RowPress
Amplifying Read Disturbance in Modern DRAM Chips

_Haocong Luo_  
Ataberk Olgun  A. Giray Yaşlalıkçı  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
<table>
<thead>
<tr>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRAM Background</td>
</tr>
<tr>
<td>What is RowPress?</td>
</tr>
<tr>
<td>Real DRAM Chip Characterization</td>
</tr>
<tr>
<td>Characterization Methodology</td>
</tr>
<tr>
<td>Key Characteristics of RowPress</td>
</tr>
<tr>
<td>Real-System Demonstration</td>
</tr>
<tr>
<td>Mitigating RowPress</td>
</tr>
<tr>
<td>Conclusion</td>
</tr>
</tbody>
</table>
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/
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, 
增加 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

- 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
Outline

DRAM Background

What is RowPress?

Real DRAM Chip Characterization

Characterization Methodology

Key Characteristics of RowPress

Real-System Demonstration

Mitigating RowPress

Conclusion
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**

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

**Metric:** The minimum number of aggressor row activations in total to cause at least one bitflip (AC\textsubscript{min})

**Access Pattern:** Single-sided (i.e., only one aggressor row). Sweep aggressor row on time (t\textsubscript{AggON}) from 36ns to 30ms

**Data Pattern:** Checkerboard (0xAA in aggressor and 0x55 in victim)

**Temperature:** 50°C

**Algorithm:** Bisection-based AC\textsubscript{min} search

- Each search iteration is capped at 60ms (<64ms refresh window)
- Repeat 5 times and report the minimum AC\textsubscript{min} value observed
- Sample 3072 DRAM rows per chip

[More sensitivity studies in the paper]
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 ($AC_{min}$) 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
Key Characteristics of RowPress

Amplifying read disturbance in DRAM

- Reduces the minimum number of row activations needed to induce a bitflip ($AC_{\text{min}}$) by \textbf{1-2 orders of magnitude}
- In extreme cases, activating a row \textbf{only once} induces bitflips
- Gets worse as \textbf{temperature increases}

Different from RowHammer

- Affects a \textbf{different set of cells} compared to RowHammer and retention failures
- \textbf{Behaves differently} as access pattern or temperature changes compared to RowHammer
Amplifying Read Disturbance (I)

How minimum activation count to induce a bitflip ($\text{AC}_\text{min}$) changes as aggressor row on time ($t_{\text{AggON}}$) increases.
How minimum activation count to induce a bitflip ($AC_{\text{min}}$) changes as aggressor row on time ($t_{\text{AggON}}$) increases.
How minimum activation count to induce a bitflip ($AC_{\text{min}}$) changes as aggressor row on time ($t_{\text{AggON}}$) increases.

$AC_{\text{min}}$ reduces by 21X on average when $t_{\text{AggON}}$ increases from 36ns to 7.8µs and 191X to 70.2µs.

RowPress significantly reduces $AC_{\text{min}}$ as $t_{\text{AggON}}$ increases.
Amplifying Read Disturbance (IV)

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 ($AC_{\text{min}}$) 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
Cells vulnerable to RowPress vs. RowHammer

- Cells vulnerable to RowPress (RowHammer) are those that flip @ ACmin
- Overlap = \( \frac{\text{Number of Cells Vulnerable to Both RowPress and RowHammer}}{\text{Number of Cells Vulnerable to RowPress}} \)

On average, only 0.013% of DRAM cells vulnerable to RowPress are also vulnerable to RowHammer, when \( t_{\text{AggON}} \geq 7.8\mu s \)
Difference Between RowPress and RowHammer (II)

**Cells vulnerable to RowPress vs. RowHammer**

- Cells vulnerable to RowPress (RowHammer) are those that flip @ ACmin
- Overlap = \( \frac{\text{Number of Cells Vulnerable to Both RowPress and RowHammer}}{\text{Number of Cells Vulnerable to RowPress}} \)

Most cells vulnerable to RowPress are NOT vulnerable to RowHammer
Directionality of RowHammer and RowPress bitflips

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

**RowPress and RowHammer bitflips have opposite directions**
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
**Real-System Demonstration (I)**

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**

```c
// 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];
    clflushopt(AGGRESSOR1[j]);
    clflushopt(AGGRESSOR2[j]);
    mfence();
activate_dummy_rows();
```

**Number of Cache Blocks Accessed Per Aggressor Row ACT**
(NUM_READS=1 is Rowhammer)
Real-System Demonstration (II)

On 1500 victim rows

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

- DRAM Background
- What is RowPress?
- Real DRAM Chip Characterization
  - Characterization Methodology
  - Key Characteristics of RowPress
- Real-System Demonstration
- Mitigating RowPress
- Conclusion
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 \(AC_{\text{min}}\).

![Diagram showing the mitigation process](image)
Mitigating RowPress (II)

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
Mitigating RowPress (III)

Key evaluation results

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

- DRAM Background
- What is RowPress?
- Real DRAM Chip Characterization
  - Characterization Methodology
  - Key Characteristics of RowPress
- Real-System Demonstration
- Mitigating RowPress
- Conclusion
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
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
Potential tAggON upper bounds

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

![Diagram of REF and DES commands with tRFC, tRFC(min), and tREFI(max)]

**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.

**Figure 157 — Refresh Command Timing (Example of 1x Refresh mode)**
Cells vulnerable to RowPress vs RowHammer

- Cells vulnerable to RowPress (RowHammer) are those that flip at AC_max
- Overlap = \( \frac{\text{Number of Cells Vulnerable to Both RowPress and RowHammer}}{\text{Number of Cells Vulnerable to RowPress}} \)
Directionality of RowHammer and RowPress bitflips

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

RowPress and RowHammer bitflips have opposite directions
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.
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.