

# Computer Architecture

## Lecture 4c: RowHammer

Prof. Onur Mutlu  
ETH Zürich  
Fall 2021  
8 October 2021

# Four Key Problems + Directions

---

- Fundamentally Secure/Reliable/Safe Architectures
- Fundamentally Energy-Efficient Architectures
  - Memory-centric (Data-centric) Architectures
- Fundamentally Low-Latency and Predictable Architectures
- Architectures for AI/ML, Genomics, Medicine, Health

# The Story of RowHammer

- One can predictably induce bit flips in commodity DRAM chips
  - >80% of the tested DRAM chips are vulnerable
- First example of how a simple hardware failure mechanism can create a widespread system security vulnerability

WIRED

Forget Software—Now Hackers Are Exploiting Physics

BUSINESS

CULTURE

DESIGN

GEAR

SCIENCE

ANDY GREENBERG SECURITY 08.31.16 7:00 AM

SHARE

 SHARE  
18276

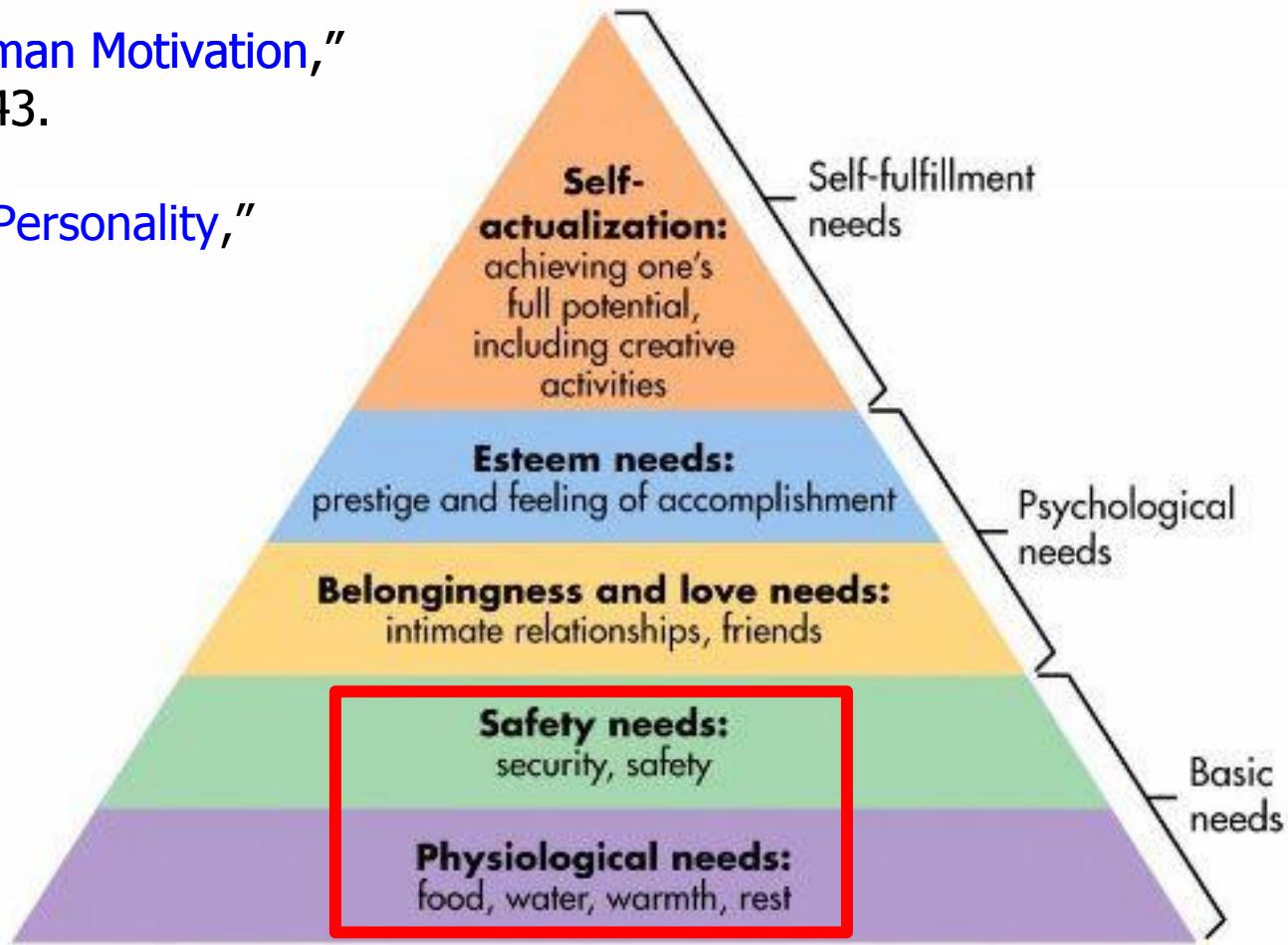
 TWEET

# FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

# Maslow's (Human) Hierarchy of Needs

Maslow, "A Theory of Human Motivation,"  
Psychological Review, 1943.

Maslow, "Motivation and Personality,"  
Book, 1954-1970.



- We need to start with **reliability and security...**

# How Reliable/Secure/Safe is This Bridge?

---



# Collapse of the “Galloping Gertie”

---



# How Secure Are These People?

---

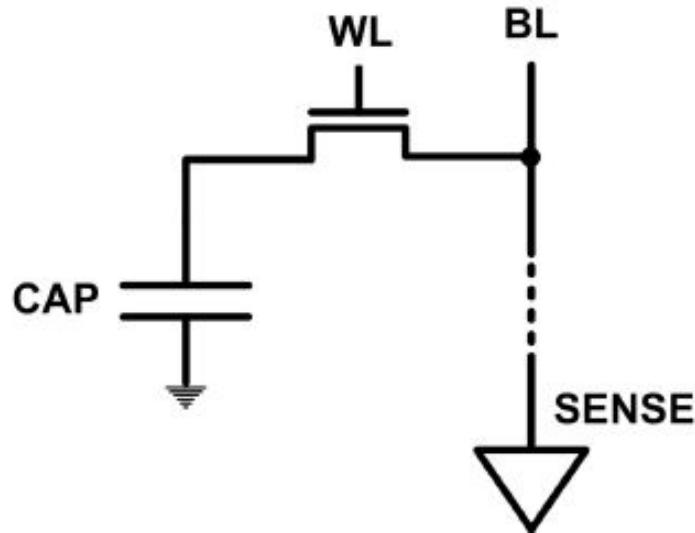


**Security is about preventing unforeseen consequences**

# The DRAM Scaling Problem

---

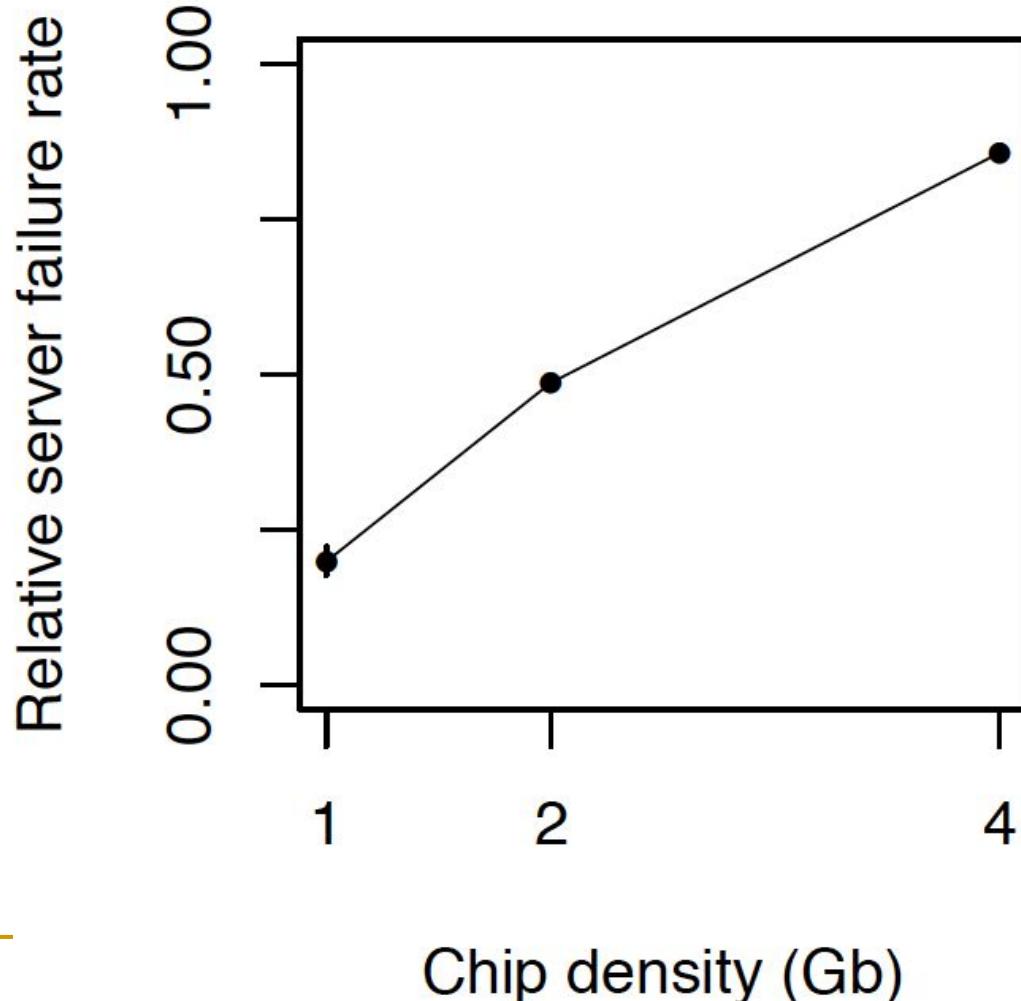
- DRAM stores charge in a capacitor (charge-based memory)
  - Capacitor must be large enough for reliable sensing
  - Access transistor should be large enough for low leakage and high retention time
  - Scaling beyond 40-35nm (2013) is challenging [ITRS, 2009]



- DRAM capacity, cost, and energy/power hard to scale

# As Memory Scales, It Becomes Unreliable

- Data from all of Facebook's servers worldwide
- Meza+, "Revisiting Memory Errors in Large-Scale Production Data Centers," DSN'15.



*Intuition:  
quadratic  
increase  
in  
capacity*

# Large-Scale Failure Analysis of DRAM Chips

---

- Analysis and modeling of memory errors found in all of Facebook's server fleet
- Justin Meza, Qiang Wu, Sanjeev Kumar, and Onur Mutlu,  
**"Revisiting Memory Errors in Large-Scale Production Data Centers: Analysis and Modeling of New Trends from the Field"**  
*Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Rio de Janeiro, Brazil, June 2015.  
[[Slides \(pptx\)](#)] [[\(pdf\)](#)] [[DRAM Error Model](#)]

Revisiting Memory Errors in Large-Scale Production Data Centers:  
Analysis and Modeling of New Trends from the Field

Justin Meza   Qiang Wu \*   Sanjeev Kumar \*   Onur Mutlu  
Carnegie Mellon University   \* Facebook, Inc.

# Infrastructures to Understand Such Issues



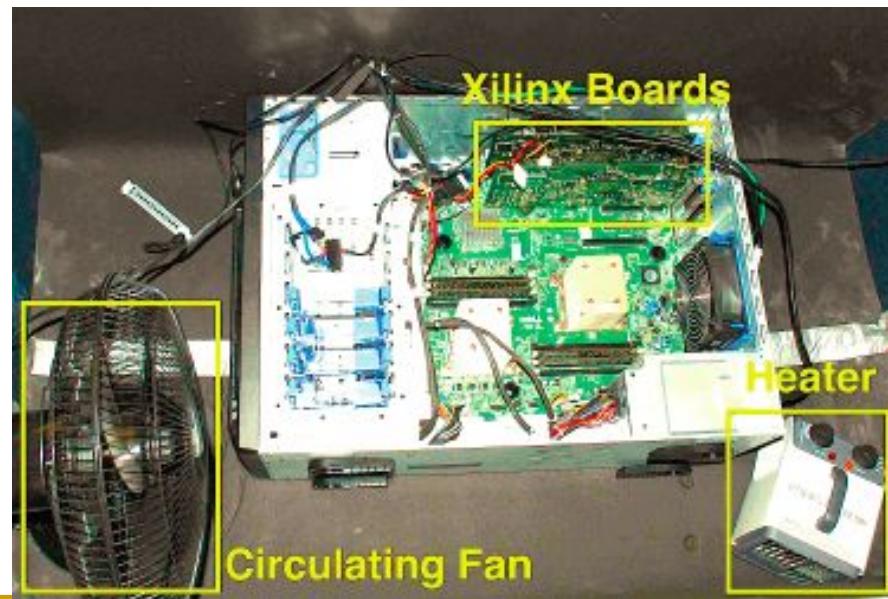
[Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors](#) (Kim et al., ISCA 2014)

[Adaptive-Latency DRAM: Optimizing DRAM Timing for the Common-Case](#) (Lee et al., HPCA 2015)

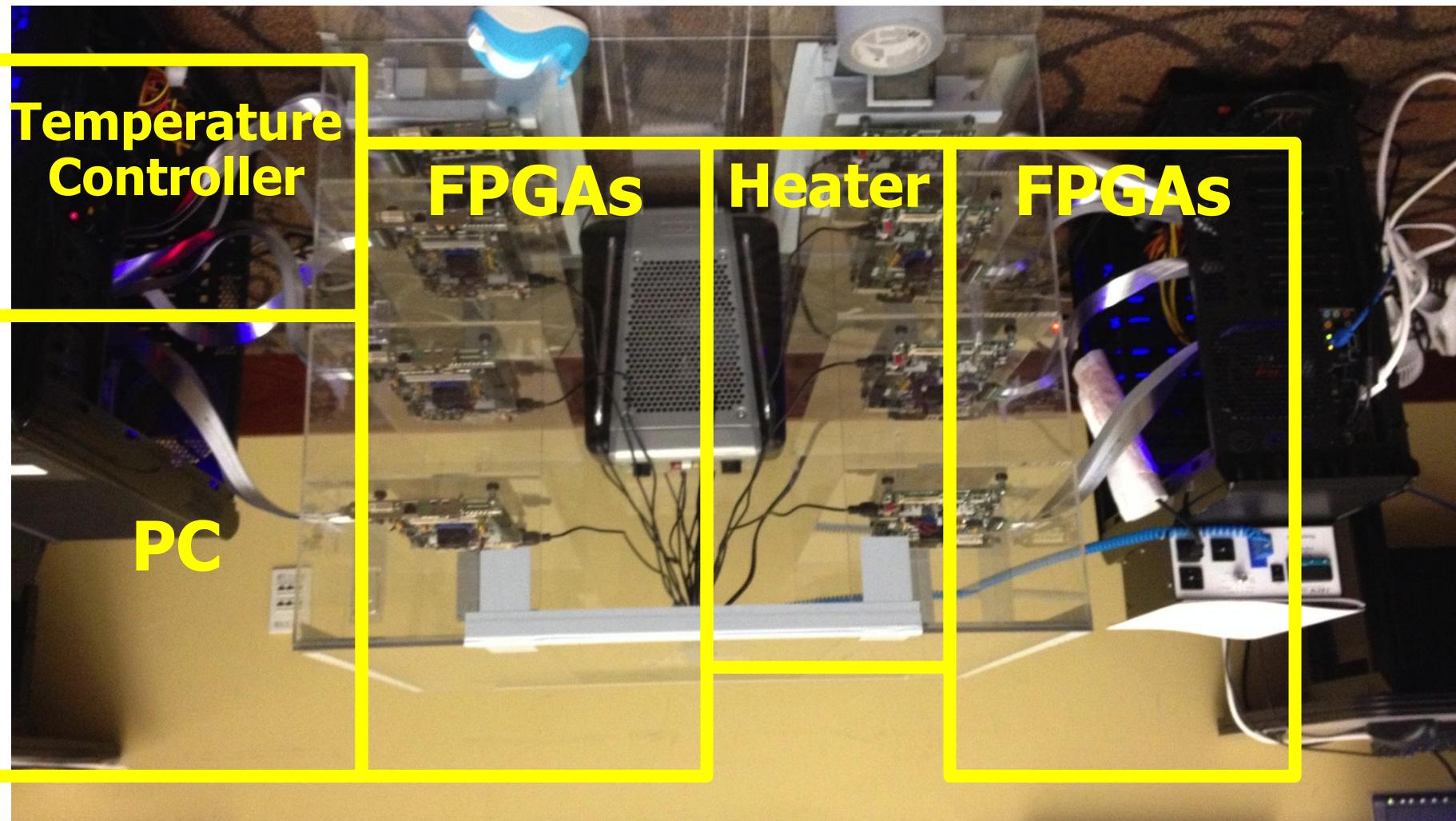
[AVATAR: A Variable-Retention-Time \(VRT\) Aware Refresh for DRAM Systems](#) (Qureshi et al., DSN 2015)

[An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms](#) (Liu et al., ISCA 2013)

[The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study](#) (Khan et al., SIGMETRICS 2014)

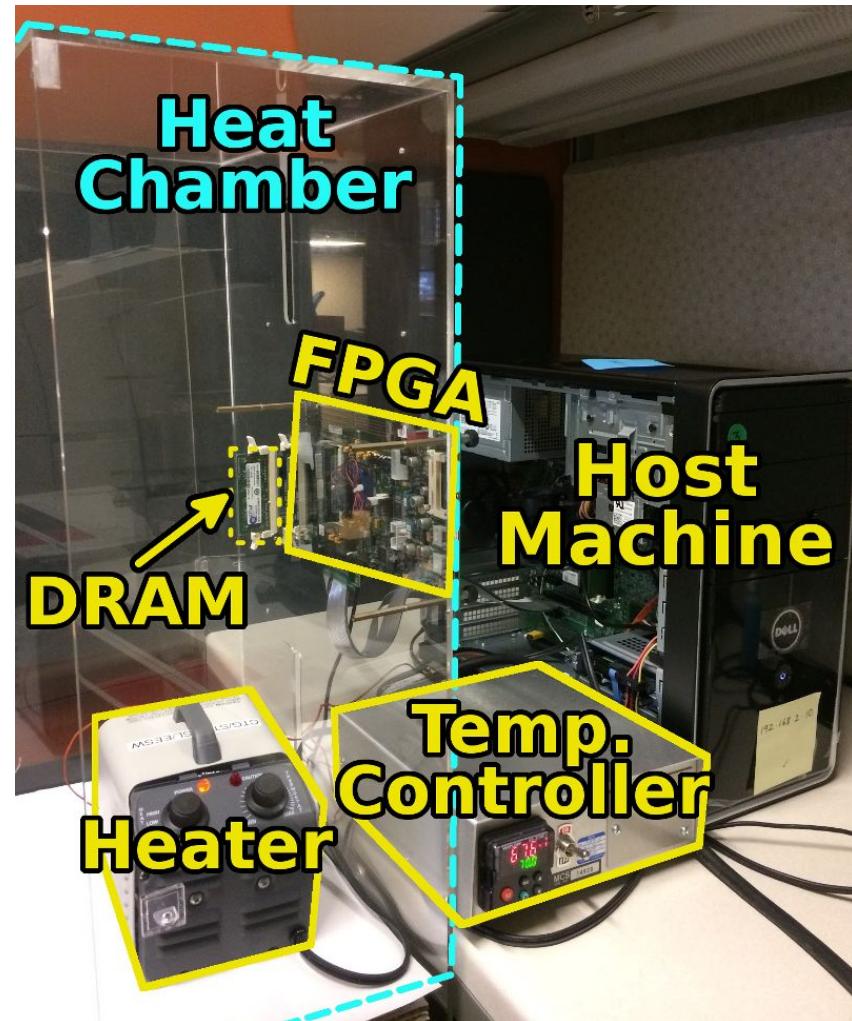


# Infrastructures to Understand Such Issues



# SoftMC: Open Source DRAM Infrastructure

- Hasan Hassan et al., "[SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies](#)," HPCA 2017.
- **Flexible**
- **Easy to Use (C++ API)**
- **Open-source**  
*[github.com/CMU-SAFARI/SoftMC](https://github.com/CMU-SAFARI/SoftMC)*



- <https://github.com/CMU-SAFARI/SoftMC>

## SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies

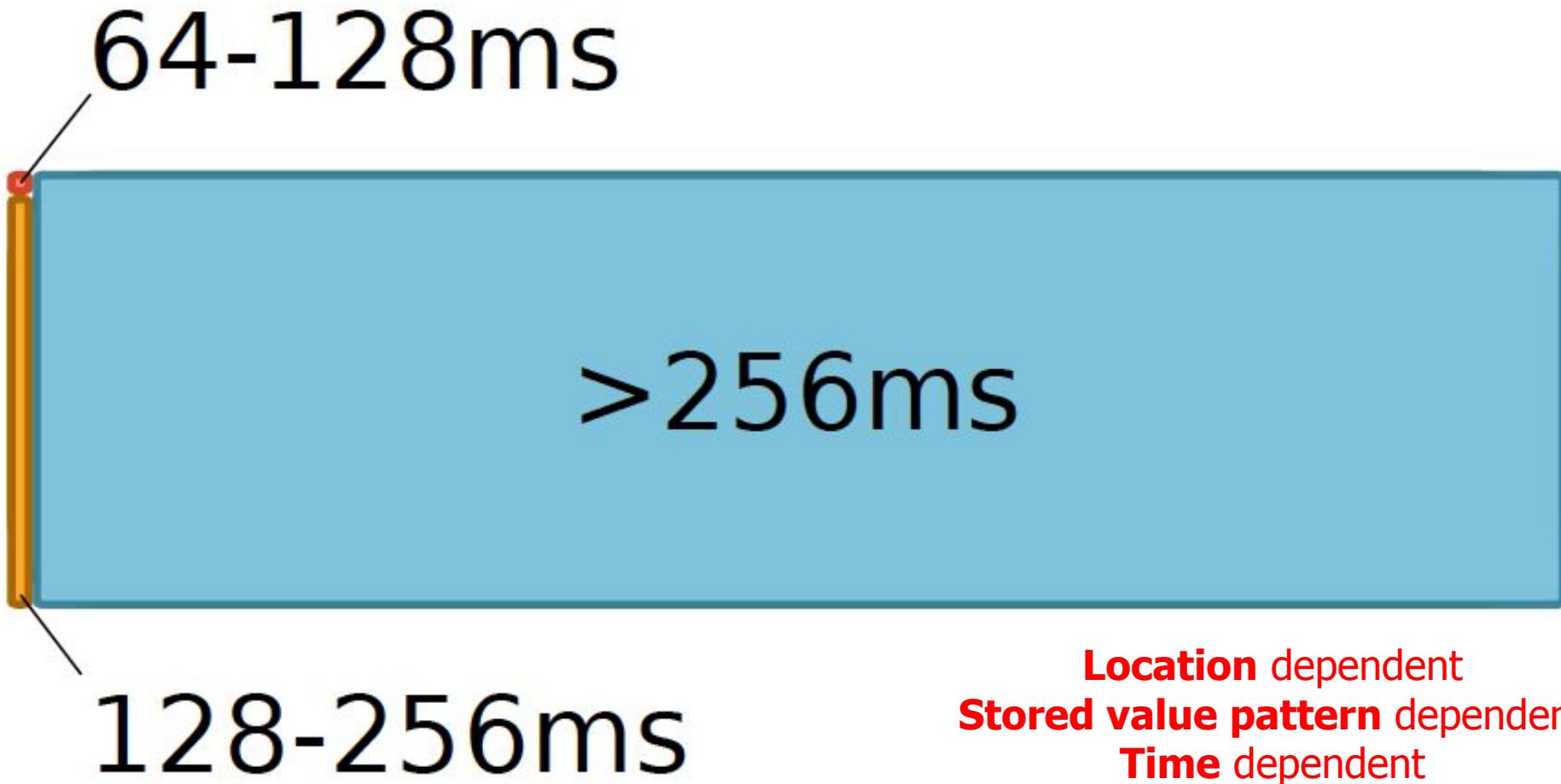
Hasan Hassan<sup>1,2,3</sup> Nandita Vijaykumar<sup>3</sup> Samira Khan<sup>4,3</sup> Saugata Ghose<sup>3</sup> Kevin Chang<sup>3</sup>  
Gennady Pekhimenko<sup>5,3</sup> Donghyuk Lee<sup>6,3</sup> Oguz Ergin<sup>2</sup> Onur Mutlu<sup>1,3</sup>

<sup>1</sup>*ETH Zürich*    <sup>2</sup>*TOBB University of Economics & Technology*    <sup>3</sup>*Carnegie Mellon University*  
<sup>4</sup>*University of Virginia*    <sup>5</sup>*Microsoft Research*    <sup>6</sup>*NVIDIA Research*

# Data Retention in Memory [Liu et al., ISCA 2013]

---

- Retention Time Profile of DRAM looks like this:



# RAIDR: Heterogeneous Refresh [ISCA'12]

---

- Jamie Liu, Ben Jaiyen, Richard Veras, and Onur Mutlu,  
**"RAIDR: Retention-Aware Intelligent DRAM Refresh"**  
*Proceedings of the 39th International Symposium on Computer Architecture (ISCA)*, Portland, OR, June 2012.  
Slides (pdf)

## RAIDR: Retention-Aware Intelligent DRAM Refresh

Jamie Liu    Ben Jaiyen    Richard Veras    Onur Mutlu  
Carnegie Mellon University

---

# Analysis of Data Retention Failures [ISCA'13]

---

- Jamie Liu, Ben Jaiyen, Yoongu Kim, Chris Wilkerson, and Onur Mutlu,  
**"An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms"**

*Proceedings of the 40th International Symposium on Computer Architecture (ISCA)*, Tel-Aviv, Israel, June 2013. [Slides \(ppt\)](#) [Slides \(pdf\)](#)

## An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms

Jamie Liu\*

Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213

jamiel@alumni.cmu.edu

Ben Jaiyen\*

Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213

bjaiyen@alumni.cmu.edu

Yoongu Kim

Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213

yoonguk@ece.cmu.edu

Chris Wilkerson

Intel Corporation  
2200 Mission College Blvd.  
Santa Clara, CA 95054  
chris.wilkerson@intel.com

Onur Mutlu

Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213  
onur@cmu.edu

# Mitigation of Retention Issues [SIGMETRICS'14]

- Samira Khan, Donghyuk Lee, Yoongu Kim, Alaa Alameldeen, Chris Wilkerson, and Onur Mutlu,

## "The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study"

*Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems* (SIGMETRICS), Austin, TX, June 2014. [[Slides \(pptx\)](#) ([pdf](#))] [[Poster \(pptx\)](#) ([pdf](#))] [[Full data sets](#)]

## The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study

Samira Khan<sup>†\*</sup>  
samirakhan@cmu.edu

Donghyuk Lee<sup>†</sup>  
donghyuk1@cmu.edu

Yoongu Kim<sup>†</sup>  
yoongukim@cmu.edu

Alaa R. Alameldeen<sup>\*</sup>  
alaa.r.alameldeen@intel.com

Chris Wilkerson<sup>\*</sup>  
chris.wilkerson@intel.com

Onur Mutlu<sup>†</sup>  
onur@cmu.edu

<sup>†</sup>Carnegie Mellon University

<sup>\*</sup>Intel Labs

# Handling Variable Retention Time [DSN'15]

---

- Moinuddin Qureshi, Dae Hyun Kim, Samira Khan, Prashant Nair, and Onur Mutlu,  
**"AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems"**

*Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Rio de Janeiro, Brazil, June 2015.*  
[Slides (pptx) (pdf)]

## AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems

Moinuddin K. Qureshi<sup>†</sup>      Dae-Hyun Kim<sup>†</sup>      Samira Khan<sup>‡</sup>      Prashant J. Nair<sup>†</sup>      Onur Mutlu<sup>‡</sup>  
<sup>†</sup>Georgia Institute of Technology      <sup>‡</sup>Carnegie Mellon University  
*{moin, dhkim, pnair6}@ece.gatech.edu*      *{samirakhan, onur}@cmu.edu*

# Handling Data-Dependent Failures [DSN'16]

---

- Samira Khan, Donghyuk Lee, and Onur Mutlu,  
**"PARBOR: An Efficient System-Level Technique to Detect Data-Dependent Failures in DRAM"**

*Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Toulouse, France, June 2016.*  
[Slides (pptx) (pdf)]

## PARBOR: An Efficient System-Level Technique to Detect Data-Dependent Failures in DRAM

Samira Khan\*

\*University of Virginia

Donghyuk Lee<sup>†‡</sup>

<sup>†</sup>Carnegie Mellon University

Onur Mutlu\*<sup>†</sup>

<sup>‡</sup>Nvidia

\*ETH Zürich

# Handling Data-Dependent Failures [MICRO'17]

---

- Samira Khan, Chris Wilkerson, Zhe Wang, Alaa R. Alameldeen, Donghyuk Lee, and Onur Mutlu,

## **"Detecting and Mitigating Data-Dependent DRAM Failures by Exploiting Current Memory Content"**

*Proceedings of the 50th International Symposium on Microarchitecture (MICRO), Boston, MA, USA, October 2017.*

[[Slides \(pptx\)](#) ([pdf](#))] [[Lightning Session Slides \(pptx\)](#) ([pdf](#))] [[Poster \(pptx\)](#) ([pdf](#))]

## **Detecting and Mitigating Data-Dependent DRAM Failures by Exploiting Current Memory Content**

Samira Khan<sup>\*</sup> Chris Wilkerson<sup>†</sup> Zhe Wang<sup>†</sup> Alaa R. Alameldeen<sup>†</sup> Donghyuk Lee<sup>‡</sup> Onur Mutlu<sup>\*</sup>

<sup>\*</sup>University of Virginia

<sup>†</sup>Intel Labs

<sup>‡</sup>Nvidia Research

<sup>\*</sup>ETH Zürich

# Handling Both DPD and VRT [ISCA'17]

---

- Minesh Patel, Jeremie S. Kim, and Onur Mutlu,  
**"The Reach Profiler (REAPER): Enabling the Mitigation of DRAM Retention Failures via Profiling at Aggressive Conditions"**

*Proceedings of the 44th International Symposium on Computer Architecture (ISCA)*, Toronto, Canada, June 2017.

[[Slides \(pptx\)](#) [\(pdf\)](#)]

[[Lightning Session Slides \(pptx\)](#) [\(pdf\)](#)]

- First experimental analysis of (mobile) LPDDR4 chips
- Analyzes the complex tradeoff space of retention time profiling
- Idea: enable fast and robust profiling at higher refresh intervals & temperatures

## The Reach Profiler (REAPER): Enabling the Mitigation of DRAM Retention Failures via Profiling at Aggressive Conditions

Minesh Patel<sup>§‡</sup>   Jeremie S. Kim<sup>‡§</sup>   Onur Mutlu<sup>§‡</sup>  
<sup>§</sup>ETH Zürich   <sup>‡</sup>Carnegie Mellon University

# In-DRAM ECC Complicates Things [DSN'19]

---

- Minesh Patel, Jeremie S. Kim, Hasan Hassan, and Onur Mutlu,  
**"Understanding and Modeling On-Die Error Correction in Modern DRAM: An Experimental Study Using Real Devices"**

*Proceedings of the 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Portland, OR, USA, June 2019.*

[Slides (pptx) (pdf)]

[Talk Video (26 minutes)]

[Full Talk Lecture (29 minutes)]

[Source Code for EINSim, the Error Inference Simulator]

***Best paper award.***

## Understanding and Modeling On-Die Error Correction in Modern DRAM: An Experimental Study Using Real Devices

Minesh Patel<sup>†</sup> Jeremie S. Kim<sup>‡†</sup> Hasan Hassan<sup>†</sup> Onur Mutlu<sup>†‡</sup>

<sup>†</sup>*ETH Zürich*    <sup>‡</sup>*Carnegie Mellon University*

# More on In-DRAM ECC [MICRO'20]

---

- Minesh Patel, Jeremie S. Kim, Taha Shahroodi, Hasan Hassan, and Onur Mutlu,  
**"Bit-Exact ECC Recovery (BEER): Determining DRAM On-Die ECC Functions by Exploiting DRAM Data Retention Characteristics"**

*Proceedings of the 53rd International Symposium on Microarchitecture (MICRO)*, Virtual, October 2020.

[[Slides \(pptx\)](#) ([pdf](#))]

[[Short Talk Slides \(pptx\)](#) ([pdf](#))]

[[Lightning Talk Slides \(pptx\)](#) ([pdf](#))]

[[Lecture Slides \(pptx\)](#) ([pdf](#))]

[[Talk Video](#) (15 minutes)]

[[Short Talk Video](#) (5.5 minutes)]

[[Lightning Talk Video](#) (1.5 minutes)]

[[Lecture Video](#) (52.5 minutes)]

[[BEER Source Code](#)]

*Best paper award.*

## Bit-Exact ECC Recovery (BEER): Determining DRAM On-Die ECC Functions by Exploiting DRAM Data Retention Characteristics

Minesh Patel<sup>†</sup> Jeremie S. Kim<sup>†‡</sup> Taha Shahroodi<sup>†</sup> Hasan Hassan<sup>†</sup> Onur Mutlu<sup>†‡</sup>

<sup>†</sup>*ETH Zürich*    <sup>‡</sup>*Carnegie Mellon University*

# Profiling In The Presence of ECC [MICRO'21]

---

- To Appear in MICRO 2021

## **HARP: Practically and Effectively Identifying Uncorrectable Errors in Memory Chips That Use On-Die Error-Correcting Codes**

Minesh Patel  
ETH Zürich

Geraldo F. Oliveira  
ETH Zürich

Onur Mutlu  
ETH Zürich

# A Curious Discovery [Kim et al., ISCA 2014]

---

One can  
predictably induce errors  
in most DRAM memory chips

# DRAM RowHammer

A simple hardware failure mechanism  
can create a widespread  
system security vulnerability

WIRED

Forget Software—Now Hackers Are Exploiting Physics

BUSINESS

CULTURE

DESIGN

GEAR

SCIENCE

ANDY GREENBERG SECURITY 08.31.16 7:00 AM

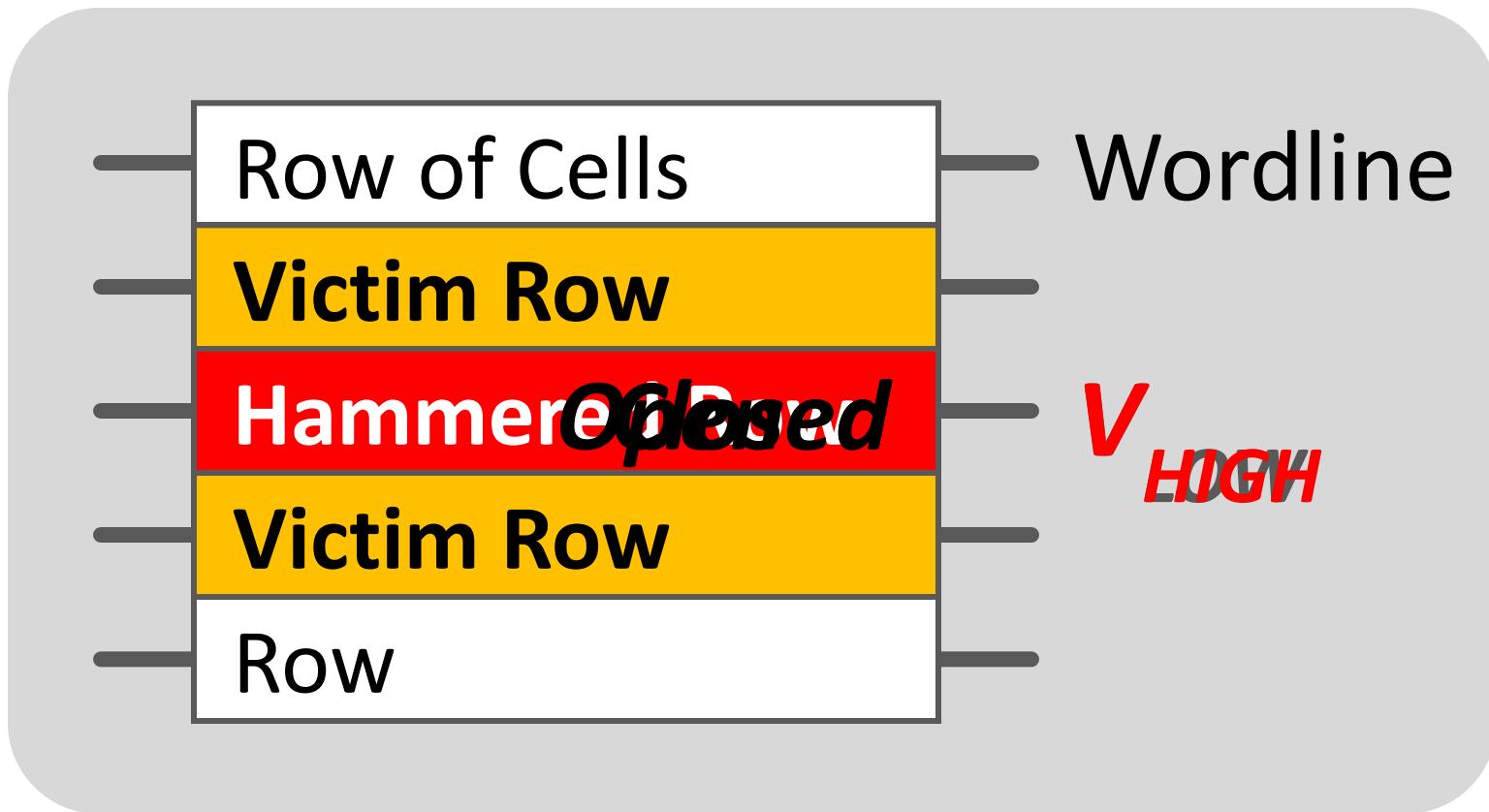
SHARE

 SHARE  
18276

 TWEET

# FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

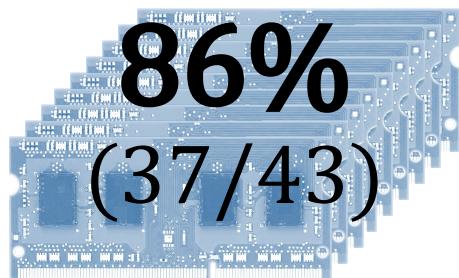
# Modern DRAM is Prone to Disturbance Errors



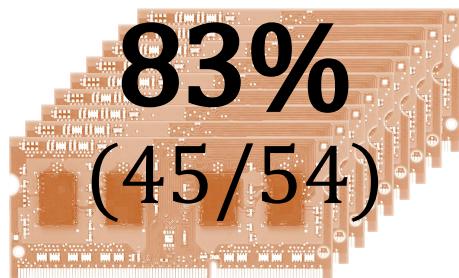
**Repeatedly reading** a row enough times (before memory gets refreshed) induces **disturbance errors** in adjacent rows in **most real DRAM chips you can buy today**

# Most DRAM Modules Are Vulnerable

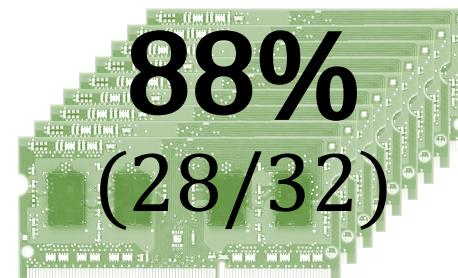
A company



B company



C company

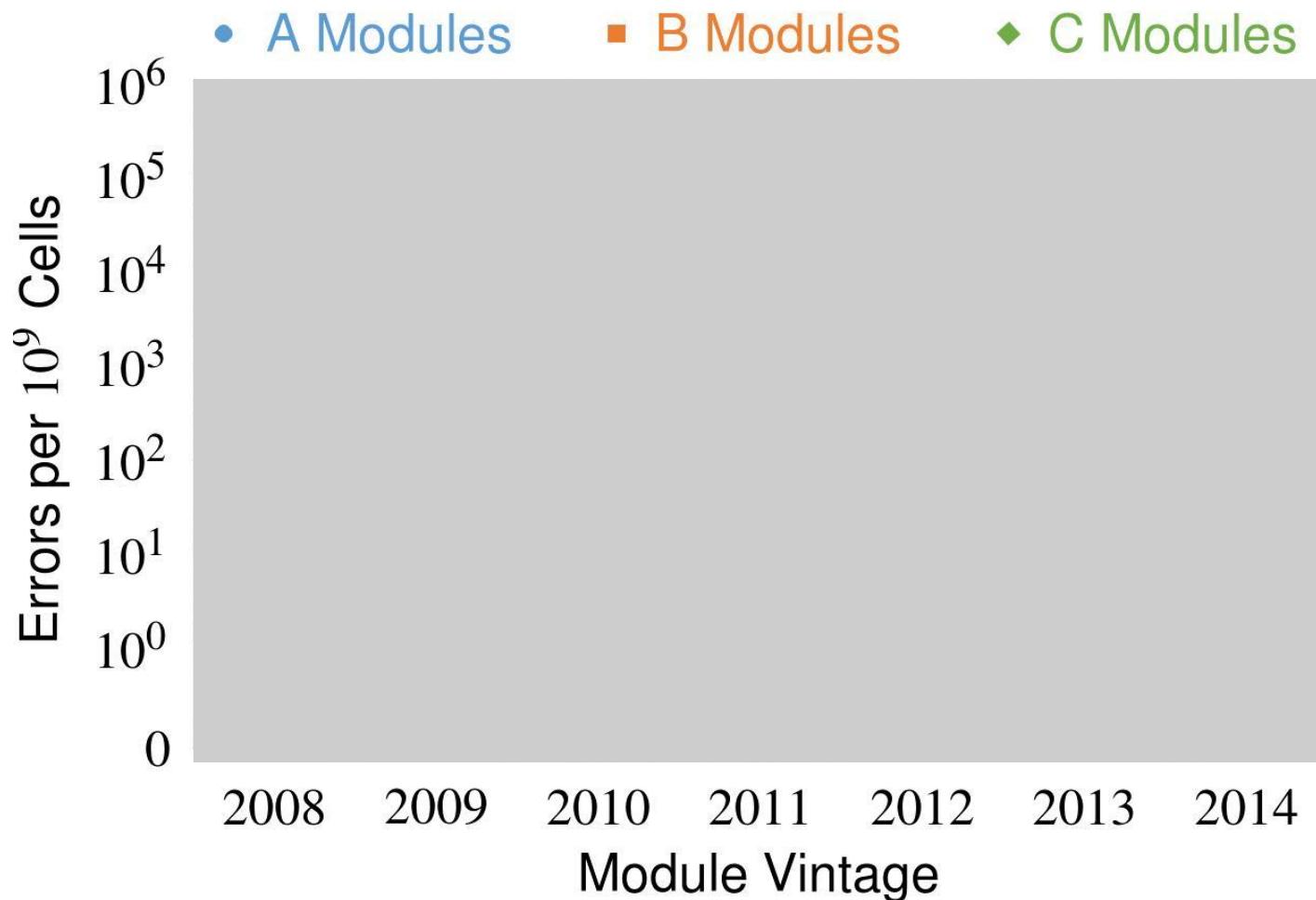


Up to  
 $1.0 \times 10^7$   
errors

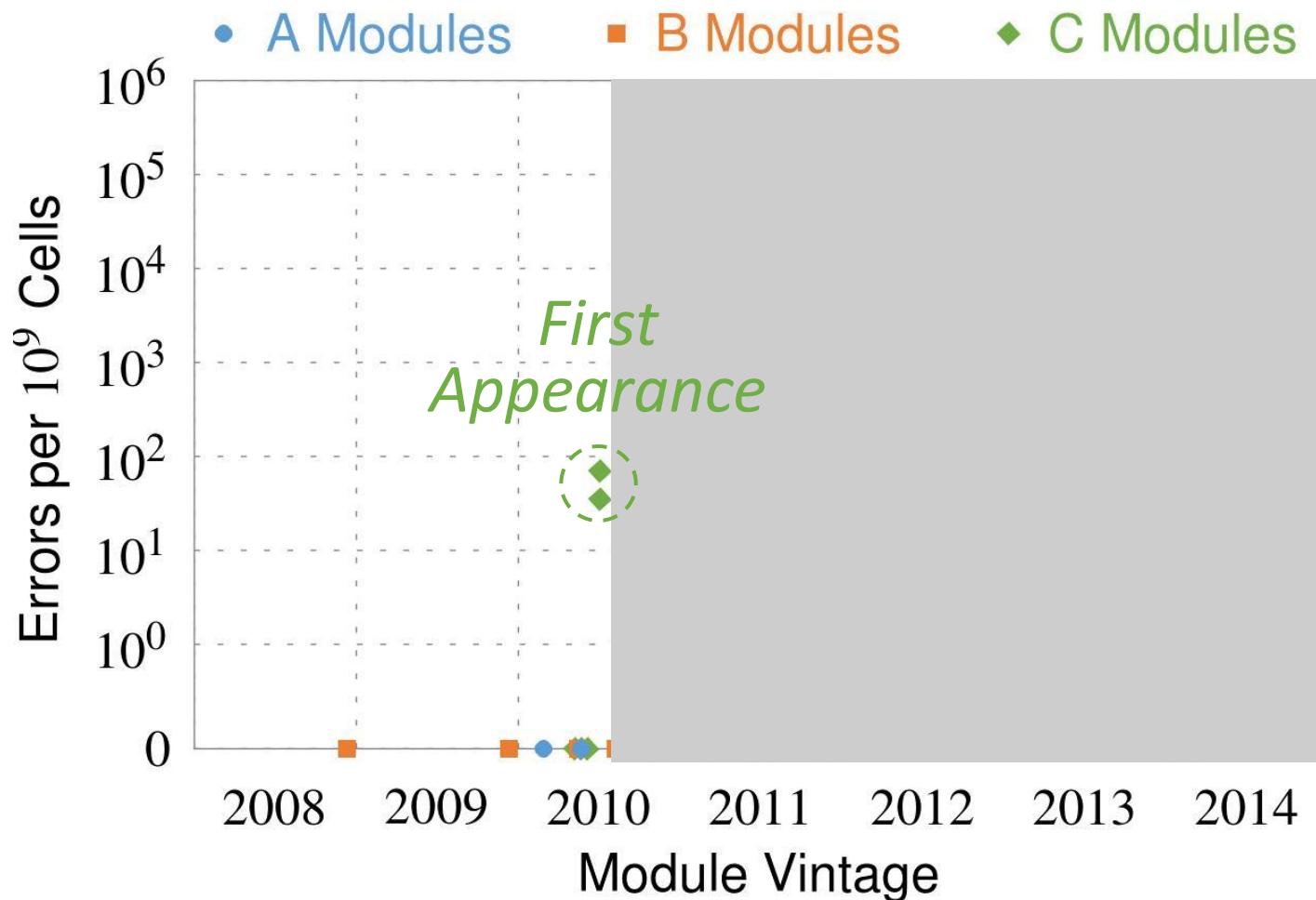
Up to  
 $2.7 \times 10^6$   
errors

Up to  
 $3.3 \times 10^5$   
errors

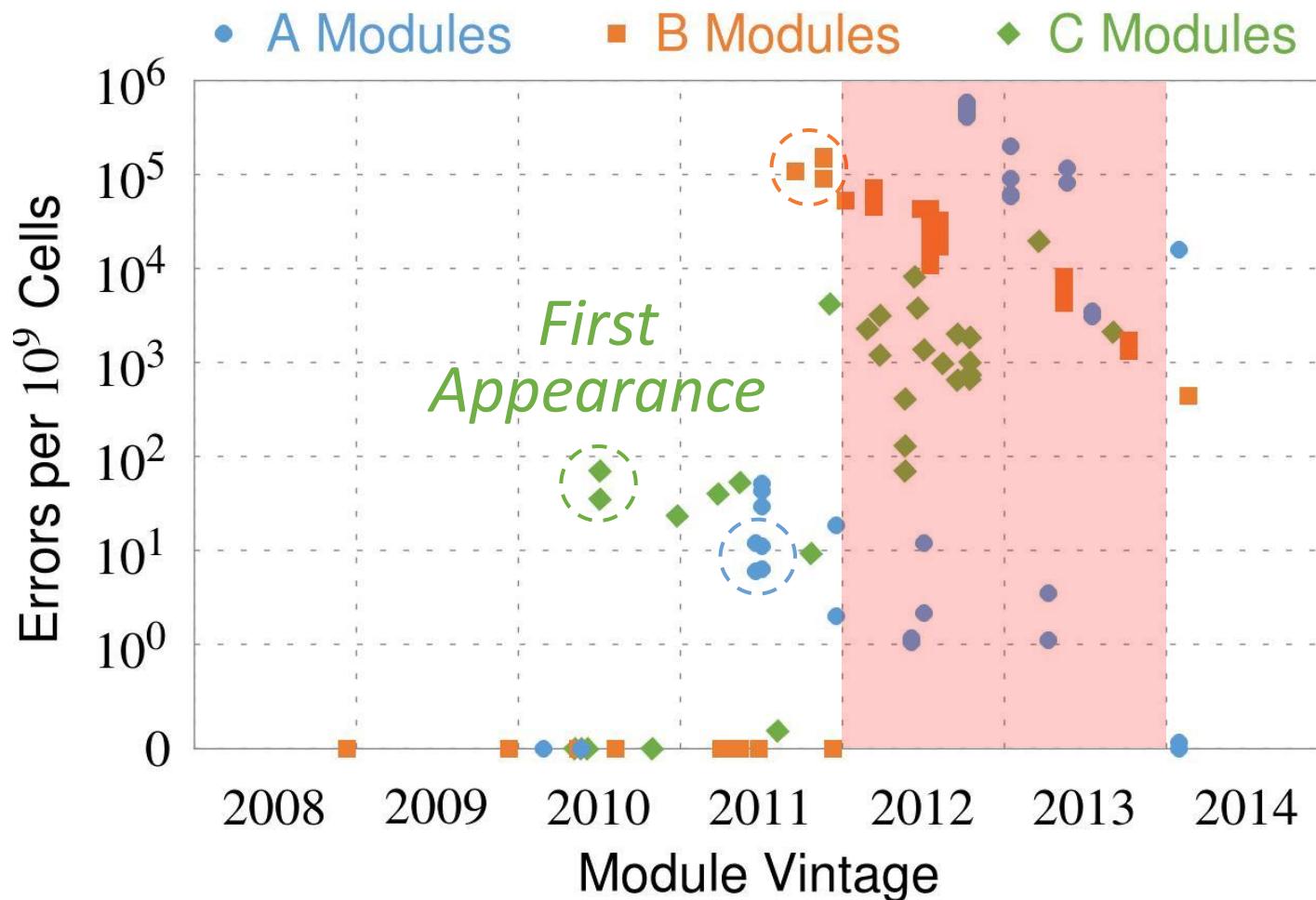
# Recent DRAM Is More Vulnerable



# Recent DRAM Is More Vulnerable



# Recent DRAM Is More Vulnerable



*All modules from 2012–2013 are vulnerable*

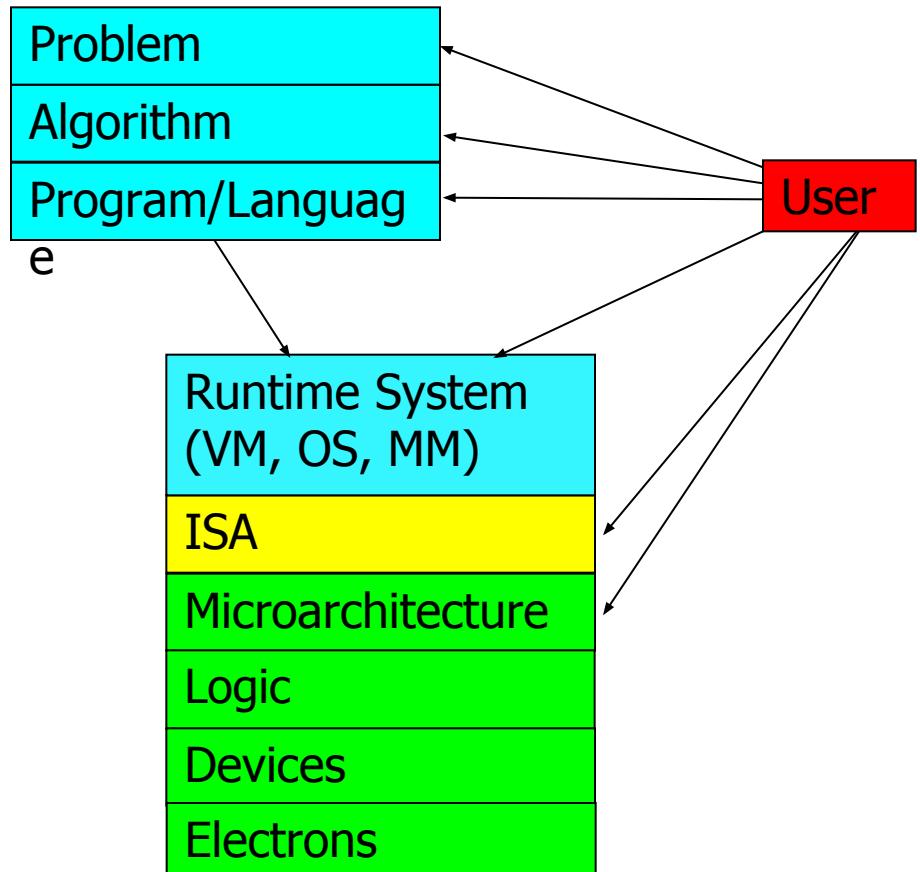
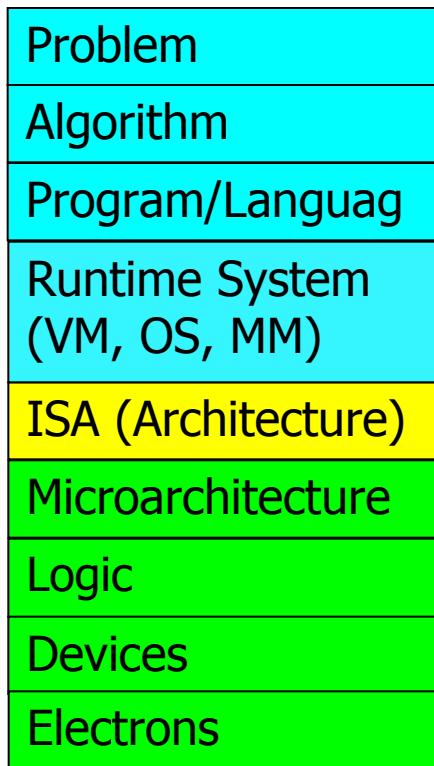
# Why Is This Happening?

---

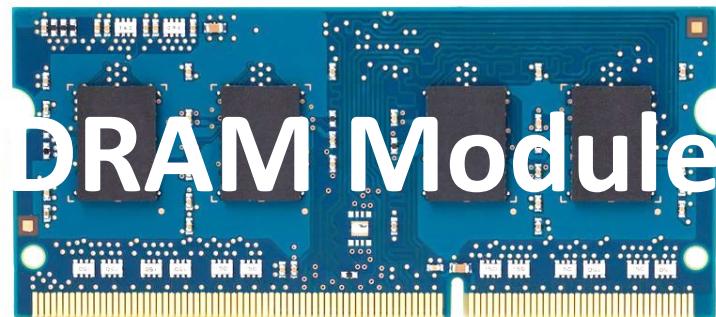
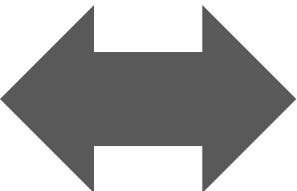
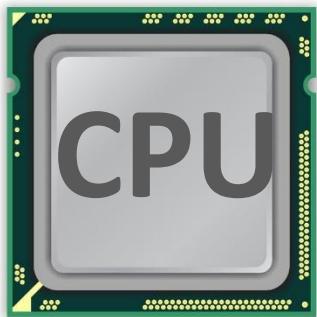
- DRAM cells are too close to each other!
  - They are not electrically isolated from each other
- Access to one cell affects the value in nearby cells
  - due to **electrical interference** between
    - the cells
    - wires used for accessing the cells
  - Also called cell-to-cell coupling/interference
- Example: When we activate (apply high voltage) to a row, an adjacent row gets slightly activated as well
  - Vulnerable cells in that slightly-activated row lose a little bit of charge
  - If row hammer happens enough times, charge in such cells gets drained

# Higher-Level Implications

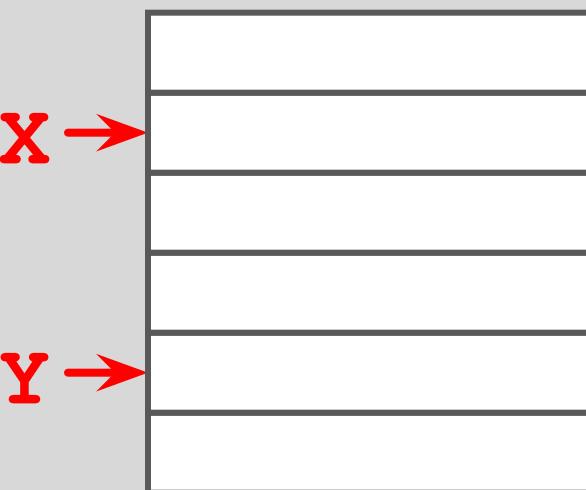
- This simple circuit level failure mechanism has enormous implications on upper layers of the transformation hierarchy



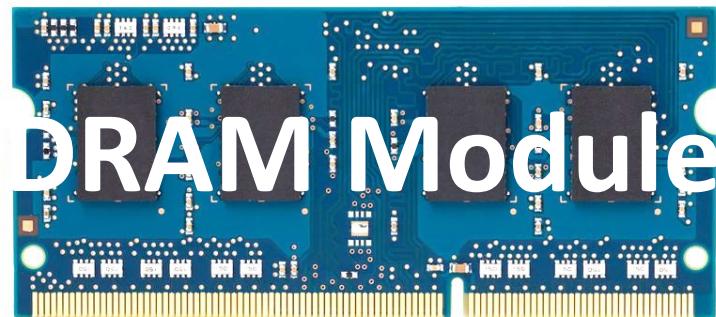
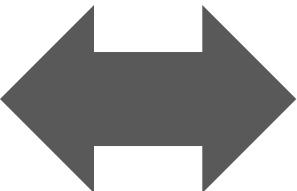
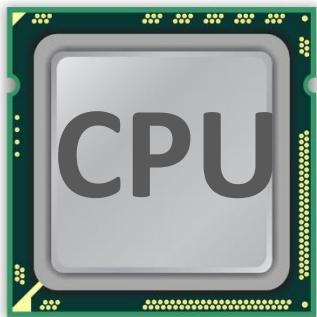
# A Simple Program Can Induce Many Errors



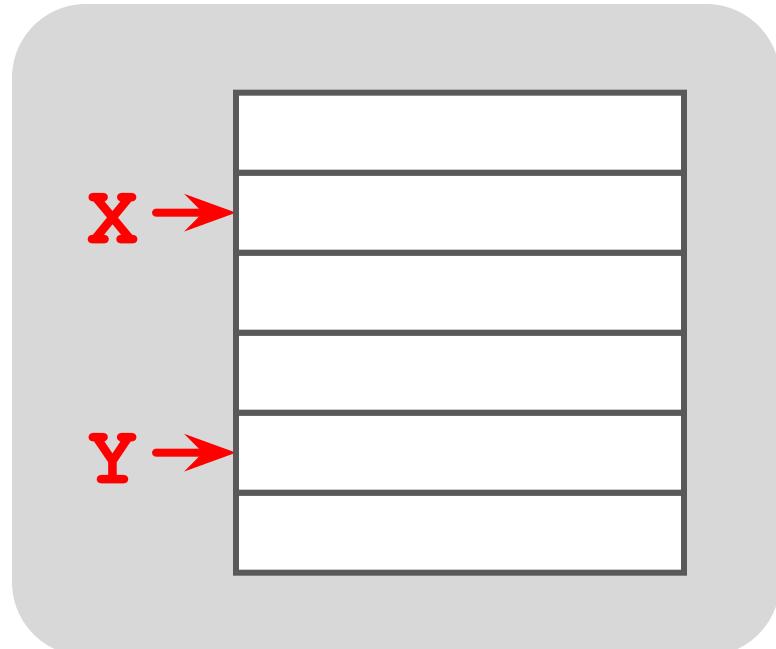
```
loop:  
    mov (%X), %eax  
    mov (%Y), %ebx  
    clflush (%X)  
    clflush (%Y)  
    mfence  
    jmp loop
```



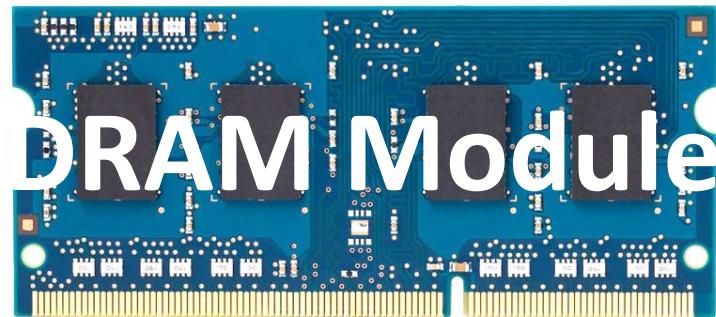
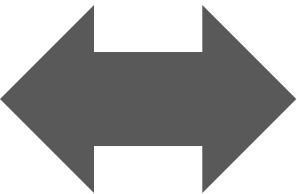
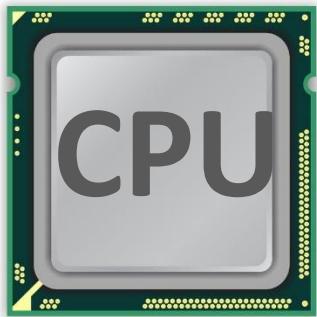
# A Simple Program Can Induce Many Errors



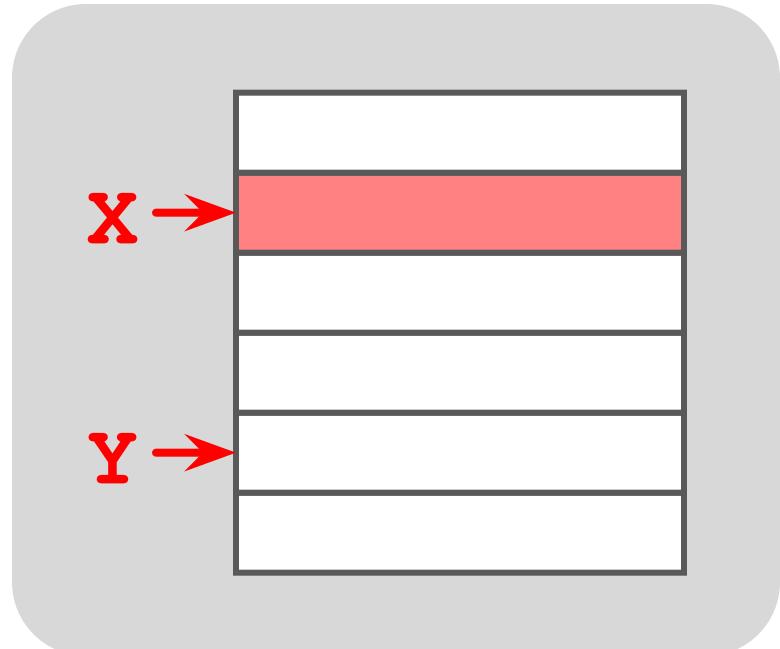
1. Avoid ***cache hits***
  - Flush **X** from cache
2. Avoid ***row hits*** to **X**
  - Read **Y** in another row



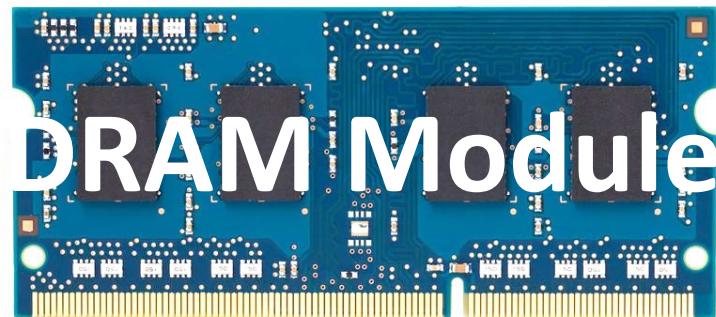
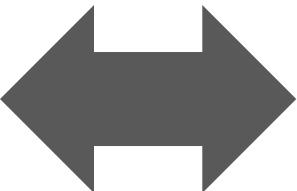
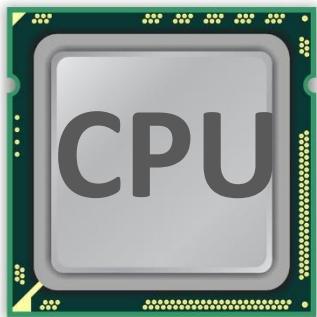
# A Simple Program Can Induce Many Errors



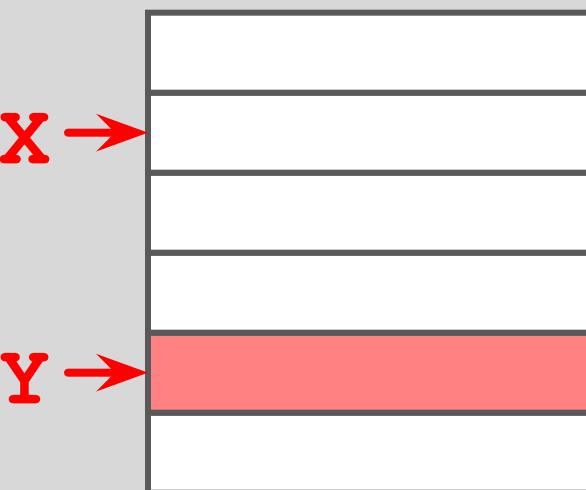
```
loop:  
    mov (%X), %eax  
    mov (%Y), %ebx  
    clflush (%X)  
    clflush (%Y)  
    mfence  
    jmp loop
```



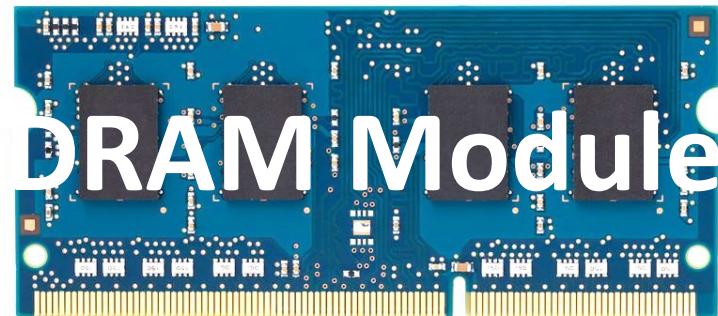
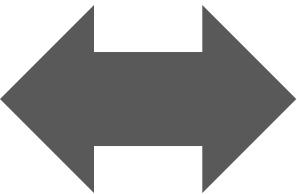
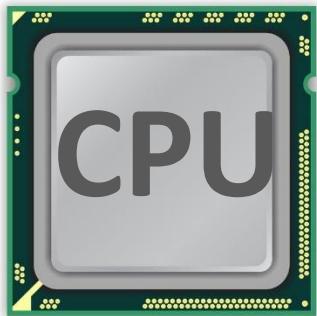
# A Simple Program Can Induce Many Errors



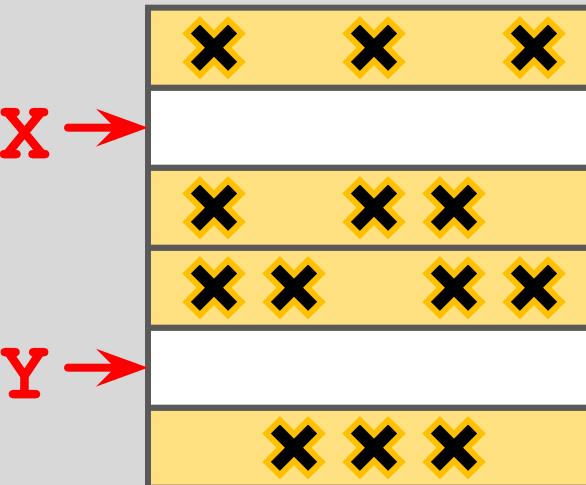
```
loop:  
    mov (%X), %eax  
    mov (%Y), %ebx  
    clflush (%X)  
    clflush (%Y)  
    mfence  
    jmp loop
```



# A Simple Program Can Induce Many Errors



```
loop:  
    mov (%X), %eax  
    mov (%Y), %ebx  
    clflush (%X)  
    clflush (%Y)  
    mfence  
    jmp loop
```



# Observed Errors in Real Systems

CPU Architecture	Errors	Access-Rate
Intel Haswell (2013)	22.9K	12.3M/sec
Intel Ivy Bridge (2012)	20.7K	11.7M/sec
Intel Sandy Bridge (2011)	16.1K	11.6M/sec
AMD Piledriver (2012)	59	6.1M/sec

A real reliability & security issue

# One Can Take Over an Otherwise-Secure System

---

## Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors

*Abstract. Memory isolation is a key property of a reliable and secure computing system — an access to one memory address should not have unintended side effects on data stored in other addresses. However, as DRAM process technology*

# Project Zero

[Flipping Bits in Memory Without Accessing Them:  
An Experimental Study of DRAM Disturbance Errors](#)  
(Kim et al., ISCA 2014)

News and updates from the Project Zero team at Google

[Exploiting the DRAM rowhammer bug to  
gain kernel privileges](#) (Seaborn, 2015)

Monday, March 9, 2015

Exploiting the DRAM rowhammer bug to gain kernel privileges

# RowHammer Security Attack Example

---

- “Rowhammer” is a problem with some recent DRAM devices in which repeatedly accessing a row of memory can cause bit flips in adjacent rows (Kim et al., ISCA 2014).
  - [Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors](#) (Kim et al., ISCA 2014)
- We tested a selection of laptops and found that a subset of them exhibited the problem.
- We built two working privilege escalation exploits that use this effect.
  - [Exploiting the DRAM rowhammer bug to gain kernel privileges](#) (Seaborn+, 2015)
- One exploit uses rowhammer-induced bit flips to gain kernel privileges on x86-64 Linux when run as an unprivileged userland process.
- When run on a machine vulnerable to the rowhammer problem, the process was able to induce bit flips in page table entries (PTEs).
- It was able to use this to gain write access to its own page table, and hence gain read-write access to all of physical memory.

# Security Implications

Rowhammer



# Security Implications



It's like breaking into an apartment by repeatedly slamming a neighbor's door until the vibrations open the door you were after

# Before RowHammer (I)

---

## Using Memory Errors to Attack a Virtual Machine

Sudhakar Govindavajhala \*

Andrew W. Appel

Princeton University

{sudhakar,appel}@cs.princeton.edu

*We present an experimental study showing that soft memory errors can lead to serious security vulnerabilities in Java and .NET virtual machines, or in any system that relies on type-checking of untrusted programs as a protection mechanism. Our attack works by sending to the JVM a Java program that is designed so that almost any memory error in its address space will allow it to take control of the JVM. All conventional Java and .NET virtual machines are vulnerable to this attack. The technique of the attack is broadly applicable against other language-based security schemes such as proof-carrying code.*

*We measured the attack on two commercial Java Virtual Machines: Sun's and IBM's. We show that a single-bit error in the Java program's data space can be exploited to execute arbitrary code with a probability of about 70%, and multiple-bit errors with a lower probability.*

*Our attack is particularly relevant against smart cards or tamper-resistant computers, where the user has physical access (to the outside of the computer) and can use various means to induce faults; we have successfully used heat. Fortunately, there are some straightforward defenses against this attack.*

## 7 Physical fault injection

If the attacker has physical access to the outside of the machine, as in the case of a smart card or other tamper-resistant computer, the attacker can induce memory errors. We considered attacks on boxes in form factors ranging from a credit card to a palmtop to a desktop PC.

We considered several ways in which the attacker could induce errors.<sup>4</sup>

# Before RowHammer (II)

---

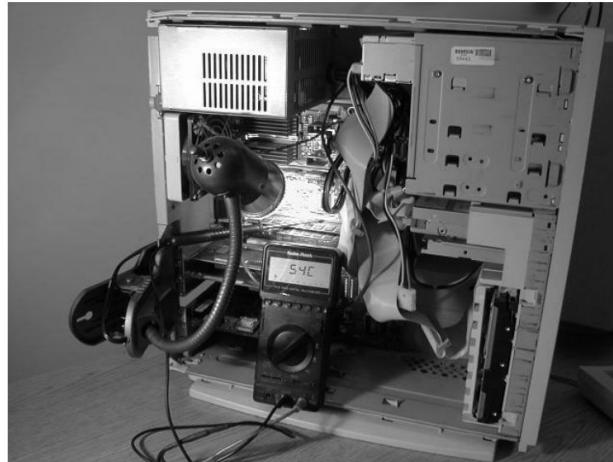
## Using Memory Errors to Attack a Virtual Machine

Sudhakar Govindavajhala \*

Andrew W. Appel

Princeton University

{sudhakar,appel}@cs.princeton.edu



**Figure 3.** Experimental setup to induce memory errors, showing a PC built from surplus components, clip-on gooseneck lamp, 50-watt spotlight bulb, and digital thermometer. Not shown is the variable AC power supply for the lamp.

# Selected Readings on RowHammer (I)

---

- Our first detailed study: [Rowhammer analysis and solutions](#) (June 2014)
  - Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu,  
**"Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors"**  
*Proceedings of the 41st International Symposium on Computer Architecture (ISCA)*, Minneapolis, MN, June 2014. [[Slides \(pptx\)](#) [\(pdf\)](#)] [[Lightning Session Slides \(pptx\)](#) [\(pdf\)](#)] [[Source Code and Data](#)]
- Our Source Code to Induce Errors in Modern DRAM Chips (June 2014)
  - <https://github.com/CMU-SAFARI/rowhammer>
- Google Project Zero's Attack to Take Over a System (March 2015)
  - [Exploiting the DRAM rowhammer bug to gain kernel privileges](#) (Seaborn+, 2015)
  - <https://github.com/google/rowhammer-test>
  - **Double-sided Rowhammer**

# Selected Readings on RowHammer (II)

---

- Remote RowHammer Attacks via JavaScript (July 2015)
  - <http://arxiv.org/abs/1507.06955>
  - <https://github.com/IAIK/rowhammerjs>
  - Gruss et al., DIMVA 2016.
  - **CLFLUSH-free Rowhammer**
  - “A fully automated attack that requires nothing but a website with JavaScript to **trigger faults on remote hardware.**”
  - “We can gain unrestricted access to systems of website visitors.”
- ANVIL: Software-Based Protection Against Next-Generation Rowhammer Attacks (March 2016)
  - <http://dl.acm.org/citation.cfm?doid=2872362.2872390>
  - Aweke et al., ASPLOS 2016
  - **CLFLUSH-free Rowhammer**
  - Software based monitoring for rowhammer detection

# Selected Readings on RowHammer (III)

---

- Dedup Est Machina: Memory Deduplication as an Advanced Exploitation Vector (May 2016)
  - <https://www.ieee-security.org/TC/SP2016/papers/0824a987.pdf>
  - Bosman et al., IEEE S&P 2016.
  - Exploits Rowhammer and Memory Deduplication to overtake a browser
  - “We report on the **first reliable remote exploit for the Rowhammer vulnerability** running entirely in Microsoft Edge.”
  - “[an attacker] ... can reliably “own” a system with all defenses up, even if the software is entirely free of bugs.”
- CAn't Touch This: Software-only Mitigation against Rowhammer Attacks targeting Kernel Memory (August 2017)
  - <https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-brasser.pdf>
  - Brasser et al., USENIX Security 2017.
  - Partitions physical memory into security domains, user vs. kernel; limits rowhammer-induced bit flips to the user domain.

# Selected Readings on RowHammer (IV)

---

- A New Approach for Rowhammer Attacks (May 2016)
  - <https://ieeexplore.ieee.org/document/7495576>
  - Qiao et al., HOST 2016
  - **CLFLUSH-free RowHammer**
  - “Libc functions memset and memcpy are found capable of rowhammer.”
  - Triggers RowHammer with malicious inputs but benign code
- One Bit Flips, One Cloud Flops: Cross-VM Row Hammer Attacks and Privilege Escalation (August 2016)
  - [https://www.usenix.org/system/files/conference/usenixsecurity16/sec16\\_paper\\_xiao.pdf](https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_xiao.pdf)
  - Xiao et al., USENIX Security 2016.
  - **“Technique that allows a malicious guest VM to have read and write accesses to arbitrary physical pages on a shared machine.”**
  - Graph-based algorithm to reverse engineer mapping of physical addresses in DRAM

# Selected Readings on RowHammer (V)

---

- Curious Case of RowHammer: Flipping Secret Exponent Bits using Timing Analysis (August 2016)
  - [https://link.springer.com/content/pdf/10.1007%2F978-3-662-53140-2\\_29.pdf](https://link.springer.com/content/pdf/10.1007%2F978-3-662-53140-2_29.pdf)
  - Bhattacharya et al., CHES 2016
  - Combines timing analysis to perform **rowhammer on cryptographic keys** stored in memory
- DRAMA: Exploiting DRAM Addressing for Cross-CPU Attacks (August 2016)
  - [https://www.usenix.org/system/files/conference/usenixsecurity16/sec16\\_paper\\_pessl.pdf](https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_pessl.pdf)
  - Pessl et al., USENIX Security 2016
  - **Shows RowHammer failures on DDR4 devices despite TRR solution**
  - Reverse engineers address mapping functions to improve existing RowHammer attacks

# Selected Readings on RowHammer (VI)

---

- **Flip Feng Shui: Hammering a Needle in the Software Stack** (August 2016)
  - [https://www.usenix.org/system/files/conference/usenixsecurity16/sec16\\_paper\\_razavi.pdf](https://www.usenix.org/system/files/conference/usenixsecurity16/sec16_paper_razavi.pdf)
  - Razavi et al., USENIX Security 2016.
  - Combines memory deduplication and RowHammer
  - **“A malicious VM can gain unauthorized access to a co-hosted VM running OpenSSH.”**
  - Breaks OpenSSH public key authentication
- **Drammer: Deterministic Rowhammer Attacks on Mobile Platforms** (October 2016)
  - <http://dl.acm.org/citation.cfm?id=2976749.2978406>
  - Van Der Veen et al., ACM CCS 2016
  - **Can take over an ARM-based Android system deterministically**
  - Exploits predictable physical memory allocator behavior
    - Can deterministically place security-sensitive data (e.g., page table) in an attacker-chosen, vulnerable location in memory

# Selected Readings on RowHammer (VII)

---

- When Good Protections go Bad: Exploiting anti-DoS Measures to Accelerate Rowhammer Attacks (May 2017)
  - <https://web.eecs.umich.edu/~misiker/resources/HOST-2017-Misiker.pdf>
  - Aga et al., HOST 2017
  - “A virtual-memory based cache-flush free attack that is sufficiently fast to **rowhammer with double rate refresh.**”
  - Enabled by Cache Allocation Technology
- SGX-Bomb: Locking Down the Processor via Rowhammer Attack (October 2017)
  - <https://dl.acm.org/citation.cfm?id=3152709>
  - Jang et al., SysTEX 2017
  - “Launches the Rowhammer attack against enclave memory to trigger the processor lockdown.”
  - **Running unknown enclave programs on the cloud can shut down servers shared with other clients.**

# Selected Readings on RowHammer (VIII)

---

- [Another Flip in the Wall of Rowhammer Defenses](#) (May 2018)
  - <https://arxiv.org/pdf/1710.00551.pdf>
  - Gruss et al., IEEE S&P 2018
  - **A new type of Rowhammer attack which only hammers one single address**, which can be done without knowledge of physical addresses and DRAM mappings
  - Defeats static analysis and performance counter analysis defenses by running inside an SGX enclave
- [GuardION: Practical Mitigation of DMA-Based Rowhammer Attacks on ARM](#) (June 2018)
  - [https://link.springer.com/chapter/10.1007/978-3-319-93411-2\\_5](https://link.springer.com/chapter/10.1007/978-3-319-93411-2_5)
  - Van Der Veen et al., DIMVA 2018
  - Presents RAMPAGE, a DMA-based RowHammer attack against the latest Android OS

# Selected Readings on RowHammer (IX)

---

- Grand Pwning Unit: Accelerating Microarchitectural Attacks with the GPU (May 2018)
  - <https://www.vusec.net/wp-content/uploads/2018/05/glitch.pdf>
  - Frigo et al., IEEE S&P 2018.
  - The first end-to-end remote Rowhammer exploit on mobile platforms that use our GPU-based primitives in orchestration to **compromise browsers on mobile devices in under two minutes.**
- Throwhammer: Rowhammer Attacks over the Network and Defenses (July 2018)
  - [https://www.cs.vu.nl/~herbertb/download/papers/throwhammer\\_atc18.pdf](https://www.cs.vu.nl/~herbertb/download/papers/throwhammer_atc18.pdf)
  - Tatar et al., USENIX ATC 2018.
  - “[We] show that **an attacker can trigger and exploit Rowhammer bit flips directly from a remote machine by only sending network packets.**”

# Selected Readings on RowHammer (X)

---

- **Nethammer: Inducing Rowhammer Faults through Network Requests** (July 2018)
  - <https://arxiv.org/pdf/1805.04956.pdf>
  - Lipp et al., arxiv.org 2018.
  - “Nethammer is the first truly **remote Rowhammer attack**, without a single attacker-controlled line of code on the targeted system.”
- **ZebRAM: Comprehensive and Compatible Software Protection Against Rowhammer Attacks** (October 2018)
  - <https://www.usenix.org/system/files/osdi18-konoth.pdf>
  - Konoth et al., OSDI 2018
  - A new pure-software protection mechanism against RowHammer.

# Selected Readings on RowHammer (XI.A)

- PassMark Software, memtest86, since 2014
  - <https://www.memtest86.com/troubleshooting.htm#hammer>

## Why am I only getting errors during Test 13 Hammer Test?

The Hammer Test is designed to detect RAM modules that are susceptible to disturbance errors caused by charge leakage. This phenomenon is characterized in the research paper **Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors** by Yoongu Kim et al. According to the research, a significant number of RAM modules manufactured 2010 or newer are affected by this defect. In simple terms, susceptible RAM modules can be subjected to disturbance errors when repeatedly accessing addresses in the same memory bank but different rows in a short period of time. Errors occur when the repeated access causes charge loss in a memory cell, before the cell contents can be refreshed at the next DRAM refresh interval.

Starting from MemTest86 v6.2, the user may see a warning indicating that the RAM may be vulnerable to high frequency row hammer bit flips. This warning appears when errors are detected during the first pass (maximum hammer rate) but no errors are detected during the second pass (lower hammer rate). See **MemTest86 Test Algorithms** for a description of the two passes that are performed during the Hammer Test (Test 13). When performing the second pass, address pairs are hammered only at the rate deemed as the maximum allowable by memory vendors (200K accesses per 64ms). Once this rate is exceeded, the integrity of memory contents may no longer be guaranteed. If errors are detected in both passes, errors are reported as normal.

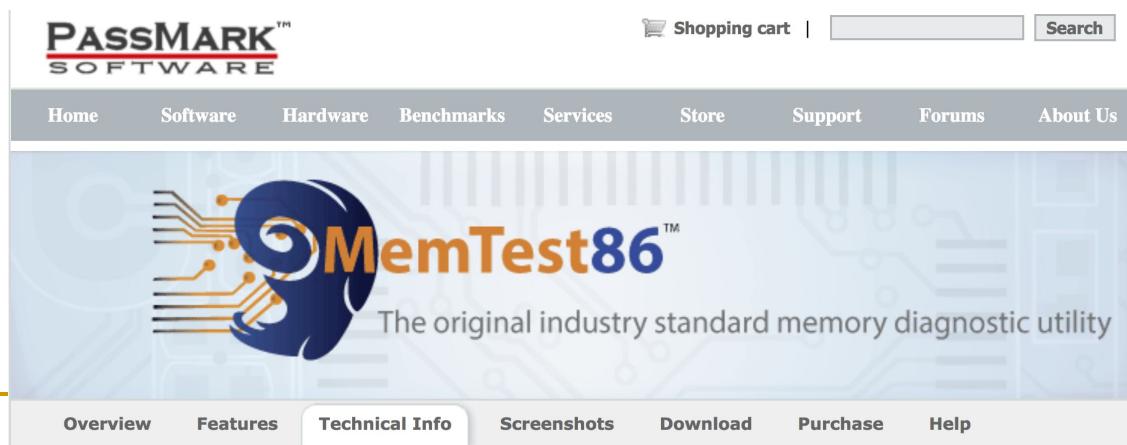
The errors detected during Test 13, albeit exposed only in extreme memory access cases, are most certainly real errors. During typical home PC usage (eg. web browsing, word processing, etc.), it is less likely that the memory usage pattern will fall into the extreme case that make it vulnerable to disturbance errors. It may be of greater concern if you were running highly sensitive equipment such as medical equipment, aircraft control systems, or bank database servers. It is impossible to predict with any accuracy if these errors will occur in real life applications. One would need to do a major scientific study of 1000 of computers and their usage patterns, then do a forensic analysis of each application to study how it makes use of the RAM while it executes. To date, we have only seen 1-bit errors as a result of running the Hammer Test.

# Selected Readings on RowHammer (XI.B)

- PassMark Software, memtest86, since 2014
  - <https://www.memtest86.com/troubleshooting.htm#hammer>

## Detection and mitigation of row hammer errors

The ability of MemTest86 to detect and report on row hammer errors depends on several factors and what mitigations are in place. To generate errors adjacent memory rows must be repeatedly accessed. But hardware features such as multiple channels, interleaving, **scrambling**, Channel Hashing, NUMA & XOR schemes make it nearly impossible (for an arbitrary CPU & RAM stick) to know which memory addresses correspond to which rows in the RAM. Various mitigations might also be in place. Different BIOS firmware might set the refresh interval to different values (tREFI). The shorter the interval the more resistant the RAM will be to errors. But shorter intervals result in higher power consumption and increased processing overhead. Some CPUs also support pseudo target row refresh (pTRR) that can be used in combination with pTRR-compliant RAM. This field allows the RAM stick to indicate the MAC (Maximum Active Count) level which is the RAM can support. A typical value might be 200,000 row activations. Some CPUs also support the Joint Electron Design Engineering Council (JEDEC) Targeted Row Refresh (TRR) algorithm. The TRR is an improved version of the previously implemented pTRR algorithm and does not inflict any performance drop or additional power usage. As a result the row hammer test implemented in MemTest86 maybe not be the worst case possible and vulnerabilities in the underlying RAM might be undetectable due to the mitigations in place in the BIOS and CPU.



# Security Implications (ISCA 2014)

- *Breach of memory protection*
  - OS page (4KB) fits inside DRAM row (8KB)
  - Adjacent DRAM row  Different OS page
- *Vulnerability: disturbance attack*
  - By accessing its own page, a program could corrupt pages belonging to another program
- *We constructed a proof-of-concept*
  - Using only user-level instructions

# More Security Implications (I)

**"We can gain unrestricted access to systems of website visitors."**

www.iaik.tugraz.at ■

Not there yet, but ...



ROOT privileges for web apps!



29

Daniel Gruss (@lavados), Clémentine Maurice (@BloodyTangerine),  
December 28, 2015 — 32c3, Hamburg, Germany

[Rowhammer.js: A Remote Software-Induced Fault Attack in JavaScript \(DIMVA'16\)](#)

# More Security Implications (II)

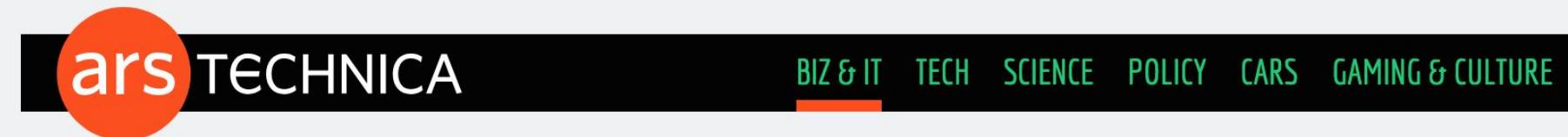
**"Can gain control of a smart phone deterministically"**



Drammer: Deterministic Rowhammer  
Attacks on Mobile Platforms, CCS'16

# More Security Implications (III)

- Using an integrated GPU in a mobile system to remotely escalate privilege via the WebGL interface



"GRAND PWNING UNIT" —

## Drive-by Rowhammer attack uses GPU to compromise an Android phone

JavaScript based GLitch pwns browsers by flipping bits inside memory chips.

DAN GOODIN - 5/3/2018, 12:00 PM

## Grand Pwning Unit: Accelerating Microarchitectural Attacks with the GPU

Pietro Frigo  
Vrije Universiteit  
Amsterdam  
p.frigo@vu.nl

Cristiano Giuffrida  
Vrije Universiteit  
Amsterdam  
giuffrida@cs.vu.nl

Herbert Bos  
Vrije Universiteit  
Amsterdam  
herbertb@cs.vu.nl

Kaveh Razavi  
Vrije Universiteit  
Amsterdam  
kaveh@cs.vu.nl

# More Security Implications (IV)

- Rowhammer over RDMA (I)

ars

TECHNICA

BIZ &amp; IT

TECH

SCIENCE

POLICY

CARS

GAMING &amp; CULTURE

**THROWHAMMER —**

## Packets over a LAN are all it takes to trigger serious Rowhammer bit flips

The bar for exploiting potentially serious DDR weakness keeps getting lower.

DAN GOODIN - 5/10/2018, 5:26 PM

### Throwhammer: Rowhammer Attacks over the Network and Defenses

Andrei Tatar  
*VU Amsterdam*

Radhesh Krishnan  
*VU Amsterdam*

Elias Athanasopoulos  
*University of Cyprus*

Cristiano Giuffrida  
*VU Amsterdam*

Herbert Bos  
*VU Amsterdam*

Kaveh Razavi  
*VU Amsterdam*

# More Security Implications (V)

- Rowhammer over RDMA (II)



Nethammer—Exploiting DRAM Rowhammer Bug Through Network Requests



## Nethammer: Inducing Rowhammer Faults through Network Requests

Moritz Lipp  
Graz University of Technology

Daniel Gruss  
Graz University of Technology

Misiker Tadesse Aga  
University of Michigan

Clémentine Maurice  
Univ Rennes, CNRS, IRISA

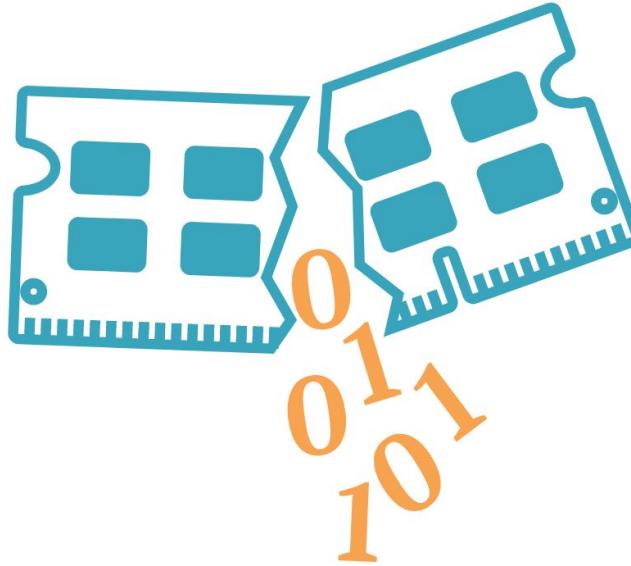
Michael Schwarz  
Graz University of Technology

Lukas Raab  
Graz University of Technology

Lukas Lamster  
Graz University of Technology

# More Security Implications (VI)

- IEEE S&P 2020



## RAMBleed

### RAMBleed: Reading Bits in Memory Without Accessing Them

Andrew Kwong

*University of Michigan*  
[ankwong@umich.edu](mailto:ankwong@umich.edu)

Daniel Genkin

*University of Michigan*  
[genkin@umich.edu](mailto:genkin@umich.edu)

Daniel Gruss

*Graz University of Technology*  
[daniel.gruss@iaik.tugraz.at](mailto:daniel.gruss@iaik.tugraz.at)

Yuval Yarom

*University of Adelaide and Data61*  
[yval@cs.adelaide.edu.au](mailto:yval@cs.adelaide.edu.au)

# More Security Implications (VII)

- Rowhammer on MLC NAND Flash (based on [Cai+, HPCA 2017])



Security

## Rowhammer RAM attack adapted to hit flash storage

Project Zero's two-year-old dog learns a new trick

By [Richard Chirgwin](#) 17 Aug 2017 at 04:27

17 SHARE ▼

---

**From random block corruption to privilege escalation:  
A filesystem attack vector for rowhammer-like attacks**

Anil Kurmus

Nikolas Ioannou

Matthias Neugschwandtner  
Thomas Parnell

Nikolaos Papandreou

*IBM Research – Zurich*

# More Security Implications (VIII)

- USENIX Security 2019

## Terminal Brain Damage: Exposing the Graceless Degradation in Deep Neural Networks Under Hardware Fault Attacks

Sanghyun Hong, Pietro Frigo<sup>†</sup>, Yiğitcan Kaya, Cristiano Giuffrida<sup>†</sup>, Tudor Dumitraş

*University of Maryland, College Park*

*†Vrije Universiteit Amsterdam*



### A Single Bit-flip Can Cause Terminal Brain Damage to DNNs

*One specific bit-flip in a DNN's representation leads to accuracy drop over 90%*

Our research found that a specific bit-flip in a DNN's bitwise representation can cause the accuracy loss up to 90%, and the DNN has 40-50% parameters, on average, that can lead to the accuracy drop over 10% when individually subjected to such single bitwise corruptions...

[Read More](#)

# More Security Implications (IX)

## ■ USENIX Security 2020

### DeepHammer: Depleting the Intelligence of Deep Neural Networks through Targeted Chain of Bit Flips

Fan Yao

*University of Central Florida*  
*fan.yao@ucf.edu*

Adnan Siraj Rakin

*Arizona State University*  
*asrakin@asu.edu*

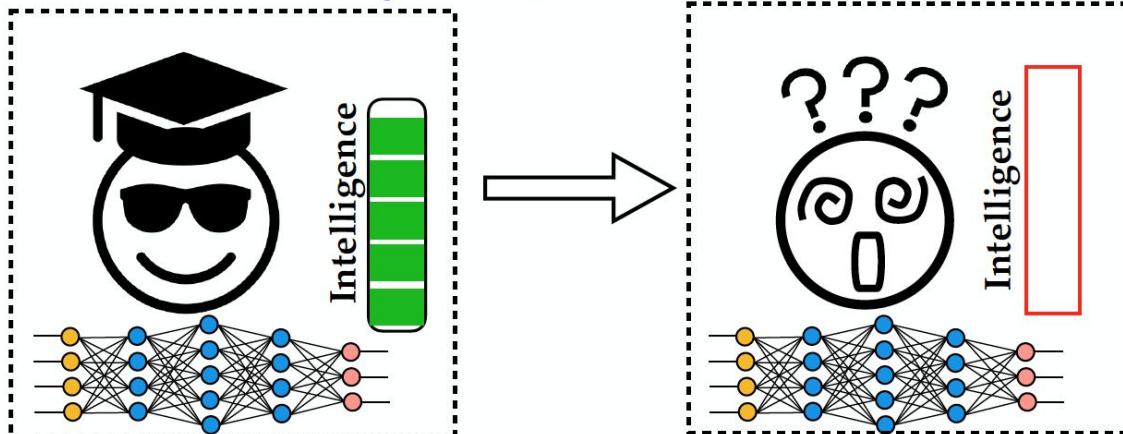
Deliang Fan

*dfan@asu.edu*

Degrade the inference accuracy to the level of Random Guess

Example: ResNet-20 for CIFAR-10, 10 output classes

Before attack, **Accuracy: 90.2%** After attack, **Accuracy: ~10% (1/10)**



# More Security Implications?

---



# Understanding RowHammer

# Root Causes of Disturbance Errors

- *Cause 1: Electromagnetic coupling*
  - Toggling the wordline voltage briefly increases the voltage of adjacent wordlines
  - Slightly opens adjacent rows ☐ Charge leakage
- *Cause 2: Conductive bridges*
- *Cause 3: Hot-carrier injection*

*Confirmed by at least one manufacturer*

# Experimental DRAM Testing Infrastructure



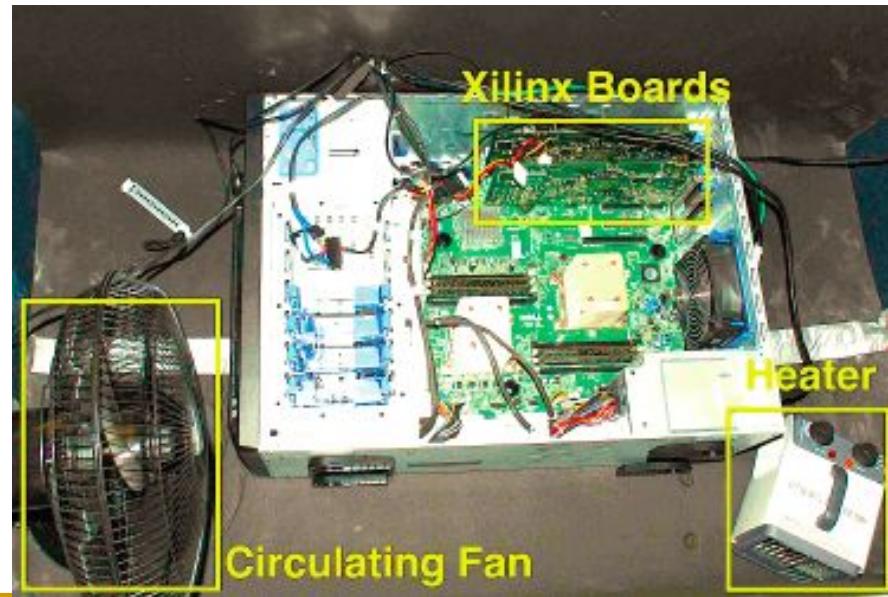
[Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors](#) (Kim et al., ISCA 2014)

[Adaptive-Latency DRAM: Optimizing DRAM Timing for the Common-Case](#) (Lee et al., HPCA 2015)

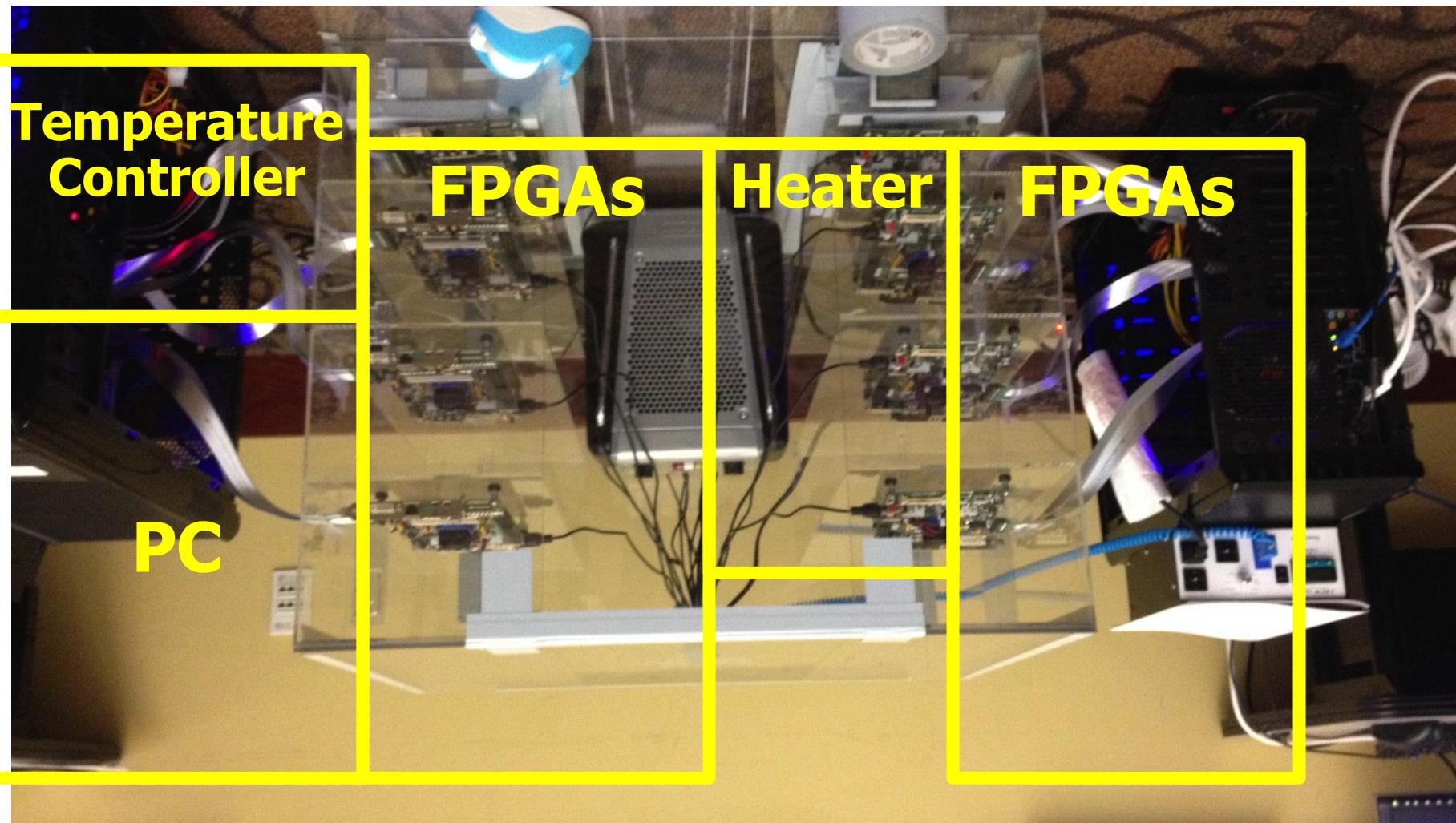
[AVATAR: A Variable-Retention-Time \(VRT\) Aware Refresh for DRAM Systems](#) (Qureshi et al., DSN 2015)

[An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms](#) (Liu et al., ISCA 2013)

[The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study](#) (Khan et al., SIGMETRICS 2014)



# Experimental DRAM Testing Infrastructure



# Tested DRAM Modules (129 total)

Manufacturer	Module	Date*	Timing†		Organization		Chip			Victims-per-Module			RF <sub>th</sub> (ms)
		(yy-ww)	Freq (MT/s)	t <sub>RC</sub> (ns)	Size (GB)	Chips	Size (Gb)‡	Pins	DieVersion§	Average	Minimum	Maximum	Min
A	A <sub>1</sub>	10-08	1066	50.625	0.5	4	1	×16	B	0	0	0	—
	A <sub>2</sub>	10-20	1066	50.625	1	8	1	×8	F	0	0	0	—
	A <sub>3-5</sub>	10-20	1066	50.625	0.5	4	1	×16	B	0	0	0	—
	A <sub>6-7</sub>	11-24	1066	49.125	1	4	2	×16	D	7.8 × 10 <sup>1</sup>	5.2 × 10 <sup>1</sup>	1.0 × 10 <sup>2</sup>	21.3
	A <sub>8-12</sub>	11-26	1066	49.125	1	4	2	×16	D	2.4 × 10 <sup>2</sup>	5.4 × 10 <sup>1</sup>	4.4 × 10 <sup>2</sup>	16.4
	A <sub>13-14</sub>	11-50	1066	49.125	1	4	2	×16	D	8.8 × 10 <sup>1</sup>	1.7 × 10 <sup>1</sup>	1.6 × 10 <sup>2</sup>	26.2
	A <sub>15-16</sub>	12-22	1600	50.625	1	4	2	×16	D	9.5	9	1.0 × 10 <sup>1</sup>	34.4
	A <sub>17-18</sub>	12-26	1600	49.125	2	8	2	×8	M	1.2 × 10 <sup>2</sup>	3.7 × 10 <sup>1</sup>	2.0 × 10 <sup>2</sup>	21.3
	A <sub>19-30</sub>	12-40	1600	48.125	2	8	2	×8	K	8.6 × 10 <sup>6</sup>	7.0 × 10 <sup>6</sup>	1.0 × 10 <sup>7</sup>	8.2
	A <sub>31-34</sub>	13-02	1600	48.125	2	8	2	×8	—	1.8 × 10 <sup>6</sup>	1.0 × 10 <sup>6</sup>	3.5 × 10 <sup>6</sup>	11.5
	A <sub>35-36</sub>	13-14	1600	48.125	2	8	2	×8	—	4.0 × 10 <sup>1</sup>	1.9 × 10 <sup>1</sup>	6.1 × 10 <sup>1</sup>	21.3
	A <sub>37-38</sub>	13-20	1600	48.125	2	8	2	×8	K	1.7 × 10 <sup>6</sup>	1.4 × 10 <sup>6</sup>	2.0 × 10 <sup>6</sup>	9.8
	A <sub>39-40</sub>	13-28	1600	48.125	2	8	2	×8	K	5.7 × 10 <sup>4</sup>	5.4 × 10 <sup>4</sup>	6.0 × 10 <sup>4</sup>	16.4
	A <sub>41</sub>	14-04	1600	49.125	2	8	2	×8	—	2.7 × 10 <sup>5</sup>	2.7 × 10 <sup>5</sup>	2.7 × 10 <sup>5</sup>	18.0
	A <sub>42-43</sub>	14-04	1600	48.125	2	8	2	×8	K	0.5	0	1	62.3
B	B <sub>1</sub>	08-49	1066	50.625	1	8	1	×8	D	0	0	0	—
	B <sub>2</sub>	09-49	1066	50.625	1	8	1	×8	E	0	0	0	—
	B <sub>3</sub>	10-19	1066	50.625	1	8	1	×8	F	0	0	0	—
	B <sub>4</sub>	10-31	1333	49.125	2	8	2	×8	C	0	0	0	—
	B <sub>5</sub>	11-13	1333	49.125	2	8	2	×8	C	0	0	0	—
	B <sub>6</sub>	11-16	1066	50.625	1	8	1	×8	F	0	0	0	—
	B <sub>7</sub>	11-19	1066	50.625	1	8	1	×8	F	0	0	0	—
	B <sub>8</sub>	11-25	1333	49.125	2	8	2	×8	C	0	0	0	—
	B <sub>9</sub>	11-37	1333	49.125	2	8	2	×8	D	1.9 × 10 <sup>6</sup>	1.9 × 10 <sup>6</sup>	1.9 × 10 <sup>6</sup>	11.5
	B <sub>10-12</sub>	11-46	1333	49.125	2	8	2	×8	D	2.2 × 10 <sup>6</sup>	1.5 × 10 <sup>6</sup>	2.7 × 10 <sup>6</sup>	11.5
	B <sub>13</sub>	11-49	1333	49.125	2	8	2	×8	C	0	0	0	—
	B <sub>14</sub>	12-01	1866	47.125	2	8	2	×8	D	9.1 × 10 <sup>5</sup>	9.1 × 10 <sup>5</sup>	9.1 × 10 <sup>5</sup>	9.8
	B <sub>15-31</sub>	12-10	1866	47.125	2	8	2	×8	D	9.8 × 10 <sup>5</sup>	7.8 × 10 <sup>5</sup>	1.2 × 10 <sup>6</sup>	11.5
	B <sub>32</sub>	12-25	1600	48.125	2	8	2	×8	E	7.4 × 10 <sup>5</sup>	7.4 × 10 <sup>5</sup>	7.4 × 10 <sup>5</sup>	11.5
	B <sub>33-42</sub>	12-28	1600	48.125	2	8	2	×8	E	5.2 × 10 <sup>5</sup>	1.9 × 10 <sup>5</sup>	7.3 × 10 <sup>5</sup>	11.5
	B <sub>43-47</sub>	12-31	1600	48.125	2	8	2	×8	E	4.0 × 10 <sup>5</sup>	2.9 × 10 <sup>5</sup>	5.5 × 10 <sup>5</sup>	13.1
	B <sub>48-51</sub>	13-19	1600	48.125	2	8	2	×8	E	1.1 × 10 <sup>5</sup>	7.4 × 10 <sup>4</sup>	1.4 × 10 <sup>5</sup>	14.7
C	B <sub>52-53</sub>	13-40	1333	49.125	2	8	2	×8	D	2.6 × 10 <sup>4</sup>	2.3 × 10 <sup>4</sup>	2.9 × 10 <sup>4</sup>	21.3
	B <sub>54</sub>	14-07	1333	49.125	2	8	2	×8	D	7.5 × 10 <sup>3</sup>	7.5 × 10 <sup>3</sup>	7.5 × 10 <sup>3</sup>	26.2
	C <sub>1</sub>	10-18	1333	49.125	2	8	2	×8	A	0	0	0	—
	C <sub>2</sub>	10-20	1066	50.625	2	8	2	×8	A	0	0	0	—
	C <sub>3</sub>	10-22	1066	50.625	2	8	2	×8	A	0	0	0	—
	C <sub>4-5</sub>	10-26	1333	49.125	2	8	2	×8	B	8.9 × 10 <sup>2</sup>	6.0 × 10 <sup>2</sup>	1.2 × 10 <sup>3</sup>	29.5
	C <sub>6</sub>	10-43	1333	49.125	1	8	1	×8	T	0	0	0	—
	C <sub>7</sub>	10-51	1333	49.125	2	8	2	×8	B	4.0 × 10 <sup>2</sup>	4.0 × 10 <sup>2</sup>	4.0 × 10 <sup>2</sup>	29.5
	C <sub>8</sub>	11-12	1333	46.25	2	8	2	×8	B	6.9 × 10 <sup>2</sup>	6.9 × 10 <sup>2</sup>	6.9 × 10 <sup>2</sup>	21.3
	C <sub>9</sub>	11-19	1333	46.25	2	8	2	×8	B	9.2 × 10 <sup>2</sup>	9.2 × 10 <sup>2</sup>	9.2 × 10 <sup>2</sup>	27.9
	C <sub>10</sub>	11-31	1333	49.125	2	8	2	×8	B	3	3	3	39.3
	C <sub>11</sub>	11-42	1333	49.125	2	8	2	×8	B	1.6 × 10 <sup>2</sup>	1.6 × 10 <sup>2</sup>	1.6 × 10 <sup>2</sup>	39.3
	C <sub>12</sub>	11-48	1600	48.125	2	8	2	×8	C	7.1 × 10 <sup>4</sup>	7.1 × 10 <sup>4</sup>	7.1 × 10 <sup>4</sup>	19.7
	C <sub>13</sub>	12-08	1333	49.125	2	8	2	×8	C	3.9 × 10 <sup>4</sup>	3.9 × 10 <sup>4</sup>	3.9 × 10 <sup>4</sup>	21.3
	C <sub>14-15</sub>	12-12	1333	49.125	2	8	2	×8	C	3.7 × 10 <sup>4</sup>	2.1 × 10 <sup>4</sup>	5.4 × 10 <sup>4</sup>	21.3
	C <sub>16-18</sub>	12-20	1600	48.125	2	8	2	×8	C	3.5 × 10 <sup>3</sup>	1.2 × 10 <sup>3</sup>	7.0 × 10 <sup>3</sup>	27.9
	C <sub>19</sub>	12-23	1600	48.125	2	8	2	×8	E	1.4 × 10 <sup>5</sup>	1.4 × 10 <sup>5</sup>	1.4 × 10 <sup>5</sup>	18.0
	C <sub>20</sub>	12-24	1600	48.125	2	8	2	×8	C	6.5 × 10 <sup>4</sup>	6.5 × 10 <sup>4</sup>	6.5 × 10 <sup>4</sup>	21.3
	C <sub>21</sub>	12-26	1600	48.125	2	8	2	×8	C	2.3 × 10 <sup>4</sup>	2.3 × 10 <sup>4</sup>	2.3 × 10 <sup>4</sup>	24.6
	C <sub>22</sub>	12-32	1600	48.125	2	8	2	×8	C	1.7 × 10 <sup>4</sup>	1.7 × 10 <sup>4</sup>	1.7 × 10 <sup>4</sup>	22.9
	C <sub>23-24</sub>	12-37	1600	48.125	2	8	2	×8	C	2.3 × 10 <sup>4</sup>	1.1 × 10 <sup>4</sup>	3.4 × 10 <sup>4</sup>	18.0
	C <sub>25-30</sub>	12-41	1600	48.125	2	8	2	×8	C	2.0 × 10 <sup>4</sup>	1.1 × 10 <sup>4</sup>	3.2 × 10 <sup>4</sup>	19.7
	C <sub>31</sub>	13-11	1600	48.125	2	8	2	×8	C	3.3 × 10 <sup>5</sup>	3.3 × 10 <sup>5</sup>	3.3 × 10 <sup>5</sup>	14.7
	C <sub>32</sub>	13-35	1600	48.125	2	8	2	×8	C	3.7 × 10 <sup>4</sup>	3.7 × 10 <sup>4</sup>	3.7 × 10 <sup>4</sup>	21.3

\* We report the manufacture date marked on the chip packages, which is more accurate than other dates that can be gleaned from a module.

† We report timing constraints stored in the module's on-board ROM [33], which is read by the system BIOS to calibrate the memory controller.

‡ The maximum DRAM die versions supported by our testing platform is 2Gb.

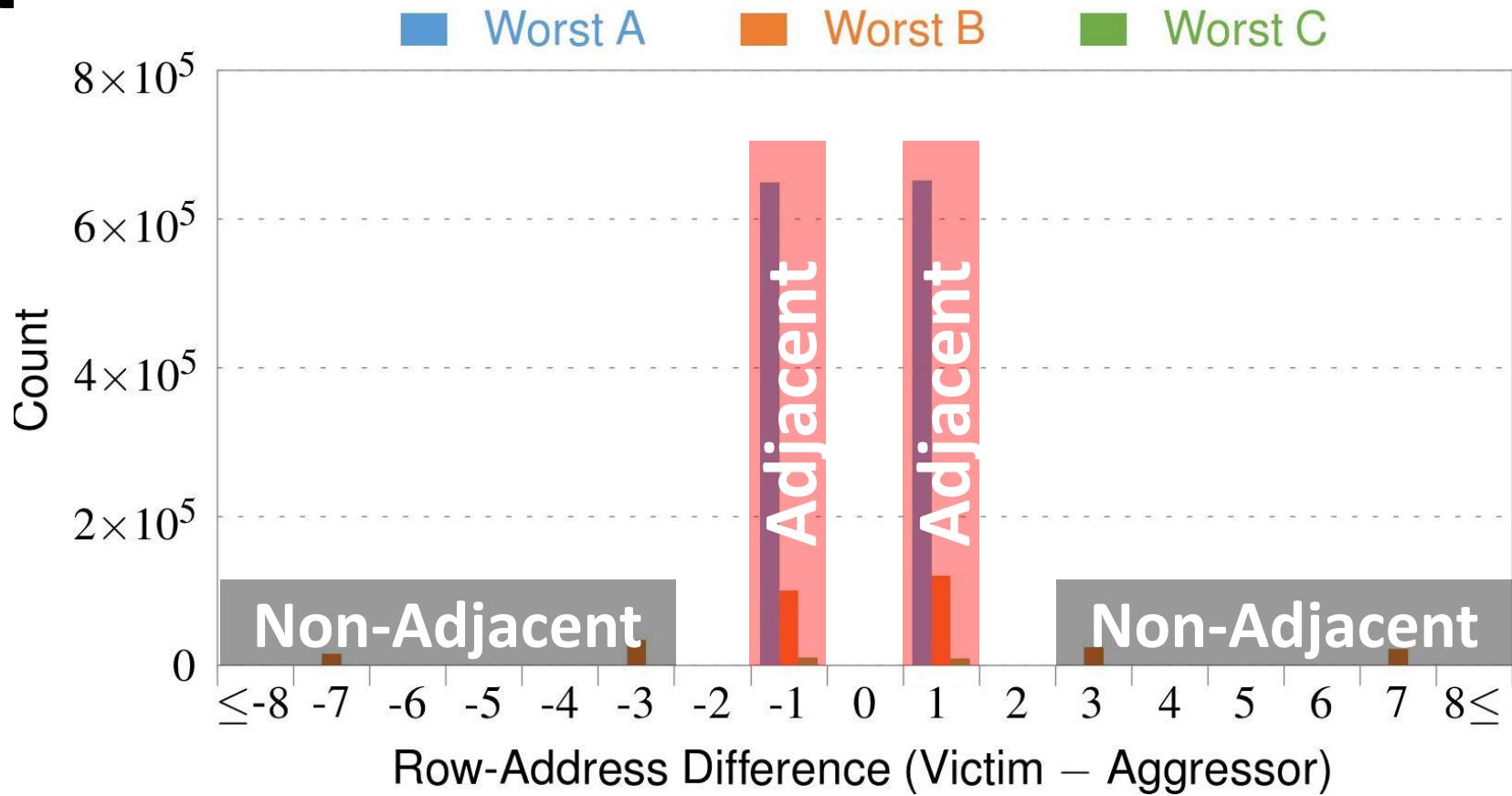
§ We report DRAM die versions marked on the chip packages, which typically progress in the following manner: M → A → B → C → ···.

Table 3. Sample population of 129 DDR3 DRAM modules, categorized by manufacturer and sorted by manufacture date

# RowHammer Characterization Results

1. Most Modules Are at Risk
2. Errors vs. Vintage
3. Error = Charge Loss
4. Adjacency: Aggressor & Victim
5. Sensitivity Studies
6. Other Results in Paper
7. Solution Space

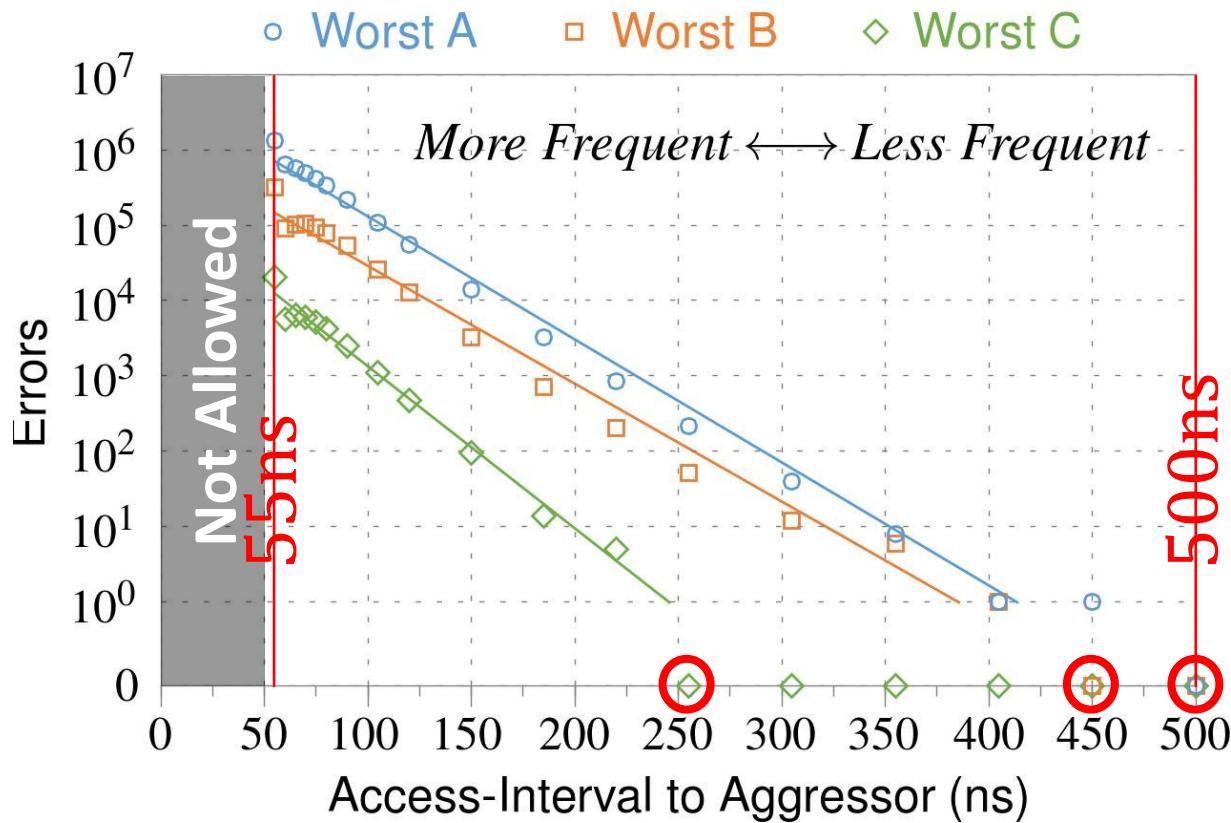
# 4. Adjacency: Aggressor & Victim



*Note: For three modules with the most errors (only first bank)*

*Most aggressors & victims are adjacent*

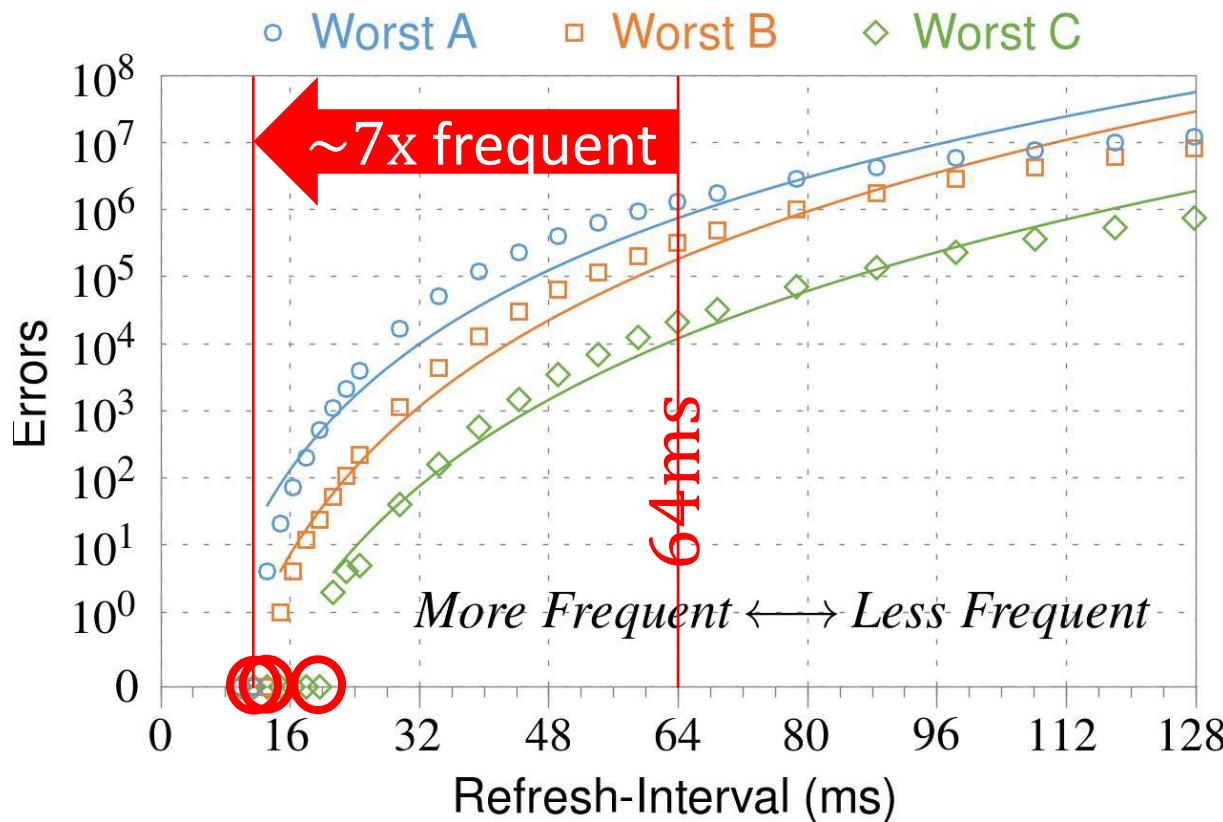
# ① Access Interval (Aggressor)



Note: For three modules with the most errors (only first bank)

*Less frequent accesses  Fewer errors*

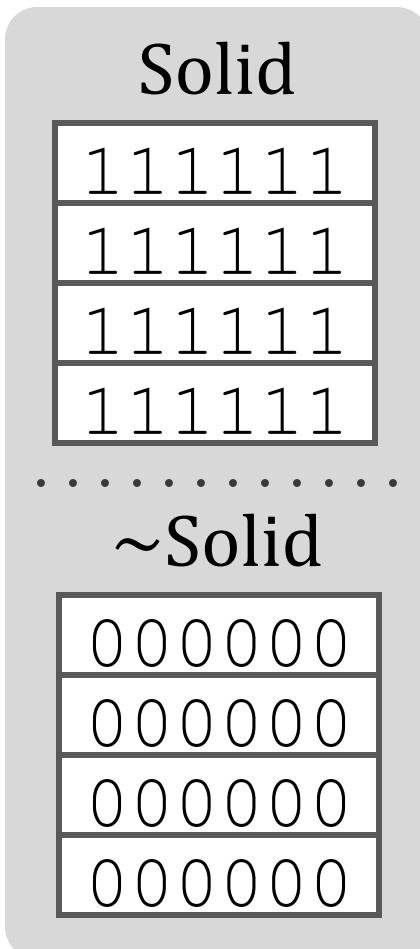
## ② Refresh Interval



Note: Using three modules with the most errors (only first bank)

*More frequent refreshes  Fewer errors*

# ③ Data Pattern



*Errors affected by data stored in other cells*

# 6. Other Results (in Paper)

- *Victim Cells  $\neq$  Weak Cells (i.e., leaky cells)*
  - Almost no overlap between them
- *Errors not strongly affected by temperature*
  - Default temperature: 50°C
  - At 30°C and 70°C, number of errors changes <15%
- *Errors are repeatable*
  - Across ten iterations of testing, >70% of victim cells had errors in every iteration

# 6. Other Results (in Paper)

## cont'd

- *As many as 4 errors per cache-line*
  - Simple ECC (e.g., SECDED) cannot prevent all errors
- *Number of cells & rows affected by aggressor*
  - Victims cells per aggressor:  $\leq 110$
  - Victims rows per aggressor:  $\leq 9$
- *Cells affected by two aggressors on either side*
  - Very small fraction of victim cells ( $< 100$ ) have an error when either one of the aggressors is toggled

# First RowHammer Analysis

---

- Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu,

## "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors"

*Proceedings of the 41st International Symposium on Computer Architecture (ISCA), Minneapolis, MN, June 2014.*

[Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)] [Source Code and Data] [Lecture Video (1 hr 49 mins), 25 September 2020]

*One of the 7 papers of 2012-2017 selected as Top Picks in Hardware and Embedded Security for IEEE TCAD (link).*

# Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors

Yoongu Kim<sup>1</sup>   Ross Daly\*   Jeremie Kim<sup>1</sup>   Chris Fallin\*   Ji Hye Lee<sup>1</sup>  
Donghyuk Lee<sup>1</sup>   Chris Wilkerson<sup>2</sup>   Konrad Lai   Onur Mutlu<sup>1</sup>

---

<sup>1</sup>Carnegie Mellon University

<sup>2</sup>Intel Labs

# RowHammer Solutions

# Two Types of RowHammer Solutions

---

- Immediate
  - To protect the vulnerable DRAM chips in the field
  - Limited possibilities
- Longer-term
  - To protect future DRAM chips
  - Wider range of protection mechanisms
- Our ISCA 2014 paper proposes both types of solutions
  - Seven solutions in total
  - PARA proposed as best solution □ already employed in the field

# Some Potential Solutions

---

- Make better DRAM chips      Cost
- Refresh frequently      Power, Performance
- Sophisticated ECC      Cost, Power
- Access counters      Cost, Power, Complexity

# Naive Solutions

## ① *Throttle accesses to same row*

- Limit access-interval:  $\geq 500\text{ns}$
- Limit number of accesses:  $\leq 128\text{K}$  ( $=64\text{ms}/500\text{ns}$ )

## ② *Refresh more frequently*

- Shorten refresh-interval by  $\sim 7\times$

*Both naive solutions introduce significant overhead in performance and power*

# Apple's Patch for RowHammer

---

- <https://support.apple.com/en-gb/HT204934>

Available for: OS X Mountain Lion v10.8.5, OS X Mavericks v10.9.5

Impact: A malicious application may induce memory corruption to escalate privileges

Description: A disturbance error, also known as Rowhammer, exists with some DDR3 RAM that could have led to memory corruption. This issue was mitigated by increasing memory refresh rates.

CVE-ID

CVE-2015-3693 : Mark Seaborn and Thomas Dullien of Google, working from original research by Yoongu Kim et al (2014)

HP, Lenovo, and other vendors released similar patches

---

# Our Solution to RowHammer

- PARA: *Probabilistic Adjacent Row Activation*
- Key Idea
  - After closing a row, we activate (i.e., refresh) one of its neighbors with a low probability:  $p = 0.005$
- Reliability Guarantee
  - When  $p=0.005$ , errors in one year:  $9.4 \times 10^{-14}$
  - By adjusting the value of  $p$ , we can vary the strength of protection against errors

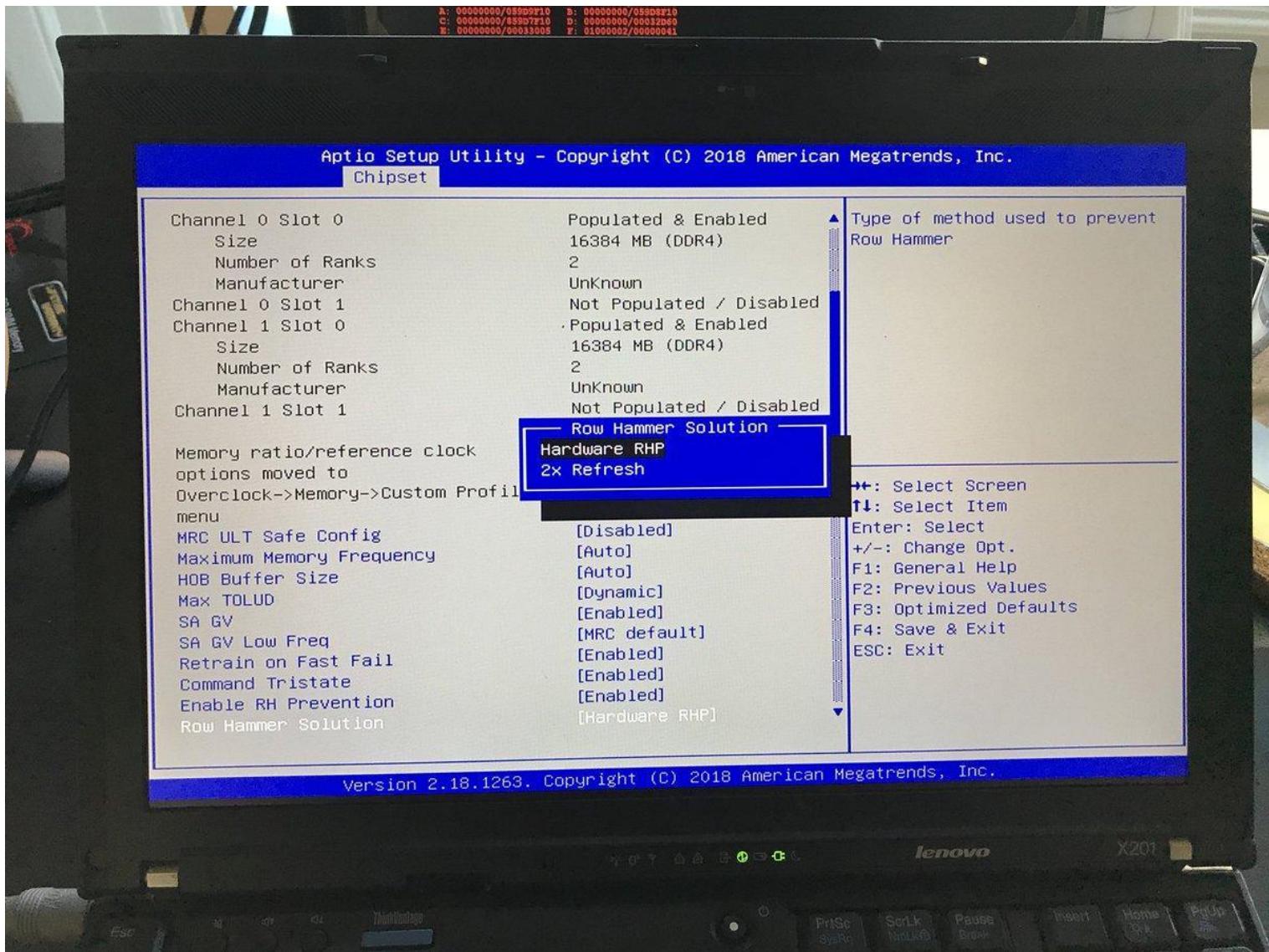
# Advantages of PARA

- *PARA refreshes rows infrequently*
  - Low power
  - Low performance-overhead
    - Average slowdown: 0.20% (for 29 benchmarks)
    - Maximum slowdown: 0.75%
- *PARA is stateless*
  - Low cost
  - Low complexity
- *PARA is an effective and low-overhead solution to prevent disturbance errors*

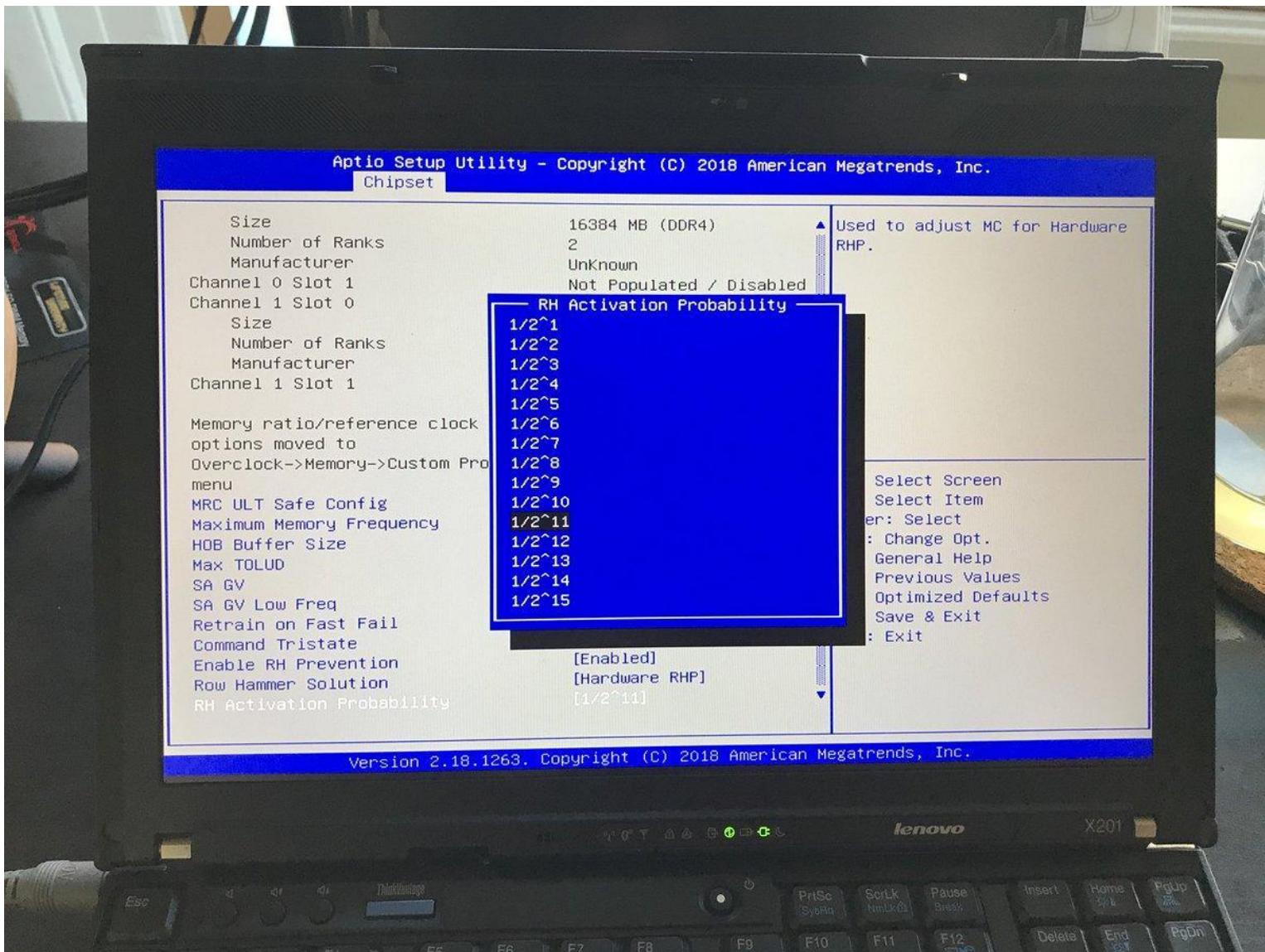
# Requirements for PARA

- If implemented in DRAM chip (done today)
  - Enough slack in timing and refresh parameters
  - Plenty of slack today:
    - Lee et al., “Adaptive-Latency DRAM: Optimizing DRAM Timing for the Common Case,” HPCA 2015.
    - Chang et al., “Understanding Latency Variation in Modern DRAM Chips,” SIGMETRICS 2016.
    - Lee et al., “Design-Induced Latency Variation in Modern DRAM Chips,” SIGMETRICS 2017.
    - Chang et al., “Understanding Reduced-Voltage Operation in Modern DRAM Devices,” SIGMETRICS 2017.
    - Ghose et al., “What Your DRAM Power Models Are Not Telling You: Lessons from a Detailed Experimental Study,” SIGMETRICS 2018.
    - Kim et al., “Solar-DRAM: Reducing DRAM Access Latency by Exploiting the Variation in Local Bitlines,” ICCD 2018.
- If implemented in memory controller
  - Better coordination between memory controller and DRAM
  - Memory controller should know which rows are physically adjacent

# Probabilistic Activation in Real Life (I)



# Probabilistic Activation in Real Life (II)



# Seven RowHammer Solutions Proposed

---

- Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu,

## "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors"

*Proceedings of the 41st International Symposium on Computer Architecture (ISCA), Minneapolis, MN, June 2014.*

[Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)] [Source Code and Data] [Lecture Video (1 hr 49 mins), 25 September 2020]

*One of the 7 papers of 2012-2017 selected as Top Picks in Hardware and Embedded Security for IEEE TCAD (link).*

## Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors

Yoongu Kim<sup>1</sup>   Ross Daly\*   Jeremie Kim<sup>1</sup>   Chris Fallin\*   Ji Hye Lee<sup>1</sup>  
Donghyuk Lee<sup>1</sup>   Chris Wilkerson<sup>2</sup>   Konrad Lai   Onur Mutlu<sup>1</sup>

---

<sup>1</sup>Carnegie Mellon University

<sup>2</sup>Intel Labs

# Main Memory Needs Intelligent Controllers for Security

# Industry Is Writing Papers About It, Too

## DRAM Process Scaling Challenges

### ❖ Refresh

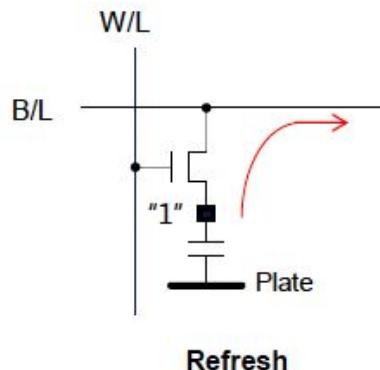
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance
- Leakage current of cell access transistors increasing

### ❖ tWR

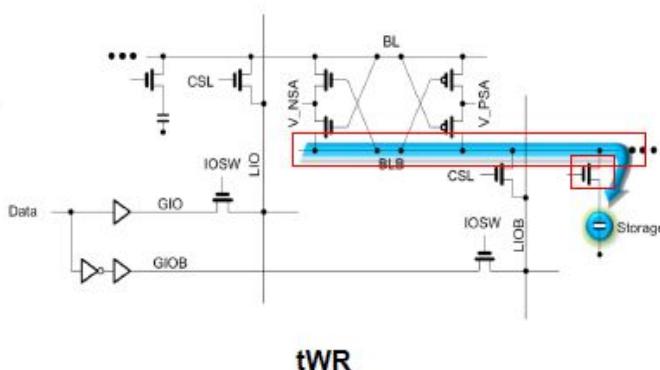
- Contact resistance between the cell capacitor and access transistor increasing
- On-current of the cell access transistor decreasing
- Bit-line resistance increasing

### ❖ VRT

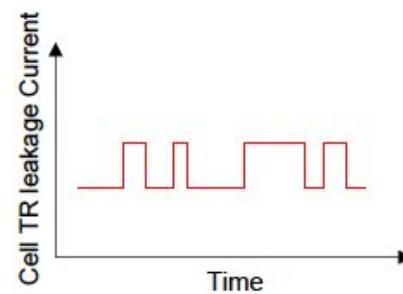
- Occurring more frequently with cell capacitance decreasing



Refresh



tWR



VRT



# Call for Intelligent Memory Controllers

## DRAM Process Scaling Challenges

### ❖ Refresh

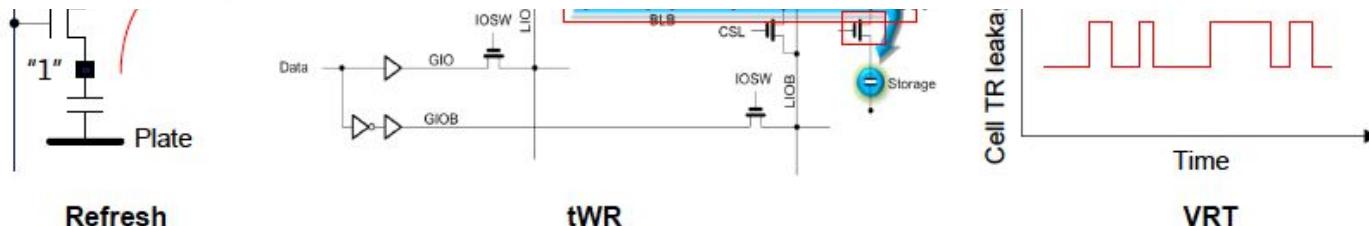
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance

THE MEMORY FORUM 2014

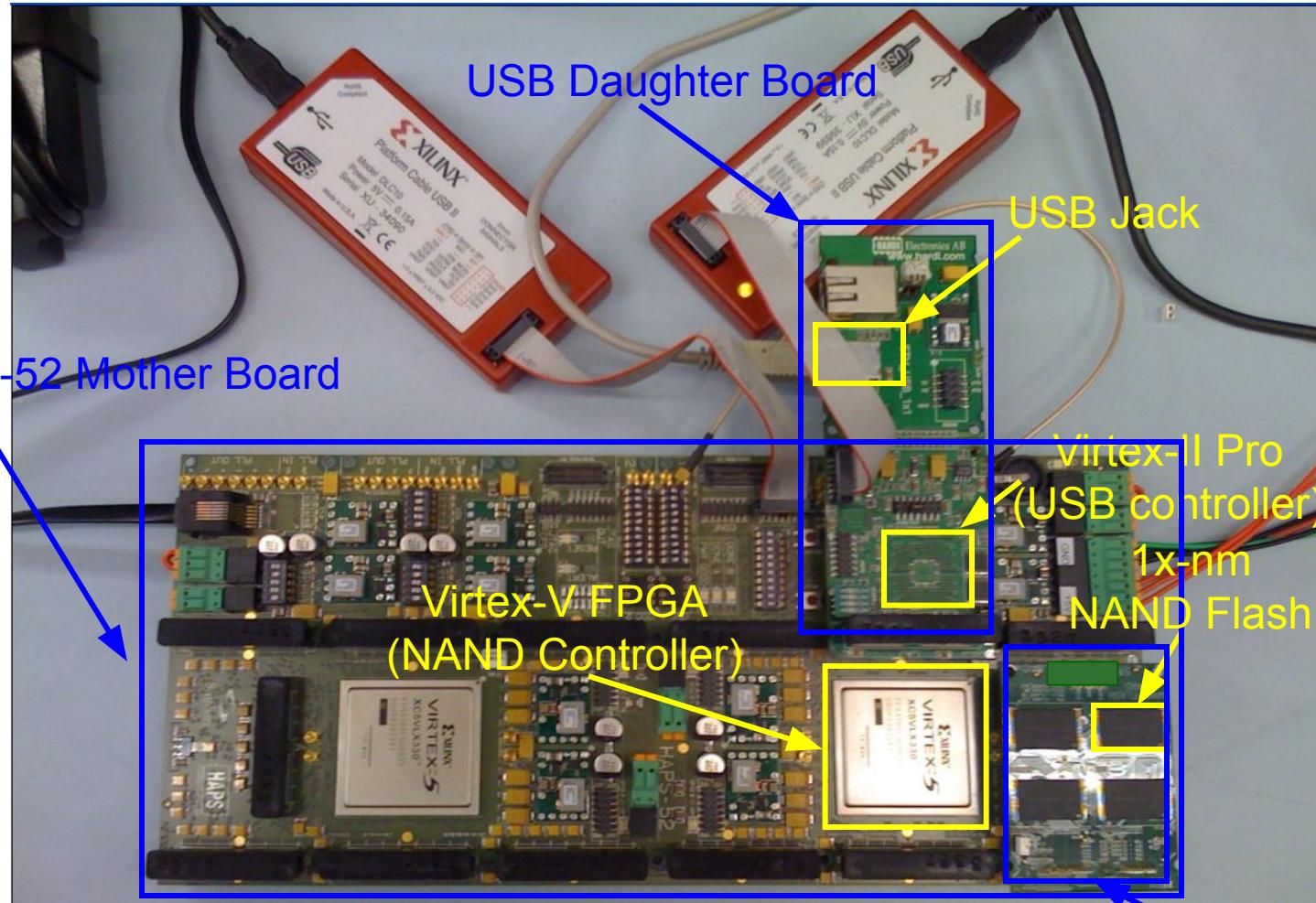
## Co-Architecting Controllers and DRAM to Enhance DRAM Process Scaling

Uksong Kang, Hak-soo Yu, Churoo Park, \*Hongzhong Zheng,  
\*\*John Halbert, \*\*Kuljit Bains, SeongJin Jang, and Joo Sun Choi

*Samsung Electronics, Hwasung, Korea / \*Samsung Electronics, San Jose / \*\*Intel*



# Aside: Intelligent Controller for NAND Flash



[DATE 2012, ICCD 2012, DATE 2013, ITJ 2013, ICCD 2013, SIGMETRICS 2014, HPCA 2015, DSN 2015, MSST 2015, JSAC 2016, HPCA 2017, DFRWS 2017, PIEEE 2017, HPCA 2018, SIGMETRICS 2018]

# Aside: Intelligent Controller for NAND Flash

---



*Proceedings of the IEEE, Sept. 2017*



## Error Characterization, Mitigation, and Recovery in Flash-Memory-Based Solid-State Drives

*This paper reviews the most recent advances in solid-state drive (SSD) error characterization, mitigation, and data recovery techniques to improve both SSD's reliability and lifetime.*

By YU CAI, SAUGATA GHOSE, ERICH F. HARATSCH, YIXIN LUO, AND ONUR MUTLU

<https://arxiv.org/pdf/1706.08642>

# First RowHammer Analysis

---

- Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu,

## "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors"

*Proceedings of the 41st International Symposium on Computer Architecture (ISCA), Minneapolis, MN, June 2014.*

[Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)] [Source Code and Data] [Lecture Video (1 hr 49 mins), 25 September 2020]

*One of the 7 papers of 2012-2017 selected as Top Picks in Hardware and Embedded Security for IEEE TCAD (link).*

# Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors

Yoongu Kim<sup>1</sup>   Ross Daly\*   Jeremie Kim<sup>1</sup>   Chris Fallin\*   Ji Hye Lee<sup>1</sup>  
Donghyuk Lee<sup>1</sup>   Chris Wilkerson<sup>2</sup>   Konrad Lai   Onur Mutlu<sup>1</sup>

---

<sup>1</sup>Carnegie Mellon University

<sup>2</sup>Intel Labs

# Retrospective on RowHammer & Future

---

- Onur Mutlu,

**"The RowHammer Problem and Other Issues We May Face as Memory Becomes Denser"**

*Invited Paper in Proceedings of the Design, Automation, and Test in Europe Conference (DATE), Lausanne, Switzerland, March 2017.*

[Slides (pptx) (pdf)]

## The RowHammer Problem and Other Issues We May Face as Memory Becomes Denser

Onur Mutlu  
ETH Zürich  
[onur.mutlu@inf.ethz.ch](mailto:onur.mutlu@inf.ethz.ch)  
<https://people.inf.ethz.ch/omutlu>

# A More Recent RowHammer Retrospective

---

- Onur Mutlu and Jeremie Kim,

## **"RowHammer: A Retrospective"**

*IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD) Special Issue on Top Picks in Hardware and Embedded Security, 2019.*

[Preliminary arXiv version]

[Slides from COSADE 2019 (pptx)]

[Slides from VLSI-SOC 2020 (pptx) (pdf)]

[Talk Video (1 hr 15 minutes, with Q&A)]

# RowHammer: A Retrospective

Onur Mutlu<sup>§‡</sup>

<sup>§</sup>ETH Zürich

Jeremie S. Kim<sup>‡§</sup>

<sup>‡</sup>Carnegie Mellon University

## A Key Takeaway

---

Main Memory Needs  
Intelligent Controllers

# RowHammer in 2020

# RowHammer in 2020 (I)

---

- Jeremie S. Kim, Minesh Patel, A. Giray Yaglikci, Hasan Hassan, Roknoddin Azizi, Lois Orosa, and Onur Mutlu,  
**"Revisiting RowHammer: An Experimental Analysis of Modern Devices and Mitigation Techniques"**

*Proceedings of the 47th International Symposium on Computer Architecture (ISCA), Valencia, Spain, June 2020.*

[[Slides \(pptx\)](#) ([pdf](#))]

[[Lightning Talk Slides \(pptx\)](#) ([pdf](#))]

[[Talk Video](#) (20 minutes)]

[[Lightning Talk Video](#) (3 minutes)]

## Revisiting RowHammer: An Experimental Analysis of Modern DRAM Devices and Mitigation Techniques

Jeremie S. Kim<sup>§†</sup>      Minesh Patel<sup>§</sup>      A. Giray Yağlıkçı<sup>§</sup>  
Hasan Hassan<sup>§</sup>      Roknoddin Azizi<sup>§</sup>      Lois Orosa<sup>§</sup>      Onur Mutlu<sup>§†</sup>

<sup>§</sup>*ETH Zürich*

<sup>†</sup>*Carnegie Mellon University*

# RowHammer in 2020 (II)

---

- Pietro Frigo, Emanuele Vannacci, Hasan Hassan, Victor van der Veen, Onur Mutlu, Cristiano Giuffrida, Herbert Bos, and Kaveh Razavi,

## **"TRRespass: Exploiting the Many Sides of Target Row Refresh"**

*Proceedings of the 41st IEEE Symposium on Security and Privacy (S&P)*, San Francisco, CA, USA, May 2020.

[[Slides \(pptx\)](#) ([pdf](#))]

[[Lecture Slides \(pptx\)](#) ([pdf](#))]

[[Talk Video](#) (17 minutes)]

[[Lecture Video](#) (59 minutes)]

[[Source Code](#)]

[[Web Article](#)]

*Best paper award.*

*Pwnie Award 2020 for Most Innovative Research.* [Pwnie Awards 2020](#)

# TRRespass: Exploiting the Many Sides of Target Row Refresh

Pietro Frigo\*†    Emanuele Vannacci\*†    Hasan Hassan§    Victor van der Veen¶  
Onur Mutlu§    Cristiano Giuffrida\*    Herbert Bos\*    Kaveh Razavi\*

# RowHammer in 2020 (III)

---

- Lucian Cojocar, Jeremie Kim, Minesh Patel, Lillian Tsai, Stefan Saroiu, Alec Wolman, and Onur Mutlu,

## **"Are We Susceptible to Rowhammer? An End-to-End Methodology for Cloud Providers"**

*Proceedings of the 41st IEEE Symposium on Security and Privacy (S&P), San Francisco, CA, USA, May 2020.*

[Slides (pptx) (pdf)]

[Talk Video (17 minutes)]

## Are We Susceptible to Rowhammer?

### An End-to-End Methodology for Cloud Providers

Lucian Cojocar, Jeremie Kim<sup>§†</sup>, Minesh Patel<sup>§</sup>, Lillian Tsai<sup>‡</sup>,  
Stefan Saroiu, Alec Wolman, and Onur Mutlu<sup>§†</sup>  
Microsoft Research, <sup>§</sup>ETH Zürich, <sup>†</sup>CMU, <sup>‡</sup>MIT

# BlockHammer Solution in 2021

---

- A. Giray Yaglikci, Minesh Patel, Jeremie S. Kim, Roknoddin Azizi, Ataberk Olgun, Lois Orosa, Hasan Hassan, Jisung Park, Konstantinos Kanellopoulos, Taha Shahroodi, Saugata Ghose, and Onur Mutlu,

## **"BlockHammer: Preventing RowHammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows"**

*Proceedings of the 27th International Symposium on High-Performance Computer Architecture (HPCA)*, Virtual, February-March 2021.

[Slides (pptx) (pdf)]

[Short Talk Slides (pptx) (pdf)]

[Talk Video (22 minutes)]

[Short Talk Video (7 minutes)]

## **BlockHammer: Preventing RowHammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows**

A. Giray Yağlıkçı<sup>1</sup> Minesh Patel<sup>1</sup> Jeremie S. Kim<sup>1</sup> Roknoddin Azizi<sup>1</sup> Ataberk Olgun<sup>1</sup> Lois Orosa<sup>1</sup>  
Hasan Hassan<sup>1</sup> Jisung Park<sup>1</sup> Konstantinos Kanellopoulos<sup>1</sup> Taha Shahroodi<sup>1</sup> Saugata Ghose<sup>2</sup> Onur Mutlu<sup>1</sup>

<sup>1</sup>ETH Zürich

<sup>2</sup>University of Illinois at Urbana-Champaign

# Two Upcoming RowHammer Papers at MICRO 2021

---

- Lois Orosa, Abdullah Giray Yaglikci, Haocong Luo, Ataberk Olgun, Jisung Park, Hasan Hassan, Minesh Patel, Jeremie S. Kim, Onur Mutlu,  
**"A Deeper Look into RowHammer's Sensitivities: Experimental Analysis of Real DRAM Chips and Implications on Future Attacks and Defenses"**

*MICRO 2021*

## A Deeper Look into RowHammer's Sensitivities: Experimental Analysis of Real DRAM Chips and Implications on Future Attacks and Defenses

Lois Orosa\*  
ETH Zürich

A. Giray Yağlıkçı\*  
ETH Zürich

Haocong Luo  
ETH Zürich

Ataberk Olgun  
ETH Zürich, TOBB ETÜ

Jisung Park  
ETH Zürich

Hasan Hassan  
ETH Zürich

Minesh Patel  
ETH Zürich

Jeremie S. Kim  
ETH Zürich

Onur Mutlu  
ETH Zürich

# Two Upcoming RowHammer Papers at MICRO 2021

---

- Hasan Hassan, Yahya Can Tugrul, Jeremie S. Kim, Victor van der Veen, Kaveh Razavi, Onur Mutlu,

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

*MICRO 2021*

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

Hasan Hassan<sup>†</sup>

<sup>†</sup>*ETH Zürich*

Yahya Can Tuğrul<sup>†‡</sup>

Kaveh Razavi<sup>†</sup>

<sup>‡</sup>*TOBB University of Economics & Technology*

Jeremie S. Kim<sup>†</sup>

Onur Mutlu<sup>†</sup>

Victor van der Veen<sup>σ</sup>

<sup>σ</sup>*Qualcomm Technologies Inc.*

# Computer Architecture

## Lecture 4c: RowHammer

Prof. Onur Mutlu  
ETH Zürich  
Fall 2021  
8 October 2021

# Slides for Next Lecture

# *Revisiting RowHammer*

## *An Experimental Analysis of Modern Devices and Mitigation Techniques*

Jeremie S. Kim

Minesh Patel

A. Giray Yağlıkçı

Hasan Hassan

Roknoddin Azizi

Lois Orosa

Onur Mutlu

**SAFARI**

# Executive Summary

- **Motivation:** Denser DRAM chips are more vulnerable to RowHammer but no characterization-based study demonstrates how vulnerability scales
- **Problem:** Unclear if existing mitigation mechanisms will remain viable for future DRAM chips that are likely to be more vulnerable to RowHammer
- **Goal:**
  1. Experimentally demonstrate how vulnerable modern DRAM chips are to RowHammer and study how this vulnerability will scale going forward
  2. Study viability of existing mitigation mechanisms on more vulnerable chips
- **Experimental Study:** First rigorous RowHammer characterization study across a broad range of DRAM chips
  - 1580 chips of different DRAM {types, technology node generations, manufacturers}
  - We find that RowHammer vulnerability worsens in newer chips
- **RowHammer Mitigation Mechanism Study:** How five state-of-the-art mechanisms are affected by worsening RowHammer vulnerability
  - Reasonable performance loss (8% on average) on modern DRAM chips
  - Scale poorly to more vulnerable DRAM chips (e.g., 80% performance loss)
- **Conclusion:** it is critical to research more effective solutions to RowHammer for future DRAM chips that will likely be even more vulnerable to RowHammer

# Motivation

- Denser DRAM chips are **more vulnerable** to RowHammer
- Three prior works [**Kim+, ISCA'14**], [**Park+, MR'16**], [**Park+, MR'16**], **over the last six years** provide RowHammer characterization data on real DRAM
- However, there is **no comprehensive experimental study** that demonstrates **how vulnerability scales** across DRAM types and technology node generations
- It is **unclear whether current mitigation mechanisms will remain viable** for future DRAM chips that are likely to be more vulnerable to RowHammer

# Goal

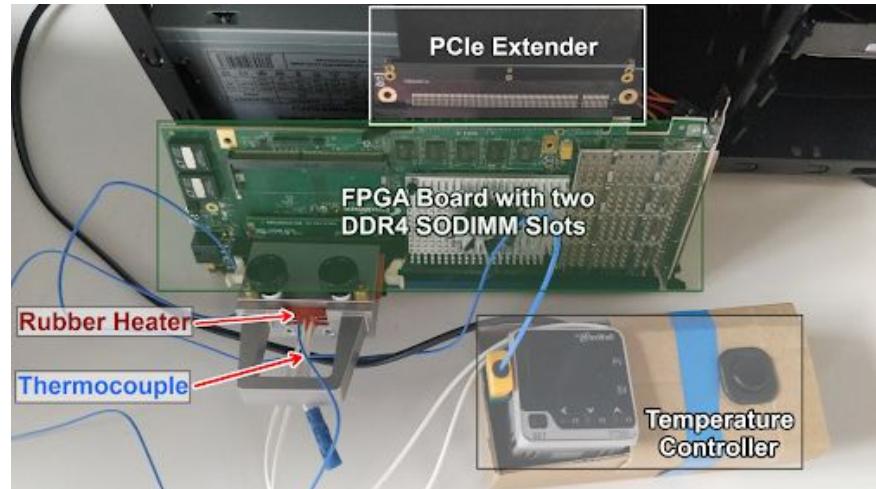
1. Experimentally demonstrate how vulnerable modern DRAM chips are to RowHammer and predict how this vulnerability will scale going forward
2. Examine the viability of current mitigation mechanisms on more vulnerable chips

# DRAM Testing Infrastructures

Three separate testing infrastructures

1. **DDR3:** FPGA-based SoftMC [Hassan+, HPCA'17]  
(Xilinx ML605)
2. **DDR4:** FPGA-based SoftMC [Hassan+, HPCA'17]  
(Xilinx Virtex UltraScale 95)
3. **LPDDR4:** In-house testing hardware for LPDDR4 chips

All provide fine-grained control over DRAM commands, timing parameters and temperature



DDR4 DRAM testing infrastructure

# DRAM Chips Tested

DRAM type-node	Number of Chips (Modules) Tested			
	Mfr. A	Mfr. B	Mfr. C	Total
DDR3-old	56 (10)	88 (11)	28 (7)	<b>172 (28)</b>
DDR3-new	80 (10)	52 (9)	104 (13)	<b>236 (32)</b>
DDR4-old	112 (16)	24 (3)	128 (18)	<b>264 (37)</b>
DDR4-new	264 (43)	16 (2)	108 (28)	<b>388 (73)</b>
LPDDR4-1x	12 (3)	180 (45)	N/A	<b>192 (48)</b>
LPDDR4-1y	184 (46)	N/A	144 (36)	<b>328 (82)</b>

**1580** total DRAM chips tested from **300** DRAM modules

- **Three** major DRAM manufacturers {A, B, C}
- **Three** DRAM *types* or *standards* {DDR3, DDR4, LPDDR4}
  - LPDDR4 chips we test implement on-die ECC
- **Two** technology nodes per DRAM type {old/new, 1x/1y}
  - Categorized based on manufacturing date, datasheet publication date, purchase date, and characterization results

**Type-node:** configuration describing a chip's type and technology node generation: **DDR3-old/new, DDR4-old/new, LPDDR4-1x/1y**

# Effective RowHammer Characterization

To characterize our DRAM chips at **worst-case** conditions, we:

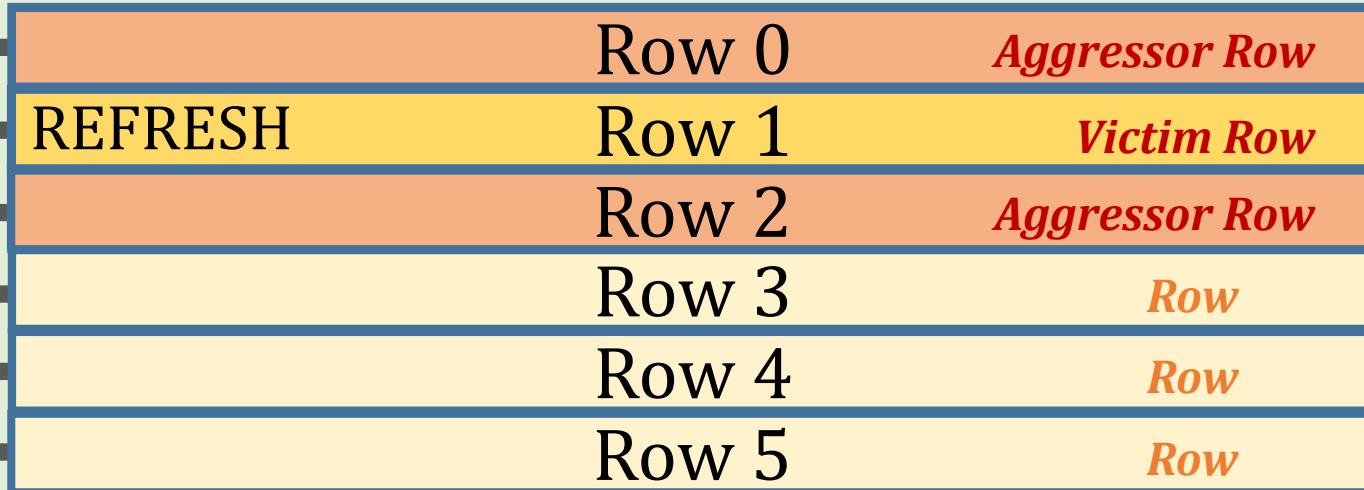
## 1. Prevent sources of interference during core test loop

- We disable:
  - **DRAM refresh**: to avoid refreshing victim row
  - **DRAM calibration events**: to minimize variation in test timing
  - **RowHammer mitigation mechanisms**: to observe circuit-level effects
- Test for **less than refresh window (32ms)** to avoid retention failures

## 2. Worst-case access sequence

- We use **worst-case** access sequence based on prior works' observations
- For each row, **repeatedly access the two directly physically-adjacent rows as fast as possible**

# Testing Methodology



DRAM RowHammer Characterization():

**foreach** row in DRAM:

```
    set victim_row to row
    set aggressor_row1 to victim_row - 1
    set aggressor_row2 to victim_row + 1
    Disable DRAM refresh
    Refresh victim_row
```

**for** n = 1 → HC: // core test loop

```
    activate aggressor_row1
    activate aggressor_row2
```

Enable DRAM refresh

Record RowHammer bit flips to storage

Restore bit flips to original values

Disable refresh to **prevent interruptions** in the core loop of our test **from refresh operations**

Induce RowHammer bit flips on a **fully charged row**

# Testing Methodology

<i>closed</i>	Row 0	<i>Aggressor Row</i>
	Row 1	<i>Aggressor Row</i>
	Row 2	<i>Row</i>
	Row 3	<i>Aggressor Row</i>
	Row 4	<i>Victim Row</i>
	Row 5	<i>Aggressor Row</i>

DRAM RowHammer Characterization():

**foreach** row in DRAM:

```
    set victim_row to row
    set aggressor_row1 to victim_row - 1
    set aggressor_row2 to victim_row + 1
    Disable DRAM refresh
    Refresh victim_row
```

**for** n = 1 → HC: // core test loop

```
    activate aggressor_row1
    activate aggressor_row2
```

Enable DRAM refresh

Record RowHammer bit flips to storage  
Restore bit flips to original values

Disable refresh to **prevent interruptions** in the core loop of our test **from refresh operations**

Induce RowHammer bit flips on a **fully charged row**

Core test loop where we alternate accesses to adjacent rows

**1 Hammer (HC) = two accesses**

Prevent further retention failures

Record bit flips for analysis

# Key Takeaways from 1580 Chips

- Chips of newer DRAM technology nodes are **more vulnerable** to RowHammer
- There are chips today whose weakest cells fail after **only 4800 hammers**
- Chips of newer DRAM technology nodes can exhibit RowHammer bit flips 1) in **more rows** and 2) **farther away** from the victim row.

# 1. RowHammer Vulnerability

*Q. Can we induce RowHammer bit flips in all of our DRAM chips?*

All chips are vulnerable, except many DDR3 chips

- A total of 1320 out of all 1580 chips (**84%**) are vulnerable
- Within **DDR3-old** chips, **only 12%** of chips (24/204) are vulnerable
- Within **DDR3-new** chips, **65%** of chips (148/228) are vulnerable

Newer DRAM chips are more vulnerable to RowHammer

# 2. Data Pattern Dependence

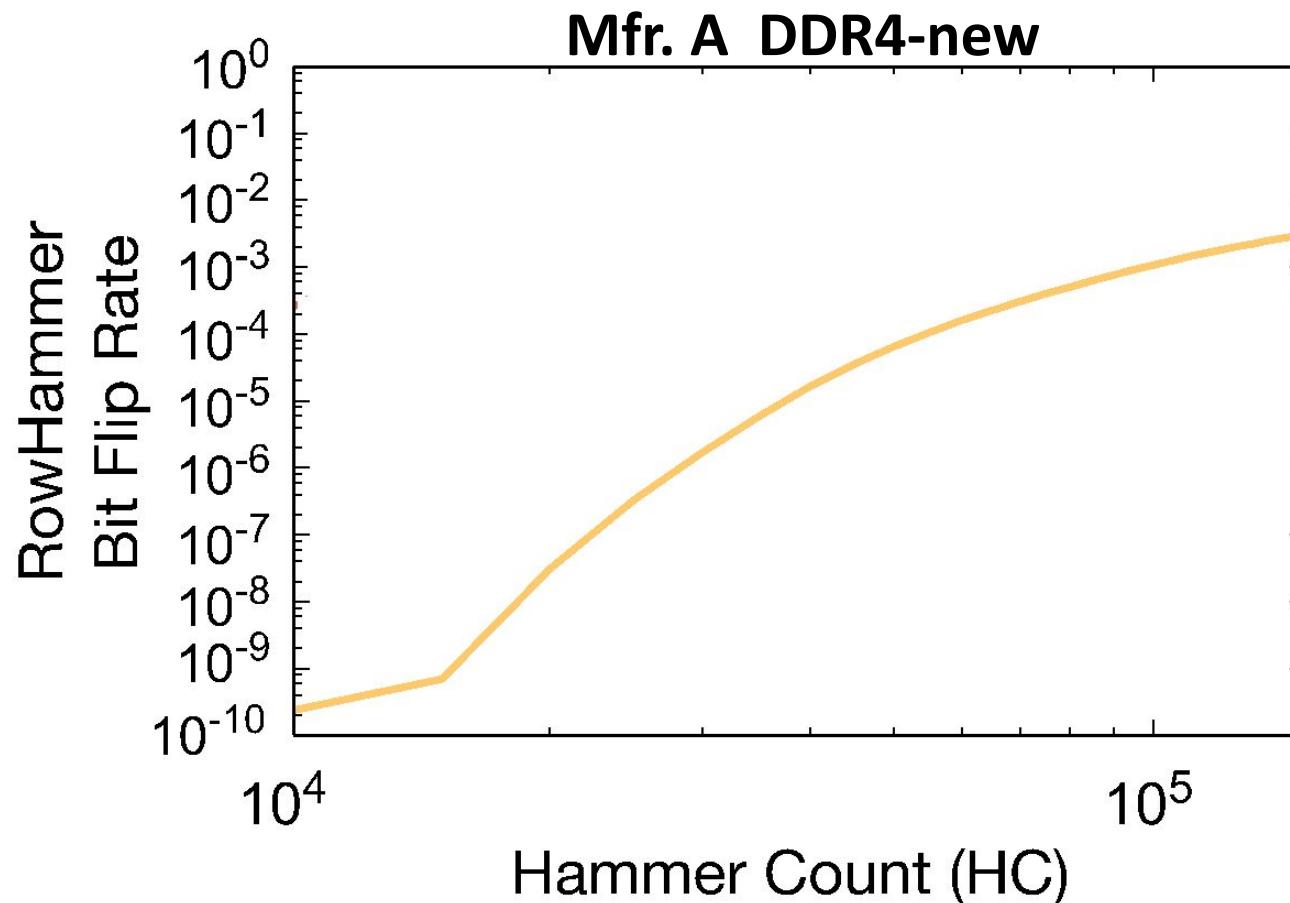
*Q. Are some data patterns more effective in inducing RowHammer bit flips?*

- We test **several data patterns** typically examined in prior work to identify the worst-case data pattern
- The worst-case data pattern is **consistent across chips** of the same manufacturer and DRAM type-node configuration
- We use the **worst-case data pattern** per DRAM chip to characterize each chip at **worst-case conditions** and **minimize the extensive testing time**

[More detail and figures in paper]

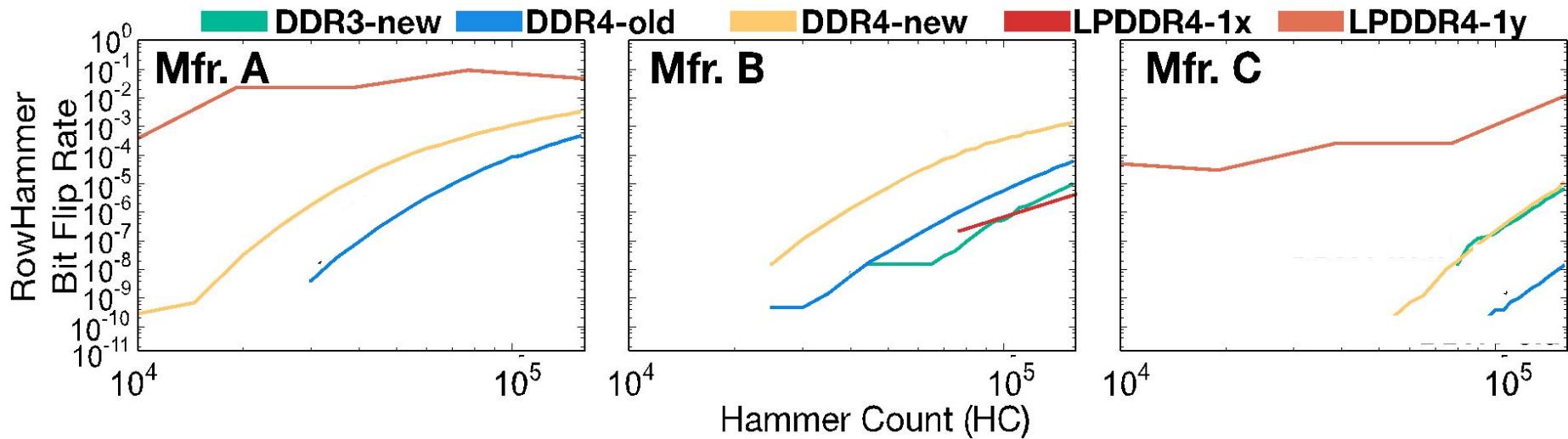
# 3. Hammer Count (HC) Effects

*Q. How does the Hammer Count affect the number of bit flips induced?*



Hammer Count = 2 Accesses,  
one to each adjacent row of victim

# 3. Hammer Count (HC) Effects

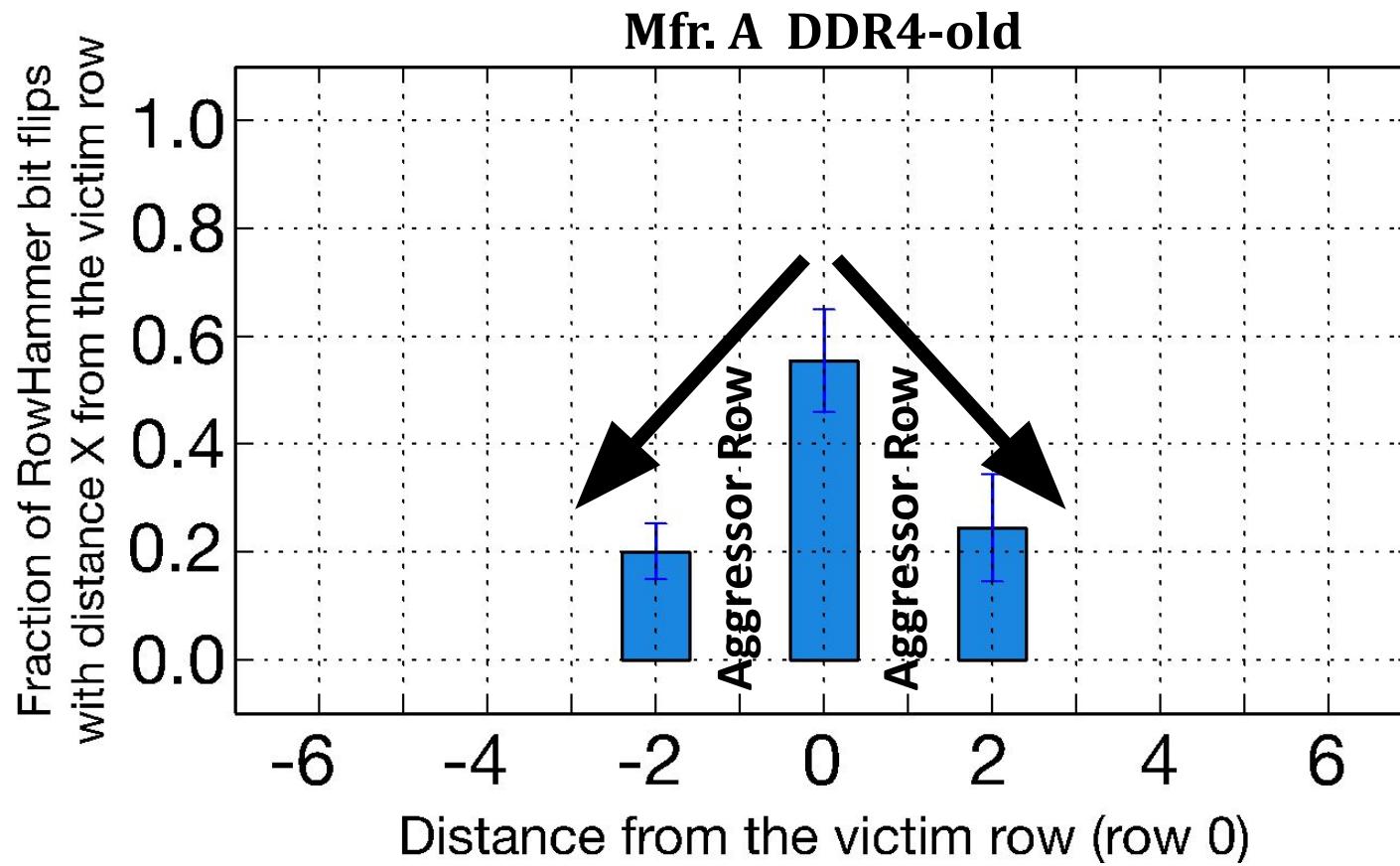


RowHammer bit flip rates **increase**  
when going **from old to new** DDR4 technology node generations

**RowHammer bit flip rates (i.e., RowHammer vulnerability)  
increase with technology node generation**

# 4. Spatial Effects: Row Distance

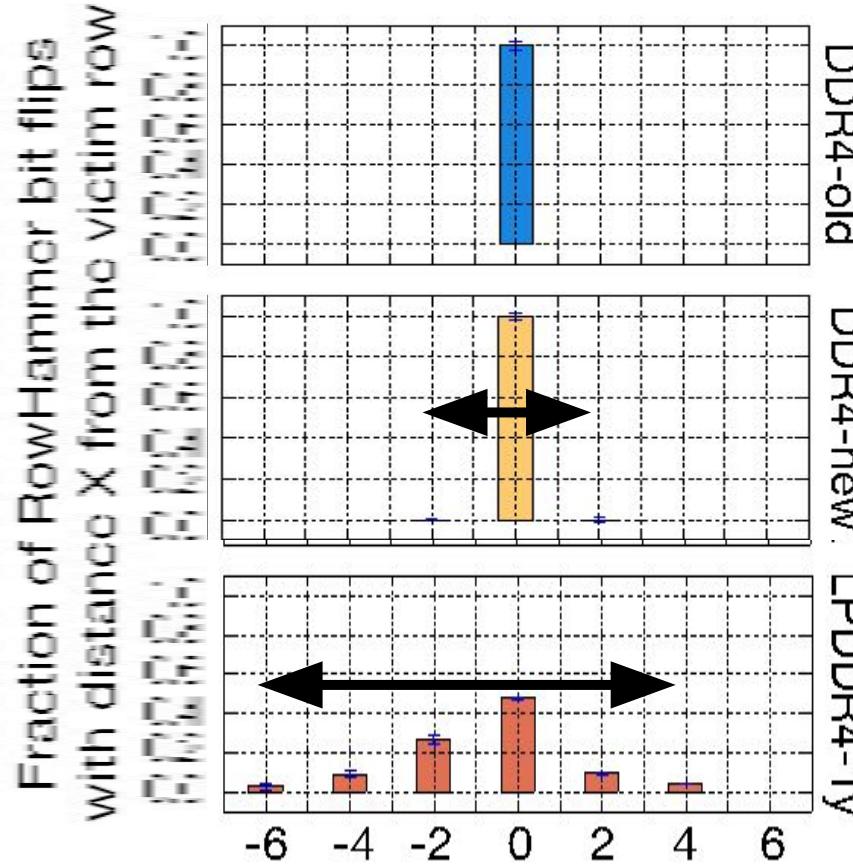
*Q. Where do RowHammer bit flips occur relative to aggressor rows?*



The number of RowHammer bit flips that occur in a given row decreases as the distance from the **victim row (row 0)** increases.

# 4. Spatial Effects: Row Distance

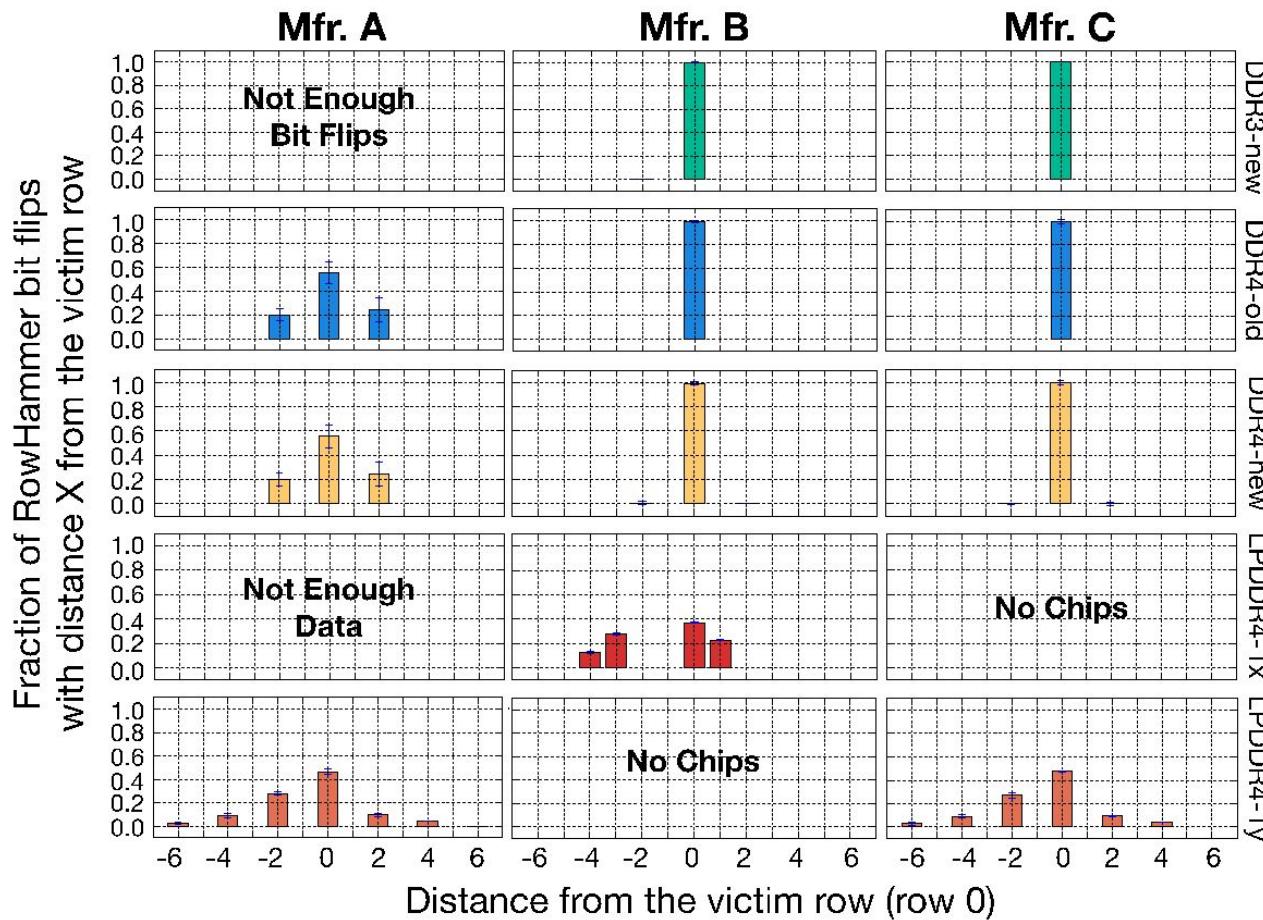
We normalize data by inducing a bit flip rate of  $10^{-6}$  in each chip



Chips of newer DRAM technology nodes can exhibit RowHammer bit flips 1) in **more rows** and 2) **farther away** from the victim row.

# 4. Spatial Effects: Row Distance

We plot this data for each DRAM type-node configuration per manufacturer

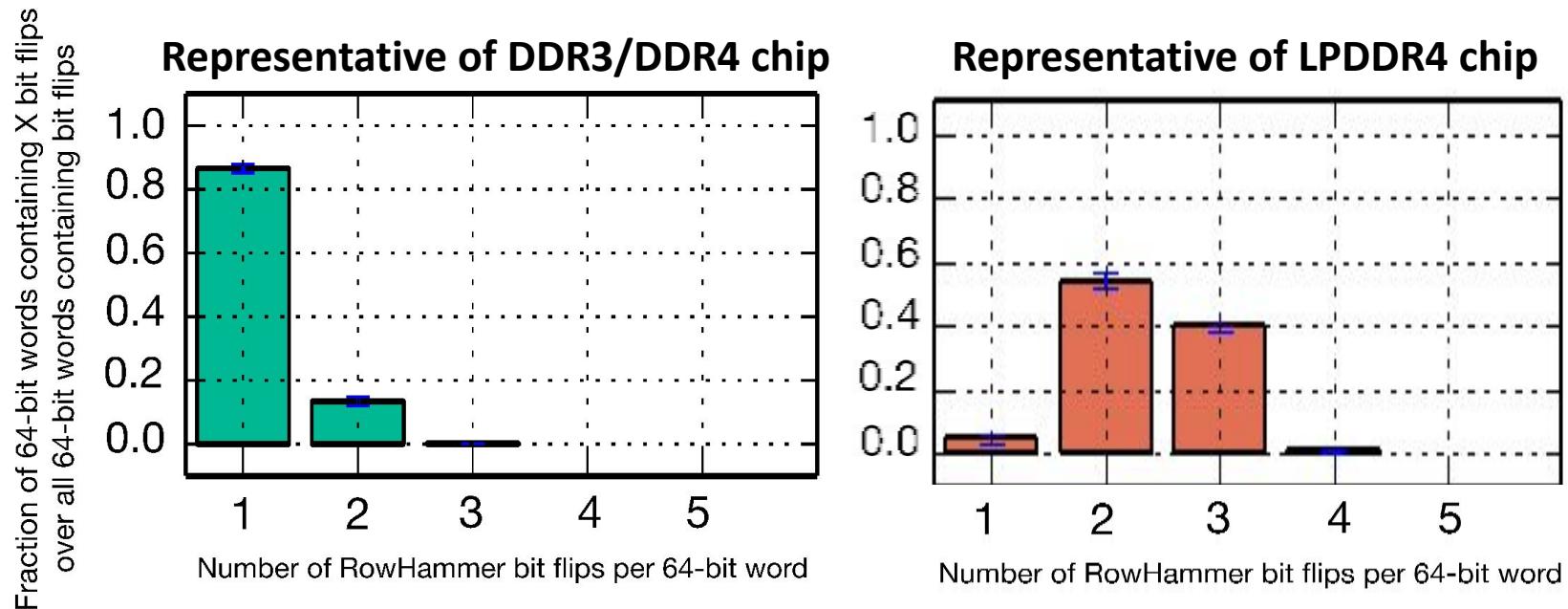


[More analysis in the paper]

# 4. Spatial Distribution of Bit Flips

*Q. How are RowHammer bit flips spatially distributed across a chip?*

We normalize data by inducing a bit flip rate of  $10^{-6}$  in each chip

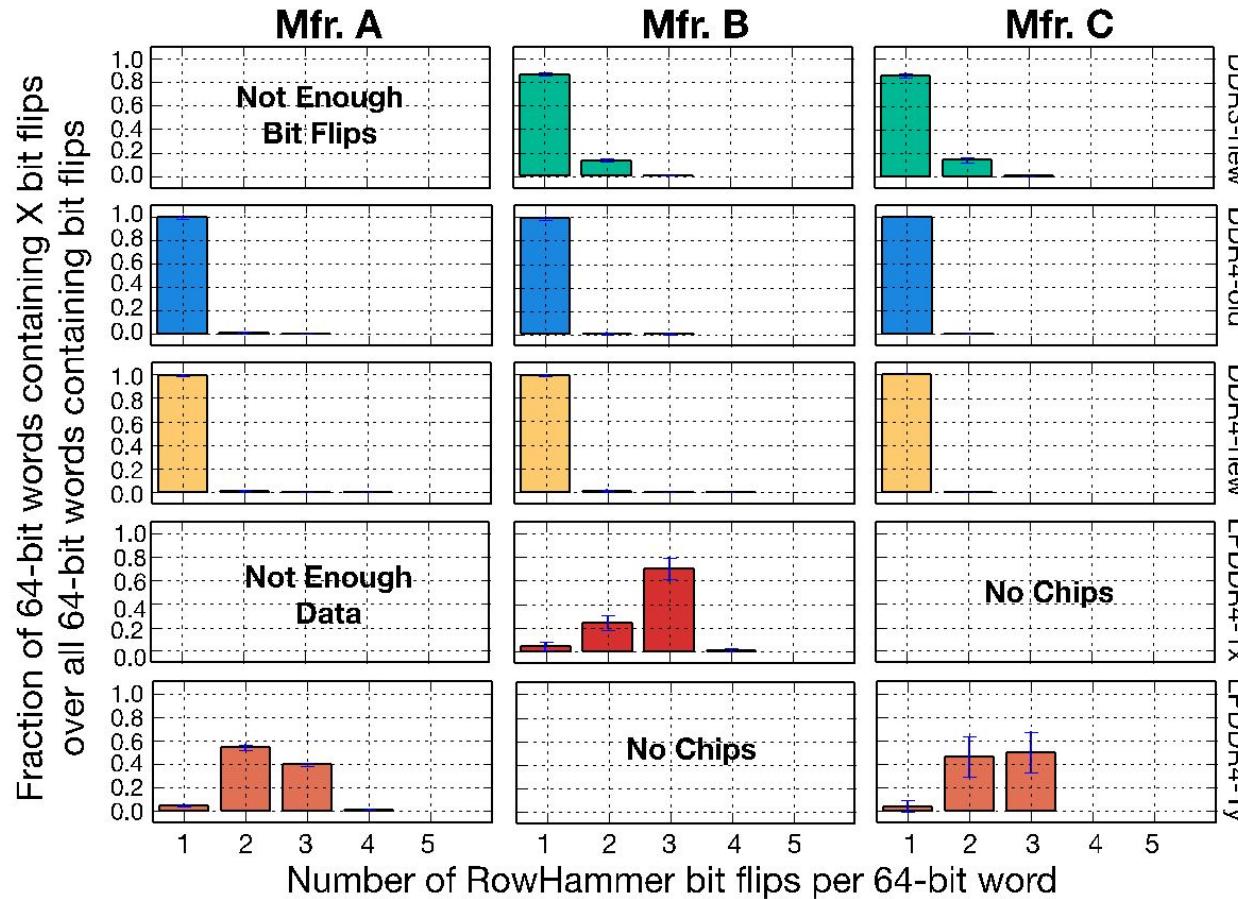


The distribution of RowHammer bit flip density per word  
**changes significantly in LPDDR4 chips** from other DRAM types

At a bit flip rate of  $10^{-6}$ , a 64-bit word can contain up to **4 bit flips**.  
Even at this very low bit flip rate, a **very strong ECC** is required

# 4. Spatial Distribution of Bit Flips

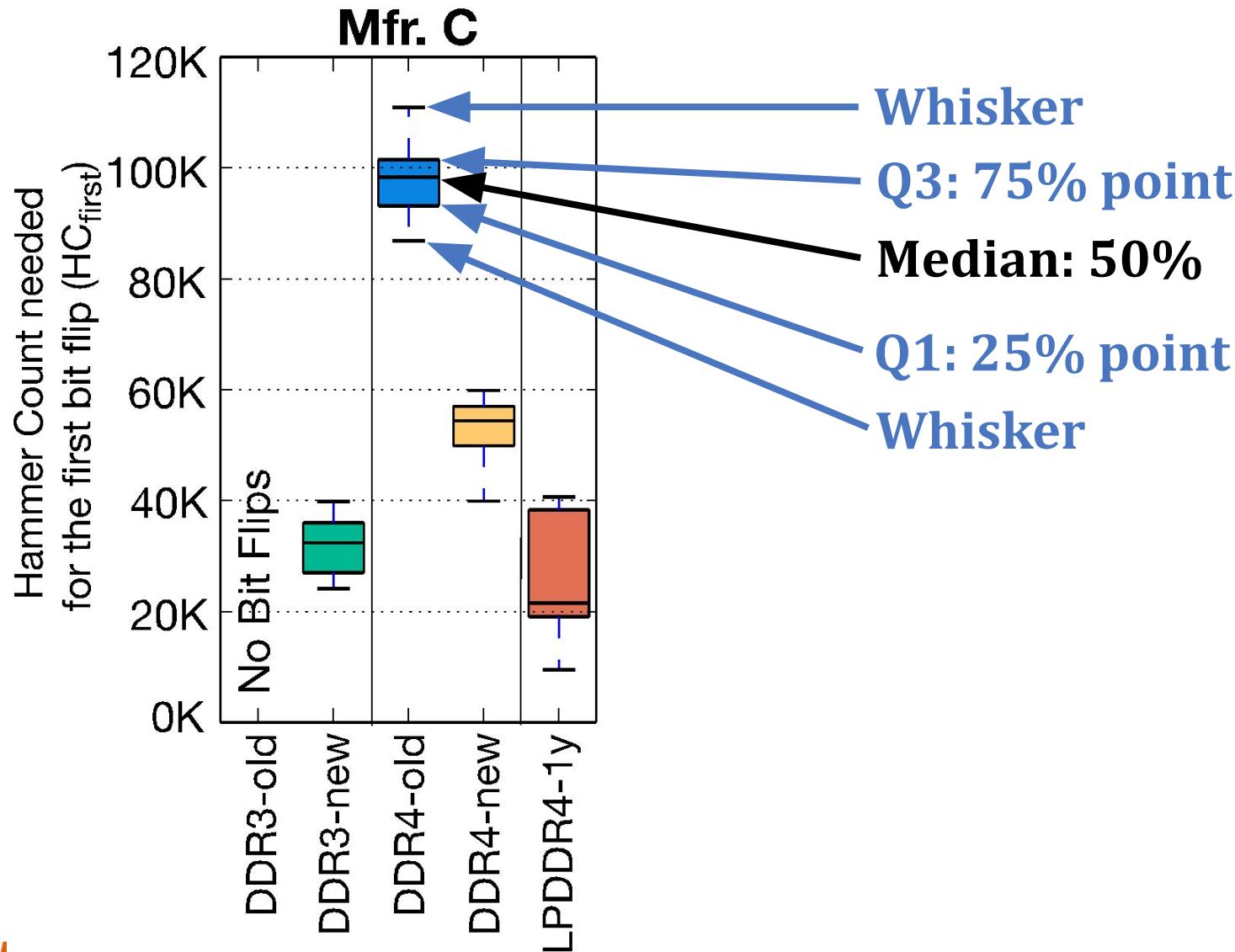
We plot this data for each DRAM type-node configuration per manufacturer



[More analysis in the paper]

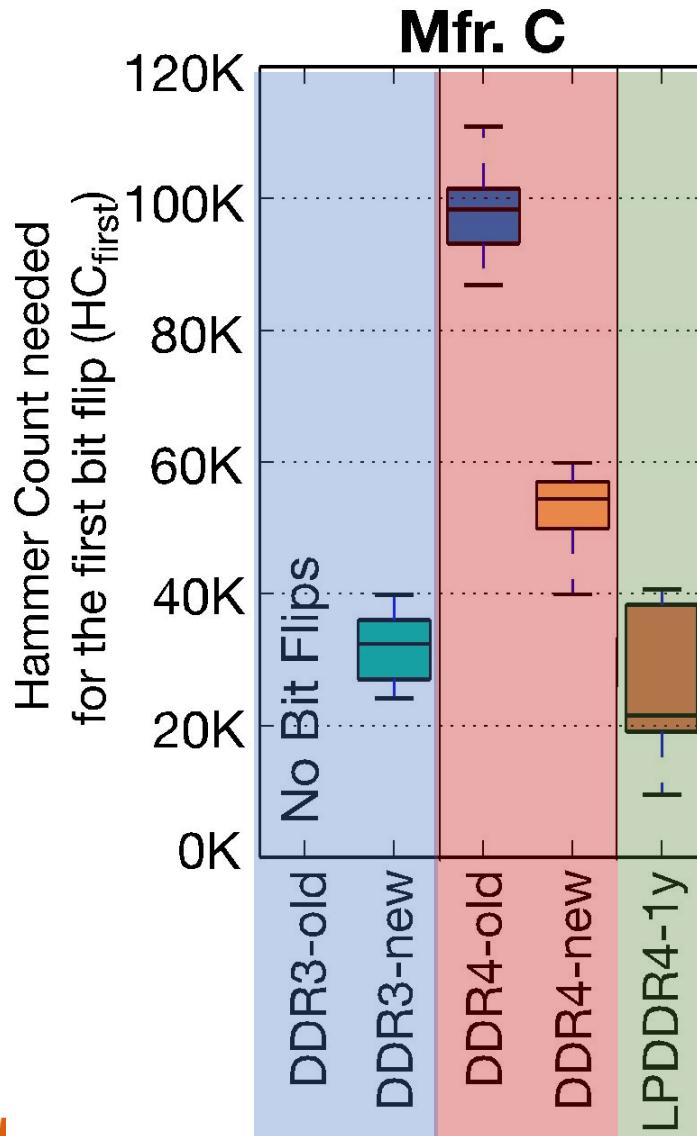
# 5. First RowHammer Bit Flips per Chip

*What is the minimum Hammer Count required to cause bit flips ( $HC_{first}$ )?*



# 5. First RowHammer Bit Flips per Chip

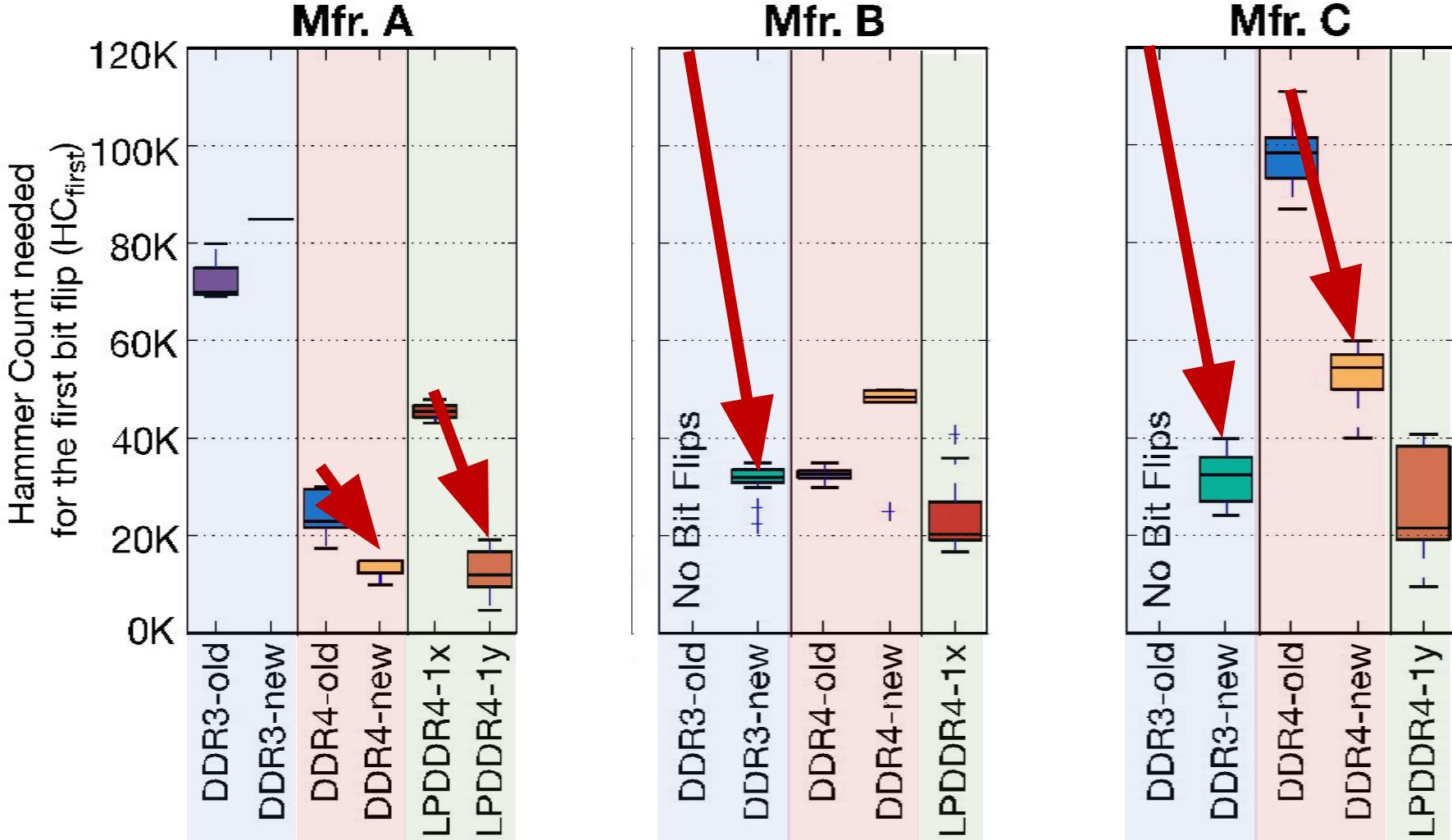
*What is the minimum Hammer Count required to cause bit flips ( $HC_{first}$ )?*



We note the different DRAM types on the x-axis:  
**DDR3, DDR4, LPDDR4.**

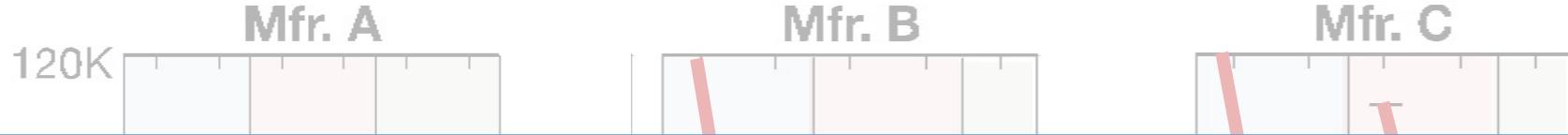
We focus on trends across chips of the same DRAM type to draw conclusions

# 5. First RowHammer Bit Flips per Chip



Newer chips from a given DRAM manufacturer  
**more** vulnerable to RowHammer

# 5. First RowHammer Bit Flips per Chip



In a DRAM type,  $HC_{first}$  reduces significantly from old to new chips, i.e., DDR3: 69.2k to 22.4k, DDR4: 17.5k to 10k, LPDDR4: 16.8k to 4.8k



There are chips whose weakest cells fail after only 4800 hammers



Newer chips from a given DRAM manufacturer  
more vulnerable to RowHammer

# Key Takeaways from 1580 Chips

- Chips of newer DRAM technology nodes are **more vulnerable** to RowHammer
- There are chips today whose weakest cells fail after **only 4800 hammers**
- Chips of newer DRAM technology nodes can exhibit RowHammer bit flips 1) in **more rows** and 2) **farther away** from the victim row.

# Evaluation Methodology

- **Cycle-level simulator:** Ramulator [Kim+, CAL'15]

<https://github.com/CMU-SAFARI/ramulator>

- 4GHz, 4-wide, 128 entry instruction window
- 48 8-core workload mixes randomly drawn from SPEC CPU2006 (**10 < MPKI < 740**)

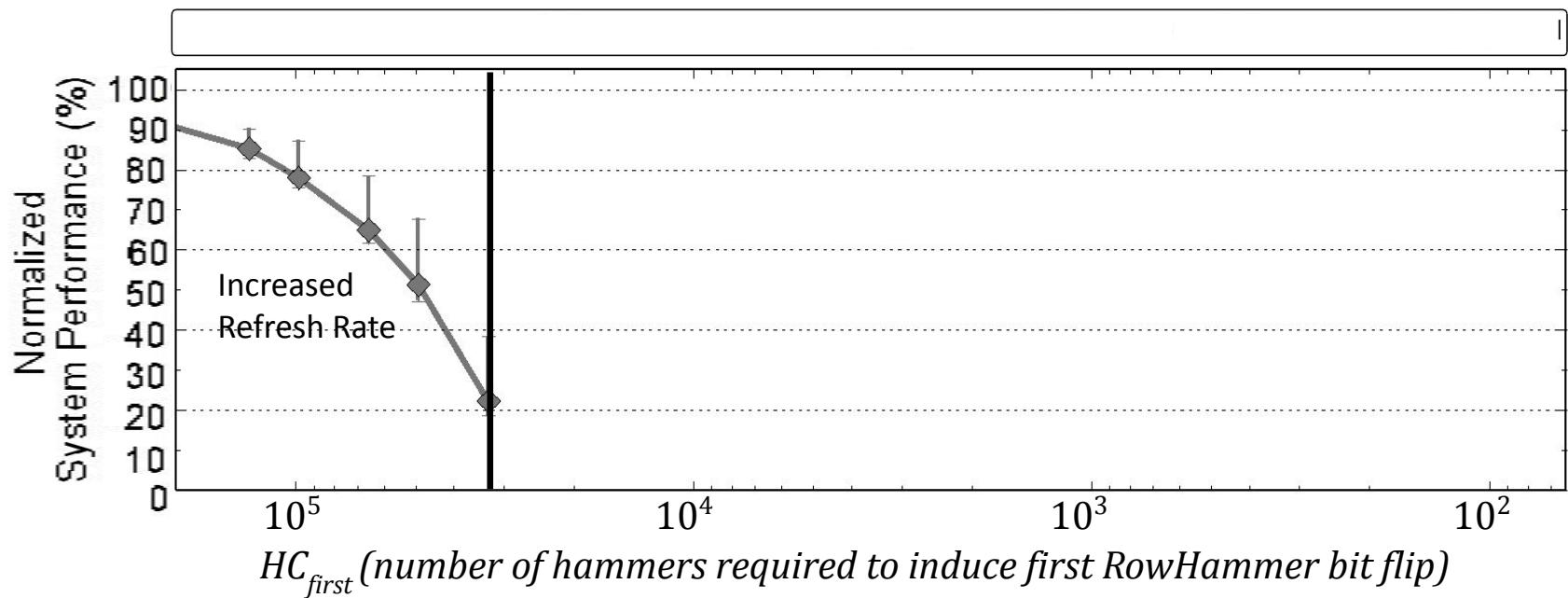
- **Metrics to evaluate mitigation mechanisms**

1. ***DRAM Bandwidth Overhead:*** fraction of total system DRAM bandwidth consumption from mitigation mechanism
2. ***Normalized System Performance:*** normalized weighted speedup to a 100% baseline

# Evaluation Methodology

- We evaluate **five** state-of-the-art mitigation mechanisms:
  - **Increased Refresh Rate** [Kim+, ISCA'14]
  - **PARA** [Kim+, ISCA'14]
  - **ProHIT** [Son+, DAC'17]
  - **MRLoc** [You+, DAC'19]
  - **TWiCe** [Lee+, ISCA'19]
- and **one** ideal refresh-based mitigation mechanism:
  - **Ideal**
- **More detailed descriptions in the paper on:**
  - Descriptions of mechanisms in our paper and the original publications
  - How we scale each mechanism to more vulnerable DRAM chips (lower  $\text{HC}_{\text{first}}$ )

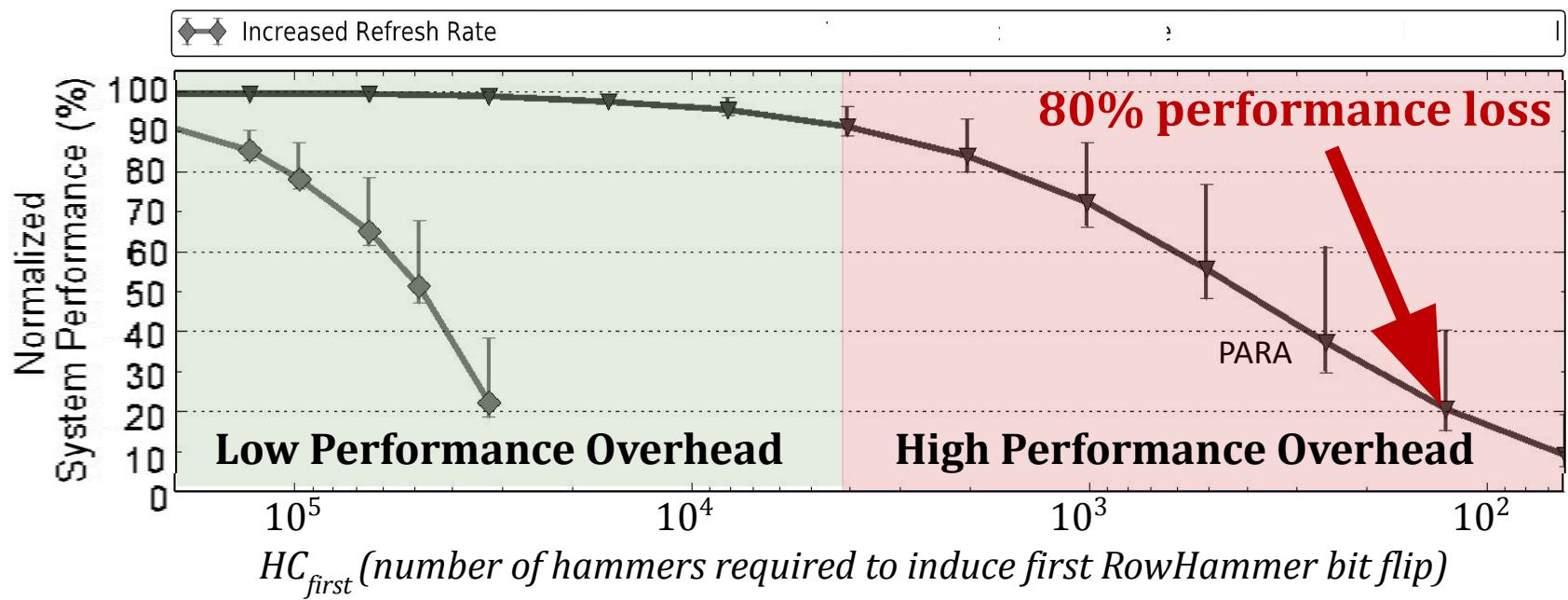
# Mitigation Mech. Eval. (Increased Refresh)



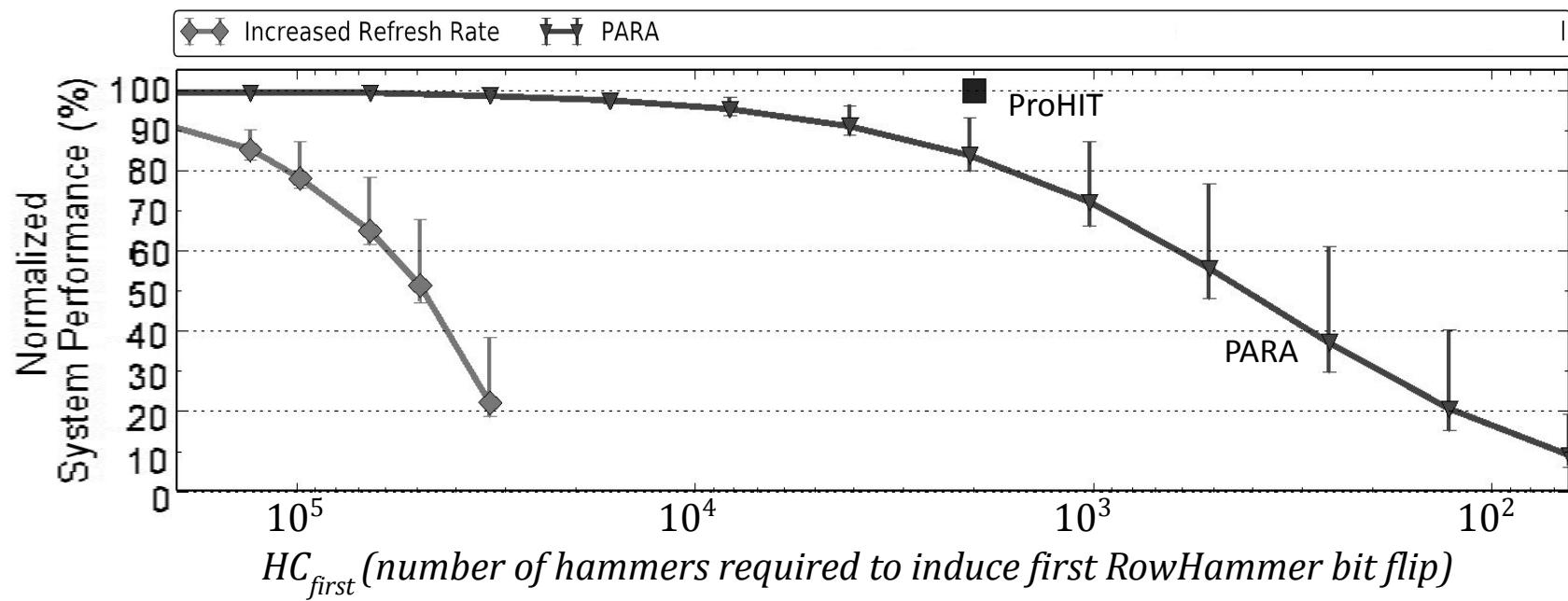
**Substantial overhead for high  $HC_{first}$  values.**

This mechanism does not support  $HC_{first} < 32k$   
due to the **prohibitively high refresh rates required**

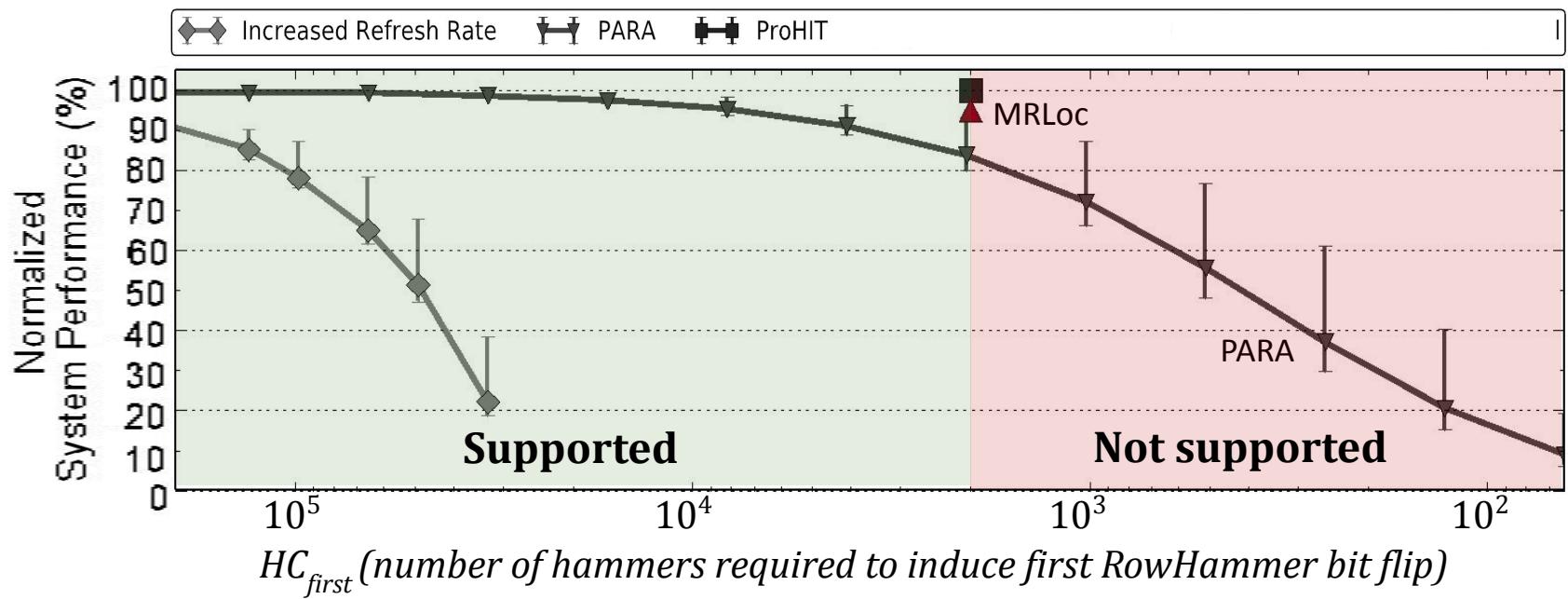
# Mitigation Mechanism Evaluation (PARA)



# Mitigation Mechanism Evaluation (ProHIT)

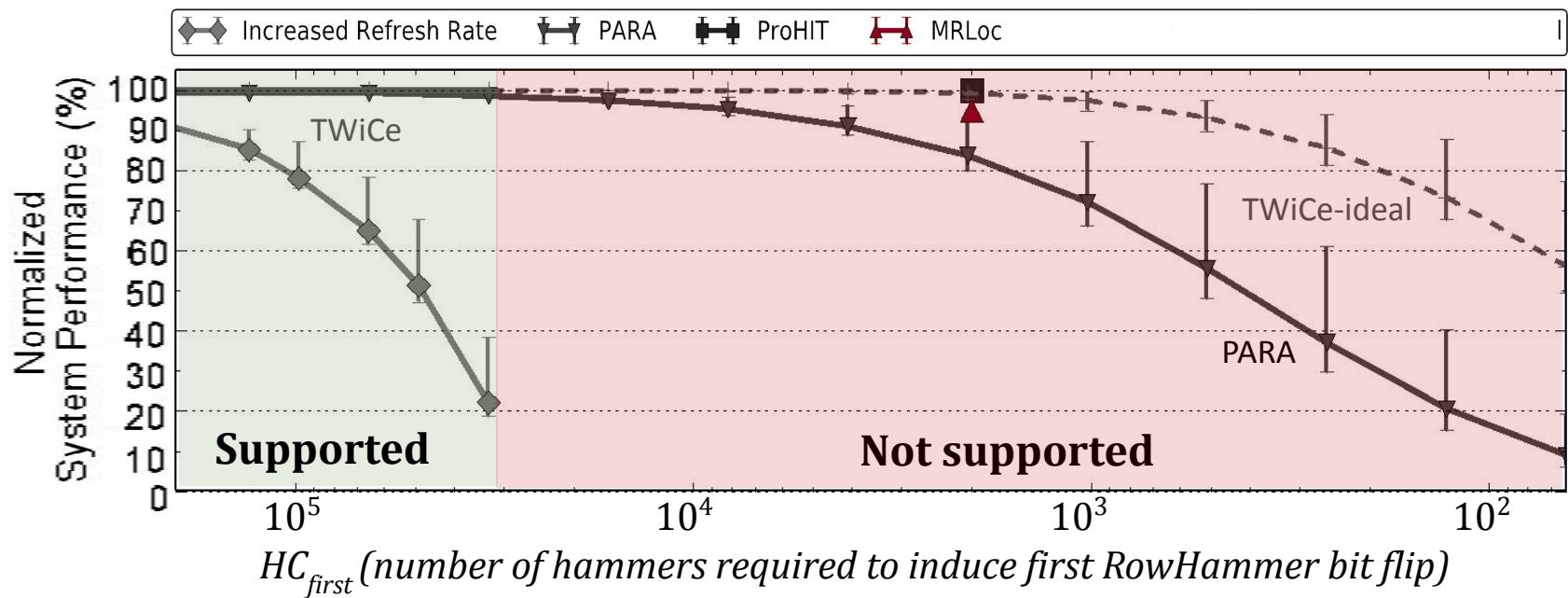


# Mitigation Mechanism Evaluation (MRLoc)



Models for scaling ProHIT and MRLoc for  $HC_{first} < 2k$   
are not provided and how to do so is not intuitive

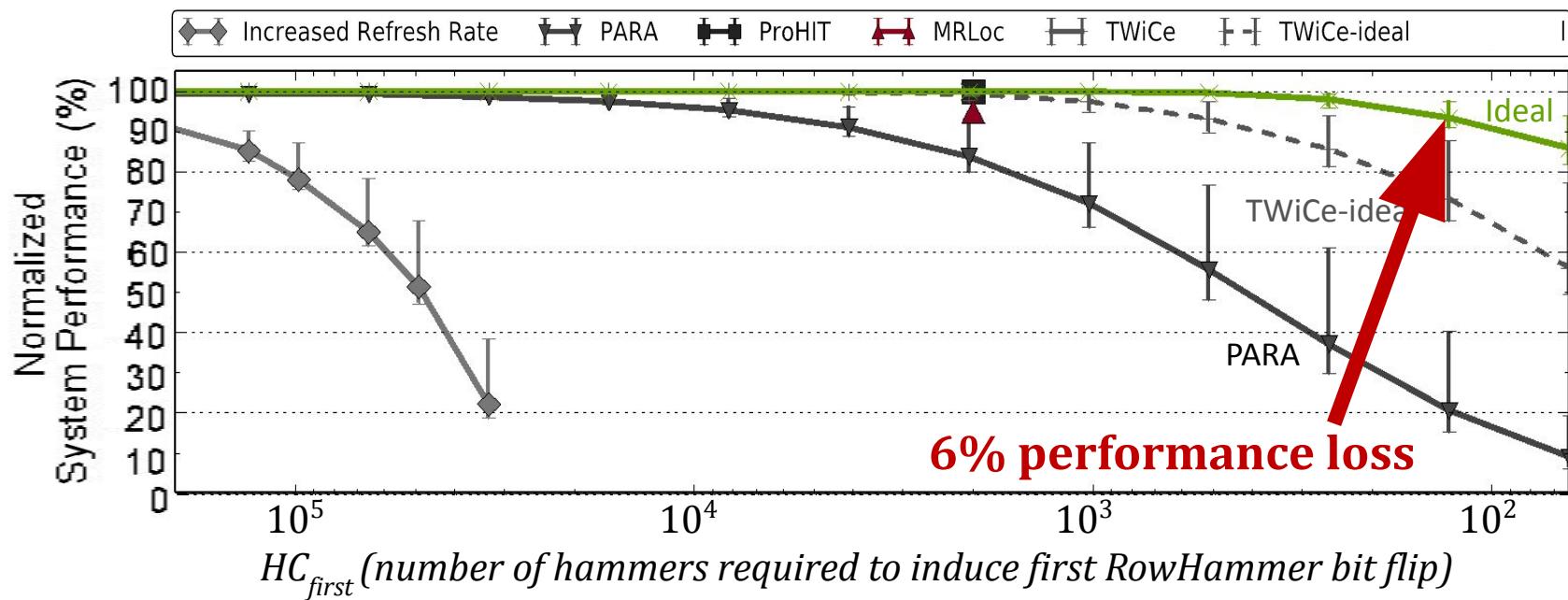
# Mitigation Mechanism Evaluation (TWiCe)



**TWiCe does not support  $HC_{first} < 32k$ .**

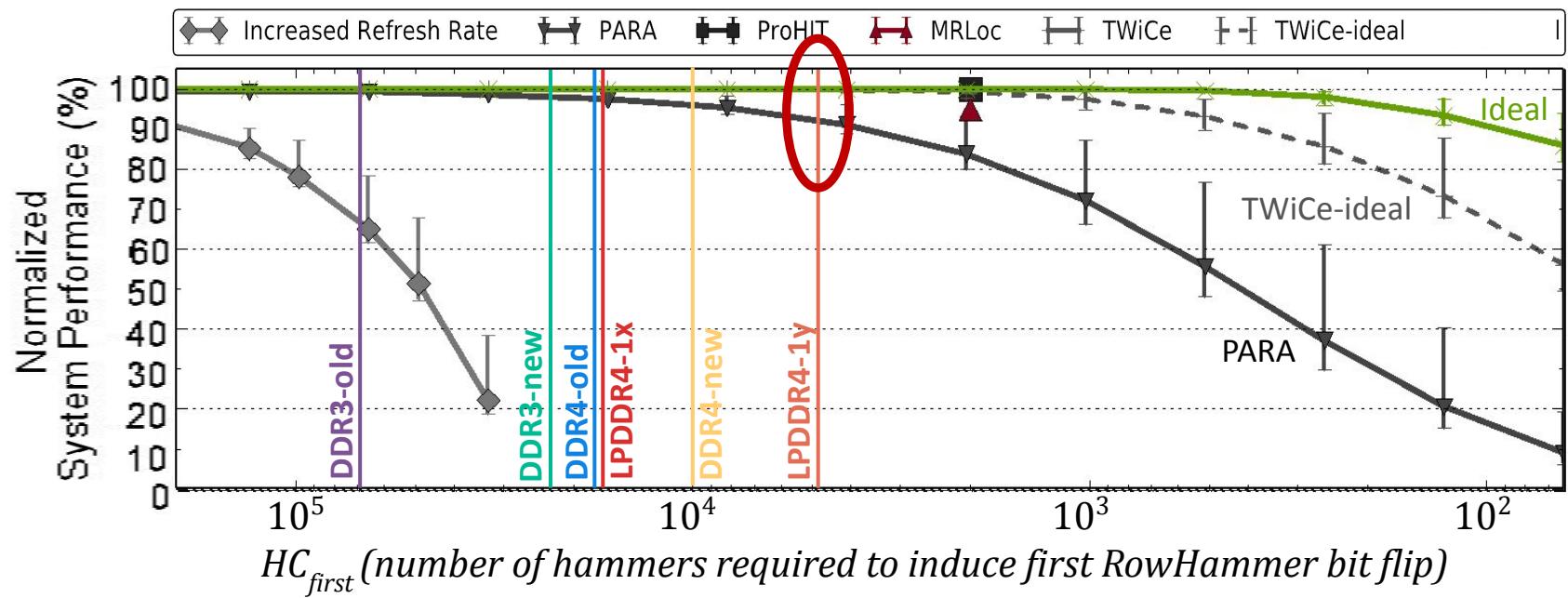
We evaluate an **ideal scalable version (TWiCe-ideal)**  
assuming it solves **two critical design issues**

# Mitigation Mechanism Evaluation (Ideal)



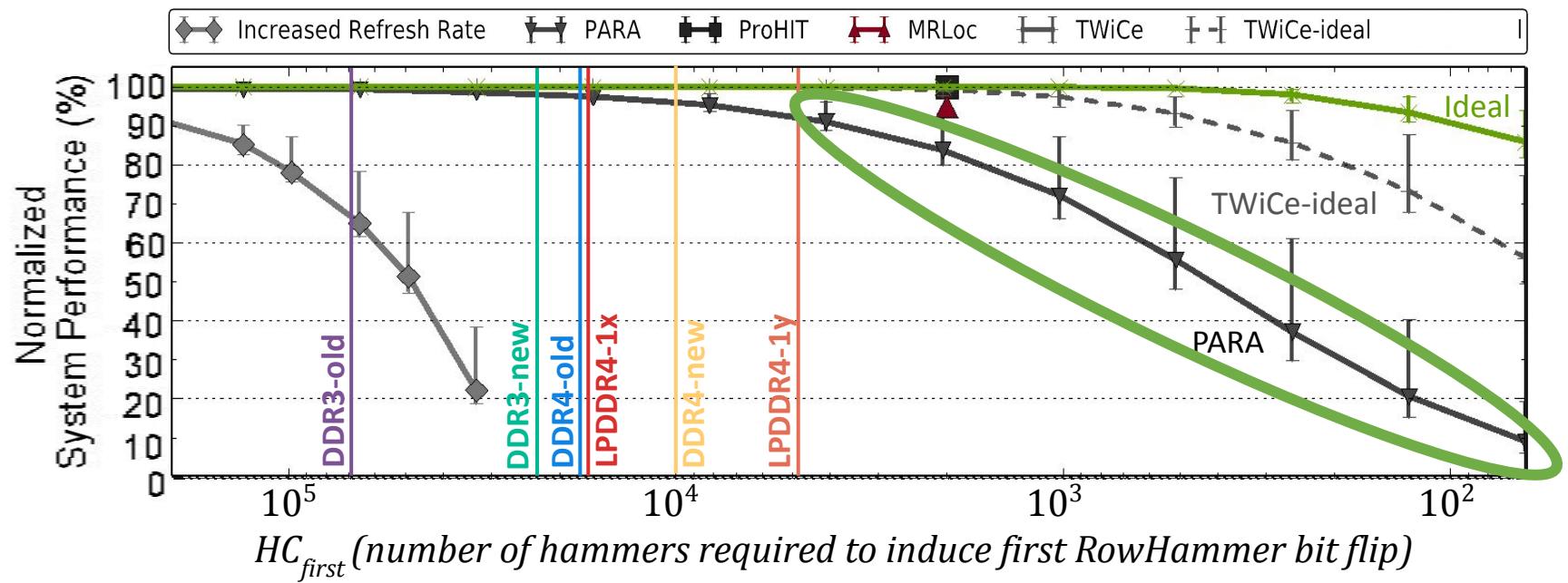
Ideal mechanism issues a refresh command  
to a row only right before the row  
can potentially experience a RowHammer bit flip

# Mitigation Mechanism Evaluation



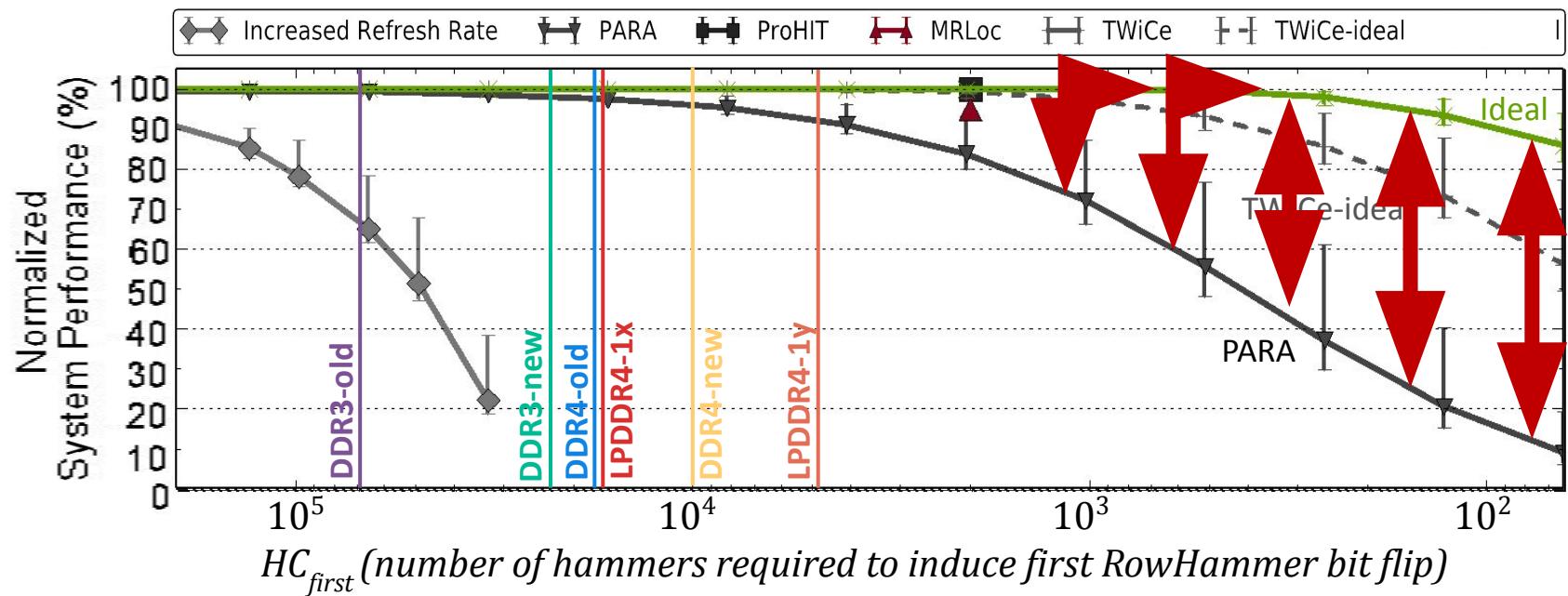
**PARA, ProHIT, and MRLoc mitigate RowHammer bit flips in **worst chips** today with reasonable system performance (92%, 100%, 100%)**

# Mitigation Mechanism Evaluation



Only PARA's design scales to low  $HC_{first}$  values  
but has very low normalized system performance

# Mitigation Mechanism Evaluation



Ideal mechanism is significantly better than any existing mechanism for  $HC_{first} < 1024$

Significant opportunity for developing a RowHammer solution with low performance overhead that supports low  $HC_{first}$

# Key Takeaways from Mitigation Mechanisms

- Existing RowHammer mitigation mechanisms can prevent RowHammer attacks with **reasonable system performance overhead** in DRAM chips today
- Existing RowHammer mitigation mechanisms **do not scale well** to DRAM chips more vulnerable to RowHammer
- There is still **significant opportunity** for developing a mechanism that is **scalable with low overhead**

# Additional Details in the Paper

- Single-cell RowHammer bit flip probability
- More details on our **data pattern dependence** study
- Analysis of **Error Correcting Codes (ECC)** in mitigating RowHammer bit flips
- Additional **observations** on our data
- **Methodology details** for characterizing DRAM
- Further discussion on comparing data across different infrastructures
- **Discussion on scaling** each mitigation mechanism

# RowHammer Solutions Going Forward

Two promising directions for new RowHammer solutions:

## 1. DRAM-system cooperation

- We believe the DRAM and system should cooperate more to provide a **holistic** solution can prevent RowHammer at **low cost**

## 2. Profile-guided

- Accurate **profile of RowHammer-susceptible cells** in DRAM provides a powerful substrate for building **targeted** RowHammer solutions, e.g.:
  - Only increase the refresh rate for rows containing RowHammer-susceptible cells
- A **fast and accurate** profiling mechanism is a key research challenge for developing low-overhead and scalable RowHammer solutions

# Conclusion

- We characterized **1580 DRAM** chips of different DRAM types, technology nodes, and manufacturers.
- We studied **five** state-of-the-art RowHammer mitigation mechanisms and an ideal refresh-based mechanism
- We made **two key observations**
  1. **RowHammer is getting much worse.** It takes much fewer hammers to induce RowHammer bit flips in newer chips
    - e.g., **DDR3: 69.2k to 22.4k, DDR4: 17.5k to 10k, LPDDR4: 16.8k to 4.8k**
  2. **Existing mitigation mechanisms do not scale** to DRAM chips that are more vulnerable to RowHammer
    - e.g., 80% performance loss when the hammer count to induce the first bit flip is 128
- We **conclude** that it is **critical** to do more research on RowHammer and develop scalable mitigation mechanisms to prevent RowHammer in future systems

# *Revisiting RowHammer*

## *An Experimental Analysis of Modern Devices and Mitigation Techniques*

Jeremie S. Kim

Minesh Patel

A. Giray Yağlıkçı

Hasan Hassan

Roknoddin Azizi

Lois Orosa

Onur Mutlu

**SAFARI**

# Revisiting RowHammer in 2020 (I)

---

- Jeremie S. Kim, Minesh Patel, A. Giray Yaglikci, Hasan Hassan, Roknoddin Azizi, Lois Orosa, and Onur Mutlu,

**"Revisiting RowHammer: An Experimental Analysis of Modern Devices and Mitigation Techniques"**

*Proceedings of the 47th International Symposium on Computer Architecture (ISCA), Valencia, Spain, June 2020.*

[Slides (pptx) (pdf)]

[Lightning Talk Slides (pptx) (pdf)]

[Talk Video (20 minutes)]

[Lightning Talk Video (3 minutes)]

## Revisiting RowHammer: An Experimental Analysis of Modern DRAM Devices and Mitigation Techniques

Jeremie S. Kim<sup>§†</sup>

Minesh Patel<sup>§</sup>

A. Giray Yağlıkçı<sup>§</sup>

Hasan Hassan<sup>§</sup>

Roknoddin Azizi<sup>§</sup>

Lois Orosa<sup>§</sup>

Onur Mutlu<sup>§†</sup>

<sup>§</sup>*ETH Zürich*

<sup>†</sup>*Carnegie Mellon University*

# Future Memory Reliability/Security Challenges

# Future of Main Memory

---

- DRAM is becoming less reliable □ more vulnerable

# Large-Scale Failure Analysis of DRAM Chips

---

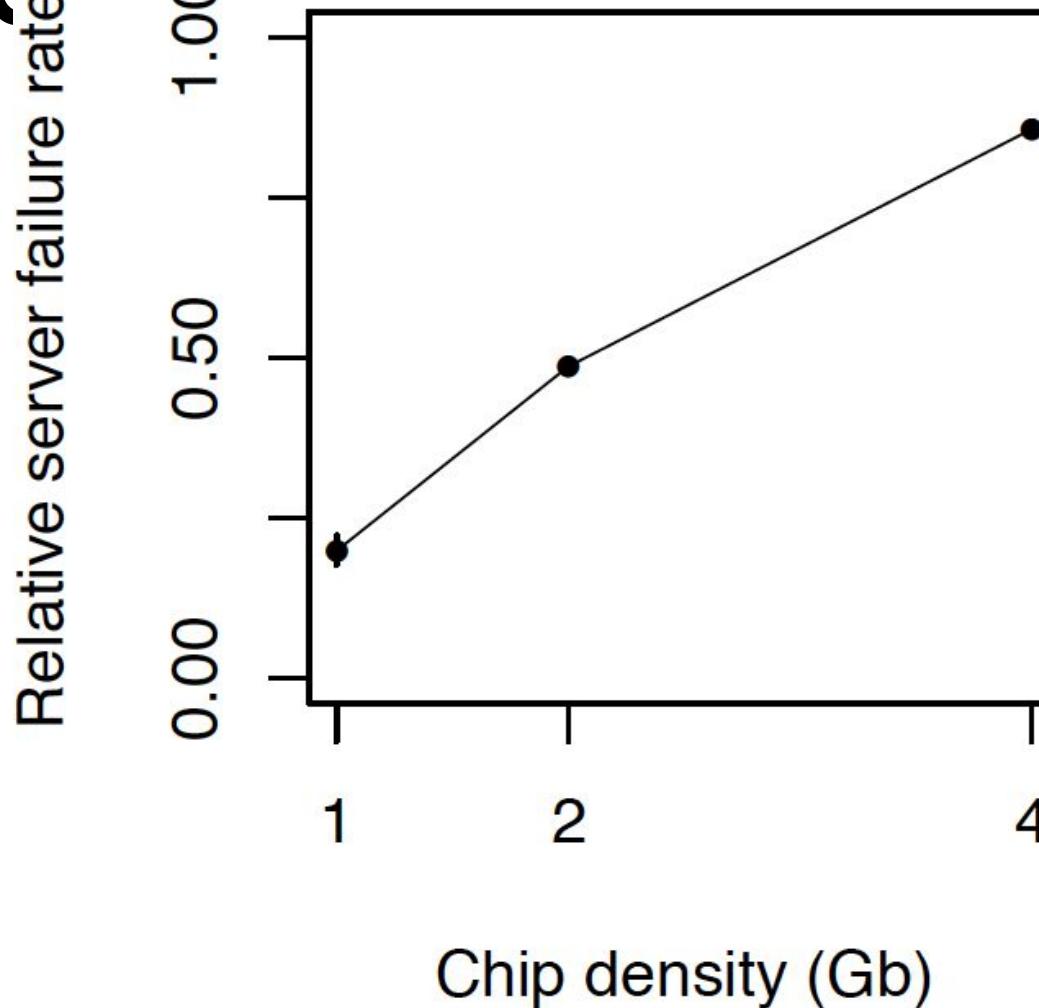
- Analysis and modeling of memory errors found in all of Facebook's server fleet
- Justin Meza, Qiang Wu, Sanjeev Kumar, and Onur Mutlu,  
**"Revisiting Memory Errors in Large-Scale Production Data Centers: Analysis and Modeling of New Trends from the Field"**  
*Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Rio de Janeiro, Brazil, June 2015.  
[Slides (pptx)] [pdf] [DRAM Error Model]

## Revisiting Memory Errors in Large-Scale Production Data Centers: Analysis and Modeling of New Trends from the Field

Justin Meza   Qiang Wu \*   Sanjeev Kumar \*   Onur Mutlu  
Carnegie Mellon University   \* Facebook, Inc.

# DRAM Reliability

Reducing chip density



*Intuition:  
quadratic  
increase in  
capacity*

# Aside: SSD Error Analysis in the Field

---

- First large-scale field study of flash memory errors
- Justin Meza, Qiang Wu, Sanjeev Kumar, and Onur Mutlu,  
**"A Large-Scale Study of Flash Memory Errors in the Field"**  
*Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, Portland, OR, June 2015.  
[Slides (pptx)] [Coverage at ZDNet]

## A Large-Scale Study of Flash Memory Failures in the Field

Justin Meza  
Carnegie Mellon University  
[meza@cmu.edu](mailto:meza@cmu.edu)

Qiang Wu  
Facebook, Inc.  
[qwu@fb.com](mailto:qwu@fb.com)

Sanjeev Kumar  
Facebook, Inc.  
[skumar@fb.com](mailto:skumar@fb.com)

Onur Mutlu  
Carnegie Mellon University  
[onur@cmu.edu](mailto:onur@cmu.edu)

# Future of Main Memory

---

- DRAM is becoming less reliable □ more vulnerable
- Due to difficulties in DRAM scaling, other problems may also appear (or they may be going unnoticed)
- Some errors may already be slipping into the field
  - Read disturb errors (Rowhammer)
  - **Retention errors**
  - Read errors, write errors
  - ...
- These errors can also pose security vulnerabilities

# DRAM Data Retention Time Failures

---

- Determining the data retention time of a cell/row is getting more difficult
- Retention failures may already be slipping into the field

# Analysis of Data Retention Failures [ISCA'13]

---

- Jamie Liu, Ben Jaiyen, Yoongu Kim, Chris Wilkerson, and Onur Mutlu,  
**"An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms"**  
*Proceedings of the 40th International Symposium on Computer Architecture (ISCA)*, Tel-Aviv, Israel, June 2013. [Slides \(ppt\)](#) [Slides \(pdf\)](#)

## An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms

Jamie Liu\*

Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213

jamiel@alumni.cmu.edu

Ben Jaiyen\*

Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213

bjaiyen@alumni.cmu.edu

Yoongu Kim

Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213

yoonguk@ece.cmu.edu

Chris Wilkerson

Intel Corporation  
2200 Mission College Blvd.  
Santa Clara, CA 95054  
chris.wilkerson@intel.com

Onur Mutlu

Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213  
onur@cmu.edu

# Two Challenges to Retention Time Profiling

---

- Data Pattern Dependence (DPD) of retention time
- Variable Retention Time (VRT) phenomenon

<https://www.youtube.com/watch?v=v702wUnaWGE>

# Industry Is Writing Papers About It, Too

## DRAM Process Scaling Challenges

### ❖ Refresh

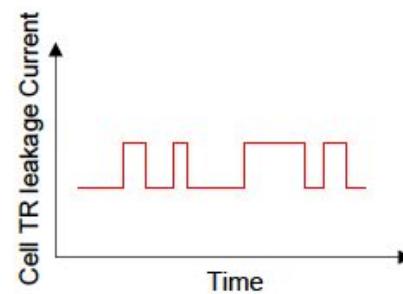
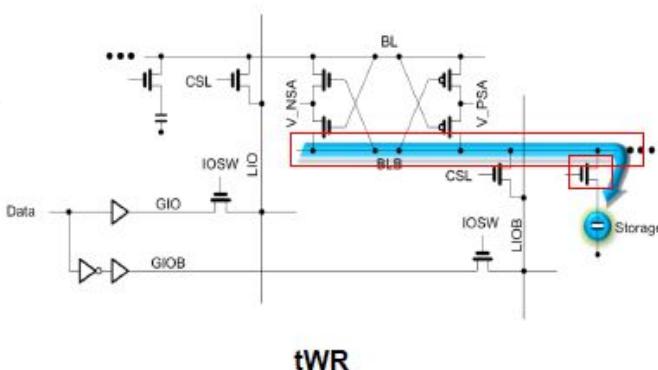
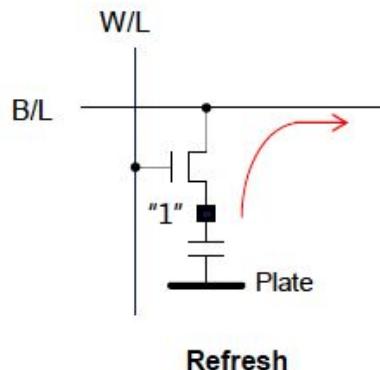
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance
- Leakage current of cell access transistors increasing

### ❖ tWR

- Contact resistance between the cell capacitor and access transistor increasing
- On-current of the cell access transistor decreasing
- Bit-line resistance increasing

### ❖ VRT

- Occurring more frequently with cell capacitance decreasing



VRT



# Industry Is Writing Papers About It, Too

## DRAM Process Scaling Challenges

### ❖ Refresh

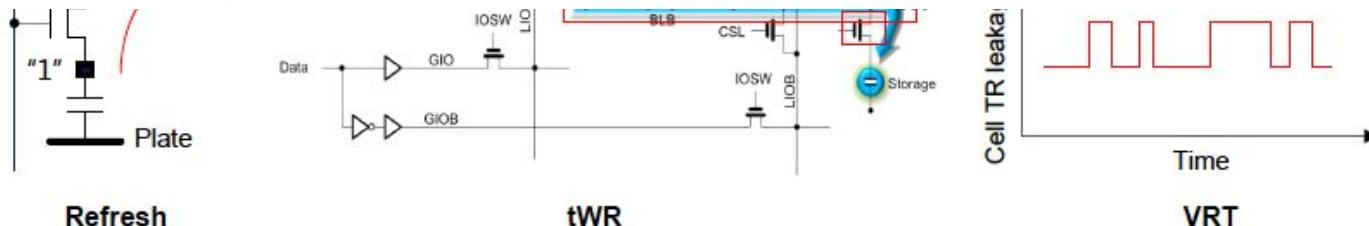
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance

THE MEMORY FORUM 2014

## Co-Architecting Controllers and DRAM to Enhance DRAM Process Scaling

Uksong Kang, Hak-soo Yu, Churoo Park, \*Hongzhong Zheng,  
\*\*John Halbert, \*\*Kuljit Bains, SeongJin Jang, and Joo Sun Choi

*Samsung Electronics, Hwasung, Korea / \*Samsung Electronics, San Jose / \*\*Intel*



# Keeping Future Memory Secure

# How Do We Keep Memory Secure?

---

- DRAM
- Flash memory
- Emerging Technologies
  - Phase Change Memory
  - STT-MRAM
  - RRAM, memristors
  - ...

# Solution Direction: Principled Designs

---

Design fundamentally secure  
computing architectures

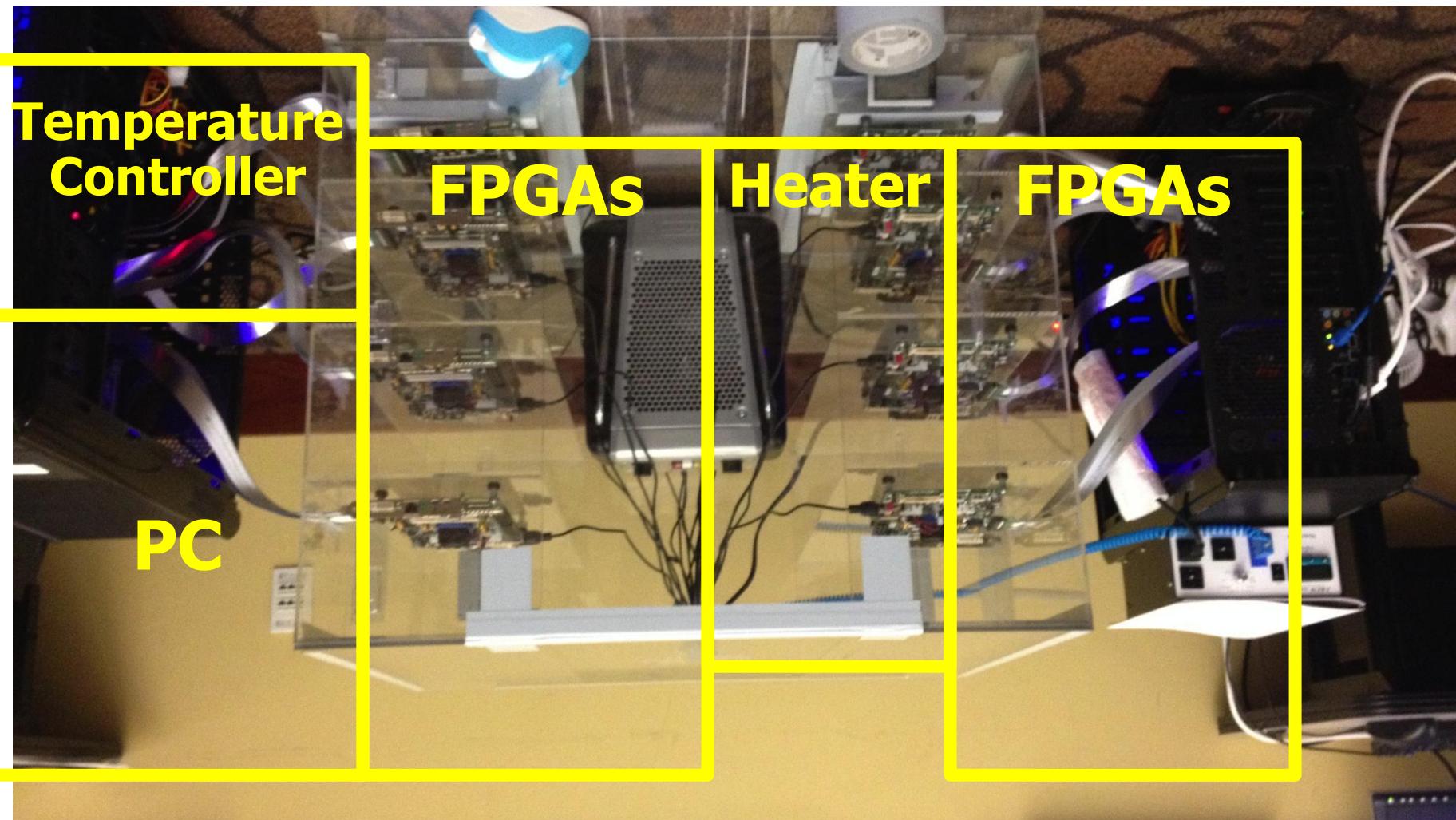
Predict and prevent  
such safety issues

# Architecting Future Memory for Security

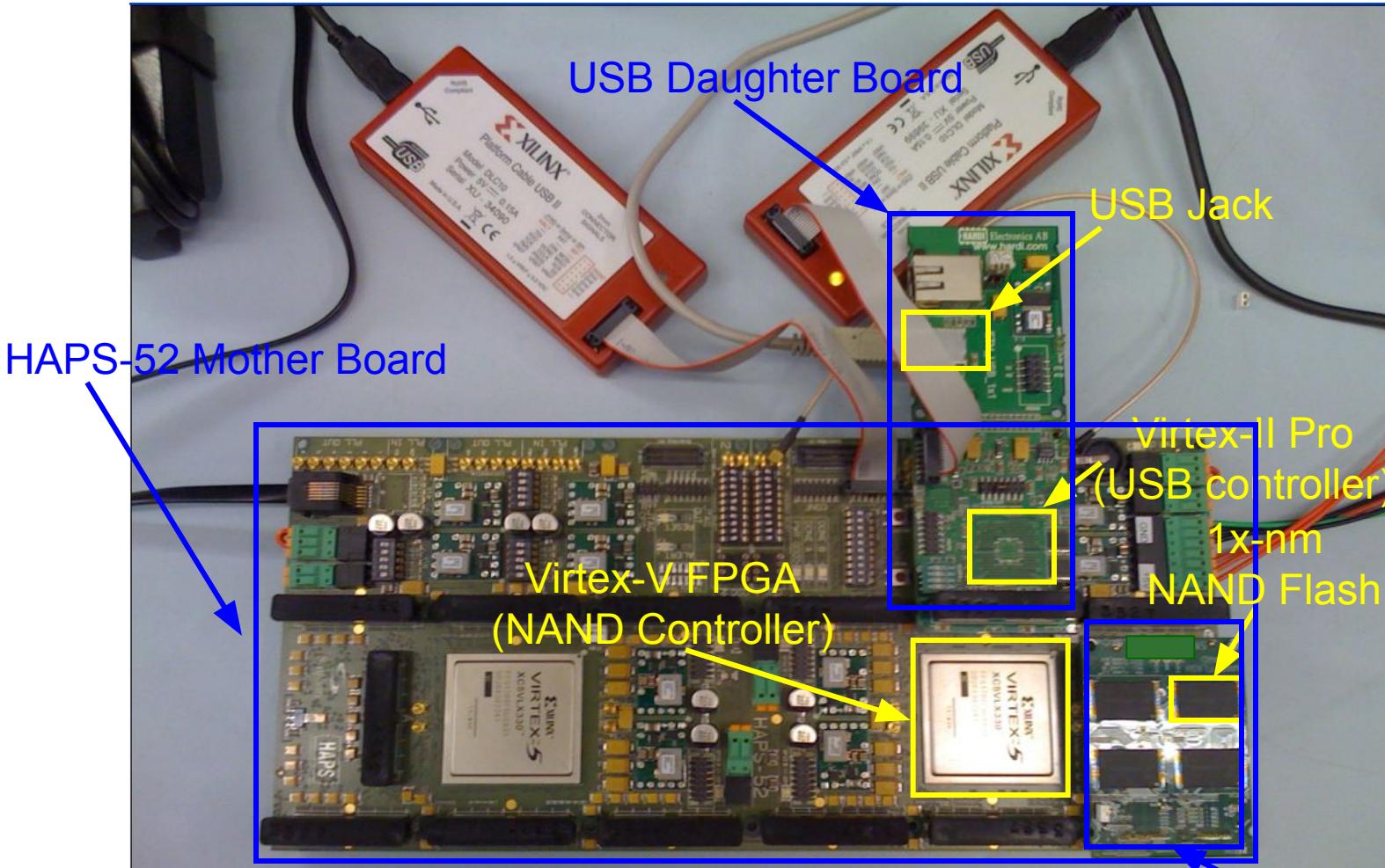
---

- **Understand:** Methods for vulnerability modeling & discovery
  - Modeling and prediction based on real (device) data and analysis
  - Understanding vulnerabilities
  - Developing reliable metrics
- **Architect:** Principled architectures with security as key concern
  - Good partitioning of duties across the stack
  - Cannot give up performance and efficiency
  - Patch-ability in the field
- **Design & Test:** Principled design, automation, (online) testing
  - Design for security
  - High coverage and good interaction with system reliability methods

# Understand and Model with Experiments (DRAM)



# Understand and Model with Experiments (Flash)



[DATE 2012, ICCD 2012, DATE 2013, ITJ 2013, ICCD 2013, SIGMETRICS 2014, HPCA 2015, DSN 2015, MSST 2015, JSAC 2016, HPCA 2017, DFRWS 2017, PIEEE 2017, HPCA 2018, SIGMETRICS 2018]

NAND Daughter Board

# Understanding Flash Memory Reliability

---



*Proceedings of the IEEE, Sept. 2017*

## Error Characterization, Mitigation, and Recovery in Flash-Memory-Based Solid-State Drives

*This paper reviews the most recent advances in solid-state drive (SSD) error characterization, mitigation, and data recovery techniques to improve both SSD's reliability and lifetime.*

By YU CAI, SAUGATA GHOSE, ERICH F. HARATSCH, YIXIN LUO, AND ONUR MUTLU

# Understanding Flash Memory Reliability

---

- Justin Meza, Qiang Wu, Sanjeev Kumar, and Onur Mutlu,  
**"A Large-Scale Study of Flash Memory Errors in the Field"**  
*Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, Portland, OR, June 2015.  
[[Slides \(pptx\)](#)] [[Coverage at ZDNet](#)] [[Coverage on The Register](#)]  
[[Coverage on TechSpot](#)] [[Coverage on The Tech Report](#)]

## A Large-Scale Study of Flash Memory Failures in the Field

Justin Meza  
Carnegie Mellon University  
meza@cmu.edu

Qiang Wu  
Facebook, Inc.  
qwu@fb.com

Sanjeev Kumar  
Facebook, Inc.  
skumar@fb.com

Onur Mutlu  
Carnegie Mellon University  
onur@cmu.edu

# NAND Flash Vulnerabilities [HPCA'17]

HPCA, Feb. 2017

## Vulnerabilities in MLC NAND Flash Memory Programming: Experimental Analysis, Exploits, and Mitigation Techniques

Yu Cai<sup>†</sup>      Saugata Ghose<sup>†</sup>      Yixin Luo<sup>‡‡</sup>      Ken Mai<sup>†</sup>      Onur Mutlu<sup>\$†</sup>      Erich F. Haratsch<sup>‡</sup>

<sup>†</sup>Carnegie Mellon University

<sup>‡</sup>Seagate Technology

<sup>\$</sup>ETH Zürich

*Modern NAND flash memory chips provide high density by storing two bits of data in each flash cell, called a multi level cell (MLC). An MLC partitions the threshold voltage range of a flash cell into four voltage states. When a flash cell is programmed, a high voltage is applied to the cell. Due to parasitic capacitance coupling between flash cells that are physically close to each other, flash cell programming can lead to cell-to-cell program interference, which introduces errors into neighboring flash cells. In order to reduce the impact of cell-to-cell interference on the reliability of MLC NAND flash memory, flash manufacturers adopt a two step programming method, which programs the MLC in two separate steps. First, the flash memory partially programs the least significant bit of the MLC to some intermediate threshold voltage. Second, it programs the most significant bit to bring the MLC up to its full voltage state.*

*In this paper, we demonstrate that two-step programming exposes new reliability and security vulnerabilities. We expe-*

belongs to a different flash memory *page* (the unit of data programmed and read at the same time), which we refer to, respectively, as the least significant bit (LSB) page and the most significant bit (MSB) page [5].

A flash cell is programmed by applying a large voltage on the control gate of the transistor, which triggers charge transfer into the floating gate, thereby increasing the threshold voltage. To precisely control the threshold voltage of the cell, the flash memory uses *incremental step pulse programming* (ISPP) [12, 21, 25, 41]. ISPP applies multiple short pulses of the programming voltage to the control gate, in order to increase the cell threshold voltage by some small voltage amount ( $V_{step}$ ) after each step. Initial MLC designs programmed the threshold voltage in *one shot*, issuing all of the pulses back-to-back to program *both* bits of data at the same time. However, as flash memory scales down, the distance between neighboring flash cells decreases, which

[https://people.inf.ethz.ch/omutlu/pub/flash-memory-programming-vulnerabilities\\_hPCA17.pdf](https://people.inf.ethz.ch/omutlu/pub/flash-memory-programming-vulnerabilities_hPCA17.pdf)

# 3D NAND Flash Reliability I [HPCA'18]

---

- Yixin Luo, Saugata Ghose, Yu Cai, Erich F. Haratsch, and Onur Mutlu,  
**"HeatWatch: Improving 3D NAND Flash Memory Device Reliability by Exploiting Self-Recovery and Temperature-Awareness"**

*Proceedings of the 24th International Symposium on High-Performance Computer Architecture (HPCA), Vienna, Austria, February 2018.*

[Lightning Talk Video]

[Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)]

## **HeatWatch: Improving 3D NAND Flash Memory Device Reliability by Exploiting Self-Recovery and Temperature Awareness**

Yixin Luo<sup>†</sup>      Saugata Ghose<sup>†</sup>      Yu Cai<sup>‡</sup>      Erich F. Haratsch<sup>‡</sup>      Onur Mutlu<sup>§†</sup>  
    <sup>†</sup>*Carnegie Mellon University*      <sup>‡</sup>*Seagate Technology*      <sup>§</sup>*ETH Zürich*

---

# 3D NAND Flash Reliability II [SIGMETRICS'18]

---

- Yixin Luo, Saugata Ghose, Yu Cai, Erich F. Haratsch, and Onur Mutlu,  
**"Improving 3D NAND Flash Memory Lifetime by Tolerating Early Retention Loss and Process Variation"**

*Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, Irvine, CA, USA, June 2018.

[[Abstract](#)]

[[POMACS Journal Version \(same content, different format\)](#)]

[[Slides \(pptx\)](#) ([pdf](#))]

## Improving 3D NAND Flash Memory Lifetime by Tolerating Early Retention Loss and Process Variation

Yixin Luo<sup>†</sup>      Saugata Ghose<sup>†</sup>      Yu Cai<sup>†</sup>      Erich F. Haratsch<sup>‡</sup>      Onur Mutlu<sup>§†</sup>

<sup>†</sup>Carnegie Mellon University

<sup>‡</sup>Seagate Technology

<sup>§</sup>ETH Zürich

# Recall: Collapse of the “Galloping Gertie”

---



# Another Example (1994)

---



# Yet Another Example (2007)

---



Source: Morry Gash/AP,  
<https://www.npr.org/2017/08/01/540669701/10-years-after-bridge-collapse-america-is-still-crumbling?t=1535427165809>

# A More Recent Example (2018)

---



In-Field Patch-ability  
(Intelligent Memory)  
Can Avoid Such Failures

# Final Thoughts on RowHammer

# Aside: Byzantine Failures

---

- This class of failures is known as **Byzantine failures**
- Characterized by
  - Undetected erroneous computation
  - Opposite of “fail fast (with an error or no result)”
- “erroneous” can be “malicious” (intent is the only distinction)
- Very difficult to detect and confine Byzantine failures
- **Do all you can to avoid them**
- Lamport et al., “The Byzantine Generals Problem,” ACM TOPLAS 1982.

# Aside: Byzantine Generals Problem

---

## The Byzantine Generals Problem

LESLIE LAMPORT, ROBERT SHOSTAK, and MARSHALL PEASE  
SRI International

---

Reliable computer systems must handle malfunctioning components that give conflicting information to different parts of the system. This situation can be expressed abstractly in terms of a group of generals of the Byzantine army camped with their troops around an enemy city. Communicating only by messenger, the generals must agree upon a common battle plan. However, one or more of them may be traitors who will try to confuse the others. The problem is to find an algorithm to ensure that the loyal generals will reach agreement. It is shown that, using only oral messages, this problem is solvable if and only if more than two-thirds of the generals are loyal; so a single traitor can confound two loyal generals. With unforgeable written messages, the problem is solvable for any number of generals and possible traitors. Applications of the solutions to reliable computer systems are then discussed.

Categories and Subject Descriptors: C.2.4. [Computer-Communication Networks]: Distributed Systems—*network operating systems*; D.4.4 [Operating Systems]: Communications Management—*network communication*; D.4.5 [Operating Systems]: Reliability—*fault tolerance*

General Terms: Algorithms, Reliability

Additional Key Words and Phrases: Interactive consistency

# RowHammer, Revisited

- One can **predictably induce bit flips** in commodity DRAM chips
  - >80% of the tested DRAM chips are vulnerable
- First example of how a **simple hardware failure mechanism** can create a **widespread system security vulnerability**

WIRED

Forget Software—Now Hackers Are Exploiting Physics

BUSINESS

CULTURE

DESIGN

GEAR

SCIENCE

ANDY GREENBERG SECURITY 08.31.16 7:00 AM

SHARE

f SHARE  
18276

tweet TWEET

# FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

# RowHammer: Retrospective

---

- New mindset that has enabled a renewed interest in HW security attack research:
  - Real (memory) chips are vulnerable, in a simple and widespread manner
    - this causes real security problems
  - Hardware reliability □ security connection is now mainstream discourse
- Many new RowHammer attacks...
  - Tens of papers in top security venues
  - **More to come** as RowHammer is getting worse (DDR4 & beyond)
- Many new RowHammer solutions...
  - Apple security release; Memtest86 updated
  - Many solution proposals in top venues (latest in ISCA 2019)
  - Principled system-DRAM co-design (in original RowHammer paper)
  - **More to come...**

# Perhaps Most Importantly...

---

- RowHammer enabled a shift of mindset in mainstream security researchers
  - General-purpose hardware is fallible, in a widespread manner
  - Its problems are exploitable
- This mindset has enabled many systems security researchers to examine hardware in more depth
  - And understand HW's inner workings and vulnerabilities
- It is no coincidence that two of the groups that discovered Meltdown and Spectre heavily worked on RowHammer attacks before
  - **More to come...**

# Summary: RowHammer

---

- DRAM reliability is reducing
- Reliability issues open up security vulnerabilities
  - Very hard to defend against
- **Rowhammer is a prime example**
  - First example of how a simple hardware failure mechanism can create a widespread system security vulnerability
  - Its implications on system security research are tremendous & exciting
- Bad news: RowHammer is getting worse.
- **Good news: We have a lot more to do.**
  - We are now fully aware hardware is easily fallible.
  - We are developing both attacks and solutions.
  - We are developing principled models, methodologies, solutions.

# For More on RowHammer...

---

- Onur Mutlu and Jeremie Kim,  
**"RowHammer: A Retrospective"**  
*IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD) Special Issue on Top Picks in Hardware and Embedded Security*, 2019.  
[Preliminary arXiv version]

## RowHammer: A Retrospective

Onur Mutlu<sup>§‡</sup>

<sup>§</sup>ETH Zürich

Jeremie S. Kim<sup>†§</sup>

<sup>†</sup>Carnegie Mellon University

# RowHammer in 2020 (I)

---

- Jeremie S. Kim, Minesh Patel, A. Giray Yaglikci, Hasan Hassan, Roknoddin Azizi, Lois Orosa, and Onur Mutlu,  
**"Revisiting RowHammer: An Experimental Analysis of Modern Devices and Mitigation Techniques"**

*Proceedings of the 47th International Symposium on Computer Architecture (ISCA), Valencia, Spain, June 2020.*

[Slides (pptx) (pdf)]

[Lightning Talk Slides (pptx) (pdf)]

[Talk Video (20 minutes)]

[Lightning Talk Video (3 minutes)]

## Revisiting RowHammer: An Experimental Analysis of Modern DRAM Devices and Mitigation Techniques

Jeremie S. Kim<sup>§†</sup>      Minesh Patel<sup>§</sup>      A. Giray Yağlıkçı<sup>§</sup>  
Hasan Hassan<sup>§</sup>      Roknoddin Azizi<sup>§</sup>      Lois Orosa<sup>§</sup>      Onur Mutlu<sup>§†</sup>

<sup>§</sup>*ETH Zürich*

<sup>†</sup>*Carnegie Mellon University*

# RowHammer in 2020 (II)

---

- Pietro Frigo, Emanuele Vannacci, Hasan Hassan, Victor van der Veen, Onur Mutlu, Cristiano Giuffrida, Herbert Bos, and Kaveh Razavi,  
**"TRRespass: Exploiting the Many Sides of Target Row Refresh"**  
*Proceedings of the 41st IEEE Symposium on Security and Privacy (S&P)*, San Francisco, CA, USA, May 2020.  
[[Slides \(pptx\)](#) ([pdf](#))]  
[[Talk Video](#) (17 minutes)]  
[[Source Code](#)]  
[[Web Article](#)]  
**Best paper award.**

## TRRespass: Exploiting the Many Sides of Target Row Refresh

Pietro Frigo\*†    Emanuele Vannacci\*†    Hasan Hassan§    Victor van der Veen¶  
Onur Mutlu§    Cristiano Giuffrida\*    Herbert Bos\*    Kaveh Razavi\*

# RowHammer in 2020 (III)

---

- Lucian Cojocar, Jeremie Kim, Minesh Patel, Lillian Tsai, Stefan Saroiu, Alec Wolman, and Onur Mutlu,

## "Are We Susceptible to Rowhammer? An End-to-End Methodology for Cloud Providers"

*Proceedings of the 41st IEEE Symposium on Security and Privacy (S&P), San Francisco, CA, USA, May 2020.*

[Slides (pptx) (pdf)]

[Talk Video (17 minutes)]

# Are We Susceptible to Rowhammer? An End-to-End Methodology for Cloud Providers

Lucian Cojocar, Jeremie Kim<sup>§†</sup>, Minesh Patel<sup>§</sup>, Lillian Tsai<sup>‡</sup>,  
Stefan Saroiu, Alec Wolman, and Onur Mutlu<sup>§†</sup>  
Microsoft Research, <sup>§</sup>ETH Zürich, <sup>†</sup>CMU, <sup>‡</sup>MIT

---

**Rowhammer**



# Some History

# Initial RowHammer Reviews

## Disturbance Errors in DRAM: Demonstration, Characterization, and Prevention

Rejected (R2)



863kB

Friday 31 May 2013 2:00:53pm PDT

|

b9bf06021da54cddf4cd0b3565558a181868b972

You are an **author** of this paper.

+ ABSTRACT

+ AUTHORS

- [Review #66A](#)
- [Review #66B](#)
- [Review #66C](#)
- [Review #66D](#)
- [Review #66E](#)
- [Review #66F](#)

OveMer	Nov	WriQua	RevExp
1	4	4	4
5	4	5	3
2	3	5	4
1	2	3	4
4	4	4	3
2	4	4	3

# Missing the Point **Reviews from Micro 2013**

## PAPER WEAKNESSES

This is an excellent test methodology paper, but there is no micro-architectural or architectural content.

## PAPER WEAKNESSES

- Whereas they show disturbance may happen in DRAM array, authors don't show it can be an issue in realistic DRAM usage scenario
- Lacks architectural/microarchitectural impact on the DRAM disturbance analysis

## PAPER WEAKNESSES

The mechanism investigated by the authors is one of many well known disturb mechanisms. The paper does not discuss the root causes to sufficient depth and the importance of this mechanism compared to others. Overall the length of the sections restating known information is much too long in relation to new work.

## PAPER WEAKNESSES

- 1) The disturbance error (a.k.a coupling or cross-talk noise induced error) is a known problem to the DRAM circuit community.
  - 2) What you demonstrated in this paper is so called DRAM row hammering issue - you can even find a Youtube video showing this! - <http://www.youtube.com/watch?v=i3-qQSnBcdo>
  - 2) The architectural contribution of this study is too insignificant.
- PAPER WEAKNESSES**
- Row Hammering appears to be well-known, and solutions have already been proposed by industry to address the issue.
  - The paper only provides a qualitative analysis of solutions to the problem. A more robust evaluation is really needed to know whether the proposed solution is necessary.

# Final RowHammer Reviews

---

## Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors

Accepted



639kB

21 Nov 2013 10:53:11pm CST |

f039be2735313b39304ae1c6296523867a485610

You are an **author** of this paper.

	Ove	Mer	Nov	Wri	Qua	Rev	Con	And
<a href="#">Review #41A</a>	8	4	5		3			
<a href="#">Review #41B</a>	7	4	4		3			
<a href="#">Review #41C</a>	6	4	4		3			
<a href="#">Review #41D</a>	2	2	5		4			
<a href="#">Review #41E</a>	3	2	3		3			
<a href="#">Review #41F</a>	7	4	4		3			

# Suggestions to Reviewers

---

- Be fair; you do not know it all
- Be open-minded; you do not know it all
- Be accepting of diverse research methods: there is no single way of doing research
- Be constructive, not destructive
- Do not have double standards...

**Do not block or delay scientific progress for non-reasons**

---

# An Interview on Research and Education

---

- Computing Research and Education (@ ISCA 2019)
  - [https://www.youtube.com/watch?v=8ffSEKZhmvo&list=PL5Q2soXY2Zi\\_4oP9LdL3cc8G6NIjD2Ydz](https://www.youtube.com/watch?v=8ffSEKZhmvo&list=PL5Q2soXY2Zi_4oP9LdL3cc8G6NIjD2Ydz)
- Maurice Wilkes Award Speech (10 minutes)
  - [https://www.youtube.com/watch?v=tcQ3zZ3JpuA&list=PL5Q2soXY2Zi8D\\_5MGV6EnXEJHnV2YFBJI&index=15](https://www.youtube.com/watch?v=tcQ3zZ3JpuA&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJI&index=15)

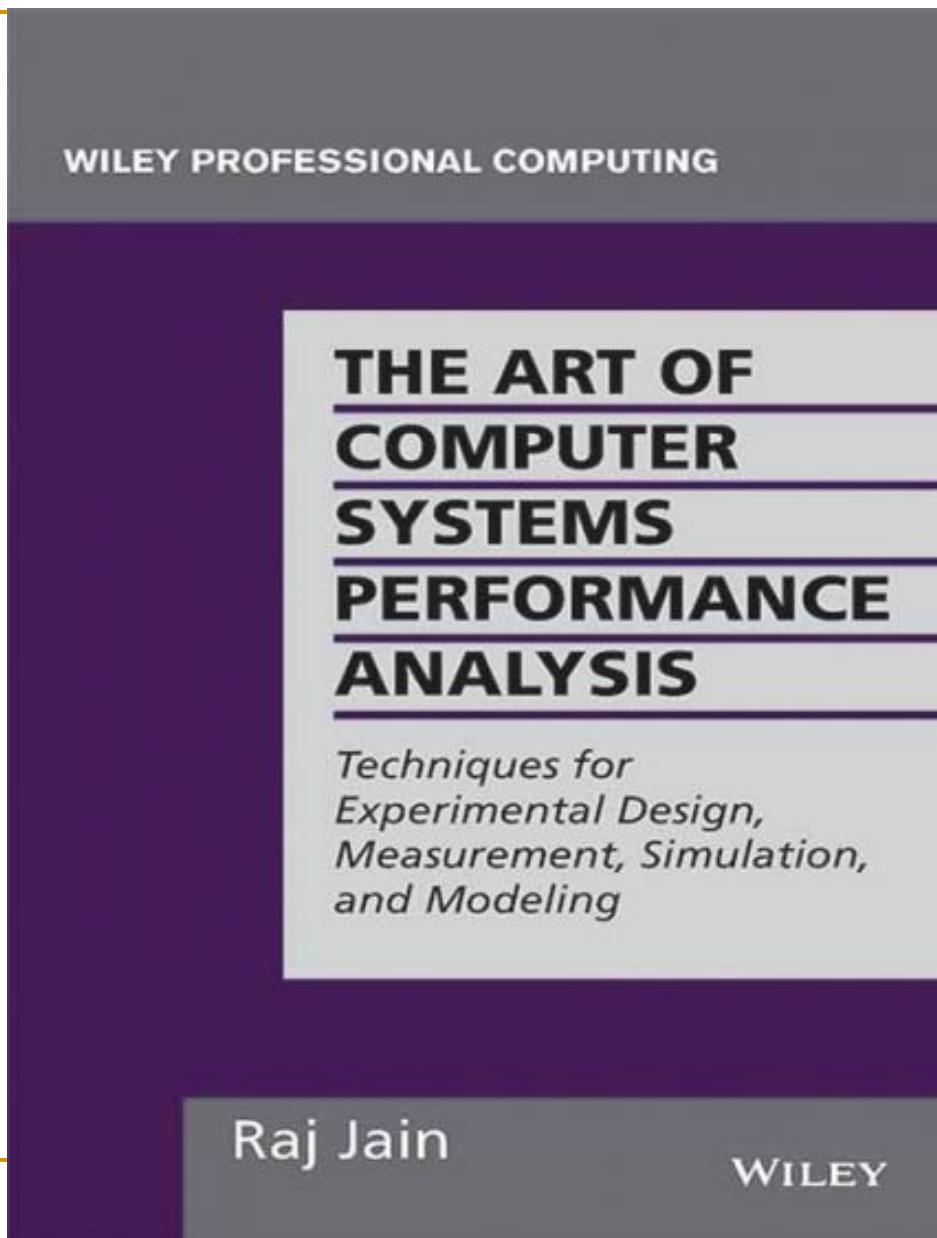
# More Thoughts and Suggestions

---

- Onur Mutlu,  
**"Some Reflections (on DRAM)"**  
Award Speech for ACM SIGARCH Maurice Wilkes Award, at the ***ISCA Awards Ceremony***, Phoenix, AZ, USA, 25 June 2019.  
[Slides (pptx) (pdf)]  
[Video of Award Acceptance Speech (Youtube; 10 minutes) (Youku; 13 minutes)]  
[Video of Interview after Award Acceptance (Youtube; 1 hour 6 minutes) (Youku; 1 hour 6 minutes)]  
[News Article on "ACM SIGARCH Maurice Wilkes Award goes to Prof. Onur Mutlu"]
  
- Onur Mutlu,  
**"How to Build an Impactful Research Group"**  
57th Design Automation Conference Early Career Workshop (DACE), Virtual, 19 July 2020.  
[Slides (pptx) (pdf)]

# Aside: A Recommended Book

---



Raj Jain, “*The Art of Computer Systems Performance Analysis*,” Wiley, 1991.

**10.8 DECISION MAKER'S GAMES**

Even if the performance analysis is correctly done and presented, it may not be enough to persuade your audience—the decision makers—to follow your recommendations. The list shown in Box 10.2 is a compilation of reasons for rejection heard at various performance analysis presentations. You can use the list by presenting it immediately and pointing out that the reason for rejection is not new and that the analysis deserves more consideration. Also, the list is helpful in getting the competing proposals rejected!

There is no clear end of an analysis. Any analysis can be rejected simply on the grounds that the problem needs more analysis. This is the first reason listed in Box 10.2. The second most common reason for rejection of an analysis and for endless debate is the workload. Since workloads are always based on the past measurements, their applicability to the current or future environment can always be questioned. Actually workload is one of the four areas of discussion that lead a performance presentation into an endless debate. These “rat holes” and their relative sizes in terms of time consumed are shown in Figure 10.26. Presenting this cartoon at the beginning of a presentation helps to avoid these areas.

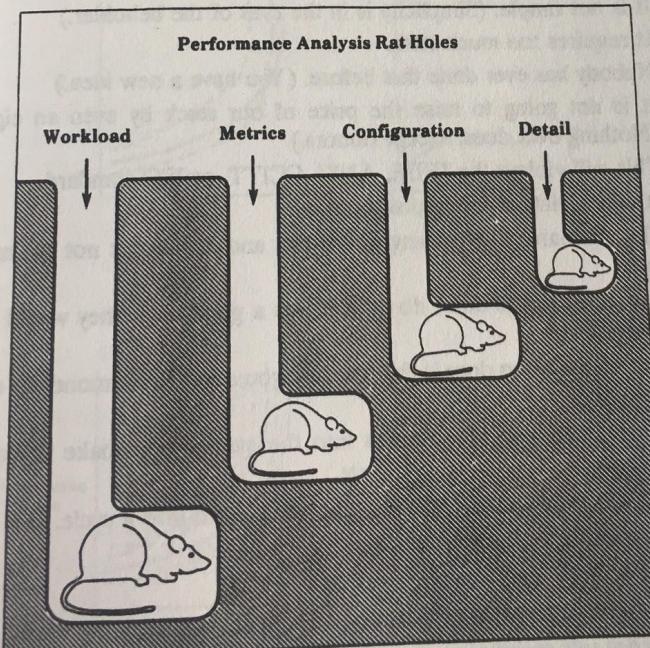


FIGURE 10.26 Four issues in performance presentations that commonly lead to endless discussion.

Raj Jain, “The Art of Computer Systems Performance Analysis,” Wiley, 1991.

**Box 10.2 Reasons for Not Accepting the Results of an Analysis**

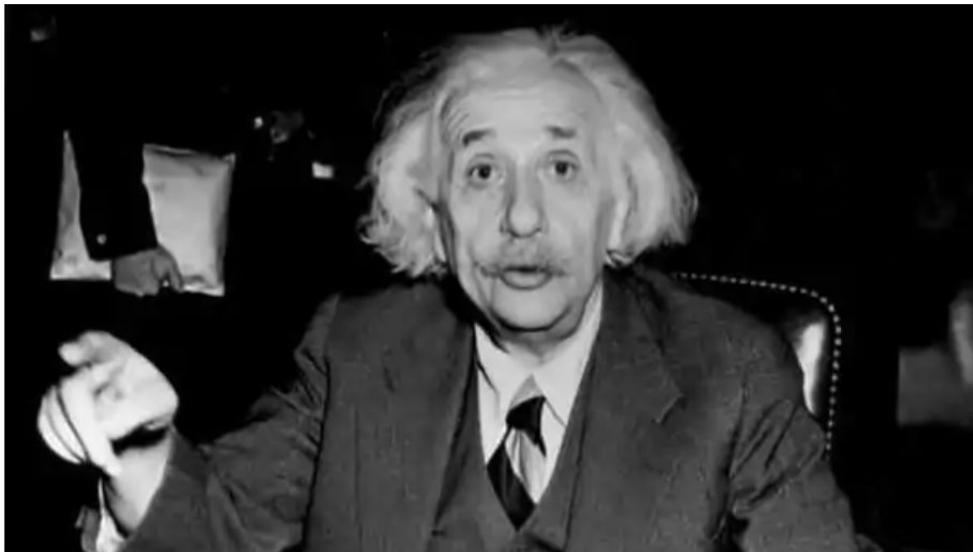
1. This needs more analysis.
2. You need a better understanding of the workload.
3. It improves performance only for long I/O's, packets, jobs, and files, and most of the I/O's, packets, jobs, and files are short.
4. It improves performance only for short I/O's, packets, jobs, and files, but who cares for the performance of short I/O's, packets, jobs, and files; it's the long ones that impact the system.
5. It needs too much memory/CPU/bandwidth and memory/CPU/bandwidth isn't free.
6. It only saves us memory/CPU/bandwidth and memory/CPU/bandwidth is cheap.
7. There is no point in making the networks (similarly, CPUs/disks/...) faster; our CPUs/disks (any component other than the one being discussed) aren't fast enough to use them.
8. It improves the performance by a factor of  $x$ , but it doesn't really matter at the user level because everything else is so slow.
9. It is going to increase the complexity and cost.
10. Let us keep it simple stupid (and your idea is not stupid).
11. It is not simple. (Simplicity is in the eyes of the beholder.)
12. It requires too much state.
13. Nobody has ever done that before. (You have a new idea.)
14. It is not going to raise the price of our stock by even an eighth. (Nothing ever does, except rumors.)
15. This will violate the IEEE, ANSI, CCITT, or ISO standard.
16. It may violate some future standard.
17. The standard says nothing about this and so it must not be important.
18. Our competitors don't do it. If it was a good idea, they would have done it.
19. Our competition does it this way and you don't make money by copying others.
20. It will introduce randomness into the system and make debugging difficult.
21. It is too deterministic; it may lead the system into a cycle.
22. It's not interoperable.
23. This impacts hardware.
24. That's beyond today's technology.
25. It is not self-stabilizing.
26. Why change—it's working OK.

Raj Jain, “The Art of Computer Systems Performance Analysis,” Wiley, 1991.

# A Fun Reading: Food for Thought

---

- <https://www.livemint.com/science/news/could-einstein-get-published-today-11601014633853.html>



A similar process of professionalization has transformed other parts of the scientific landscape. (Central Press/Getty Images)

THE WALL STREET JOURNAL.

## Could Einstein get published today?

3 min read . Updated: 25 Sep 2020, 11:51 AM IST

The Wall Street Journal

Scientific journals and institutions have become more professionalized over the last century, leaving less room for individual style

# Computer Architecture

## Lecture 4b: RowHammer

Prof. Onur Mutlu

ETH Zürich

Fall 2020

25 September 2020