



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

报告人：唐浩晋

中国科学院计算技术研究所

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



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

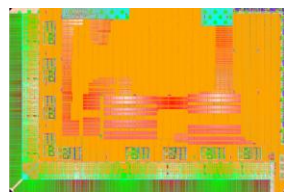
- 工业界需要定制化需求和快速开发迭代
- 高可配置性让领域特定架构的快速生成和海量的定制化需求成为可能



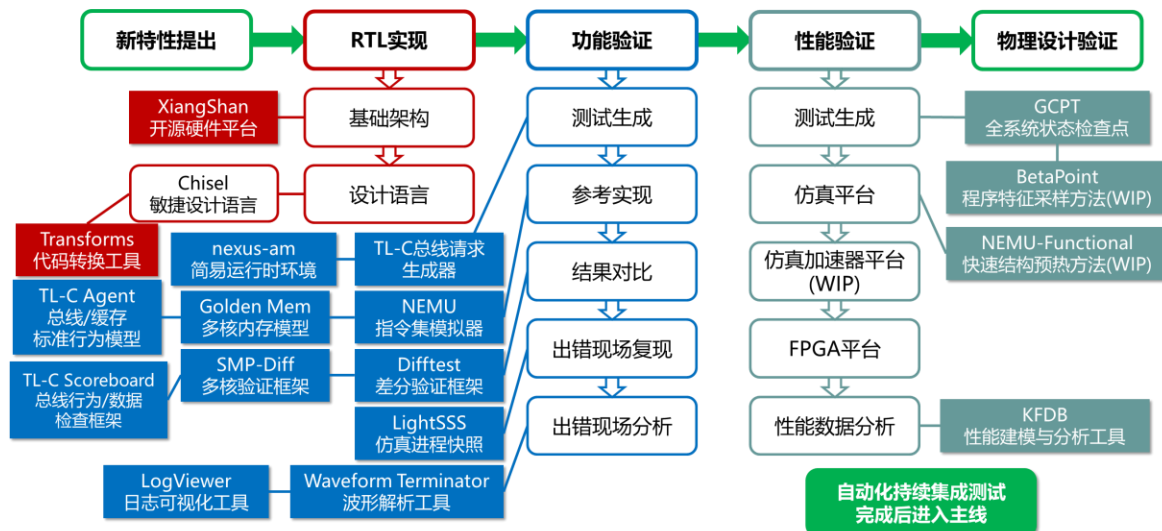
➤ “处理器界的 Linux”

开源开放的处理器芯片研发能力体系

- 香山核心价值是构建一套**芯片敏捷设计基础设施**，缩短迭代优化周期
- **开源开放能力体系**，联合企业加速处理器研发节奏，支持按需快速定制芯片



成果开源



能力开放

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

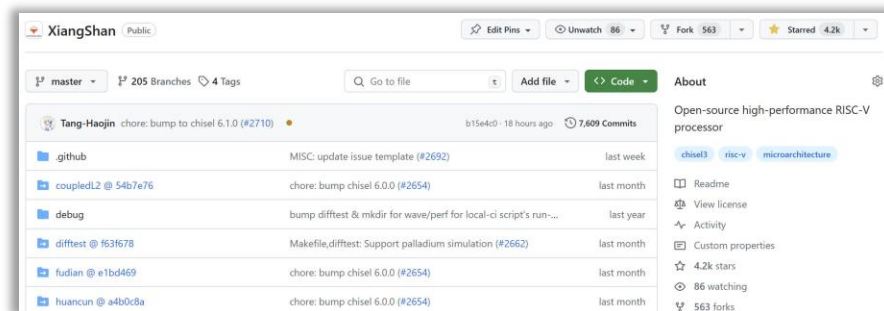
- 2021/5: 开始设计探索和架构设计
- 2023/4: GDSII 交付
- **14-nm 流片, SPEC CPU2006 20 Est. @2GHz**

• 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 Est. @3GHz**



香山

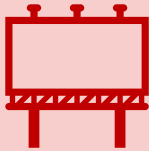


>4.5K stars, >630 forks on GitHub

第三代昆明湖架构



功能与性能

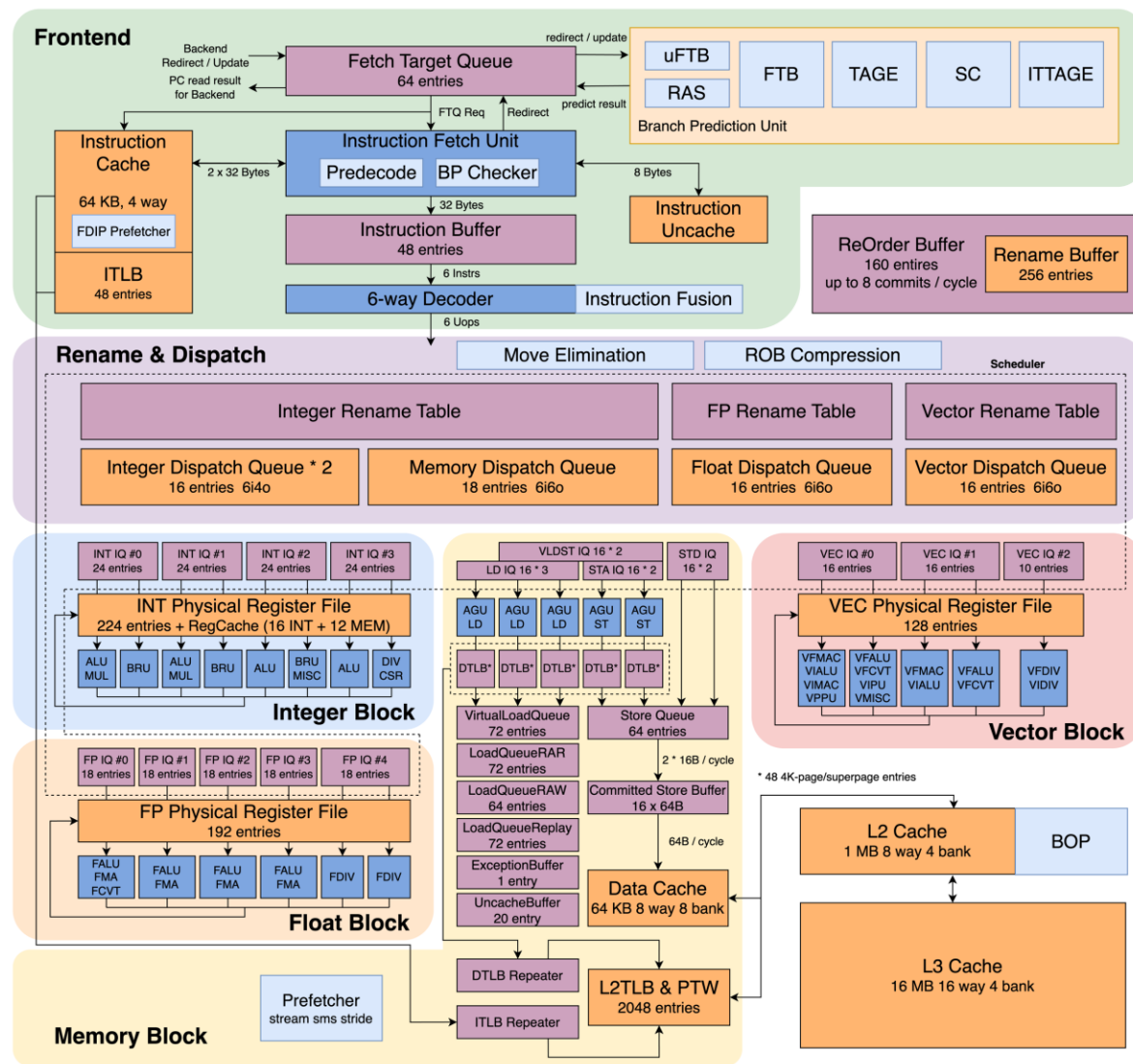


产品化交付

- 功能与性能
 - 指令集扩展与 RVA23
 - 微架构设计改进升级
 - 性能评估与对比
 - 低功耗
- 产品化交付
 - 工业级的验证流程
 - 与会员企业的深度合作
 - 灵活可配置的交付单元

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
- 48-entry ITLB / DTLB + 2048-entry L2TLB
- FDIP 指令预取
- stream / sms / stride / bop / tp 预取器



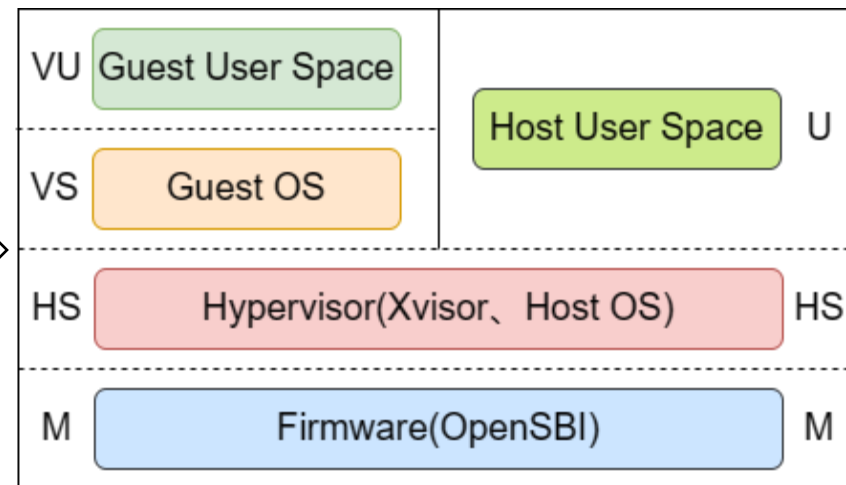
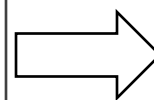
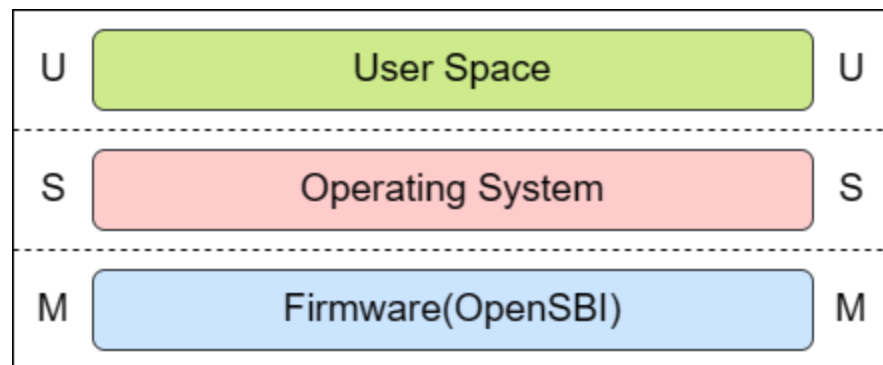
功能：昆明湖架构新支持的指令集扩展

- H 扩展：硬件虚拟化
- V 扩展：向量运算与向量访存
- Sv48：支持 48 位虚拟地址
- RVA23 Profile
 - Sstc：S 态时钟中断（同时支持 rdttime 指令）
 - Svpbmt：基于页的内存属性
 - Zicbop：软件预取指令
 -

🏔️ RISC-V 虚拟化 H 扩展

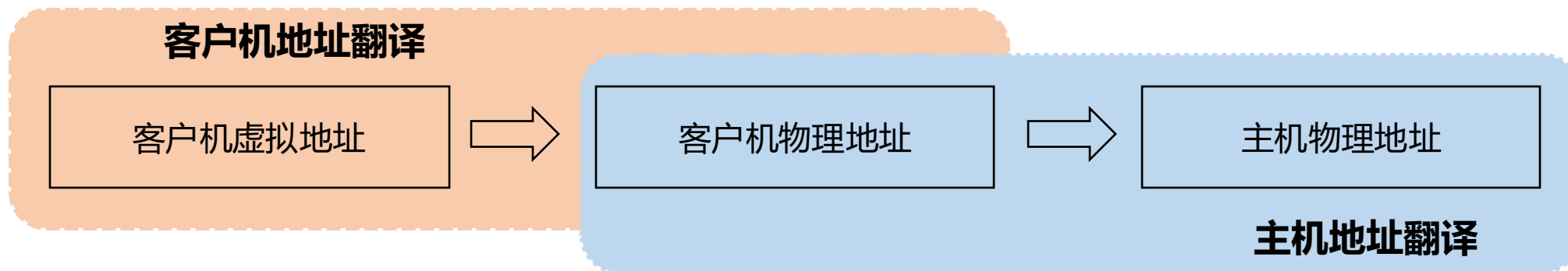
• CPU虚拟化

- 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



昆明湖向量 v 扩展简介

- 昆明湖基本配置
 - VLEN=128
 - 支持数据类型
 - 整数 8 ~ 64, FP64、FP32、FP16min
 - 支持全部 LMUL 配置 (**最大向量长度1024bit**)
- 基于 SimPoint 估计的 SPEC CPU2006 456.hmmmer 向量性能与标量相比**提高 24.4%**

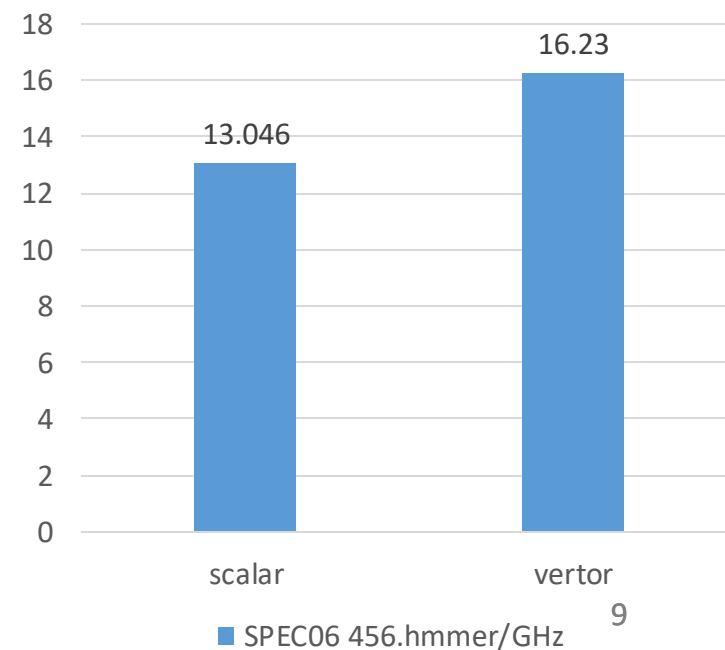
详见稍后的“香山处理器昆明湖架构向量扩展的设计与演进”报告

| |
|-----|
| v0 |
| v1 |
| ... |
| v31 |

新增向量寄存器

| |
|--------|
| v1 |
| vtype |
| vlenb |
| vstart |
| vxsat |
| vxrm |
| vcsr |

新增向量 CSR



RVA23 和 Sv48

- RVA23 Profile 对于服务器生态十分重要
 - 安全: Zkt、Zvkt...
 - 功能: Sscounterenw、Shcounterenw...
 - 性能: V、Sstc、Zicbop、Zicboz、Zicbom、Zicclsm、Zicond、Zcb...
 - 功耗: Zawrs、Zihintpause
 - 调试: Sdext...
- 昆明湖将实现**所有 RVA23 必选项**和部分认为重要的可选项
- Sv48 是服务器芯片的**基本要求**
 - Sv39 -> Sv48: 512GB -> **256TB**
 - 最大物理地址支持也由 64GB (36 位地址线) -> 增长为 **256TB (48 位地址线)**

性能、功耗和面积 (PPA) : 昆明湖架构改进

• 前端

- 全新 ICache 设计
- 功耗面积优化

• 后端

- 寄存器重命名优化
- 发射后读寄存器堆

• 访存

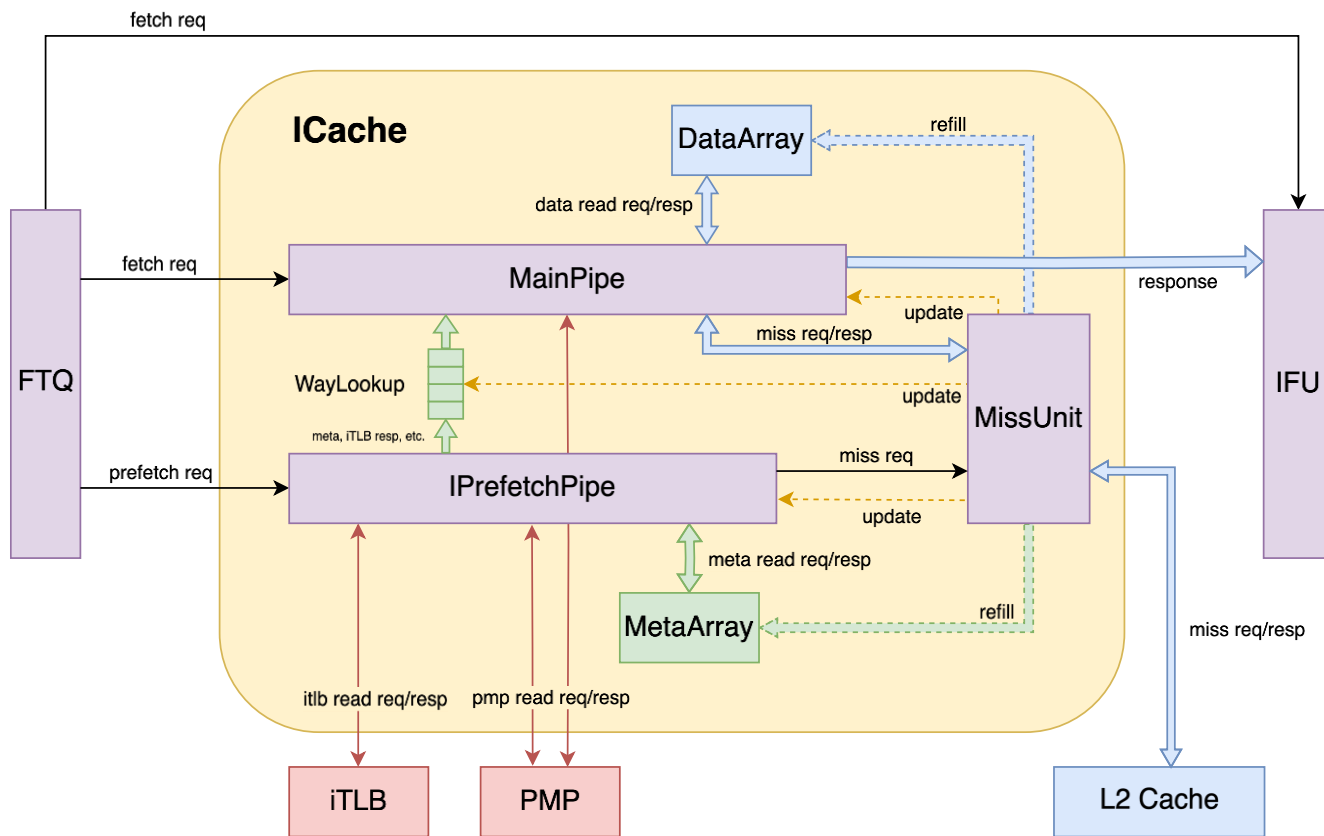
- LoadQueue 拆分设计
- L1 DCache Load 预取

• 缓存

- 性能联合优化
- CHI 总线协议支持

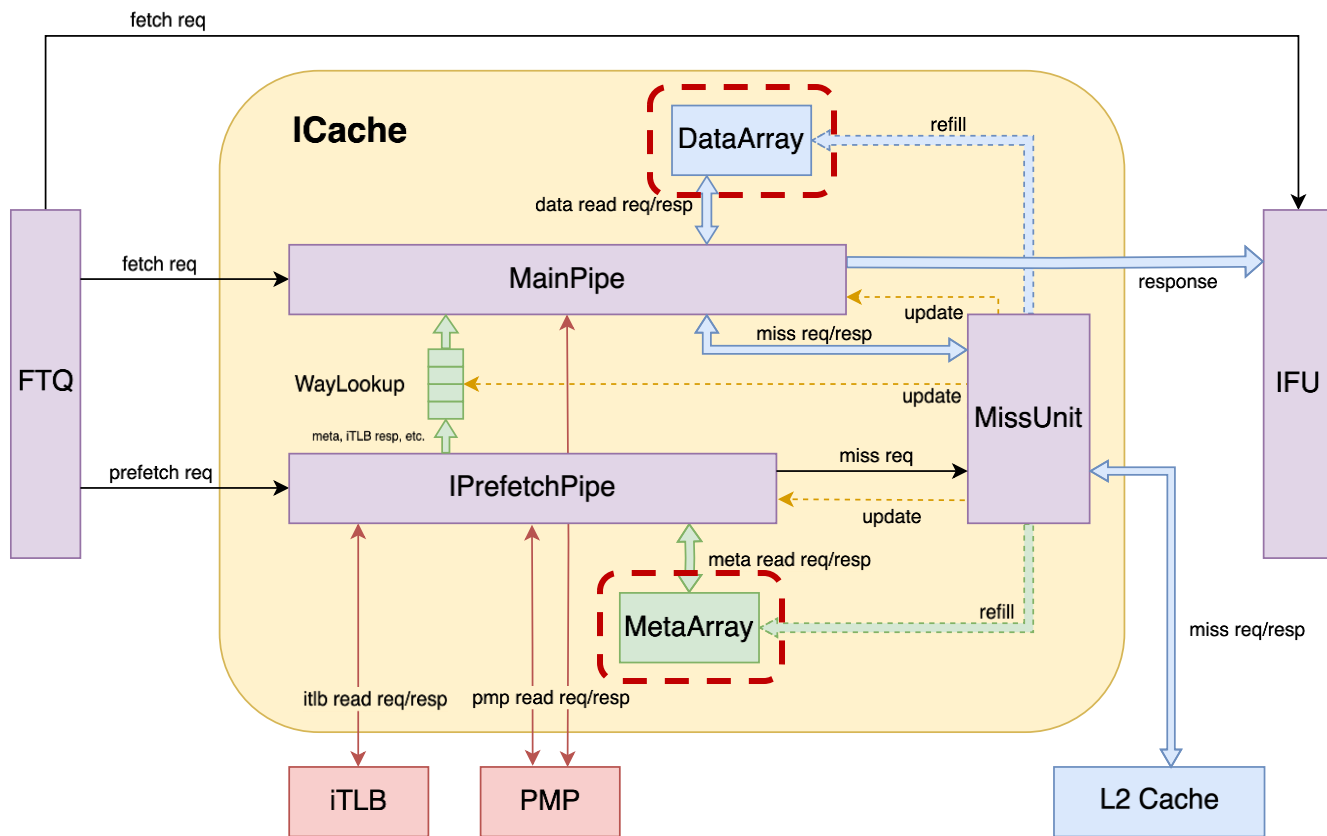
全新设计的 ICache —— 高性能

- FDIP 预取至 L1 ICache
 - 对比原先预取至 L2, 提高了 L1 ICache 的利用率
- 支持 Zicbop prefetch.i 软件预取
- MSHR 数量可配
 - Fetch MSHR 1 项 -> 4 项
 - 减小重定向取指延迟



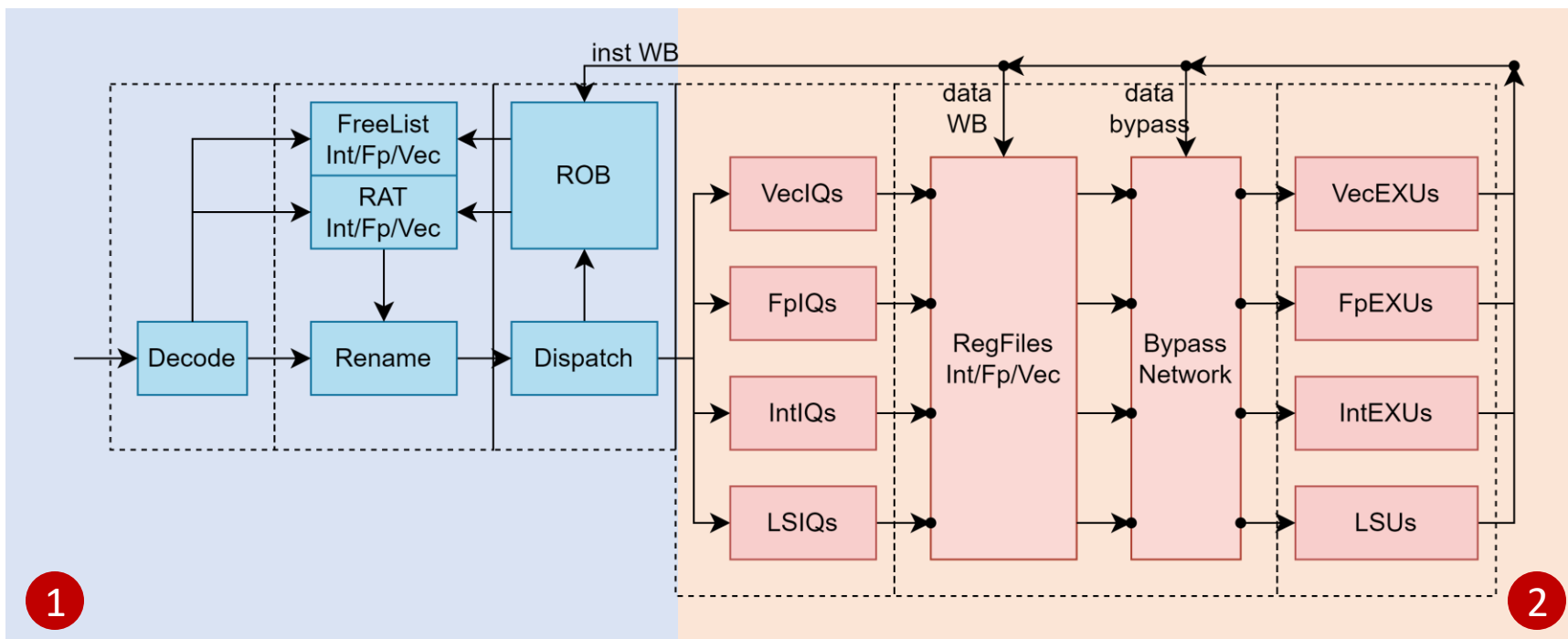
全新设计的 ICache —— 低功耗

- Meta 和 Data Array 读取解耦
 - 先读取 MetaArray, **命中后**再读取 dataArray
 - 未命中的路无需读取 dataArray
- 细粒度的 dataArray
 - 以 8 字节为粒度 **分 bank** 存储
 - 只读取需要的 bank, 节省功耗
- ICache 功耗 0.339W -> 0.105W
 - 功耗 **下降了 69%**



全新设计的后端流水线

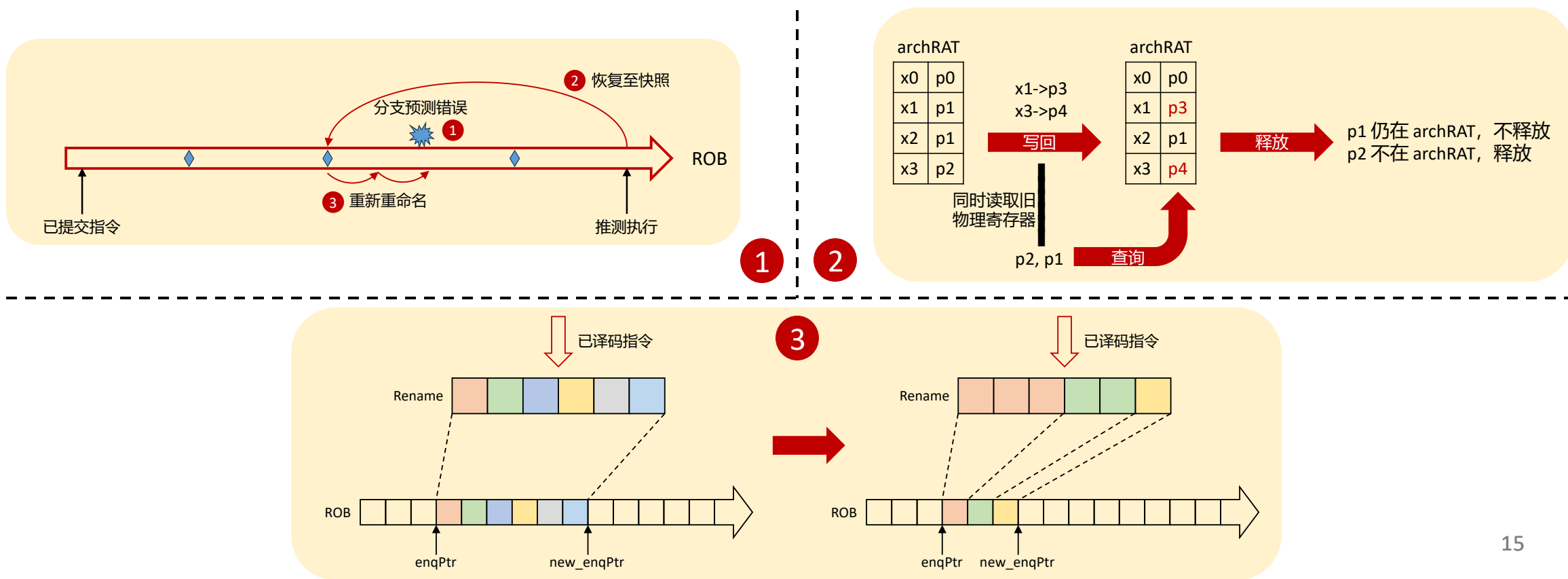
- ① 寄存器重命名优化
- ② 发射后读寄存器堆



后端流水线结构示意图

🔥 寄存器重命名优化

- ① 重命名快照：降低后端重定向的代价
- ② 基于 RAT 的 Move 消除：优化面积功耗表现
- ③ ROB 压缩：减少 ROB 的消耗，增大乱序窗口



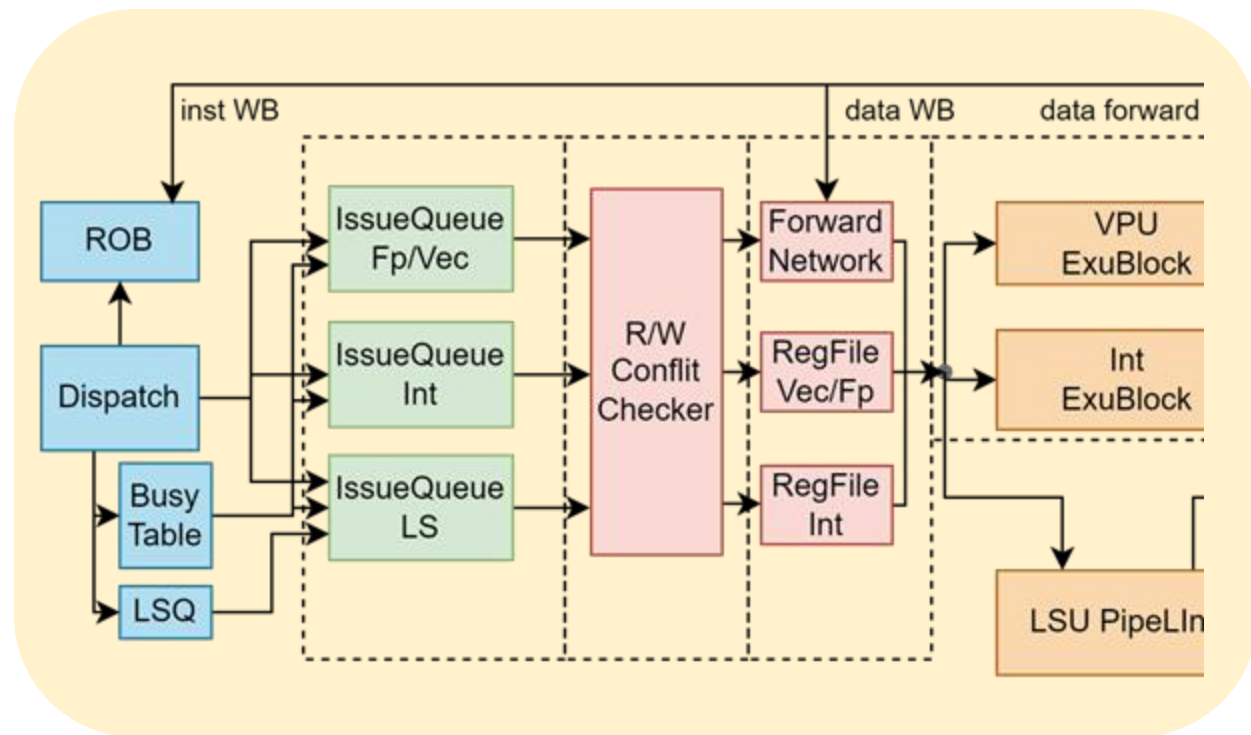
发射后读寄存器堆

• 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



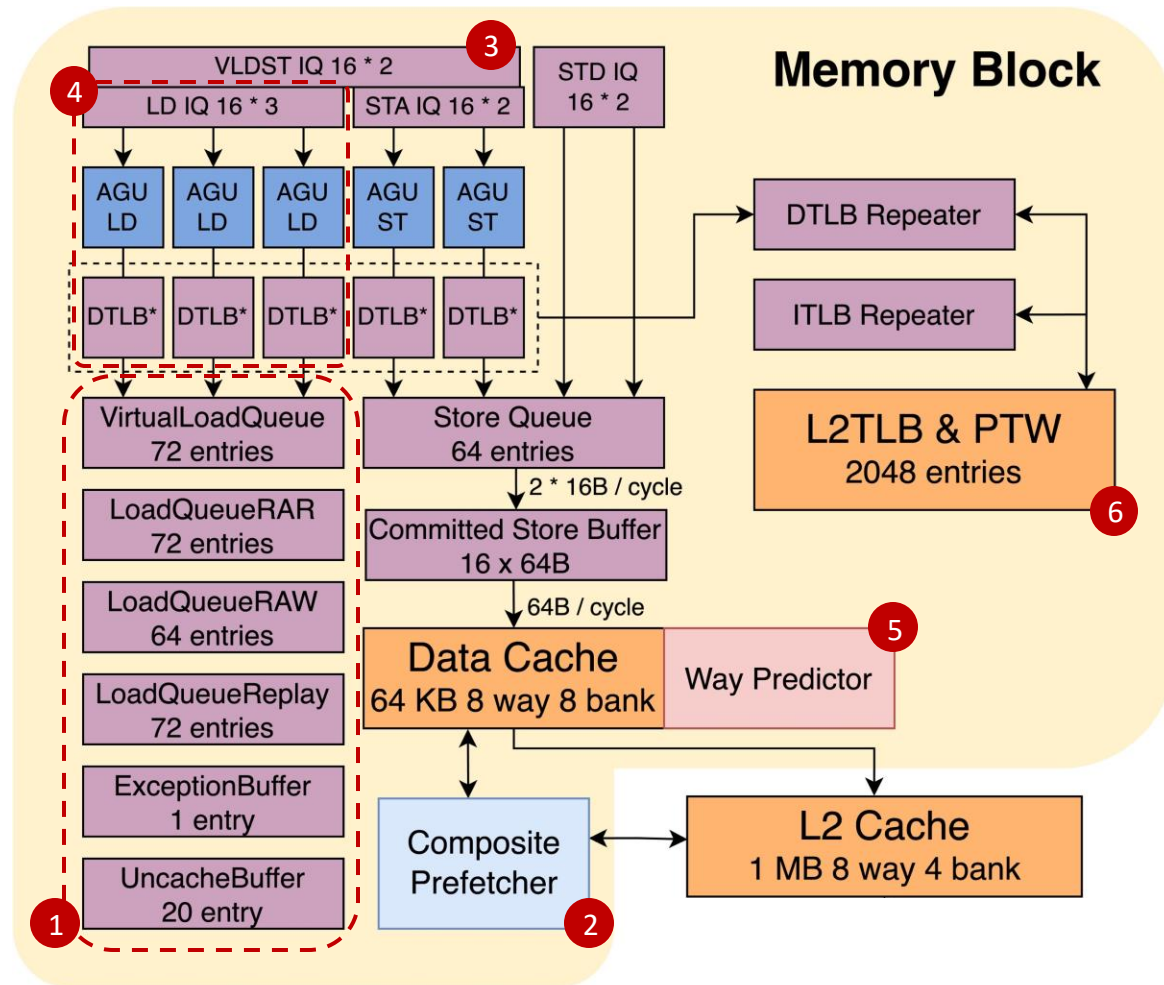
发射后读寄存器堆



昆明湖架构访存子系统

• 相对南湖架构的优化

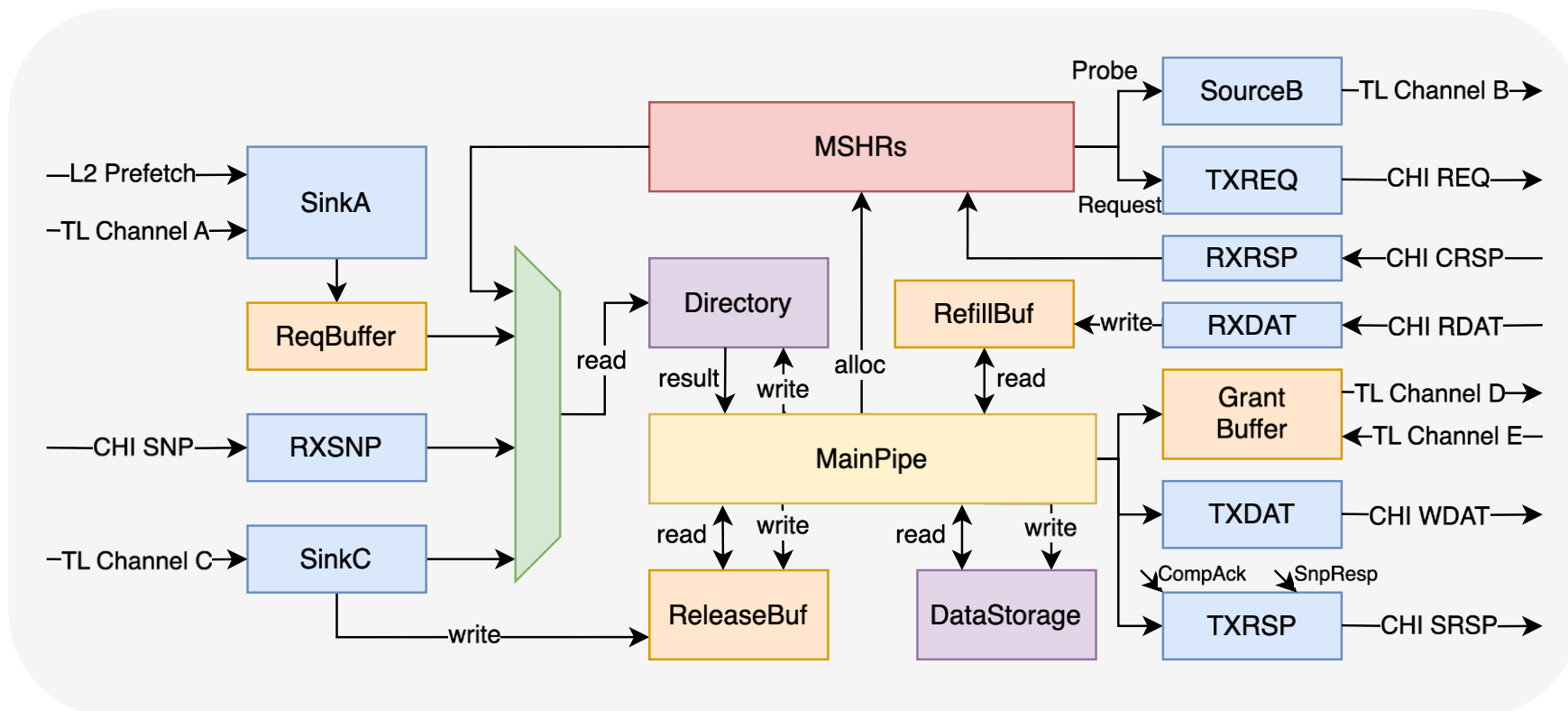
- ① 拆分 load queue
- ② 增加预取
- ③ 向量访存
- ④ 增大 load 带宽
- ⑤ 添加路预测器
- ⑥ 优化 MMU



昆明湖 CoupledL2 总览

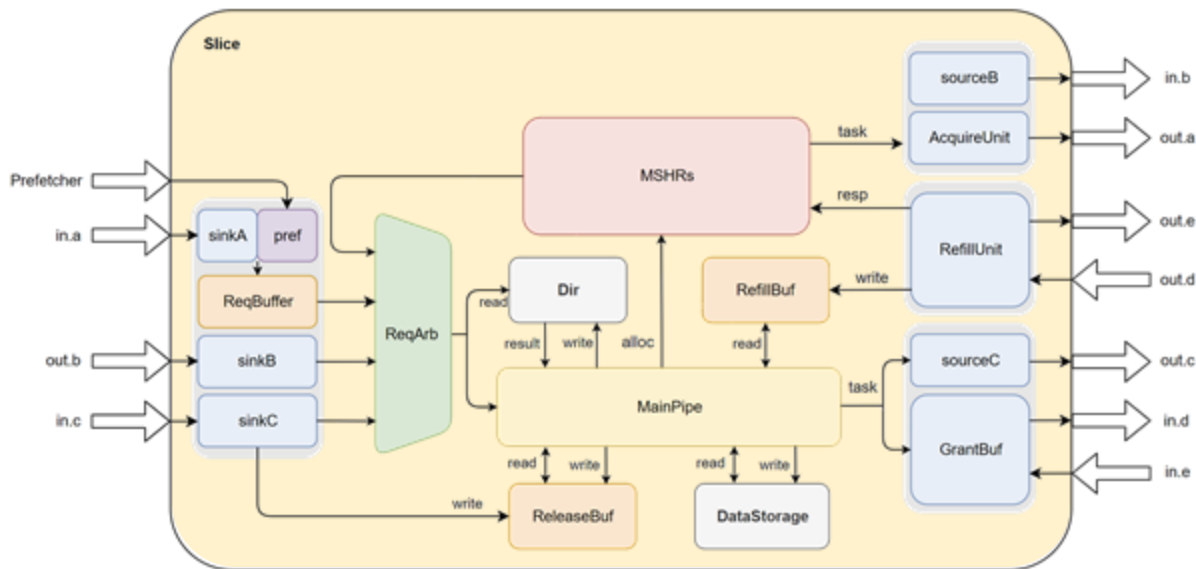
- 8 路组相联，最高 1MB 容量
- 取消和 ICache 的硬件一致性

- L1-L2 协同优化
- 同 set 请求并行
- Evict on refill

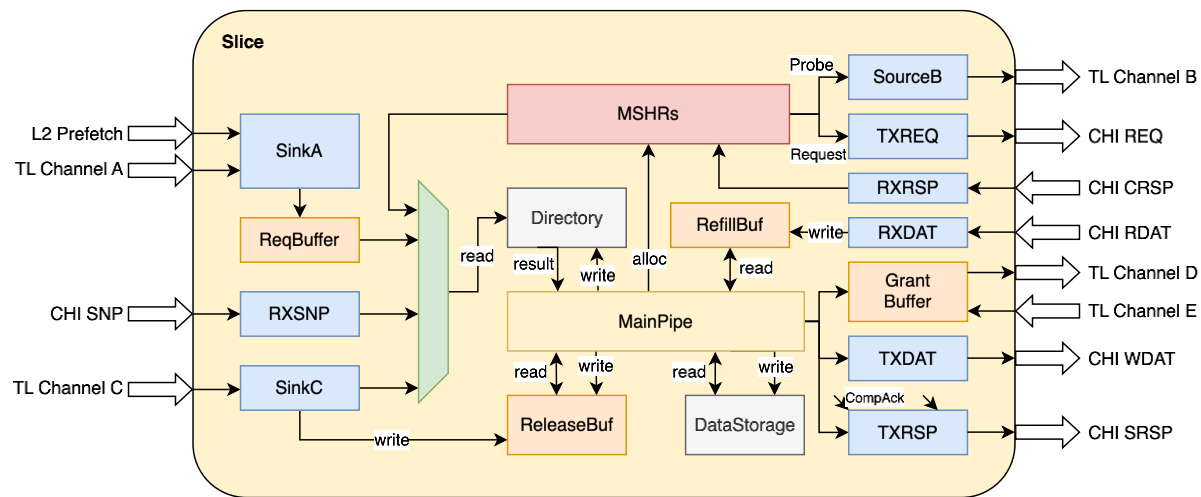


CoupledL2 对外协议可配置

- L1 <-> L2 采用 TileLink 1 协议
- L2 <-> LLC / NoC 可选择 CHI Issue B / CHI Issue E.b / TileLink 1 协议



L2 对 LLC 为 TileLink 1 协议

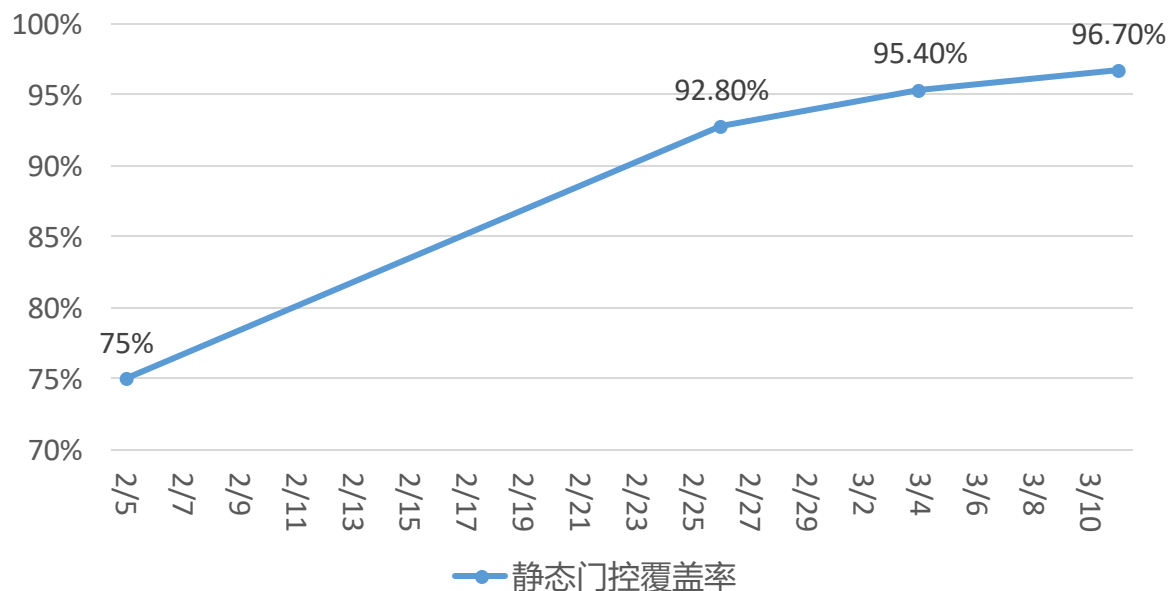


L2 对 LLC / NoC 为 CHI 协议

低功耗设计

- 昆明湖在关注性能、功能和面积的同时，也更注重**低功耗**方面的设计
 - 为寄存器添加**使能**信号 (RegNext -> RegEnable)
 - 对未使用的模块进行**时钟关断**
- 静态门控覆盖率显著提升 (~75% -> 96.7%)

后端流水线静态门控覆盖率



🏔️ 小结：昆明湖微架构的改进

- 更多的指令集扩展
- 低功耗的 ICache
- 发射后读寄存器堆
- 时序更优的 LSU 设计
- CHI 总线的 Coupled L2

| 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 |

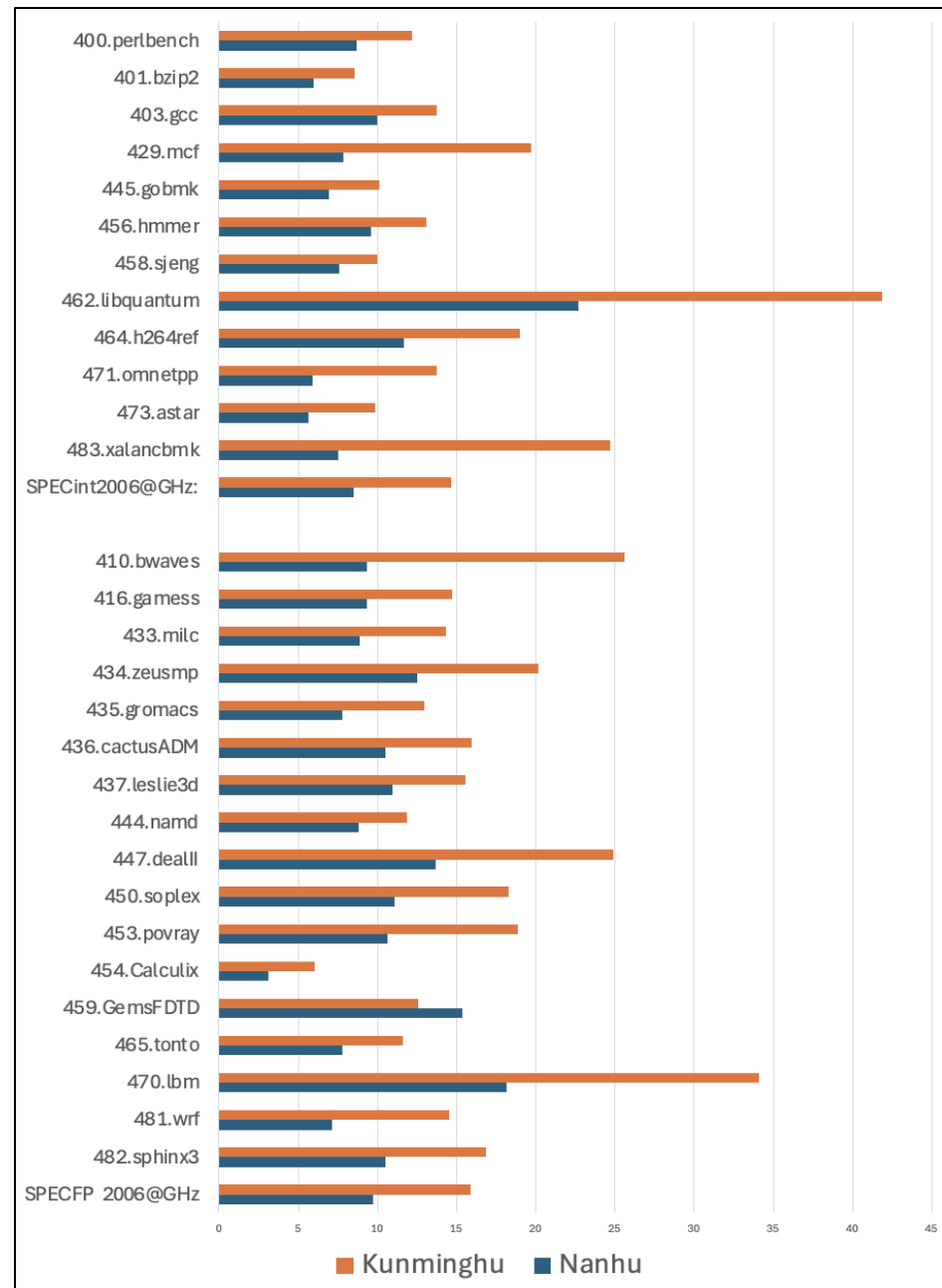
性能评估

- **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

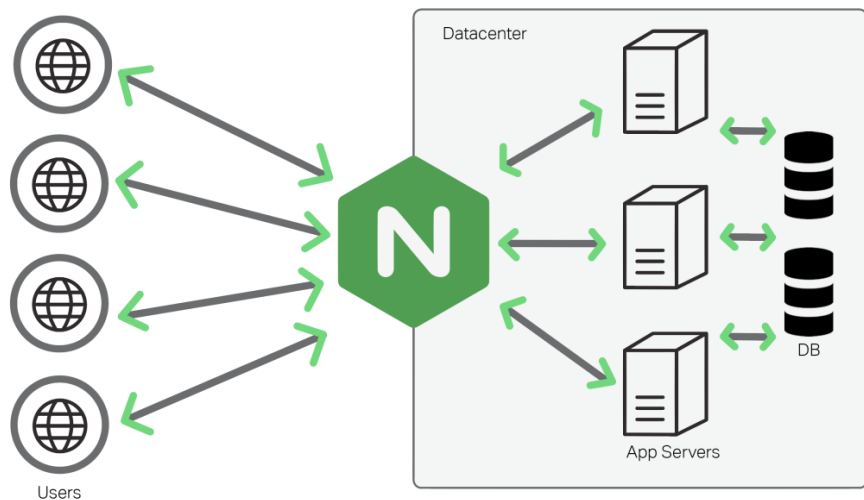
评估结果

| 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



香山作为产品，需要更高的可靠性



长时间运行**后台服务**程序
Web Server, etc.

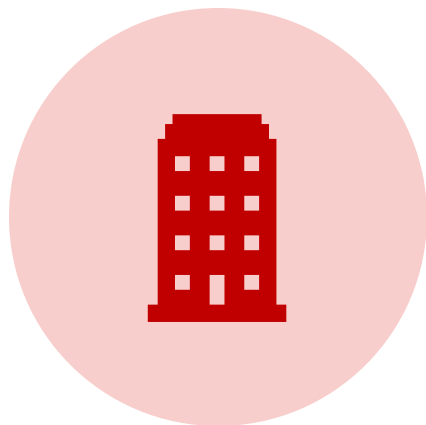


稳定运行**高负载**应用程序
Video Codec, Game, etc.

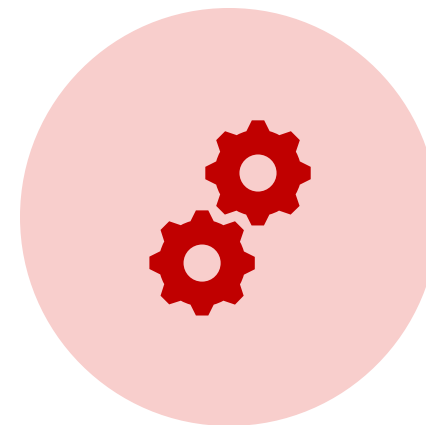
昆明湖产品化交付



工业级验证流程



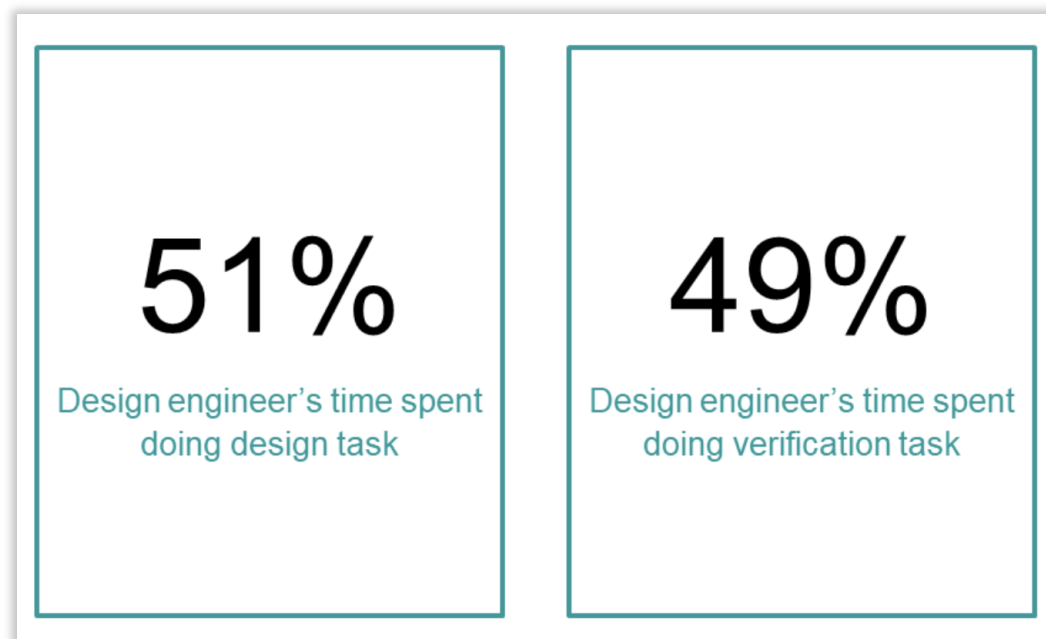
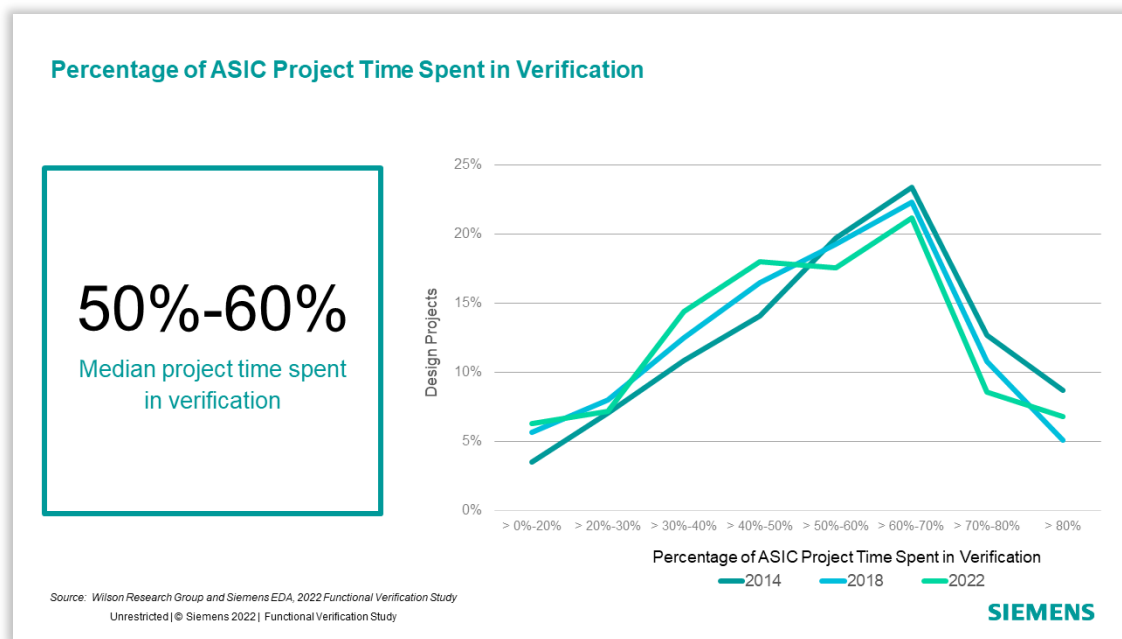
与业界深度合作



交付灵活可配置

验证在硬件开发流程中占据重要地位

- 项目平均需要花费 50%-60% 的时间在验证工作上
- 设计工程师也需要将 49% 的时间投入到验证工作中
- 若验证不充分，将会提高流片失败风险，浪费大量人力物力财力



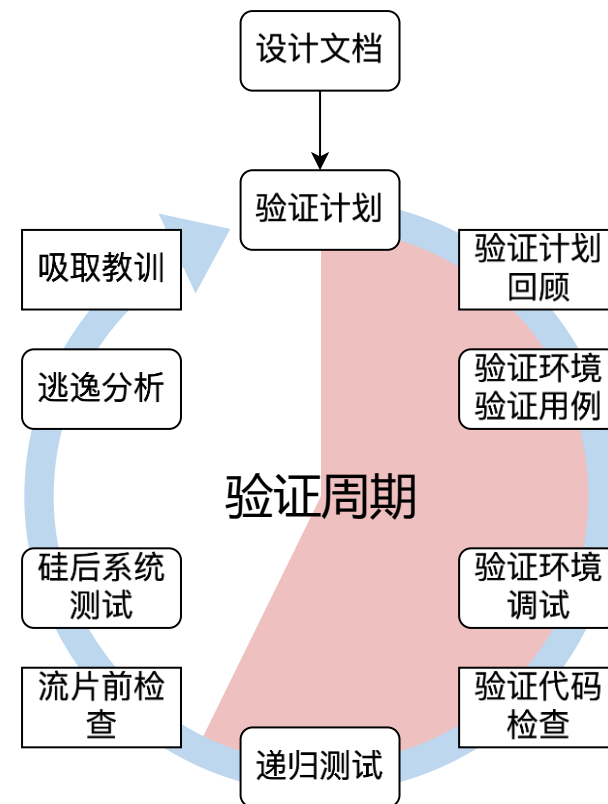
🏔️ 处理器芯片验证的基本流程

• 硅前验证

- 发现尽可能多的 Bug
- 保证设计实现和设计文档保持一致

• 硅后验证

- 对流片后的真实芯片进行验证
- 硅后验证出现的 Bug，需在硅前环境进行复现
- 积累经验，避免下一验证周期出现类似问题

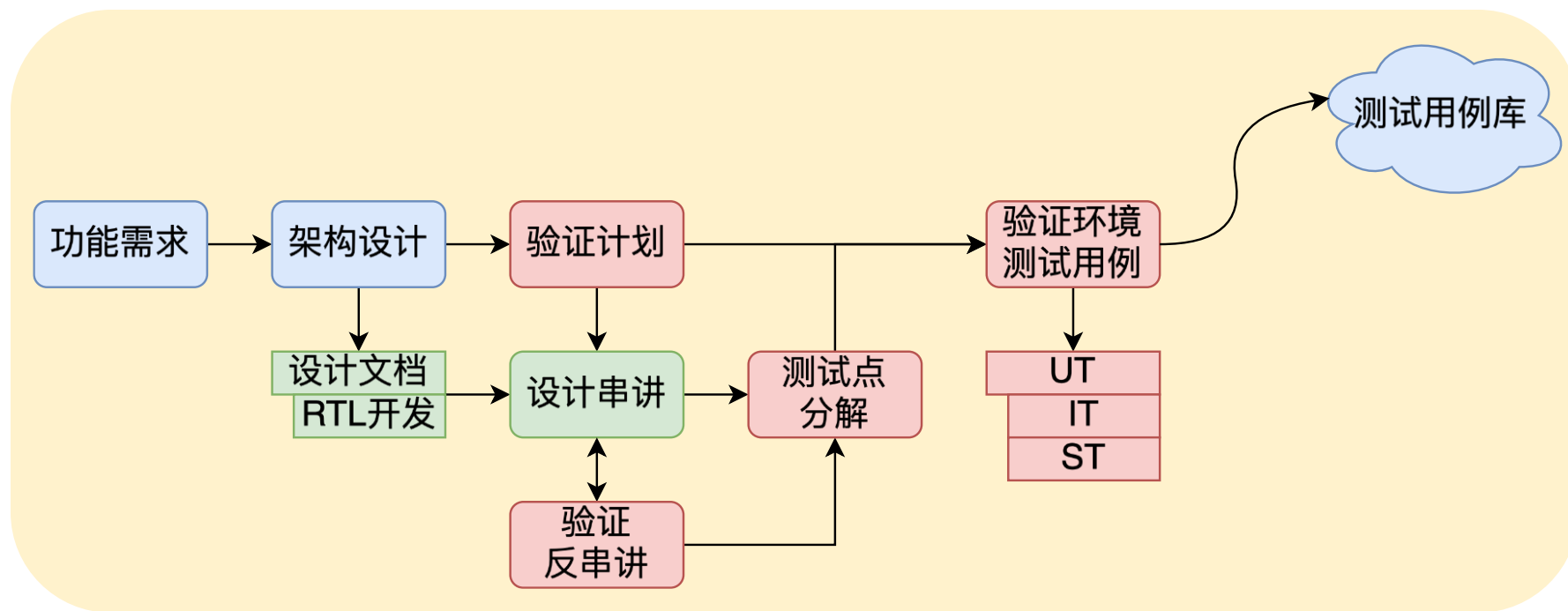


🏔️ 昆明湖架构硅前验证流程

- 富有经验的验证团队
 - 参与过南湖系列芯片的验证

- 验证计划、验证环境和用例

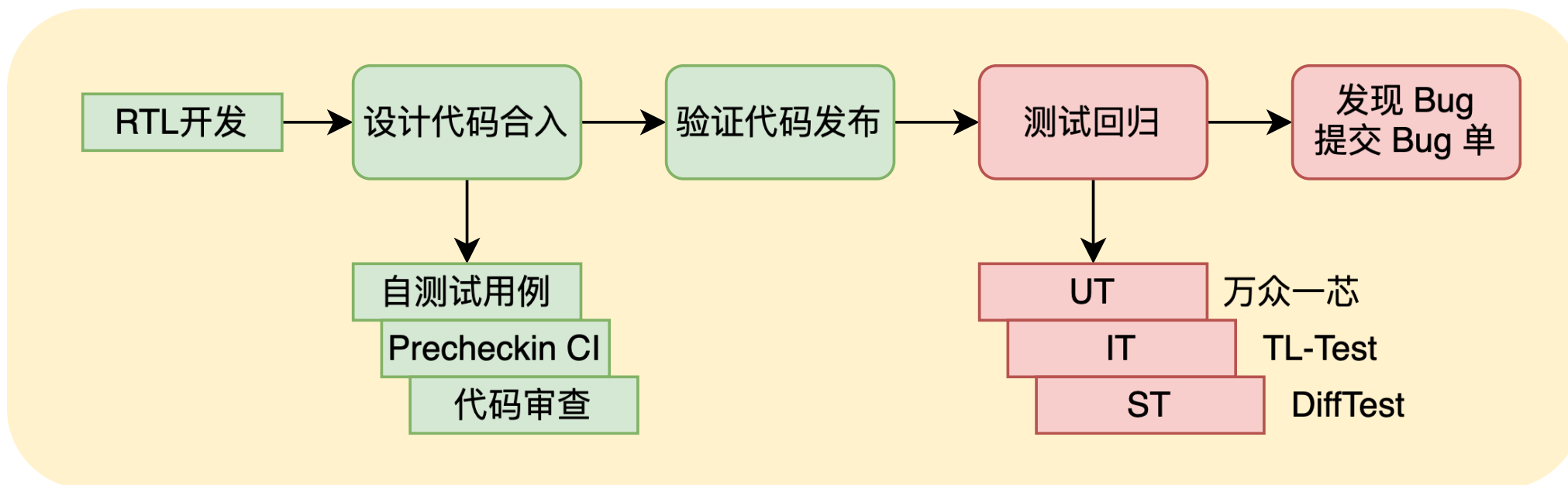
- 串讲和反串讲
- 测试点分解
- 搭建验证环境
- 形成测试用例
- 构建用例库



昆明湖架构硅前验证流程

• 代码发布、测试回归和缺陷提单

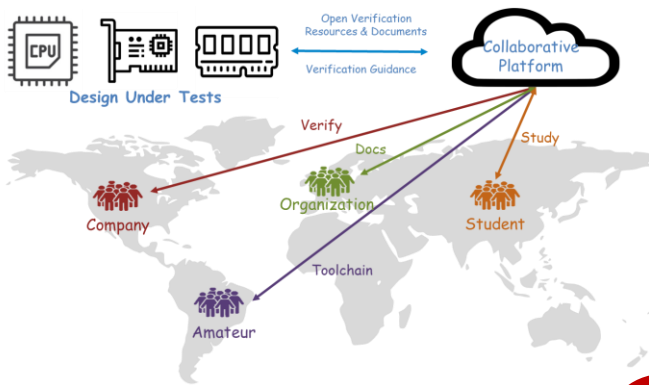
- RTL 代码开发合入需要经过设计自测、pre-checkin CI 和代码审查三道关口
- UT、IT、ST 不同粒度的验证
- 将香山敏捷开发验证平台引入传统验证工具和流程
- 发现 Bug 后，验证提缺陷反馈给设计



万众一芯 (UnityChip)

将软件开发者和硬件开发者共同引入硬件验证

吸引社区参与，发现问题
累积验证案例，减少重复劳动
开源验证资料，促进人才培养
帮助企业投入更多精力到设计



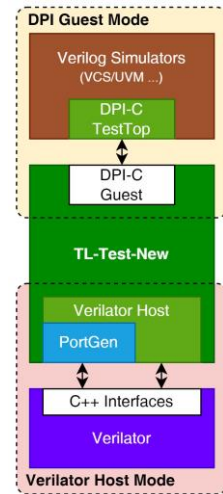
UT

IT

TL-Test-new

支持商业仿真环境的 TL-Test

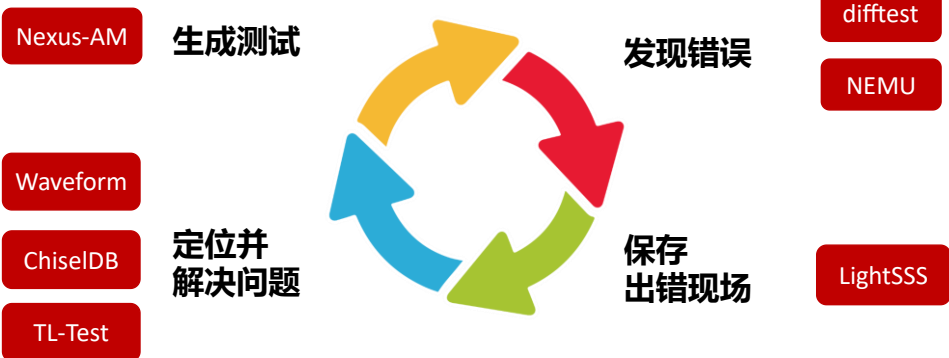
- 初期没有可用的开源 CHI 下游
 - 目前 AMBA CHI 仅有商业 RTL IP 和 VIP
- 保证原有 workflow 可用
 - 完善 ChiselDB 对商业仿真器的支持
- 同时支持开源、商业仿真环境
 - 一套代码，部署到多个不同定位的工作流
 - 更好的随机激励生成器 (Fuzzer)



开发流程永远需要基础设施的支持

敏捷功能验证

ST



欢迎关注我们的更多报告 / Tutorial

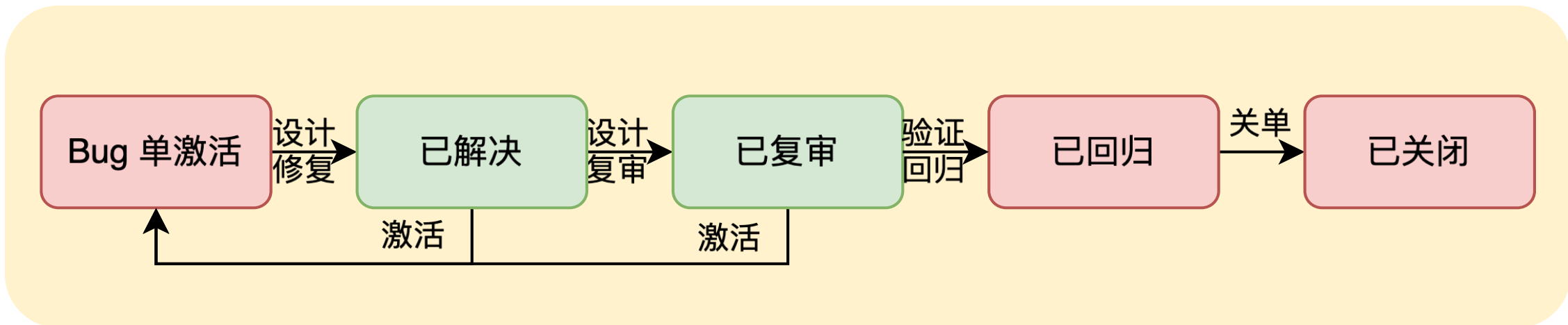
10:20 – 10:40 @ 主会场 B
万众一芯：基于开源众包芯片验证的探索与实践

10:00 – 10:20 @ 主会场 B
面向服务器的香山处理器多核解决方案

8月22日 下午 @ 紫荆厅 1
香山开发者大会 (含 Tutorial)

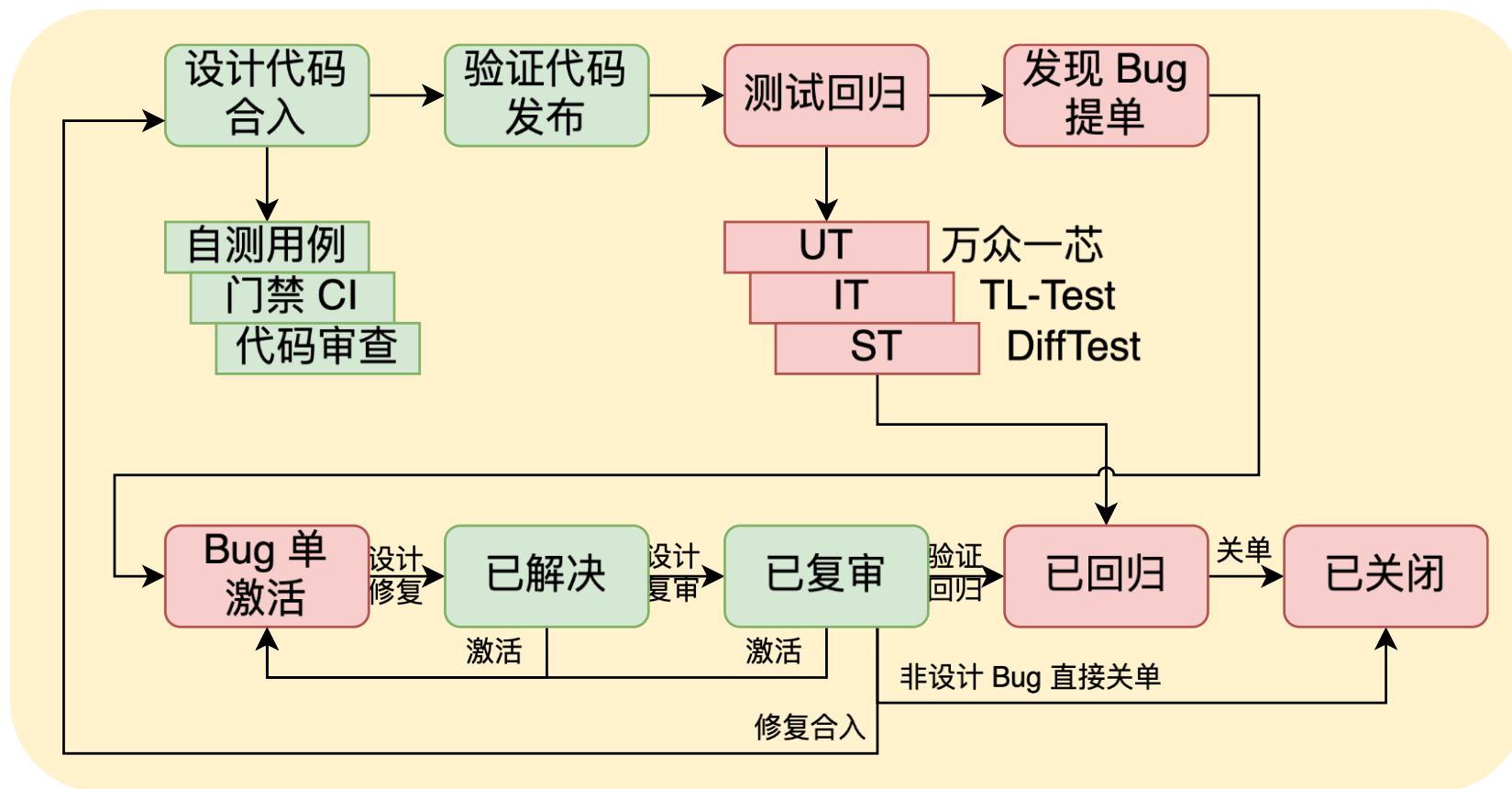
🌄 昆明湖架构硅前验证流程

- 缺陷解决、复审和关单
 - 五步关单
 - 设计修复后需要小组组长进行**设计复审**



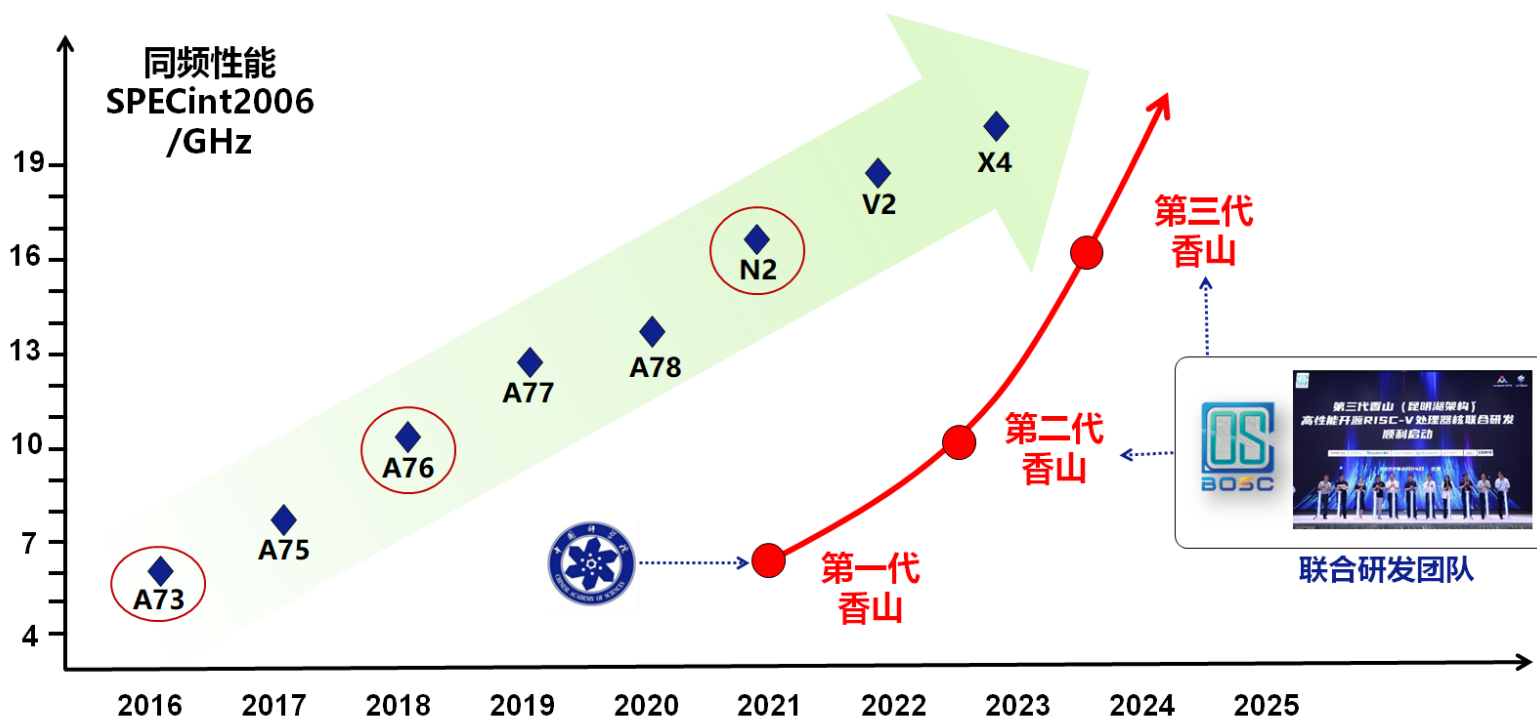
🏔️ 昆明湖架构硅前验证流程

- 缺陷解决、复审和关单
 - 复审后的代码进入**代码合入流程**
 - 合入后**验证回归**并关单



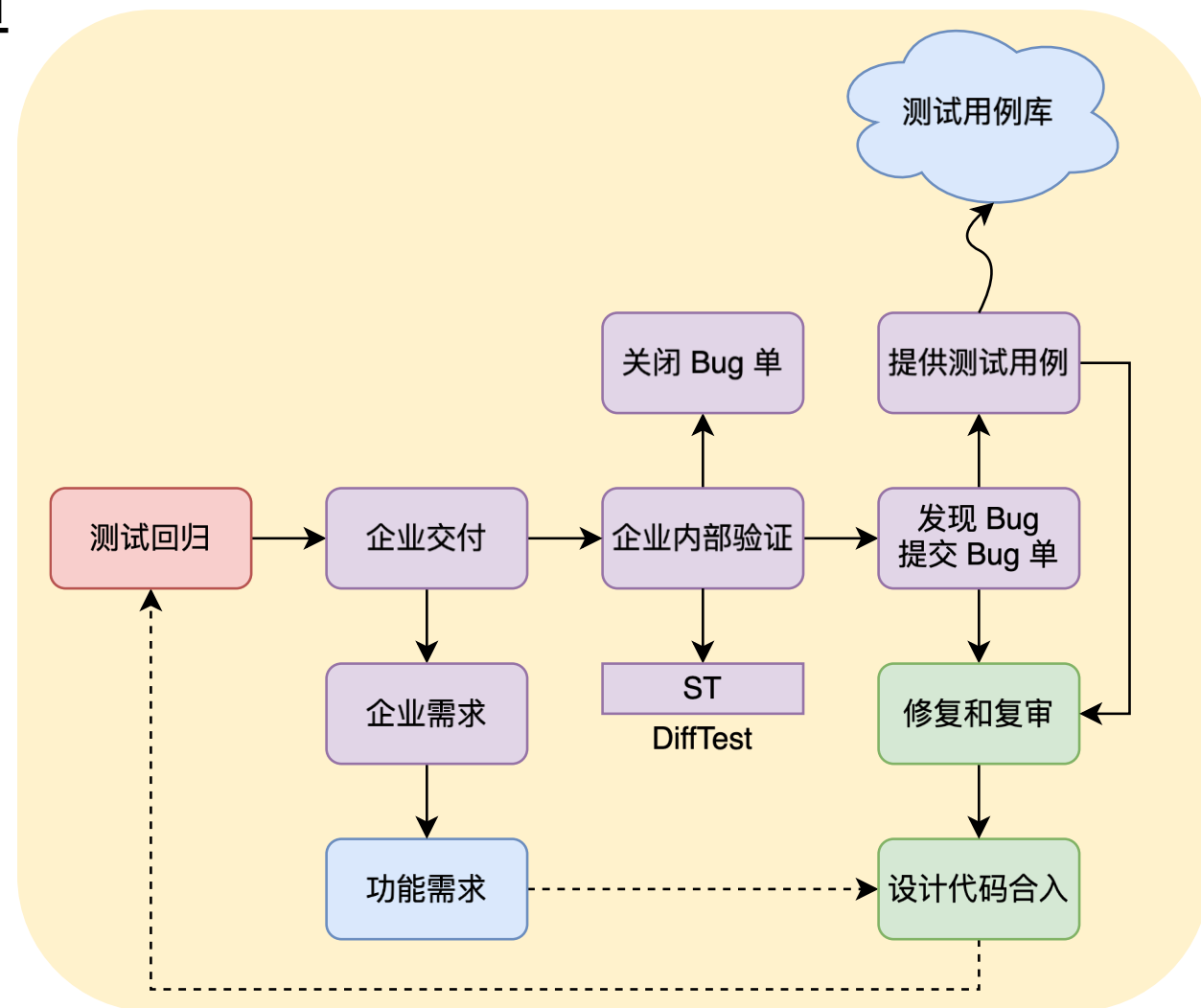
联合开发 联合验证

- 从第二代香山开始，由“开芯院”联合企业组建**联合研发团队**
 - 香山听取来自企业界的一线需求
 - 企业**深度参与**昆明湖架构的验证流程



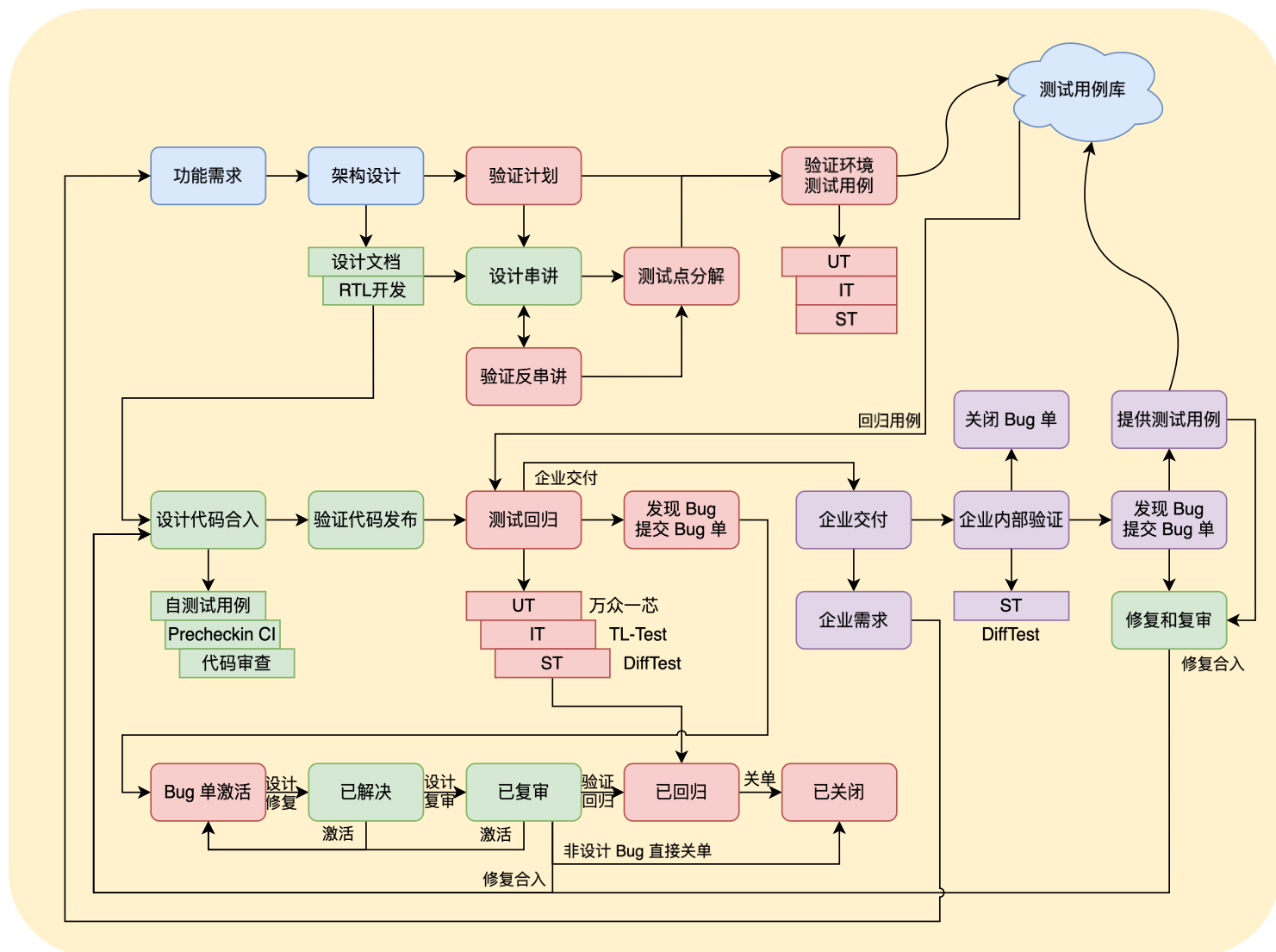
企业参与昆明湖架构硅前验证流程

- 交付企业、内部验证和缺陷提单
 - 测试回归后，代码交付给企业
 - 企业内部进行验证
 - 推荐企业验证流程集成 DiffTest 框架
 - 发现缺陷后提交缺陷单
- 触发缺陷的用例加入用例库
 - 确保该缺陷后续能被覆盖



小结：昆明湖架构的硅前验证

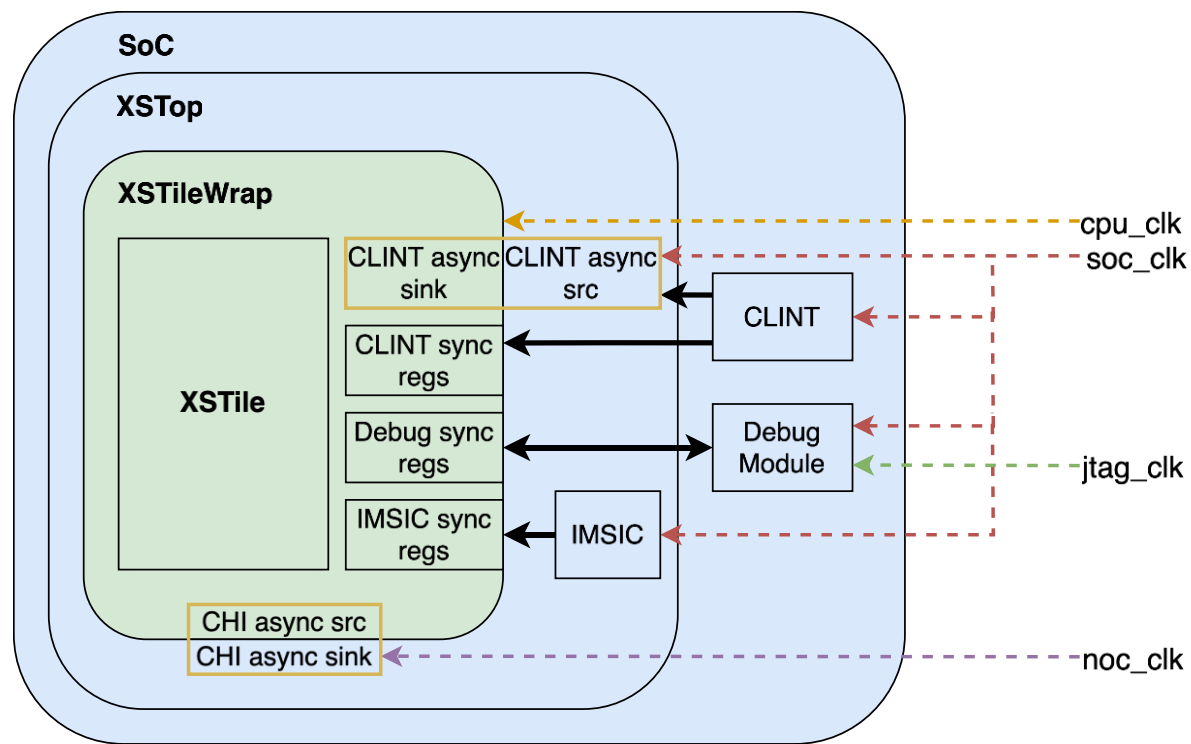
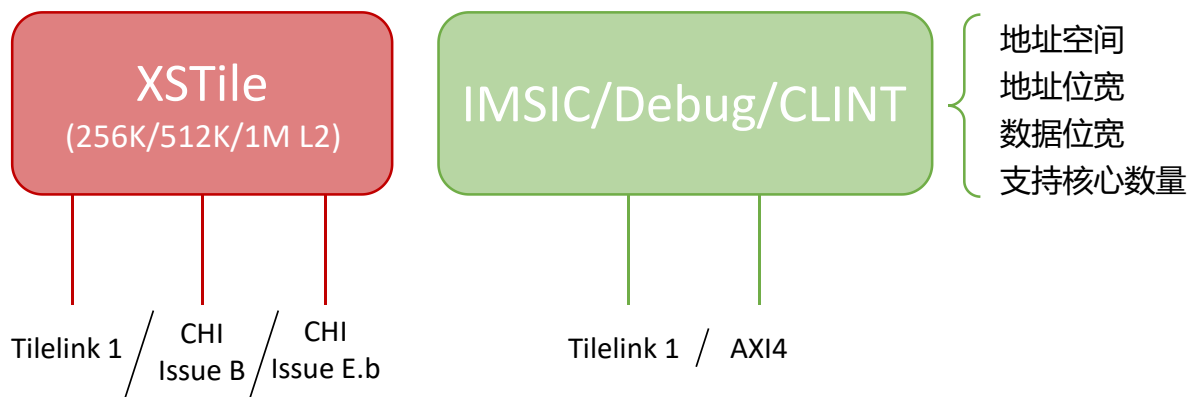
- 富有经验的**验证团队**
- 完整的**硅前验证流程**
- **合入规范和测试回归**
- **敏捷开发和敏捷验证**
- **企业联合开发深度参与**



丰富可配置的交付选项

- 香山昆明湖架构的用户众多，**灵活可配置**的交付选项可以满足不同的定制化需求

- ① 总线协议、地址空间、位宽、核心数量、核心参数可配
- ② 高低电压域分离、跨模块信号异步处理，可灵活进行 SoC 设计



1

2

高电压域

低电压域

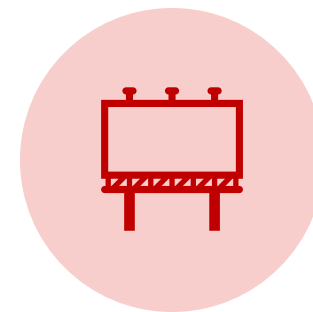
- CPU 时钟
- SoC 时钟
- JTAG 时钟
- NoC 时钟

总结：香山昆明湖架构的演进

- PPA 的全面优化
 - SPEC CPU 2006 ~45分 @ 3GHz
 - 低功耗设计
 - 针对 PPA 优化的架构重构
- 更完善的处理器功能
 - 虚拟化、向量、Sv48、RVA23
- 工业级的验证流程
 - 敏捷验证
 - 企业联合
- 灵活的交付选项



功能与性能



产品化交付

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



香山技术讨论 QQ 群



香山技术讨论微信四群