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

香山昆明湖架构访存单元的设计演进

王华强 **李昕**[1] 冯浩原[1] 吴凌云[2] 李燕琴[1] 马建露[3]

[1]中国科学院计算技术研究所

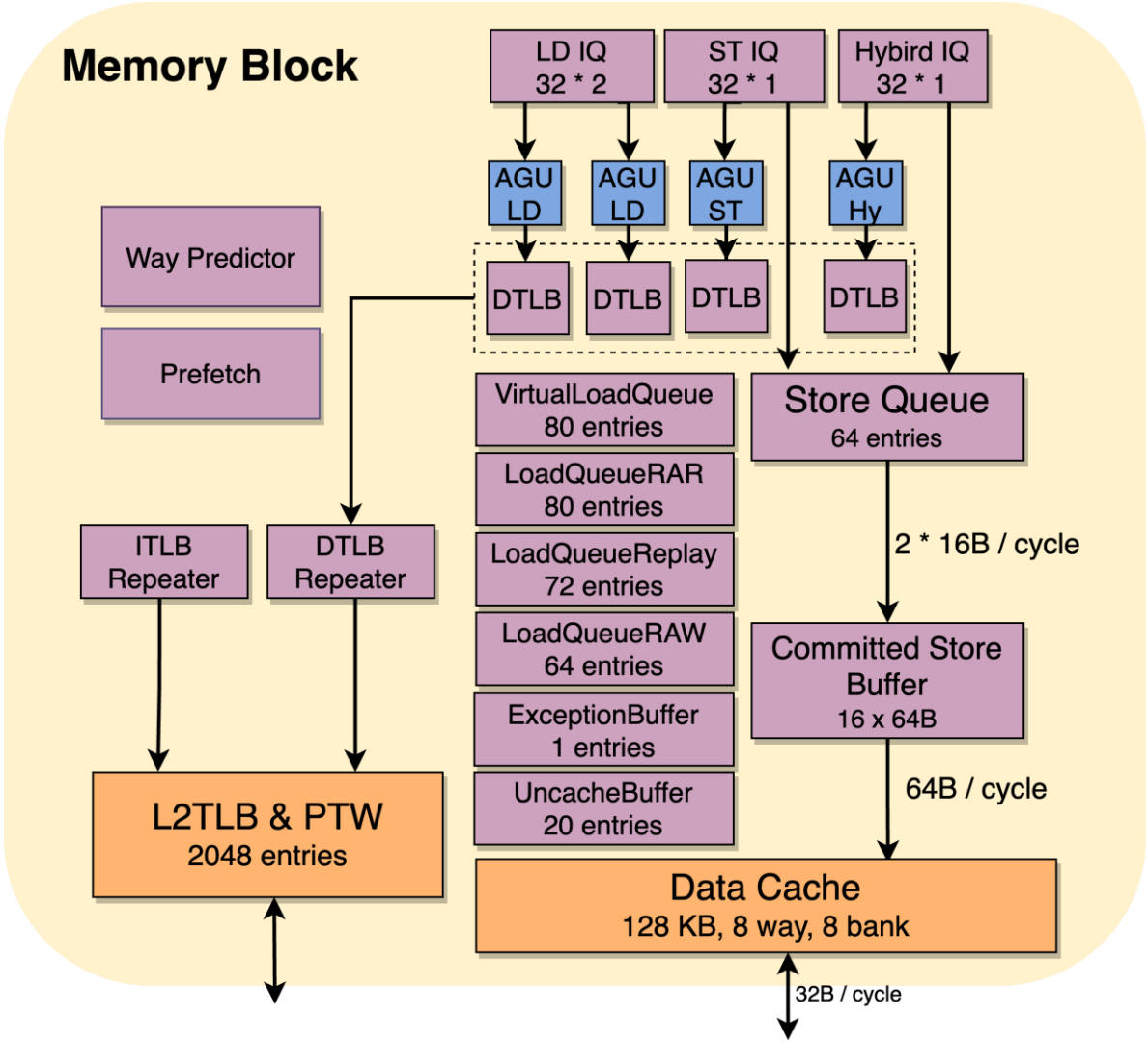
[2]上海科技大学

[3]北京大学

2023 年 8 月 24 日 @第三届 RISC-V 中国峰会

昆明湖架构访存子系统

- 访存子系统的整体架构
- 相对于南湖架构变化
  - 拆分 load queue
  - 增大 load 带宽
  - 添加路预测器
  - 优化 MMU
  - 增加预取

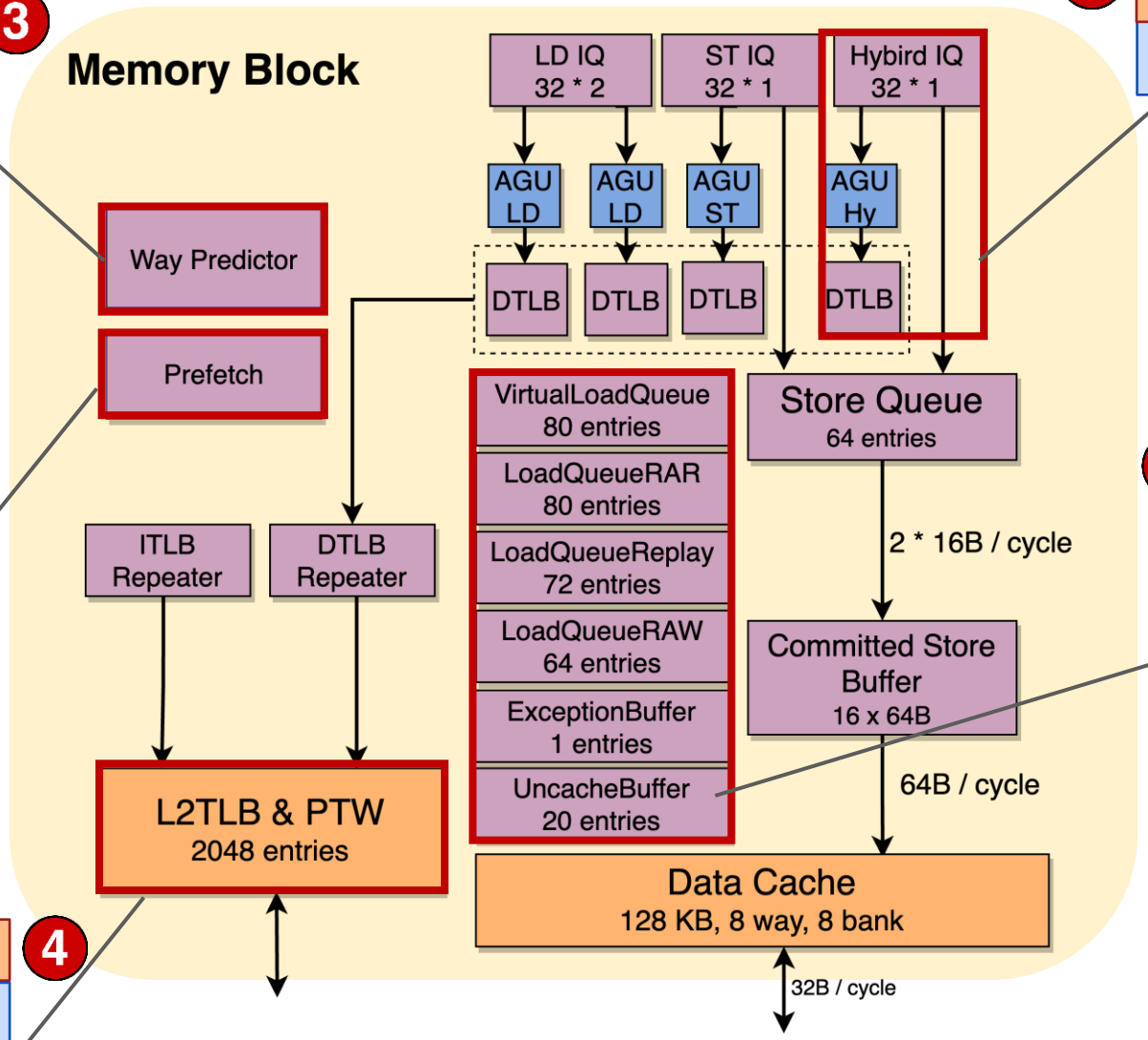


香山处理器访存子系统存在的瓶颈

缺点：L1D DataSram 功耗大 ③
优化：添加路预测器

缺点：L1D 命中率低 ⑤
优化：添加预取器

缺点：MMU 时序紧张 ④
优化：优化 MMU 布局与结构



② 缺点：Load 带宽不够
优化：添加 Hybrid 访存单元

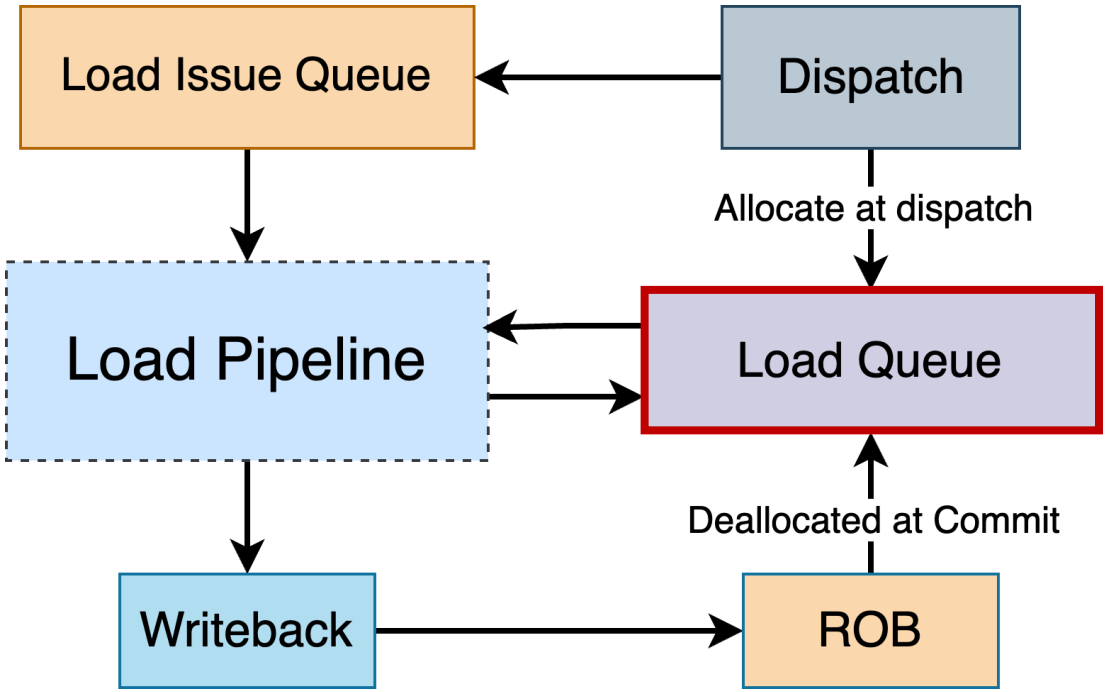
① 缺点：LQ 时序紧张，复杂
优化：拆分 load queue

参数对比：雁栖湖 -> 南湖 -> 昆明湖

AGU &Pipeline & Queue & features	雁栖湖	南湖	昆明湖
AGUs	2-LD, 2-ST	2-LD, 2-STA	3-LD, 2-STA
L1 LD hit latency	3	4	4
L1 LD hit bandwidth	2x8B/cycle	2x8B/cycle	3x16B/cycle
Store data bandwidth	2x8B/cycle	2x8B/cycle	2x16B/cycle
Load queue entry	64	80	80 (splited)
Store queue entry	48	64	64
Store buffer (CSB)	16 cachelines	16 cachelines	16 cachelines
L1 Data Cache	32KB, 8Way	128KB,8Ways, 8Banks	128KB,8Ways,8Banks
Data Cache Probe Queue	16	8	8
Data Cache Miss Queue	16	16	16
Data Cache Write-back Queue	16	18	18
Software prefetch	No	Yes	Yes
L1 Hardware prefetch	No	No	Yes
Way predictor	No	No	Yes
Vector extension	No	No	Yes
Memory dependency predictor	Wait Table	Store Sets	Store Sets
L1/L2 interface	TileLink	TileLink (improve)	Self-defined

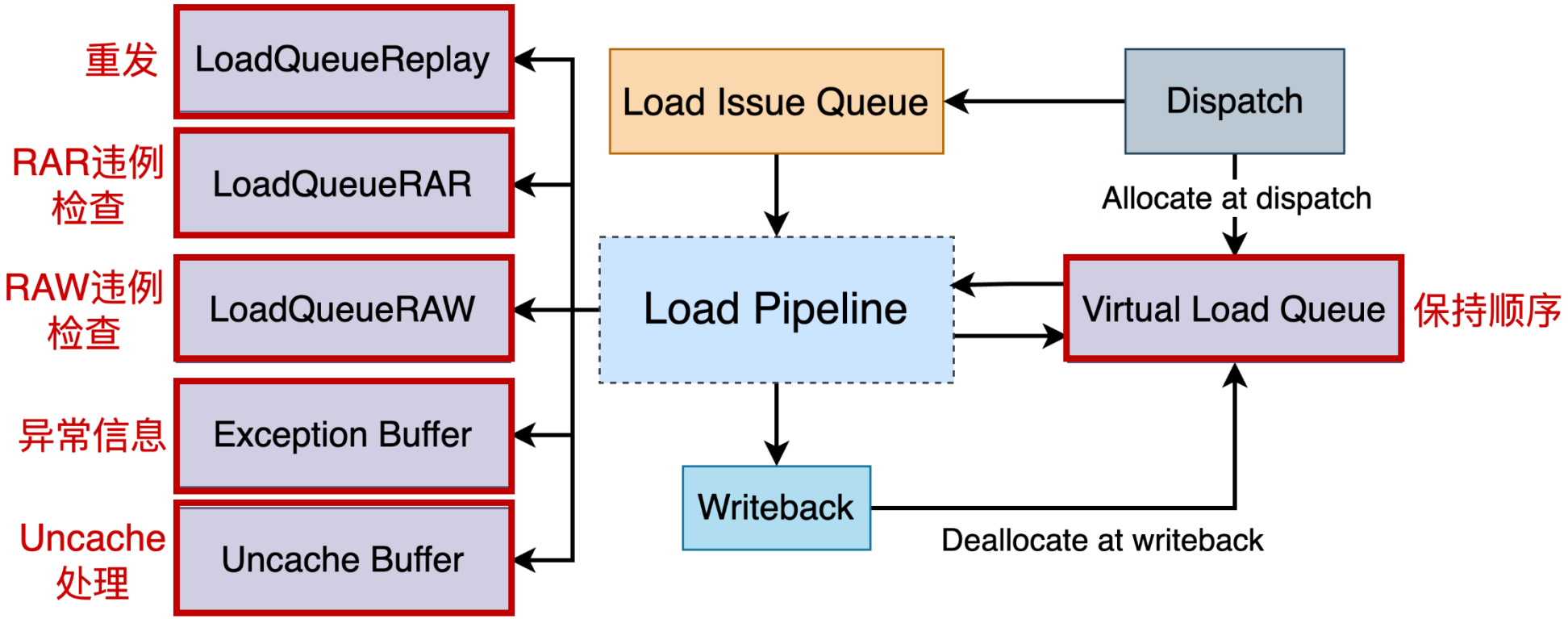
Load queue 拆分

- 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



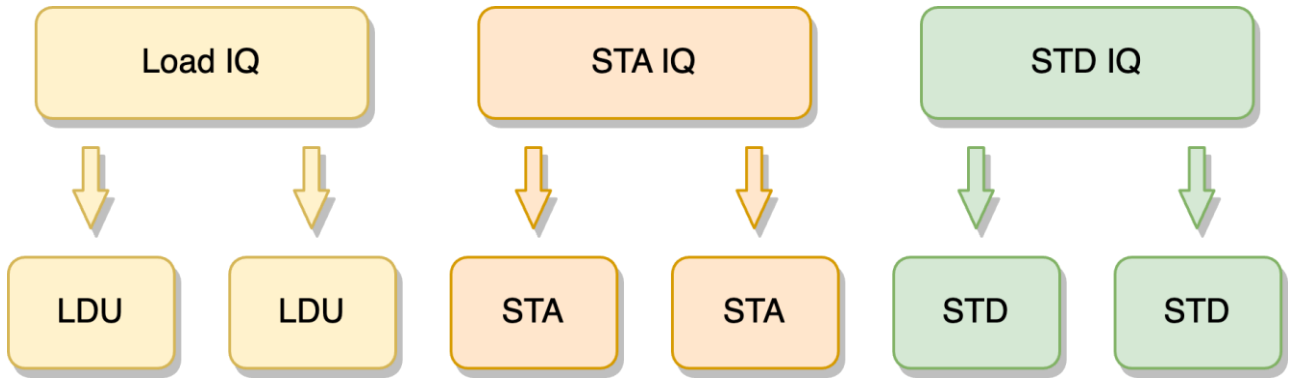
Load queue 拆分

- 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

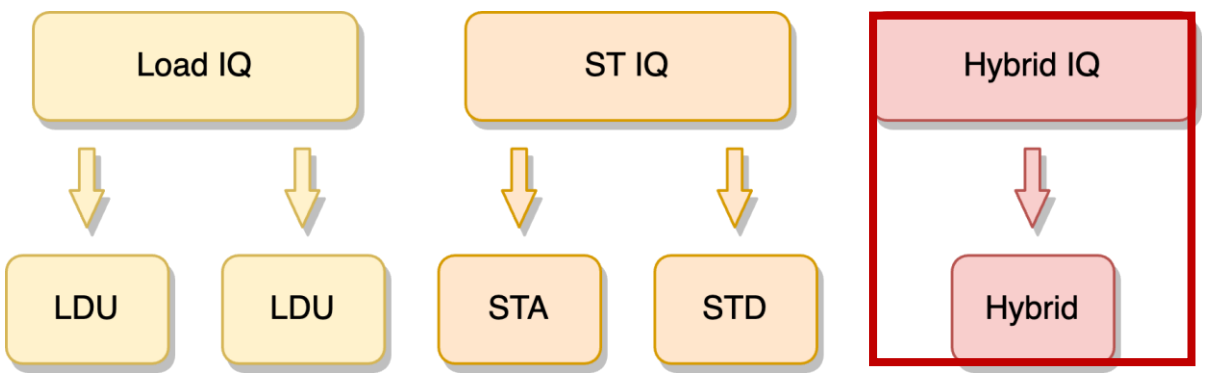


Load 带宽

- 昆明湖采用 **3-LD，2-ST** 设计
  - 每一个访存执行单元都独占一个保留站
  - Hybrid Unit：Load/Store **混合访存**执行单元，能够处理 load/store 指令
  - Hybrid IQ 区别于 STA 和 STD IQ，不将 store 指令的地址与数据拆分



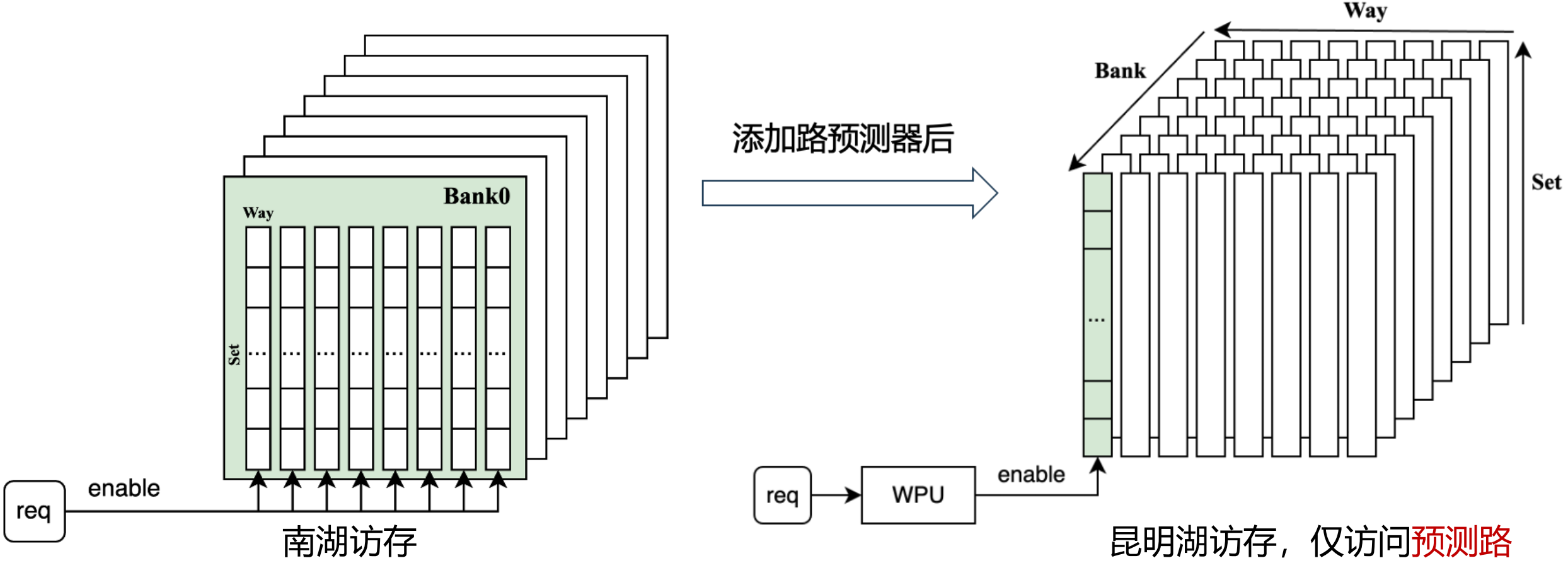
1. 南湖架构中 load/store 组织结构



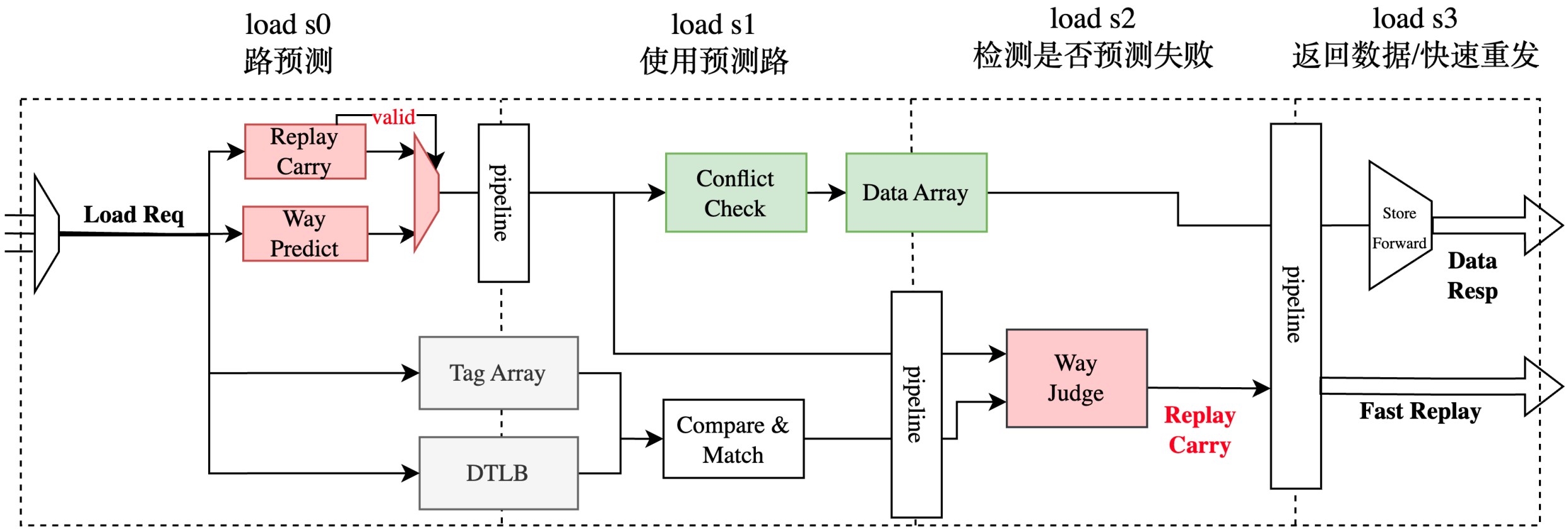
2. 昆明湖架构中 load/store 组织结构

路预测器：节省 N-1 路动态功耗

- 南湖访存缺陷
  - VIPT 缓存，在获取**命中路**前，需要访问 N 路数据，**产生大量动态功耗**

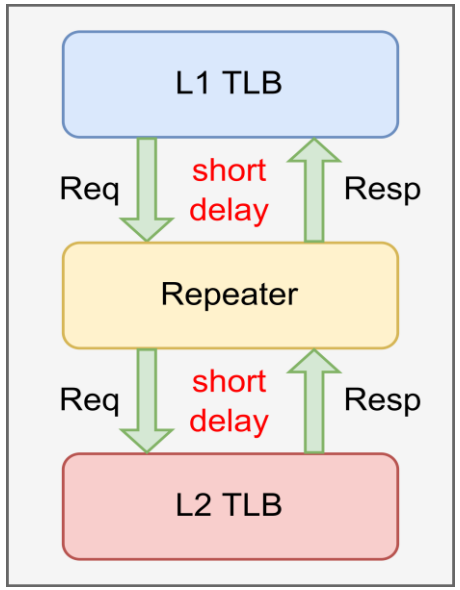


路预测器：通路设计

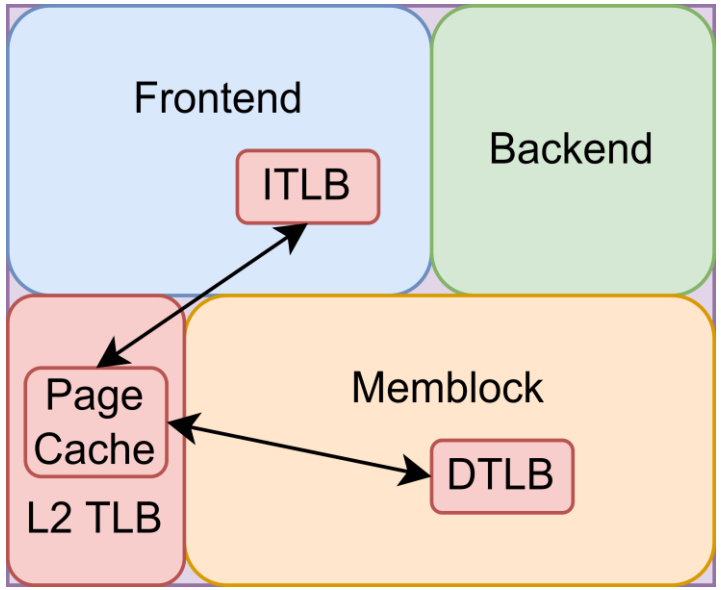

load s0
路预测
load s1
使用预测路
load s2
检测是否预测失败
load s3
返回数据/快速重发
valid
Replay Carry
Way Predict
Load Req
pipeline
Conflict Check
Data Array
Tag Array
DTLB
Compare & Match
pipeline
Way Judge
Replay Carry
pipeline
Store Forward
Data Resp
Fast Replay

MMU — 优化物理布局

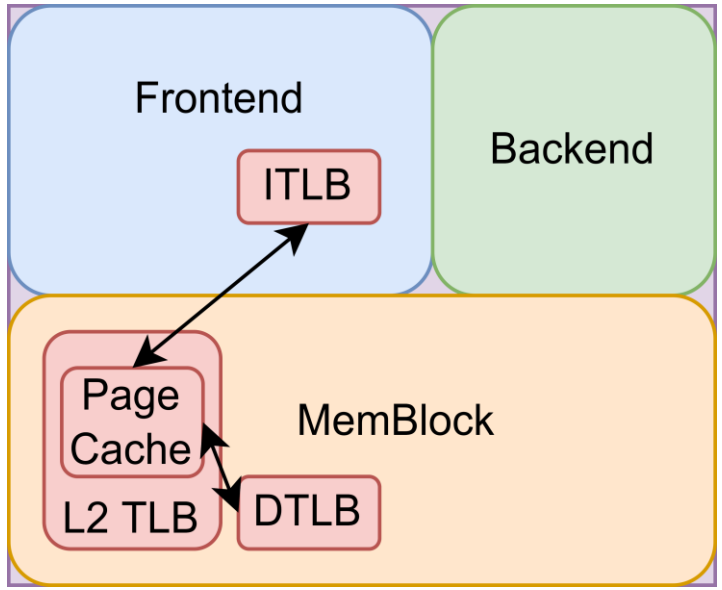
- 南湖架构 DTLB 和 L2 TLB 的**物理距离较远**，延迟较高，对性能影响较大
- 昆明湖架构将 **L2 TLB 合并至 Memblock** 中
  - 缩短 **DTLB 缺失时**访问 L2 TLB 的物理距离和延迟
  - ITLB 缺失时访问 L2 TLB 的延迟保持不变



南湖架构需要在两级 TLB 间增加中继器



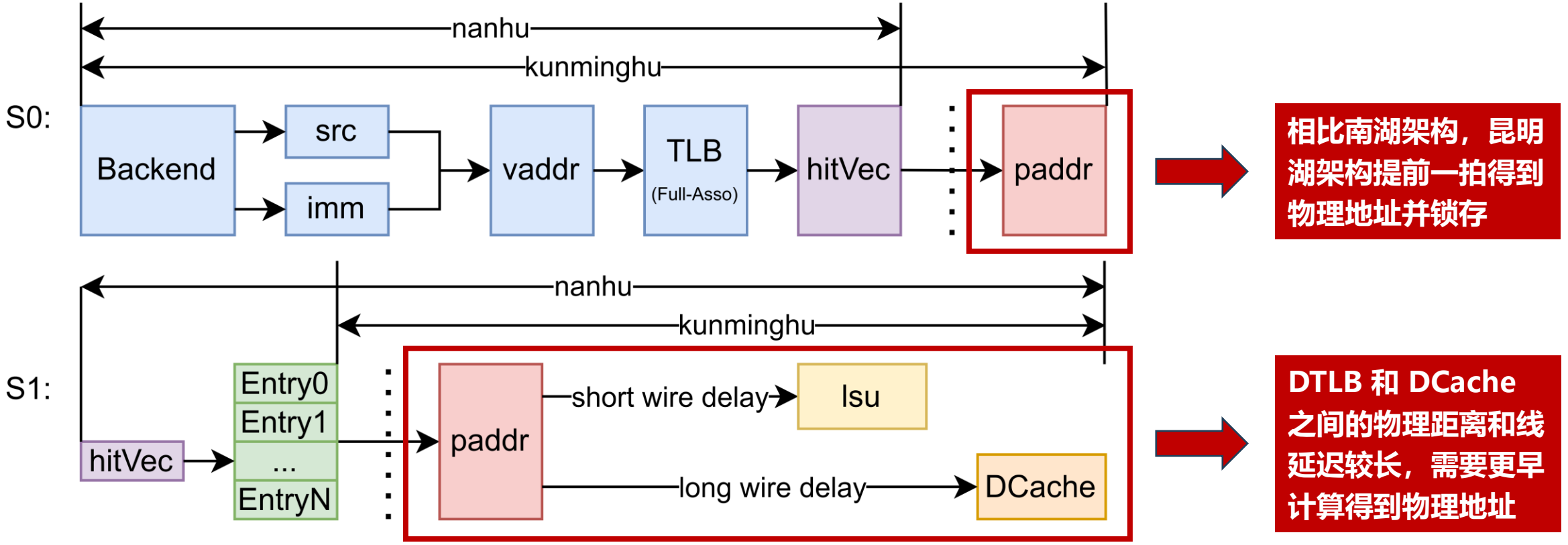
香山南湖架构的物理版图实例



合并 L2 TLB 至 MemBlock 中，无需中继器

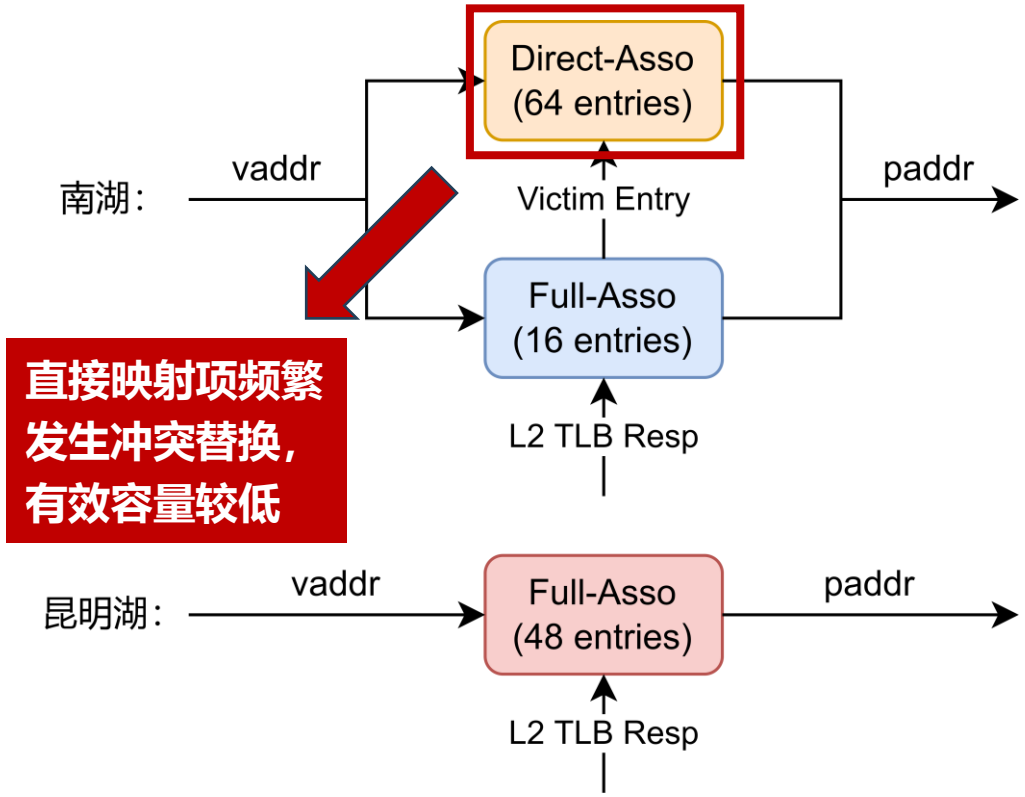
MMU — 优化 DTLB 查询时序

- 南湖架构中，S0 只查询 TLB 的命中项，S1 **索引得到 paddr** 并发送给 DCache
- 昆明湖架构中，S0 查询得到 paddr 并锁存，S1 将锁存的 **paddr 直接发送**

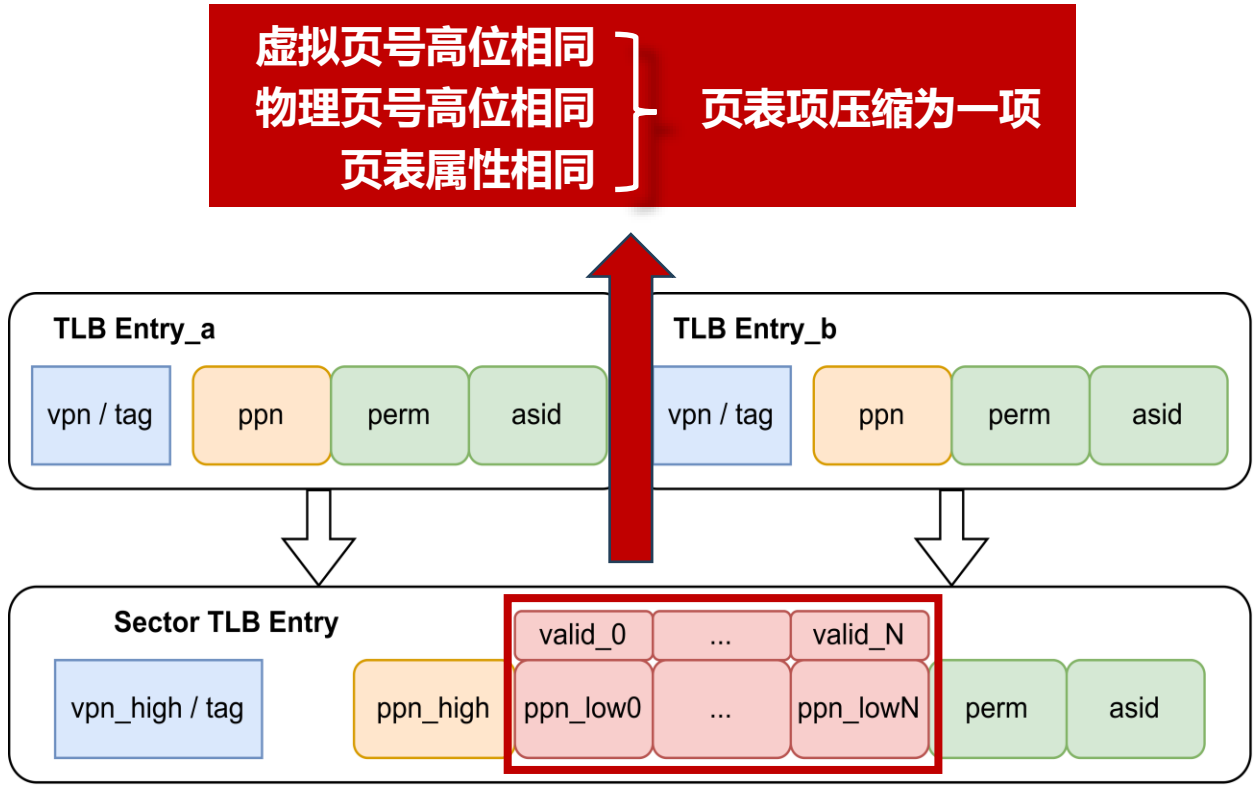


MMU — 提升 DTLB 有效容量

- 16 项全相联项 + 64 项直接映射项并行查询 -> **48 项全相联**结构
- 实现 TLB 压缩，合并虚拟页号和物理页号均**连续的页表项**



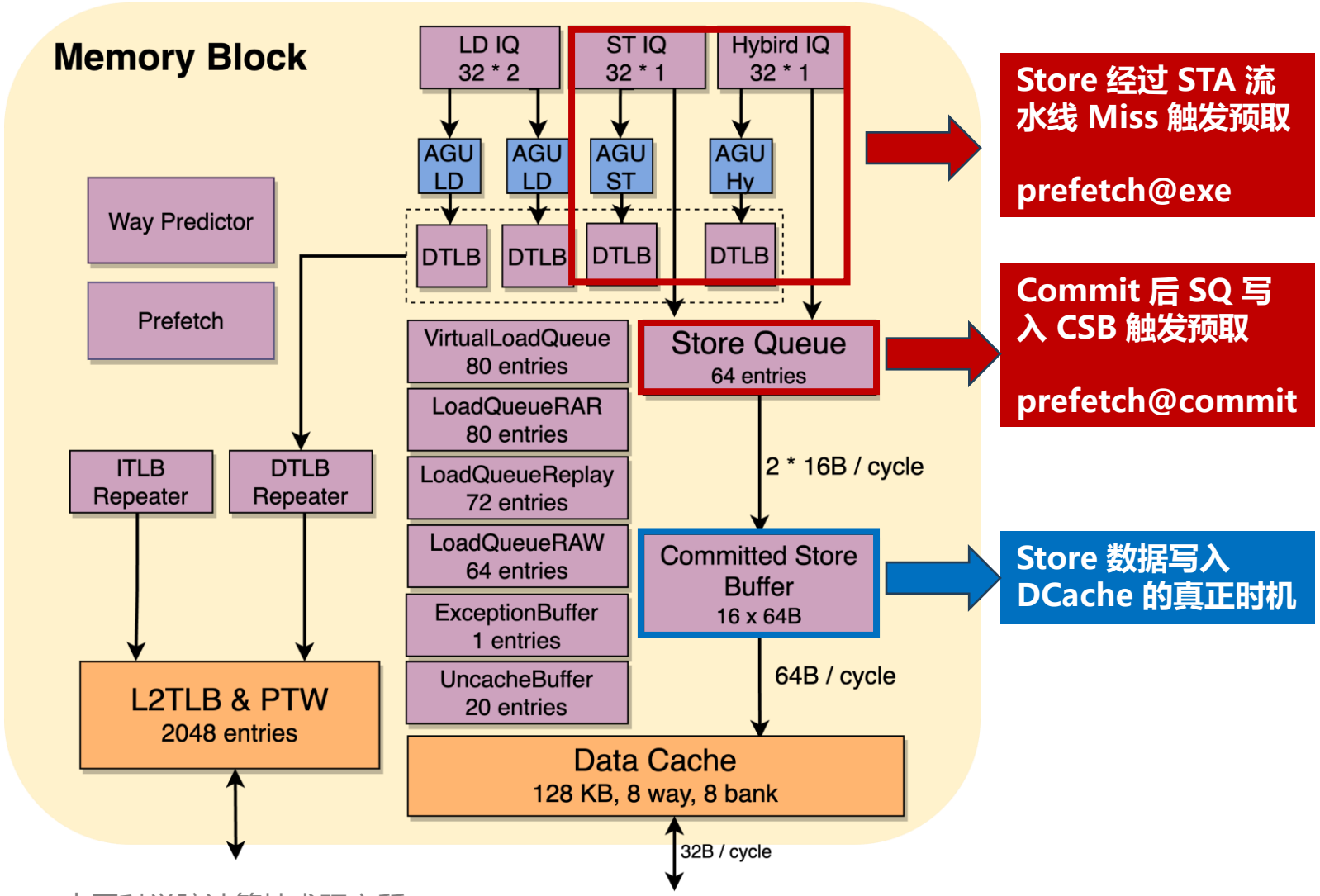
南湖和昆明湖架构中 DTLB 的组织结构



TLB 压缩示意图，将具有合并特征的 TLB 项压缩为一项保存

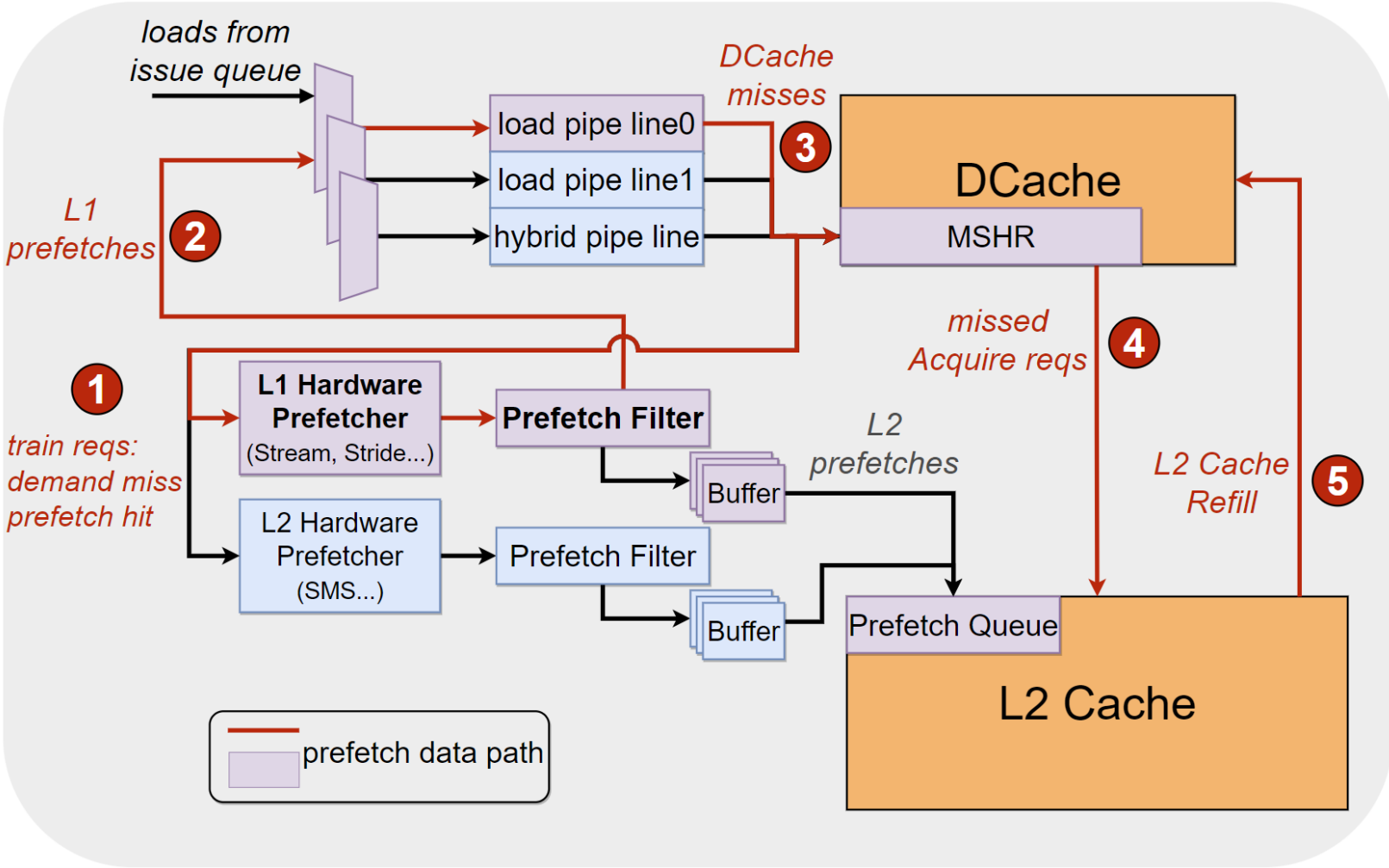
L1 Store 预取机制 — 提高 Store 命中率

- 特点
  - Store 执行/提交距离真实写入 DCache 有一定的**时间差**
- 预取时机
  - prefetch@exe
  - prefetch@commit
- 目前启用
  - **prefetch@exe**



L1 Load 预取机制 — 提高 Load 命中率

- L1 预取流程
  - 训练，模式检测
  - **复用 load 流水线**
  - DCache MSHR 取回数据
- 预取算法
  - **Stream[1]**
    - (x,x+1,x+2,x+3...)
  - **Stride[2]**
    - (x,x+n,x+2n,x+3n...)



[1] S. Srinath, O. Mutlu, H. Kim and Y. N. Patt, "Feedback Directed Prefetching: Improving the Performance and Bandwidth-Efficiency of Hardware Prefetchers," 2007 IEEE 13th International Symposium on High Performance Computer Architecture, Scottsdale, AZ, USA, 2007

[2] J. -L. Baer and T. -F. Chen, "An effective on-chip preloading scheme to reduce data access penalty," Supercomputing '91:Proceedings of the 1991 ACM/IEEE Conference on Supercomputing, Albuquerque, NM, USA, 1991

小结

- 昆明湖在南湖的基础上做了几个方面的修改
  - 为增加 load queue 有效容量，拆分 load queue
  - 为提高 load 处理效率，增加 load 带宽
  - 为降低 L1 DataSram 的动态功耗，添加路预测器
  - 为优化 MMU 时序，修改 MMU 的结构和布局
  - 为提升 L1 DCache 命中率，添加预取器

敬请批评指正！