



Code
Reproducible



Dataset
Reproducible

ScaleDisturb: Exploiting Temporal Asymmetry to Amplify Read Disturbance in Modern DRAM Chips

Jikun Wang

Haocong Luo Ataberk Olgun Ismail Emir Yüksel A. Giray Yağlıkçı
Yu Liang F. Nisa Bostanci Mohammad Sadrosadati Onur Mutlu

SAFARI

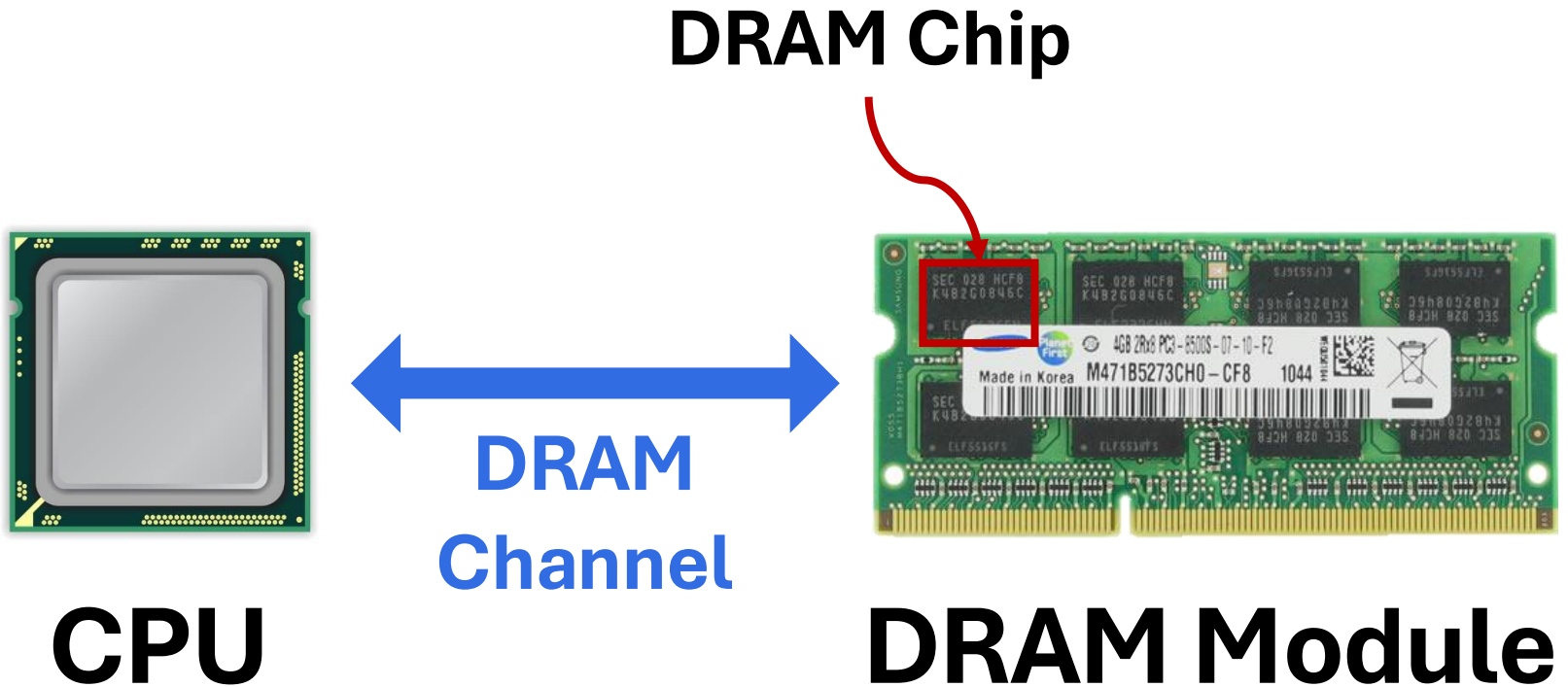
ETH zürich



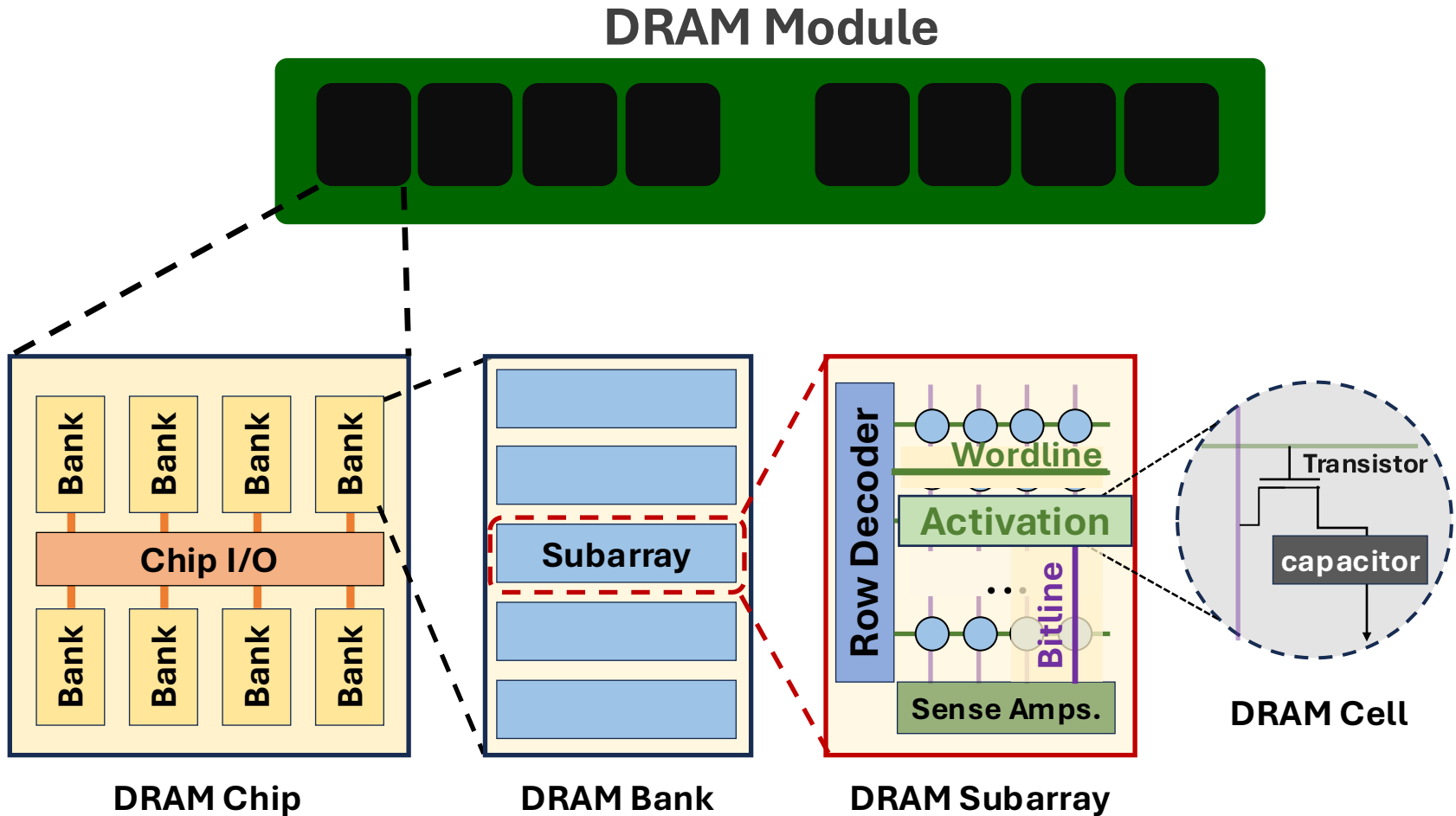
CISPA
HELMHOLTZ CENTER FOR
INFORMATION SECURITY

Inria

A Typical DRAM-Based Computing System

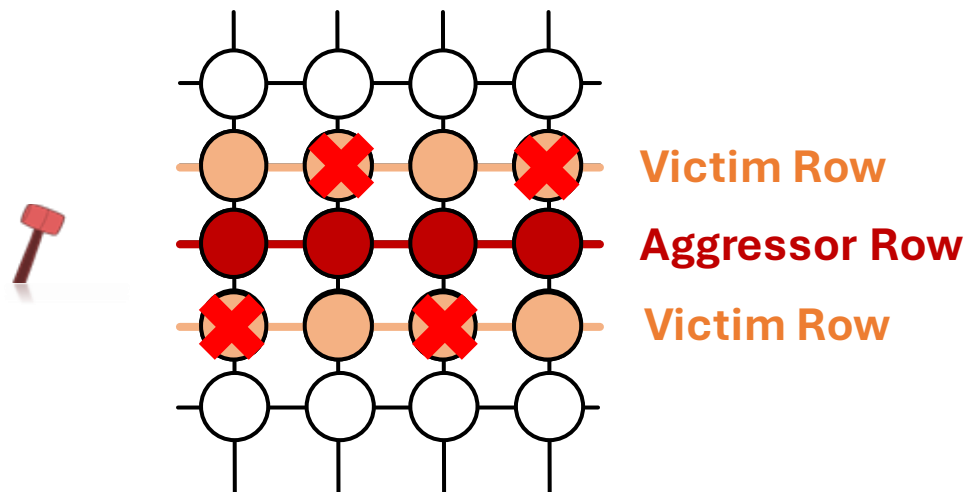


DRAM Organization



DRAM Read Disturbance (I): RowHammer

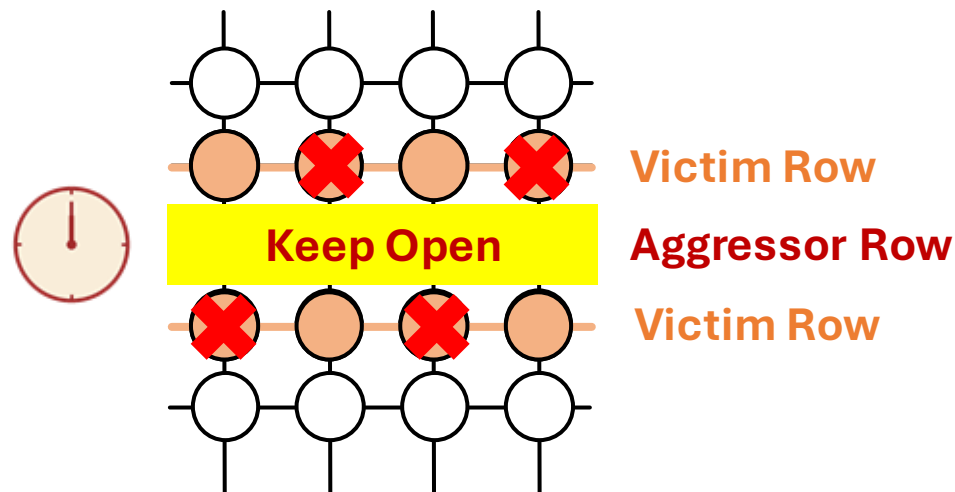
- Read disturbance in DRAM **breaks memory isolation**
- Prominent example: **RowHammer** [Kim+, ISCA 2014]



Repeatedly **opening** and **closing** a DRAM row **many times** causes **bitflips** in physically adjacent victim rows

DRAM Read Disturbance (II): RowPress

- A new read disturbance phenomenon: **RowPress** [Luo+, ISCA 2023]



Keeping a DRAM row **open for a long time** causes **bitflips** in physically adjacent victim rows

Read Disturbance Solutions

There are many solutions to mitigate read disturbance bitflips

- More robust DRAM chips and/or error-correcting codes
- Increased refresh rate
- Physical isolation
- Row remapping
- Preventive refresh
- Proactive throttling

Each solution offers a different system design point in reliability, performance, energy, and area tradeoff space

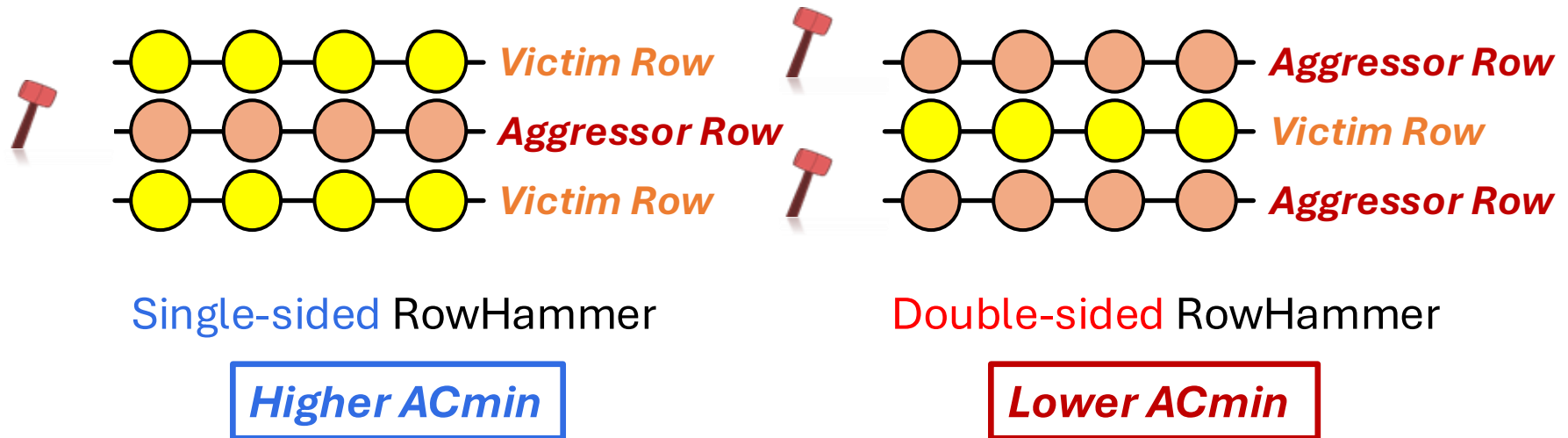
Accurate Identification of Read Disturbance Threshold

- Many secure read disturbance solutions take a **preventive action** before a bitflip manifests
 - E.g., **preventively refresh** a victim row
- Must **accurately quantify** the **amount of disturbance** that a row can **withstand** before experiencing a bitflip
- **Read Disturbance Threshold (ACmin)**
 - The **minimum** aggressor row **activation count** required to induce the first bitflip

To **securely prevent** read disturbance bitflips, the **ACmin** of DRAM rows must be **accurately identified**

DRAM Access Pattern Affects ACmin (I)

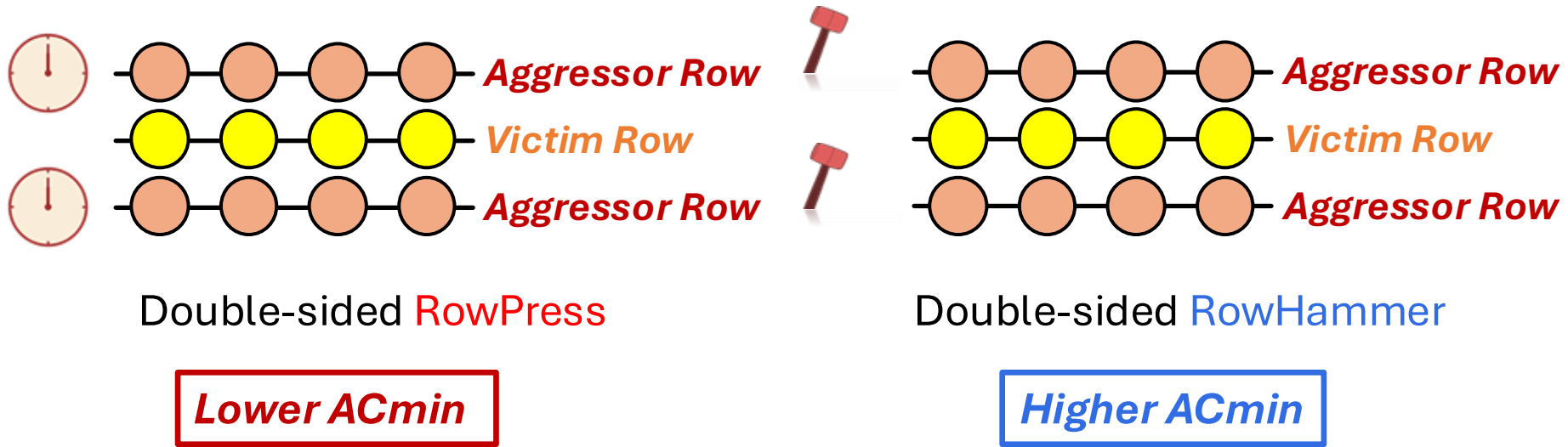
A DRAM row's **ACmin** is strongly **affected** by the **DRAM access pattern** used in real DRAM chip characterization



DRAM is **more vulnerable** to read disturbance under **double-sided** RowHammer than **single-sided** RowHammer

[Kim+, ISCA 2020]

DRAM Access Pattern Affects ACmin (II)



DRAM is **more vulnerable** to read disturbance under double-sided **RowPress** than double-sided **RowHammer**

[Luo+, ISCA 2023]

Problem & Goal

Problem:

- Existing read disturbance characterization works explore DRAM's vulnerability under **only a limited set of DRAM access patterns**
- A more effective access pattern (i.e., one causes bitflips with lower ACmin), if undetected, would lead to **insufficient protection**

Goal:

- To explore a **new DRAM access pattern** that can induce bitflips with **fewer row activations** compared to existing access patterns
- To **evaluate its implications** on modern DRAM-based systems

Talk Outline

ScaleDisturb Access Pattern

Testing Infrastructure

Real DRAM Chip Results

Real System Demonstration of ScaleDisturb

Implications and Mitigations

Conclusion

Talk Outline

ScaleDisturb Access Pattern

Testing Infrastructure

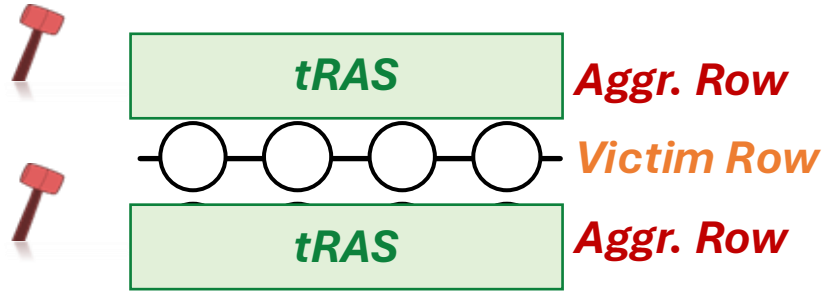
Real DRAM Chip Results

Real System Demonstration of ScaleDisturb

Implications and Mitigations

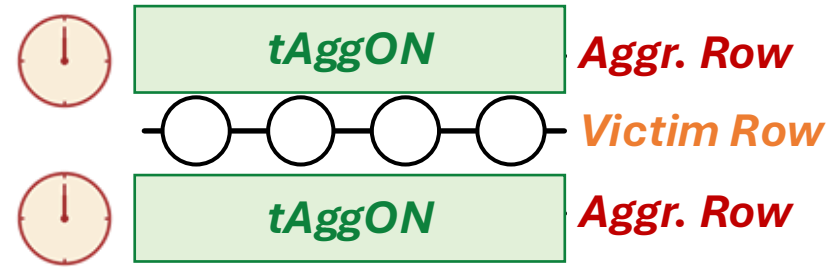
Conclusion

Existing Symmetric Access Pattern



Double-sided
RowHammer

Symmetric open times
 $tRAS$



Double-sided
RowPress

Symmetric open times
 $tAggON$

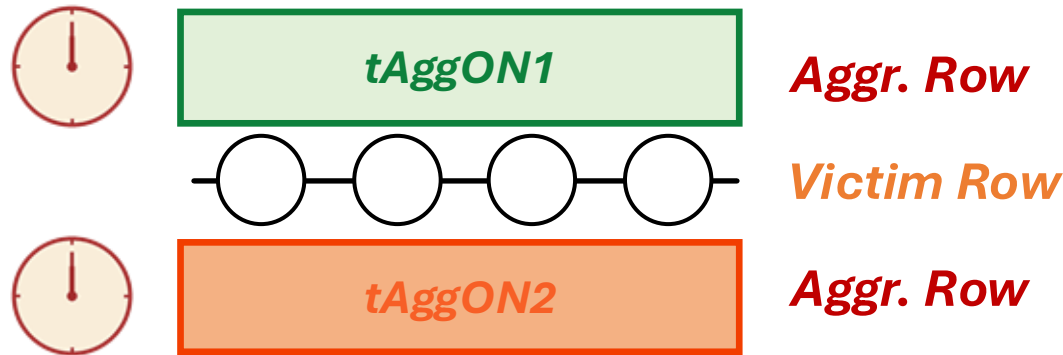
Existing **double-sided DRAM access patterns**
keep the aggressor rows open for **equal durations**

ScaleDisturb Access Pattern

What if the two aggressor rows have different open durations?

ScaleDisturb

$(t_{\text{AggON1}} \neq t_{\text{AggON2}})$



ScaleDisturb is a **new DRAM access pattern** that **asymmetrically extends** the open time of two aggressor rows

Talk Outline

ScaleDisturb Access Pattern

Testing Infrastructure

Real DRAM Chip Results

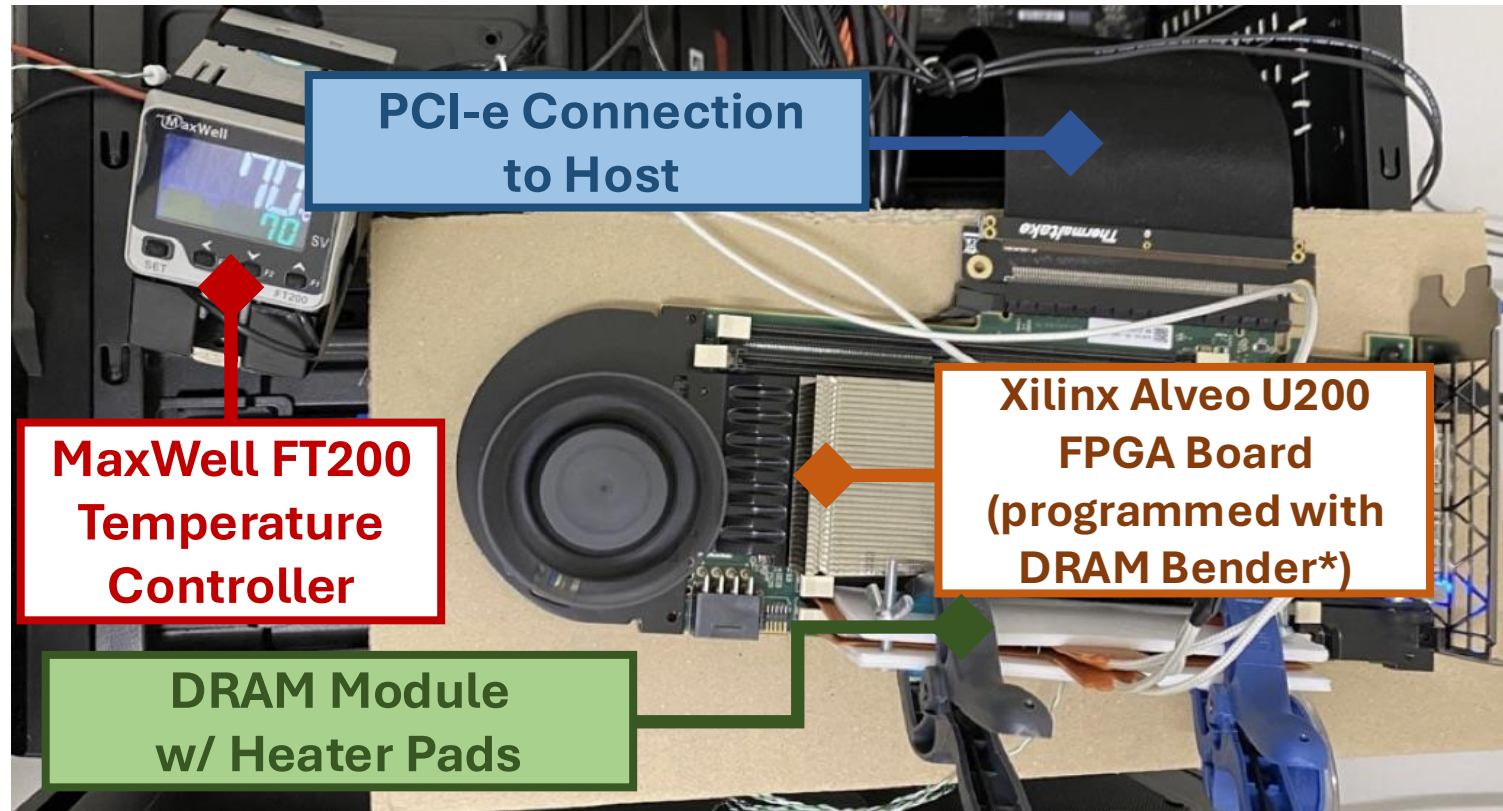
Real System Demonstration of ScaleDisturb

Implications and Mitigations

Conclusion

DDR4 DRAM Testing Infrastructure

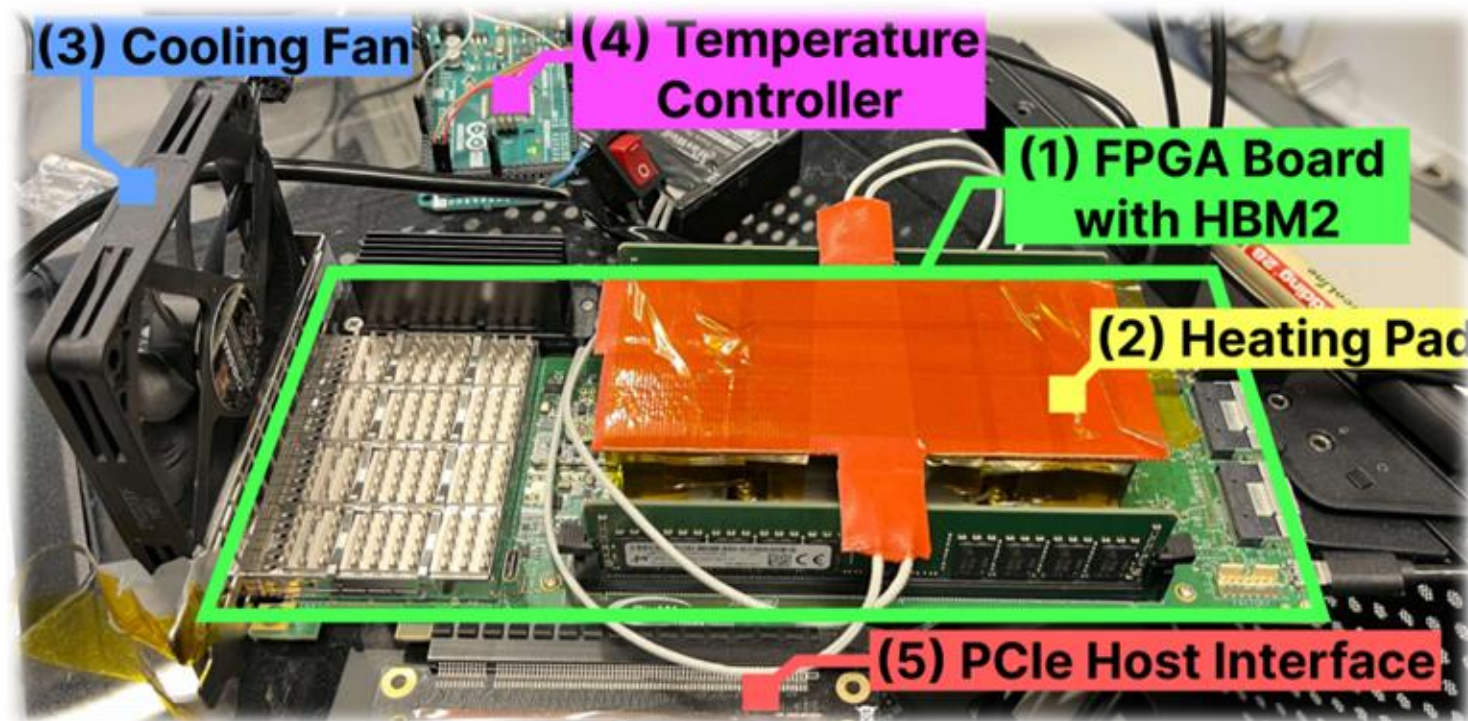
DRAM Bender on a Xilinx Virtex UltraScale+ XCU200



Fine-grained control over **DRAM commands**,
timing parameters ($\pm 1.5\text{ns}$), and **temperature** ($\pm 0.5^\circ\text{C}$)

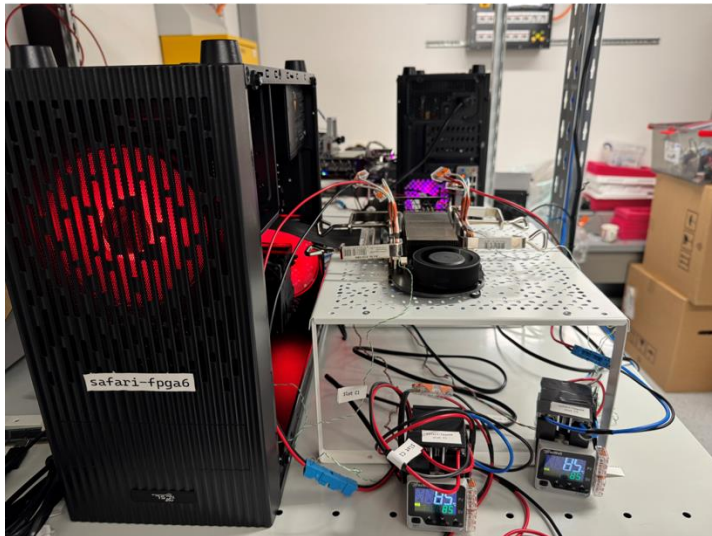
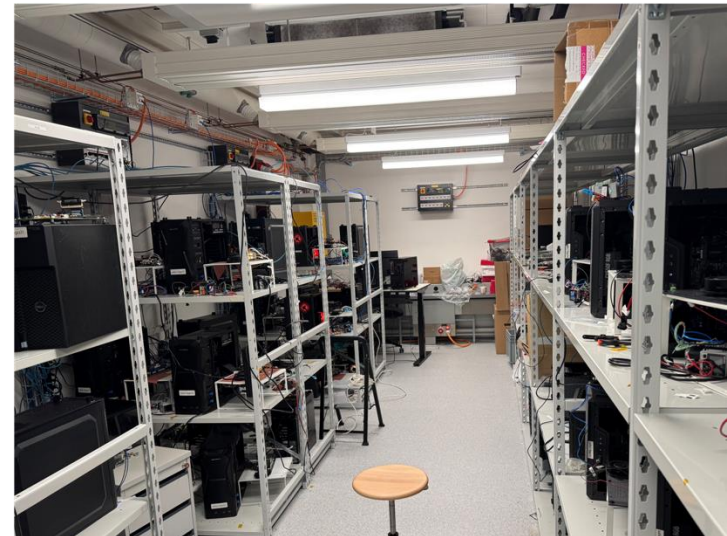
HBM2 DRAM Testing Infrastructure

DRAM Bender on a Bittware XUPVH



Fine-grained control over DRAM commands,
timing parameters ($\pm 1.67\text{ns}$), and temperature ($\pm 0.5^\circ\text{C}$)

Laboratory for Understanding Memory



Tested DRAM Chips

196 DDR4 and 3 HBM2 Chips from Samsung, SK Hynix, and Micron

Chip Mfr.	Module IDs	#Chips	Die Rev.	Density	Org.
Mfr. S (Samsung)	S0, S1	8	B	16Gb	x8
	S2, S3, S4	8	A	16Gb	x8
	S5, S6	8	D	8Gb	x8
	S7	16	C	8Gb	x4
	S8	8	F	4Gb	x8
Mfr. H (SK Hynix)	H0	16	A	16Gb	x8
	H1	8	D	8Gb	x8
	H2	8	C	8Gb	x8
	H3	16	C	8Gb	x8
	H4	8	C	4Gb	x8
Mfr. M (Micron)	M0	16	F	16Gb	x8
	M1	8	F	16Gb	x8
	M2, M3	4	E	16Gb	x16
	M4	4	B	16Gb	x16
	M5	16	E	8Gb	x8
Samsung	HBM2 Chips	3	N/A	N/A	N/A

^a We report “N/A” if the information is not publicly available.

Testing Methodology: Sources of Interference

Prevent sources of interference during core test loop

- **No DRAM refresh**
 - to avoid refreshing victim rows
- **Within a single refresh window**
 - to avoid retention failures
- **No read disturbance mitigation mechanisms**
 - to observe circuit-level effects
- **No error correcting codes (ECC)**
 - to observe all bitflips

Testing Methodology: Metrics & Parameters

Metric: Minimum aggressor row activation count required to induce the first bitflip (**ACmin**)

Lower
is worse

Access patterns: ScaleDisturb, double-/single-sided RowPress, double-/single-sided RowHammer

Row Open Time Budget (OTB) = $t_{AggON1} + t_{AggON2} - 2 * t_{RAS}$

OTB: 0ns, 48ns, 120ns, 300ns, 1200ns, 7800ns

Data Pattern: Rowstripe0 (0x00 in victim and 0xFF in aggressor) and Rowstripe1 (0xFF in victim and 0x00 in aggressor)

Temperature: 50°C and 80°C

Algorithm: Bisection-based ACmin search

ScaleDisturb: Exploiting Temporal Asymmetry to Amplify Read Disturbance in Modern DRAM Chips

Jikun Wang[§] Haocong Luo[§] Ataberk Olgun[§] İsmail Emir Yüksel[§] A. Giray Yağlıkçı^{§,†}
Yu Liang[‡] F. Nisa Bostancı[§] Mohammad Sadrosadati[§] Onur Mutlu[§]
[§]ETH Zürich [†]CISPA [‡]Inria Paris

DRAM suffers from read disturbance phenomena (e.g., RowHammer and RowPress), where repeatedly accessing or continuously keeping open a DRAM row (aggressor row) induces bitflips in other physically nearby unaccessed rows (victim rows). The disturbance mechanism is practically exploitable from the software stack and worsens across generations with continued density scaling. DRAM read disturbance is highly sensitive to memory access patterns, yet prior work explores read disturbance under only a limited set of access patterns.

We present ScaleDisturb, a new DRAM access pattern that can amplify DRAM read disturbance by asymmetrically extending the open time of two aggressor rows. Our rigorous experimental characterization of 196 DDR4 and 3 HBM2 DRAM chips shows that ScaleDisturb (1) leads to bitflips at significantly fewer row activations, compared to state-of-the-art memory access patterns, (2) makes read disturbance attacks easier across all tested DRAM chips, (3) increases DRAM vulnerability to read disturbance as DRAM manufacturing technology scales down to smaller node sizes. We showcase a proof-of-

access pattern (i.e., alternately activating two adjacent aggressor rows within an access loop) is much more effective (i.e., requires far fewer aggressor row activations to induce bitflips) than a single-sided RowHammer access pattern, which activates only one aggressor row in each access loop. For RowPress, a longer t_{AggON} makes the RowPress access pattern more effective at inducing bitflips than RowHammer access patterns [9, 12, 13, 15–19, 21]. Prior characterization works conservatively quantify a DRAM row’s read disturbance vulnerability under the most effective access pattern using a row-level metric: the minimum number of aggressor row activations required to induce a bitflip in a victim row (AC_{min}). The reported minimum AC_{min} across all tested rows in a DRAM module is then used as the threshold for triggering protective operations in read disturbance mitigation mechanisms to prevent any erroneous bits. Identifying more effective DRAM row access patterns is critical to 1) accurately characterizing DRAM vulnerability and 2) designing robust mitigation mechanisms. However, prior works cumulatively examine subset of all possible memory

<https://arxiv.org/pdf/2606.07761>

Talk Outline

ScaleDisturb Access Pattern

Testing Infrastructure

Real DRAM Chip Results

Real System Demonstration of ScaleDisturb

Implications and Mitigations

Conclusion

Key Takeaways from Major Characterization Experiments

Takeaway 1

ScaleDisturb significantly **amplifies DRAM's vulnerability to read disturbance**

Takeaway 2

ScaleDisturb **widely affects DRAM chips**, and **worsens** as DRAM technology **scales down** to smaller node sizes

Key Takeaways from Major Characterization Experiments

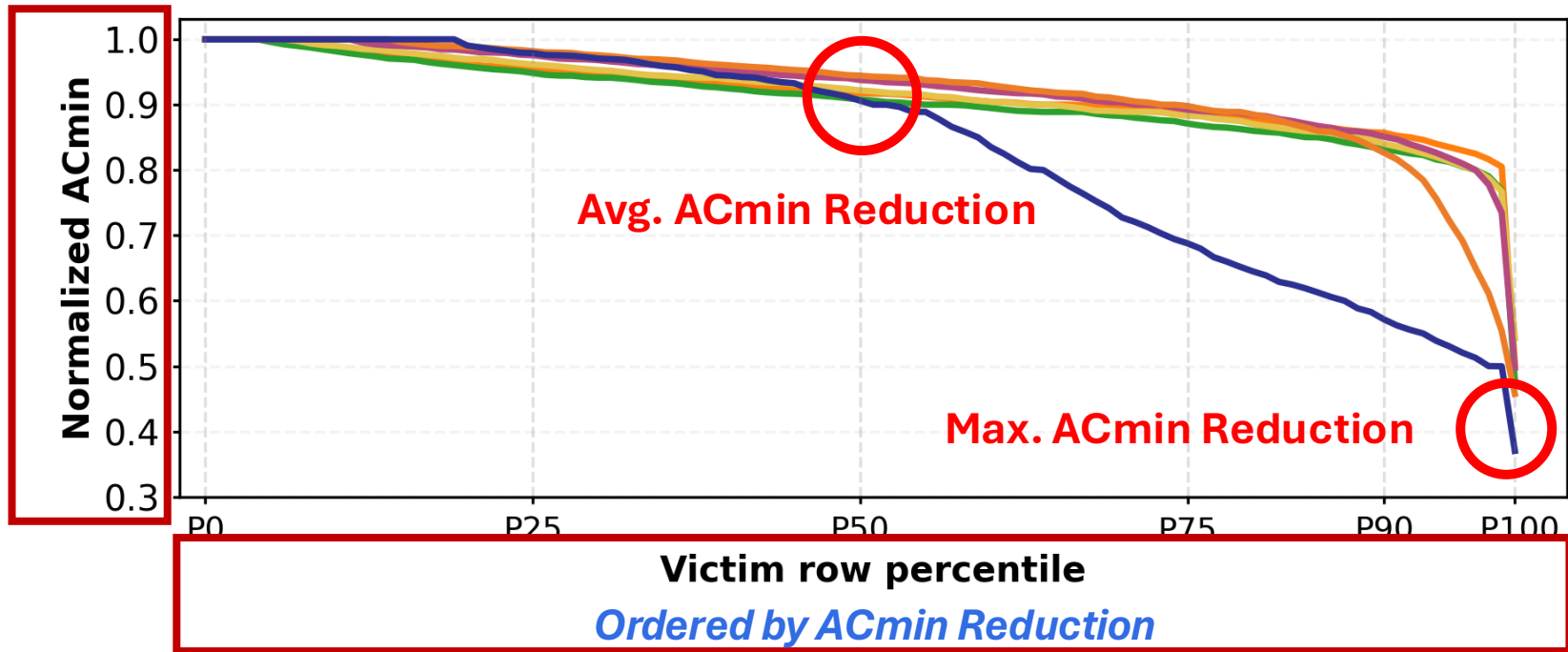
Takeaway 1

ScaleDisturb significantly **amplifies DRAM's vulnerability to read disturbance**

Takeaway 2

ScaleDisturb **widely affects DRAM chips**, and **worsens** as DRAM technology **scales down** to smaller node sizes

ScaleDisturb vs. Double-sided RowPress



ScaleDisturb reduces ACmin by **9.6% on average** (up to **63%**) across **all tested DRAM rows** compared to double-sided RowPress

Key Takeaways from Major Characterization Experiments

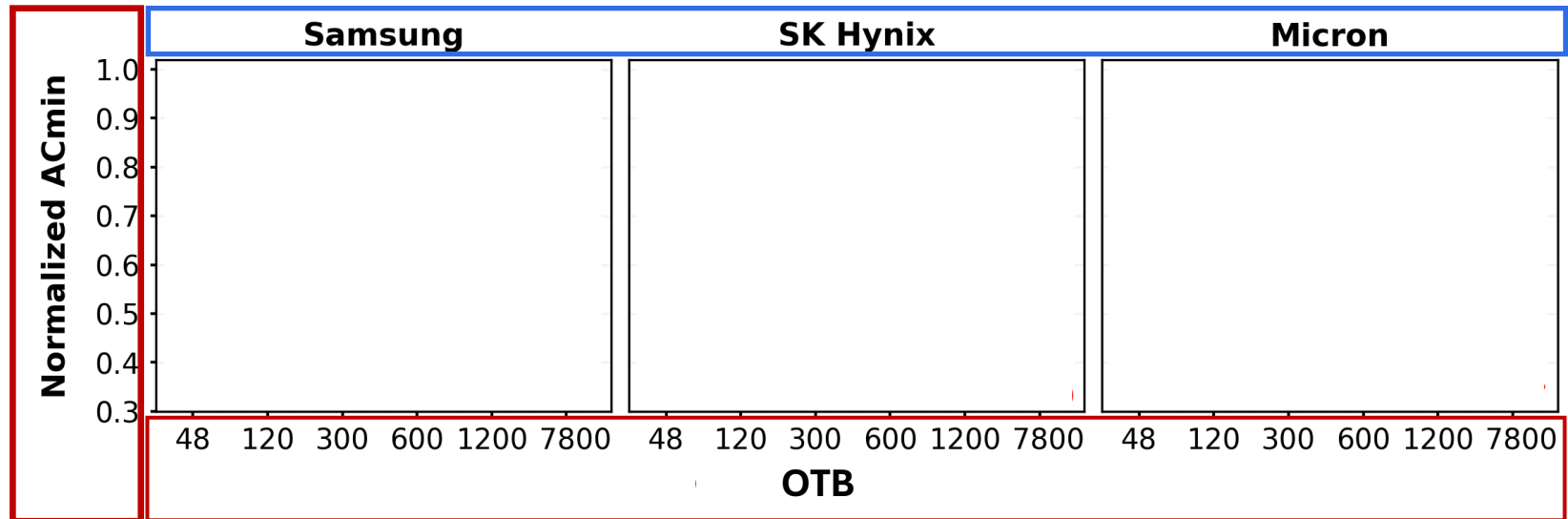
Takeaway 1

ScaleDisturb significantly **amplifies DRAM's vulnerability to read disturbance**

Takeaway 2

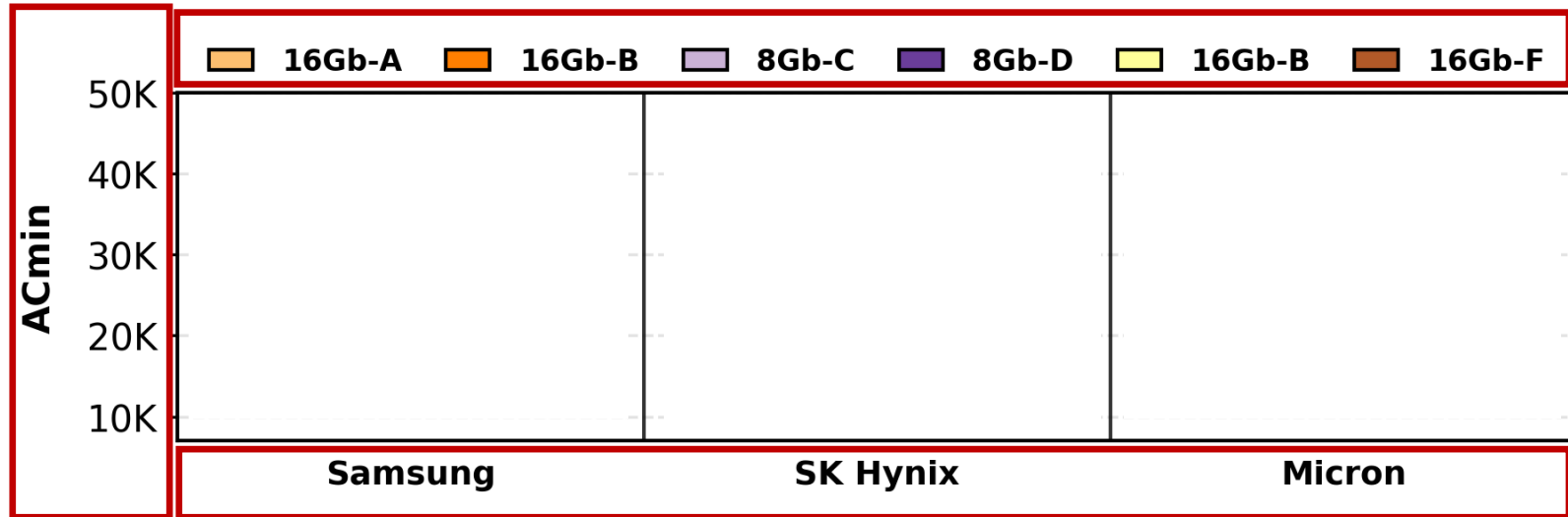
ScaleDisturb **widely affects DRAM chips**, and **worsens** as DRAM technology **scales down** to smaller node sizes

Prevalence of ScaleDisturb



ScaleDisturb **widely affects** DRAM chips **vulnerability** to read disturbance from **all three major DRAM manufacturers**

Newer Chips Are More Vulnerable



Newer chips tend to be **more vulnerable** to read disturbance under ScaleDisturb access pattern

Talk Outline

ScaleDisturb Access Pattern

Testing Infrastructure

Real DRAM Chip Results

Real System Demonstration of ScaleDisturb

Implications and Mitigations

Conclusion

Methodology of Real System Demonstration



**Intel Core i5-10400
(Comet Lake)**



**Samsung DDR4 Module
M378A2K43CB1-CTD
w/ TRR RowHammer Mitigation**

Our ScaleDisturb program

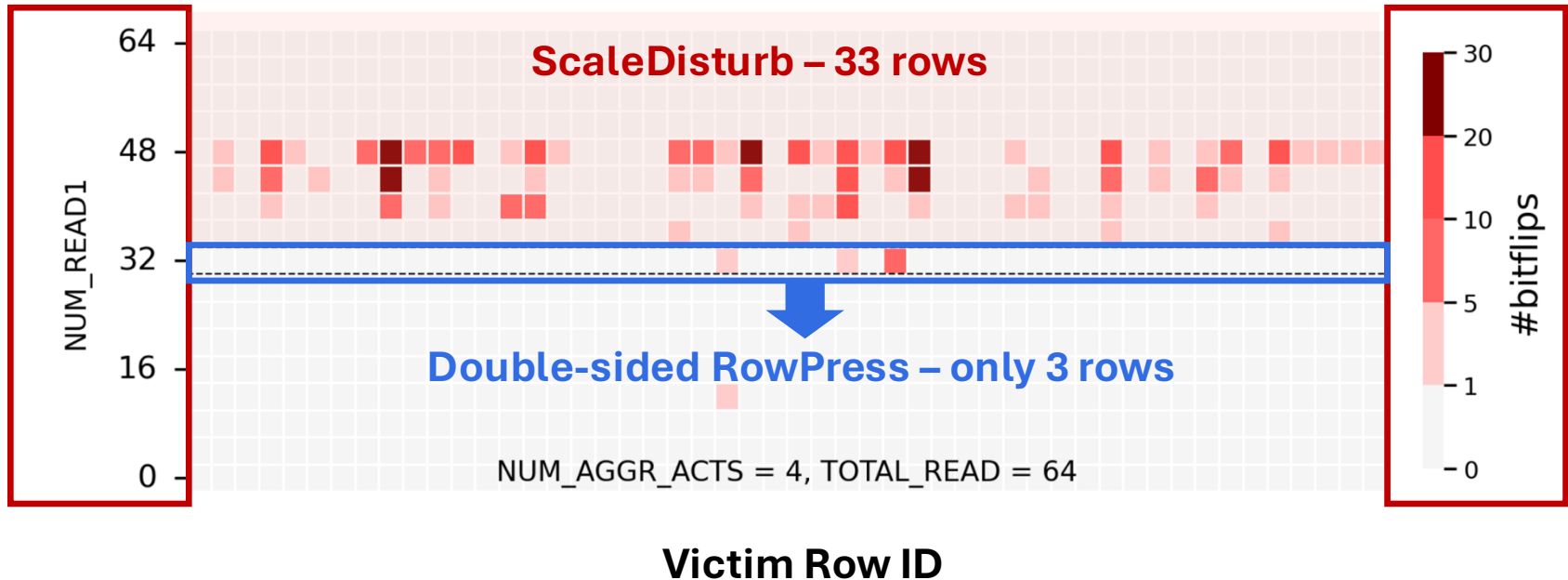
- keeps a DRAM row open for a longer time by accessing more cache blocks in the row [Luo+, ISCA 2023]
- keeps the two DRAM rows open for different times by accessing different numbers of cache blocks in the two aggressor rows

Real System Demonstration Program

```
// Sync with Refresh and Loop Below
for(NUM_READ1 = 0; NUM_READ1 < TOTAL_READ; NUM_READ1++)
    // Sweep two rows' activation time
    NUM_READ2 = TOTAL_READ - NUM_READ1;
    for (k = 0; k < NUM_AGGR_ACTS; k++) # Cache Blocks Accessed Per Row Activation
        for (j = 0; j < NUM_READ1; j++) NUM_READ1 != NUM_READ2 (ScaleDisturb)
            *AGGRESSOR1[j];
            NUM_READ1 = NUM_READ2 > 1 (RowPress)
            c1flushopt(AGGRESSOR1[j]);
        for (j = 0; j < NUM_READ2; j++) NUM_READ1 = NUM_READ2 = 1 (RowHammer)
            *AGGRESSOR2[j]; NUM_AGGR_ACTS: Dummy row activation frequency
            c1flushopt(AGGRESSOR2[j]);
        mfence();
    activate_dummy_rows();
```

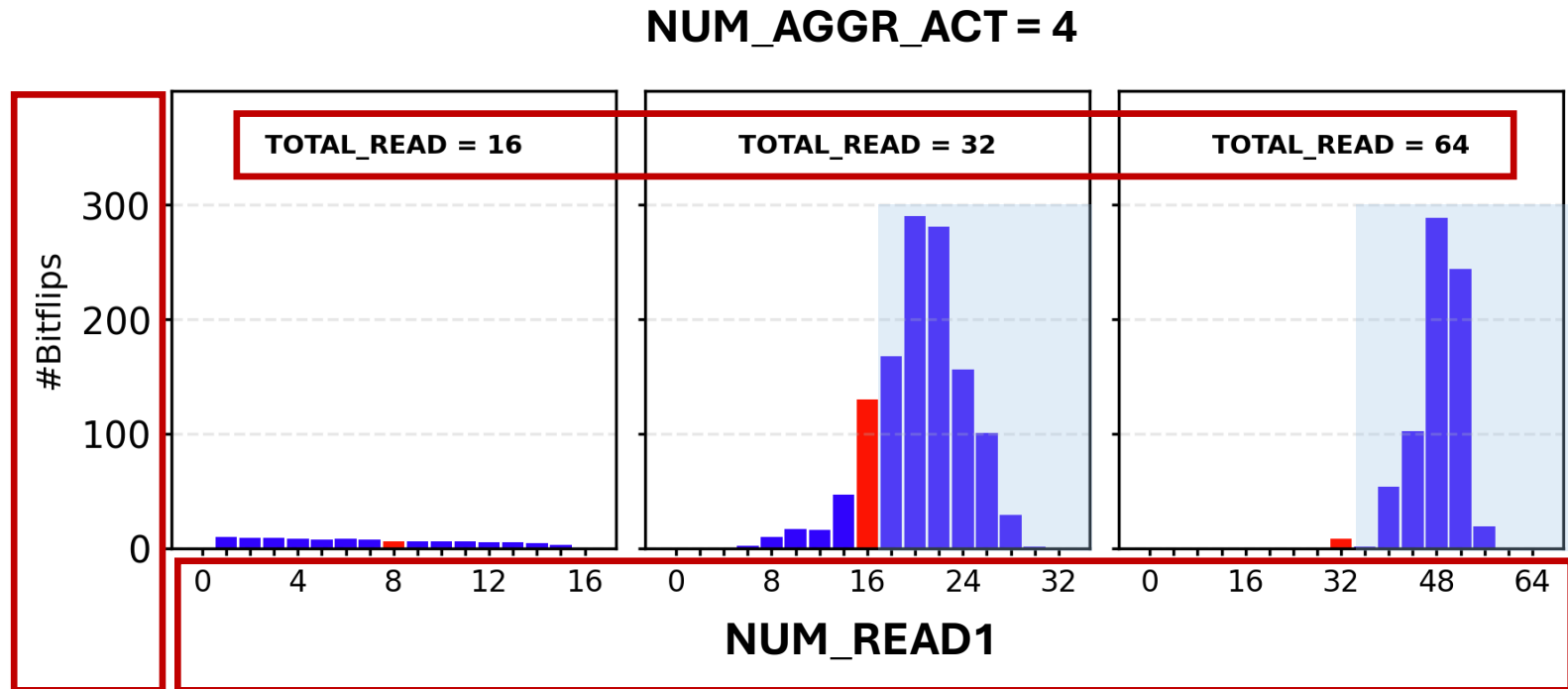
Real System Demonstration Results (I)

Top 50 rows with highest #bitflips



A user-level program leveraging ScaleDisturb induces bitflips in cells where double-sided RowPress cannot

Real System Demonstration Results (II)



ScaleDisturb induces **more bitflips** than **double-sided RowPress**, under low dummy row activation frequency

Talk Outline

ScaleDisturb Access Pattern

Testing Infrastructure

Real DRAM Chip Results

Real System Demonstration of ScaleDisturb

Implications and Mitigations

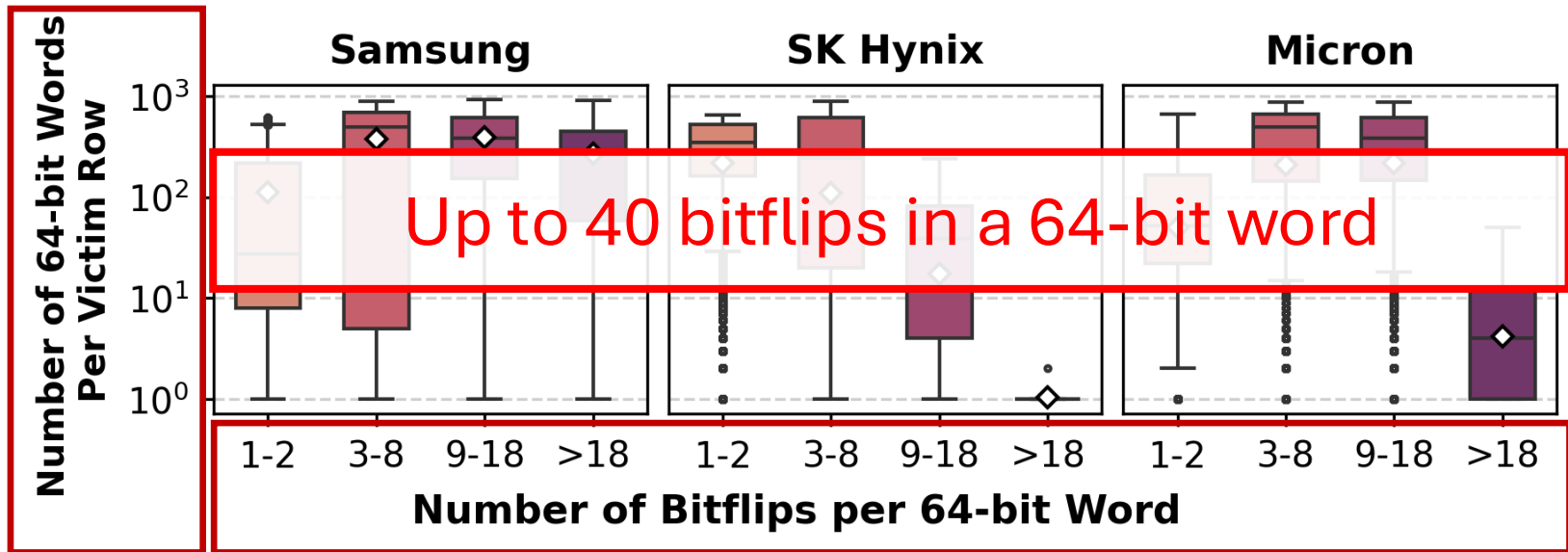
Conclusion

Implications and Mitigations Summary

- Security guarantees provided by mitigation techniques rely on accurately profiling the read disturbance threshold (AC_{min}) using existing DRAM access patterns
- ScaleDisturb leads to much lower AC_{min} than existing access patterns, defenses should take ScaleDisturb into account
- We analyze ScaleDisturb's implication on three classes of potential solutions
- We propose a new solution with low additional overhead

ECC Is Ineffective

Error Correcting Code (ECC)



ScaleDisturb induces more bitflips (up to 40 bitflips) per 64-bit word than widely used ECC techniques can correct

Methodology for Safety Margin Evaluation

Applying Safety Margin to existing SOTA mitigation mechanisms

- **Performance evaluation:**

- cycle-level simulations using **Ramulator 2.0** [Luo+, CAL 2023]

- **Threshold baselines: 1024 and 128**

- **Safety margins: 10%, 25%, and 60%**

- **Five state-of-the-art mitigation mechanisms**

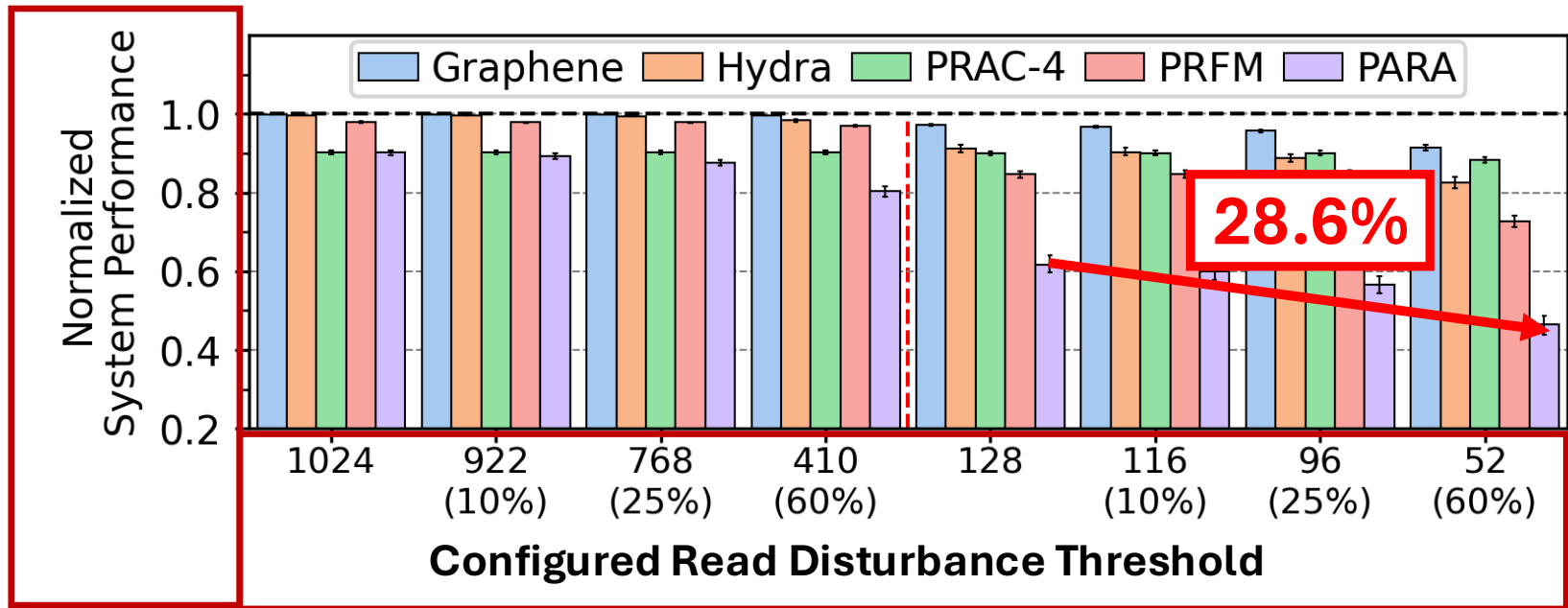
- Graphene, Hydra, PRAC, PRFM, PARA

- **System Configuration:**

Processor	4 cores, 4.2GHz clock frequency, 4-wide issue, 128-entry instruction window
DRAM	DDR5, 1 channel, 2 rank/channel, 8 bank groups, 4 banks/bank group, 64K rows/bank
Memory Ctrl.	64-entry read and write requests queues, Scheduling policy: FR-FCFS with a column cap of 4 Last-Level Cache 8 MiB (4-core)

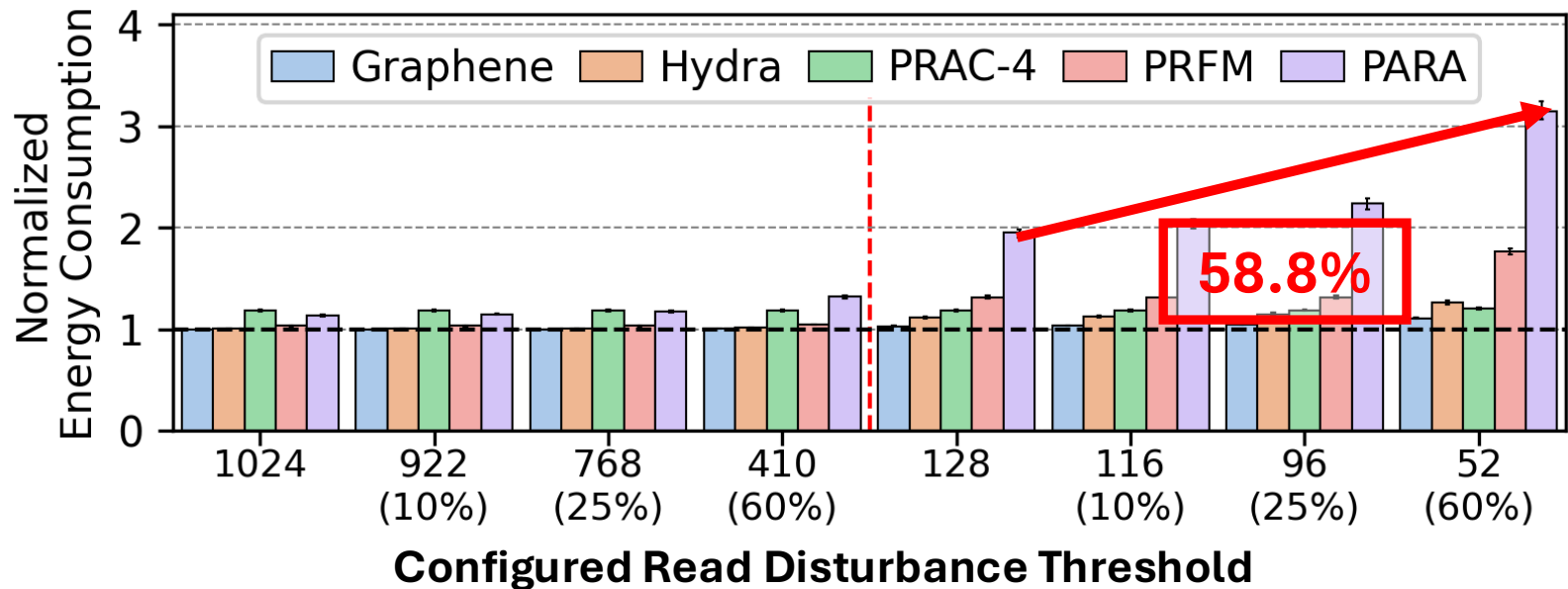
- **Workloads:** 60 4-core workload mixes from 5 benchmark suites

Performance Overhead



Adding a **large safety margin** to mitigation mechanisms incurs **up to 28.6% (8.8% on average)** performance overhead

Energy Overhead

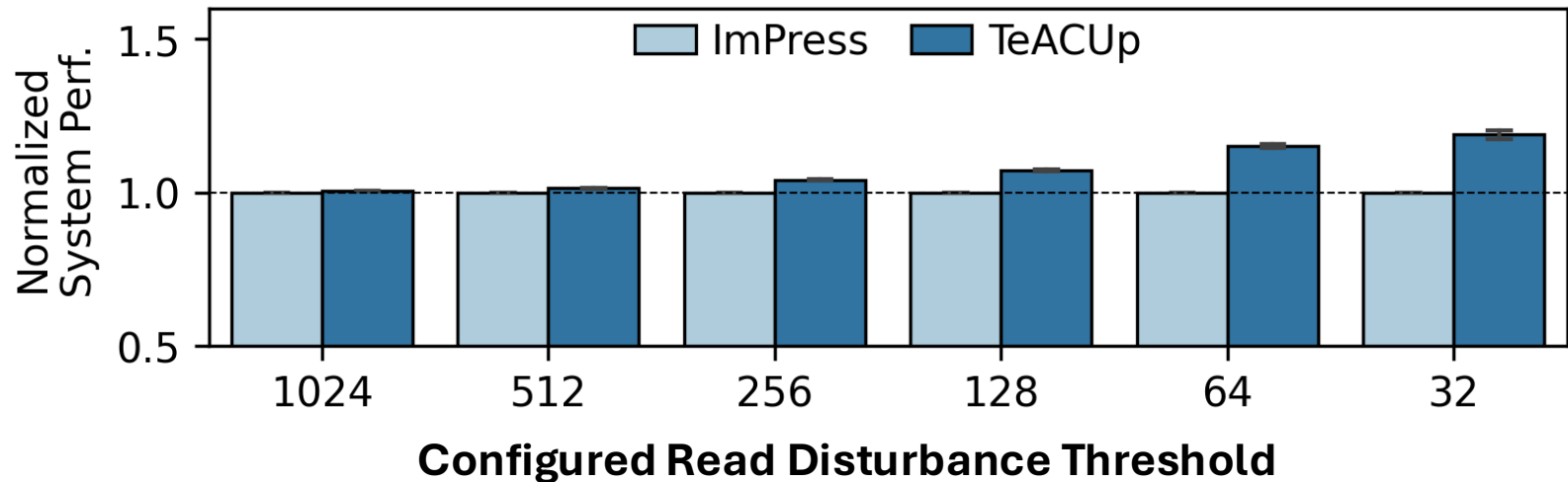


Using safety margin to existing mitigation mechanisms incurs
up to 58.8% (12.3% on average) energy overhead

Our Solution: TeACUp

- We develop a new mitigation, *Temporal Asymmetry-aware Counter Update (TeACUp)*, by adapting a state-of-the-art RowPress mitigation, *ImPress*
- The **key idea** of TeACUp is to **scale** the row counter increment **adaptively** based on the **asymmetric activation times**

TeACUp Evaluation Results



TeACUp improves system performance by **3.2% on average (up to 9.2%)** over state-of-the-art RowPress mitigation

More in the Paper

- **Many more characterization results**
 - Bitflip index analysis
 - Access pattern comparison
 - Data pattern analysis
 - Temperature analysis
 - HBM2 analysis
- **Hypothetical explanation for ScaleDisturb**
- **Mitigation implications**
 - ECC capability
 - Adapting existing RowHammer/RowPress mitigation mechanisms
 - TeACUp: security analysis, evaluation methodology, overhead analysis

ScaleDisturb: Exploiting Temporal Asymmetry to Amplify Read Disturbance in Modern DRAM Chips

Jikun Wang[§] Haocong Luo[§] Ataberk Olgun[§] İsmail Emir Yüksel[§] A. Giray Yağlıkçı^{§,†}
Yu Liang[‡] F. Nisa Bostancı[§] Mohammad Sadrosadati[§] Onur Mutlu[§]
[§]ETH Zürich [†]CISPA [‡]Inria Paris

DRAM suffers from read disturbance phenomena (e.g., RowHammer and RowPress), where repeatedly accessing or continuously keeping open a DRAM row (aggressor row) induces bitflips in other physically nearby unaccessed rows (victim rows). The disturbance mechanism is practically exploitable from the software stack and worsens across generations with continued density scaling. DRAM read disturbance is highly sensitive to memory access patterns, yet prior work explores read disturbance under only a limited set of access patterns.

We present ScaleDisturb, a new DRAM access pattern that can amplify DRAM read disturbance by asymmetrically extending the open time of two aggressor rows. Our rigorous experimental characterization of 196 DDR4 and 3 HBM2 DRAM chips shows that ScaleDisturb (1) leads to bitflips at significantly fewer row activations, compared to state-of-the-art memory access patterns, (2) makes read disturbance attacks easier across all tested DRAM chips, (3) increases DRAM vulnerability to read disturbance as DRAM manufacturing technology scales down to smaller node sizes. We showcase a proof-of-

access pattern (i.e., alternately activating two adjacent aggressor rows within an access loop) is much more effective (i.e., requires far fewer aggressor row activations to induce bitflips) than a single-sided RowHammer access pattern, which activates only one aggressor row in each access loop. For RowPress, a longer t_{AggON} makes the RowPress access pattern more effective at inducing bitflips than RowHammer access patterns [9, 12, 13, 15–19, 21]. Prior characterization works conservatively quantify a DRAM row’s read disturbance vulnerability under the most effective access pattern using a row-level metric: the minimum number of aggressor row activations required to induce a bitflip in a victim row (AC_{min}). The reported minimum AC_{min} across all tested rows in a DRAM module is then used as the threshold for triggering protective operations in read disturbance mitigation mechanisms to prevent any erroneous bits. Identifying more effective DRAM row access patterns is critical to 1) accurately characterizing DRAM vulnerability and 2) designing robust mitigation mechanisms. However, prior works cumulatively examine subset of all possible memory

<https://arxiv.org/pdf/2606.07761>

Open-Sourced & Artifact-Evaluated

CMU-SAFARI / ScaleDisturb

Type to search

<> Code

Issues

Pull requests

Agents

Actions

Projects

Security and quality

Insights

Settings

SAFARI

ScaleDisturb

Public

Watch

0

main

1 Branch

0 Tags

Go to file

Add file

<> Code

jwang

Initial commit

1213ab0 · 3 weeks ago

1 Commit

<div>characterization</div>	Initial commit	3 weeks ago
<div>demonstration</div>	Initial commit	3 weeks ago
<div>simulation</div>	Initial commit	3 weeks ago
<div>.gitignore</div>	Initial commit	3 weeks ago
<div>README.md</div>	Initial commit	3 weeks ago
<div>requirements.txt</div>	Initial commit	3 weeks ago
<div>setup_all.sh</div>	Initial commit	3 weeks ago

README

ScaleDisturb: Exploiting Temporal Asymmetry to Amplify Read Disturbance in Modern DRAM Chips

Talk Outline

ScaleDisturb Access Pattern

Testing Infrastructure

Real DRAM Chip Results

Real System Demonstration of ScaleDisturb

Implications and Mitigations

Conclusion

Conclusion

We propose ScaleDisturb, a new DRAM access pattern that amplifies DRAM vulnerability to read disturbance by asymmetrically extending the open time of two aggressor rows

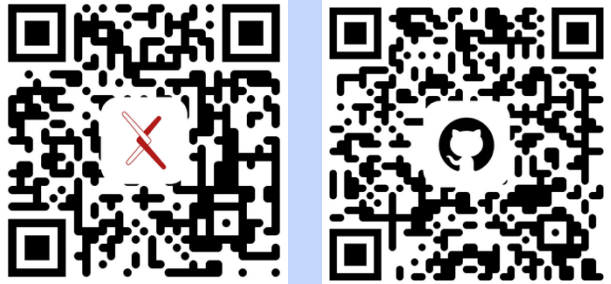
We characterize ScaleDisturb on 196 DDR4 and 3 HBM2 DRAM chips

- ScaleDisturb worsens DRAM's vulnerability to read disturbance
- ScaleDisturb becomes worse as DRAM technology node size reduces

We demonstrate ScaleDisturb bitflips on a real system with TRR mitigation

- ScaleDisturb induces bitflips in cells where double-sided RowPress cannot
- ScaleDisturb induce more bitflips than double-sided RowPress

We evaluate the implications of ScaleDisturb on existing DRAM-based systems and propose a new solution with low overhead



Code
Reproducible



Dataset
Reproducible

ScaleDisturb: Exploiting Temporal Asymmetry to Amplify Read Disturbance in Modern DRAM Chips

Jikun Wang

Haocong Luo Ataberk Olgun Ismail Emir Yüksel A. Giray Yağlıkçı
Yu Liang F. Nisa Bostanci Mohammad Sadrosadati Onur Mutlu

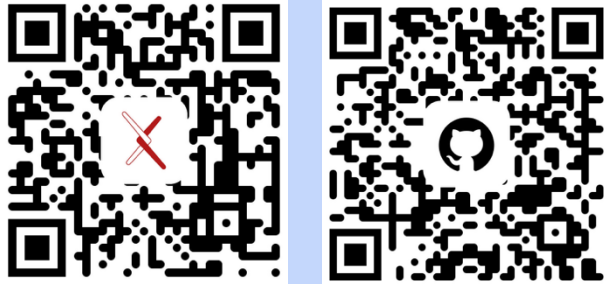
SAFARI

ETH zürich



CISPA
HELMHOLTZ CENTER FOR
INFORMATION SECURITY

Inria



Code
Reproducible



Dataset
Reproducible

ScaleDisturb: Exploiting Temporal Asymmetry to Amplify Read Disturbance in Modern DRAM Chips

Backup Slides

Jikun Wang

Haocong Luo Ataberk Olgun Ismail Emir Yüksel A. Giray Yağlıkçı
Yu Liang F. Nisa Bostanci Mohammad Sadrosadati Onur Mutlu

SAFARI

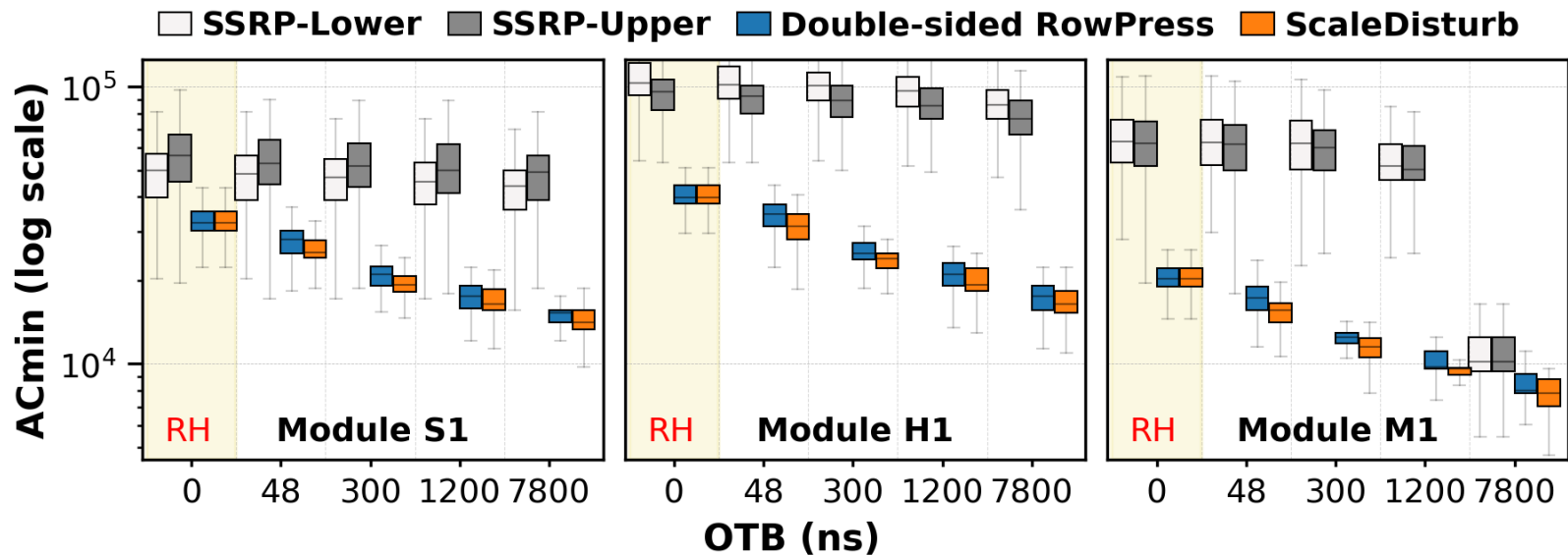
ETH zürich



CISPA
HELMHOLTZ CENTER FOR
INFORMATION SECURITY

Inria

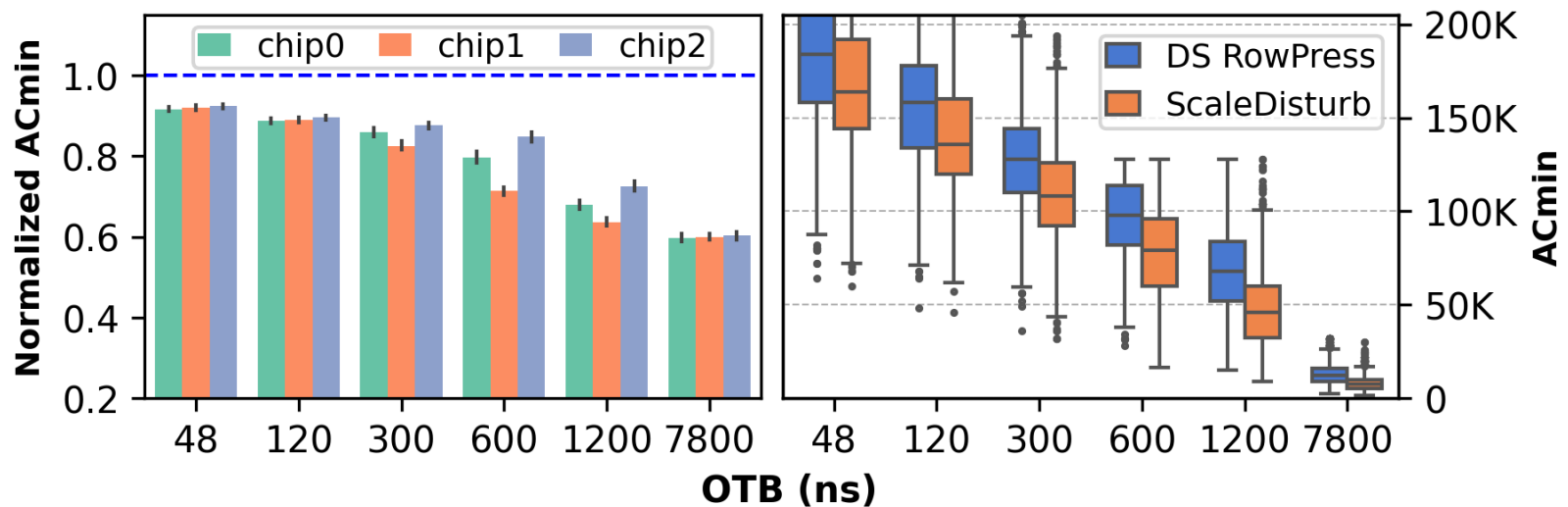
DRAM Access Pattern Comparison



Single-sided access patterns causes **asymmetric vulnerability** based on victim row's position

ScaleDisturb consistently leads to the **lowest ACmin** across all access patterns under tested OTB values.

HBM2 DRAM Chips – ACmin Reduction

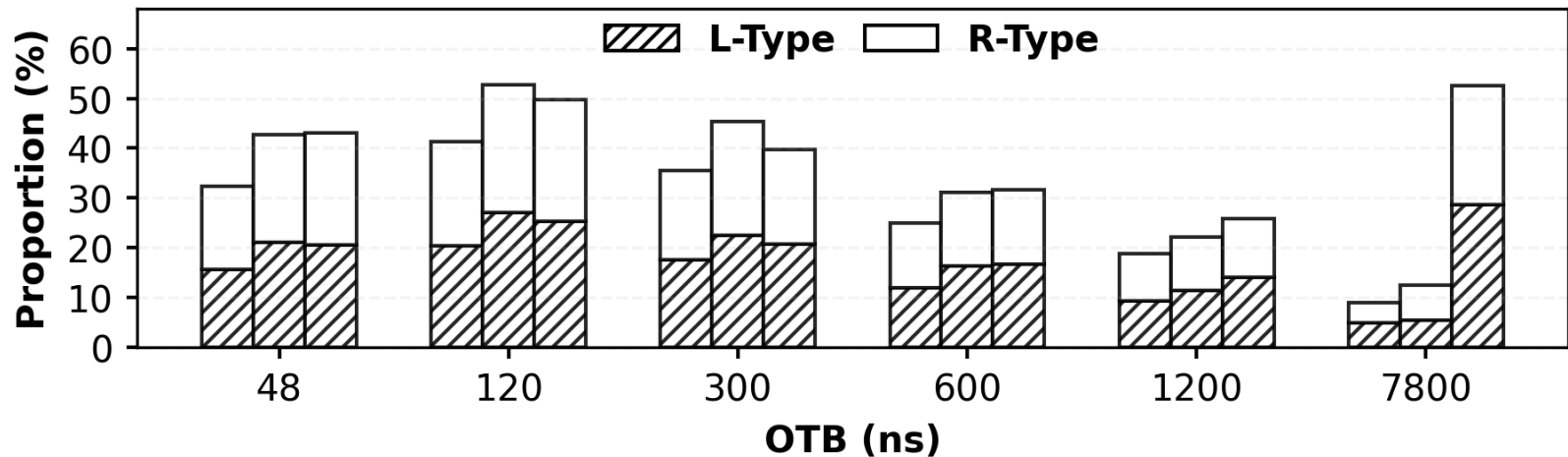


HBM2 chips are **more vulnerable** to read disturbance under **ScaleDisturb** access pattern than **double-sided RowPress**

Hypothetical explanation

- Enhanced electric fields drive electron migration more effectively than trap-assisted leakage or crosstalk-induced mechanisms
- In double-sided patterns, the electric fields from the two aggressor wordlines can interact and partially cancel each other
- ScaleDisturb introduces temporal asymmetry that breaks this interference, thereby increasing net electron migration

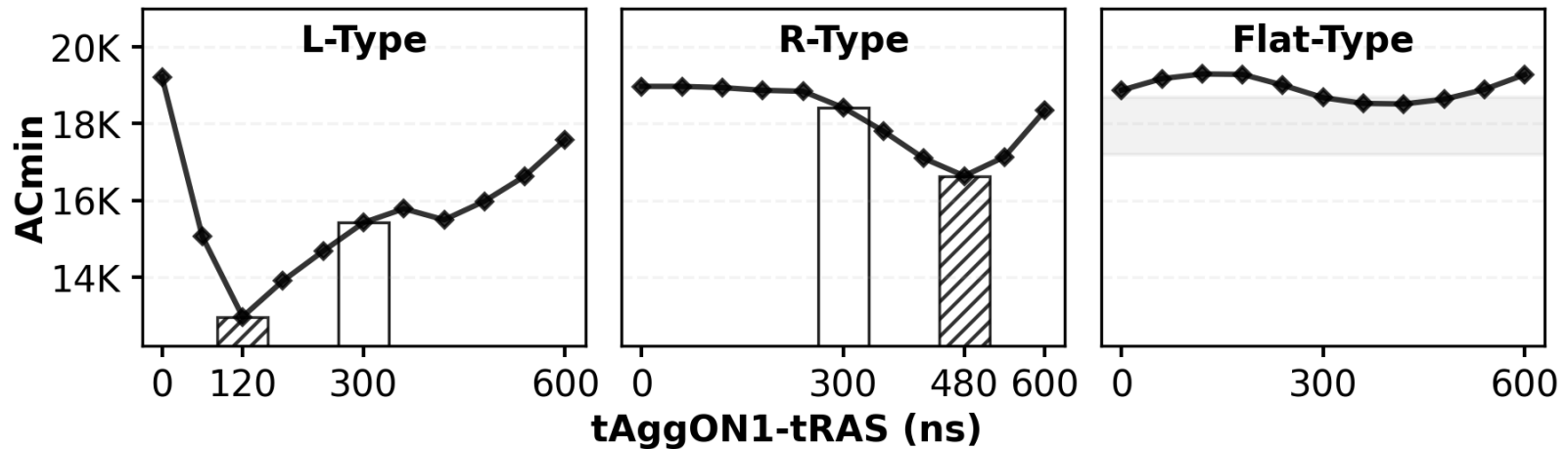
Proportions of three ACmin Changing Pattern



L-type and R-type patterns constitute **a large fraction of rows**, and their proportions vary with OTB

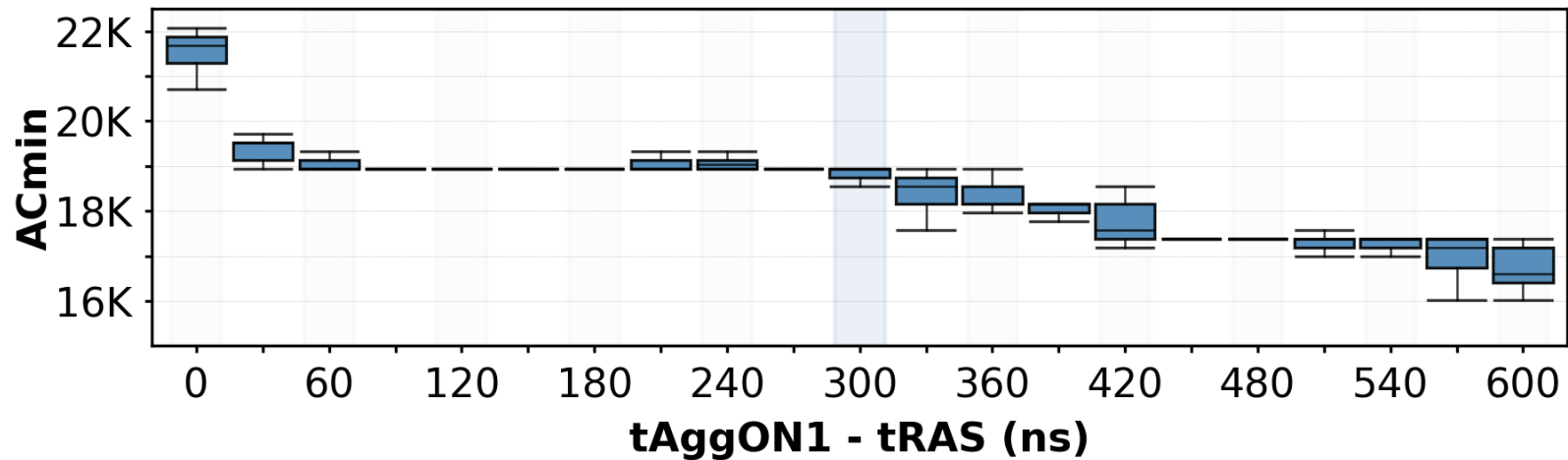
L-type and R-type patterns occur at **nearly equal rates** across all evaluated OTB values

Three Observed ACmin Changing Pattern



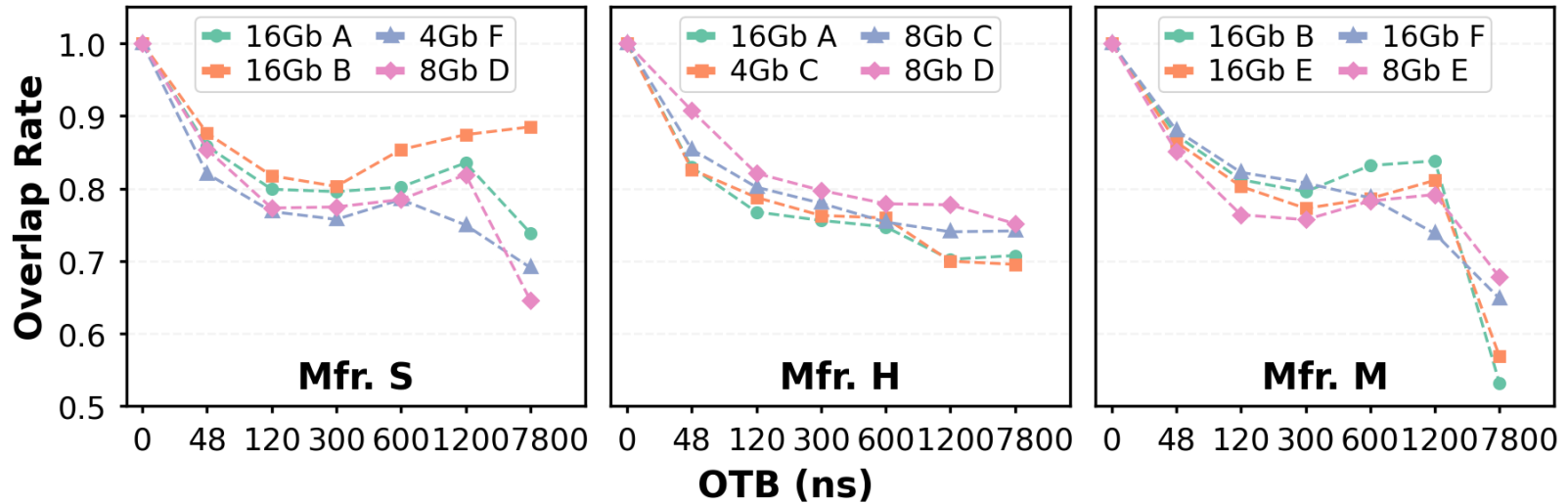
DRAM rows under ScaleDisturb exhibit three major ACmin reduction patterns: L-, R- and Flat-type

Distinguish ScaleDisturb and Variable Read Disturbance (VRD)



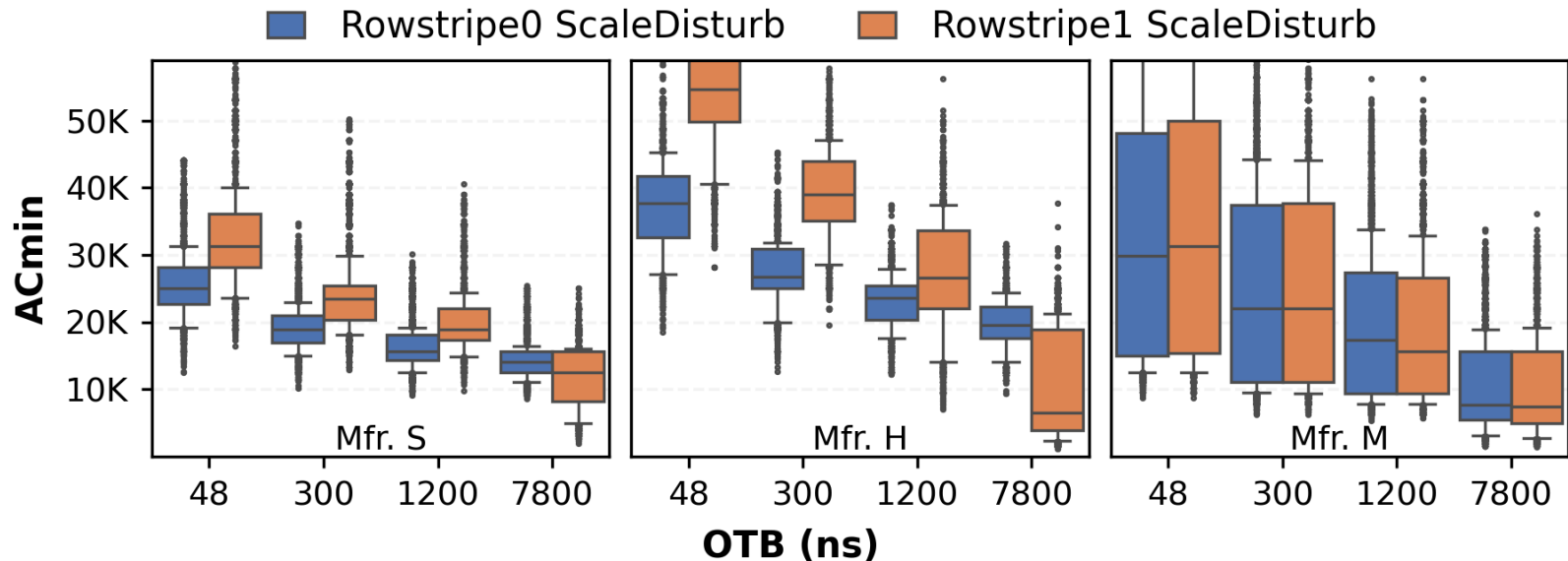
ScaleDisturb is **fundamentally different** from **Variable Read Disturbance** in both **stability** and **ACmin reduction**

Bitflip Index Analysis



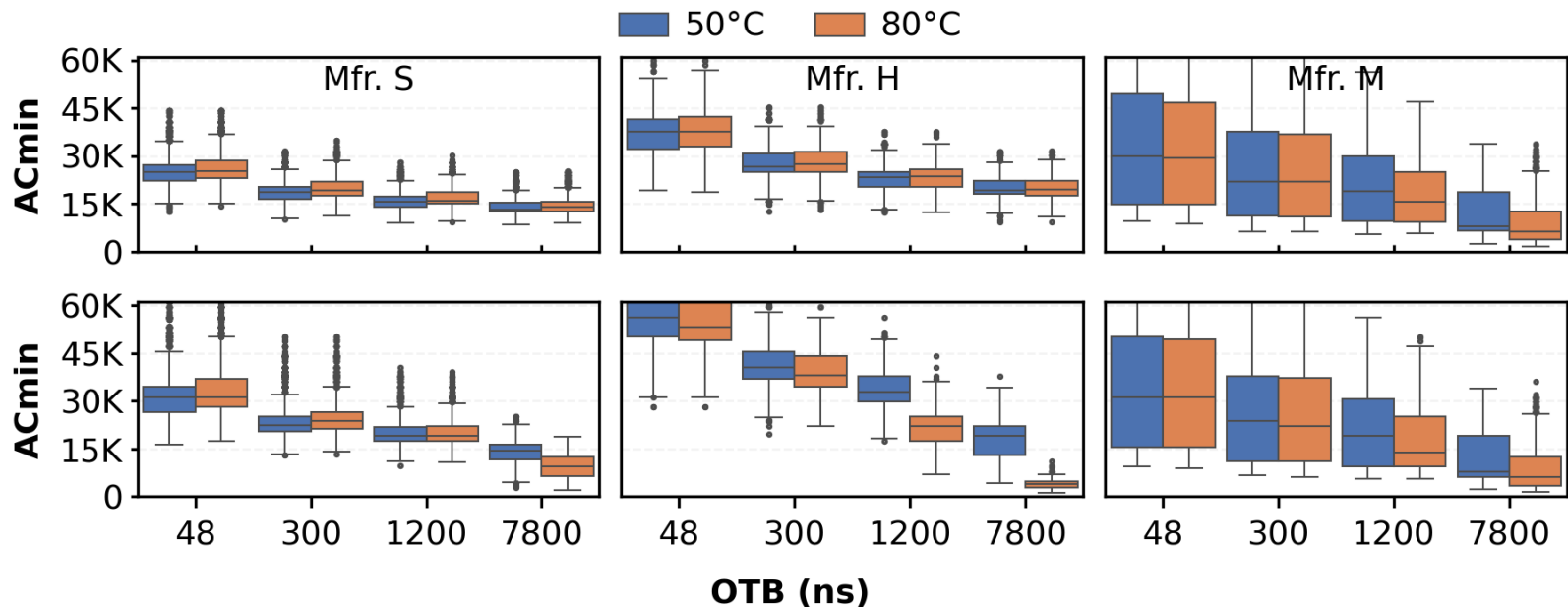
ScaleDisturb flips a distinct set of cells
than double-sided RowPress

Data Pattern' Effects on ScaleDisturb



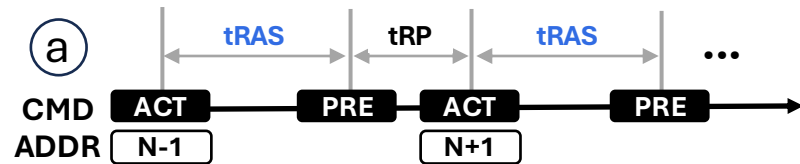
Rowstripe0 yields lower ACmin at small OTBs
Rowstripe1 yields lower ACmin at large OTBs

Temperature' Effects on ScaleDisturb

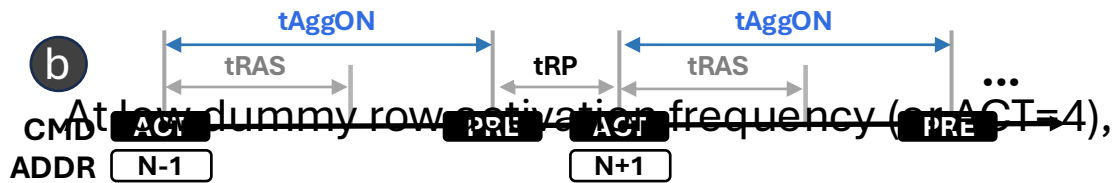


Under ScaleDisturb access pattern,
ACmin is less sensitive to a higher temperature for Rowstripe0
than for Rowstripe1 as OTB increases

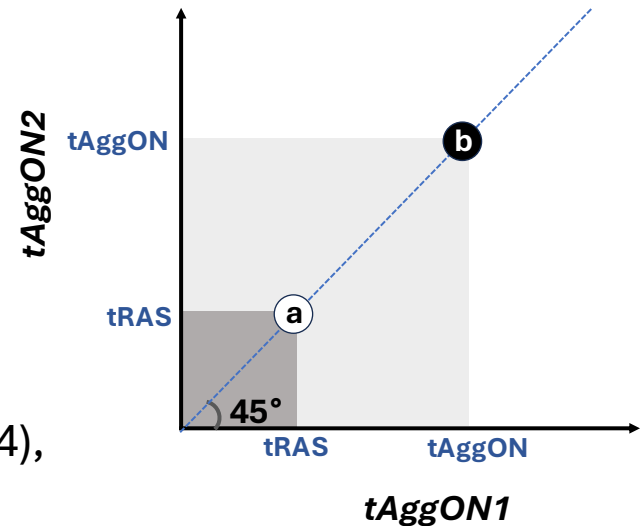
ScaleDisturb Access Pattern (I)



Double-sided RowHammer



Double-sided RowPress

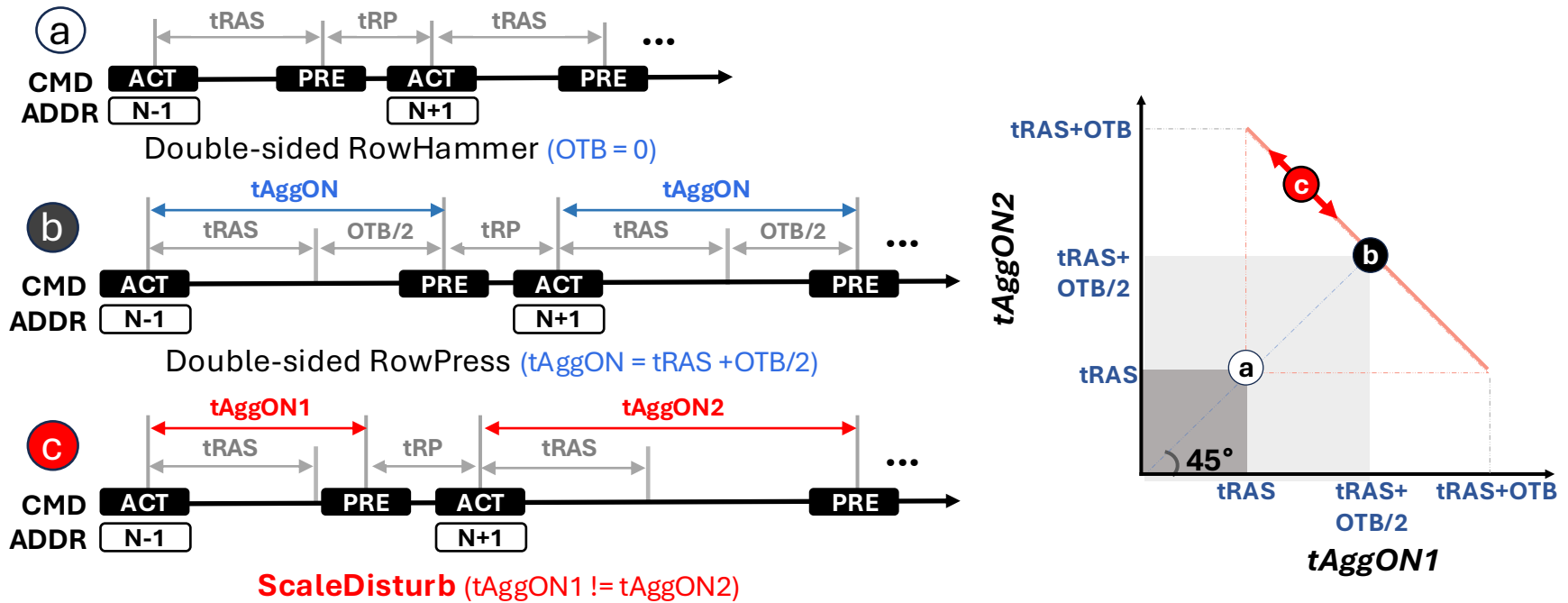


Existing **double-sided DRAM** access patterns
always keep the two aggressor rows open for **equal durations**

ScaleDisturb Access Pattern (II)

What if the two aggressor rows have different open durations?

New Parameter: Additional Row Open Time Budget (OTB) = $t_{\text{AggON1}} + t_{\text{AggON2}} - 2 \cdot t_{\text{RAS}}$



ScaleDisturb is a **new DRAM access pattern**

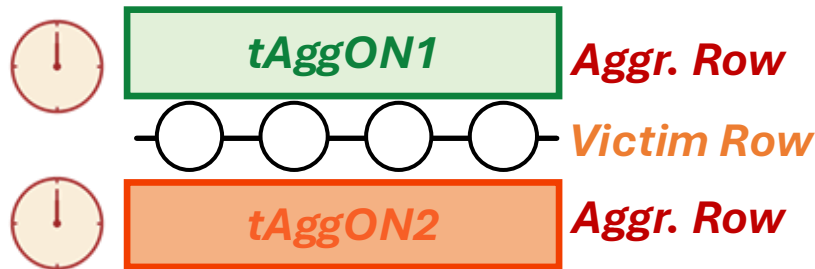
that **asymmetrically extends** the open time of two aggressor rows

ScaleDisturb Access Pattern (III)

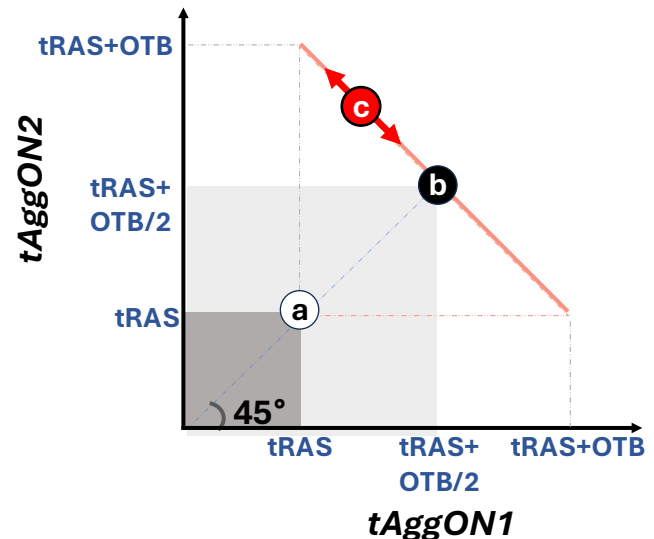
What if the two aggressor rows have different open durations?

New Parameter: Extra Row Open Time Budget (OTB) = $t_{\text{AggON1}} + t_{\text{AggON2}} - 2 \cdot t_{\text{RAS}}$

- a** Double-sided RowHammer (OTB = 0)
- b** Double-sided RowPress ($t_{\text{AggON}} = t_{\text{RAS}} + \text{OTB}/2$)

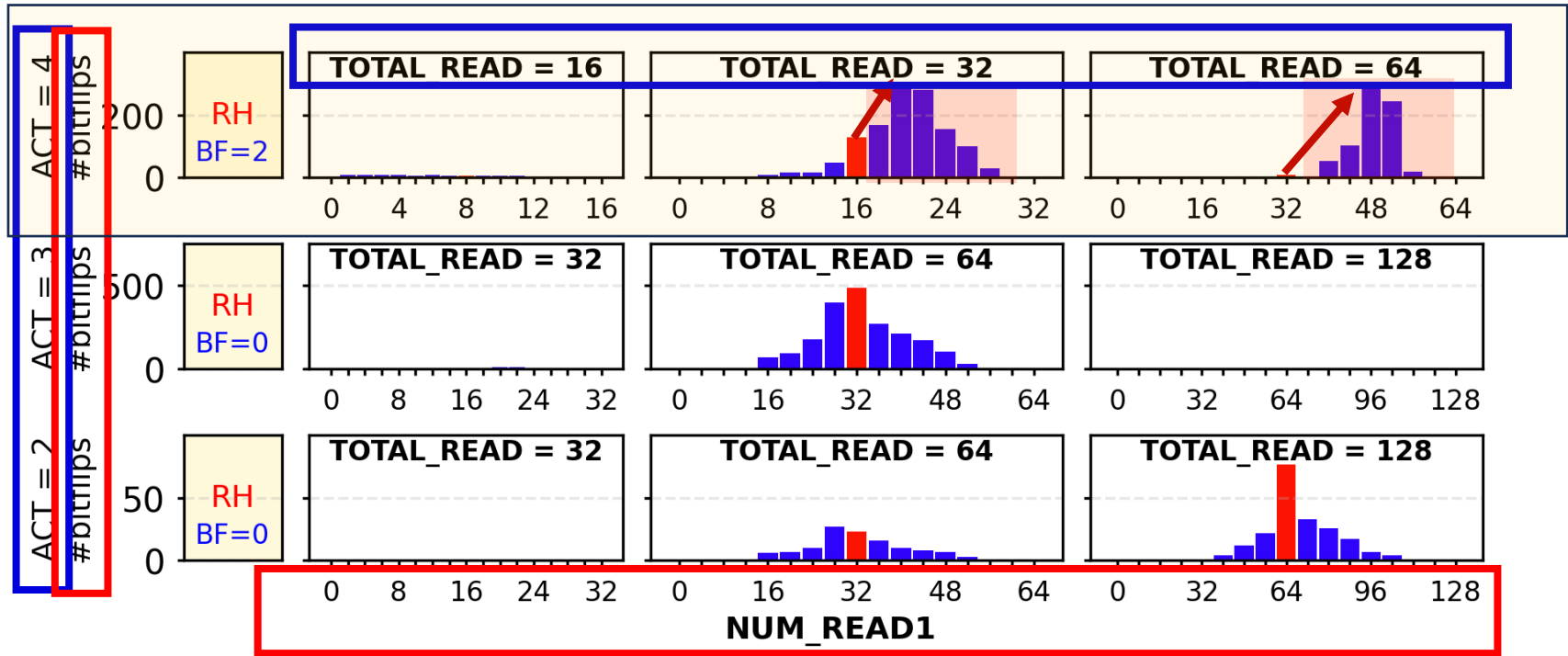


c **ScaleDisturb**
($t_{\text{AggON1}} \neq t_{\text{AggON2}}$)



ScaleDisturb is a **new DRAM access pattern** that **asymmetrically extends** the open time of two aggressor rows

Real System Demonstration Results (II)

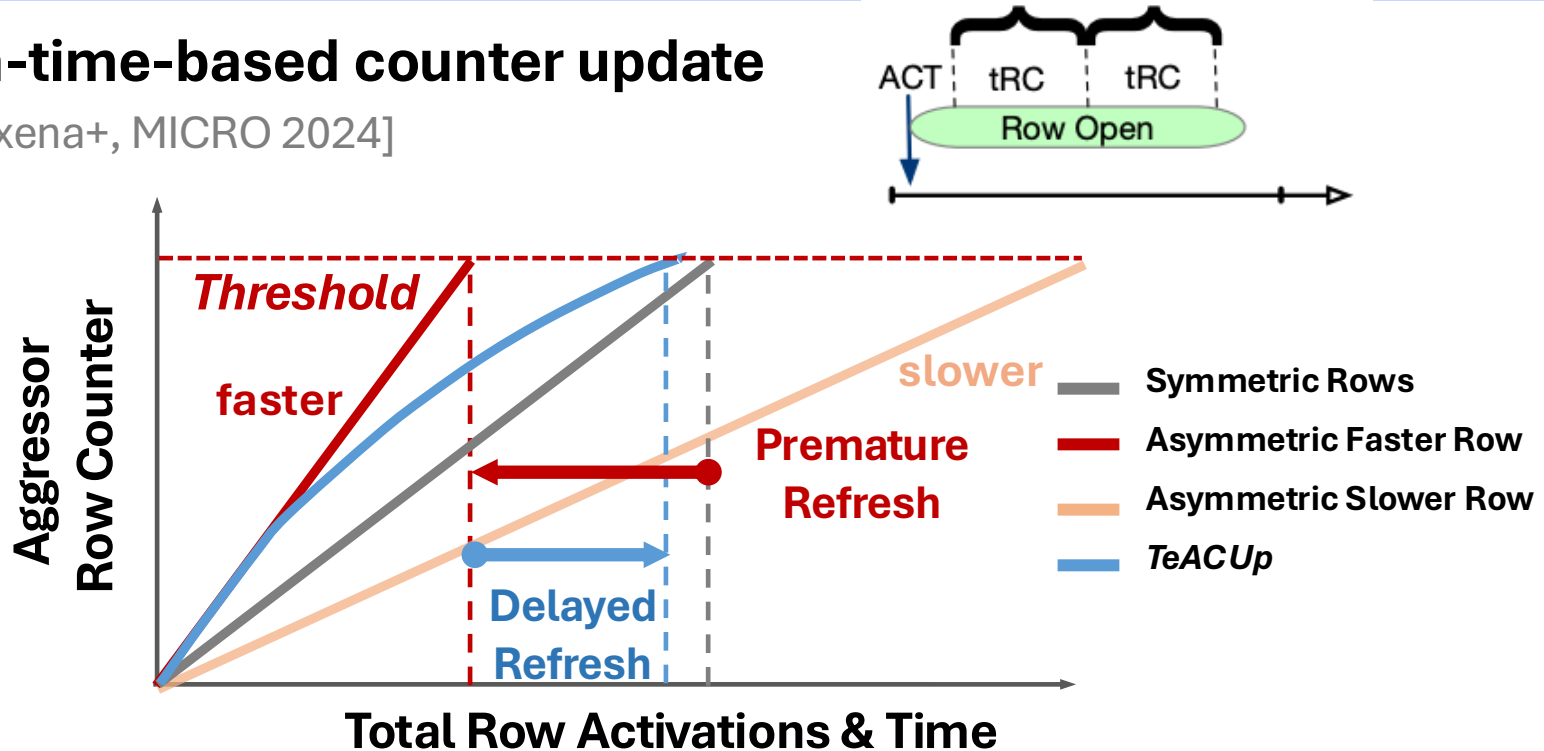


ScaleDisturb induces bitflips in **more cells than double-sided RowPress**, under low dummy row activation frequency

Implications and Mitigations (III)

Activation-time-based counter update

ImPress, [Saxena+, MICRO 2024]



Naively mapping activation time to counter increment causes unnecessary protective refreshes

Temporal Asymmetry-aware Counter Update (TeACUp) safely slows the increment of faster row to delay the refresh