北京开源芯片研究院
BEIJING INSTITUTE OF OPEN SOURCE CHIP

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

中国开放指令生态（RISC-V）联盟
China RISC-V Alliance

香山开源处理器昆明湖架构的设计演进

唐浩晋[1] 王凯帆[1] 胡轩[1] 张林隽[2] 裴晓坤[1] 徐泽凡[4] 李昕[12] 李衍君[3]

[1]中国科学院计算技术研究所 [2]北京开源芯片研究院

[3]中国科学院大学 [4]中国科学技术大学

2024年8月22日

香山：开源高性能处理器

- **迄今性能最高的开源处理器系列**
  - 开源的 RTL 设计和文档
  - 开源的开发工具和平台
- **旨在解决开源处理器芯片生态的两大挑战**

XIANGSHAN

托管在 GitHub 上

> 4.5K stars

> 630 forks

高性能

- 算力需求正在快速增长
- 由于较高的微架构设计、优化和设计难度，很少有开源处理器芯片能够做到较高的性能

高可配置性

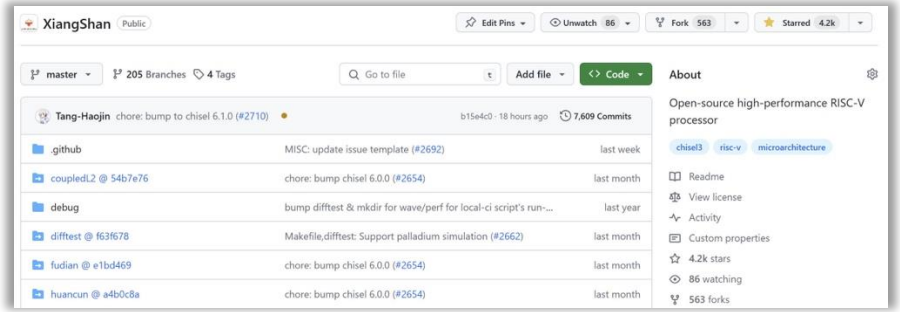
- 工业界需要定制化需求和快速开发迭代
- 高可配置性让领域特定架构的快速生成和海量的定制化需求成为可能

➢ **“处理器界的 Linux”**

香山：开源高性能处理器

- **第一代架构：雁栖湖**
  - 2020/6：RTL 设计的第一个提交
  - 2021/7：**28nm 流片，1.3GHz**
  - 性能：**SPEC CPU2006 7.01@1GHz, DDR4-1600**
- **第二代架构：南湖**
  - 2021/5：开始设计探索和架构设计
  - 2023/4：GDSII 交付
  - **14-nm 流片，SPEC CPU2006 20@2GHz**
- **第三代架构：昆明湖**
  - 新的指令集扩展（虚拟化、向量等）
  - 优化设计的流水线部件
  - CHI-CoupledL2 缓存
  - 性能：**SPEC CPU2006 45@3GHz**

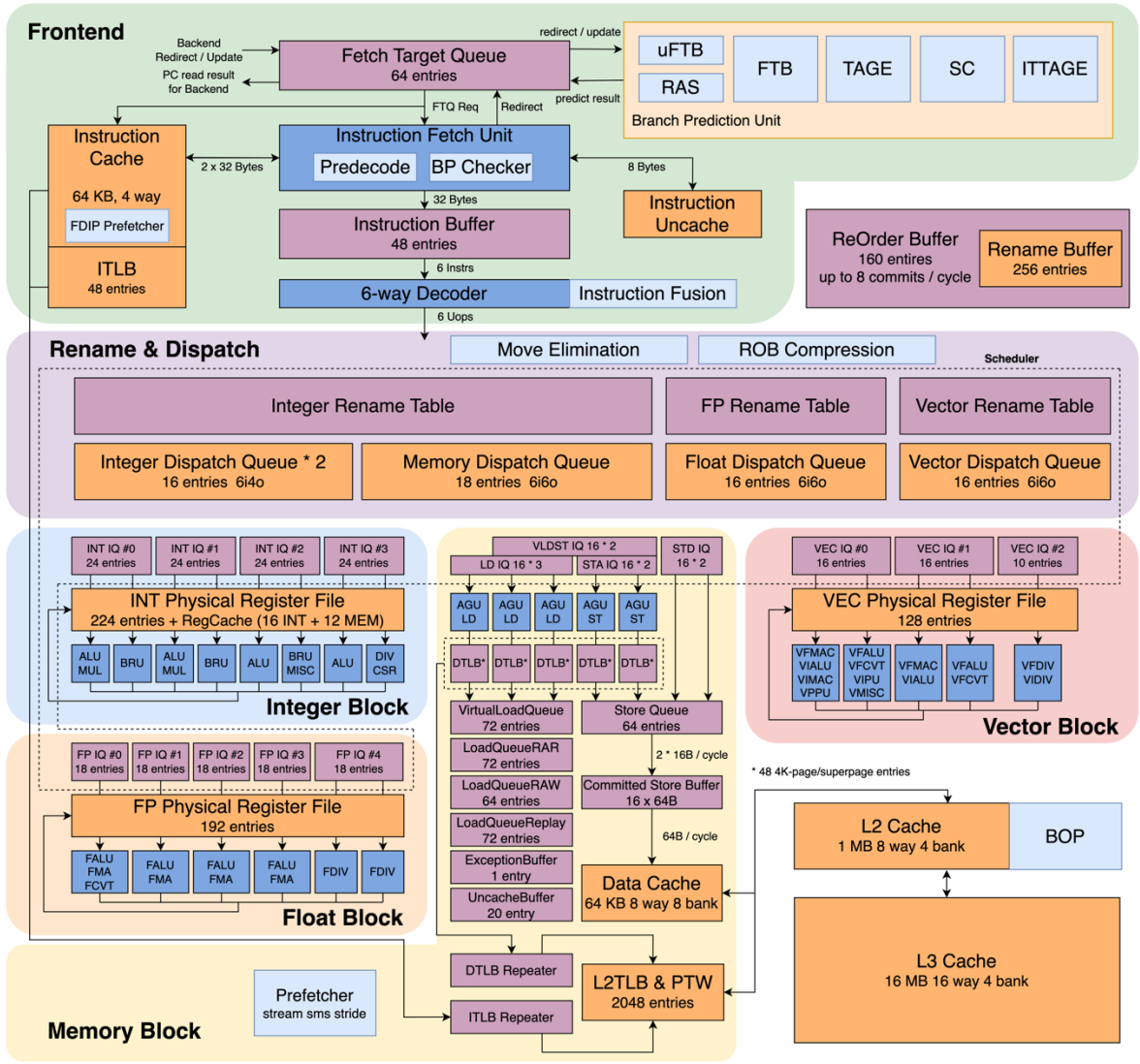
北京香山



>4.2K stars, >500 forks on GitHub

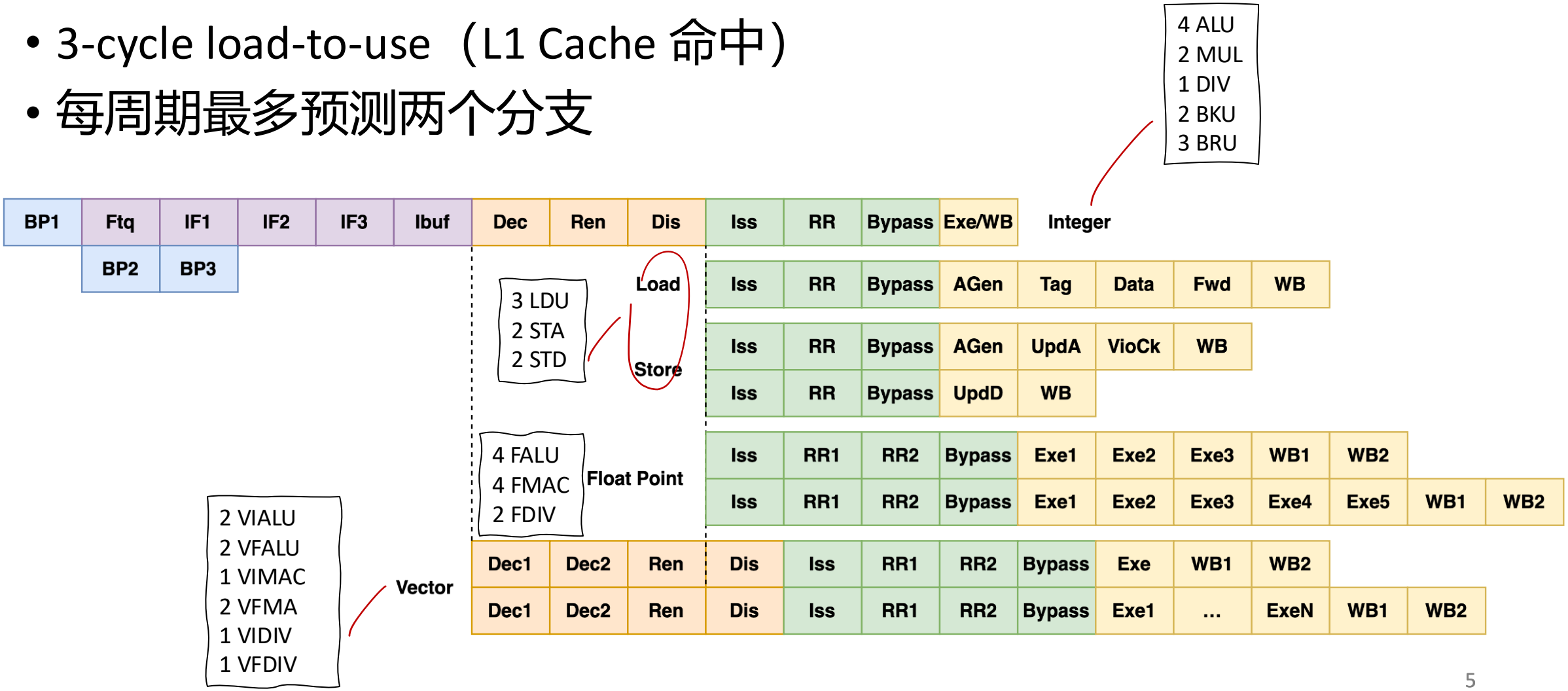
昆明湖架构总览

- 6 宽度重命名
- 多级覆盖分支预测
- 224 INT + 192 FP + 128 VEC 物理寄存器
- 160 ROB + 256 RAB（RenAme Buffer）
- 64 KB ICache / DCache
- 1 MB L2 Cache
- FDIP 指令预取
- stream / sms / stride / bop / tp 预取器
- 48-entry ITLB / DTLB + 2048-entry L2TLB



昆明湖流水线概览

- 13 级流水深度
- 3-cycle load-to-use（L1 Cache 命中）
- 每周期最多预测两个分支

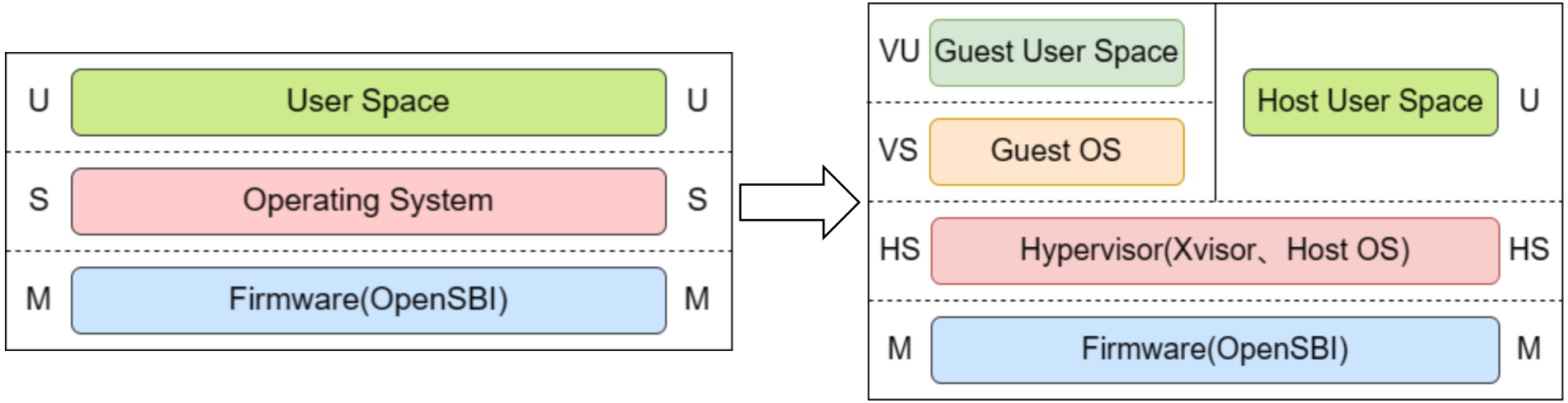


昆明湖微架构改进

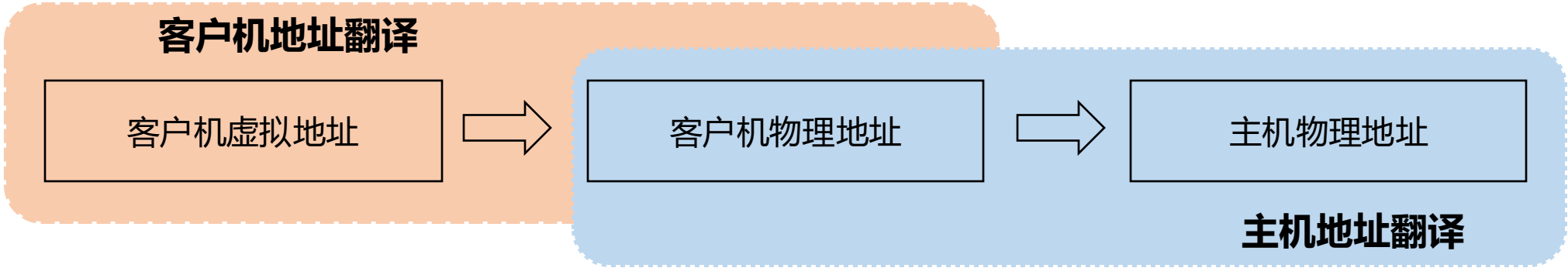
- 新支持指令集扩展
- 发射后读寄存器堆的新后端
- 更低功耗的 ICache
- 时序更优的 LoadQueue 和 LSU 设计
- 支持 CHI 总线的 Coupled L2 缓存

RISC-V 虚拟化 H 扩展

- CPU虚拟化
  - 特权级拓展
  - CSR拓展
  - 指令拓展
  - Trap拓展

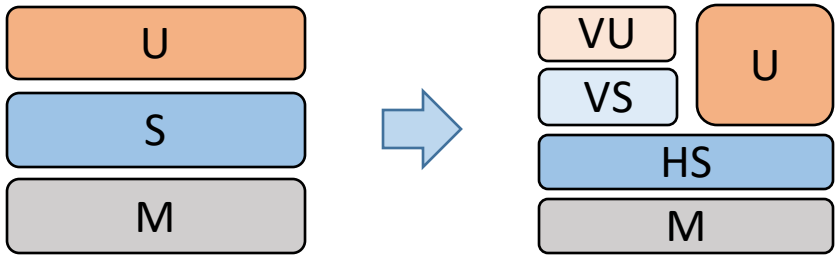


- 内存虚拟化
  - 两阶段地址翻译：客户机的地址翻译、主机的地址翻译



昆明湖虚拟化扩展设计·CPU虚拟化

- 特权级
  - 新增V位，区分VS和HS、VU和U



- CSR寄存器

Hypervisor CSR	hstatus、hedeleg、hideleg、hvip、hip、hie、hgatp等
Virtual Supervisor CSR	vsstatus、vsip、vsie、vstvec、vsepc、vsatp等
Machine CSR	mstatus、mideleg、mip、mie、mtval2（新增）、mtinst（新增）

- Hypervisor指令

访存指令	HLV.width、HLVX.HU/WU、HSV.width
Fence指令	HFENCE.VVMA/GVMA

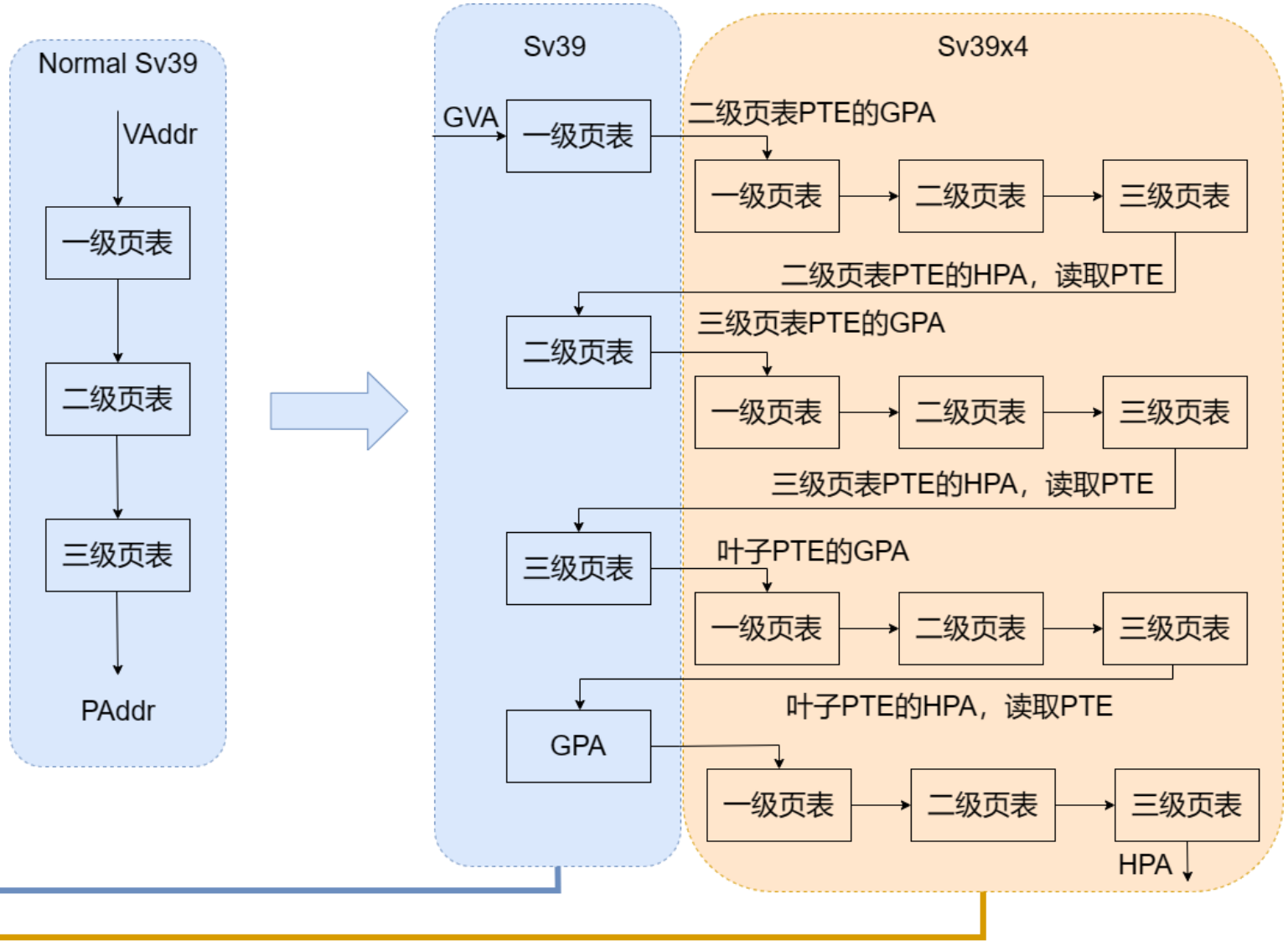
- Trap
  - 增加VS级陷入陷出的处理

新增中断	VS software interrupt、VS timer interrupt、VS external interrupt、Supervisor guest external interrupt
新增异常	Environment call from VS-mode、Instruction guest-page fault、Load guest-page fault、Virtual instruction、Store/AMO guest-page fault

昆明湖虚拟化扩展设计·内存虚拟化

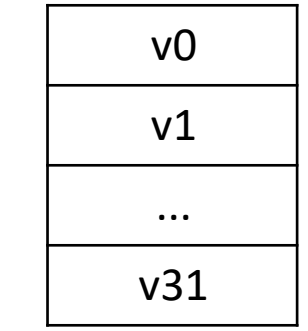
- 地址翻译模式
  - Sv39 --> Sv39 + Sv39x4
- 内存页表查询
  - 访存维度
  - 访存次数

4 * 3 = 12

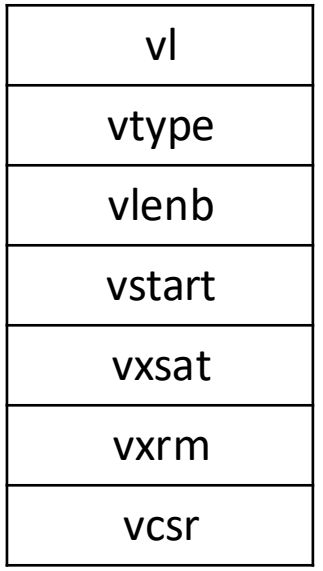


RISC-V向量扩展简介[1]

- 可编程寄存器
  - v0~v31　32个向量寄存器，v0还用做谓词寄存器
  - vstart　向量起始位置
  - vl　向量长度
  - vtype　向量配置，包括元素位宽和寄存器组个数等
  - …
- 特点
  - 可变向量长度
  - 支持向量谓词化
  - 寄存器组可扩充 8x 向量长度
  - 向量配置指令指定位宽，指令不编码位宽信息



新增向量寄存器



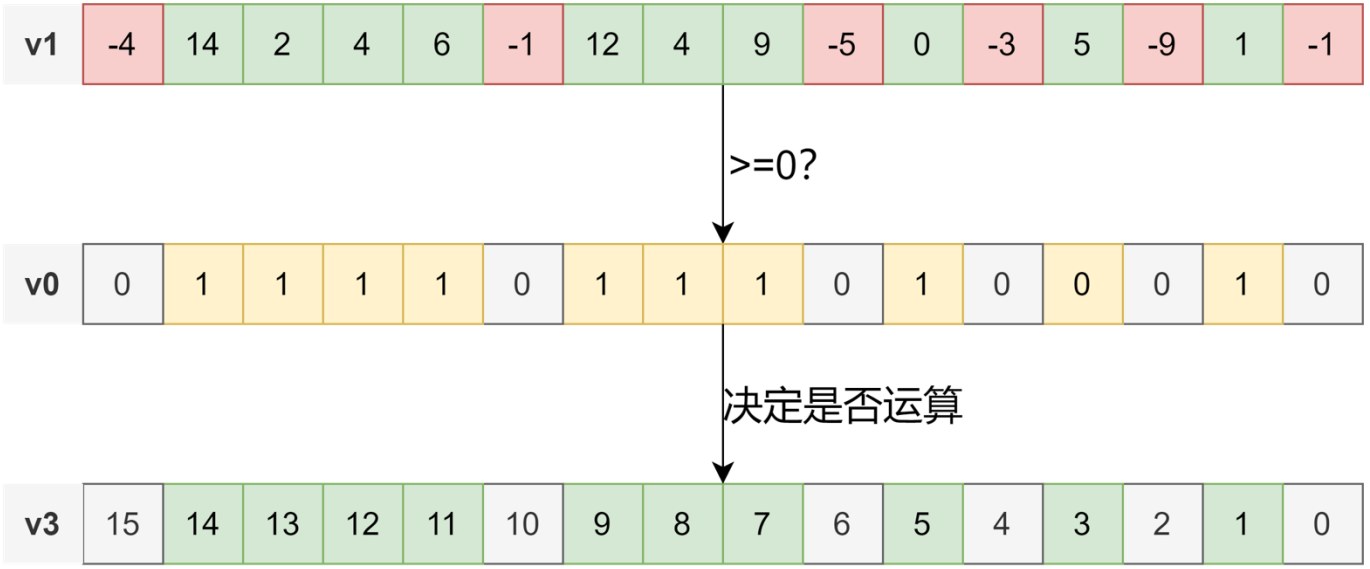
新增向量 CSR

[1] https://github.com/riscv/riscv-v-spec

向量的优势

- 提升单次操作的数据位宽
- 可转换控制依赖为数据依赖，消除难预测分支

```
for (int i = 0; i < 16; i++) {
    if (a[i] >= 0)
        a[i] = a[i] * 2;
}
```



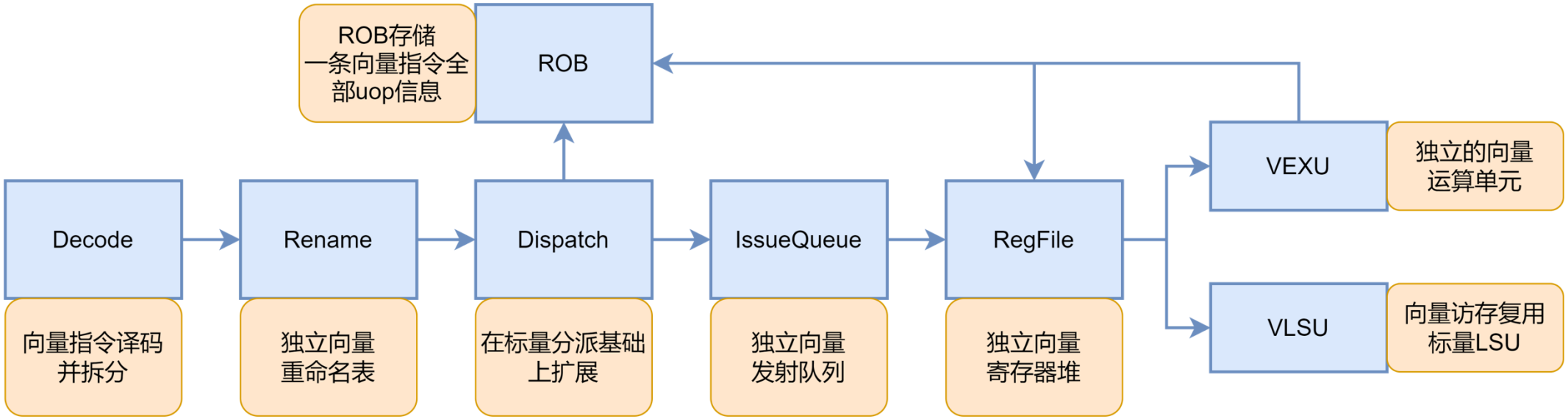
向量化消除循环内分支

香山 RVV 实现

- 基本配置
  - VLEN=128
  - 支持数据类型
    - 整数8~64，FP64、FP32
  - 支持全部 LMUL 配置（**最大向量长度1024bit**）
- 紧耦合实现
  - 向量长度寄存器 vl 重命名
  - 复用标量访存流水线（扩展数据位宽至128bit）
  - **以向量寄存器粒度拆分 uop**
  - **推测向量配置信息**

香山向量设计架构

- 紧耦合式向量扩展设计
- 复用流水线：译码/分派/乱序/访存



支持向量扩展对主流水线的修改

香山向量现状及展望

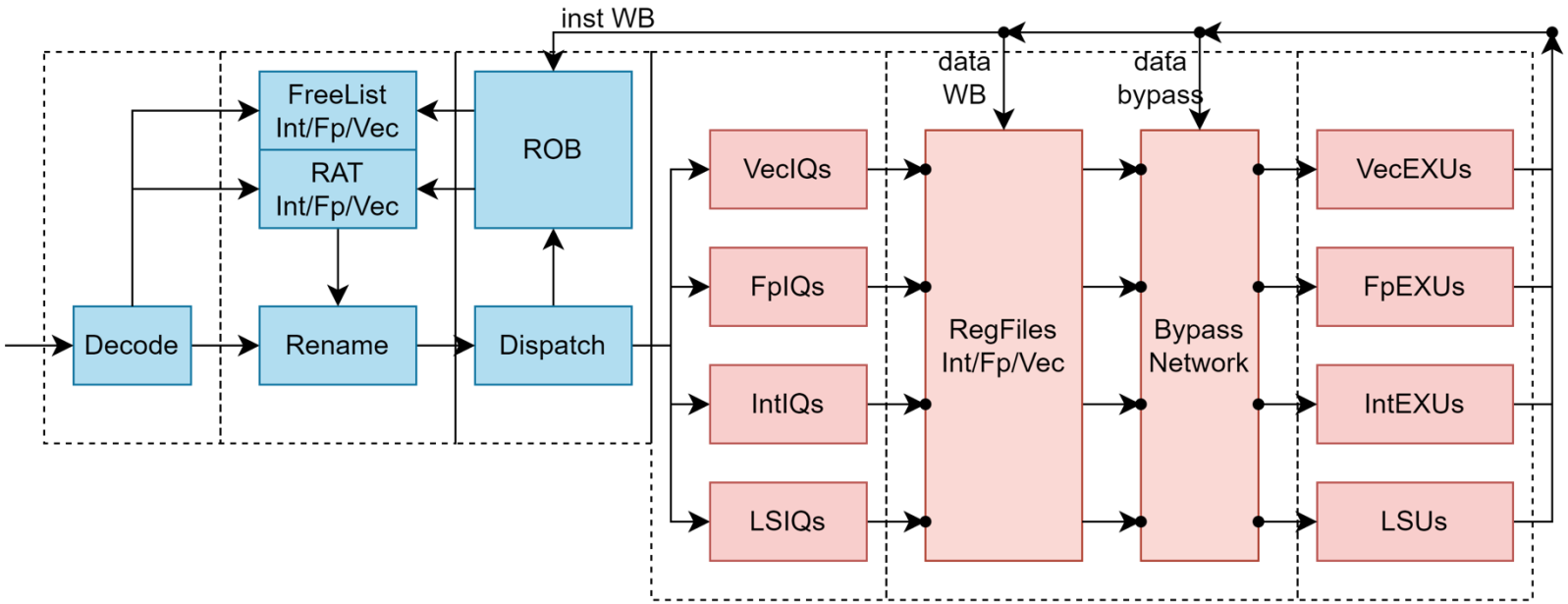
- 现状
  - 支持 RISC-V Vector 1.0
  - 自动向量化程序在部分 Benchmark 对比标量有性能提升
- 展望
  - 更高效的乱序调度
  - 更长的向量长度
  - 更强大的乱序访存

昆明湖微架构改进

- 新支持指令集扩展
- 发射后读寄存器堆的新后端
- 更低功耗的 ICache
- 容纳更大乱序窗口的 LoadQueue 和 LSU 设计
- 支持 CHI 总线的 Coupled L2 缓存

全新设计的后端流水线

- 重命名快照
- 基于 RAT 的 Move 消除
- 发射后读寄存器堆
- 指令提交和寄存器提交解耦（ROB压缩）

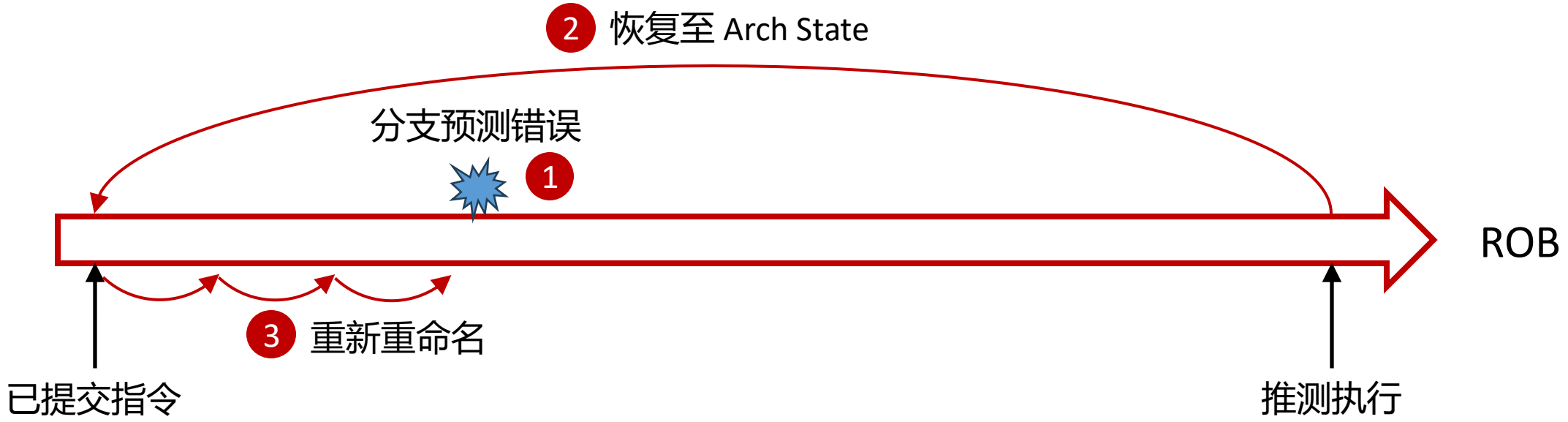


昆明湖重命名恢复方案

- 重新重命名： 改进的 Arch State + ROB Walk
  - 新指令无需等待 in-flight 指令全部执行完成

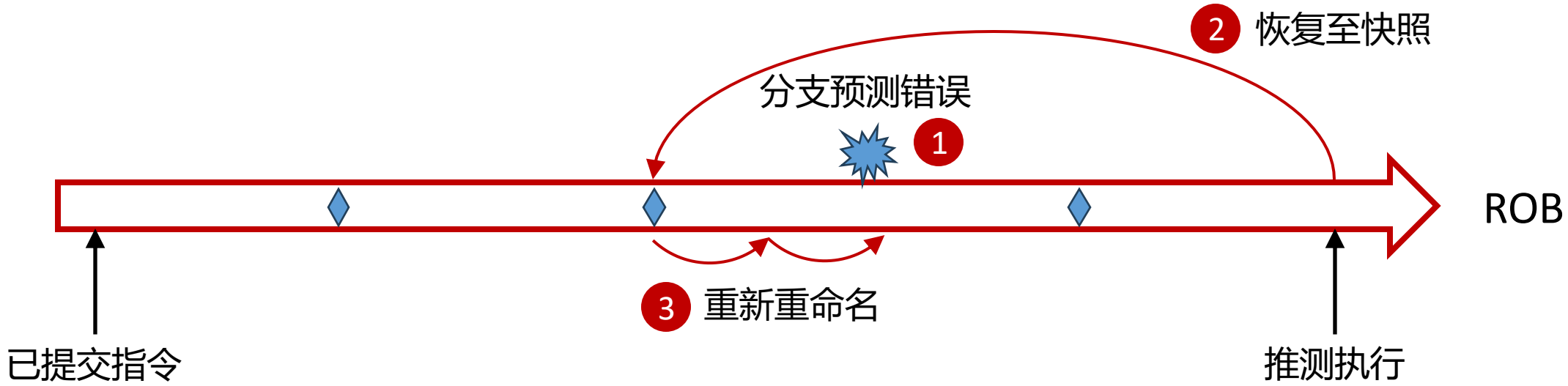
统一

  - 需要恢复的指令多 -> 重新重命名时间短 -> 较短时间无法进入新指令
  - 需要恢复的指令多 -> 正确路径 in-flight 指令少 -> 较短时间即可完成既有指令



昆明湖重命名恢复方案

- 重命名快照：改进的 Checkpoint
  - 无需为每个分支指令生成检查点
  - 恢复到最近的快照后重新重命名——**缩短**重新重命名时间



重命名快照带来的存储开销问题

- RefCounter 占用空间大
  - 香山第二代 Move 消除使用 RefCounter 记录物理寄存器引用计数
  - 总比特数

$$S = \log(\text{ROB项数}) \times \text{整数物理寄存器数} = 8 \times 192 = 1536$$

  - 加入快照后总比特数

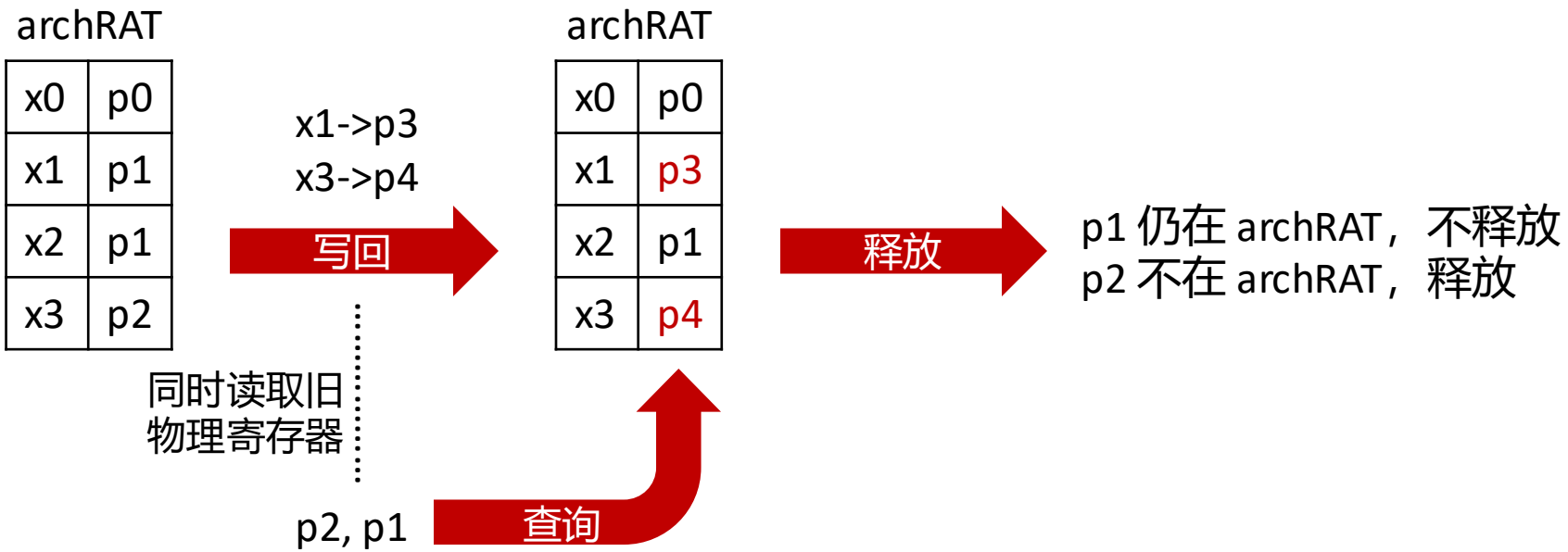
$$S' = (1 + 4 + 1)S = 9216$$

快照数量　　Arch State

  - 尺寸与物理寄存器堆大小相当

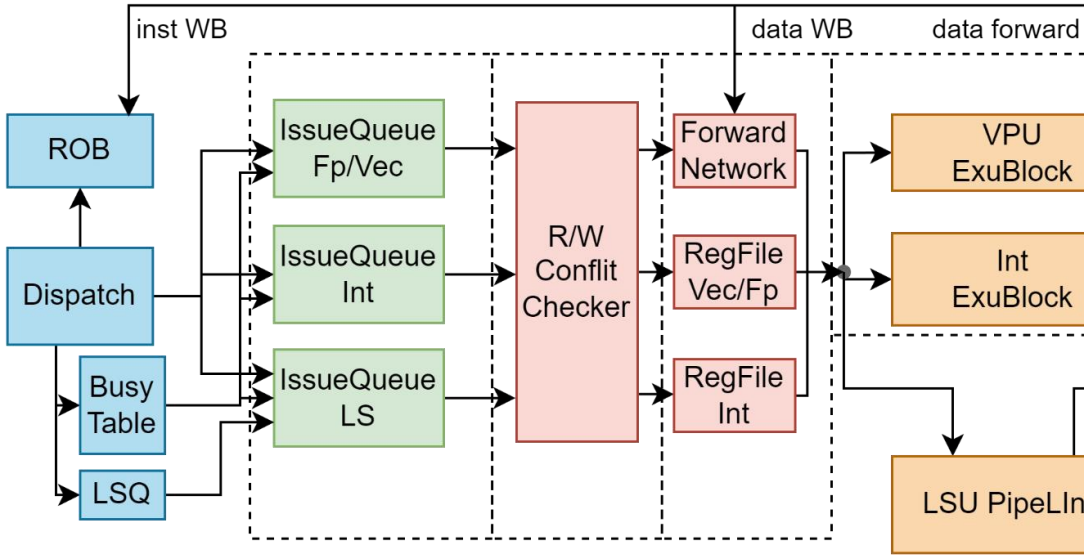
存储开销问题——解决方法

- 基于 RAT 的 Move 消除机制
  - 直接查询旧物理寄存器是否还在 RAT 中
  - 去除复杂的结构，时序、功耗、面积更佳



发射后读寄存器堆

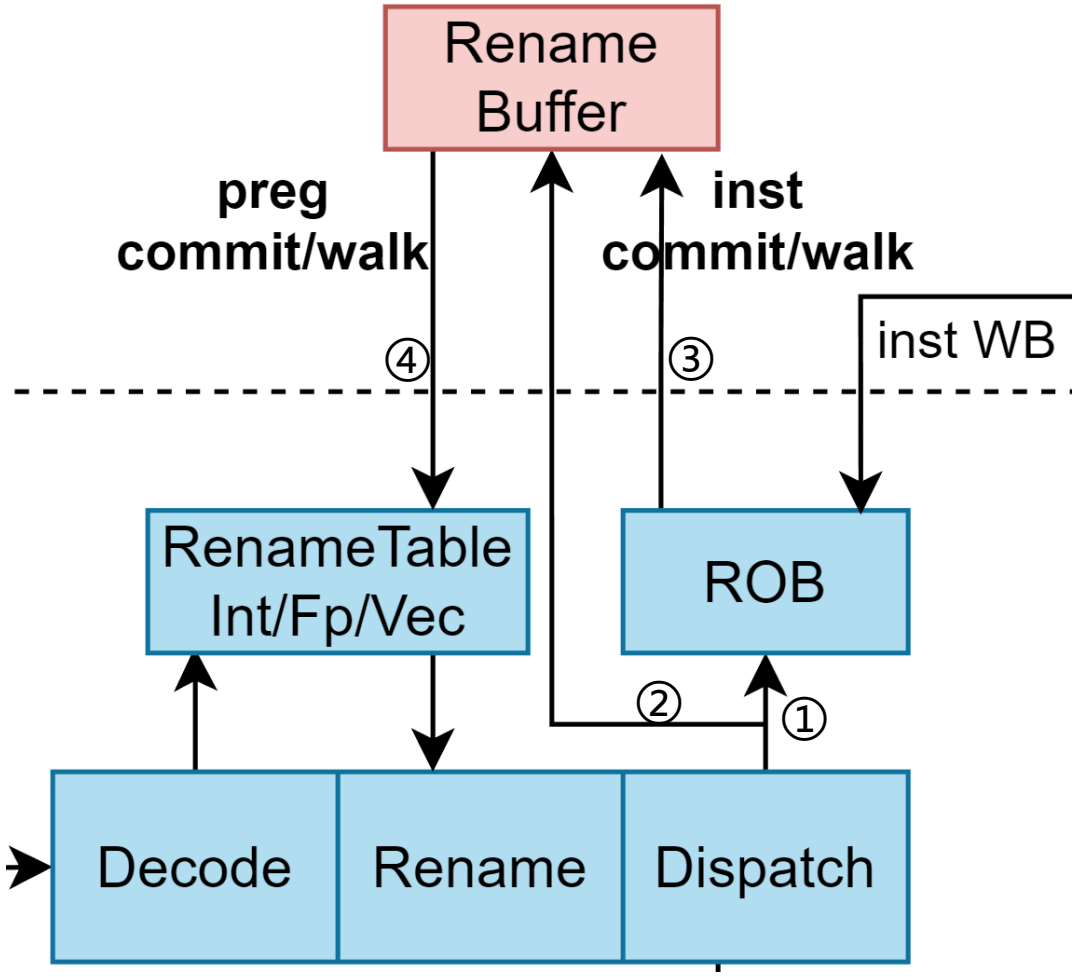
- 优点
  - 指令发射不再因为寄存器读口受限
  - 减少发射队列占用的存储空间
  - 降低延迟，增大发射队列面积
- 实现特点
  - 高效的寄存器读口共享和仲裁策略
  - 准确的推测唤醒和推测取消



Read regfile after issue

指令提交和寄存器提交解耦

- 重命名缓存 RAB：记录重命名表和寄存器堆的修改历史
  - ① 指令派遣后，将指令存入 ROB
  - ② 将寄存器映射修改信息存入 RAB
  - ③ 指令提交或重定向时，唤醒 RAB
  - ④ 由 RAB 指导重命名表 RAT 的更新
- 优势
  - 更快的 ROB 项释放
  - 支持 ROB 压缩
  - 更小的 ROB 项数需求，优化时序
  - 支持向量指令 μop 拆分

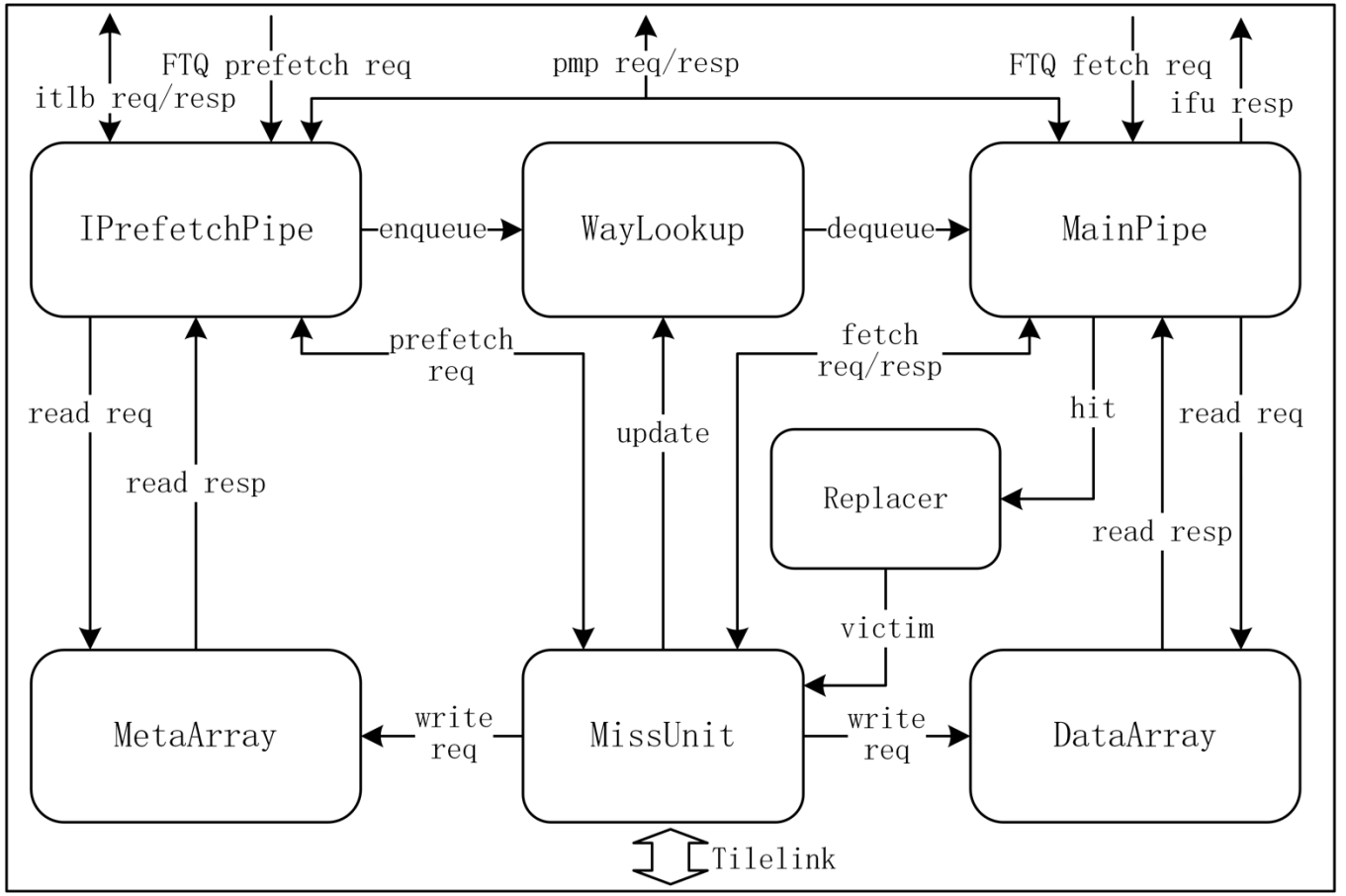


昆明湖微架构改进

- 新支持指令集扩展
- 发射后读寄存器堆的新后端
- 更低功耗的 ICache
- 时序更优的 LoadQueue 和 LSU 设计
- 支持 CHI 总线的 Coupled L2 缓存

新 ICache

- 低功耗设计
  - MetaArray 和 DataArray 解耦分别读取
  - 更细粒度的数据存储方式
- 基于 FDIP 的 L1I 预取
- MSHR 数量可配置
  - 4 Fetch MSHR
  - 10 Prefetch MSHR

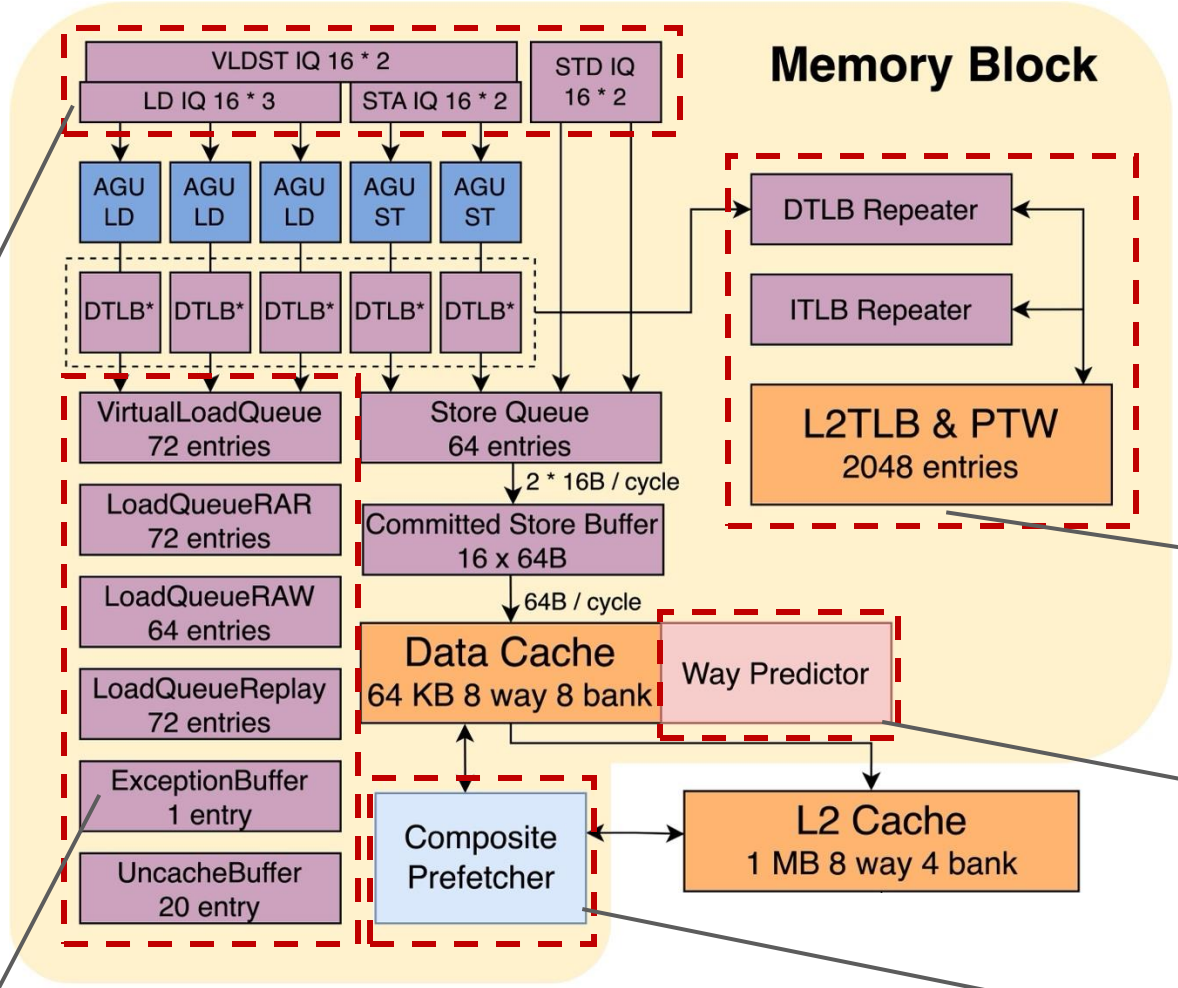


昆明湖微架构改进

- 新支持指令集扩展
- 发射后读寄存器堆的新后端
- 更低功耗的 ICache
- 时序更优的 LoadQueue 和 LSU 设计
- 支持 CHI 总线的 Coupled L2 缓存

昆明湖 Load Store Unit 总览

功能：向量访存独立发射队列
复用标量访存功能单元



问题：TLB 时序紧张
优化：重构 L1/L2 TLB 的结构

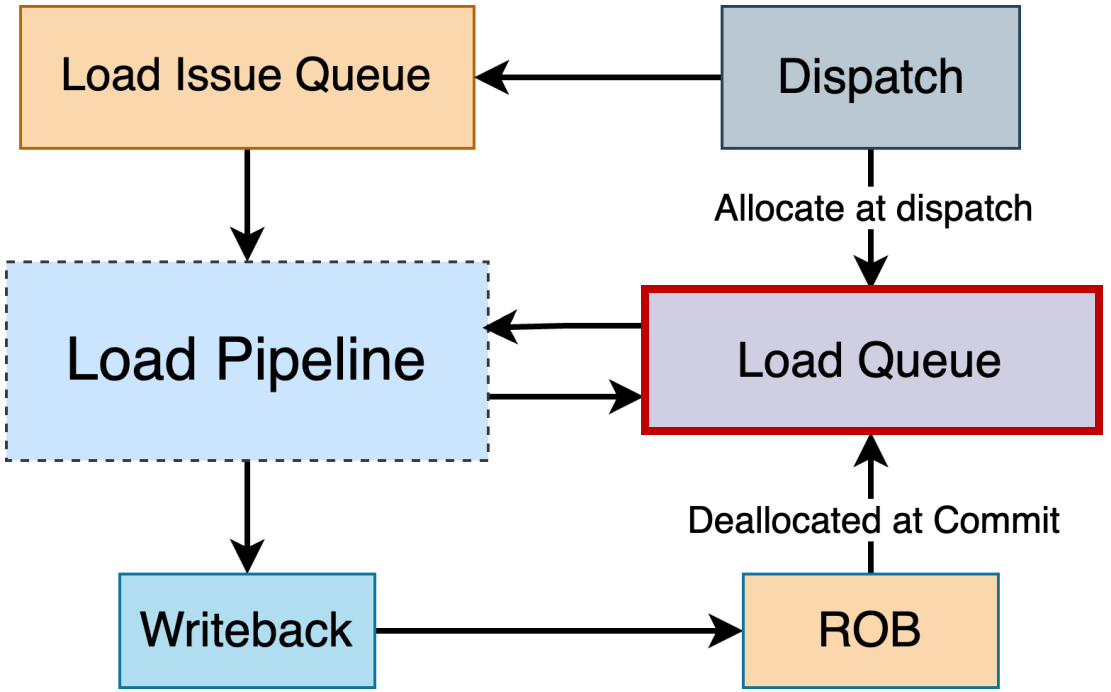
问题：高 L1D SRAM 功耗
优化：支持路预测

问题：高 L1D miss 率
优化：支持 L1D 预取

问题：Load Queue 时序较差
优化：分离式 Load Queue

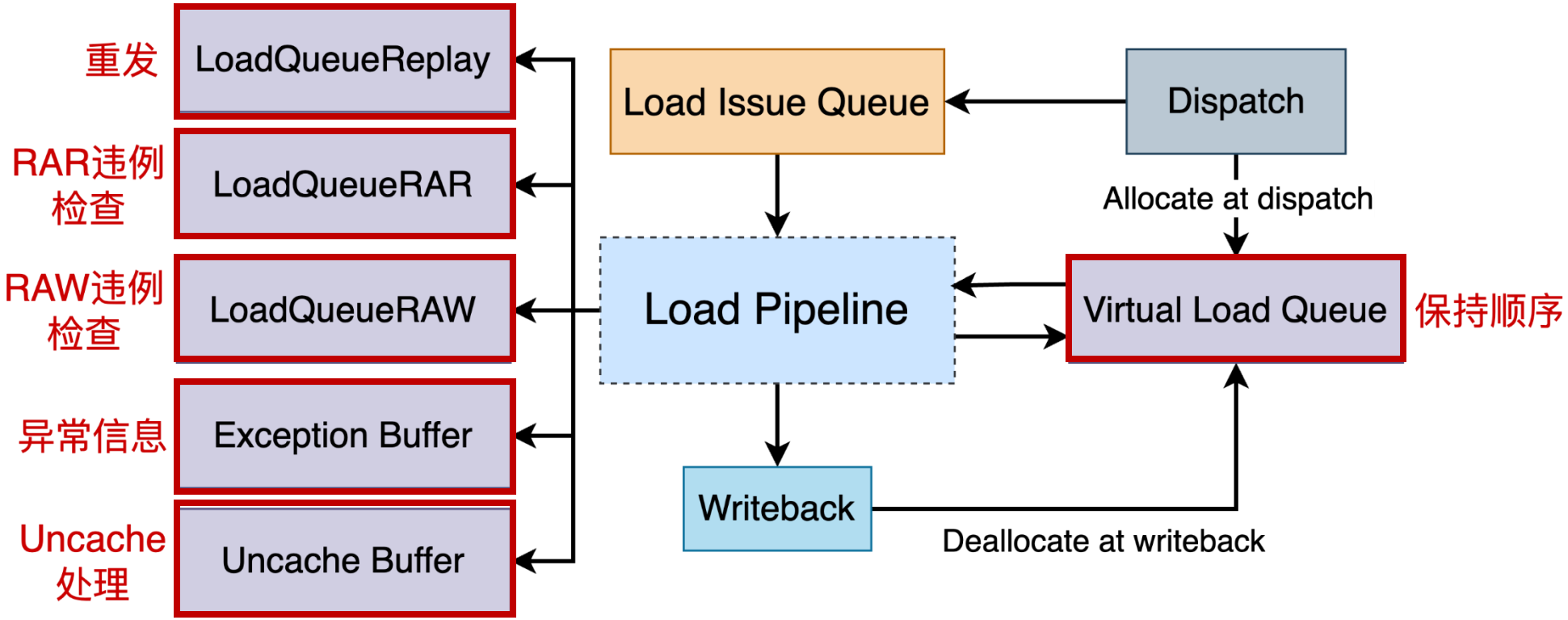
Load Queue 拆分

- 雁栖湖，南湖采用 **统一的 load queue** 设计
  - **职责多**：负责 uncache， RAR/RAW 违例检查，维护 load 状态，异常信息等
  - **资源需求多**：多个 CAM 端口，维护的信息多等
  - **释放比较缓慢**： ROB 提交之后才能从 load queue 移除



Load Queue 拆分

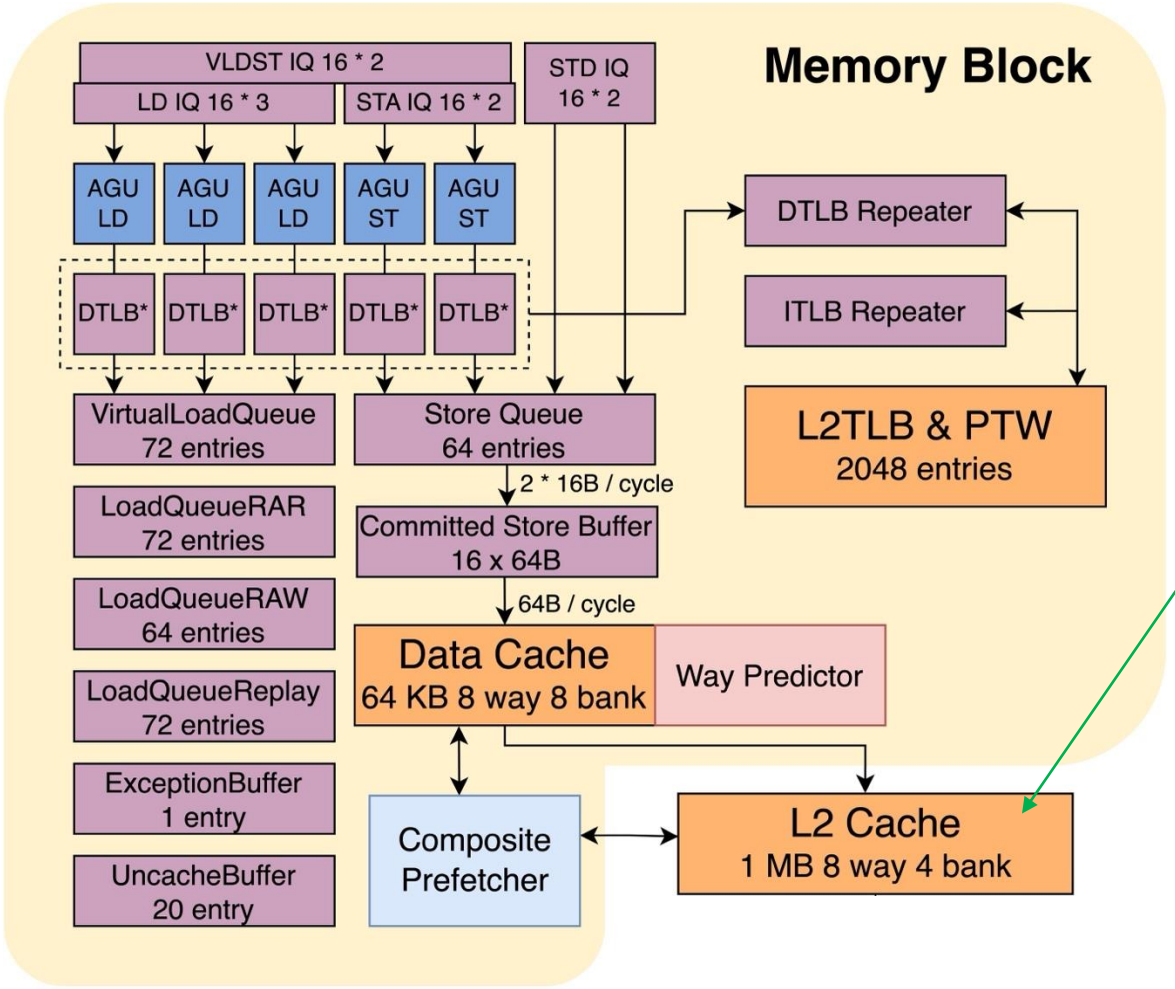
- 昆明湖的 load queue 根据**不同职责**采用拆分设计
  - **early commit**：load 指令在写回后可从 virtual load queue 中按顺序移除
  - 逻辑简单、时序友好、可接受资源消耗



昆明湖微架构改进

- 新支持指令集扩展
- 发射后读寄存器堆的新后端
- 更低功耗的 ICache
- 时序更优的 LoadQueue 和 LSU 设计
- 支持 CHI 总线的 Coupled L2 缓存

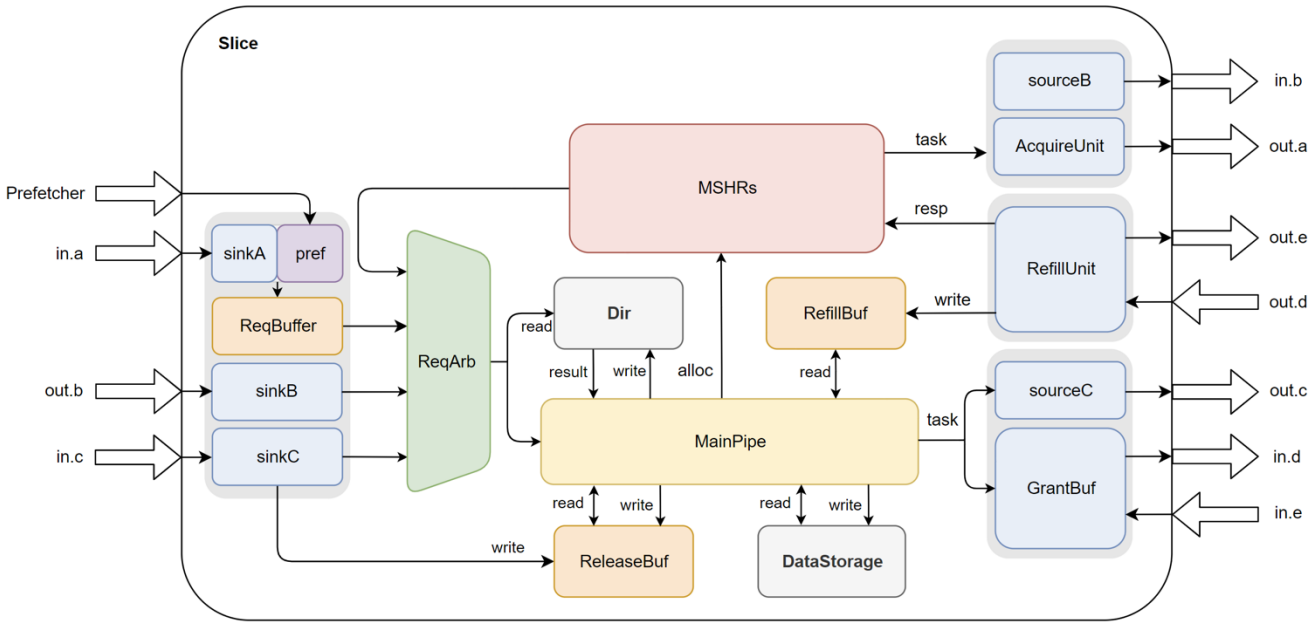
昆明湖 CoupledL2 总览

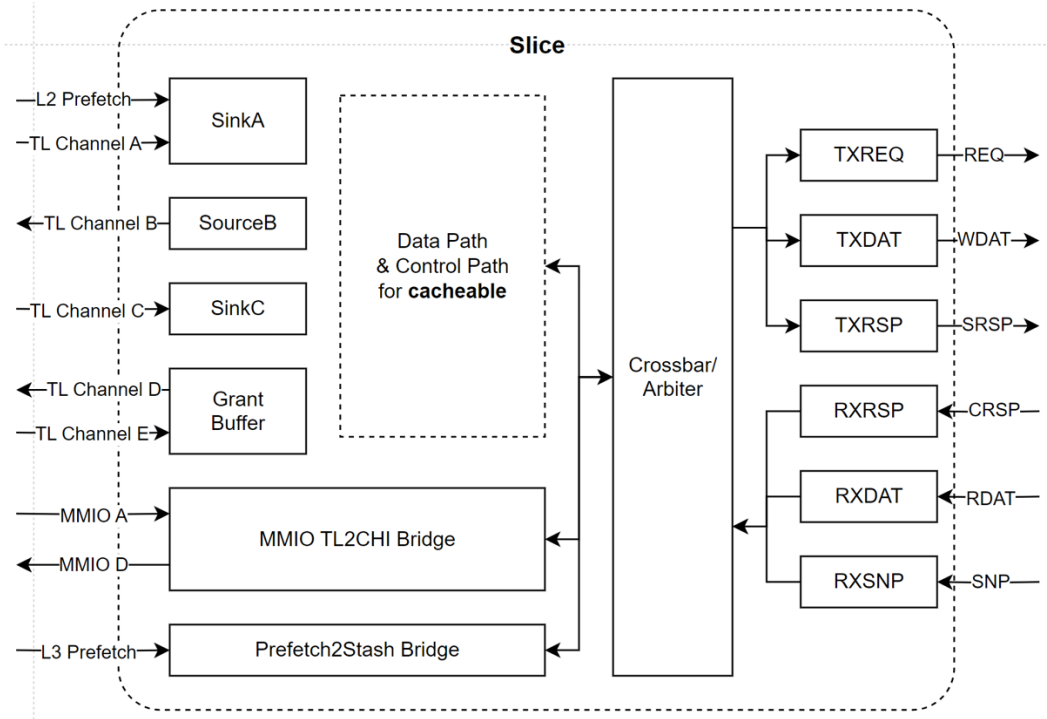


- **8 路**组相联，最高 **1 MB** 容量
- 缓存块 **64 字节**
- 对 DCache inclusive
- 对 ICache non-inclusive
- **10 周期** load to use 延迟
- PLRU/RRIP 替换算法

CoupledL2 对外协议可配置

- L1 <-> L2 采用 TileLink 1.0 协议
- L2 <-> LLC / NoC 可选择 CHI Issue B / CHI Issue E.b / TileLink 1.0 协议





昆明湖南湖微架构对比

- 新支持指令集扩展
- 发射后读寄存器堆的新后端
- 更低功耗的 ICache
- 时序更优的 LoadQueue 和 LSU 设计
- 支持 CHI 总线的 Coupled L2 缓存

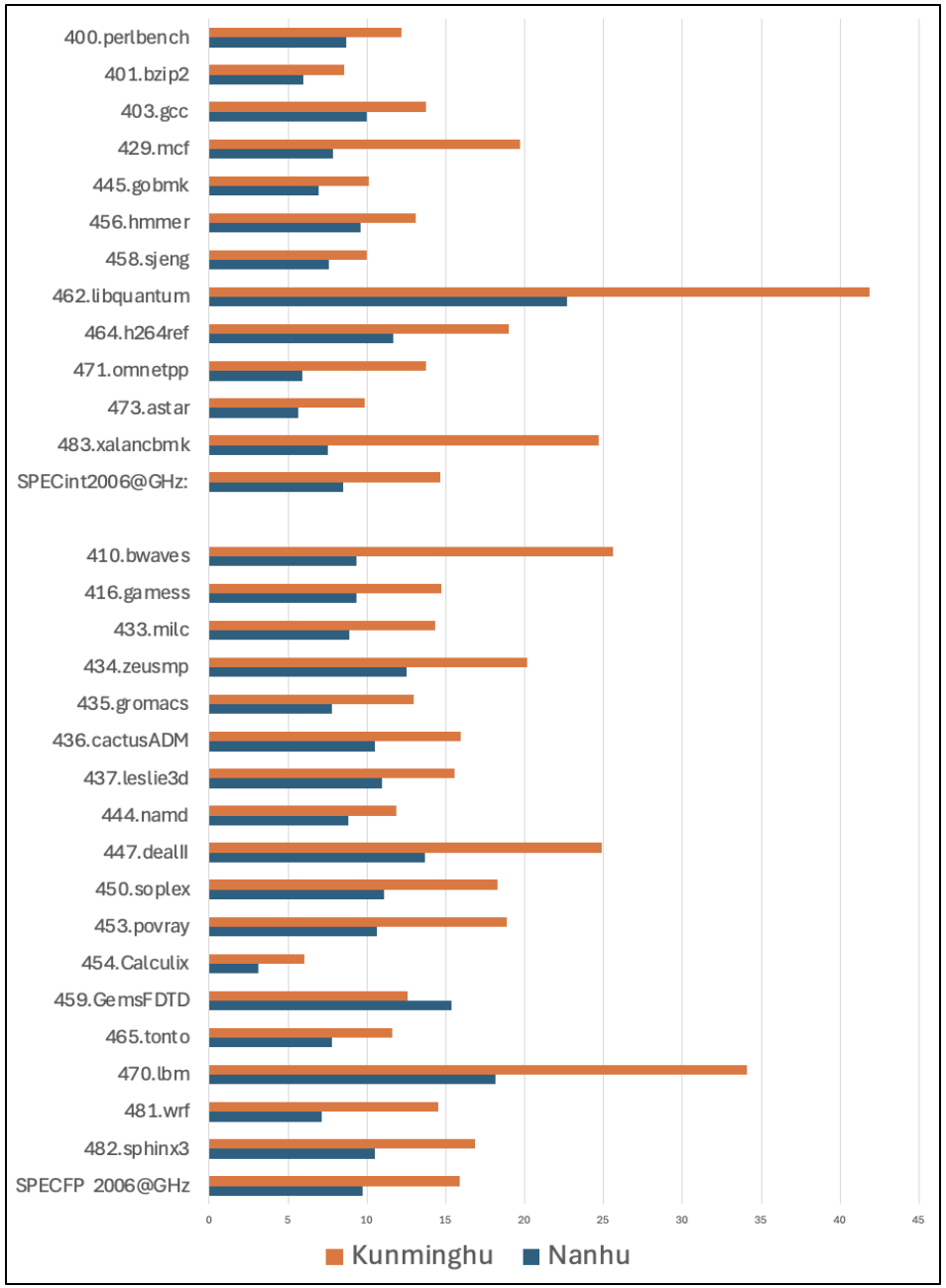
Feature	Kunminghu	Neoverse N2	Nanhu	Cortex A76
Pipeline depth	13	10	13	13
Rename width	6	5	4	4
Rename checkpoint	Y	Y	N	N
ROB size	160 (x6)	160+	192	128
ALUs	4	4	4	3
L1 instruction cache	64KB	64KB	64KB	64KB
L1 data cache	64KB	64KB	64KB	64KB
L2 cache	1024KB	512/1024KB	256KB	256/512KB
L3 cache	Up to 16MB	4MB per slice	Up to 4MB	Up to 4MB
NoC support	Y	Y	N	N
L2 outstanding txns	64	64	32	46
ITLB	48	48	32	48
DTLB	48	44	128 direct mapped	48
L2 TLB	2048	1280	2048	1280
Vector	Y	Y	Y	Y
Virtualization	Y	Y	N	Y
ECC support	Y	Y	Y	Y
PMA/PMP support	Y	Y	Y	Y
Debug support	Y	Y	Y	Y
External interface	AXI4/TL/CHI	AXI4/CHI	AXI4/TL	AXI4/CHI

性能评估

- **RTL 仿真：基于 SimPoint 的程序采样评估**
  - 编译器：GCC 12 -O3，RV64GCB，jemalloc
  - 缓存配置：64KB I$/D$ + 256KB L2$ + 4MB L3$（南湖）
    64KB I$/D$ + 1MB L2$ + 16MB L3$（昆明湖）
  - 内存模型：DRAMsim3 DDR4@3200MHz
    - 70ns 延迟
    - 双通道，2 x 64
- **评估结果**

Base score	SPECint 2006	SPECfp 2006
南湖@2GHz	**16.94**	**19.42**
昆明湖@3GHz	**44.00 → 49.96***	**47.63**

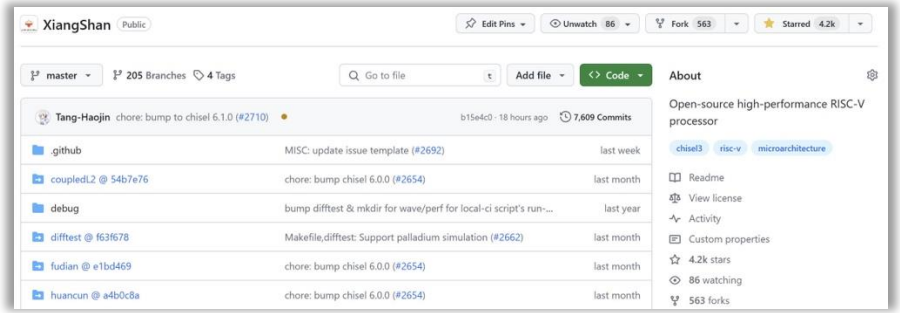
* 使用了更激进的编译优化



香山：开源高性能处理器

- **第一代架构：雁栖湖**
  - 2020/6：RTL 设计的第一个提交
  - 2021/7：**28nm 流片，1.3GHz**
  - 性能：**SPEC CPU2006 7.01@1GHz, DDR4-1600**
- **第二代架构：南湖**
  - 2021/5：开始设计探索和架构设计
  - 2023/4：GDSII 交付
  - **14-nm 流片，SPEC CPU2006 20@2GHz**
- **第三代架构：昆明湖**
  - 新的指令集扩展（虚拟化、向量等）
  - 优化设计的流水线部件
  - CHI-CoupledL2 缓存
  - 性能：**SPEC CPU2006 45@3GHz**

北京香山



>4.2K stars, >500 forks on GitHub

北京开源芯片研究院
BEIJING INSTITUTE OF OPEN SOURCE CHIP

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

中国开放指令生态（RISC-V）联盟
China RISC-V Alliance

感谢！

欢迎参加下午的香山开发者大会（香山 Tutorial）

8月22日下午 紫荆1