Chapter 3 – Instruction-Level Parallelism and its Exploitation (Part 2)

ILP vs. Parallel Computers
Dynamic Scheduling (Section 3.4, 3.5)
Dynamic Branch Prediction (Section 3.3, 3.9, and Appendix C)
Hardware Speculation and Precise Interrupts (Section 3.6)
Multiple Issue (Section 3.7)
Static Techniques (Section 3.2, Appendix H)
Limitations of ILP
Multithreading (Section 3.11)
Putting it Together (Mini-projects)

Dynamic Branch Prediction
Reducing penalties from control dependences
Basic idea
  Hardware guesses
    * Whether branch will be taken/not taken
    * Where the branch will go
Especially important for multiple issue processors
Desirable properties
  Good prediction rate
  Make correct prediction fast
  Don’t slow too much on misprediction

Branch Prediction Buffer (Appendix C)
Maintain a buffer with prediction bits
Index buffer with LSBs of branch instruction PC

<table>
<thead>
<tr>
<th>Prediction bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

Predict based on indexed bit, change bit on misprediction
Accessed in ID stage (not useful for simple 5-stage pipeline)
Limitation of 1-bit predictor?

Variations on Branch Prediction Buffer
Variations
  n-bit predictor
  Correlating predictors
  Tournament predictors
**N-bit Predictor**

Contains n-bit saturating counter
Count up if taken, down if not taken
Predict taken if ≥ 2**(n-1); predict not taken if < 2**(n-1)
2-bit good for loops

**Correlating Predictors: (m,n) Predictor**

Use outcome of previous m branches and n-bit predictors
For each branch, the prediction buffer contains
An entry for each possible history of previous m branches
Each entry is an n-bit predictor

---

**Correlating Predictors (Cont.)**

(1,1) predictor
Prediction based on 1 previous branch, 1 bit predictor

Number of prediction entries per branch = ??
Number of bits per prediction entry = ??

**Correlating Predictors Example**

Loop:

```
if a == 1  /* b1 */
a = 0
if a == 0  /* b2 */
```

... Let a = 1, 3, 1, 3, 1, 3, ...
Notation: N=not taken; T=taken
Initialize (1,1) prediction buffer entries of b2 to NT
(1st entry for previous branch taken, 2nd for not taken)
Direction of b1:
Direction of b2:
History at b2:
Prediction entries of b2:
Prediction for b2:
Tournament Predictor

Combine multiple predictors with a selector
Often combine a global predictor and a local predictor
Selector typically two bit saturating counter
Increment when predicted predictor correct, other incorrect

Tournament Predictor Example - Alpha 21264

Uses 4K 2-bit counters to choose from global and local predictor
Global predictor
4K entries of 2-bit predictors
Indexed by history of last 12 branches
Local predictor is a two-level predictor
History table with 1K 10-bit entries (for that branch)
Each entry gives 10 most recent branch outcomes
Indexes table of 1K entries with 3-bit counters
Total of 29K bits
Misprediction rate
SPECfp95 – 1 per 1000
SPECint95 – 11.5 per 1000

More Predictors

Lots of work on branch prediction
International Branch Prediction Competition!