ECE 4750
Computer Architecture
Prof. José F. Martínez
Dynamic Branch Prediction

• Control hazards limit performance
  • must figure out (1) it’s a branch, (2) target address, (3) outcome
  • gets worse when (a) issue ↑, (b) pipeline depth ↑

• Static schemes
  • delayed branches, predict (not) taken
  • do not capture dynamic behavior

• Dynamic schemes (today)
  • predictive hardware
  • action dependent on run-time behavior
Dynamic Branch Prediction

• Goal: prevent control dependences from causing stalls
• Means: predict branch outcome based on run-time behavior
• Effectiveness determined by
  • prediction accuracy
  • cost when prediction is correct
  • penalty when prediction is not correct
• Factors that affect effectiveness
  • pipeline structure
  • type of predictor
  • misprediction recovery scheme
Pattern History Table (PHT)

• Small memory indexed by lower portion of branch PC (why lower?)
  • similar to instruction cache, only every access is a hit
    • small, tagless, direct mapped

• Simplest: remember last outcome – one-bit PHT
  • targets highly biased branches
  • however, mispredictions come in pairs!
    ex.: loop branch taken 9/10 times; what is the misprediction rate?
Pattern History Table (PHT)

• Small memory indexed by lower portion of branch PC (why lower?)
  • similar to instruction cache, only every access is a hit
    • small, tagless, direct mapped

• Simplest: remember last outcome – one-bit PHT
  • targets highly biased branches
  • however, mispredictions come in pairs!
    ex.: loop branch taken 9/10 times; what is the misprediction rate?

<table>
<thead>
<tr>
<th>Branch Instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
</tr>
<tr>
<td>Prediction</td>
</tr>
<tr>
<td>Outcome</td>
</tr>
</tbody>
</table>
N-bit PHT

• Obvious solution: two-bit prediction (*saturating counter*)
  • taken increments (up to 11), not taken decrements (down to 00)
  • 0X predict not taken, 1X predict taken

• Generalization: n-bit prediction
  • 0XX...X predict not taken, 1XX...X predict taken
  • empirically, n=2 often good enough

• Complication vs. one-bit: requires more updates
Bimodal
Local

for (i=1; i<4; i++)
...

Yeh and Patt, ISCA ’92

Scott McFarling
SPEC ’89 average

![Graph showing conditional branch prediction accuracy vs predictor size in bytes. The graph compares local and bimodal prediction methods.](image-url)
Global

\[
\begin{align*}
\text{if } (x < 5) \\
& \quad \text{...}
\end{align*}
\]

\[
\begin{align*}
\text{if } (x < 8) \\
& \quad \text{...}
\end{align*}
\]

Yeh and Patt, ISCA ’92

Scott McFarling
Combined (aka Tournament or Selection)

McFarling, TR DEC-WRL TN-36

Counts

<table>
<thead>
<tr>
<th>P1c</th>
<th>P2c</th>
<th>P1c-P2c</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>(no change)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>(decrement counter)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>(increment counter)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>(no change)</td>
</tr>
</tbody>
</table>

P1c-P2c → useP1 → P1 → P2

PC
Branch Target Buffer (BTB)

- PHT guesses direction; BTB tries to obtain target address
- If target determined by the end of IF, no penalty
  - PHT must be used at ID → one-cycle penalty
  - BTB can be used at IF → zero-cycle penalty
- Small cache indexed, tagged by PC
  - unlike PHT, should miss if not the right branch – or a branch at all!
- Allocate when branch and taken
- Problem: hit implies taken
Return Address Stack (RAS)

• Predict return address of a procedure
• Cannot use BTB – caller address may change every time
• Solution: use specialized stack
  • push PC+4 when jal
  • pop address, set PC upon returning
• Nested/recursive calls: remember last n return addresses