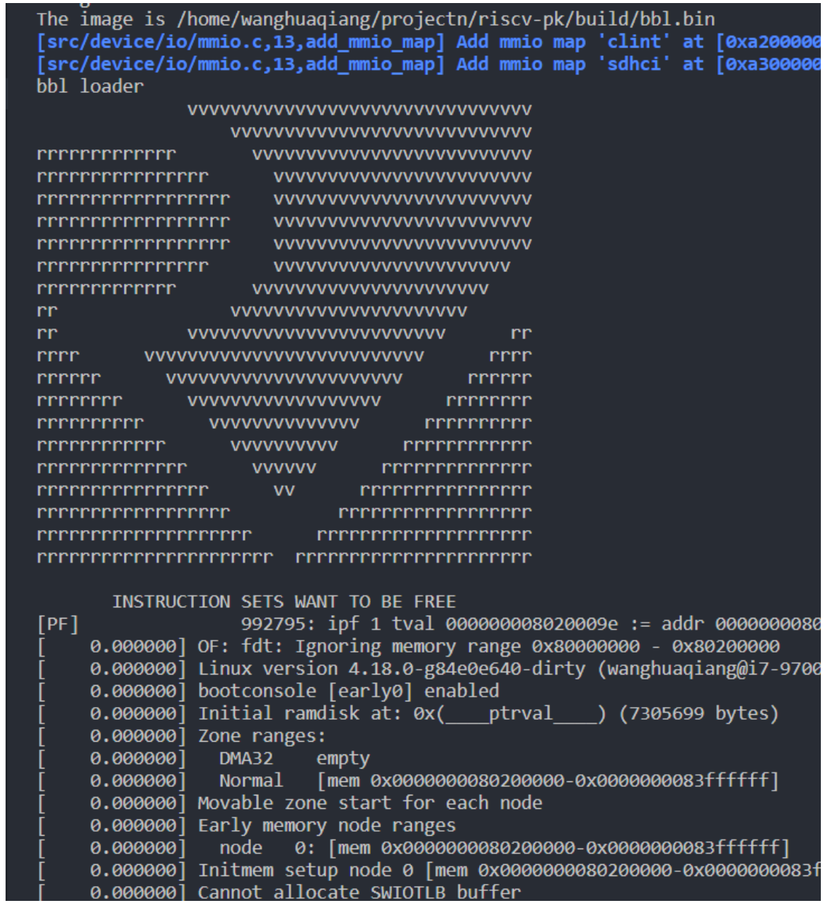
- 相比于顺序的 NutShell 架构，新增了 **2898** 行代码
- 实现了基于隐式重命名，分布式保留站，发射前读的乱序处理器 Argo



特性

- 继承自 NutShell 的特性
  - RV64IMAC指令集，支持M/S/U特权级
  - 两位饱和计数器分支预测，包含返回地址栈
  - 指令 / 数据 L1 cache，L2 cache
  - 支持Sv39分页机制
  - Chisel3语言开发
  - 支持启动Linux内核

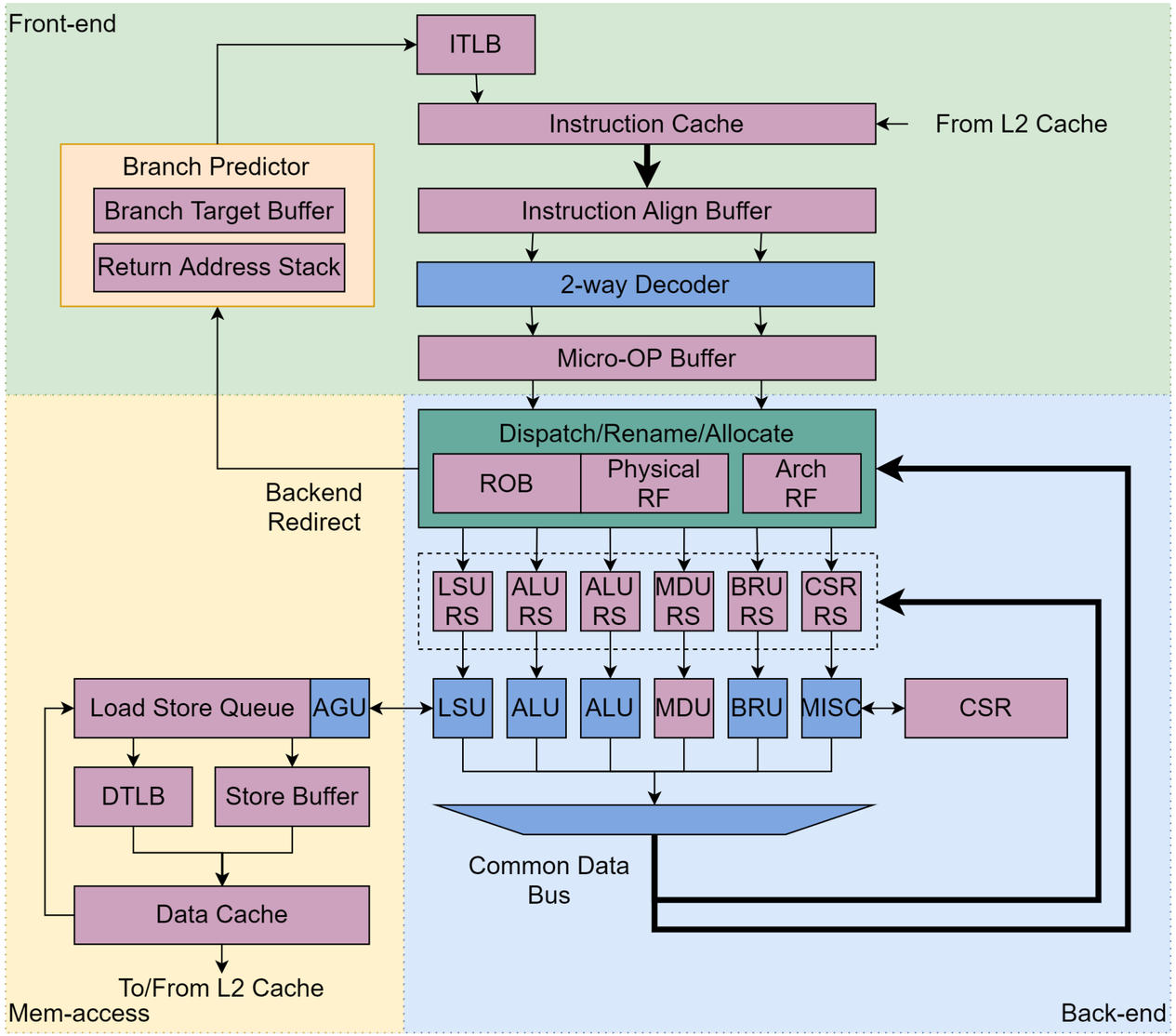
- 简单乱序超标量处理器核
- CoreMark IPC 1.090



使用 Argo 在仿真中启动 Linux

整体架构

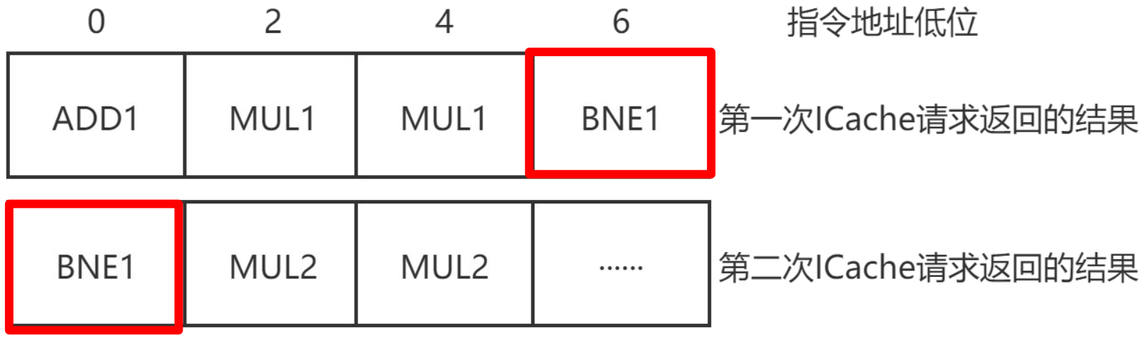
- 前端取指部分
  - 指令译码宽度为2
- 后端乱序执行部分
  - Rename-in-ROB 设计
  - 最多支持同时分派和提交两条指令
  - 分布式保留站
  - 发射前读寄存器堆
- 访存部分
  - 一条公用的访存流水线
  - 硬件TLB填充



前端取指变动

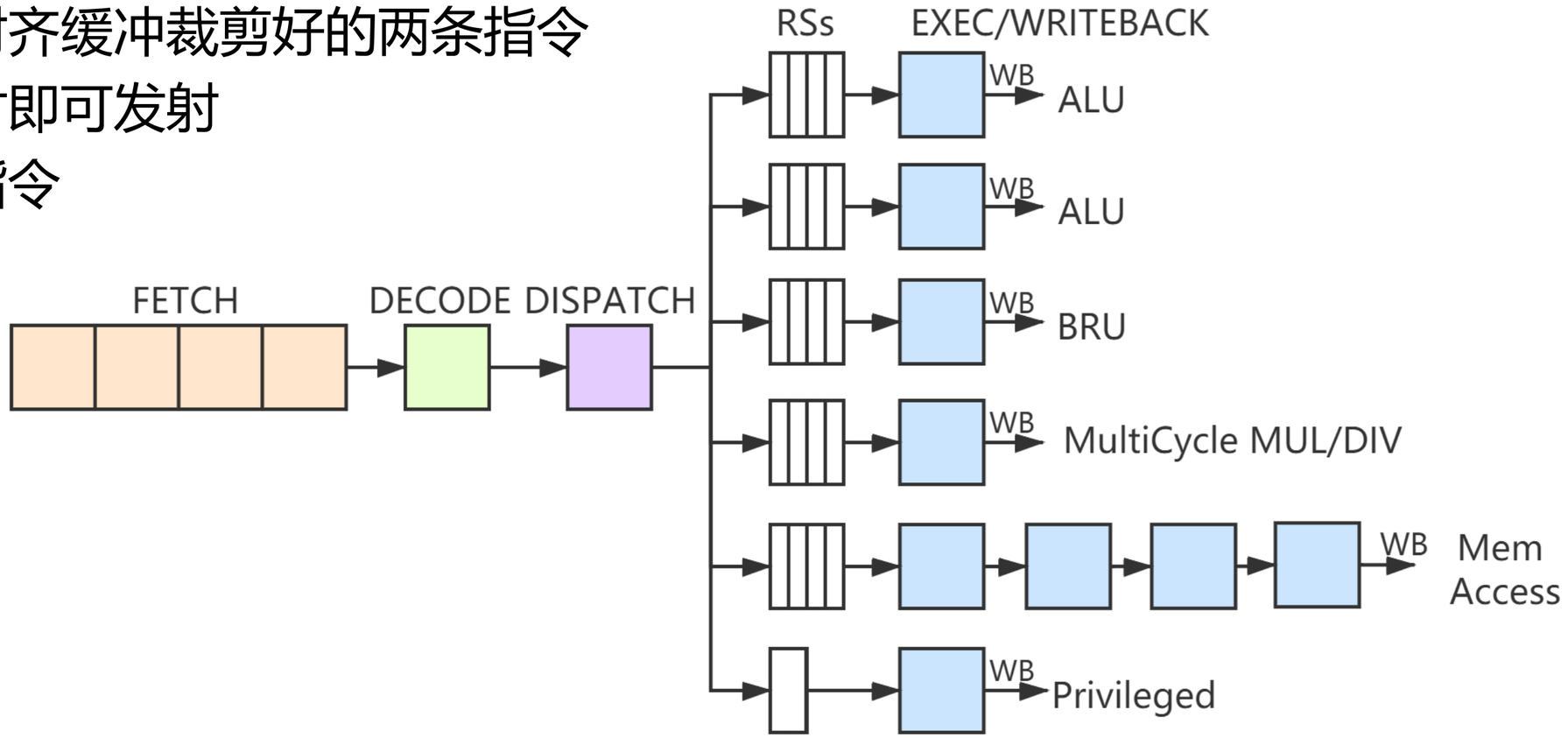
- 前端整体宽度为2
- ICache 每周期返回 64bit 的指令
- 指令对齐缓冲
- 处理非对齐指令的跳转

```
0: ADD(Compressed) a0, a0, a1
2: MUL a2, a2, a3
6: BNE a1, a2, 24
10: MUL a2, a2, a3
12: ......
```



分派、发射与功能部件

- 分派逻辑
  - 每次检查指令对齐缓冲裁剪好的两条指令
  - 功能部件充足时即可发射
  - 否则阻塞后续指令
- 功能部件配置
  - ALU x 2
  - BRU x 1
  - MDU x 1
  - LSU x 1
  - MISC



分派、发射与功能部件

- 发射逻辑
  - 保留站使用**掩码**判断指令顺序
  - 最老的指令优先发射
  - 发射前读寄存器

ID	掩码 (mask)	有效位 (valid)	进入保留站的顺序
0	0000	0	X
1	0000	1	1
2	0010	1	2
3	0110	1	3

使用掩码追踪指令优先级示例
三条有效指令进入保留站的顺序是1->2->3

- 访存保留站保存更多信息
  - 会与 ROB，访存队列通信来获取访存指令的执行状态
  - 用于控制访存指令的执行顺序

乱序执行与检查点实现

- 寄存器重命名
  - 为简化实现，使用**隐式**寄存器重命名
  - 需要在提交时将 ROB 中的数据搬运到体系结构寄存器中
  - 默认配置下使用 32 项的 ROB

- 分支误预测恢复
  - 提供4个**检查点**进行分支误预测恢复
  - 检查点记录 ROB 中的数据是否有效
  - 跳转指令分派时更新检查点
  - BRU发现跳转指令预测错误时从检查点恢复状态

访存子系统

- **一条**访存流水线
- 限制了访存指令的执行顺序
  - Load 可以乱序执行
  - Load **不会**越过之前的 store 执行
  - 只需要处理 store to load forward

- 默认复用 NutShell 的 VIVT blocking Cache
  - 也支持 non-blocking Cache

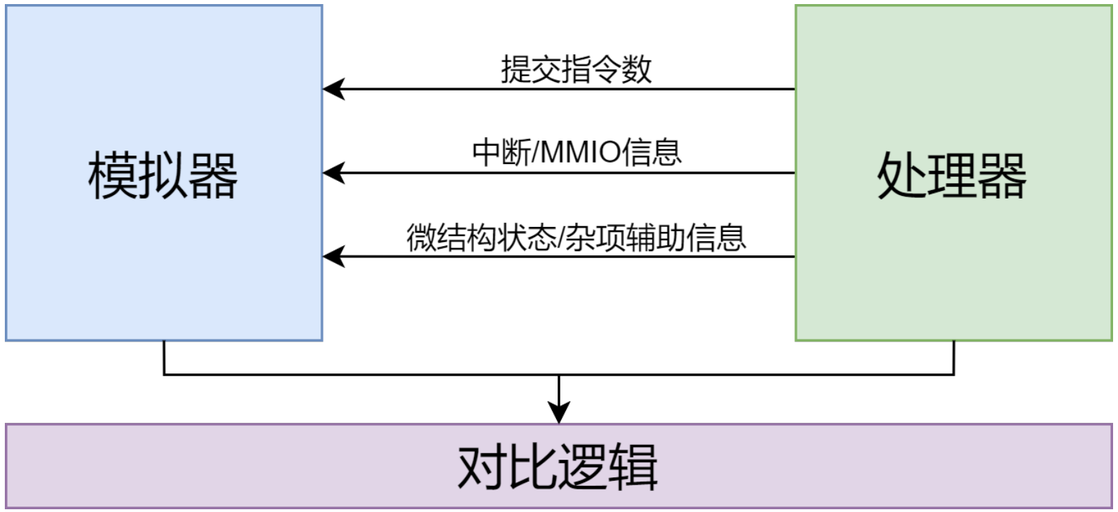
写回与提交

- 可配置使用2个或3个写回端口
  - 分别对应 6R2W / 6R3W 寄存器堆

- ROB 每次提交至多两条指令
- 在写回时更新体系结构寄存器堆

验证框架

- 使用与 NutShell 相同的基础设施
  - 与模拟器在线比对



- 提高每次提交的对比宽度到2
- 在触发中断、例外时，每周期仍然最多只能提交一条指令

小结

- 此设计提供了一个较为完整的乱序超标量处理器设计
- 可以作为入门乱序处理器设计的参考

感谢
Q&A

王华强
2021.6