Not Quite My Tempo[1]

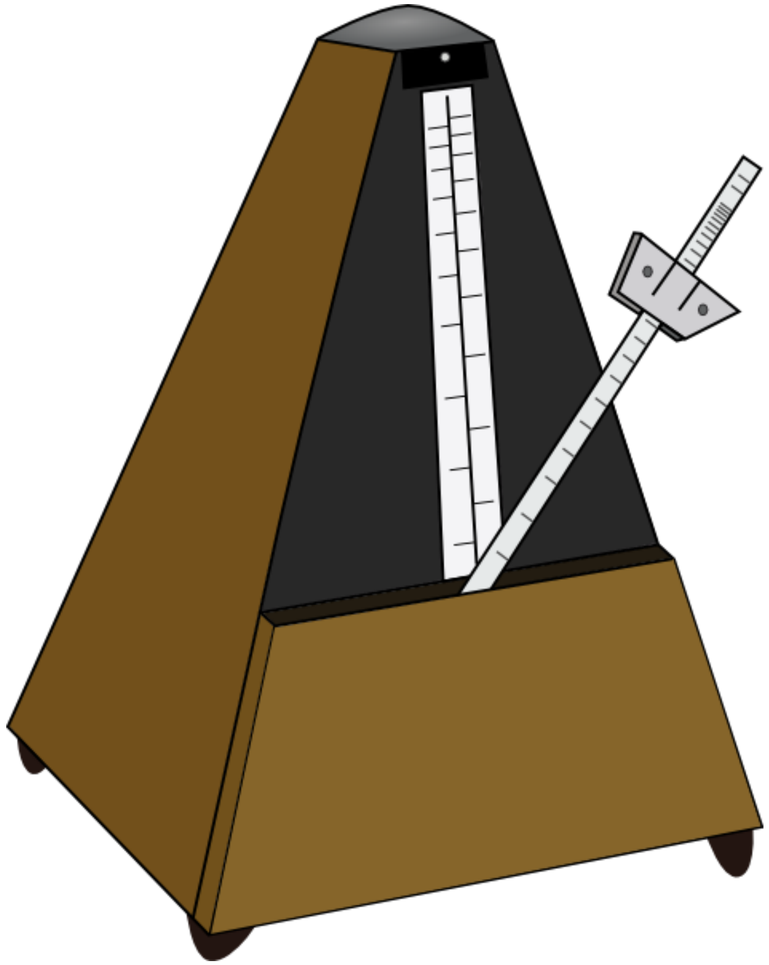
Matching Prefetches to Memory Access Times

Mark Sutherland
Ajaykumar Kannan
Natalie Enright Jerger

{suther68,kannanaj,enright}@ece.utoronto.ca

The Edward S. Rogers Sr. Department
of Electrical & Computer Engineering
UNIVERSITY OF TORONTO

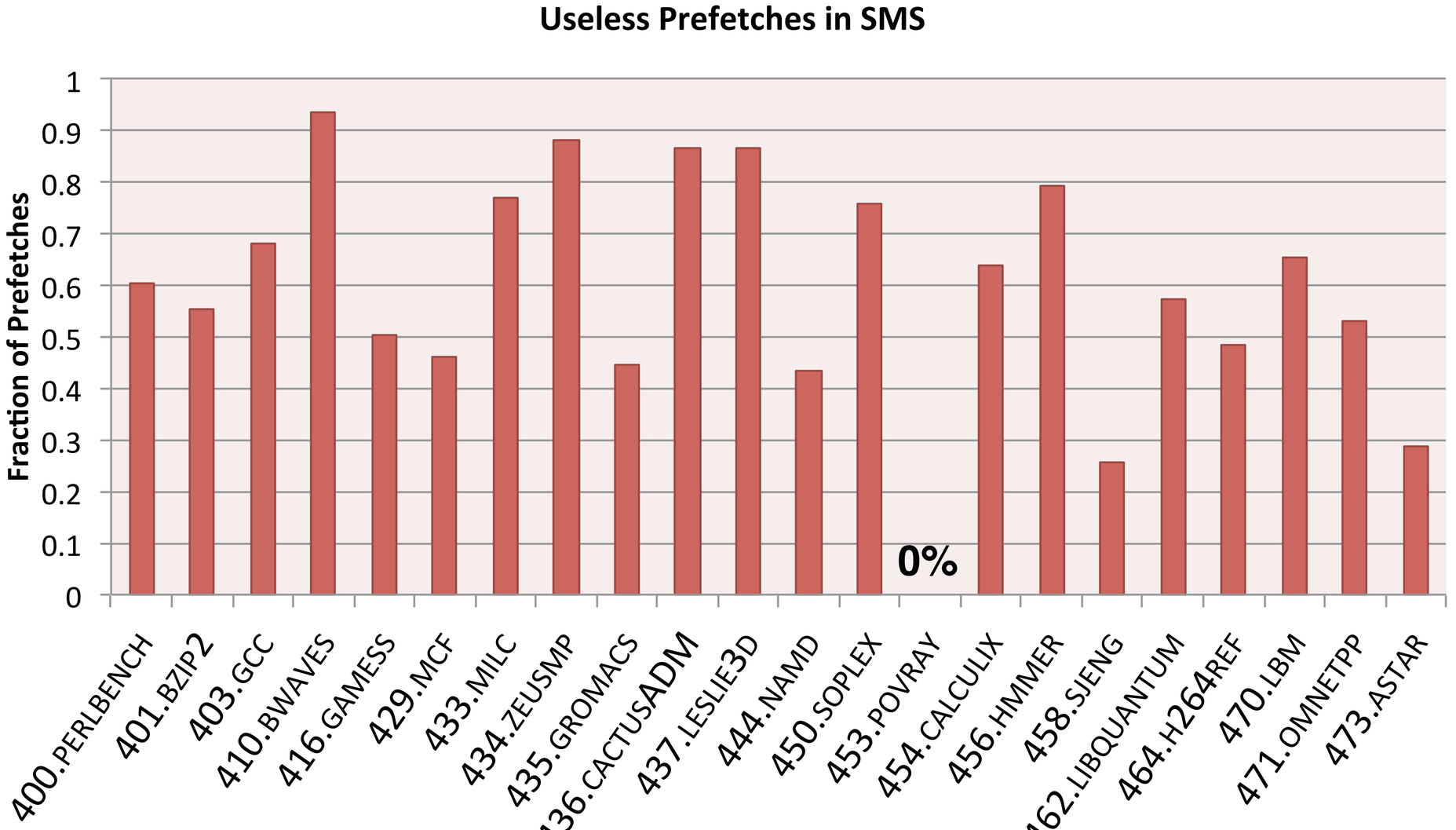
1. This work was inspired by Chazelle, Simmons, and Teller's foundational work in this area: *Whiplash*.




Map
Next-Line
History
Access
Stride
Feedback
GHB
Markov
Pattern
SMS
Stream
Local

Goal: Add temporal information to an SMS prefetcher.

Motivation

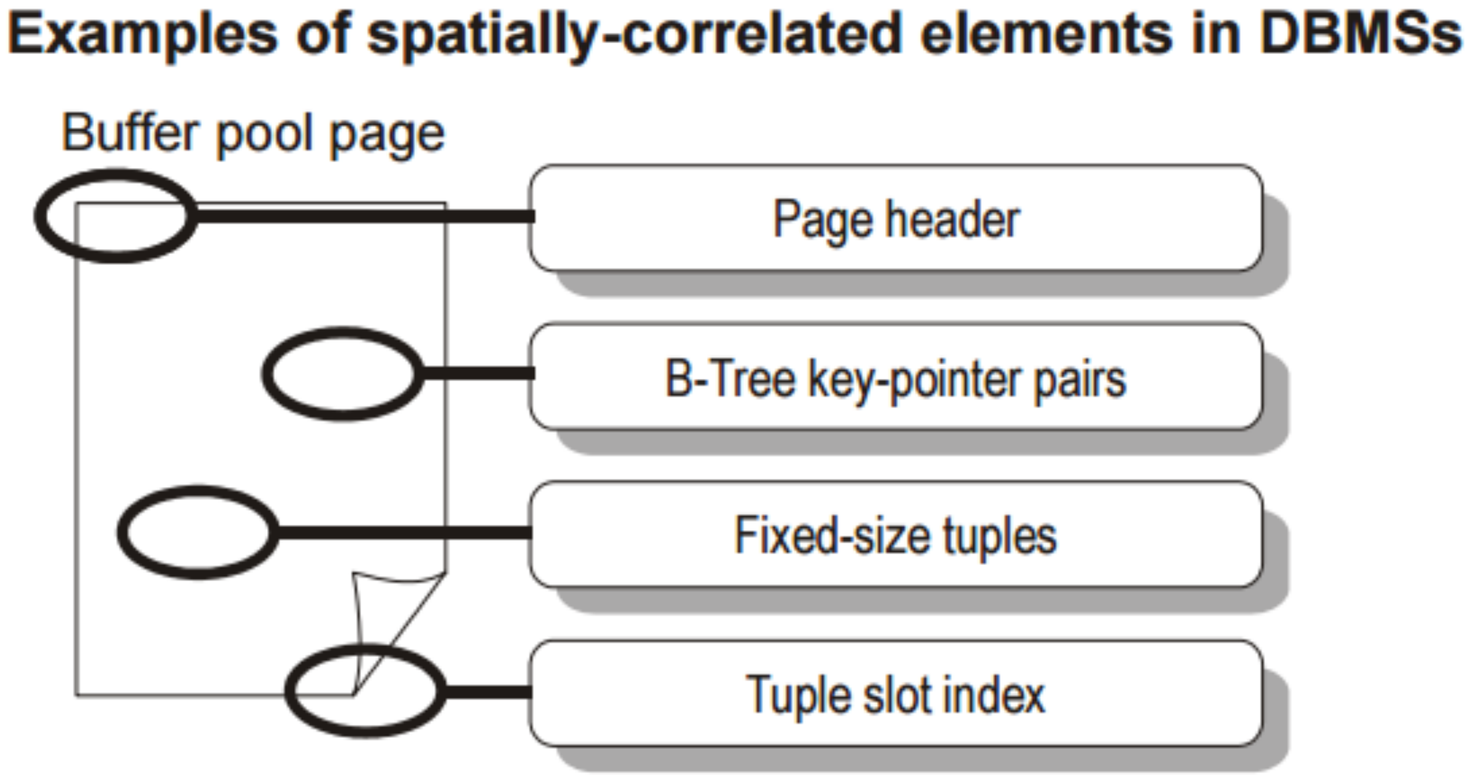

Useless Prefetches in SMS
Fraction of Prefetches
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0%
400.PERLBENCH
401.BZIP2
403.GCC
410.BWAVES
416.GAMESS
429.MCF
433.MILC
434.ZEUSMP
435.GROMACS
436.CACTUSADM
437.LESLIE3D
444.NAMD
450.SOPLEX
453.POVRAY
454.CALCULIX
456.HMMER
458.SJENG
462.LIBQUANTUM
464.H264REF
470.LBM
471.OMNETPP
473.ASTAR

Our Contributions

1. Propose a framework for timely prefetching in SMS, which utilizes *temporal* information to filter prefetches in the Pattern History Table.

2. Demonstrate practical improvements over SMS with a 32kB storage budget.

SMS Prefetcher Background



Source: *S. Somogyi, T. F.Wenisch, A. Ailamaki, B. Falsafi, and A. Moshovos. Spatial Memory Streaming. SIGARCH Comput. Archit. News, 2006.*

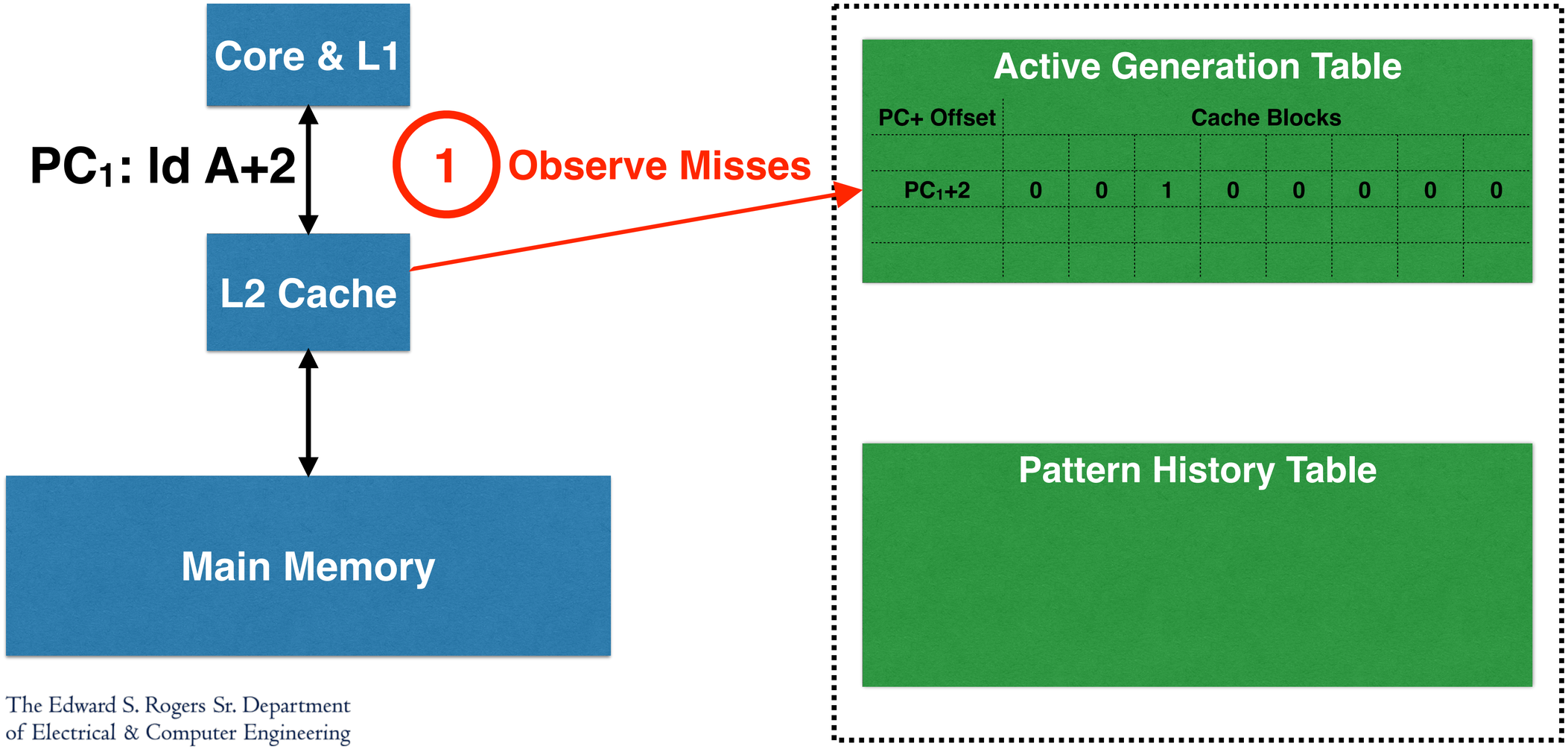
SMS Prefetcher Background

Representation of a Memory Region

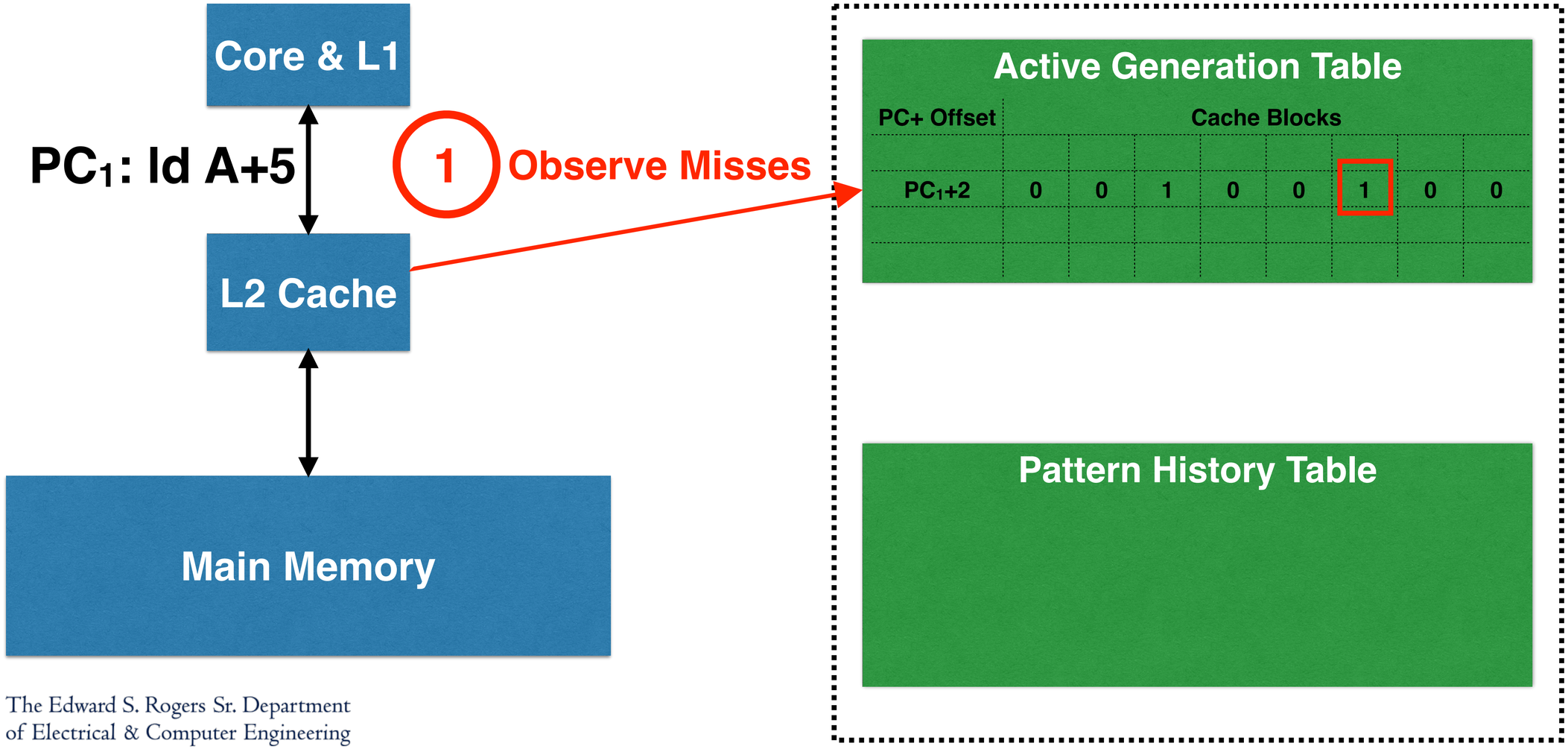
Page Bits		Accessed Cache Blocks = 1															
100110100	…	1	1	1	1	0	0	0	1	1	0	1	0	1	0	1	1
110100101	…	1	0	1	0	0	1	0	1	0	0	1	0	0	1	1	0
100000010	…	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0

- Memory regions (most often the size of a physical page) are bit vectors, where a set bit indicates that the program accessed this block.

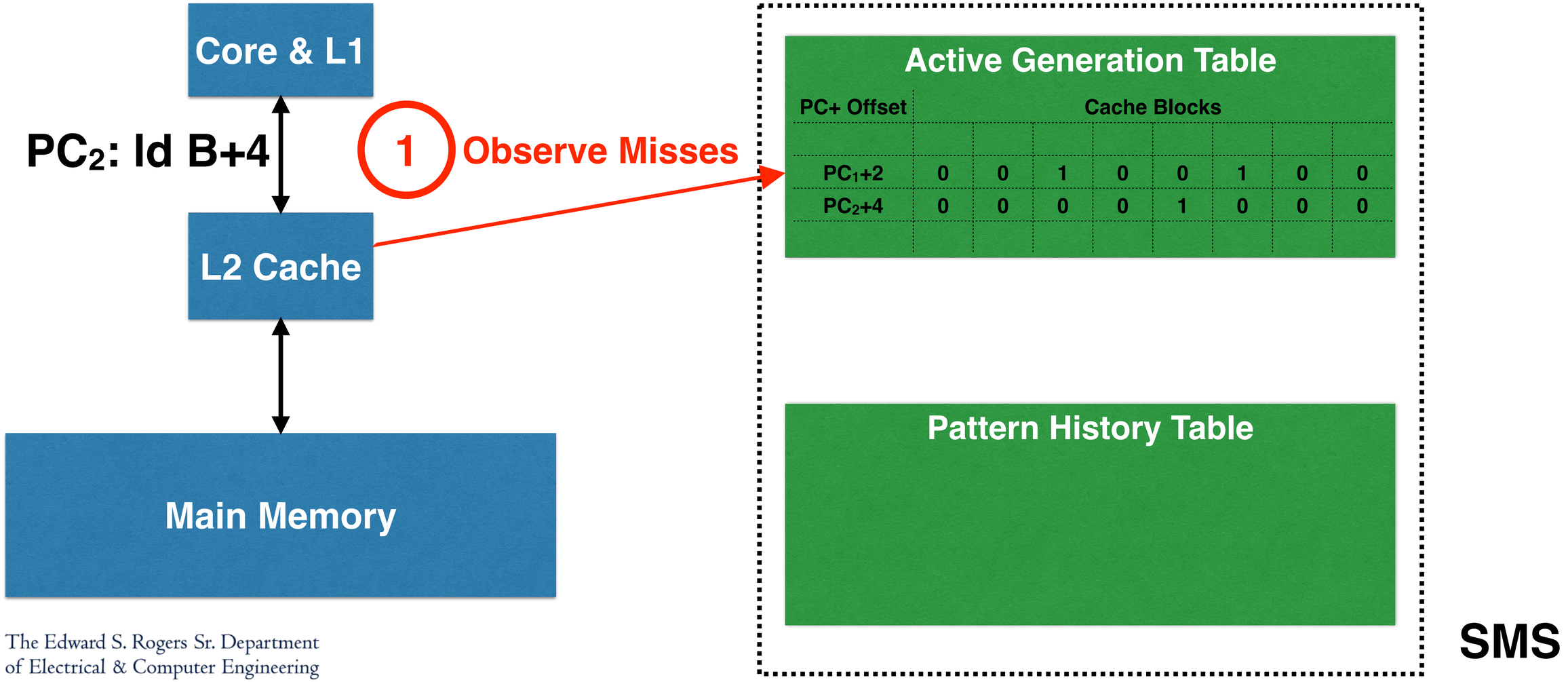
SMS Prefetcher Background



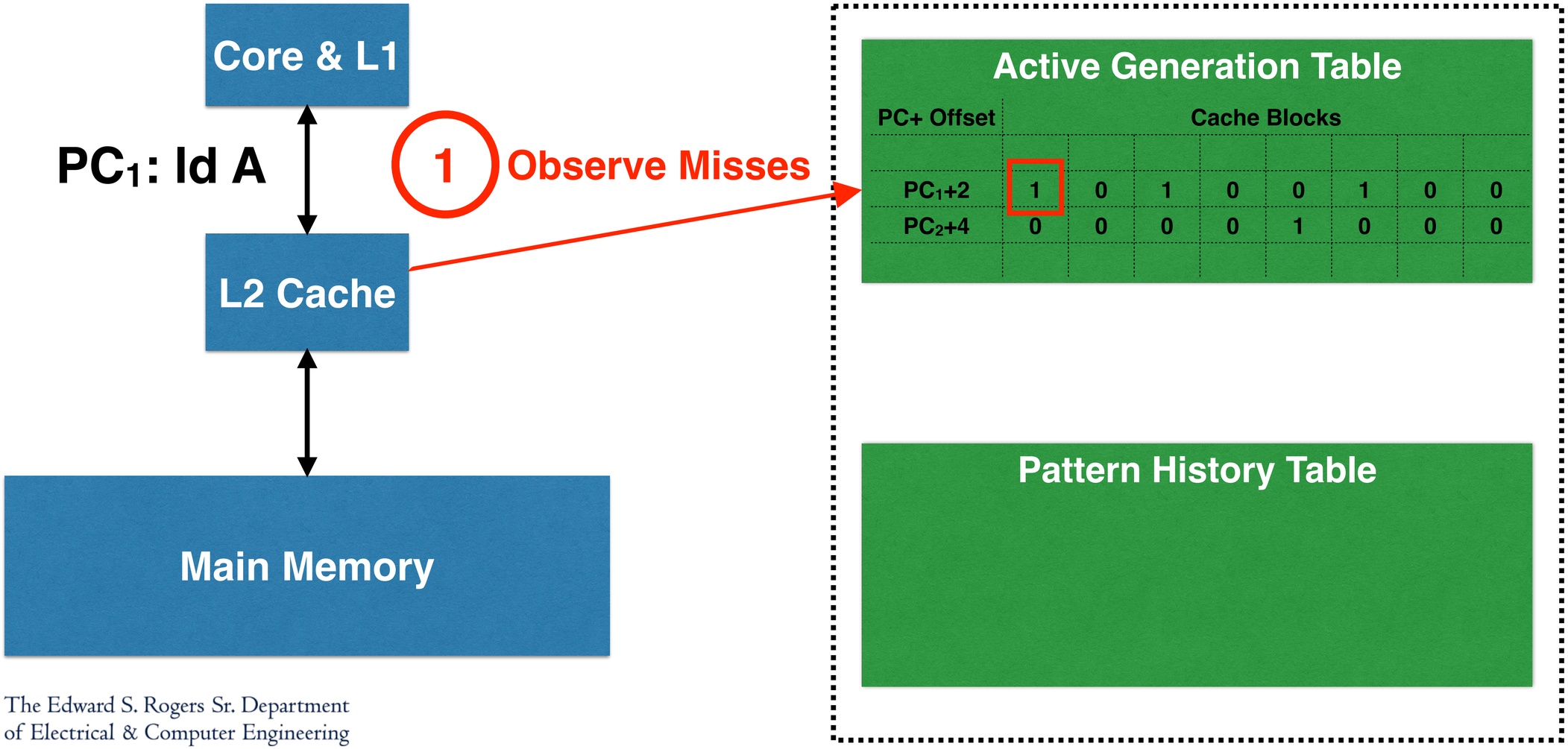
SMS Prefetcher Background



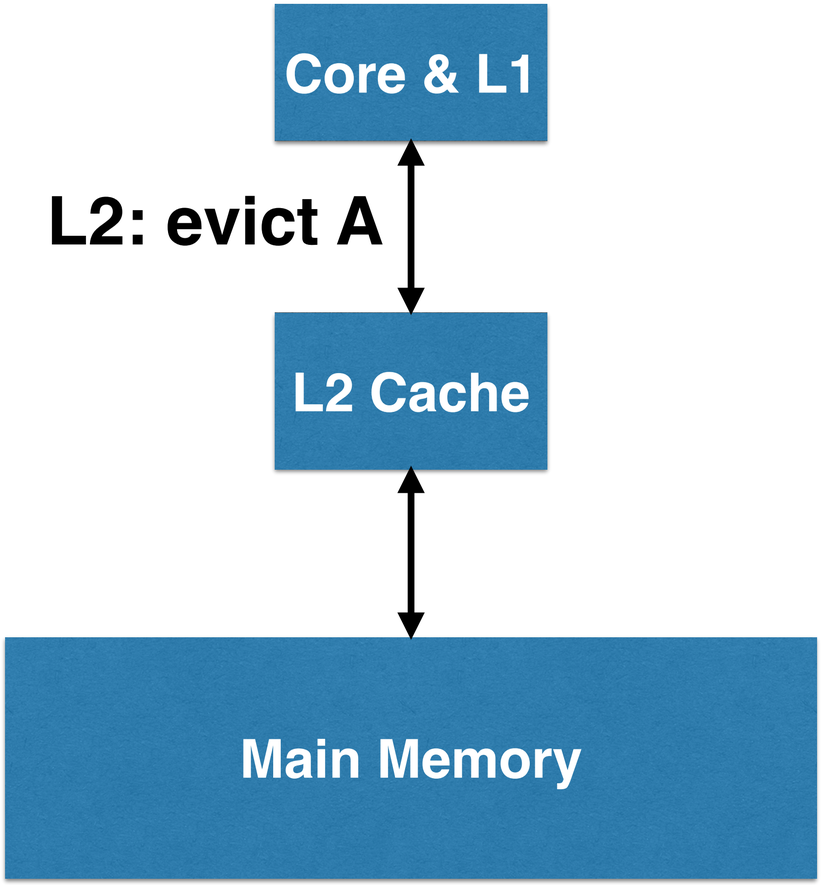
SMS Prefetcher Background

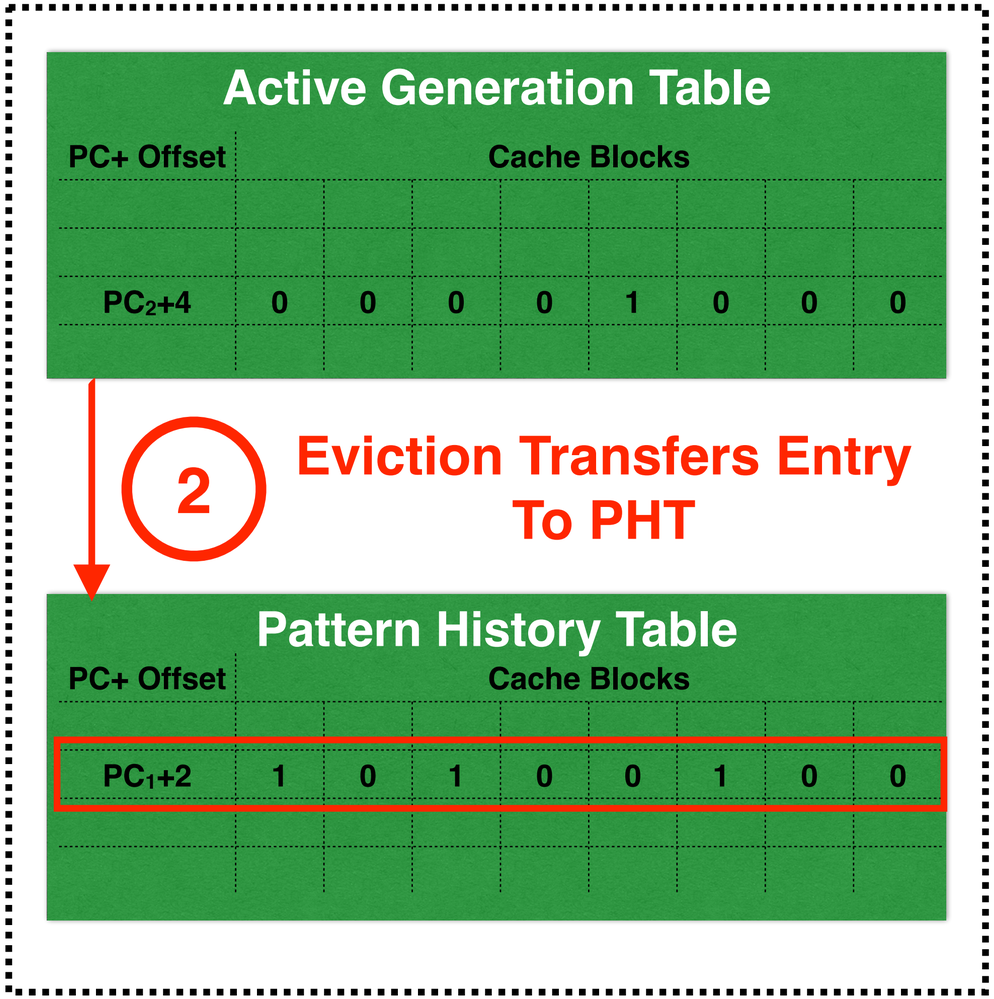


SMS Prefetcher Background



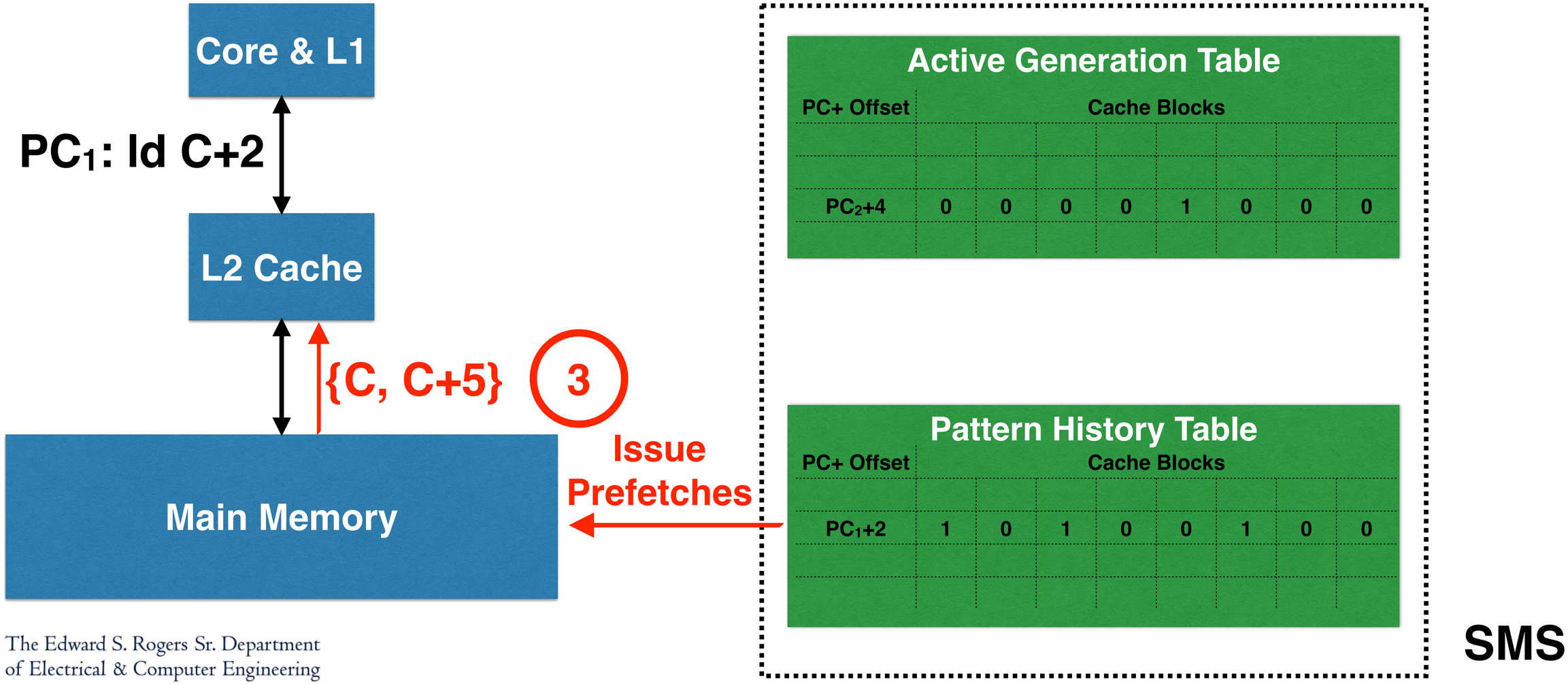
SMS Prefetcher Background



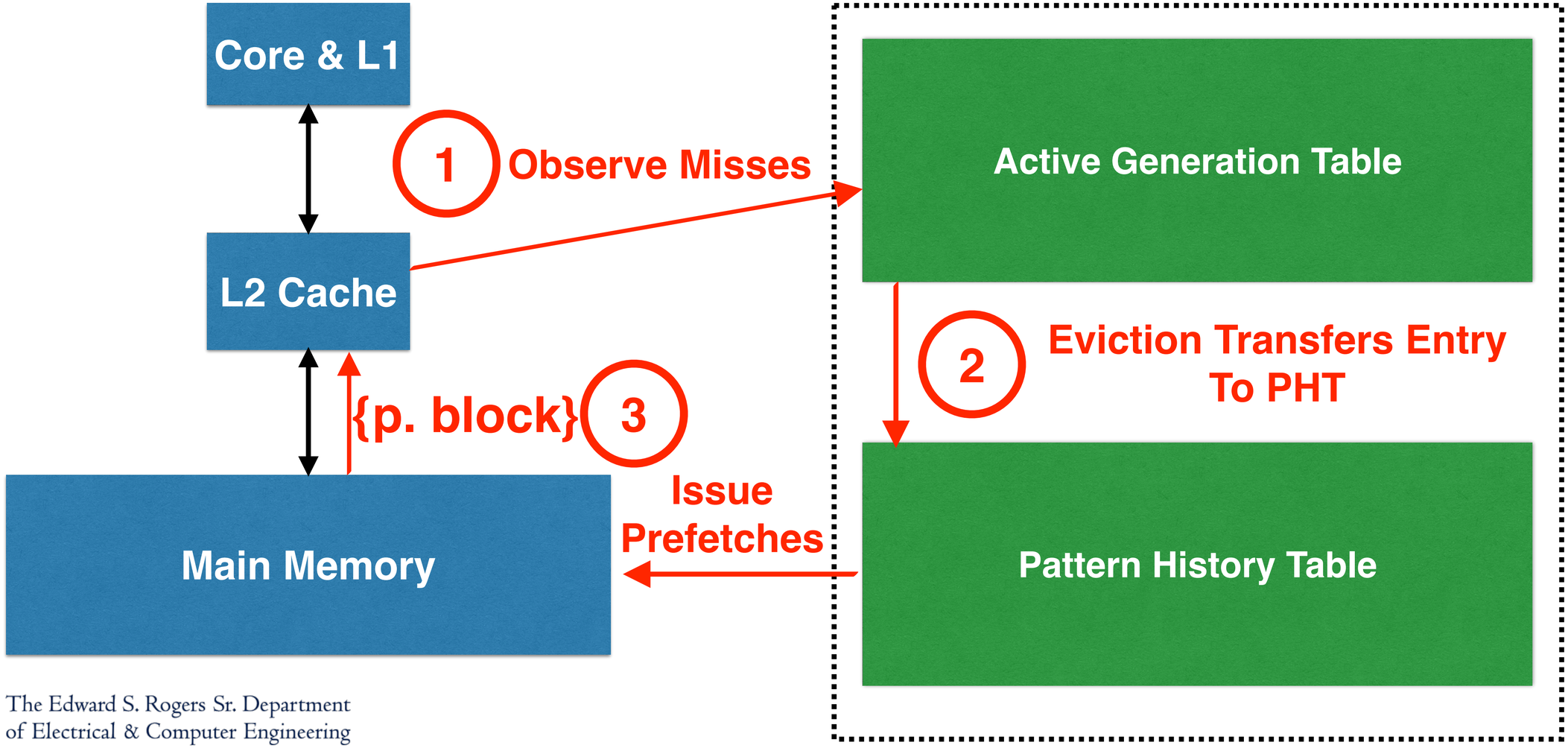


SMS

SMS Prefetcher Background



SMS Prefetcher Background



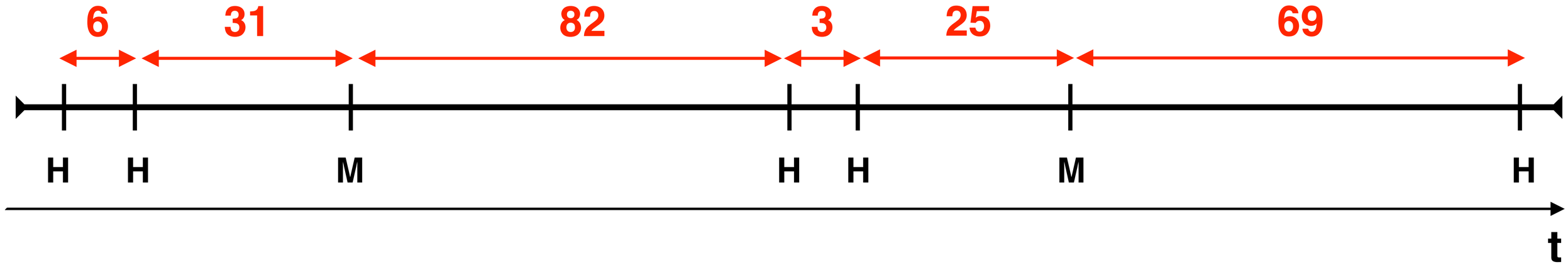
Adding Temporal Information

- **Problem:** In the bit vectors stored by SMS, we do not have any information about the *order* of the accesses.

- **Goal**: A mechanism to distinguish various accesses within a spatial region, preferably one that allows us to prefetch blocks in a more timely fashion.

Adding Temporal Information

- Observe that many of our applications see the same PC generating memory requests with different time deltas between each request.

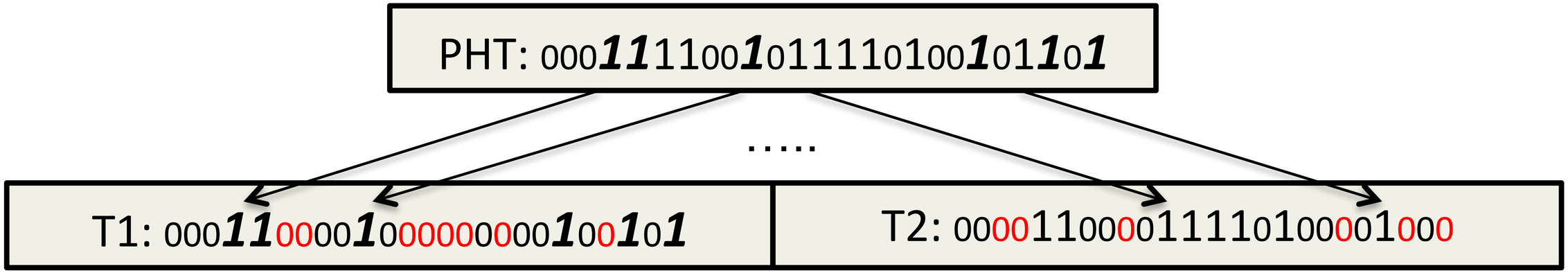


Adding Temporal Information

- Add a global counter to the L2 that ticks on every cache miss.
- Define **access tempo**: Delta between two cache accesses.

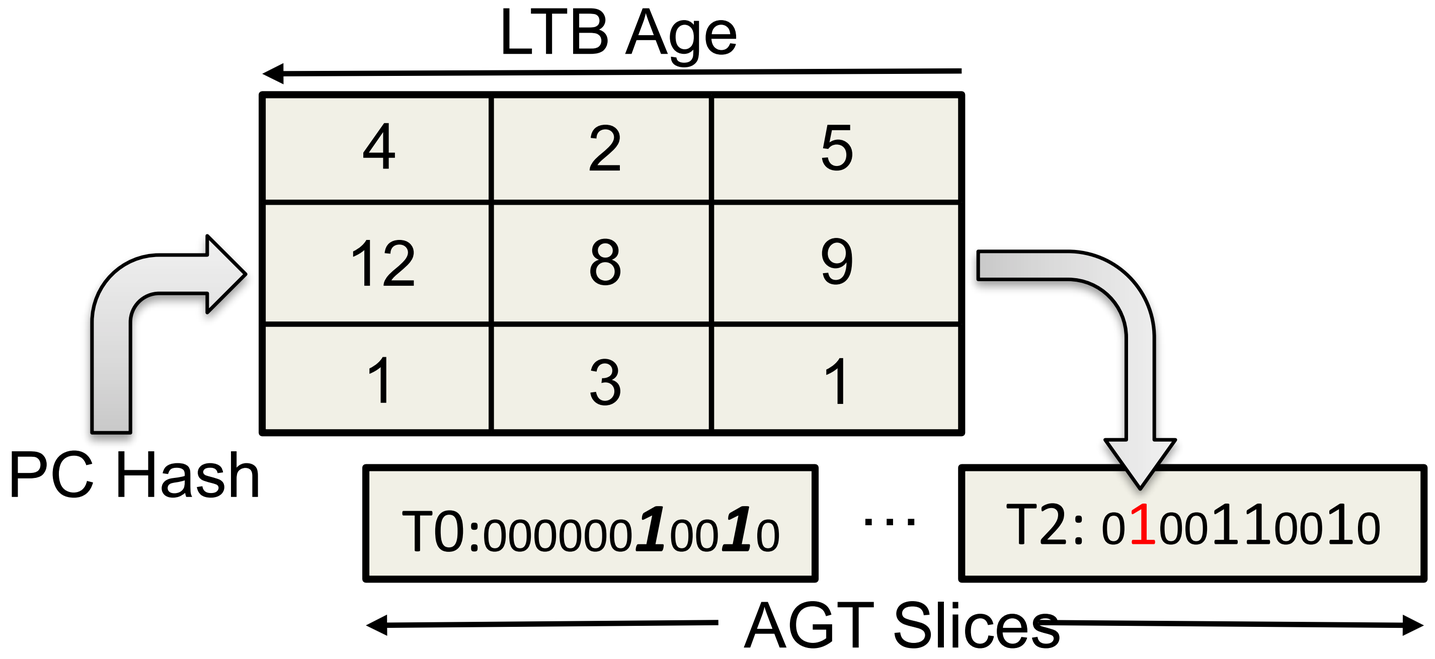
Adding Temporal Information

- Decompose all PHT entries into multiple **access tempos.**
- Store each disjoint tempo in its own “slice” of SMS.



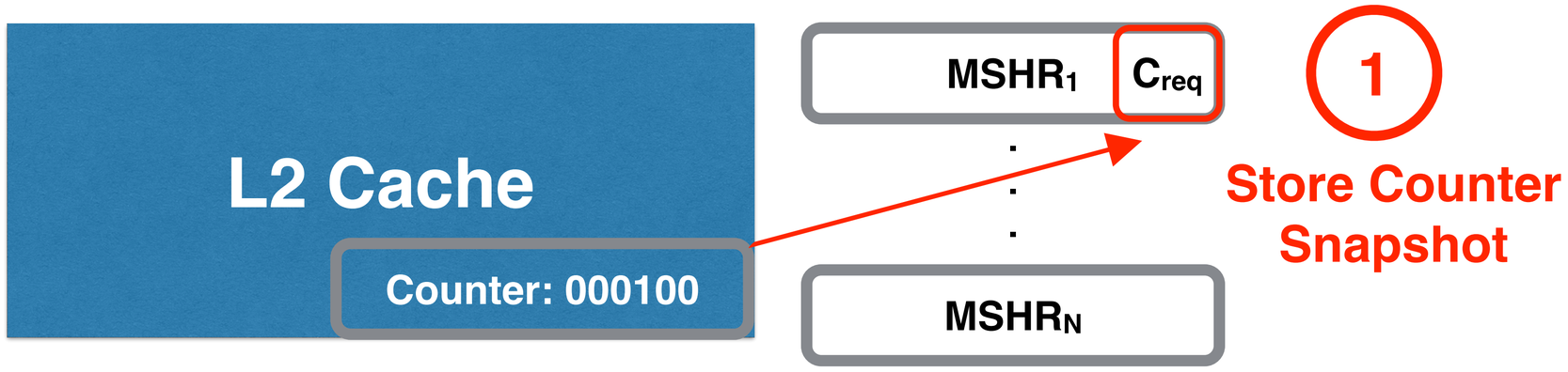
Training Based on Tempo

- Store the access tempos in on-chip Localized Tempo Buffers (LTBs).
- During training, only set the bits of the vector corresponding to the **currently measured tempo**.



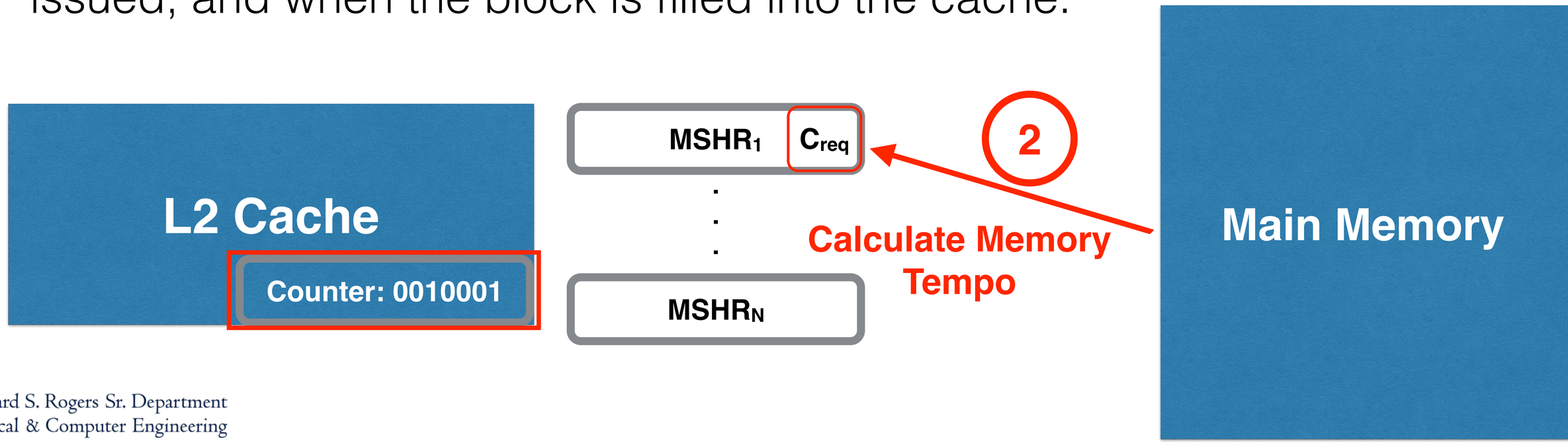
Prefetching Based on Tempo

- Simply knowing our current *access* tempo is not enough to ensure prefetch timeliness.
- Define **memory tempo**: Delta between when a memory request is issued, and when the block is filled into the cache.



Prefetching Based on Tempo

- Simply knowing our current *access* tempo is not enough to ensure prefetch timeliness.
- Define **memory tempo**: Delta between when a memory request is issued, and when the block is filled into the cache.

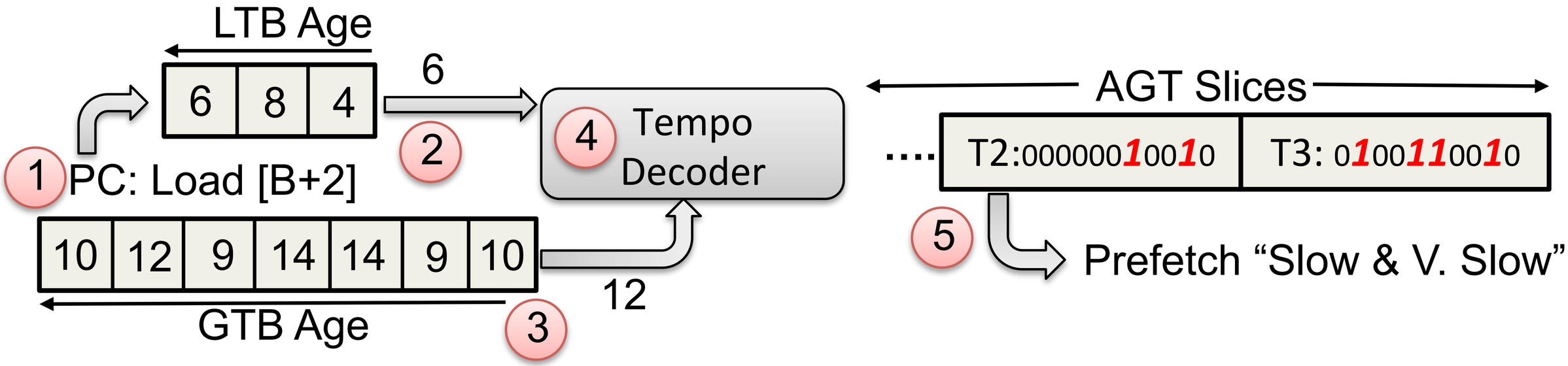


Rushing or Dragging?

- Can compare the memory's tempo against the current PC's access tempo at prefetch time.

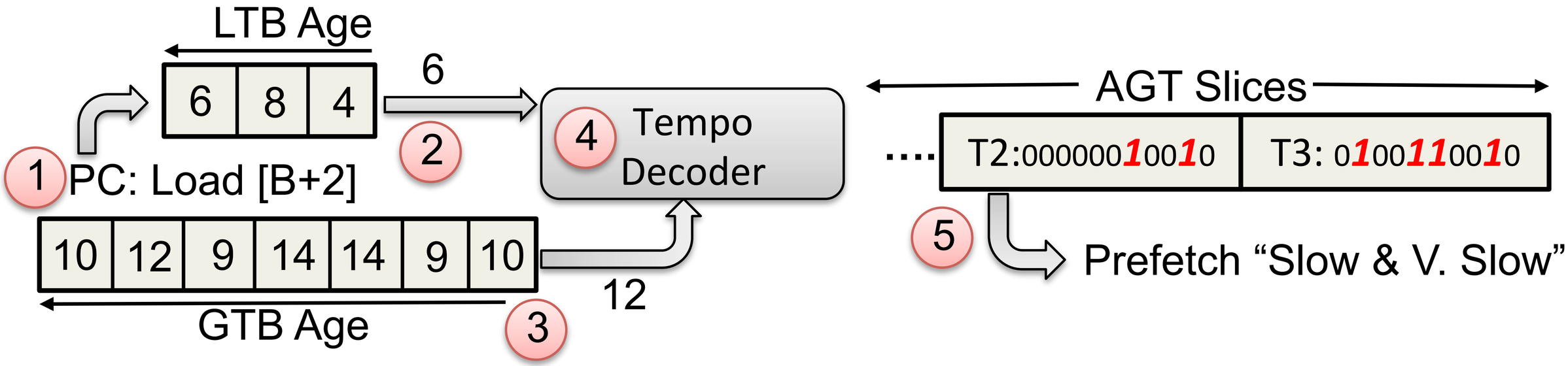
PC > Memory (Rushing)	Memory > PC (Dragging)
• Generating requests faster than memory is returning them. • Prefetch only **slower** tempos.	• Memory is fast enough to keep up with our requests. • Prefetch **faster** tempos**.**

Walkthrough Example



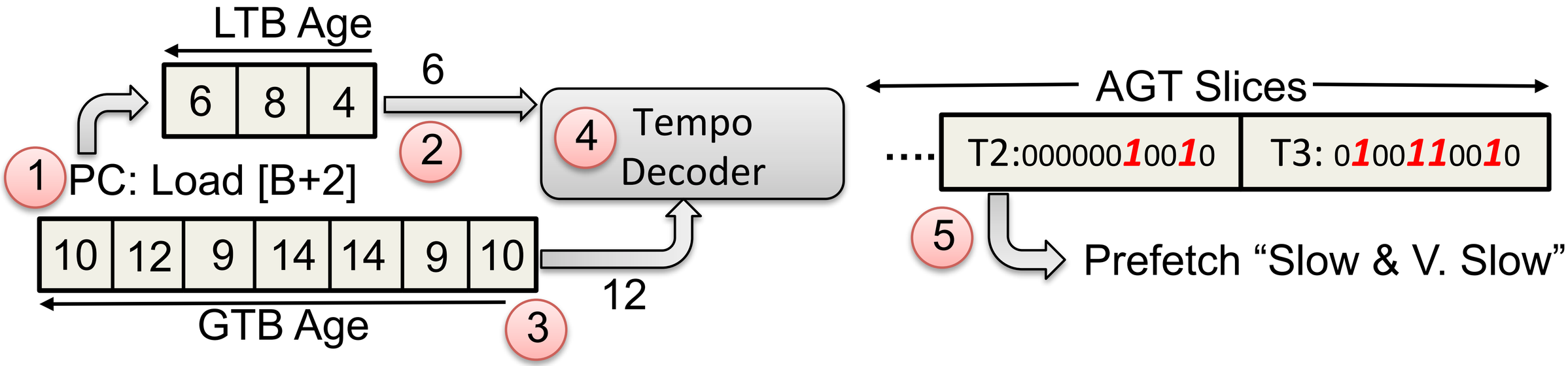
1. PC Observes Demand Request to Address [B+2]

Walkthrough Example



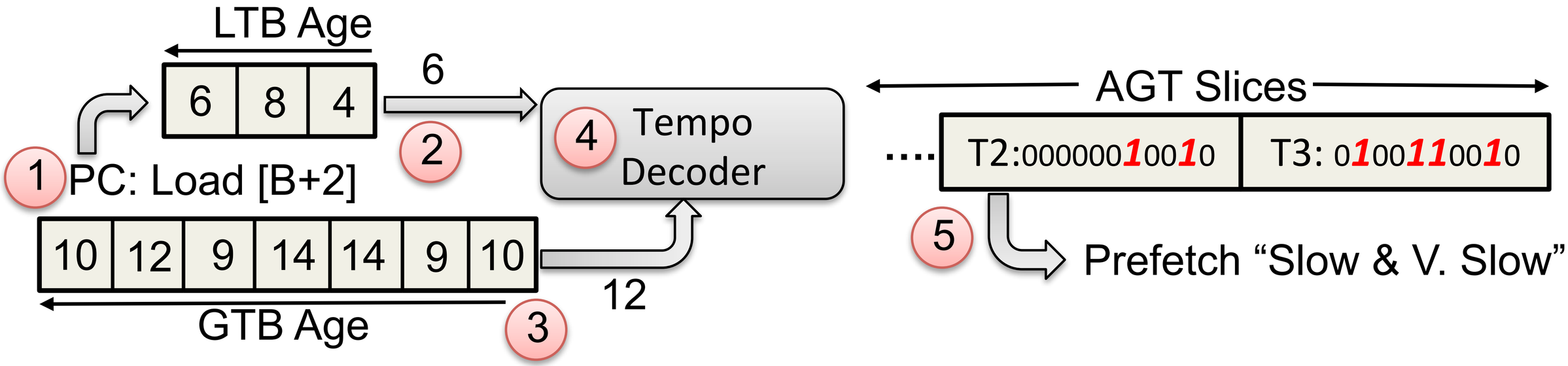
2. Lookup Current Local Access Tempo (6)

Walkthrough Example



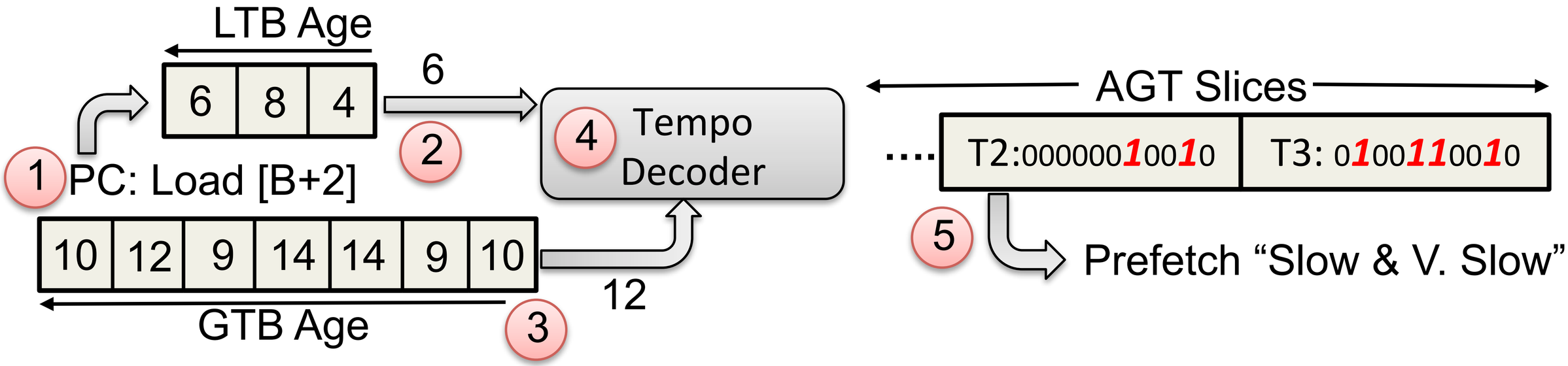
3. Average All Elements of Global Tempo Buffer (12)

Walkthrough Example



4. Decode “Rushing” or “Dragging”.

Walkthrough Example

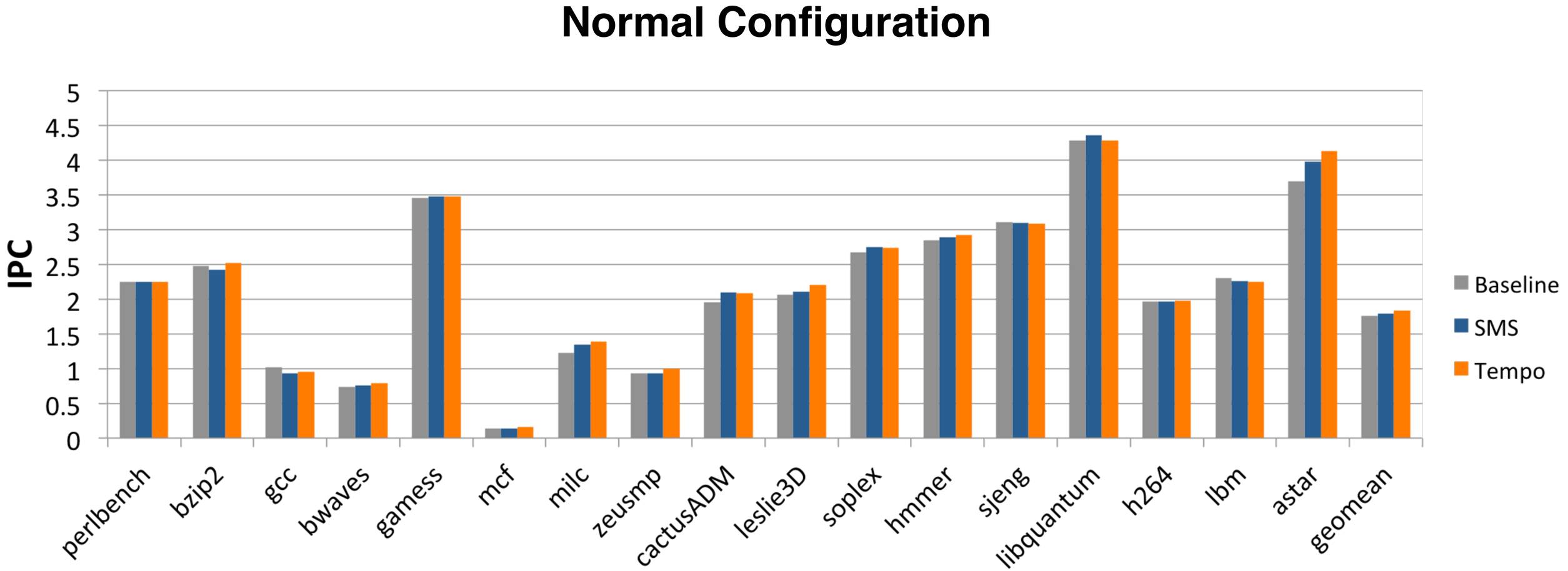


5. Prefetch Matching Tempos

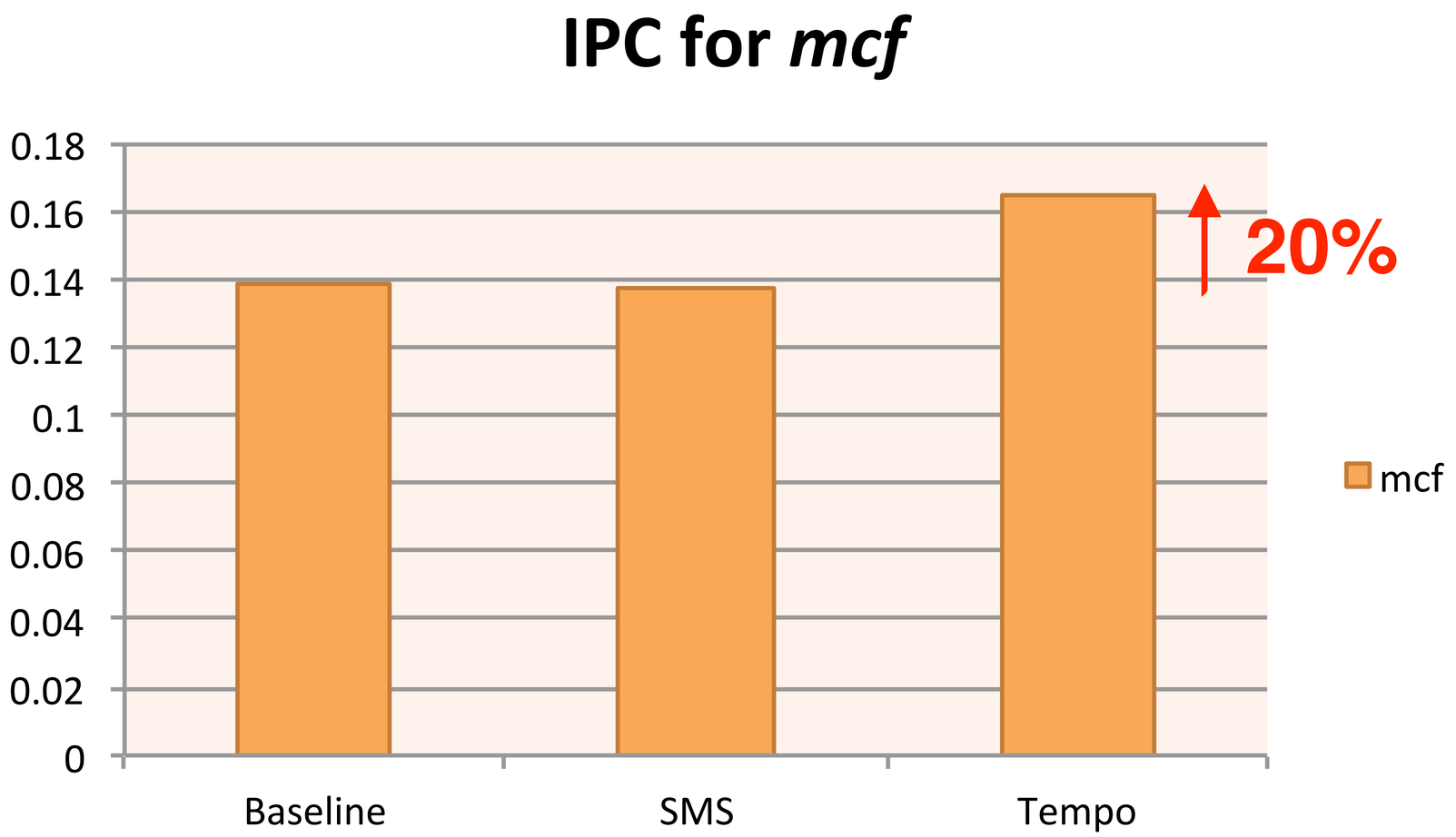
Evaluation

Core (OoO)	Memory
6-wide, 256 entry instruction window	16kB L1, 128kB L2, Fully inclusive L3
No fetch hazards.	32 entry L2 request queue
Issue 2 loads & 1 store / cycle.	Open Row FR-FCFS Mem. Scheduling
	t_{l2} = 10 cycles
	Config 1: 1MB L3, 12.8 GB/s memory
	Config 2: 256kB L3, 12.8 GB/s memory
	Config 3: 1 MB L3, 3.2 GB/s memory

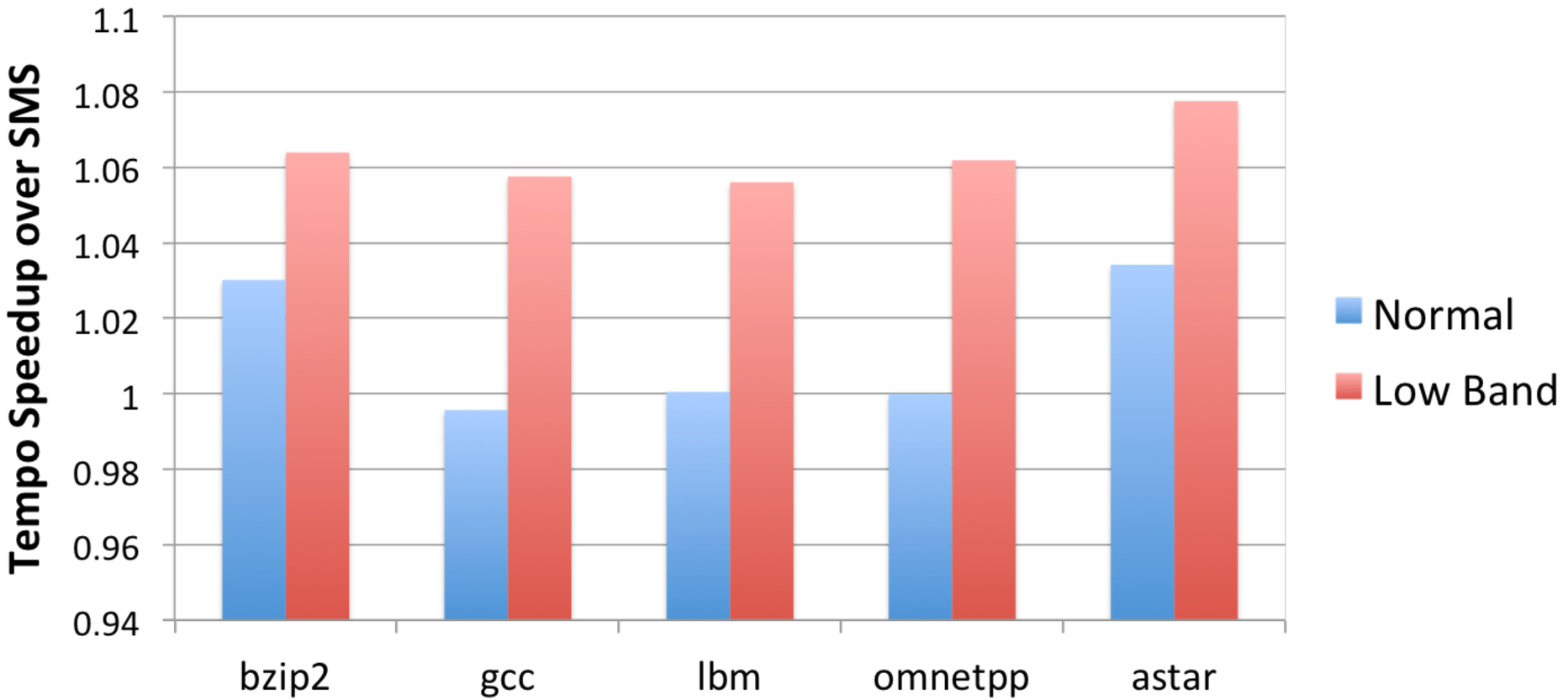
Absolute IPC Results



Absolute IPC Results

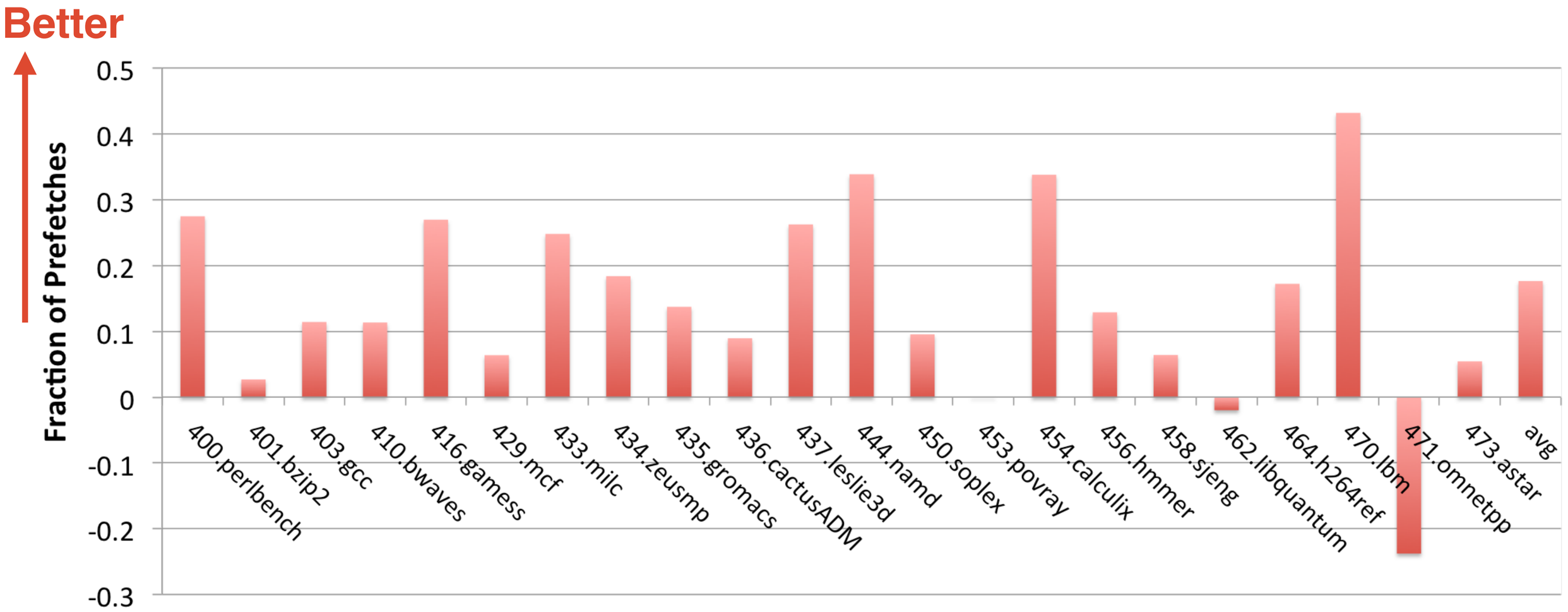


Increased Low B/W Performance



- Decreasing memory B/W increases Tempo's gains.

Reduction in Useless Prefetches



Conclusion

1. Presented a “sliced” version of SMS that filters prefetching decisions based on repeated PC access times.

2. Achieved **1.45%** and **2.57%** IPC improvement on Normal and Low Bandwidth configurations.

3. Reduced the number of useless prefetches by **17.6%**.

?