

# lab4 分支预测 实验报告

PB17111614

王嵘晟

## 1. 实验设计

- 分支预测器的设计：本实验只对 br 指令进行。分支预测器输入 IF 时获得的 PC, PC\_IF, 流水段更新后的 PC: PC\_EX+4, 以及 br 指令的目标地址 br\_target, 将 br 跳转当作输入的 update 信号, 输出施行分支预测后的 NPC, 预测失败信号 br\_predict\_miss, 以及预测进行信号 br\_predict\_result。分支预测器中使用 BTB 或者 BHT 两种方法来实现
- BTB 的设计：维护4个数组来实现对于分支情况的储存, 并根据数组中的值生成 BTB\_hit BTB\_br 信号, 分别表示 BTB 策略成功预测和 BTB 策略成功预测了 br 指令, 最后将预测结果的 pc\_predict 输出。

```
reg [19:0] TAG [SET_ADDR_SIZE-1:0]; // 存储高位地址
reg [31:0] DATA [SET_ADDR_SIZE-1:0]; // 存储预测的 PC
reg VALID[SET_ADDR_SIZE-1:0]; // 当前数组单元被更新则 VALID 为1, 表示有效
reg ISBR [SET_ADDR_SIZE-1:0]; // 是否是 BR 指令

assign BTB_hit = ( (TAG[query_addr] == query_tag) && (VALID[query_addr] == 1'b1) )? 1'b1 : 1'b0;
assign BTB_br = ( (TAG[query_addr] == query_tag) && (VALID[query_addr] == 1'b1) && (ISBR[query_addr] == 1'b1) )? 1'b1 : 1'b0;
assign pc_predict = DATA[query_addr];
```

- BHT 的设计：BHT 的方法要基于 BTB, 当 BTB 成功预测后, 需要使用 4 阶段状态机来判断是否预测有效, 当两种方法都成功预测时, 分支预测器才会输出做出成功预测的 pc 值。

```
assign BHT_br = (STATE[query_addr] > 2'b01) ? 1'b1 : 1'b0;
```

## 2. BHT 表格

BTB	BHT	REAL	NPC_PRED	flush	NPC_REAL	BTB UPADTE
Y	Y	Y	BUF	N	BUF	N
Y	Y	N	BUF	Y	PC_EX+4	N
Y	N	Y	PC_IF+4	Y	BUF	N
Y	N	N	PC_IF+4	N	PC_EX+4	N
N	Y	Y	PC_IF+4	Y	Br_target	Y
N	Y	N	PC_IF+4	N	PC_EX+4	N
N	N	Y	PC_IF+4	Y	Br_target	Y

BTB	BHT	REAL	NPC_PRED	flush	NPC_REAL	BTB UPADTE
N	N	N	PC_IF+4	N	PC_EX+4	N

### 3. 实验结果

- btb.s

策略	总周期数	分支指令数目	分支预测正确次数	分支预测错误次数
不使用	512.5	-/-	-/-	-/-
BTB	312	101	98	2

- bht.s

策略	总周期数	分支指令数目	分支预测正确次数	分支预测错误次数
不使用	538.5	-/-	-/-	-/-
BHT	366	110	91	15

- QuickSort.s

策略	总周期数	分支指令数目	分支预测正确次数	分支预测错误次数
不使用	54174.5	-/-	-/-	-/-
BTB	61830.5	11598	1087	2063
BHT	63321	13048	1227	2257

- MatMul.s

策略	总周期数	分支指令数目	分支预测正确次数	分支预测错误次数
不使用	336296.5	-/-	-/-	-/-
BTB	333955	4896	3787	818
BHT	351785	5168	4576	566

### 4. 分析与总结

通过 4 个样例的比对，可以发现：

- 当样例代码比较简单时，使用动态分支预测可以有效的缩短程序的运行时间
- 对于快速排序，使用动态分支预测的运行总周期数都比不使用分支预测大，这是分支预测带来的开销
- 对于矩阵乘法，使用 BTB 策略对于减少程序运行周期起到了一定的作用，但 BHT 策略依然增加了运行的总周期数。所以可以说明 BHT 方法比 BTB 方法带来的分支代价更大
- 对于同一种测试样例，使用不同的分支预测方法：可以发现使用 BTB 的分支预测正确率要远低于使用 BHT，但 BHT 的运行周期更长。因为 BHT 方法是基于 BTB 的实现，所以 BHT 通过增大开销的方法来提高分支预测正确率

### 实验总结：

- 本次实验用 BTB 和 BHT 两种方法来实现动态分支预测，使得我们对于 RV32I CPU 中的分支预测有了更深入的理解
- 分支收益与分支代价是同时存在且相互依赖的，想要提高分支预测正确率必然会降低程序运行速率。在真正实现 CPU 时应当对二者做出相应的权衡以便得到更好的性能