

Computer Architecture Lab4

PB19071501 李平治

实验目的

- 实现BTB (Branch Target Buffer) 和BHT (Branch History Table) 两种动态分支预测器
- 体会动态分支预测对流水线性能的影响

实验环境

- Vivado 2019.1
- VMWare Fusion12.2.3虚拟机中的Windows 10

实验内容

I. 阶段一

在Lab3阶段二的RV32I Core基础上，实现BTB

在BTB.sv中维护一个buffer数组，用于储存预测的PC；同时维护一个valid数组和state数组，分别表示某位置处预测PC是否有效和是否跳转。

```
1 reg [32 - 1 : 0] buffer [BUFFER_SIZE];
2 reg valid [BUFFER_SIZE];
3 reg state [BUFFER_SIZE];
```

类似于Cache中的动机，地址同样被分为tag和set两部分，用于分组缓存中的标签和地址

```
1 wire [SET_ADDR_LEN - 1 : 0] read_addr, write_addr;
2 wire [TAG_ADDR_LEN - 1 : 0] read_tag, write_tag;
3 reg [TAG_ADDR_LEN - 1 : 0] tag_buffer [BUFFER_SIZE];
4 assign {read_tag, read_addr} = pc_rd;
5 assign {write_tag, write_addr} = pc_wr;
```

当tag匹配且valid为1时命中，进行预测

```
1 assign btb_hit = (tag_buffer[read_addr] == read_tag) && valid[read_addr];
2 assign btb_br = (tag_buffer[read_addr] == read_tag) && valid[read_addr] &&
  state[read_addr];
```

BTB接在BranchPredict.sv中，其中在EX段对是否命中进行了判断，并在miss时对BTB进行写回

```
1 always @ (*) begin
2     if (write) begin
3         if (br_pred_EX == br) begin
4             NPC = br_pred_IF ? btb_pc_predict : PC_rd_IF_4;
5             br_predict_miss = 0;
6         end else begin
7             NPC = br ? PC_br_target : PC_rd_EX;
8             br_predict_miss = 1;
9         end
10    end else begin
11        NPC = br_pred_IF ? btb_pc_predict : PC_rd_IF_4;
12        br_predict_miss = 0;
13    end
14 end
```

II. 阶段二

在阶段一的基础上，实现BHT

BHT和BTB一起接入BranchPredict.sv，并在BHT中维持一个2bit长的状态位；当状态超过阈值1时（即为10或11时），则进行跳转

```
1 reg [1 : 0] STATE [BUFFER_SIZE];
2 assign bht_br = (STATE[read_addr] > THRESHOLD) ? 1 : 0;
```

BranchPredict.sv中跳转的条件修改为BTB预测跳转和BHT达到阈值同时满足

```
1 assign br_pred_IF = btb_hit & bht_br;
```

III. 阶段三

1. 分支收益和分支代价

- 没有分支预测的情况下，一个分支跳转指令的代价为固定2个周期。
- 在有分支预测的情况下，一个分支跳转指令预测失败的代价为2个周期，成功的收益为2个周期。

2. 统计未使用分支预测和使用分支预测的总周期数及差值

	btb.s	bht.s	QuickSort.s	MatMul.s
无分支预测	512	538	45337	173605
BTB分支预测	316	382	45798	171537
BTB分支预测差值	196	156	-461	2068
BHT分支预测	318	370	44634	170006
BHT分支预测差值	194	168	703	3599

3. 统计分支指令数目、动态分支预测正确次数和错误次数

	btb.s	bht.s	QuickSort.s	MatMul.s
分支指令数目	101	110	10198	4896
BTB预测正确次数	99	88	8176	4077
BTB预测错误次数	2	22	2022	819
BHT预测正确次数	98	95	8749	4329
BHT预测错误次数	3	15	1449	567

4. 对比与分析

- 使用动态分支预测通常会带来正收益，但也有反例。例如在QuickSort测试中，简单BTB预测反而会增大运行周期数
- BHT分支预测效果通常好于BTB，这符合“机制过于简单效果不会太好”的直觉
- 不同测试样例上不同分支结构带来的运行周期数优化效果差别显著，表明程序性能优化与程序本身有较强的相关性，而没有银弹。

实验总结

- 本次实验实现了BTB和BHT动态分支预测中的BTB和BHT，加深了对动态分支预测的原理理解。
- 在本次实验结果分析中，通过各种程序测试和指标分析，体会到动态分支预测对程序性能的实际优化效果。
- 本次试验用时6h：

- 阶段一：3.5h
- 阶段二：1h
- 阶段三与实验报告：1.5h