

Lecture 9: Branch Prediction

Basic idea, saturating counter, BHT, BTB, return address prediction, correlating prediction

1

Reducing Branch Penalty

Branch penalty in dynamically scheduled processors: wasted cycles due to pipeline flushing on mis-predicted branches

Reduce branch penalty:

1. Predict branch/jump instructions AND branch direction (taken or hot taken)
2. Predict branch/jump target address (for taken branches)
3. Speculatively execute instructions along the predicted path

2

What to Use and What to Predict

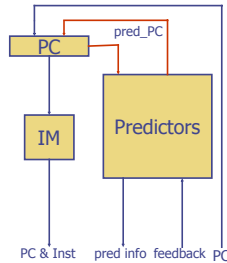
Available info:

- Current predicted PC
- Past branch history (direction and target)

What to predict:

- Conditional branch inst: branch direction and target address
- Jump inst: target address
- Procedure call/return: target address

May need instruction pre-decoded



3

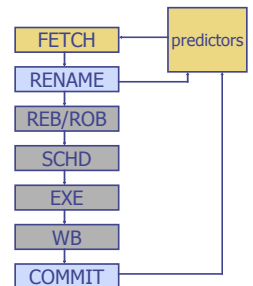
Mis-prediction Detections and Feedbacks

Detections:

- At the end of decoding
 - Target address known at decoding, and not match
 - Flush fetch stage
- At commit (most cases)
 - Wrong branch direction or target address not match
 - Flush the whole pipeline (at EXE: MIPS R10000)

Feedbacks:

- Any time a mis-prediction is detected
- At a branch's commit (at EXE: called speculative update)



4

Branch Direction Prediction

- Predict branch direction: taken or not taken (T/NT)



- Static prediction: compilers decide the direction
- Dynamic prediction: hardware decides the direction using dynamic information

- 1-bit Branch-Prediction Buffer
- 2-bit Branch-Prediction Buffer
- Correlating Branch Prediction Buffer
- Tournament Branch Predictor
- and more ...

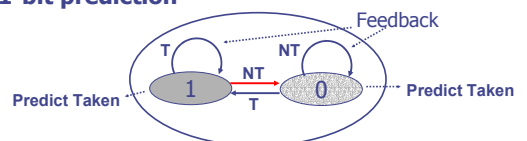
5

Predictor for a Single Branch

General Form

1. Access PC
2. Predict Output T/NT
3. Feedback T/NT

1-bit prediction

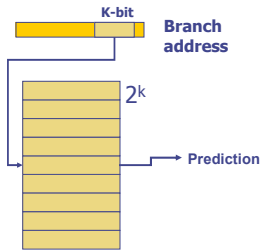


6

Branch History Table of 1-bit Predictor

BHT also Called Branch Prediction Buffer in textbook

- Can use only one 1-bit predictor, but accuracy is low
- BHT: use a table of simple predictors, indexed by bits from PC
- Similar to direct mapped cache
- More entries, more cost, but less conflicts, higher accuracy
- BHT can contain complex predictors



7

1-bit BHT Weakness

- Example: in a loop, 1-bit BHT will cause 2 mispredictions
- Consider a loop of 9 iterations before exit:

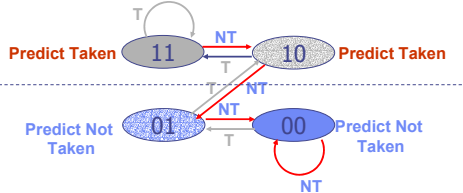

```
for (...) {
    for (i=0; i<9; i++)
        a[i] = a[i] * 2.0;
}
```

 - End of loop case, when it exits instead of looping as before
 - First time through loop on *next* time through code, when it predicts *exit* instead of looping
 - Only 80% accuracy even if loop 90% of the time

8

2-bit Saturating Counter

- Solution: 2-bit scheme where change prediction only if get misprediction *twice*: (Figure 3.7, p. 249)

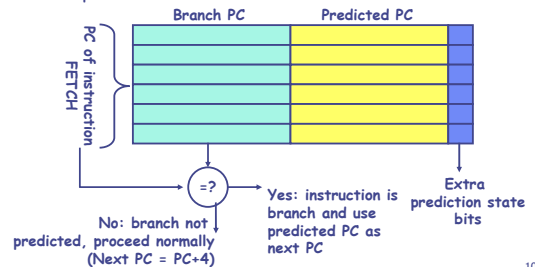


- Blue: stop, not taken
- Green: go, taken
- Adds *hysteresis* to decision making process

9

Branch Target Buffer

- Branch Target Buffer (BTB): Address of branch index to get prediction AND branch address (if taken)
 - Note: must check for branch match now, since can't use wrong branch address
- Example: BTB combined with BHT



10

Return Addresses Prediction

- Register indirect branch hard to predict address
 - Many callers, one callee
 - Jump to multiple return addresses from a single address (no PC-target correlation)
- SPEC89 85% such branches for procedure return
- Since stack discipline for procedures, save return address in small buffer that acts like a stack: 8 to 16 entries has small miss rate

11

Correlating Branches

Code example showing the potential Assemble code

```
If (d==0)                                BNEZ R1, L1
    d=1;                                  DADDIU R1,R0,#1
If (d==1)                                L1: DADDIU R3,R1,#-1
    ...                                    BNEZ R3, L2
                                           L2:
    ...
```

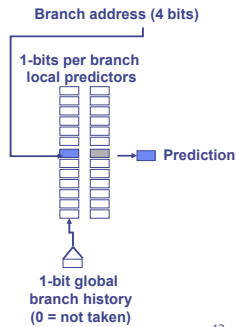
Observation: if BNEZ1 is not taken, then BNEZ2 is taken

12

Correlating Branch Predictor

Idea: taken/not taken of recently executed branches is related to behavior of next branch (as well as the history of that branch behavior)

- Then behavior of recent branches selects between, say, 2 predictions of next branch, updating just that prediction
- (1,1) predictor: 1-bit global, 1-bit local

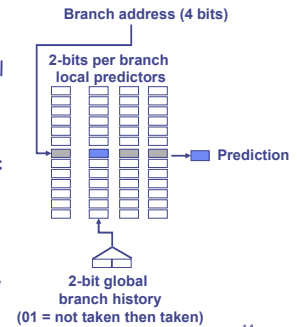


13

Correlating Branch Predictor

General form: (m, n) predictor

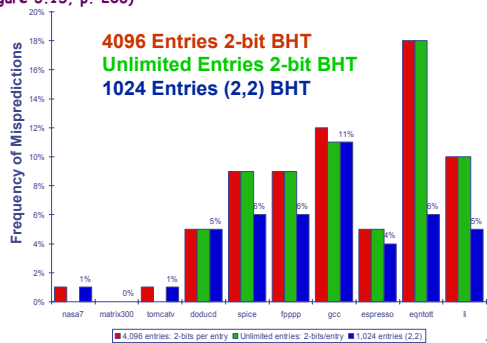
- m bits for global history, n bits for local history
- Records correlation between m+1 branches
- Simple implementation: global history can be store in a shift register
- Example: (2,2) predictor, 2-bit global, 2-bit local



14

Accuracy of Different Schemes

(Figure 3.15, p. 206)



15

Estimate Branch Penalty

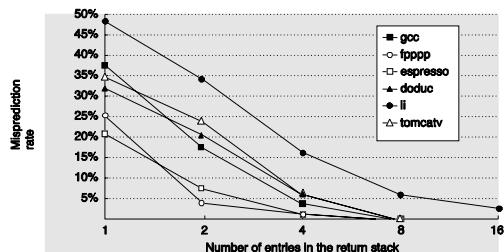
EX: BHT correct rate is 95%, BTB hit rate is 95%

Average miss penalty is 15 cycles

How much is the branch penalty?

16

Accuracy of Return Address Predictor



17