Sectored DRAM
A Practical Energy-Efficient and High-Performance Fine-Grained DRAM Architecture

Ataberk Olgun
olgunataberk@gmail.com

F. Nisa Bostanci Geraldo F. Oliveira Yahya Can Tugrul Rahul Bera
A. Giray Yaglikci Hasan Hassan Oguz Ergin Onur Mutlu

ETH zürich SAFARI kasırga TOBB ETÜ Ekonomi ve Teknoloji Üniversitesi

Sectored DRAM Summary

Problem: DRAM-based systems suffer from two sources of energy inefficiency

1. Coarse-grained cache-block-sized (typically 64-byte) data transfer
2. Coarse-grained DRAM-row-sized (typically 8-kilobyte) activation

A workload does not use all data fetched from DRAM

Goal: Design a fine-grained, low-cost, and high-throughput DRAM substrate

- Mitigate excessive energy consumption from coarse-grained DRAM

Key Ideas: Small modifications to memory controller and DRAM chip enable

1. Transferring sub-cache-block-sized data in a variable number of clock cycles
2. Activating relatively small physically isolated regions of a DRAM row

based on the workload memory access pattern

Key Results: For the evaluated memory-intensive workloads, Sectored DRAM

- Improves system energy consumption by 14%, system performance by 17%
- Incurs 0.39 mm^2 (1.7%) DRAM chip area overhead
- Performs within 11% of a state-of-the-art prior work (Half-DRAM), with 12% smaller DRAM energy and 34% smaller area overhead

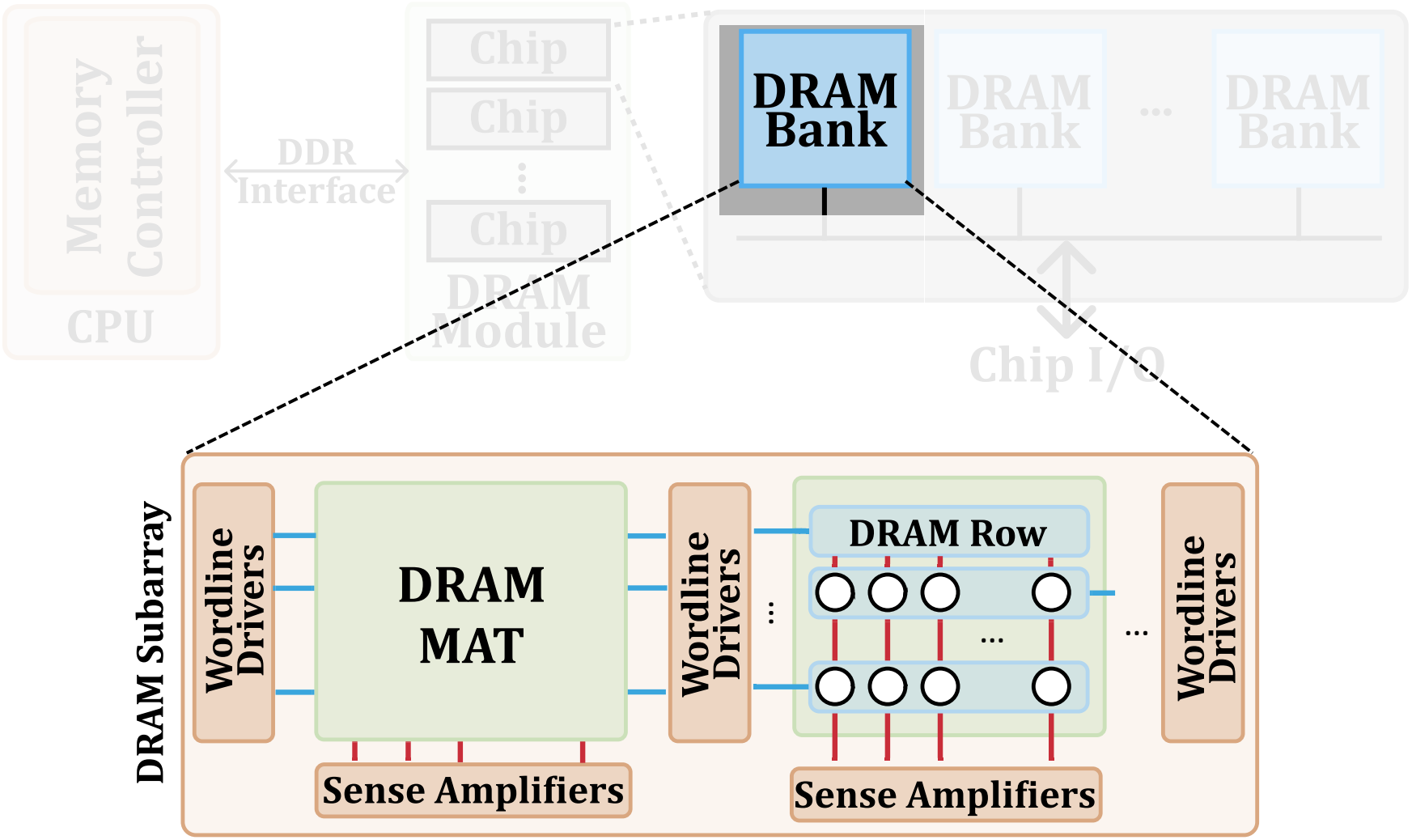
Outline

1. Background & Motivation
2. Sectored DRAM: Design
3. Sectored DRAM: System Integration
4. Evaluation
5. Conclusion

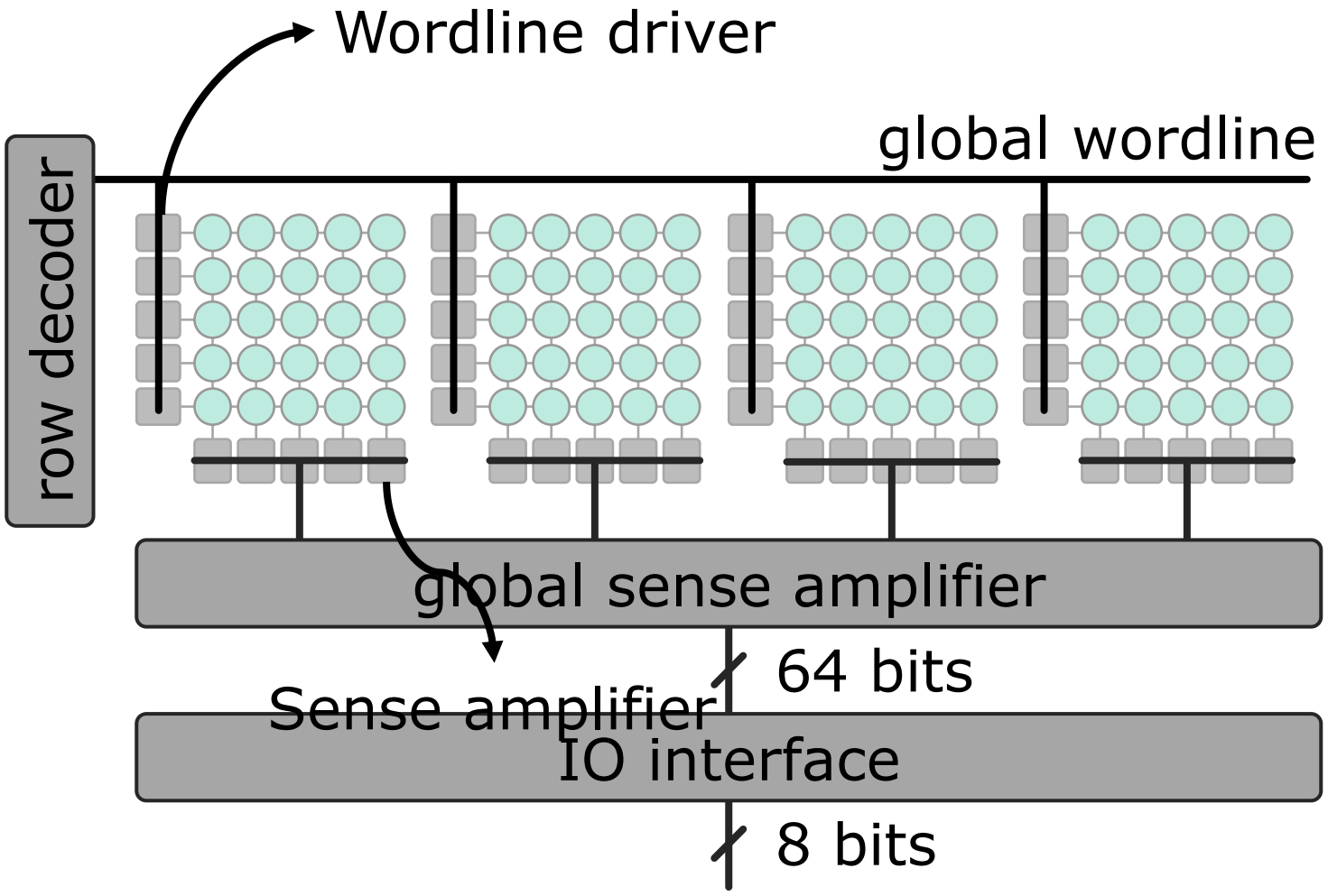
Outline

1. Background & Motivation
2. Sectored DRAM: Design
3. Sectored DRAM: System Integration
4. Evaluation
5. Conclusion

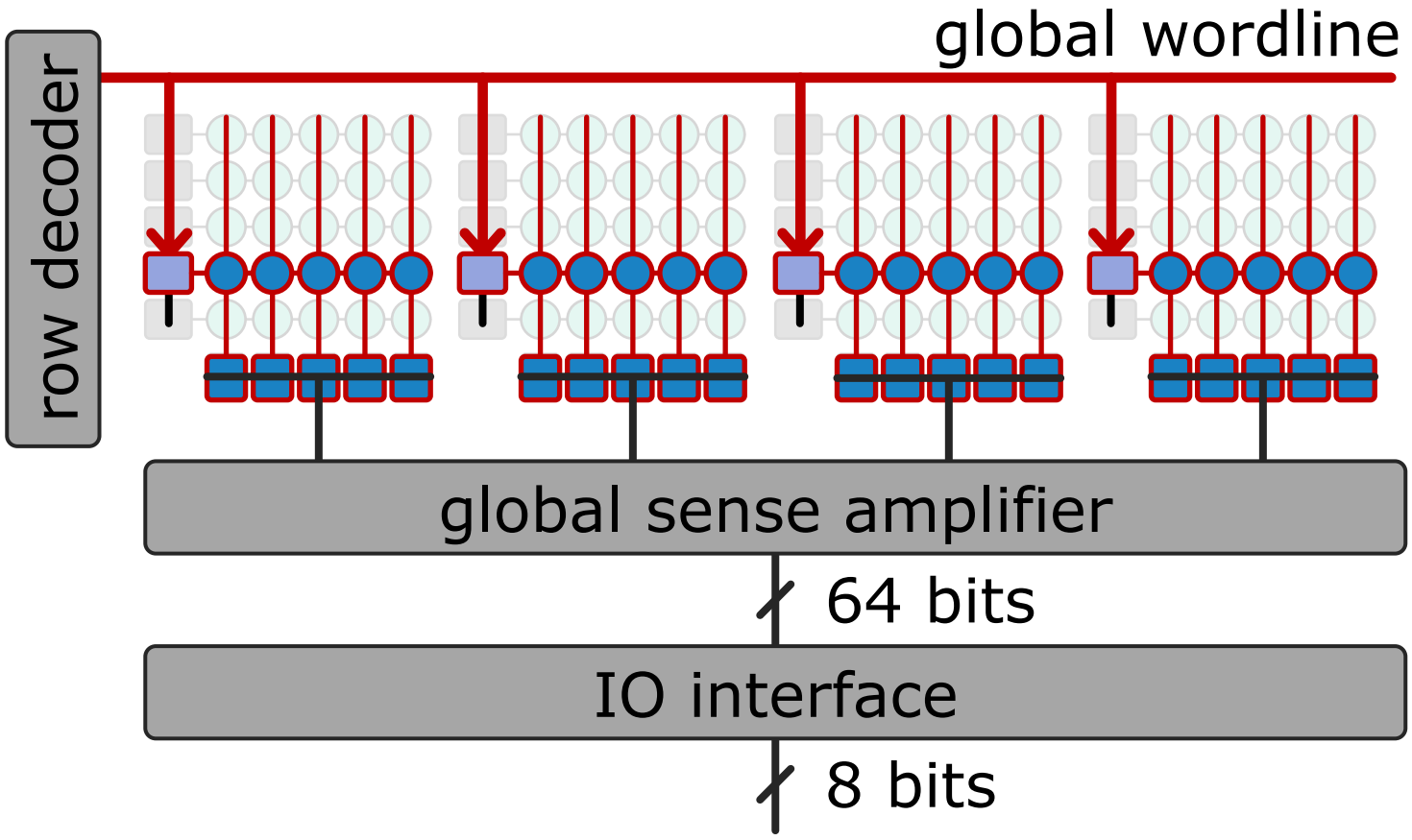
DRAM is Organized Hierarchically



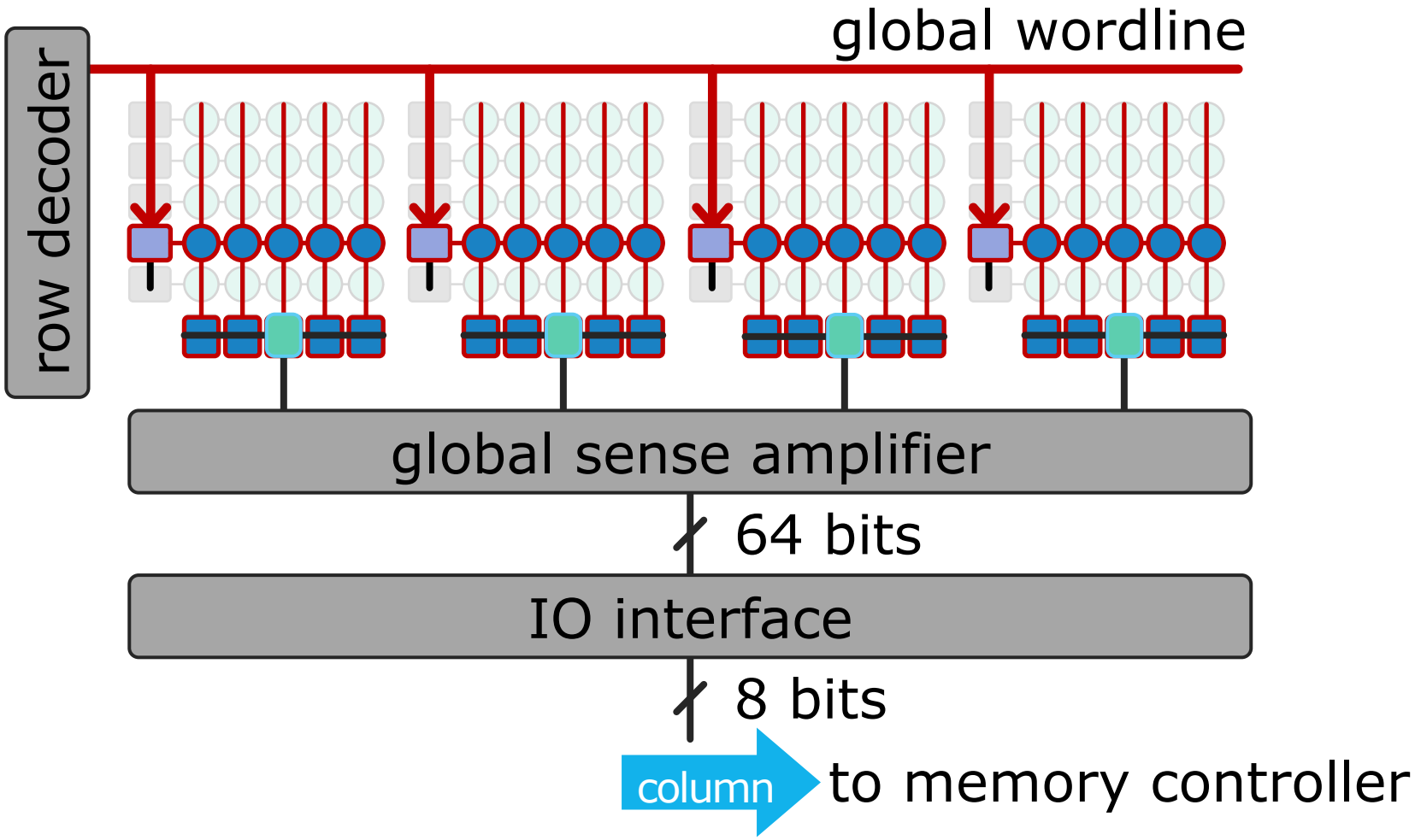
DRAM Row Activate Operation



DRAM Row Activate Operation



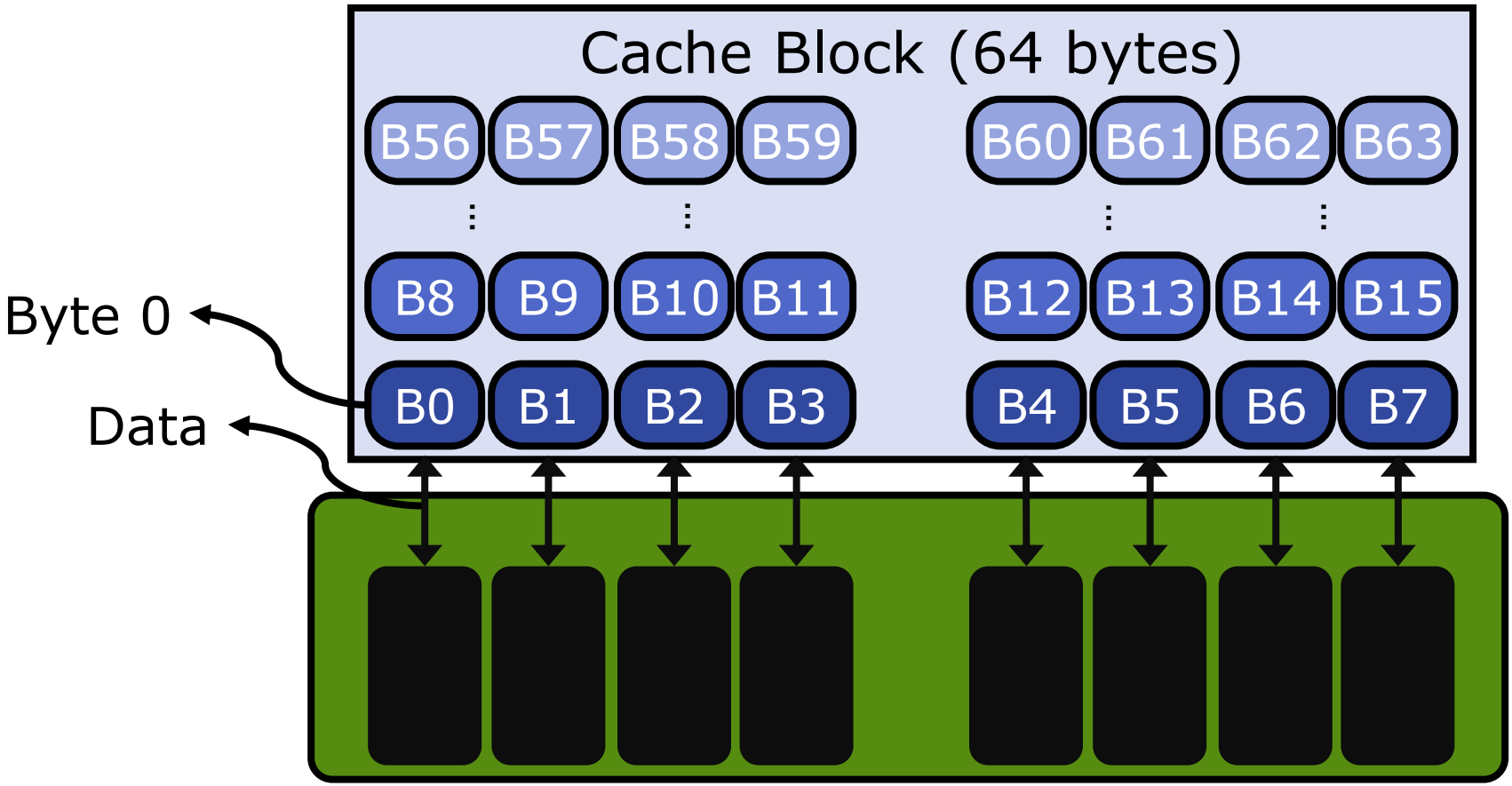
DRAM Column Read Operation



[Oliveira+, HPCA'24]

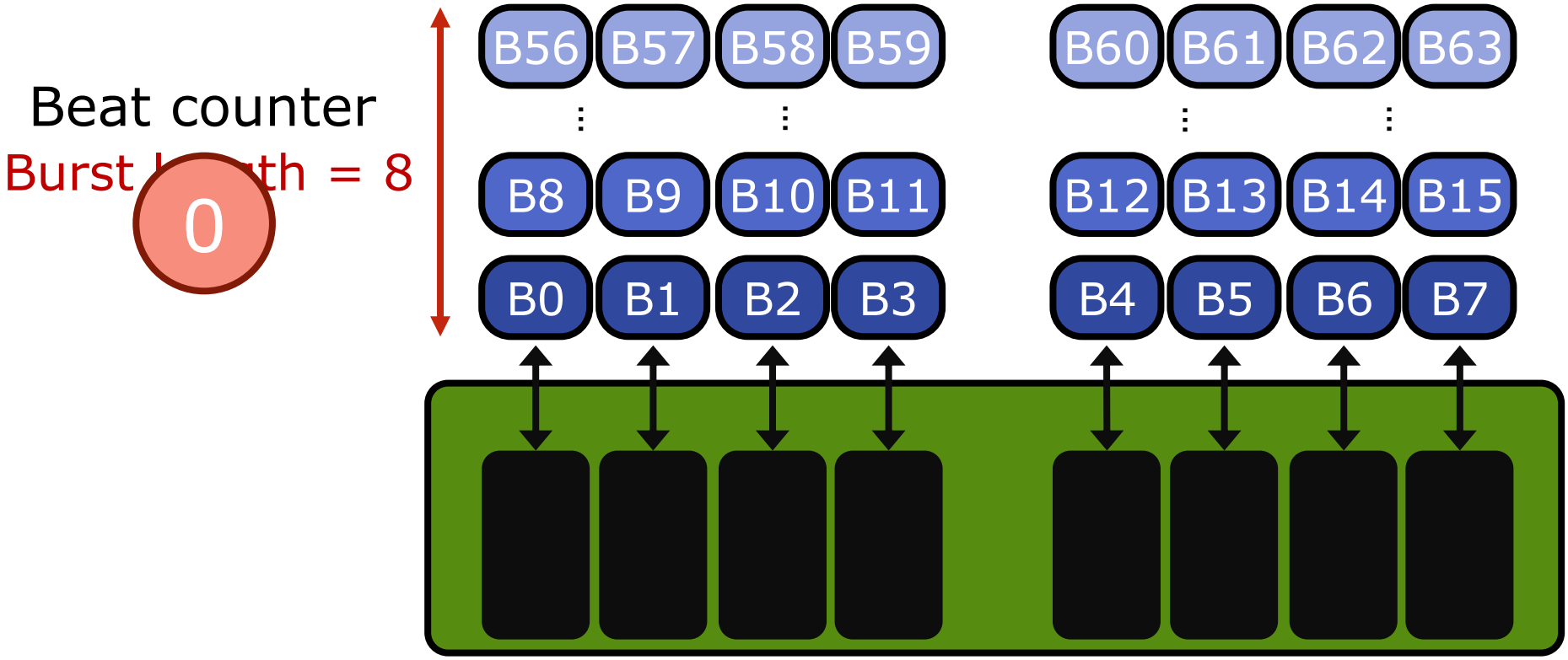
DRAM Data Transfer (I)

- DRAM data transfer happens in cache block granularity



DRAM Data Transfer (I)

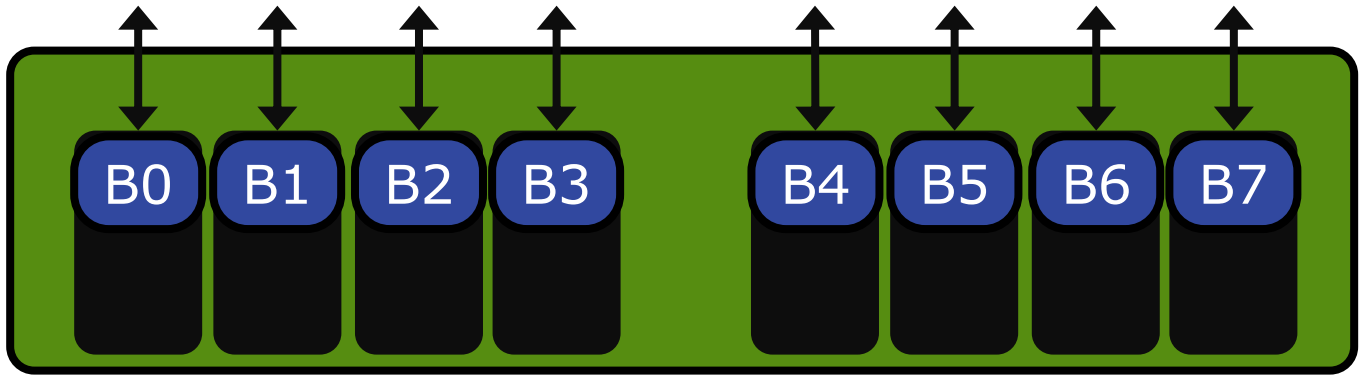
- DRAM data transfer happens in cache block granularity
- Using data transfer bursts (or bursts)



DRAM Data Transfer (I)

- DRAM data transfer happens in cache block granularity
- Using data transfer bursts (or bursts)

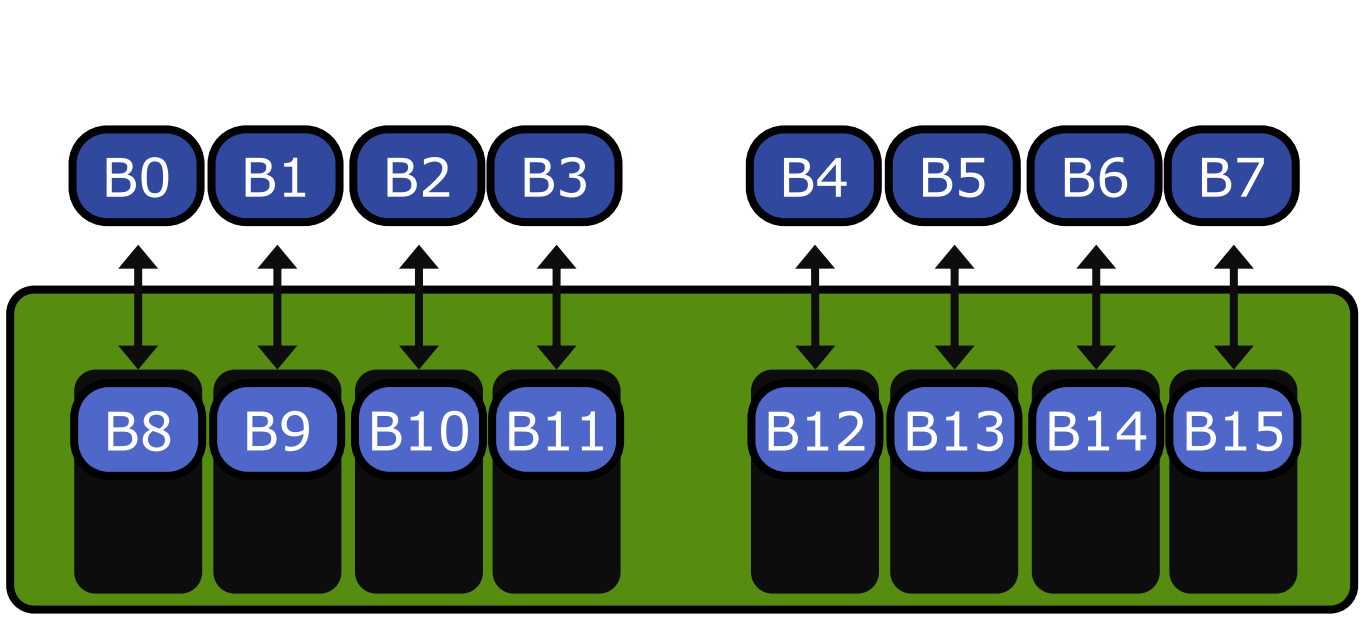
Beat counter



DRAM Data Transfer (I)

- DRAM data transfer happens in cache block granularity
- Using data transfer bursts (or bursts)

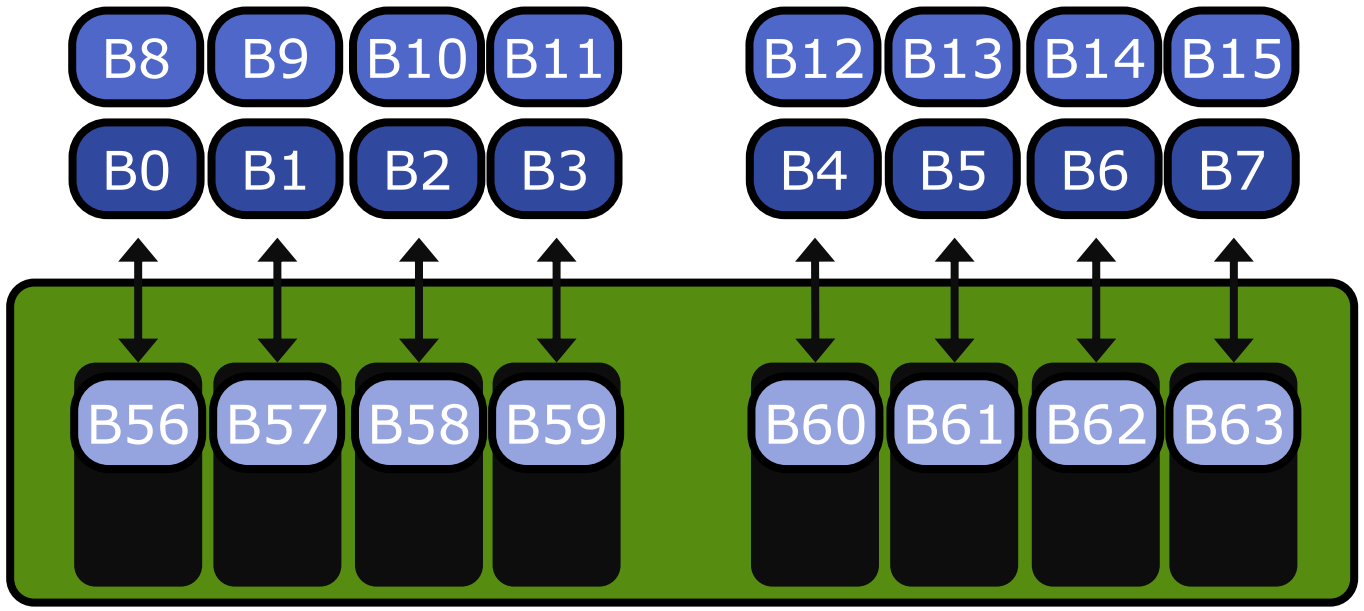
Beat counter



DRAM Data Transfer (I)

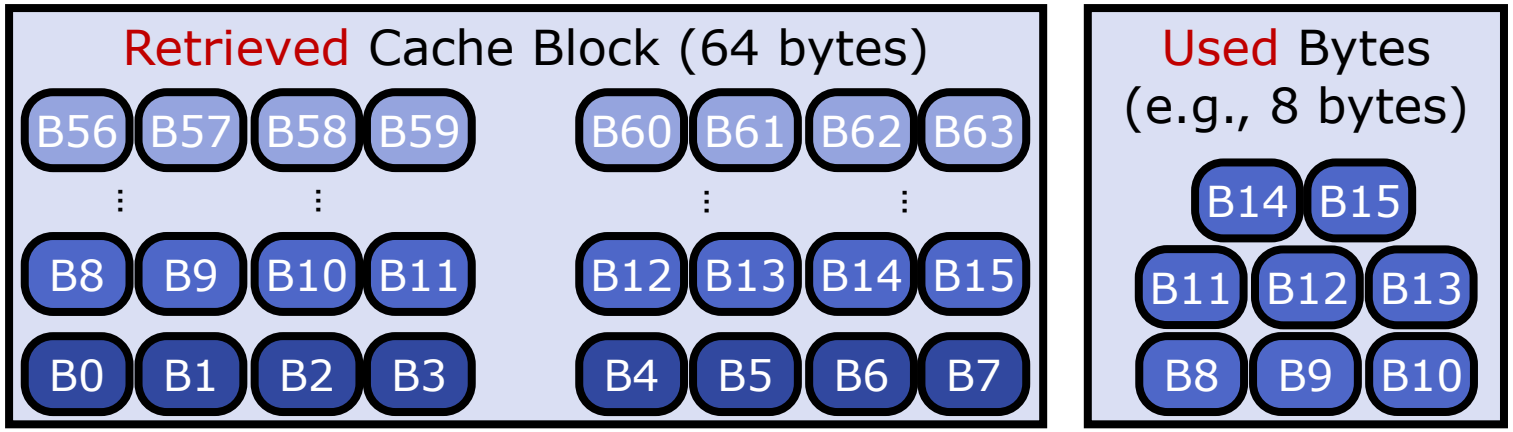
- DRAM data transfer happens in cache block granularity
- Using data transfer bursts (or bursts)

Beat counter



Coarse-Grained DRAM Data Transfer Wastes Energy

- Retrieve more bytes than necessary with each word (e.g., 8 bytes) access

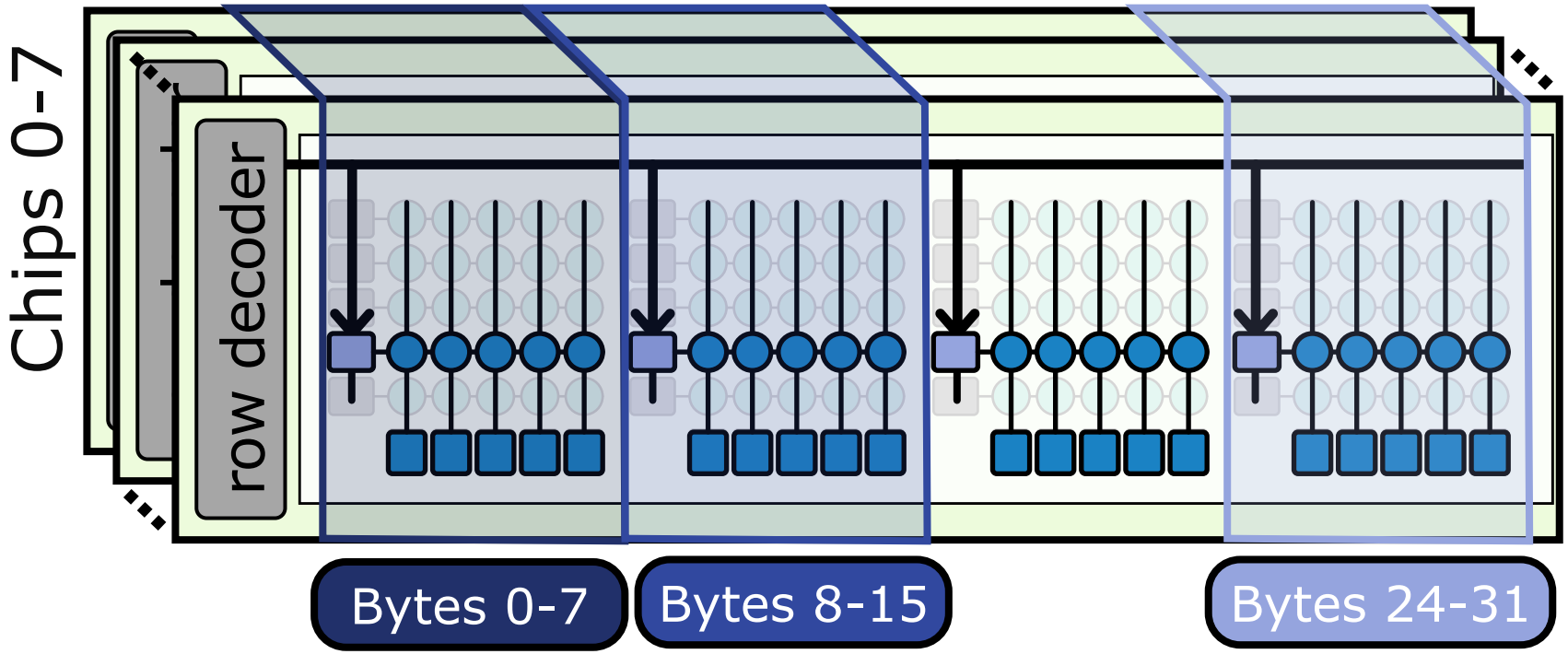


- Exploit spatial locality
- Not all words in a cache block are referenced by CPU load/store instructions

Less than 60% of words used on average (e.g., [Qureshi+, HPCA'07])

Coarse-Grained DRAM Row Activation Wastes Energy

- Activate more mats than necessary with each DRAM row activation

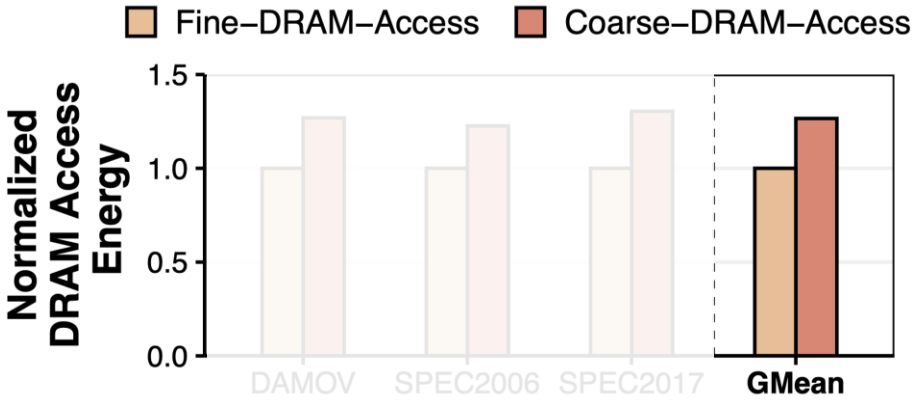


- Transfer all words of a cache block in one burst
- Not all mats need to be read or updated

Fine-Grained DRAM Can Greatly Improve System Energy Efficiency

Fine-DRAM-Access: Enable word-sized (8-byte) data transfers

Fine-DRAM-Activation: Enable per-mat DRAM row activation



Fine-Grained DRAM can improve READ/WRITE (ACTIVATE) energy by 27% (4%)

Challenges of Enabling Fine-Grained DRAM

Prior works

FGA
SBA
HalfDRAM
HalfPage
PRA

1. Maintaining high DRAM data transfer throughput
2. Incurring low DRAM area overhead
3. Fully exploiting fine-grained DRAM

Problem and Goal

1. Maintaining high DRAM data transfer throughput
2. Incurring low DRAM area overhead
3. Fully exploiting fine-grained DRAM

Problem

No prior work overcomes all three challenges

Goal

Develop a new, low-cost, and high-throughput DRAM substrate that can mitigate the excessive energy consumption of coarse-grained DRAM

Outline

1. Background & Motivation
2. Sectored DRAM: Design
3. Sectored DRAM: System Integration
4. Evaluation
5. Conclusion

Two Key Design Components

Two key observations regarding DRAM chip design enable Sectored DRAM at low cost

- **Observation:** DRAM mats naturally split DRAM rows into small fixed-size portions

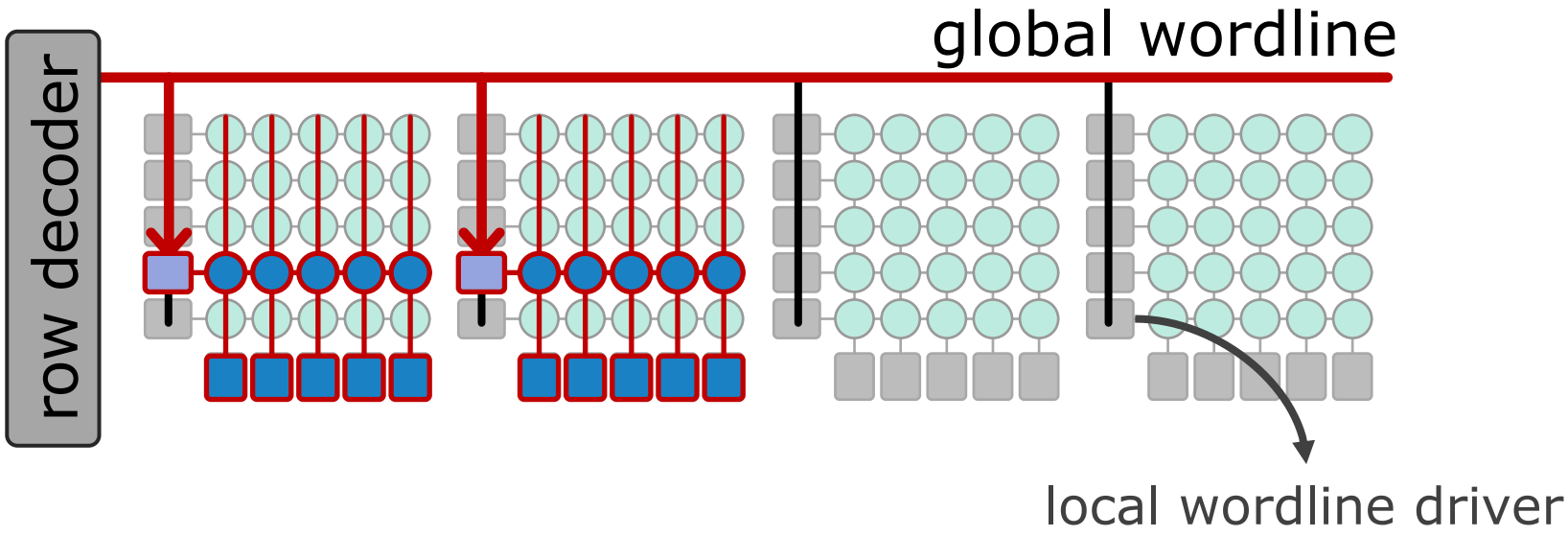
1 Sectored Activation (SA)

- **Observation:** DRAM I/O circuitry can already transfer a small portion of a cache block in one beat

2 Variable Burst Length (VBL)

Component 1: Sectored Activation

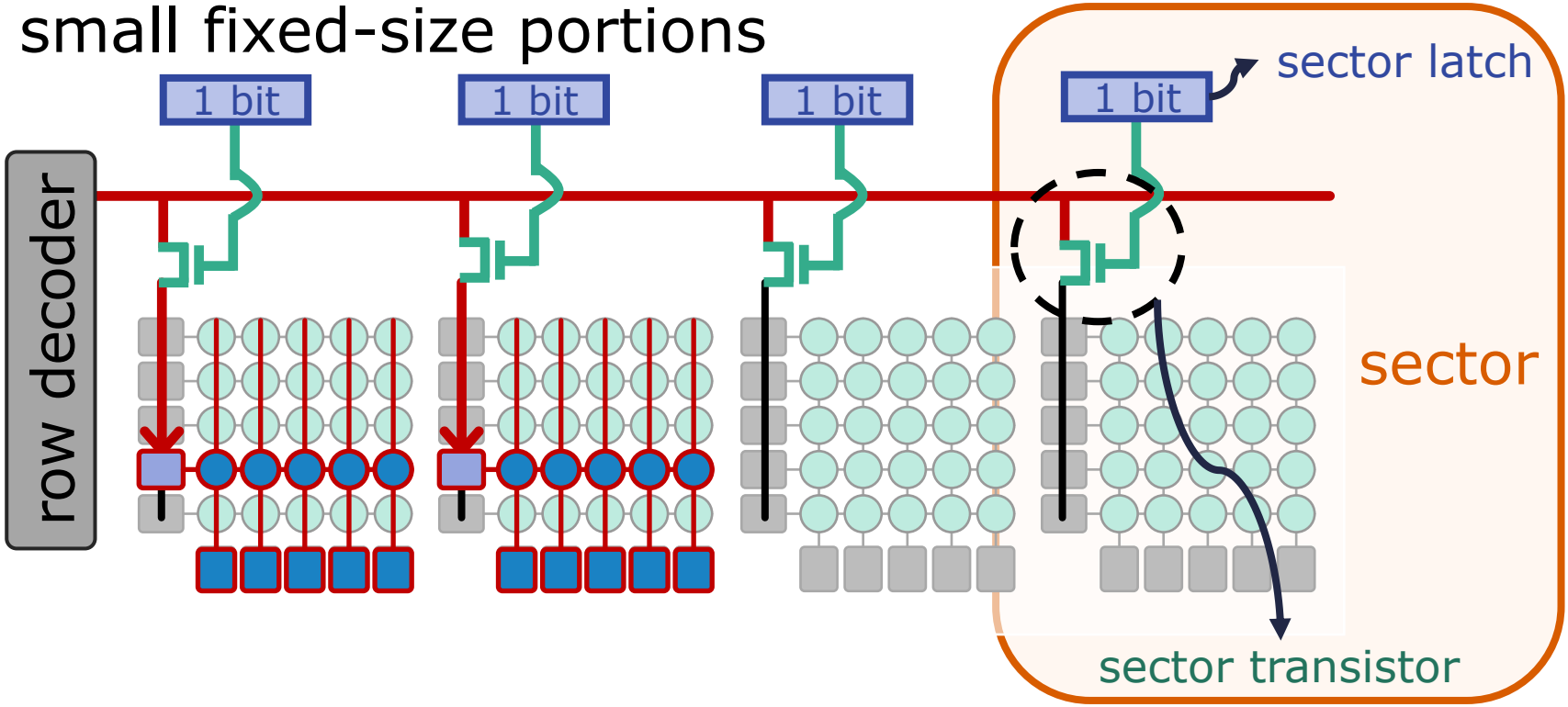
- **Observation:** DRAM mats naturally split DRAM rows into small fixed-size portions



- To select and activate one or multiple mats:
  1. Isolate the global wordline from local wordline drivers

Component 1: Sectored Activation

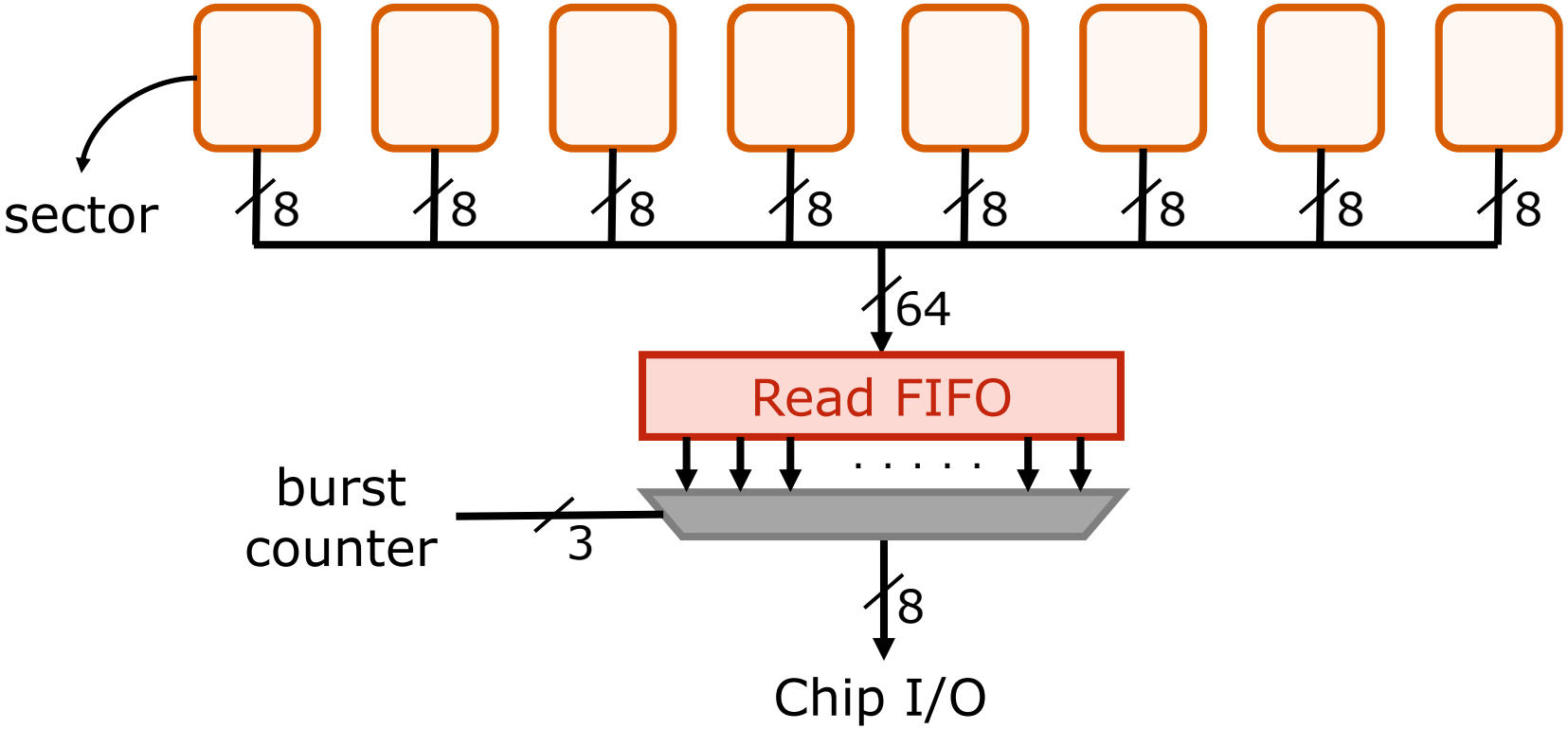
- **Observation:** DRAM mats naturally split DRAM rows into small fixed-size portions



- To select and activate one or multiple mats:
  1. Isolate the global wordline from local wordline drivers
  2. Add a control signal (1 bit) for each mat

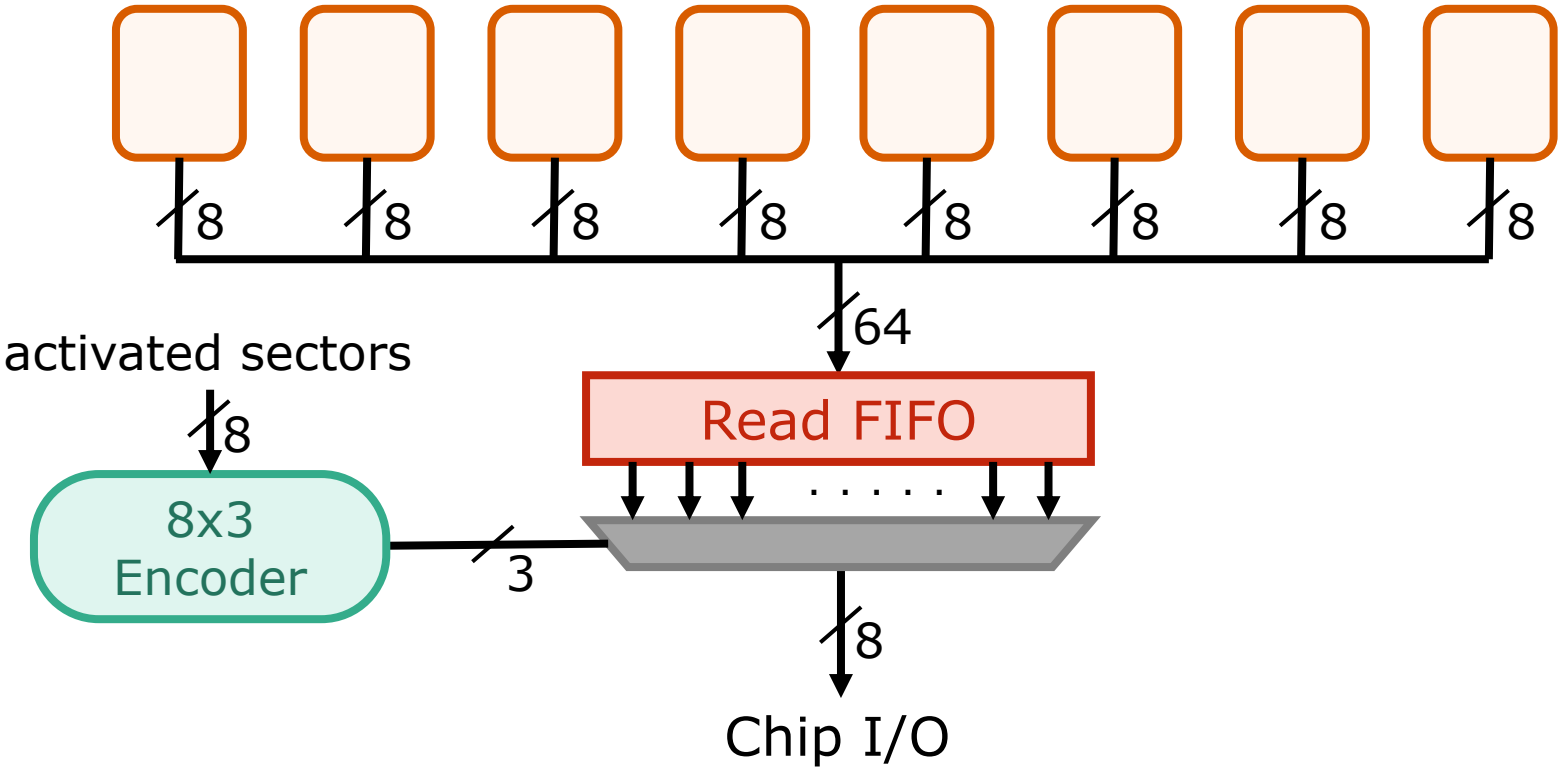
Component 2: Variable Burst Length

- **Observation:** DRAM I/O circuitry can already transfer a small portion of a cache block in one beat



Component 2: Variable Burst Length

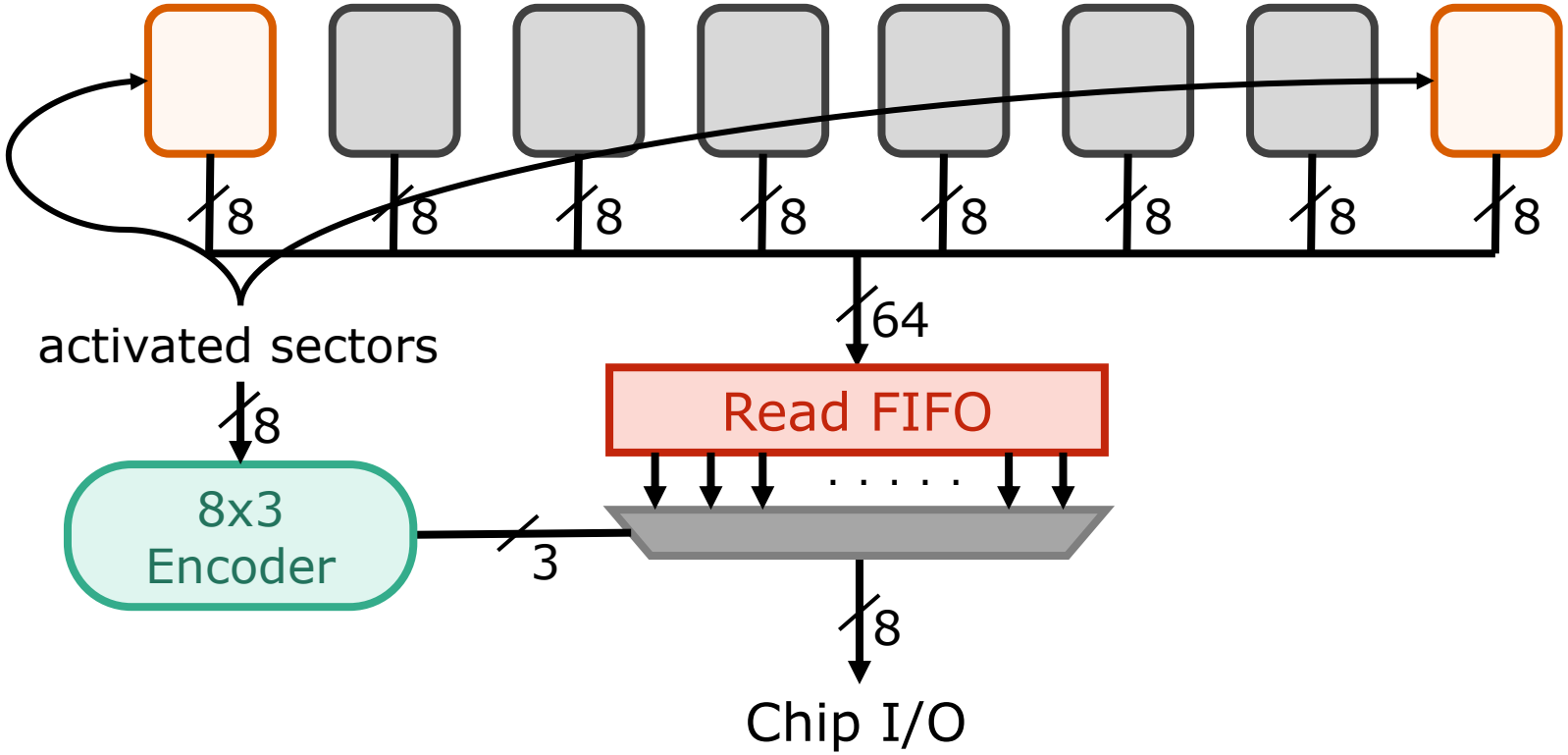
- **Observation:** DRAM I/O circuitry can already transfer a small portion of a cache block in one beat



- Replace the burst counter with an encoder that selects only the open/activated sectors

Component 2: Variable Burst Length

- **Observation:** DRAM I/O circuitry can already transfer a small portion of a cache block in one beat



- Replace the burst counter with an encoder that selects only the open/activated sectors

Exposing Sectored DRAM to the Memory Controller

- Goal: Give memory controller (MC) control over sectors
  - E.g., activate 3 out of 8 sectors in a subarray
- Modifications to the standard interface not required

Precharge (PRE) command closes the open DRAM row

- More than 10 unused bits in PRE command encoding
- The previously unused bits now determine the sectors opened by the next activate (ACT) command

A memory controller can leverage Sectored DRAM without any physical DRAM interface modifications

Sectored DRAM: A Practical Energy-Efficient and High-Performance Fine-Grained DRAM Architecture

Ataberk Olgun[§] F. Nisa Bostancı[§†] Geraldo F. Oliveira[§] Yahya Can Tuğrul[§†] Rahul Bera[§]
A. Giray Yağlıkcı[§] Hasan Hassan[§] Oğuz Ergin[†] Onur Mutlu[§]

[§]*ETH Zürich* [†]*TOBB University of Economics and Technology*

Modern computing systems access data in main memory at coarse granularity *(e.g., at 512-bit cache block granularity). Coarse-grained access leads to wasted energy because the system does* not *use all individually accessed small portions (e.g.,* words, *each of which typically is 64 bits) of a cache block. In modern DRAM-based computing systems, two key coarse-grained access mechanisms lead to wasted energy: large and fixed-size (i) data transfers between DRAM and the memory controller and (ii) DRAM row activations.*

We propose Sectored DRAM, a new, low-overhead DRAM substrate that reduces wasted energy by enabling fine-grained *DRAM data transfer and DRAM row activation. To retrieve only useful data from DRAM, Sectored DRAM exploits the observation that many cache blocks are not fully utilized in many workloads due to poor spatial locality. Sectored DRAM predicts the words in a cache block that will likely be accessed during the cache block's*

1. Introduction

DRAM [22] is hierarchically organized to improve scaling in density and performance. At the highest level of the hierarchy, a DRAM chip is partitioned into banks that can be accessed simultaneously [87, 57, 58, 59, 63]. At the lowest level, a collection of DRAM rows (DRAM cells that are activated together) are typically divided into multiple *DRAM mats* that can operate individually [52, 42, 125, 58]. Even though DRAM chips are hierarchically organized, standard DRAM interfaces (e.g., DDRx [43, 44, 45]) do *not* expose DRAM mats to the memory controller. To access even a single DRAM cell, the memory controller needs to activate a large number of DRAM cells (e.g., 65,536 DRAM cells in a DRAM row in DDR4 [80]) and transfer many bits (e.g., a cache block, typically 512 bits [32]) over the memory channel. Thus, in current systems, both DRAM data transfer and activation are *coarse-grained.* Coarse-grained data

https://arxiv.org/pdf/2207.13795.pdf

Outline

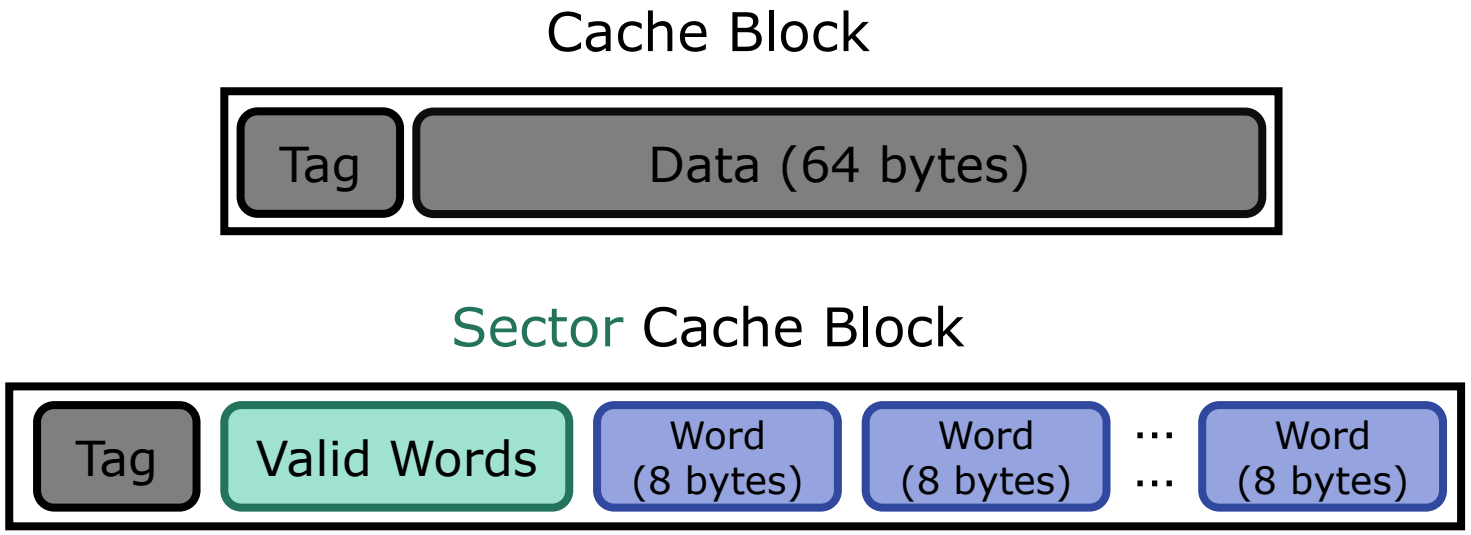
1. Background & Motivation
2. Sectored DRAM: Design
3. Sectored DRAM: System Integration
4. Evaluation
5. Conclusion

Efficient System Integration of Sectored DRAM is Challenging (I)

Challenge 1: Requires system-wide modifications to enable sub-cache-block (e.g., word) granularity data transfers

Solution: Use sector caches (e.g., [Liptay+,1968])

- Extend a cache block with 1 bit for each word
- A bit indicates if its corresponding word is valid

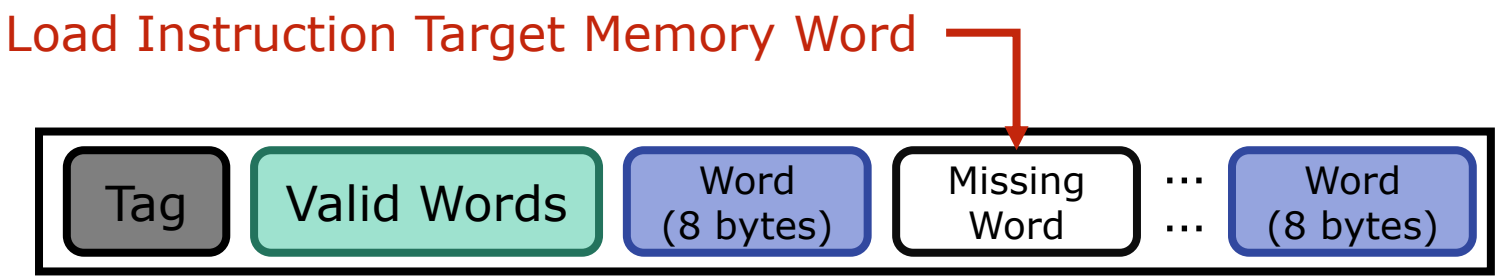


Efficient System Integration of Sectored DRAM is Challenging (II)

Challenge 2: Missing words (sectors) in a cache block cause additional performance overhead

Solution: Develop two prediction techniques

1) A technique to exploit the spatial locality in subsequent load/store (LD/ST) instructions
2) A spatial pattern predictor (e.g., [Kumar+,1998]) tailored for predicting useful words (similar to [Yoon+, 2012])



Efficient System Integration of Sectored DRAM is Challenging (II)

Challenge 2: Missing words (sectors) in a cache block cause additional performance overhead

Solution: Develop two prediction techniques

1) A technique to exploit the spatial locality in subsequent load/store (LD/ST) instructions

2) A spatial pattern predictor (e.g., [Kumar+,1998]) tailored for predicting useful words (similar to [Yoon+, 2012])

Load/Store Queue (LSQ) Lookahead

- One load/store instruction *references* one word in main memory
- **Key Mechanism:** 1) Collect references from *younger* load/store instructions
  2) store the collected references in the *oldest* load/store instr.

A load/store instruction retrieves all words in a cache block that will be referenced in the near future to the L1 cache with only one cache access

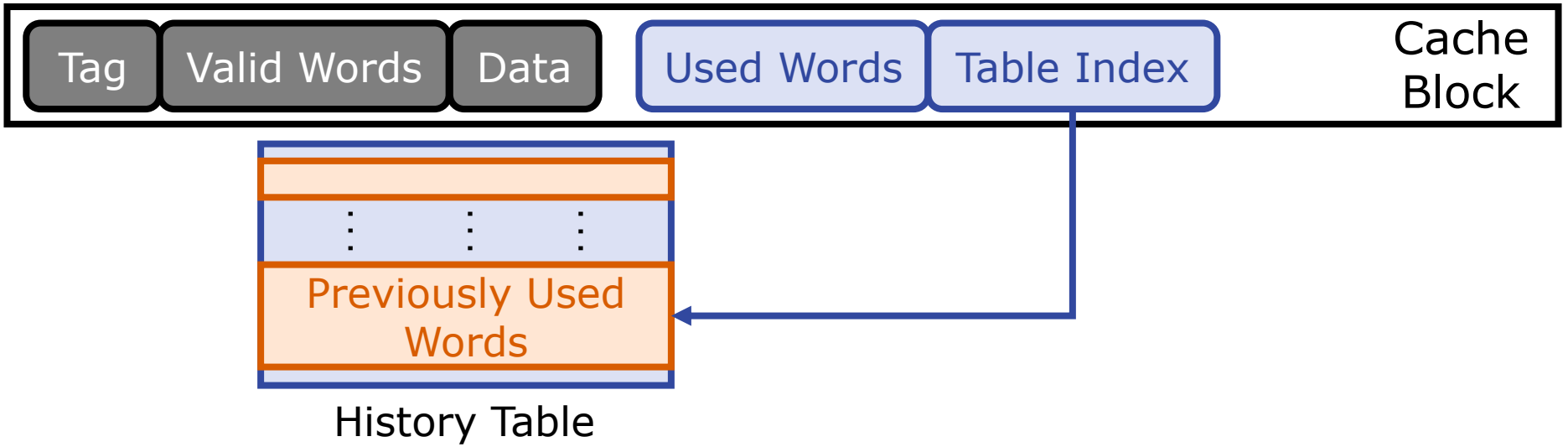
LSQ Lookahead has two key drawbacks

- LSQ is not large enough to store many LD/ST instructions
- Dependencies prevent computation of future LD/ST instruction addresses

Sector Predictor (SP)

Key Idea: Complement LSQ Lookahead and minimize sector misses

- Used (referenced) words in a cache block form a signature
- Reuse this signature when the same cache block misses in the cache



Outline

1. Background & Motivation
2. Sectored DRAM: Design
3. Sectored DRAM: System Integration
4. Evaluation
5. Conclusion

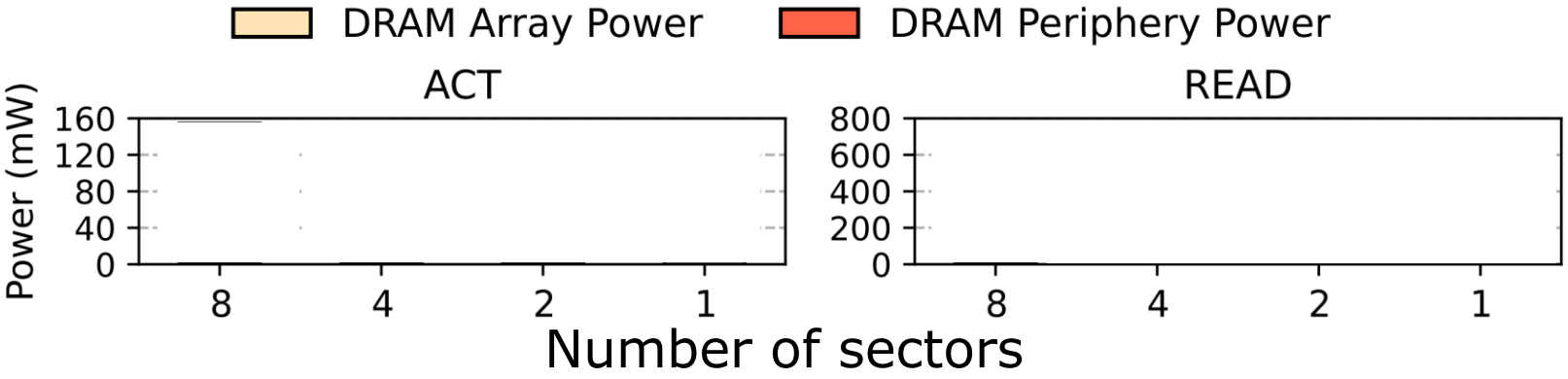
Evaluation Methodology

- **Performance and energy consumption evaluation:**
  Cycle-level simulations using Ramulator
  Rambus Power Model and DRAMPower for DRAM energy
  CACTI & McPAT for processor energy estimation
- **System Configuration:**

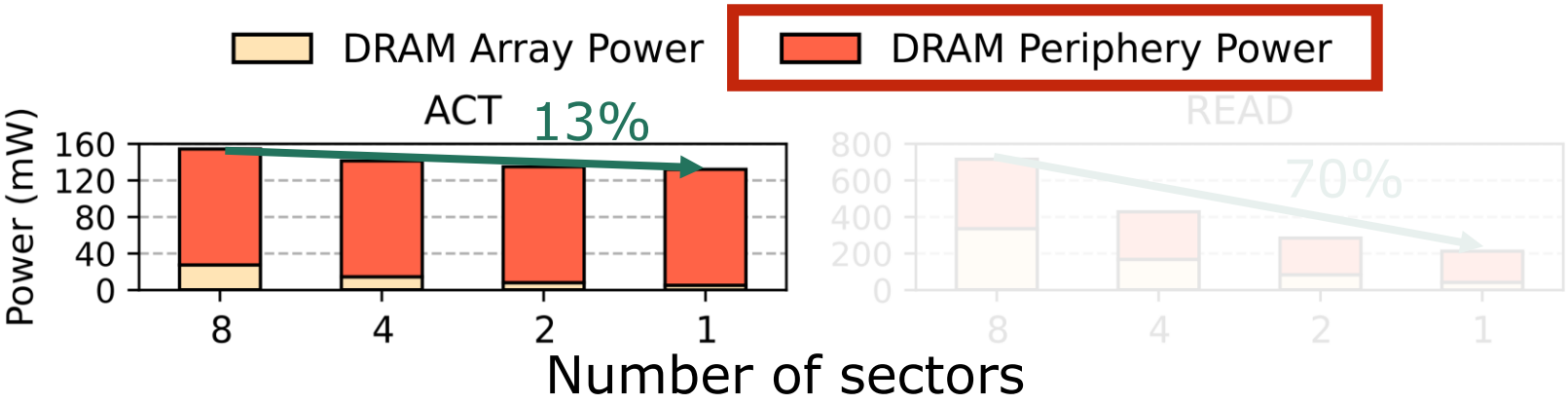
Processor	1-16 cores, 3.6GHz clock frequency, 4-wide issue, 128-entry instruction window 32 KiB L1, 256 KiB L2, and 8 MiB L3 caches
DRAM	DDR4, 1-4 channel, 4 rank/channel, 4 bank groups, 4 banks/bank group, 32K rows/bank, 3200 MT/s
Memory Ctrl.	64-entry read and write requests queues, FR-FCFS with a column cap of 16

- **Sectored DRAM Policies:** Always-On and Dynamic
  - Always-On: Never disable Sectored DRAM
  - Dynamic: Dynamically turn on Sectored DRAM based on workload memory intensity
- **Comparison Points:** 3 state-of-the-art fine-grained DRAM mechanisms
  - HalfDRAM [Zhang+, ISCA'14] (best performing),
  - Fine-Grained Activation [Cooper-Balis+, IEEE MICRO'10] (lowest area overhead),
  - Partial Row Activation [Lee+, HPCA'17]
- **Workloads:** 41 1-,2-,4-,8-,16-core (multiprogrammed) workloads
  - SPEC CPU2006, SPEC CPU2017, DAMOV benchmark suites

Sectored DRAM Can Greatly Reduce DRAM ACT and READ Power



Sectored DRAM Can Greatly Reduce DRAM ACT and READ Power



Reading from (activating) one sector takes 70% (13%) less power than reading from (activating) all 8 sectors

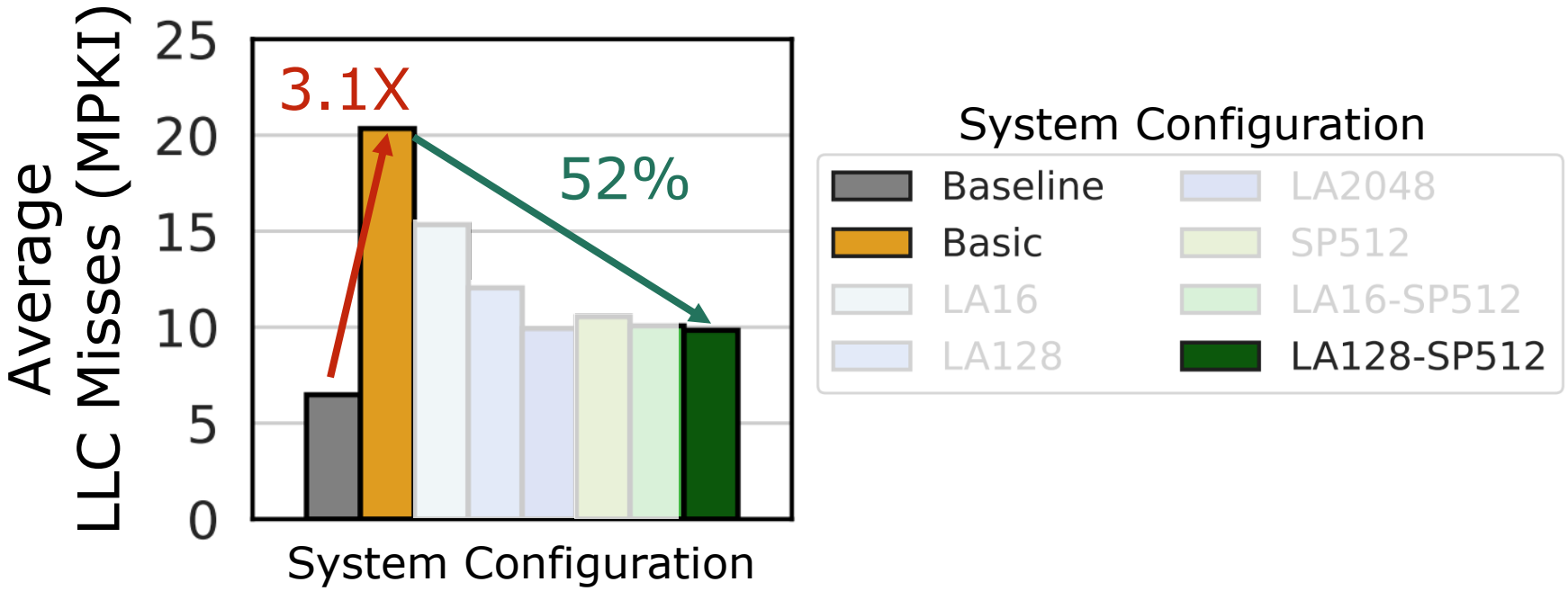
ACT power is dominated by periphery power not affected by the number of sectors activated

Number of Sector Misses

Basic = Sectored DRAM without any sector prediction

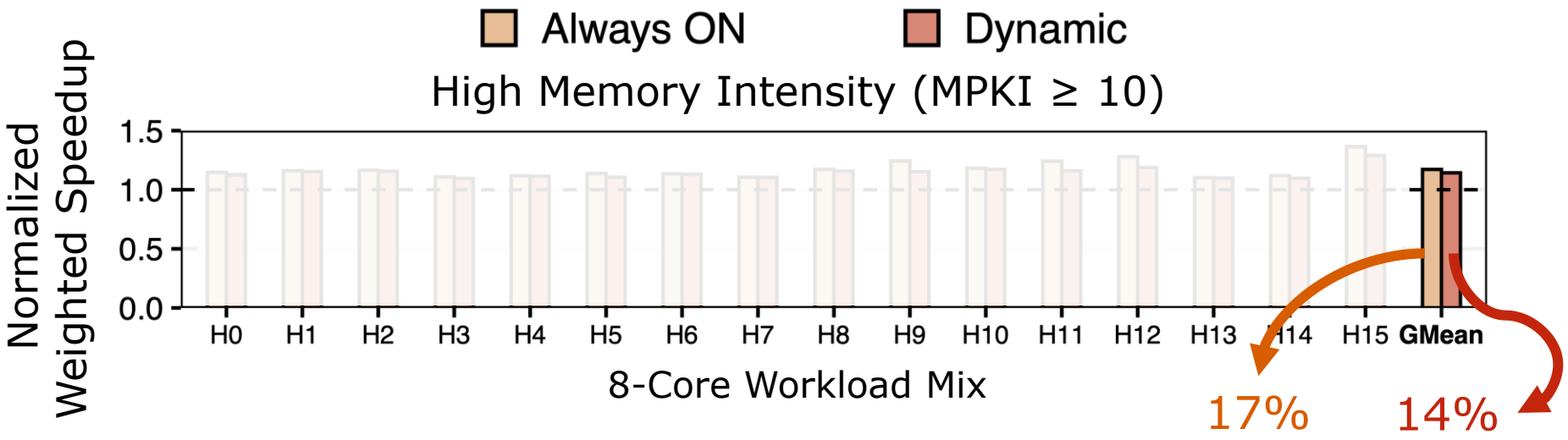
LA<N> = LSQ Lookahead with N LSQ entries

SP512 = Sector Predictor with a history table size of 512



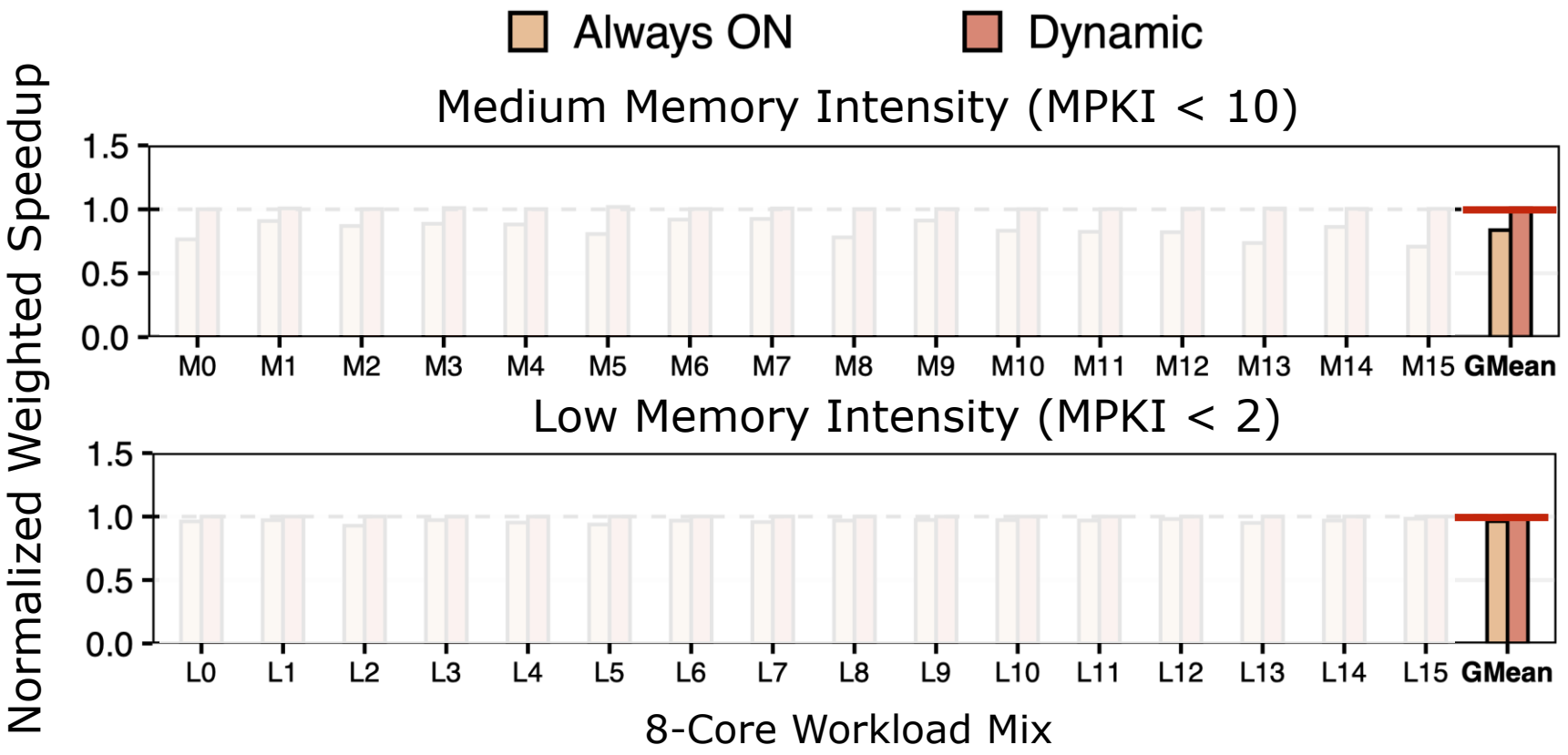
LSQ Lookahead 128 with SP 512
minimizes the LLC misses caused by sector misses

Speedup



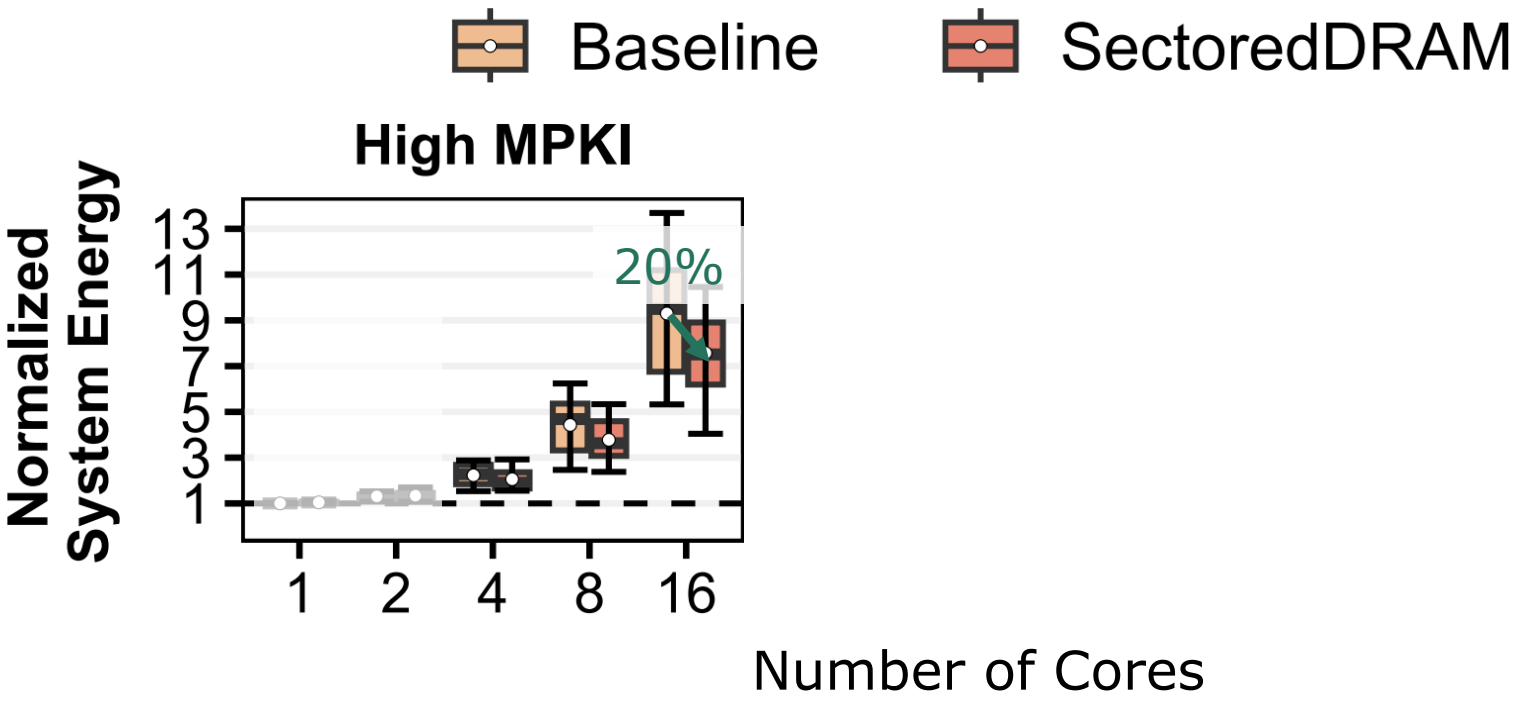
Sectored DRAM provides significant speedups for highly memory intensive workloads

Performance Degradation for Non-Memory-Intensive Workloads



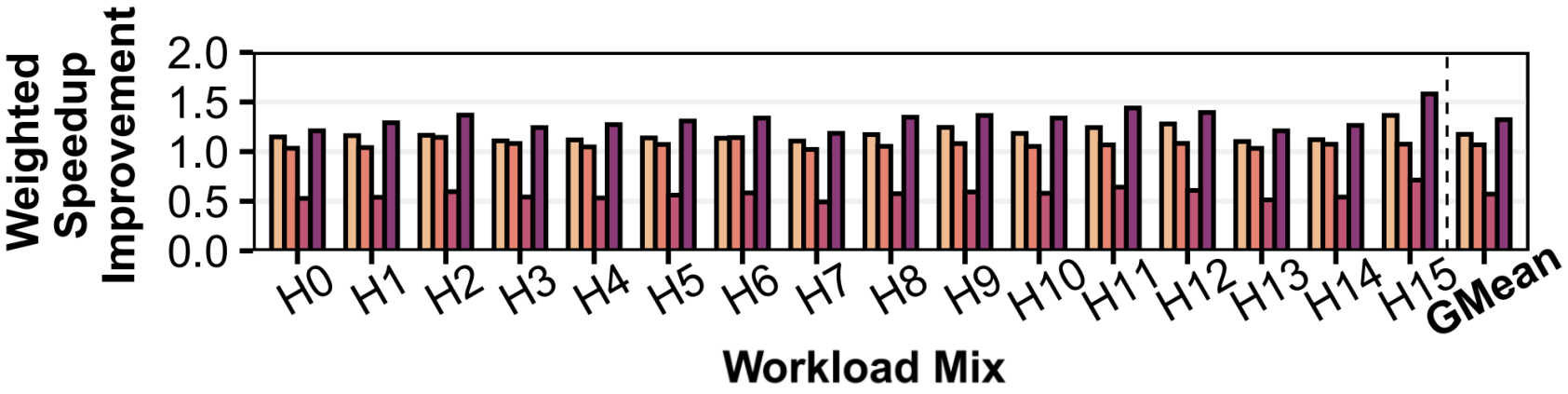
Dynamic policy overcomes the performance degradation in non-memory-intensive workloads

System Energy

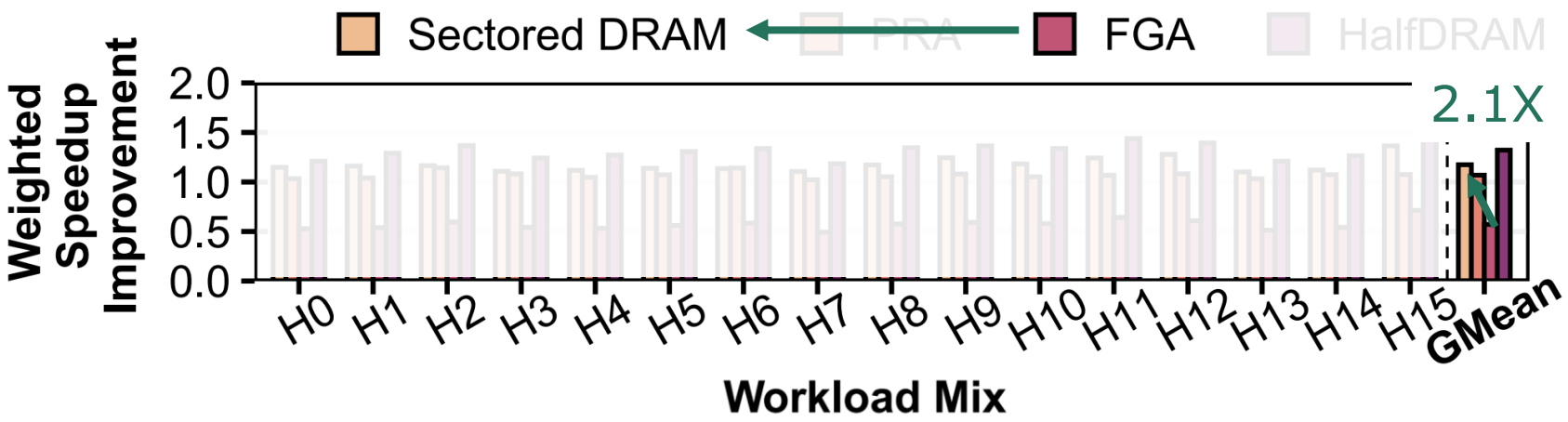


Sectored DRAM provides significant system energy savings for highly memory intensive workloads at core count > 2

Workload Mix Performance Comparison

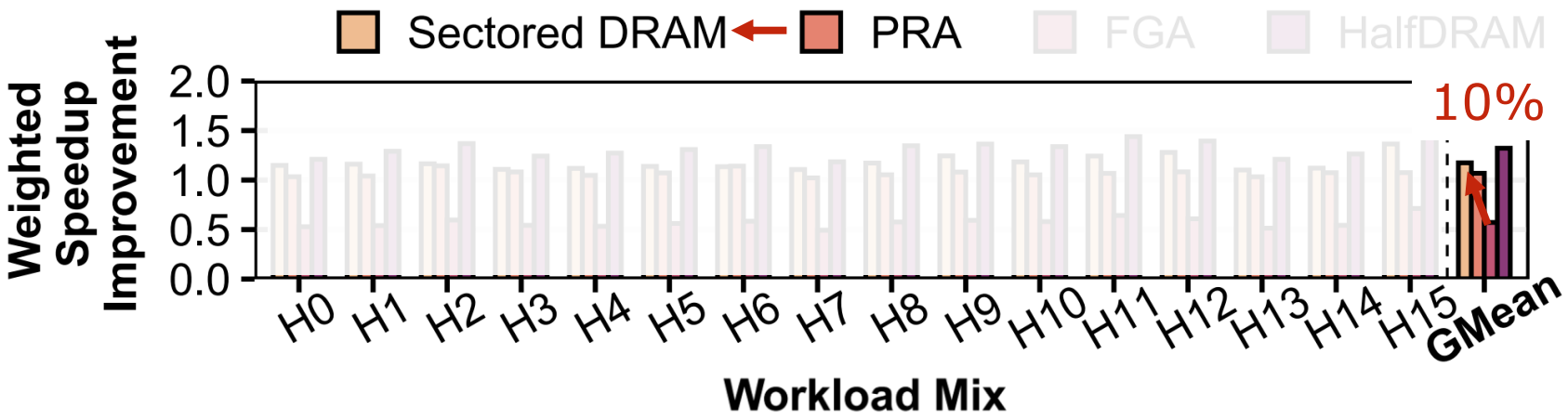


Workload Mix Performance Comparison



Outperforms fine-grained activation by 2.1X

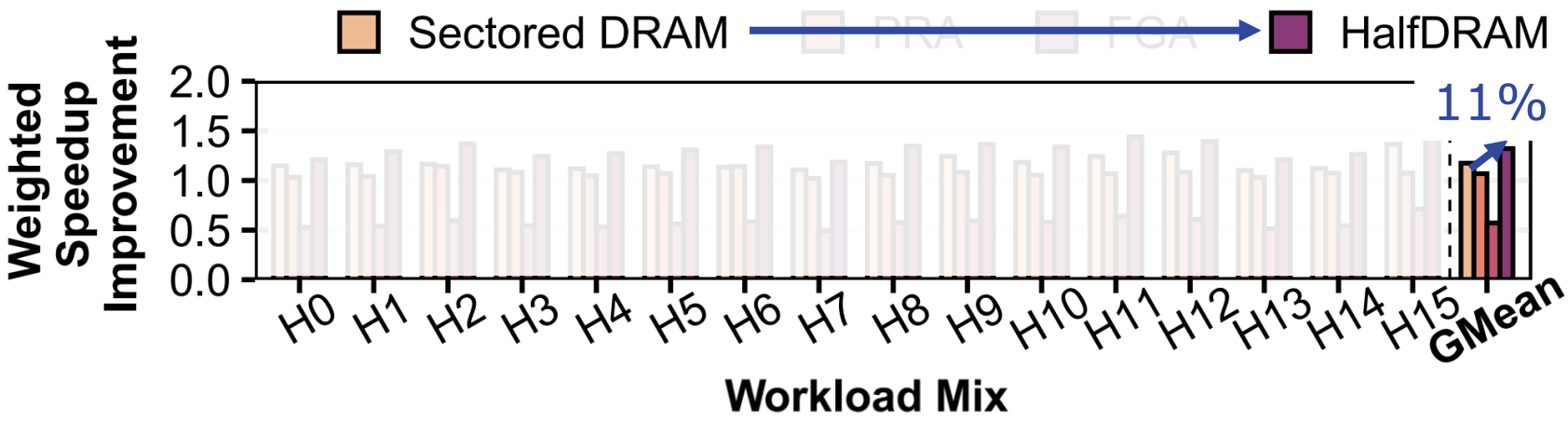
Workload Mix Performance Comparison



Outperforms fine-grained activation by 2.1X

Outperforms Partial Row Activation by 10%

Workload Mix Performance Comparison

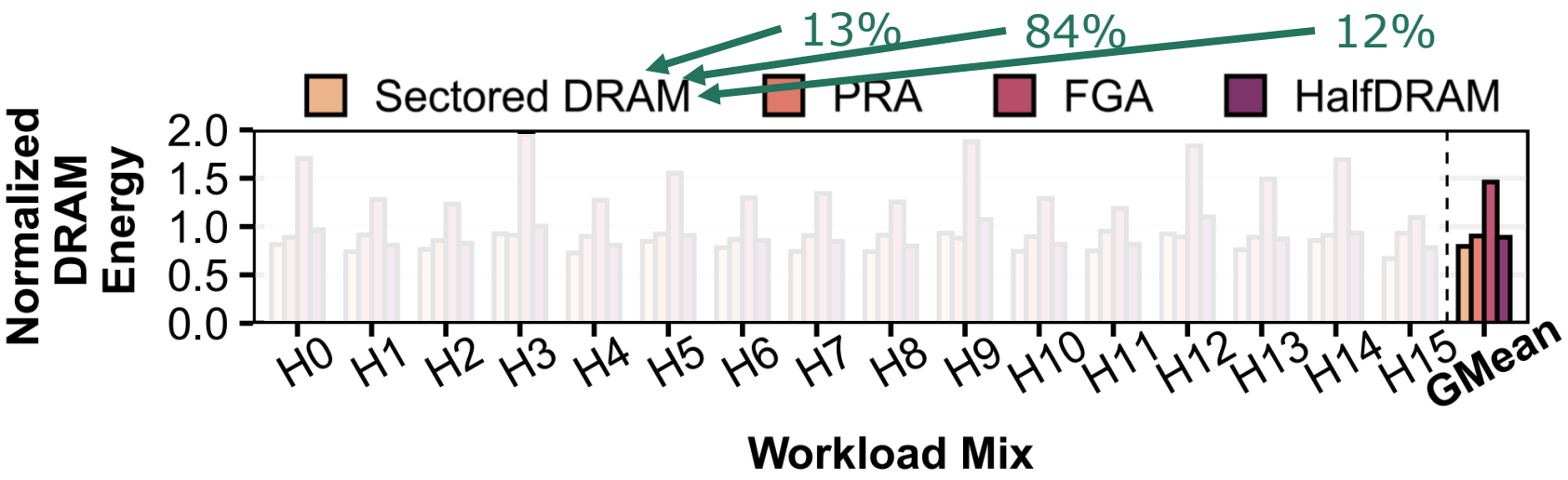


Outperforms fine-grained activation by 2.1X

Outperforms Partial Row Activation by 10%

Performs within 11% of HalfDRAM

Workload Mix DRAM Energy Comparison



Sectored DRAM enables larger DRAM energy savings compared to prior works

Savings are attributed to
i) finer-grained data transfer and activation than HalfDRAM
ii) background power reduction compared to PRA and FGA

Area Overhead Estimation

DRAM

- Sector transistors, sector latches, wiring
- 8 additional local wordline driver stripes
- Model DRAM chip using CACTI
  - Sectored DRAM: 1.7% of DRAM chip area
  - Partial Row Activation and Fine Grained Activation: 1.7%
  - HalfDRAM: 2.6%

Processor

- Sector bits (indicate valid words): 1 byte/cache block
- Sector predictor: 1088 bytes/core
- Model processor storage area overhead using CACTI
  - 8-core processor area increases by 1.2%

More in the Paper

- Microbenchmark performance evaluation
  - Sectored DRAM greatly benefits random access workloads
    - Provides 1.87x parallel speedup over Baseline
  - Adversarial access patterns can reduce performance
    - Incurs 33% performance overhead for a strided access single-core workload

- Performance & energy sensitivity analysis
  - Number of DRAM channels
  - Performance with prefetching enabled

- Discussion on
  - Finer-granularity sector support (i.e., >8 sectors)
  - Compatibility with DRAM Error Correcting Codes

More in the Paper

Sectored DRAM: A Practical Energy-Efficient and High-Performance Fine-Grained DRAM Architecture

Ataberk Olgun[§] F. Nisa Bostancı[§†] Geraldo F. Oliveira[§] Yahya Can Tuğrul[§†] Rahul Bera[§]
A. Giray Yağlıkcı[§] Hasan Hassan[§] Oğuz Ergin[†] Onur Mutlu[§]

[§]*ETH Zürich* [†]*TOBB University of Economics and Technology*

Modern computing systems access data in main memory at coarse granularity *(e.g., at 512-bit cache block granularity). Coarse-grained access leads to wasted energy because the system does* not *use all individually accessed small portions (e.g.,* words, *each of which typically is 64 bits) of a cache block. In modern DRAM-based computing systems, two key coarse-grained access mechanisms lead to wasted energy: large and fixed-size (i) data transfers between DRAM and the memory controller and (ii) DRAM row activations.*

We propose Sectored DRAM, a new, low-overhead DRAM substrate that reduces wasted energy by enabling fine-grained *DRAM data transfer and DRAM row activation. To retrieve only useful data from DRAM, Sectored DRAM exploits the observation that many cache blocks are not fully utilized in many workloads due to poor spatial locality. Sectored DRAM predicts the words in a cache block that will likely be accessed during the cache block's*

1. Introduction

DRAM [22] is hierarchically organized to improve scaling in density and performance. At the highest level of the hierarchy, a DRAM chip is partitioned into banks that can be accessed simultaneously [87, 57, 58, 59, 63]. At the lowest level, a collection of DRAM rows (DRAM cells that are activated together) are typically divided into multiple *DRAM mats* that can operate individually [52, 42, 125, 58]. Even though DRAM chips are hierarchically organized, standard DRAM interfaces (e.g., DDRx [43, 44, 45]) do *not* expose DRAM mats to the memory controller. To access even a single DRAM cell, the memory controller needs to activate a large number of DRAM cells (e.g., 65,536 DRAM cells in a DRAM row in DDR4 [80]) and transfer many bits (e.g., a cache block, typically 512 bits [32]) over the memory channel. Thus, in current systems, both DRAM data transfer and activation are *coarse-grained.* Coarse-grained data

https://arxiv.org/pdf/2207.13795.pdf

Outline

1. Background & Motivation
2. Sectored DRAM: Design
3. Sectored DRAM: System Integration
4. Evaluation
5. Conclusion

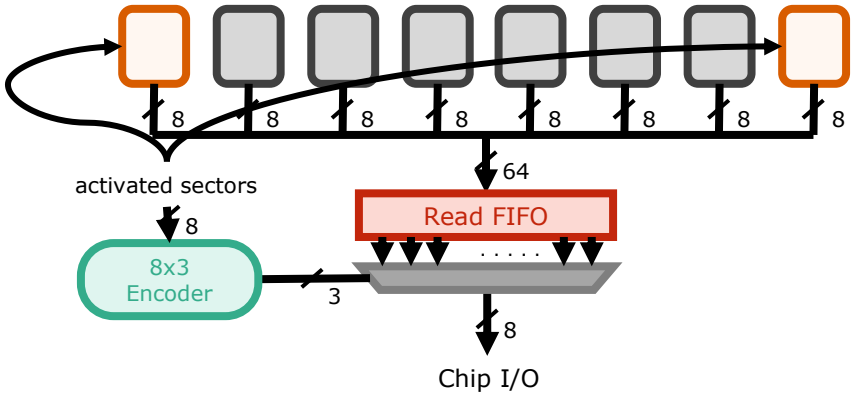
Sectored DRAM Conclusion

Designed a fine-grained, low-cost, and high-throughput DRAM substrate

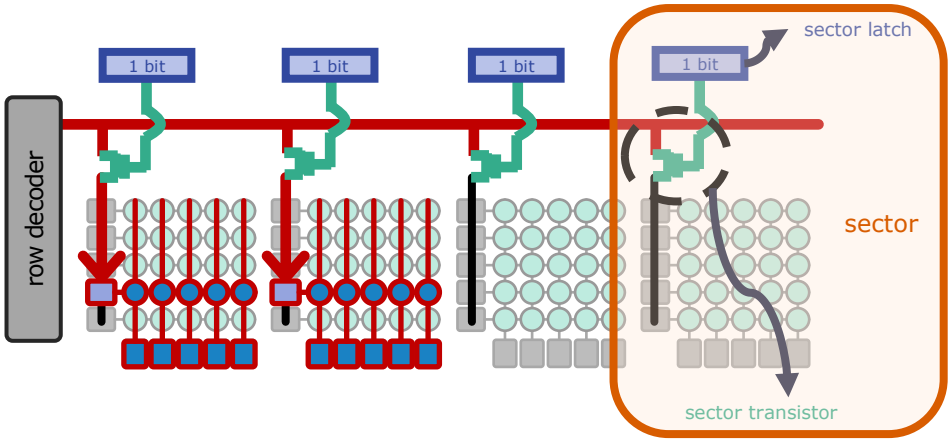
- Mitigates excessive energy consumption of coarse-grained DRAM

Key Ideas: Small modifications to memory controller and DRAM chip enable

Variable Burst Length



Sectored Activation

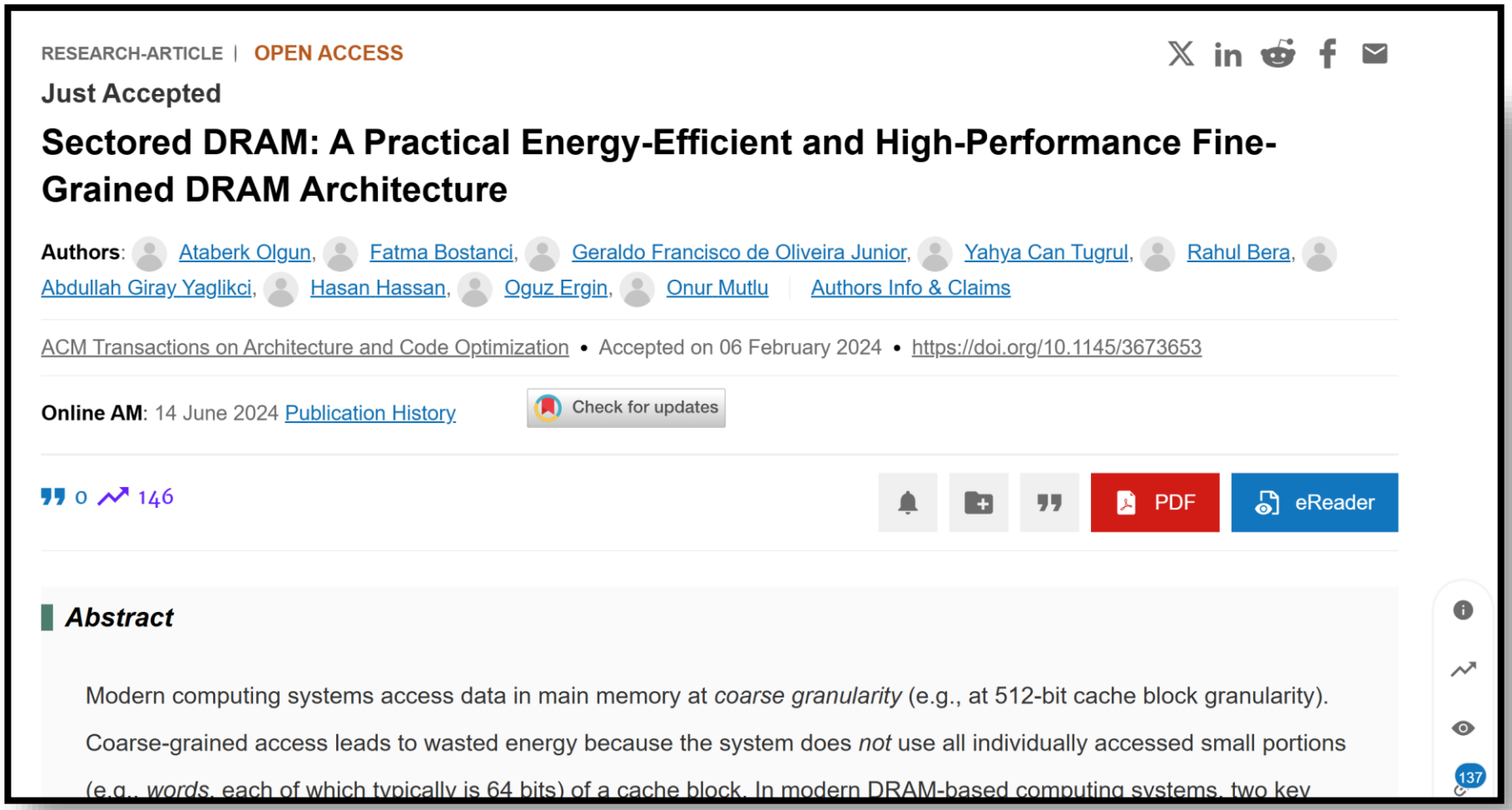


Key Results: For the evaluated memory-intensive workloads, Sectored DRAM

- Improves system energy consumption by 14%, system performance by 17%
- Incurs 0.39 mm^2 (1.7%) DRAM chip area overhead
- Performs within 11% of a state-of-the-art prior work (Half-DRAM), with 12% less DRAM energy and 34% less area overhead

Sectored DRAM is Published in ACM TACO

https://dl.acm.org/doi/abs/10.1145/3673653



RESEARCH-ARTICLE | OPEN ACCESS

Just Accepted

Sectored DRAM: A Practical Energy-Efficient and High-Performance Fine-Grained DRAM Architecture

Authors: Ataberk Olgun, Fatma Bostanci, Geraldo Francisco de Oliveira Junior, Yahya Can Tugrul, Rahul Bera, Abdullah Giray Yaglikci, Hasan Hassan, Oguz Ergin, Onur Mutlu | Authors Info & Claims

ACM Transactions on Architecture and Code Optimization • Accepted on 06 February 2024 • https://doi.org/10.1145/3673653

Online AM: 14 June 2024 Publication History

Check for updates

0 146

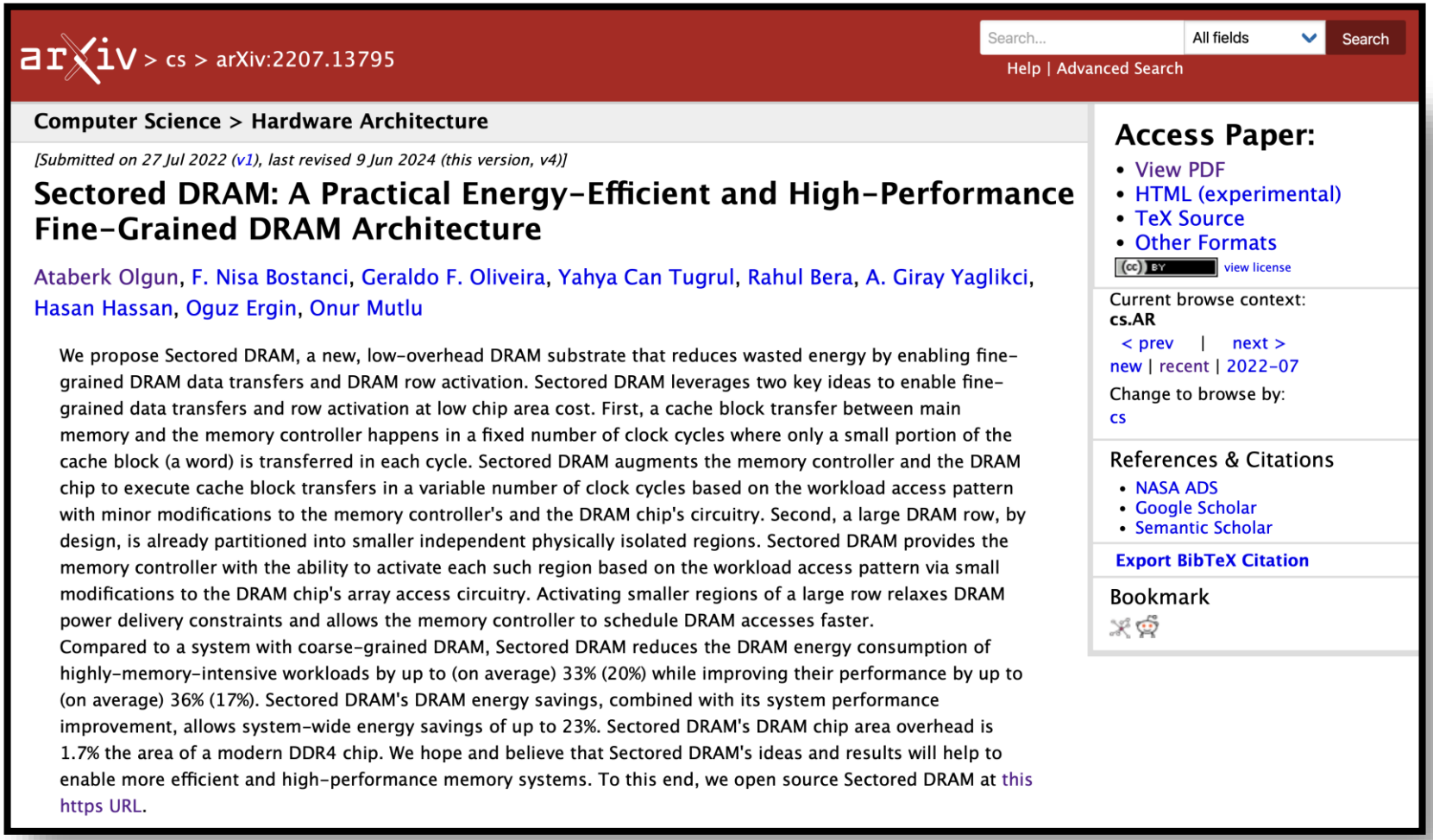
PDF eReader

Abstract

Modern computing systems access data in main memory at *coarse granularity* (e.g., at 512-bit cache block granularity). Coarse-grained access leads to wasted energy because the system does *not* use all individually accessed small portions (e.g., *words*, each of which typically is 64 bits) of a cache block. In modern DRAM-based computing systems, two key

Extended Version on Arxiv

https://arxiv.org/pdf/2207.13795.pdf



arXiv > cs > arXiv:2207.13795

Search... All fields Search

Help | Advanced Search

Computer Science > Hardware Architecture

[Submitted on 27 Jul 2022 (v1), last revised 9 Jun 2024 (this version, v4)]

Sectored DRAM: A Practical Energy-Efficient and High-Performance Fine-Grained DRAM Architecture

Ataberk Olgun, F. Nisa Bostanci, Geraldo F. Oliveira, Yahya Can Tugrul, Rahul Bera, A. Giray Yaglikci, Hasan Hassan, Oguz Ergin, Onur Mutlu

We propose Sectored DRAM, a new, low-overhead DRAM substrate that reduces wasted energy by enabling fine-grained DRAM data transfers and DRAM row activation. Sectored DRAM leverages two key ideas to enable fine-grained data transfers and row activation at low chip area cost. First, a cache block transfer between main memory and the memory controller happens in a fixed number of clock cycles where only a small portion of the cache block (a word) is transferred in each cycle. Sectored DRAM augments the memory controller and the DRAM chip to execute cache block transfers in a variable number of clock cycles based on the workload access pattern with minor modifications to the memory controller's and the DRAM chip's circuitry. Second, a large DRAM row, by design, is already partitioned into smaller independent physically isolated regions. Sectored DRAM provides the memory controller with the ability to activate each such region based on the workload access pattern via small modifications to the DRAM chip's array access circuitry. Activating smaller regions of a large row relaxes DRAM power delivery constraints and allows the memory controller to schedule DRAM accesses faster.
Compared to a system with coarse-grained DRAM, Sectored DRAM reduces the DRAM energy consumption of highly-memory-intensive workloads by up to (on average) 33% (20%) while improving their performance by up to (on average) 36% (17%). Sectored DRAM's DRAM energy savings, combined with its system performance improvement, allows system-wide energy savings of up to 23%. Sectored DRAM's DRAM chip area overhead is 1.7% the area of a modern DDR4 chip. We hope and believe that Sectored DRAM's ideas and results will help to enable more efficient and high-performance memory systems. To this end, we open source Sectored DRAM at this https URL.

Access Paper:

- View PDF
- HTML (experimental)
- TeX Source
- Other Formats

(cc) BY view license

Current browse context: **cs.AR**

< prev | next >

new | recent | 2022-07

Change to browse by: cs

References & Citations

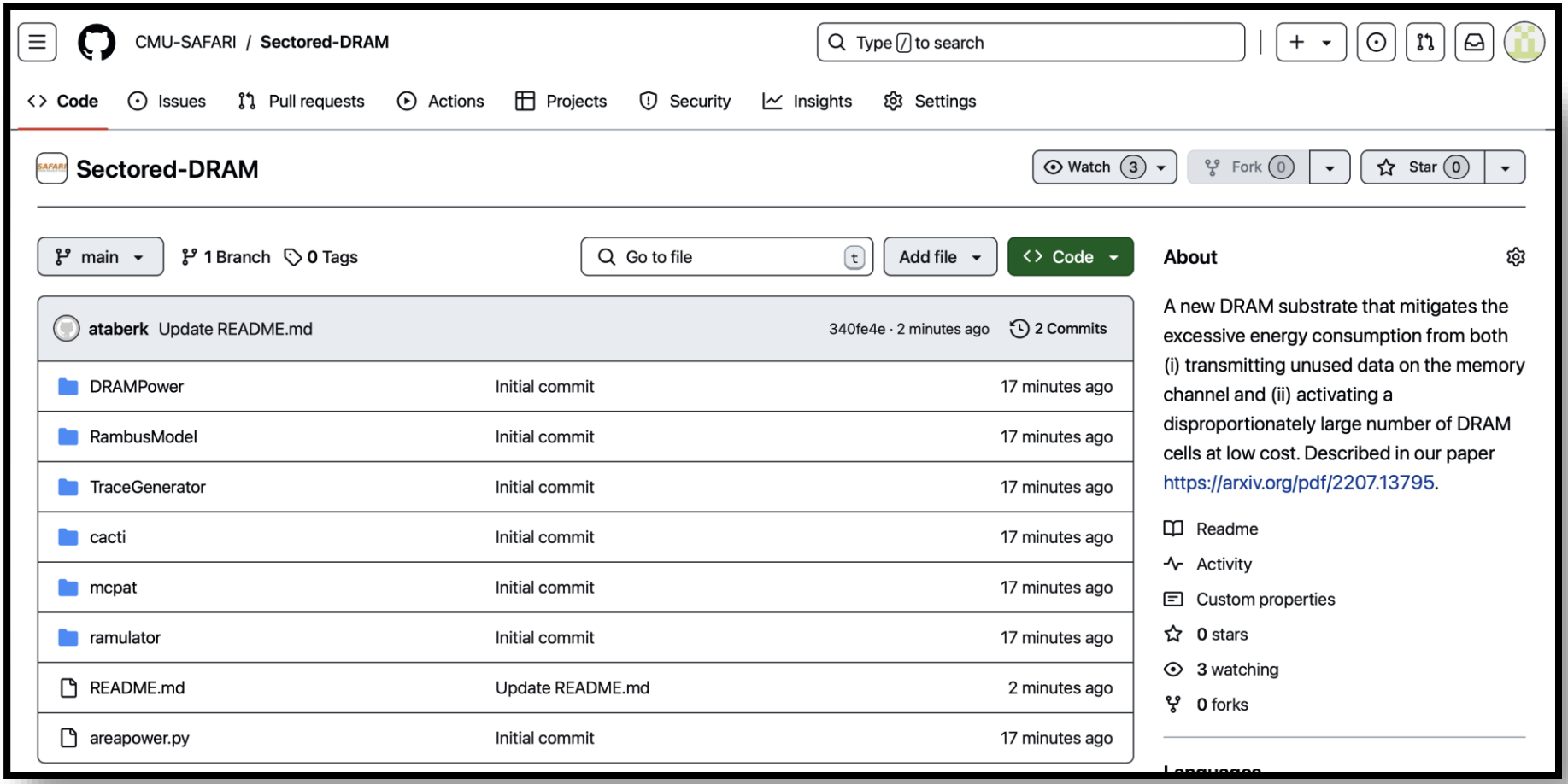
- NASA ADS
- Google Scholar
- Semantic Scholar

Export BibTeX Citation

Bookmark

Sectored DRAM is Open Source

https://github.com/CMU-SAFARI/Sectored-DRAM



Sectored DRAM
A Practical Energy-Efficient and High-Performance Fine-Grained DRAM Architecture

Paper

GitHub

Ataberk Olgun
olgunataberk@gmail.com

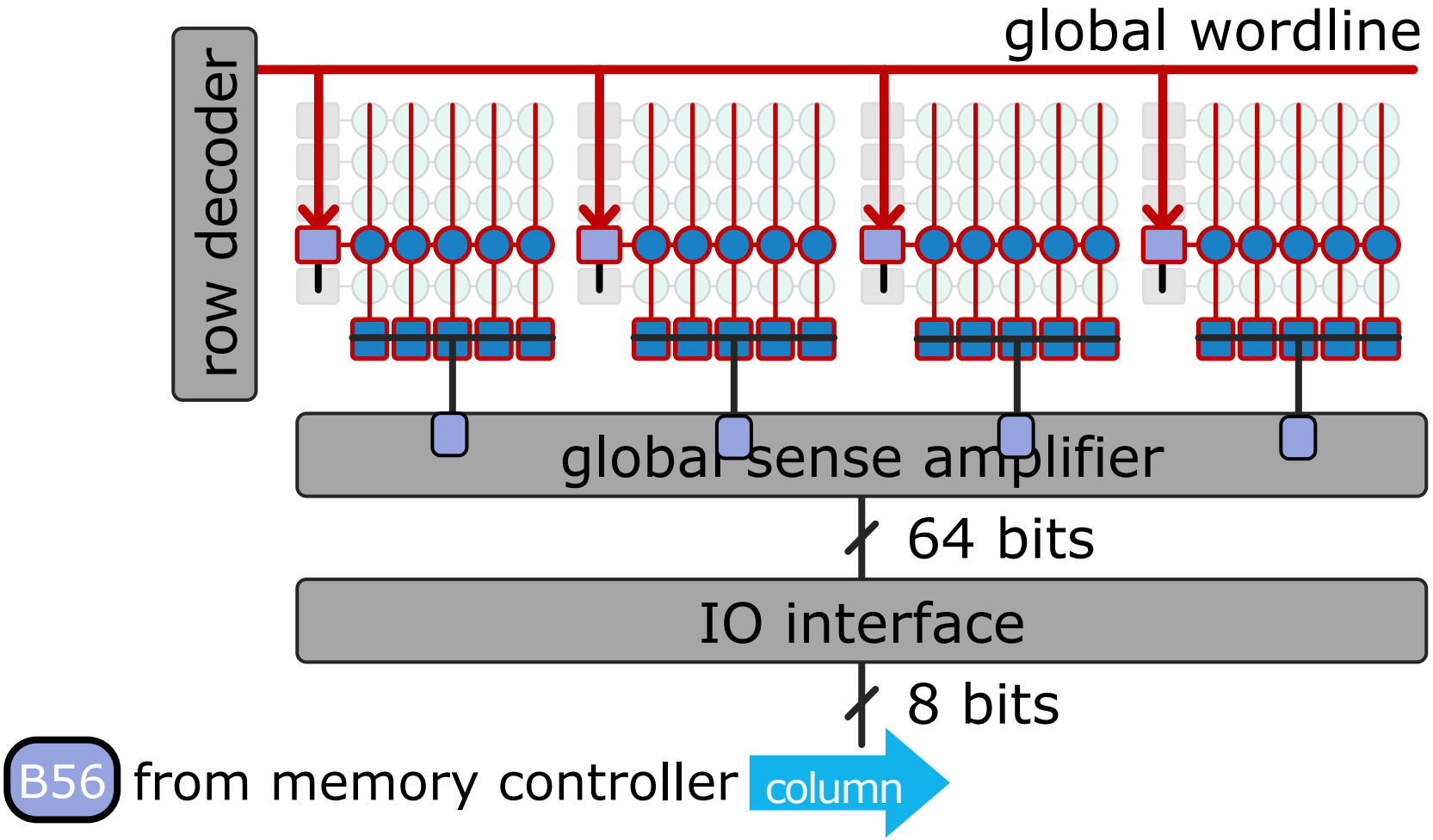
F. Nisa Bostanci Geraldo F. Oliveira Yahya Can Tugrul Rahul Bera
A. Giray Yaglikci Hasan Hassan Oguz Ergin Onur Mutlu

ETH zürich SAFARI kasırga TOBB ETÜ Ekonomi ve Teknoloji Üniversitesi

Backup Slides

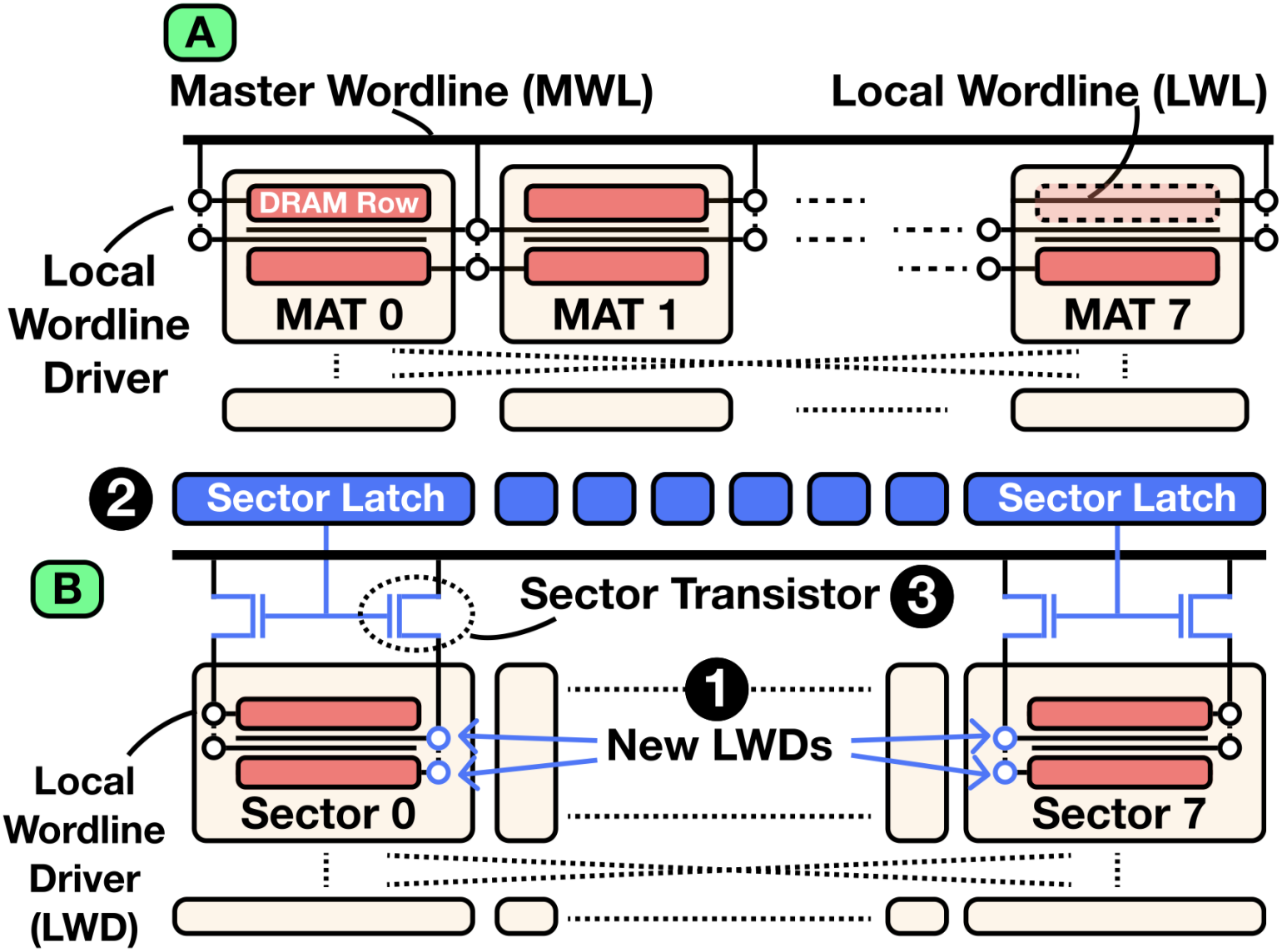
DRAM Data Transfer (II)

- Bits of a burst split across DRAM mats



[Oliveira+, HPCA'24]

Sectored DRAM Subarray Organization



Exposing Sectored DRAM to the Memory Controller with No Interface Modifications

1 Sectored Activation (SA)

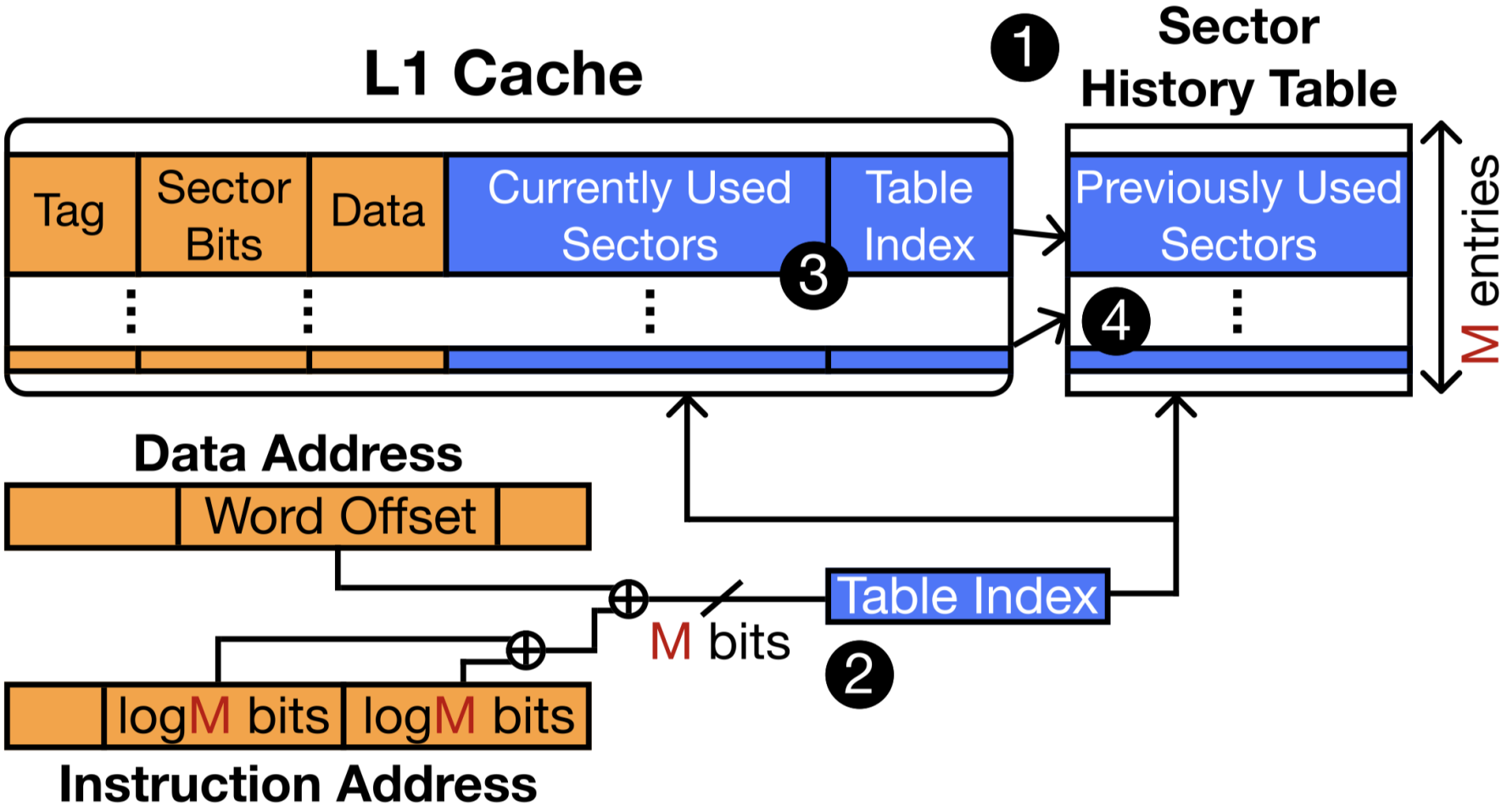
- More than 10 unused bits in precharge (PRE) command encoding
- Determine the sectors opened for the next activate (ACT) command

Activating fewer than all 8 sectors relaxes power constraints allows for higher ACT command throughput

2 Variable Burst Length (VBL)

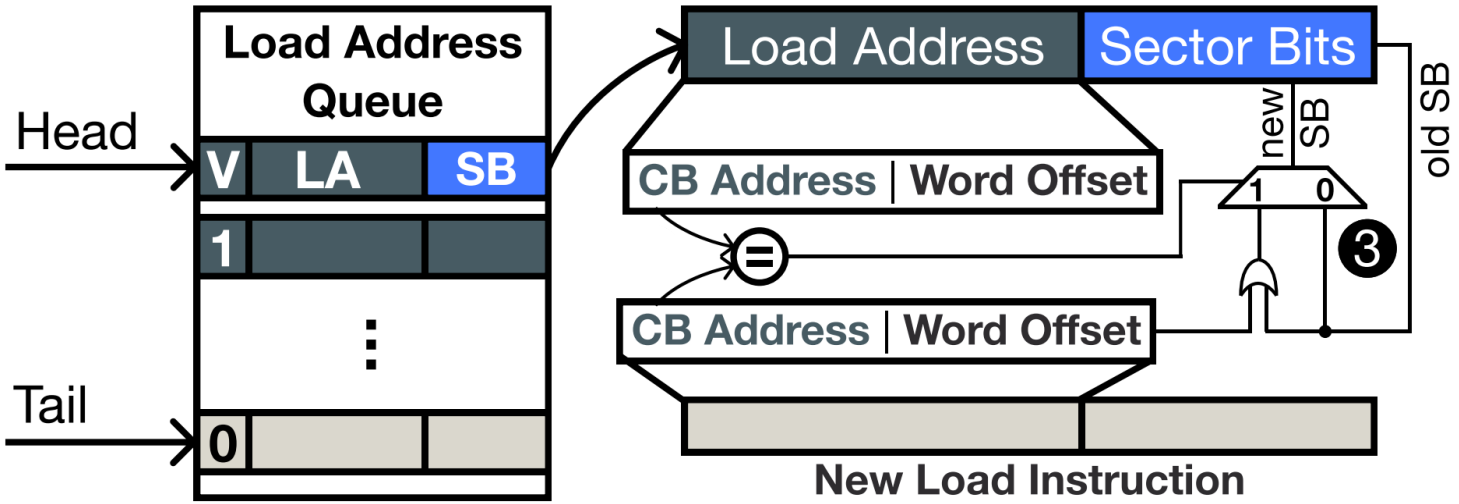
- DRAM and memory controller must agree on burst length
- DRAM and memory controller store sector bits for each bank
- Low overhead popcount circuitry to count set (logic-1) sector bits

Sector Predictor



Load/Store Queue (LSQ) Lookahead

- One load/store instruction *references* one word in main memory
- **Key Mechanism:** 1) Collect references from *younger* load/store instructions
  2) store the collected references in the *oldest* load/store instr.



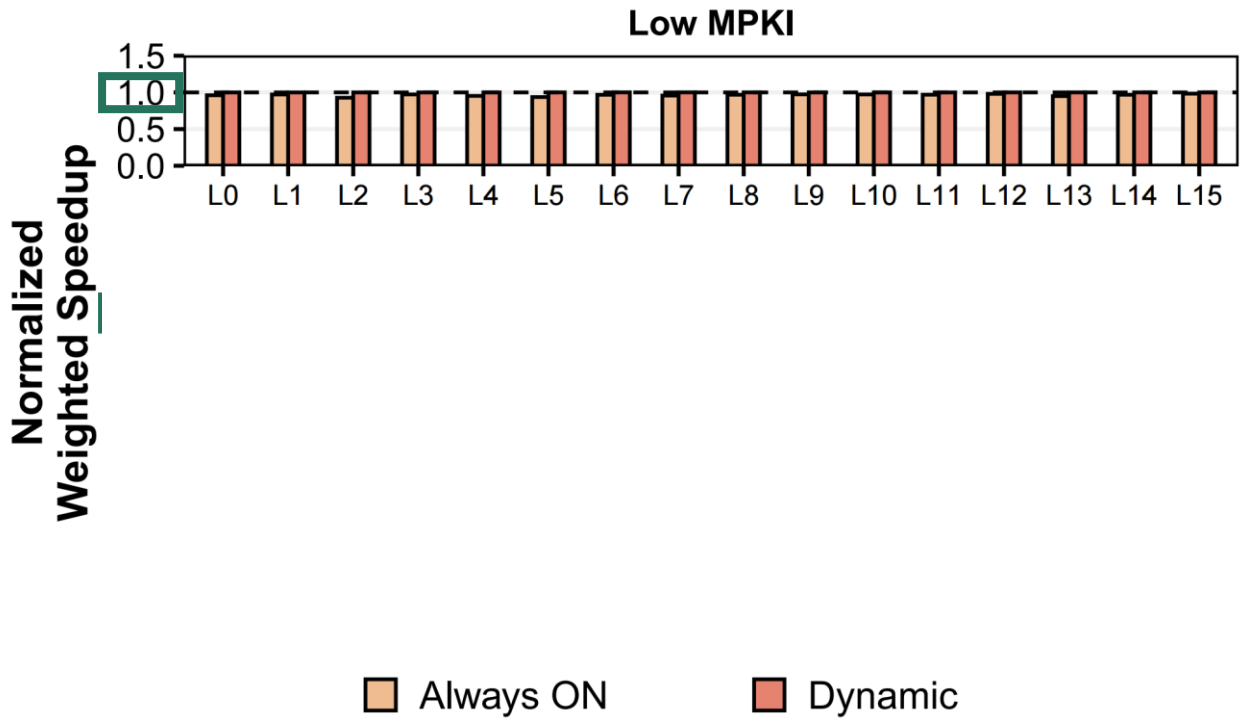
A load/store instruction retrieves all words in a cache block that will be referenced in the near future to the L1 cache with only one cache access

Evaluated Workloads

LLC MPKI	Workloads
≥ 10 (High)	ligraPageRank, mcf-2006, libquantum-2006, gobmk-2006, ligraMIS, GemsFDTD-2006, bwaves-2006, lbm-2006, lbm -2017, hashjoinPR
1..10 (Medium)	omnetpp-2006, gcc-2017, mcf-2017, cactusADM-2006, zeusmp-2006, xalancbmk-2006, ligraKCore, astar-2006, cactus-2017, parest-2017, ligraComponents
≤ 1 (Low)	splash2Ocean, tonto-2006, xz-2017, wrf-2006, bzip2-2006, xalancbmk-2017, h264ref-2006, hmmer-2006, namd-2017, blender-2017, sjeng-2006, perlbench-2006, x264-2017, deepsjeng-2017, gromacs-2006, gcc-2006, imagick-2017, leela-2017, povray-2006, calculix-2006

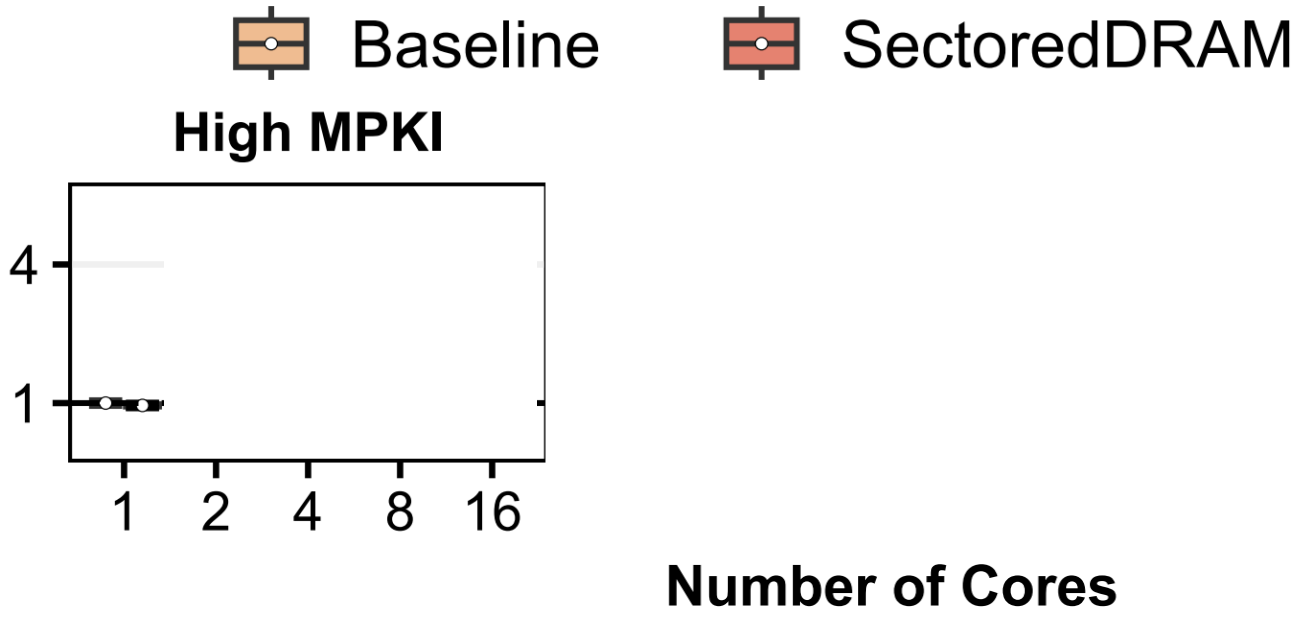
Performance Degradation for Non-Memory-Intensive Workloads

- Fetch all sectors of a cache block if the workload access pattern does not favor sub-cache-block data transfers
  - Based on average MPKI and thresholding

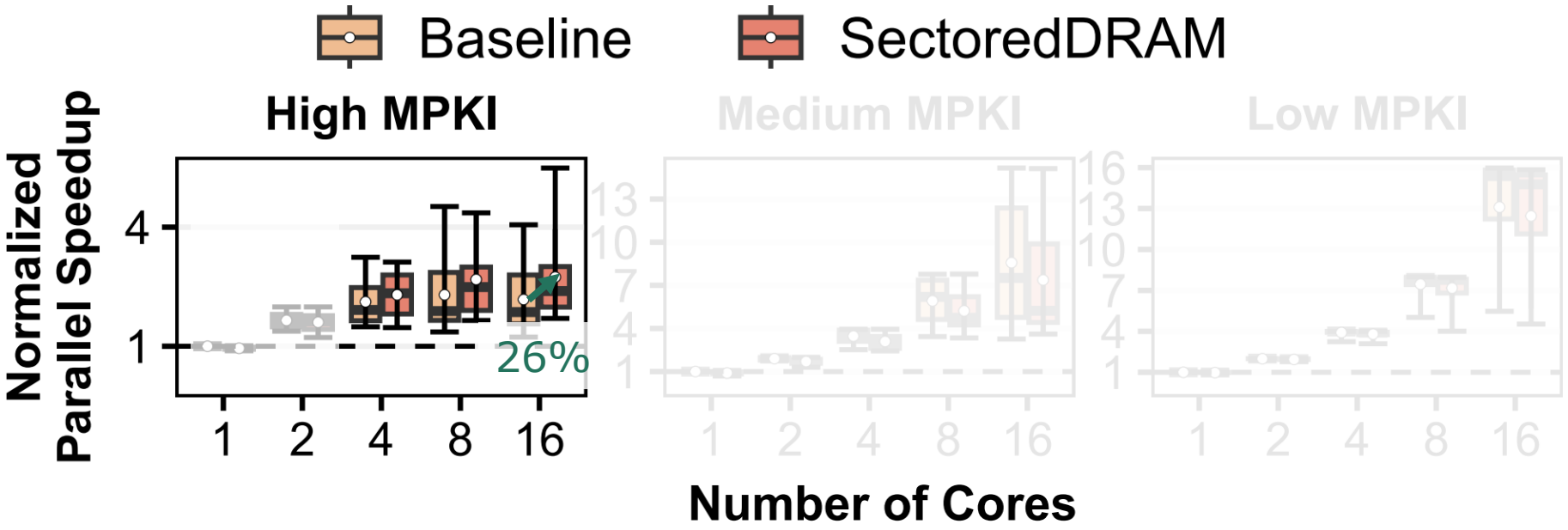


Dynamic policy overcomes the performance degradation in non-memory-intensive workloads

Speedup

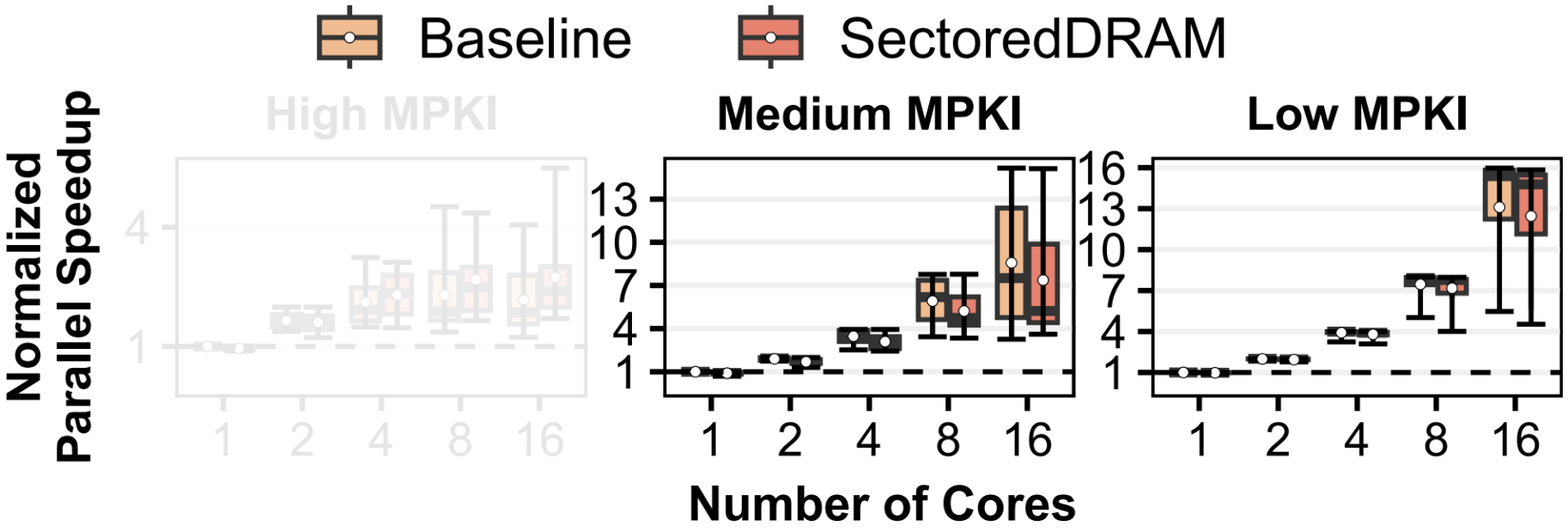


Speedup



Sectored DRAM provides significant speedups
for highly memory intensive workloads at core count > 2

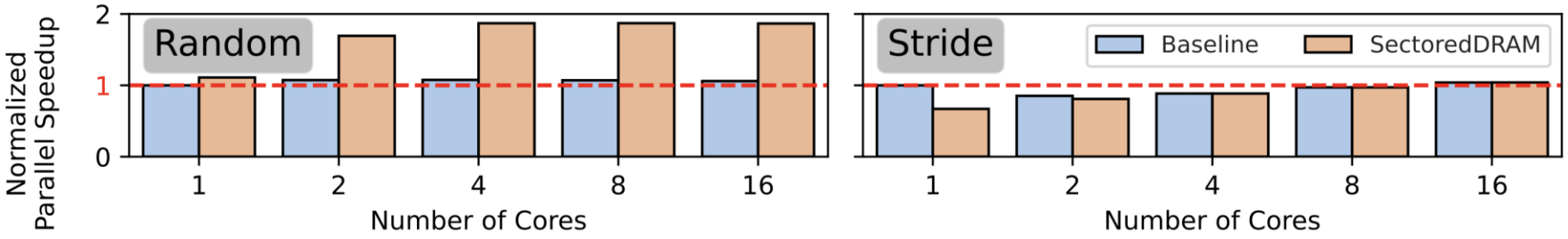
Speedup



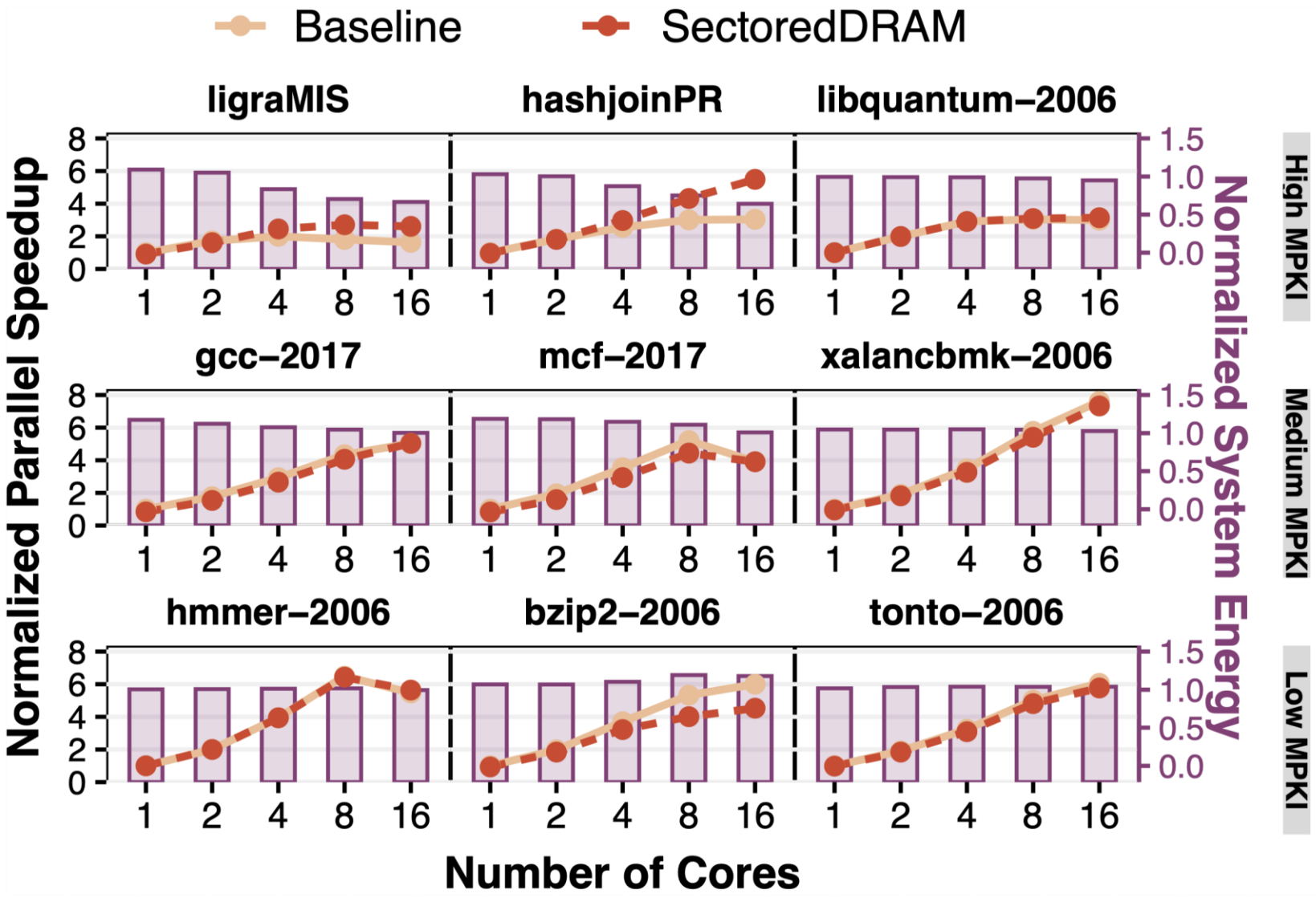
Sectored DRAM provides significant speedups for highly memory intensive workloads at core count > 2

Sectored DRAM provides smaller parallel speedup than Baseline for non-memory-intensive workloads

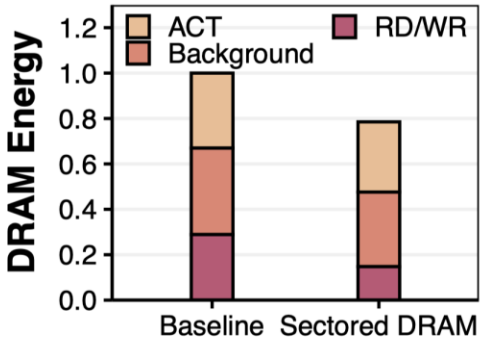
Microbenchmark Performance

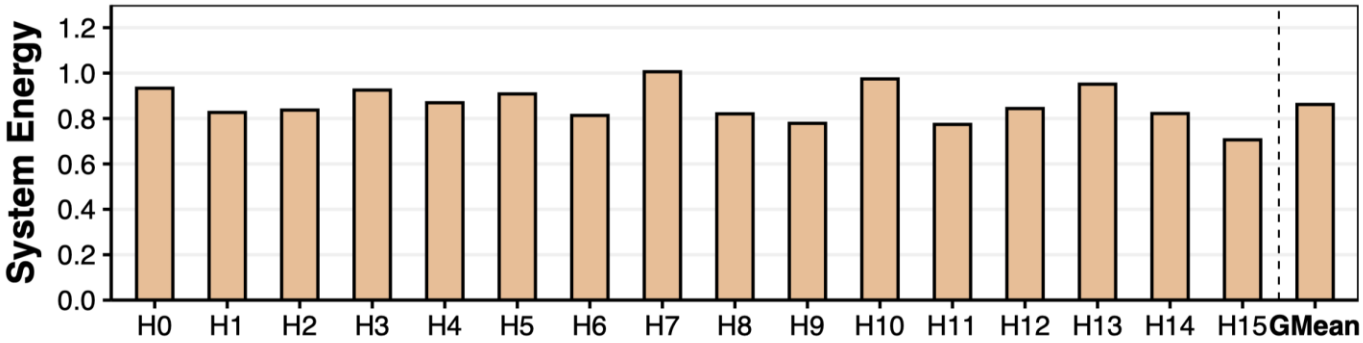


Parallel Speedup and System Energy per Workload

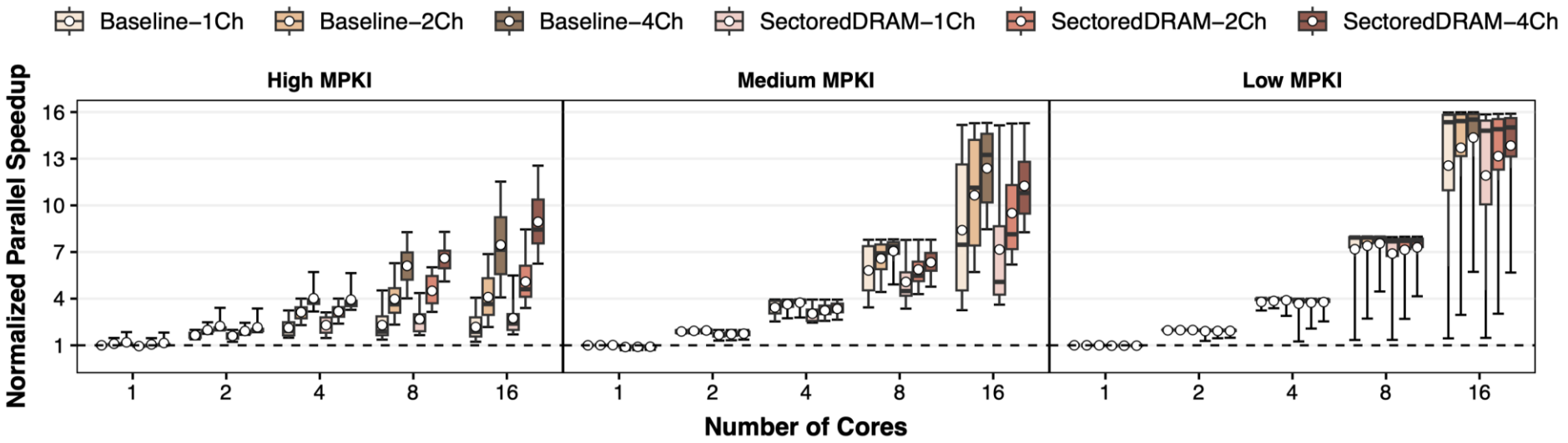


DRAM Energy Breakdown and System Energy

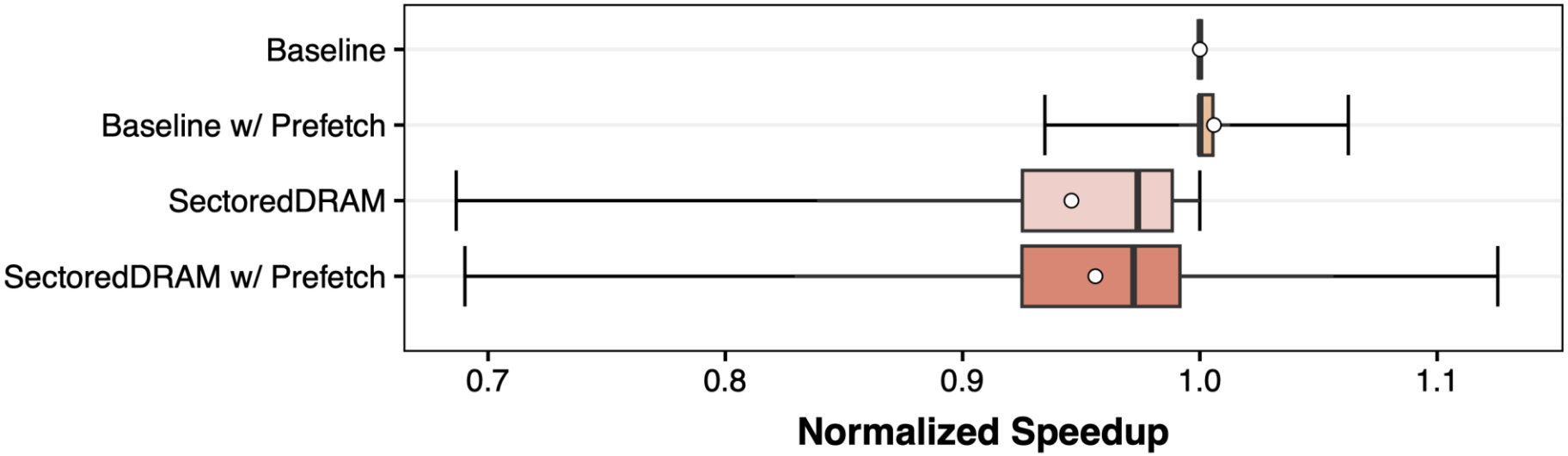




Performance Sensitivity to Number of Channels



Sectored DRAM with Prefetching



Enabling Higher Row Activation Rate

- tFAW = 25 nanoseconds (ns)
- 32 sectors can be activated in a tFAW
- Only 10 activate commands can be issued in 25 ns due to tRRD_L and tRRD_S

- 10 ACT, each of which activate one sector takes 20% less power than
  4 ACT, each of which activates 8 sectors

Sectored DRAM vs Module-Level Mechanisms

- DRAM interface modifications vs. DRAM chip modifications

- Low overhead module-level mechanism induces 23% overhead where Sectored DRAM provides 17% speedup
  - Command bus becomes the bottleneck
  - Alleviating command bus bottleneck is area expensive

- System integration heavily inspired by DGMS