Entangling Prefetchers

Alberto Ros

University of Murcia, Spain

Oct 22, 2021

PREFETCHING

- Processors need to fetch instructions and data from memory
- High-performance processors would need a very large memory with a low access latency
- This is not possible due to technology limitations
- Computer architects came with a solution to this problem: prefetching

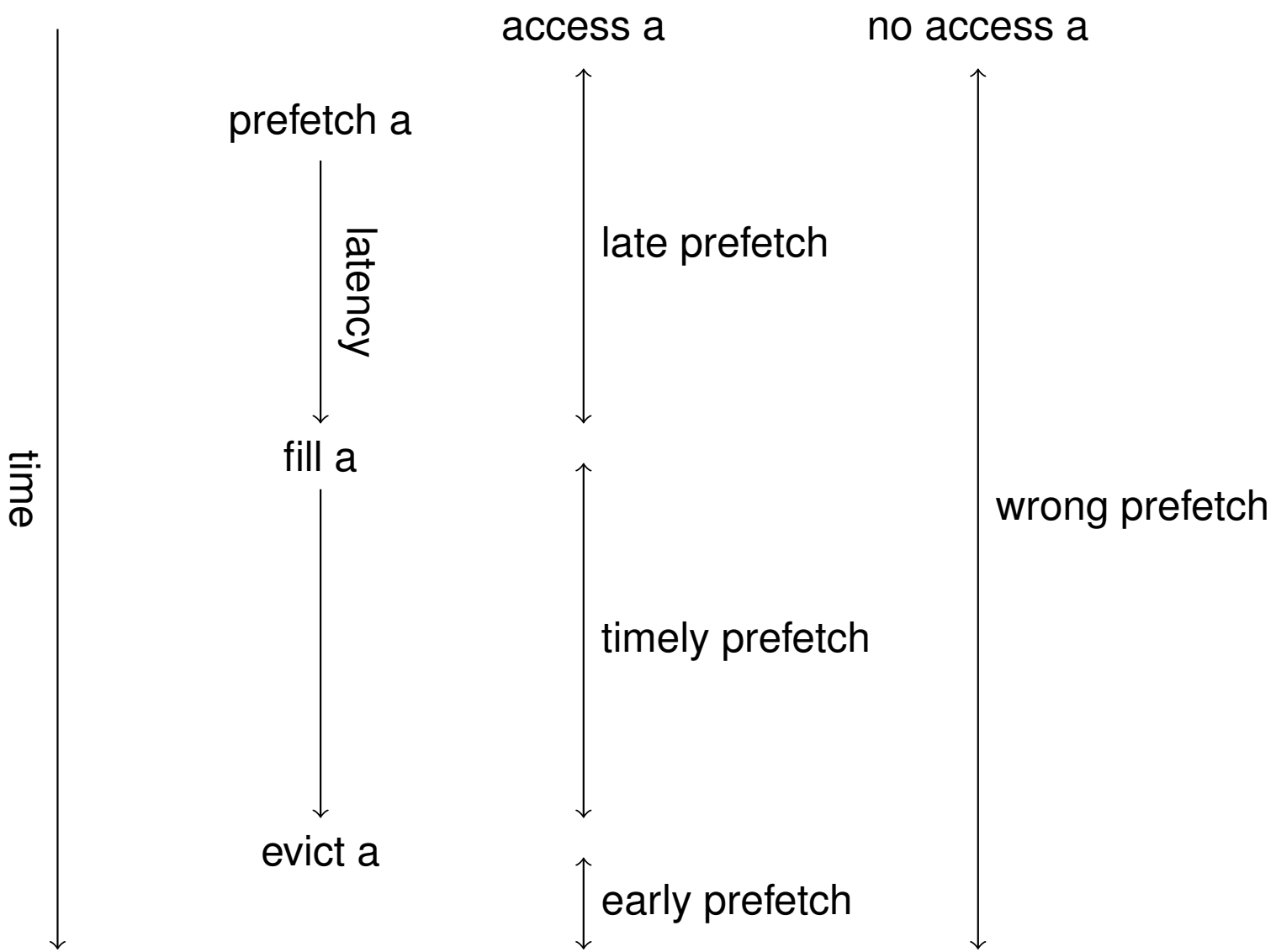
PREFETCHING

- Processors need to fetch instructions and data from memory
- High-performance processors would need a very large memory with a low access latency
- This is not possible due to technology limitations
- Computer architects came with a solution to this problem: prefetching

PREFETCHING

Predict which memory addresses will be accessed by the processor and fetch them before the processor requests them

TIMELINESS


access a
no access a
prefetch a
latency
late prefetch
time
fill a
wrong prefetch
timely prefetch
evict a
early prefetch

PREFETCHING METRICS

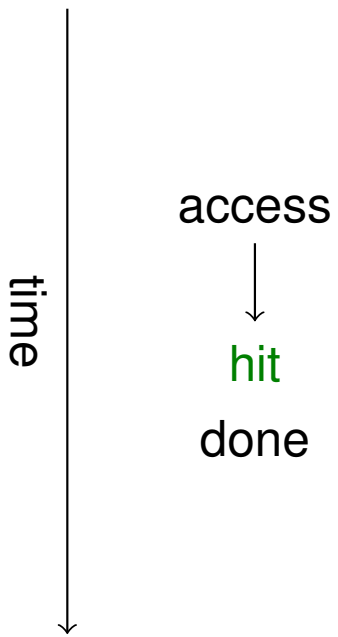
- COVERAGE
  - Fraction of cache misses covered by the prefetch
  - Indicator of performance benefits

$$Coverage = \frac{timely\ \ prefs}{timely\ \ prefs + cache\ \ misses}$$

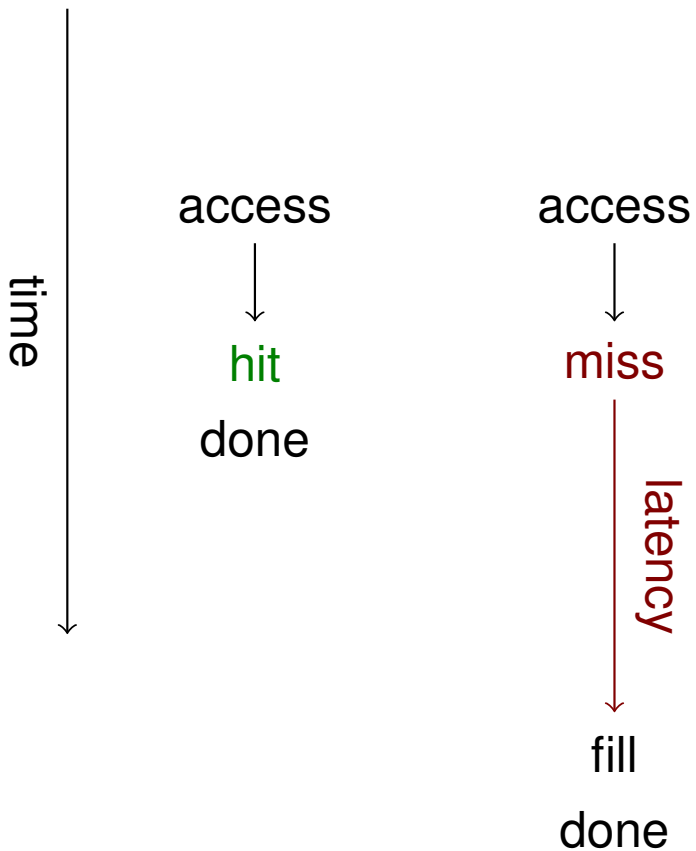
- ACCURACY
  - Fraction of useful prefetches
  - Indicator of efficiency

$$Accuracy = \frac{timely\ \ prefs + late\ \ prefs}{prefs\ \ to\ \ next\ \ cache\ \ level}$$

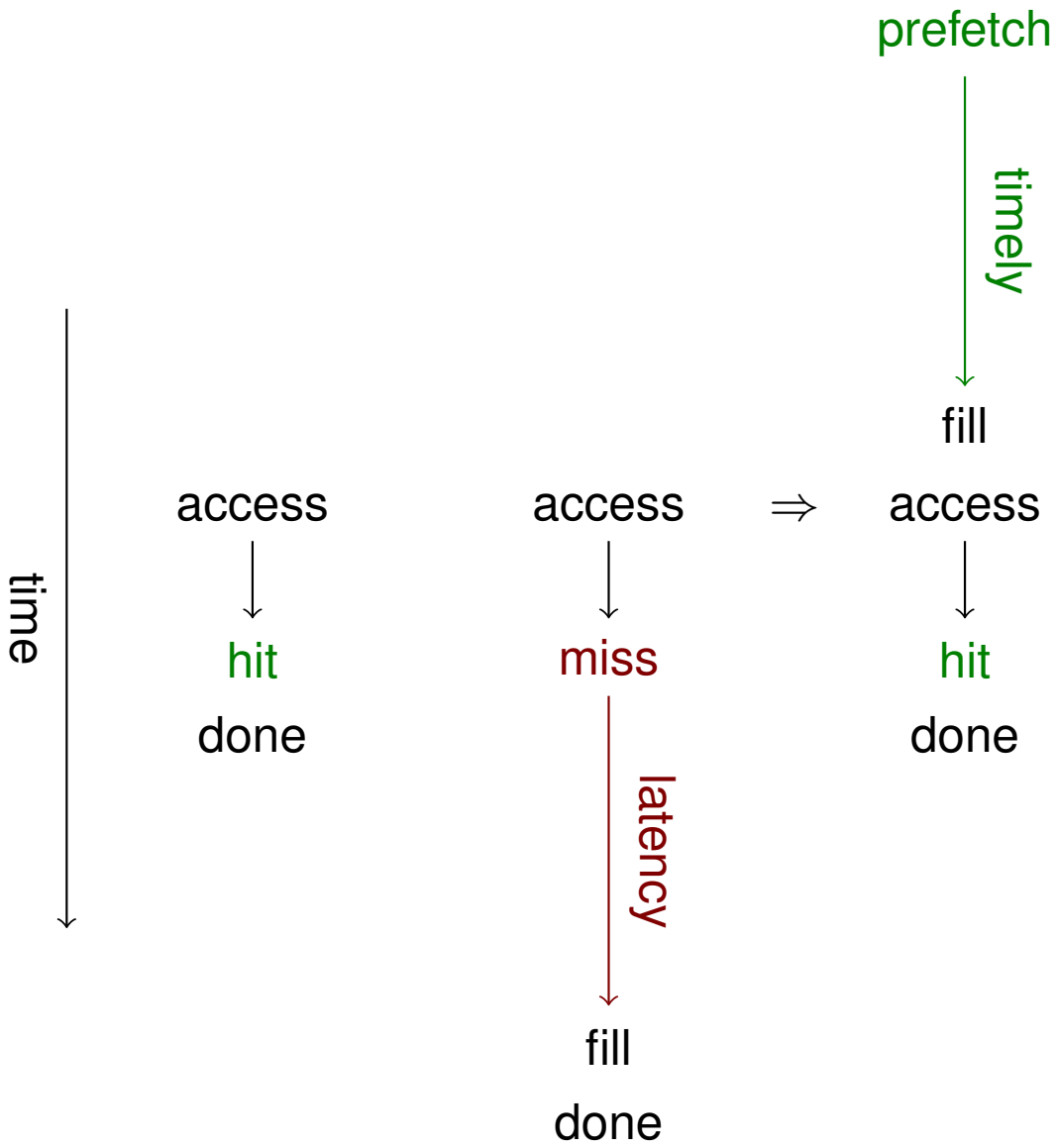
MOTIVATION: TIMELINESS



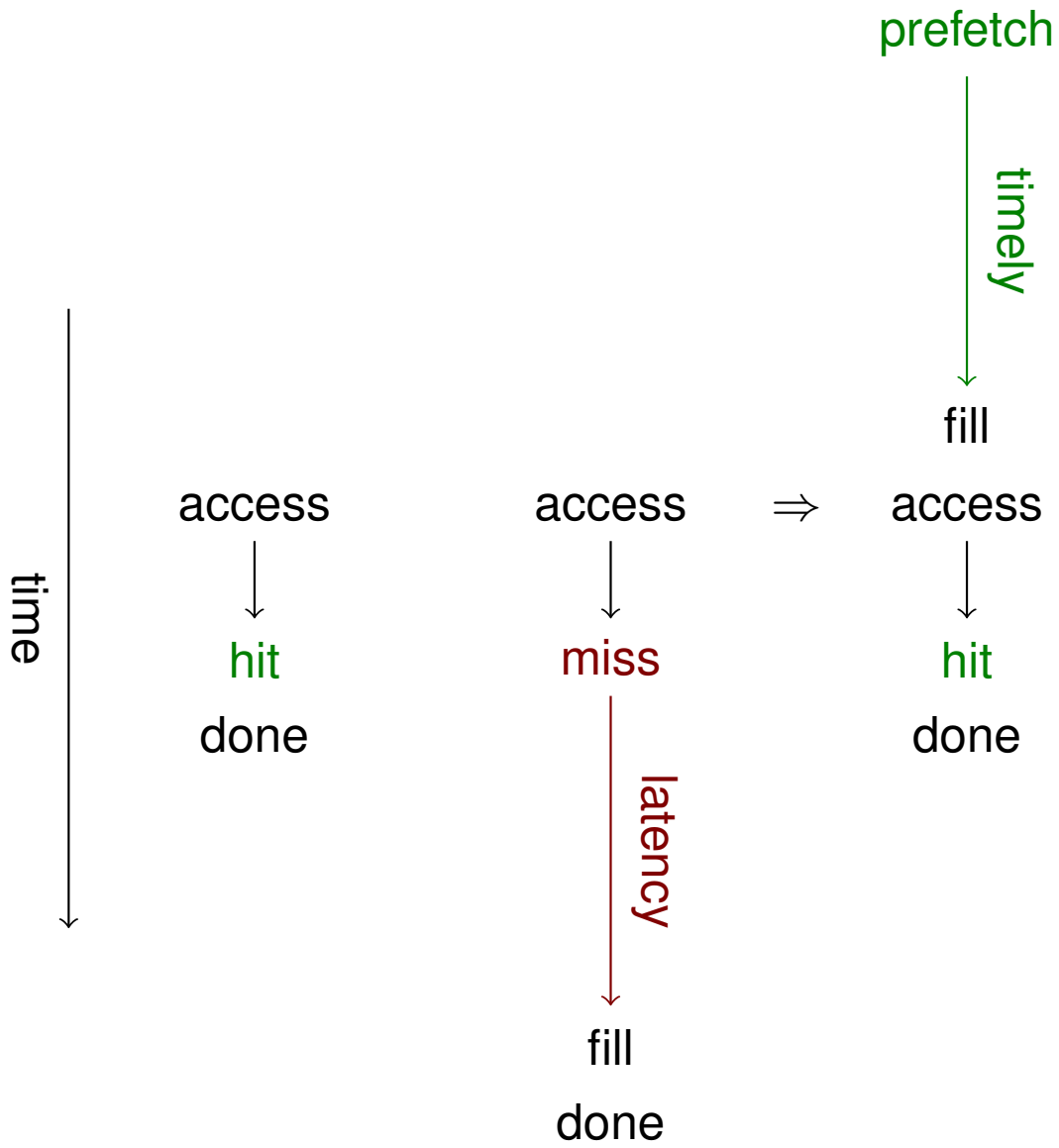
MOTIVATION: TIMELINESS



MOTIVATION: TIMELINESS



MOTIVATION: TIMELINESS



Timely prefetches for all misses:
Coverage 100%

And only for misses:
Accuracy 100%

Overview: Instruction prefetcher

- Server and cloud apps getting larger, far from fitting in L1I
  - $\Rightarrow$ stalls processor front-end, performance degradation

Overview: Instruction prefetcher

- Server and cloud apps getting larger, far from fitting in L1I
  - $\Rightarrow$ stalls processor front-end, performance degradation
- Prefetching instructions is fundamental for performance
  - Even when a decoupled front-end is implemented

Overview: Instruction prefetcher

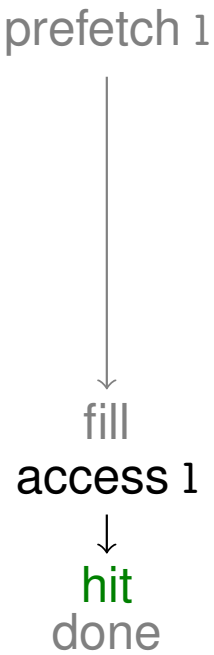
- Server and cloud apps getting larger, far from fitting in L1I
  - $\Rightarrow$ stalls processor front-end, performance degradation
- Prefetching instructions is fundamental for performance
  - Even when a decoupled front-end is implemented
- Our contribution: An Entangling prefetcher
  - Entangling: adaptive correlation based on latency
  - Winner of the 1st Instruction Prefetching Championship
  - A cost-effective prefetcher
  - Prefetcher code is available[1]

[1] https://github.com/alberto-ros/EntanglingInstructionPrefetcher

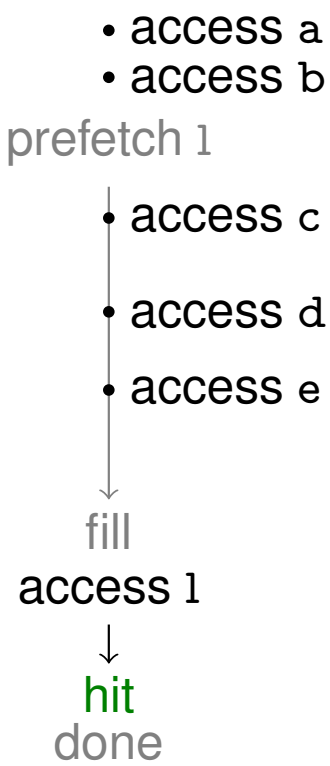
CONCEPT OF ENTANGLED ACCESSES

CONCEPT OF ENTANGLED ACCESSES

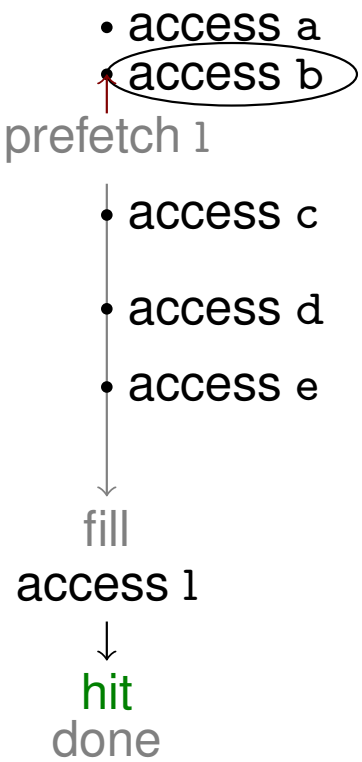
CONCEPT OF ENTANGLED ACCESSES



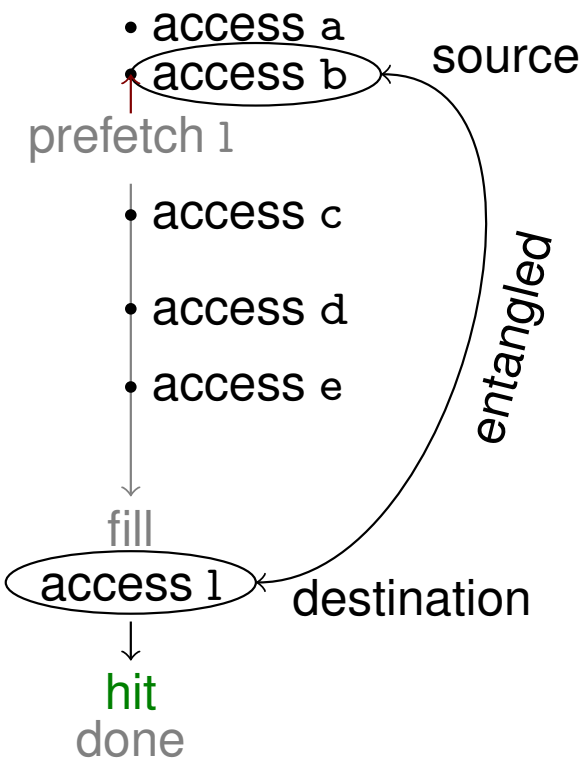
CONCEPT OF ENTANGLED ACCESSES



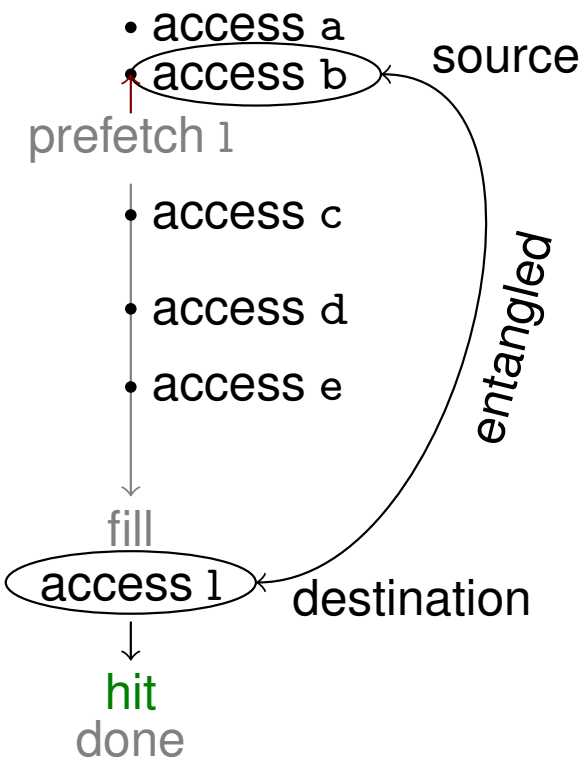
CONCEPT OF ENTANGLED ACCESSES

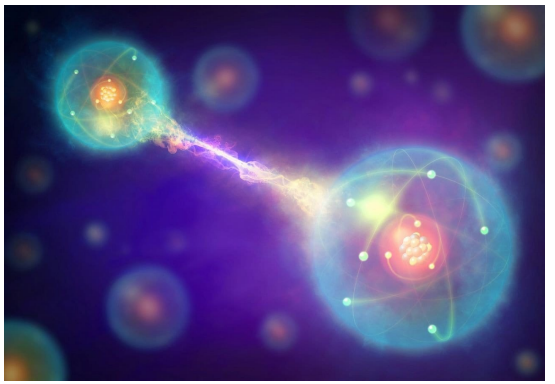


CONCEPT OF ENTANGLED ACCESSES



CONCEPT OF ENTANGLED ACCESSES

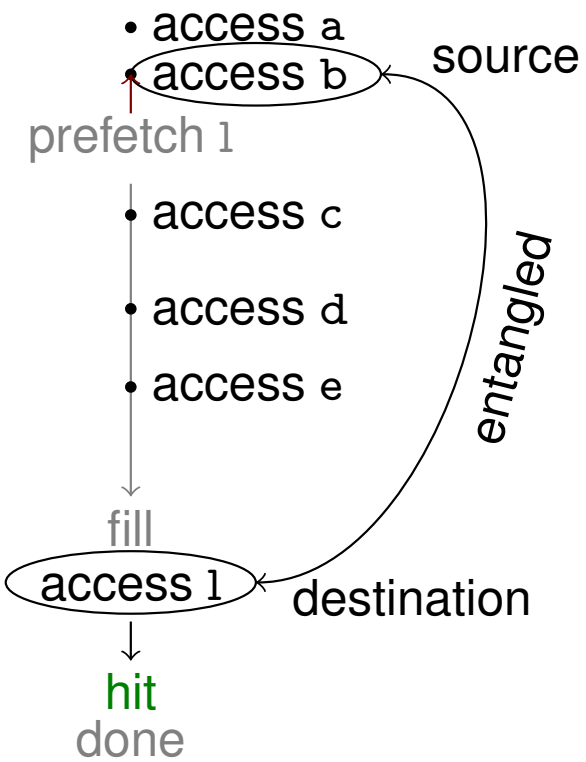


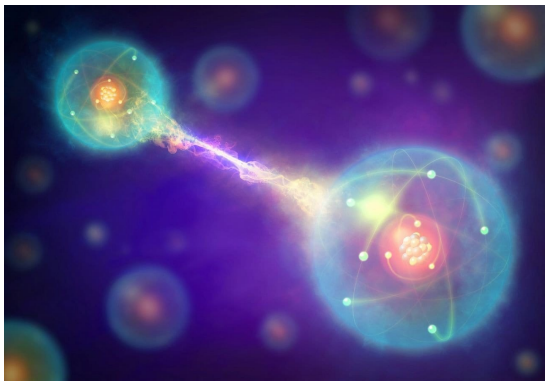


Quantum entanglement

(Image: © MARK GARLICK/SCIENCE PHOTO LIBRARY/Getty)

Concept of entangled accesses



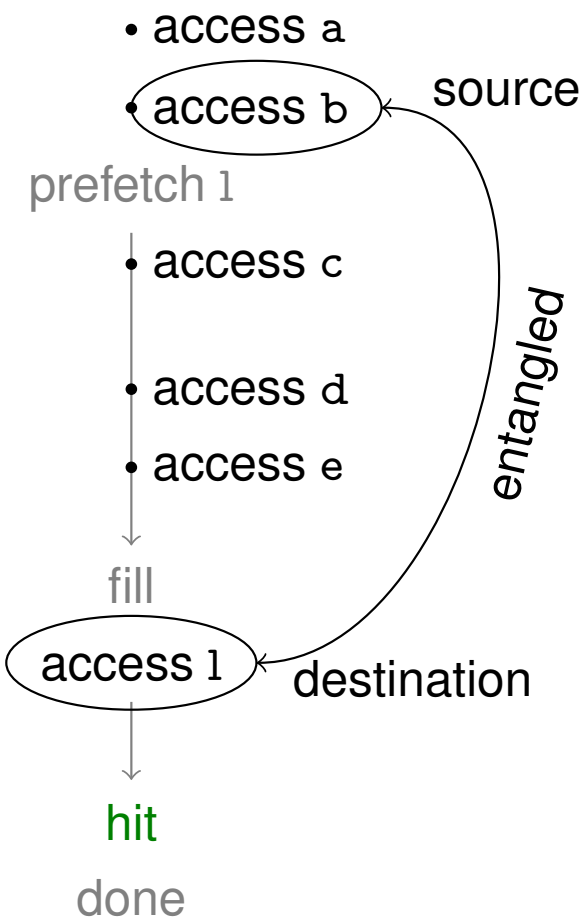


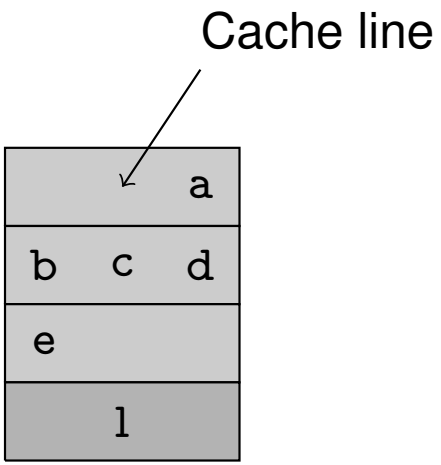
Quantum entanglement

(Image: © MARK GARLICK/SCIENCE PHOTO LIBRARY/Getty)

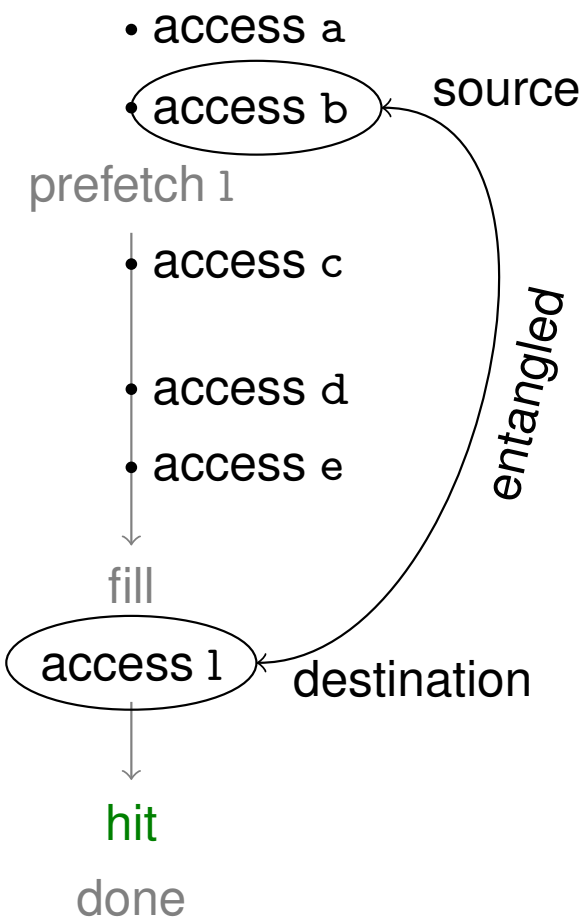
The Entangling Prefetcher for Instructions

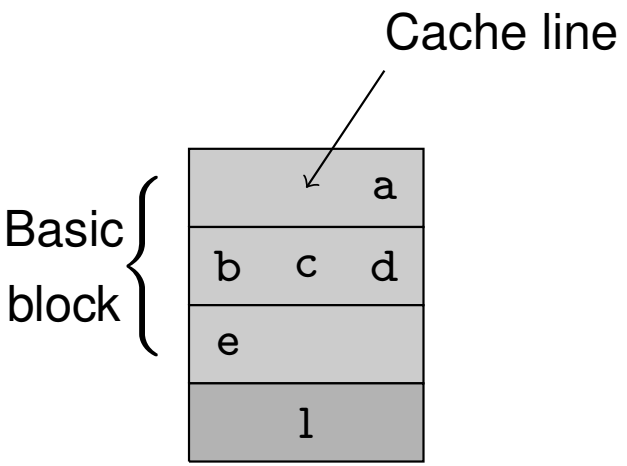
ENTANGLING CACHE LINES HEAD OF BASIC BLOCKS



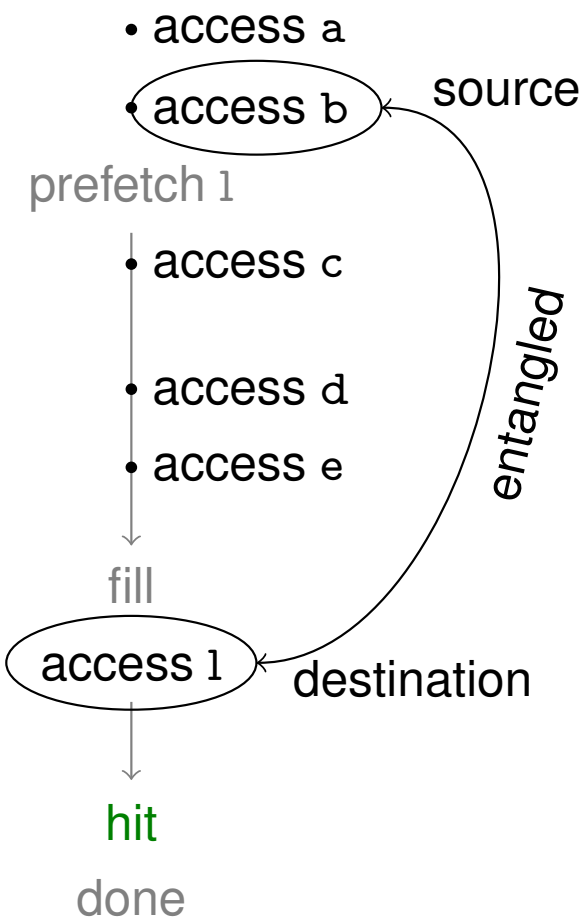


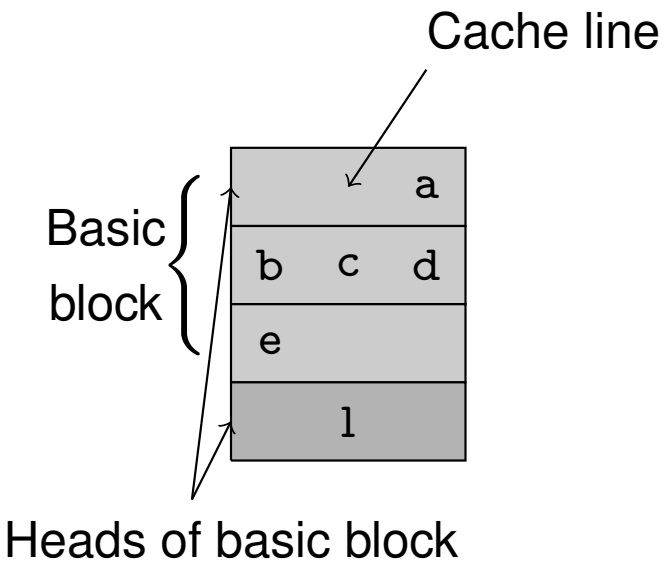
ENTANGLING CACHE LINES HEAD OF BASIC BLOCKS



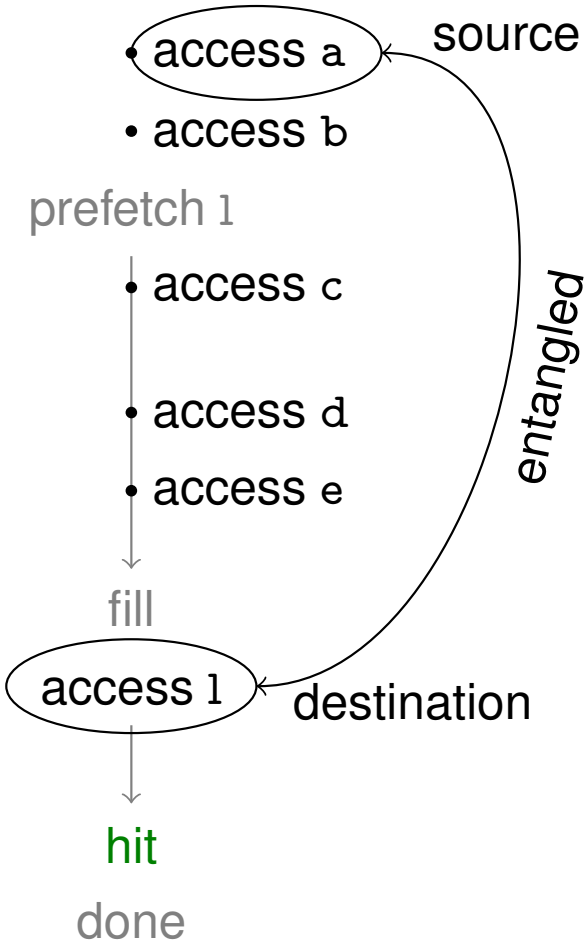


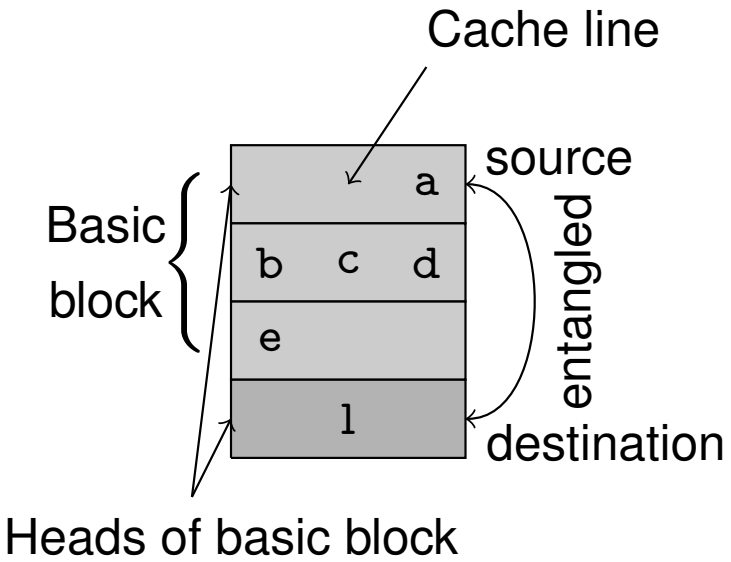
ENTANGLING CACHE LINES HEAD OF BASIC BLOCKS



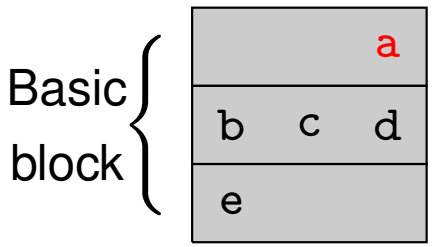


ENTANGLING CACHE LINES HEAD OF BASIC BLOCKS

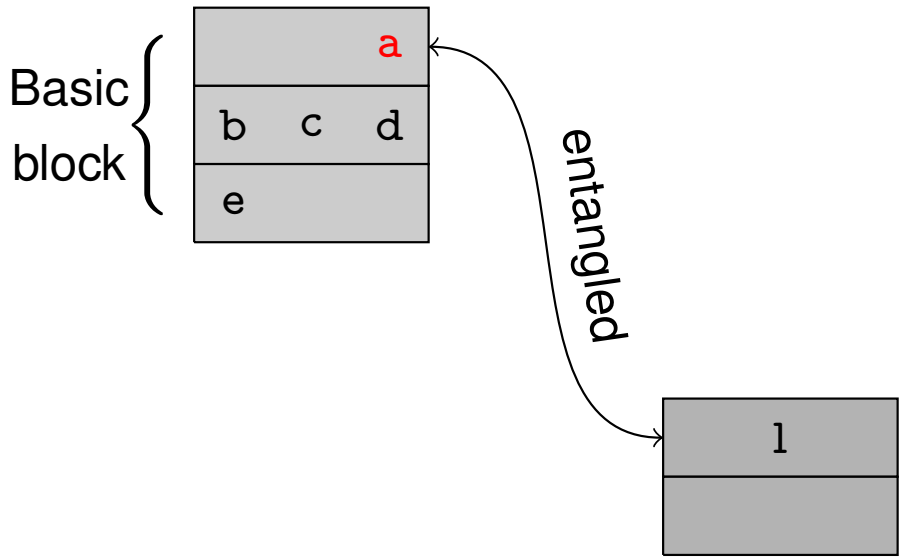




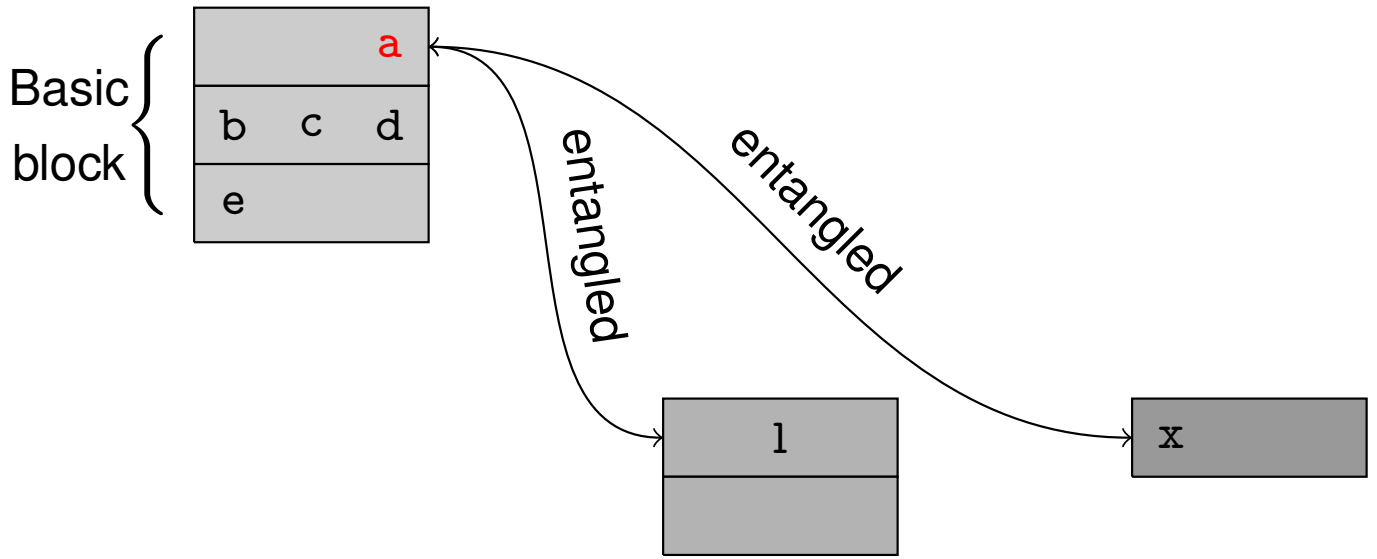
WHAT TO PREFETCH ON AN ACCESS TO a?



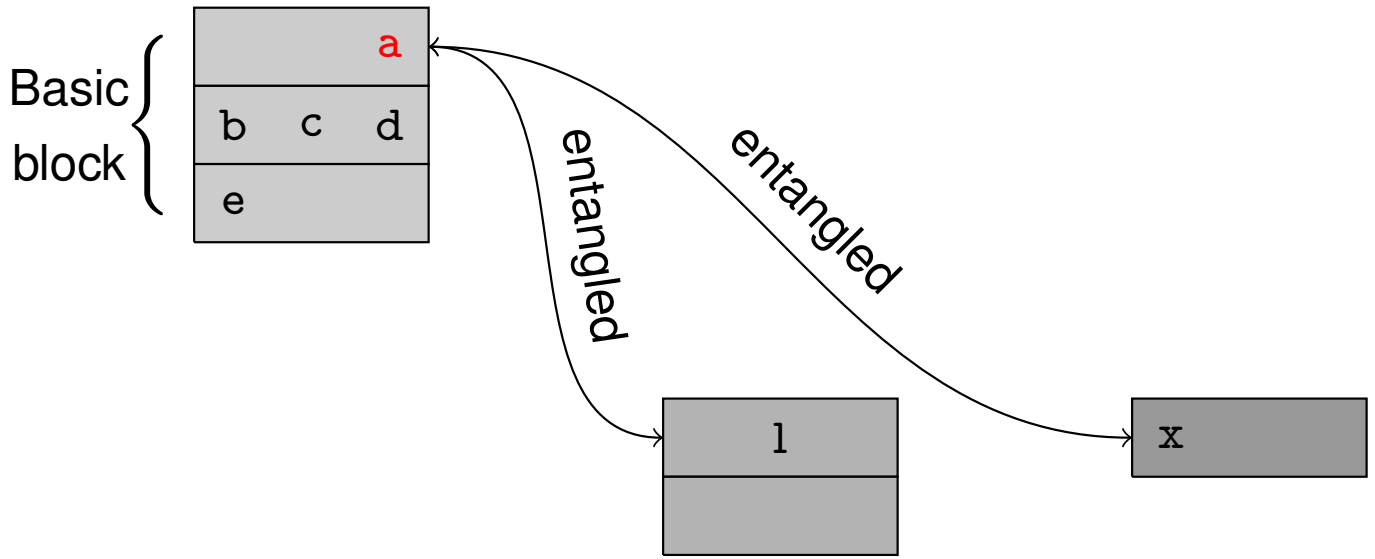
WHAT TO PREFETCH ON AN ACCESS TO a?



WHAT TO PREFETCH ON AN ACCESS TO a?

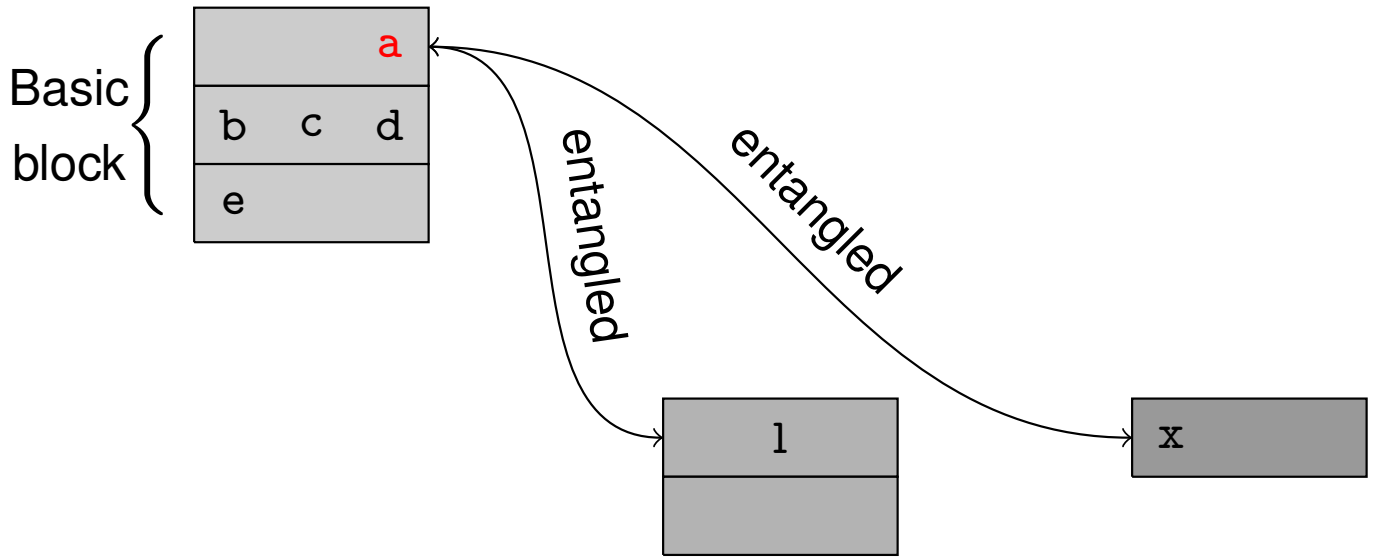


WHAT TO PREFETCH ON AN ACCESS TO a?



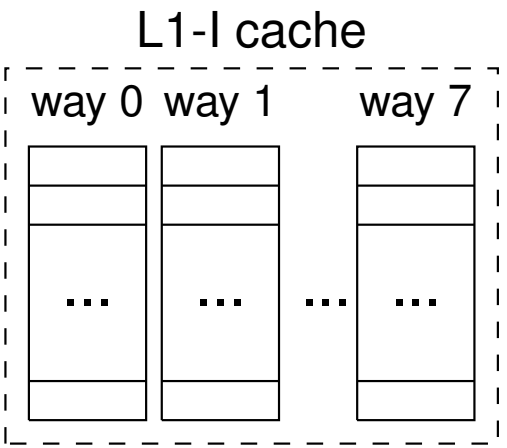
- Too much? (Max entangled = 6, Max BB size = 64)

WHAT TO PREFETCH ON AN ACCESS TO a?

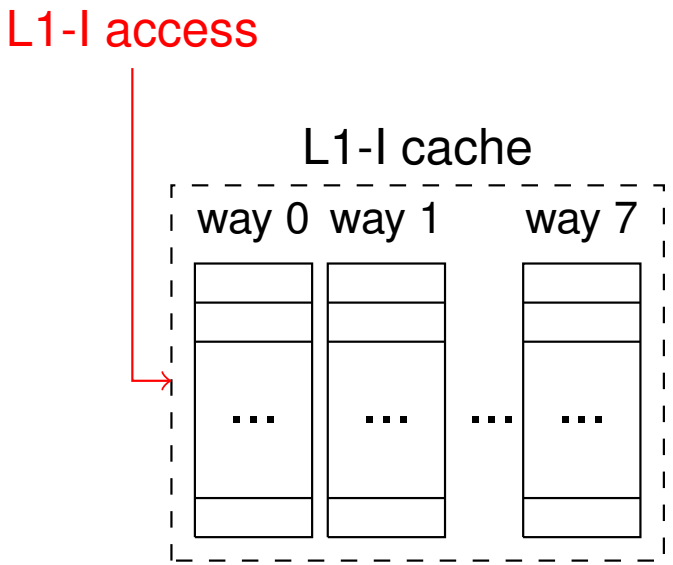


- Too much? (Max entangled = 6, Max BB size = 64)
  - Most of the time no prefetches are issued (no head of basic block)
  - Average number of prefetches per access to head of basic block ranging from ≈9 to ≈17

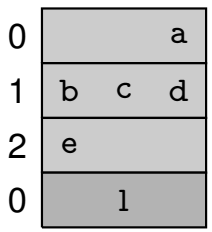
Design of the Entangling Prefetcher

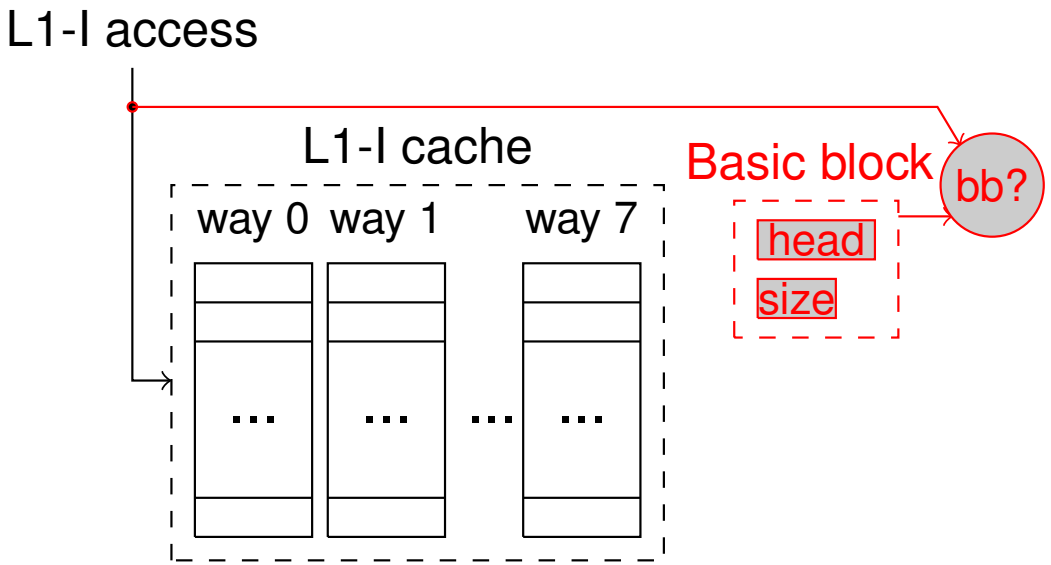


Design of the Entangling Prefetcher

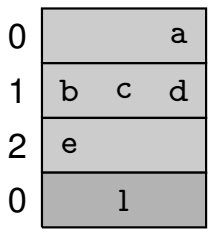


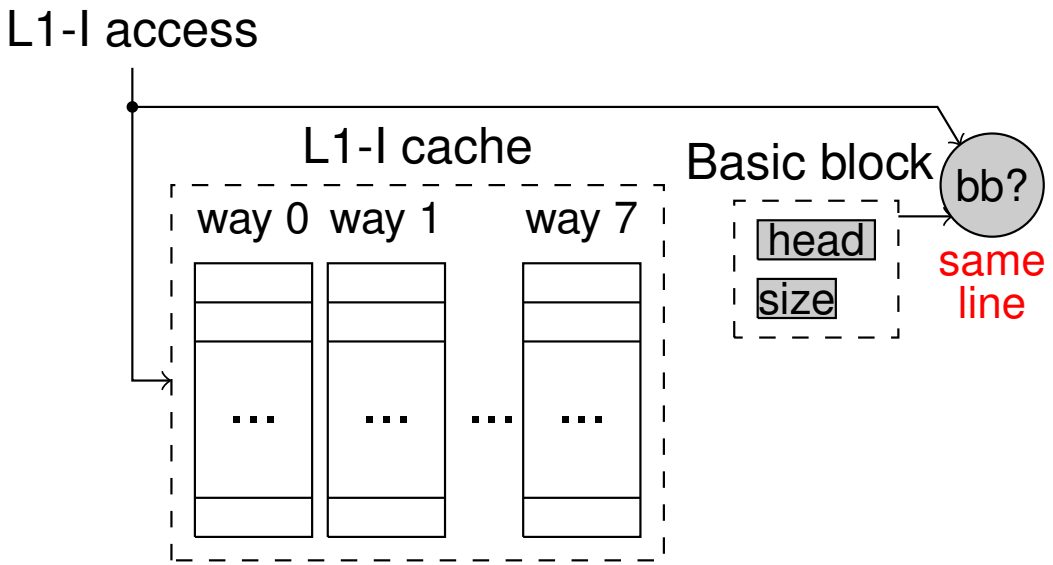
Design of the Entangling Prefetcher - Finding basic blocks



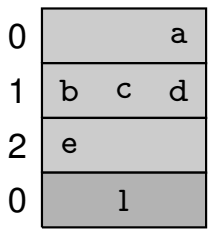


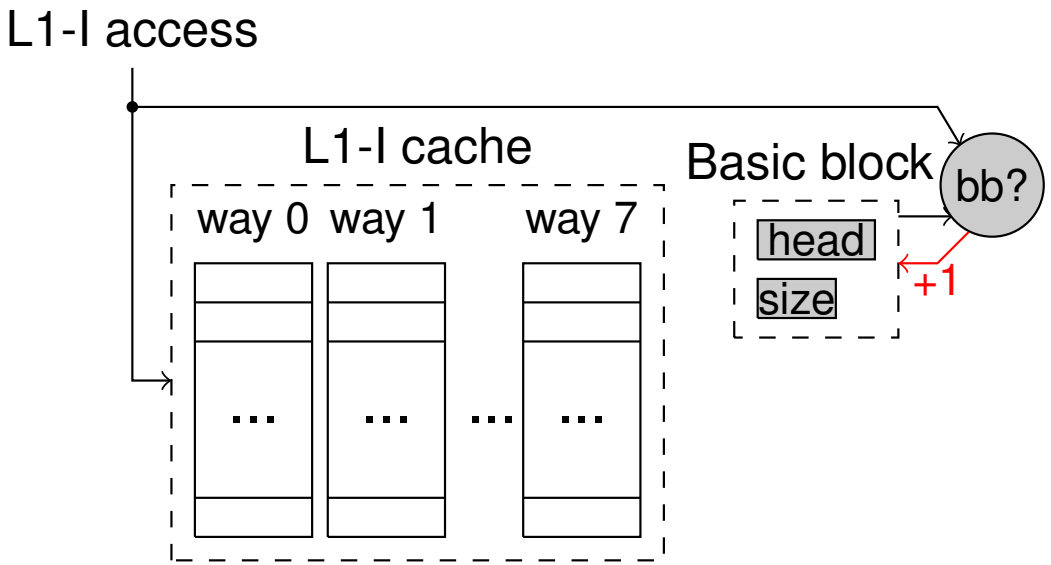
Design of the Entangling Prefetcher - Finding basic blocks



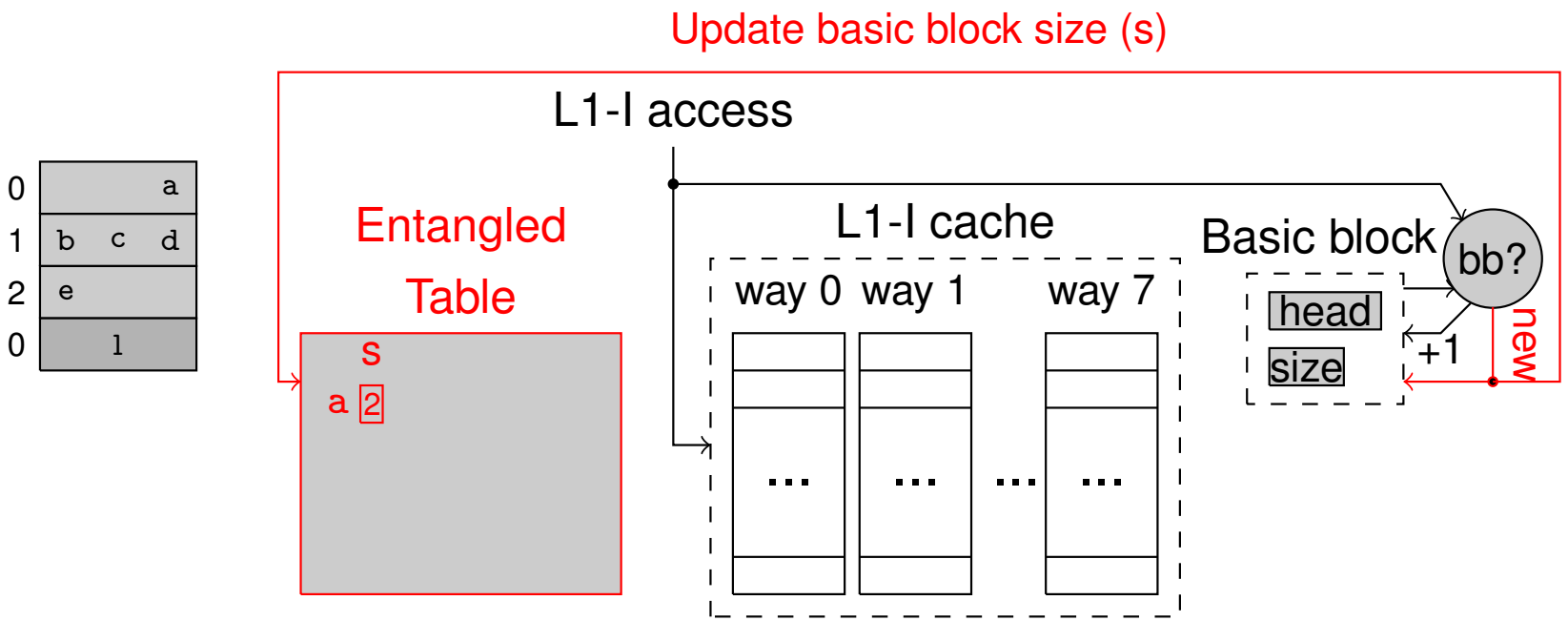


Design of the Entangling Prefetcher - Finding basic blocks

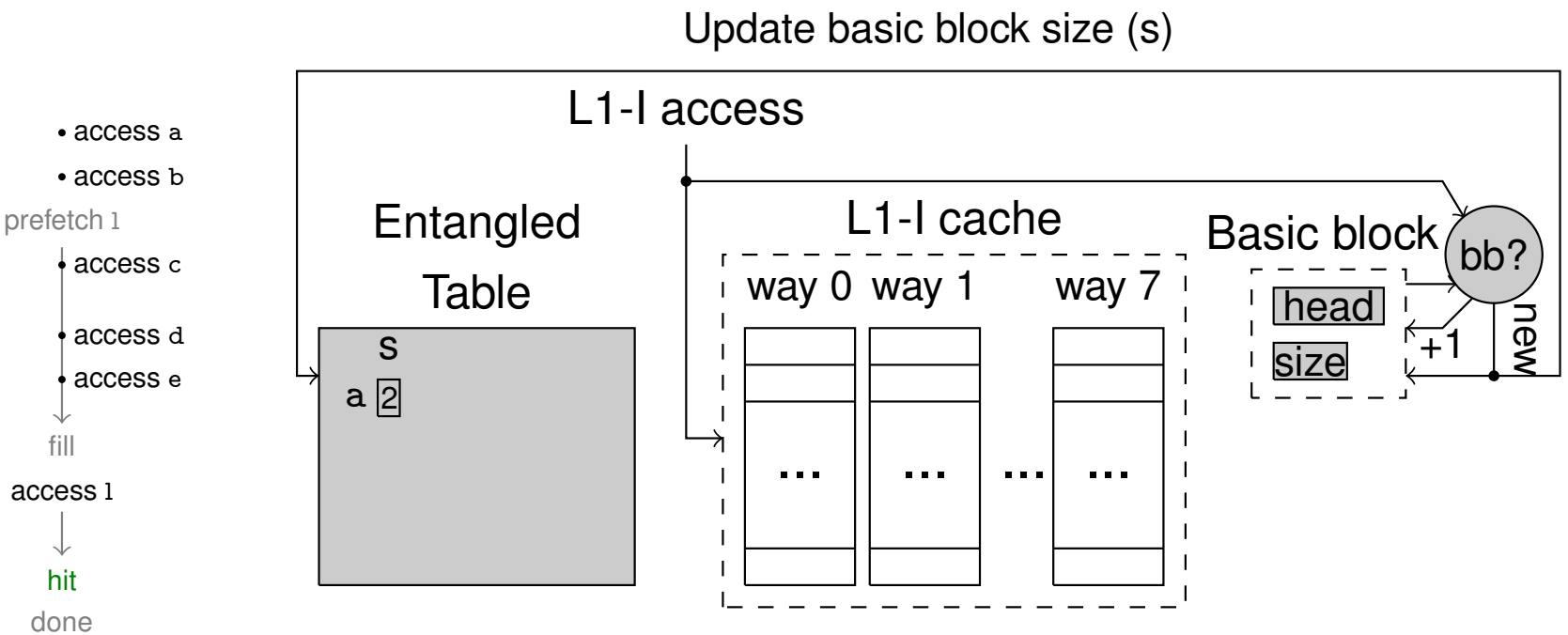




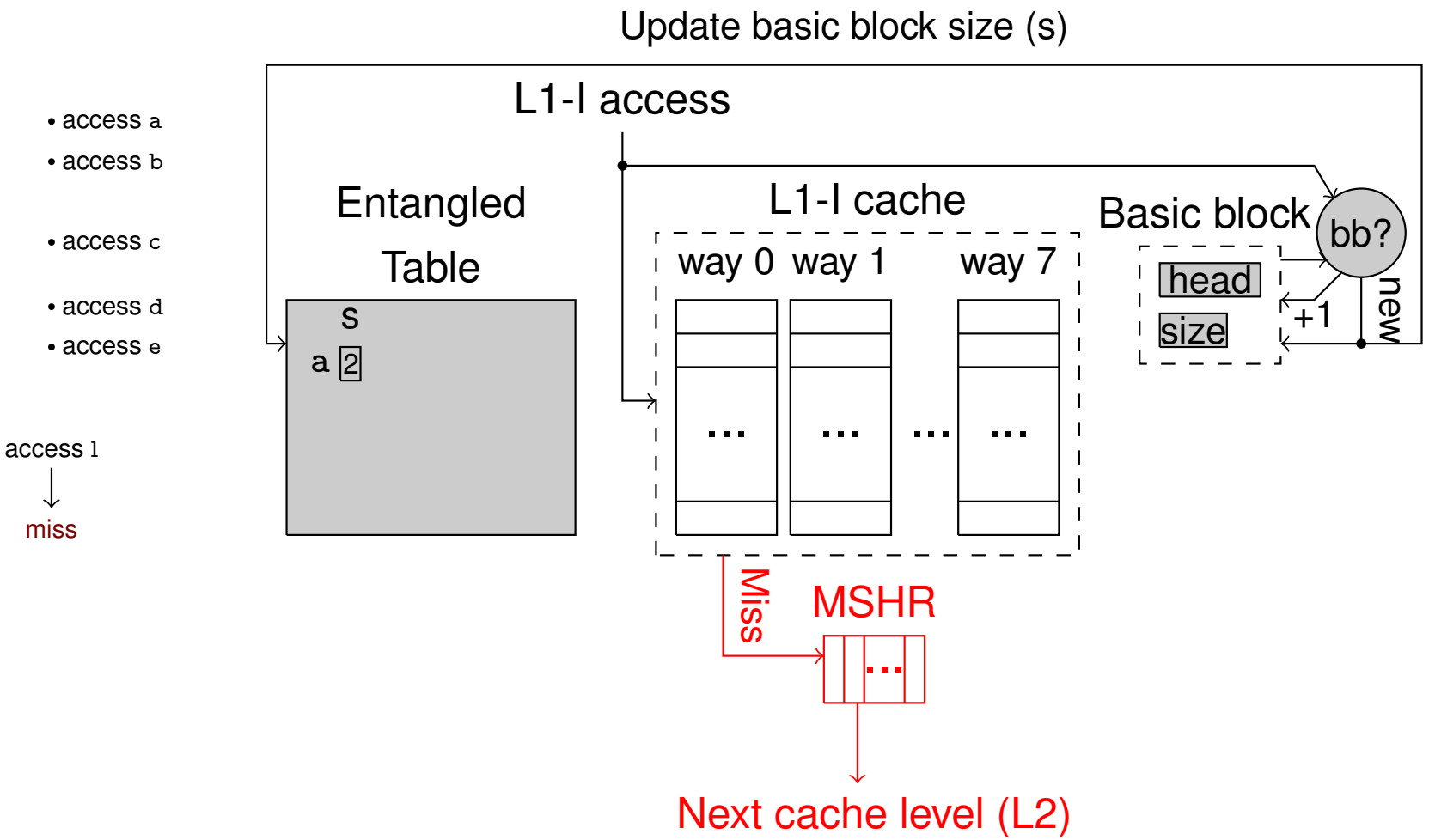
Design of the Entangling Prefetcher - Finding basic blocks



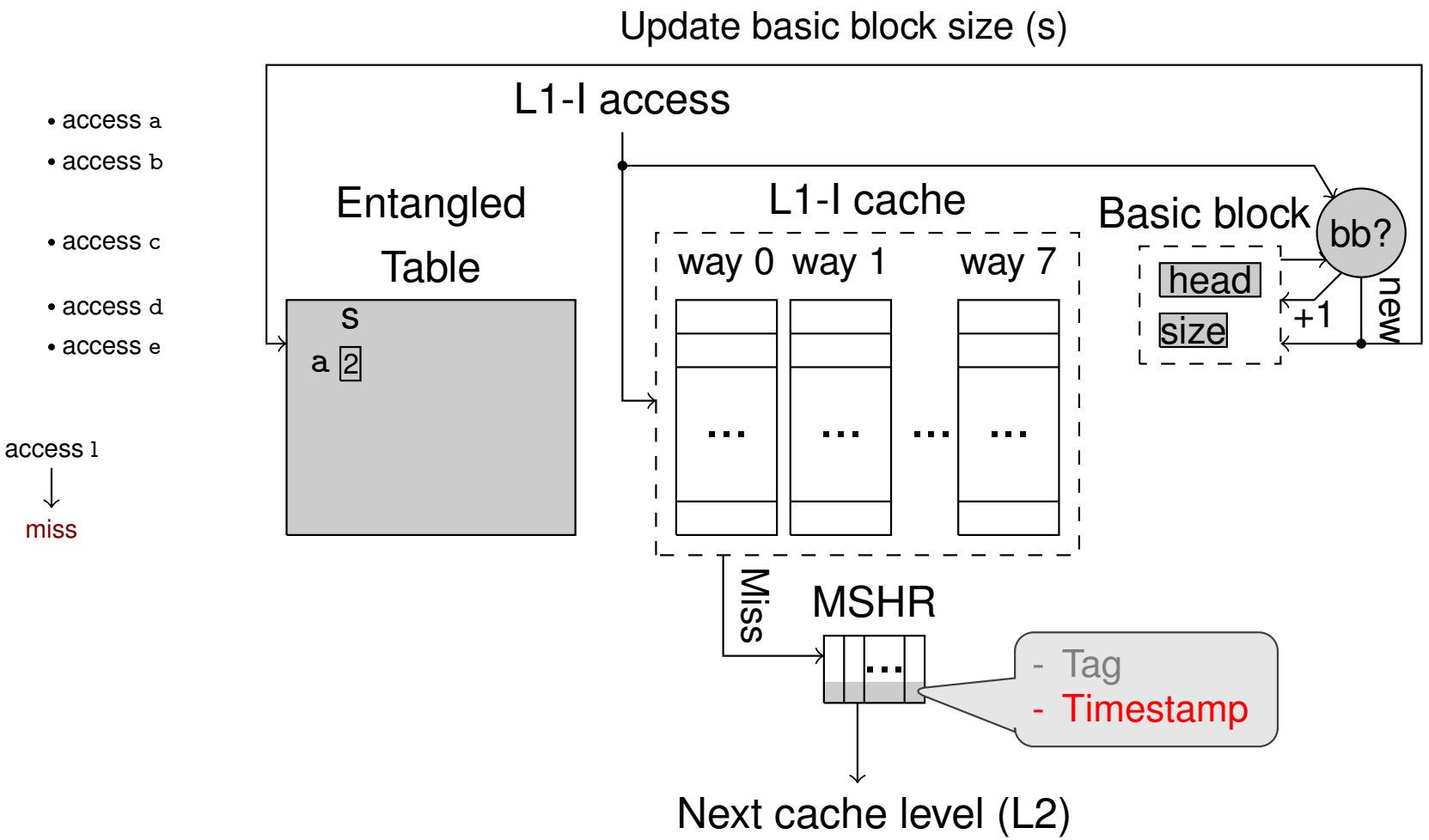
Design of the Entangling Prefetcher - Entangling cache lines



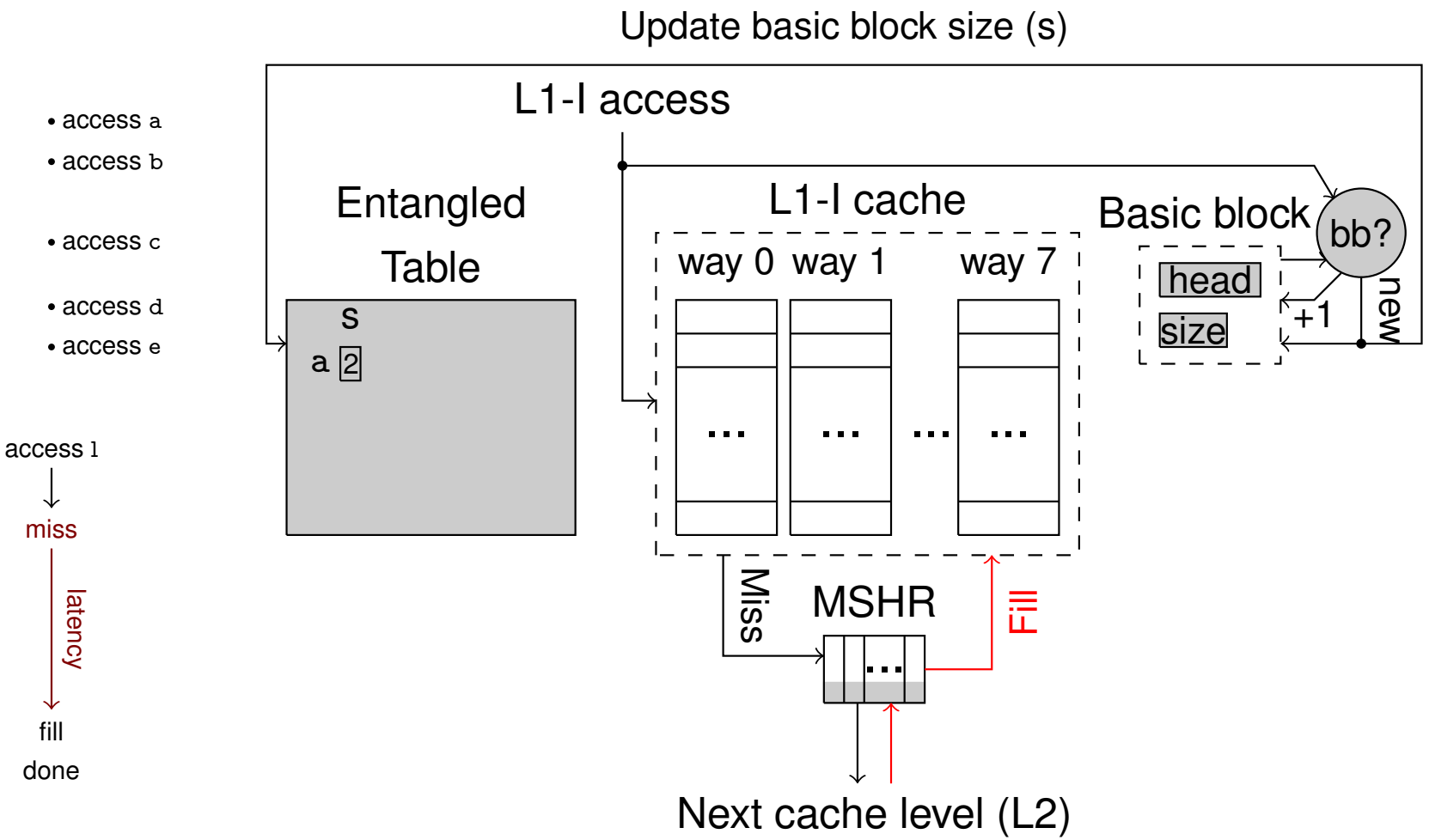
Design of the Entangling Prefetcher - Entangling cache lines



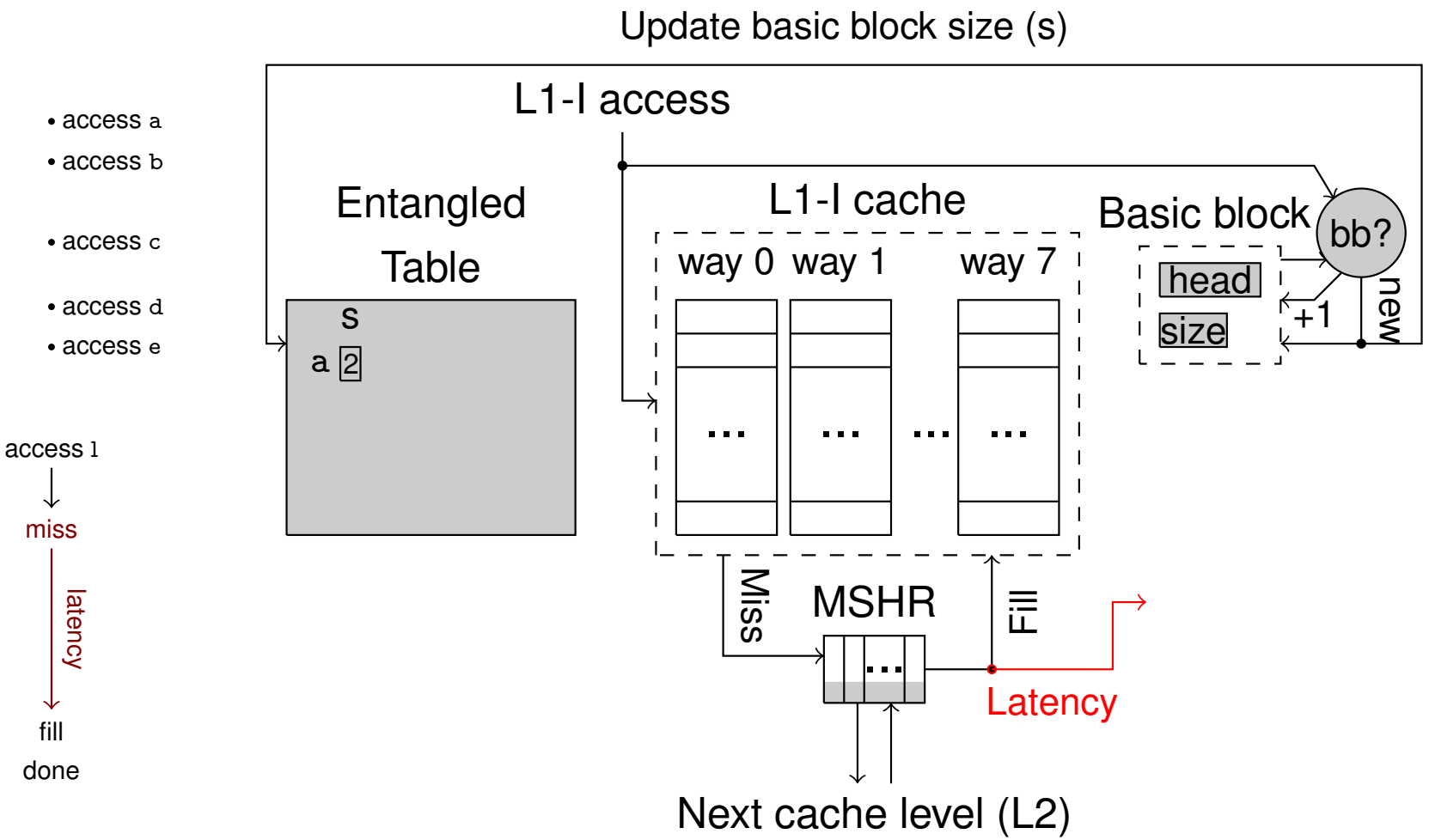
Design of the Entangling Prefetcher - Entangling cache lines



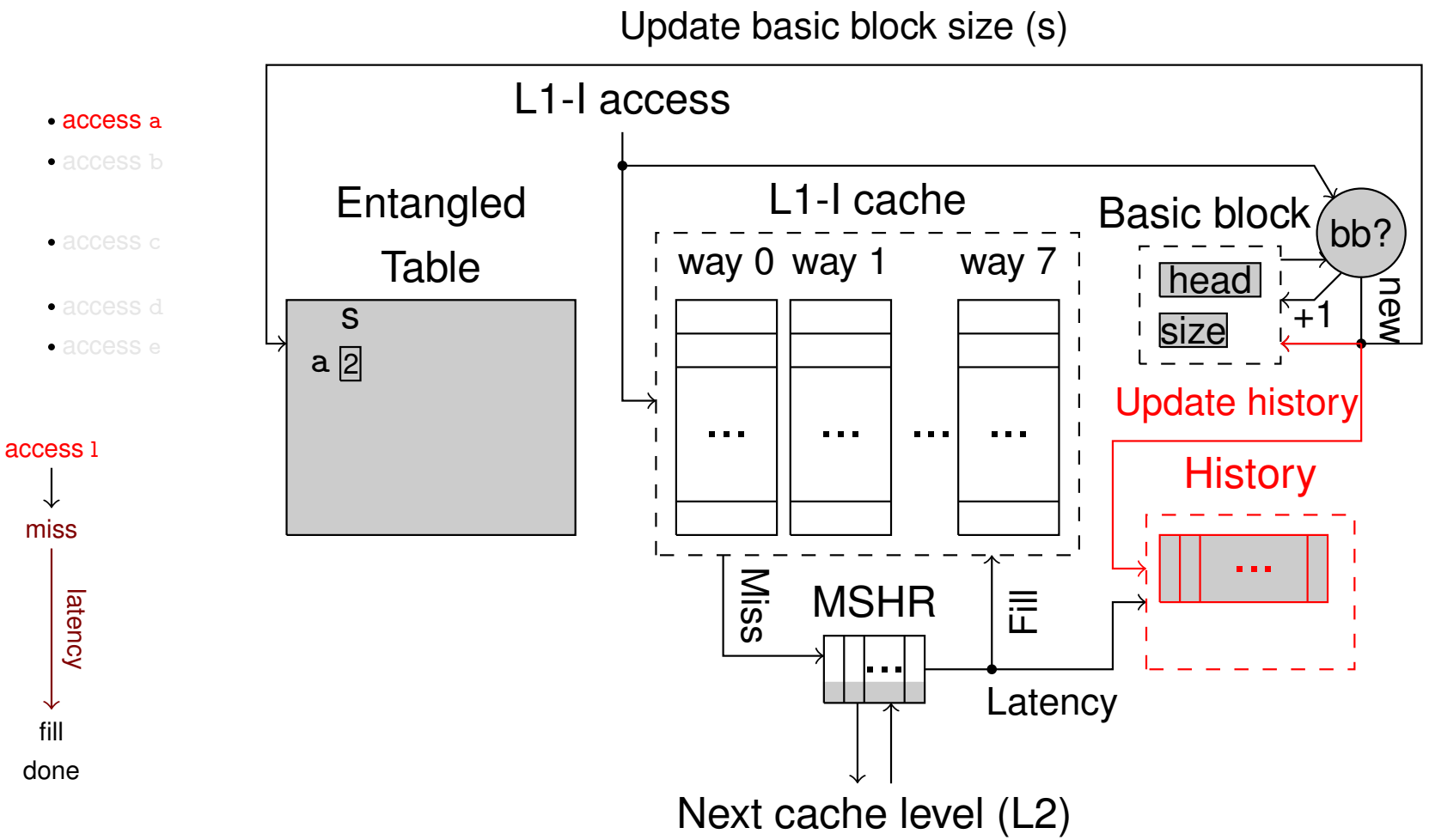
Design of the Entangling Prefetcher - Entangling cache lines



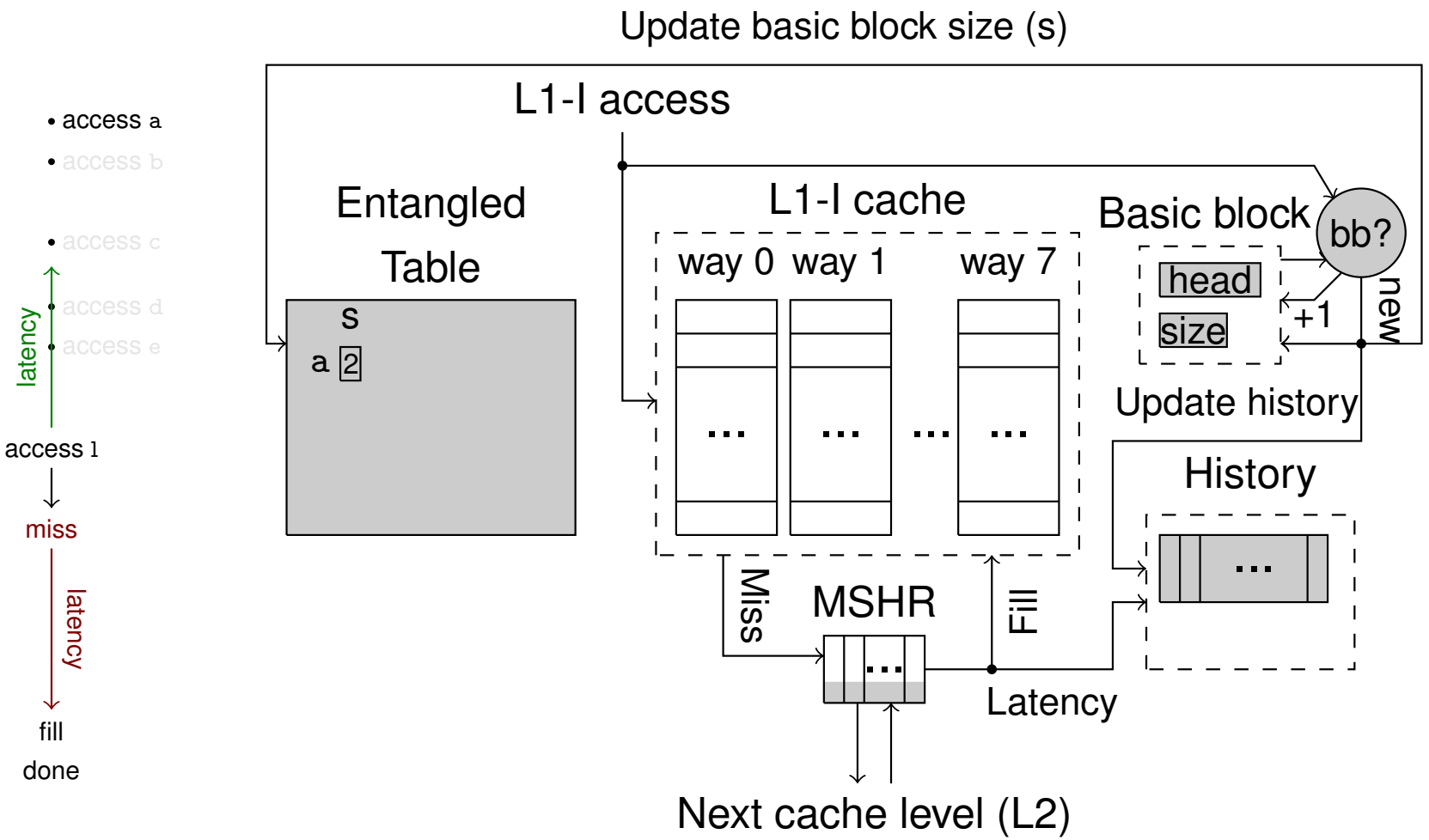
Design of the Entangling Prefetcher - Entangling cache lines



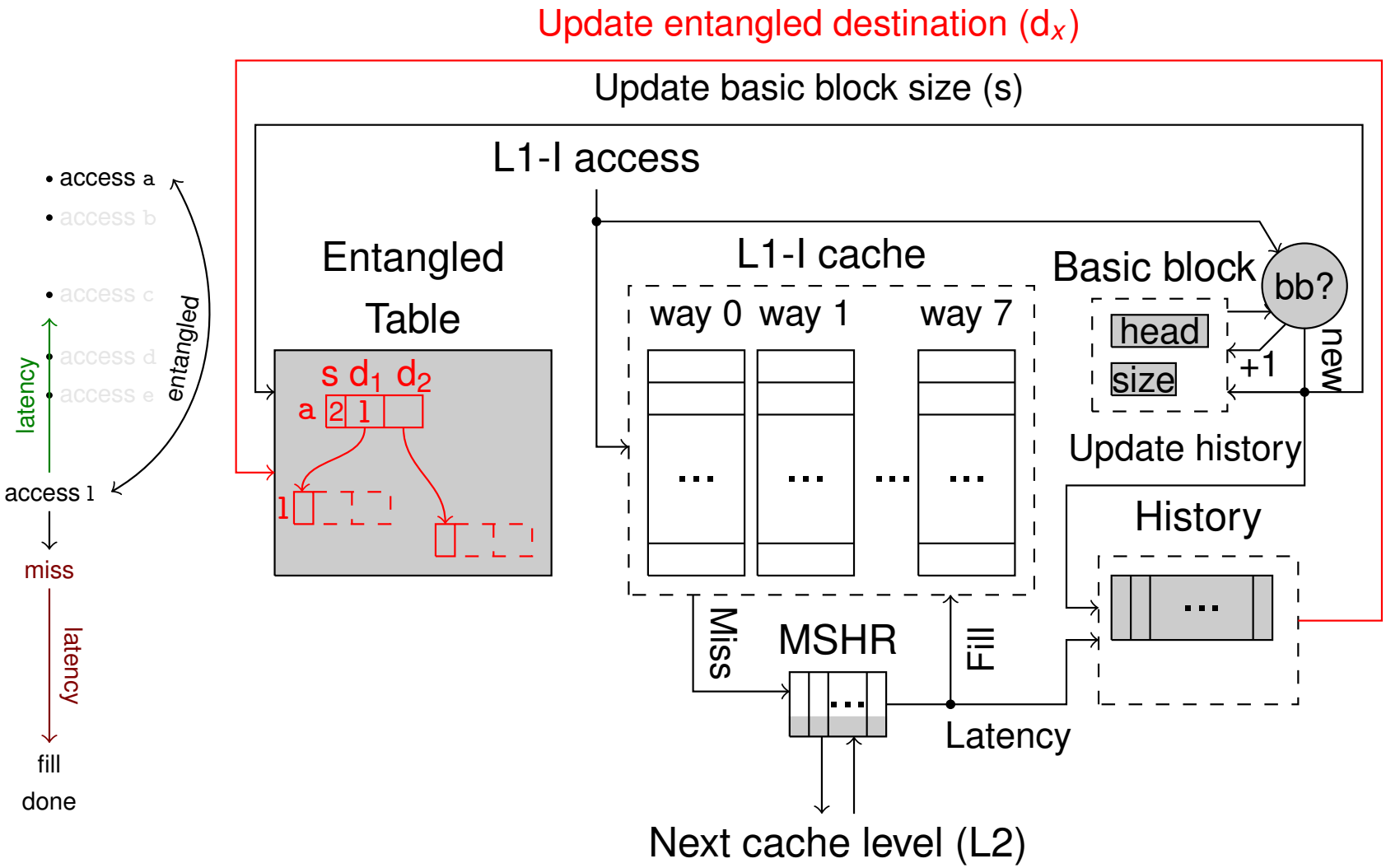
Design of the Entangling Prefetcher - Entangling cache lines



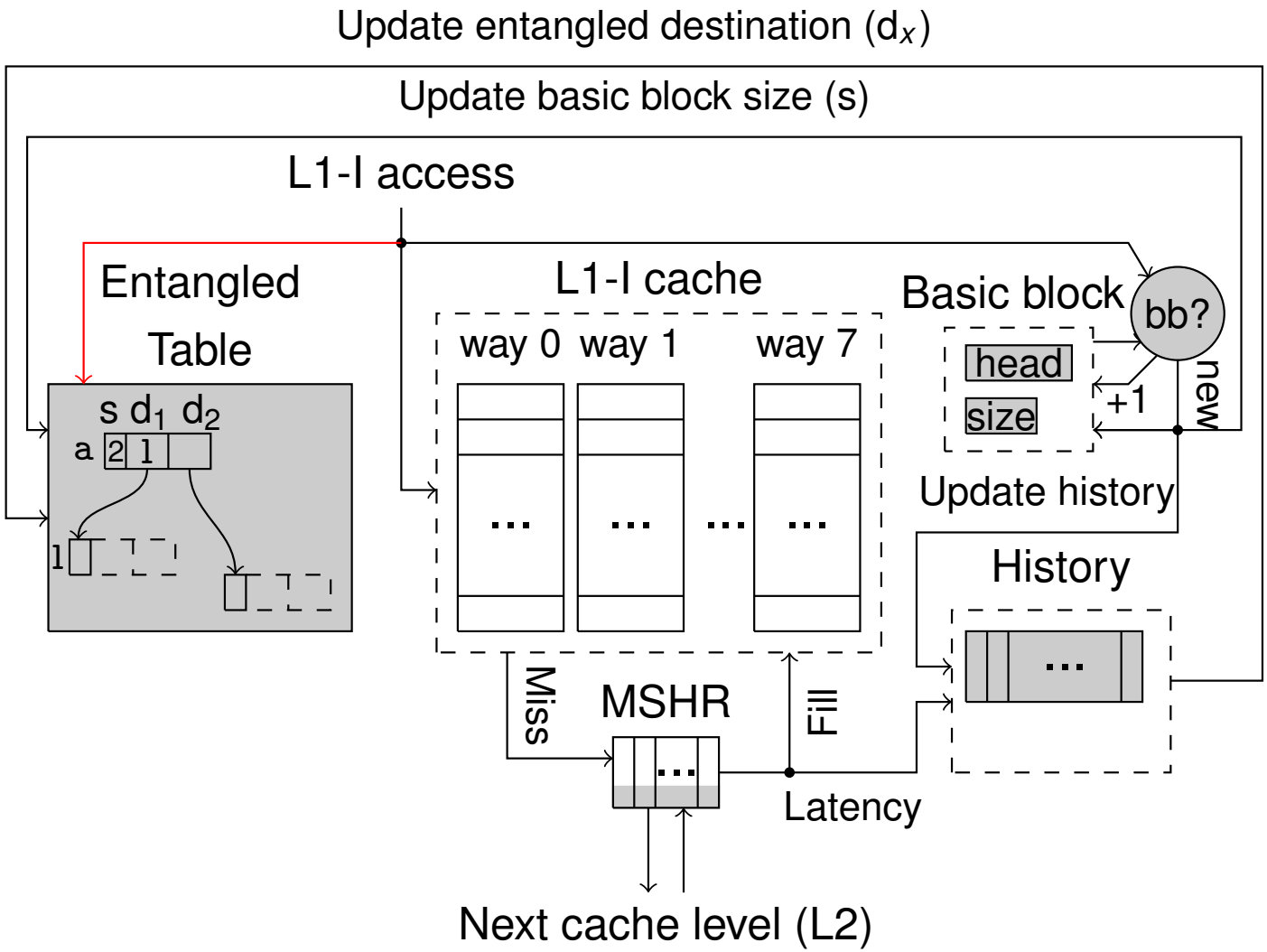
Design of the Entangling Prefetcher - Entangling cache lines



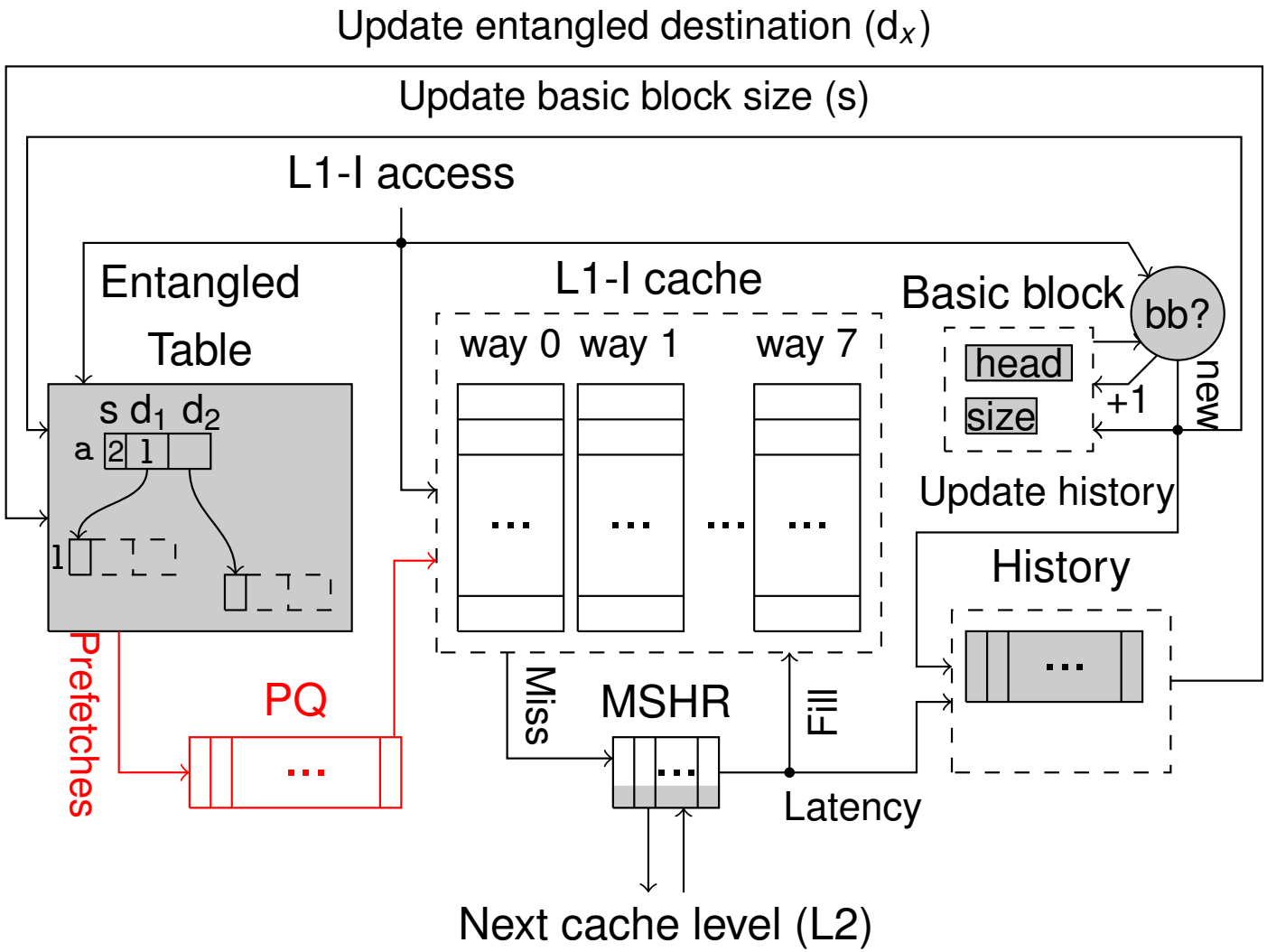
Design of the Entangling Prefetcher - Entangling cache lines



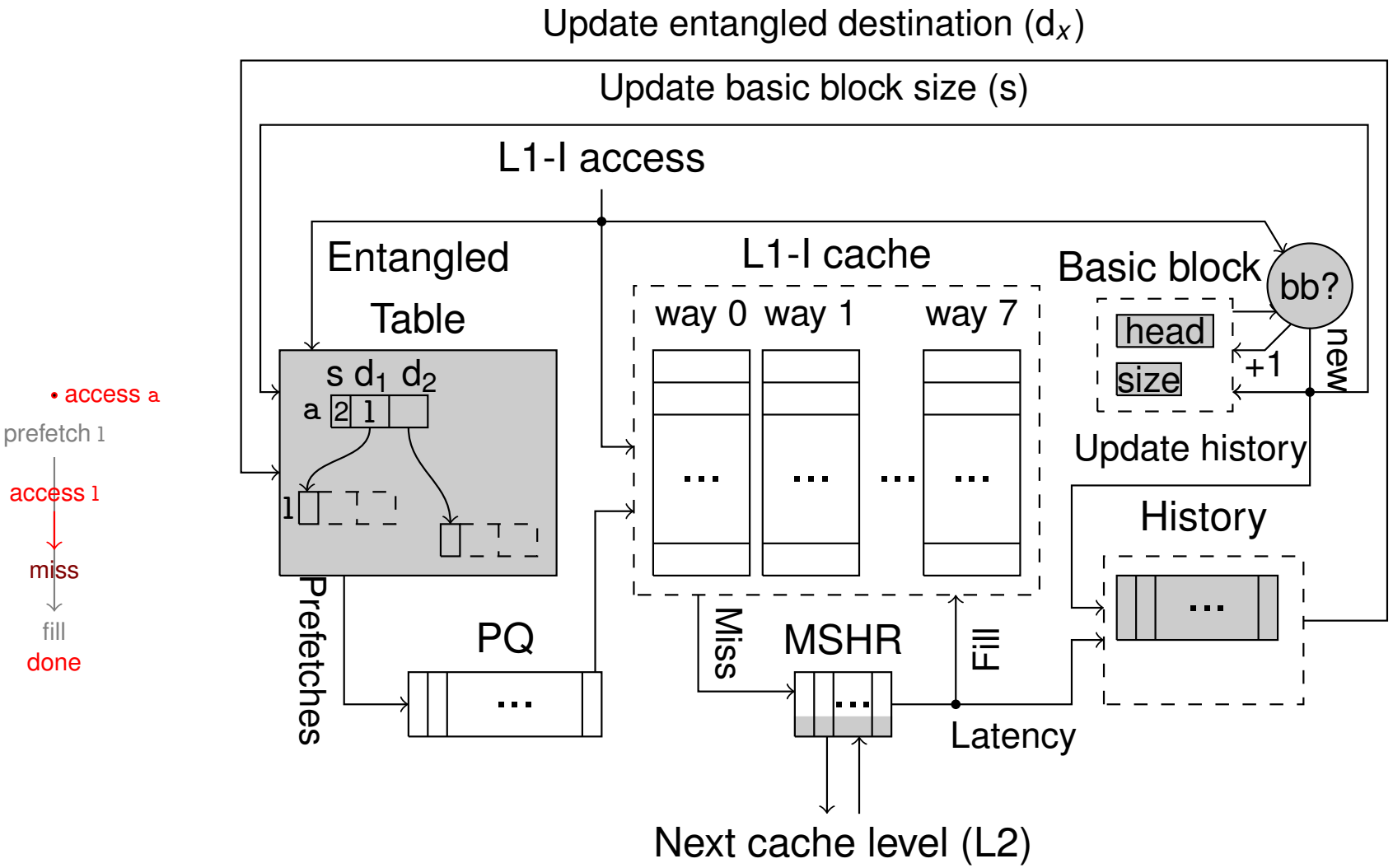
Design of the Entangling Prefetcher - Issuing prefetches



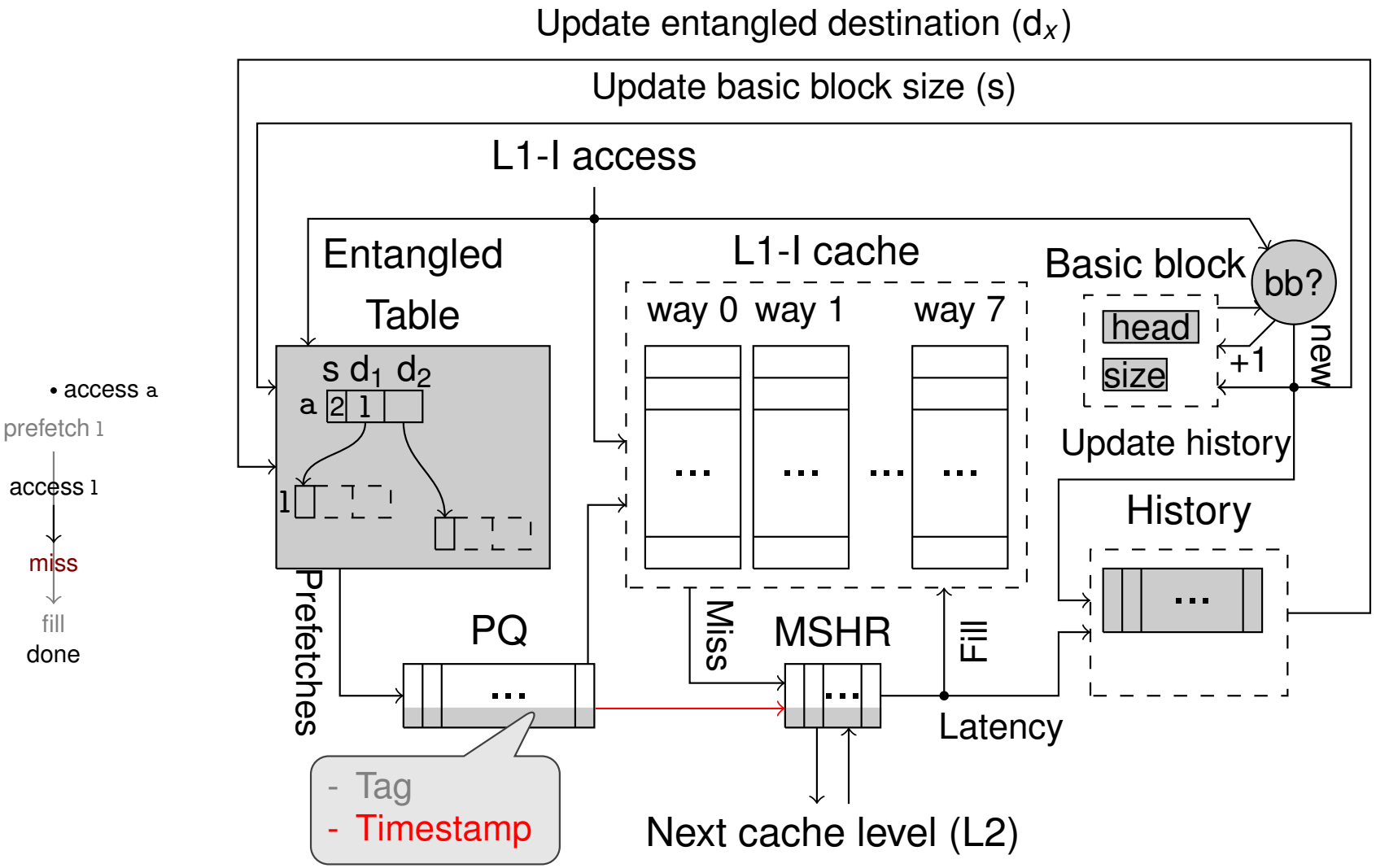
Design of the Entangling Prefetcher - Issuing prefetches



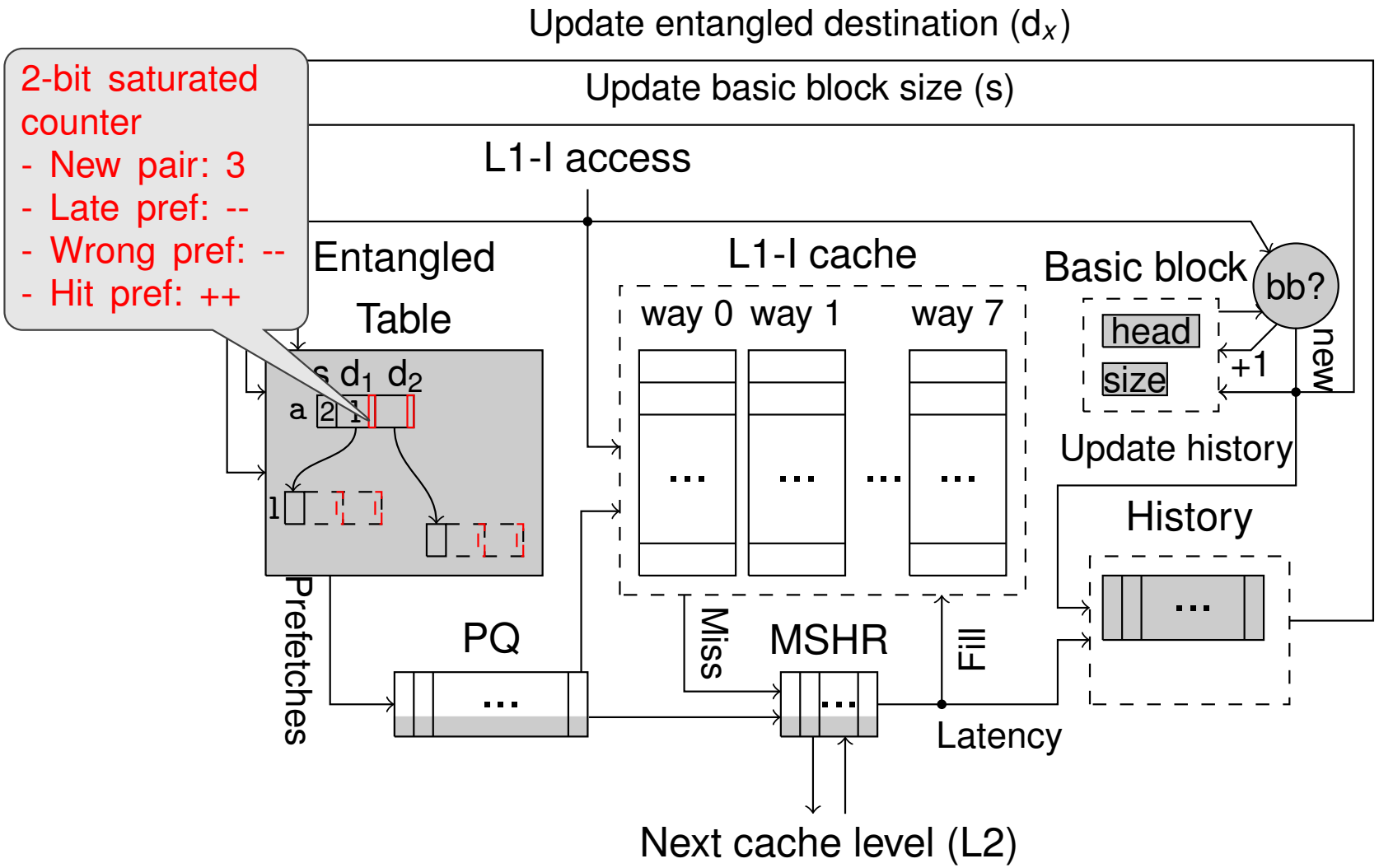
Design of the Entangling Prefetcher - Fixing Late Prefetches



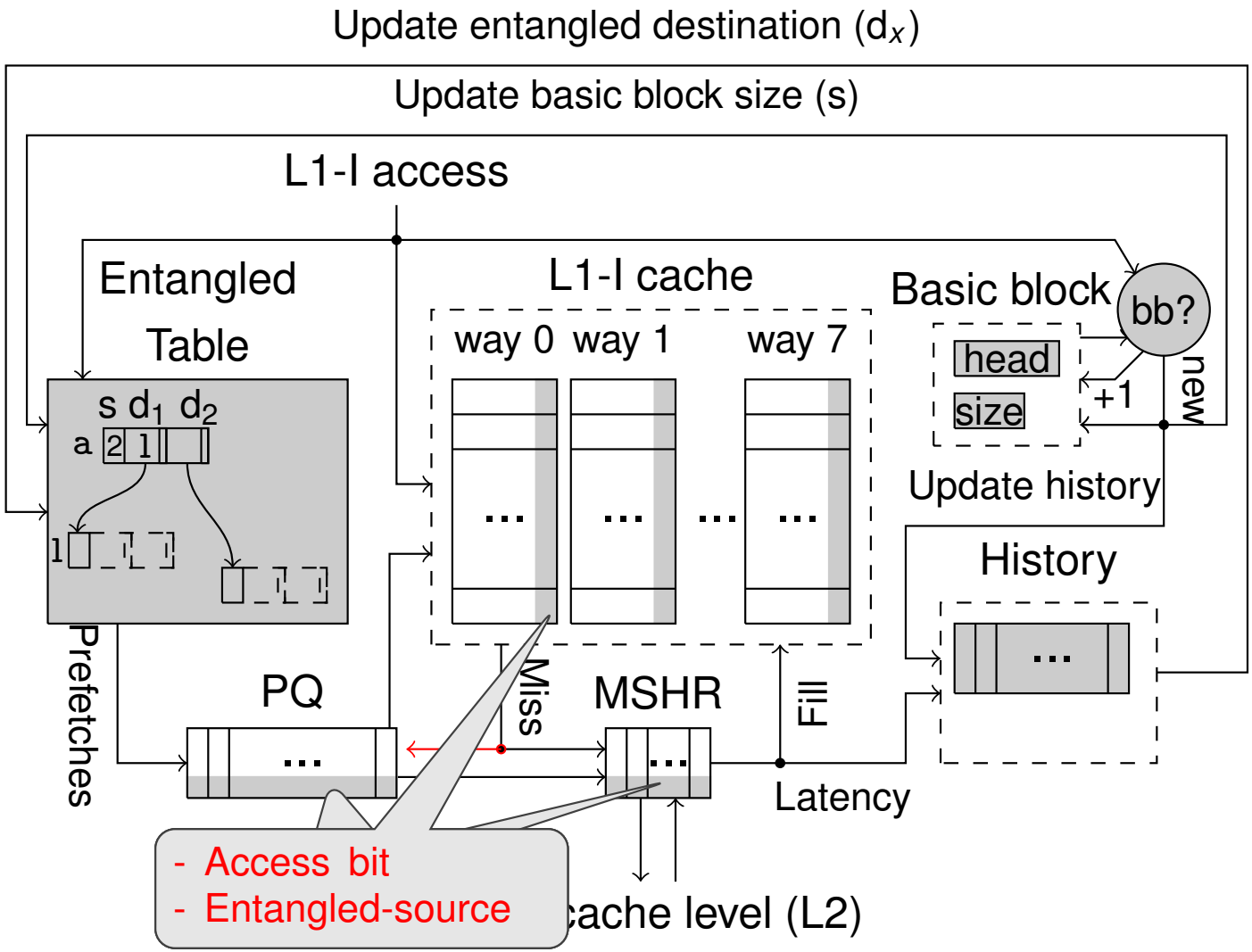
Design of the Entangling Prefetcher - Fixing late prefetches



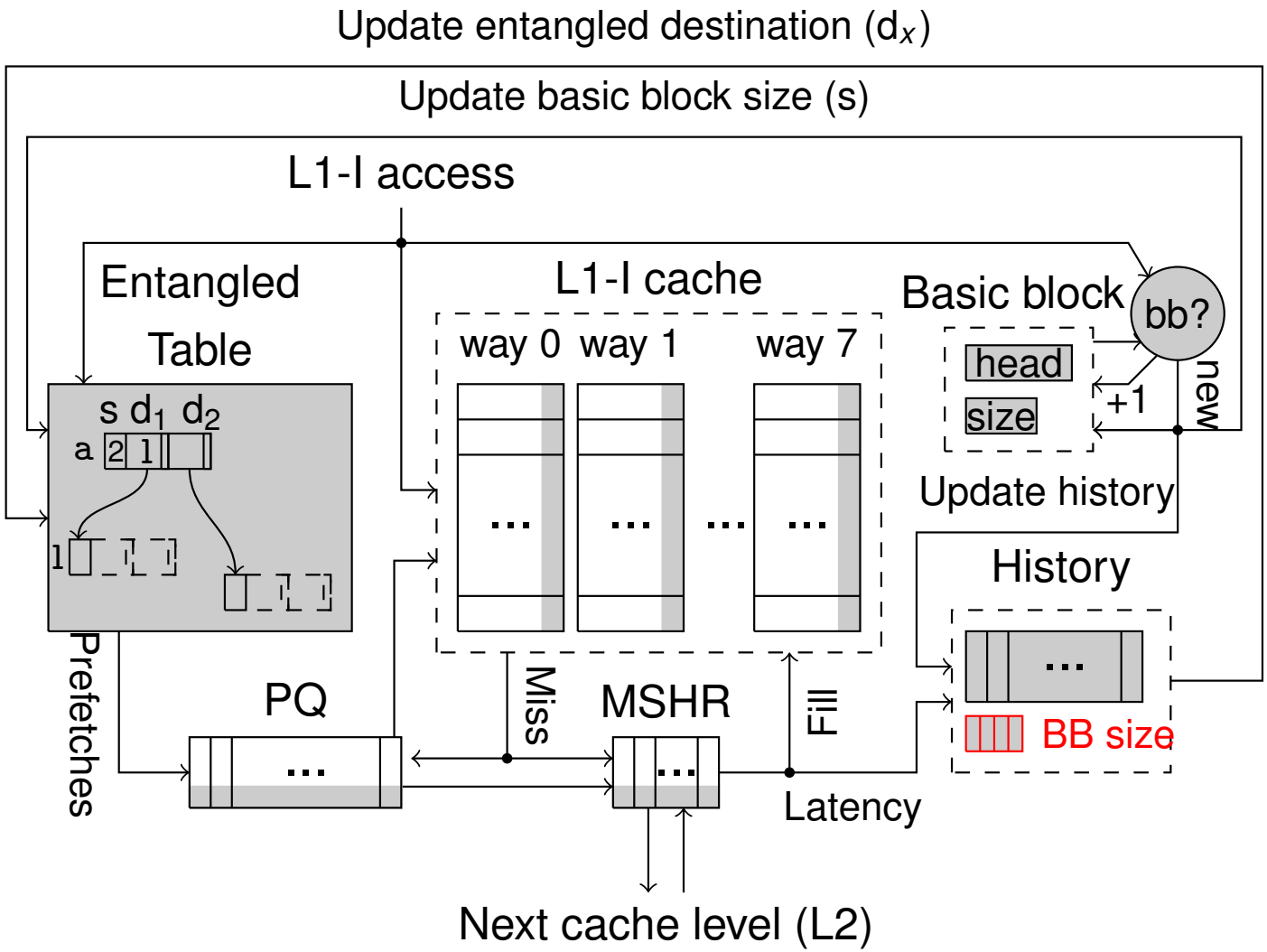
Design of the Entangling Prefetcher - Confidence for entangled pairs



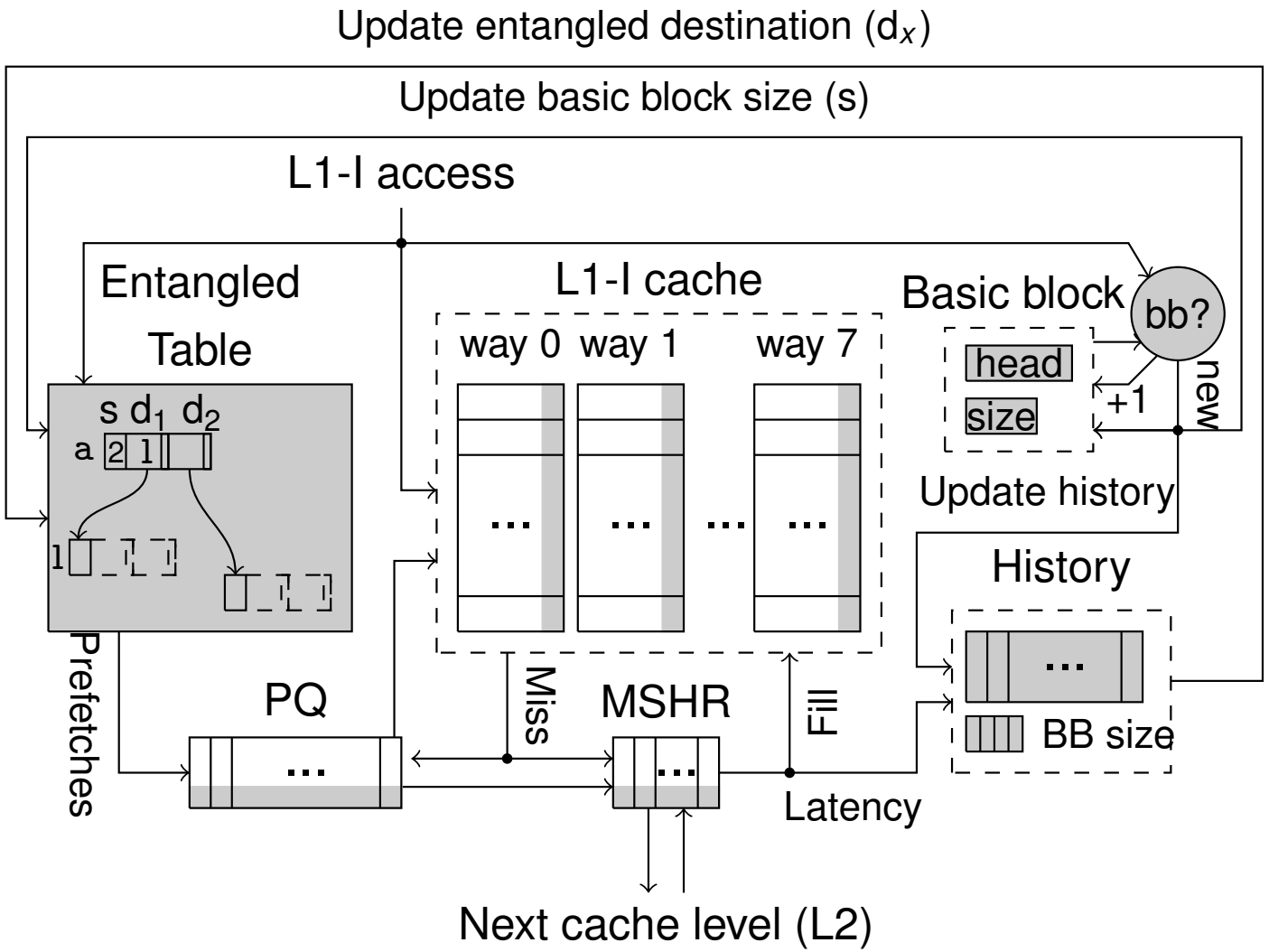
Design of the Entangling Prefetcher - Confidence for entangled pairs



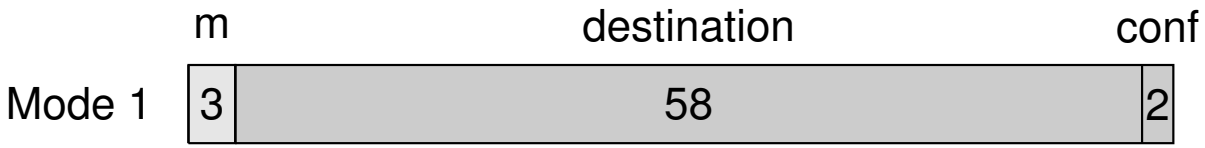
Design of the Entangling Prefetcher - Merging basic blocks



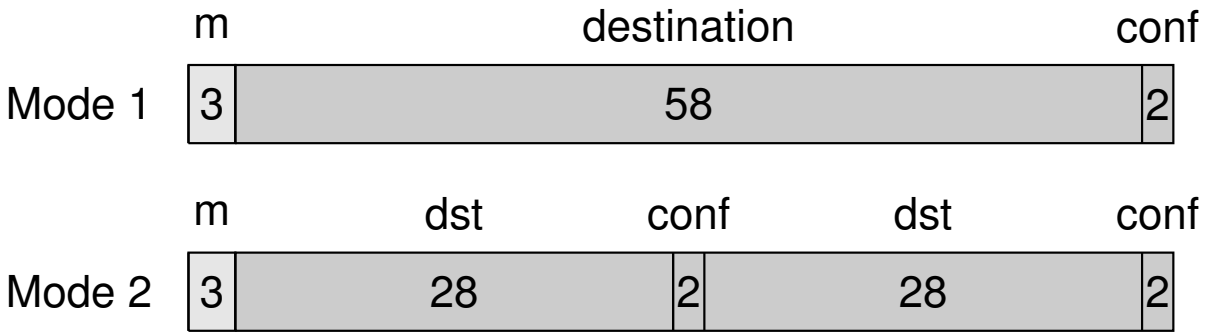
Design of the Entangling Prefetcher



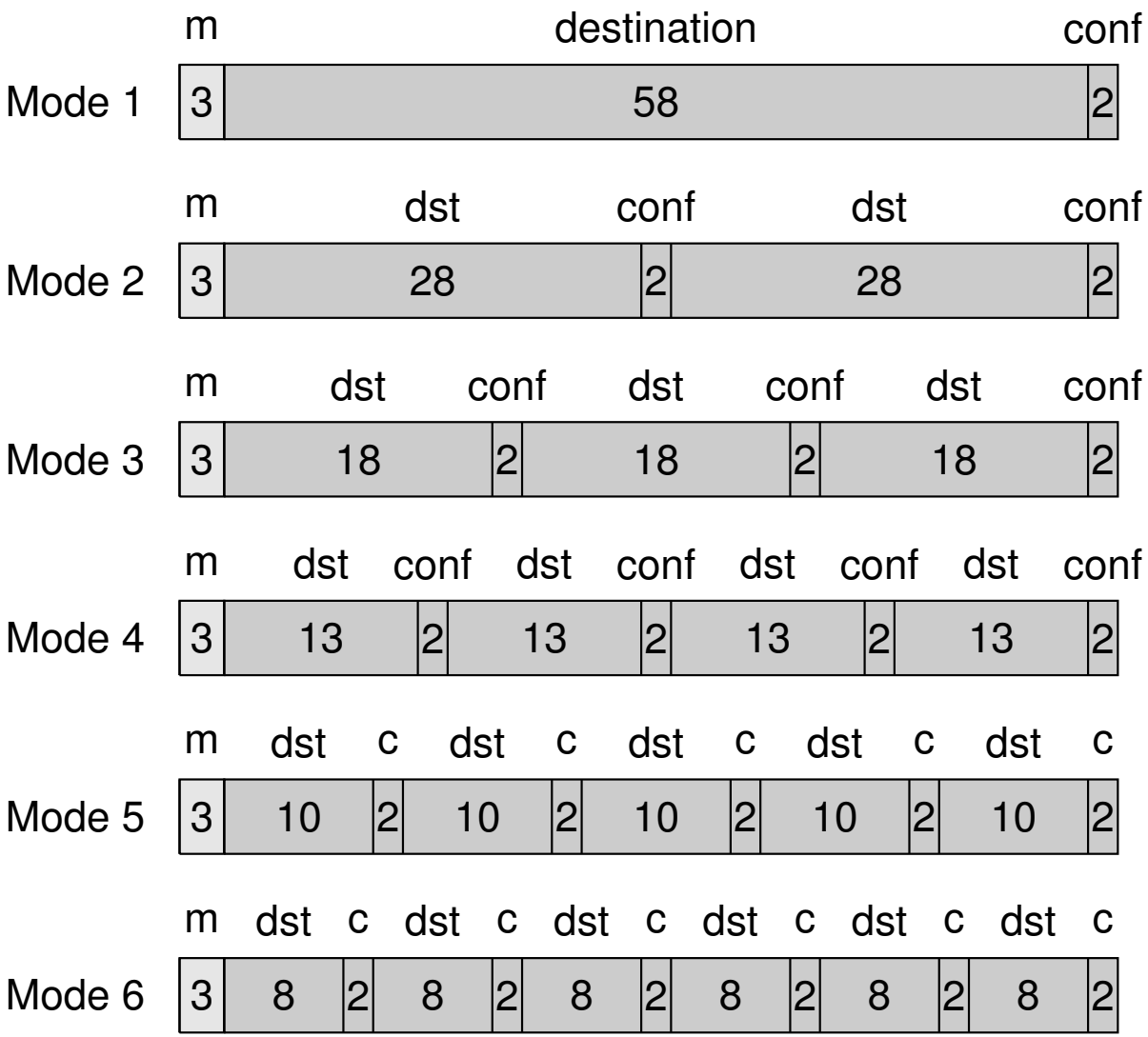
COMPRESSING DESTINATIONS



COMPRESSING DESTINATIONS



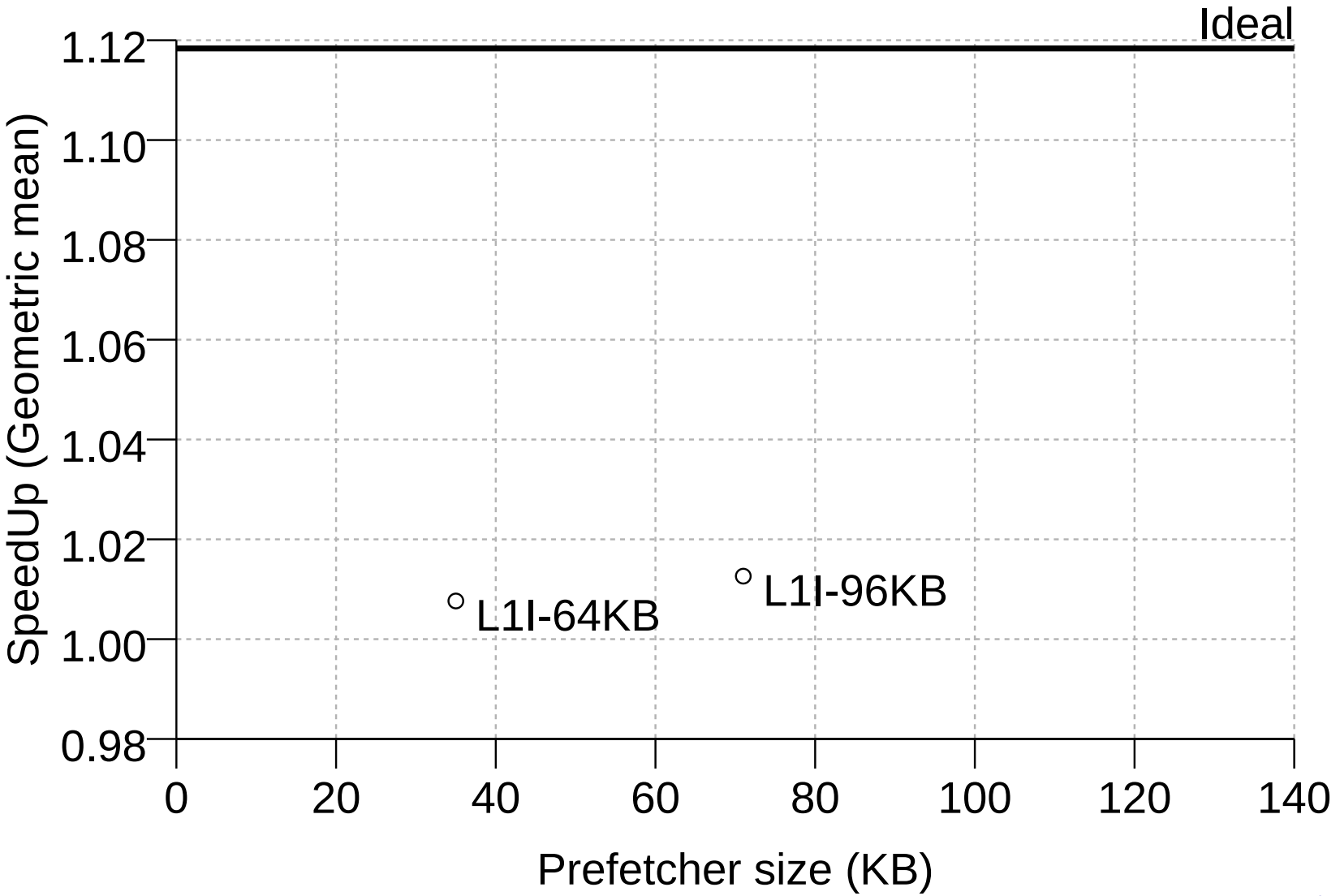
COMPRESSING DESTINATIONS



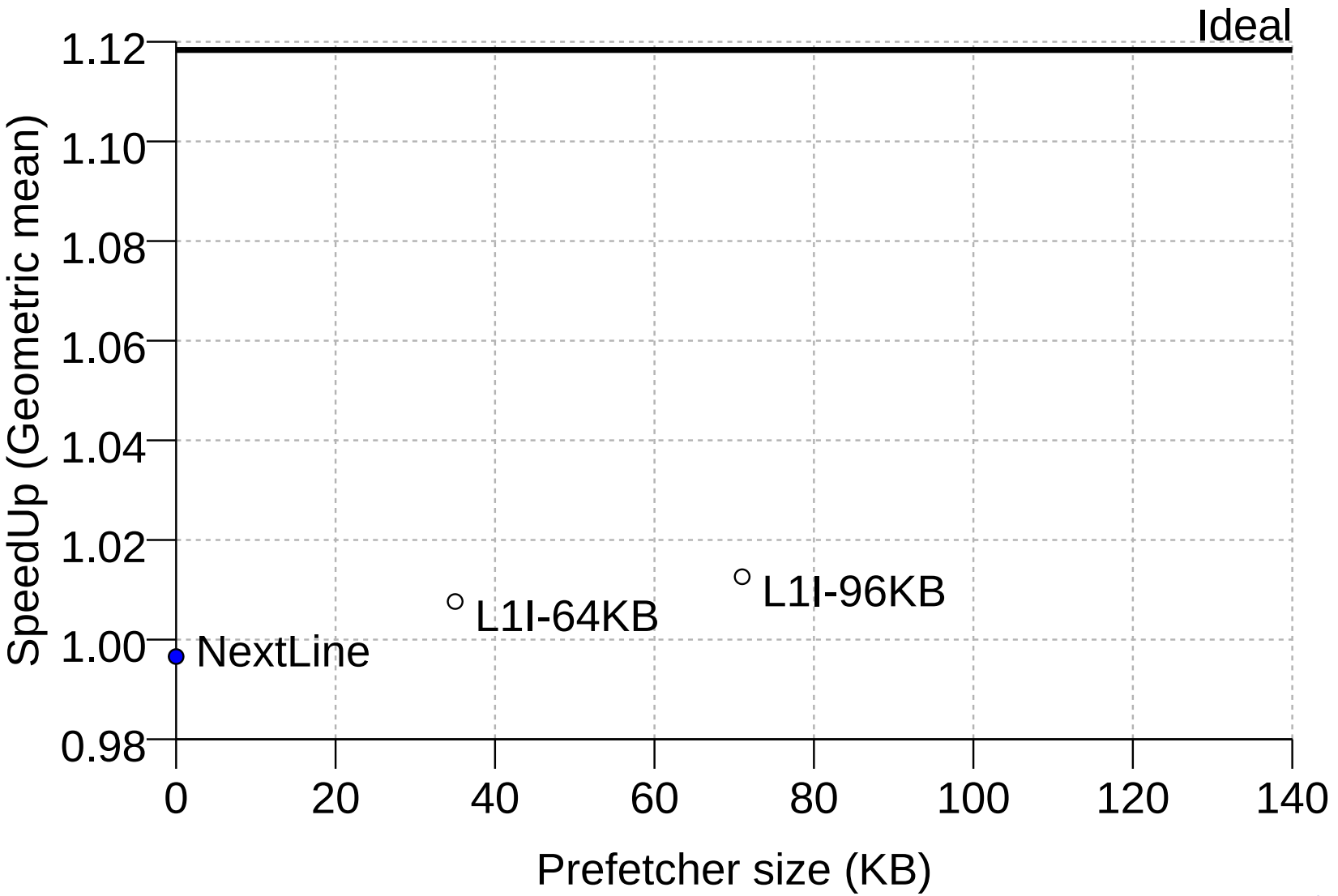
METHODOLOGY

- ChampSim develop branch (nov 2020)
- Baseline:
  - Sunny Cove-like system
  - Decoupled front-end (64-entry fetch queue)
  - 32KB L1I
- ENTANGLED:
  - *History buffer*: 16 entries
  - *Entangled table*: 2K, 4K and 8K entries
- Applications
  - 959 traces from the Championship Value Prediction (provided by Qualcomm)
  - Cloud Suite
- Analysis both for virtual and physical prefetching

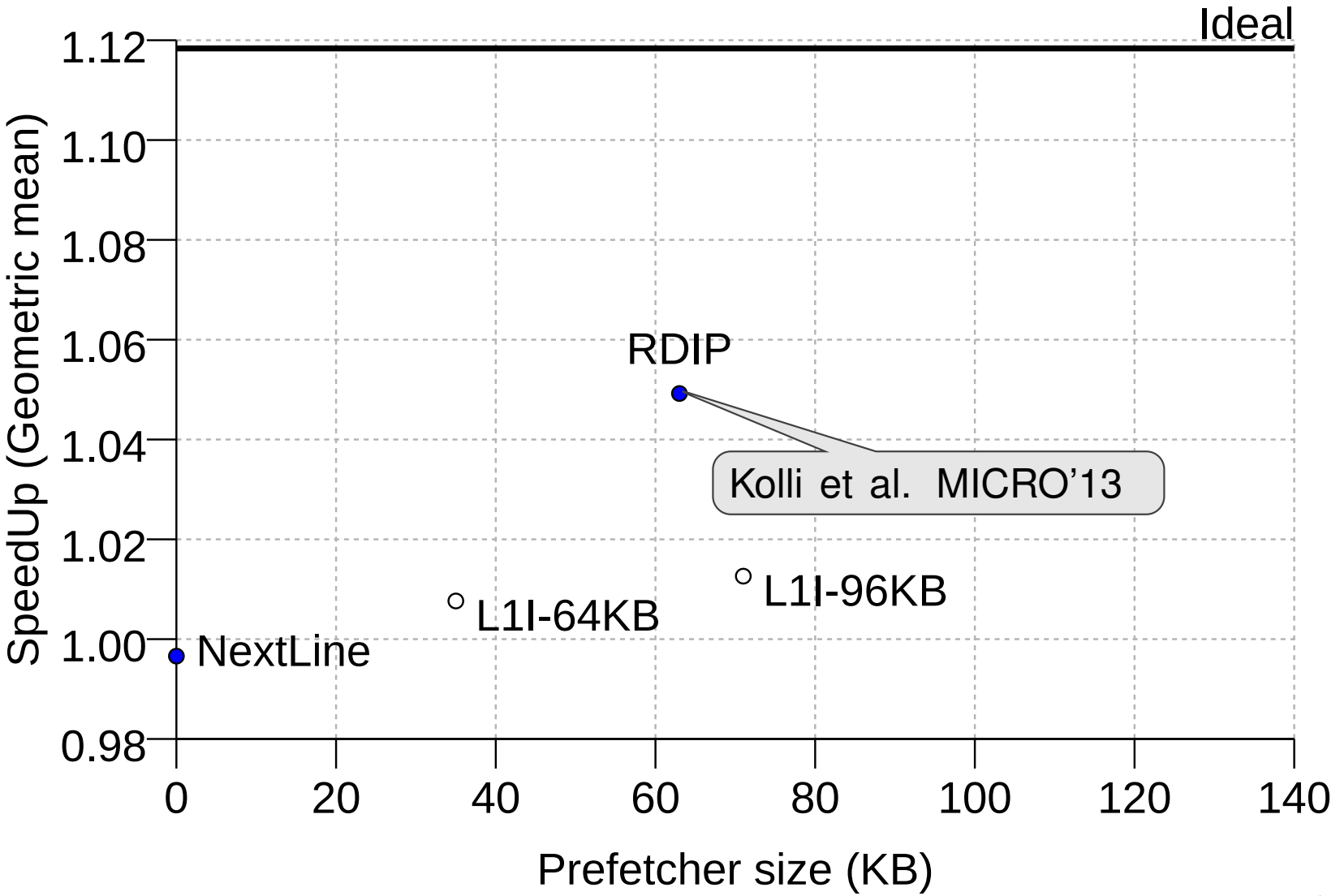
RESULTS: IPC VS MEMORY OVERHEAD



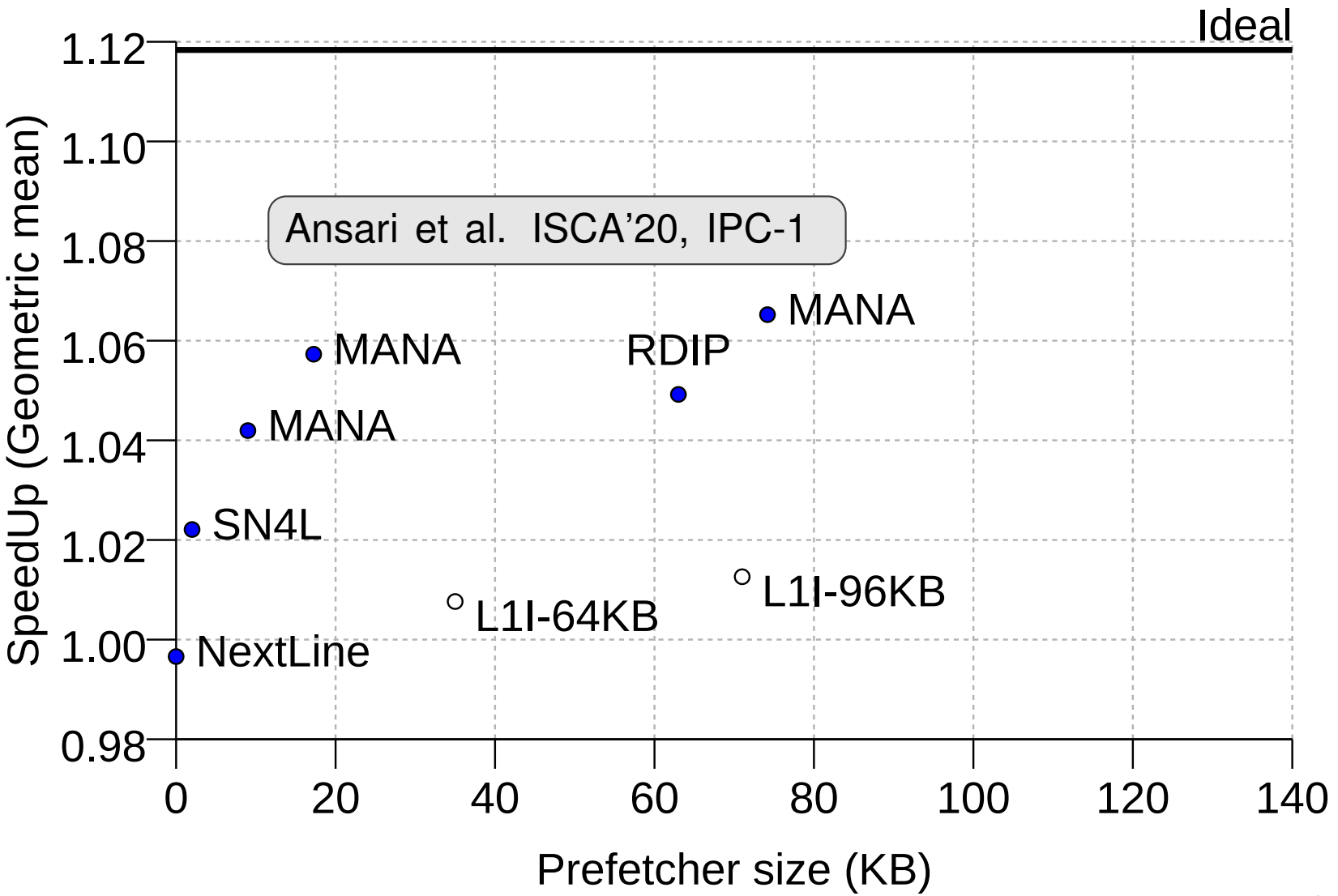
RESULTS: IPC VS MEMORY OVERHEAD



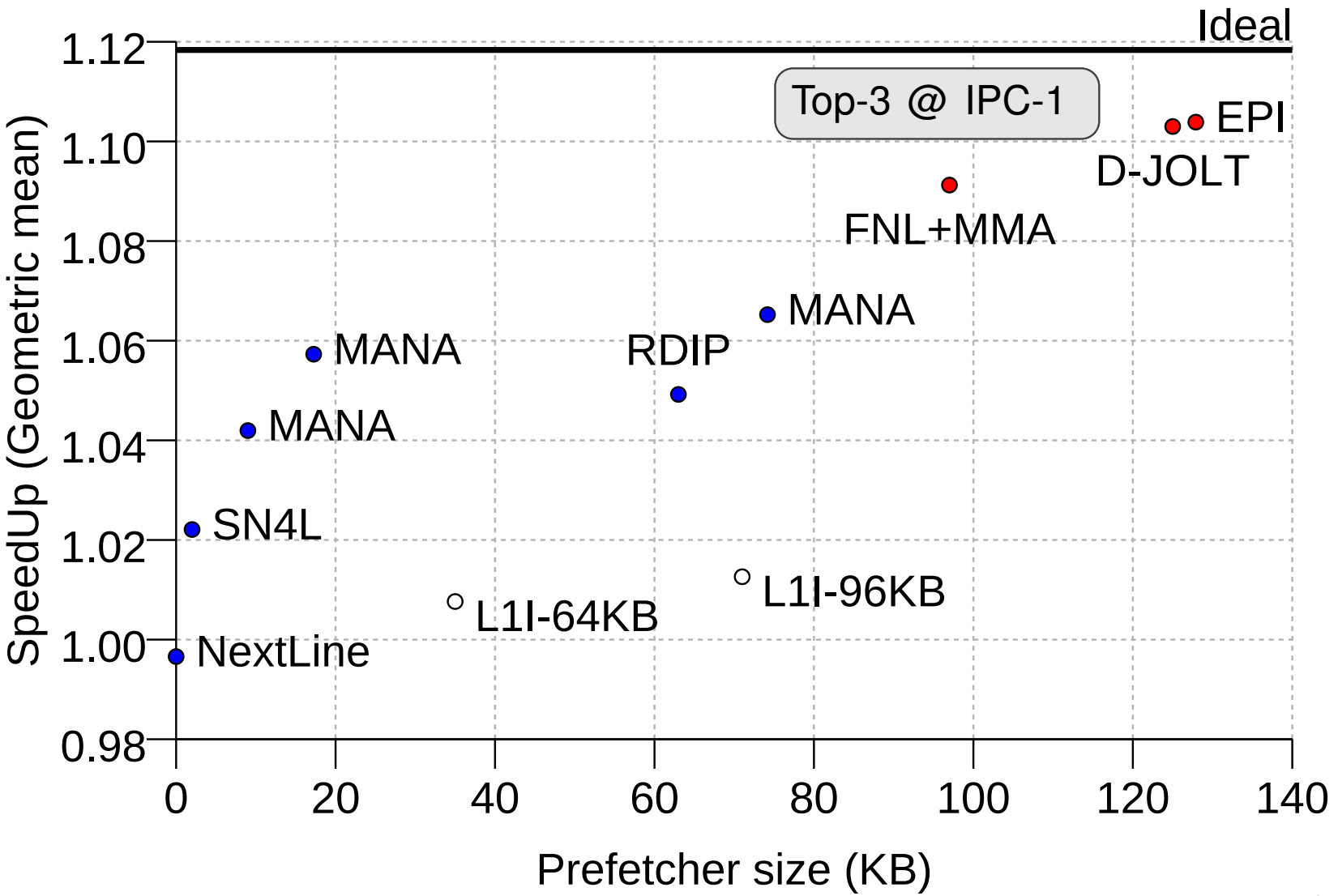
RESULTS: IPC VS MEMORY OVERHEAD



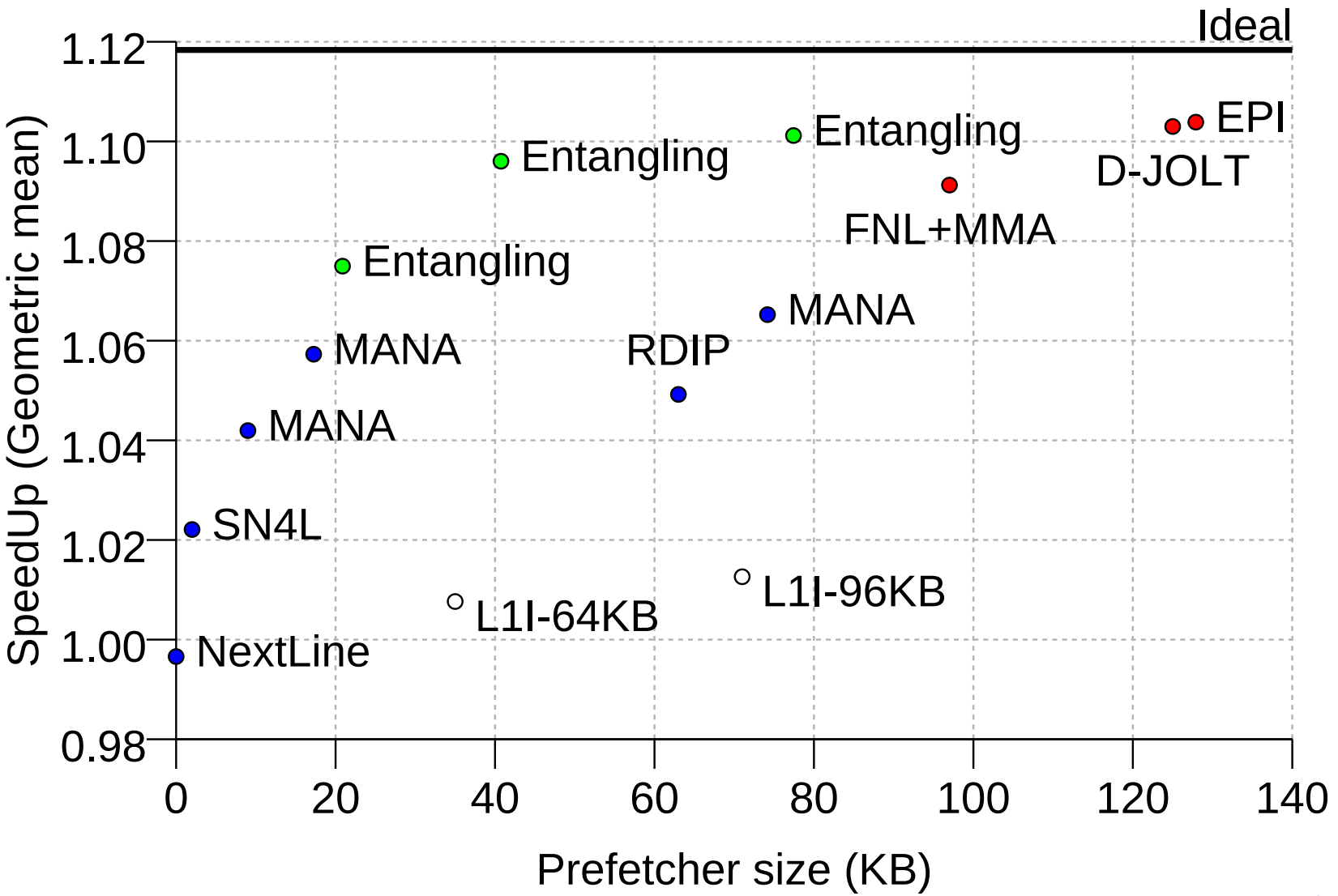
RESULTS: IPC VS MEMORY OVERHEAD



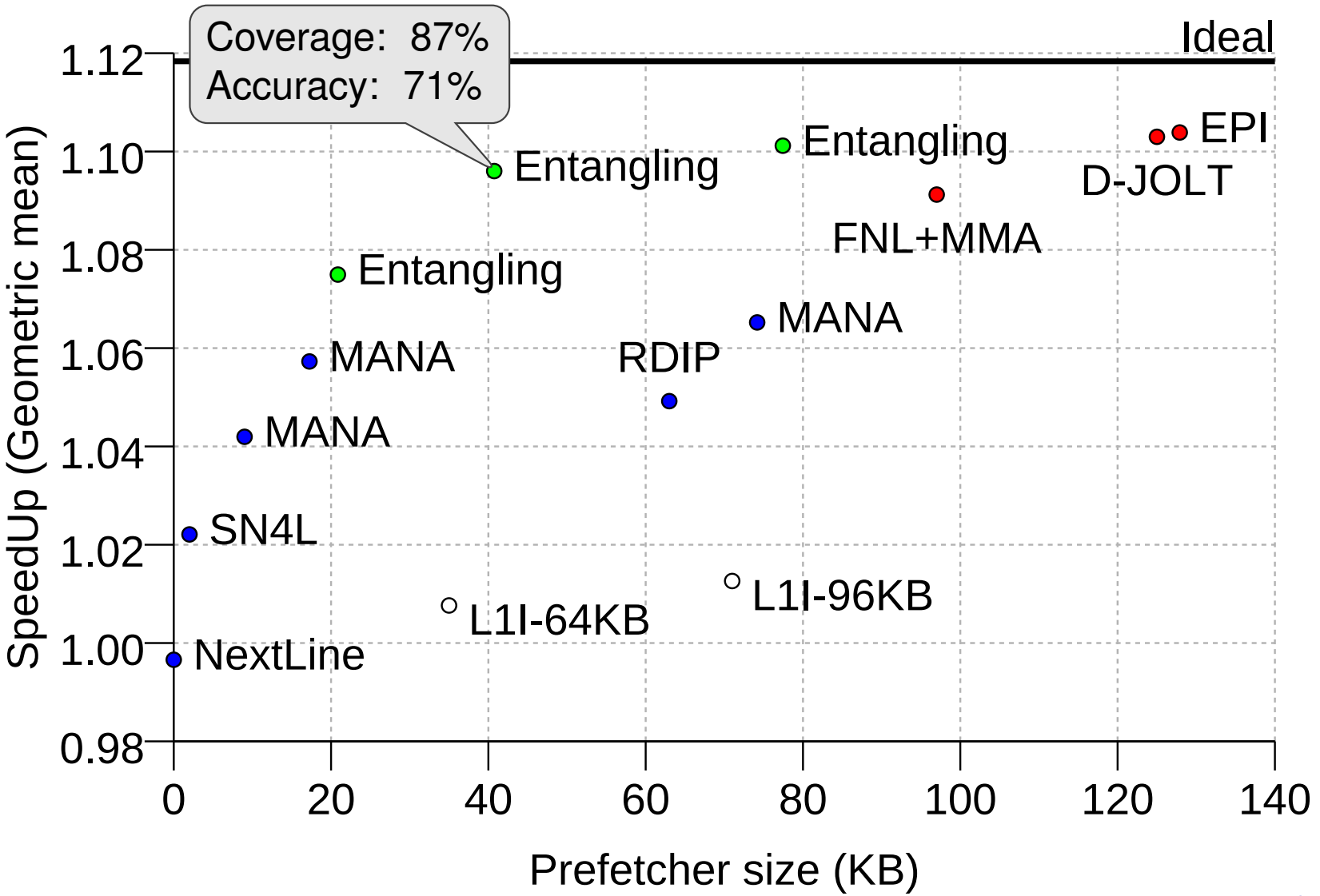
RESULTS: IPC VS MEMORY OVERHEAD



RESULTS: IPC VS MEMORY OVERHEAD



RESULTS: IPC VS MEMORY OVERHEAD



CONCLUDING REMARKS

- Timeliness as a key property
- Entangles heads of basic blocks to trigger timely prefetches
- Near ideal performance with just 40KB

ENTANGLING PREFETCHERS

Alberto Ros

aros@ditec.um.es

Thank you!

ECHO, ERC Consolidator Grant (No 819134)