RowPress
Amplifying Read Disturbance in Modern DRAM Chips

Haocong Luo

Ataberk Olgun   A. Giray Yağlıkçı   Yahya Can Tuğrul
Steve Rhyner   Meryem Banu Cavlak   Joël Lindegger
Mohammad Sadrosadati   Onur Mutlu

SAFARI   ETH Zürich

High-Level Summary

- We demonstrate and analyze **RowPress, a new read disturbance phenomenon** that causes bitflips in real DRAM chips

- We show that RowPress is **different from the RowHammer vulnerability**

- We demonstrate RowPress **using a user-level program** on a real Intel system with real DRAM chips

- We provide **effective solutions** to RowPress

SAFARI

Outline
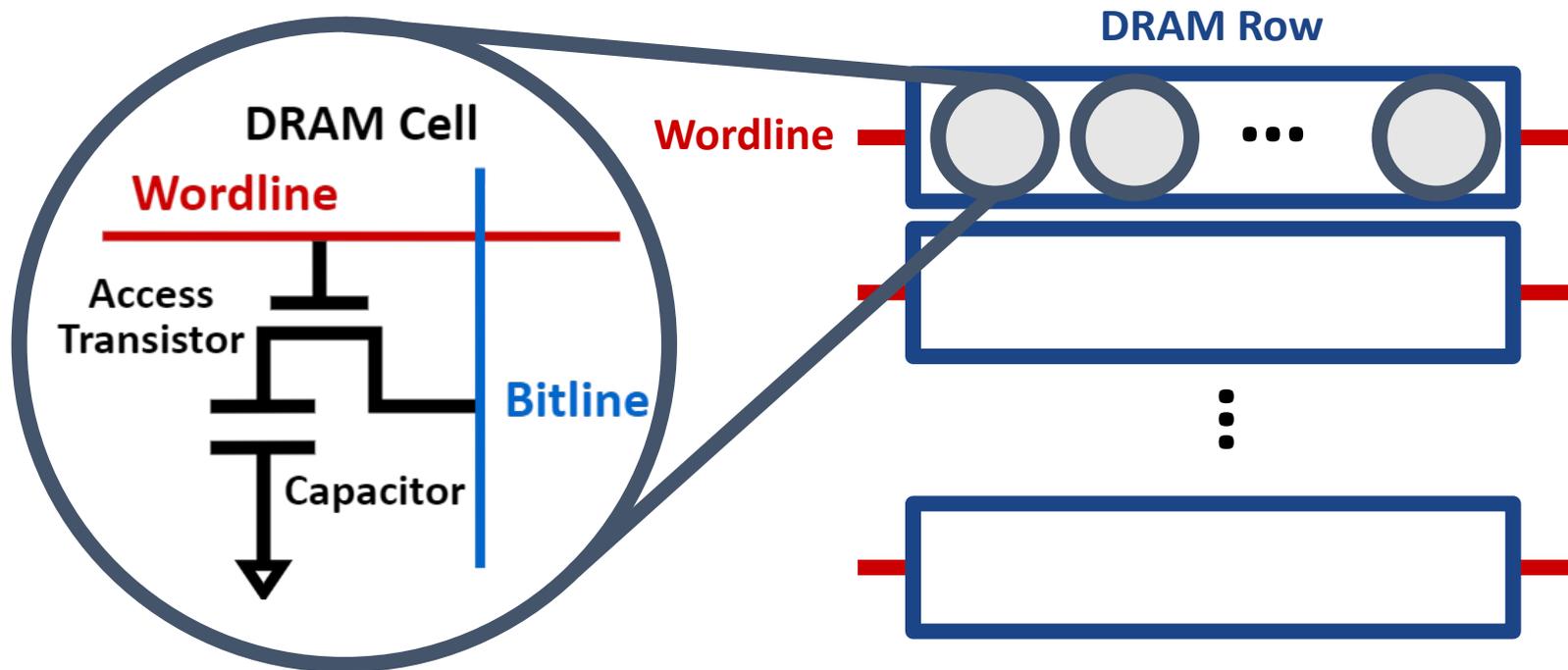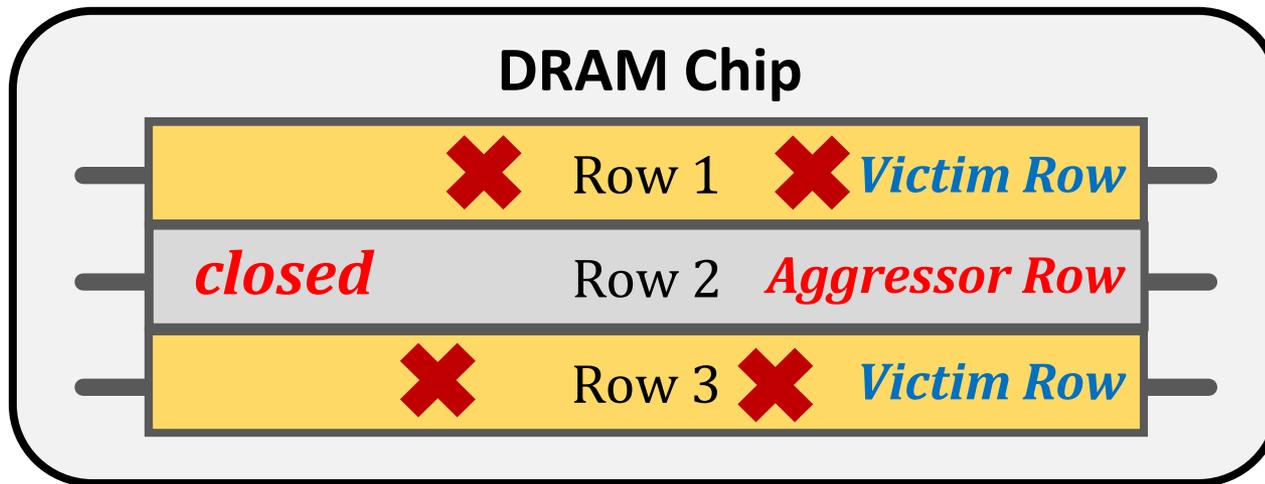
DRAM Organization

DRAM is the prevalent technology for main memory

- A **DRAM cell** stores 1 bit of information in a **leaky** capacitor
- DRAM cells are organized into **DRAM rows**

Read Disturbance in DRAM

- Read disturbance in DRAM breaks memory isolation
- **Prominent example: RowHammer**

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

Are There Other Read-Disturb Issues in DRAM?

- RowHammer is the only studied read-disturb phenomenon
- Mitigations work by detecting **high row activation count**

What if there is another read-disturb phenomenon that **does NOT rely on high row activation count**?

https://www.reddit.com/r/CrappyDesign/comments/arw0q8/now_this_this_is_poor_fencing/

SAFARI

Outline

What is RowPress?

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

These bitflips do **NOT** require many row activations

Only one activation is enough in some cases!

☞ Now, let's see how this is **different from RowHammer**
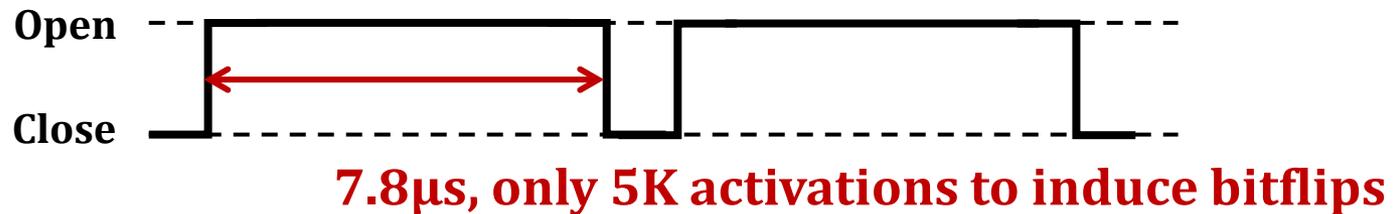
RowPress vs. RowHammer

Instead of using a high activation count,
☞ increase the time that the aggressor row stays open

RowHammer Aggressor Row

Open

Close

36ns, 47K activations to induce bitflips

RowPress Aggressor Row

Open

Close

7.8μs, only 5K activations to induce bitflips

We observe bitflips even with **ONLY ONE activation** in extreme cases where the row stays open for 30ms

Outline

Major Takeaways from Real DRAM Chips

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

RowPress has a **different** underlying failure **mechanism** from RowHammer

Outline

FPGA-based DDR4 testing infrastructure

- Developed from SoftMC [Hassan+, HPCA'17] and DRAM Bender [Olgun+, TCAD'23]

- Fine-grained control over DRAM commands, timings, and temperature
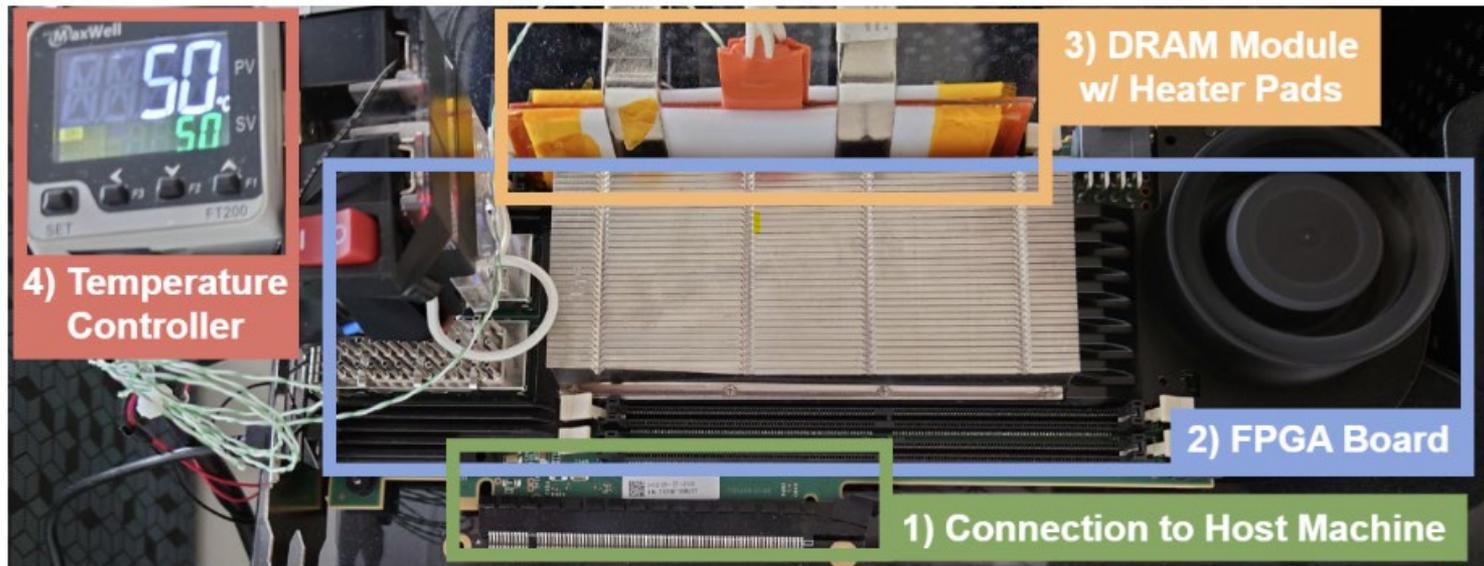
DRAM chips tested

- 164 DDR4 chips from all 3 major DRAM manufacturers
- Covers different die densities and revisions

Mfr.	#DIMMs	#Chips	Density	Die Rev.	Org.	Date
Mfr. S (Samsung)	2	8	8Gb	B	x8	20-53
	1	8	8Gb	C	x8	N/A
	3	8	8Gb	D	x8	21-10
	2	8	4Gb	F	x8	N/A
Mfr. H (SK Hynix)	1	8	4Gb	A	x8	19-46
	1	8	4Gb	X	x8	N/A
	2	8	16Gb	A	x8	20-51
	2	8	16Gb	C	x8	21-36
Mfr. M (Micron)	1	16	8Gb	B	x4	N/A
	2	4	16Gb	B	x16	21-26
	1	16	16Gb	E	x4	20-14
	2	4	16Gb	E	x16	20-46
	1	4	16Gb	F	x16	21-50

SAFARI

14

Characterization Methodology (III)

Metric: The minimum number of aggressor row activations in total to cause at least one bitflip (**ACmin**)

Access Pattern: Single-sided (i.e., only one aggressor row). Sweep aggressor row on time (**tAggON**) from 36ns to 30ms
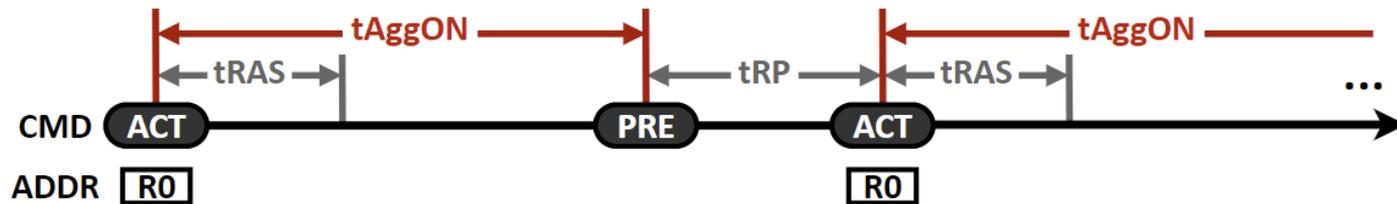
Data Pattern: Checkerboard (`0xAA` in aggressor and `0x55` in victim)

Temperature: 50°C

Algorithm: Bisection-based ACmin search
- **Each search iteration is capped at 60ms (<64ms refresh window)**
- Repeat 5 times and report the minimum ACmin value observed
- Sample 3072 DRAM rows per chip

[More sensitivity studies in the paper]

SAFARI

Outline

DRAM Background

What is RowPress?

Real DRAM Chip Characterization

Characterization Methodology

Key Characteristics of RowPress

Real-System Demonstration

Mitigating RowPress

Conclusion

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

RowPress has a **different** underlying failure **mechanism** from RowHammer

Key Characteristics of RowPress

Amplifying read disturbance in DRAM

- Reduces the minimum number of row activations needed to induce a bitflip (**ACmin**) by **1-2 orders of magnitude**

- In extreme cases, activating a row **only once** induces bitflips

- Gets worse as **temperature increases**

Different from RowHammer

- Affects a **different set of cells** compared to RowHammer and retention failures

- **Behaves differently** as access pattern or temperature changes compared to RowHammer

SAFARI

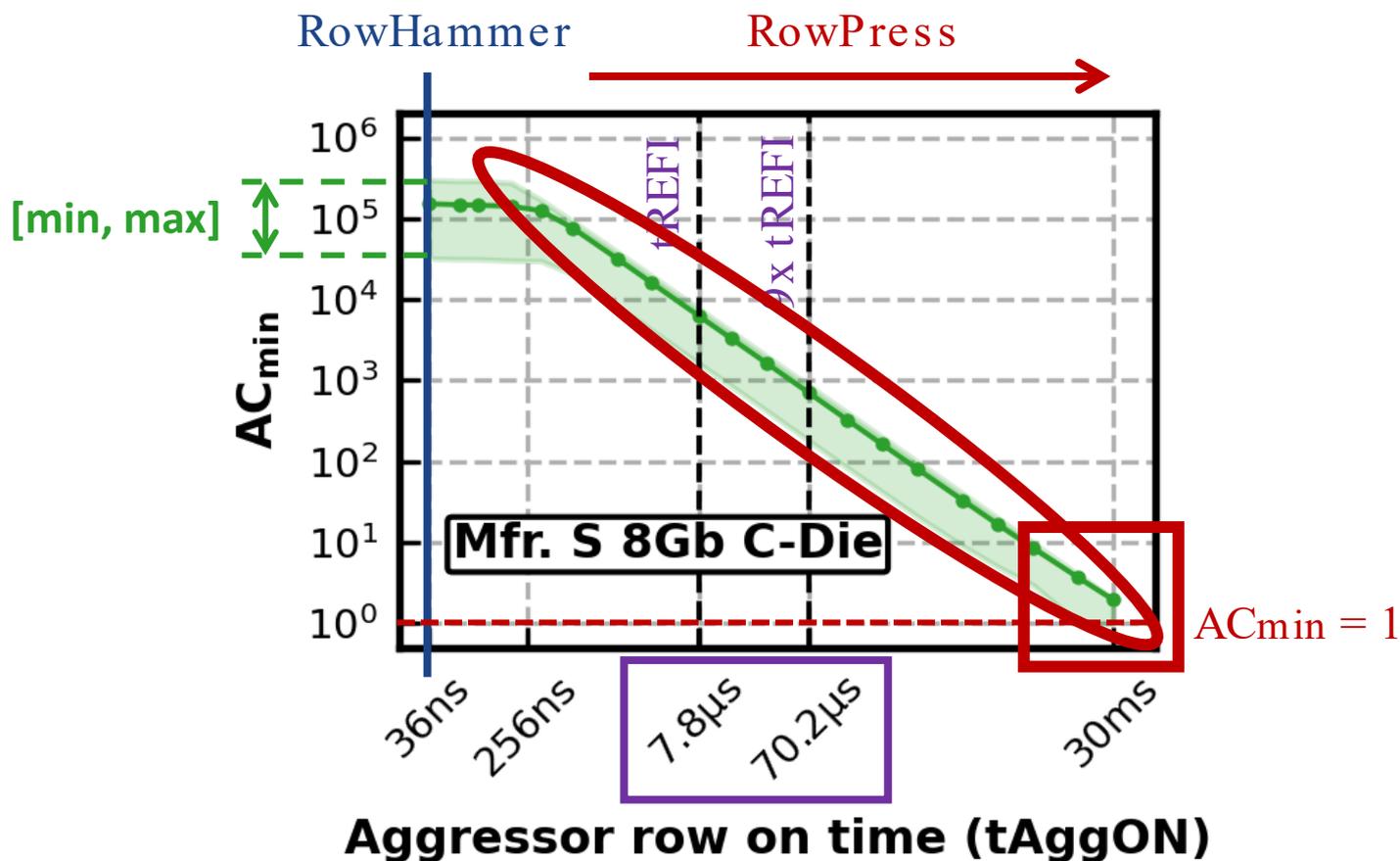# Key Characteristics of RowPress

Amplifying read disturbance in DRAM

- Reduces the minimum number of row activations needed to induce a bitflip (**ACmin**) by **1-2 orders of magnitude**

- In extreme cases, activating a row **only once** induces bitflips

- Gets worse as **temperature increases**

Different from RowHammer

- Affects a **different set of cells** compared to RowHammer and retention failures

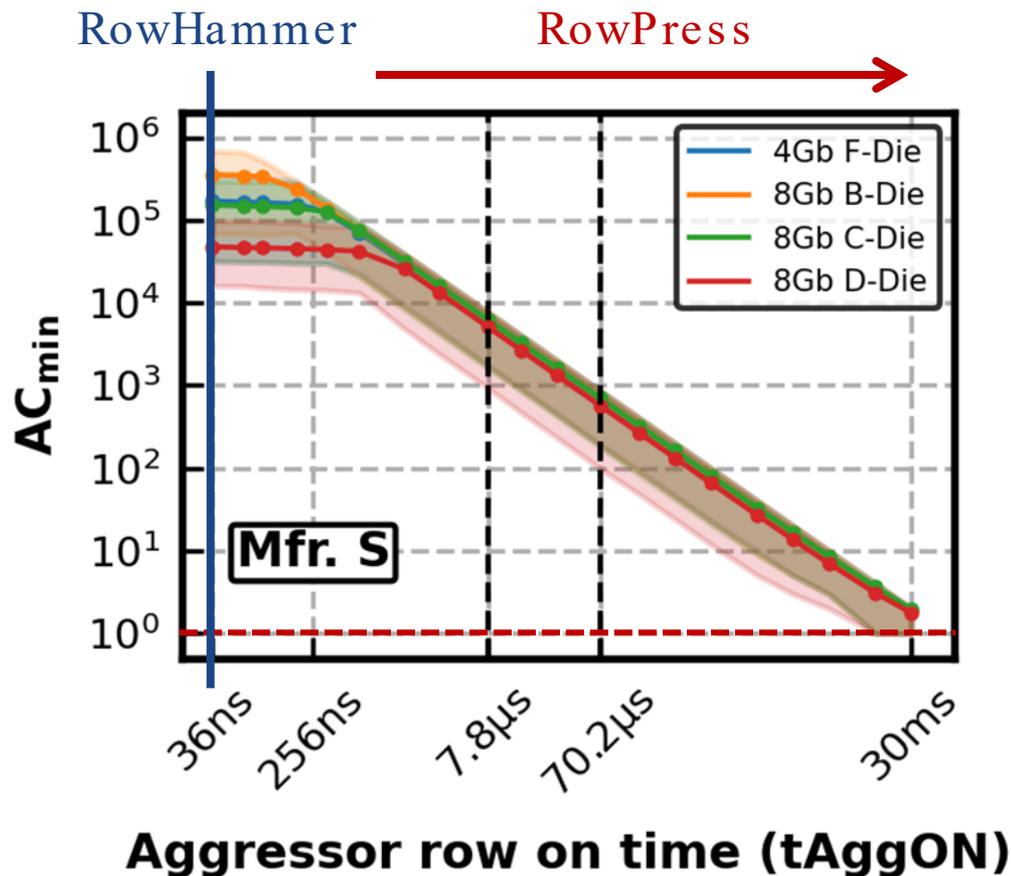- **Behaves differently** as access pattern or temperature changes compared to RowHammer

SAFARI

How minimum activation count to induce a bitflip (ACmin) changes as aggressor row on time (tAggON) increases

How minimum activation count to induce a bitflip (AC_{min}) changes as aggressor row on time (tAggON) increases

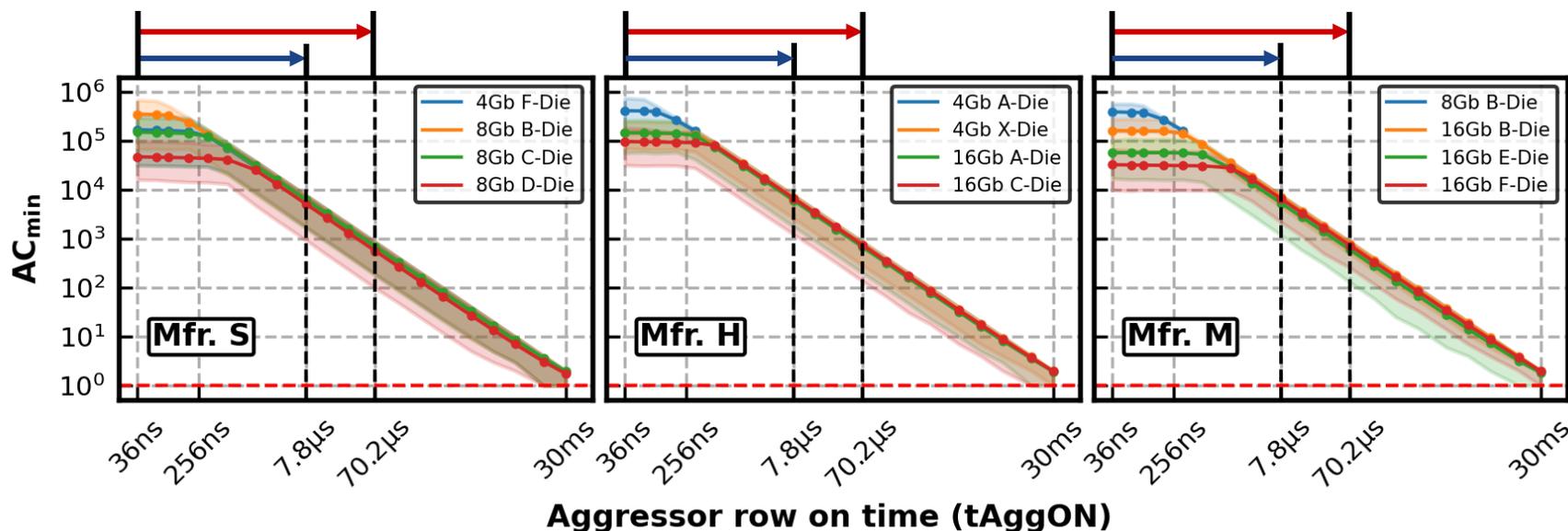**How minimum activation count to induce a bitflip (ACmin) changes as aggressor row on time (tAggON) increases**

ACmin reduces by **21X** on average when tAggON increases from 36ns to **7.8μs**

191X   **70.2μs**
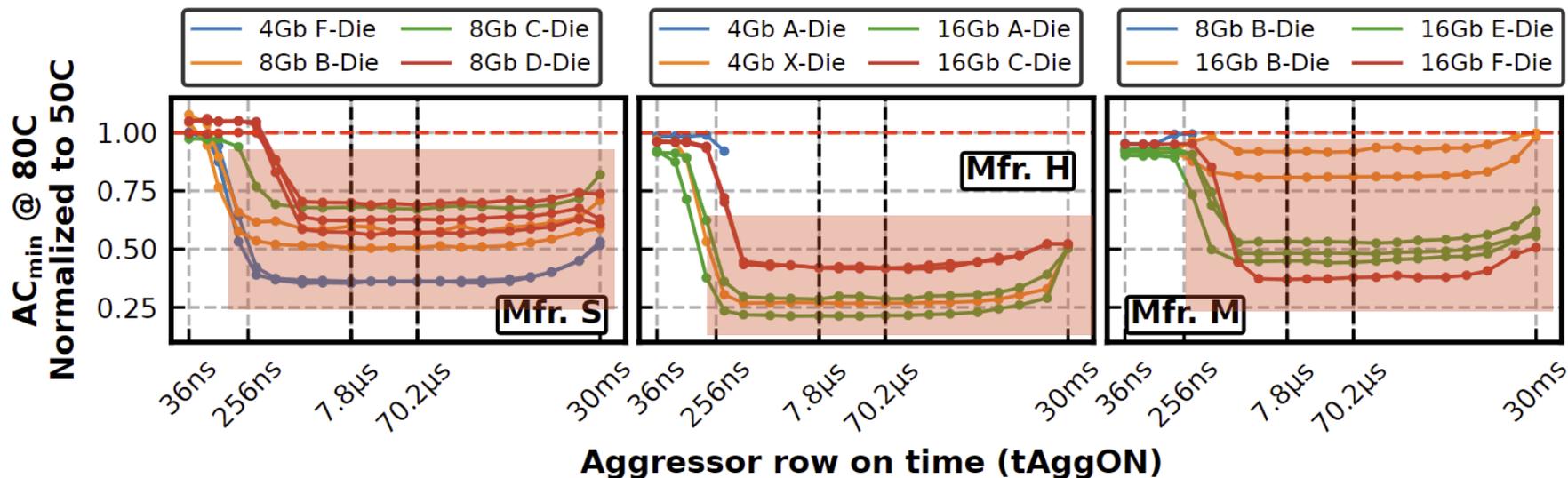
RowPress significantly reduces ACmin as tAggON increases

AC$_{min}$ @ 80°C normalized to AC$_{min}$ @ 50°C

- Data point below 1 means fewer activations to cause bitflips @ 80°C compared to 50°C

When tAggON is 7.8 µs, RowPress requires about 50% fewer activations to induce bitflips at 80°C compared to 50°C

RowPress gets worse as temperature increases

Key Characteristics of RowPress

Amplifying read disturbance in DRAM

- Reduces the minimum number of row activations needed to induce a bitflip (**ACmin**) by **1-2 orders of magnitude**

- In extreme cases, activating a row **only once** induces bitflips

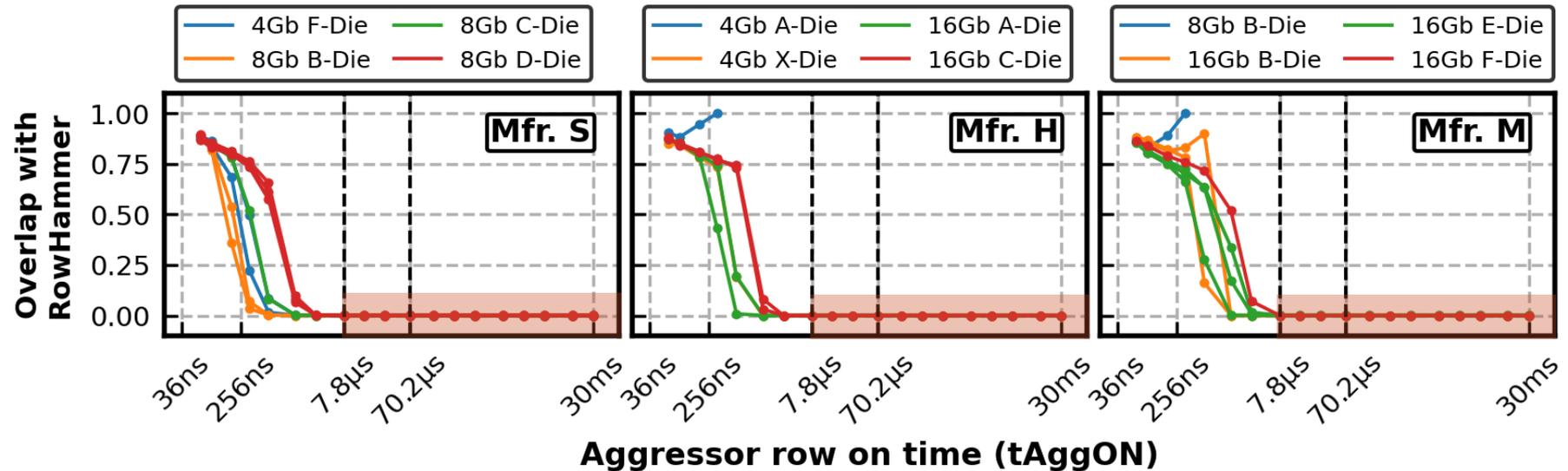- Gets worse as **temperature increases**

Different from RowHammer

- Affects a **different set of cells** compared to RowHammer and retention failures

- **Behaves differently** as access pattern or temperature changes compared to RowHammer

SAFARI

Cells vulnerable to RowPress vs. RowHammer

- Cells vulnerable to RowPress (RowHammer) are those that flip @ ACmin

- Overlap = $\dfrac{Number\ of\ Cells\ Vulnerable\ to\ Both\ RowPress\ and\ RowHammer}{Number\ of\ Cells\ Vulnerable\ to\ RowPress}$
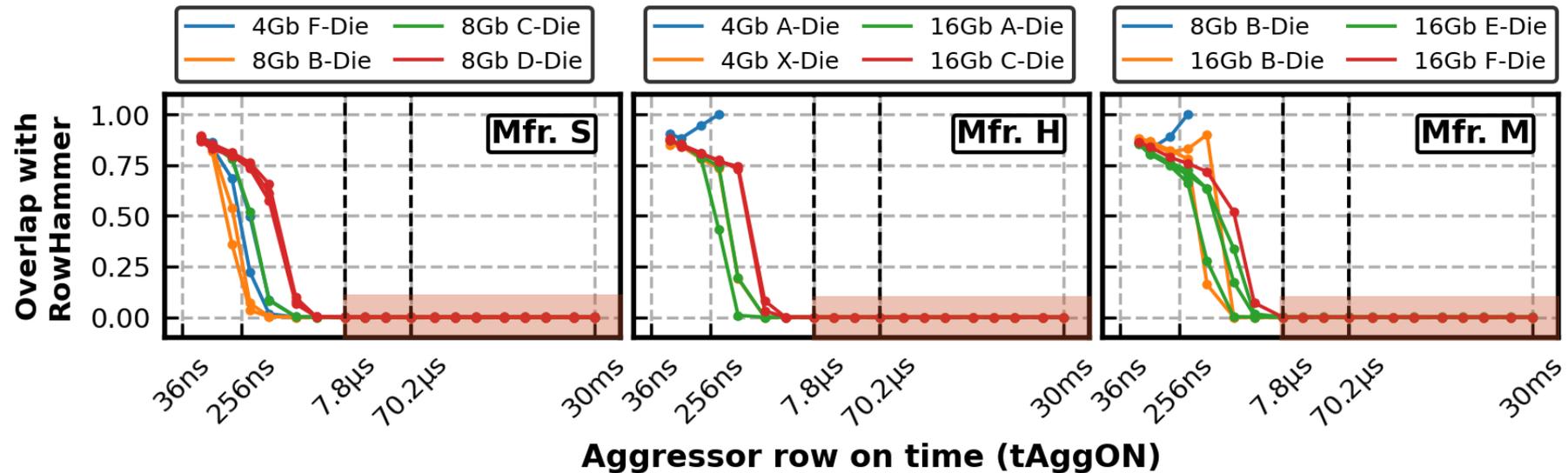

On average, only 0.013% of DRAM cells vulnerable to RowPress are also vulnerable to RowHammer, when tAggON ≥ 7.8us
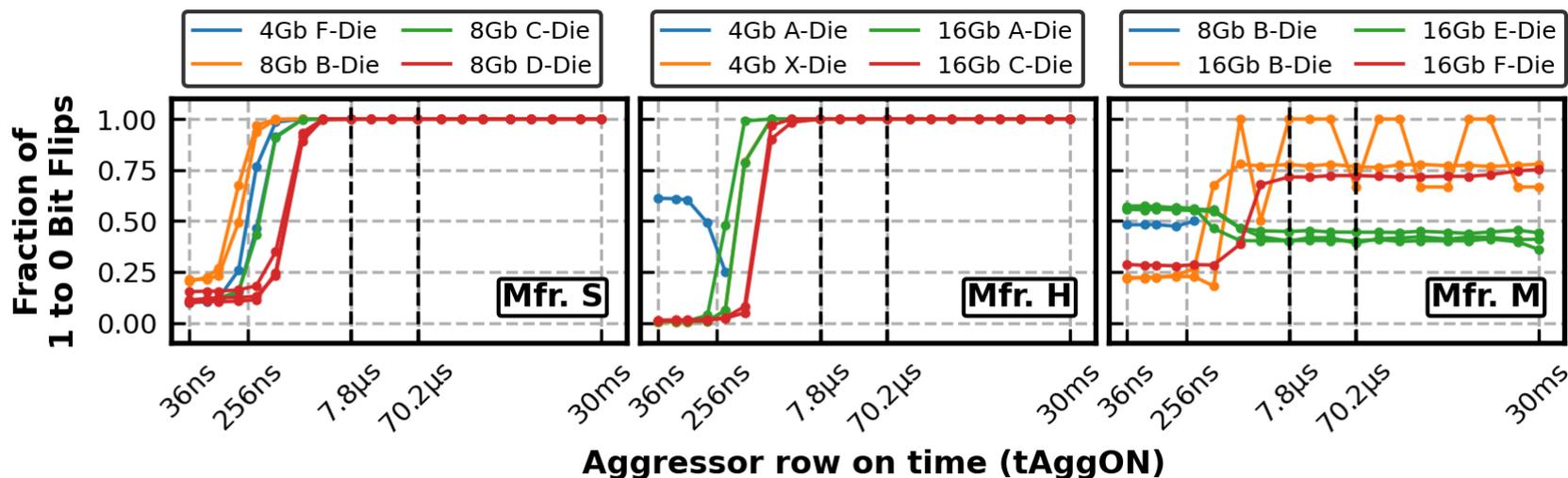
Cells vulnerable to RowPress vs. RowHammer

- Cells vulnerable to RowPress (RowHammer) are those that flip @ ACmin

- Overlap = $\dfrac{Number\ of\ Cells\ Vulnerable\ to\ Both\ RowPress\ and\ RowHammer}{Number\ of\ Cells\ Vulnerable\ to\ RowPress}$

**Most cells vulnerable to RowPress
are NOT vulnerable to RowHammer**

SAFARI

Directionality of RowHammer and RowPress bitflips
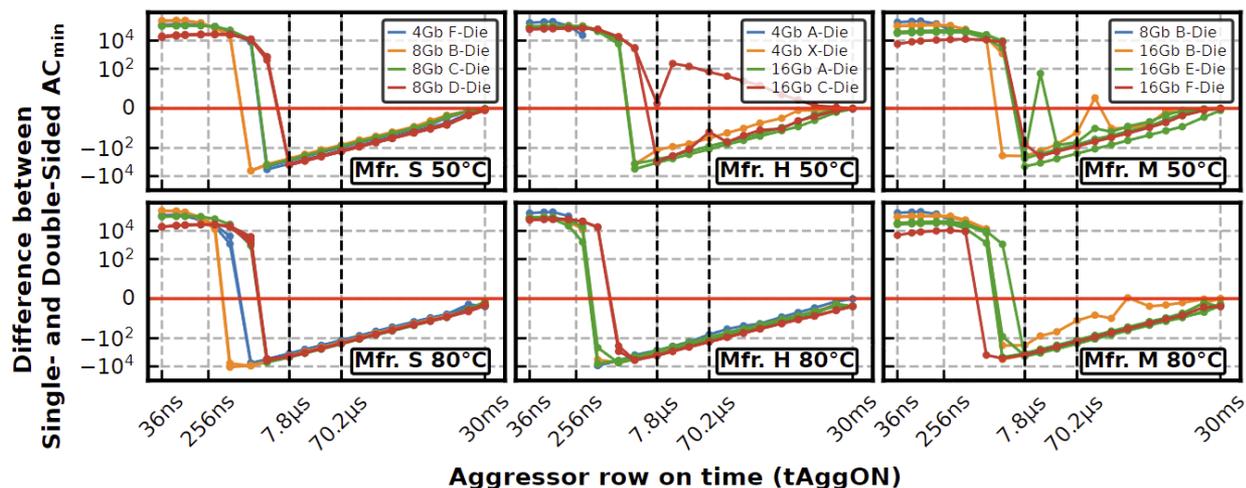

The majority of **RowHammer** bitflips are **0 to 1**
The majority of **RowPress** bitflips are **1 to 0**

RowPress and RowHammer bitflips have opposite directions

SAFARI

Effectiveness of single-sided vs. double-sided RowPress

- Data point below 0 means fewer activations to cause bitflips with single-sided RowPress compared to double-sided RowPress

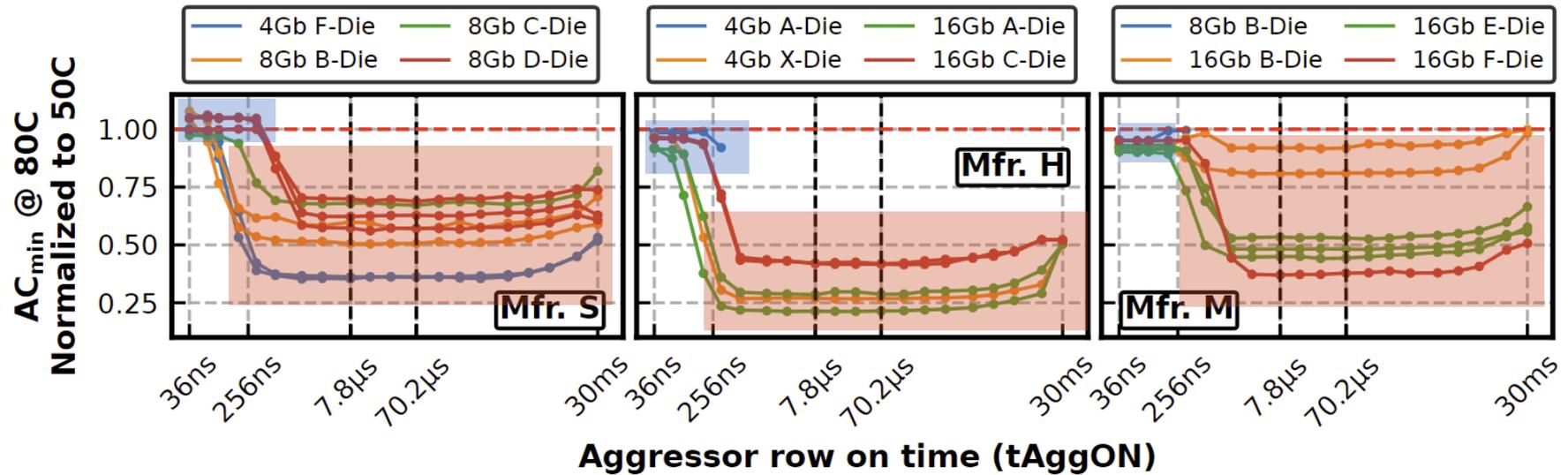As tAggON increases beyond a certain level, **single-sided RowPress becomes more effective** compared to double-sided

Different from RowHammer where **double-sided is more effective**

Sensitivity to temperature

- Data point below 1 means fewer activations to cause bitflips @ 80°C compared to 50°C

RowPress gets worse as temperature increases, which is **very different from RowHammer**

Outline

Intel Core i5-10400
(Comet Lake)

Samsung DDR4 Module
M378A2K43CB1-CTD
(Date Code: 20-10)
w/ TRR RowHammer Mitigation

Key Idea: A proof-of-concept RowPress program keeps a DRAM row open for a longer period by **keeping on accessing different cache blocks in the row**

Facilitated by adaptive row policy in the memory controller

```
// Sync with Refresh and Loop Below
  for (k = 0; k < NUM_AGGR_ACTS; k++)
    for (j = 0; j < NUM_READS; j++)  *AGGRESSOR1[j];
    for (j = 0; j < NUM_READS; j++)  *AGGRESSOR2[j];
    for (j = 0; j < NUM_READS; j++)
      clflushopt(AGGRESSOR1[j]);
      clflushopt(AGGRESSOR2[j]);
    mfence();
  activate_dummy_rows();
```
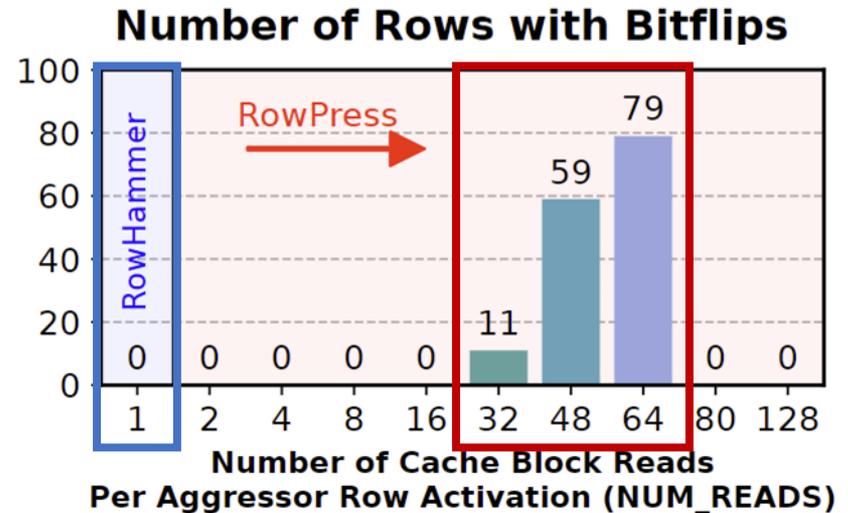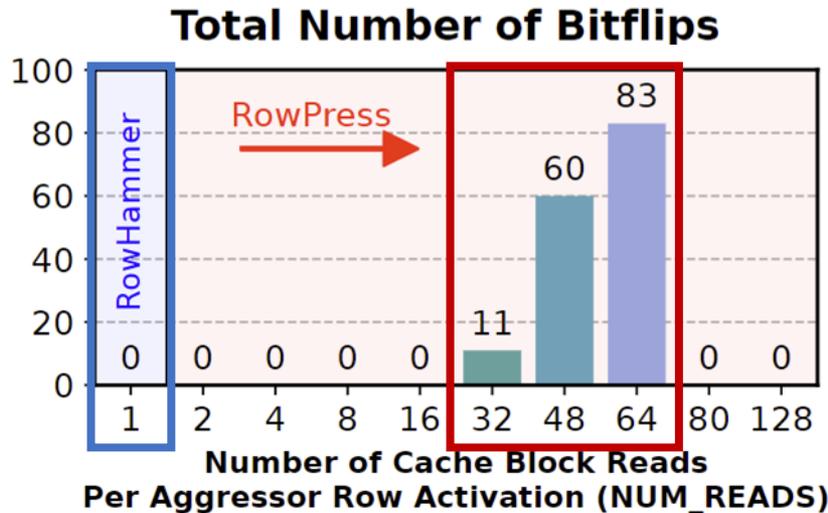
Number of Cache Blocks Accessed Per Aggressor Row ACT
(NUM_READS=1 is Rowhammer)

On 1500 victim rows

Total Number of Bitflips

Number of Rows with Bitflips

Number of Cache Block Reads Per Aggressor Row Activation (NUM_READS)

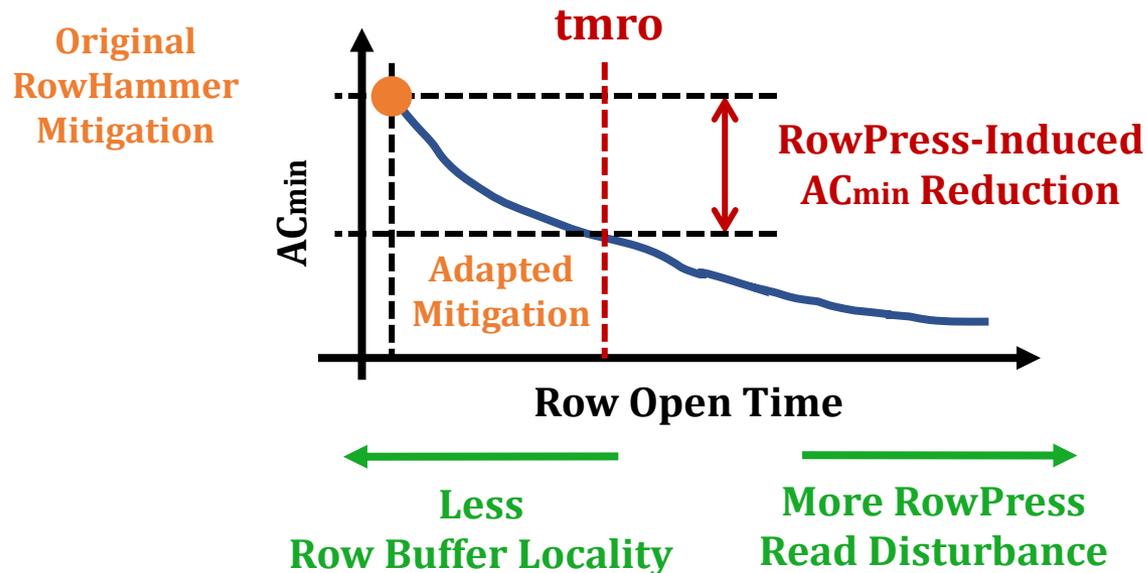**Leveraging RowPress, our user-level program induces bitflips when RowHammer cannot**

Outline

Mitigating RowPress (I)

We propose a methodology to adapt existing RowHammer mitigations to **also mitigate RowPress**

Key Idea:

1. Limit the maximum row open time (**tmro**)
2. Configure the RowHammer mitigation to account for the **RowPress-induced reduction in AC_{min}**

Evaluation methodology

- **Adapted RowHammer Mitigations:** Graphene (**Graphene-RP**) and PARA (**PARA-RP**)

- Cycle-accurate DRAM simulator: Ramulator [Kim+, CAL'15]
  - 4 GHz Out-of-Order Core, dual-rank DDR4 DRAM
  - FR-FCFS scheduling
  - Open-row policy (with limited maximum row open time)

- 58 four-core multiprogrammed workloads from SPEC CPU2017, TPC-H, and YCSB

- **Metric: Additional performance overhead** of Graphene-RP (PARA-RP) over Graphene (PARA)
  - Measured by weighted speedup

SAFARI

Mitigating RowPress (III)

Key evaluation results

Additional Performance Overhead of
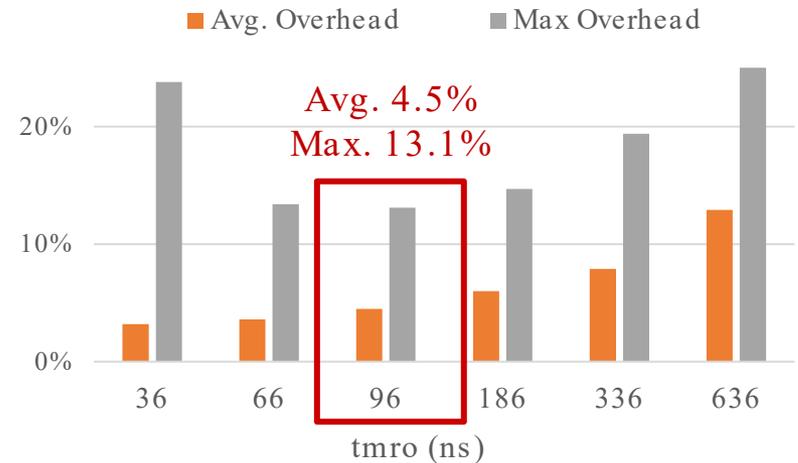Graphene-RP

■ Avg. Overhead   ■ Max Overhead

Avg. -0.63%
Max. 6.4%
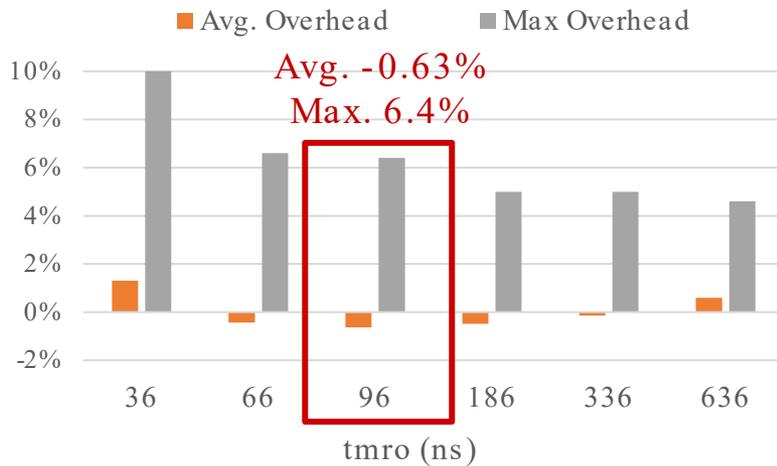
Additional Performance Overhead of
PARA-RP

■ Avg. Overhead   ■ Max Overhead

Avg. 4.5%
Max. 13.1%

tmro (ns)

tmro (ns)

Our solutions **mitigate RowPress** at **low additional performance overhead**

Outline

Conclusion

We demonstrate and analyze **RowPress, a widespread read disturbance phenomenon** that causes bitflips in real DRAM chips

We **characterize RowPress** on 164 DDR4 chips from all 3 major DRAM manufacturers
- RowPress greatly amplifies read disturbance: minimum activation count reduces by 1-2 orders of magnitude
- RowPress has a different mechanism from RowHammer & retention failures

We **demonstrate RowPress** using **a user-level program**
- Induces bitflips when RowHammer cannot

We provide **effective solutions** to RowPress
- Low additional performance overhead

SAFARI

More Results & Source Code

Many more results & analyses in the paper

- 6 major takeaways

- 19 major empirical observations

- 3 more potential mitigations

Fully open source and artifact evaluated

- https://github.com/CMU-SAFARI/RowPress

RowPress
Amplifying Read Disturbance in Modern DRAM Chips

Haocong Luo

Ataberk Olgun   A. Giray Yağlıkçı   Yahya Can Tuğrul

Steve Rhyner   Meryem Banu Cavlak   Joël Lindegger
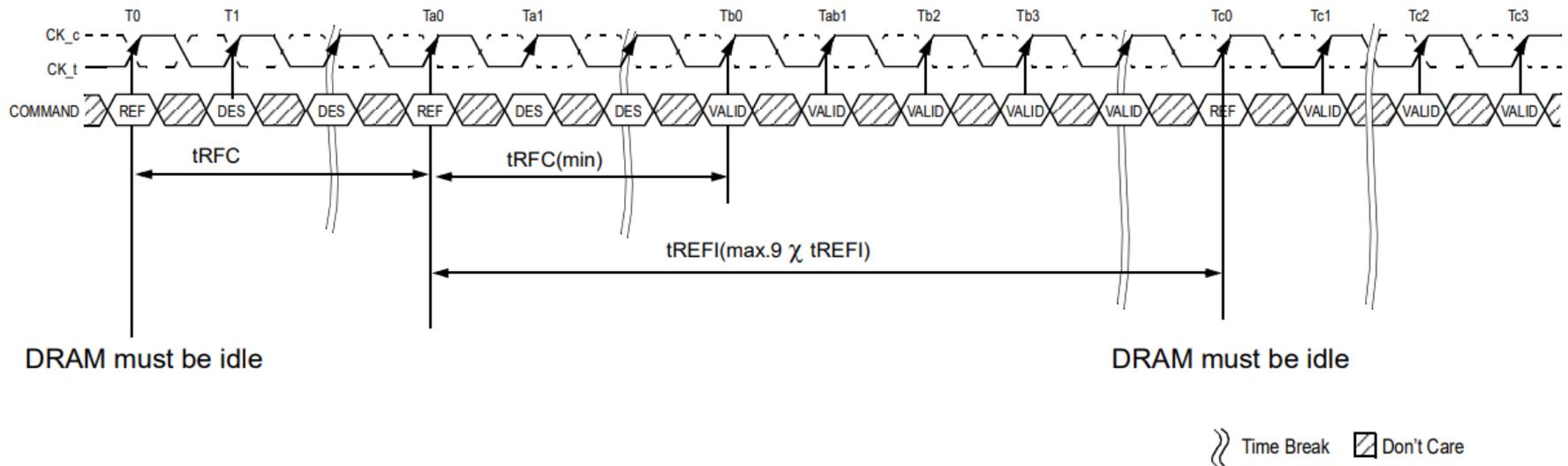
Mohammad Sadrosadati   Onur Mutlu

https://github.com/CMU-SAFARI/RowPress

SAFARI   ETH Zürich

Potential tAggON upper bounds

- **tREFI**: Interval between two REF commands
- **9tREFI**:

Figure 157 — Refresh Command Timing (Example of 1x Refresh mode)

NOTE 1 Only DES commands allowed after Refresh command registered until tRFC(min) expires.
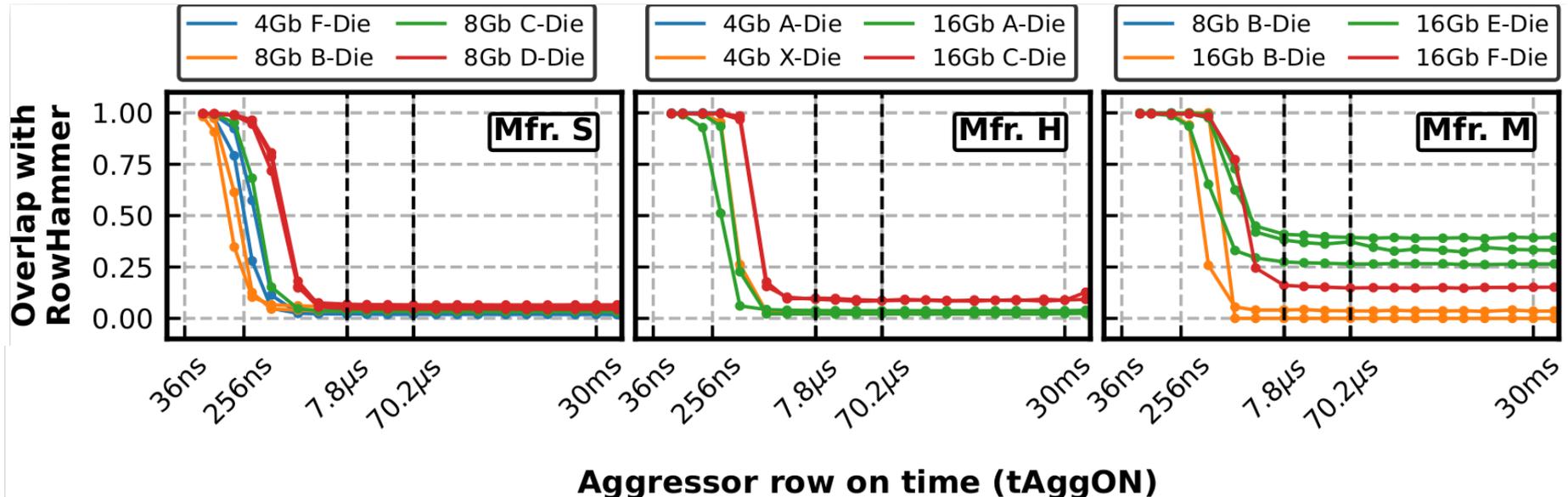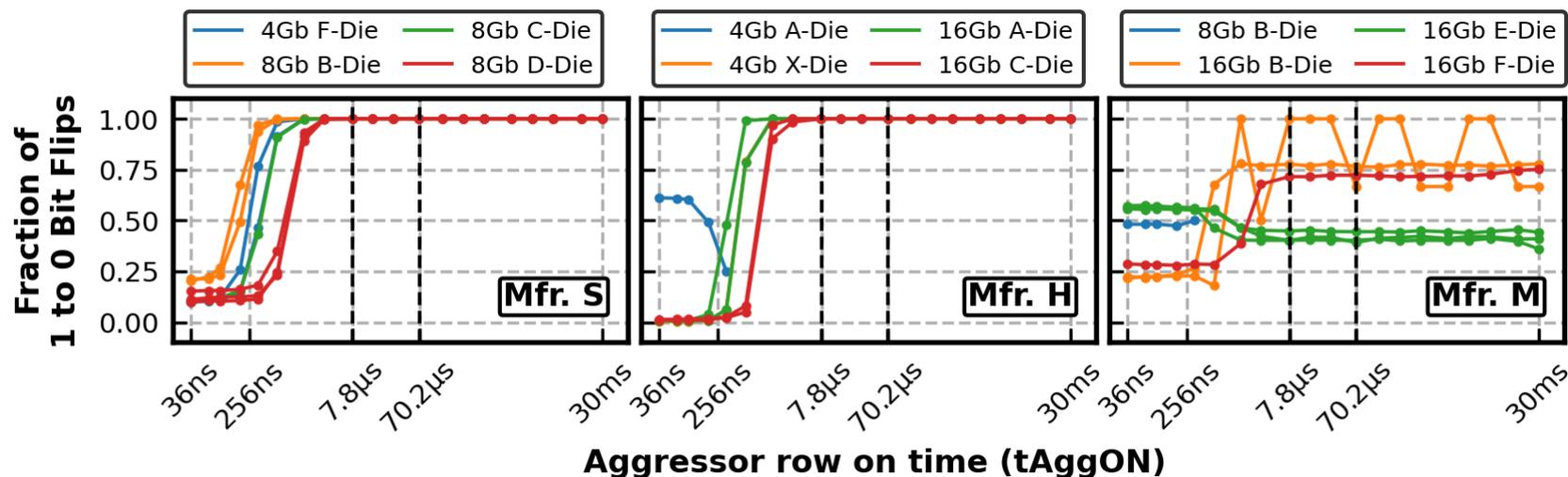NOTE 2 Time interval between two Refresh commands may be extended to a maximum of 9 X tREFI.

JESD79-4C

SAFARI

Cells vulnerable to RowPress vs RowHammer

- Cells vulnerable to RowPress (RowHammer) are those that flip @ ACmax

- Overlap = $\dfrac{Number\ of\ Cells\ Vulnerable\ to\ Both\ RowPress\ and\ RowHammer}{Number\ of\ Cells\ Vulnerable\ to\ RowPress}$

SAFARI

Backup Slide

Directionality of RowHammer and RowPress bitflips

The majority of **RowHammer** bitflips are **0 to 1**
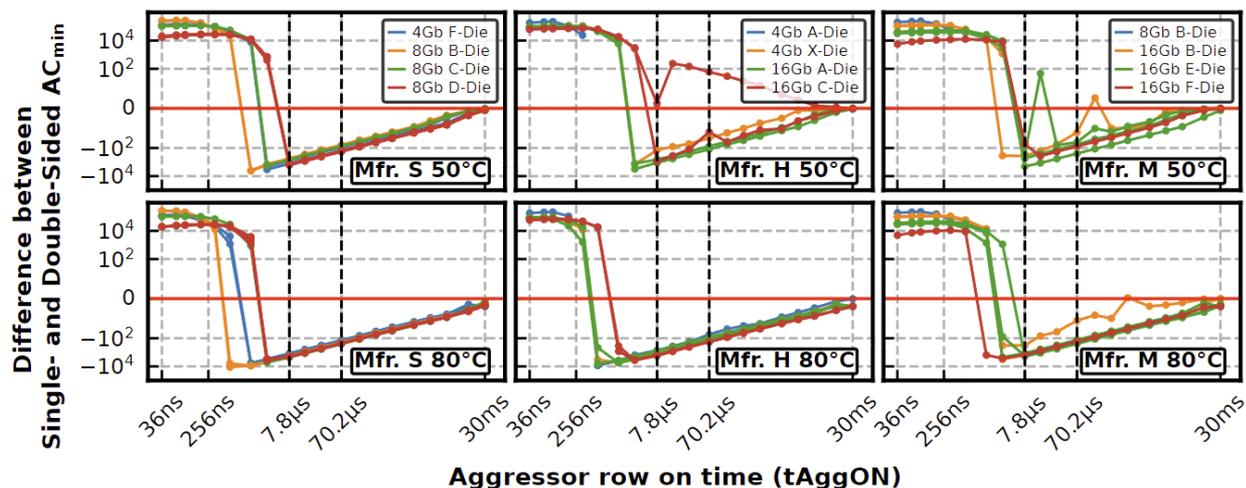The majority of **RowPress** bitflips are **1 to 0**

RowPress and RowHammer bitflips have opposite directions

SAFARI

Effectiveness of single-sided vs double-sided RowPress

- Data point below 0 means fewer activations to cause bitflips with single-sided RowPress compared to double-sided RowPress
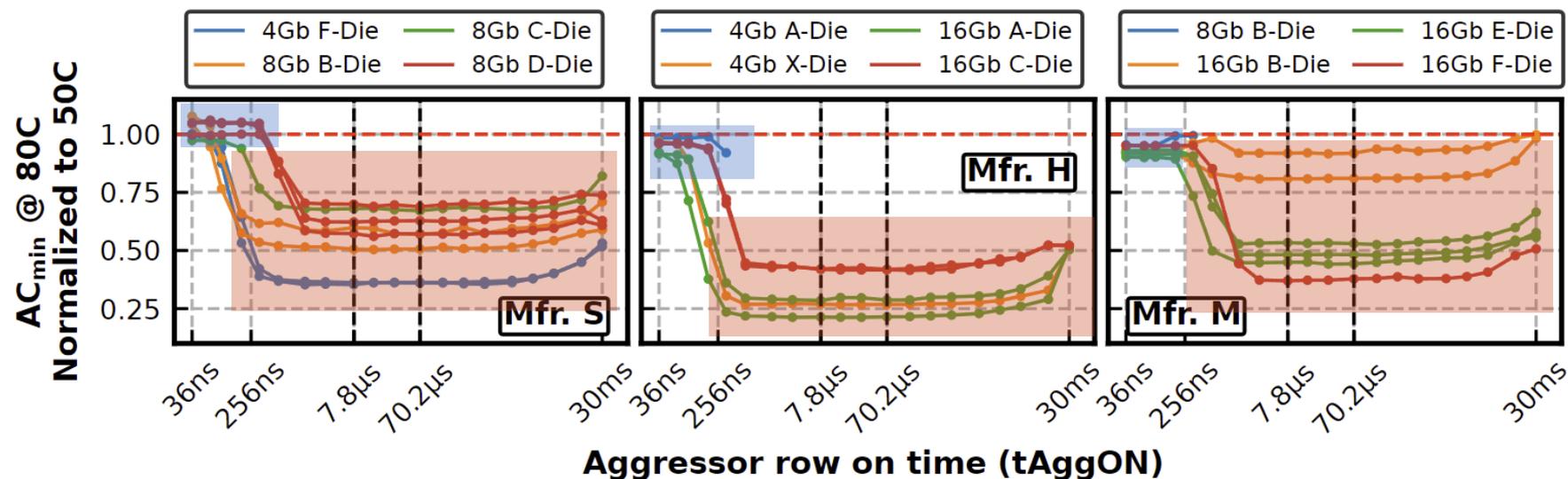

As tAggON increases beyond a certain level, **single-sided RowPress becomes more effective** compared to double-sided

Different from RowHammer where **double-sided is more effective**

SAFARI

Sensitivity to temperature

- Data point below 1 means fewer activations to cause bitflips @ 80°C compared to 50°C

RowPress gets worse as temperature increases,
which is **very different from RowHammer**

Backup Slide

RowPress significantly reduces ACmin as tAggON increases

**Most Cells Vulnerable to RowPress
are NOT vulnerable to RowHammer**

RowPress and RowHammer bitflips have opposite directions

As tAggON increases beyond a certain level, **single-sided RowPress becomes more effective** compared to double-sided

SAFARI

Device-Level Explanations

- AWL-induced "1" failure is mainly caused by electron migration during the OFF period of AWL, not affected by increased tAggON

- AWL-induced "0" failure is mainly caused by the increased victim SN leakage caused by the strong electric field near AWL when AWL is ON

- PWL-induced "0" failure is mainly caused by charge capture by e-traps near PWL, not affected by increased tAggON after trap saturation

- PWL-induced "1" failure is mainly caused by the strong electric field near PWL attracting electrons towards the direction of victim SN
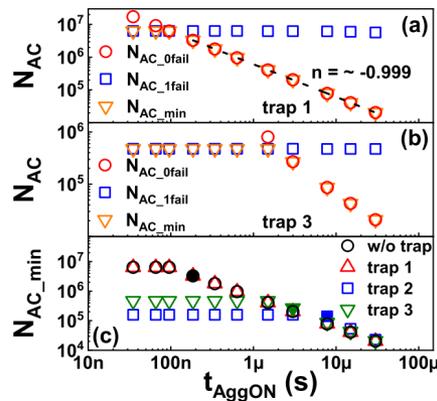

Fig. 3. AWL-induced RH and RP: N_{AC_0fail}, N_{AC_1fail}, and N_{AC_min} as functions of t_{AggON} for (a) trap 1 and (b) trap 3. (c) Dependence of N_{AC_min} on t_{AggON} under different trap conditions, highlighted by a solid point at the transition t_{AggON}.
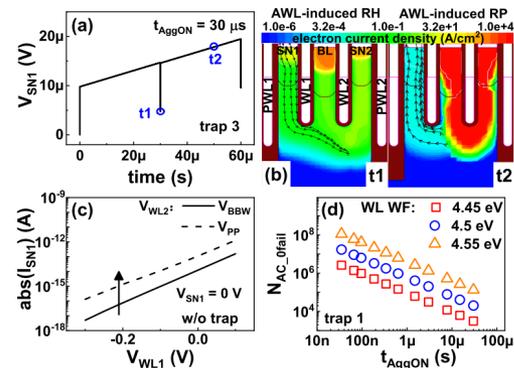
Fig. 4. Analysis of the EM mechanism in AWL-induced RP. (a) Time kinetics of V_{SN1} with t_{AggON} = 30 μs for "0" failure. (b) ECD contours at different time points in (a) are depicted, along with the direction of the ECD. (c) I_{SN1}–V_{WL1} curves for different V_{WL2} conditions. (d) Dependence of N_{AC_0fail} on t_{AggON} under different WL WFs.

Zhou et al., "Unveiling RowPress in Sub-20 nm DRAM Through Comparative Analysis With RowHammer: From Leakage Mechanisms to Key Features," in IEEE TRANSACTIONS ON ELECTRON DEVICES, 2014

SAFARI

47