



香山处理器的取指单元实现

金越¹

勾凌睿¹

邹江瑞²

张林隽¹

¹中科院计算所

²深圳大学

2021年6月@上海



目录

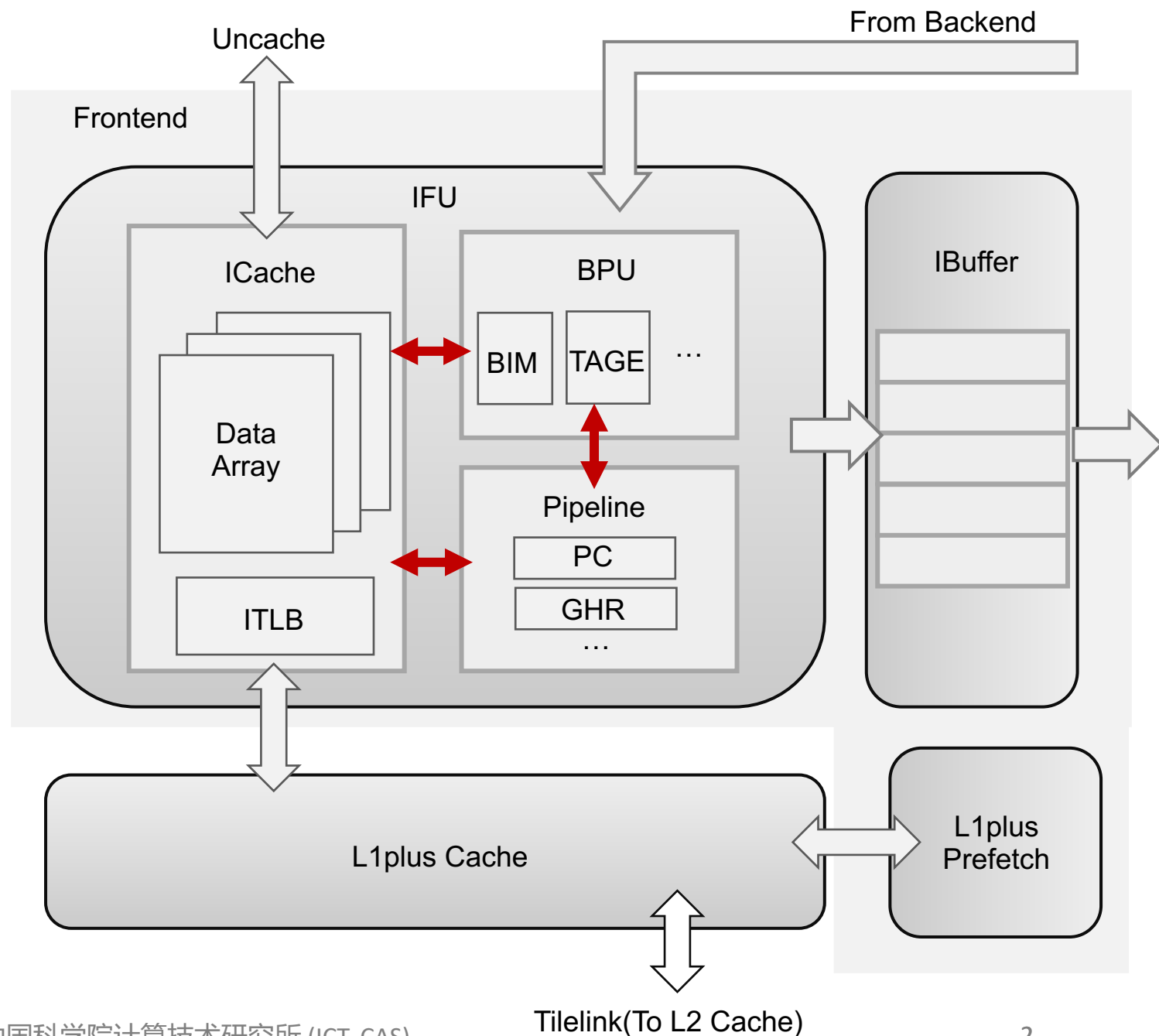
■取指单元设计

- 四级流水线
- 分支预测前向覆盖

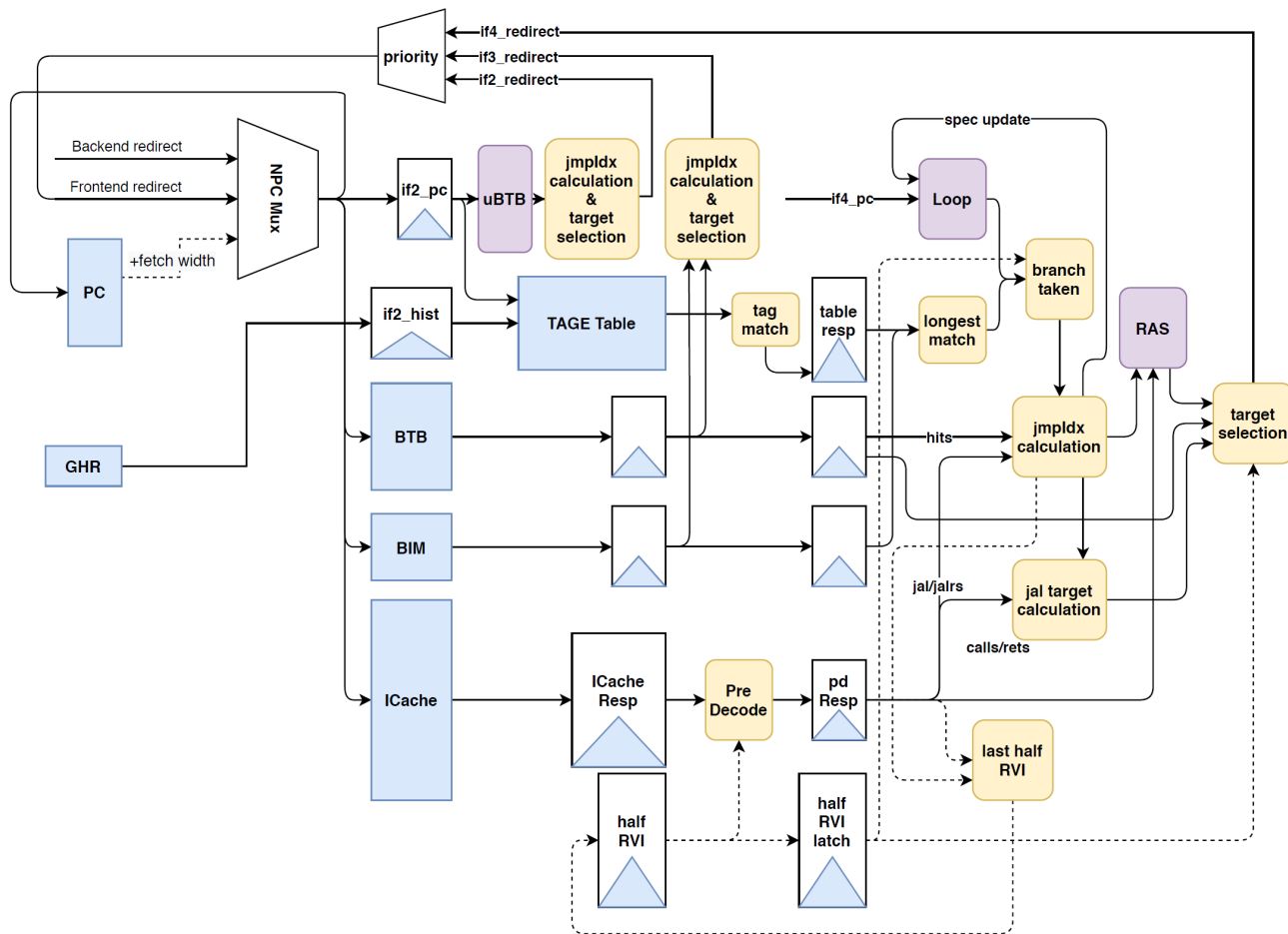
■指令缓存架构设计

- 指令端Memory Hierarchy
- 指令Cache设计
- L1plus Cache设计
- 指令预取器

■未来版本的设计方向



取指单元设计

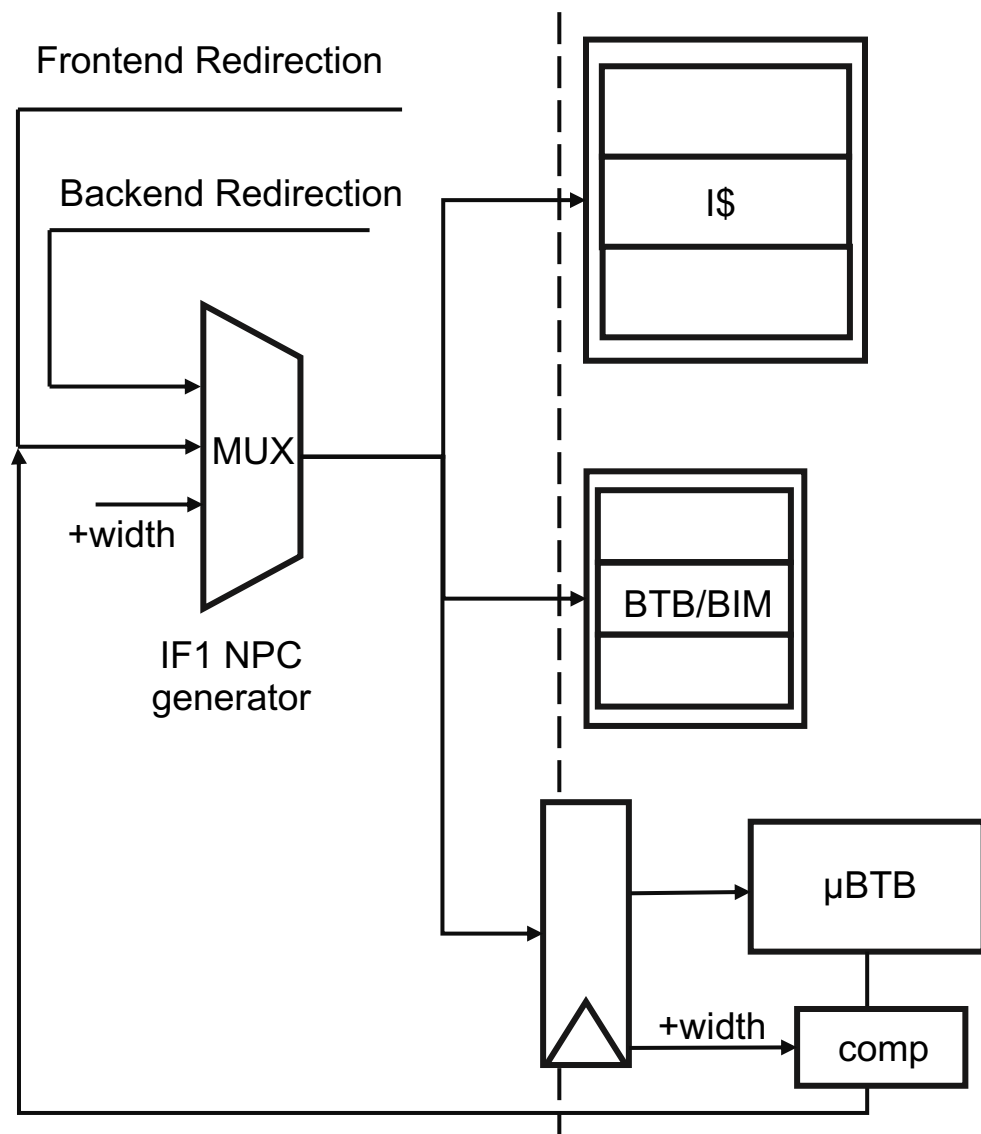


设计主要特点

- 参考BOOMv3的实现[*]
- 四级取指流水线 (IF1 , IF2 , IF3 , IF4)
- 多种预测器 (μ BTB , BTB , TAGE , RAS , Loop Predictor)
- 预测结果前向覆盖 (优先级重定向)

*: Zhao, Jerry, et al. "Sonicboom: The 3rd generation berkeley out-of-order machine." *Fourth Workshop on Computer Architecture Research with RISC-V*. 2020.

四级流水线与多种预测器 (1)



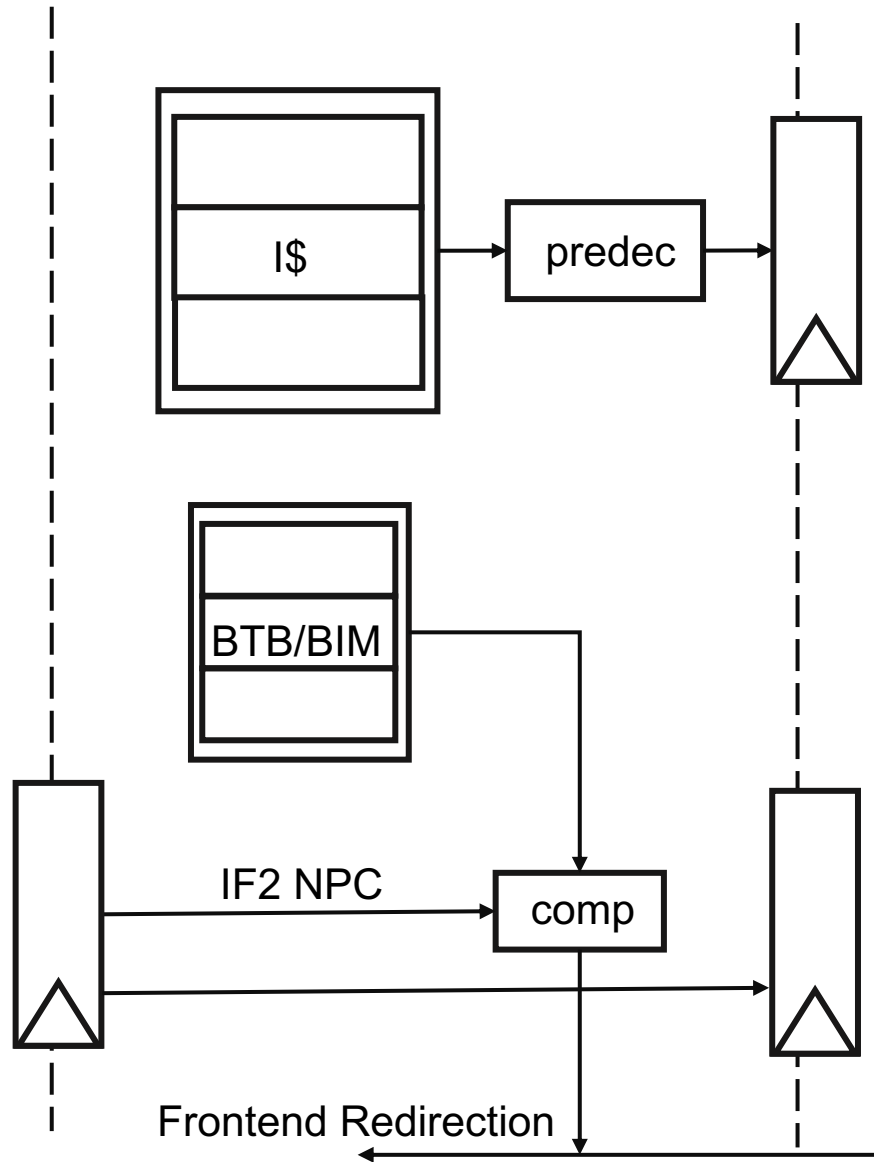
IF1

- 产生NEXT PC；来源包括：
 - 后端重定向（分支预测错误、Load Replay、处理器例外等）
 - 前端重定向（前向覆盖）
 - 顺序取指令（PC + fetch width）
- 发送指令Cache读请求（取指）
- 发送地址给BTB（预测）

IF2

- μBTB进行的方向+目标预测
- 若跳转则发送IF2重定向

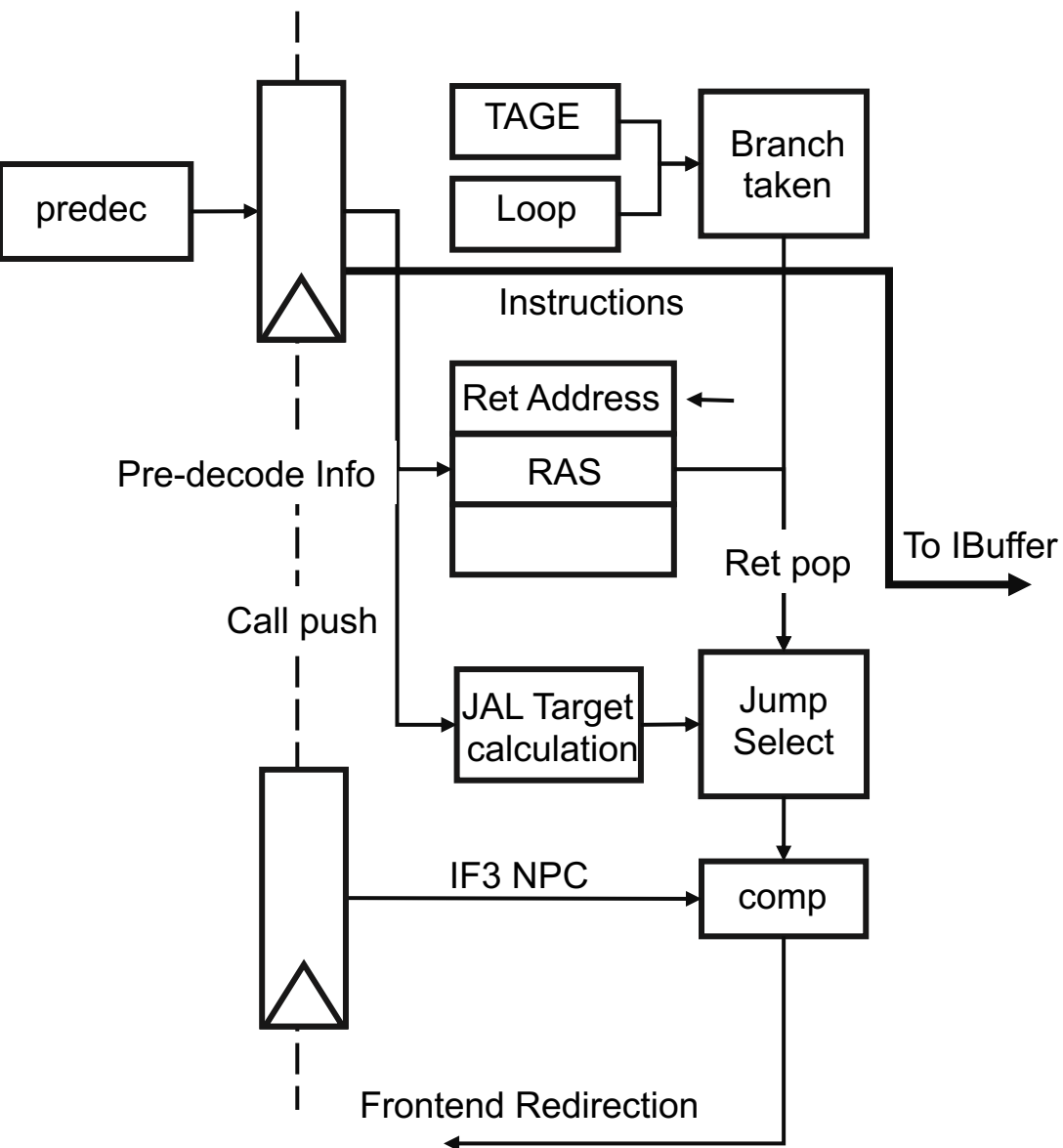
四级流水线与多种预测器 (2)



■ IF3

- 指令Cache回应，对指令进行预译码
- 得到BTB/BIM的预测结果
- 与IF2的预测结果进行比较，若不同则要发送IF3重定向

四级流水线与多种预测器 (3)



IF4

- TAGE、Loop Predictor得到branch预测结果
- 使用预译码得到的指令信息
 - 计算jal、branch指令的目标地址
 - RAS中对call指令进行压栈，对ret指令进行弹出
- 根据指令信息和预测结果计算出第一个跳转指令的目标地址
- 与IF3的预测结果进行比较，若不同则要发送IF4重定向

分支预测前向覆盖

■ 出发点

越靠后出结果的分支预测器

- 得到的指令信息越多

IF4直接可以根据预译码信息计算出准确的target

- 允许的容量更大（可以使用同步SRAM搭建）
- 预测算法更复杂

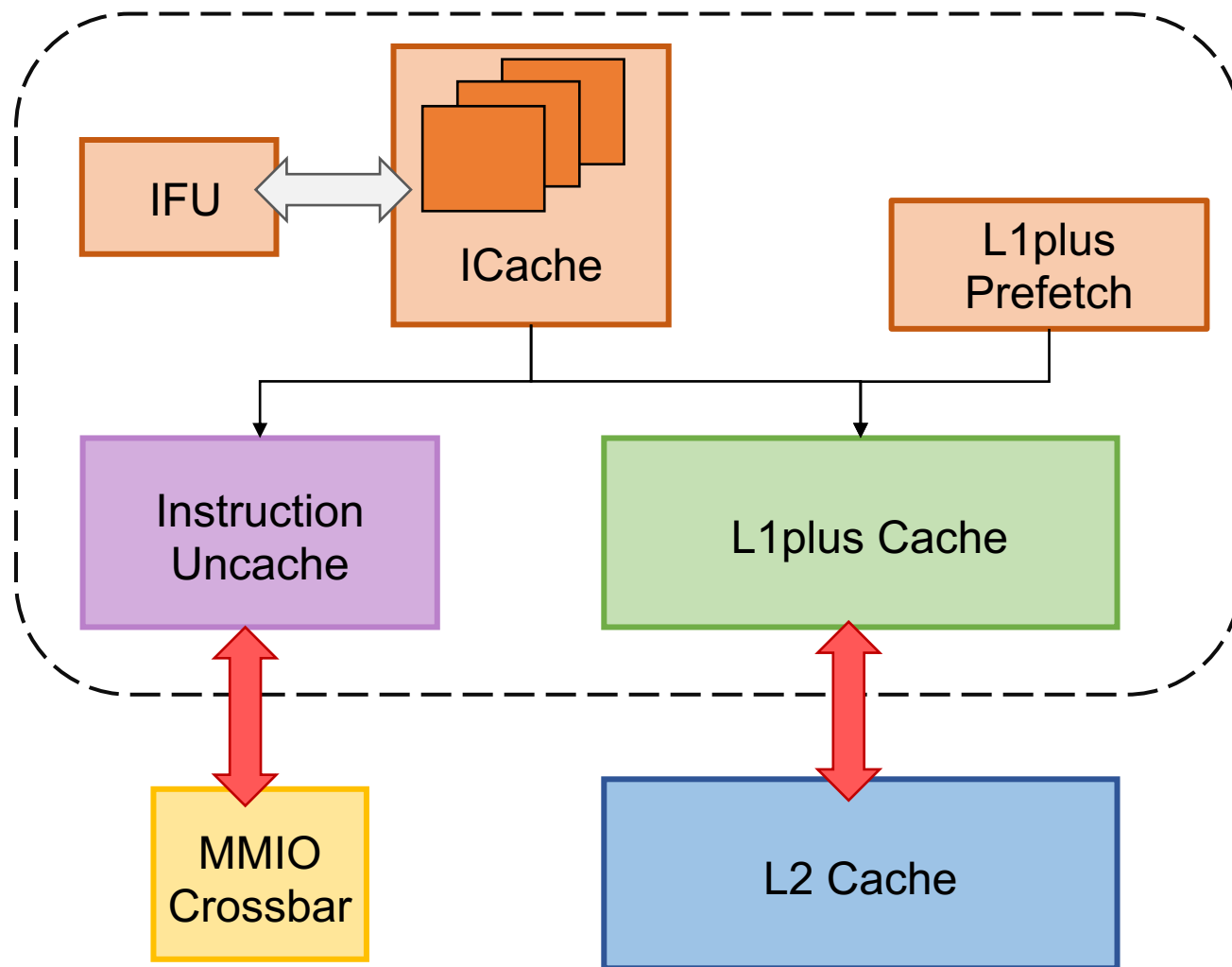
BIM基于两位饱和计数器，而TAGE基于全局历史查表

越靠后出结果的分支预测器具有更高的预测准确率

■ 目的

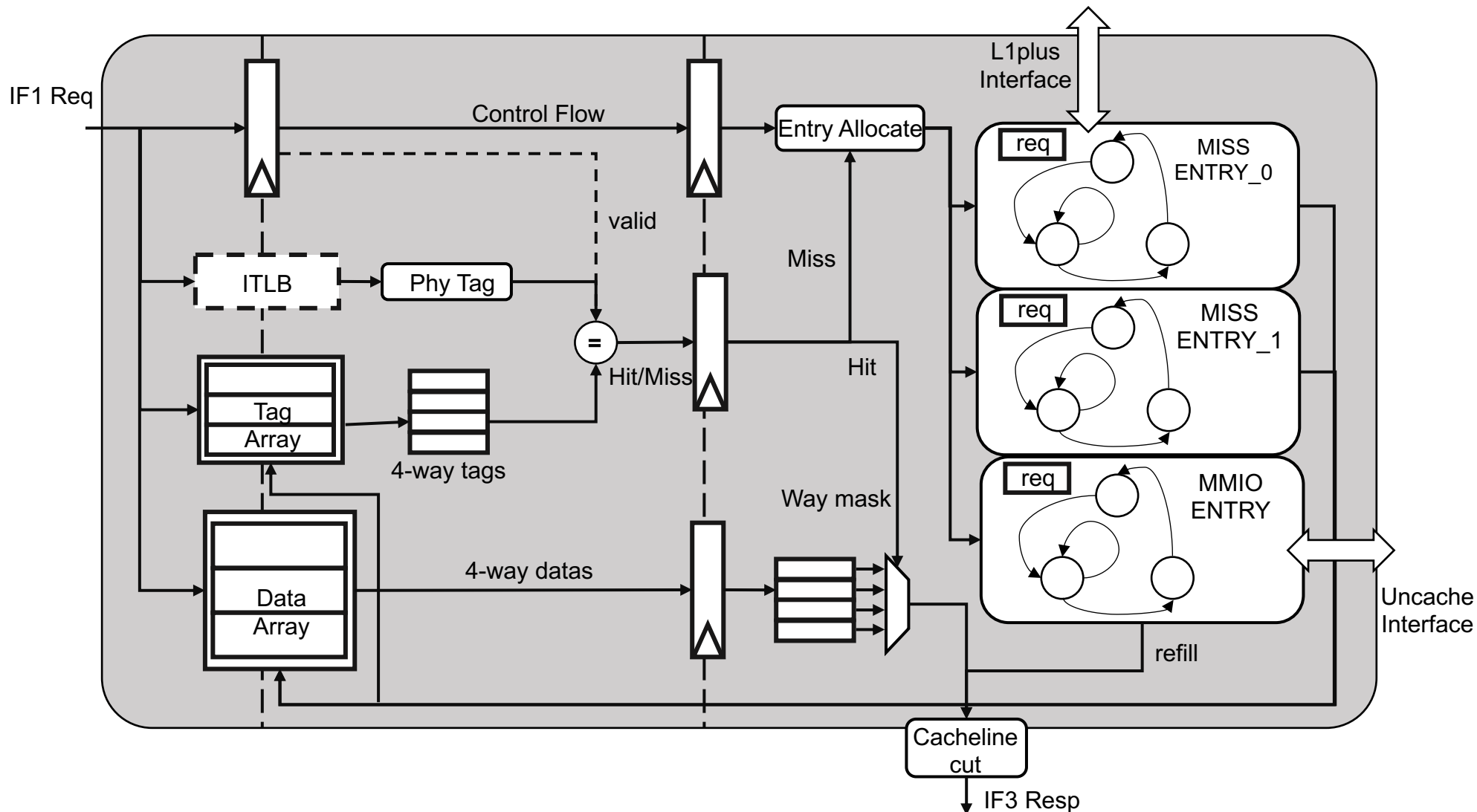
- 尽可能早地纠正分支预测错误

指令缓存架构设计



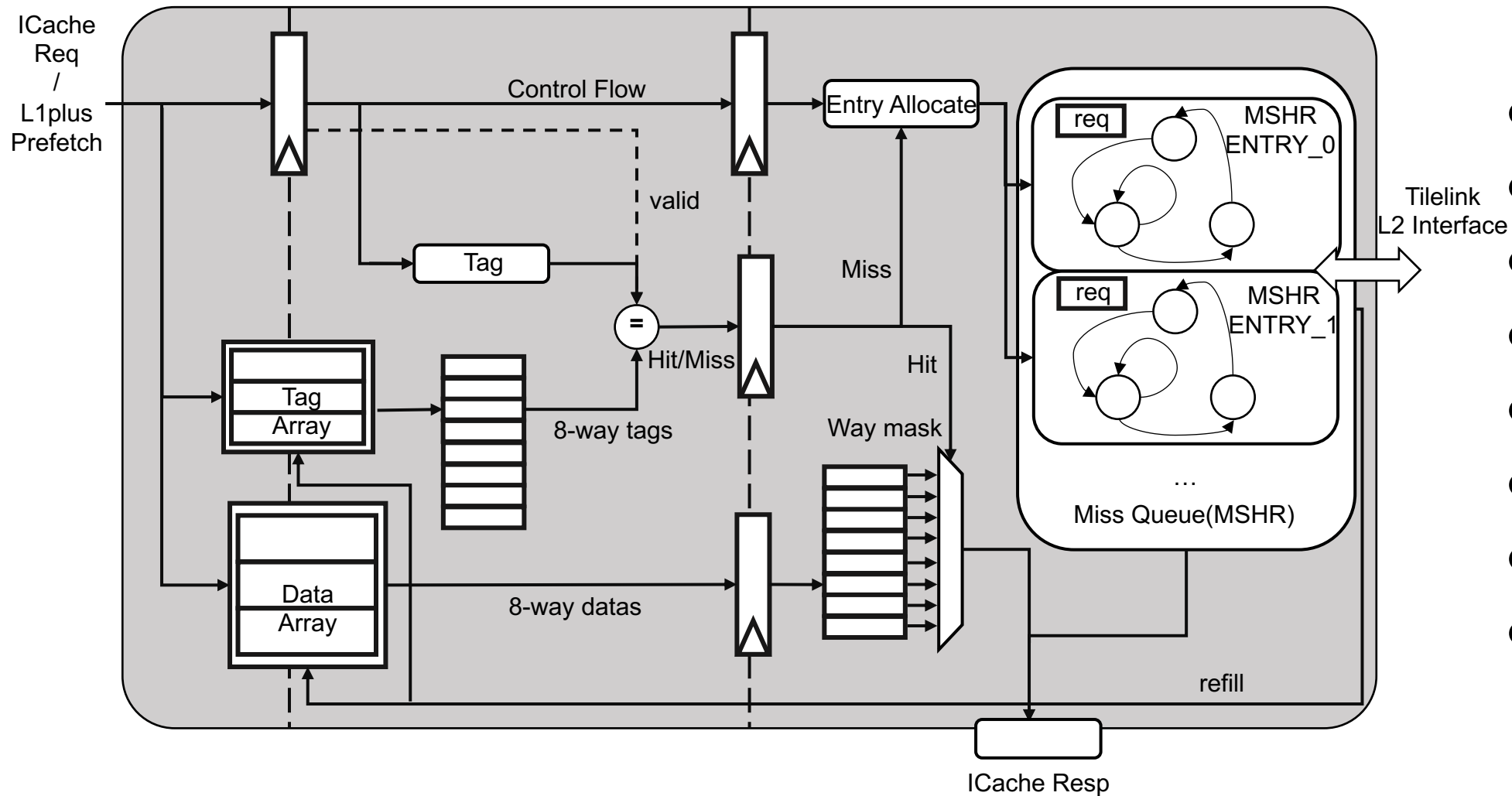
- 16KB , 4路组相联指令Cache
- 128KB , 8路组相联L1plus Cache
- 两者维护Inclusive的关系
- 两者之间采用自定义的协议
- 支持MMIO空间取指（从flash取指令）

指令Cache设计



- 4路 16KB
- 3级流水线
- Blocking Cache
- VIPT
- 2项Miss Entries
+1项MMIO Entry
- 奇偶校验
- PLRU替换策略

L1plus Cache设计

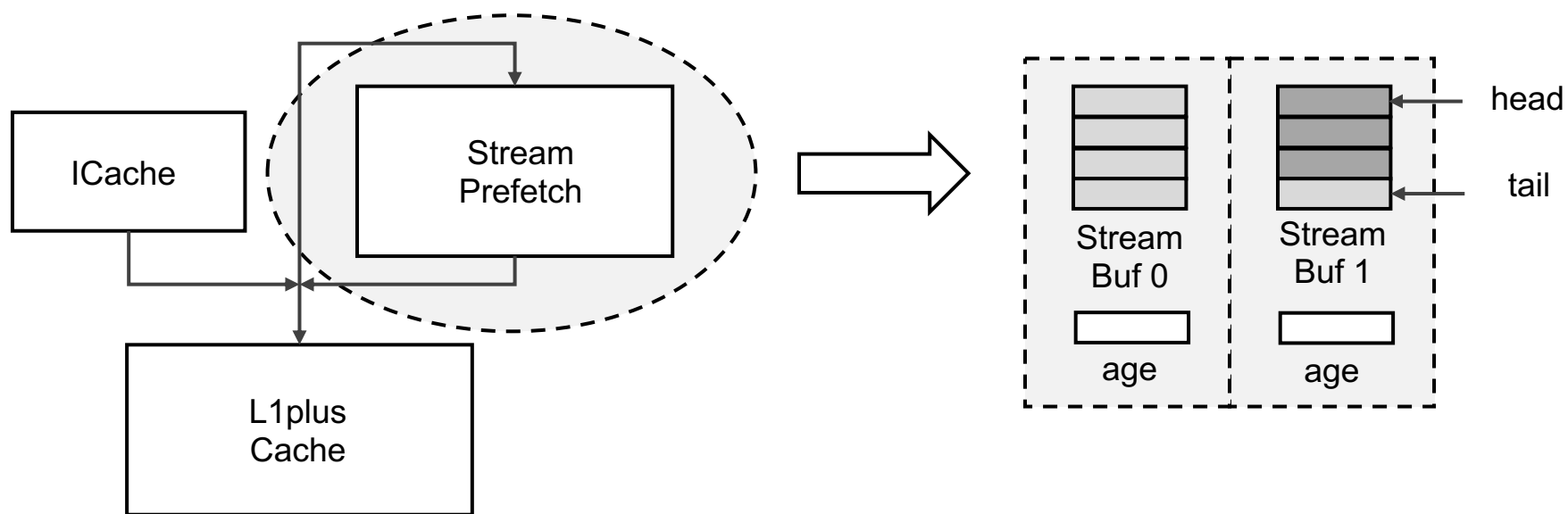


- 8路 128KB
- 3级流水线
- Blocking Cache
- PIPT
- Tilelink Node
- ECC校验
- Stream预取器
- PLRU替换策略

指令预取器

■流预取

- 侦听ICache Miss请求
- 支持两条指令流并行预取，流深度为4
- 预取数据存入L1plus Cache（预取器不存储数据）



L1plus Cache设计考虑

■ 前端紧耦合下的Trade off

- 分支预测、指令Cache与取指单元紧耦合
 - 控制逻辑互相纠缠导致比较差的时序
 - 增大指令Cache带来时序上的困难
 - 容量小的指令Cache上的预取效果不好
-
- 解决方案：在与取指单元紧耦合的Cache之外再添加一个**额外的指令Cache**，两者维护**Inclusive**的关系，后者与前端采用一个自定义的快速传递缓存块的协议，同时**增大容量**和**提高关联度**。

未来版本的设计方向

■ 紧耦合→解耦

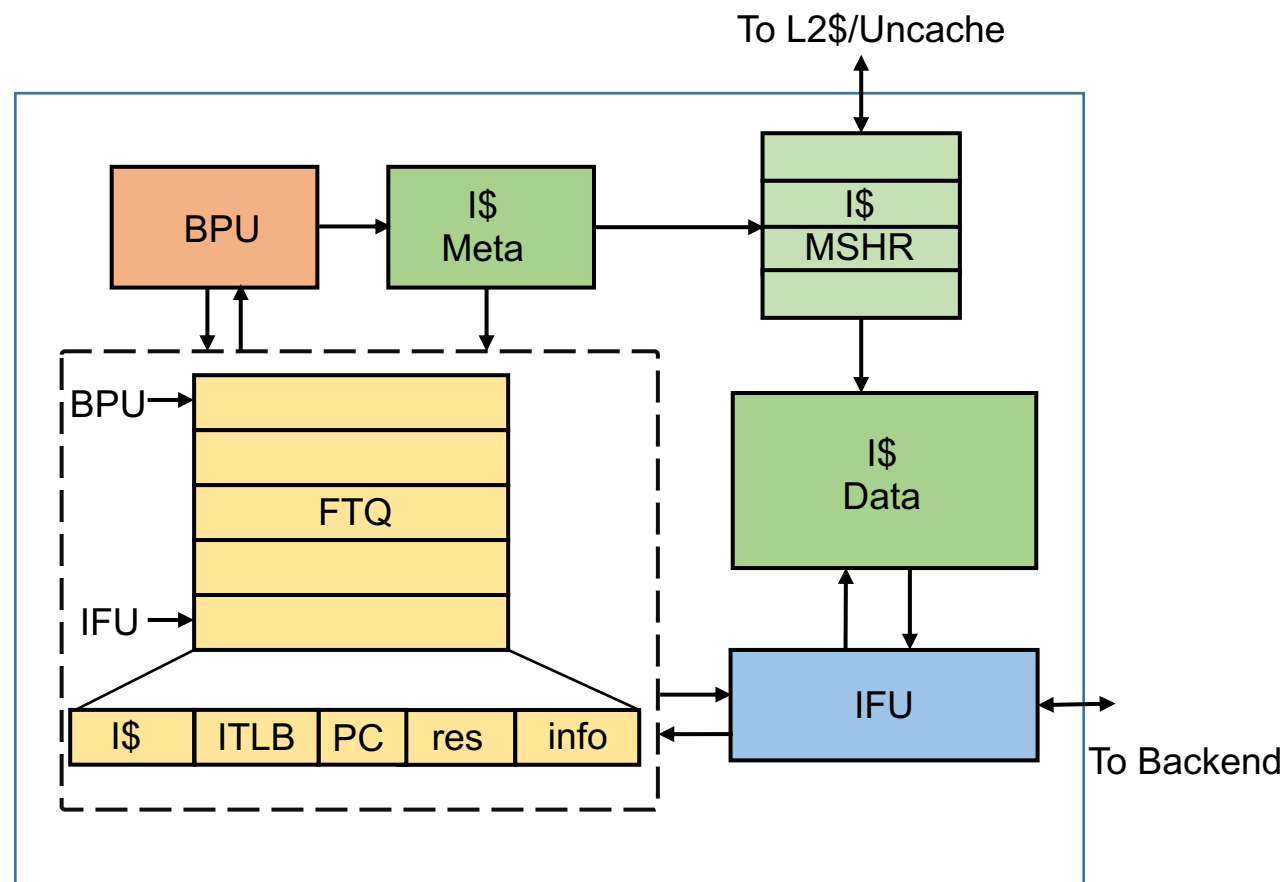
- 取指令和分支预测分开

■ 阻塞→非阻塞

- 指令Cache缺失时，分支预测继续执行
- 非阻塞的指令Cache

■ 简化缓存层次结构

- 去除L1plus Cache



感谢 

北京微核芯科技有限公司
BEIJING VCORE TECHNOLOGY CO., LTD.

提供产业经验、联合完成结构设计及物理设计

招募香山处理器二期联合开发合作伙伴



北京微核芯科技有限公司
BEIJING VCORE TECHNOLOGY CO., LTD.



ESWIN

优矽科技

欢迎更多伙伴加入！

联系人：李迪 13811881360