

计算机体系结构

实验2 分支预测器设计

江仲鸣



HITSZ 实验与创新实践教育中心
Education Center of Experiments and Innovations, HITSZ

- 加深对分支预测原理的理解，掌握分支预测器的软件实现方法
- 进一步熟悉程序插桩的原理，掌握Pin插桩工具的编写和使用方法
- 了解使用Pin插桩工具进行体系结构分析的方法



- 实现**BHT**、**全局历史**2种基本动态预测方法
- 实现**锦标赛**预测、**TAGE**预测2种复合预测方法
- 只需预测分支跳转方向，不需预测分支目标地址
- 要求分支预测器的实际大小不超过**33KB**

- 1、分支预测

- 与流水线相互配合，提高ILP
- 分类：静态分支预测、动态分支预测
- 静态分支预测
 - 总是预测跳转 or 总是预测不跳转
 - BTFN (Back Taken, Foreword Not Taken)
- Pros：易实现、开销小； Cons：准确度较低

• 2-1、基于BHT的分支预测

- 采用分支历史表记录分支历史，并以此预测分支行为

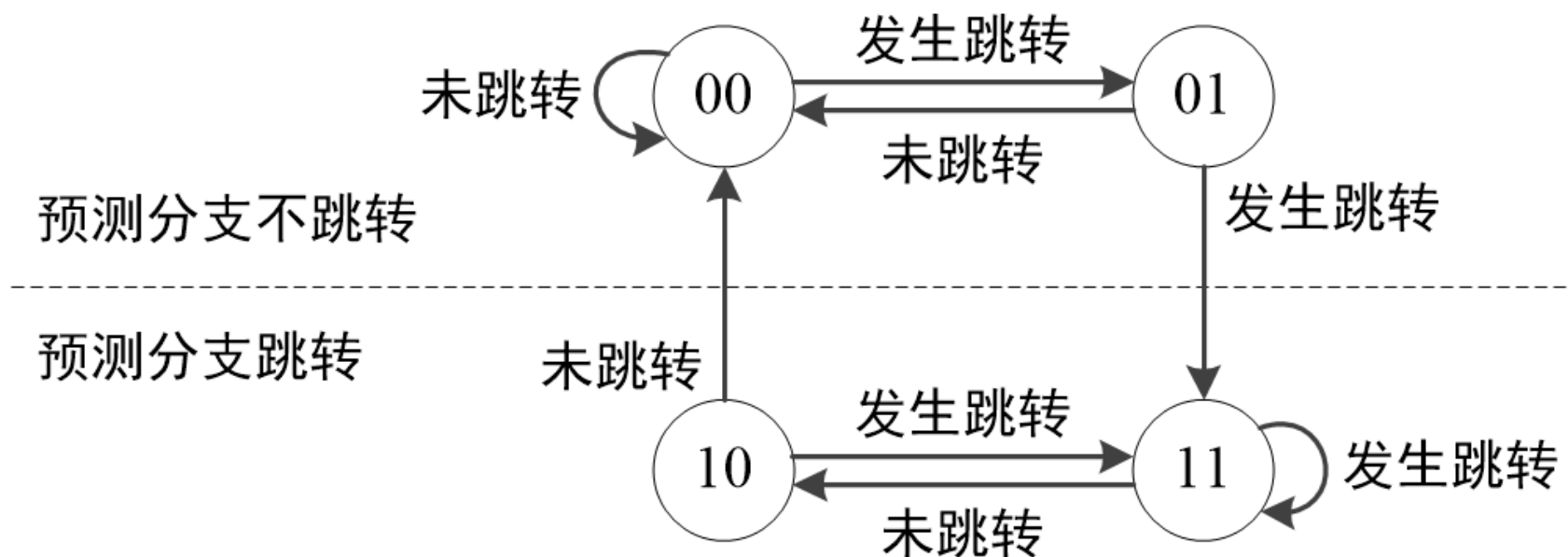
BHT:

Tag	分支历史(2bit)
A_0	H_0
A_1	H_1
...	...
A_{k-1}	H_{k-1}

- Tag —— 分支指令地址的一部分，类似于Cache的Tag
- 分支历史 —— 2bit饱和计数器

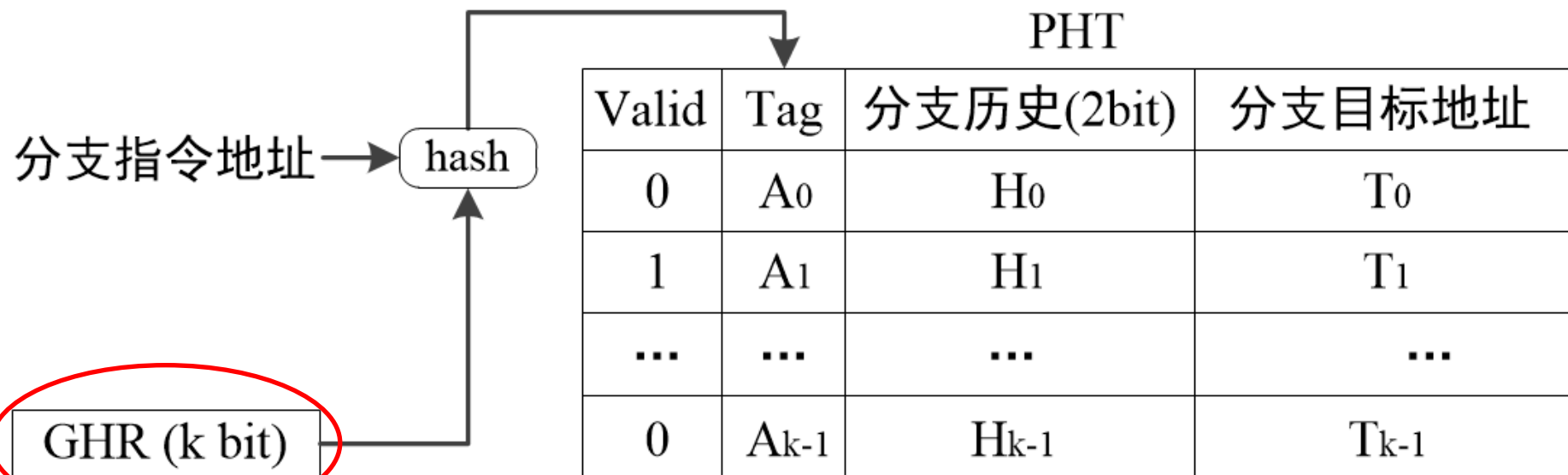
• 2-1、基于BHT的分支预测

- 采用分支历史表记录分支历史，并以此预测分支行为
- 先用指令地址查BHT，再根据分支历史预测是否跳转



• 2-2、基于全局历史的分支预测

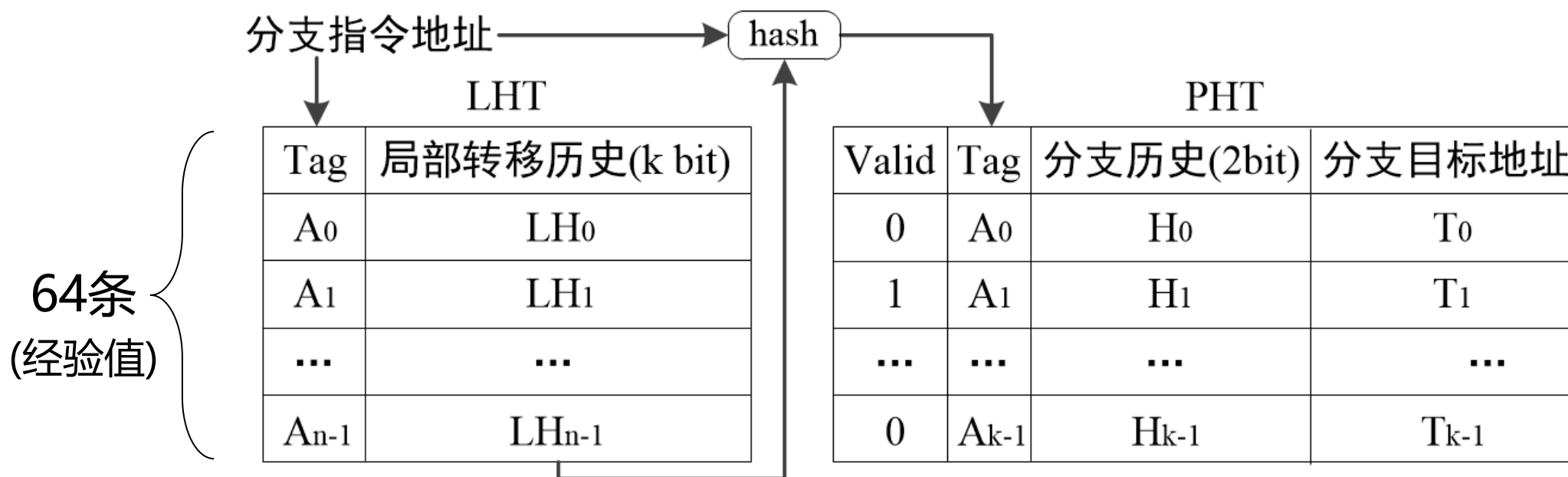
- BHT方法忽视了分支指令之前的关联性
- 使用GHR关联所有分支指令，使用PHT记录分支历史行为



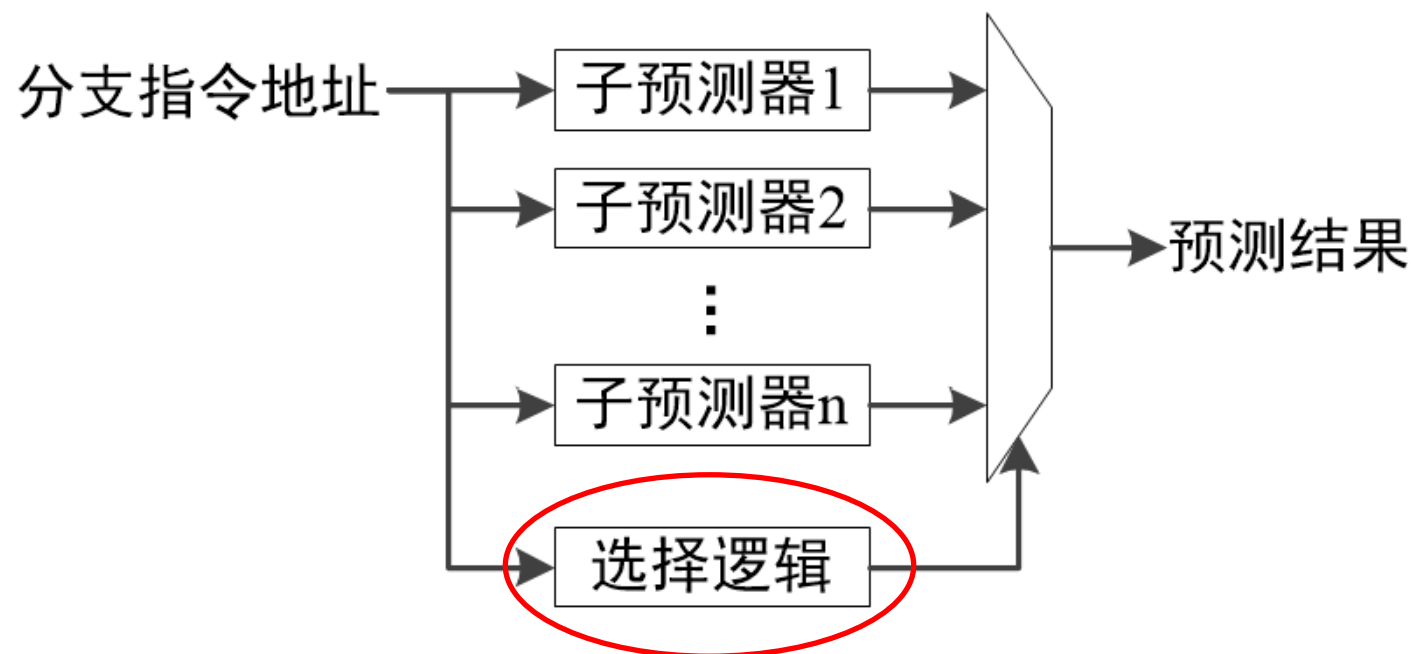
跳转则移进1，不跳转移进0

• *2-3、基于局部历史的分支预测

- 全局历史方法只能记录一个局部
- 使用LHT关联局部的分支指令，使用PHT记录分支历史行为



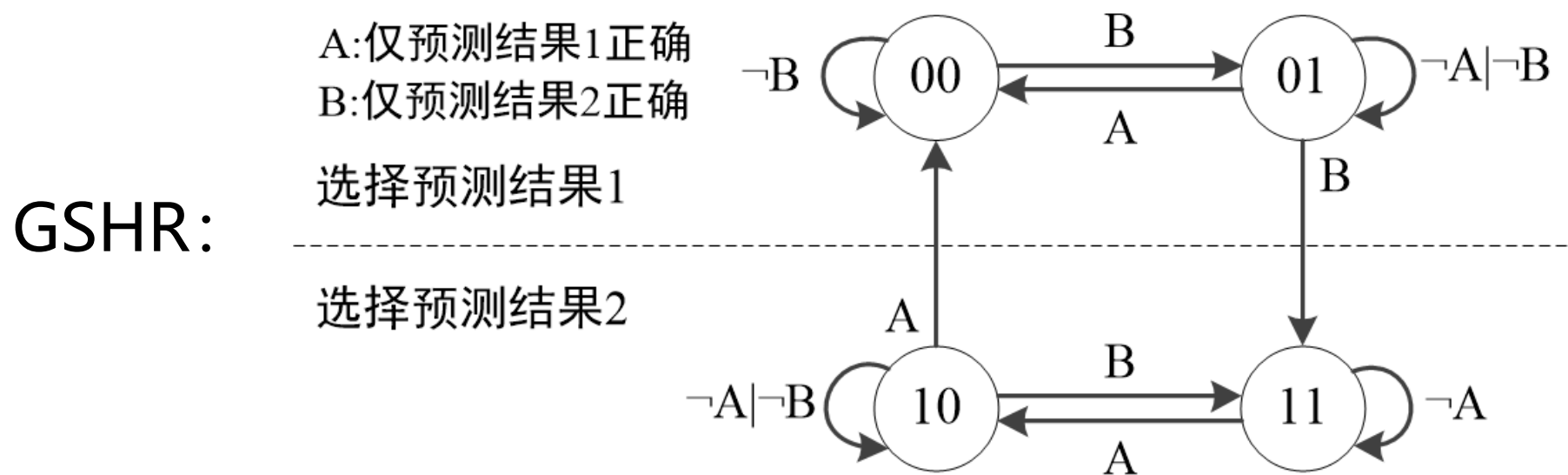
• 2-4、锦标赛分支预测 —— 集各家所长



- 选择策略 {
 - 基于全局选择历史选择
 - 基于局部选择历史选择

• 2-4、锦标赛分支预测 —— 集各家所长

- 基于全局选择历史的选择策略



- 基于局部选择历史的选择策略

LSHT: 由Tag索引的LSHR表, 一般有4096条记录

• 2-5、TAGE分支预测

- partially TAgged Geometric history length branch predictor
- 基于BHT基础预测器与不同GHR长度的全局历史预测器来完成预测
 - 多个不同长度的GHR，可记录不同大小的分支局部性
 - 应用：RISC-V BOOM、AMD Zen2, etc.

• 2-5、TAGE分支预测

• 预测器结构:

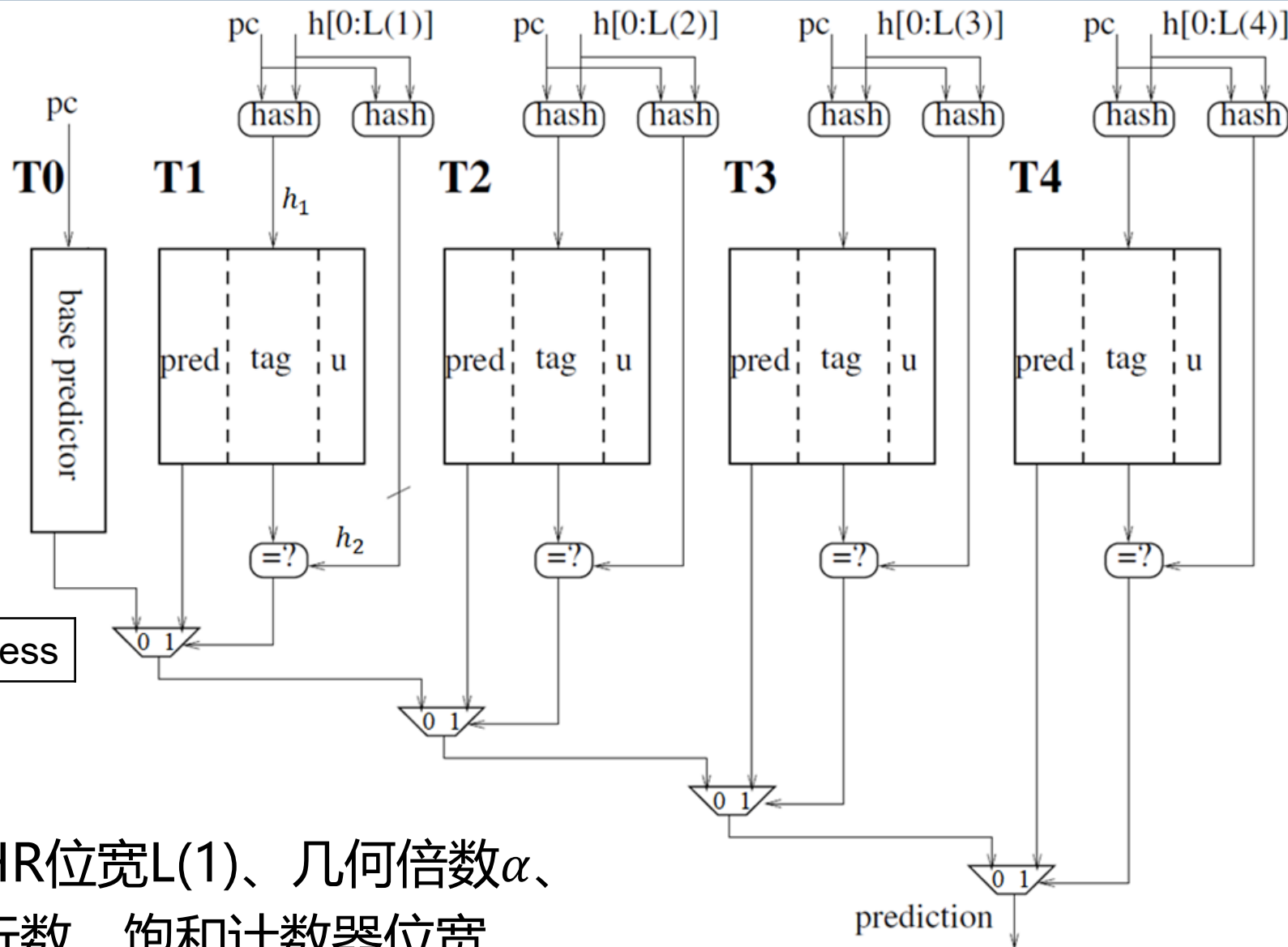
- T_0 : 基础预测器 (BHT)
- T_i : 全局历史预测器
 - $L(i) = \alpha^{i-1} \cdot L(1)$

entry结构:

3bit	x bit	2bit
pred	tag	usefulness

• 可配置参数:

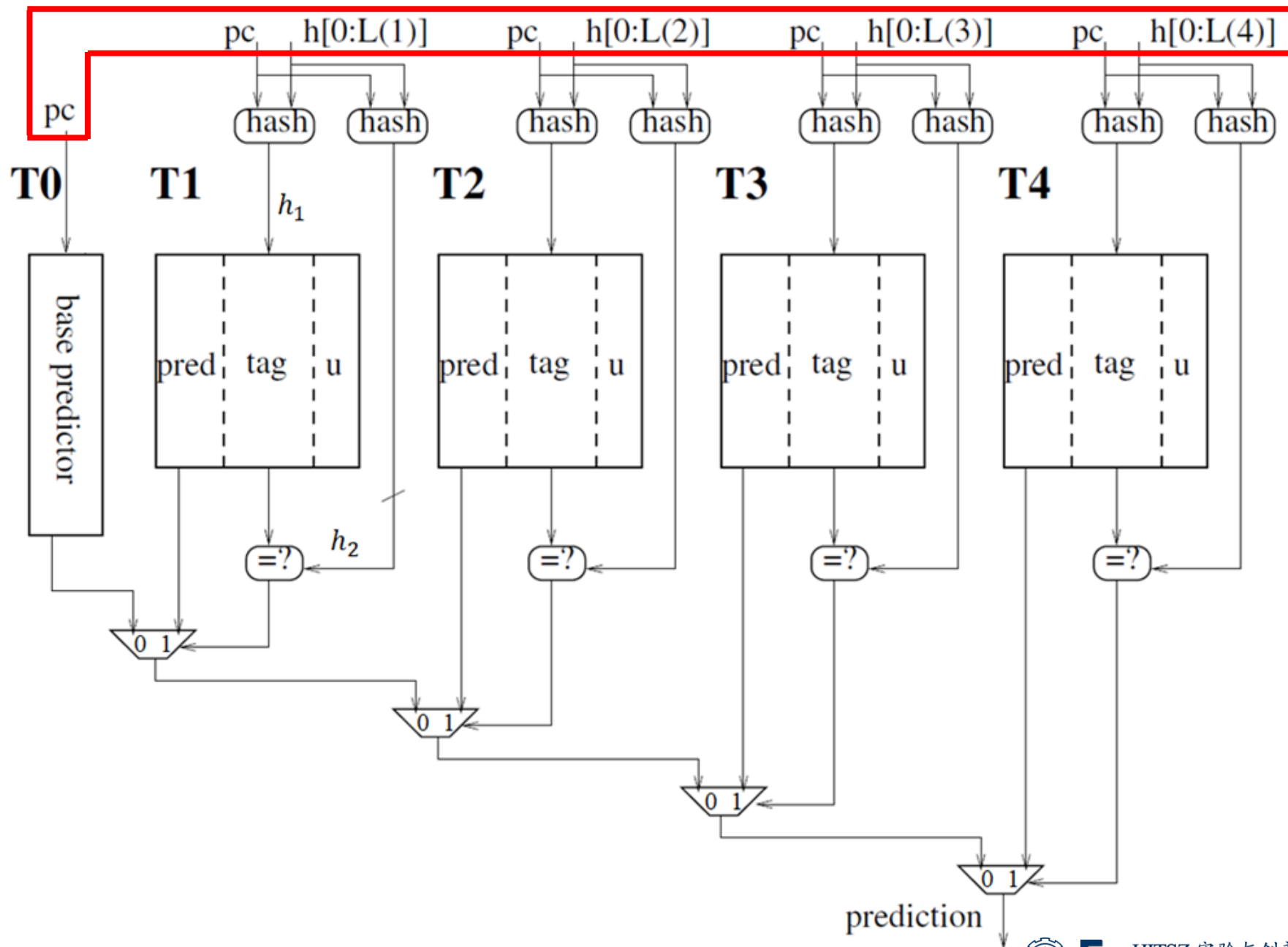
- 子预测器个数M、 T_1 的GHR位宽L(1)、几何倍数 α 、 T_0 的BHT行数、 T_i 的PHT行数、饱和计数器位宽、usefulness复位周期、hash函数、Tag位宽, etc.



• 2-5、TAGE

• 预测过程:

- ① 将PC送至各个子预测器



• 2-5、TAGE

• 预测过程：

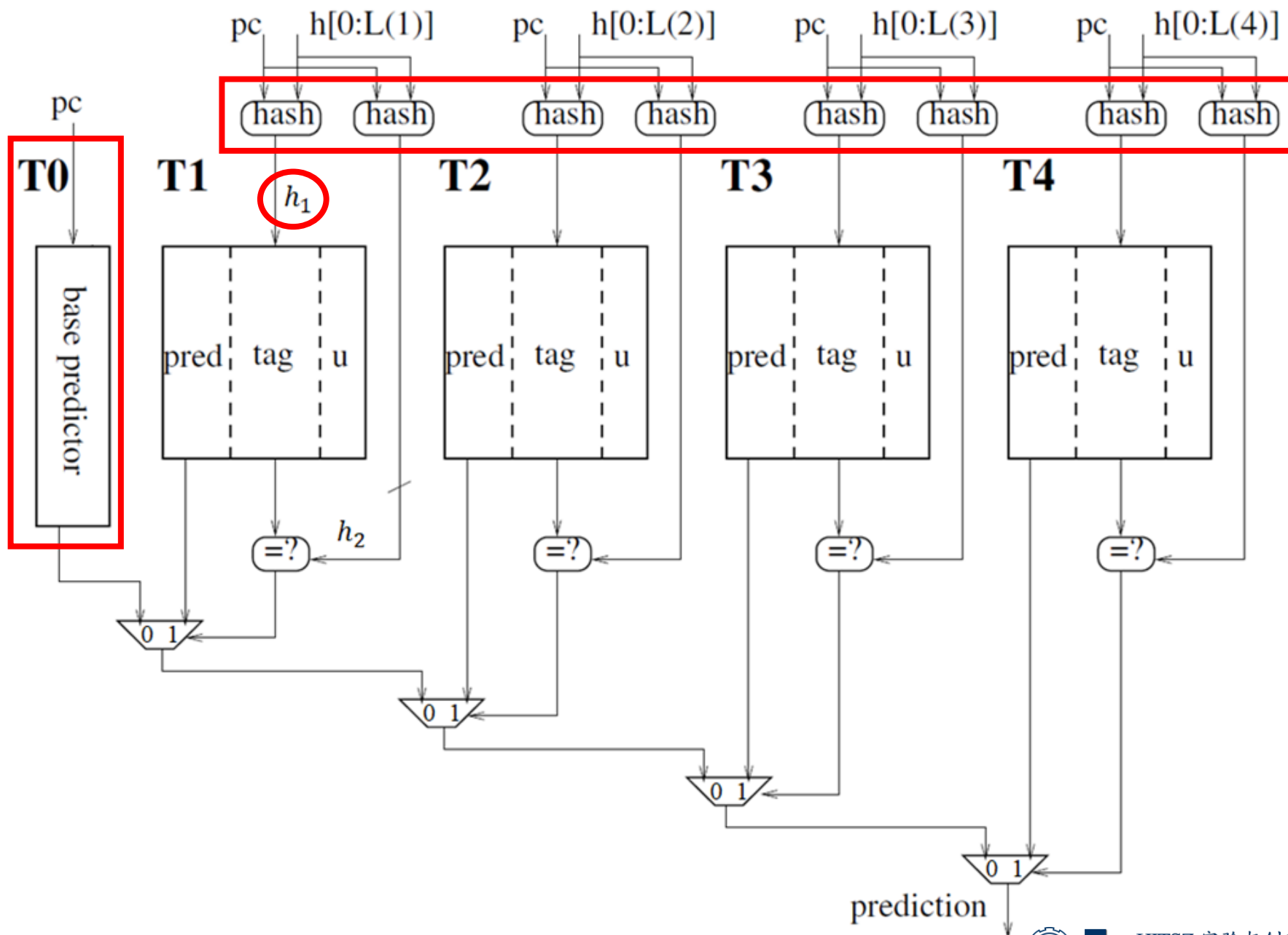
② 各子预测器进行

预测：

T_0 提供基础预测

T_i 根据hash值 h_1

查PHT表作预测



• 2-5、TAGE

• 预测过程：

③ 根据子预测器的Tag匹配情况，选出最终预测

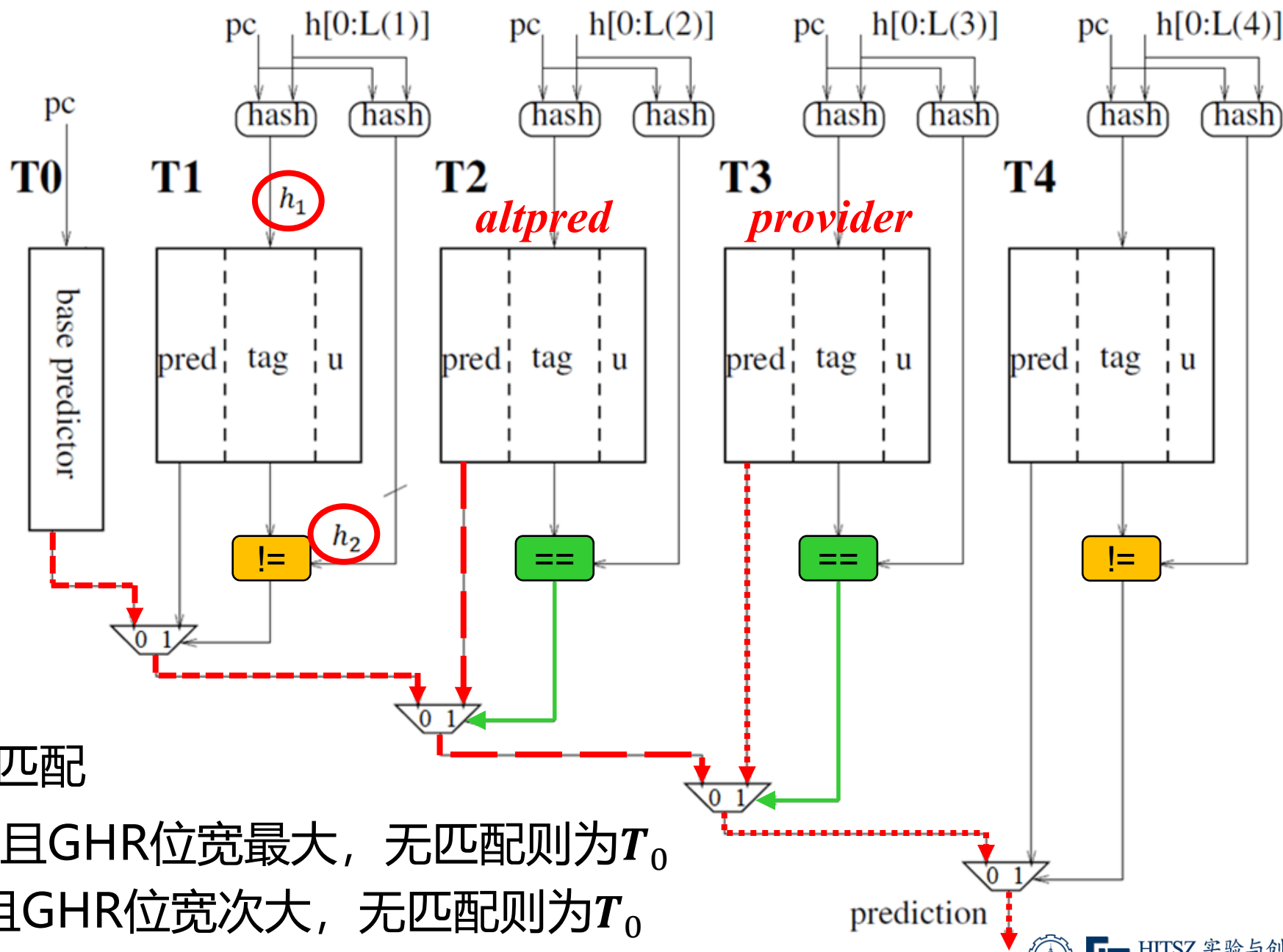
• Tag匹配：

$$\text{tag}[h_1] == h_2$$

假设 T_2 、 T_3 发生Tag匹配

➤ *provider*: Tag匹配且GHR位宽最大，无匹配则为 T_0

➤ *altpred*: Tag匹配且GHR位宽次大，无匹配则为 T_0



• 2-5、TAGE分支预测

• 更新策略

① 更新 $provider$

- 更新对象：发生Tag匹配的一条entry
- $pred$ 更新：预测成功， $++pred$ ；预测失败， $--pred$
- $usefulness$ 更新：仅当 $provider$ 、 $altpred$ 预测不同时更新
 - 预测成功， $++usefulness$ ；预测失败， $--usefulness$

② 定期重置 $usefulness$

- 防止某些entry被一直标记为 $useful$ ，从而影响预测器的更新效果
- 每预测N条分支指令时，将所有 $usefulness$ 的高、低位分别清零

• 2-5、TAGE分支预测

• 更新策略

③ 预测失败时，分配entry

- 更新对象： $T_i \in A$, $A = \{t | t \text{ 是HR 位宽比} \textit{provider} \text{ 大的子预测器}\}$
 - 令 $B = \{t' | t' \in A \wedge t' \text{ 对应entry的} \textit{usefulness} \text{ 为} 0\}$
 - 若 $B \neq \emptyset$ ，在 B 中找到GHR位宽最小的子预测器进行更新
 - 将对应entry的 \textit{pred} 设置为 $\textit{weaktaken}$
 - 对于3bit饱和计数器， $\textit{weaktaken}$ 即3'b100
 - 若 $B = \emptyset$ ，将 A 中所有子预测器的对应entry的 $\textit{usefulness}$ 减1

• 2-5、TAGE分支预测

- 相比其他预测器的改进之处

- 同时采用多个不同GHR位宽的子预测器，记录不同大小的局部性
- 增加*usefulness* 字段，为预测器更新提供指引
 - *usefulness* 不为0则不被分配 —— 模拟LRU
 - *usefulness* 定期清零 —— 剔除使用率不高的entry、防止过拟合
- 分配entry时选择最“靠左”的子预测器 —— 防止乒乓效应、保证预测效果

- 1. 补全brchPredict.cpp代码，编写预测器

```
if (INS_IsControlFlow(ins) && INS_HasFallThrough(ins))
{
    INS_InsertCall(ins, IPOINT_TAKEN_BRANCH, (AFUNPTR)predictBranch,
        IARG_INST_PTR, IARG_BOOL, TRUE, IARG_END);

    INS_InsertCall(ins, IPOINT_AFTER, (AFUNPTR)predictBranch,
        IARG_INST_PTR, IARG_BOOL, FALSE, IARG_END);
}
```

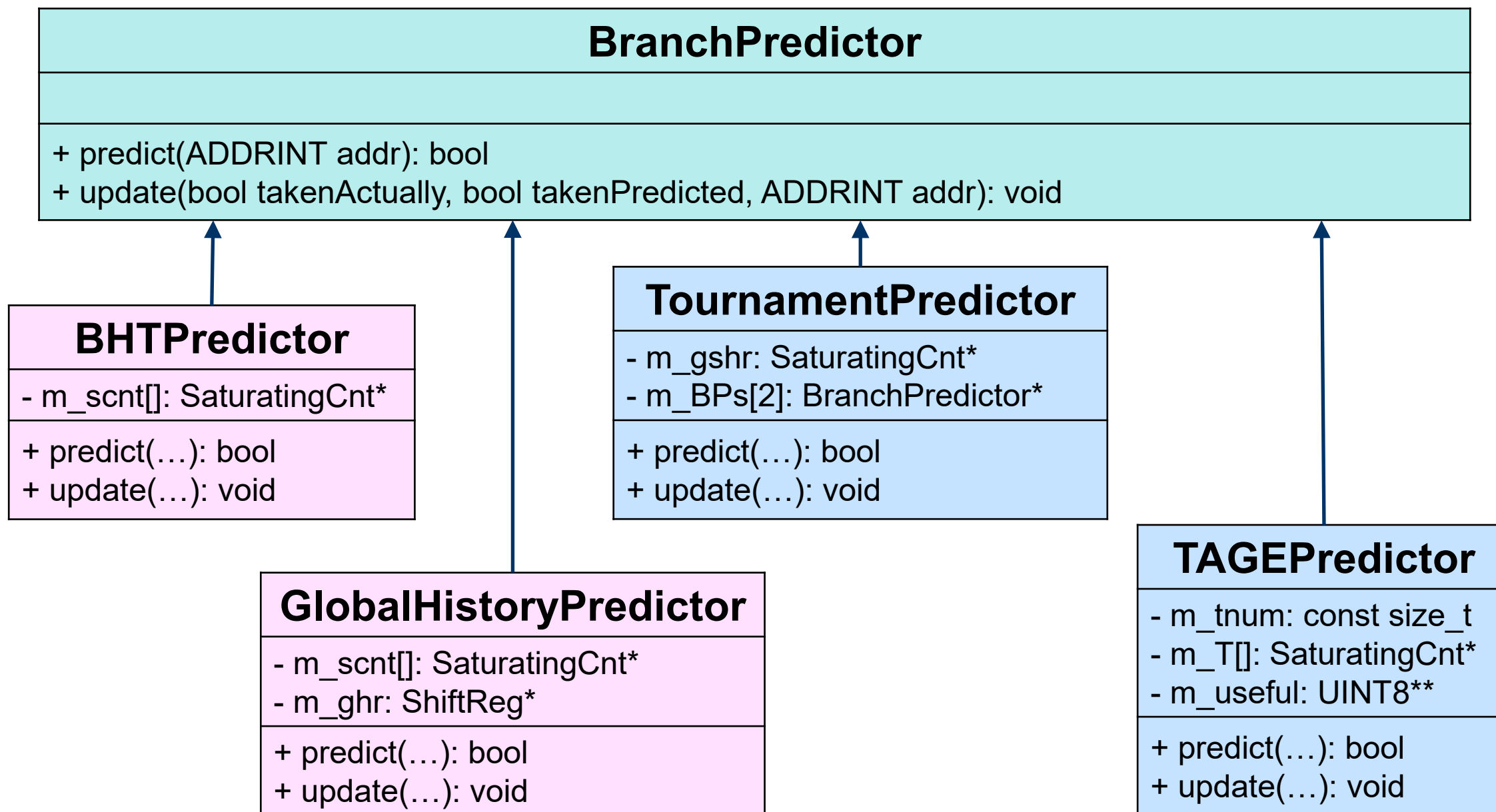
```
BOOL prediction = BP->predict(pc);
BP->update(direction, prediction, pc);
if (prediction)
{
    if (direction)
        takenCorrect++;
    else
        takenIncorrect++;
}
else
```

```
int main(int argc, char * argv[])
{
    // TODO: New your Predictor below.
    // BP = new BranchPredictor();
}
```

```
template<UINT128 (*hash)(UINT128 addr, UINT128 history)>
class GlobalHistoryPredictor: public BranchPredictor
```

```
// TODO: New your Predictor below.
BP = new GlobalHistoryPredictor<f_xor>(...);
```





- 2. 测试与分析
 - 测试程序：SPEC CPU 2006
 - 填写测试表格，对比分析结果

准确率 预测器 \ bench- mark	bzip2	sjeng	wrf	sphinx3
BHT预测器				
全局历史预测器				
锦标赛预测器				
TAGE预测器				

- 题目1：实现分支目标预测（+2分）
 - 修改必做题的锦标赛预测器代码，使其支持分支目标地址的预测
 - 要求：
 - 测试预测准确率
 - 在实验报告中，将设计思路、测试结果列成表格并分析实验结果
 - 关键：获取分支指令的目标地址
 - IARG_BRANCH_TARGET_ADDR

- **题目2：实现TAGE的其他变体（+2分）**
 - 自行查阅论文，实现TAGE变体
 - 要求：
 - 变体的预测性能需明显高于TAGE
 - 在实验报告介绍变体的架构、预测过程、更新策略
 - 测试预测准确率，以图表形式写入实验报告并分析实验结果

- 课堂检查：4个benchmark的测试表
- 将**源码**、**实验报告**打包提交
 - 命名规则：**学号_姓名_ARCH实验2.zip**
 - 提交方法：<https://hitsz-cslab.gitee.io/arch/ojguide>
 - Deadline：下周同一上课时间前
- 附加题：将设计思路、关键代码等写入报告，与源码一起打包提交（**+2**分）

开始实验



HITSZ 实验与创新实践教育中心
Education Center of Experiments and Innovations, HITSZ