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

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

香山开源处理器社区
XiangShan Open-Source Processor Community



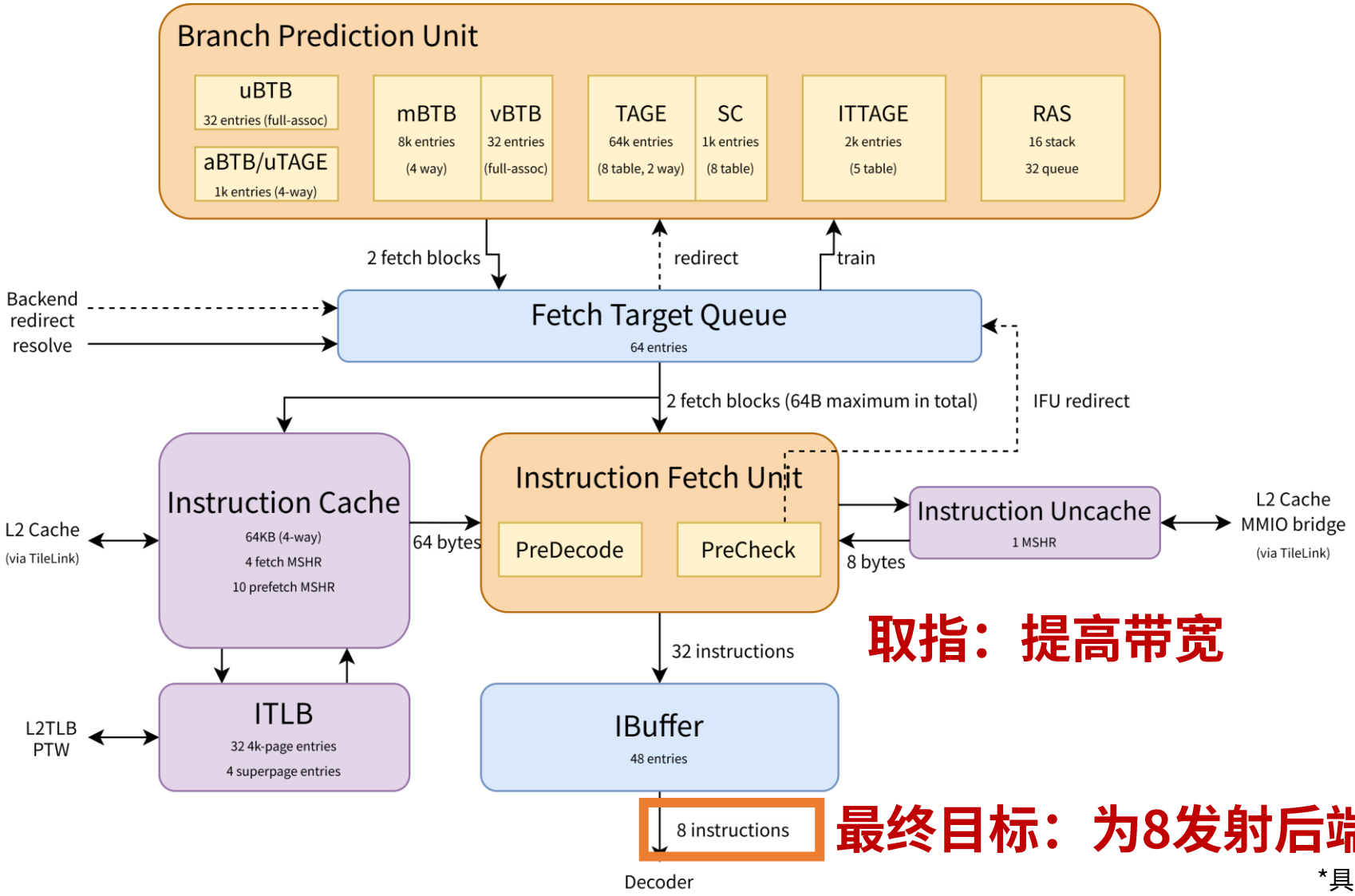
昆明湖 V3 前端架构演进

徐之皓
中国科学院计算技术研究所

2026 年 10 月

整体设计

分支预测：提高带宽+准确率



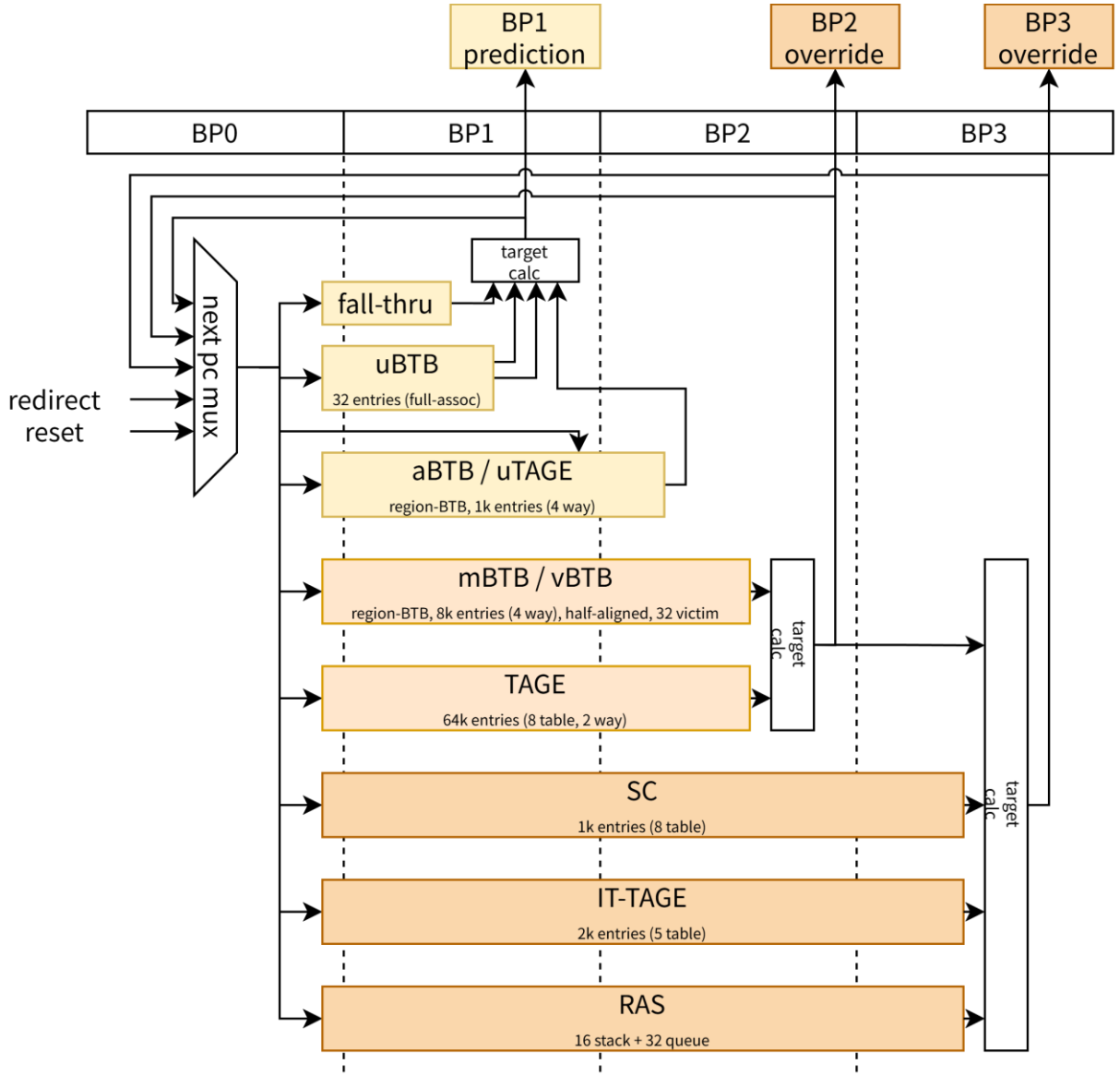
取指：提高带宽

最终目标：为8发射后端提供足够指令

*具体参数仍在探索

BPU 整体结构

- 三级预测流水（overriding）
  - BP1：uBTB + aBTB **+ uTAGE**
  - **BP2：** mBTB + TAGE
  - BP3：SC + RAS + IT-TAGE
- 禁止预测块跨页
- Half-aligned BTB
- Path-history
- **Victim BTB**
- **Resolve 训练**
- 2-taken（WIP）

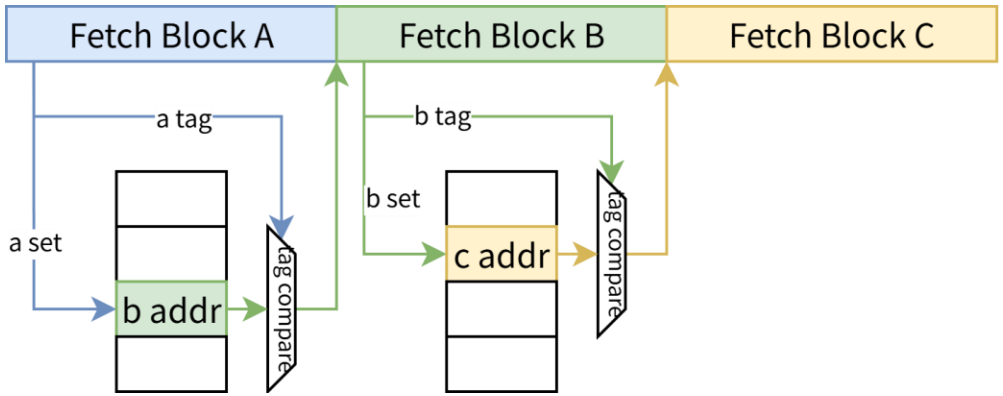


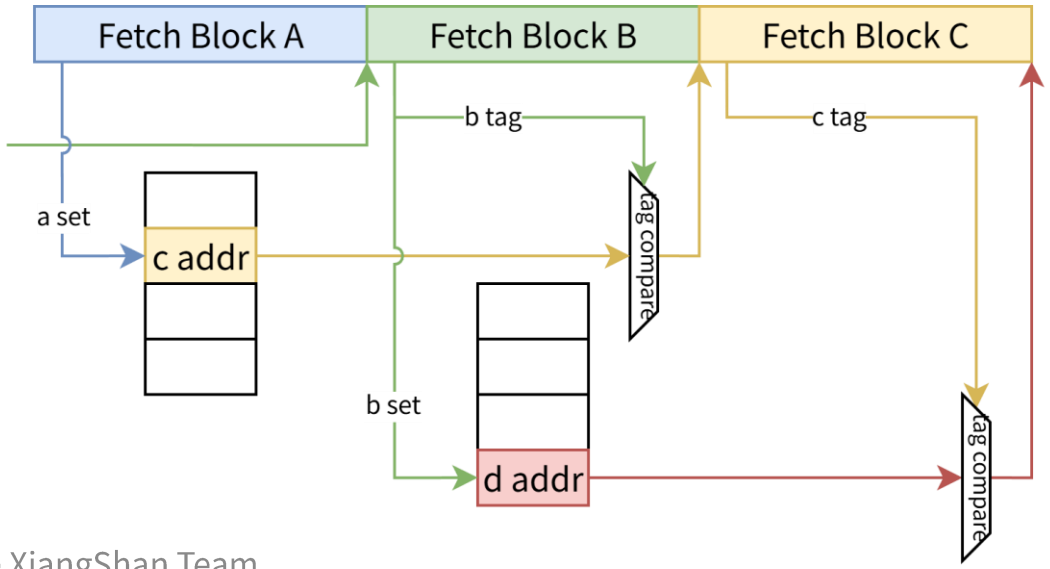
*具体参数仍在探索

AheadBTB / MicroTAGE：Ahead pipeline

- 用前一个预测块的地址索引 SRAM
- 1K 项（4 路组相联）BTB **+ 1K 项（2 张不同历史长度表）TAGE**

预测器	索引（查表）	Tag（判断命中）	预测
uBTB（左）	A	A	B
aBTB（右）	A	B	C

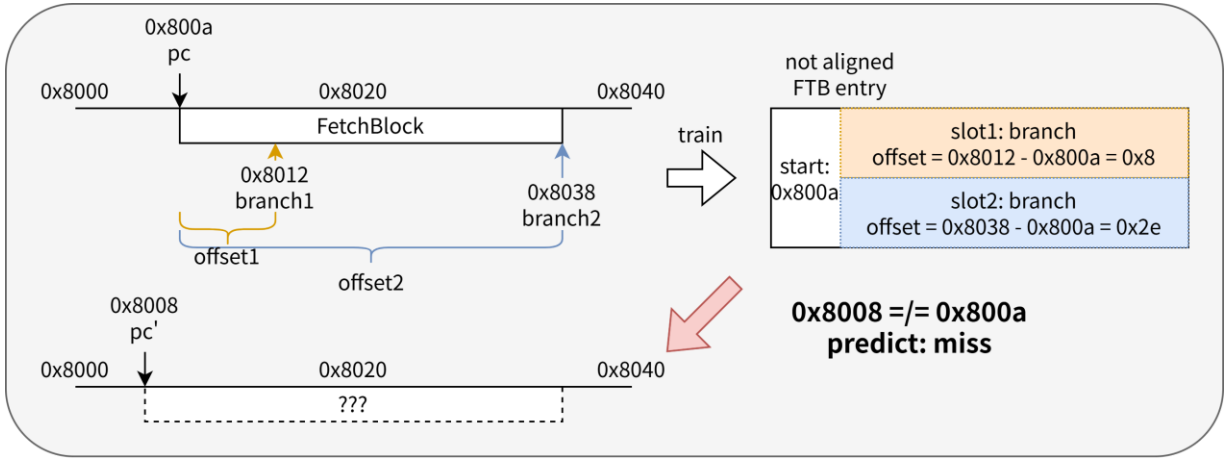


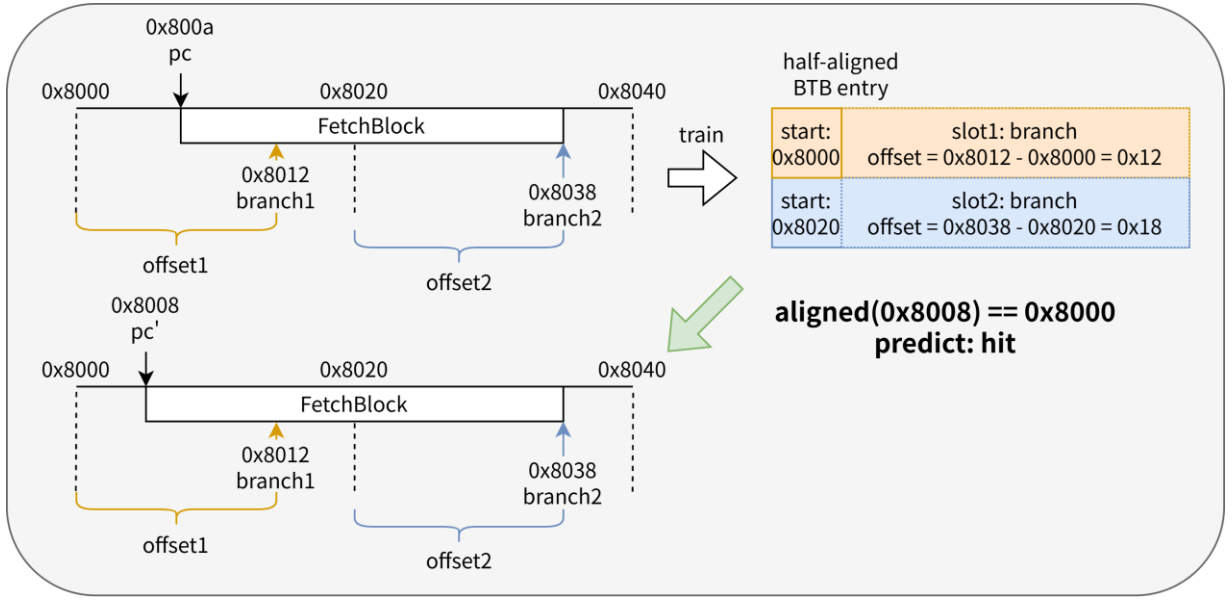


MainBTB：Region-BTB

- FTB：
  - 用pc直接索引 SRAM
  - 不同路径无法识别
  - 存在容量浪费

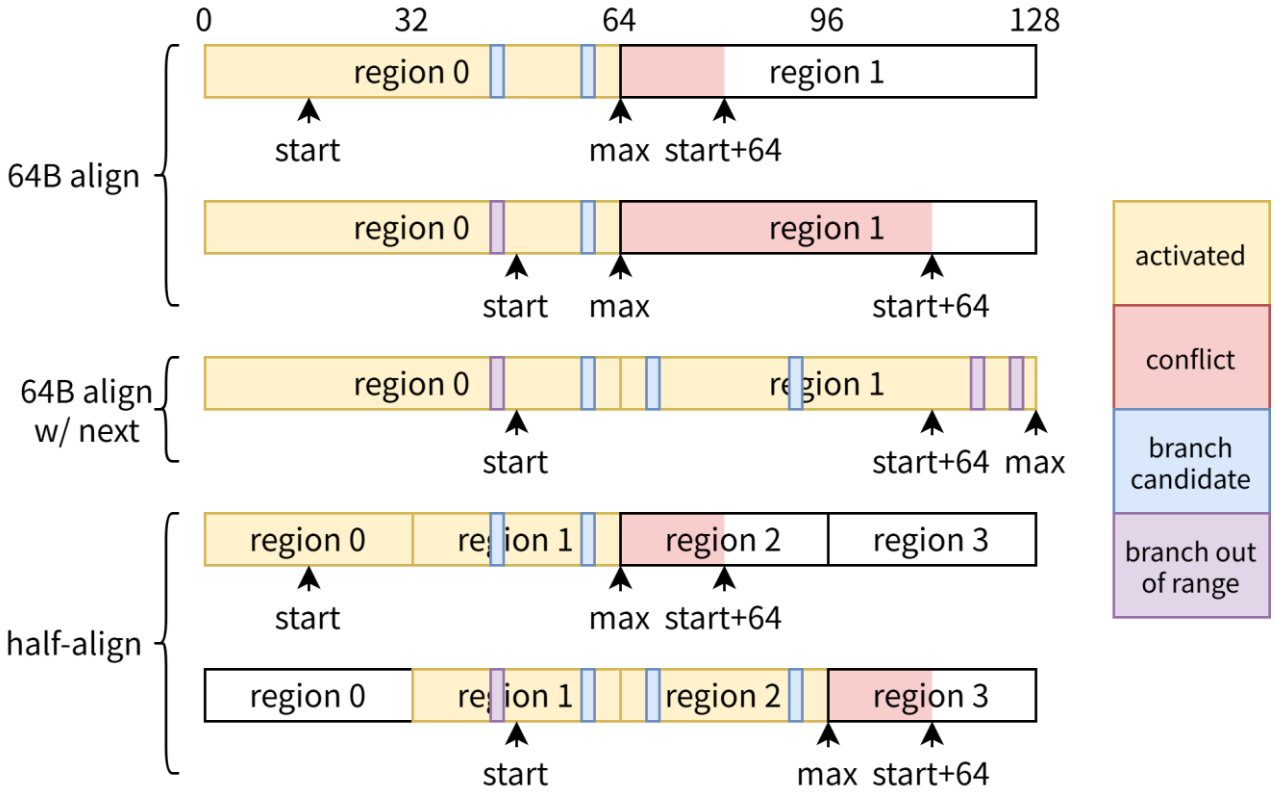
- Region BTB：
  - 用pc高位索引 SRAM
  - 执行路径无关
  - 面积利用率高
  - 存在时序压力
  - 存在相联度压力





MainBTB：Half-align

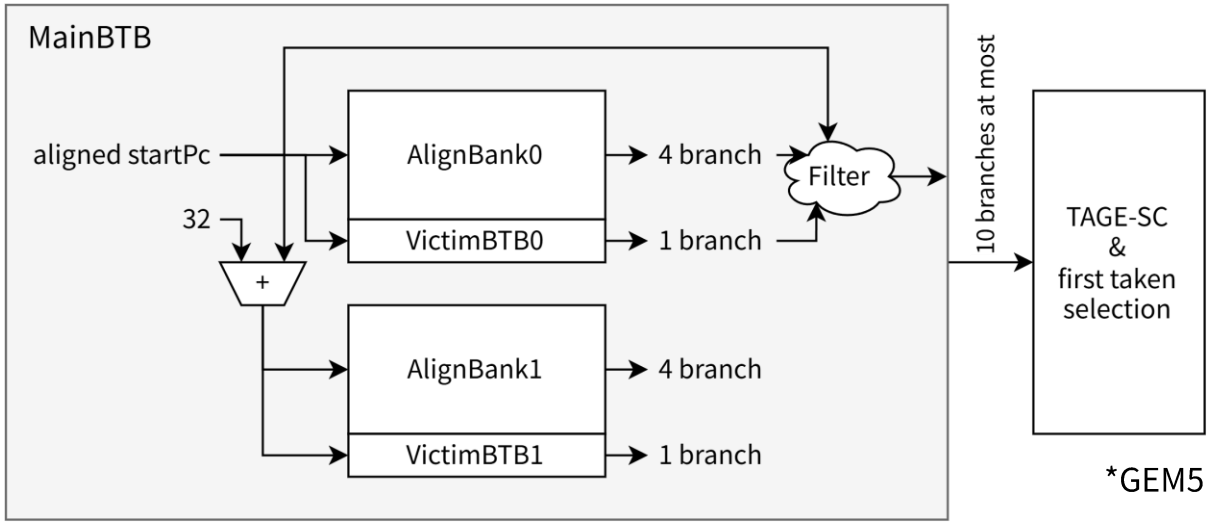
- Full-align：
  - Region 大小和取指块大小一致
  - 8 分支 / 64B region
  - pc 靠近 region 尾时取指块偏小
- Half-align：
  - Region 为取指块大小的一半
  - 4 分支 / 32B region
  - 预测时访问连续 2 region
  - 取指块大小稳定
  - 分支分布不均时相联度压力更大



MainBTB： VictimBTB

- MainBTB 相联度压力
  - SPEC06 dealII 23.62% 的块有 =5 条分支，99.98% <= 5 条
  - SPEC06 平均 2.73%、99.53%；SPEC17 平均 6.38%、99.20%
- 相联度 =5 需求不可忽略，但暴力增大 MainBTB 性价比低
- 2 * (4K 项（4 路组相联）Table **+ 16 项（全相联）Victim**)
- TAGE-SC 提供 10 条分支的方向预测

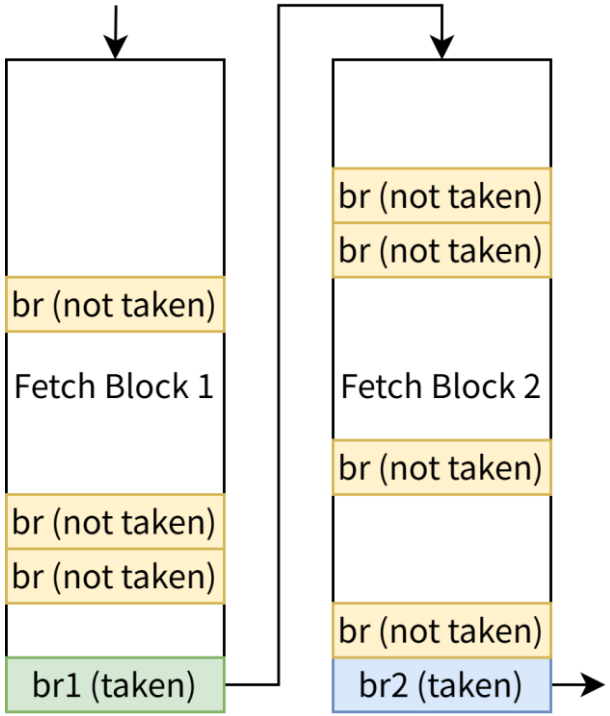
SPEC06 0.3c
dealII +43.487%
SPECint +0.526%
SPECfp +2.312%

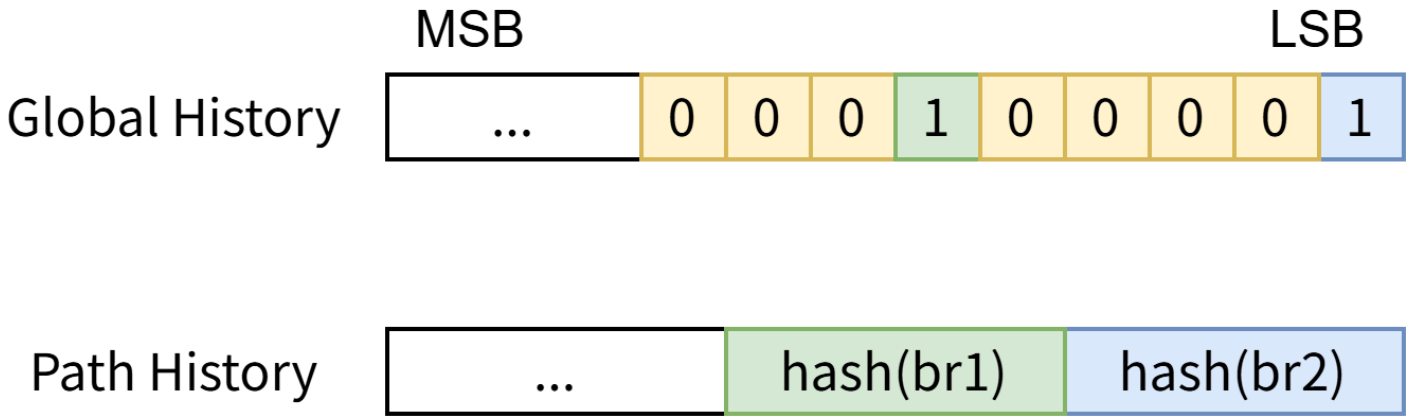


*GEM5 运行 gcc16 rva23 SimPoint 估计

TAGE-SC、ITTAGE：Path history

- 每周期能预测的非跳转分支增多，干扰全局历史 → 换用路径历史
- 记录所有分支方向 → 记录所有**跳转**分支的 pc/target
- 新增历史位宽固定、上下文识别能力强





其余预测器

- TAGE
  - 8 张表，共 64K 项；适配 BTB 结构重写
- SC
  - global、path、backward 各两张表；imli、bias 各一张表。共 1K 项
- IT-TAGE
  - 基本与 V2 保持一致，5张表，共 2K 项
- RAS
  - 基本与 V2 保持一致，16 项带压缩的提交栈 + 32 项的推测队列
  - **MicroRAS：** RAS 栈顶+BPU s2/3 流水信息，提供 s1 预测，不做栈维护

- https://github.com/OpenXiangShan/XiangShan-Design-Doc

BPU 训练机制

- 分支预测器的训练从指令提交（commit）提前到执行（resolve）
  - 准确度下降：可能用错误路径上的分支结果更新BPU
  - 及时性上升：BPU 可以更早看到新数据

- aBTB/uTAGE
  - 仅使用精确预测器训练：需要更早一个预测块的 startPc 作为 index，resolve 难获得
- uBTB
  - 优先 resolve：精确预测器 miss 场景可以更早看到数据，redirect 后命中
  - 无 resolve 时使用精确预测器训练

2-taken（WIP）

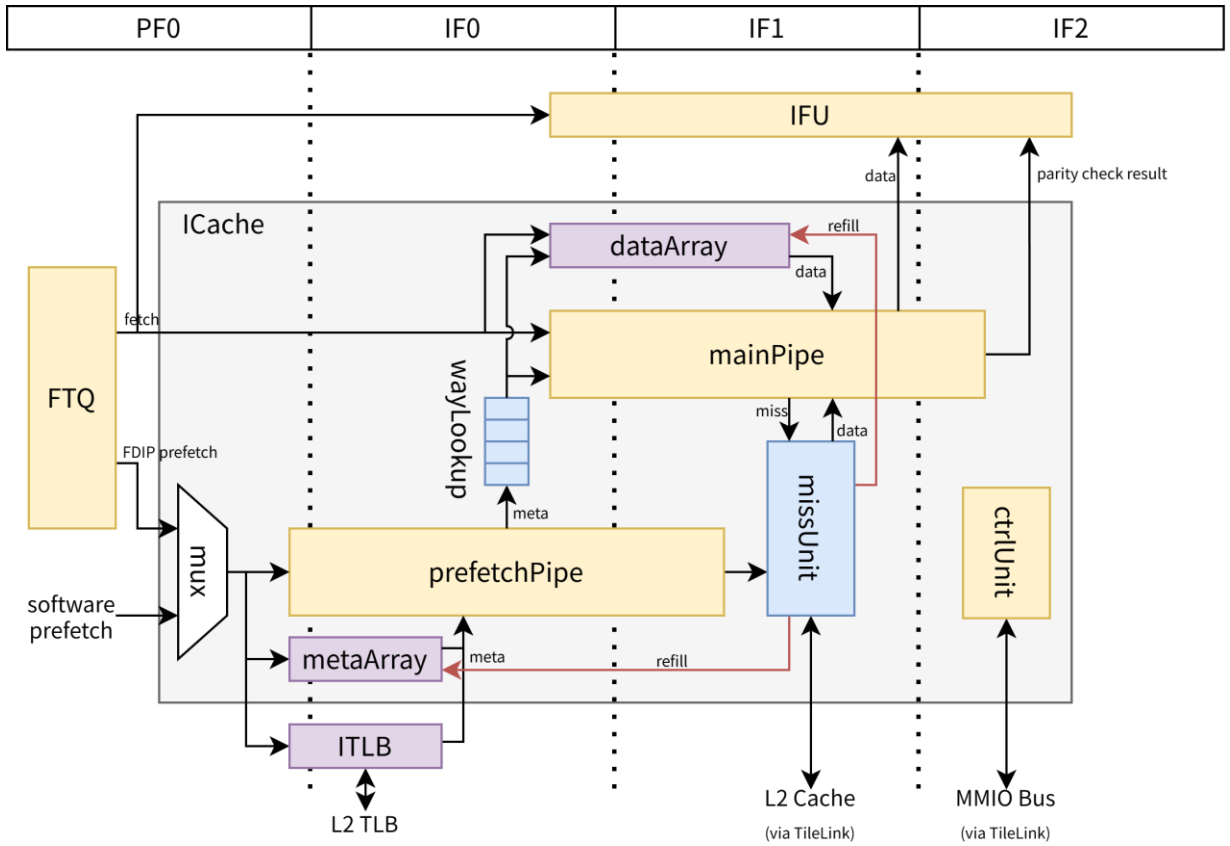
- 分支密集场景下：取指块平均偏小，供指带宽不足

- 快速预测器：类 trace prediction 设计
  - 单 entry 内存储至多两个分支
  - 与 FTB 不同，允许第一个分支是 taken 分支
- 精确预测器（WIP）：
  - 目标预测：重查 uBTB
  - 方向预测：扩展 TAGE

*初步方案，以后续实现/文档发布为准

ICache

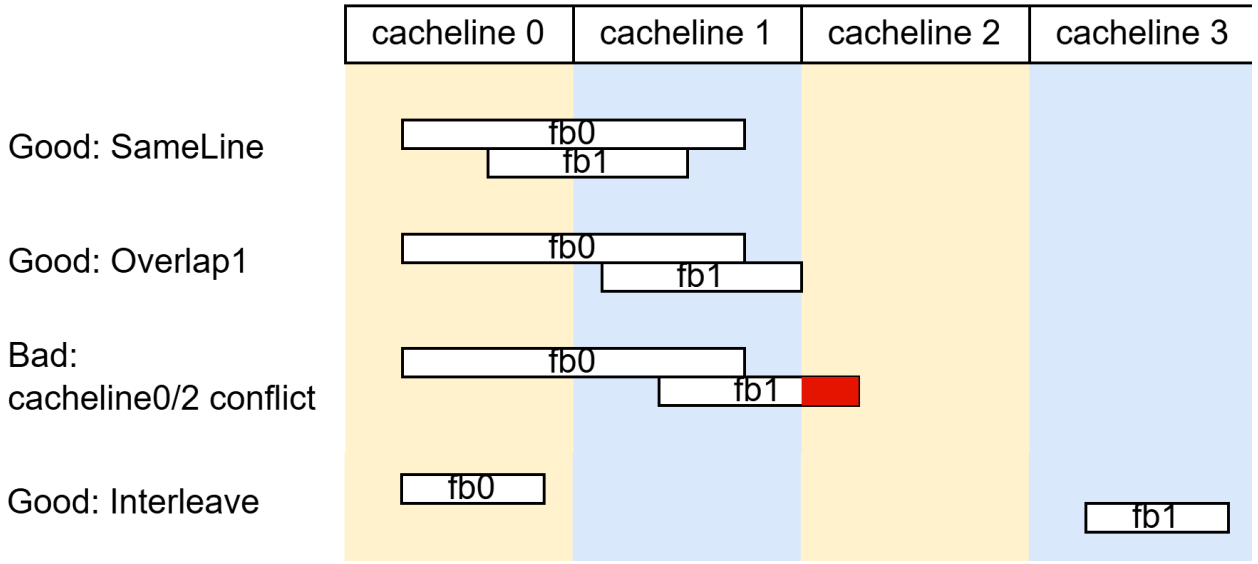
- 64KB（4 路组相联）
- 64B 缓存行
- 8B 细粒度存储（低功耗）
- 4 + 10 MSHR
- Parity 校验、错误注入
- **快慢路径拆分**
  - 指令数据、定界元数据 IF1 发送
  - Parity 校验、异常数据 IF2 发送
- **2-prefetch/2-fetch 支持**



2-(pre)fetch

- 与 2-taken 配合，处理分支密集场景的供指带宽问题
- 硬件复杂度/面积成本 – 性能 tradeoff：利用已有机制

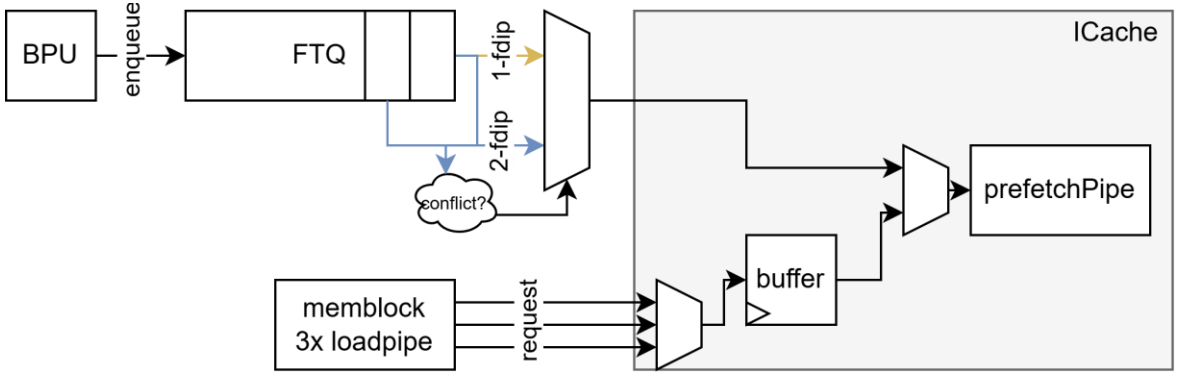
- 仅在无 SRAM 冲突情况下进行
  - metaArray interleave
  - dataArray banking
- meta、data 冲突条件不一致
  - 利用 wayLookup 解耦

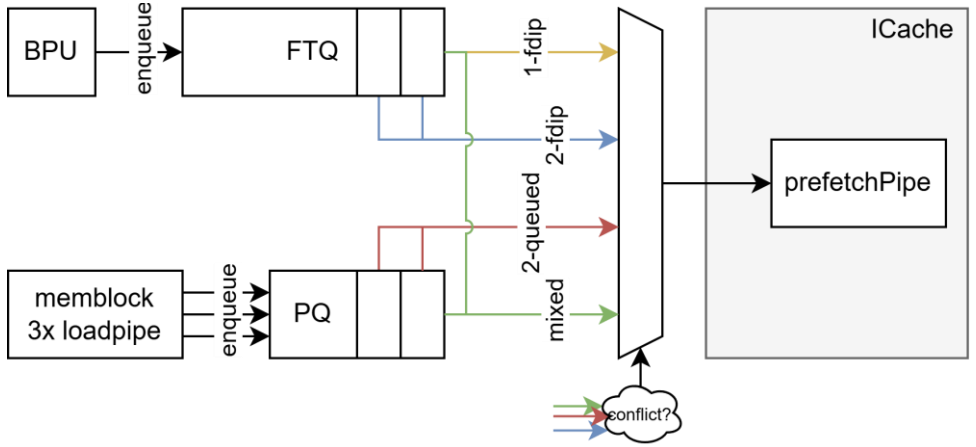


*更多冲突示例请参考设计文档

指令预取优化（WIP）

- 可扩展的预取队列设计
- 更完善、稳定的 Zicbop prefetch.i 支持
- 新预取算法探索中





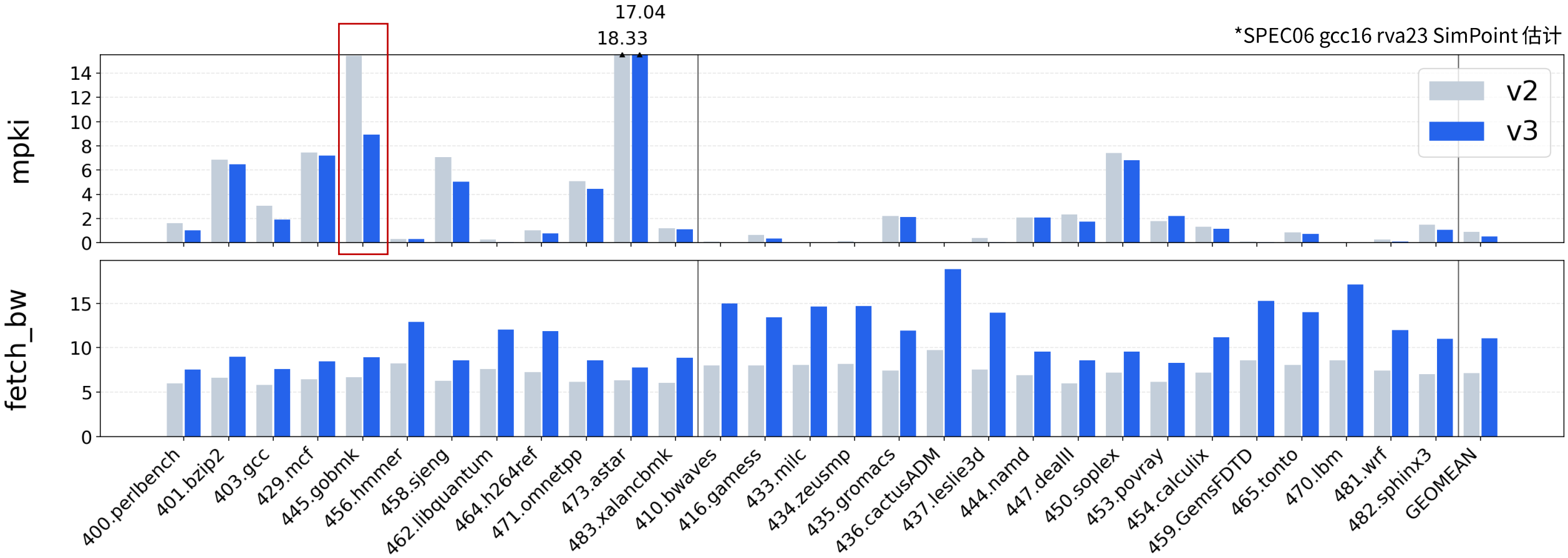
*初步方案，以后续实现/文档发布为准

IFU 优化

- 2-fetch 支持
- 时序优化
  - 单周期供指上限16 → 32，部分串行逻辑时序困难
  - IBuffer 入队预对齐：每个指令可能进入 48 项任意一项 → 进入特定 bank
  - 压缩指令信息预计算：从 data 读出后计算 → 重填时提前计算，存入 meta
- IFU resolve
  - 直接跳转类分支在预译码时已知目标、方向，直接训练 BPU
- 统一跨块指令处理
  - 4B/2B 指令混合情况，存在 4B 指令落在两个预测块中间
  - 更极端：两个预测块属于两个属性不一致的页（MMIO – cacheable 边界）
  - 新架构+UT

性能数据

- 平均分支预测 MPKI： 0.920 → **0.523** **(-43.15%)**
- 平均供指带宽： 7.168 → **11.037** **(+53.98%)**



架构演进总结

	特性	目标
V.S. 昆明湖 V2	2-taken（WIP）	每周期提供两个预测块，提高预测带宽
	AheadBTB	提高 s1 预测器命中率
	Half-aligned BTB 结构	允许识别来自不同跳转路径的分支，提高命中率
	Path-history	允许识别不同到达路径，提高上下文识别能力
	2-fetch	每周期处理两个预测块，提高供指带宽
	IFU 优化	更好的边界情况处理、时序优化等
V.S. RVSC 2025	MicroTAGE	提高 s1 预测器方向预测能力
	BPU S2 override	减少平均预测延迟
	VictimBTB	补充 MainBTB 相联度，提高命中率
	解耦 2-prefetch/2-fetch	减少因 SRAM 冲突导致无法 2-fetch 的概率
	指令预取优化（WIP）	提高指令缓存命中率

感谢大家对香山前端的关注！