Verilog-Lab4 实验报告

彭浩然 **PB19051055**

1 测试方法

针对每一个测试样例，我们使用三种不同分支预测策略（预测不跳转、BTB、BHT）分别运行，记录分支次数、预测错误次数和程序运行的总时钟周期数。

为了排除其他因素干扰，我们将cache尺寸调整得足够大以减少cache miss带来的开销。此处采用参数为8组、每组16路组相联，每路16个字。

另外，BTB在实现时采用直接映射方法，为了减少记录冲突带来的影响，我们将BTB的大小也设得足够大。经过实际测试，在采用末尾7位地址作为分组时（即记录128组，每组1个记录），在各个样例上均不会出现记录冲突。

2 在 `btb.S` 上的测试

策略	分支次数	预测错误次数	预测错误率	时钟周期
预测不跳转	101	100	99.01%	509
BTB	101	2	1.98%	311
BHT	101	2	1.98%	311

`btb.S` 是一个简单的单层循环，因此除了最后一次以外的每一次分支都应该跳转。在这种简单的模式下，BTB和BHT两种策略的效果均为除了第一次预测不跳转之外以后全部预测跳转，所以预测错误次数是一样的。

3 在 `bht.S` 上的测试

策略	分支次数	预测错误次数	预测错误率	时钟周期
预测不跳转	110	99	90.00%	535
BTB	110	22	20.00%	379
BHT	110	13	11.82%	361

`bht.S` 是一个双层循环，内外均从0循环到9。这个样例中BHT的效果开始优于BTB。这是因为在每次的外层循环中，内层循环退出前会有一次不跳转，而BTB策略会因为这个原因而在下一次外层循环中第一次执行内层循环时也预测不跳转，导致预测错误。BHT相对BTB更为稳定，在这个样例上表现更好。

4 在 `QuickSort.S` 上的测试

策略	分支次数	预测错误次数	预测错误率	时钟周期
预测不跳转	6756	1636	24.22%	35312
BTB	6756	1868	27.65%	35774
BHT	6756	1150	17.02%	34338

`QuickSort.S` 是一个快速排序算法。这个样例中，BTB的预测错误率甚至高于直接预测不跳转，而BHT的预测错误率也较高。初步推测这是因为快速排序算法中循环的主要任务是找出大小关系不符合要求的元素对并对其进行调换以完成partition，也就是说，循环执行多少次甚至是否执行取决于数组中元素的排列，而这在运行过程中是相对随机的。预测不跳转的错误率只有24.22%，这可以部分说明大部分循环其实执行次数较少甚至没有执行。在这种跳转情况随时容易改变的场景下，BTB策略只根据上次结果进行预测，会更容易出现抖动，导致更高的错误率。BHT则能够更好地学习在一段较长的时间内是否跳转的规律，因此在三个策略里效果最好。

5 在`MatMul.S`上的测试

策略	分支次数	预测错误次数	预测错误率	时钟周期
预测不跳转	4624	4350	94.07%	68264
BTB	4624	548	11.85%	60658
BHT	4624	278	6.01%	60118

`QuickSort.S`是一个快伪矩阵乘法算法。这个样例的循环次数全部都是固定的，而且循环次数较多（一般是16次），特征上类似`bht.S`。稳定性较好的BHT会在除第一次以外的所有情况下预测跳转，在这个样例上效果也优于其他策略。

6 总结

在上面的4个样例中，1次正确的预测平均可以减少2个时钟周期的运行时间，这与分支错误需要清空流水线F、D两级的特性是一致的。在循环次数较多且确定的情景下，BTB和BHT的效果均优于直接预测不跳转。而在分支本身与数据相关、较难预测的场景下，BTB和BHT的预测效果会下降。稳定性较差的BTB策略甚至可能会比直接预测不跳转更差。