Superscalar Design

Instruction Flow

Virendra Singh

Associate Professor

Computer **A**rchitecture and **D**ependable **S**ystems **L**ab

Department of Electrical Engineering

Indian Institute of Technology Bombay

http://www.ee.iitb.ac.in/~viren/

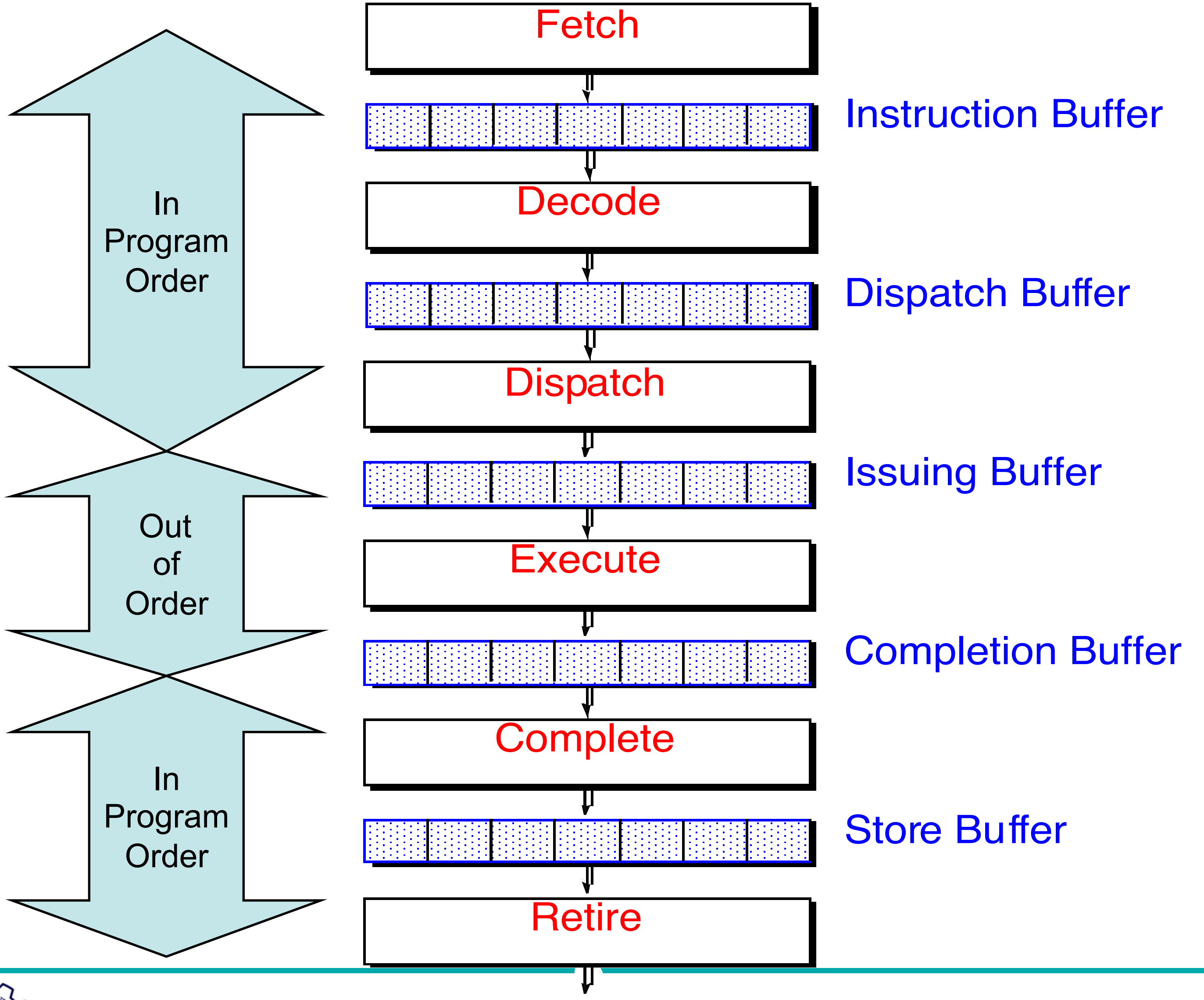
E-mail: viren@ee.iitb.ac.in

EE-739: Processor Design

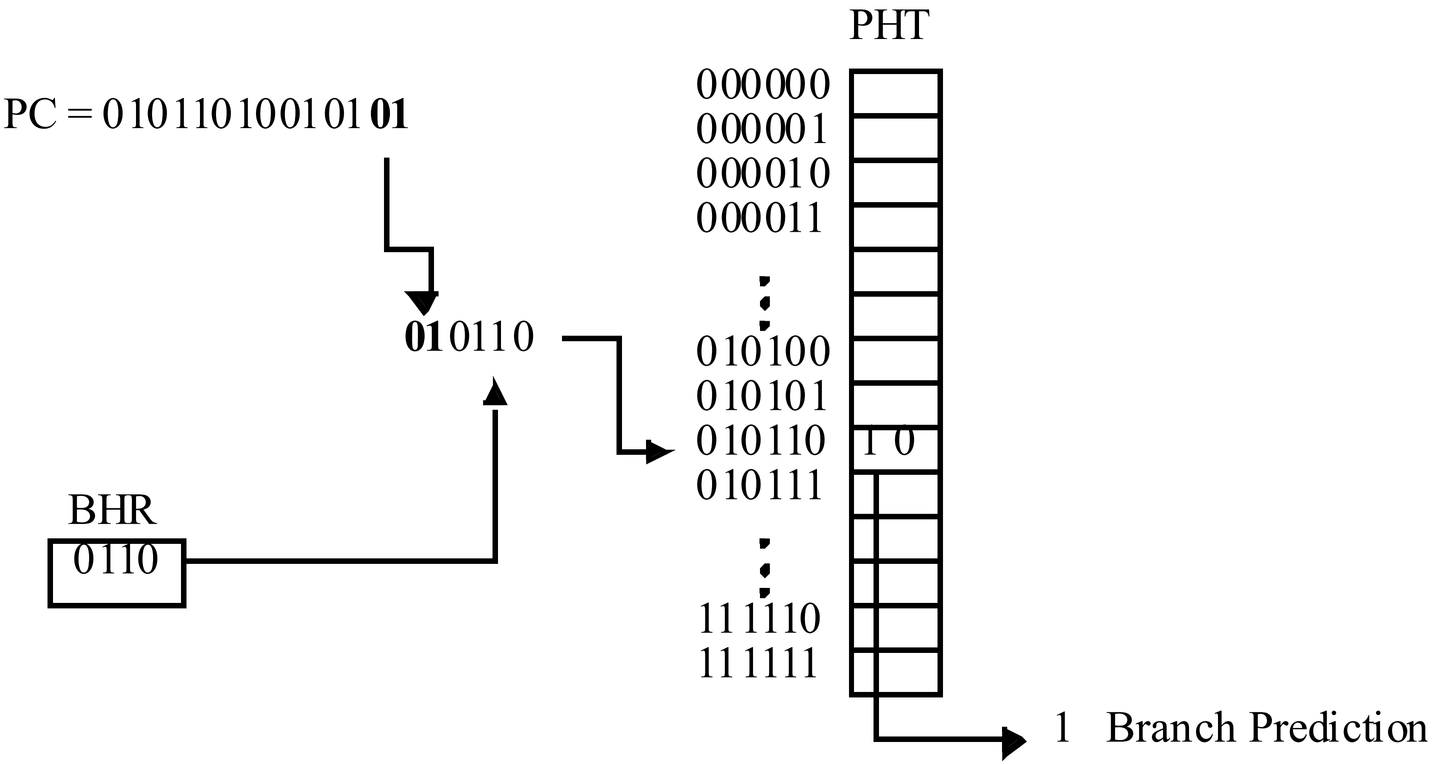
Lecture 5 (28 Jan 2015)

CADSL

Superscalar Pipeline Stages



Two-level Branch Prediction

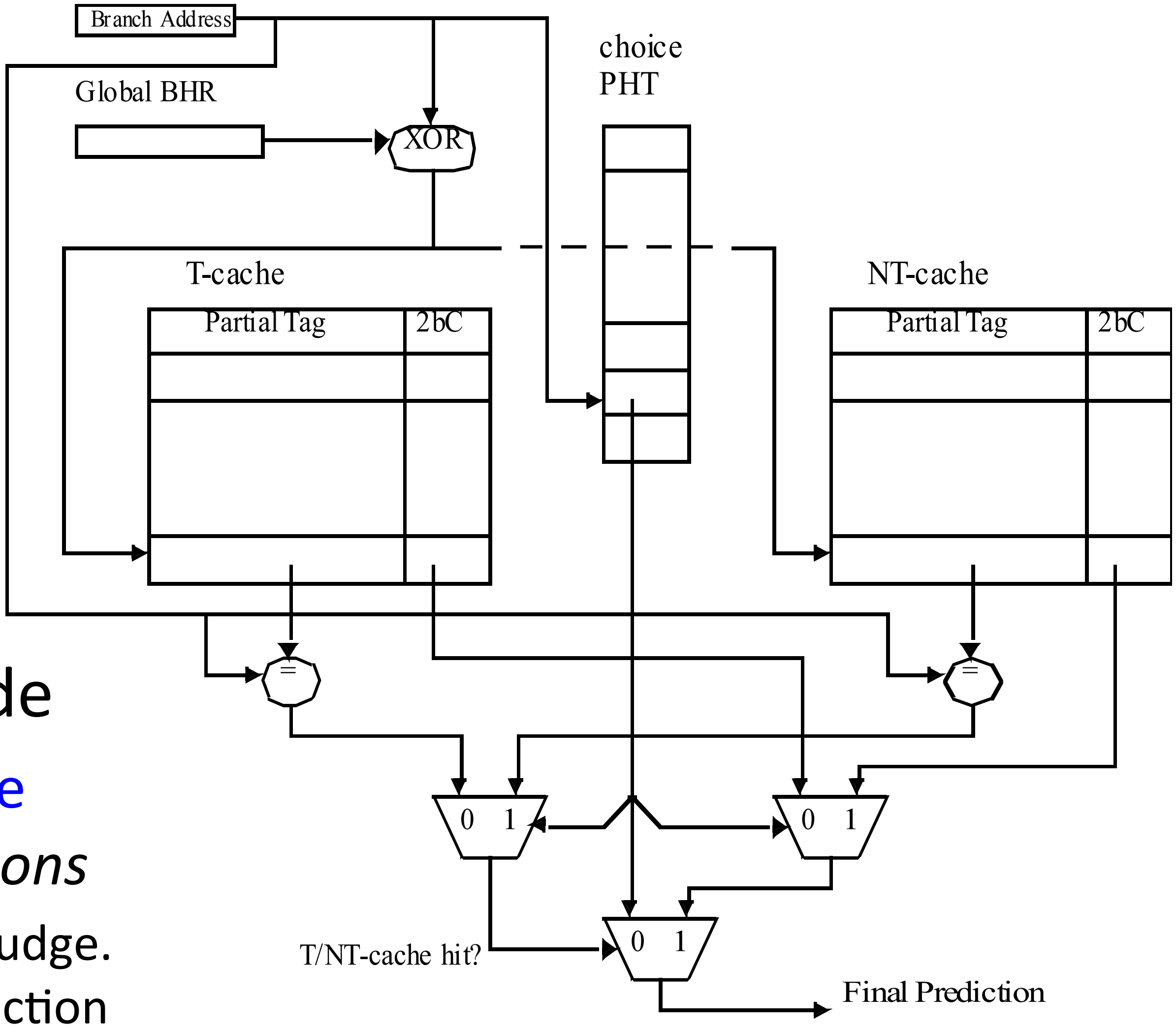


- BHR adds *global* branch history
  - Provides more context
  - Can differentiate multiple instances of the same static branch
  - Can correlate behavior across multiple static branches

Sources of Mispredictions

- Lack of history (training time)
- Randomized behavior
  - Usually due to randomized input data
  - Surprisingly few branches depend on input data values
- BHR capacity
  - Correlate to branch that already shifted out
  - e.g., loop count > BHR width
- PHT capacity
  - Aliasing/interference
    - Positive
    - Negative

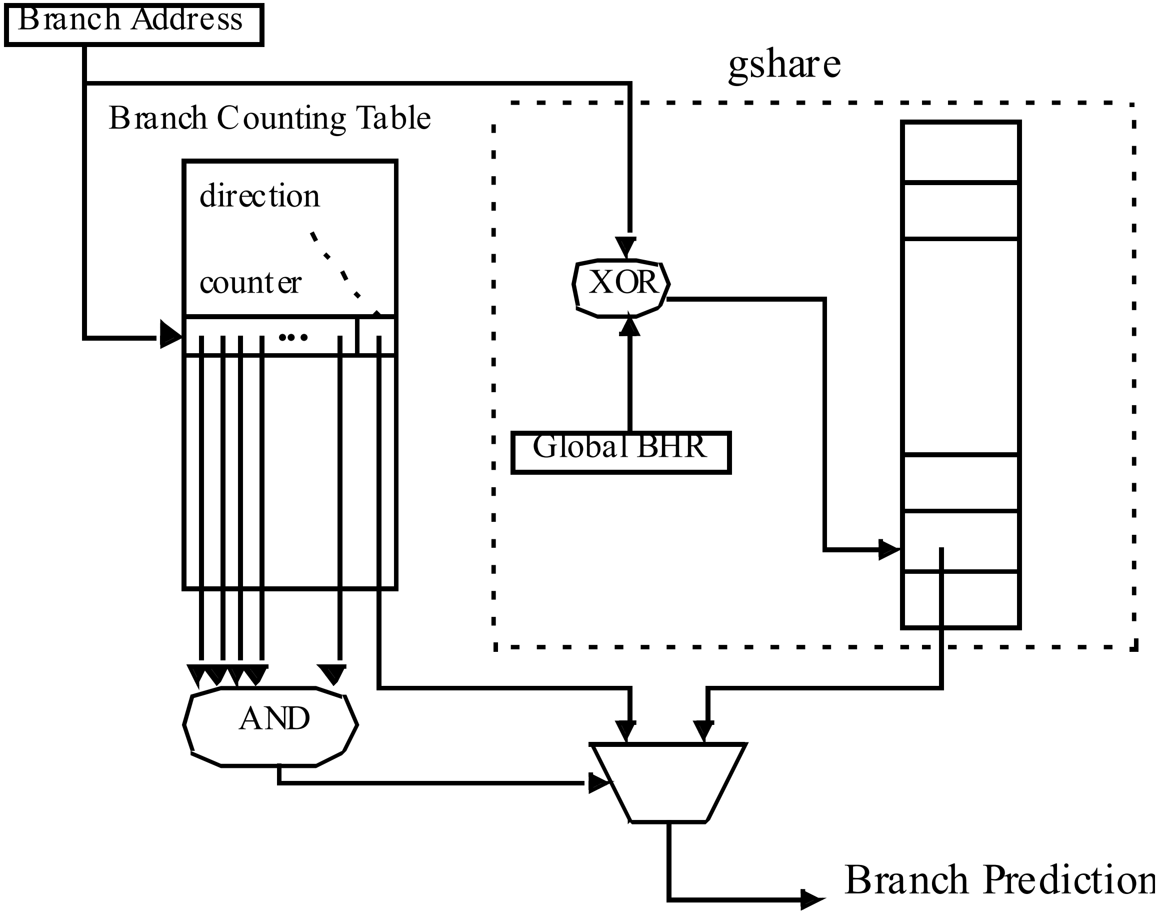
YAGS Predictor



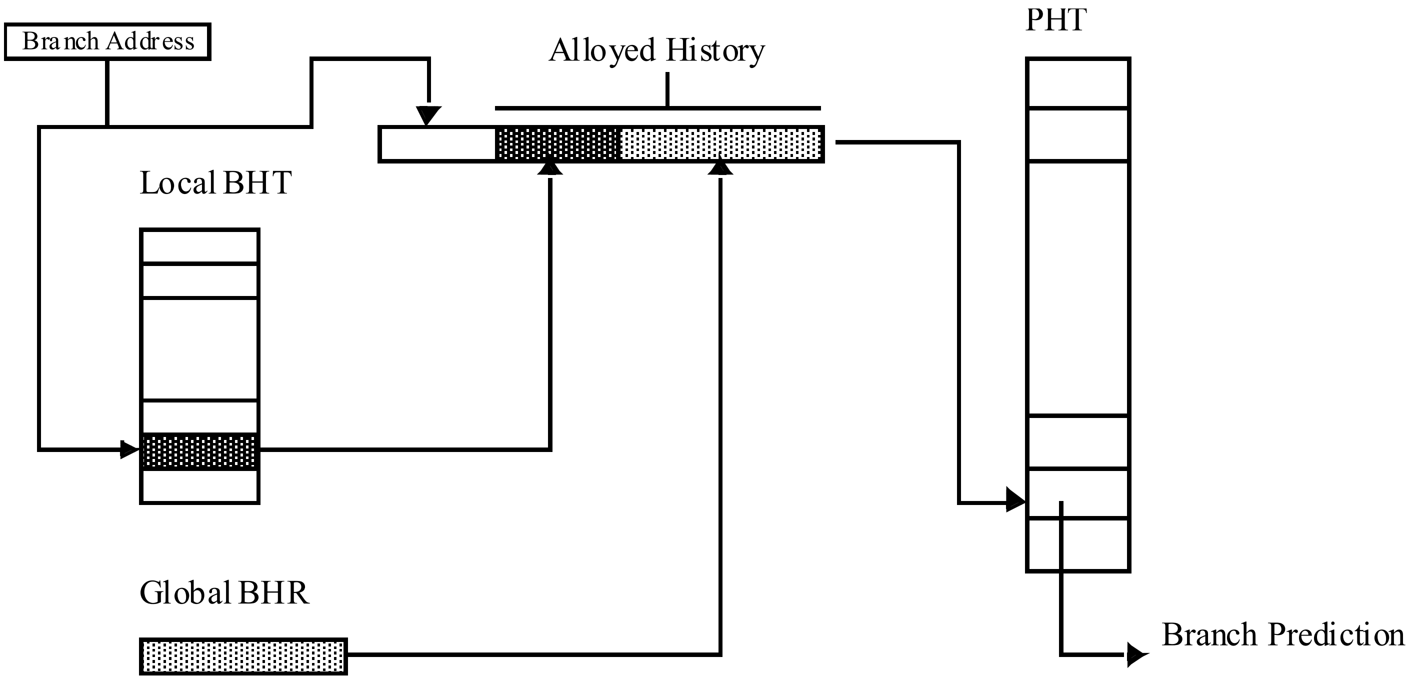
- Based on bi-mode
  - T/NT PHTs cache only the *exceptions*
- A. N. Eden and T. N. Mudge. The YAGS Branch Prediction Scheme. MICRO, Dec 1998.

Branch Filtering

- Highly-biased branches
  - e.g. '11111' history
  - Eliminated from PHT
- P-Y Chang, M. Evers, and Y Patt. Improving Branch Prediction Accuracy by Reducing Pattern History Table Interference. PACT, October 1996.

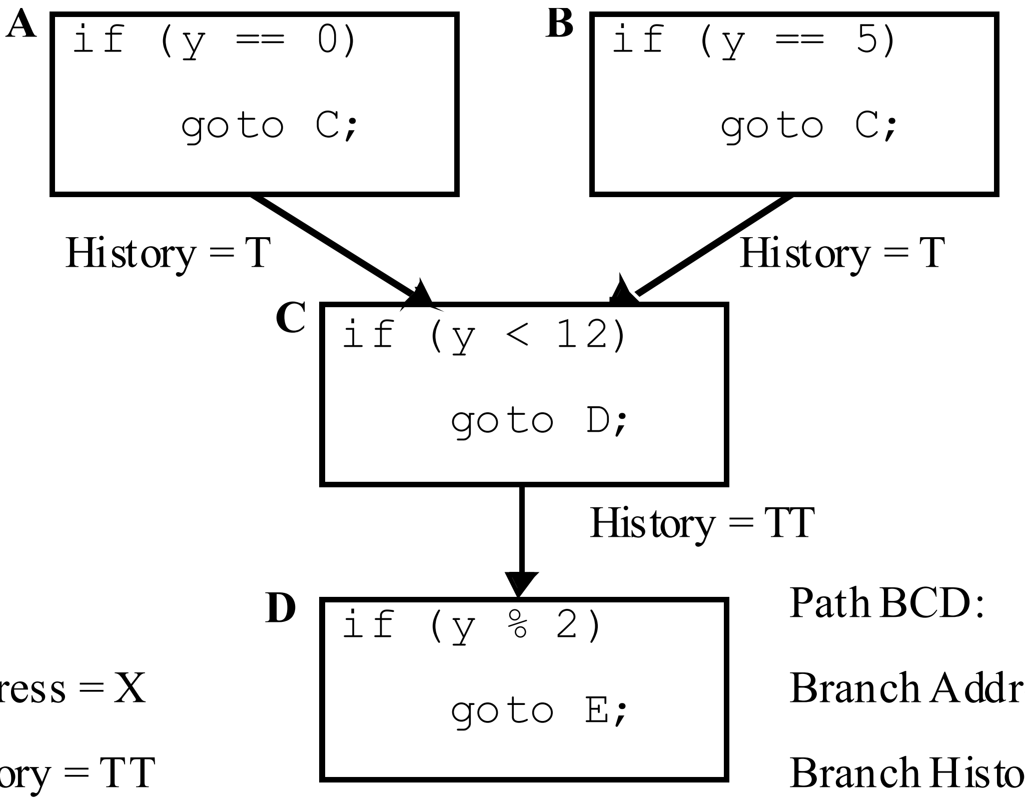


Alloyed-History Predictors



- Local history vs. global history
- Kevin Skadron, Margaret Martonosi, and Douglas W. Clark. Alloyed Global and Local Branch History: A Robust Solution to Wrong-History Mispredictions. International Journal of Parallel Programming, 31(2), April 2003.

Path History



A

```
if (y == 0)
    goto C;
```

B

```
if (y == 5)
    goto C;
```

History = T

History = T

C

```
if (y < 12)
    goto D;
```

History = TT

D

```
if (y % 2)
    goto E;
```

Path ACD:

Branch Address = X

Branch History = TT

Branch Outcome = Not Taken

Path BCD:

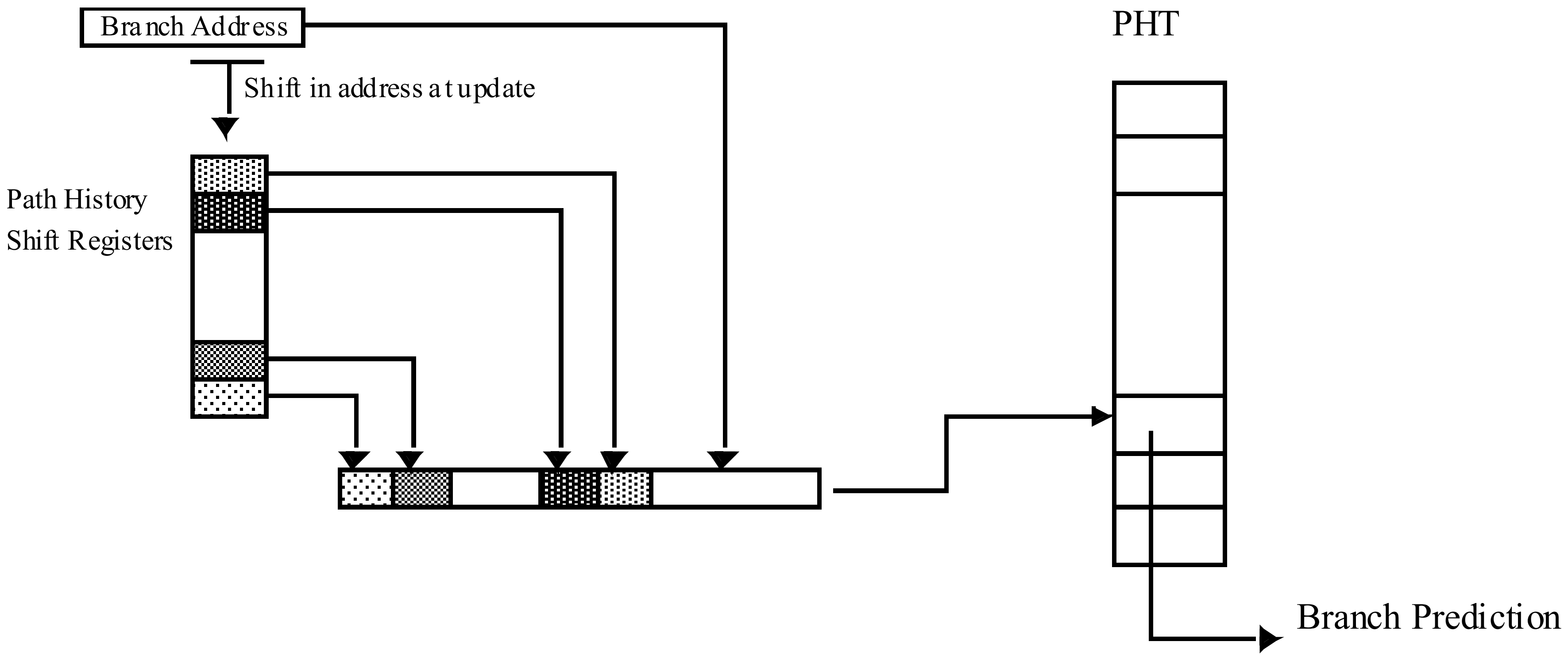
Branch Address = X

Branch History = TT

Branch Outcome = Taken

- Sometimes T/NT history is not enough
- Path history (PC values) can help

Path-Based Branch Predictor



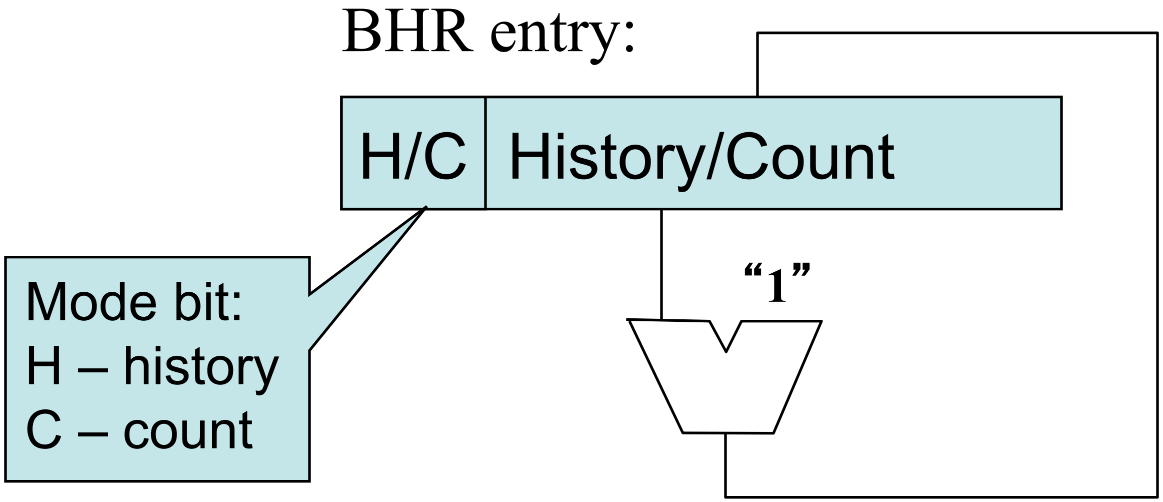
- Ravi Nair. Dynamic Path-Based Branch Correlation. International Symposium on Microarchitecture, pp 15-23, December 1995.

Dynamic History Length

- Branch history length:
  - Some prefer short history (less training time)
  - Some require longer history (complex behavior)
- Vary history length
  - Choose through profile/compile-time hints
  - Or learn dynamically

- References
  - Maria-Dana Tarlescu, Kevin B. Theobald, and Guang R. Gao. Elastic History Buffer: A Low-Cost Method to Improve Branch Prediction Accuracy. ICCD, October 1996.
  - Toni Juan, Sanji Sanjeevan, and Juan J. Navarro. Dynamic History-Length Fitting: A Third Level of Adaptivity for Branch Prediction. ISCA, June 1998.
  - Jared Stark, Marius Evers, and Patt. Variable Path Branch Prediction. ACM SIGPLAN Notices, 33(11):170-179, 1998

Loop Count Predictors



- To predict last loop iteration's NT branch:
  - Must have length(BHR) > loop count
  - Not feasible for large loop counts
- Instead, BHR has mode bit
  - Once history == '111…11' or '000…00' switch to count mode
  - Now nth entry in PHT trains to NT and predicts nth iteration as last one
  - Now length(BHR) > $\log_2$(loop count) is sufficient
- Used in Intel Pentium M/Core Duo/ Core 2 Duo

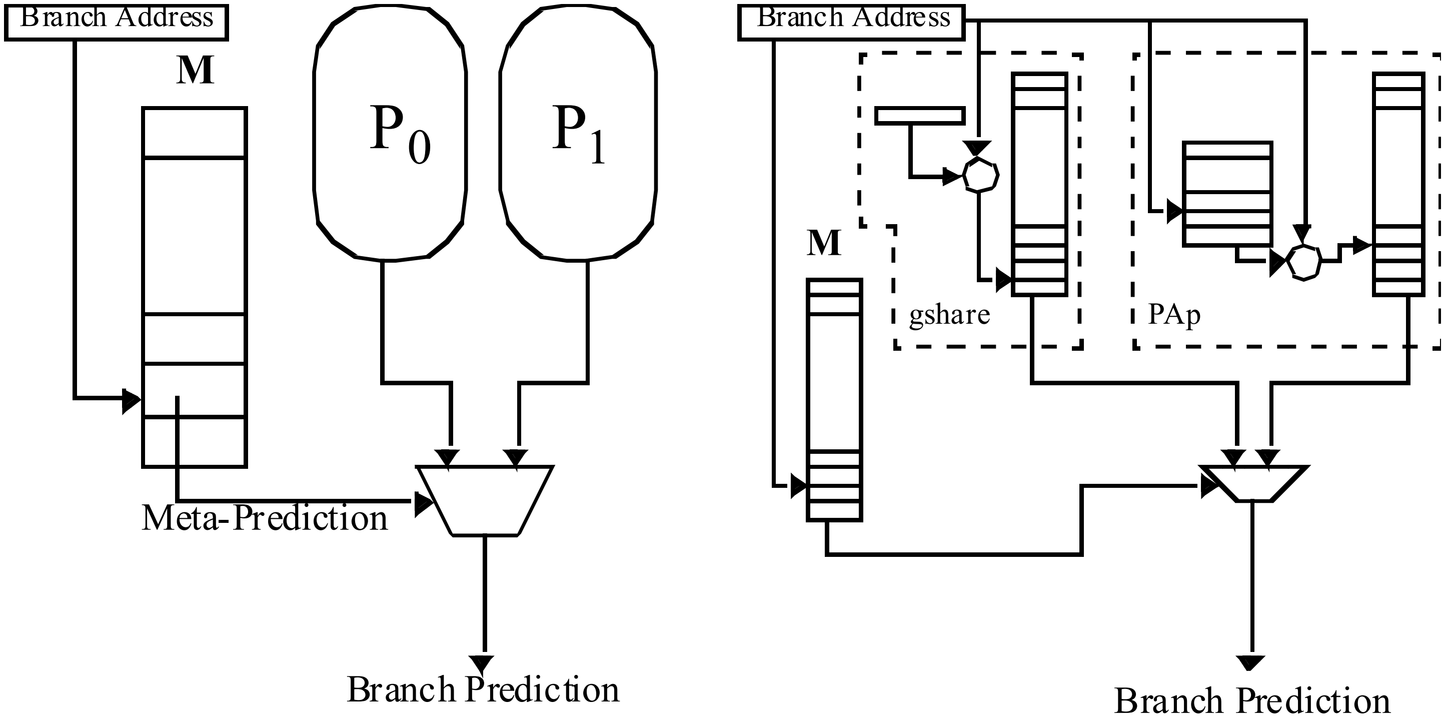
Understanding Advanced Predictors

- Four types of history
  - Local (bimodal) history (Smith predictor)
    - Table of counters summarizes local history
    - Simple, but only effective for biased branches
  - Local outcome history (correlate with self)
    - Shift register of individual branch outcomes
    - Separate counter for each outcome history
  - Global outcome history (correlate with others)
    - Shift register of recent branch outcomes
    - Separate counter for each outcome history
  - Path history (overcomes CFG convergence aliasing)
    - Shift register of recent (partial) block addresses
    - Can differentiate similar global outcome histories
- Can combine or "alloy" histories in many ways

Understanding Advanced Predictors

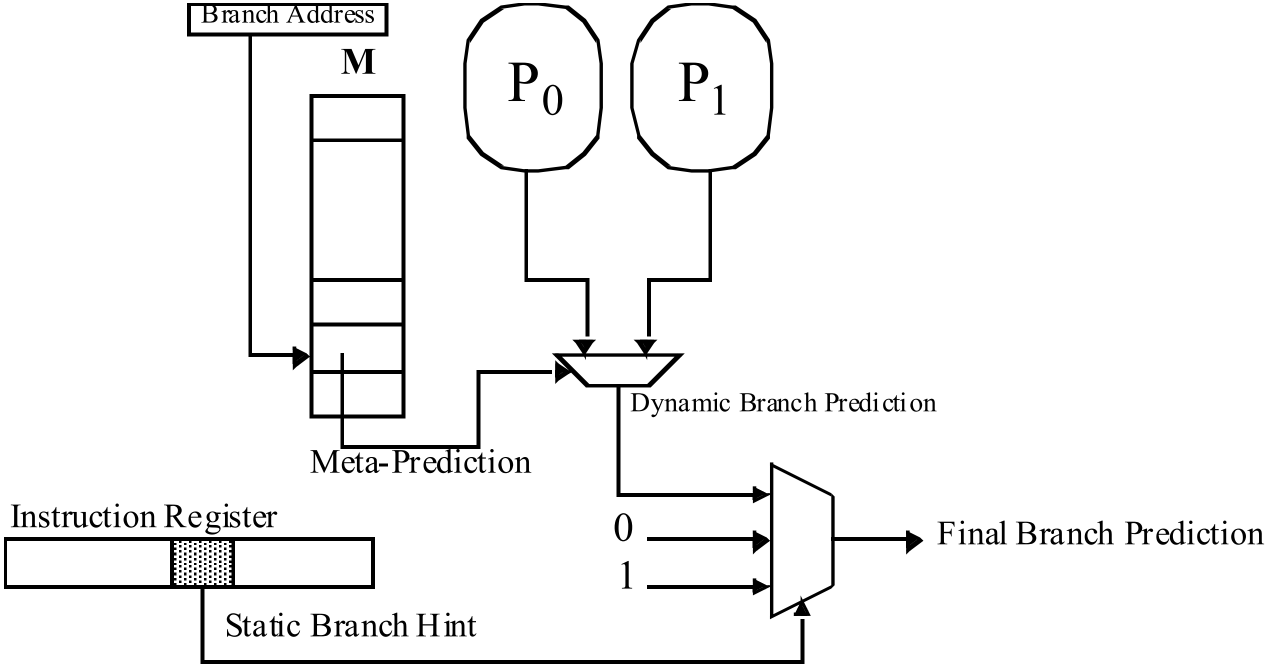
- History length
  - Short history—lower training cost
  - Long history—captures macro-level behavior
  - Variable history length predictors
- Really long history (long loops)
  - Loop count predictors
  - Fourier transform into frequency domain
- Limited capacity & interference
  - Constructive vs. destructive
  - Bi-mode, gskewed, agree, YAGS

Combining or Hybrid Predictors



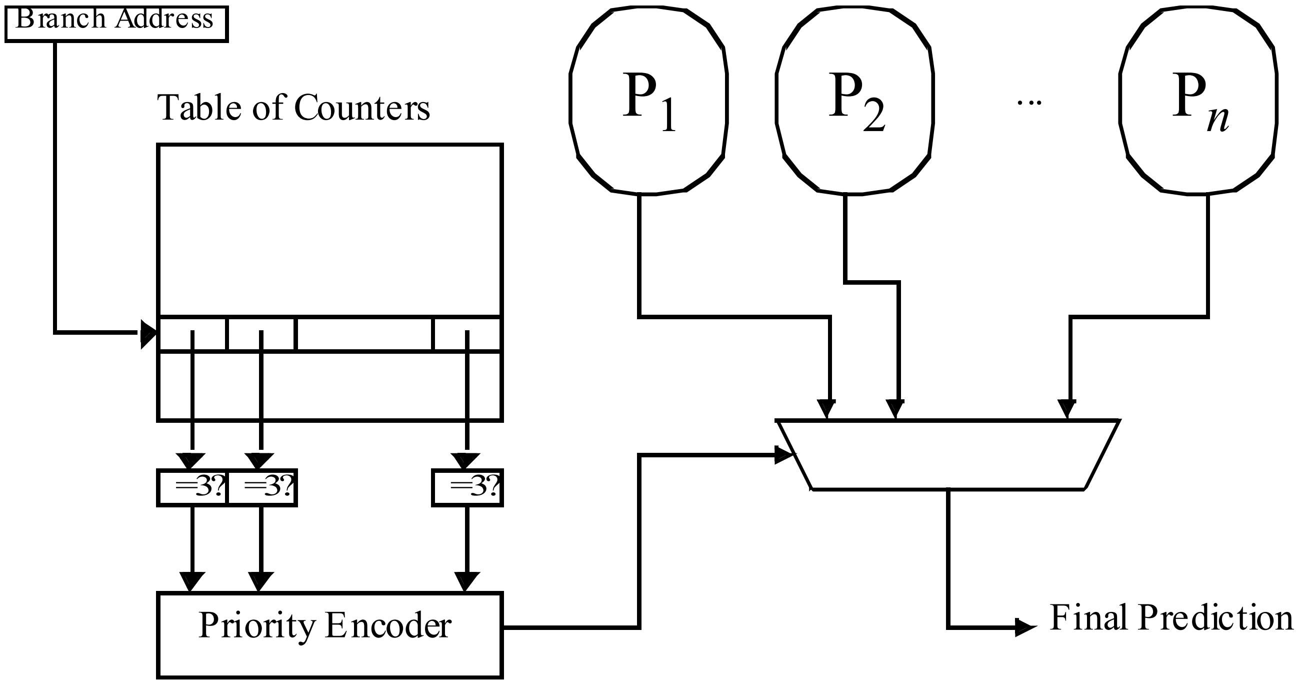
- Select best history
- Reduce interference w/partial updates
- Scott McFarling. Combining Branch Predictors. TN-36, Digital Equipment Corporation Western Research Laboratory, June 1993.

Branch Classification



- Static (profile-based) branch hints select which prediction to use
  - Static T/Static NT/Dynamic
  - PowerPC y-bit overrides static BTFN
- P-Y Chang, E Hao, TY Yeh, and Y Patt. Branch Classification: a New Mechanism for Improving Branch Predictor Performance. MICRO, Nov. 1994.
- D Grunwald, D Lindsay, and B Zorn. Static Methods in Hybrid Branch Prediction. PACT, October 1998

Multi-Hybrid Predictor

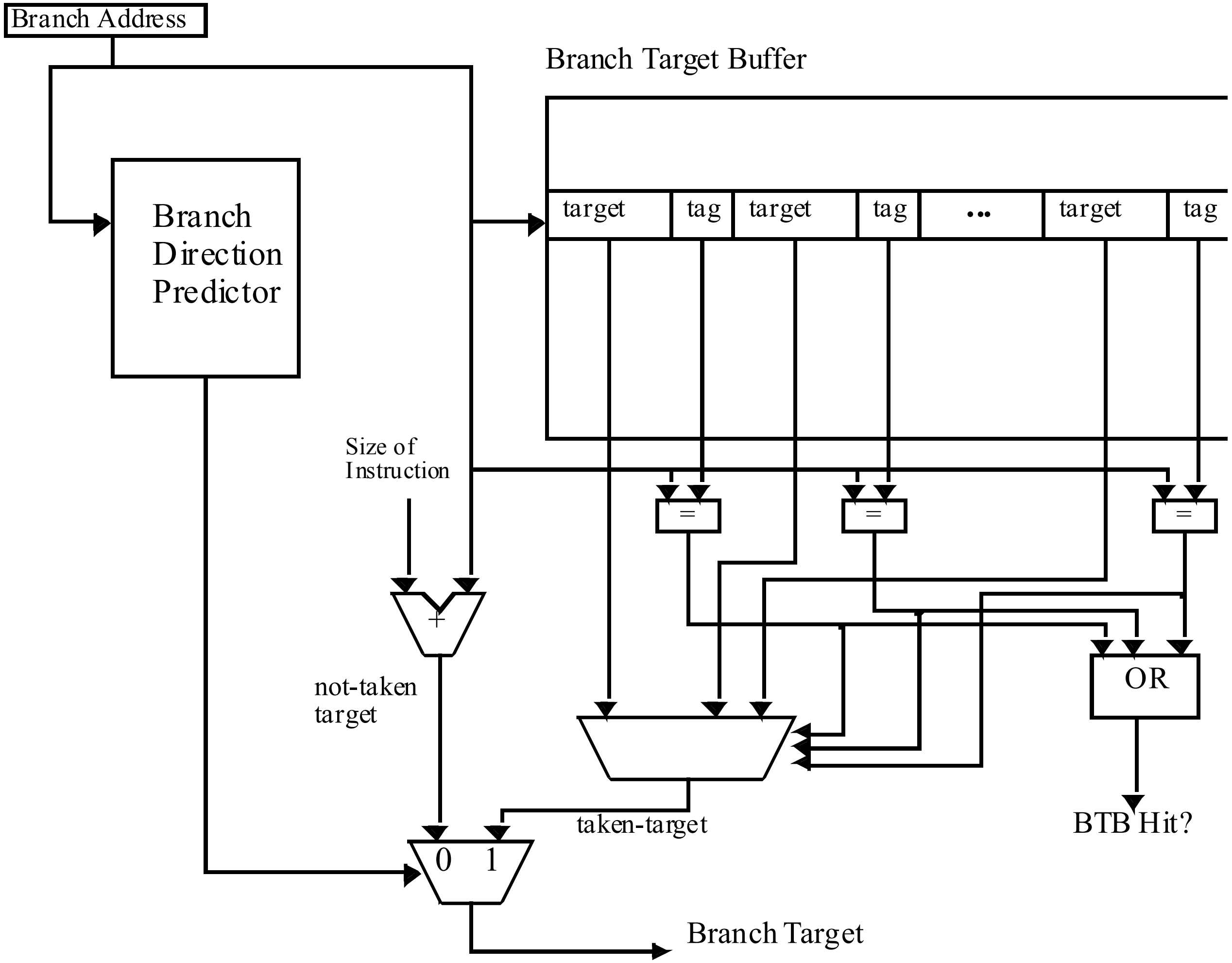


- Generalizes selector to choose from > 2 predictors
- Marius Evers, Po-Yung Chang, and Yale N. Patt. Using Hybrid Branch Predictors to Improve Branch Prediction Accuracy in the Presence of Context Switches. International Symposium on Computer Architecture, pages 3-11, May 1996.

Multiple History Lengths

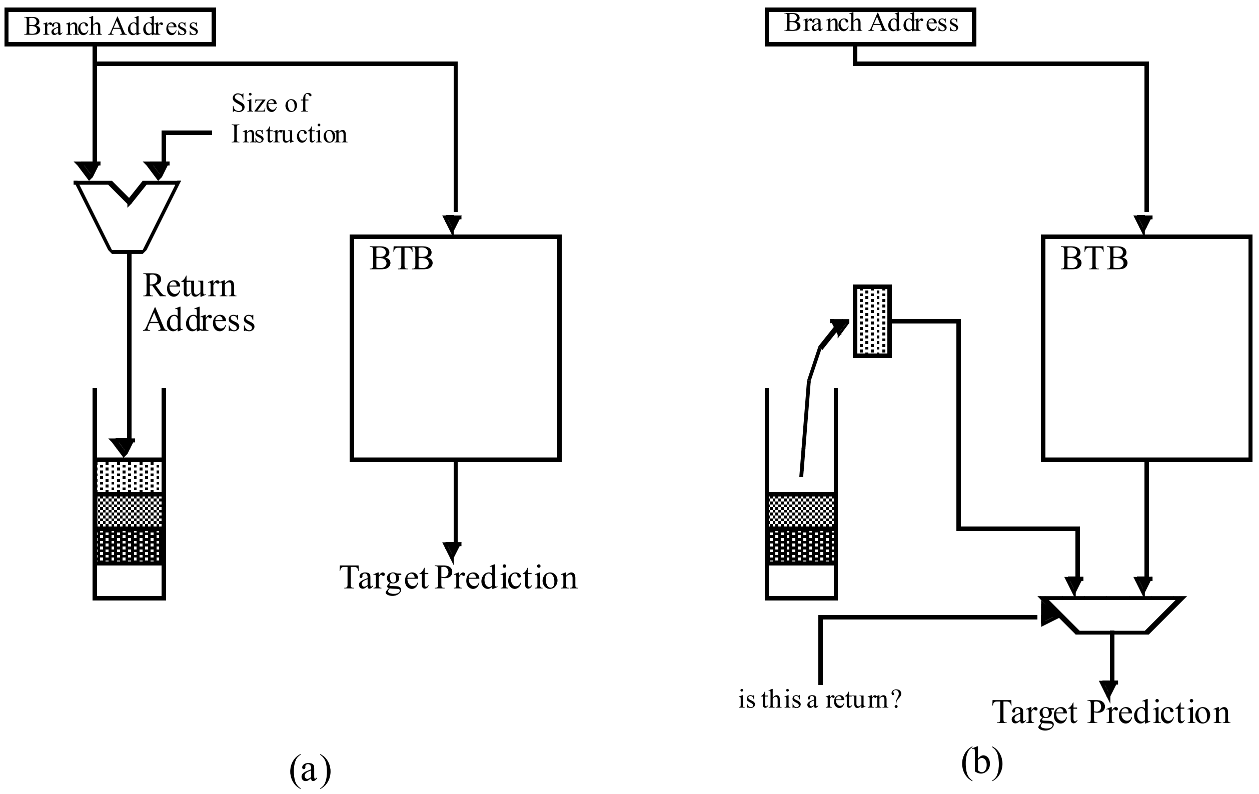
- Championship Branch Prediction (CBP)
  - 2 contests, standardized methods and traces
- Insight from perceptron BP:
  - Some branches need short history
  - Others need very long history
- Geometric history length (O-GEHL) [Seznec]
  - Geometric series of history lengths
- Tagged Geometric History Length (TAGE)
  - Choose longest matching history

Branch Target Prediction



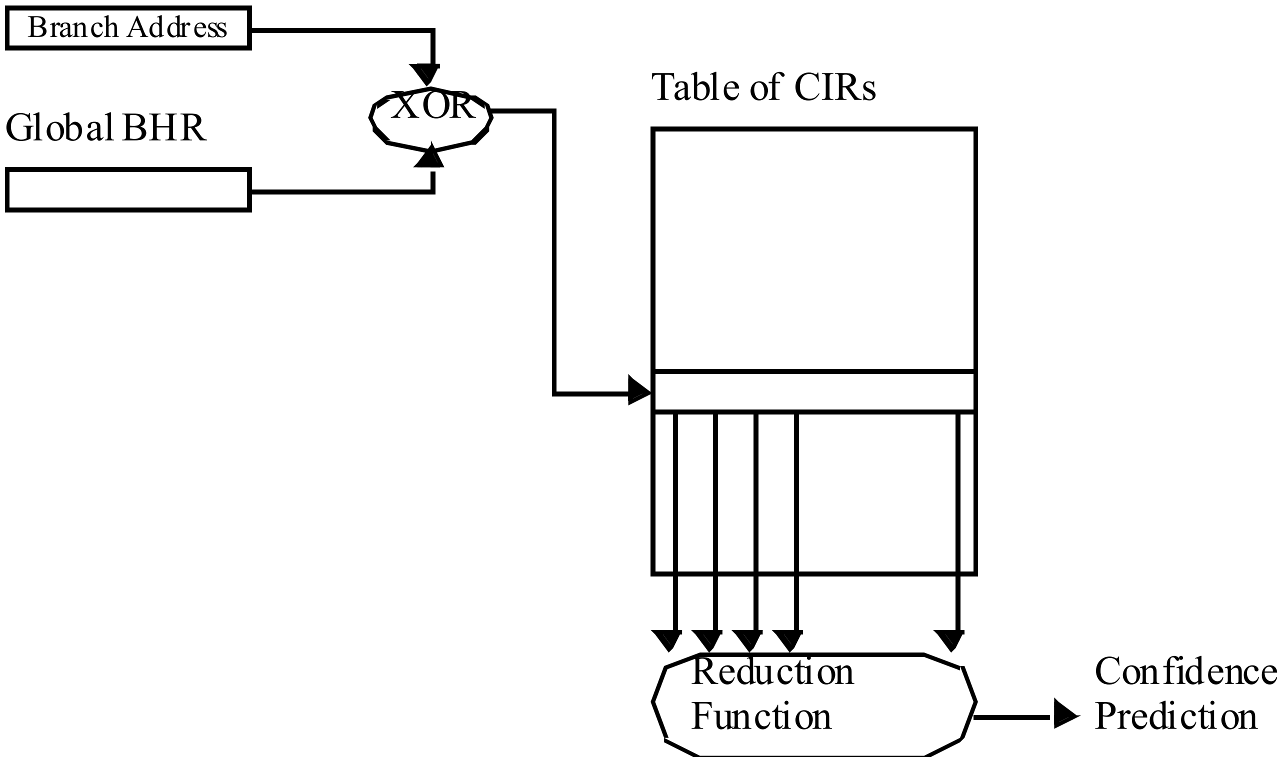
- Partial tags sufficient in BTB

Return Address Stack



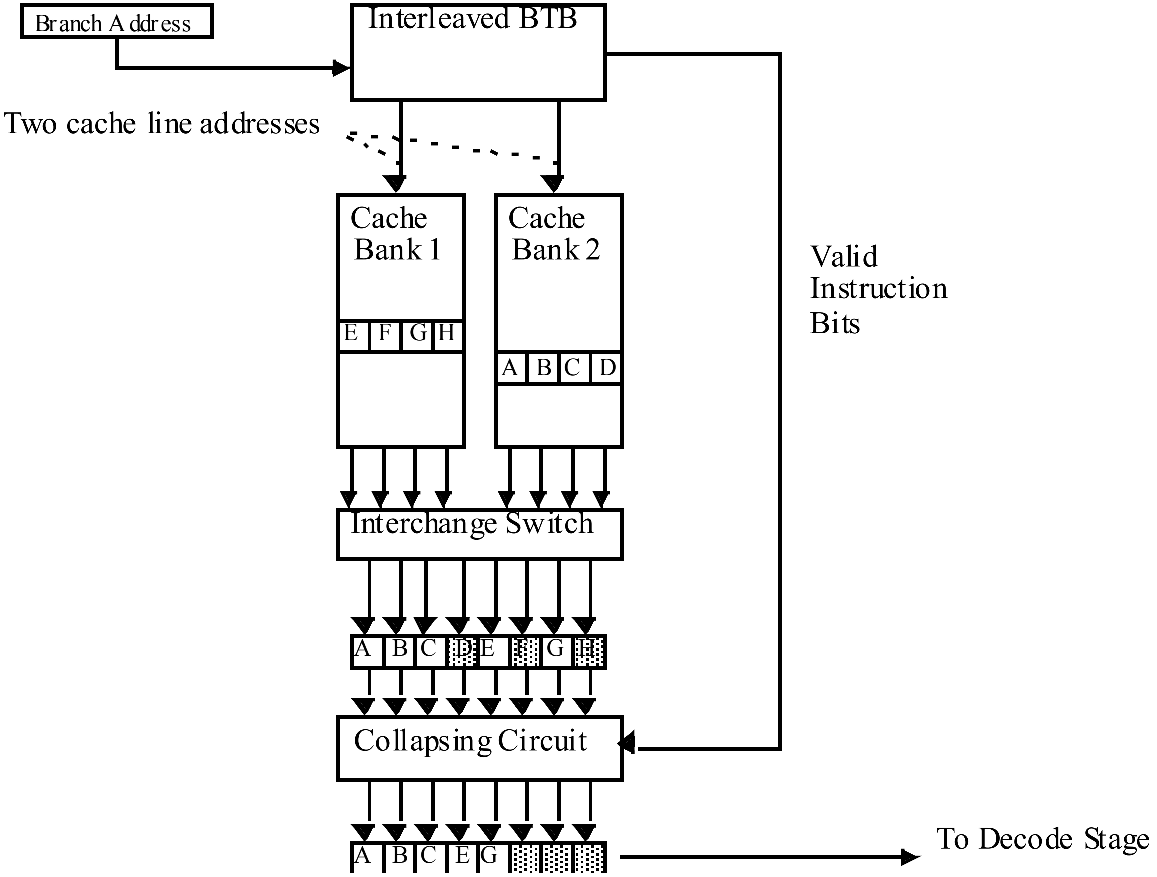
- Speculative update is painful
  - On each predicted branch, checkpoint head/tail
  - Further, checkpoint stack contents since speculative pop/push sequence is destructive
  - Conditional call/return causes more headaches

Branch Confidence Estimation



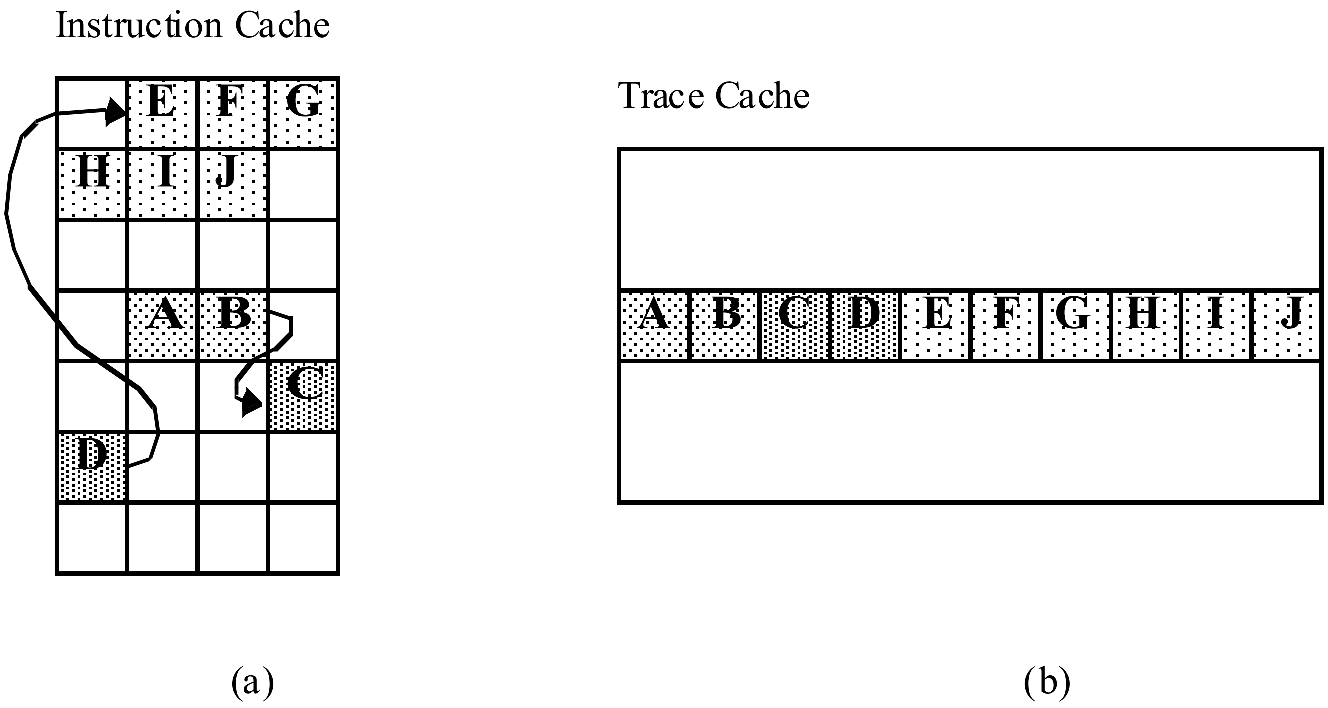
- Limit speculation (energy), reverse predictions, guide fetch for multithreaded processors
- Q Jacobson, E Rotenberg, and JE Smith. Assigning Confidence to Conditional Branch Predictions. MICRO, Dec 1996.

High-Bandwidth Fetch: Collapsing Buffer



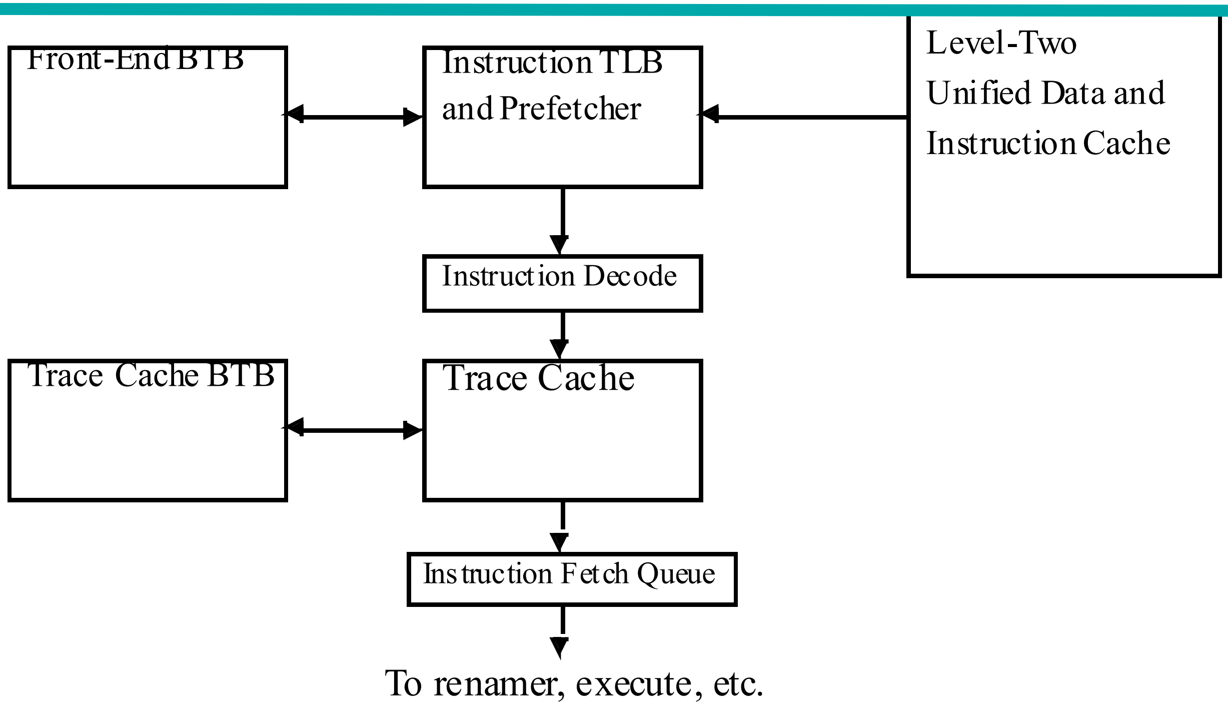
- Fetch from two cache blocks, rotate, collapse past taken branches
- Thomas M. Conte, Kishore N. Menezes, Patrick M. Mills and Burzin A. Patel. Optimization of Instruction Fetch Mechanisms for High Issue Rates. International Symposium on Computer Architecture, June 1995.

High-Bandwidth Fetch: Trace Cache



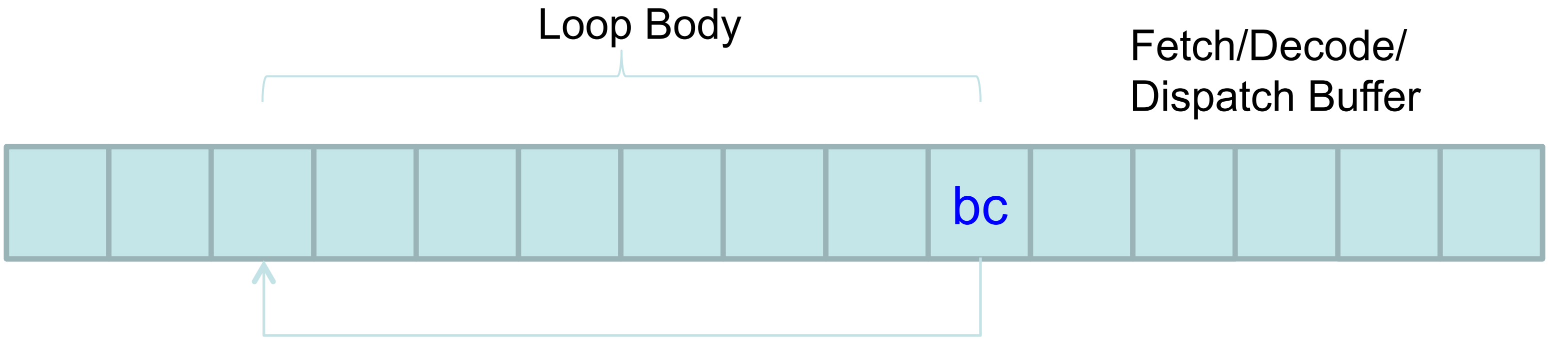
- Fold out taken branches by *tracing* instructions as they commit into a *fill buffer*
- Eric Rotenberg, S. Bennett, and James E. Smith. Trace Cache: A Low Latency Approach to High Bandwidth Instruction Fetching. MICRO, Dec 1996.

Intel Pentium 4 Trace Cache


Front-End BTB
Instruction TLB and Prefetcher
Level-Two Unified Data and Instruction Cache
Instruction Decode
Trace Cache BTB
Trace Cache
Instruction Fetch Queue
To renamer, execute, etc.

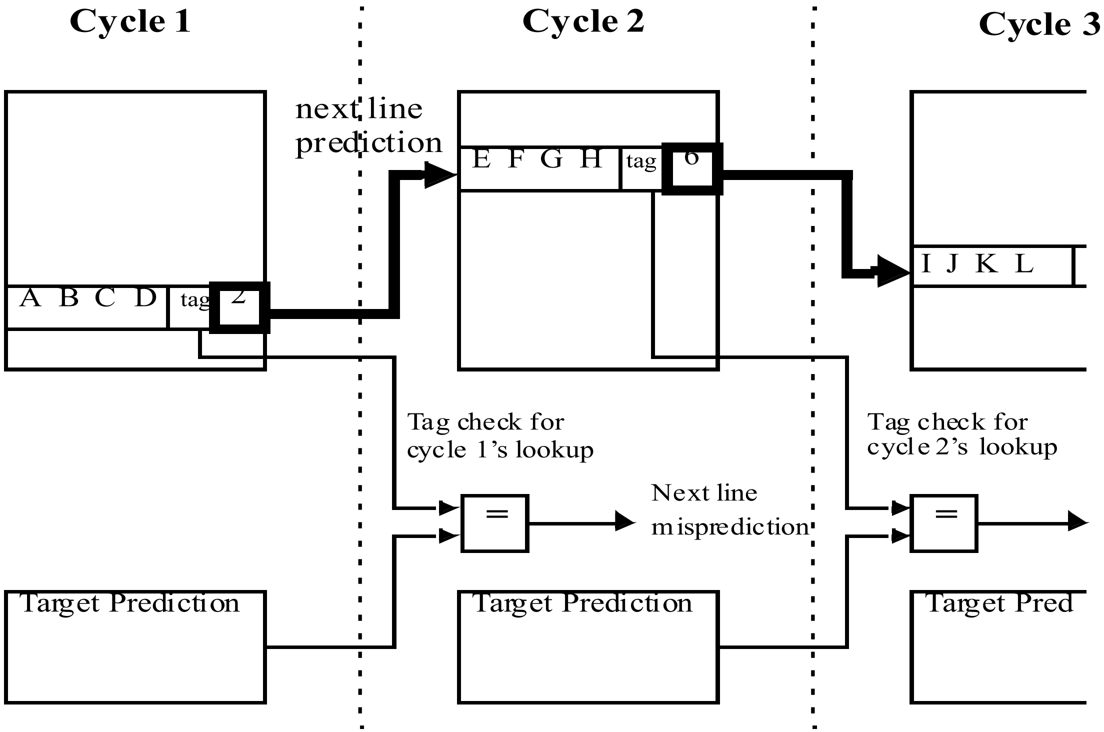
- No first-level instruction cache: trace cache only
- Trace cache BTB identifies next trace
- Miss leads to fetch from level two cache
- Trace cache instructions are decoded (uops)
- Cache capacity 12k uops
  - Overwhelmed for database applications
  - Serial decoder becomes performance bottleneck

High-Bandwidth Fetch: Loop Buffers



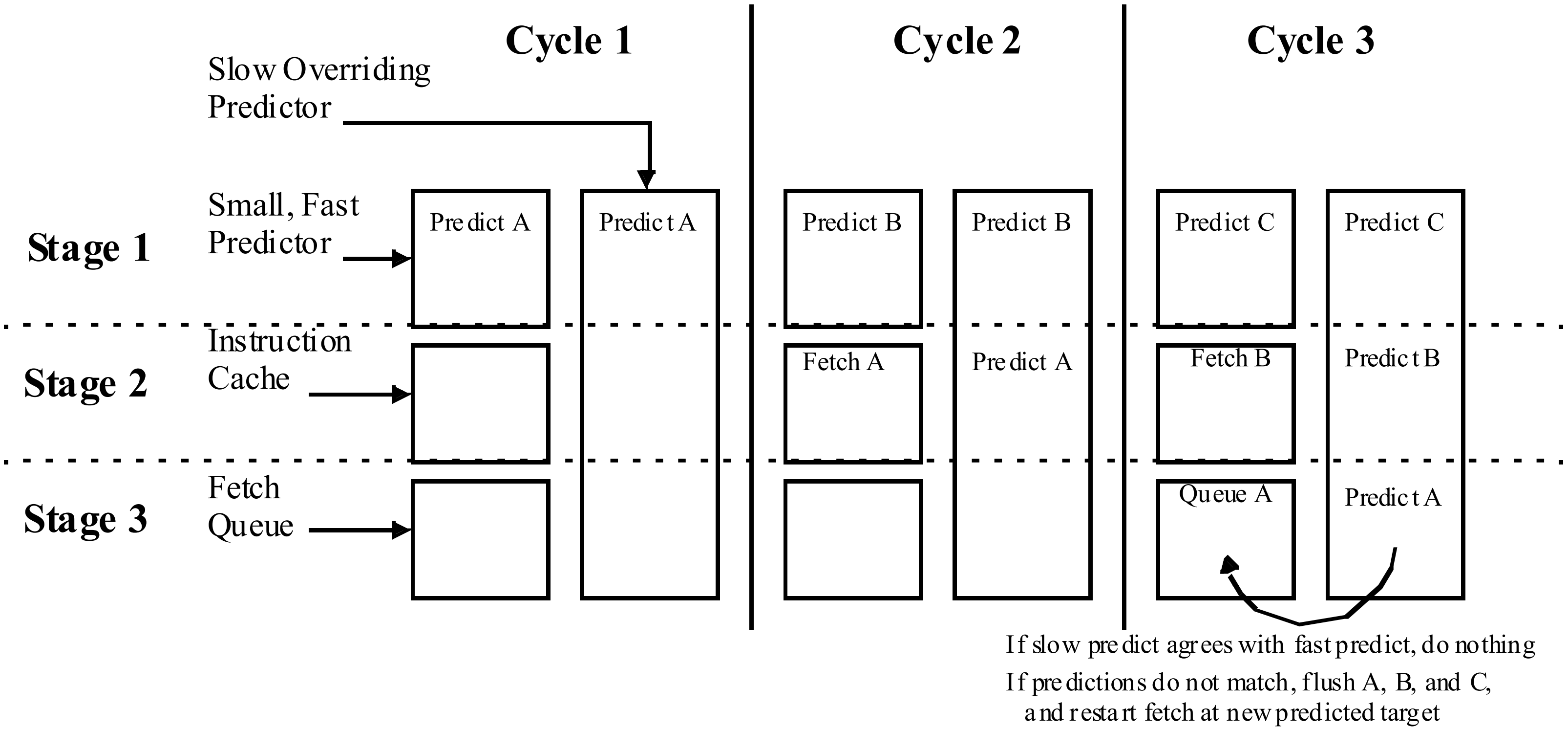
- History: AMD29K Branch Target Cache
  - Don't cache the target address; cache 4 instructions from the target itself
  - Avoid accessing I$ for first fetch group following a taken branch
  - If loop body is <= 4 instructions, effectively a loop cache
  - Room for 32/64 branch targets
- Also common in DSP designs, under s/w control (e.g. Lucent)
- Introduced in Intel Merom (Core 2 Duo)
  - Fetch buffer detects short backward branches, inhibits refetch from I$
- Intel Nehalem (Core i7)
  - Moved loop buffer after decoders: contains uops
- Intel Sandybridge
  - General-purpose uop cache (not just loops)
  - 1.5K capacity

High Frequency: Next-line Prediction



- Embed next fetch address in instruction cache
  – Enables high-frequency back-to-back fetch
- Brad Calder and Dirk Grunwald. Next Cache Line and Set Prediction. International Symposium on Computer Architecture, pp. 287-296, June 1995.

High Frequency: Overriding Predictors



- Simple, fast predictor turns around every cycle
- Smarter, slower predictor can override
- Widely used: PowerPC 604, 620, Alpha 21264

Advanced Branch Prediction Summary

- Control Flow Speculation
  - Branch Speculation
  - Mis-speculation Recovery
- Branch Direction Prediction
  - Static Prediction
  - Dynamic Prediction
  - Hybrid Prediction
- Branch Target Prediction
- High-bandwidth Fetch
- High-Frequency Fetch

Thank You