

## Lab4 实验报告

PB18051098 徐碧涵

# 实验目的

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

# 实验环境

Vivado2018.3

# 实验过程

## 实现 BTB

BTB 实现一个 buffer, 保存当前地址高位、目标地址和有效位, 使用 1bit 预测位进行分支预测, 预测位为 1 表示预测跳转, 为 0 表示预测不跳转。在分支指令的 EX 段根据其真实跳转情况对预测位进行修改, 如果真实指令跳转那么该跳转指令预测位设为 1, 反之设为 0。在 IF 段对当前 PC 值在 Buffer 中进行查找如果命中且预测位为 1, 则下一条指令的地址不是 pc+4, 而是 buffer 中的内容。BTB 的命中: 当前指令的低位用于寻址, 对比指令的高位和 buffer 中是否相等并且有效位为 1, 表示命中。

Buffer 实现如下: 用寄存器存储指令地址高位, 目标跳转地址, 有效位以及预测位

```
reg [TAG_ADDR_LEN-1 : 0] tag[0 : SET_SIZE-1];
```

```
reg [31:0] target[0 : SET_SIZE-1];
```

```
reg valid[0 : SET_SIZE-1];
```

```
reg state[0 : SET_SIZE-1];
```

在分支指令的 EX 段根据是否跳转更新 buffer 中的数据:

```
if (update)
    begin
        if (br)
            begin
                tag[PCX_set_addr] <= PCX_tag;
                target[PCX_set_addr] <= br_target_PCX;
                valid[PCX_set_addr] <= 1'b1;
                state[PCX_set_addr] <= 1'b1;    //跳转预测位置 1
            end
        else
```

```

begin
    tag[PCX_set_addr] <= PCX_tag;
    target[PCX_set_addr] <= br_target_PCX;
    valid[PCX_set_addr] <= 1'b1;
    state[PCX_set_addr] <= 1'b0;        //未跳转预测位置 0
end
end

```

## 在 BTB 的基础上实现 BHT

使用 2bit 进行分支预测，使用 buffer 记录分支指令的预测位

reg [1 : 0] state[0 : SET\_SIZE-1];

根据真实跳转对状态值进行修改

if (update)

begin

if (br == 1)

begin

state[PCX\_set\_addr] <= (state[PCX\_set\_addr] == 2'b11) ? 2'b11 :  
state[PCX\_set\_addr] + 1;

end

else

begin

state[PCX\_set\_addr] <= (state[PCX\_set\_addr] == 2'b00) ? 2'b00 :  
state[PCX\_set\_addr] - 1;

end

end

对于给定 PC 值进行预测，如果该 PC 值在 BHT 中预测跳转，那么即预测该 branch 指令 taken，如果此时在 BTB 中命中，那么下一条指令的地址即为 BTB buffer 中的值，如果 BTB 没有命中，那么需要一个 cycle 计算出分支目标地址，在下一个 cycle 计算出分支目标地址后，flush 掉一条指令，并将下一条指令地址赋为计算出的分支目标地址。如果预测错误需要 flush 掉 2 条指令。

## 实验结果

分支收益：如果 BTB 命中且分支预测正确，那么能带来 2 个 cycle 的收益，如果 BTB 未命中但 BHT 分支预测正确那么带来 1 个 cycle 的收益，分支预测失败则无收益。

统计四个测试样例未使用分支预测的执行总周期数，以及使用分支预测执行周期数，分支指令数，分支错误指令数，分支正确指令数。

使用 BTB 进行分支预测

	未使用分支 预测时钟周 期数	使用分支预 测时钟周期 数	分支指令条 数	分支错误指 令数	分支正确指 令数

Btb	506	310	101	3	98
Bht	554	378	110	22	88
Quicksort	53530	45689	6116	2196	3920
Matmul	341323	333171	4624	548	4076

#### 使用 BHT 进行分支预测

	未使用分支 预测时钟周 期数	使用分支预 测时钟周期 数	分支指令条 数	分支错误指 令数	分支正确指 令数
Btb	506	312	101	3	98
Bht	554	364	110	15	95
Quicksort	53530	44069	6116	1174	4942
Matmul	341323	332639	4624	282	4342

## 总结

从上述实验结果可见使用分支预测将带来收益减短程序执行时间，其中 BHT 比 BTB 预测准确率更高，减少的时钟周期数也更多。