



香山处理器访存、缓存架构 解决方案与演进

丁昊楠¹ 刘威丁²

¹中国科学院计算技术研究所 ²北京开源芯片研究院

2025 年 7 月 16 日



第一部分 访存

Svpbmt: 基于页表的内存属性

- 实现 Svpbmt 的页表属性与总线适配

- 越过 L1 & L2 Cache

直接到达 PoS (Point of Serialization)

- NC**: Non-cacheable, RVWMO

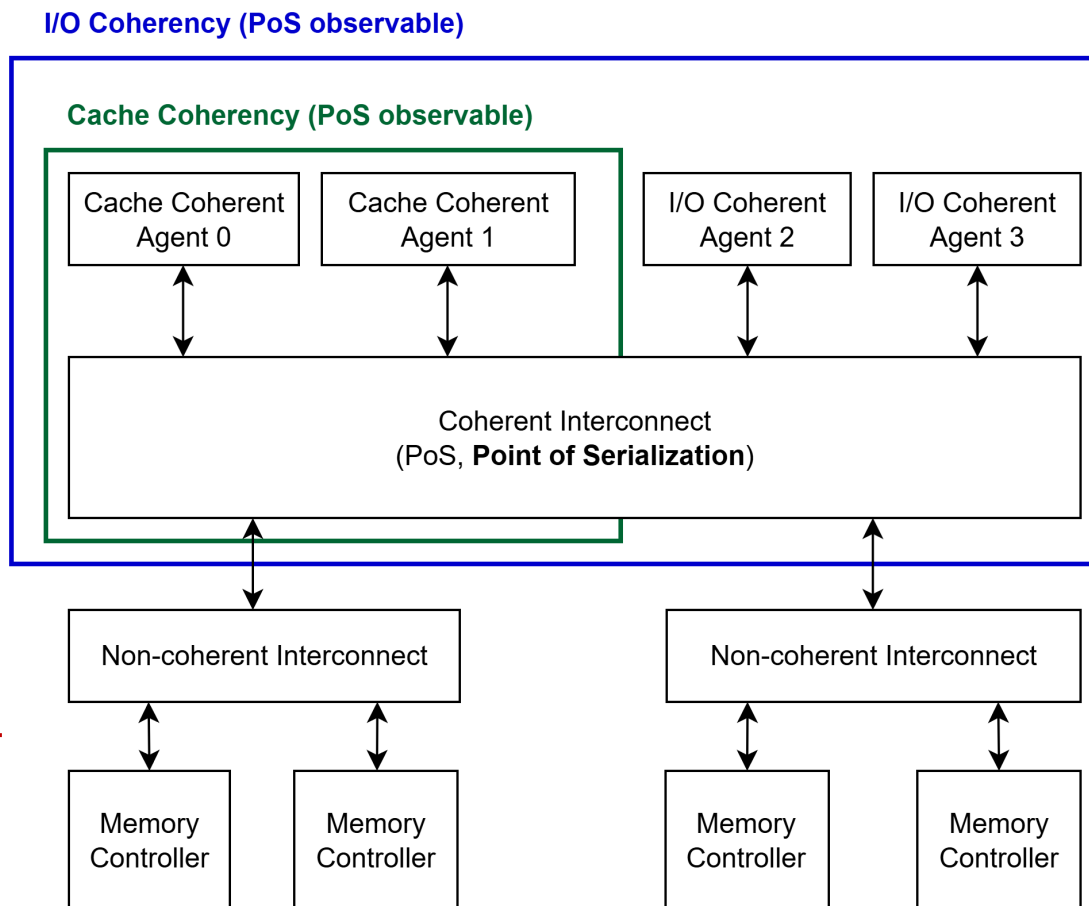
可合并, 可并行, 可推测

- IO**: Non-cacheable

强制顺序

- 意义

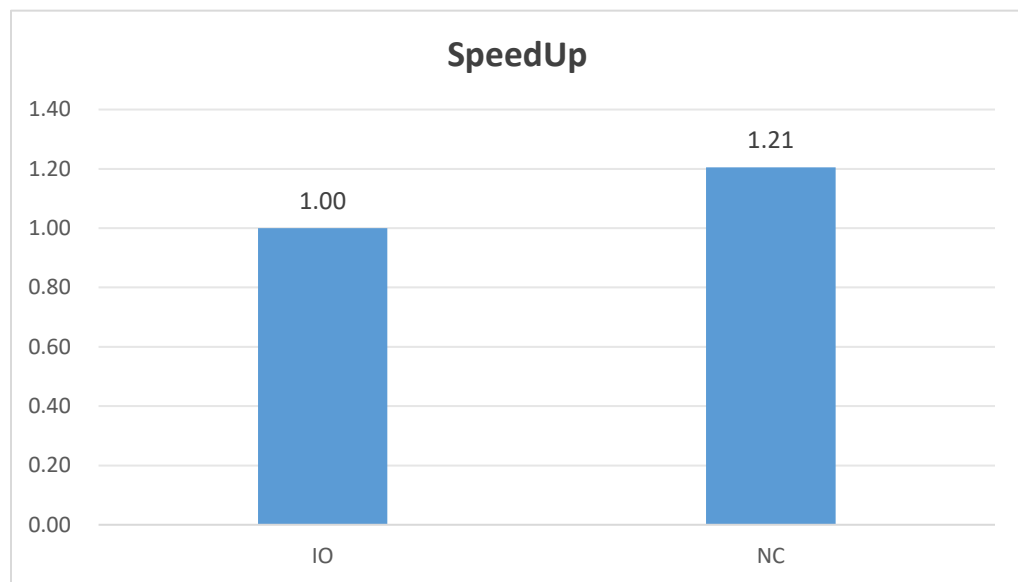
- 非 Memory 属性地址**不再只能非推测顺序执行**
 - 软件定义副作用影响范围
 - 高带宽需求的 I/O Coherent 设备
 - 更好地适配不同的 SoC 需求, 逐渐弥补 RISC-V 对 SoC 理解的空缺



Svpbmt: 基于页表的内存属性

• ① NC 推测执行

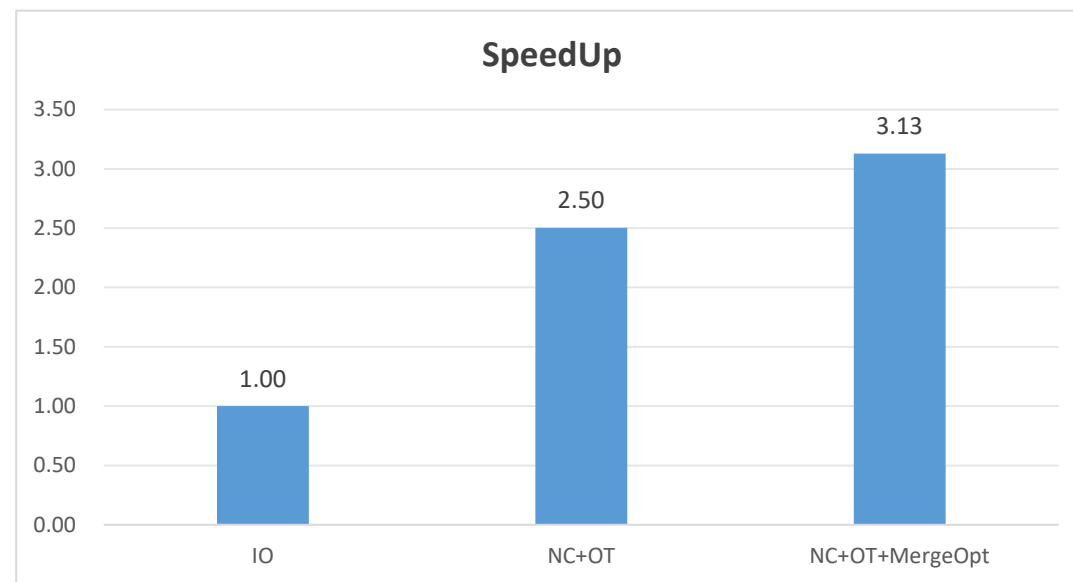
- NC 请求 RVWMO
- 推测执行 NC 请求相比顺序执行 IO 请求获得 **1.21** 加速比



自建访存密集型测试收益

• ② NC 请求合并

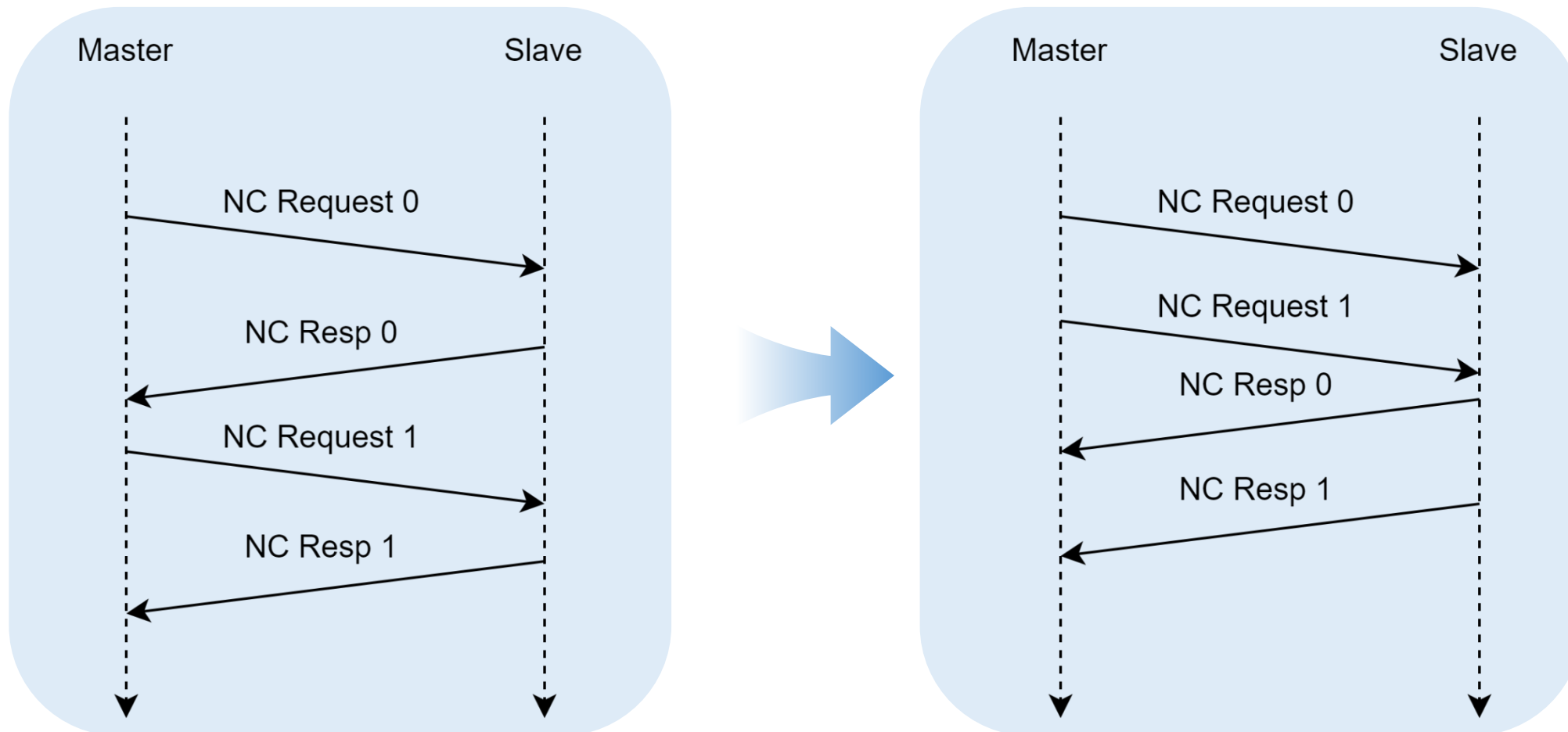
- 8-Byte 对齐的同属性同地址的请求入队合并
- 请求合并与 Outstanding 协同优化获得 **3.13** 加速比



自建访存密集型测试收益

Svpbmt: 基于页表的内存属性

- ③ NC Outstanding
 - 最大 4 个 Outstanding NC 请求



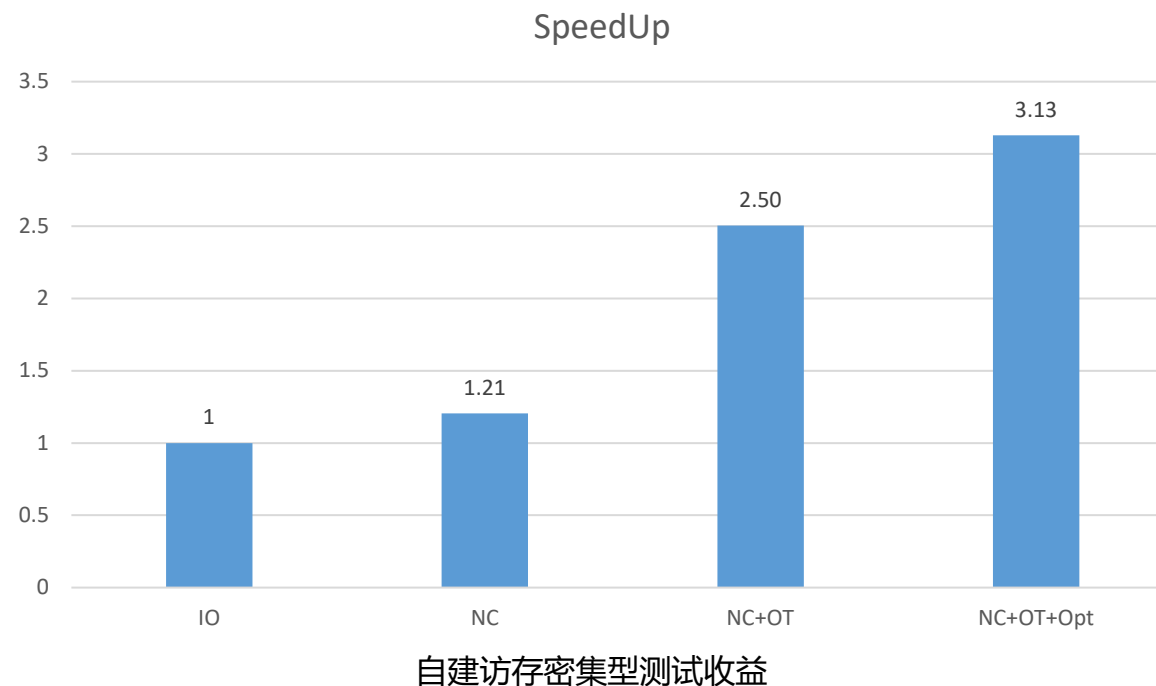
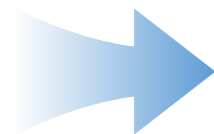
Svpbmt: 基于页表的内存属性

• 昆明湖 Svpbmt 优化点总结

NC 推测执行

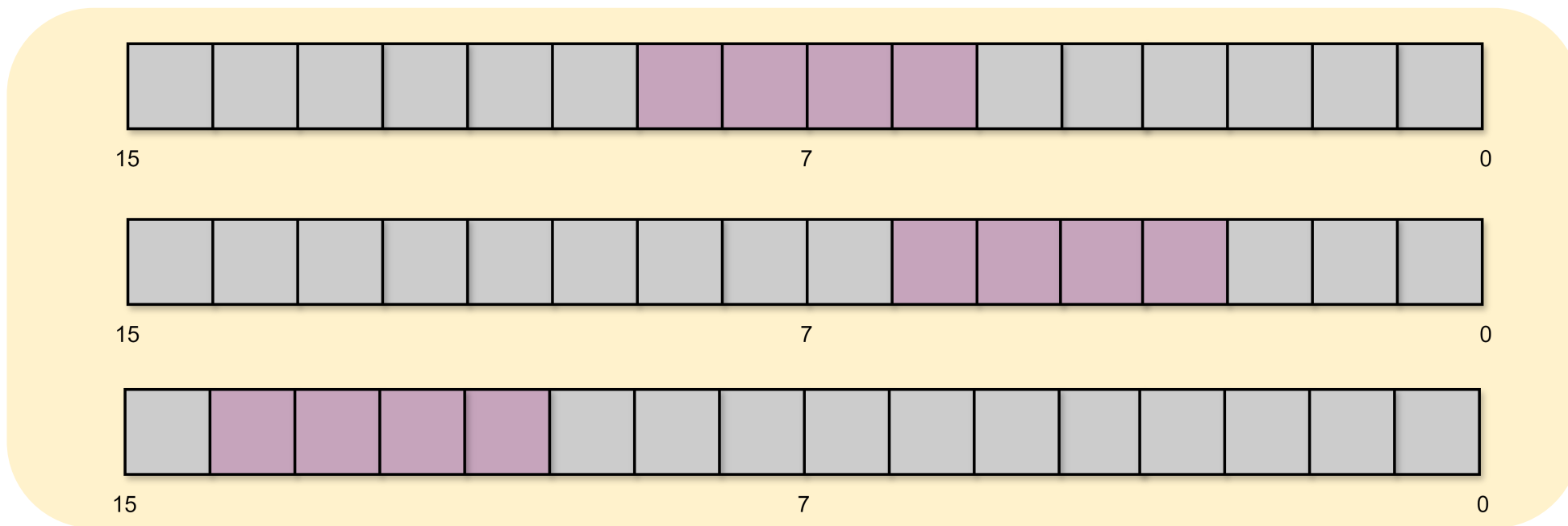
NC 合并优化

NC Outstanding



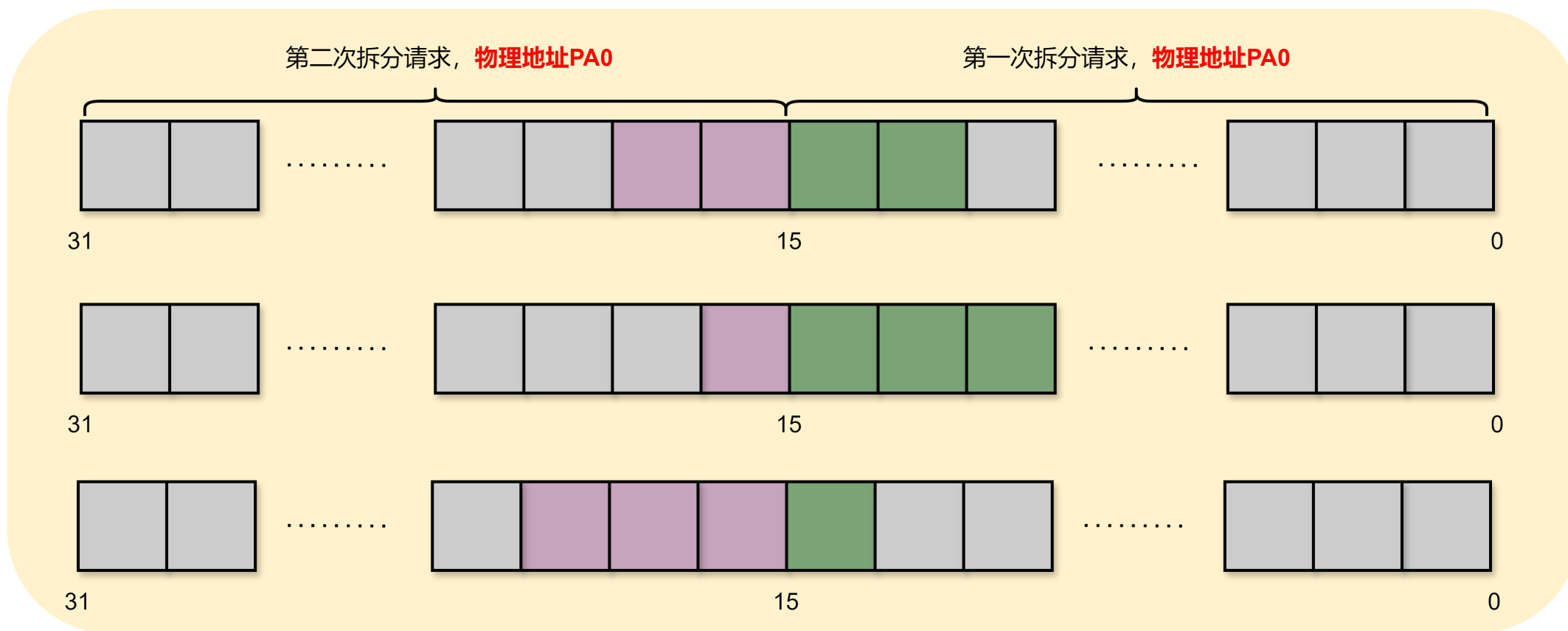
Zicclsm: 非对齐Load/Store访问

- 支持标量与向量非对齐访问
- **目标：尽量高效的拆分+支持向量非对齐**
- ① 16 Byte 内非对齐请求处理



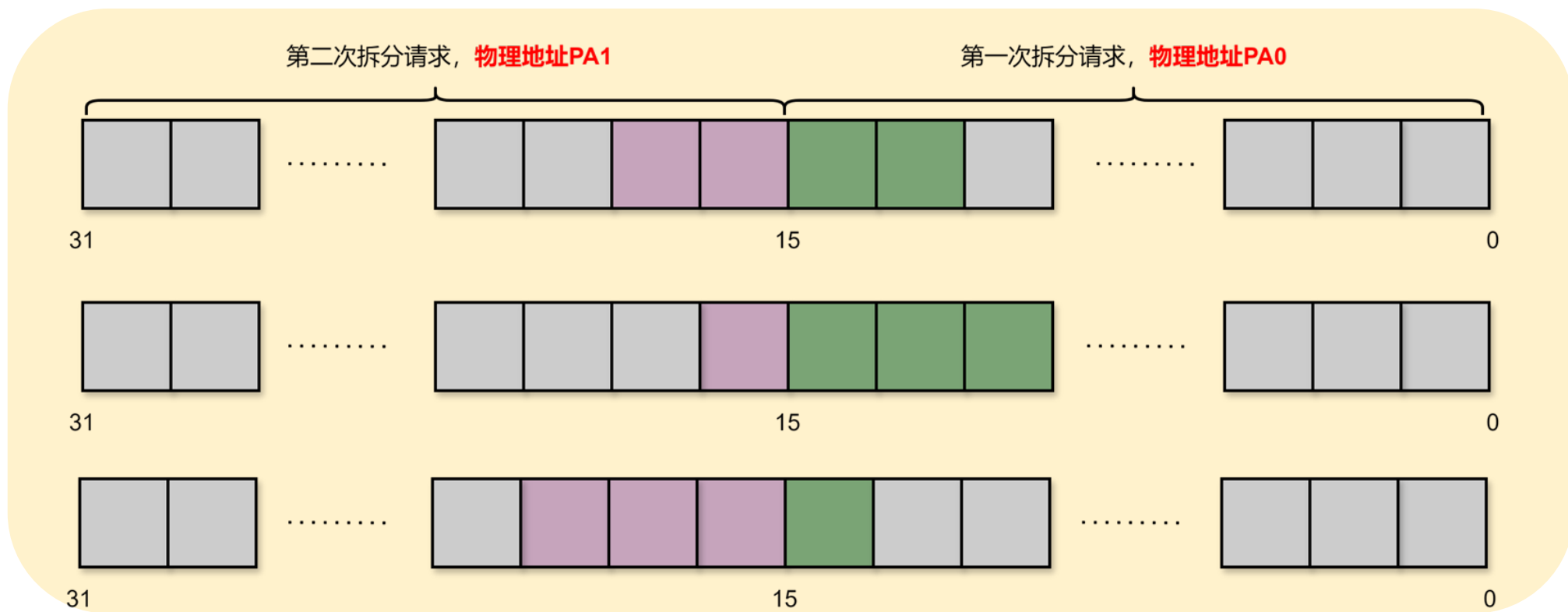
Zicclsm: 非对齐Load/Store访问

- ① 16 Byte 内非对齐请求处理
- ③ 跨 16 Byte 非对齐请求拆分



Zicclsm: 非对齐Load/Store访问

- ① 16 Byte 内非对齐请求处理
- ③ 跨 16 Byte 非对齐请求拆分
- ② 跨页的非对齐请求拆分





第二部分 缓存

香山缓存系统路线图

开源高性能 RISC-V 处理器	HuanCun 2022	性能 CoupledL2 2023	规模 TL2CHI CoupledL2 2024-2025	PPA KunminghuV3 Cache IP 2026
	Nanhu	Kunminghu	Kunminghu V2	kunminghu V3
频率	2GHz	↑ 3GHz	3GHz	↑ 3GHz +
总线宽度	32B up + 32B down	32B up + 32B down	32B up + 32B down	32B up + 32B down
总线协议	TileLink	TileLink	TileLink + CHI	↑ Private + CHI
一致性协议	MESI-ish	MESI-ish	MESI-ish	↑ MOESI-ish
L1 一致性	I+D	D	D	↑ D/I+D
包含策略	NINE	Inclusive	Inclusive	Inclusive, NINE for I
L3 与互联	-	HuanCun	OpenLLC, or Any CHI NoC	TBD, or Any CHI NoC
扩展性	~ 8 cores	~ 8 cores	↑ 64+ cores	64+ cores
事务并行度	14 / slice	16 / slice	16 / slice	↑ 32+ / slice
一致性热插拔	-	-	↑ 支持	支持
原子指令下放	-	-	-	↑ 支持
L3 预取节流	-	-	-	↑ 支持



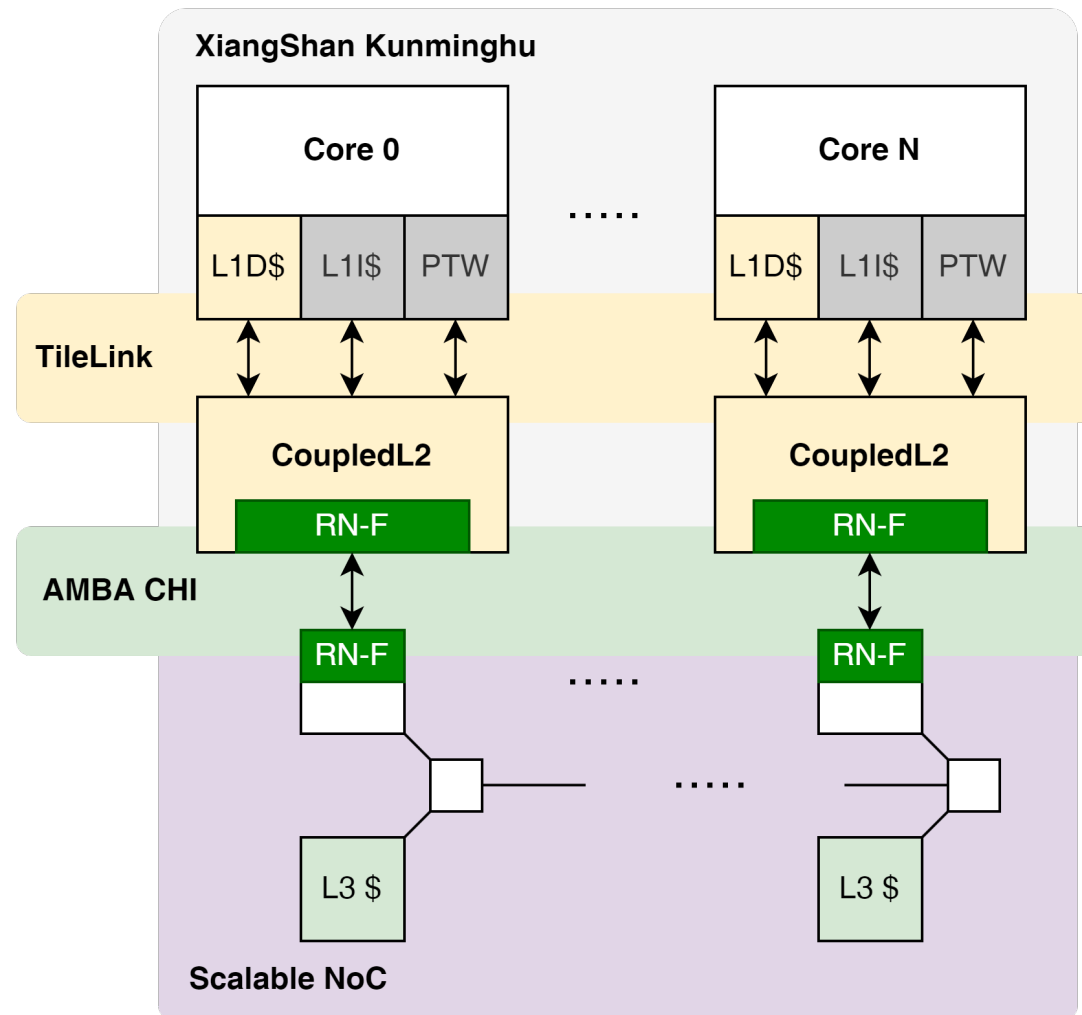
CoupledL2: 适用于香山昆明湖的高可扩展一致性 L2 缓存

• 规格

- **基于目录**的一致性维护
- Inclusive 包含策略
- 总线协议: **TileLink 1.8.1 (custom)**
(KMHv2 TL2CHI) **CHI Issue B/C/E**

• 特性

- **非阻塞的缓存流水线**
- 替换算法: Random/PLRU/DDRIP
- 预取算法: Stream/Stride/
Best-Offset/Temporal
- 与 L1 Cache 的耦合优化
- 在 NoC 上可扩展到 **64+ cores**





CoupledL2: 适用于香山昆明湖的高可扩展一致性 L2 缓存

- 非阻塞流水线，尽可能保证不同请求之间的并行度

- 支持 Multi-Cycle Path，以支持更大的 L2 SRAM 规格

- 经过充分测试、验证的一致性协议转换

- TileLink \leftrightarrow AMBA CHI

- 支持的总线协议

- CHI Issue E

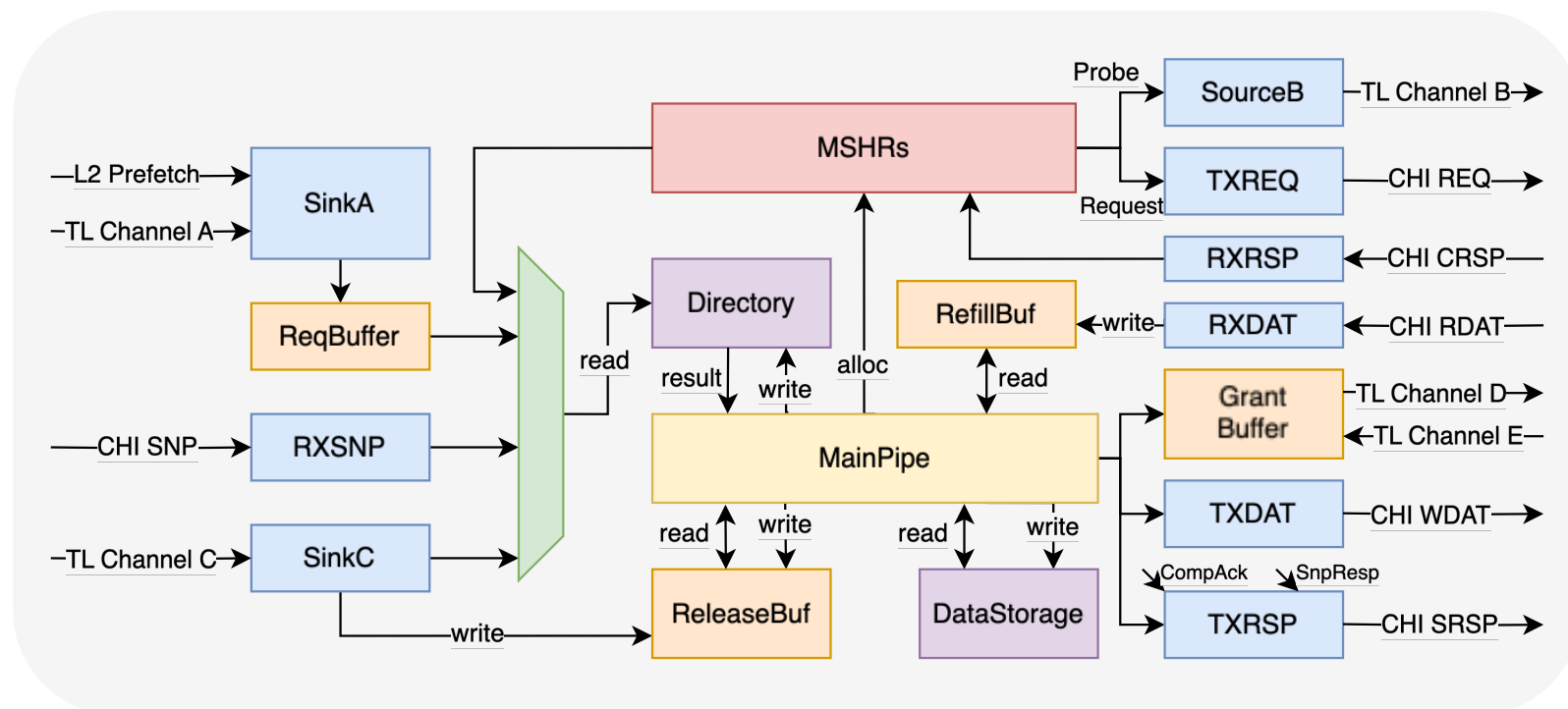
- CHI Issue C

- CHI Issue B

提供 SoC 支持的多样性

- TileLink

以连接 L1 Cache 等





CoupledL2: 适用于香山昆明湖的高可扩展一致性 L2 缓存

- **新特性: 支持一致性热插拔 (单核关断)**

- 支持维护一致性的热插拔, 以支持单个核心的电源关断
 - 在服务器级 IP 平台中非常重要的特性
- 以其为基础可支持平台层级的**能耗管理、热备份、故障隔离**
- **支持基于 CHI 标准的一致性域热介入、热撤离**
 - 需要小心地处理过程中的状态竞争, 保证总线协议不丢失消息、不出现死锁
- 在处理器下电前保证清空缓存, 写回所有数据
- **在 L2 实现了电源管理状态机与交互接口**
 - 支持 CSR 控制的一致性撤离
 - 支持中断唤醒

→ **与合作单位共同完成单核关断的测试、验证与设计反馈**



问题：通用 TileLink 协议功能不足、耦合性不足

特性	TileLink	AMBA CHI
一致性状态	类 MESI	类 MOESI
流控	Ready-valid	Credit
拓扑感知	无	有
QoS	无	有
内存序支持	未定义	有(支持多种)
电源、复位、时钟管理	无	有
Cache-to-cache Direct Transfer	未定义	有
Cache-to-memory Direct Transfer	未定义	有
CMO	无	有
Exclusive	无	有
Atomic	有	有

支持复杂总线拓扑（如 Mesh）
的必要标准化定义

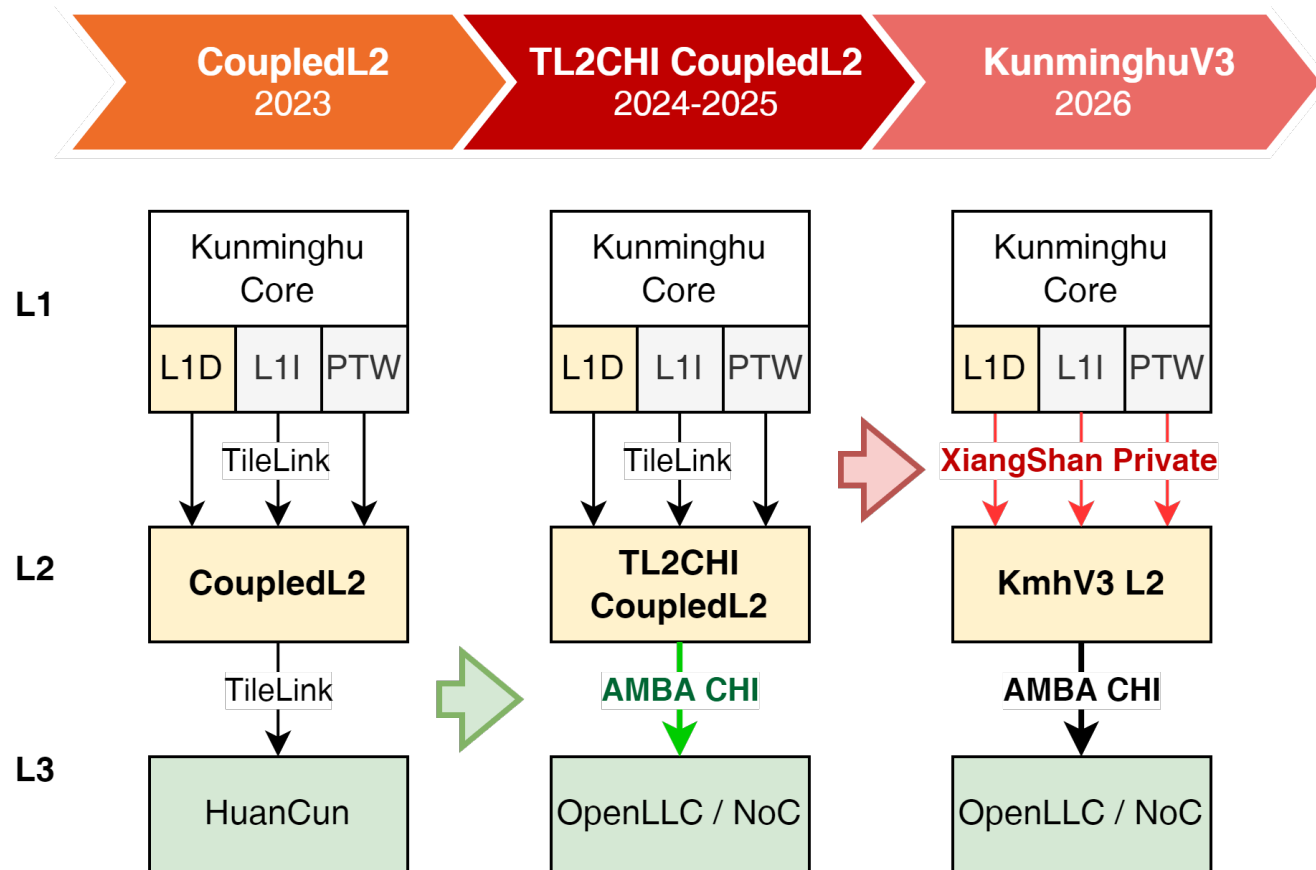
支持复杂总线性能优化
的必要附加功能定义

支持一致性操作性能优化
的必要附加功能定义



问题：通用 TileLink 协议功能不足、耦合性不足

- **我们已经在修改 TileLink 协议**
 - 以支持缓存维护 (CMO) 指令
 - 以达成 L1-L2 之间的耦合优化
- **一致性协议转换存在设计负担**
 - TileLink <-> AMBA CHI
 - 需要更加复杂的验证手段
 - 协议的设计语言不同
 - 事务优先级、通道优先级规则不同
 - 导致 L2 事务流程设计偏向保守
 - LSU-L1-L2 之间的耦合协同远远不足
 - L1-L2 严重依赖 Rocket-chip Diplomacy
 - Rocket-chip 组件**不总是那么美好**
- **→ 下一代 L1-L2 间将采用私有协议**





反思：RISC-V CBO 指令作用范围包含非一致性设备

• 软件角度

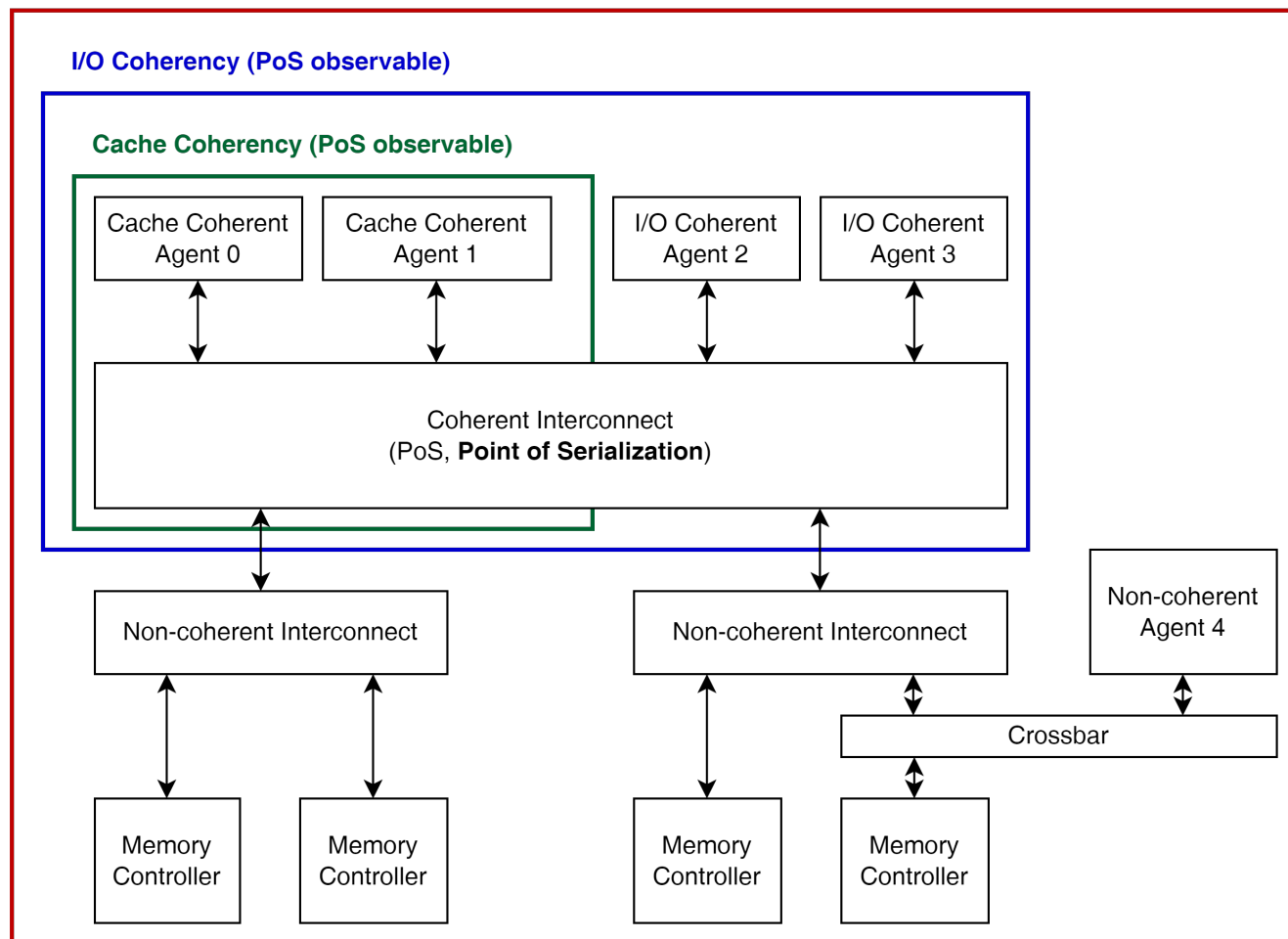
- 作用范围过大的 CBO 指令
影响软件性能

• 硬件角度

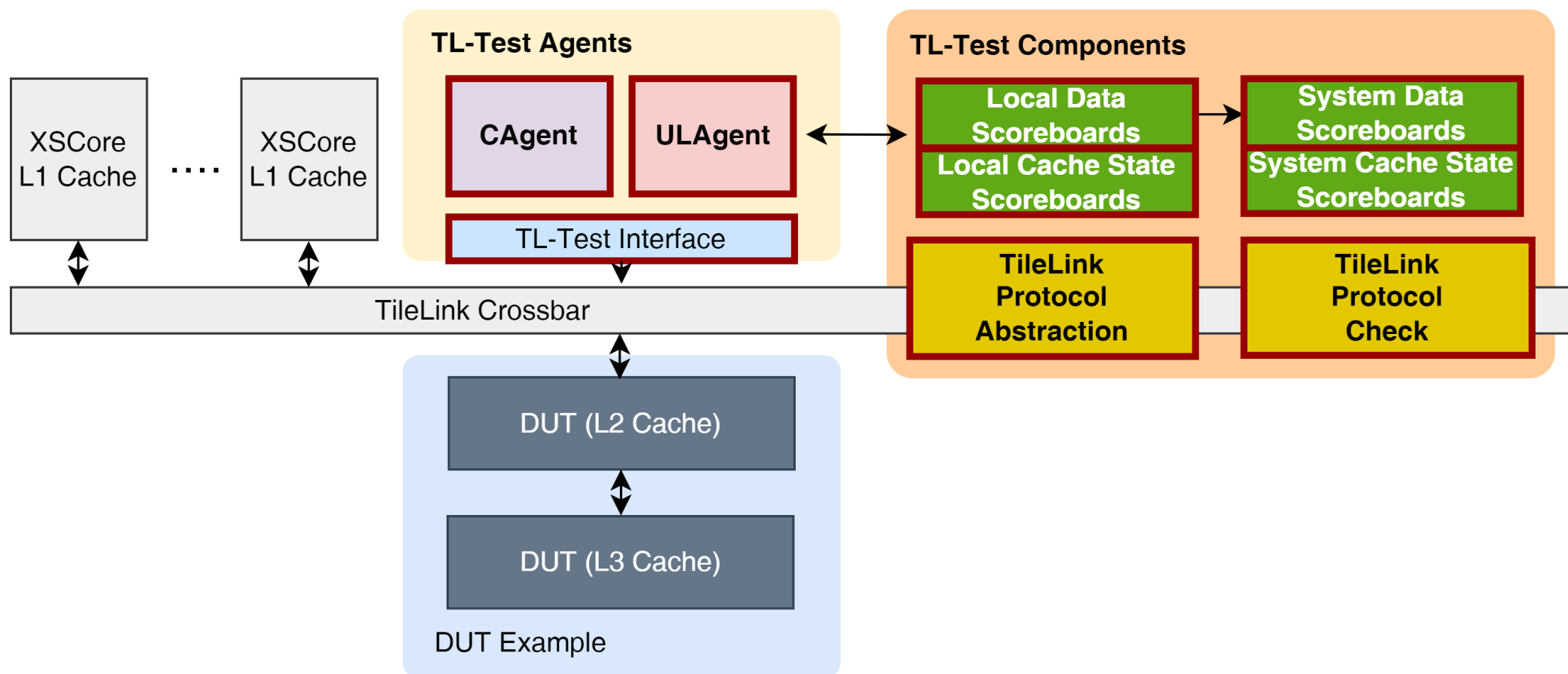
- 要求 CMO 参与可见性定序的目标**至少为 I/O 一致性设备**
- 仅依靠标准的 AMBA CHI 实现 CMO **无法满足 RISC-V CBO 的要求**
- 需要 SoC 协同设计才能保证符合 RISC-V CBO 的要求

• → 此种需求是否真的合理？

No Coherency (Out of PoS, System observable)



TL-Test: 基于 TileLink 的多功能缓存验证框架

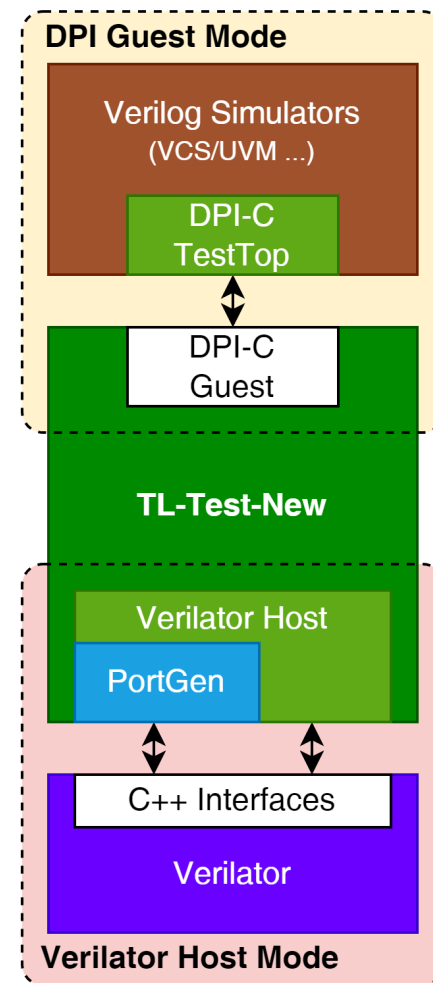


- 基于 TileLink 总线抽象出的一整套**离核**验证、调试、性能采样基础设施

🚧 软件工具演进

- 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**
 - 早期 AMBA CHI 仅有商业 RTL IP 和 VIP
 - 保证原有工作流可用
 - 保持缓存系统**离核**开发、测试、性能分析的方法论
 - 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（如 GitHub CI）
 - 更好的随机激励生成器（Fuzzer）与专用测试序列
 - 根据我们整体的开发、测试经验，不断**完善设计自测**

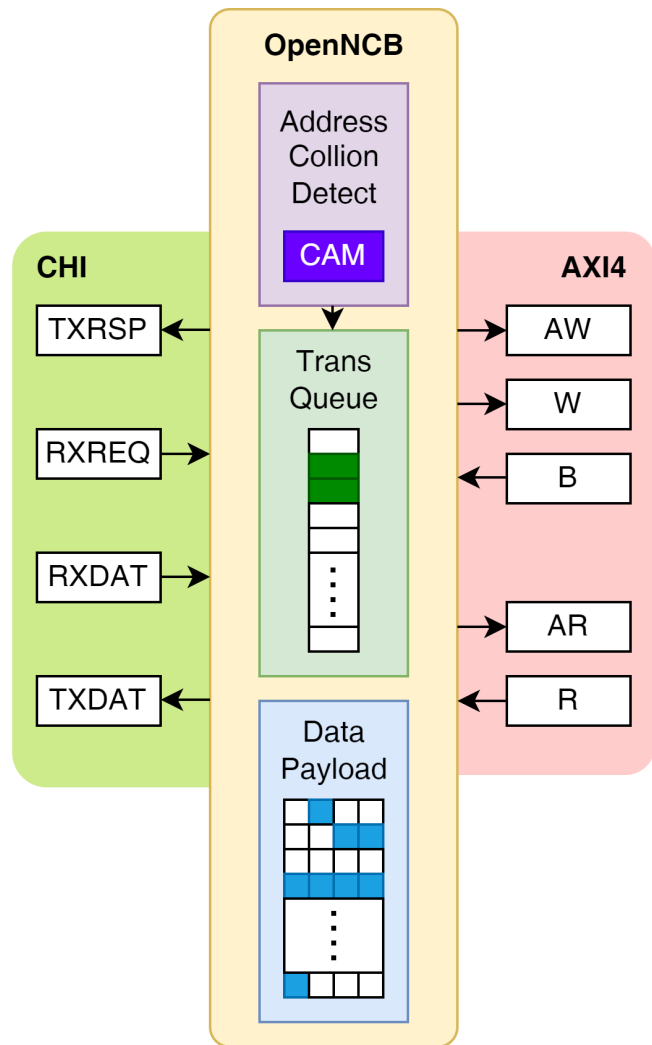
→ 开发流程永远需要基础设施的支持



开源 CHI SoC IP

- OpenNCB: 开源高性能 CHI to AXI4 协议桥

- 到 AXI4 的非一致性桥接
 - 主要桥接内存设备
- Credit-controlled Zero Retry
- 支持参数化配置 outstanding 深度
- 支持参数化配置总线数据宽度
 - CHI: 128, 256, or 512 bits
 - AXI: 32, 64, 128, 256 or 512 bits
- 可配置的事务乱序，适配不同的 SoC 场景
 - 全乱序、部分乱序、不乱序
- 保证可观测性顺序的事务提前返回，给予 HN 更大的乱序空间

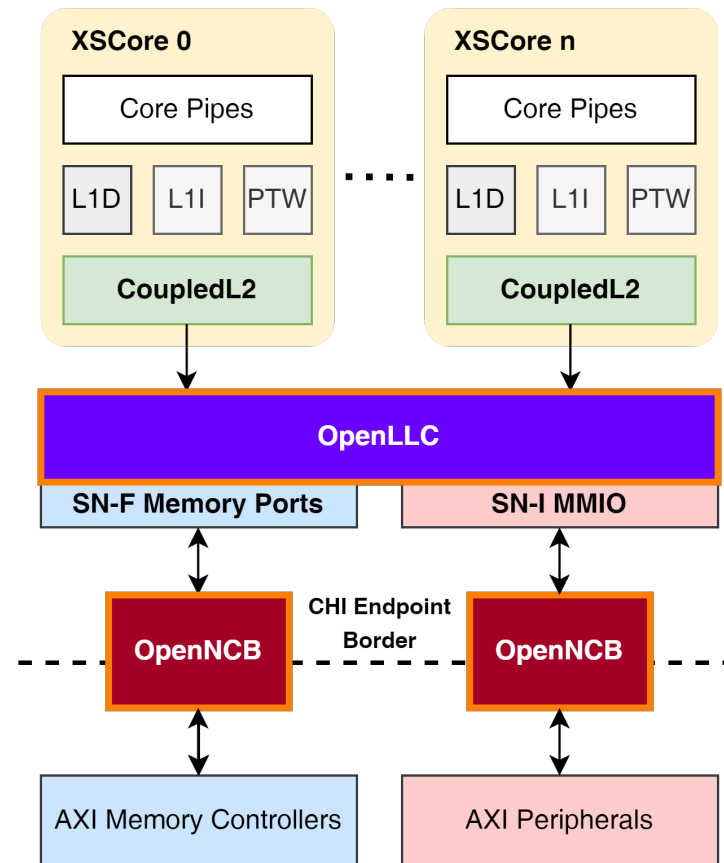


开源 CHI SoC IP

• OpenLLC: 开源高性能 CHI HN-F 实现

- 向上为 RN-F 端口，向下为 SN-F 端口
- 支持必要的 CHI Issue E 事务
- 支持 CMO 请求
- 基于目录的 MESI 一致性协议 (w/ Snoop Filter)
- 支持多种缓存包含策略
 - 单核为 exclusive，多核为 non-inclusive
- 高并发度设计与非阻塞主流水线架构
 - 多 slices 并行，同缓存 set 内请求并行
 - 支持部分事务数据 bypass

→ OpenLLC 与 OpenNCB 组成 CHI 最小互联系统 IP



开源 CHI 基础设施

- **CHIron: 开源 AMBA CHI 基础设施**

- 世界首个开源 AMBA CHI 工具集
- 使用 C++ 描述 AMBA CHI 标准，利用编译器特性实现类 Machine Readable Specification
- 支持 CHI Issue B/C/E，目前主要维护 Issue E
- **基于模板的高扩展性协议层、事务层抽象**
- 支持大部分事务层抽象，正在逐渐完善，并将于今年内实现完整的一致性状态追踪
 - 目前已经完全覆盖 Kunminghu V2 的需求，并协助解决数个总线协议问题
- **目标成为基础设施的基础设施 (Infrastructure of infrastructures)**
 - 大部分代码都遵循一定的设计模式，几乎所有功能以 API 形式提供
 - 可以成为 UVM 系统的内核，但 CHIron 可以将我们从 UVM 中解放出来，VIP 不再是黑盒
 - 将支持多线程分析，为 Palladium 离线分析等提供更好的性能 **(包括多线程解算、检查缓存行状态)**
 - 在开源项目中可以随意使用

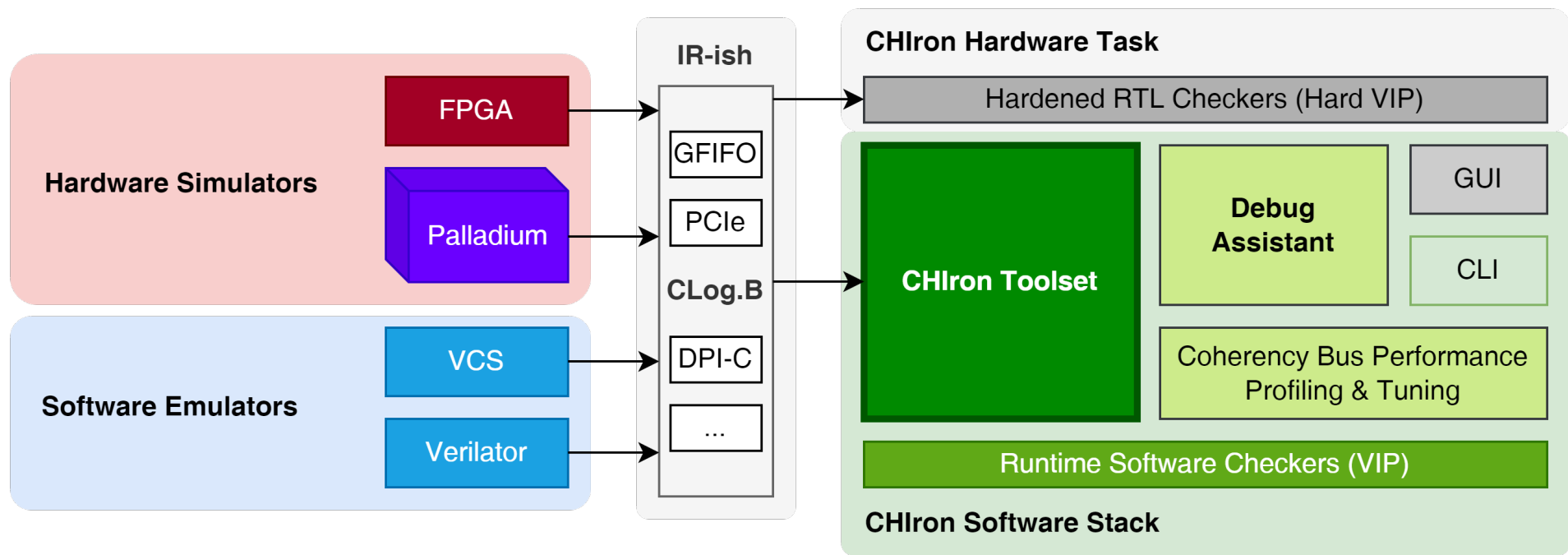
🚧 开源 CHI 基础设施

• CHIron: 开源 AMBA CHI 基础设施

• 支持基于总线事务 Trace 的在线、离线分析

- 支持带压缩的流式二进制格式 (CLog.B), **压缩率接近 37%, 将支持多线程离线分析**
- 将支持上下文无关的 Trace 分析 (例如仅记录、分析部分 Trace 片段)

• 更好地支持基础设施需求, 协助打造开发流程、工具栈中的任意链条, 加快设计反馈效率



敬请批评指正！