Lecture 10: Branch Prediction and Instruction Delivery

Branch target buffer, return address prediction, tournament predictor, high-performance instruction delivery

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 stored in a shift register
- Example: \((2,2)\) predictor, 2-bit global, 2-bit local

Accuracy of Different Schemes

(Figure 3.15, p. 206)

- 4096 Entries 2-bit BHT
- Unlimited Entries 2-bit BHT
- 1024 Entries \((2,2)\) BHT

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

Estimate Branch Penalty

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

Average miss penalty is 6 cycles

How much is the branch penalty?

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
**Accuracy of Return Address Predictor**

![Accuracy of Return Address Predictor Graph]

**Tournament Branch Predictor**

- Used in Alpha 21264: Track both "local" and global history
- Intended for mixed types of applications
- Global history: T/NT history of past k branches, e.g. 0 1 0 1 0 1 (NT T NT T NT T)

![Tournament Branch Predictor Diagram]

**Branch Prediction With n-way Issue**

1. Branches will arrive up to n times faster in an n-issue processor
2. Amdahl's Law => relative impact of the control stalls will be larger with the lower potential CPI in an n-issue processor

**Integrated Instruction Fetch Units**

1. Integrated branch prediction: branch predictor becomes part of the instruction fetch unit
2. Instruction prefetch: fetch ahead to deliver multiple instructions per cycle
3. Instruction memory access and buffering: may access multiple cache lines in one cycle, use prefetch to hide the cost
   - Another approach: trace cache

**Instruction Fetch Unit**

- Fetch predictor: Predicts next fetch addresses to avoid fetch delay; may pre-predict branch direction; may be integrated with I-cache
- Branch predictor: overrides and trains fetch predictor
Pitfall: Sometimes bigger and dumber is better

- 21264 uses tournament predictor (29 Kbits)
- Earlier 21164 uses a simple 2-bit predictor with 2K entries (or a total of 4 Kbits)
- SPEC95 benchmarks, 21264 outperforms, 21164 avg. 11.5 mispredictions per 1000 instructions
- 21164 avg. 16.5 mispredictions per 1000 instructions

- Reversed for transaction processing (TP)
  - 21264 avg. 17 mispredictions per 1000 instructions
  - 21164 avg. 15 mispredictions per 1000 instructions

- TP code much larger & 21164 hold 2X branch predictions based on local behavior (2K vs. 1K local predictor in the 21264)

Dynamic Branch Prediction Summary

- Prediction becoming important part of scalar execution
- Branch History Table: 2 bits for loop accuracy
- Correlation: Recently executed branches correlated with next branch.
  - Either different branches
  - Or different executions of same branches
- Tournament Predictor: more resources to competitive solutions and pick between them
- Branch Target Buffer: include branch address & prediction
- Return address stack for prediction of indirect jump