

PuDHammer

Experimental Analysis of Read Disturbance Effects of Processing-using-DRAM in Real DRAM Chips

İsmail Emir Yüksel

Akash Sood Ataberk Olgun Oğuzhan Canpolat Haocong Luo

Nisa Bostancı Mohammad Sadrosadati Giray Yağlıkçı Onur Mutlu

SAFARI

ETH zürich

Executive Summary

Motivation: Processing-using-DRAM (PuD) alleviates data movement bottlenecks

- DRAM can perform **many** PuD operations by activating **multiple DRAM rows** in quick succession or simultaneously (i.e., **multiple-row activation**)
- Modern DRAM is subject to **read disturbance** (e.g., RowHammer)
 - Repeatedly activating **even a single** row induces **bitflips** in **unaccessed** rows

Problem: No prior work studies the read disturbance effects of multiple-row activation

Goal: Understand and analyze read disturbance effects of **multiple-row activation**

Experimental Study: 316 COTS DDR4 chips from **four** major manufacturers to study read disturbance effects of multiple-row activation, which we call **PuDHammer**

- PuDHammer significantly **exacerbates** DRAM read disturbance, causing up to **158.58x reduction** in min. hammer count to induce first bitflip
- PuDHammer bypasses in-DRAM RowHammer mitigation in DDR4 and induces **11340x** more bitflips than RowHammer

Mitigation: We adapt Per Row Activation Counting (PRAC) for PuDHammer

- Adapted PRAC solution incurs an average system performance overhead of **48.26%**

Outline

Background

Problem & Goal

Testing Infrastructure

Read Disturbance Effect of CoMRA

Read Disturbance Effect of SiMRA

Mitigating PuDHammer

Conclusion

Outline

Background

Problem & Goal

Testing Infrastructure

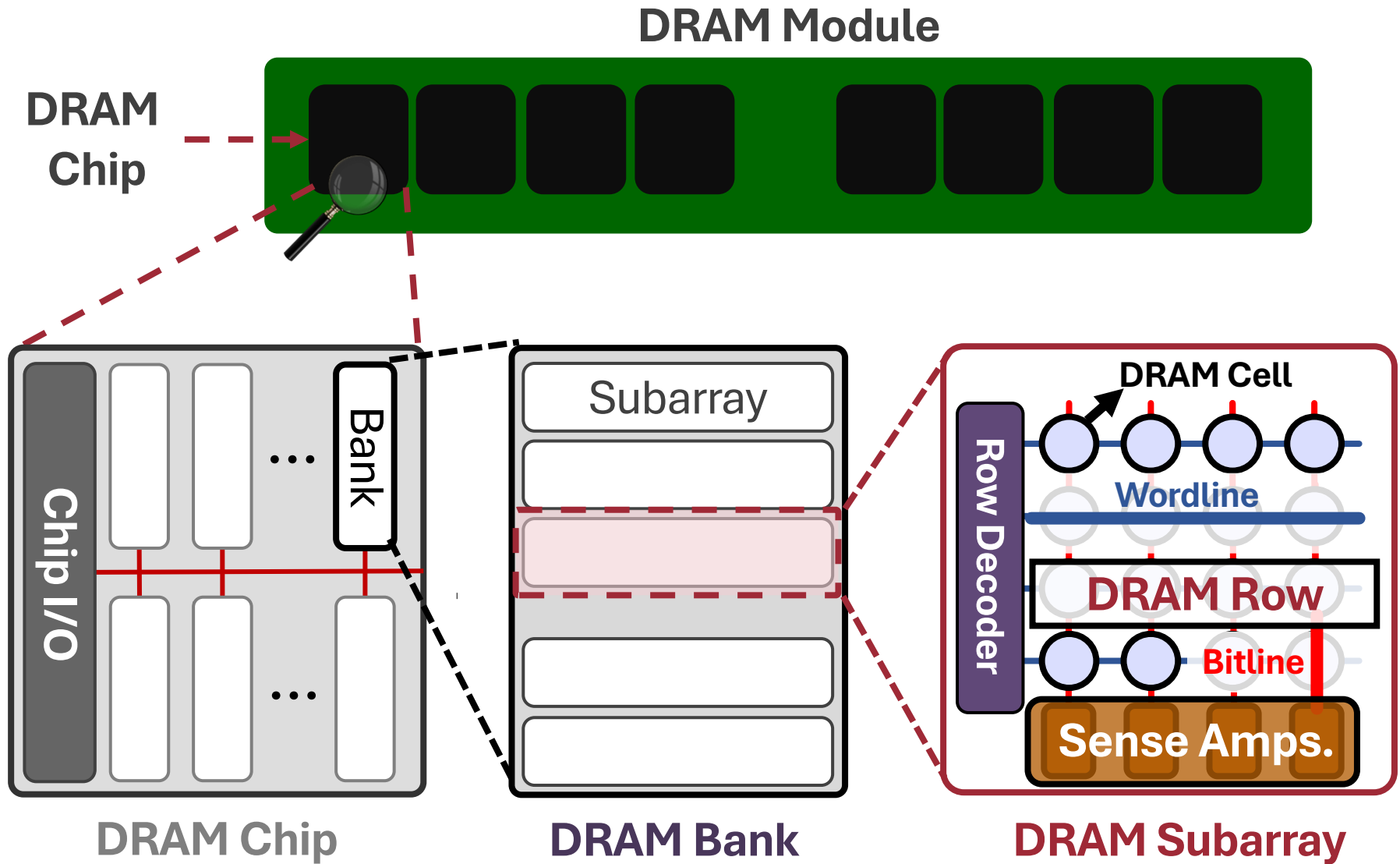
Read Disturbance Effect of CoMRA

Read Disturbance Effect of SiMRA

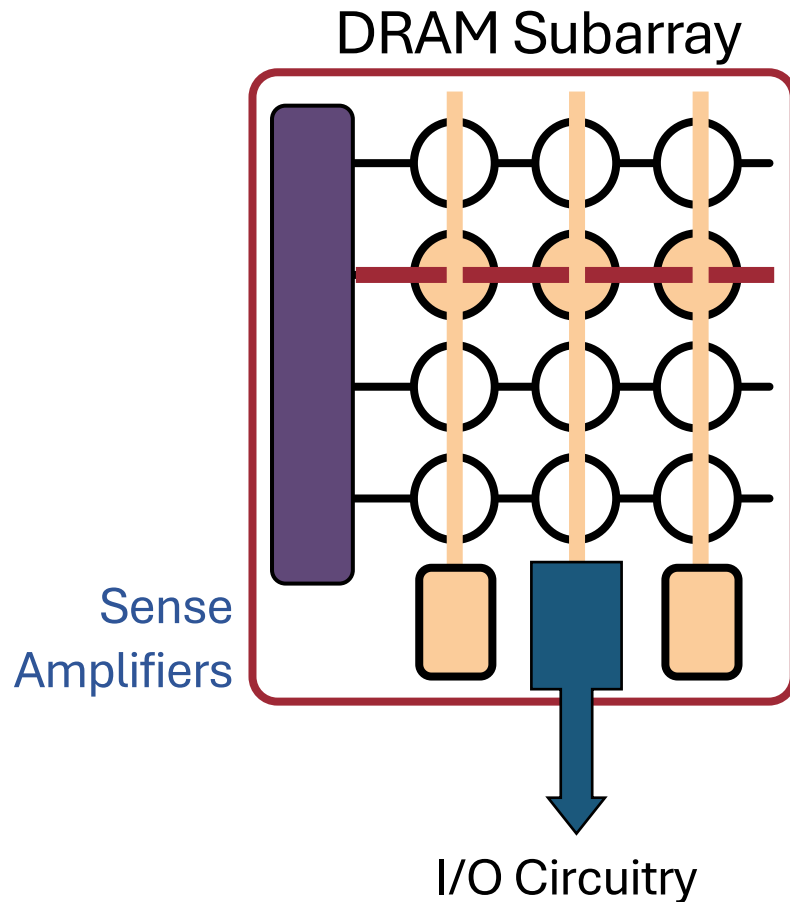
Mitigating PuDHammer

Conclusion

DRAM Organization



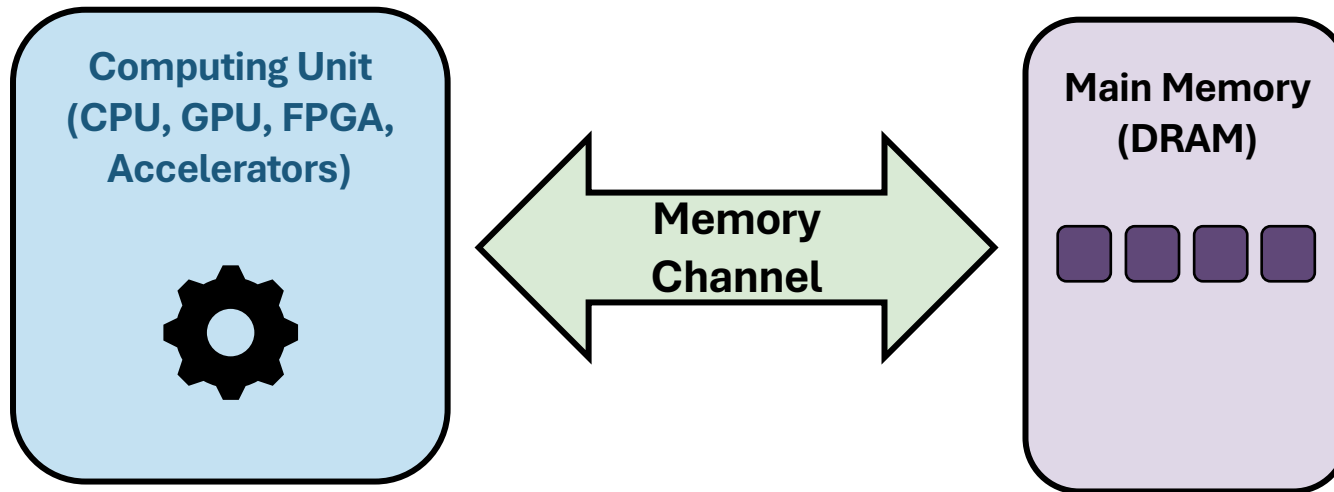
DRAM Operation



- 1 ACTIVATE (ACT):**
Fetch the row's content into the **sense amplifiers**
- 2 Column Access (RD/WR):**
Read/Write the target column and drive to I/O
- 3 PRECHARGE (PRE):**
Prepare the bank for a new ACTIVATE

Data Movement Bottleneck

- Today's computing systems are processor centric
- All data is processed in the processor → **at great system cost**

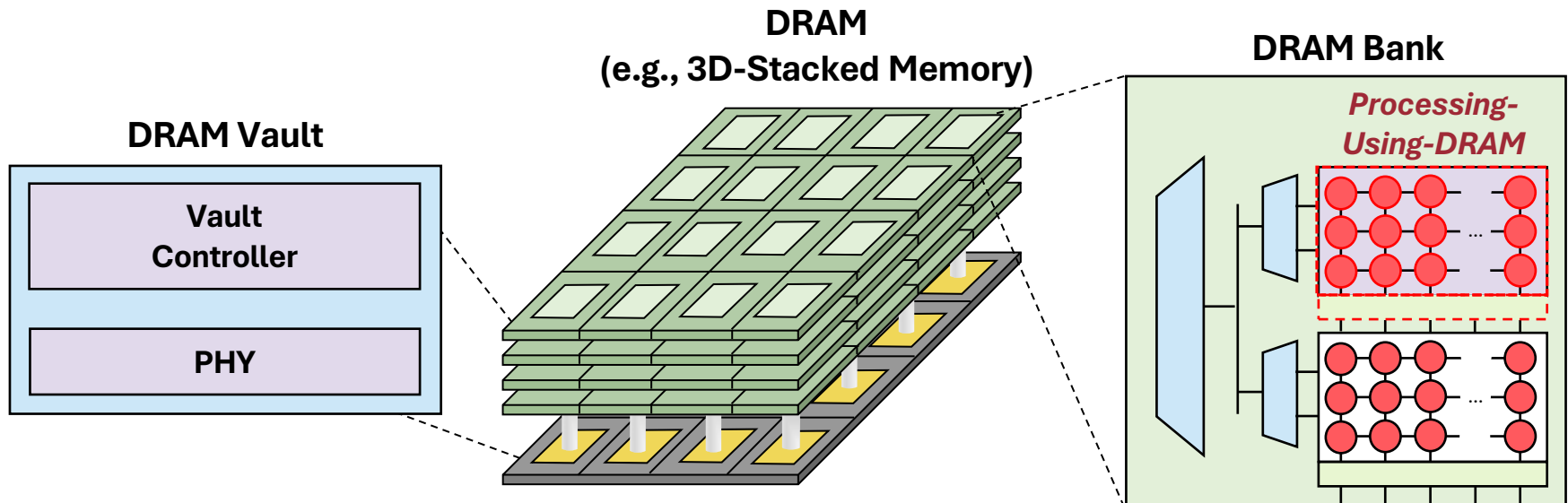


More than **60%** of the total system energy is spent on **data movement**¹

Processing-using-DRAM (PuD)

Processing-using-DRAM uses the analog **operational** principles of **DRAM cells** to perform computation

Multiple-row activation is a key technique to realize many PuD operations (e.g., AND and OR)

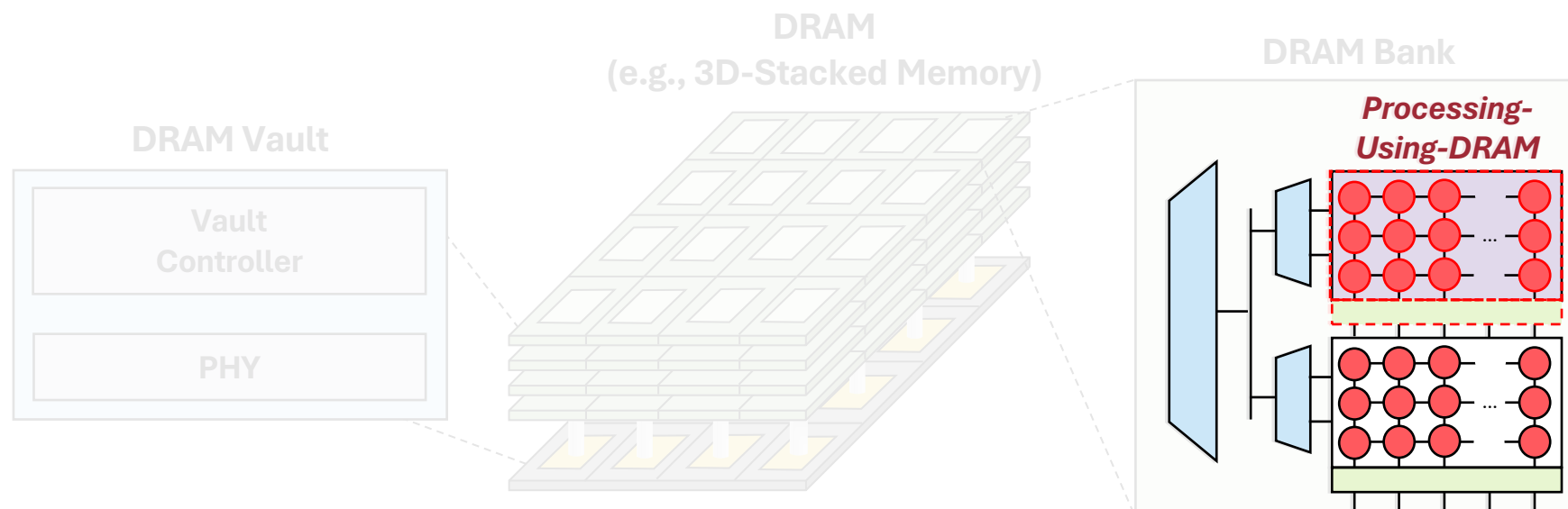


Processing-using-DRAM (PuD)

Multiple-row activation is a key technique to realize many PuD operations

1. Consecutive Multiple-Row Activation (CoMRA)

2. Simultaneous Multiple-Row Activation (SiMRA)

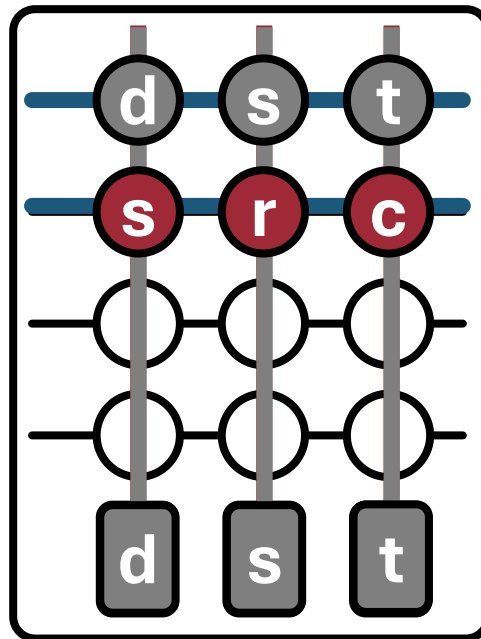


Consecutive Multiple-Row Activation

Violating timing of PRE command **greatly**
in **COTS** DRAM chips
activates two rows **consecutively**

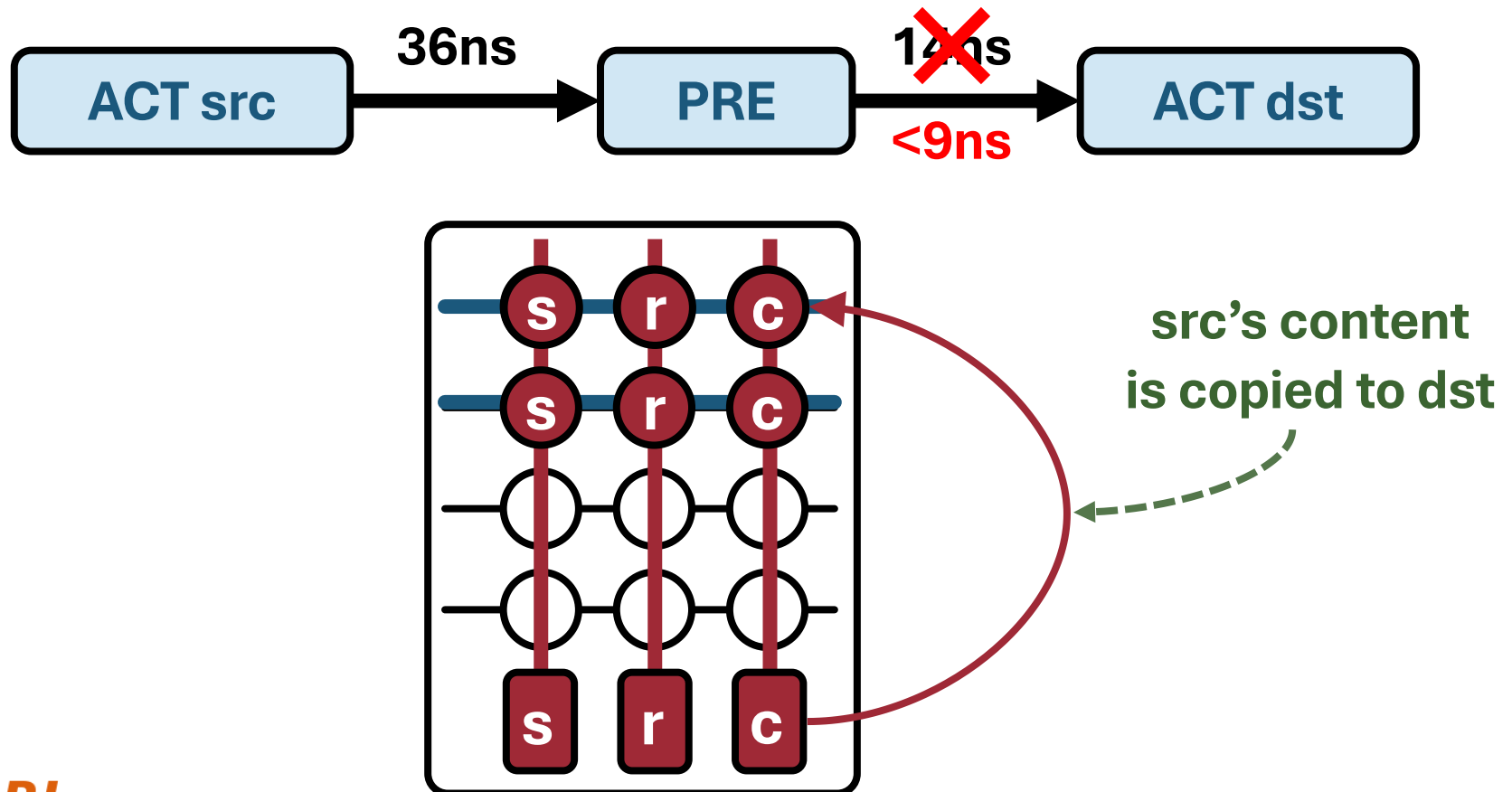
Consecutive Multiple-Row Activation

Violating timing of PRE command **greatly**
in **COTS** DRAM chips
activates two rows **consecutively**



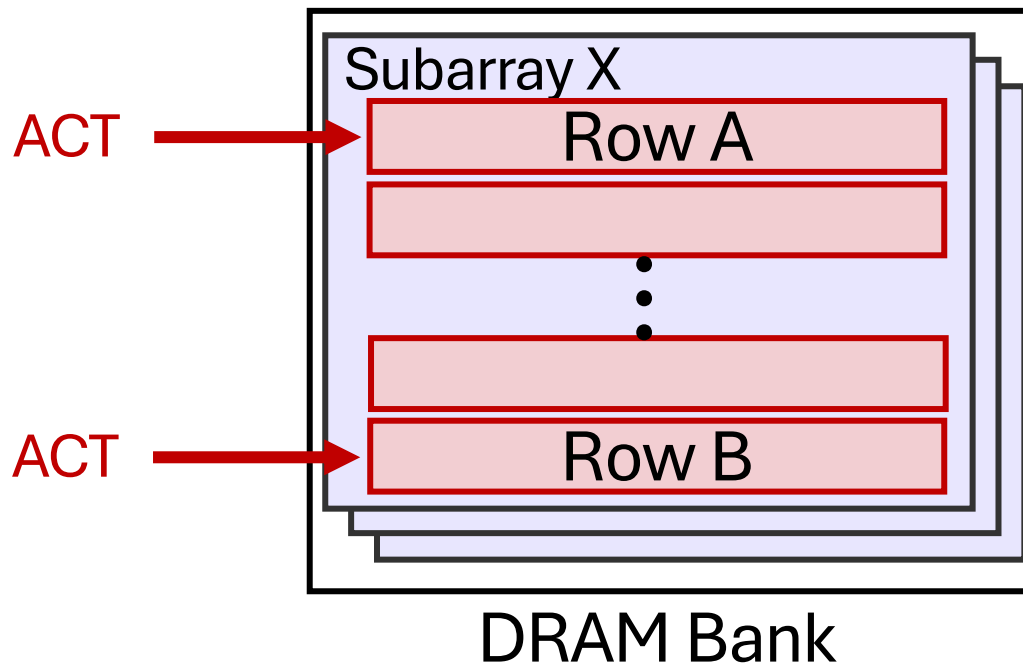
Consecutive Multiple-Row Activation

Violating timing of PRE command **greatly**
in **COTS** DRAM chips
activates two rows **consecutively**

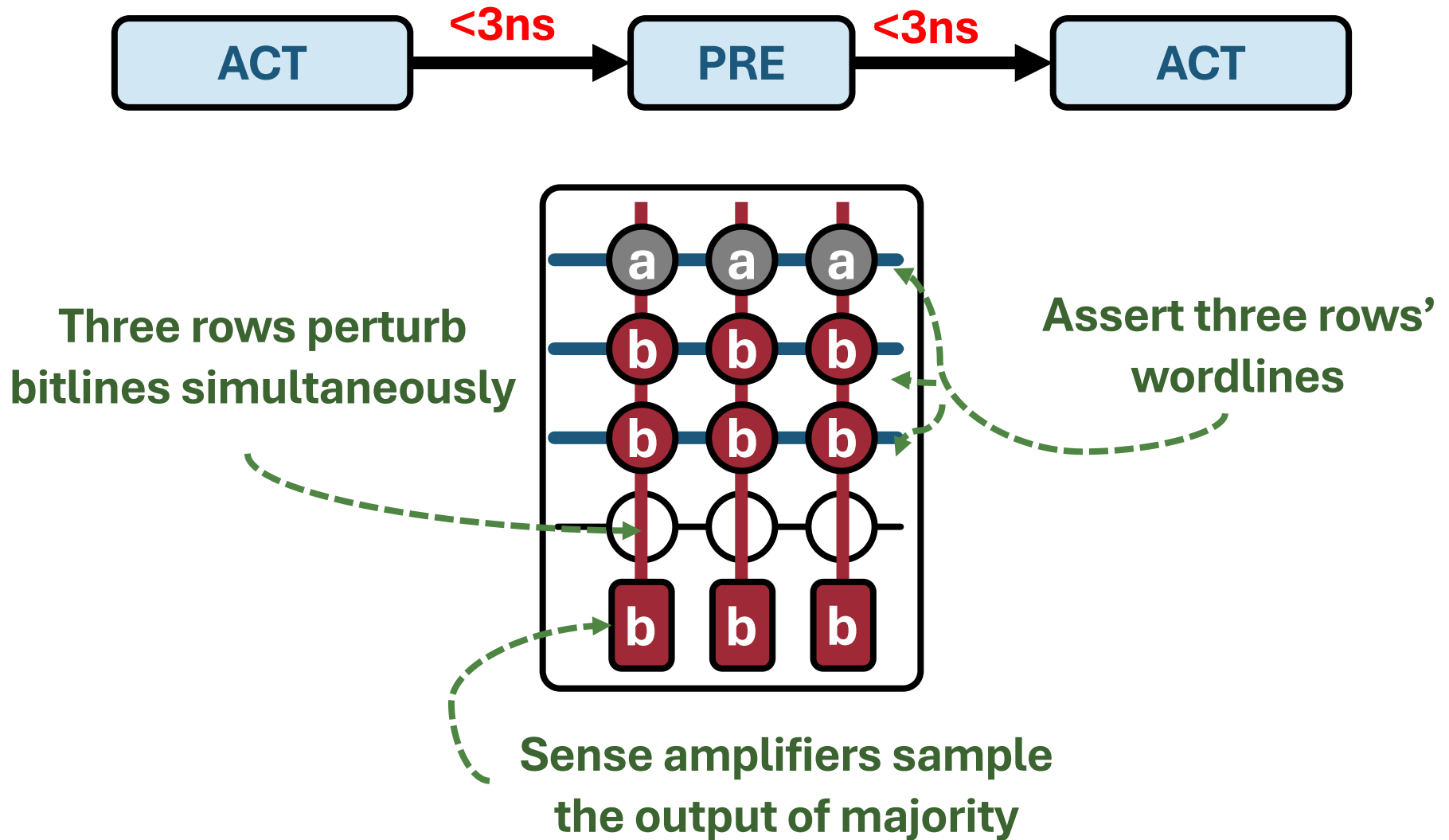


Simultaneous Multiple-Row Activation

Activating two rows in **quick succession** in **COTS** DRAM chips **simultaneously** activates **multiple rows** in a subarray

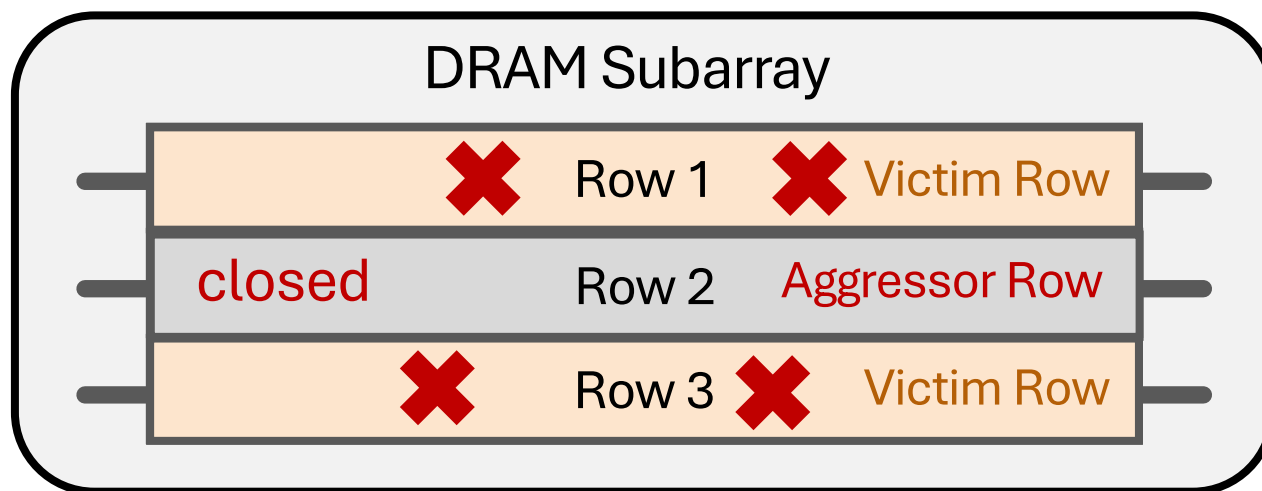


Example: In-DRAM Majority-of-Three (MAJ3)



Read Disturbance in DRAM

- Unfortunately, even activating a **single DRAM row** **disturbs** the data integrity of other **unaccessed DRAM rows**
- Prominent example: **RowHammer**



Repeatedly **opening (activating)** and **closing** a DRAM row **many times** causes **RowHammer bitflips** in adjacent rows

Outline

Background

Problem & Goal

Testing Infrastructure

Read Disturbance Effect of CoMRA

Read Disturbance Effect of SiMRA

Combined Effect of CoMRA, SiMRA, and RowHammer

Conclusion

Problem & Goal

Problem

*No prior work **investigates** the **read disturbance effects** of **multiple-row activation***

Goal

*Understand how **multiple-row activation** affects **read disturbance vulnerability** in **DRAM***

Outline

Background

Problem & Goal

Real DRAM Chip Testing Infrastructure

Read Disturbance Effect of CoMRA

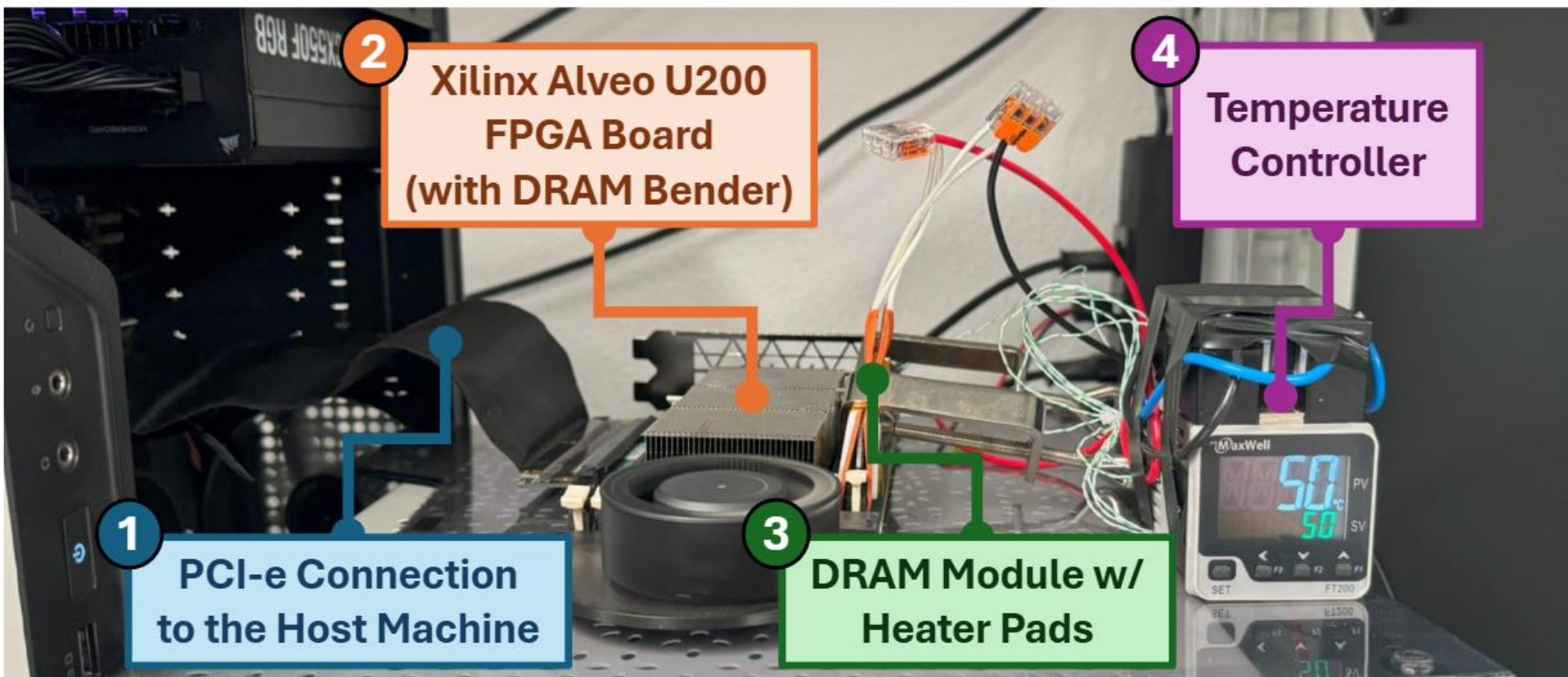
Read Disturbance Effect of SiMRA

Mitigating PuDHammer

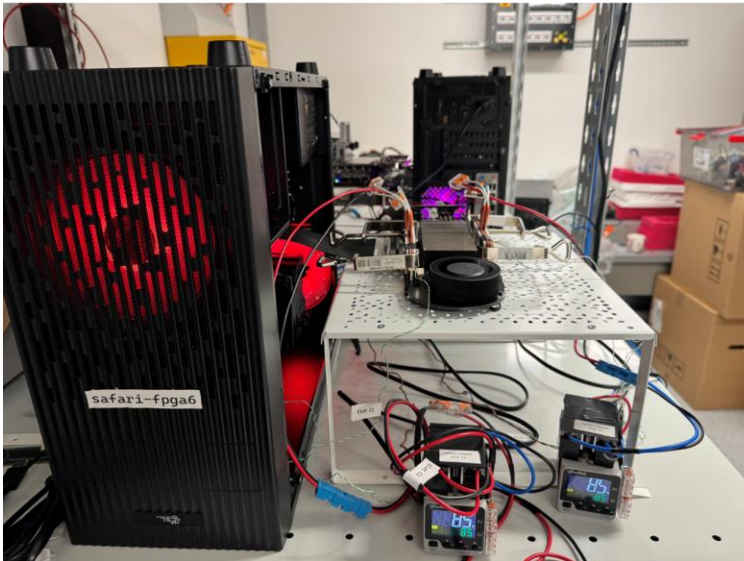
Conclusion

DRAM Bender: DRAM Testing Infrastructure

Fine-grained control over DRAM commands, timings, temperature, and voltage



Laboratory for Understanding Memory



DRAM Chips Tested

- 316 DDR4 chips from four major DRAM manufacturers
- Covers different die revisions and chip densities

Chip Mfr.	#Modules	#Chips	Die Rev.	Density	Org.
SK Hynix	1	8	A	4Gb	x8
	8	64	A	8Gb	x8
	2	16	C	16Gb	x8
	6	48	D	8Gb	x8
Micron	1	8	B	4Gb	x8
	4	32	E	16Gb	x16
	4	32	F	16Gb	x8
	2	16	R	8Gb	x8
Samsung	1	8	A	16Gb	x8
	5	40	B	16Gb	x8
	1	4	C	4Gb	x16
	1	8	C	16Gb	x8
	1	8	E	4Gb	x8
Nanya	3	24	C	8Gb	x8

Outline

Background

Problem & Goal

Testing Infrastructure

Read Disturbance Effect of CoMRA

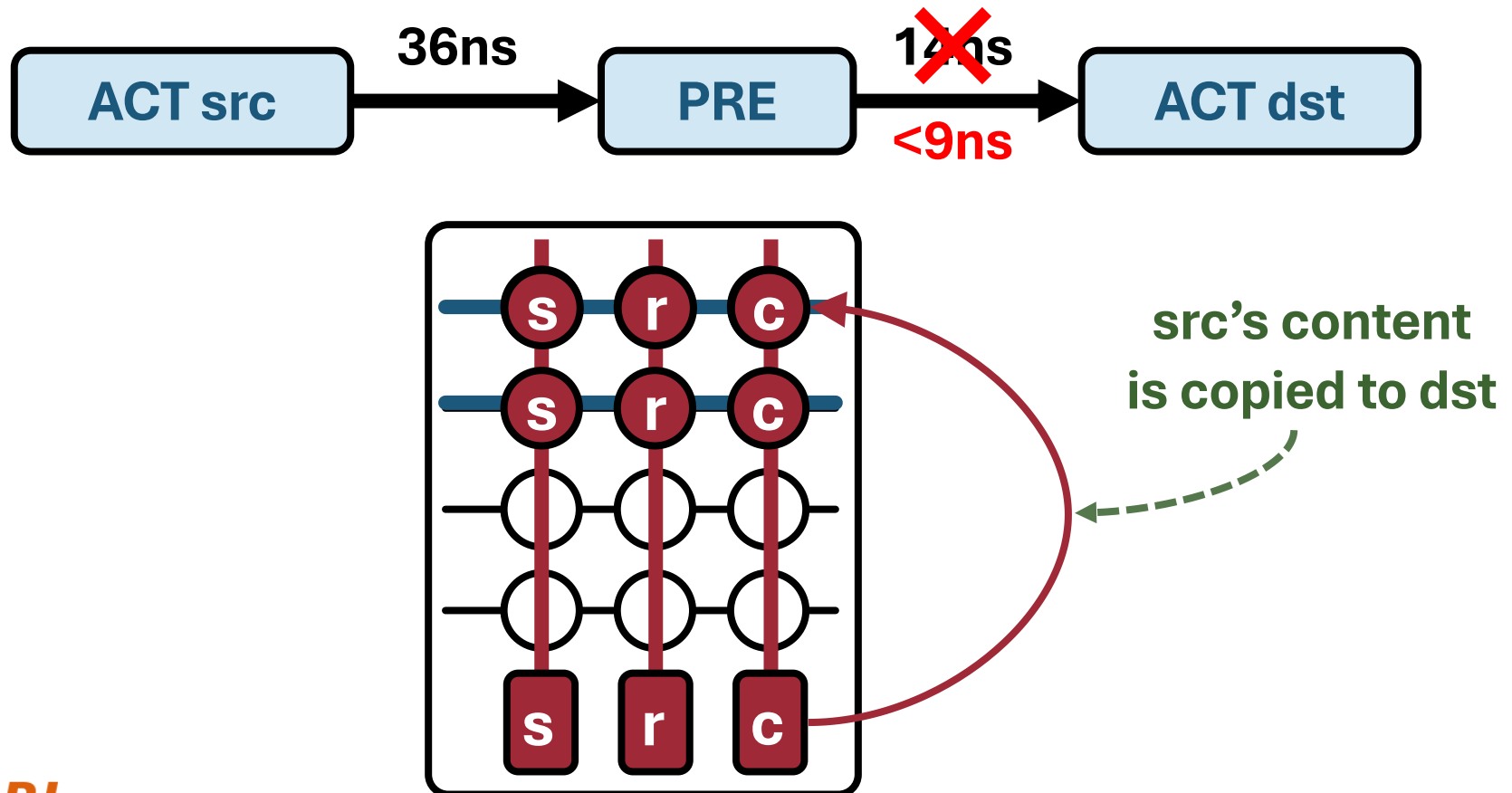
Read Disturbance Effect of SiMRA

Mitigating PuDHammer

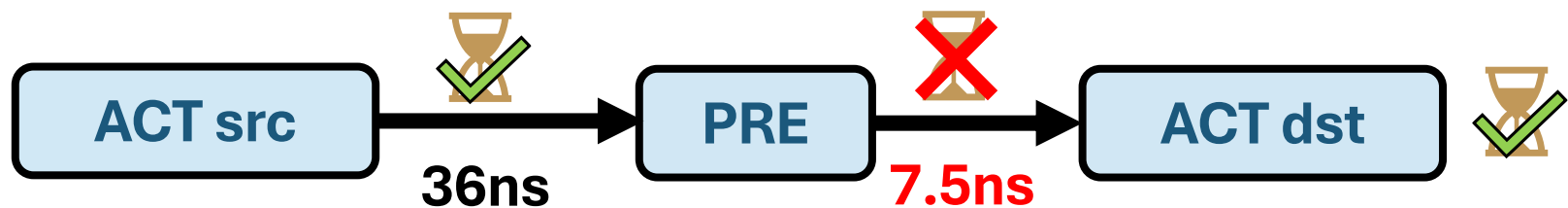
Conclusion

Consecutive Multiple-Row Activation

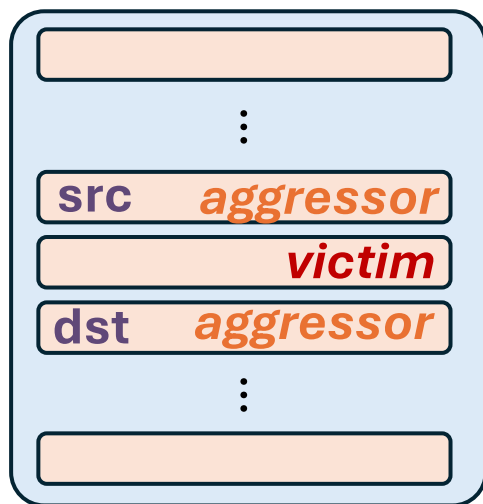
Violating timing of PRE command **greatly**
in **COTS** DRAM chips
activates two rows **consecutively**



Hammering with CoMRA

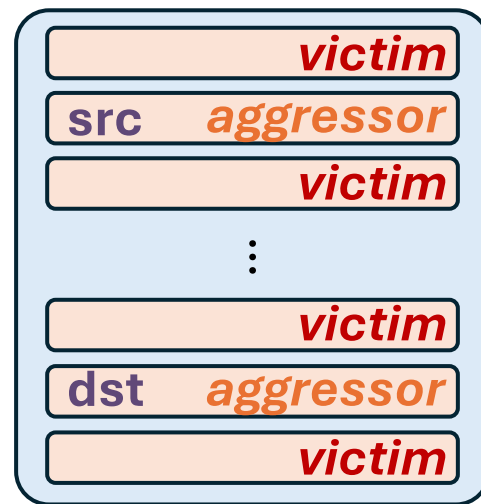


Double-Sided



DRAM Subarray

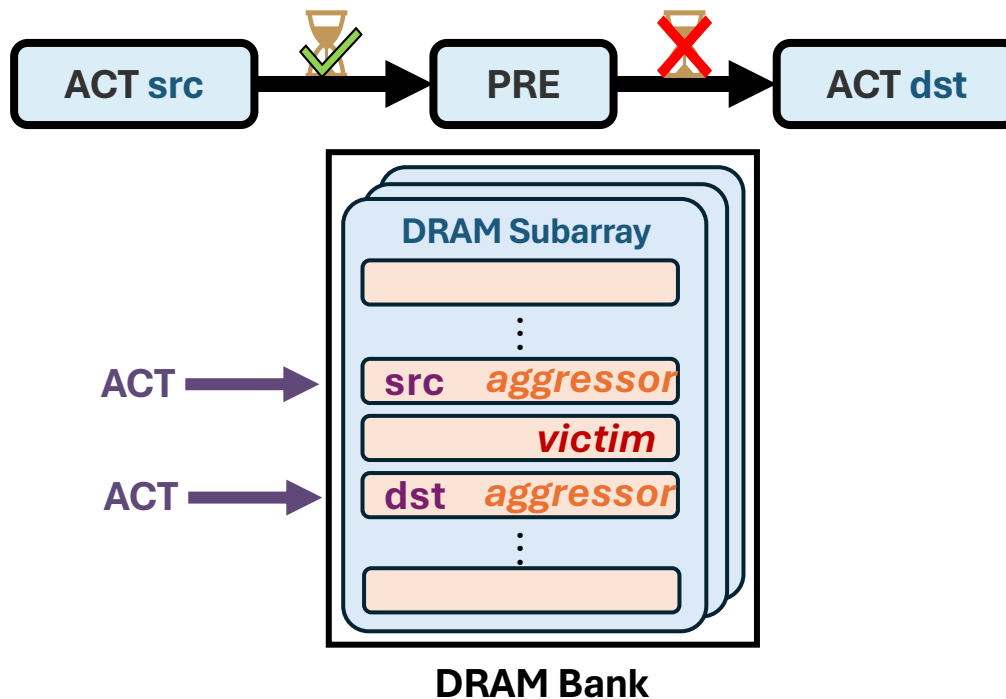
Single-Sided



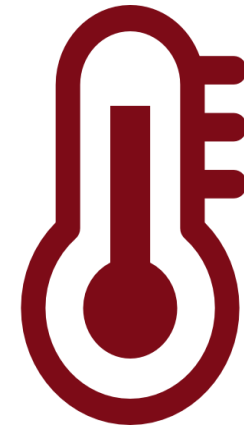
DRAM Subarray

Characterization Methodology

- Carefully sweep
 - Row addresses: src and dst
 - Timing parameters: Between PRE → ACT
 - Data Pattern: 0x00, 0xFF, 0xAA, and 0x55
 - Temperature (°C): 50, 60, 70, and 80



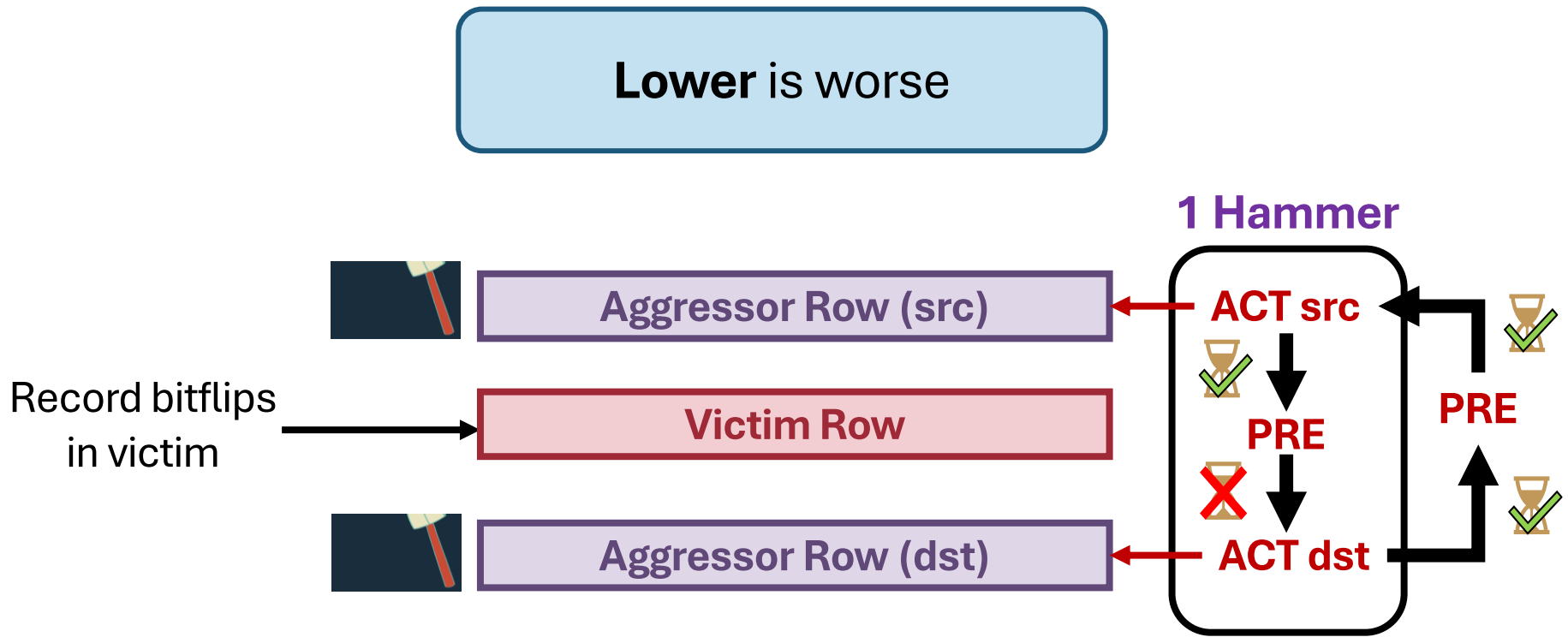
Temperature



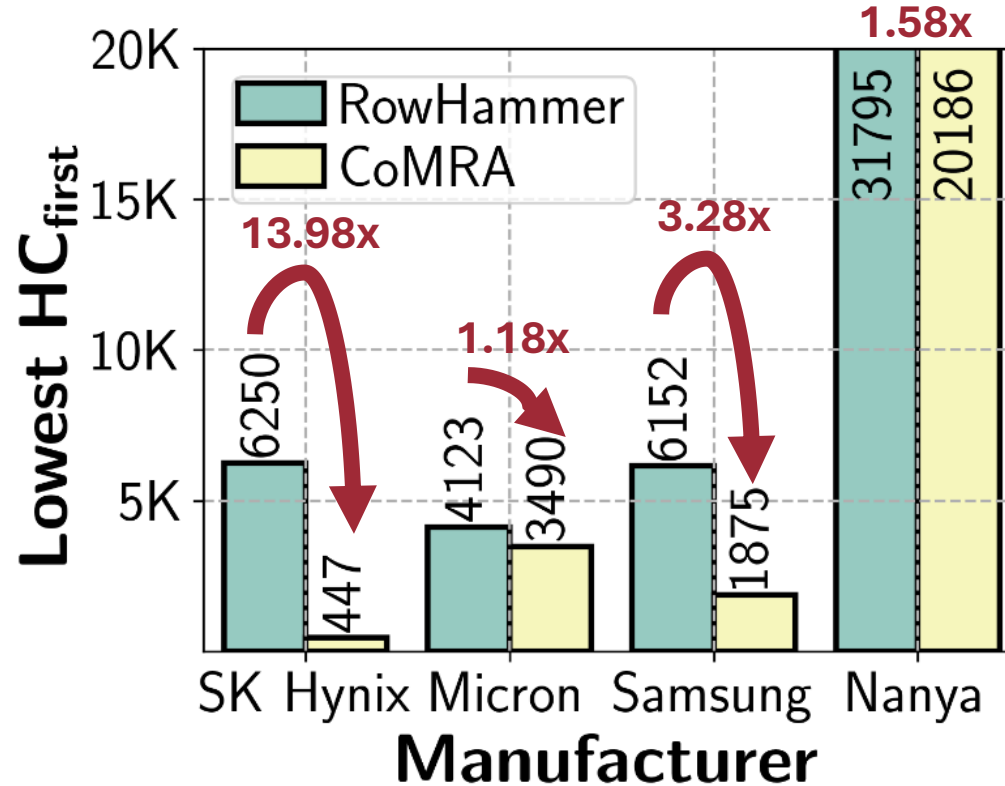
50 °C → 80 °C

Read Disturbance Vulnerability Metric: HC_{first}

The minimum hammer count
required to induce the **first** bitflip



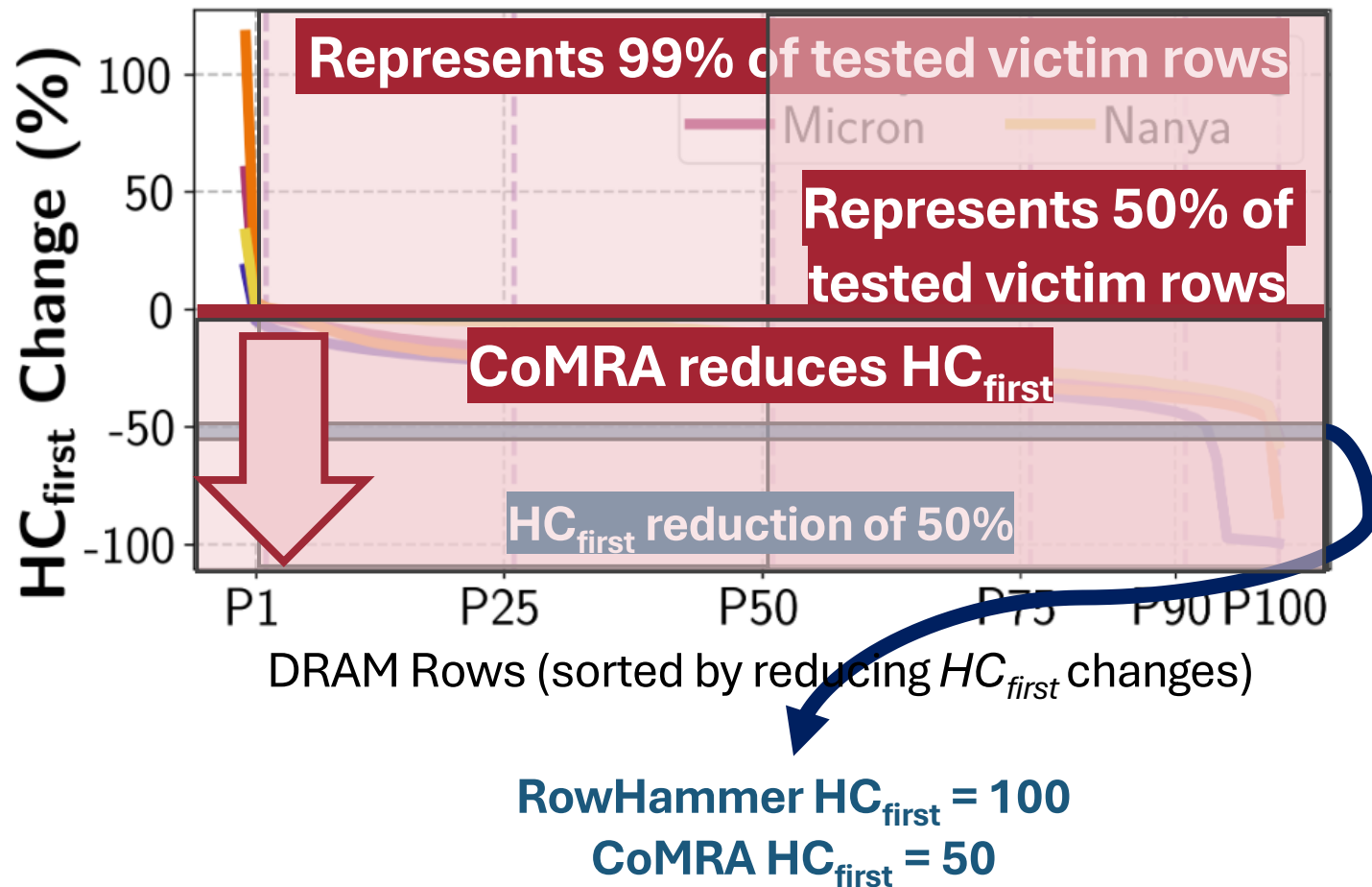
Double-Sided CoMRA vs. RowHammer (I)



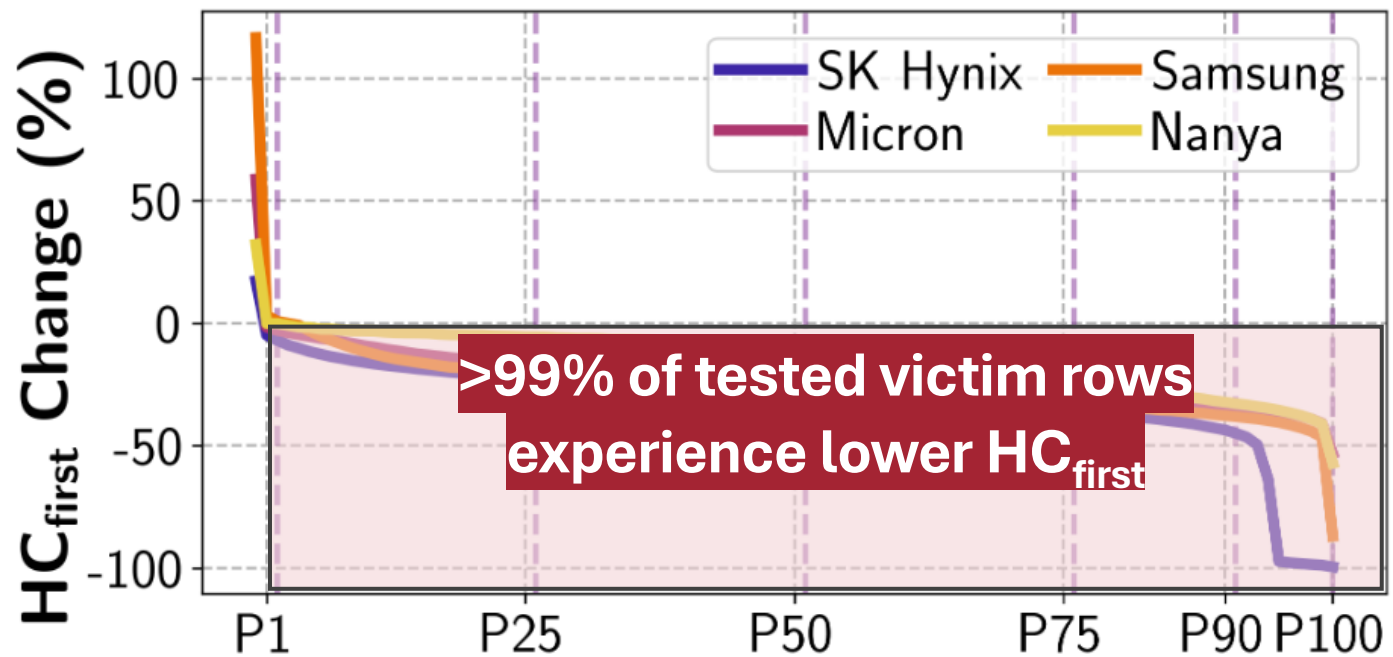
Double-sided CoMRA decreases **lowest HC_{first}** by **13.98x** compared to double-sided RowHammer

Lowest HC_{first} reduction trend consistent across all four manufacturer

Double-Sided CoMRA vs. RowHammer (II)

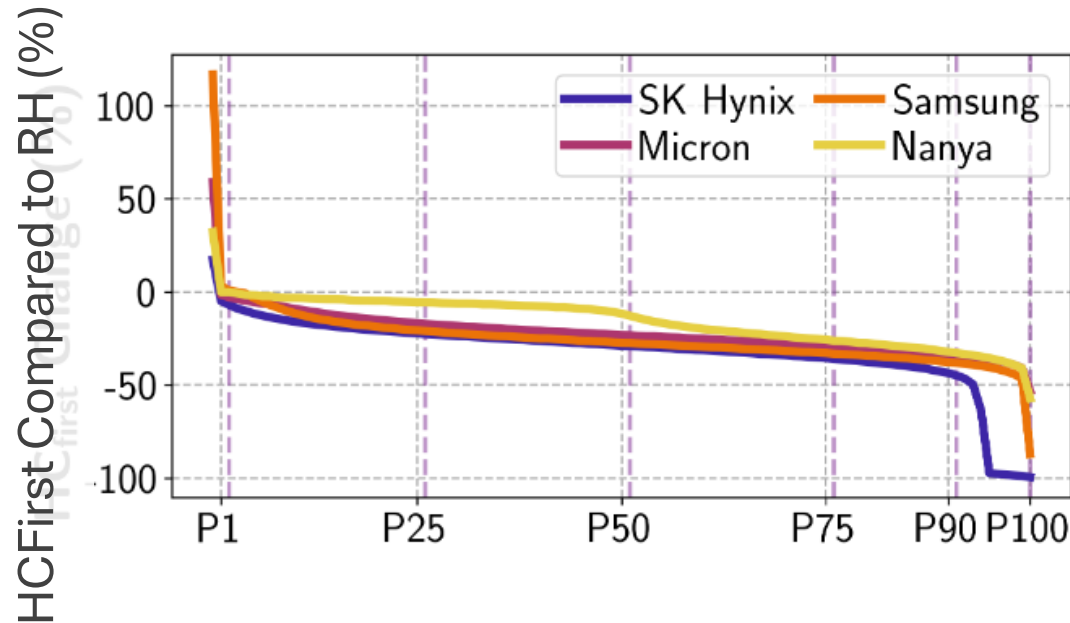
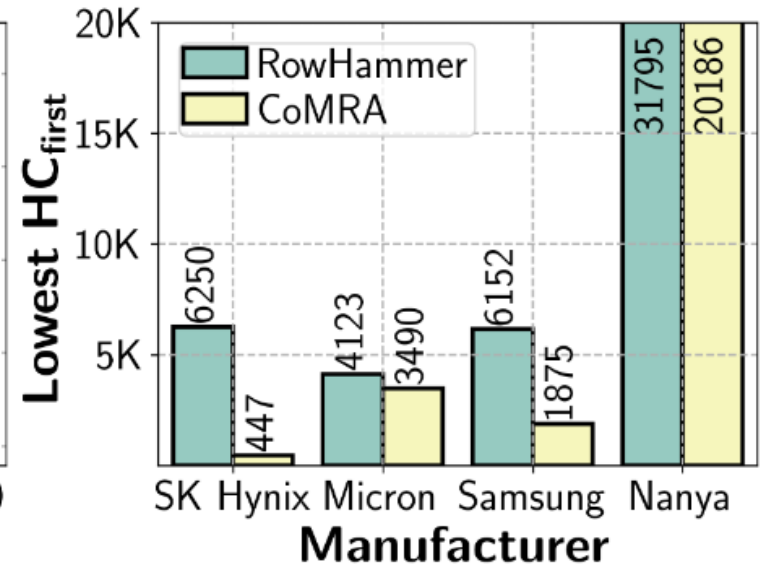


Double-Sided CoMRA vs. RowHammer (II)



Double-sided CoMRA decreases HC_{first} for a large fraction of rows compared to double-sided RowHammer

Double-Sided CoMRA vs. RowHammer



Key Takeaway

CoMRA exacerbates DRAM's vulnerability to read disturbance in all four major manufacturers

Outline

Background

Problem & Goal

Testing Infrastructure

Read Disturbance Effect of CoMRA

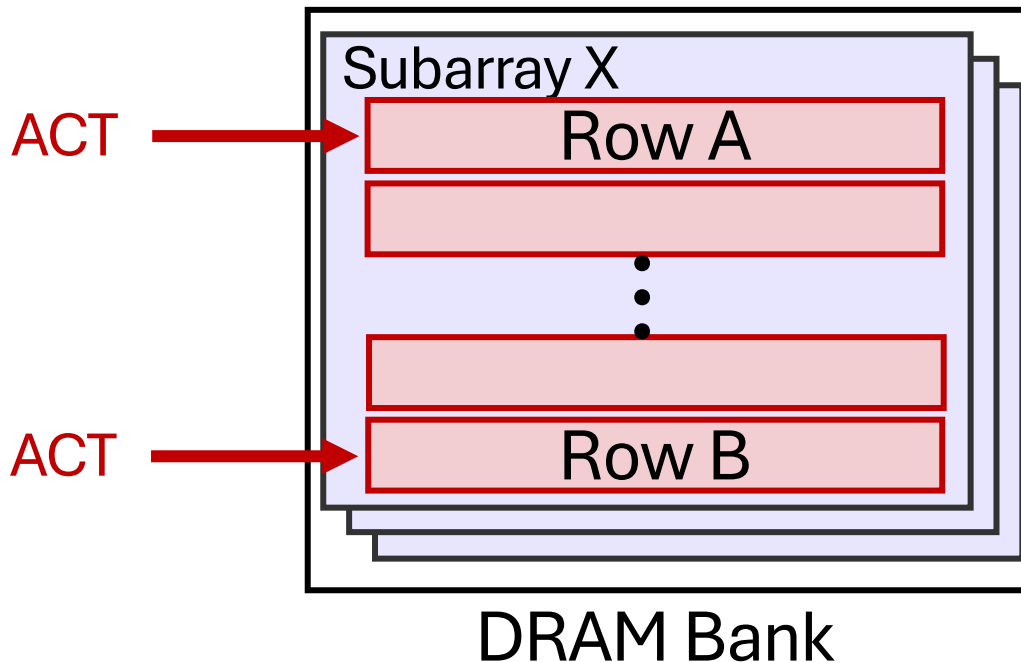
Read Disturbance Effect of SiMRA

Mitigating PuDHammer

Conclusion

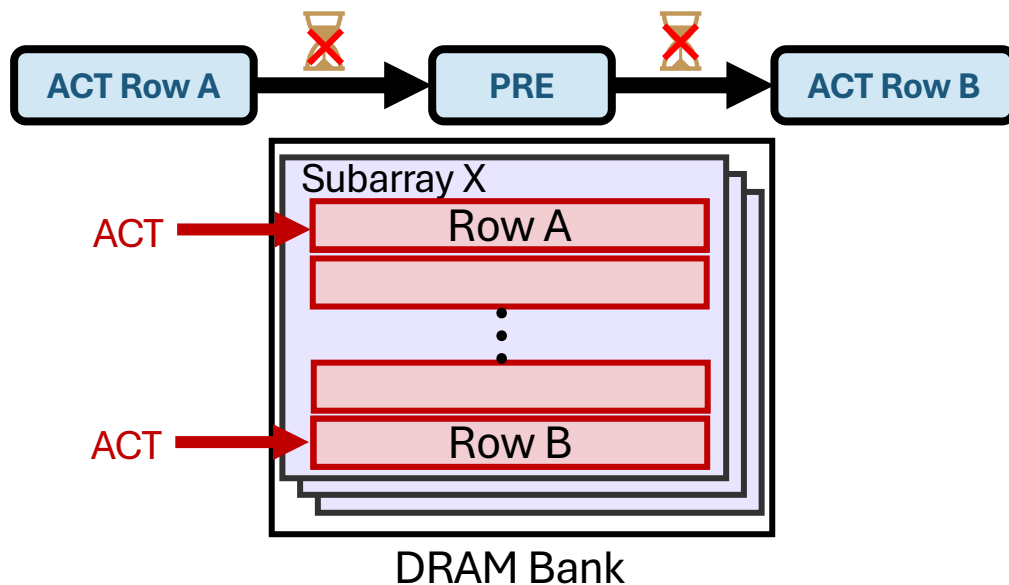
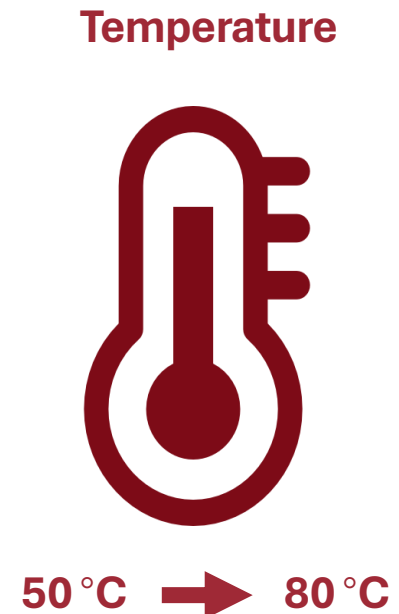
Recall: SiMRA

Activating two rows in **quick succession** in **COTS** DRAM chips **simultaneously** activates **multiple rows** in a subarray

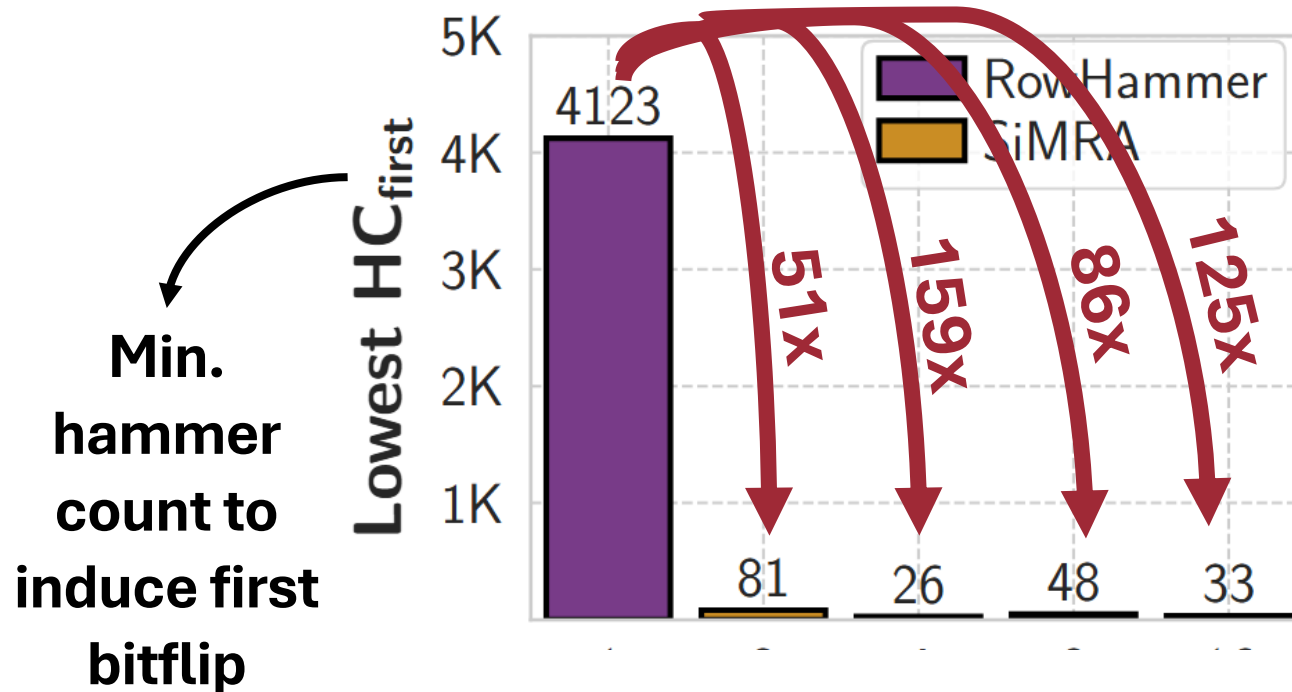


Characterization Methodology

- Carefully sweep
 - Row addresses: Row A and Row B
 - Timing parameters: Between ACT → PRE and PRE → ACT
 - Data Pattern: 0x00, 0xFF, 0xAA, and 0x55
 - Temperature (°C): 50, 60, 70, and 80



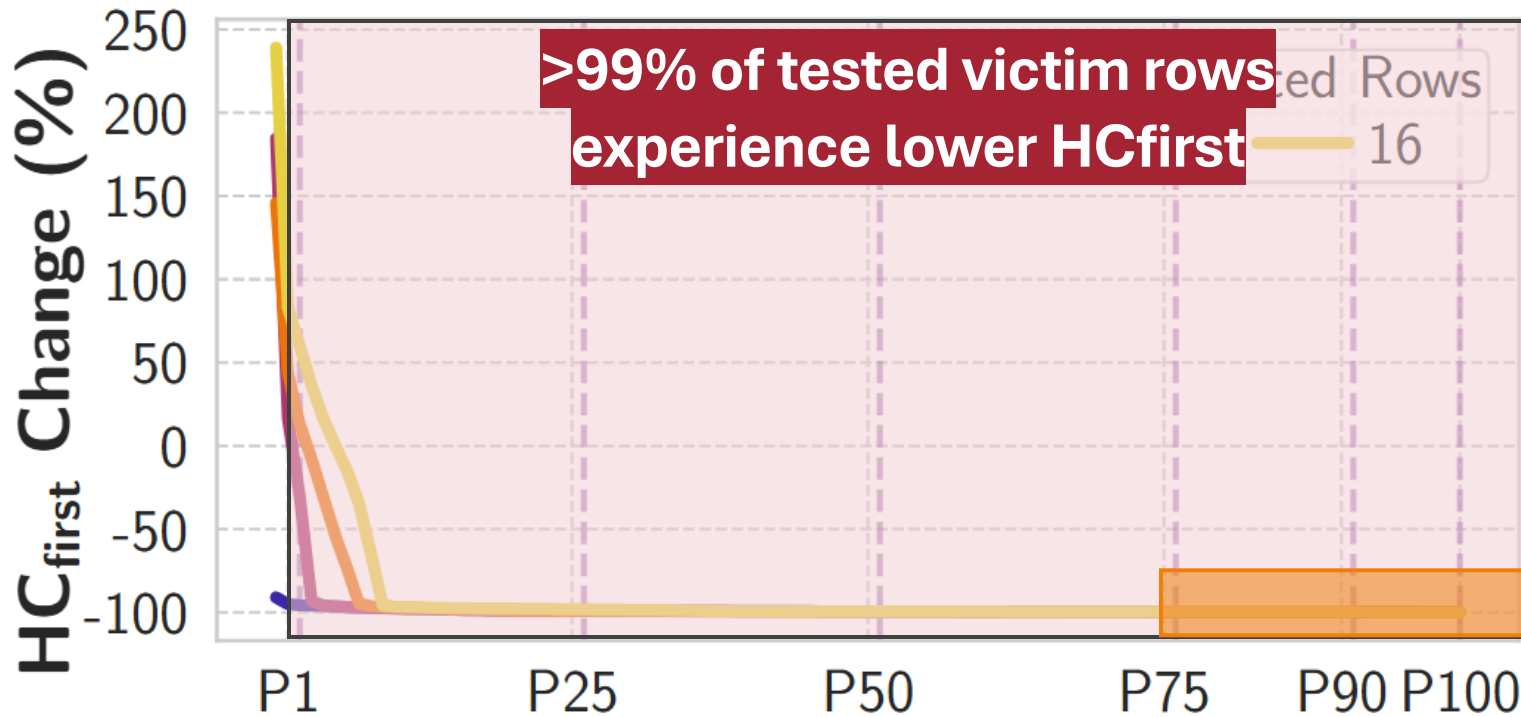
Double-Sided SiMRA vs. RowHammer (I)



Double-sided SiMRA decreases **lowest HC_{first}** by 159x compared to double-sided RowHammer

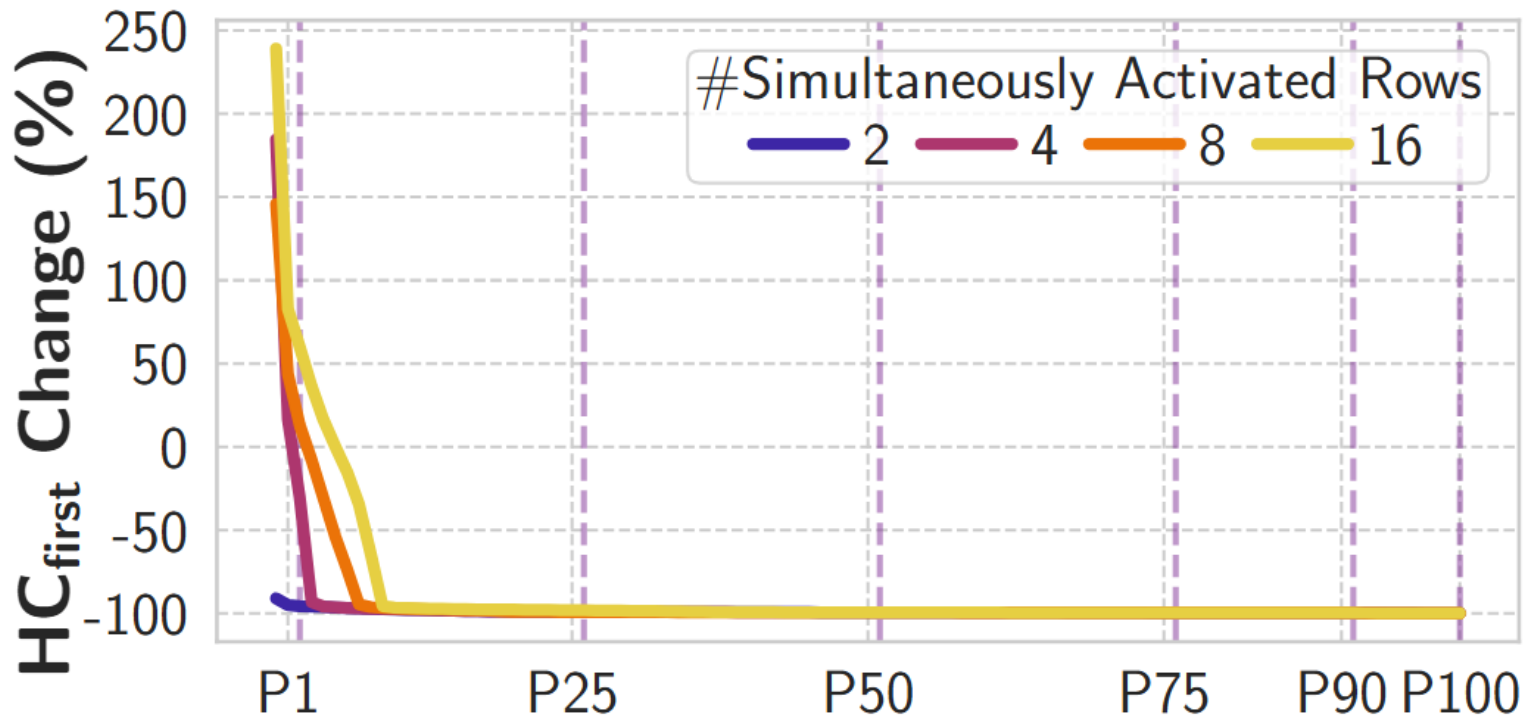
Lowest HC_{first} reduction trend consistent across all tested number of simultaneously activated rows

Double-Sided SiMRA vs. RowHammer (II)



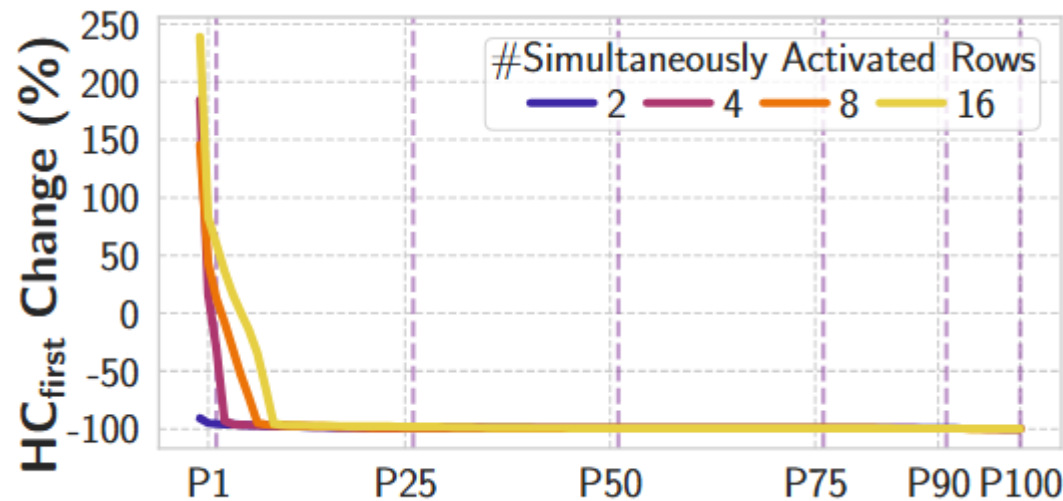
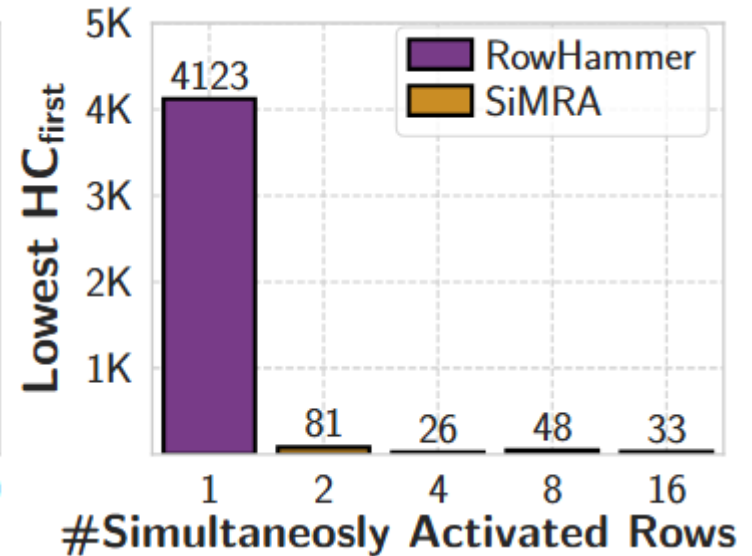
at least 25.19% of tested victim rows experience >99% HC_{first} reduction

Double-Sided SiMRA vs. RowHammer (II)



Double-sided SiMRA
significantly decreases HC_{first} for a majority of rows
compared to double-sided RowHammer

Double-sided SiMRA vs. RowHammer



Key Takeaway

**SiMRA drastically exacerbates
DRAM's vulnerability to read disturbance**

Outline

Background

Problem & Goal

Testing Infrastructure

Read Disturbance Effect of CoMRA

Read Disturbance Effect of SiMRA

Mitigating PuDHammer

Conclusion

Mitigating PuDHammer

1

Analyze the **effectiveness of existing in-DRAM RowHammer mitigation against PuDHammer**

2

Adapt RowHammer mitigations to account for PuDHammer

Mitigating PuDHammer

1

Analyze the **effectiveness of existing in-DRAM RowHammer mitigation against PuDHammer**

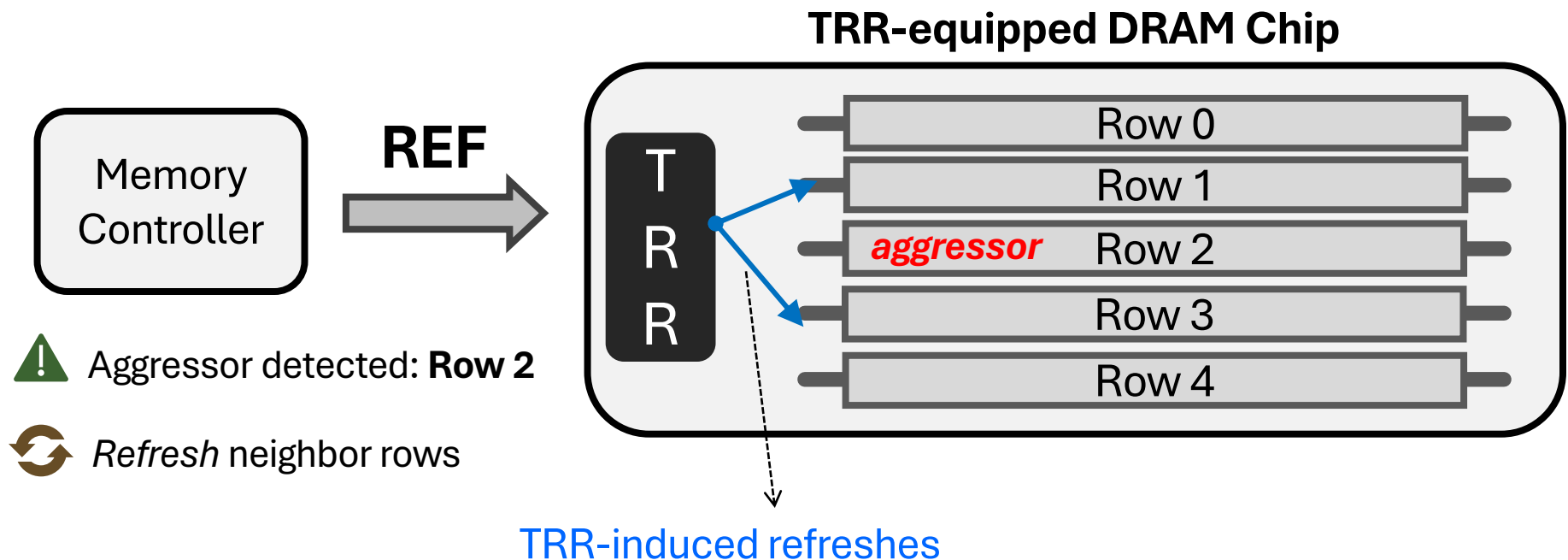
2

Adapt RowHammer mitigations
to account for PuDHammer

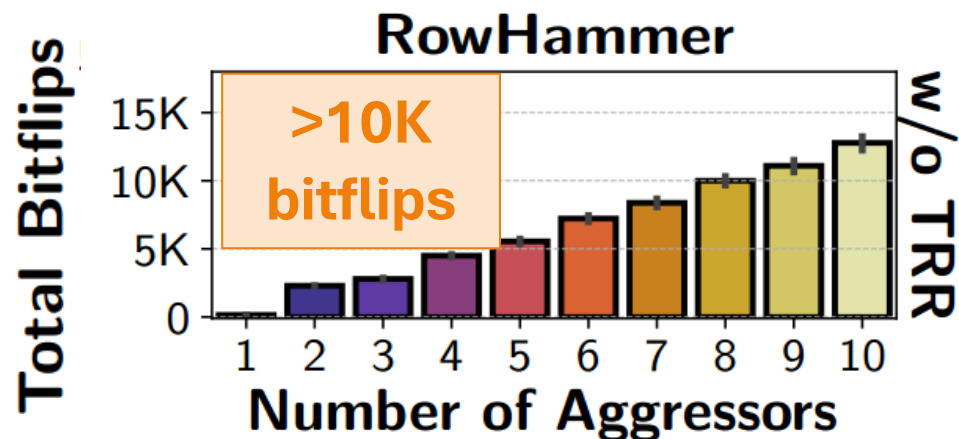
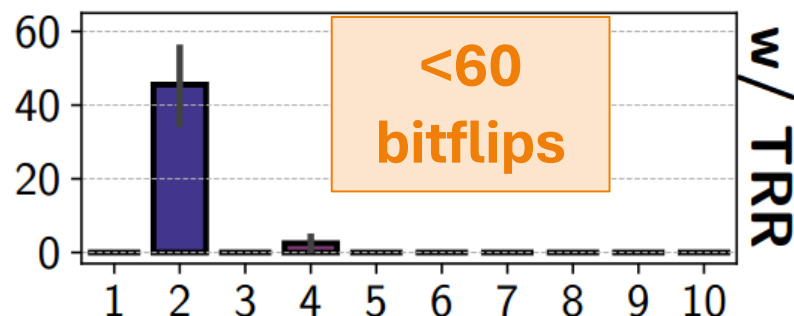
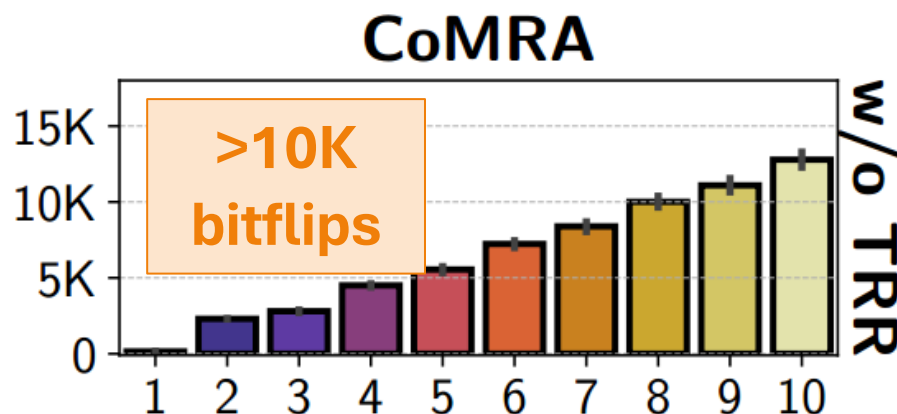
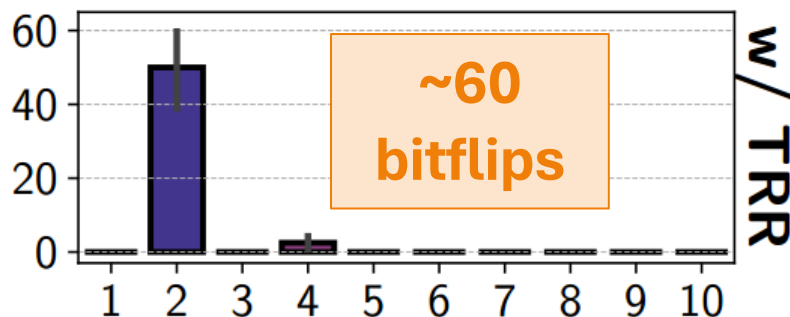
Target Row Refresh (TRR)

DRAM vendors equip their DRAM chips with a *proprietary* mitigation mechanisms known as **Target Row Refresh (TRR)**

Key Idea: TRR refreshes nearby rows upon detecting an aggressor row



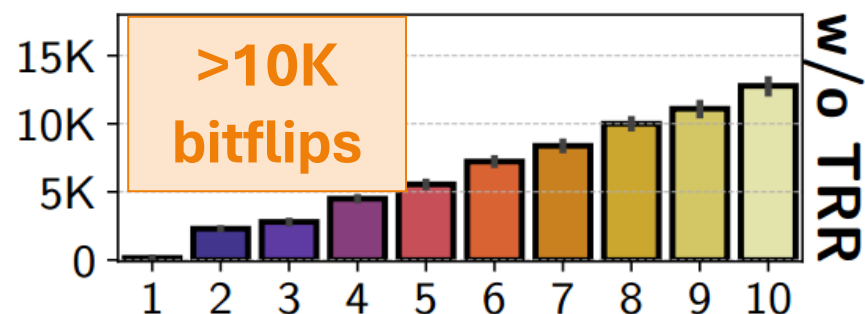
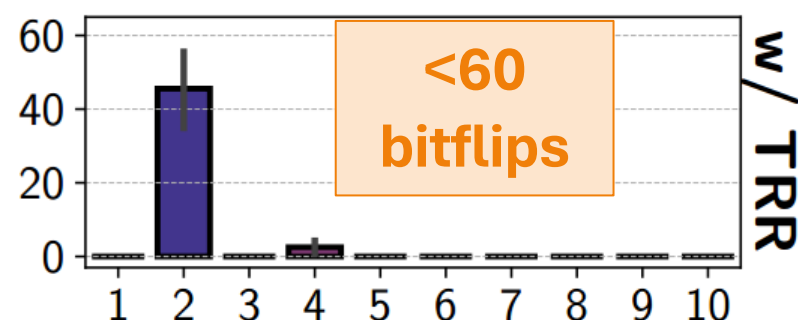
Real DRAM Chip Results



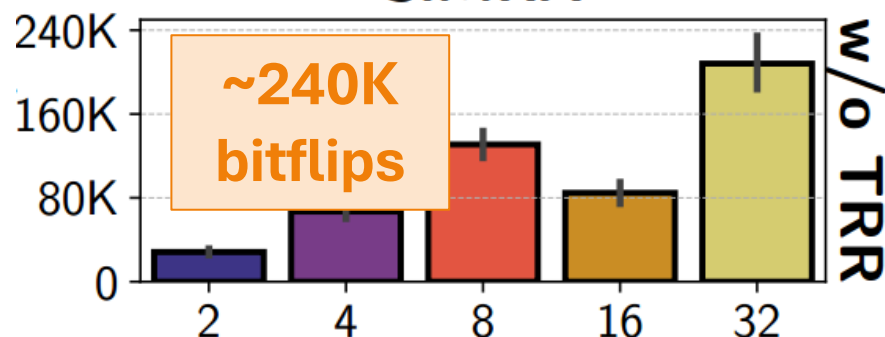
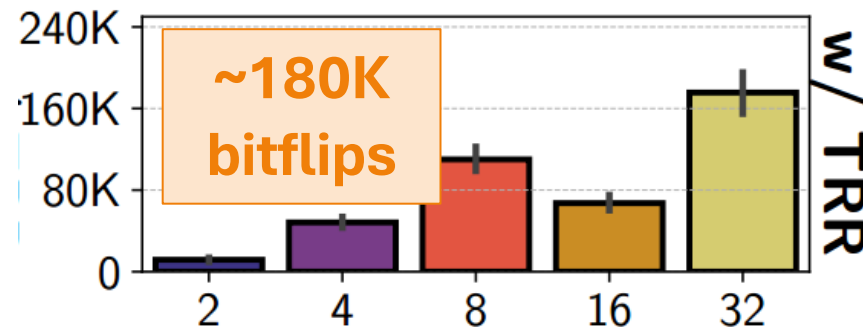
CoMRA induces 1.10x more bitflips than RowHammer on average in the presence of TRR

Real DRAM Chip Results

RowHammer



SiMRA



SiMRA induces 11340x more bitflips than RowHammer on average in the presence of TRR

Mitigating PuDHammer

1

Analyze the effectiveness of existing in-DRAM RowHammer mitigation against PuDHammer

2

**Adapt RowHammer mitigations
to account for PuDHammer**

A Naïve Approach (PRAC-Naïve)

*Reducing RowHammer threshold
to SiMRA's lowest observed HCfirst*

Weighted Counting (PRAC-WC)

*Increase counters based on the vulnerability
caused by the operation (RH, CoMRA, and SiMRA)*

Methodology

- **Environment:** Cycle-level DRAM simulator **Ramulator2** [Kim+ CAL'15, Luo+ CAL'23]

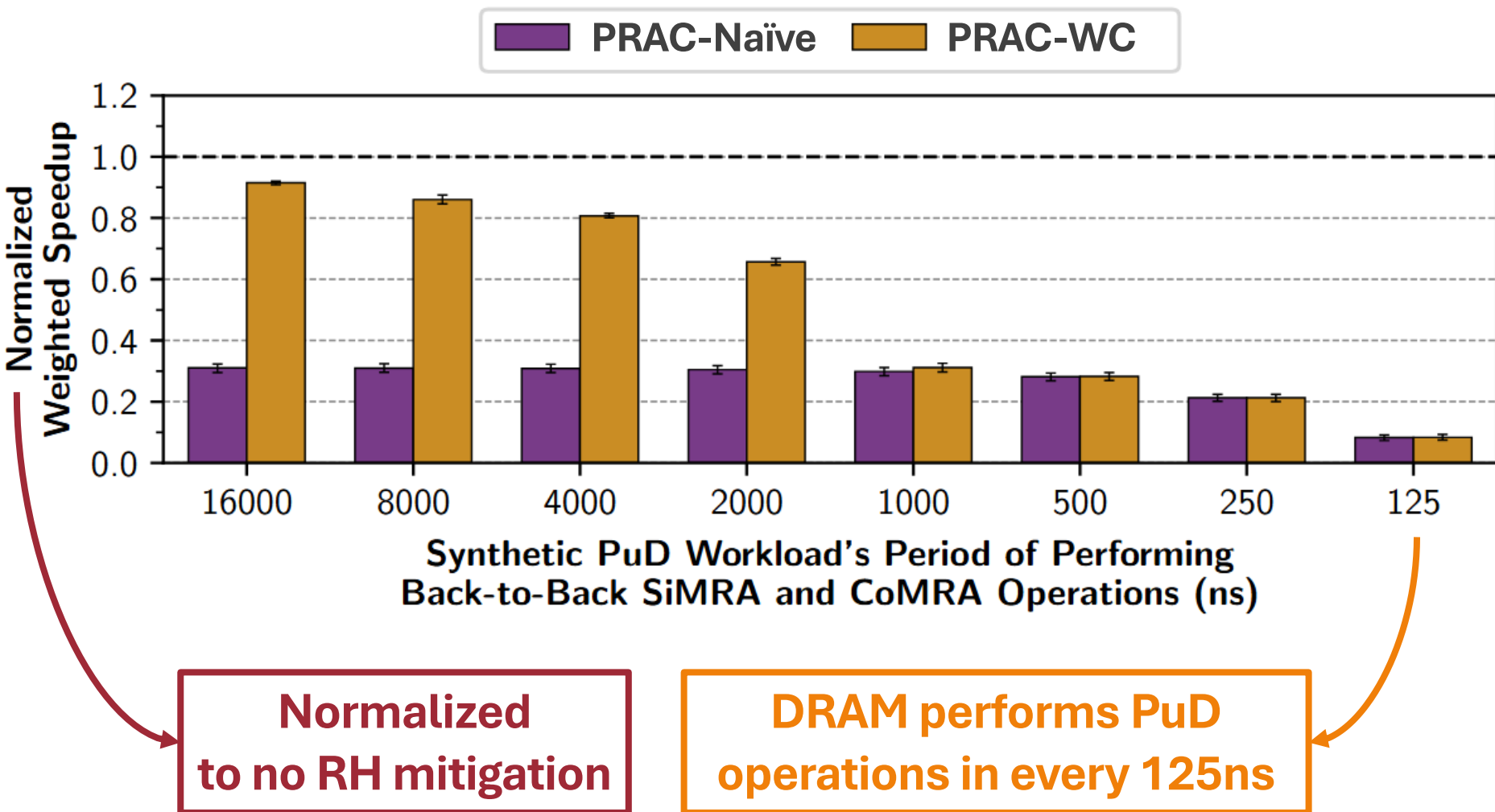
- **System Configuration:**

Processor	4-core, out-of-order, 4.2GHz clock frequency
DRAM	DDR5, 1 channel, 2 rank/channel, 8 bank groups, 4 banks/bank group, 128K rows/bank
Memory Ctrl.	64-entry read and write requests queues, Scheduling policy: FR-FCFS with a column cap of 4
PRAC	4 RFMs per Back-off
PIM Unit	1-core, issues CoMRA & SiMRA operations with a fixed interval

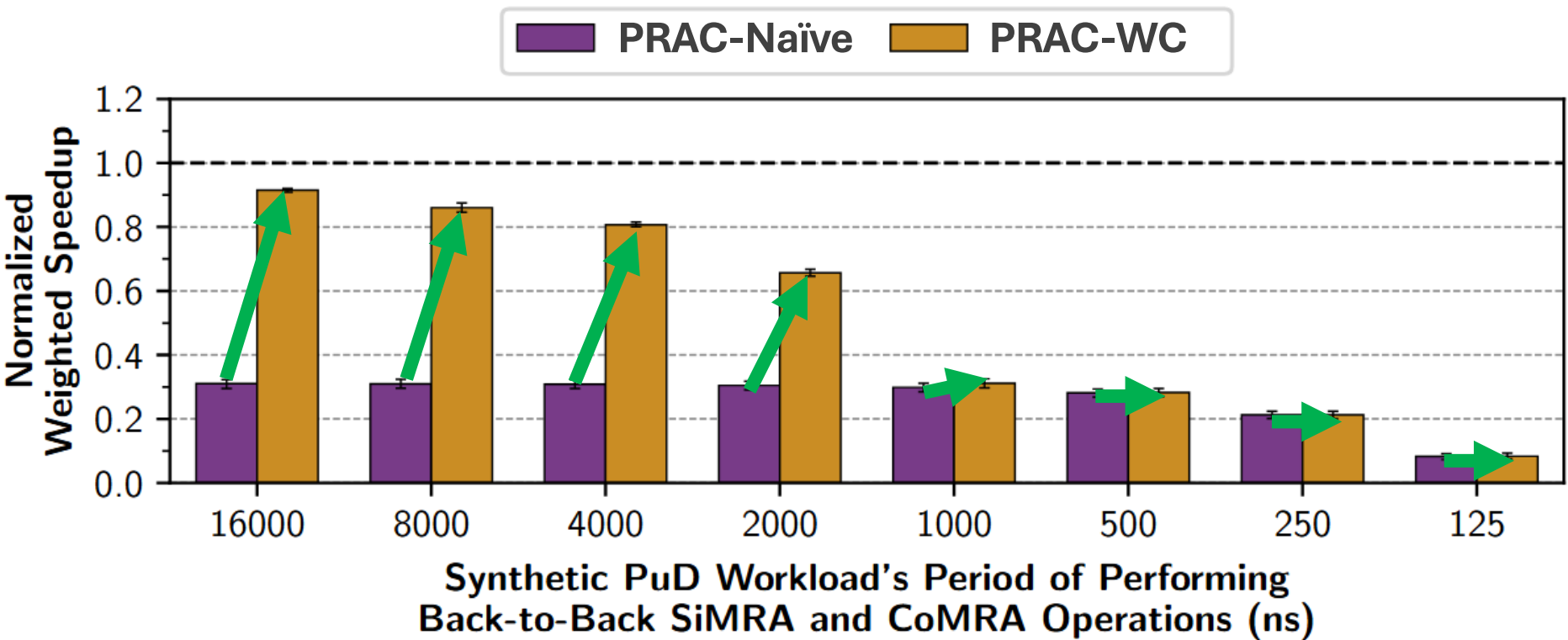
- **Workloads:**

- 60 five-core multiprogrammed workload mixes.
- Each mix is composed of
 - four workloads from five major benchmark suites and
 - one synthetic workload that periodically performs back-to-back one SiMRA with 32-row activation and one CoMRA operation

Weighted Speedup Results



Weighted Speedup Results



PRAC-WC greatly outperforms PRAC-Naïve until 2us

PRAC-WC incurs an average (maximum) performance overhead of 48.26% (98.83%)

More in the Paper

- **More characterization results**
 - Temperature, spatial variation, timing parameters, access pattern, data pattern, aggressor row on time analyses for CoMRA and SiMRA
 - Combined Read Disturbance Effect of RowHammer with PuD
 - Combined pattern is more effective than RowHammer alone (e.g., 1.66x reduction in HCfirst on average)
- **Details and analyses on mitigating PuDHammer**
 - Discussion on how modifying DRAM chips could help mitigating PuDHammer
 - More details on PRAC adaptation
 - Challenges on adapting for PuDHammer
 - An area-efficient PRAC solution
 - ..

PuDHammer: Experimental Analysis of Read Disturbance Effects of Processing-using-DRAM in Real DRAM Chips

Ismail Emir Yüksel Akash Sood Ataberk Olgun Oğuzhan Canpolat Haocong Luo
F. Nisa Bostancı Mohammad Sadrosadati A. Giray Yağlıkçı Onur Mutlu

ETH Zürich

Processing-using-DRAM (PuD) is a promising paradigm for alleviating the data movement bottleneck using a DRAM array's massive internal parallelism and bandwidth to execute very wide data-parallel operations. Performing a PuD operation involves activating multiple DRAM rows in quick succession or simultaneously, i.e., multiple-row activation. Multiple-row activation is fundamentally different from conventional memory access patterns that activate one DRAM row at a time. However, repeatedly activating even one DRAM row (e.g., RowHammer) can induce bitflips in unaccessed DRAM rows because modern DRAM is subject to read disturbance, a worsening safety, security, and reliability issue. Unfortunately, no prior work investigates the effects of multiple-row activation, as commonly used by PuD operations, on DRAM read disturbance.

In this paper, we present the first characterization study of read disturbance effects of multiple-row activation-based PuD (which we call PuDHammer) using 316 real DDR4 DRAM chips from four major DRAM manufacturers. Our detailed characterization results covering various operational conditions and parameters (i.e., temperature, data patterns, access patterns, timing parameters,

CPU and GPU) [1, 2]. Unfortunately, this data movement is a major bottleneck that consumes a large fraction of execution time and energy in many modern applications [1–28]. Processing-using-DRAM (PuD) [29–34] is a promising paradigm that can alleviate the data movement bottleneck. PuD uses the analog operational properties of the DRAM array circuitry to enable massively parallel in-DRAM computation (i.e., PuD operations), which can be used to accelerate important applications including databases and web search [29, 30, 32, 35–43], data analytics [29, 44–48], graph processing [32, 48–51], genome analysis [52–57], cryptography [58, 59], hyper-dimensional computing [60–62], and generative AI [63–72].

A wide variety of PuD operations (e.g., in-DRAM data copy and bulk bitwise operations) rely on a key PuD technique called multiple-row activation, which accesses (activates) multiple DRAM rows in quick succession or simultaneously [29–32, 40, 73–84]. Multiple-row activation is fundamentally different from conventional DRAM operations that access only a single DRAM row at a time.

<https://arxiv.org/pdf/2506.12947>

Outline

Background

Problem & Goal

Testing Infrastructure

Read Disturbance Effect of CoMRA

Read Disturbance Effect of SiMRA

Mitigating PuDHammer

Conclusion

Conclusion

We extensively study the interaction between read disturbance and multiple-row activation-based Processing-using-DRAM operations

We characterize 316 DDR4 chips from four major manufacturers

- PuDHammer **significantly exacerbates** the **read disturbance vulnerability** (e.g., HCfirst reduces from 4K to 26)
- PuDHammer **bypasses in-DRAM RowHammer mitigation** (TRR) and induces **11340x more bitflips** than RowHammer

We **adapt Per Row Activation Counting (PRAC)** for PuDHammer

- Incurs **48.26% performance overhead** on average across 60 multiprogrammed workload mixes

We hope our findings guide system-level and architectural solutions to enable read-disturbance-resilient future PuD systems

PuDHammer

Experimental Analysis of Read Disturbance Effects of Processing-using-DRAM in Real DRAM Chips

İsmail Emir Yüksel

Akash Sood Ataberk Olgun Oğuzhan Canpolat Haocong Luo

Nisa Bostancı Mohammad Sadrosadati Giray Yağlıkçı Onur Mutlu

SAFARI

ETH zürich

PuDHammer

Experimental Analysis of Read Disturbance Effects of Processing-using-DRAM in Real DRAM Chips

Backup Slides

İsmail Emir Yüksel

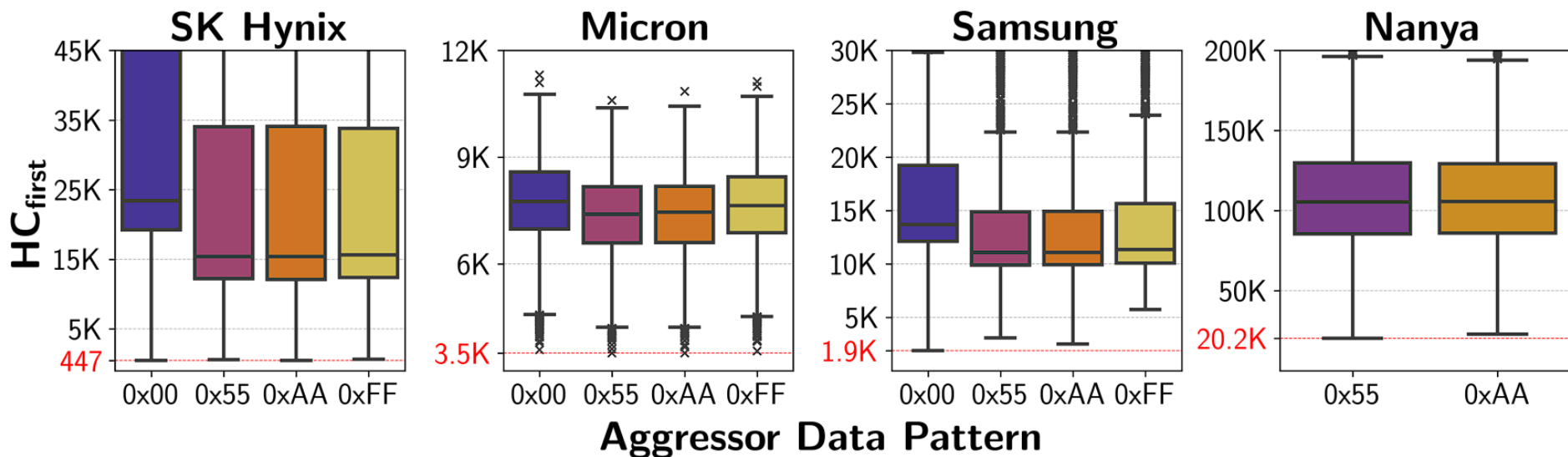
Akash Sood Ataberk Olgun Oğuzhan Canpolat Haocong Luo

Nisa Bostancı Mohammad Sadrosadati Giray Yağlıkçı Onur Mutlu

SAFARI

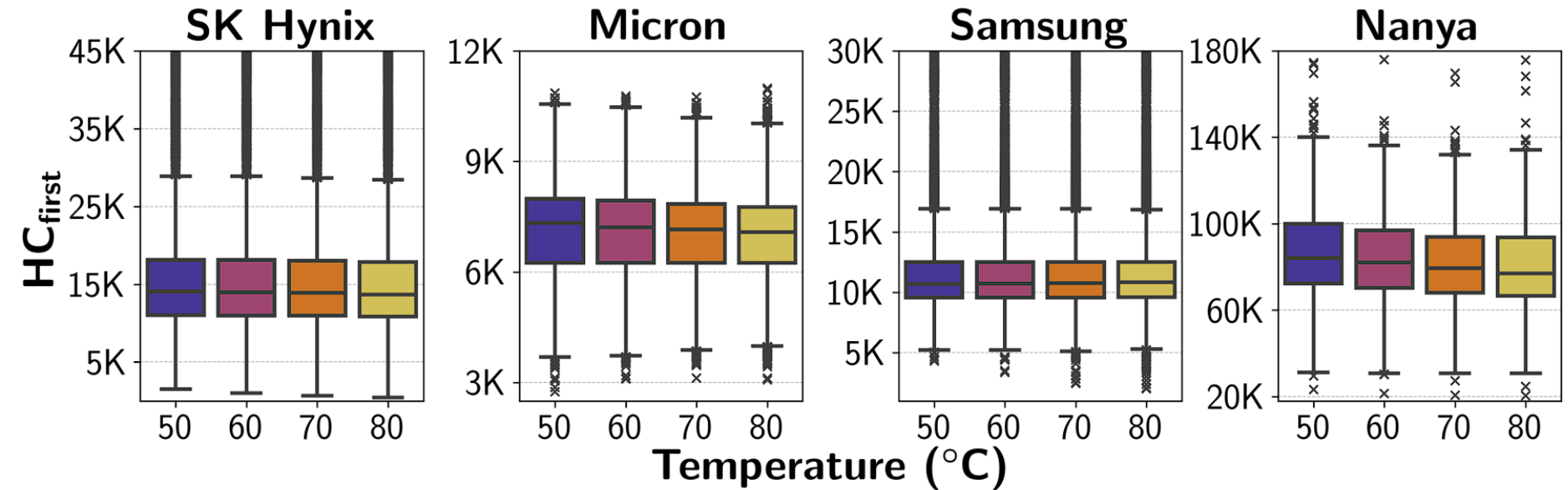
ETH zürich

CoMRA – Data Pattern



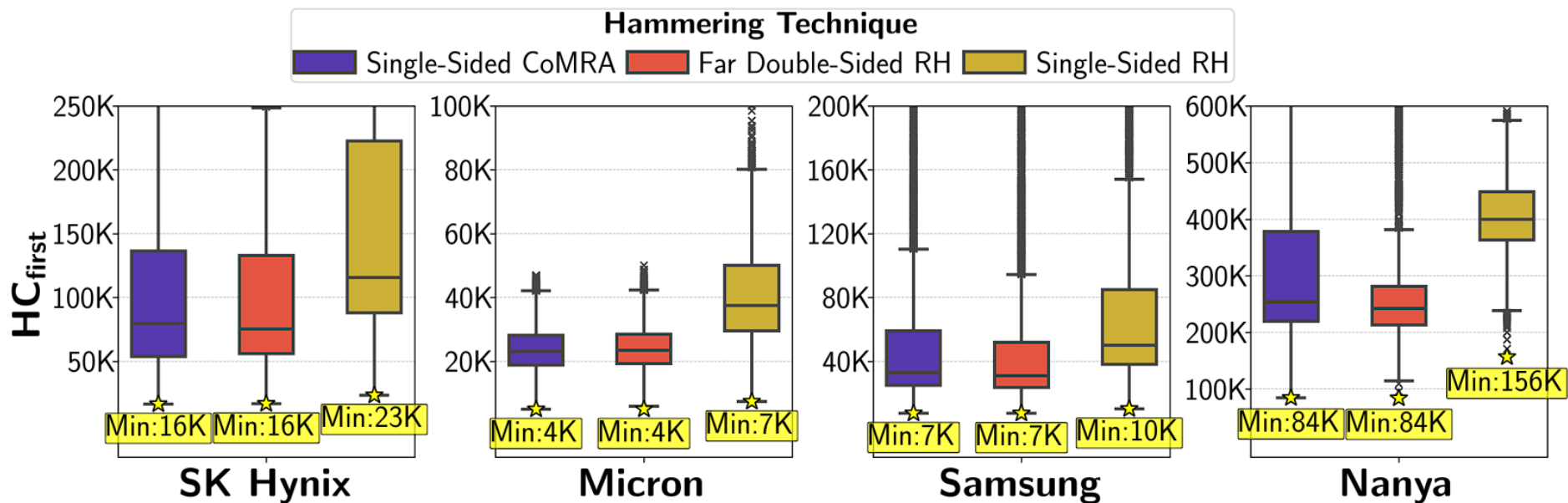
CheckerBoard pattern (i.e., 0x55/0xAA) is, in general, the most effective data pattern among the ones tested.

CoMRA – Temperature



Read disturbance effects of hammering with double-sided CoMRA tend to get worse as temperature increases.

Single-Sided CoMRA vs. RowHammer

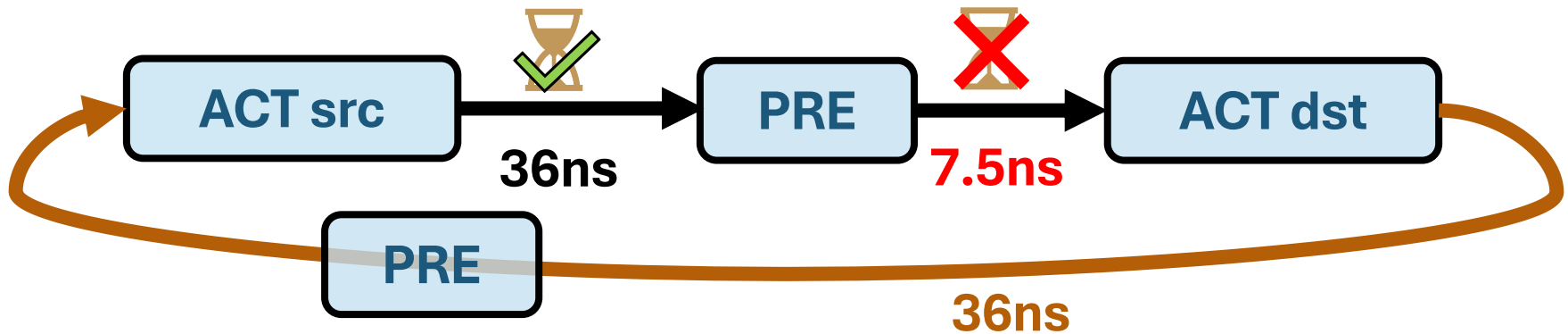


Single-sided CoMRA decreases HCfirst compared to single-sided RowHammer

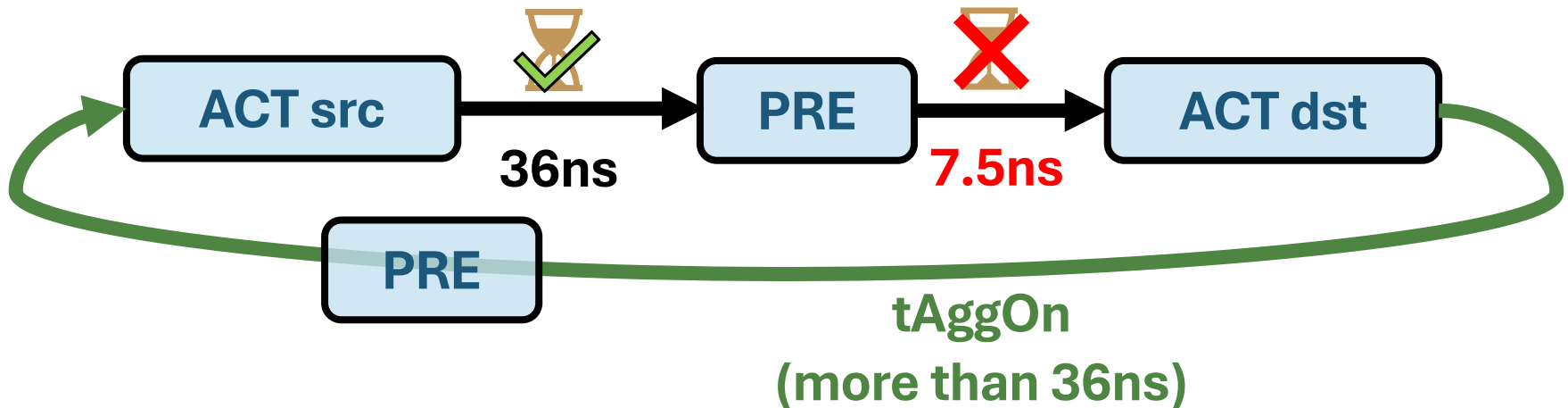
Single-sided CoMRA exhibits similar HCfirst distribution with far double-sided RowHammer.

CoMRA (w/ increased tAggOn)

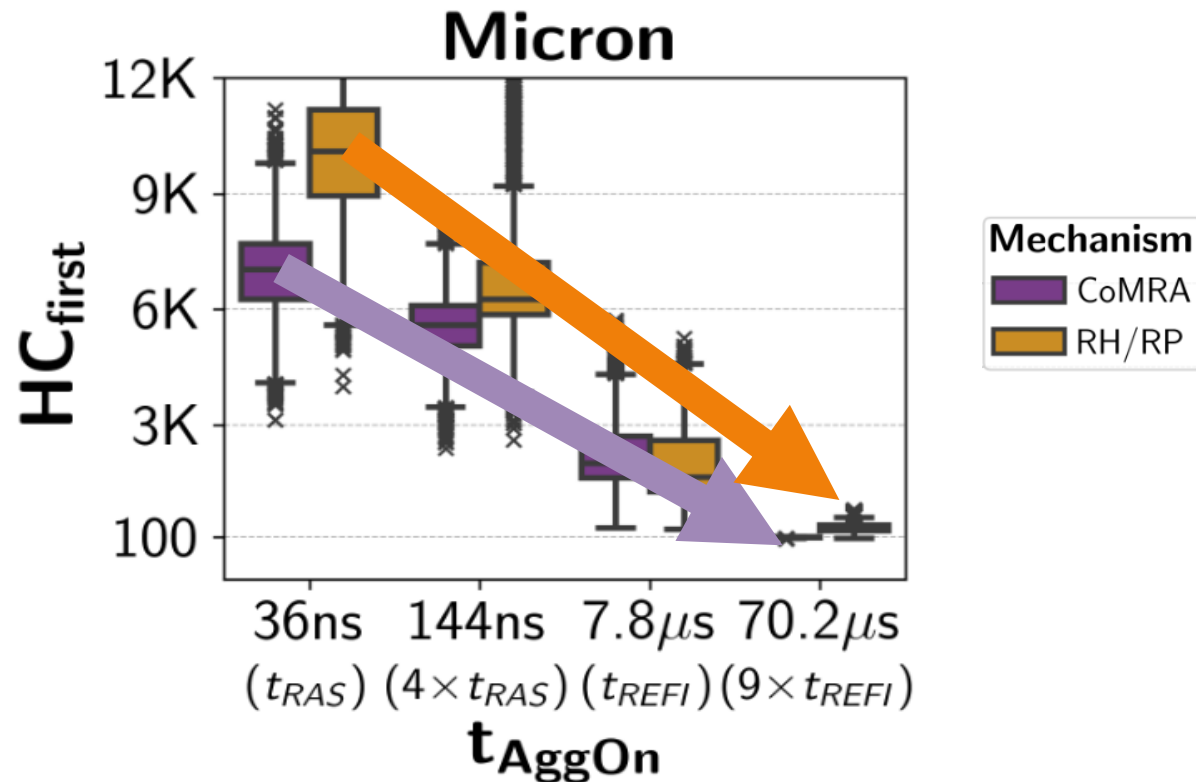
Hammering with CoMRA



Pressing with CoMRA



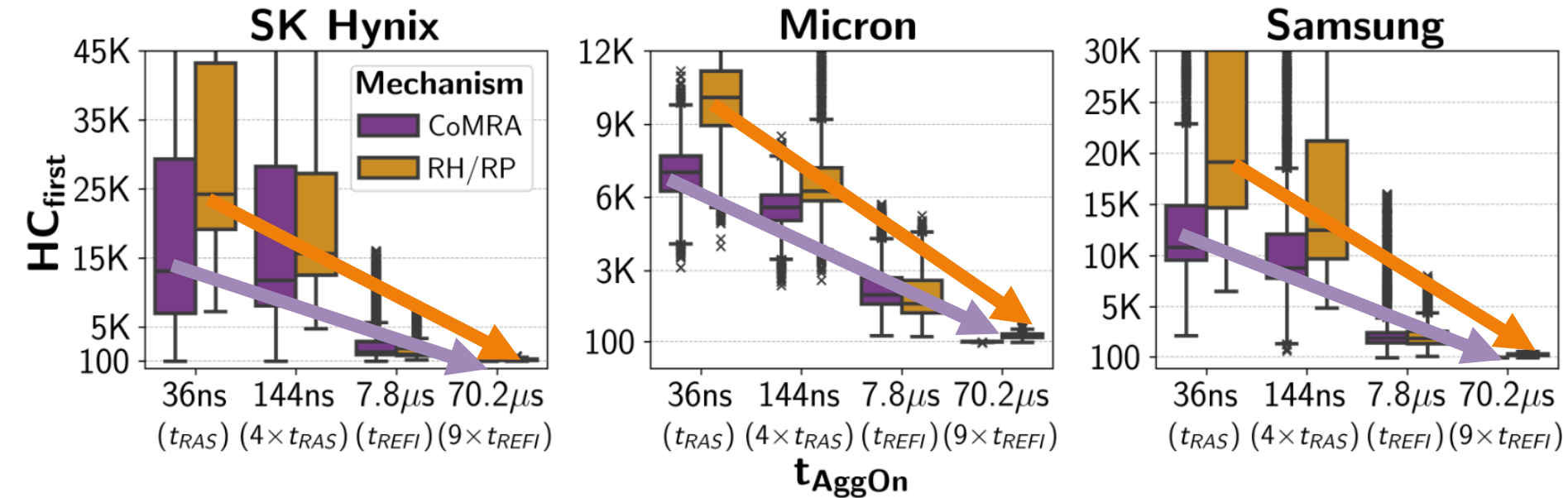
CoMRA (w/ Aggressor Row On Time) vs RowPress



For CoMRA
From 36ns to 70.2us
79x reduction

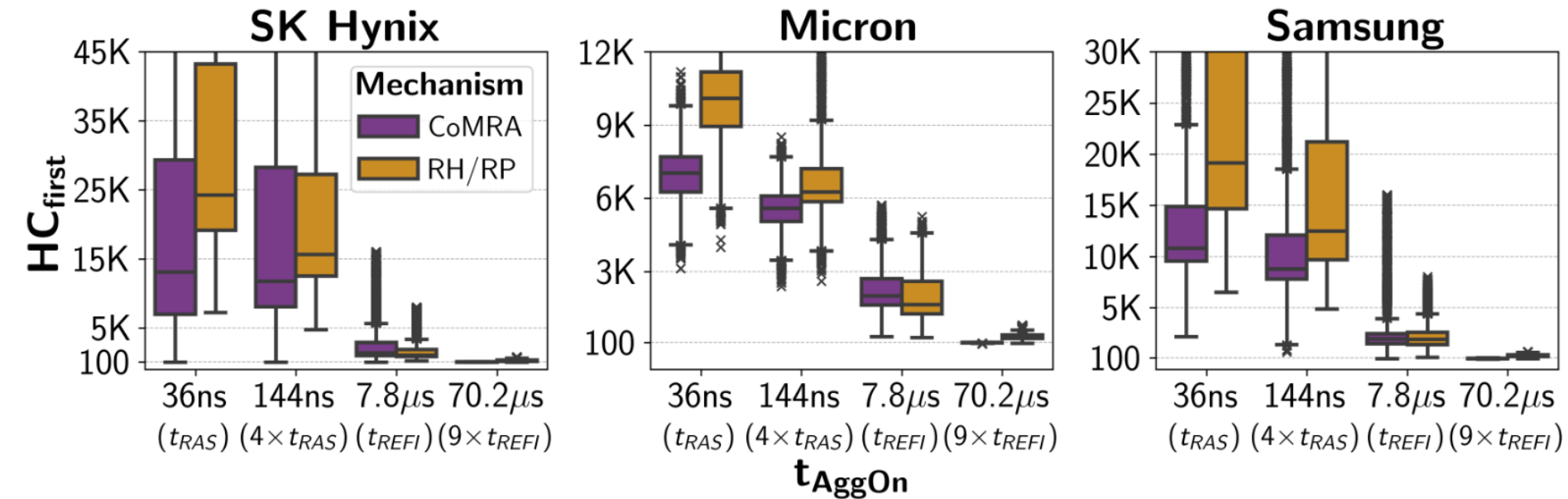
For RH/RP
From 36ns to 70.2us
31x reduction

CoMRA (w/ Aggressor Row On Time) vs RowPress



Trend consistent across all four manufacturer:
Increasing t_{AggOn} significantly reduces HC_{first}
for both CoMRA and RH/RP

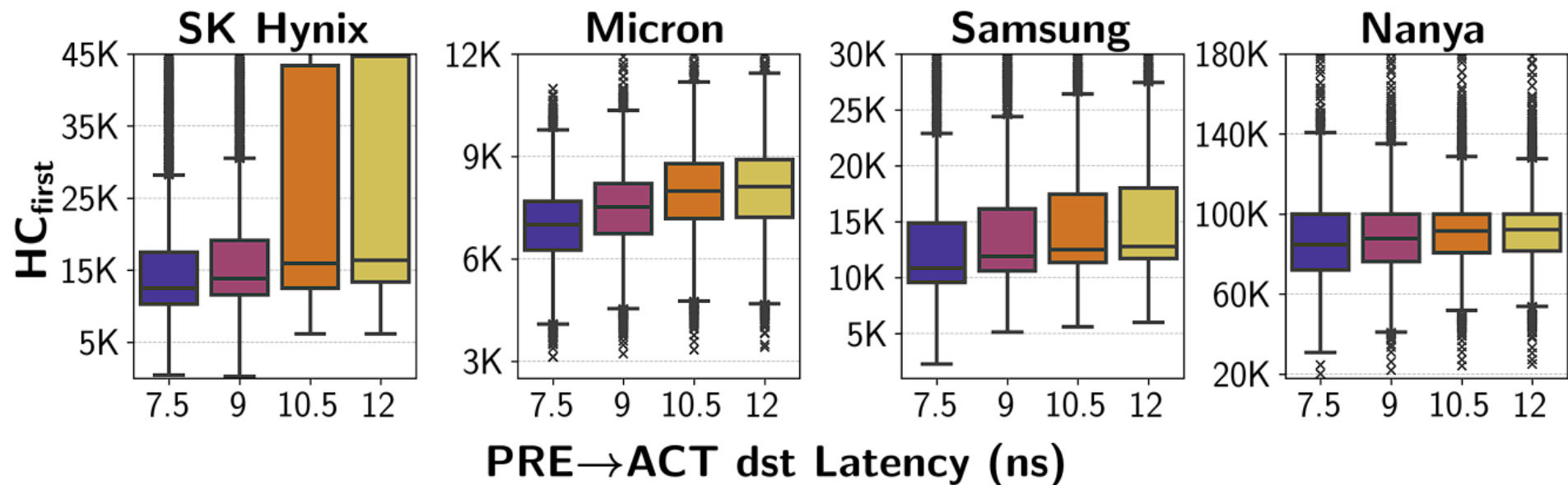
CoMRA (w/ Aggressor Row On Time) vs RowPress



Key Takeaway 2

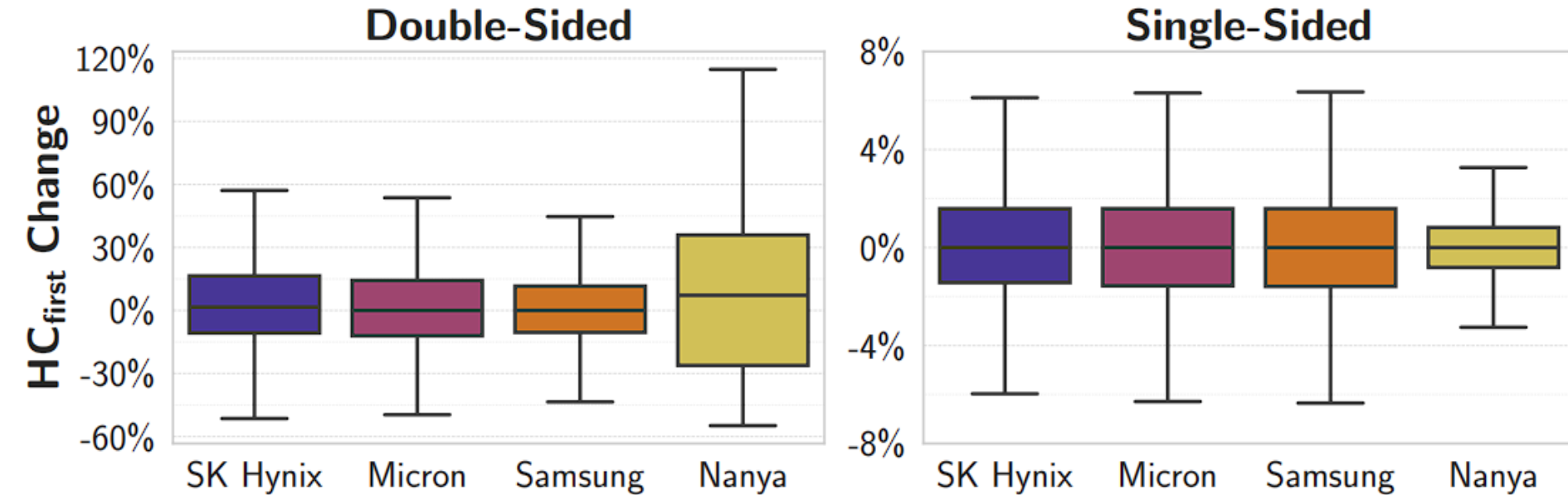
Pressing with CoMRA is more effective than hammering with CoMRA

CoMRA – Timing Delay



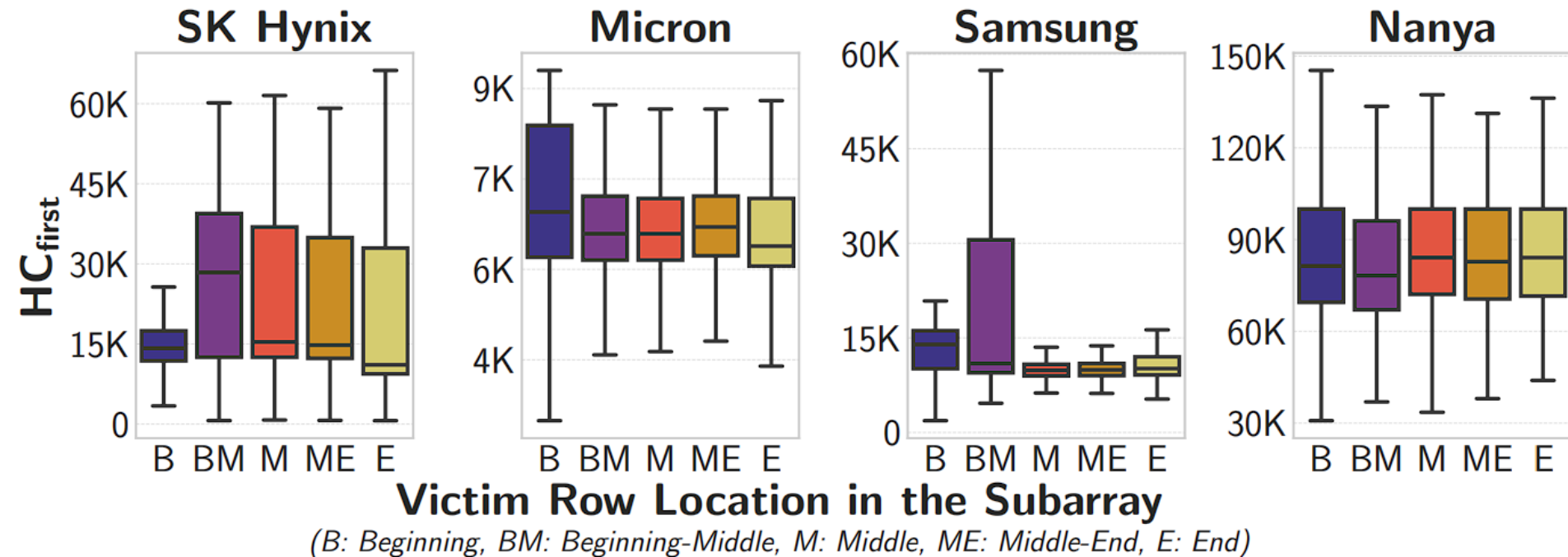
HC_{first} increases as the latency of PRE→ACT dst increases

CoMRA – Copy Direction



In most cases, copy direction has a small effect on HCfirst.

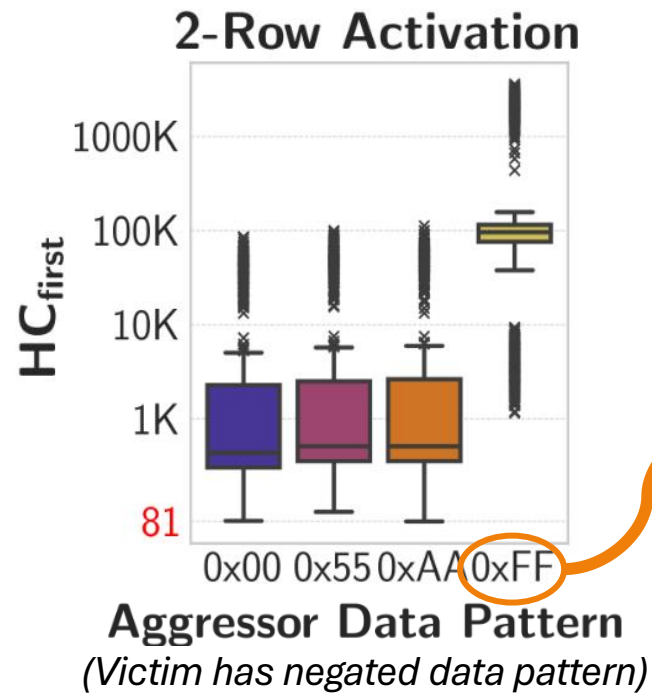
CoMRA – Spatial Variation



HCfirst varies based on
the victim row's location in a subarray

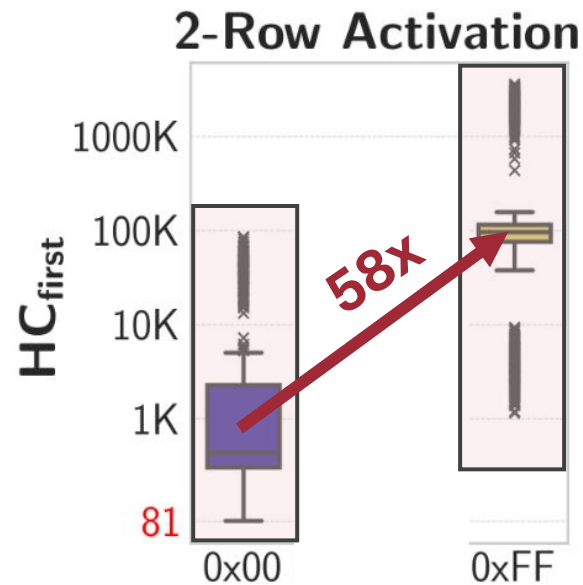
SiMRA Results

SiMRA -- Impact of the Data Pattern



Aggressor Rows = 0xFF..FF
Victim Rows = 0x00..00

SiMRA -- Impact of the Data Pattern



Aggressor Data Pattern

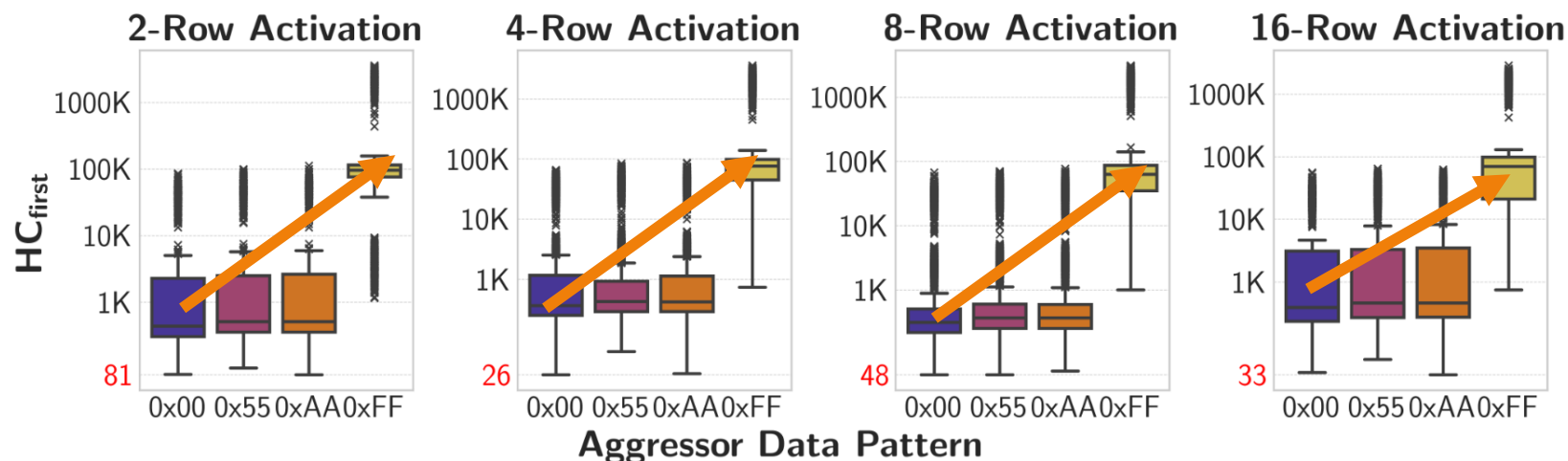
(Victim has negated data pattern)

Initializing victim rows with 0x00
increases avg. HCfirst by 58x

Dominant bitflip direction for SiMRA
1->0

Dominant bitflip direction for RH*
0->1

SiMRA -- Impact of the Data Pattern

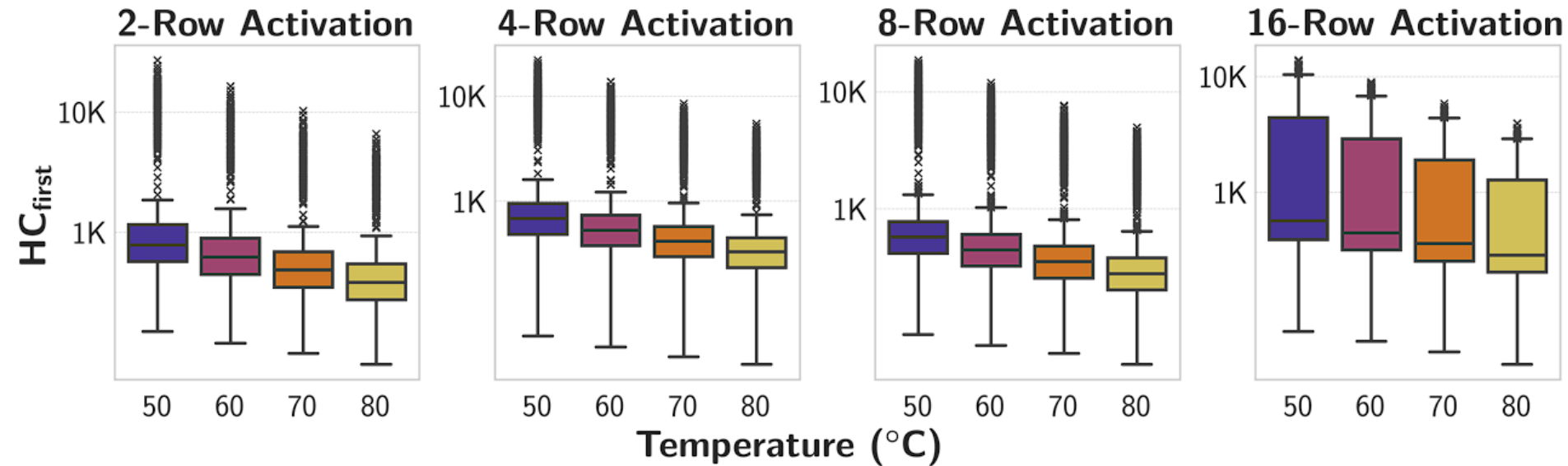


Trend consistent across all tested number of simultaneously activated rows

Key Takeaway

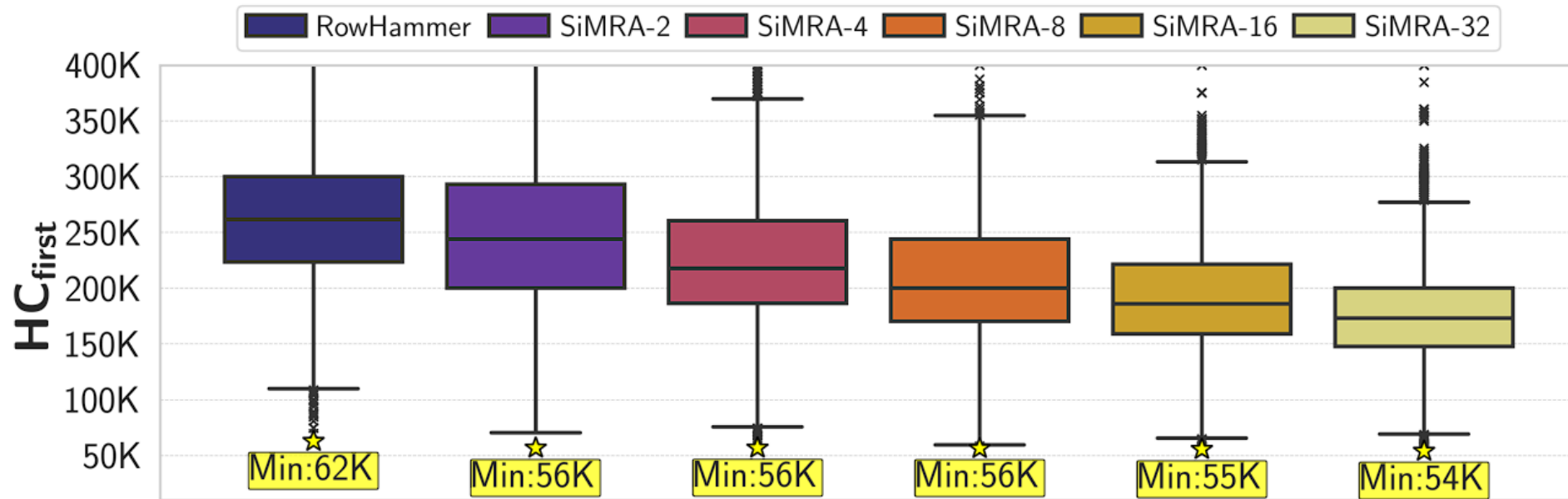
SiMRA is significantly affected by data pattern and directionality of SiMRA and RowHammer bitflips are opposite

SiMRA – Temperature



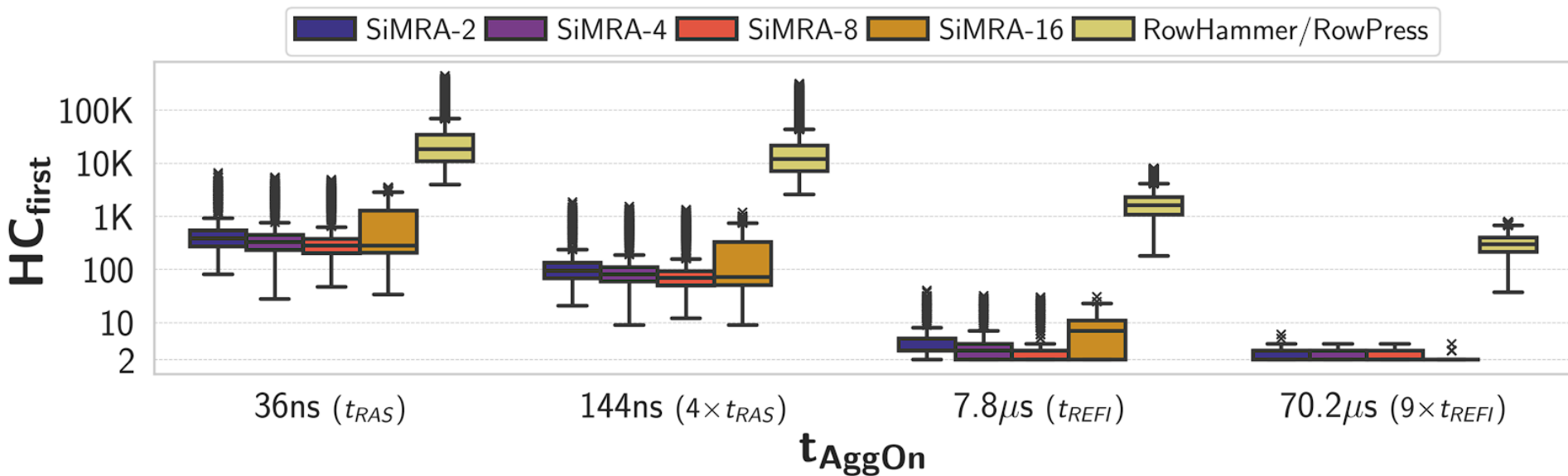
HCfirst decreases as temperature increases

Single-Sided SiMRA vs. RowHammer



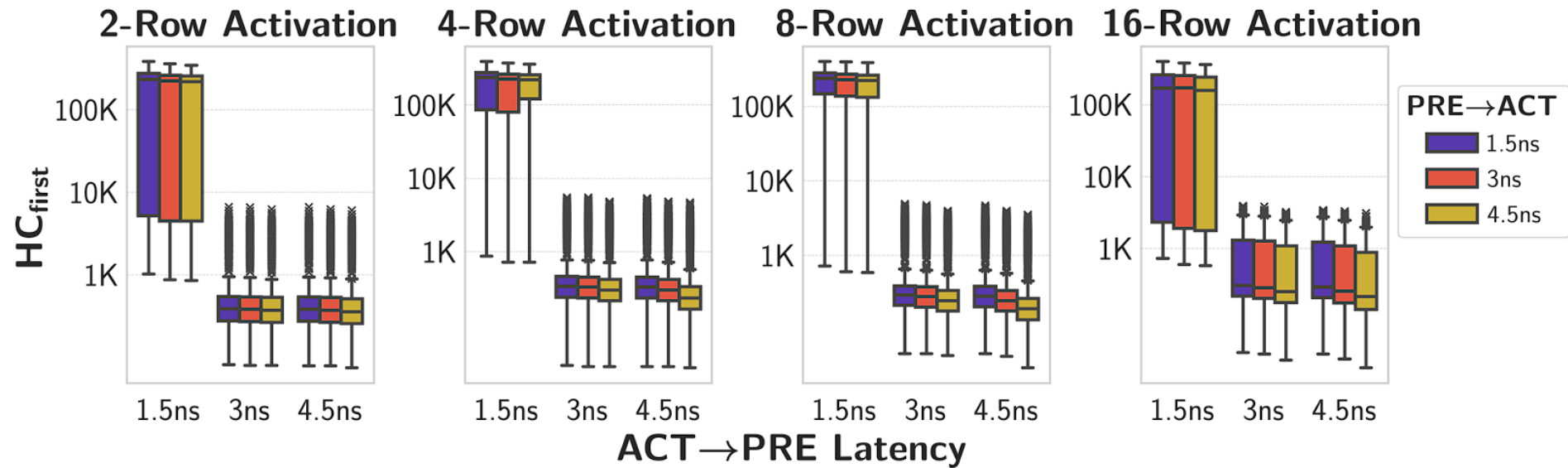
HC_{first} consistently decreases as the number of simultaneously activated rows increases when performing single-sided SiMRA.

SiMRA vs. RowPress



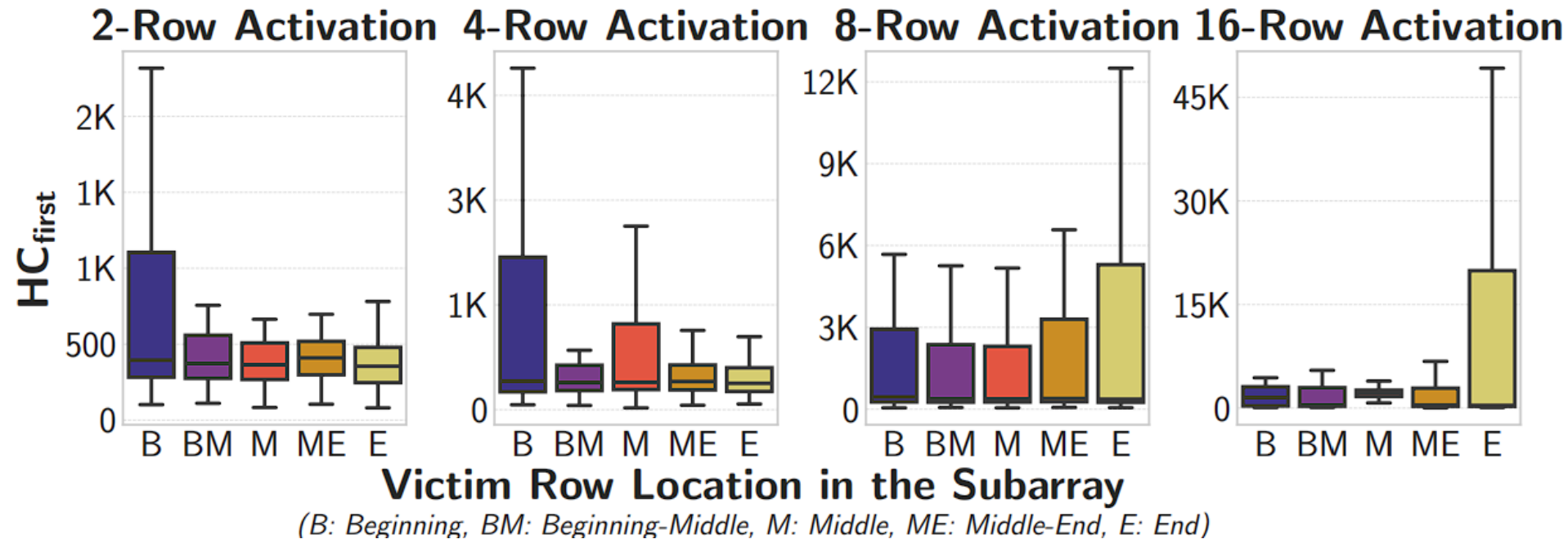
HCfirst decreases greatly as t_{AggOn} increases

SiMRA -- Timing Delay



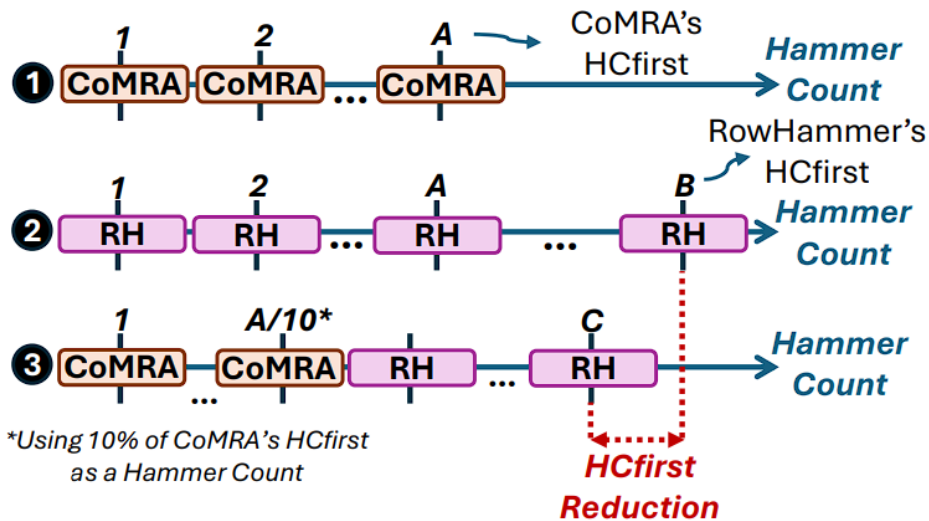
HC_{first} decreases greatly as tAggOn increases

SiMRA – Spatial Variation

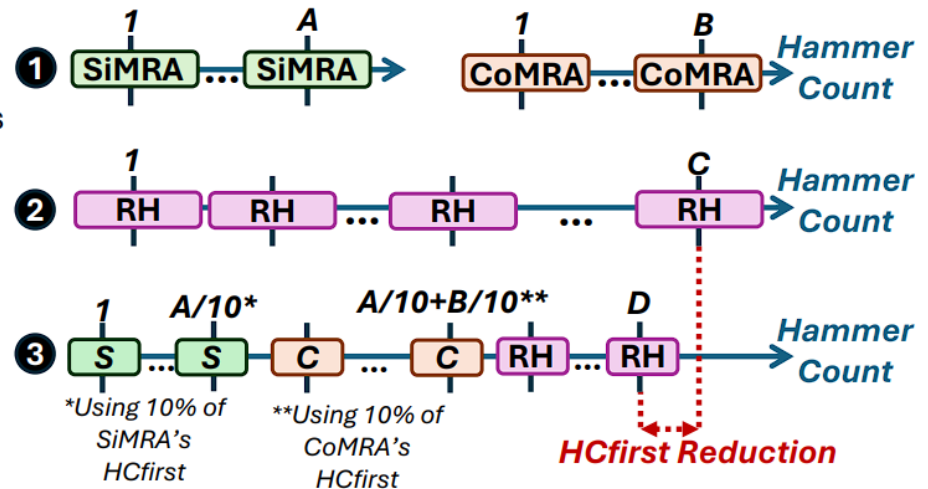


The effectiveness of SiMRA depends on the victim row's location in a subarray.

Combined PuDHammer, RH & RP

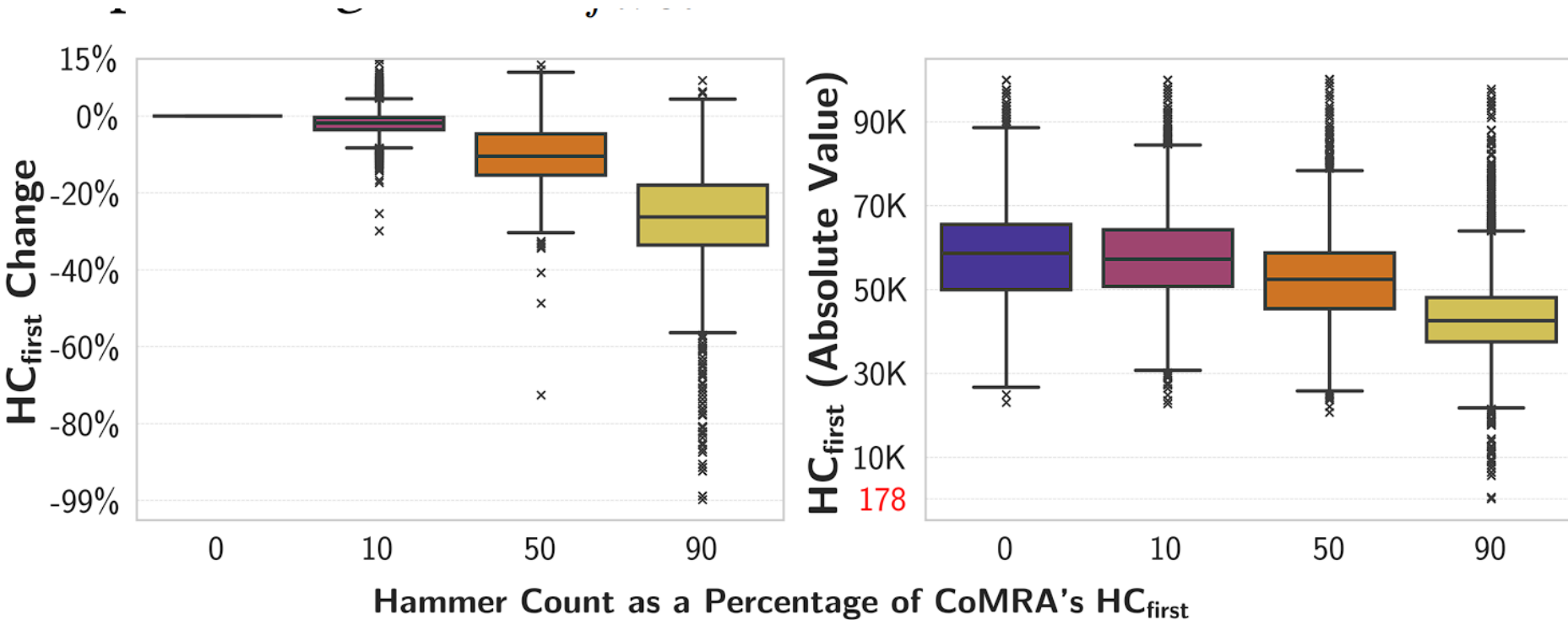


(a) Access Pattern for CoMRA (1), RowHammer (2), and Combined RowHammer and CoMRA (3)



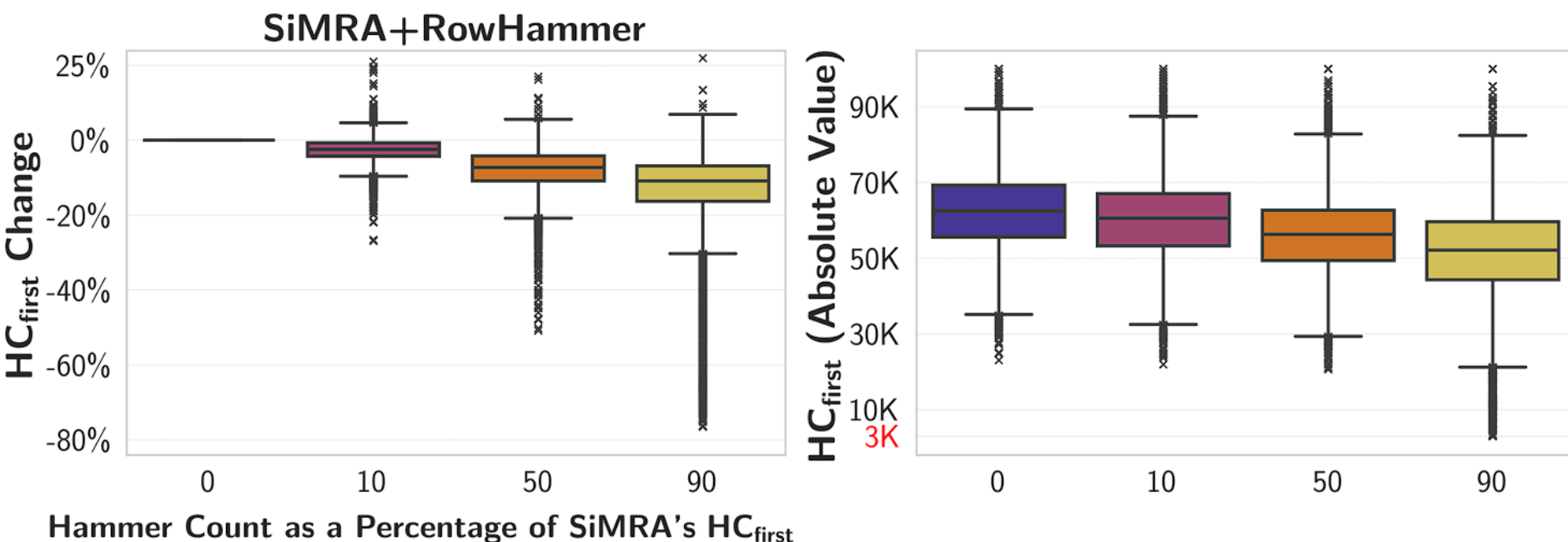
(b) Access Pattern for SiMRA and CoMRA (1), RowHammer (2), and Combined RowHammer, SiMRA and CoMRA (3)

Combined CoMRA & RH



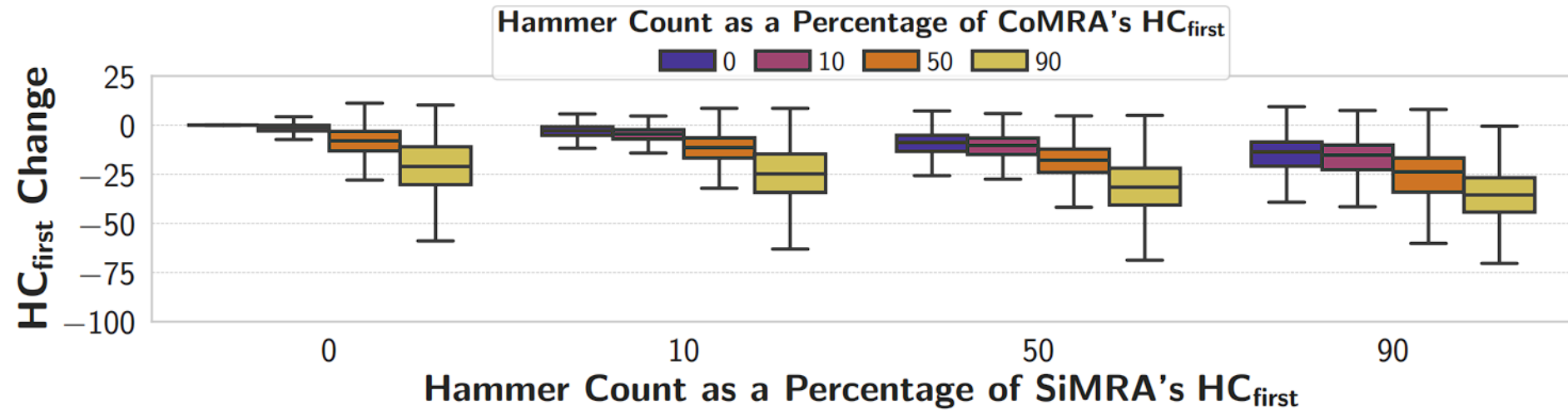
DRAM rows experience significantly lower HC_{first} when performing combined RowHammer and CoMRA.

Combined SiMRA & RH



DRAM rows tend to experience lower HC_{first} when performing combined RowHammer and SiMRA.

Combined CoMRA, SiMRA & RH



Combined RowHammer with CoMRA & SiMRA together is the most effective combined access pattern to decrease HC_{first}.

Reverse Engineering TRR in Real DRAM

Key idea: Use data retention failures as a side channel to detect when a row is refreshed by on-die mitigation

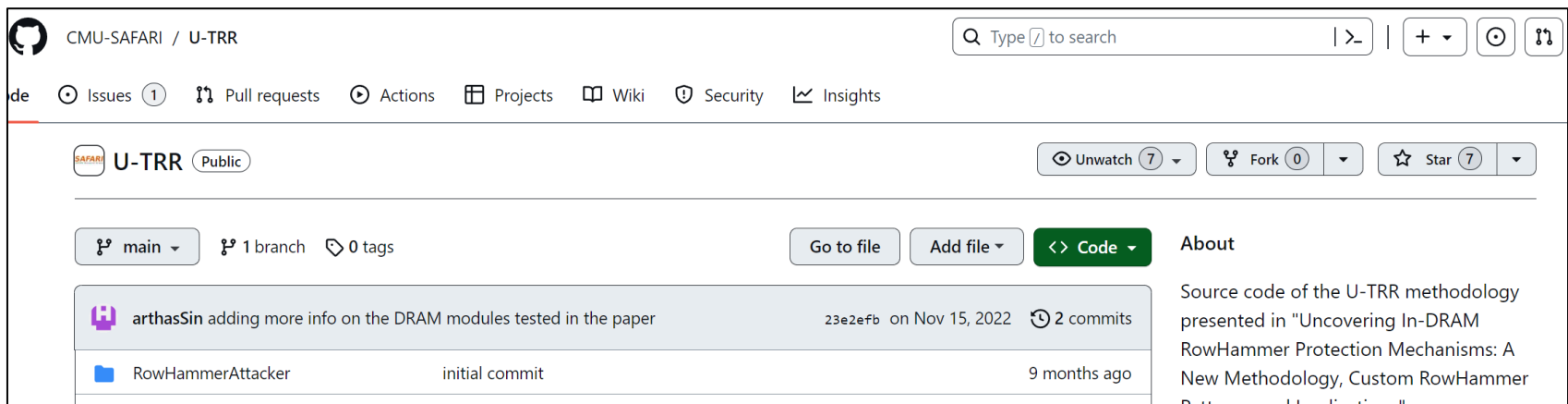
Uncovering In-DRAM RowHammer Protection Mechanisms: A New Methodology, Custom RowHammer Patterns, and Implications

Hasan Hassan[†]
[†]*ETH Zürich*

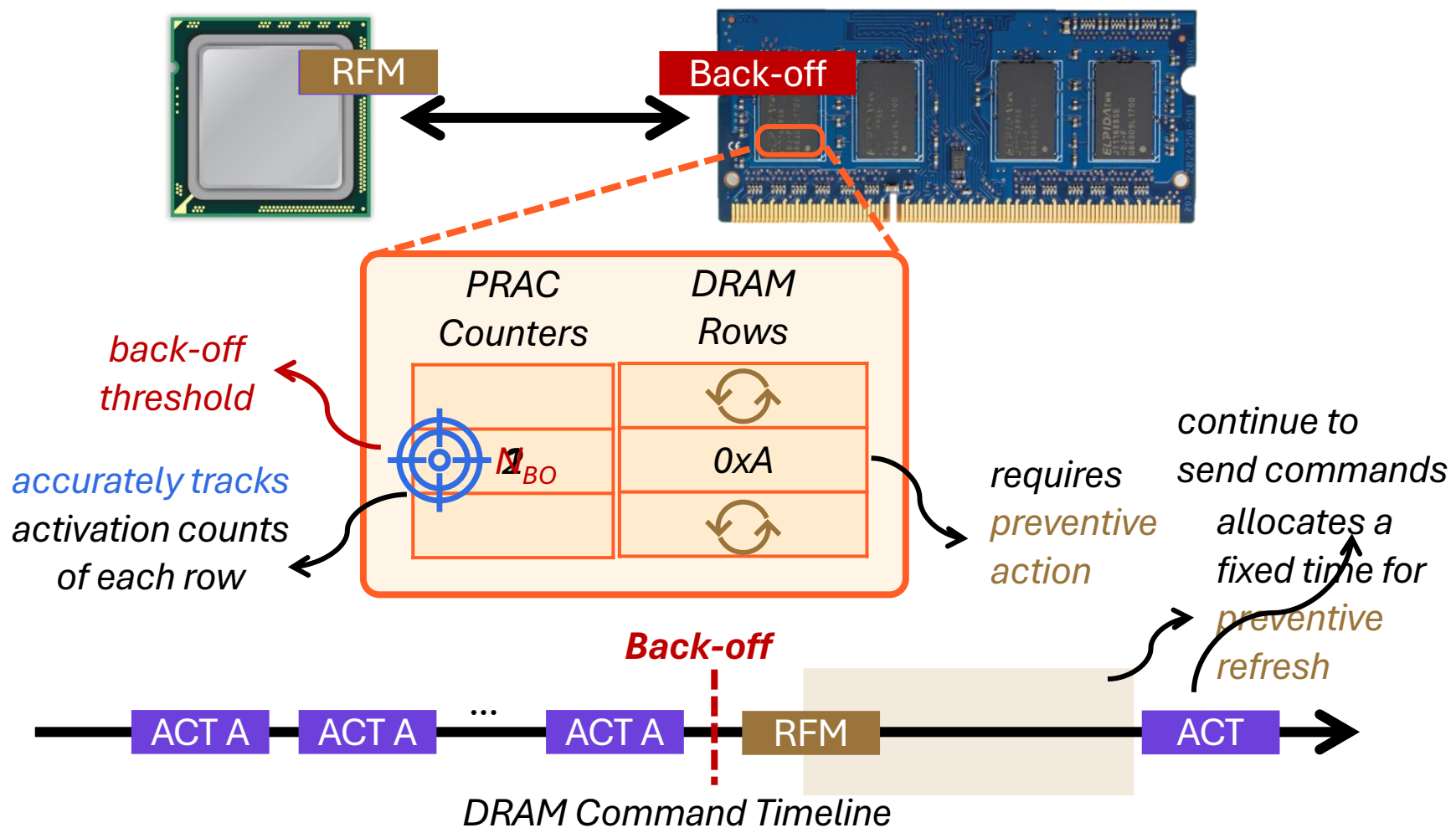
Yahya Can Tuğrul^{†‡}
Kaveh Razavi[†]
[‡]*TOBB University of Economics & Technology*

Jeremie S. Kim[†]
Onur Mutlu[†]

Victor van der Veen^σ
^σ*Qualcomm Technologies Inc.*



Per Row Activation Counting in DDR5 (April 2024)



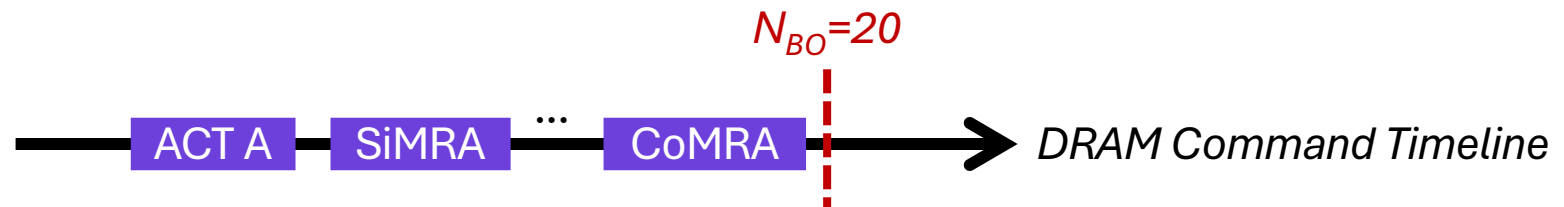
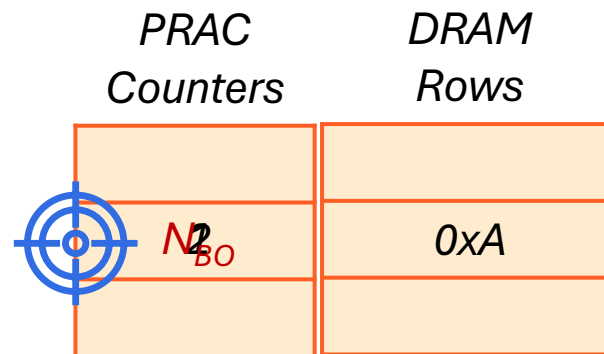
A Naïve Approach (PRAC-Naïve)

Lowest HCfirst for tested DRAM chips

RH: ~4K

CoMRA: ~400

SiMRA: ~20



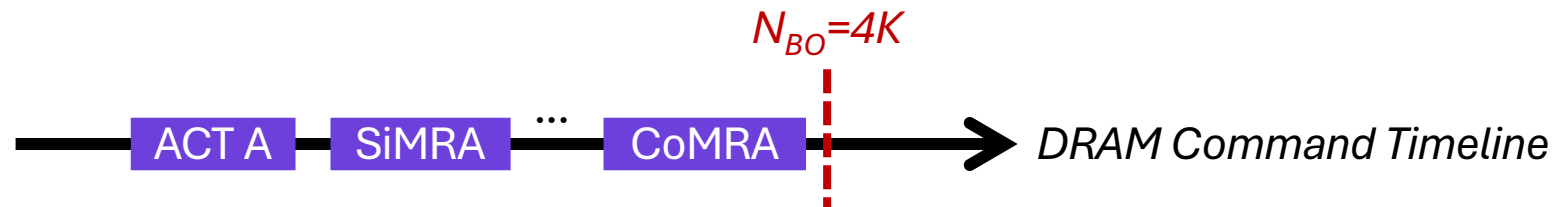
Weighted Counting (PRAC-WC)

Threshold is set to 4K (RH Threshold)
Each operation increases counters by

RH (ACT): 1

CoMRA: 10

SiMRA: 200



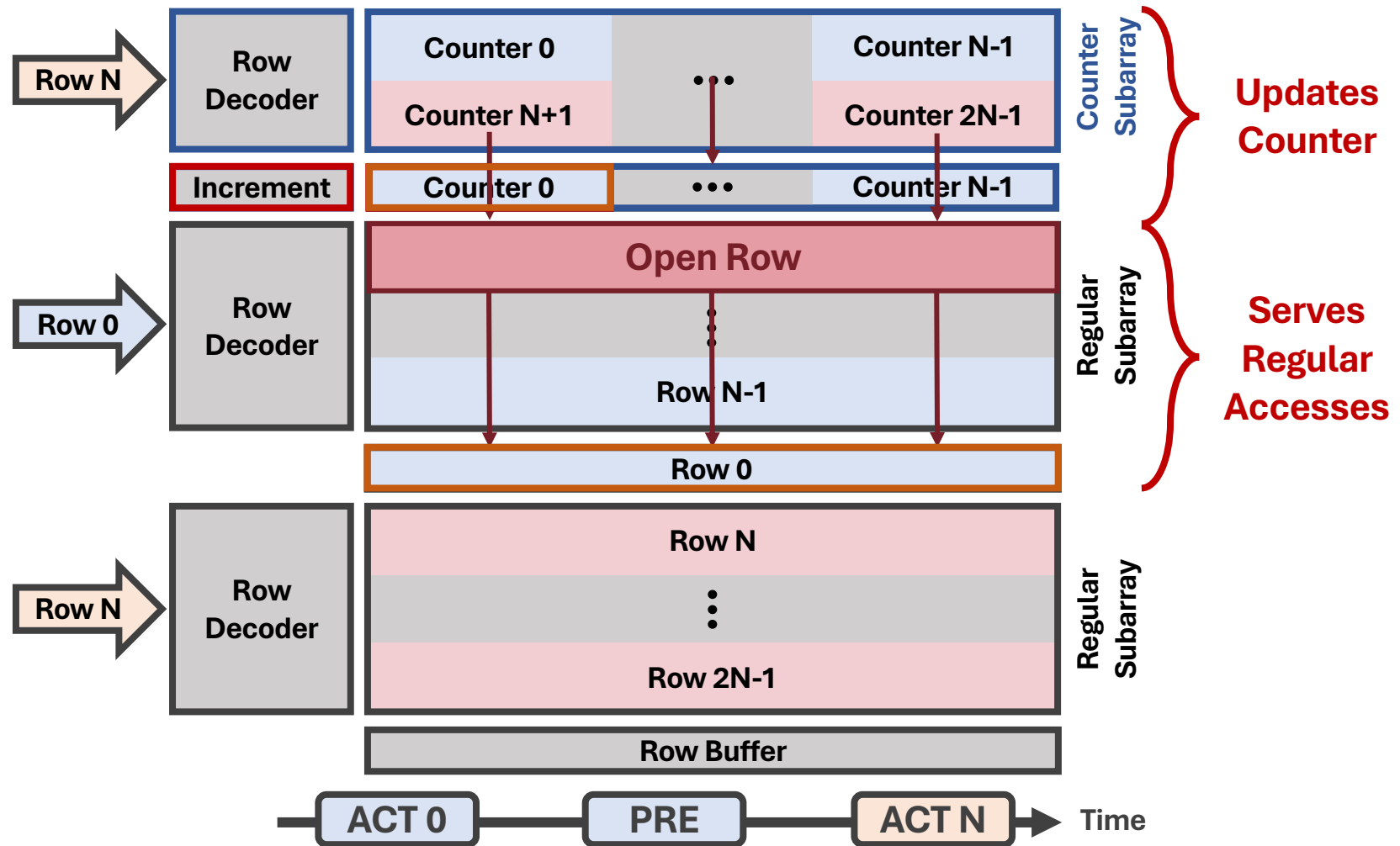
PRAC Counters	DRAM Rows
 201 N_{BO}	0xA

Updating Multiple Counters in PRAC

- SiMRA operation activates multiple rows simultaneously
 - Which requires updating multiple PRAC row activation counters simultaneously
- However, PRAC is designed to update only one counter upon an ACT command
 - We need to modify the counter organization and operation in PRAC to support updating multiple counters.
- Assuming we can activate up to N rows simultaneously
 - Simultaneous access to N different counter values
 - N incrementer circuits to update all the counters simultaneously

A Solution: Chronus + N Incrementers

Chronus concurrently updates counters while serving accesses

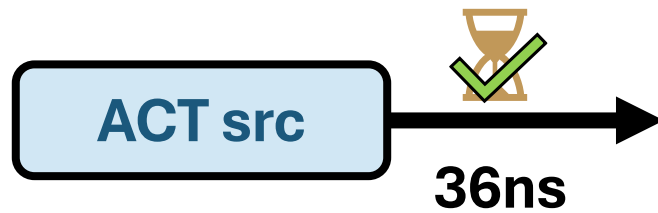


Limitations of Tested COTS DRAM Chips

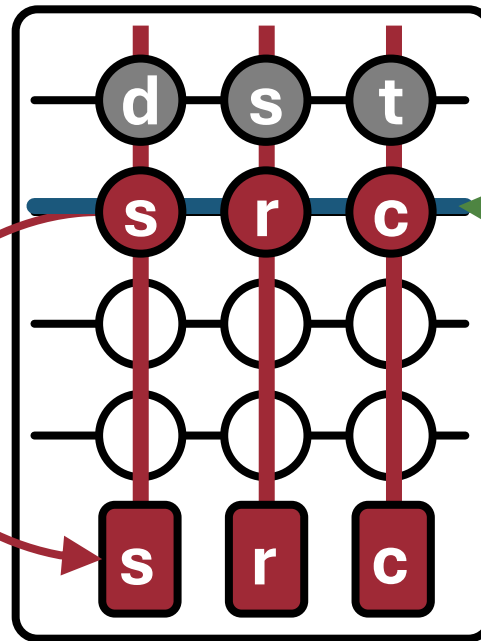
- **Some COTS DRAM chips do not support SiMRA**
 - We do not observe SiMRA in all tested Samsung/Micron chips
- Hypothesis
 - Internal DRAM circuitry ignores the PRE command or the second ACT command when the timing parameters are greatly violated

If such a limitation were not imposed, we believe these DRAM chips are also fundamentally capable of performing the operations we examine in this work

In-DRAM Row-Copy (RowClone)

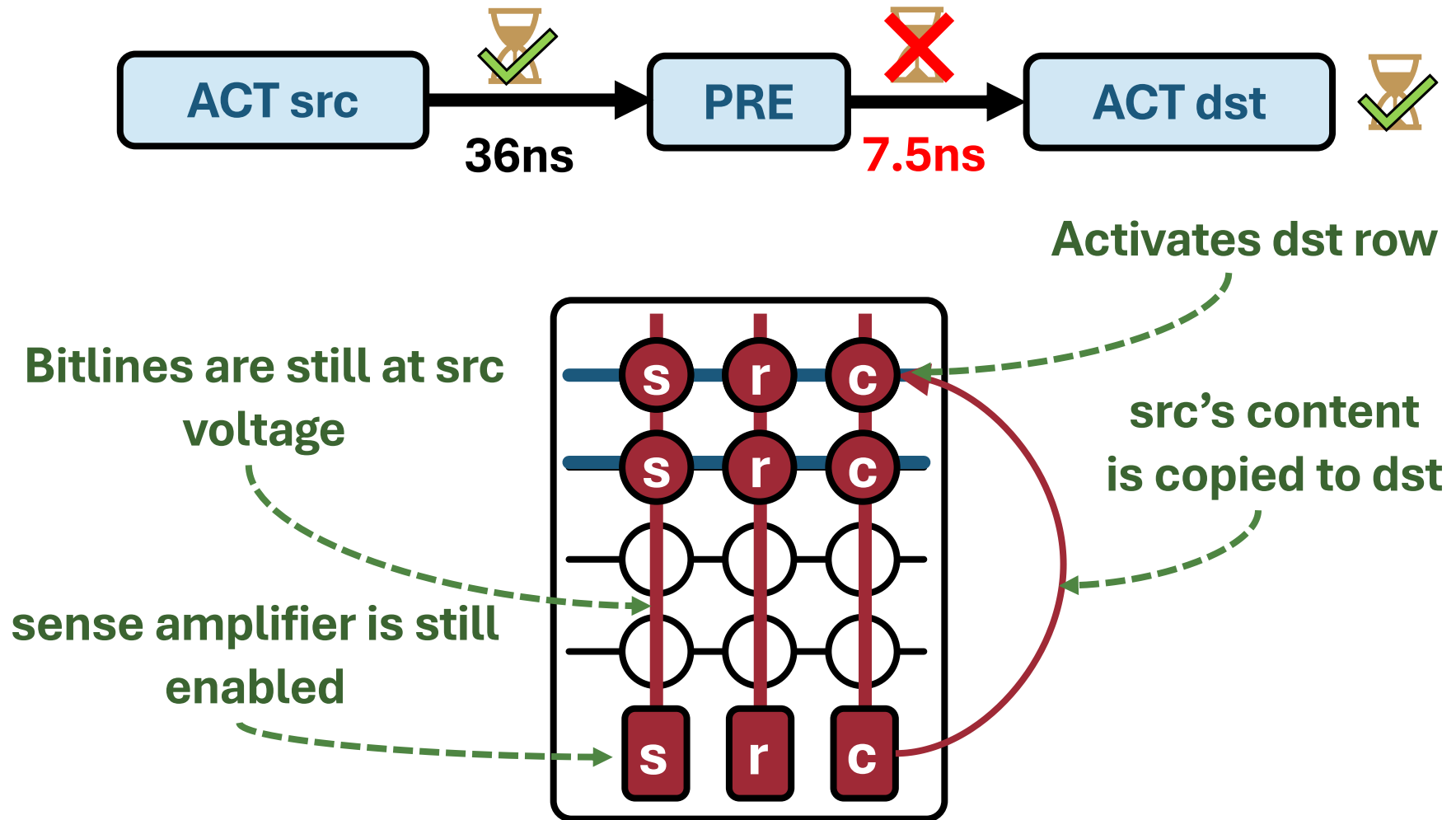


Fetch src's content
into the sense amplifiers

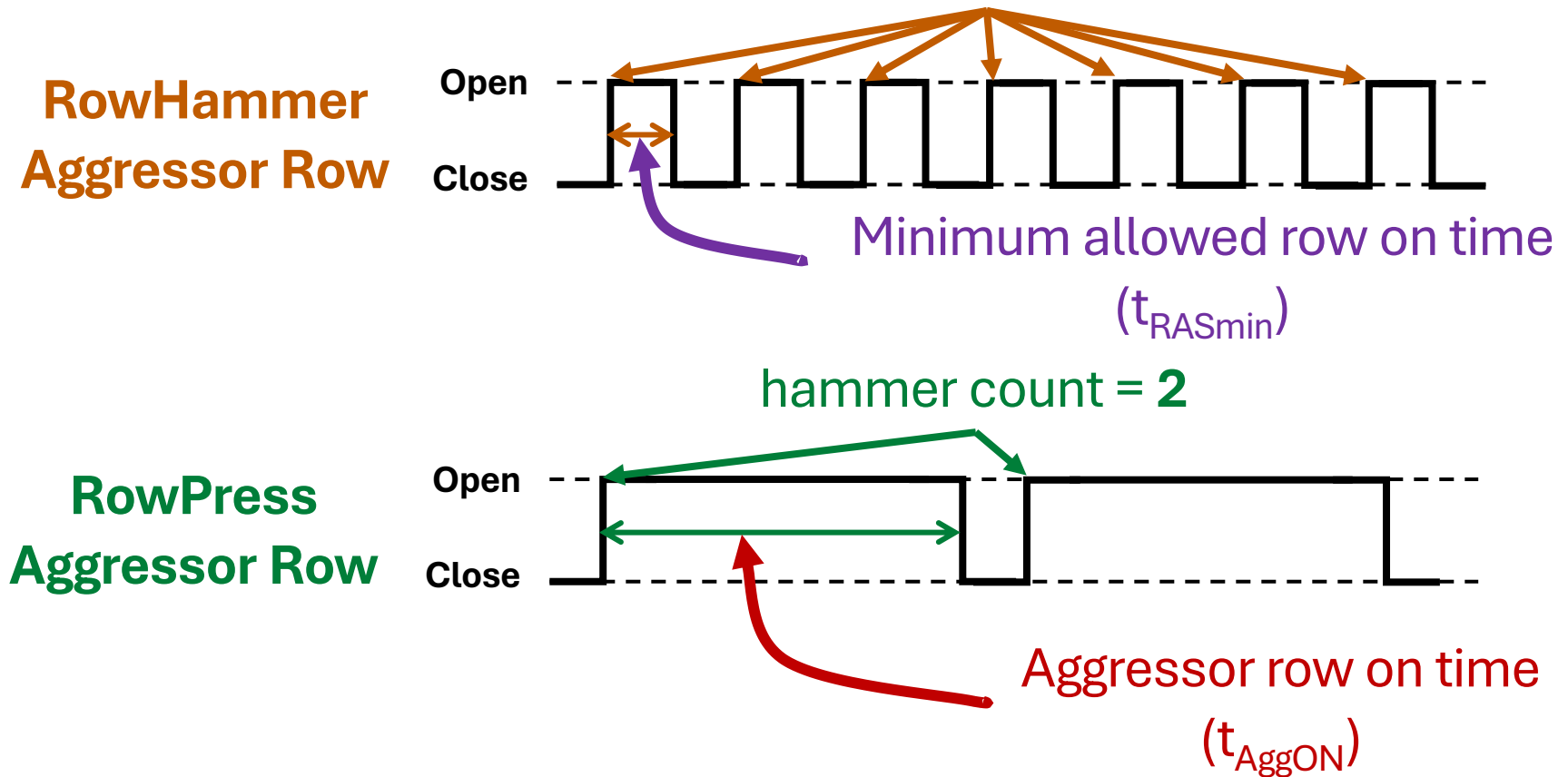


Assert the src's
wordline

In-DRAM Row-Copy (RowClone)



Aggressor Row On Time & RowPress



Instead of using a high hammer count,
👉 increase the time that the aggressor row stays open