RowHammer, RowPress & Beyond

Can We Be Free of Bitflips (Soon)?

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
13 June 2024
University of Colorado Boulder

SAFARI ETH zürich Carnegie Mellon

How Reliable/Secure/Safe is This Bridge?

Source: http://www.technologystudent.com/struct1/tacom1.png

Collapse of the "Galloping Gertie"

Source: AP
http://www.wsdot.wa.gov/tnbhistory/connections/connections3.htm

How Safe & Secure Are These People?

Security is about preventing unforeseen consequences

Source: https://s-media-cache-ak0.pinimg.com/originals/48/09/54/4809543a9c7700246a0cf8acdae27abf.jpg

How Safe & Secure Are Our Platforms?

Security is about preventing unforeseen consequences

What Is RowHammer?

- One can predictably induce bit flips in commodity DRAM chips
  - All recent DRAM chips are fundamentally 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

An "Early" Position Paper [IMW'13]

- Onur Mutlu,
  "Memory Scaling: A Systems Architecture Perspective"
  *Proceedings of the 5th International Memory Workshop (**IMW**)*, Monterey, CA, May 2013. Slides (pptx) (pdf)
  EETimes Reprint

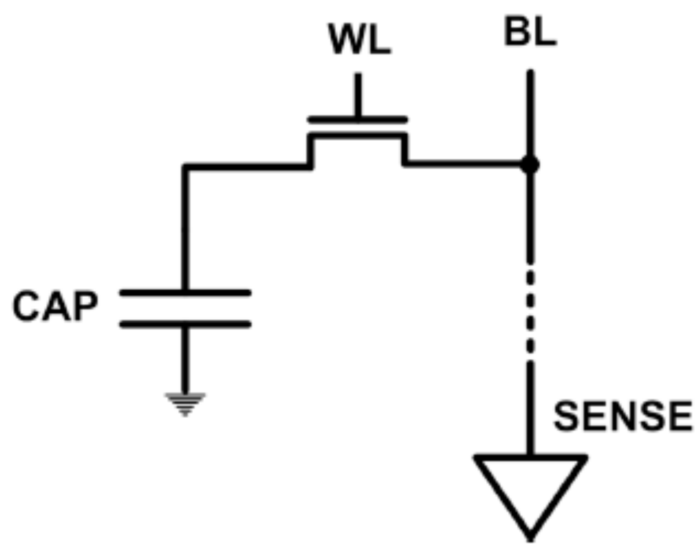
Memory Scaling: A Systems Architecture Perspective

Onur Mutlu
Carnegie Mellon University
onur@cmu.edu
http://users.ece.cmu.edu/~omutlu/

https://people.inf.ethz.ch/omutlu/pub/memory-scaling_memcon13.pdf

The DRAM Scaling Problem

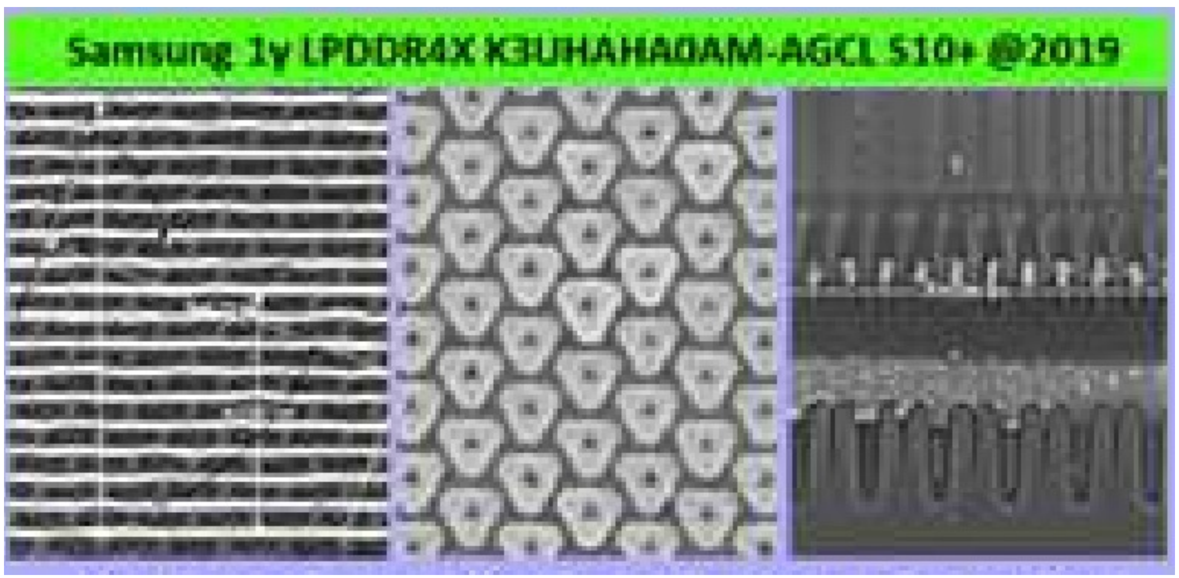
- DRAM stores charge in a capacitor (charge-based memory)
  - Capacitor must be large enough for reliable sensing
  - Access transistor should be large enough for low leakage and high retention time
  - Scaling beyond 40-35nm (2013) is challenging [ITRS, 2009]



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

The DRAM Scaling Problem

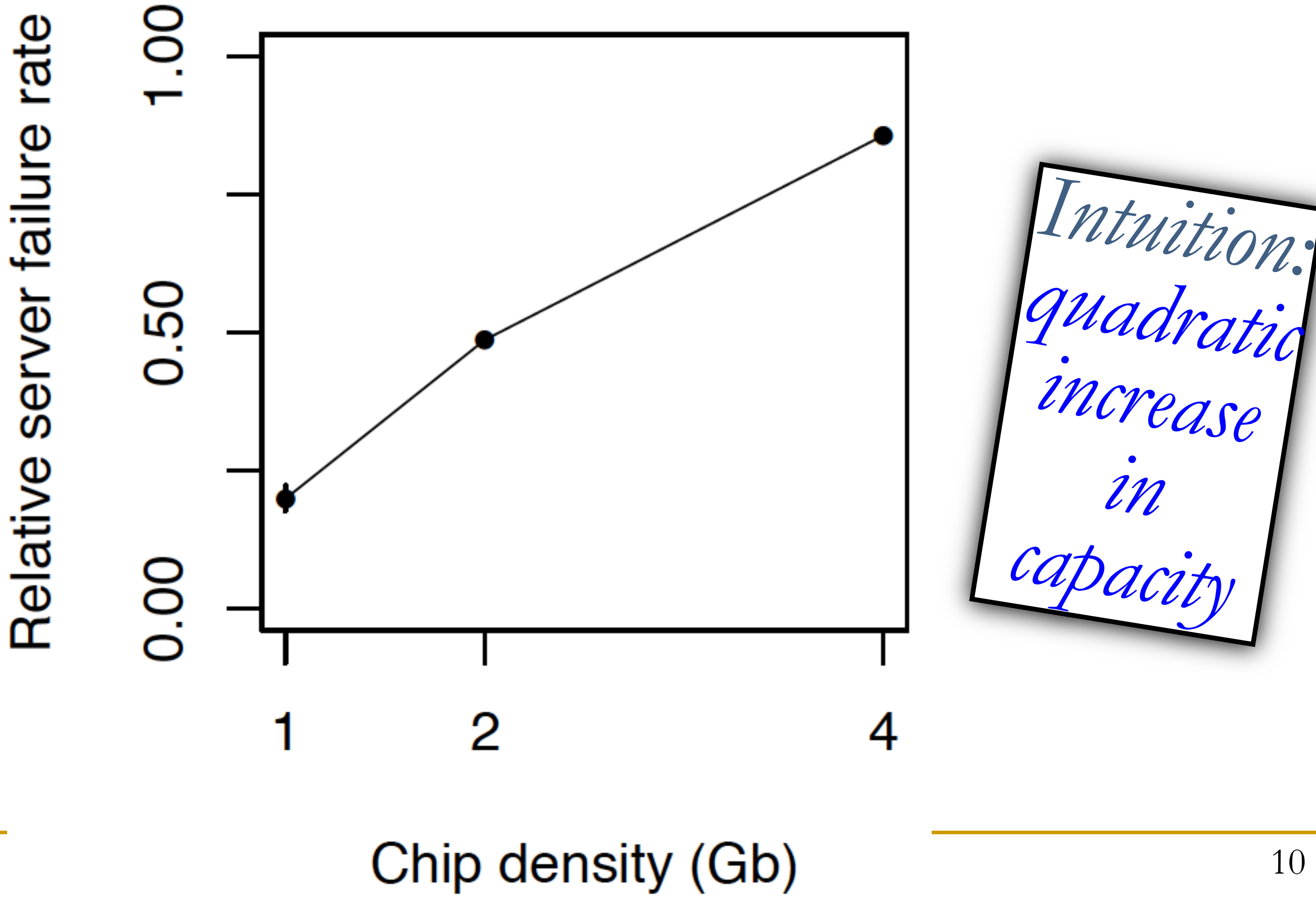
- DRAM stores charge in a capacitor (charge-based memory)
  - Capacitor must be large enough for reliable sensing
  - Access transistor should be large enough for low leakage and high retention time
  - Scaling beyond 40-35nm (2013) is challenging [ITRS, 2009]



- 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.



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

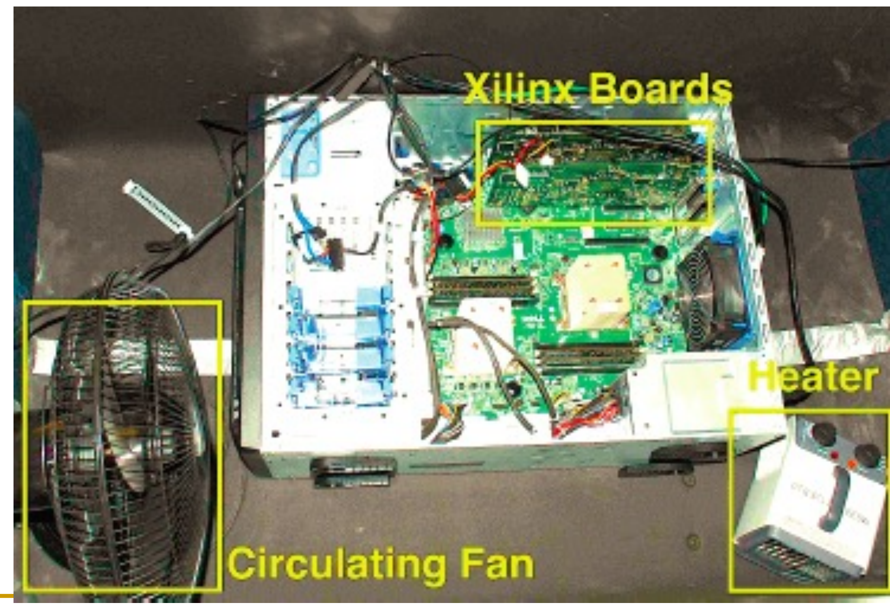
Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors (Kim et al., ISCA 2014)

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

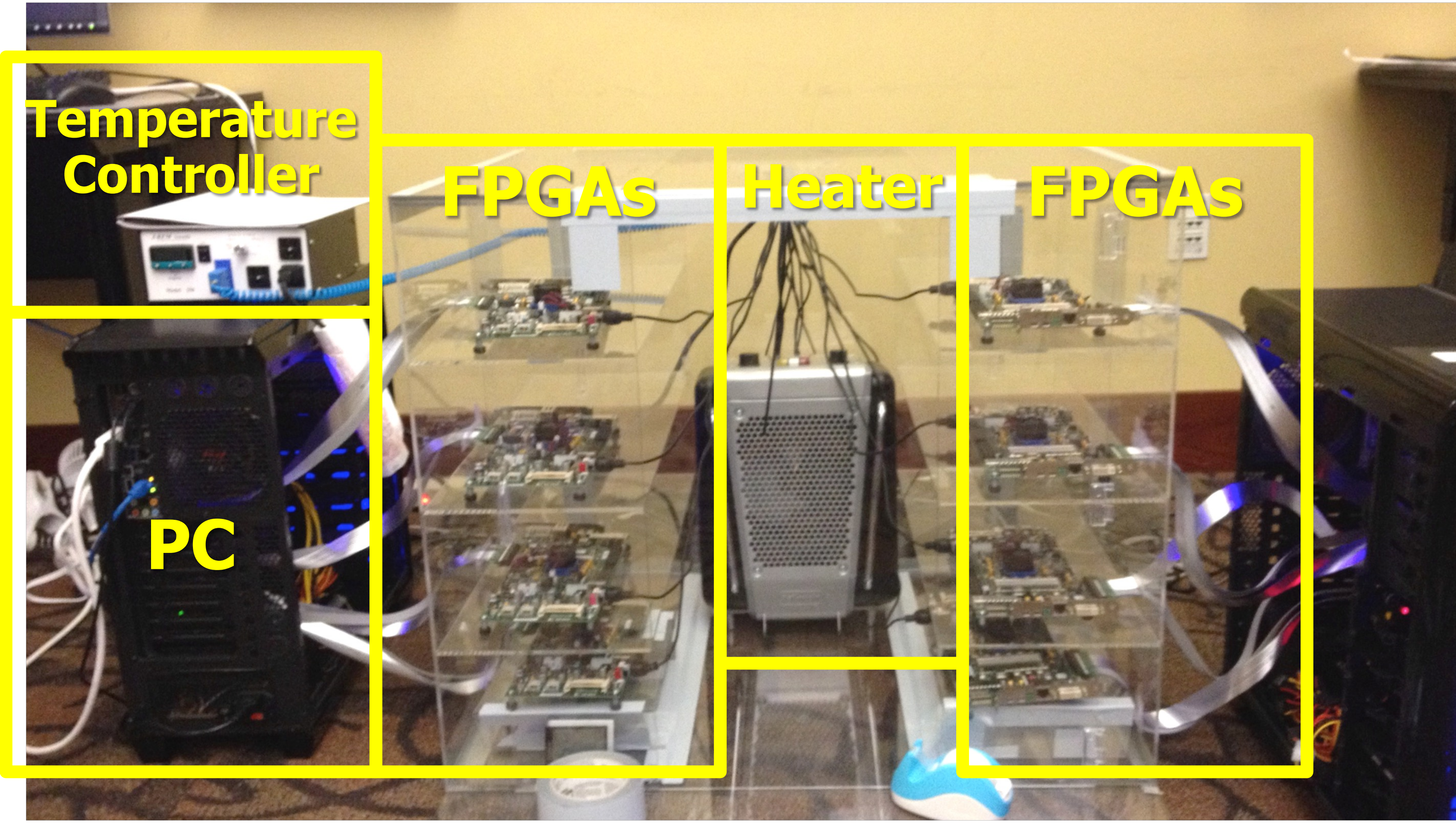
AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems (Qureshi et al., DSN 2015)

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



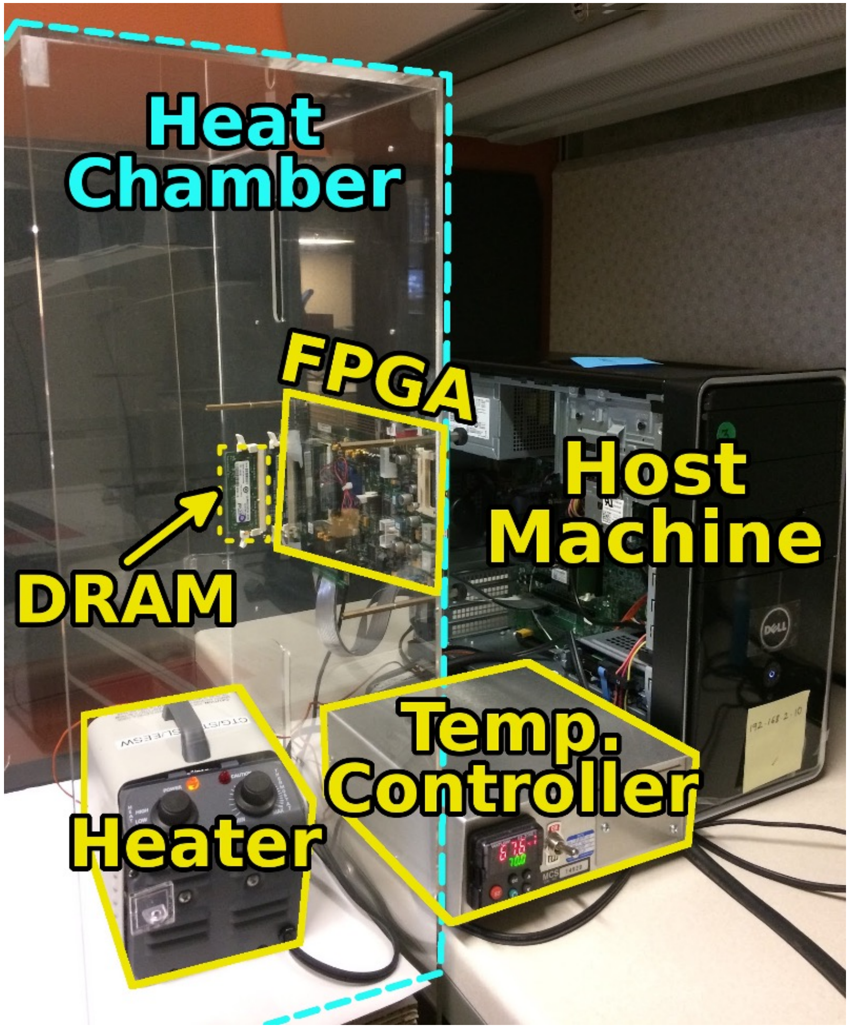
Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.

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



SoftMC: Open Source DRAM Infrastructure

- Hasan Hassan, Nandita Vijaykumar, Samira Khan, Saugata Ghose, Kevin Chang, Gennady Pekhimenko, Donghyuk Lee, Oguz Ergin, and Onur Mutlu,
  "SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies"
  *Proceedings of the 23rd International Symposium on High-Performance Computer Architecture (**HPCA**)*, Austin, TX, USA, February 2017.
  [Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)]
  [Full Talk Lecture (39 minutes)]
  [Source Code]

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

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

[1]*ETH Zürich* [2]*TOBB University of Economics & Technology* [3]*Carnegie Mellon University*
[4]*University of Virginia* [5]*Microsoft Research* [6]*NVIDIA Research*

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

DRAM Bender

- Ataberk Olgun, Hasan Hassan, A Giray Yağlıkçı, Yahya Can Tuğrul, Lois Orosa, Haocong Luo, Minesh Patel, Oğuz Ergin, and Onur Mutlu,
  "DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips"
  *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (**TCAD**)*, 2023.
  [Extended arXiv version]
  [DRAM Bender Source Code]
  [DRAM Bender Tutorial Video (43 minutes)]

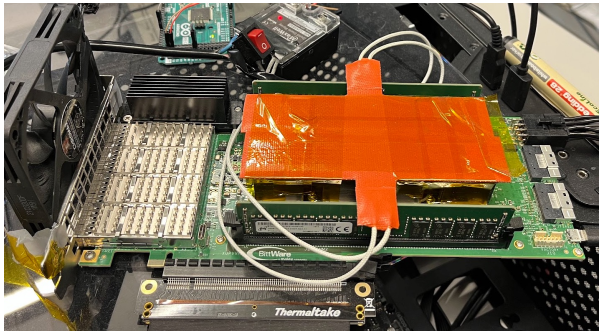
DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips

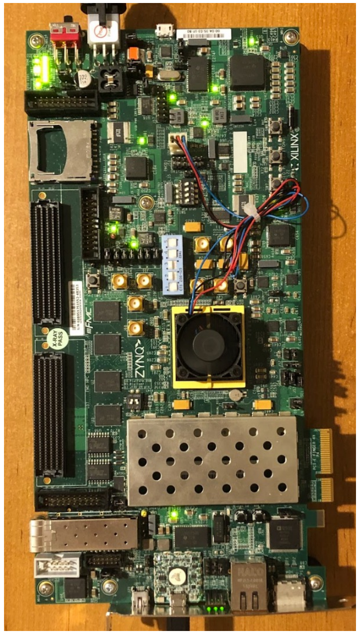
Ataberk Olgun[§] Hasan Hassan[§] A. Giray Yağlıkçı[§] Yahya Can Tuğrul[§†]
Lois Orosa[§⊙] Haocong Luo[§] Minesh Patel[§] Oğuz Ergin[†] Onur Mutlu[§]
[§]*ETH Zürich* [†]*TOBB ETÜ* [⊙]*Galician Supercomputing Center*

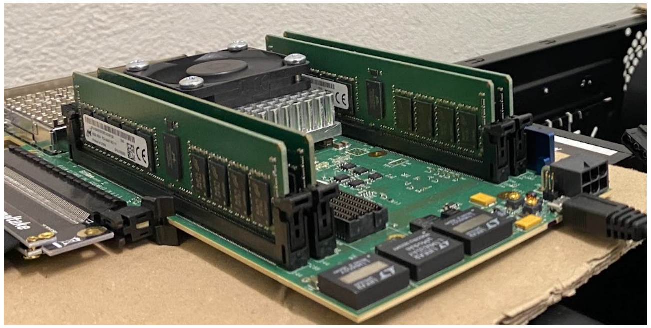
DRAM Bender: Prototypes

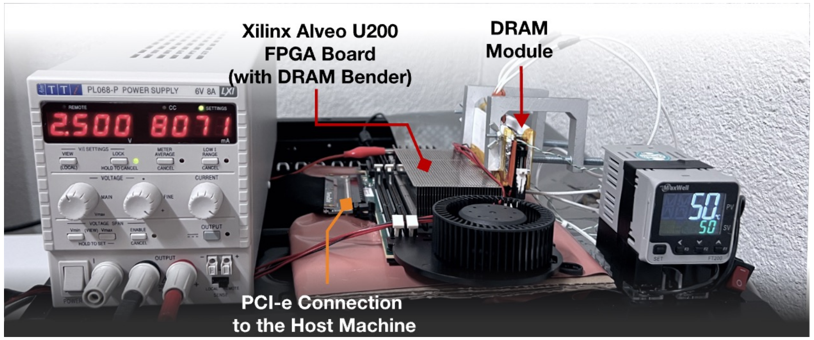
Testing Infrastructure	Protocol Support	FPGA Support
SoftMC [134]	DDR3	One Prototype
LiteX RowHammer Tester (LRT) [17]	DDR3/4, LPDDR4	Two Prototypes
DRAM Bender (this work)	**DDR3/DDR4**	**Five Prototypes**

Five out of the box FPGA-based prototypes



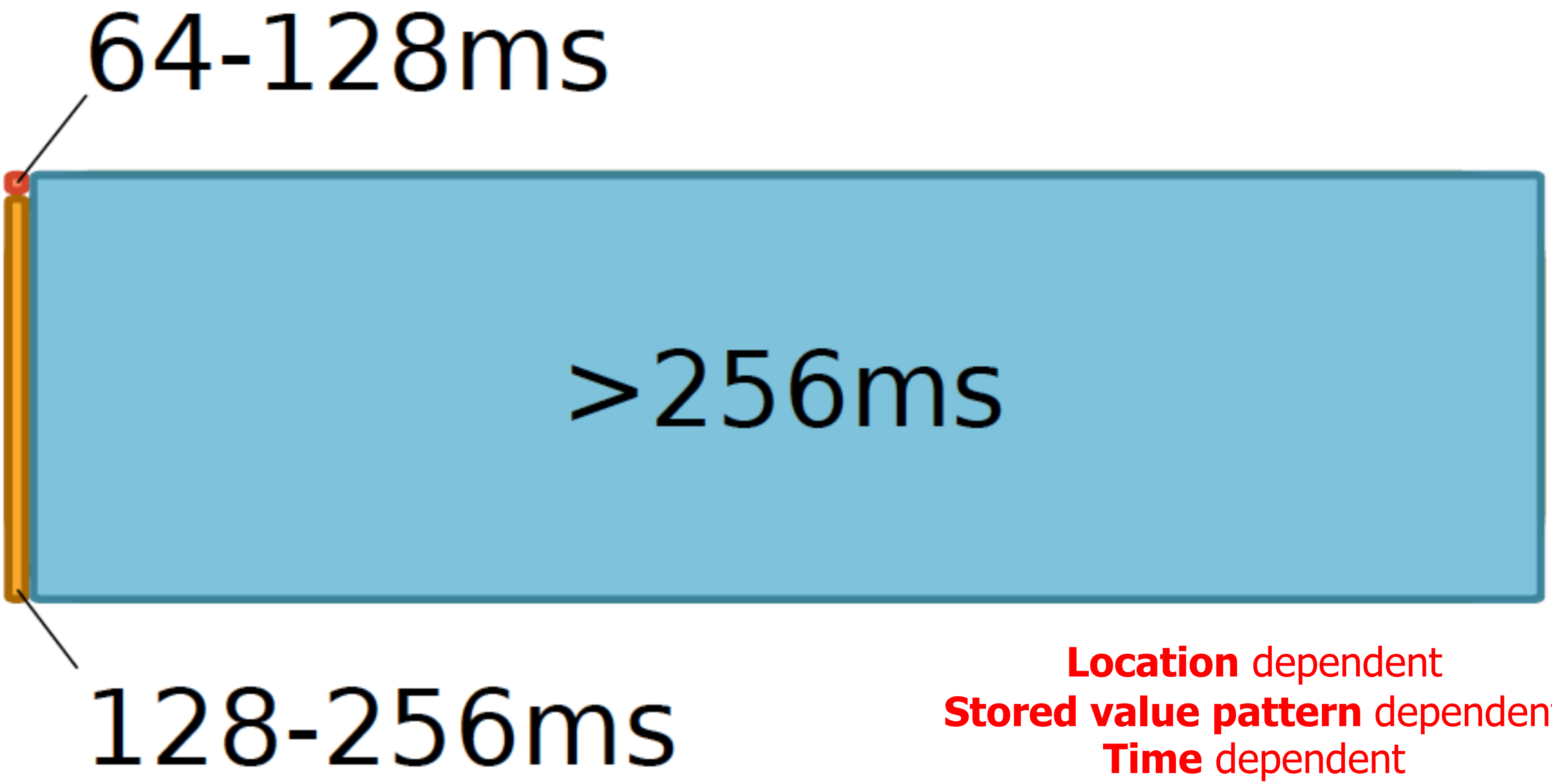






Data Retention in Memory [Liu et al., ISCA 2013]

- Retention Time Profile of DRAM looks like this:



Location dependent
Stored value pattern dependent
Time dependent

Liu+, "RAIDR: Retention-Aware Intelligent DRAM Refresh," ISCA 2012.

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)
  [Invited Retrospective at 50 Years of ISCA, 2023 (pdf)]
  Selected to the ISCA-50 25-Year Retrospective Issue covering 1996-2020 in 2023 (Retrospective (pdf) Full Issue).

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)
  [Invited Retrospective at 50 Years of ISCA, 2023 (pdf)]
  Selected to the ISCA-50 25-Year Retrospective Issue covering 1996-2020 in 2023 (Retrospective (pdf) Full Issue).

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

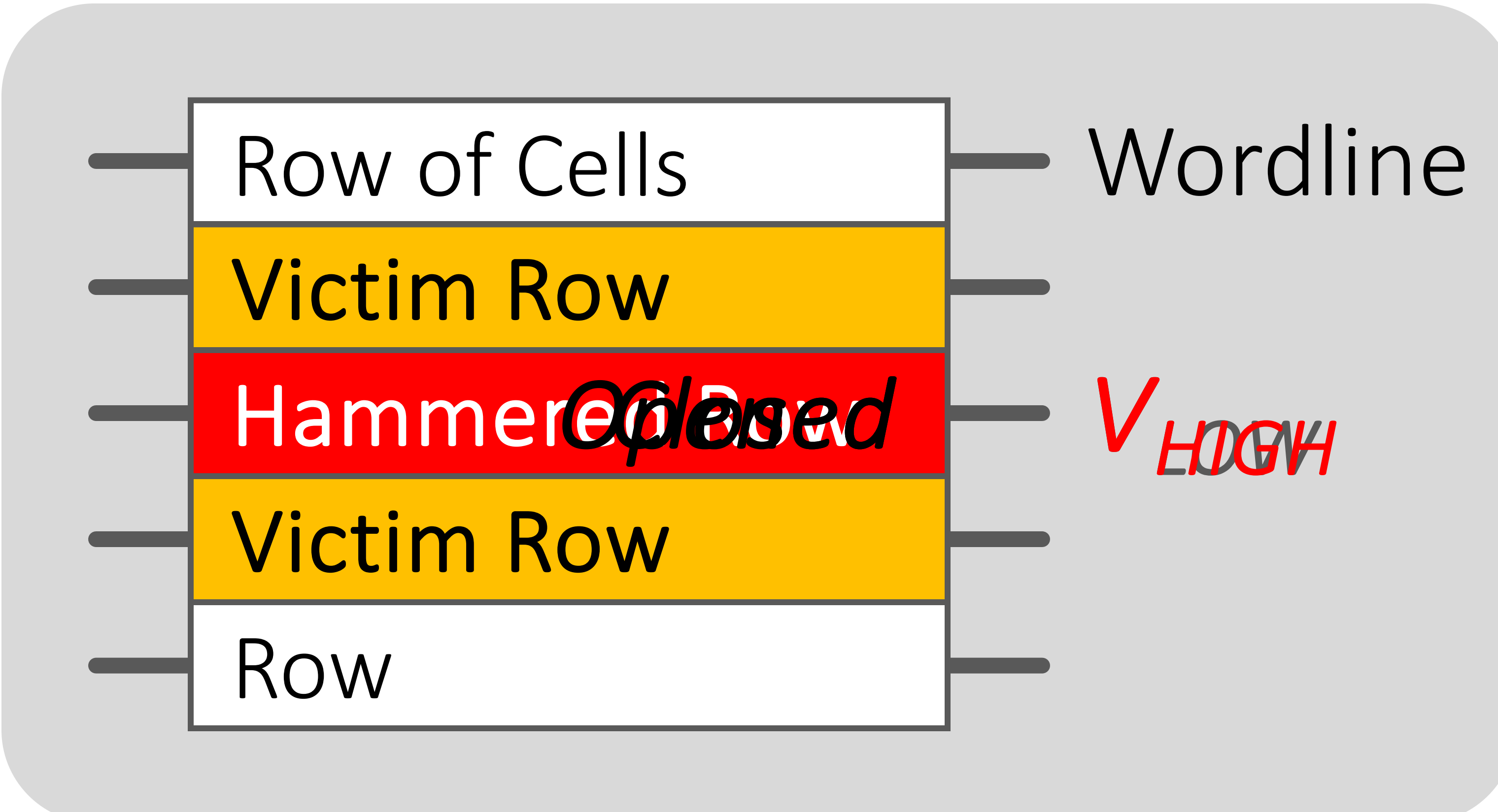
Yoongu Kim
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
yoonguk@ece.cmu.edu

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

A Curious Phenomenon

Modern DRAM is Prone to Disturbance Errors

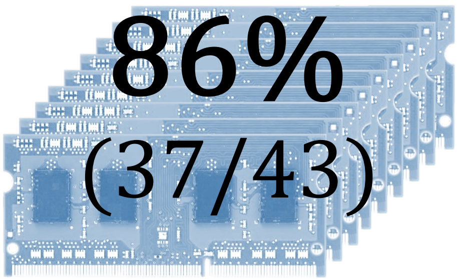


Repeatedly reading a row enough times (before memory gets refreshed) induces disturbance errors in **adjacent rows** in most real DRAM chips you can buy today

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

Most DRAM Modules Are Vulnerable

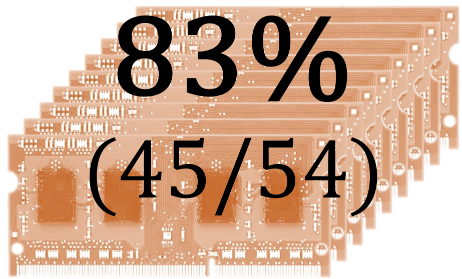
A company



86%
(37/43)

Up to
1.0×10^{7}
errors

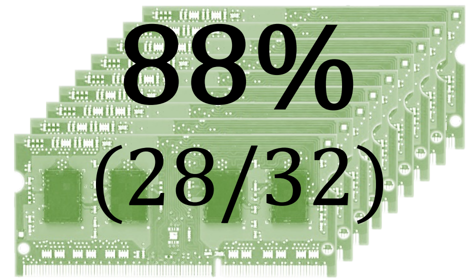
B company



83%
(45/54)

Up to
2.7×10^{6}
errors

C company

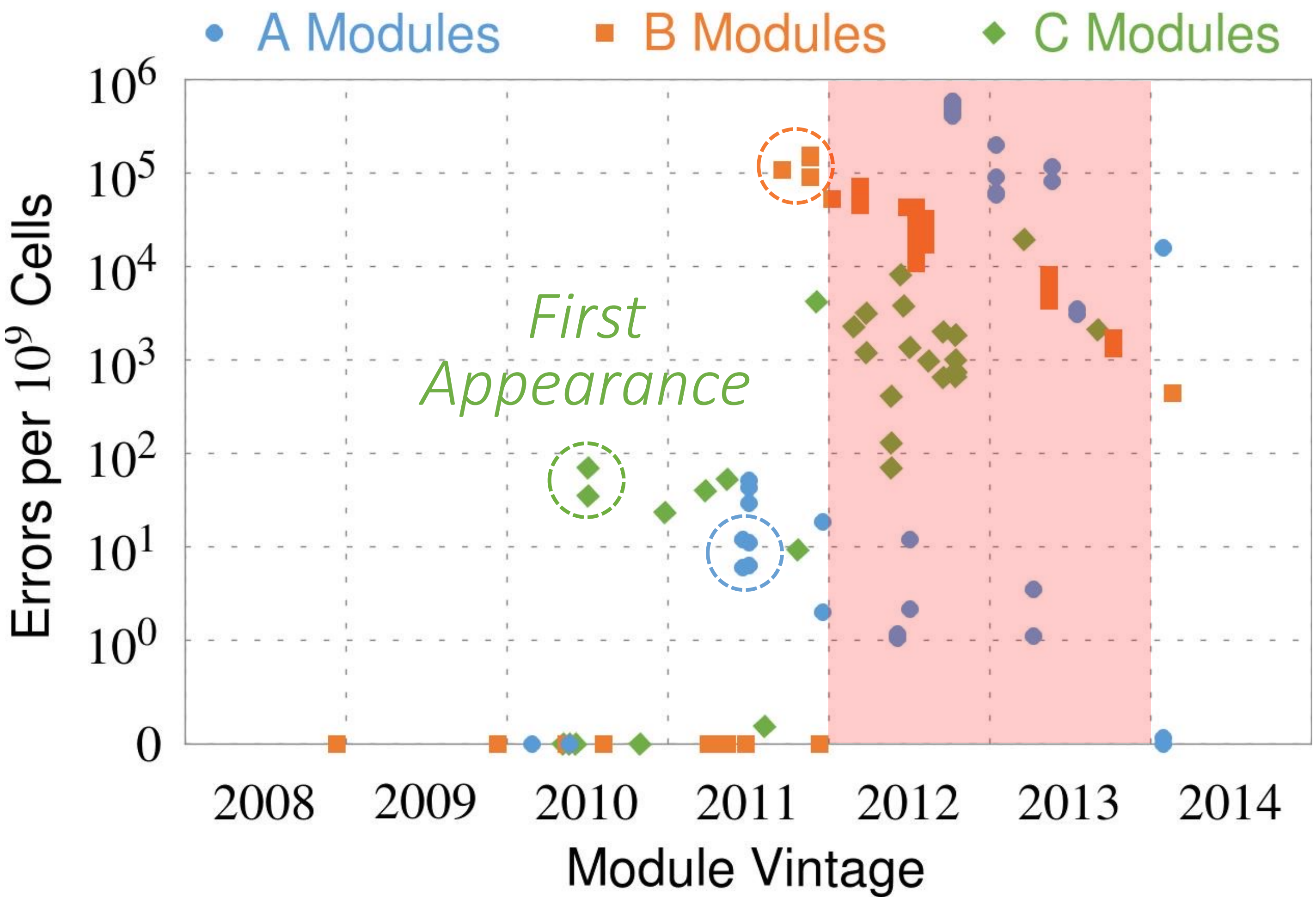


88%
(28/32)

Up to
3.3×10^{5}
errors

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

Recent DRAM Is More Vulnerable



All modules from 2012–2013 are vulnerable

Why Is This Happening?

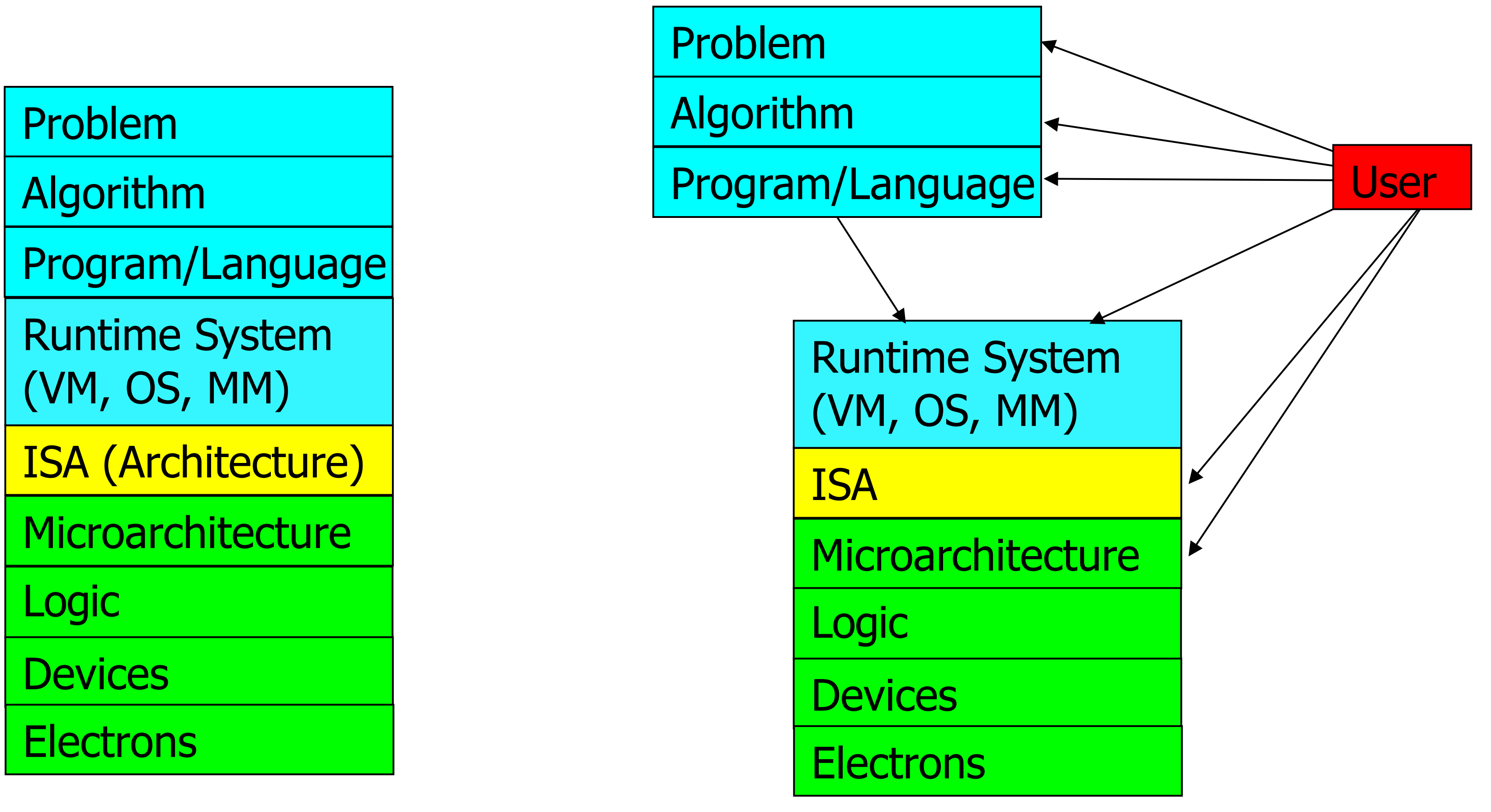
- DRAM cells are too close to each other!
  - They are not electrically isolated from each other

- 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

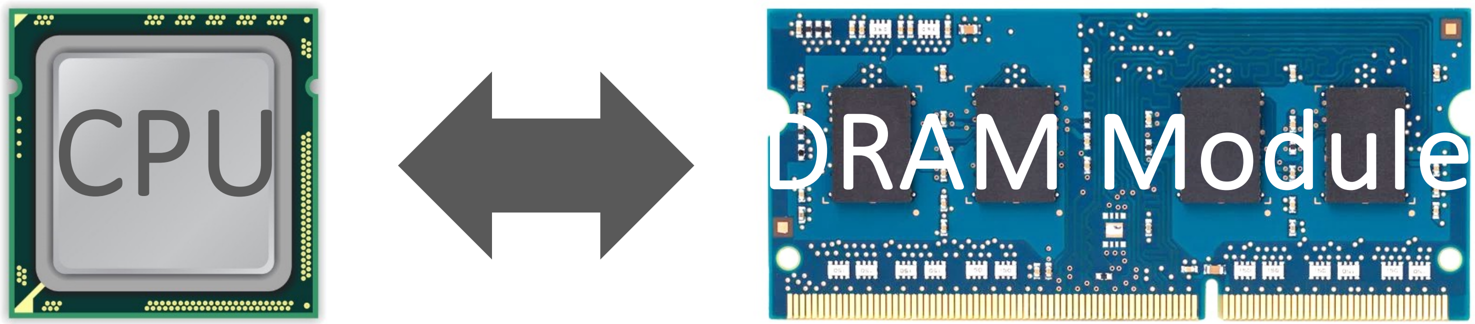
- Example: When we activate (apply high voltage) to a row, an adjacent row gets slightly activated as well
  - Vulnerable cells in that slightly-activated row lose a little bit of charge
  - If RowHammer happens enough times, charge in such cells gets drained

Higher-Level Implications

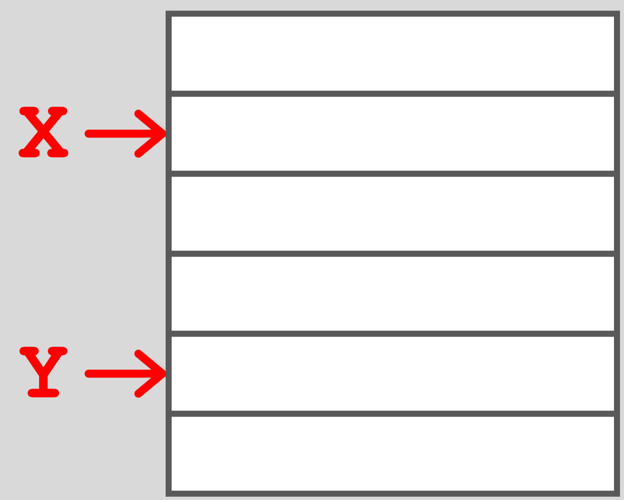
- This simple circuit level failure mechanism has enormous implications on upper layers of the transformation hierarchy



A Simple Program Can Induce Many Errors

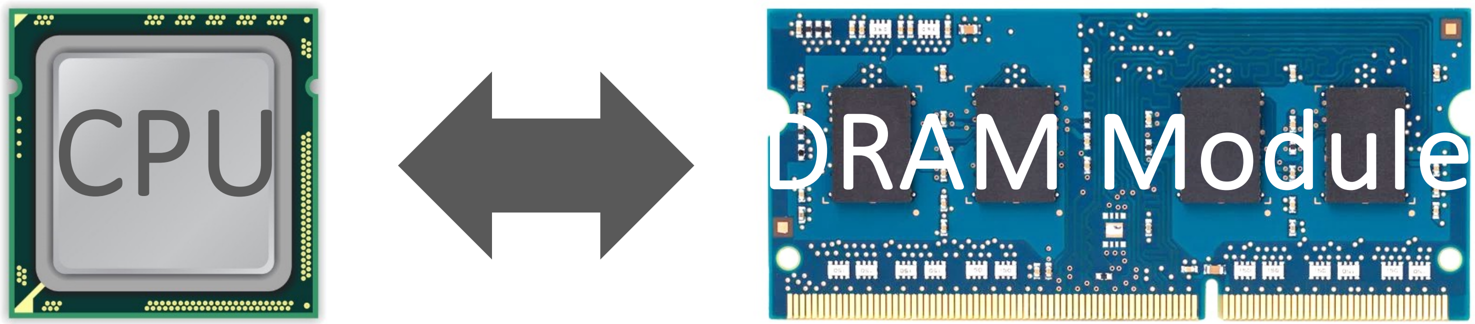



```
loop:
   mov (X), %eax
   mov (Y), %ebx
   clflush (X)
   clflush (Y)
   mfence
   jmp loop
```



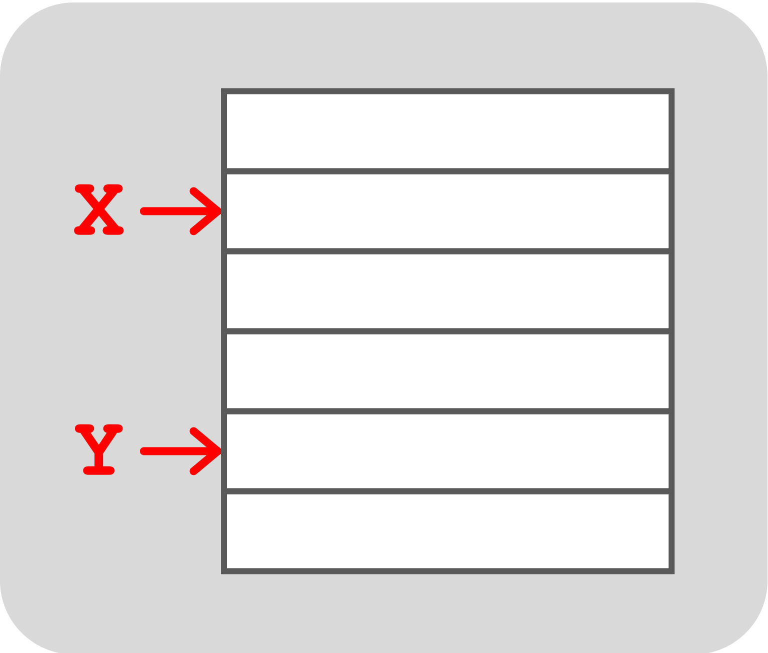
Download from: https://github.com/CMU-SAFARI/rowhammer

A Simple Program Can Induce Many Errors



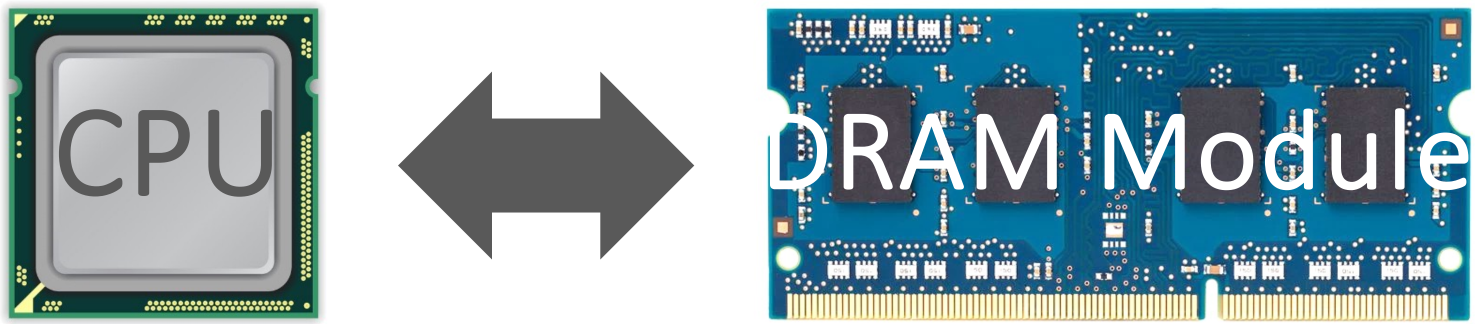
1. Avoid ***cache hits***
   – Flush **X** from cache

2. Avoid ***row hits*** to **X**
   – Read **Y** in another row

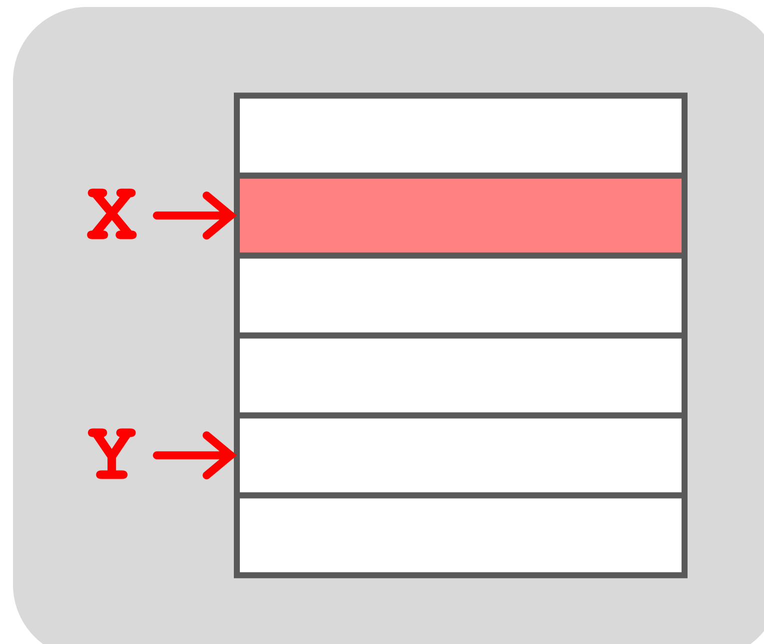


Download from: https://github.com/CMU-SAFARI/rowhammer

A Simple Program Can Induce Many Errors

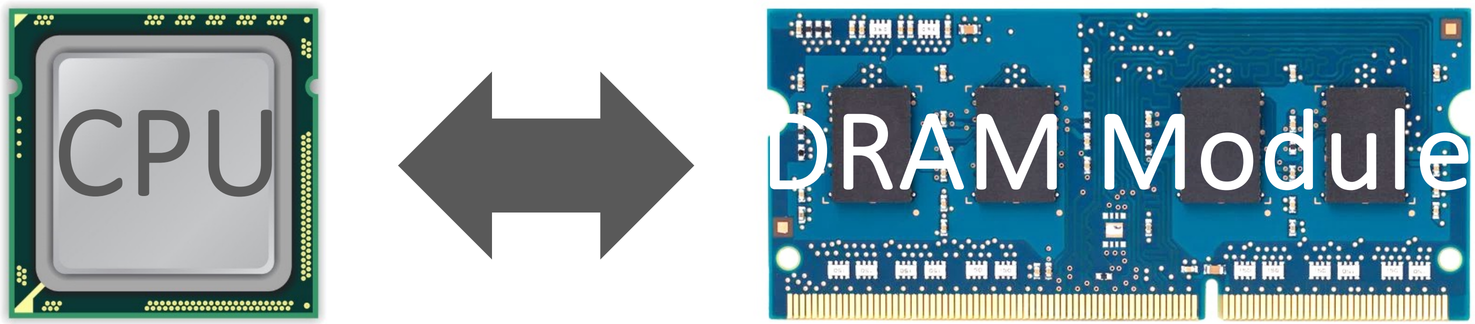



```
loop:
  mov (X), %eax
  mov (Y), %ebx
  clflush (X)
  clflush (Y)
  mfence
  jmp loop
```

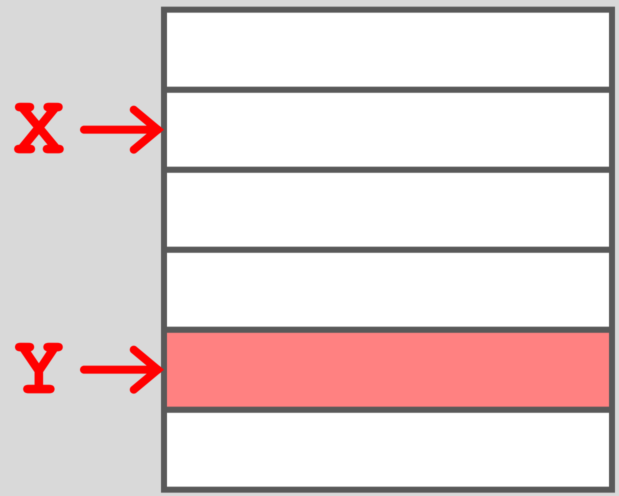


Download from: https://github.com/CMU-SAFARI/rowhammer

A Simple Program Can Induce Many Errors

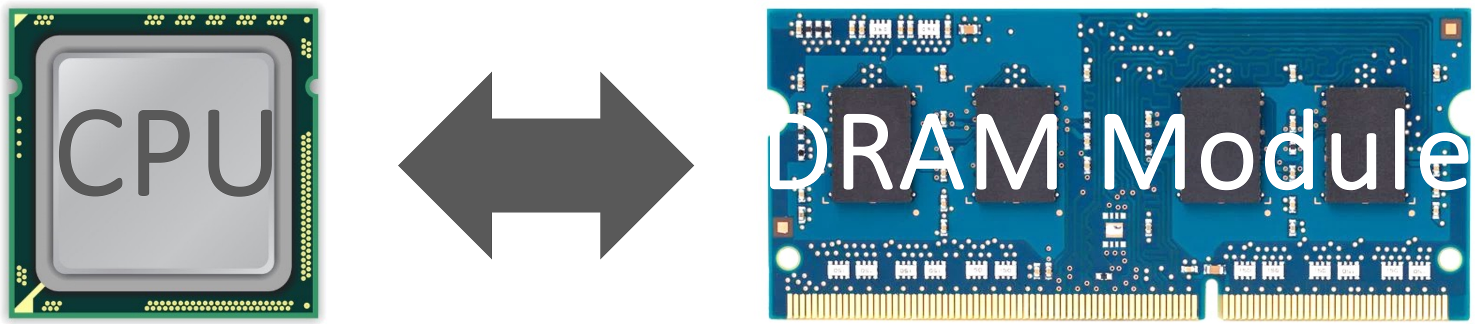



```
loop:
  mov (X), %eax
  mov (Y), %ebx
  clflush (X)
  clflush (Y)
  mfence
  jmp loop
```

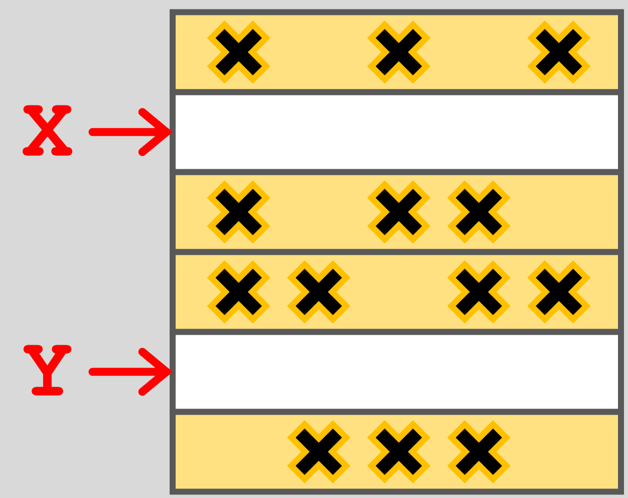


Download from: https://github.com/CMU-SAFARI/rowhammer

A Simple Program Can Induce Many Errors




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



Download from: https://github.com/CMU-SAFARI/rowhammer

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, safety issue

Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.

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.

Exploiting the DRAM rowhammer bug to gain kernel privileges (Seaborn & Dullien, 2015)

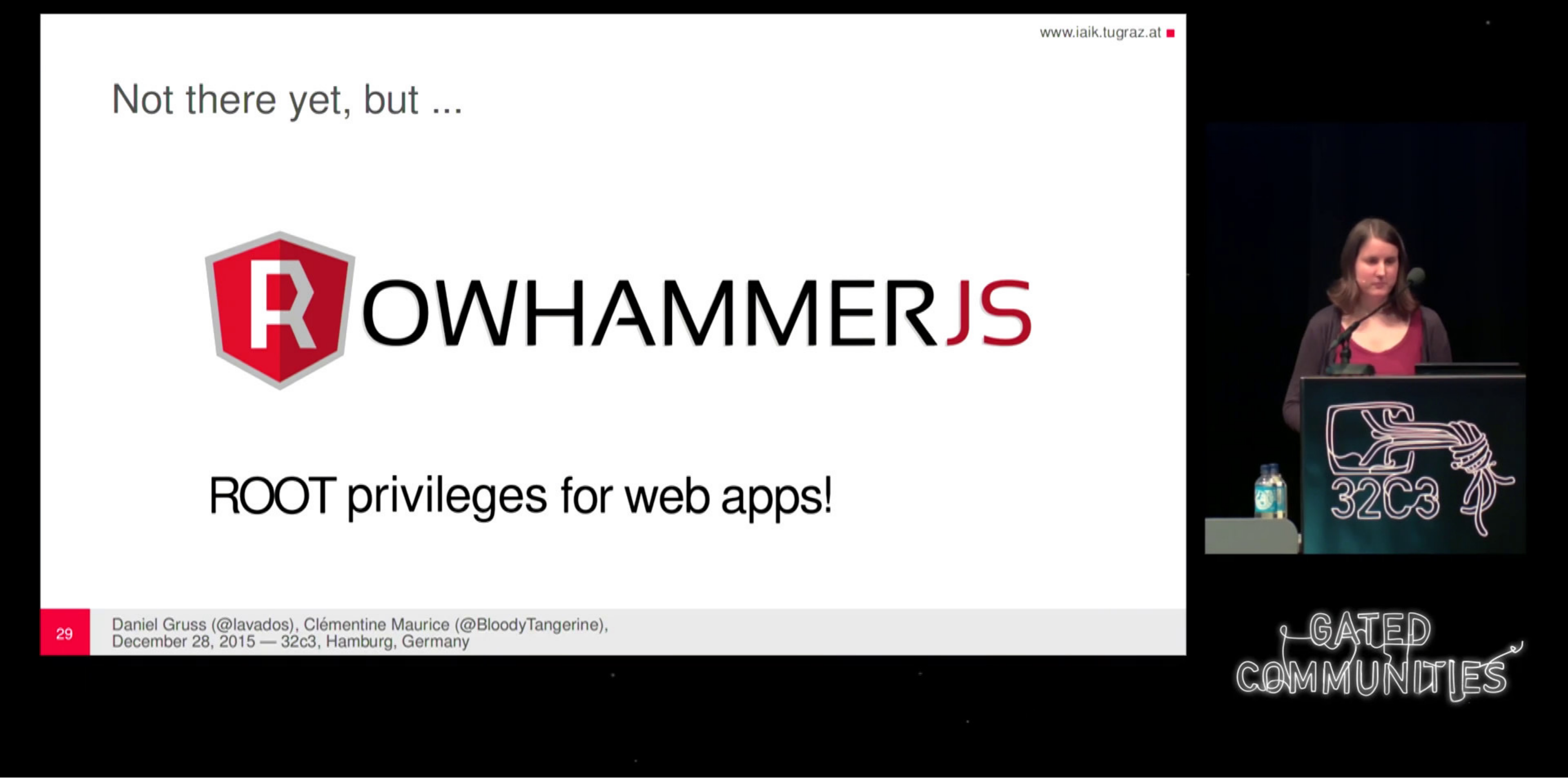
Security Implications

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

More Security Implications (I)

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



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

Source: https://lab.dsst.io/32c3-slides/7197.html

More Security Implications (II)

"Can gain control of a smart phone deterministically"

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

Source: https://fossbytes.com/drammer-rowhammer-attack-android-root-devices/

More Security Implications (III)

- Using an integrated GPU in a mobile system to remotely escalate privilege via the WebGL interface. IEEE S&P 2018

ars TECHNICA — BIZ & IT | TECH | SCIENCE | POLICY | CARS | GAMING & CULTURE

"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) USENIX ATC 2018

ars TECHNICA

BIZ & IT TECH SCIENCE POLICY CARS GAMING & 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)

The Hacker News™
Security in a serious way

Nethammer—Exploiting DRAM Rowhammer Bug Through Network Requests

Nethammer:
Inducing Rowhammer Faults through Network Requests

Moritz Lipp
Graz University of Technology

Misiker Tadesse Aga
University of Michigan

Michael Schwarz
Graz University of Technology

Daniel Gruss
Graz University of Technology

Clémentine Maurice
Univ Rennes, CNRS, IRISA

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

Daniel Genkin
University of Michigan
genkin@umich.edu

Daniel Gruss
Graz University of Technology
daniel.gruss@iaik.tugraz.at

Yuval Yarom
University of Adelaide and Data61
yval@cs.adelaide.edu.au

More Security Implications (VII)

- USENIX Security 2019

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

Sanghyun Hong, Pietro Frigo[†], Yiğitcan Kaya, Cristiano Giuffrida[†], Tudor Dumitraș

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 (VIII)

- USENIX Security 2020

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

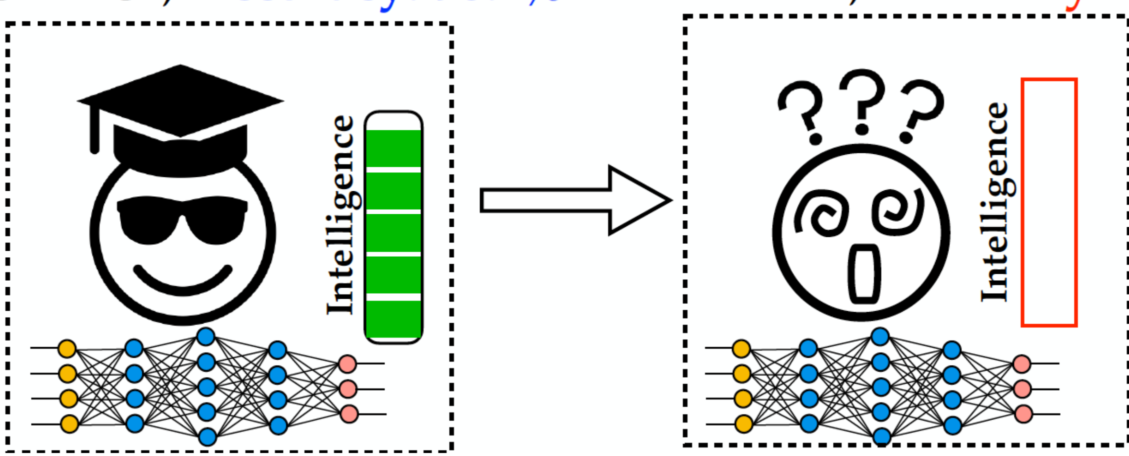
Fan Yao
University of Central Florida
fan.yao@ucf.edu

Adnan Siraj Rakin, Deliang Fan
Arizona State University
asrakin@asu.edu *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)**



Google's Half-Double RowHammer Attack (May 2021)

Google Security Blog

The latest news and insights from Google on security and safety on the Internet

---

Introducing Half-Double: New hammering technique for DRAM Rowhammer bug

May 25, 2021

Research Team: Salman Qazi, Yoongu Kim, Nicolas Boichat, Eric Shiu & Mattias Nissler

Today, we are sharing details around our discovery of Half-Double, a new Rowhammer technique that capitalizes on the worsening physics of some of the newer DRAM chips to alter the contents of memory.

Rowhammer is a DRAM vulnerability whereby repeated accesses to one address can tamper with the data stored at other addresses. Much like speculative execution vulnerabilities in CPUs, Rowhammer is a breach of the security guarantees made by the underlying hardware. As an electrical coupling phenomenon within the silicon itself, Rowhammer allows the potential bypass of hardware and software memory protection policies. This can allow untrusted code to break out of its sandbox and take full control of the system.

https://security.googleblog.com/2021/05/introducing-half-double-new-hammering.html

More Security Implications (VIII)

- **USENIX Security 2022**

- Google's Half-Double RowHammer Attack

Google Security Blog

The latest news and insights from Google on security and safety on the Internet

Introducing Half-Double: New hammering technique for DRAM Rowhammer bug

May 25, 2021

Research Team: Salman Qazi, Yoongu Kim, Nicolas Boichat, Eric Shiu & Mattias Nissler

Today, we are sharing details around our discovery of Half-Double, a new Rowhammer technique that capitalizes on the worsening physics of some of the newer DRAM chips to alter the contents of memory.

Rowhammer is a DRAM vulnerability whereby repeated accesses to one address can tamper with the data stored at other addresses. Much like speculative execution vulnerabilities in CPUs, Rowhammer is a breach of the security guarantees made by the underlying hardware. As an electrical coupling phenomenon within the silicon itself, Rowhammer allows the potential bypass of hardware and software memory protection policies. This can allow untrusted code to break out of its sandbox and take full control of the system.

Half-Double: Hammering From the Next Row Over

Andreas Kogler[1] Jonas Juffinger[1,2] Salman Qazi[3] Yoongu Kim[3] Moritz Lipp[4*]
Nicolas Boichat[3] Eric Shiu[5] Mattias Nissler[3] Daniel Gruss[1]

[1]*Graz University of Technology* [2]*Lamarr Security Research* [3]*Google*
[4]*Amazon Web Services* [5]*Rivos*

https://www.usenix.org/system/files/sec22-kogler-half-double.pdf

More Security Implications?

A RowHammer Survey Across the Stack

- 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[§‡] Jeremie S. Kim[‡§]
[§]ETH Zürich [‡]Carnegie Mellon University

A RowHammer Survey: Recent Update

- Onur Mutlu, Ataberk Olgun, and A. Giray Yaglikci,
  "Fundamentally Understanding and Solving RowHammer"
  *Invited Special Session Paper at the 28th Asia and South Pacific Design Automation Conference (**ASP-DAC**)*, Tokyo, Japan, January 2023.
  [arXiv version]
  [Slides (pptx) (pdf)]
  [Talk Video (26 minutes)]

Fundamentally Understanding and Solving RowHammer

Onur Mutlu
onur.mutlu@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

Ataberk Olgun
ataberk.olgun@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

A. Giray Yağlıkcı
giray.yaglikci@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

https://arxiv.org/pdf/2211.07613.pdf

Understanding RowHammer

First RowHammer Analysis [ISCA 2014]

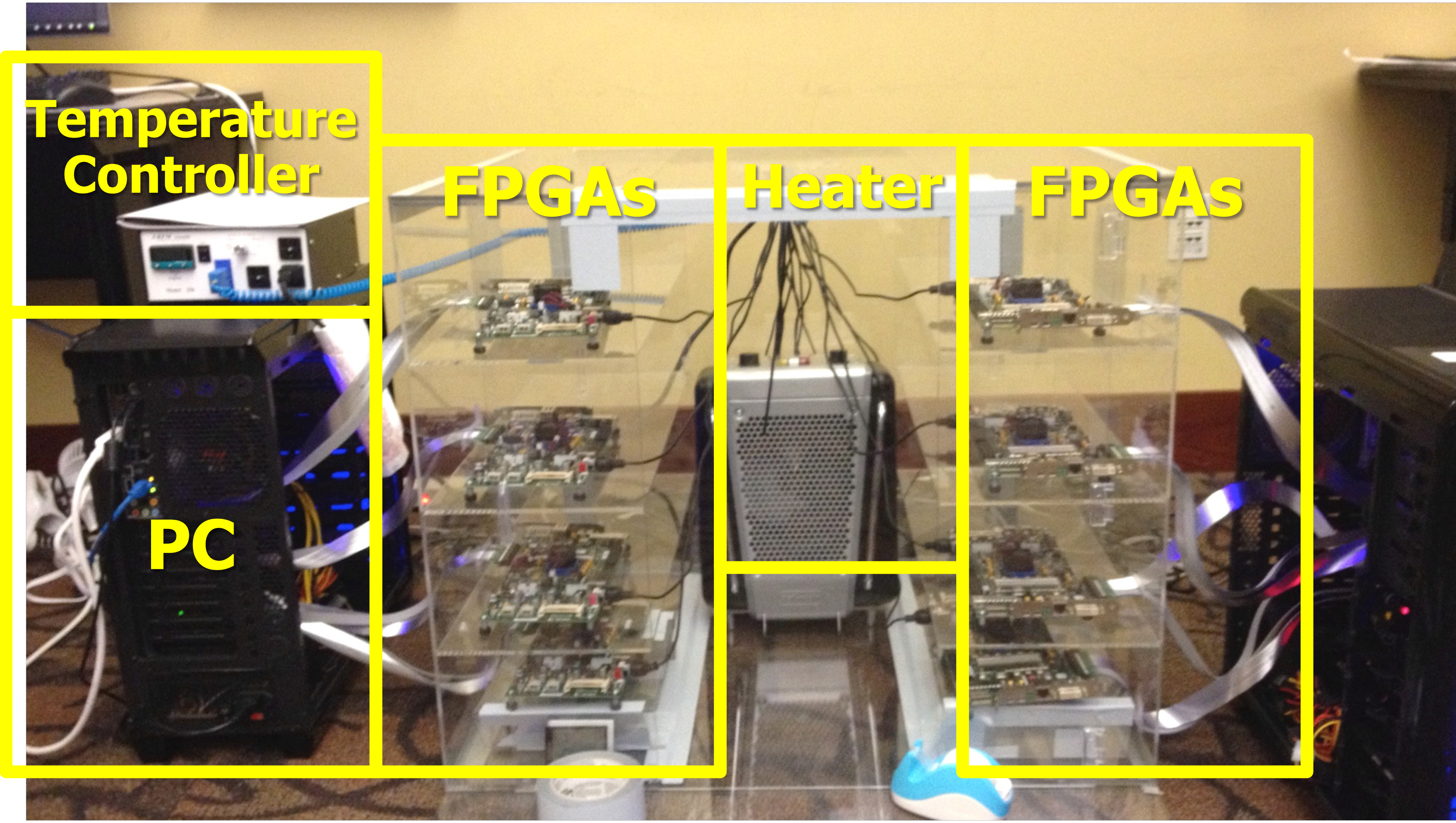
- Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu,
  "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors"
  *Proceedings of the 41st International Symposium on Computer Architecture (**ISCA**)*, Minneapolis, MN, June 2014.
  [Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)] [Source Code and Data] [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).
  Selected to the ISCA-50 25-Year Retrospective Issue covering 1996-2020 in 2023 (Retrospective (pdf) Full Issue).

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

Yoongu Kim[1] Ross Daly* Jeremie Kim[1] Chris Fallin* Ji Hye Lee[1]
Donghyuk Lee[1] Chris Wilkerson[2] Konrad Lai Onur Mutlu[1]

[1]Carnegie Mellon University [2]Intel Labs

RowHammer Infrastructure (2012-2014)



Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.

Tested DRAM Modules from 2008-2014

(129 total)

Manufacturer	Module	Date* (yy-ww)	Timing† Freq (MT/s)	Timing† t_{RC} (ns)	Organization Size (GB)	Organization Chips	Chip Size (Gb)‡	Chip Pins	Chip DieVersion§	Victims-per-Module Average	Victims-per-Module Minimum	Victims-per-Module Maximum	RI_{th} (ms) Min
A Total of 43 Modules	A_1	10-08	1066	50.625	0.5	4	1	×16	$\mathcal{B}$	0	0	0	–
	A_2	10-20	1066	50.625	1	8	1	×8	$\mathcal{F}$	0	0	0	–
	$A_{3\text{-}5}$	10-20	1066	50.625	0.5	4	1	×16	$\mathcal{B}$	0	0	0	–
	$A_{6\text{-}7}$	11-24	1066	49.125	1	4	2	×16	$\mathcal{D}$	7.8×10^1	5.2×10^1	1.0×10^2	21.3
	$A_{8\text{-}12}$	11-26	1066	49.125	1	4	2	×16	$\mathcal{D}$	2.4×10^2	5.4×10^1	4.4×10^2	16.4
	$A_{13\text{-}14}$	11-50	1066	49.125	1	4	2	×16	$\mathcal{D}$	8.8×10^1	1.7×10^1	1.6×10^2	26.2
	$A_{15\text{-}16}$	12-22	1600	50.625	1	4	2	×16	$\mathcal{D}$	9.5	9	1.0×10^1	34.4
	$A_{17\text{-}18}$	12-26	1600	49.125	2	8	2	×8	$\mathcal{M}$	1.2×10^2	3.7×10^1	2.0×10^2	21.3
	$A_{19\text{-}30}$	12-40	1600	48.125	2	8	2	×8	$\mathcal{K}$	8.6×10^6	7.0×10^6	$\mathbf{1.0 \times 10^7}$	**8.2**
	$A_{31\text{-}34}$	13-02	1600	48.125	2	8	2	×8	–	1.8×10^6	1.0×10^6	3.5×10^6	11.5
	$A_{35\text{-}36}$	13-14	1600	48.125	2	8	2	×8	–	4.0×10^1	1.9×10^1	6.1×10^1	21.3
	$A_{37\text{-}38}$	13-20	1600	48.125	2	8	2	×8	$\mathcal{K}$	1.7×10^6	1.4×10^6	2.0×10^6	9.8
	$A_{39\text{-}40}$	13-28	1600	48.125	2	8	2	×8	$\mathcal{K}$	5.7×10^4	5.4×10^4	6.0×10^4	16.4
	A_{41}	14-04	1600	49.125	2	8	2	×8	–	2.7×10^5	2.7×10^5	2.7×10^5	18.0
	$A_{42\text{-}43}$	14-04	1600	48.125	2	8	2	×8	$\mathcal{K}$	0.5	0	1	62.3
B Total of 54 Modules	B_1	08-49	1066	50.625	1	8	1	×8	$\mathcal{D}$	0	0	0	–
	B_2	09-49	1066	50.625	1	8	1	×8	$\mathcal{E}$	0	0	0	–
	B_3	10-19	1066	50.625	1	8	1	×8	$\mathcal{F}$	0	0	0	–
	B_4	10-31	1333	49.125	2	8	2	×8	$\mathcal{C}$	0	0	0	–
	B_5	11-13	1333	49.125	2	8	2	×8	$\mathcal{C}$	0	0	0	–
	B_6	11-16	1066	50.625	1	8	1	×8	$\mathcal{F}$	0	0	0	–
	B_7	11-19	1066	50.625	1	8	1	×8	$\mathcal{F}$	0	0	0	–
	B_8	11-25	1333	49.125	2	8	2	×8	$\mathcal{C}$	0	0	0	–
	B_9	11-37	1333	49.125	2	8	2	×8	$\mathcal{D}$	1.9×10^6	1.9×10^6	1.9×10^6	11.5
	$B_{10\text{-}12}$	11-46	1333	49.125	2	8	2	×8	$\mathcal{D}$	2.2×10^6	1.5×10^6	$\mathbf{2.7 \times 10^6}$	11.5
	B_{13}	11-49	1333	49.125	2	8	2	×8	$\mathcal{C}$	0	0	0	–
	B_{14}	12-01	1866	47.125	2	8	2	×8	$\mathcal{D}$	9.1×10^5	9.1×10^5	9.1×10^5	**9.8**
	$B_{15\text{-}31}$	12-10	1866	47.125	2	8	2	×8	$\mathcal{D}$	9.8×10^5	7.8×10^5	1.2×10^6	11.5
	B_{32}	12-25	1600	48.125	2	8	2	×8	$\mathcal{E}$	7.4×10^5	7.4×10^5	7.4×10^5	11.5
	$B_{33\text{-}42}$	12-28	1600	48.125	2	8	2	×8	$\mathcal{E}$	5.2×10^5	1.9×10^5	7.3×10^5	11.5
	$B_{43\text{-}47}$	12-31	1600	48.125	2	8	2	×8	$\mathcal{E}$	4.0×10^5	2.9×10^5	5.5×10^5	13.1
	$B_{48\text{-}51}$	13-19	1600	48.125	2	8	2	×8	$\mathcal{E}$	1.1×10^5	7.4×10^4	1.4×10^5	14.7
	$B_{52\text{-}53}$	13-40	1333	49.125	2	8	2	×8	$\mathcal{D}$	2.6×10^4	2.3×10^4	2.9×10^4	21.3
	B_{54}	14-07	1333	49.125	2	8	2	×8	$\mathcal{D}$	7.5×10^3	7.5×10^3	7.5×10^3	26.2
C Total of 32 Modules	C_1	10-18	1333	49.125	2	8	2	×8	$\mathcal{A}$	0	0	0	–
	C_2	10-20	1066	50.625	2	8	2	×8	$\mathcal{A}$	0	0	0	–
	C_3	10-22	1066	50.625	2	8	2	×8	$\mathcal{A}$	0	0	0	–
	$C_{4\text{-}5}$	10-26	1333	49.125	2	8	2	×8	$\mathcal{B}$	8.9×10^2	6.0×10^2	1.2×10^3	29.5
	C_6	10-43	1333	49.125	1	8	1	×8	$\mathcal{T}$	0	0	0	–
	C_7	10-51	1333	49.125	2	8	2	×8	$\mathcal{B}$	4.0×10^2	4.0×10^2	4.0×10^2	29.5
	C_8	11-12	1333	46.25	2	8	2	×8	$\mathcal{B}$	6.9×10^2	6.9×10^2	6.9×10^2	21.3
	C_9	11-19	1333	46.25	2	8	2	×8	$\mathcal{B}$	9.2×10^2	9.2×10^2	9.2×10^2	27.9
	C_{10}	11-31	1333	49.125	2	8	2	×8	$\mathcal{B}$	3	3	3	39.3
	C_{11}	11-42	1333	49.125	2	8	2	×8	$\mathcal{B}$	1.6×10^2	1.6×10^2	1.6×10^2	39.3
	C_{12}	11-48	1600	48.125	2	8	2	×8	$\mathcal{C}$	7.1×10^4	7.1×10^4	7.1×10^4	19.7
	C_{13}	12-08	1333	49.125	2	8	2	×8	$\mathcal{C}$	3.9×10^4	3.9×10^4	3.9×10^4	21.3
	$C_{14\text{-}15}$	12-12	1333	49.125	2	8	2	×8	$\mathcal{C}$	3.7×10^4	2.1×10^4	5.4×10^4	21.3
	$C_{16\text{-}18}$	12-20	1600	48.125	2	8	2	×8	$\mathcal{C}$	3.5×10^3	1.2×10^3	7.0×10^3	27.9
	C_{19}	12-23	1600	48.125	2	8	2	×8	$\mathcal{E}$	1.4×10^5	1.4×10^5	1.4×10^5	18.0
	C_{20}	12-24	1600	48.125	2	8	2	×8	$\mathcal{C}$	6.5×10^4	6.5×10^4	6.5×10^4	21.3
	C_{21}	12-26	1600	48.125	2	8	2	×8	$\mathcal{C}$	2.3×10^4	2.3×10^4	2.3×10^4	24.6
	C_{22}	12-32	1600	48.125	2	8	2	×8	$\mathcal{C}$	1.7×10^4	1.7×10^4	1.7×10^4	22.9
	$C_{23\text{-}24}$	12-37	1600	48.125	2	8	2	×8	$\mathcal{C}$	2.3×10^4	1.1×10^4	3.4×10^4	18.0
	$C_{25\text{-}30}$	12-41	1600	48.125	2	8	2	×8	$\mathcal{C}$	2.0×10^4	1.1×10^4	3.2×10^4	19.7
	C_{31}	13-11	1600	48.125	2	8	2	×8	$\mathcal{C}$	3.3×10^5	3.3×10^5	$\mathbf{3.3 \times 10^5}$	**14.7**
	C_{32}	13-35	1600	48.125	2	8	2	×8	$\mathcal{C}$	3.7×10^4	3.7×10^4	3.7×10^4	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 chip size supported by our testing platform is 2Gb.
§ We report DRAM die versions marked on the chip packages, which typically progress in the following manner: $\mathcal{M} \rightarrow \mathcal{A} \rightarrow \mathcal{B} \rightarrow \mathcal{C} \rightarrow \cdots$.

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

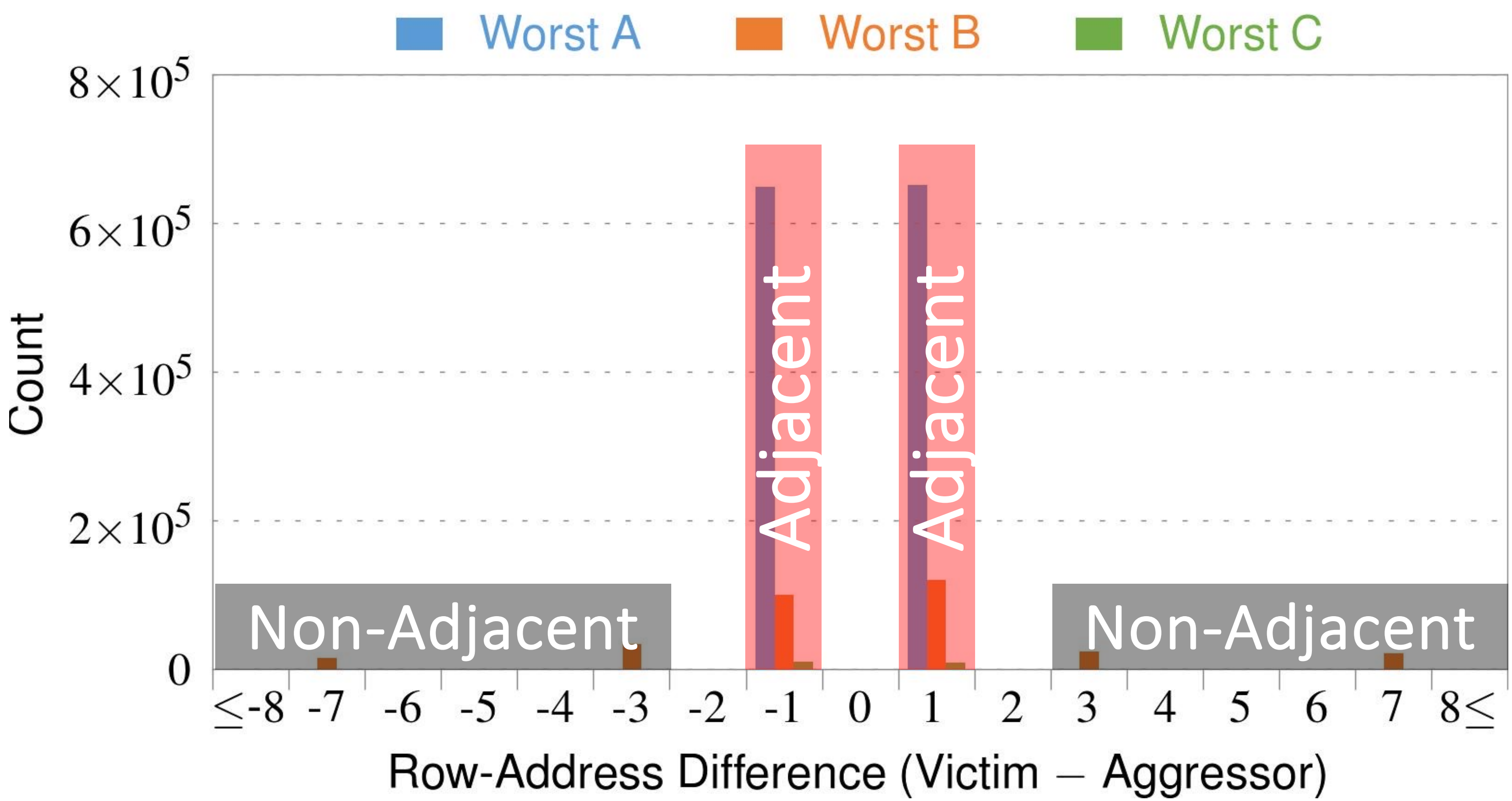
SAFARI

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

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

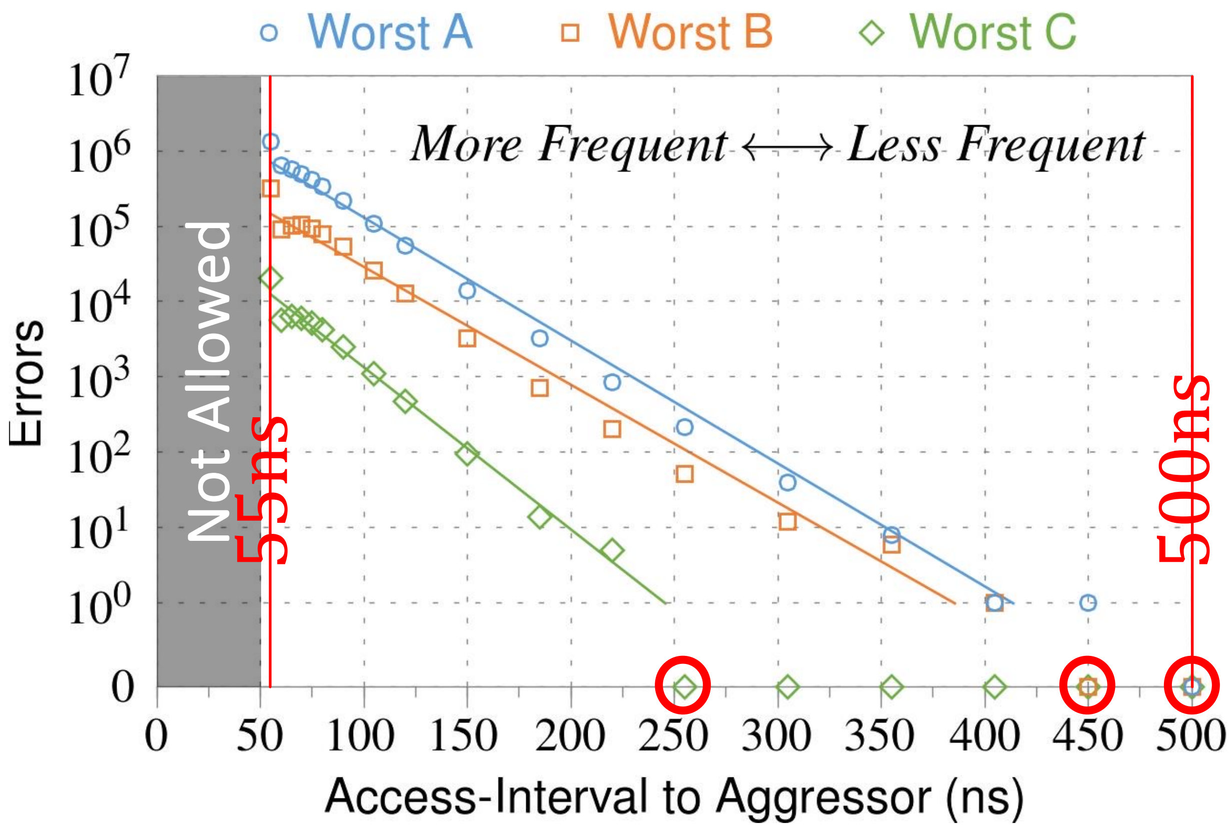
4. Adjacency: Aggressor & Victim



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

Most aggressors & victims are adjacent

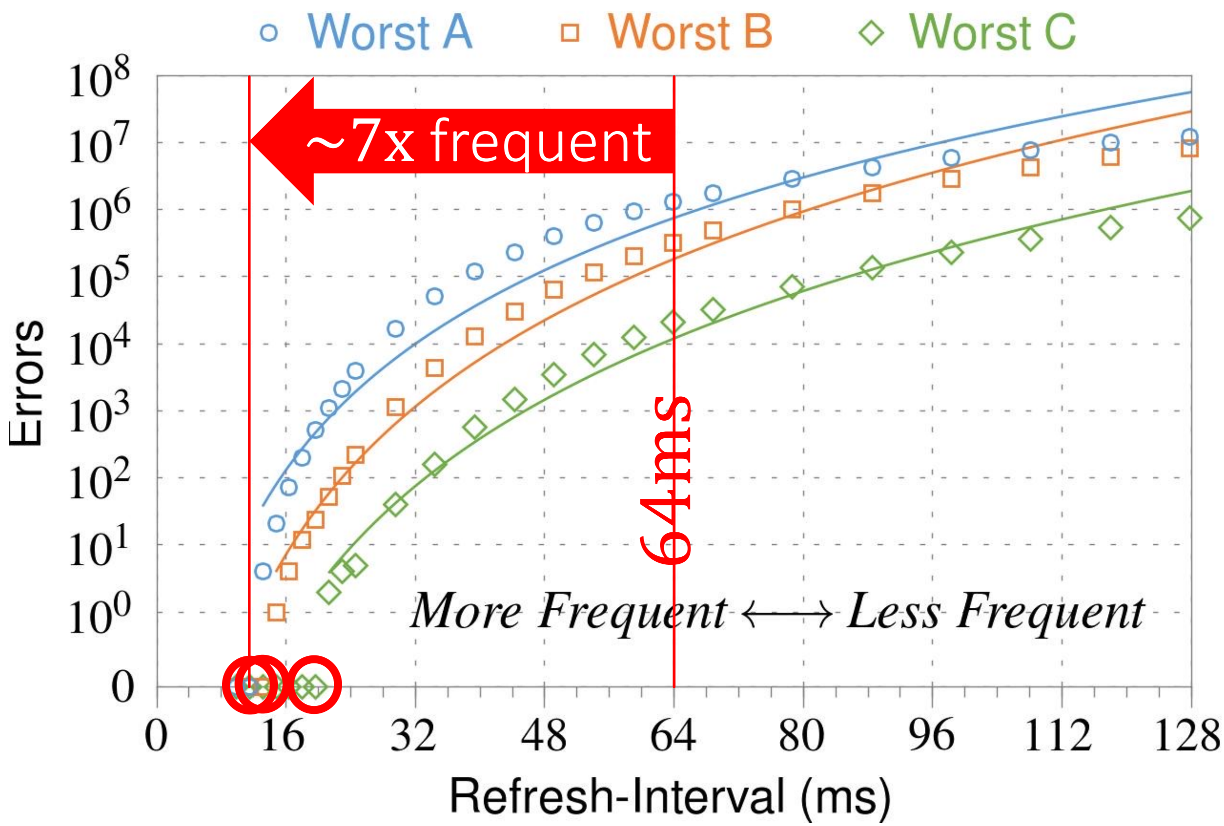
❶ Access Interval (Aggressor)



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

Less frequent accesses → Fewer errors

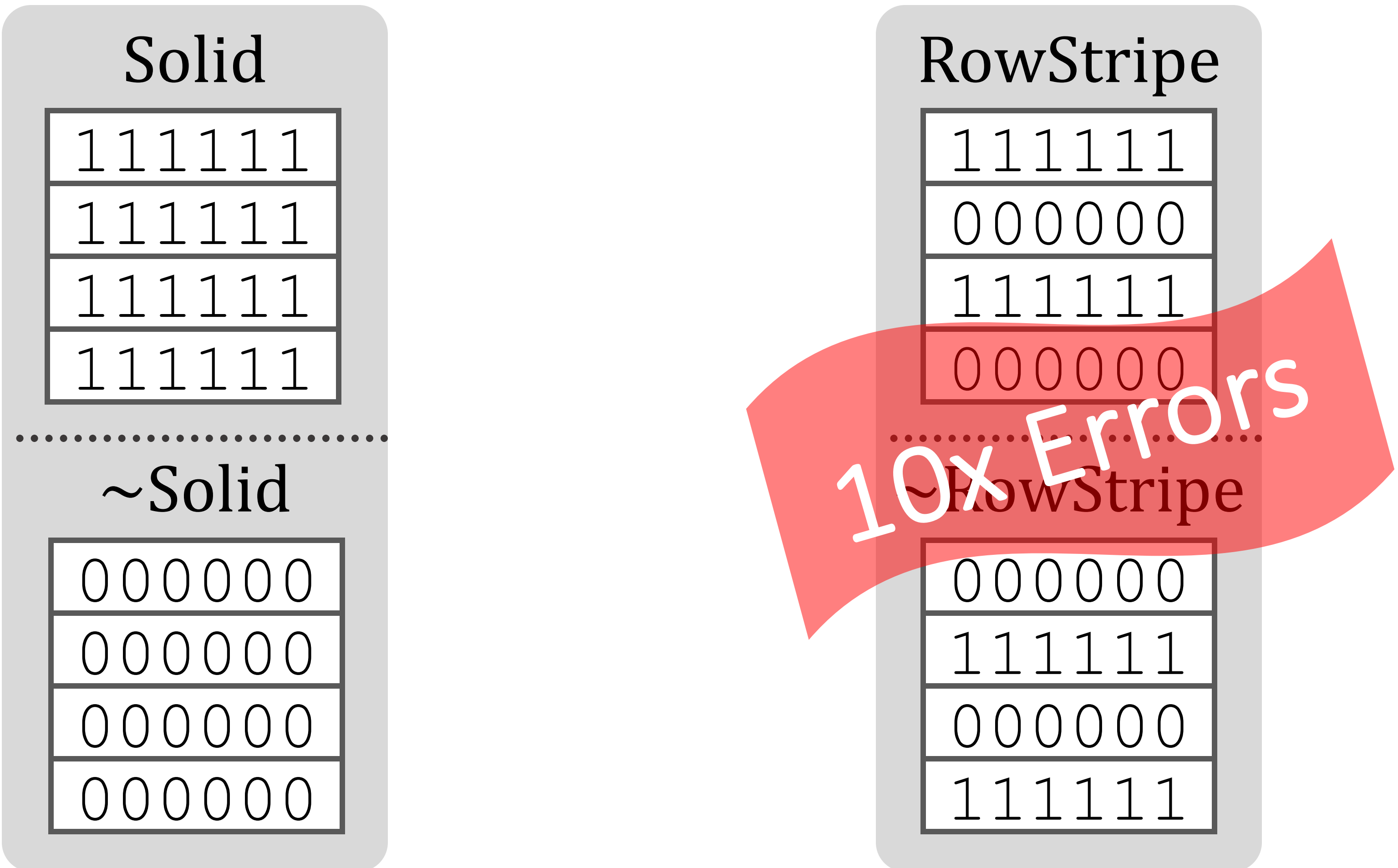
❷ Refresh Interval



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

More frequent refreshes → Fewer errors

❸ Data Pattern



Errors affected by data stored in other cells

6. Other Key Observations [ISCA'14]

- *Victim Cells $\neq$ Retention-Weak Cells*
  - Almost no overlap between them
- *Errors are repeatable*
  - Across ten iterations of testing, >70% of victim cells had errors in every iteration
- *As many as 4 errors per cache-line*
  - Simple ECC (e.g., SECDED) cannot prevent all errors
- *Cells affected by two aggressors on either side*
  - Double sided hammering

Major RowHammer Characteristics (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] [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).
  Selected to the ISCA-50 25-Year Retrospective Issue covering 1996-2020 in 2023 (Retrospective (pdf) Full Issue).

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

Yoongu Kim[1] Ross Daly* Jeremie Kim[1] Chris Fallin* Ji Hye Lee[1]
Donghyuk Lee[1] Chris Wilkerson[2] Konrad Lai Onur Mutlu[1]

[1]Carnegie Mellon University [2]Intel Labs

RowHammer is Getting Much Worse (2020)

- 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[§†] Minesh Patel[§] A. Giray Yağlıkçı[§]
Hasan Hassan[§] Roknoddin Azizi[§] Lois Orosa[§] Onur Mutlu[§†]

[§]*ETH Zürich* [†]*Carnegie Mellon University*

RowHammer Has Many Dimensions (2021)

- Lois Orosa, Abdullah Giray Yaglikci, Haocong Luo, Ataberk Olgun, Jisung Park, Hasan Hassan, Minesh Patel, Jeremie S. Kim, and Onur Mutlu,
  "A Deeper Look into RowHammer's Sensitivities: Experimental Analysis of Real DRAM Chips and Implications on Future Attacks and Defenses"
  Proceedings of the 54th International Symposium on Microarchitecture (***MICRO***), Virtual, October 2021.
  [Slides (pptx) (pdf)]
  [Short Talk Slides (pptx) (pdf)]
  [Lightning Talk Slides (pptx) (pdf)]
  [Talk Video (21 minutes)]
  [Lightning Talk Video (1.5 minutes)]
  [arXiv version]

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

RowHammer vs. Wordline Voltage (2022)

- A. Giray Yağlıkçı, Haocong Luo, Geraldo F. de Oliviera, Ataberk Olgun, Minesh Patel, Jisung Park, Hasan Hassan, Jeremie S. Kim, Lois Orosa, and Onur Mutlu, **"Understanding RowHammer Under Reduced Wordline Voltage: An Experimental Study Using Real DRAM Devices"**
*Proceedings of the 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (**DSN**)*, Baltimore, MD, USA, June 2022.
[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[arXiv version]
[Talk Video (34 minutes, including Q&A)]
[Lightning Talk Video (2 minutes)]

Understanding RowHammer Under Reduced Wordline Voltage: An Experimental Study Using Real DRAM Devices

A. Giray Yağlıkçı[1] Haocong Luo[1] Geraldo F. de Oliviera[1] Ataberk Olgun[1] Minesh Patel[1]
Jisung Park[1] Hasan Hassan[1] Jeremie S. Kim[1] Lois Orosa[1,2] Onur Mutlu[1]
[1] *ETH Zürich* [2] *Galicia Supercomputing Center (CESGA)*

RowHammer in HBM Chips (2023)

- Ataberk Olgun, Majd Osserian, A. Giray Yağlıkçı, Yahya Can Tugrul, Haocong Luo, Steve Rhyner, Behzad Salami, Juan Gomez-Luna, and Onur Mutlu,
  "An Experimental Analysis of RowHammer in HBM2 DRAM Chips"
  *Proceedings of the 53nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Disrupt Track (**DSN Disrupt**)*, Porto, Portugal, June 2023.
  [arXiv version]
  [Slides (pptx) (pdf)]
  [Talk Video (24 minutes, including Q&A)]

An Experimental Analysis of RowHammer in HBM2 DRAM Chips

Ataberk Olgun[1] Majd Osseiran[1,2] A. Giray Yağlıkçı[1] Yahya Can Tuğrul[1]
Haocong Luo[1] Steve Rhyner[1] Behzad Salami[1] Juan Gomez Luna[1] Onur Mutlu[1]
[1]*SAFARI Research Group, ETH Zürich* [2]*American University of Beirut*

https://arxiv.org/pdf/2305.17918.pdf

RowHammer in HBM Chips (2024)

- **Appears at DSN 2024**

Read Disturbance in High Bandwidth Memory: A Detailed Experimental Study on HBM2 DRAM Chips

Ataberk Olgun[1] Majd Osseiran[1] A. Giray Yağlıkçı[1] Yahya Can Tuğrul[1]
Haocong Luo[1] Steve Rhyner[1] Behzad Salami[2] Juan Gomez Luna[1] Onur Mutlu[1]
[1]*ETH Zürich* [2]*BSC*

https://arxiv.org/pdf/2310.14665

RowHammer Spatial Variation Analysis (2024)

- **Appears at HPCA 2024**

Spatial Variation-Aware Read Disturbance Defenses: Experimental Analysis of Real DRAM Chips and Implications on Future Solutions

Abdullah Giray Yağlıkçı   Yahya Can Tuğrul   Geraldo F. Oliveira
İsmail Emir Yüksel   Ataberk Olgun   Haocong Luo   Onur Mutlu
ETH Zürich

https://arxiv.org/pdf/2402.18652

RowHammer Solutions

Two Types of RowHammer Solutions

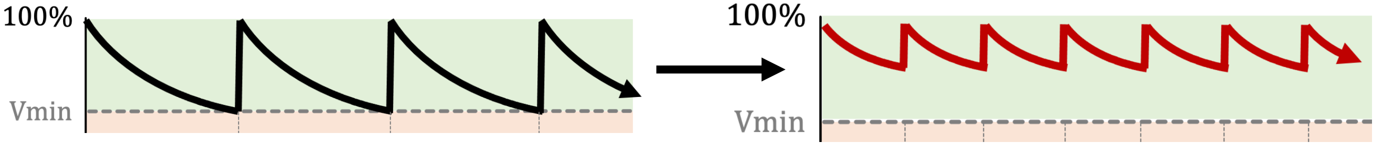
- Immediate
  - To protect the vulnerable DRAM chips in the field
  - Limited possibilities

- 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

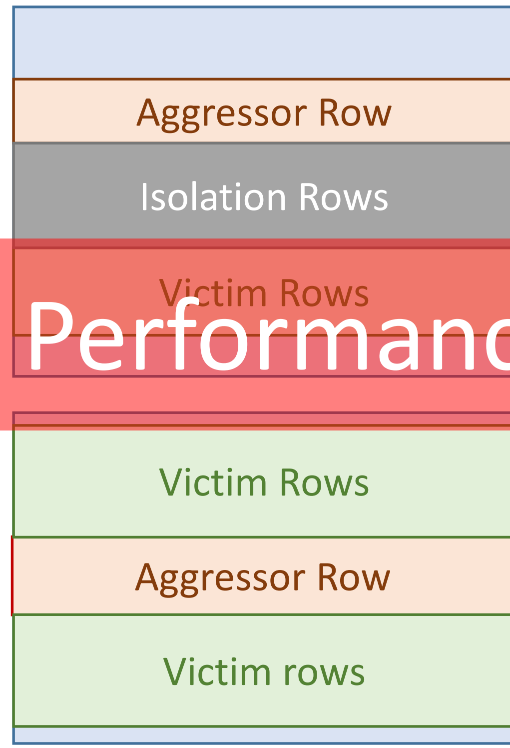
RowHammer Solution Approaches

- More robust DRAM chips **and/or** error-correcting codes
- Increased refresh rate



Fewer activations possible in a refresh interval

- Physical isolation



Large-enough distance

Cost, Power, Performance, Complexity

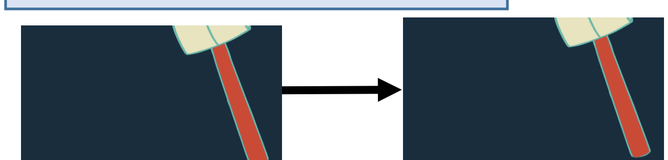
- Reactive refresh

Refresh

Rapidly activated (hammered)

Refresh

- Proactive throttling



Fewer activations allowed for aggressive applications

Apple's Security 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 many other vendors released similar patches

Our First Solution to RowHammer

- **PARA:** *Probabilistic Adjacent Row Activation*
- Key Idea
  - After closing a row, activate (i.e., refresh) its neighbors with a low probability: $p = 0.005$
- Reliability Guarantee
  - When $p=0.005$, errors in one year: 9.4×10^{-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

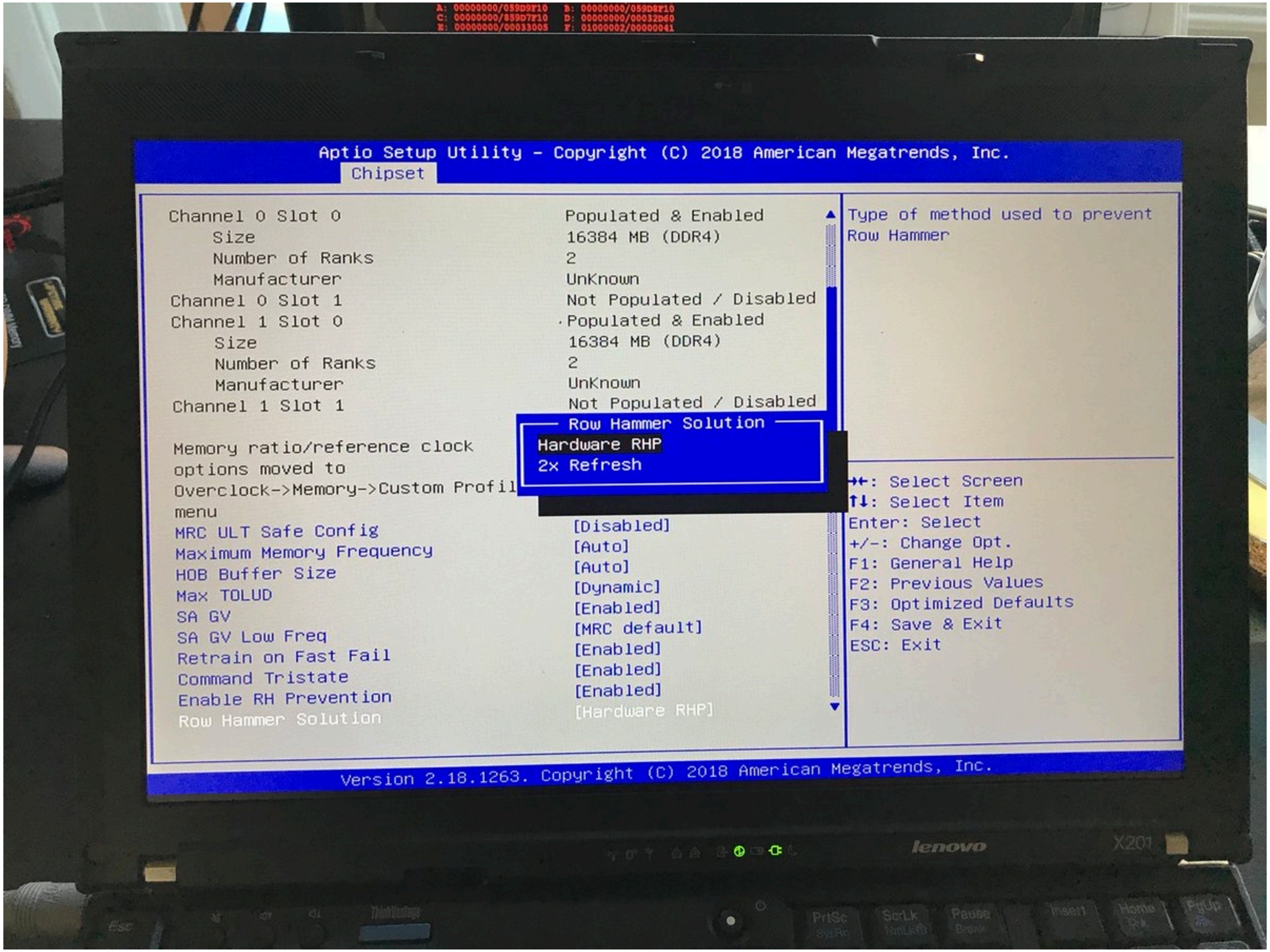
- *PARA is an effective and low-overhead solution to prevent disturbance errors*

Requirements for PARA

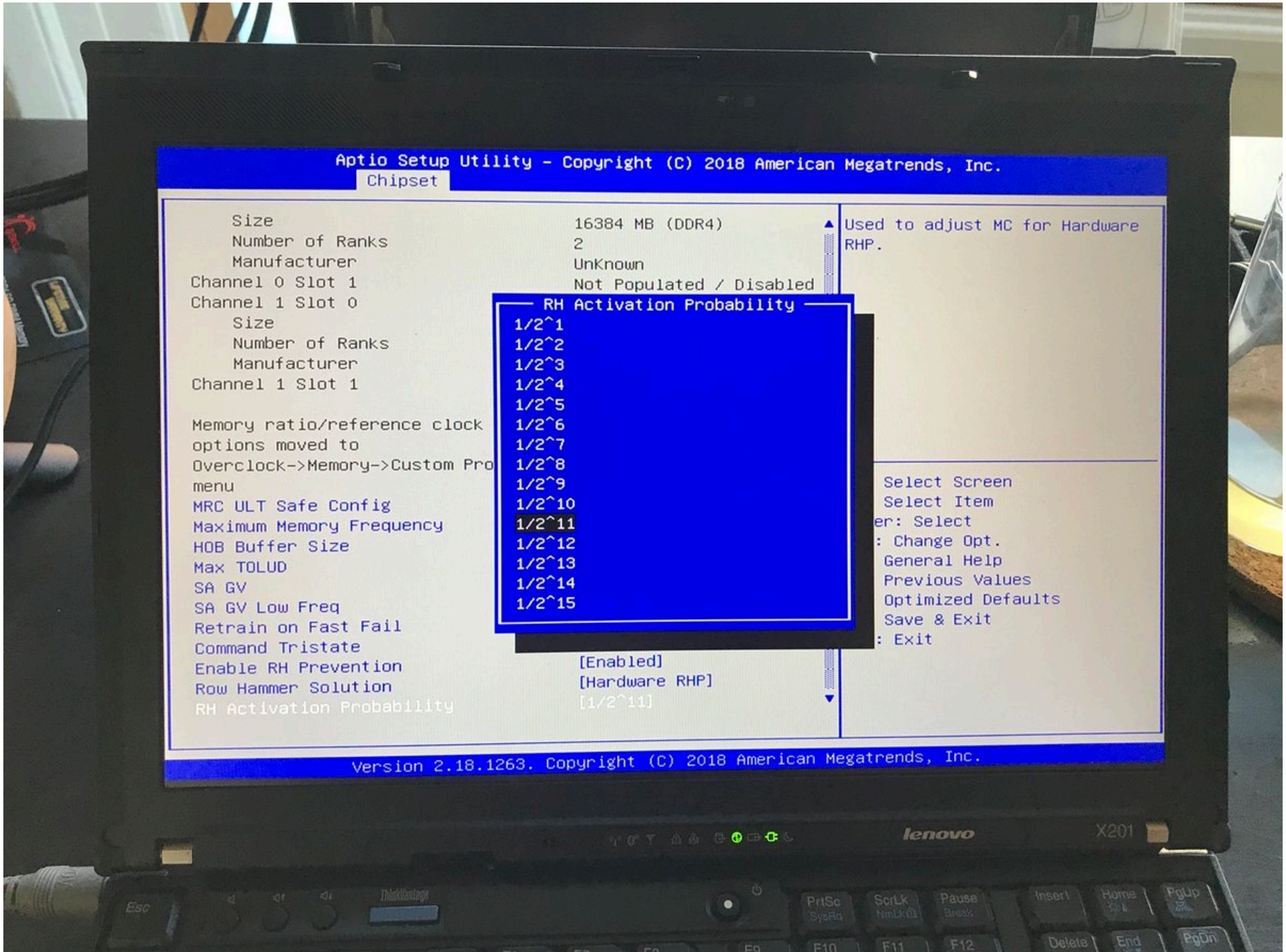
- If implemented in DRAM chip (done today)
  - Enough slack in timing and refresh parameters
  - Plenty of slack today:
    - Lee et al., "Adaptive-Latency DRAM: Optimizing DRAM Timing for the Common Case," HPCA 2015.
    - Chang et al., "Understanding Latency Variation in Modern DRAM Chips," SIGMETRICS 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
  - Need coordination between 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).
Selected to the ISCA-50 25-Year Retrospective Issue covering 1996-2020 in 2023 (Retrospective (pdf) Full Issue).

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

Yoongu Kim[1] Ross Daly* Jeremie Kim[1] Chris Fallin* Ji Hye Lee[1]
Donghyuk Lee[1] Chris Wilkerson[2] Konrad Lai Onur Mutlu[1]

[1]Carnegie Mellon University [2]Intel Labs

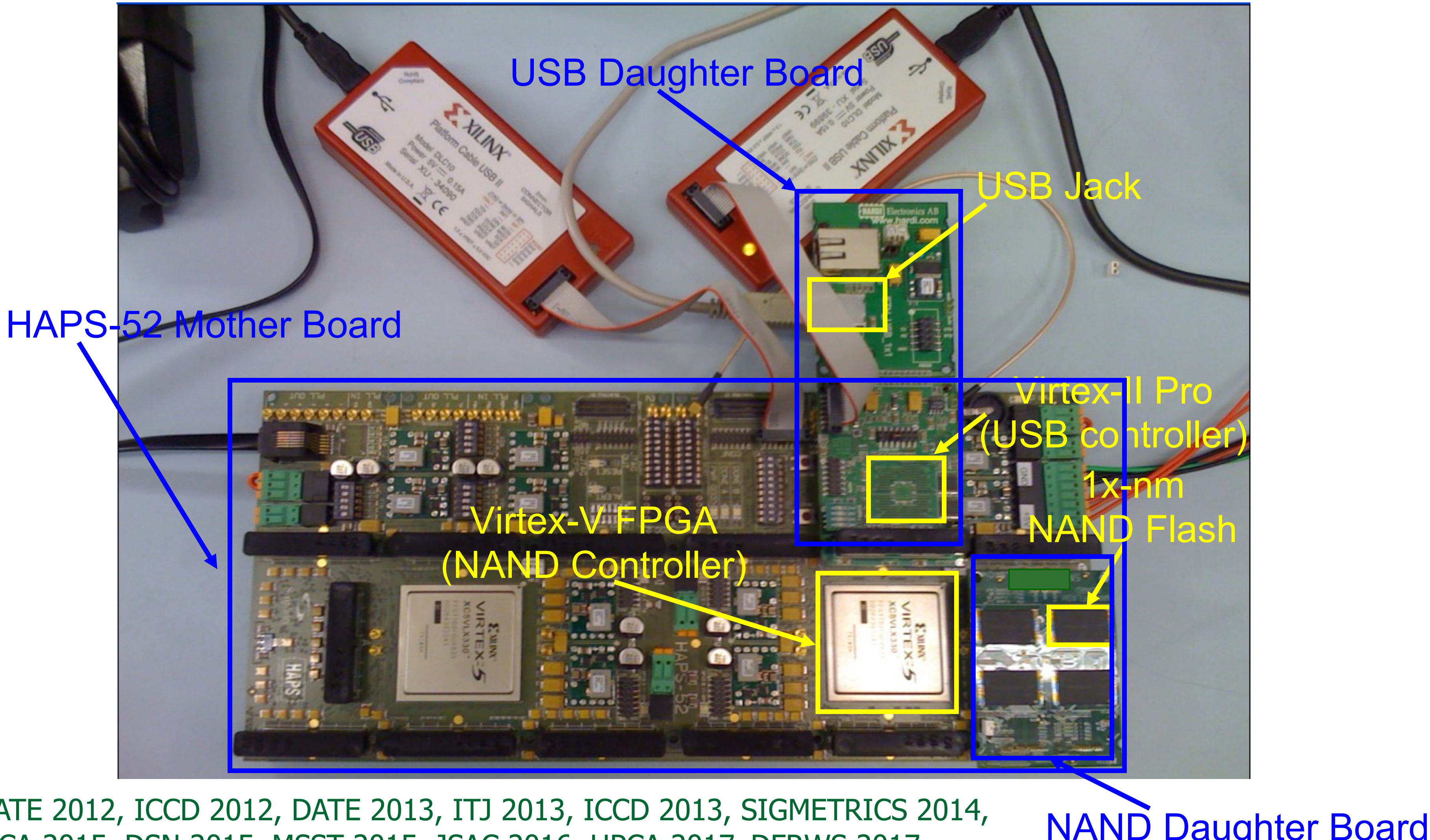
A Takeaway

Main Memory Needs

Intelligent Controllers

for Security, Safety, Reliability, Scaling

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]

Cai+, "Error Characterization, Mitigation, and Recovery in Flash Memory Based Solid State Drives," Proc. IEEE 2017.

Intelligent Flash Controllers **[PIEEE'17]**

INVITED PAPER

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

Two Major RowHammer Directions

- **Understanding RowHammer**
  - Many effects still need to be rigorously examined
    - Aging of DRAM Chips
    - Environmental Conditions (e.g., Process, Voltage, Temperature)
    - Memory Access Patterns
    - Memory Controller & System Design Decisions
    - …

- **Solving RowHammer**
  - Flexible and efficient solutions are necessary
    - In-field patchable / reconfigurable / programmable solutions
  - Co-architecting System and Memory is important
    - To avoid performance and denial-of-service problems

RowHammer in 2020-2024

Revisiting RowHammer

RowHammer is Getting Much Worse

- 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[§,†] Minesh Patel[§] A. Giray Yağlıkçı[§]
Hasan Hassan[§] Roknoddin Azizi[§] Lois Orosa[§] Onur Mutlu[§,†]

[§]*ETH Zürich* [†]*Carnegie Mellon University*

Key Takeaways from 1580 Chips

- **Newer DRAM chips are much more vulnerable to RowHammer (more bit flips, happening earlier)**

- There are new chips 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.

- **Existing mitigation mechanisms are NOT effective at future technology nodes**

1580 DRAM Chips Tested

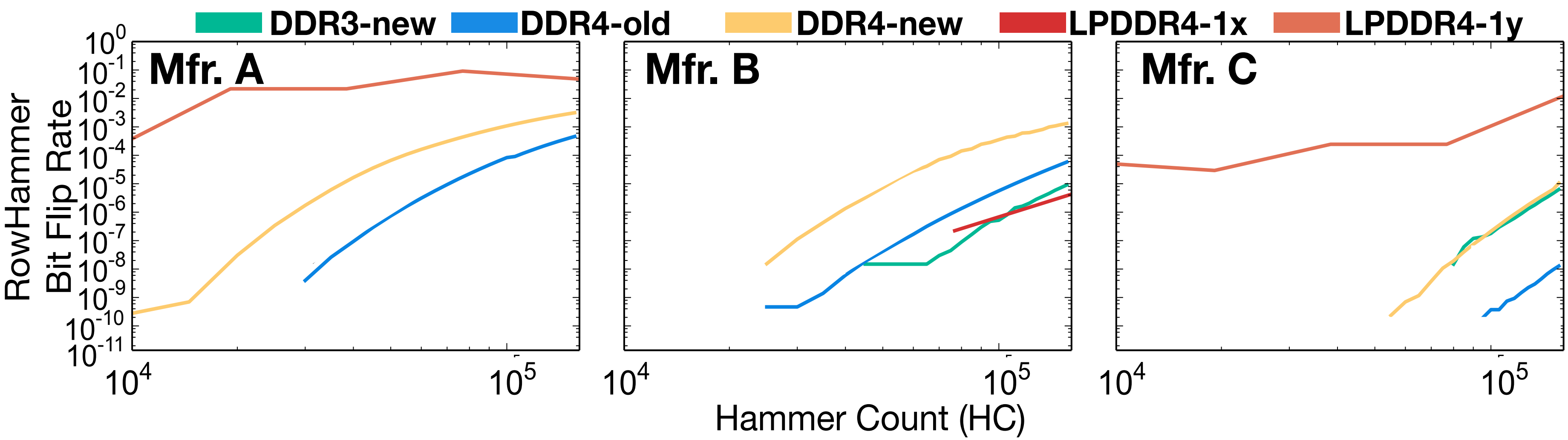
DRAM type-node	Number of Chips (Modules) Tested			
	Mfr. A	*Mfr. B*	*Mfr. C*	*Total*
DDR3-old	56 (10)	88 (11)	28 (7)	**172 (28)**
DDR3-new	80 (10)	52 (9)	104 (13)	**236 (32)**
DDR4-old	112 (16)	24 (3)	128 (18)	**264 (37)**
DDR4-new	264 (43)	16 (2)	108 (28)	**388 (73)**
LPDDR4-1x	12 (3)	180 (45)	N/A	**192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

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**

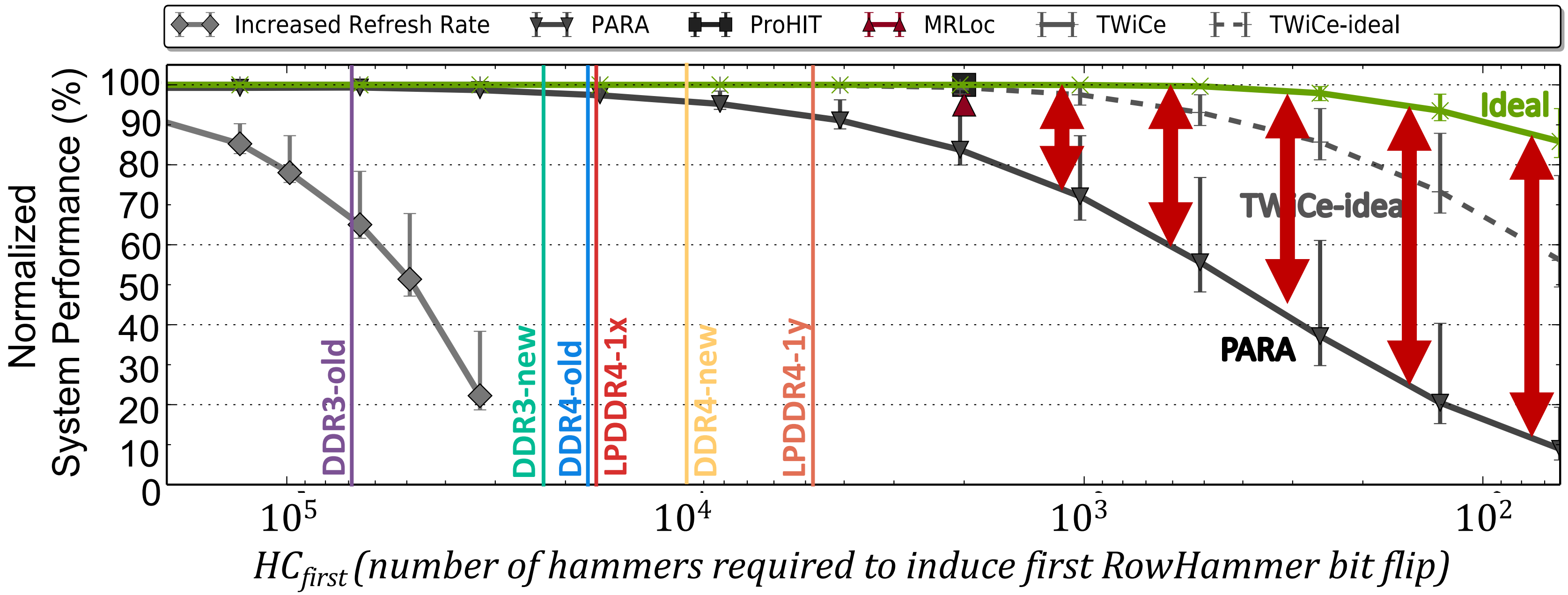
3. Hammer Count (HC) Effects



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

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}

New RowHammer Characteristics

RowHammer Has Many Dimensions

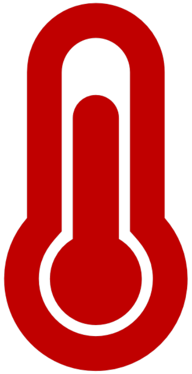
- Lois Orosa, Abdullah Giray Yaglikci, Haocong Luo, Ataberk Olgun, Jisung Park, Hasan Hassan, Minesh Patel, Jeremie S. Kim, and Onur Mutlu,
  "A Deeper Look into RowHammer's Sensitivities: Experimental Analysis of Real DRAM Chips and Implications on Future Attacks and Defenses"
  Proceedings of the 54th International Symposium on Microarchitecture (***MICRO***), Virtual, October 2021.
  [Slides (pptx) (pdf)]
  [Short Talk Slides (pptx) (pdf)]
  [Lightning Talk Slides (pptx) (pdf)]
  [Talk Video (21 minutes)]
  [Lightning Talk Video (1.5 minutes)]
  [arXiv version]

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

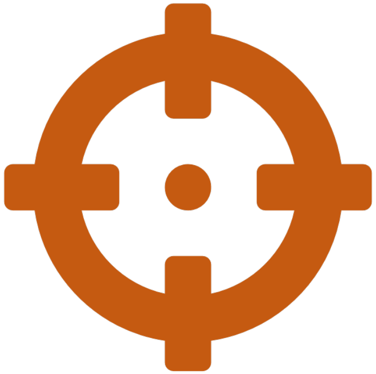
Our Goal

Provide insights into **three fundamental properties**



Temperature

Aggressor Row Active Time



Victim DRAM Cell's Physical Location

To find **effective and efficient** attacks and defenses

Summary of The Study & Key Results

- **272 DRAM chips** from **four major manufacturers**

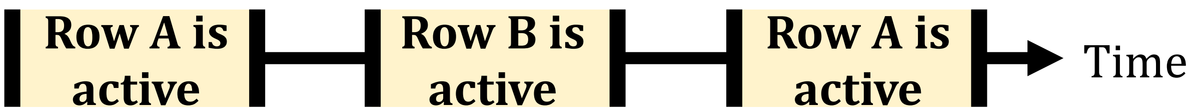
- **6 major takeaways** from **16 novel observations**

- A RowHammer bit flip is **more likely to occur**
  1) in **a bounded range of temperature**
  2) if the aggressor row is **active for longer time**
  3) in **certain physical regions** of the DRAM module under attack

- Our novel observations can inspire and aid future work
  - Craft **more effective attacks**
  - Design **more effective and efficient defenses**

Example Attack Improvement 3:
Bypassing Defenses with Aggressor Row Active Time

Activating aggressor rows as frequently as possible:



Keeping aggressor rows active for a longer time:

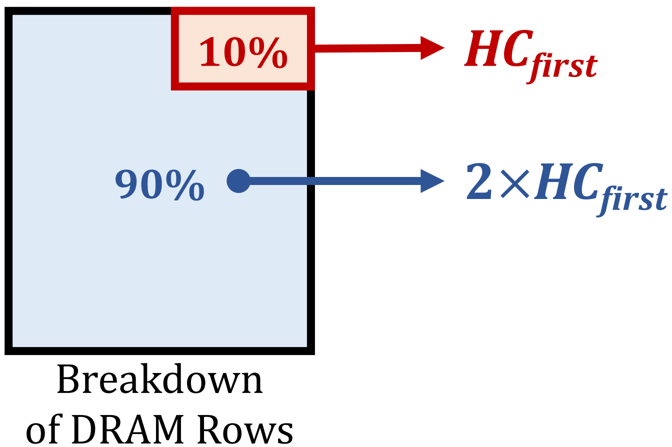
36% reduction in HC_{first}

Reduces the minimum activation count to induce a bit flip **by 36%**

Bypasses defenses that do not account for this reduction

Example Defense Improvements

- **Example 1: Leveraging variation across DRAM rows**



Aggressiveness can be reduced:

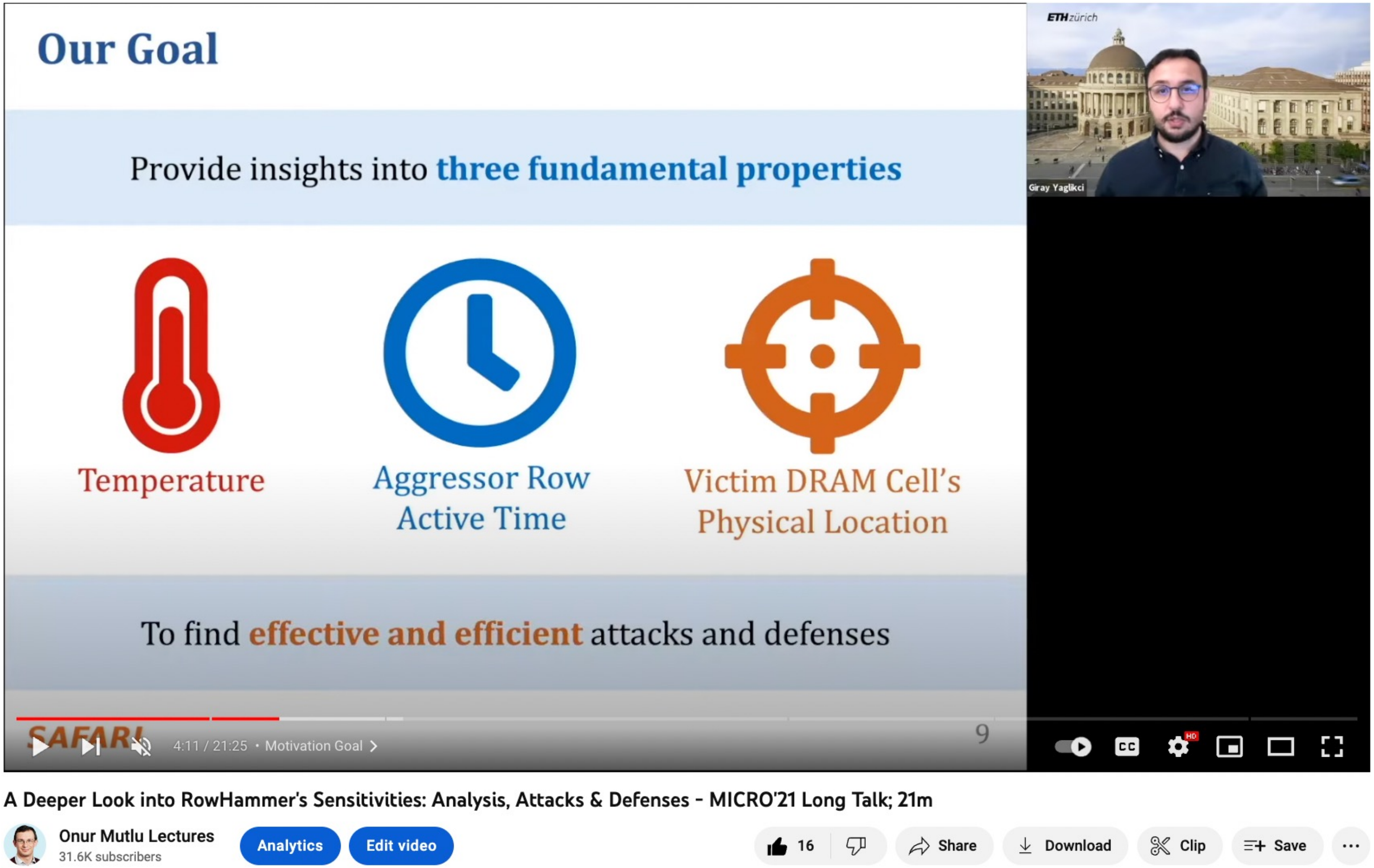
33% area reduction
for BlockHammer [Yağlıkçı+, HPCA'21]

80% area reduction
for Graphene [Park+, MICRO'20]

- **Example 2: Leveraging variation with temperature**
  - A DRAM cell experiences **bit flips** within **a bounded temperature range**

  - A row can be **disabled** within the row's **vulnerable temperature range**

Deeper Look into RowHammer: Talk Video



https://youtube.com/watch?v=fkM32oA0u6U&si=EnSIkaIECMiOmarE

More RowHammer Analysis

RowHammer vs. Wordline Voltage (2022)

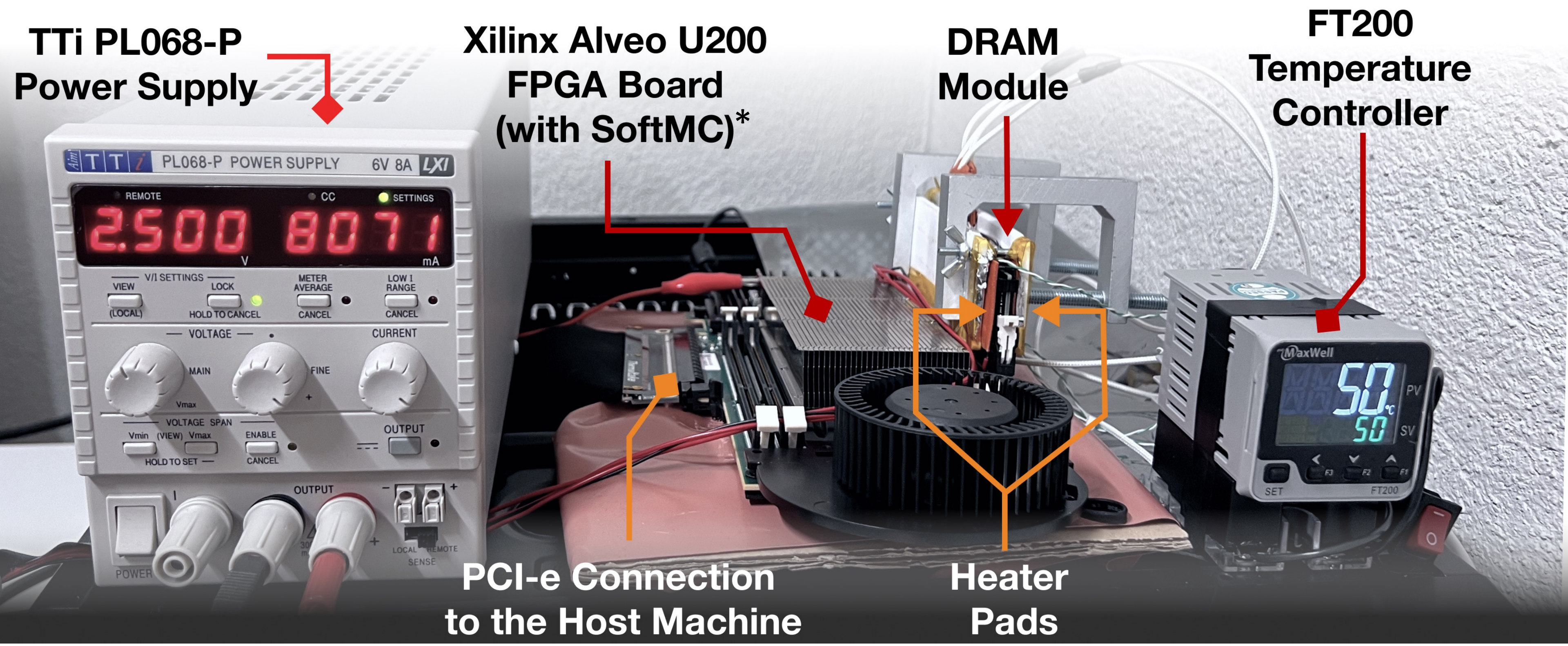
- A. Giray Yağlıkçı, Haocong Luo, Geraldo F. de Oliviera, Ataberk Olgun, Minesh Patel, Jisung Park, Hasan Hassan, Jeremie S. Kim, Lois Orosa, and Onur Mutlu, **"Understanding RowHammer Under Reduced Wordline Voltage: An Experimental Study Using Real DRAM Devices"**
Proceedings of the 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (***DSN***), Baltimore, MD, USA, June 2022.
[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[arXiv version]
[Talk Video (34 minutes, including Q&A)]
[Lightning Talk Video (2 minutes)]

Understanding RowHammer Under Reduced Wordline Voltage: An Experimental Study Using Real DRAM Devices

A. Giray Yağlıkçı[1] Haocong Luo[1] Geraldo F. de Oliviera[1] Ataberk Olgun[1] Minesh Patel[1]
Jisung Park[1] Hasan Hassan[1] Jeremie S. Kim[1] Lois Orosa[1,2] Onur Mutlu[1]
[1]*ETH Zürich* [2]*Galicia Supercomputing Center (CESGA)*

Updated DRAM Testing Infrastructure

FPGA-based SoftMC (Xilinx Virtex UltraScale+ XCU200)



Fine-grained control over **DRAM commands**, **timing parameters (±1.5ns)**, **temperature (±0.1°C)**, and **wordline voltage (±1mV)**

*Hassan et al., "[SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies](#)," in HPCA, 2017. [Available on GitHub: https://github.com/CMU-SAFARI/SoftMC]

Summary

We provide *the first* RowHammer characterization **under reduced wordline voltage**

Experimental results with *272 real DRAM chips* show that **reducing wordline voltage:**

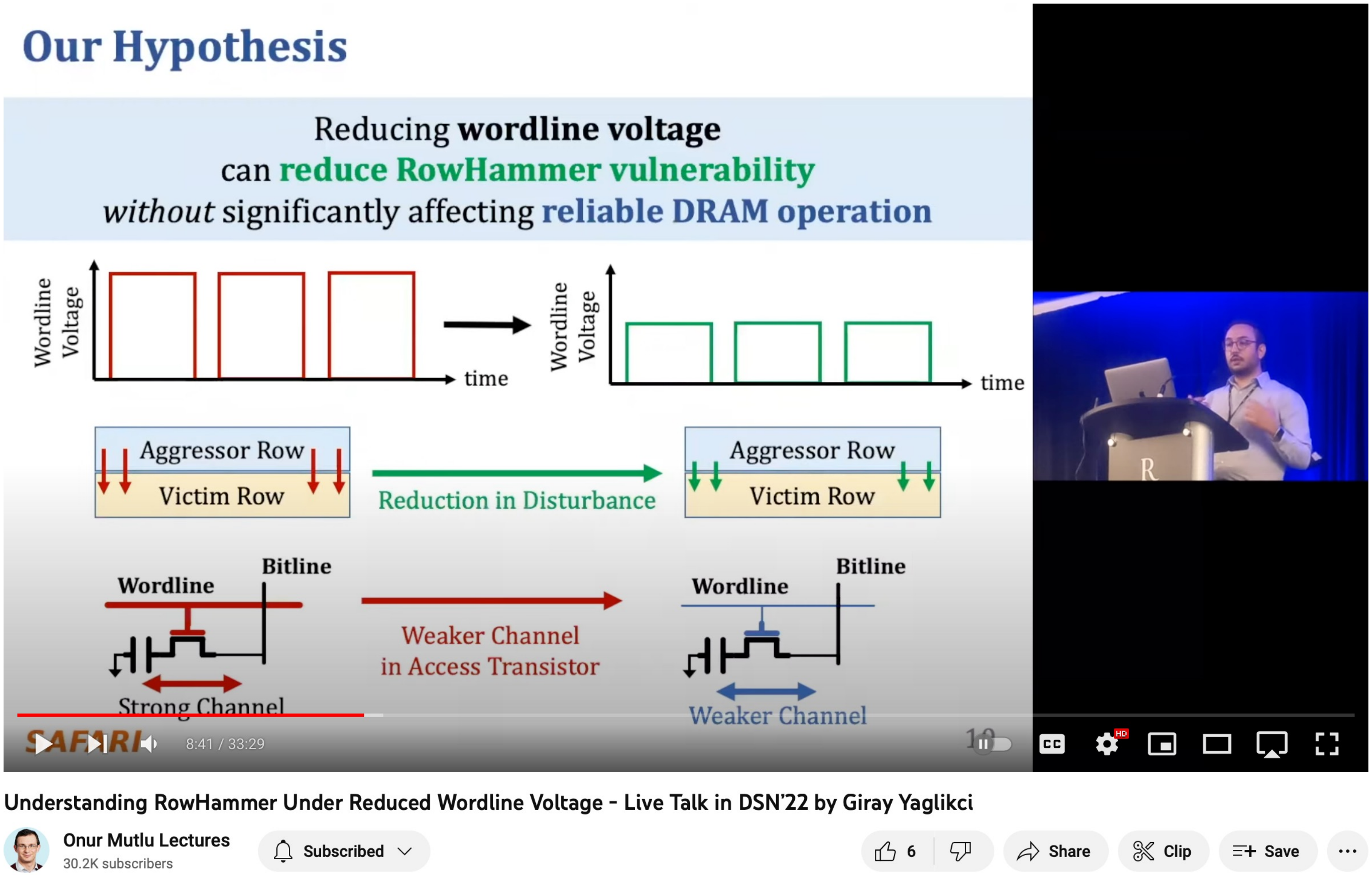
1. **Reduces RowHammer vulnerability**
   - **Bit error rate** caused by a RowHammer attack reduces by **15.2% (66.9% max)**
   - A row needs to be activated **7.4% more times (85.8% max)** to induce *the first* bit flip

2. **Increases row activation latency**
   - More than **76%** of the tested DRAM chips **reliably operate** using **nominal** timing parameters
   - Remaining **24%** **reliably operate** with **increased** (up to 24ns) row activation latency

3. **Reduces data retention time**
   - **80%** of the tested DRAM chips **reliably operate using nominal refresh rate**
   - Remaining **20%** **reliably operate** by
     - Using **single error correcting codes**
     - **Doubling the refresh rate** for **a small fraction (16.4%) of DRAM rows**

Reducing wordline voltage can **reduce RowHammer vulnerability**
without significantly affecting **reliable DRAM operation**

RowHammer vs. Wordline Voltage: Talk Video



https://youtube.com/watch?v=fkM32oA0u6U&si=EnSIkaIECMiOmarE

RowHammer in HBM Chips (2023)

- Ataberk Olgun, Majd Osserian, A. Giray Yağlıkçı, Yahya Can Tugrul, Haocong Luo, Steve Rhyner, Behzad Salami, Juan Gomez-Luna, and Onur Mutlu,
"An Experimental Analysis of RowHammer in HBM2 DRAM Chips"
*Proceedings of the 53nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Disrupt Track (**DSN Disrupt**)*, Porto, Portugal, June 2023.
[arXiv version]
[Slides (pptx) (pdf)]
[Talk Video (24 minutes, including Q&A)]

An Experimental Analysis of RowHammer in HBM2 DRAM Chips

Ataberk Olgun[1] Majd Osseiran[1,2] A. Giray Yağlıkçı[1] Yahya Can Tuğrul[1]
Haocong Luo[1] Steve Rhyner[1] Behzad Salami[1] Juan Gomez Luna[1] Onur Mutlu[1]
[1]*SAFARI Research Group, ETH Zürich* [2]*American University of Beirut*

https://arxiv.org/pdf/2305.17918.pdf

RowHammer in HBM Chips (2024)

- **Appears at DSN 2024**

Read Disturbance in High Bandwidth Memory: A Detailed Experimental Study on HBM2 DRAM Chips

Ataberk Olgun[1] Majd Osseiran[1] A. Giray Yağlıkçı[1] Yahya Can Tuğrul[1]
Haocong Luo[1] Steve Rhyner[1] Behzad Salami[2] Juan Gomez Luna[1] Onur Mutlu[1]
[1]*ETH Zürich* [2]*BSC*

https://arxiv.org/pdf/2310.14665

RowHammer Spatial Variation Analysis (2024)

- **Appears at HPCA 2024**

Spatial Variation-Aware Read Disturbance Defenses: Experimental Analysis of Real DRAM Chips and Implications on Future Solutions

Abdullah Giray Yağlıkçı   Yahya Can Tuğrul   Geraldo F. Oliveira
İsmail Emir Yüksel   Ataberk Olgun   Haocong Luo   Onur Mutlu
ETH Zürich

https://arxiv.org/pdf/2402.18652

New RowHammer Solutions

TRRespass

Industry-Adopted Solutions Do Not Work

- 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. IEEE Micro Top Pick Honorable Mention.
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*

*Vrije Universiteit Amsterdam §ETH Zürich ¶Qualcomm Technologies Inc.

A Poor RowHammer Solution

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

TRRespass

- First work to show that TRR-protected DRAM chips are vulnerable to RowHammer in the field
  - Mitigations advertised as secure are not secure
- Introduces the Many-sided RowHammer attack
  - Idea: Hammer many rows to bypass TRR mitigations (e.g., by overflowing proprietary TRR tables that detect aggressor rows)
- (Partially) reverse-engineers the TRR and pTRR mitigation mechanisms implemented in DRAM chips and memory controllers
- Provides an automatic tool that can effectively create many-sided RowHammer attacks in DDR4 and LPDDR4(X) chips

Example Many-Sided Hammering Patterns

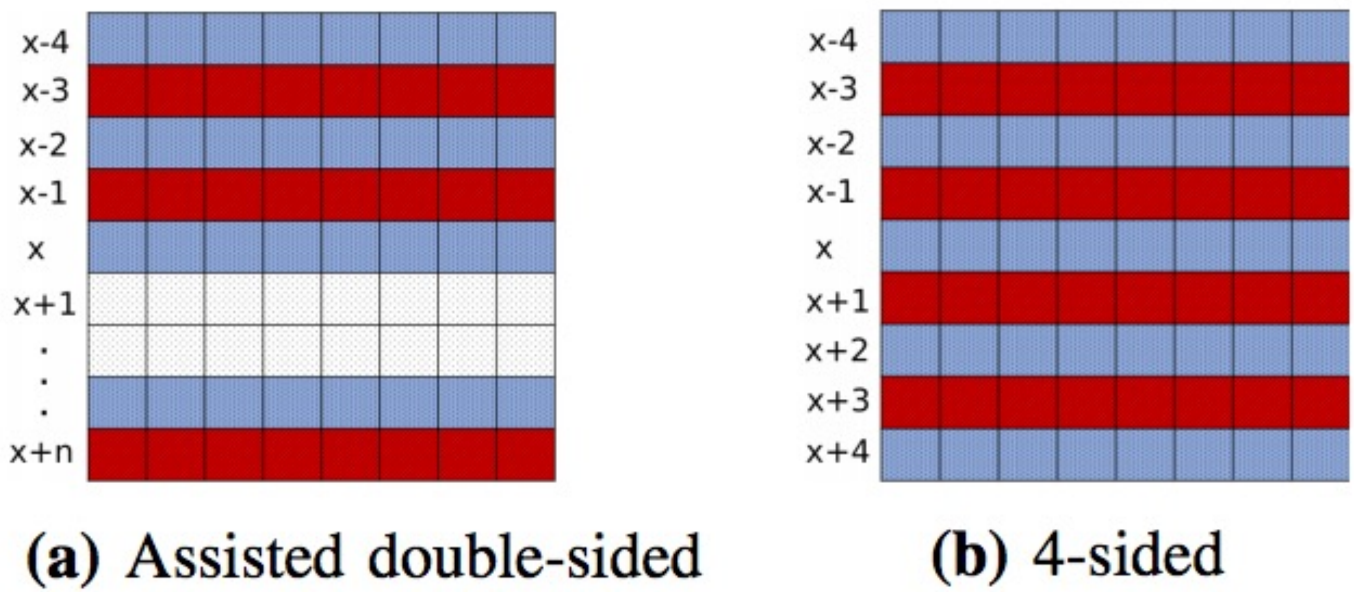


Fig. 12: Hammering patterns discovered by *TRRespass*. Aggressor rows are in red (■) and victim rows are in blue (■).

BitFlips vs. Number of Aggressor Rows

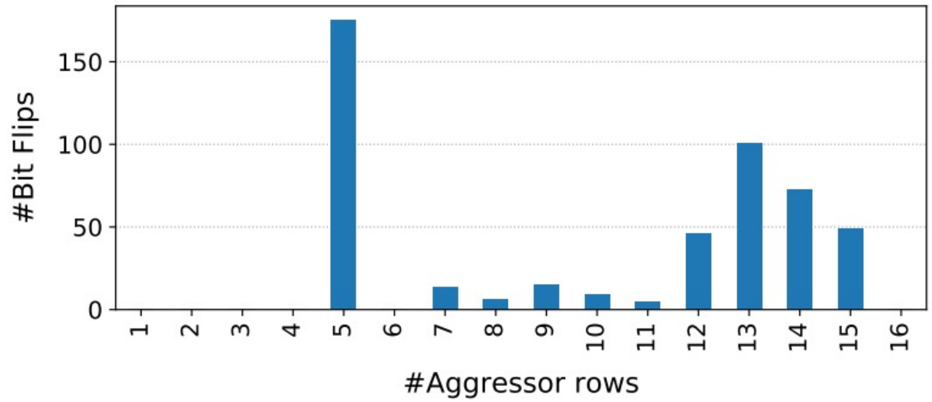


Fig. 10: Bit flips vs. number of aggressor rows. Module $\mathcal{C}_{12}$: Number of bit flips in bank 0 as we vary the number of aggressor rows. Using SoftMC, we refresh DRAM with standard `tREFI` and run the tests until each aggressor rows is hammered 500K times.

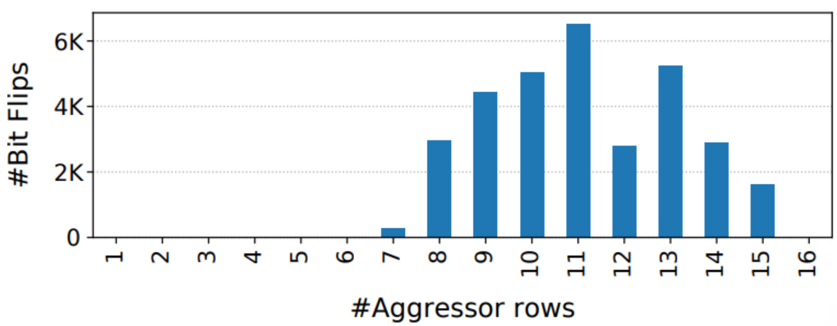


Fig. 11: Bit flips vs. number of aggressor rows. Module $\mathcal{A}_{15}$: Number of bit flips in bank 0 as we vary the number of aggressor rows. Using SoftMC, we refresh DRAM with standard `tREFI` and run the tests until each aggressor rows is hammered 500K times.

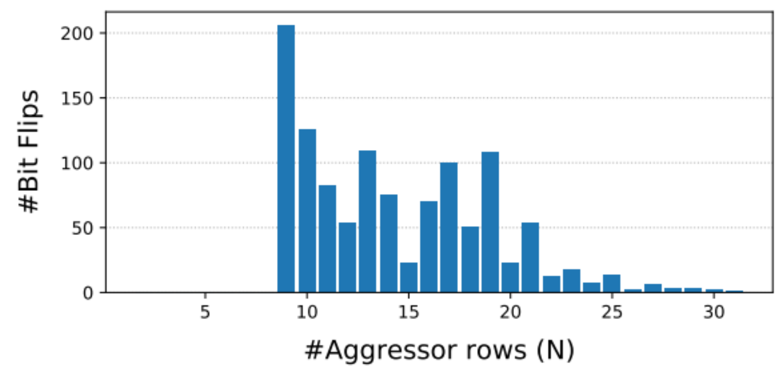


Fig. 13: Bit flips vs. number of aggressor rows. Module $\mathcal{A}_{10}$: Number of bit flips triggered with *N-sided* RowHammer for varying number of *N* on Intel Core i7-7700K. Each aggressor row is one row away from the closest aggressor row (i.e., `VAVAVA...` configuration) and aggressor rows are hammered in a round-robin fashion.

TRRespass Vulnerable DRAM Modules

TABLE II: *TRRespass* results. We report the number of patterns found and bit flips detected for the 42 DRAM modules in our set.

Module	*Date (yy-ww)*	*Freq. (MHz)*	*Size (GB)*	*Organization*			*MAC*	*Found Patterns*	*Best Pattern*	*Corruptions*			*Double Refresh*
				Ranks	*Banks*	*Pins*				*Total*	$1 \to 0$	$0 \to 1$	
$\mathcal{A}_{0,1,2,3}$	16-37	2132	4	1	16	×8	UL	—	—	—	—	—	—
$\mathcal{A}_4$	16-51	2132	4	1	16	×8	UL	4	9-sided	7956	4008	3948	—
$\mathcal{A}_5$	18-51	2400	4	1	8	×16	UL	—	—	—	—	—	—
$\mathcal{A}_{6,7}$	18-15	2666	4	1	8	×16	UL	—	—	—	—	—	—
$\mathcal{A}_8$	17-09	2400	8	1	16	×8	UL	33	19-sided	20808	10289	10519	—
$\mathcal{A}_9$	17-31	2400	8	1	16	×8	UL	33	19-sided	24854	12580	12274	—
$\mathcal{A}_{10}$	19-02	2400	16	2	16	×8	UL	488	10-sided	11342	1809	11533	✓
$\mathcal{A}_{11}$	19-02	2400	16	2	16	×8	UL	523	10-sided	12830	1682	11148	✓
$\mathcal{A}_{12,13}$	18-50	2666	8	1	16	×8	UL	—	—	—	—	—	—
$\mathcal{A}_{14}$	19-08†	3200	16	2	16	×8	UL	120	14-sided	32723	16490	16233	—
$\mathcal{A}_{15}$‡	17-08	2132	4	1	16	×8	UL	2	9-sided	22397	12351	10046	—
$\mathcal{B}_0$	18-11	2666	16	2	16	×8	UL	2	3-sided	17	10	7	—
$\mathcal{B}_1$	18-11	2666	16	2	16	×8	UL	2	3-sided	22	16	6	—
$\mathcal{B}_2$	18-49	3000	16	2	16	×8	UL	2	3-sided	5	2	3	—
$\mathcal{B}_3$	19-08†	3000	8	1	16	×8	UL	—	—	—	—	—	—
$\mathcal{B}_{4,5}$	19-08†	2666	8	2	16	×8	UL	—	—	—	—	—	—
$\mathcal{B}_{6,7}$	19-08†	2400	4	1	16	×8	UL	—	—	—	—	—	—
$\mathcal{B}_8$◇	19-08†	2400	8	1	16	×8	UL	—	—	—	—	—	—
$\mathcal{B}_9$◇	19-08†	2400	8	1	16	×8	UL	2	3-sided	12	—	12	✓
$\mathcal{B}_{10,11}$	16-13†	2132	8	2	16	×8	UL	—	—	—	—	—	—
$\mathcal{C}_{0,1}$	18-46	2666	16	2	16	×8	UL	—	—	—	—	—	—
$\mathcal{C}_{2,3}$	19-08†	2800	4	1	16	×8	UL	—	—	—	—	—	—
$\mathcal{C}_{4,5}$	19-08†	3000	8	1	16	×8	UL	—	—	—	—	—	—
$\mathcal{C}_{6,7}$	19-08†	3000	16	2	16	×8	UL	—	—	—	—	—	—
$\mathcal{C}_8$	19-08†	3200	16	2	16	×8	UL	—	—	—	—	—	—
$\mathcal{C}_9$	18-47	2666	16	2	16	×8	UL	—	—	—	—	—	—
$\mathcal{C}_{10,11}$	19-04	2933	8	1	16	×8	UL	—	—	—	—	—	—
$\mathcal{C}_{12}$‡	15-01†	2132	4	1	16	×8	UT	25	10-sided	190037	63904	126133	✓
$\mathcal{C}_{13}$‡	18-49	2132	4	1	16	×8	UT	3	9-sided	694	239	455	—

† The module does not report manufacturing date. Therefore, we report purchase date as an approximation.
‡ Analyzed using the FPGA-based SoftMC.
◇ The system runs with double refresh frequency in standard conditions. We configured the refresh interval to be 64 ms in the BIOS settings.

UL = Unlimited
UT = Untested

TRRespass Vulnerable Mobile Phones

TABLE III: LPDDR4(X) results. Mobile phones tested against *TRRespass* on ARMv8 sorted by production date. We found bit flip inducing RowHammer patterns on 5 out of 13mobile phones.

Mobile Phone	*Year*	*SoC*	*Memory (GB)*	*Found Patterns*
Google Pixel	2016	MSM8996	$4^{\dagger}$	✓
Google Pixel 2	2017	MSM8998	4	—
Samsung G960F/DS	2018	Exynos 9810	4	—
Huawei P20 DS	2018	Kirin 970	4	—
Sony XZ3	2018	SDM845	4	—
HTC U12+	2018	SDM845	6	—
LG G7 ThinQ	2018	SDM845	$4^{\dagger}$	✓
Google Pixel 3	2018	SDM845	4	✓
Google Pixel 4	2019	SM8150	6	—
OnePlus 7	2019	SM8150	8	✓
Samsung G970F/DS	2019	Exynos 9820	6	✓
Huawei P30 DS	2019	Kirin 980	6	—
Xiaomi Redmi Note 8 Pro	2019	Helio G90T	6	—

† LPDDR4 (not LPDDR4X)

TRRespass Based RowHammer Attack

TABLE IV: Time to exploit. Time to find the first exploitable template on two sample modules from each DRAM vendor.

Module	τ *(ms)*	*PTE* [81]	*RSA-2048* [79]	*sudo* [27]
$\mathcal{A}_{14}$	188.7	4.9s	6m 27s	—
$\mathcal{A}_{4}$	180.8	38.8s	39m 28s	—
$\mathcal{B}_{1}$	360.7	—	—	—
$\mathcal{B}_{2}$	331.2	—	—	—
$\mathcal{C}_{12}$	300.0	2.3s	74.6s	54m16s
$\mathcal{C}_{13}$	180.9	3h 15m	—	—

τ: Time to template a single row: time to fill the victim and aggressor rows + hammer time + time to scan the row.

TRRespass Key Results

- 13 out of 42 tested DDR4 DRAM modules are vulnerable
  - From all 3 major manufacturers
  - 3-, 9-, 10-, 14-, 19-sided hammer attacks needed
- 5 out of 13 mobile phones tested vulnerable
  - From 4 major manufacturers
  - With LPDDR4(X) DRAM chips
- These results are scratching the surface
  - TRRespass tool is not exhaustive
  - There is a lot of room for uncovering more vulnerable chips and phones

TRRespass Key Takeaways

RowHammer is still
an open problem

Security by obscurity
is likely not a good solution

Improvements in JEDEC (2020-2021)

JEDEC®
Global Standards for the Microelectronics Industry

STANDARDS & DOCUMENTS | COMMITTEES | NEWS | EVENTS & MEETINGS

NEAR-TERM DRAM LEVEL ROWHAMMER MITIGATION

JEP300-1

Published: Mar 2021

RAM process node transistor scaling for power and DRAM capacity has made DRAM cells more sensitive to disturbances or transient faults. This sensitivity becomes much worse if external stresses are applied in a meticulously manipulated sequence, such as Rowhammer. Rowhammer related papers have been written outside of JEDEC, but some assumptions used in those papers didn't explain the problem very clearly or correctly, so the perception for this matter is not precisely understood within the industry. This publication defines the problem and recommends following mitigations to address such concerns across the DRAM industry or academia. Item 1866.01.

Committee(s): JC-42

https://www.jedec.org/standards-documents/docs/jep300-1

JEDEC®
Global Standards for the Microelectronics Industry

STANDARDS & DOCUMENTS | COMMITTEES | NEWS | EVENTS & MEETINGS

SYSTEM LEVEL ROWHAMMER MITIGATION

JEP301-1

Published: Mar 2021

A DRAM rowhammer security exploit is a serious threat to cloud service providers, data centers, laptops, smart phones, self-driving cars and IoT devices. Hardware research and development will take time. DRAM components, DRAM DIMMs, System-on-chip (SoC), chipsets and system products have their own design cycle time and overall life time. This publication recommends best practices to mitigate the security risks from rowhammer attacks. Item 1866.02.

Committee(s): JC-42

https://www.jedec.org/standards-documents/docs/jep301-1

Improvements in JEDEC (2024)

JEDEC®

Global Standards for the Microelectronics Industry

STANDARDS & DOCUMENTS | COMMITTEES | NEWS | EVENTS & MEETINGS | JOI

DDR5 SDRAM JESD79-5C Apr 2024

Release Number: Version 1.30

Version 1.30

This standard defines the DDR5 SDRAM specification, including features, functionalities, AC and DC characteristics, packages, and ball/signal assignments. The purpose of this Standard is to define the minimum set of requirements for JEDEC compliant 8 Gb through 32 Gb for x4, x8, and x16 DDR5 SDRAM devices. This standard was created based on the DDR4 standards (JESD79-4) and some aspects of the DDR, DDR2, DDR3, and LPDDR4 standards (JESD79, JESD79-2, JESD79-3, and JESD209-4).

Committee(s): JC-42, JC-42.3

Uncovering TRR Almost Completely

Industry-Adopted Solutions Are Very Poor

- Hasan Hassan, Yahya Can Tugrul, Jeremie S. Kim, Victor van der Veen, Kaveh Razavi, and Onur Mutlu,
  "Uncovering In-DRAM RowHammer Protection Mechanisms: A New Methodology, Custom RowHammer Patterns, and Implications"
  *Proceedings of the 54th International Symposium on Microarchitecture (**MICRO**)*, Virtual, October 2021.
  [Slides (pptx) (pdf)]
  [Short Talk Slides (pptx) (pdf)]
  [Lightning Talk Slides (pptx) (pdf)]
  [Talk Video (25 minutes)]
  [Lightning Talk Video (100 seconds)]
  [arXiv version]

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

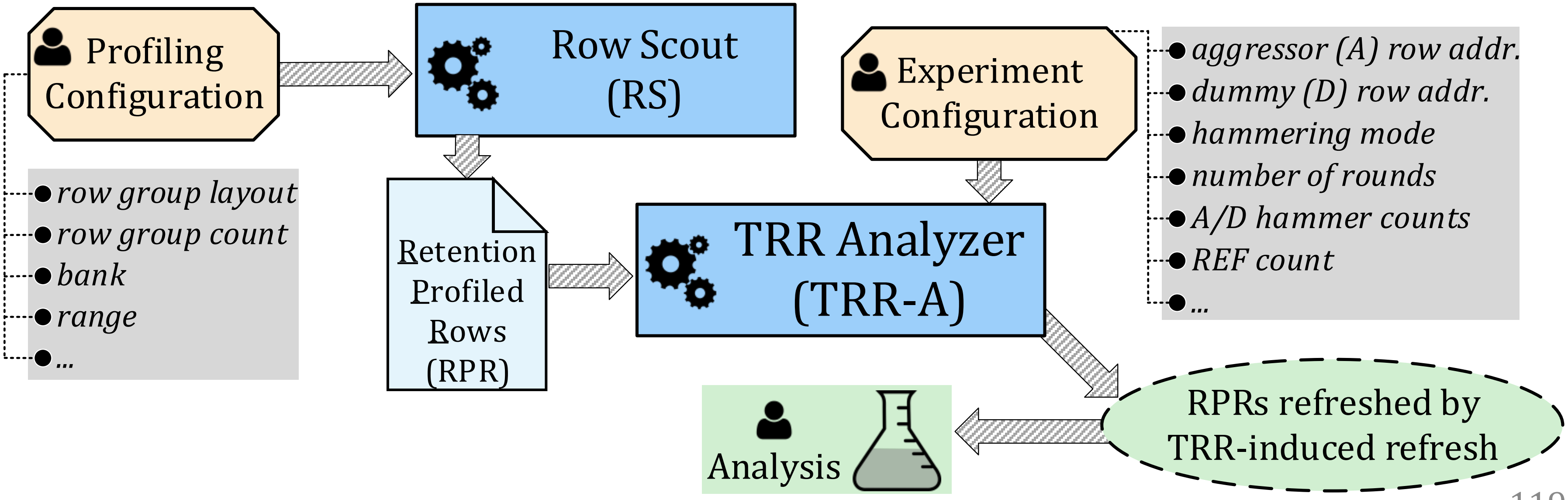
Hasan Hassan[†] Yahya Can Tuğrul[†‡] Jeremie S. Kim[†] Victor van der Veen[σ]
Kaveh Razavi[†] Onur Mutlu[†]
[†]ETH Zürich *[‡]TOBB University of Economics & Technology* *[σ]Qualcomm Technologies Inc.*

https://github.com/CMU-SAFARI/U-TRR

Overview of U-TRR

U-TRR: A new methodology to *uncover* the inner workings of TRR

Key idea: Use data retention failures as a side channel to detect when a row is refreshed by TRR



https://github.com/CMU-SAFARI/U-TRR

Key Takeaways

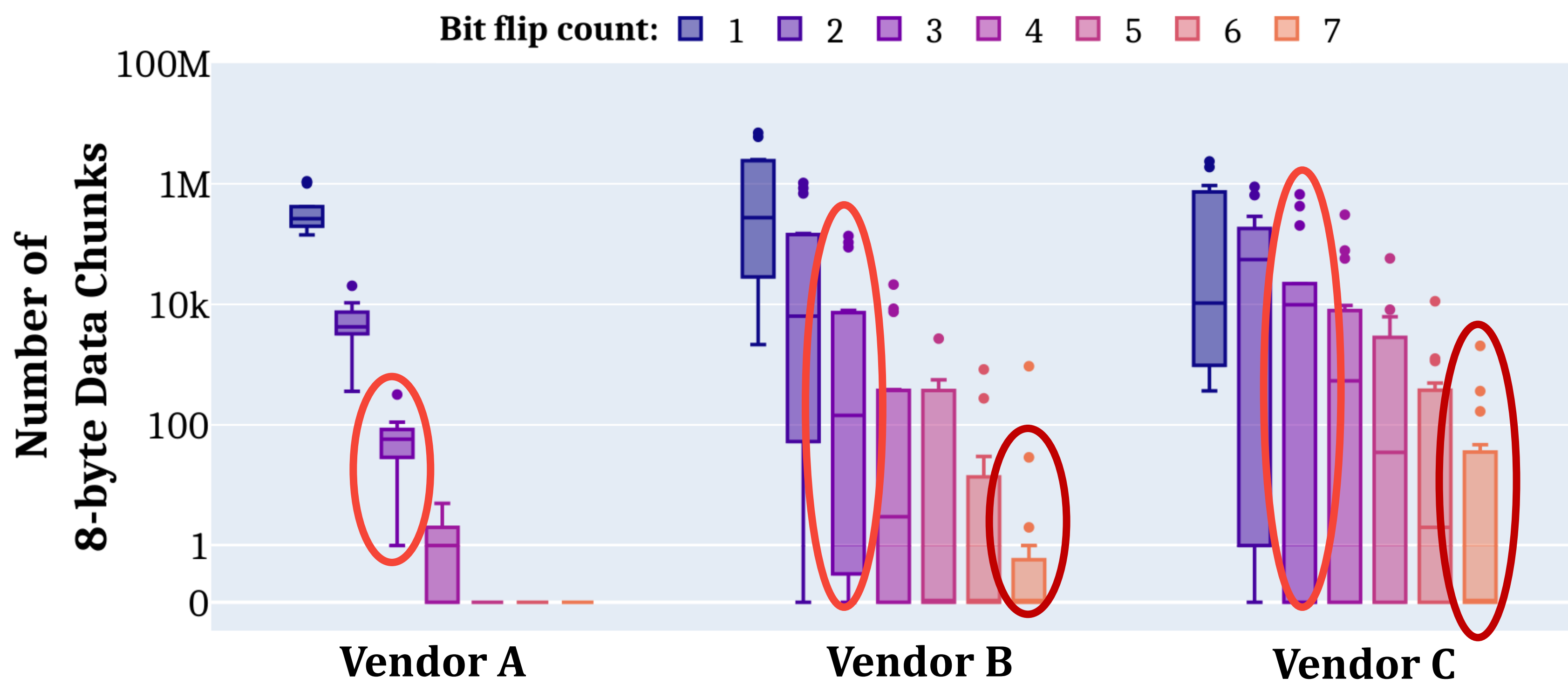
All 45 modules we test are vulnerable

99.9% of rows in a DRAM bank
experience **at least one RowHammer bit flip**

ECC is ineffective: up to 7 RowHammer bit flips
in an 8-byte dataword

Module	Date (yy-ww)	Chip Density (Gbit)	Organization			HC_{first}†	Our Key TRR Observations and Results							
			Ranks	Banks	Pins		Version	Aggressor Detection	Aggressor Capacity	Per-Bank TRR	TRR-to-REF Ratio	Neighbors Refreshed	% Vulnerable DRAM Rows†	Max. Bit Flips per Row per Hammer†
A0	19-50	8	1	16	8	16K	A_{TRR1}	Counter-based	16	✓	1/9	4	73.3%	1.16
A1-5	19-36	8	1	8	16	13K-15K	A_{TRR1}	Counter-based	16	✓	1/9	4	99.2% - 99.4%	2.32 - 4.73
A6-7	19-45	8	1	8	16	13K-15K	A_{TRR1}	Counter-based	16	✓	1/9	4	99.3% - 99.4%	2.12 - 3.86
A8-9	20-07	8	1	16	8	12K-14K	A_{TRR1}	Counter-based	16	✓	1/9	4	74.6% - 75.0%	1.96 - 2.96
A10-12	19-51	8	1	16	8	12K-13K	A_{TRR1}	Counter-based	16	✓	1/9	4	74.6% - 75.0%	1.48 - 2.86
A13-14	20-31	8	1	8	16	11K-14K	A_{TRR2}	Counter-based	16	✓	1/9	2	94.3% - 98.6%	1.53 - 2.78
B0	18-22	4	1	16	8	44K	B_{TRR1}	Sampling-based	1	✗	1/4	2	99.9%	2.13
B1-4	20-17	4	1	16	8	159K-192K	B_{TRR1}	Sampling-based	1	✗	1/4	2	23.3% - 51.2%	0.06 - 0.11
B5-6	16-48	4	1	16	8	44K-50K	B_{TRR1}	Sampling-based	1	✗	1/4	2	99.9%	1.85 - 2.03
B7	19-06	8	2	16	8	20K	B_{TRR1}	Sampling-based	1	✗	1/4	2	99.9%	31.14
B8	18-03	4	1	16	8	43K	B_{TRR1}	Sampling-based	1	✗	1/4	2	99.9%	2.57
B9-12	19-48	8	1	16	8	42K-65K	B_{TRR2}	Sampling-based	1	✗	1/9	2	36.3% - 38.9%	16.83 - 24.26
B13-14	20-08	4	1	16	8	11K-14K	B_{TRR3}	Sampling-based	1	✓	1/2	4	99.9%	16.20 - 18.12
C0-3	16-48	4	1	16	x8	137K-194K	C_{TRR1}	Mix	Unknown	✓	1/17	2	1.0% - 23.2%	0.05 - 0.15
C4-6	17-12	8	1	16	x8	130K-150K	C_{TRR1}	Mix	Unknown	✓	1/17	2	7.8% - 12.0%	0.06 - 0.08
C7-8	20-31	8	1	8	x16	40K-44K	C_{TRR1}	Mix	Unknown	✓	1/17	2	39.8% - 41.8%	9.66 - 14.56
C9-11	20-31	8	1	8	x16	42K-53K	C_{TRR2}	Mix	Unknown	✓	1/9	2	99.7%	9.30 - 32.04
C12-14	20-46	16	1	8	x16	6K-7K	C_{TRR3}	Mix	Unknown	✓	1/8	2	99.9%	4.91 - 12.64

Bypassing ECC with New RowHammer Patterns



Modules from all three vendors have many **8-byte data chunks** with 3 and more (up to 7) RowHammer bit flips

Conventional DRAM ECC cannot protect against our new RowHammer access patterns

Google's Half-Double RowHammer Attack (May 2021)

Google Security Blog

The latest news and insights from Google on security and safety on the Internet

Introducing Half-Double: New hammering technique for DRAM Rowhammer bug

May 25, 2021

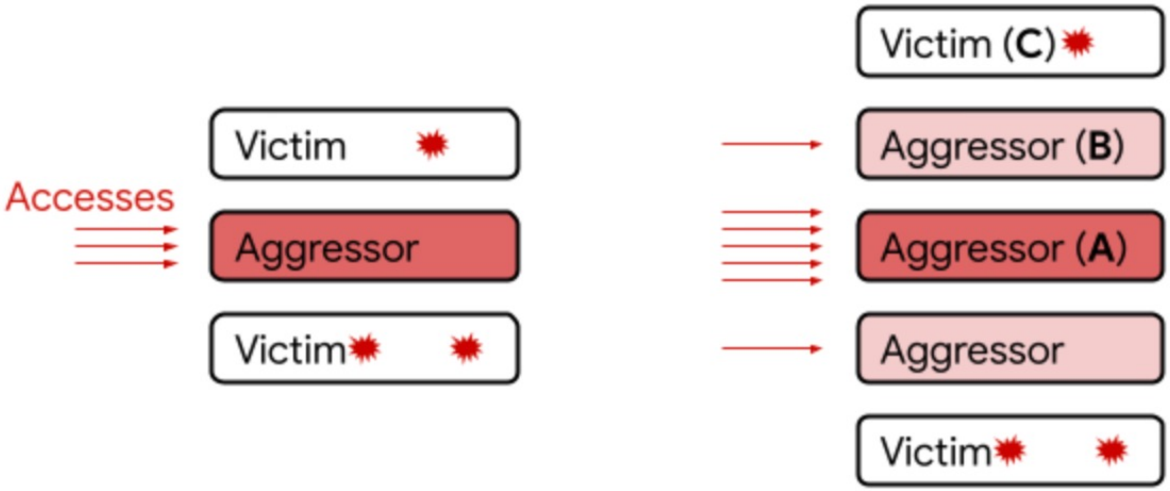
Research Team: Salman Qazi, Yoongu Kim, Nicolas Boichat, Eric Shiu & Mattias Nissler

Today, we are sharing details around our discovery of Half-Double, a new Rowhammer technique that capitalizes on the worsening physics of some of the newer DRAM chips to alter the contents of memory.

Rowhammer is a DRAM vulnerability whereby repeated accesses to one address can tamper with the data stored at other addresses. Much like speculative execution vulnerabilities in CPUs, Rowhammer is a breach of the security guarantees made by the underlying hardware. As an electrical coupling phenomenon within the silicon itself, Rowhammer allows the potential bypass of hardware and software memory protection policies. This can allow untrusted code to break out of its sandbox and take full control of the system.

https://security.googleblog.com/2021/05/introducing-half-double-new-hammering.html

Google's Half-Double RowHammer Attack (May 2021)



(a) Classic Rowhammer (b) Half-Double

- Given three consecutive rows A, B, and C, we were able to attack C by directing a very large number of accesses to A, along with just a handful (~dozens) to B.
- Based on our experiments, accesses to B have a non-linear gating effect, in which they appear to "transport" the Rowhammer effect of A onto C.
- This is likely an indication that the electrical coupling responsible for **Rowhammer** is a property of distance, **effectively becoming stronger** and longer-ranged as cell geometries shrink down.

Google's Half-Double RowHammer Attack

- **Appears at USENIX Security 2022**

Half-Double: Hammering From the Next Row Over

Andreas Kogler[1] Jonas Juffinger[1,2] Salman Qazi[3] Yoongu Kim[3] Moritz Lipp[4*]
Nicolas Boichat[3] Eric Shiu[5] Mattias Nissler[3] Daniel Gruss[1]

[1]*Graz University of Technology* [2]*Lamarr Security Research* [3]*Google*
[4]*Amazon Web Services* [5]*Rivos*

https://www.usenix.org/system/files/sec22-kogler-half-double.pdf

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)]
  [Intel Hardware Security Academic Awards Short Talk Slides (pptx) (pdf)]
  [Talk Video (22 minutes)]
  [Short Talk Video (7 minutes)]
  [Intel Hardware Security Academic Awards Short Talk Video (2 minutes)]
  [BlockHammer Source Code]
  Intel Hardware Security Academic Award Finalist (one of 4 finalists out of 34 nominations)

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

A. Giray Yağlıkçı[1] Minesh Patel[1] Jeremie S. Kim[1] Roknoddin Azizi[1] Ataberk Olgun[1] Lois Orosa[1]
Hasan Hassan[1] Jisung Park[1] Konstantinos Kanellopoulos[1] Taha Shahroodi[1] Saugata Ghose[2] Onur Mutlu[1]
[1]*ETH Zürich* [2]*University of Illinois at Urbana–Champaign*

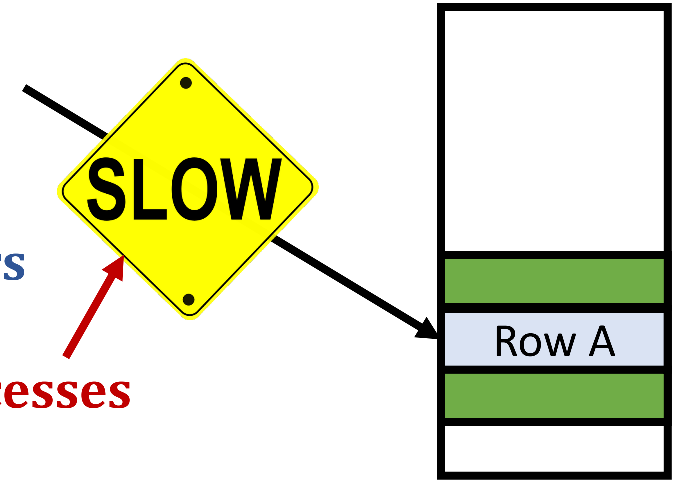
Two Key Challenges

1 **Scalability**
with worsening RowHammer vulnerability

2 **Compatibility**
with commodity DRAM chips

BlockHammer: Practical Throttling-based Mechanism

- A RowHammer attack hammers Row A
- BlockHammer detects a RowHammer attack using **area-efficient Bloom filters**
- BlockHammer **selectively throttles accesses** from within **the memory controller**
- Bit flips **do not** occur
- BlockHammer can *optionally* **inform the system software** about the attack



Physical Row Layout

BlockHammer is compatible with commodity DRAM chips
No need for proprietary info of or modifications to DRAM chips

A Takeaway

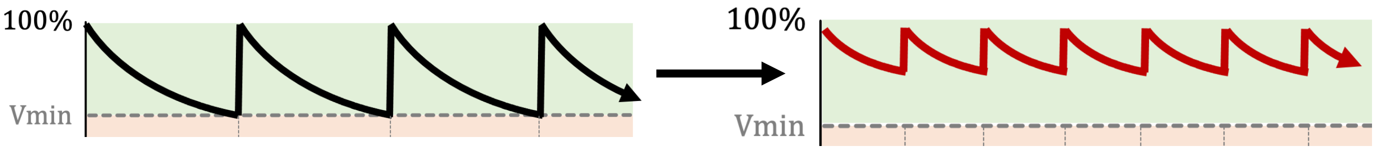
Main Memory Needs

Intelligent Controllers

for Security, Safety, Reliability, Scaling

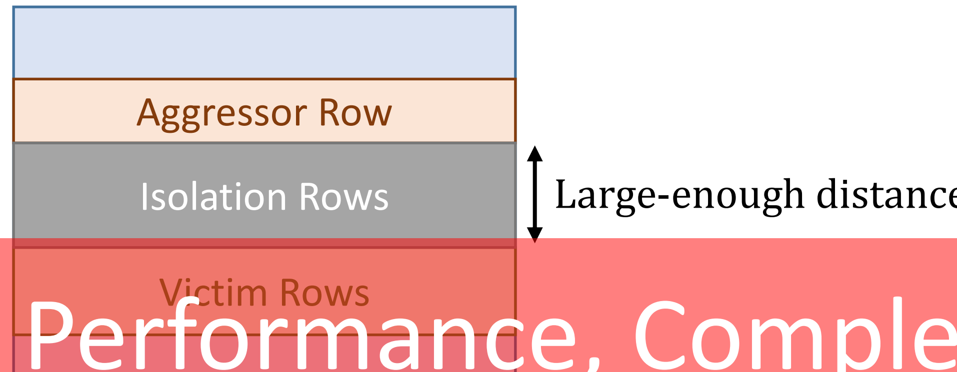
RowHammer Solution Approaches

- More robust DRAM chips **and/or** error-correcting codes
- Increased refresh rate



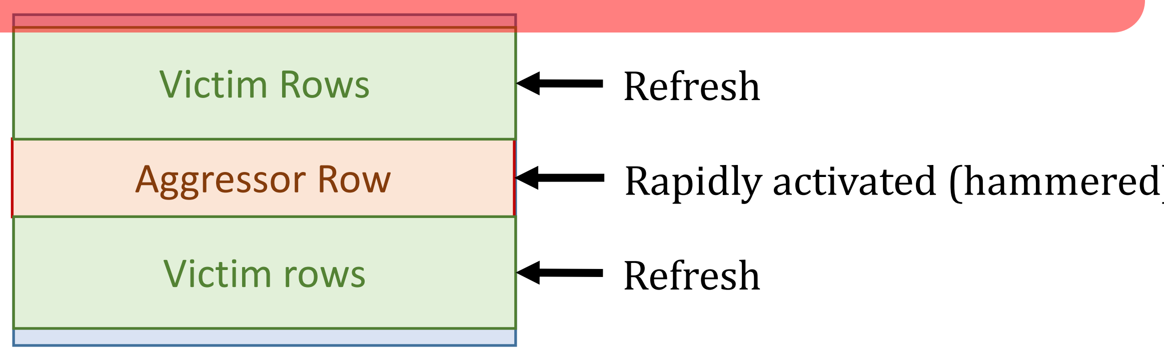
Fewer activations possible in a refresh interval

- Physical isolation

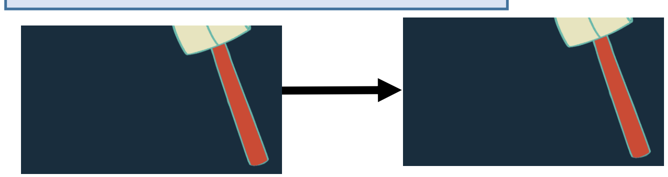


Cost, Power, Performance, Complexity

- Reactive refresh



- Proactive throttling



Fewer activations allowed for aggressive applications

RowHammer in 2023: SK Hynix

ISSCC 2023 / SESSION 28 / HIGH-DENSITY MEMORIES /

28.8 A 1.1V 16Gb DDR5 DRAM with Probabilistic-Aggressor Tracking, Refresh-Management Functionality, Per-Row Hammer Tracking, a Multi-Step Precharge, and Core-Bias Modulation for Security and Reliability Enhancement

Woongrae Kim, Chulmoon Jung, Seongnyuh Yoo, Duckhwa Hong, Jeongjin Hwang, Jungmin Yoon, Ohyong Jung, Joonwoo Choi, Sanga Hyun, Mankeun Kang, Sangho Lee, Dohong Kim, Sanghyun Ku, Donhyun Choi, Nogeun Joo, Sangwoo Yoon, Junseok Noh, Byeongyong Go, Cheolhoe Kim, Sunil Hwang, Mihyun Hwang, Seol-Min Yi, Hyungmin Kim, Sanghyuk Heo, Yeonsu Jang, Kyoungchul Jang, Shinho Chu, Yoonna Oh, Kwidong Kim, Junghyun Kim, Soohwan Kim, Jeongtae Hwang, Sangil Park, Junphyo Lee, Inchul Jeong, Joohwan Cho, Jonghwan Kim

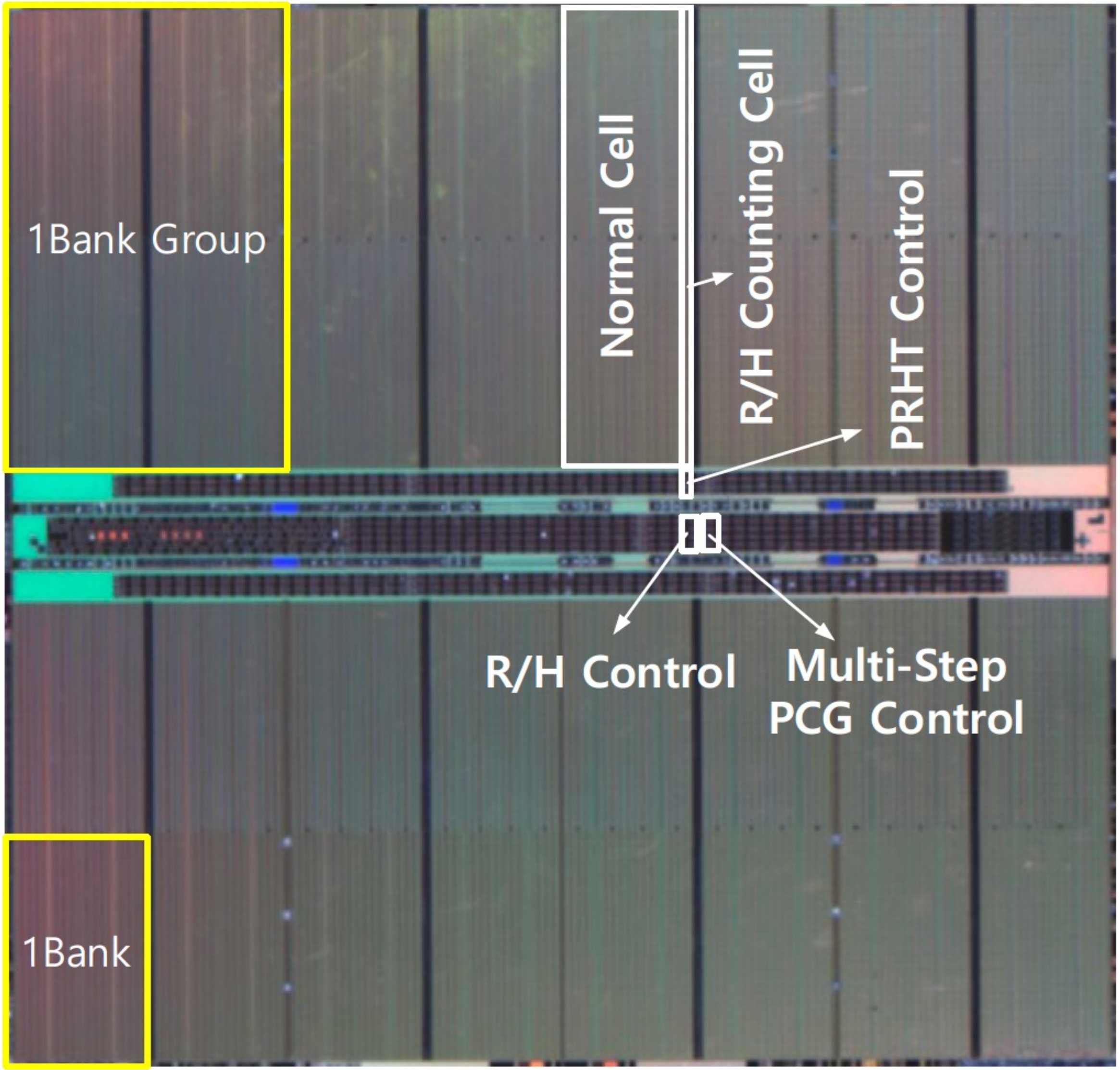
SK hynix Semiconductor, Icheon, Korea

Industry’s RowHammer Solutions (I)

SK hynix Semiconductor, Icheon, Korea

DRAM products have been recently adopted in a wide range of high-performance computing applications: such as in cloud computing, in big data systems, and IoT devices. This demand creates larger memory capacity requirements, thereby requiring aggressive DRAM technology node scaling to reduce the cost per bit [1,2]. However, DRAM manufacturers are facing technology scaling challenges due to row hammer and refresh retention time beyond 1a-nm [2]. Row hammer is a failure mechanism, where repeatedly activating a DRAM row disturbs data in adjacent rows. Scaling down severely threatens reliability since a reduction of DRAM cell size leads to a reduction in the intrinsic row hammer tolerance [2,3]. To improve row hammer tolerance, there is a need to probabilistically activate adjacent rows with carefully sampled active addresses and to improve intrinsic row hammer tolerance [2]. In this paper, row-hammer-protection and refresh-management schemes are presented to guarantee DRAM security and reliability despite the aggressive scaling from 1a-nm to sub 10-nm nodes. The probabilistic-aggressor-tracking scheme with a refresh-management function (RFM) and per-row hammer tracking (PRHT) improve DRAM resilience. A multi-step precharge reinforces intrinsic row-hammer tolerance and a core-bias modulation improves retention time: even in the face of cell-transistor degradation due to technology scaling. This comprehensive scheme leads to a reduced probability of failure, due to row hammer attacks, by 93.1% and an improvement in retention time by 17%.

Industry's RowHammer Solutions (II)



ISSCC 2023 / SESSION 28 / HIGH-DENSITY MEMORIES /

28.8 A 1.1V 16Gb DDR5 DRAM with Probabilistic-Aggressor Tracking, Refresh-Management Functionality, Per-Row Hammer Tracking, a Multi-Step Precharge, and Core-Bias Modulation for Security and Reliability Enhancement

Woongrae Kim, Chulmoon Jung, Seongnyuh Yoo, Duckhwa Hong, Jeongjin Hwang, Jungmin Yoon, Ohyong Jung, Joonwoo Choi, Sanga Hyun, Mankeun Kang, Sangho Lee, Dohong Kim, Sanghyun Ku, Donhyun Choi, Nogeun Joo, Sangwoo Yoon, Junseok Noh, Byeongyong Go, Cheolhoe Kim, Sunil Hwang, Mihyun Hwang, Seol-Min Yi, Hyungmin Kim, Sanghyuk Heo, Yeonsu Jang, Kyoungchul Jang, Shinho Chu, Yoonna Oh, Kwidong Kim, Junghyun Kim, Soohwan Kim, Jeongtae Hwang, Sangil Park, Junphyo Lee, Inchul Jeong, Joohwan Cho, Jonghwan Kim

SK hynix Semiconductor, Icheon, Korea

RowHammer in 2023: Samsung

DSAC: Low-Cost Rowhammer Mitigation Using In-DRAM Stochastic and Approximate Counting Algorithm

Seungki Hong Dongha Kim Jaehyung Lee Reum Oh
Changsik Yoo Sangjoon Hwang Jooyoung Lee

DRAM Design Team, Memory Division, Samsung Electronics

https://arxiv.org/pdf/2302.03591v1.pdf

A Solution from Microsoft

Panopticon: A Complete In-DRAM Rowhammer Mitigation

Tanj Bennett[§], Stefan Saroiu, Alec Wolman, and Lucian Cojocar
Microsoft, [§]Avant-Gray LLC

https://stefan.t8k2.com/publications/dramsec/2021/panopticon.pdf

Solutions in JEDEC (2024)

JEDEC®
Global Standards for the Microelectronics Industry

STANDARDS & DOCUMENTS | COMMITTEES | NEWS | EVENTS & MEETINGS | JOI

DDR5 SDRAM JESD79-5C Apr 2024

Release Number: Version 1.30

Version 1.30

This standard defines the DDR5 SDRAM specification, including features, functionalities, AC and DC characteristics, packages, and ball/signal assignments. The purpose of this Standard is to define the minimum set of requirements for JEDEC compliant 8 Gb through 32 Gb for x4, x8, and x16 DDR5 SDRAM devices. This standard was created based on the DDR4 standards (JESD79-4) and some aspects of the DDR, DDR2, DDR3, and LPDDR4 standards (JESD79, JESD79-2, JESD79-3, and JESD209-4).

Committee(s): JC-42, JC-42.3

Question

Are we now

RowHammer-free

in 2024 and Beyond?

Are We Now **RowHammer Free** in 2023?

- **Appeared at ISCA in June 2023**

RowPress: Amplifying Read-Disturbance in Modern DRAM Chips

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

https://arxiv.org/pdf/2306.17061.pdf

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

RowPress

RowPress [ISCA 2023]

- Haocong Luo, Ataberk Olgun, Giray Yaglikci, Yahya Can Tugrul, Steve Rhyner, M. Banu Cavlak, Joel Lindegger, Mohammad Sadrosadati, and Onur Mutlu,
"RowPress: Amplifying Read Disturbance in Modern DRAM Chips"
*Proceedings of the 50th International Symposium on Computer Architecture (**ISCA**)*, Orlando, FL, USA, June 2023.
[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Lightning Talk Video (3 minutes)]
[RowPress Source Code and Datasets (Officially Artifact Evaluated with All Badges)]
Officially artifact evaluated as available, reusable and reproducible.
Best artifact award at ISCA 2023. IEEE Micro Top Pick in 2024.

RowPress: Amplifying Read-Disturbance in Modern DRAM Chips

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

RowPress

Amplifying Read Disturbance in Modern DRAM Chips

ISCA 2023 Session 2B: Monday 19 June, 2:15 PM EDT

Haocong Luo

Ataberk Olgun *A. Giray Yağlıkçı* *Yahya Can Tuğrul*

Steve Rhyner *Meryem Banu Cavlak* *Joël Lindegger*

Mohammad Sadrosadati *Onur Mutlu*

SAFARI ETH Zürich

High-Level Summary

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

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

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

- We provide **effective solutions** to RowPress

What is RowPress?

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

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

Only one activation is enough in some cases!

Now, let's delve into some background and see how this is **different from RowHammer**

Are There Other Read-Disturb Issues in DRAM?

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

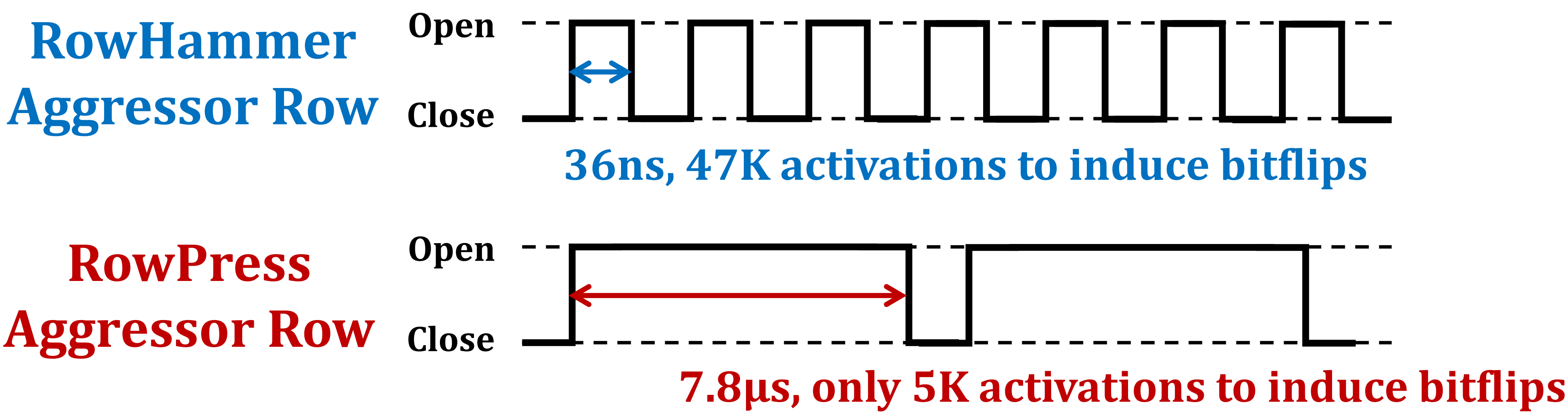
What if there is another read-disturb phenomenon that **does NOT rely on high row activation count**?

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

RowPress vs. RowHammer

Instead of using a high activation count,

☞ increase the time that the aggressor row stays open

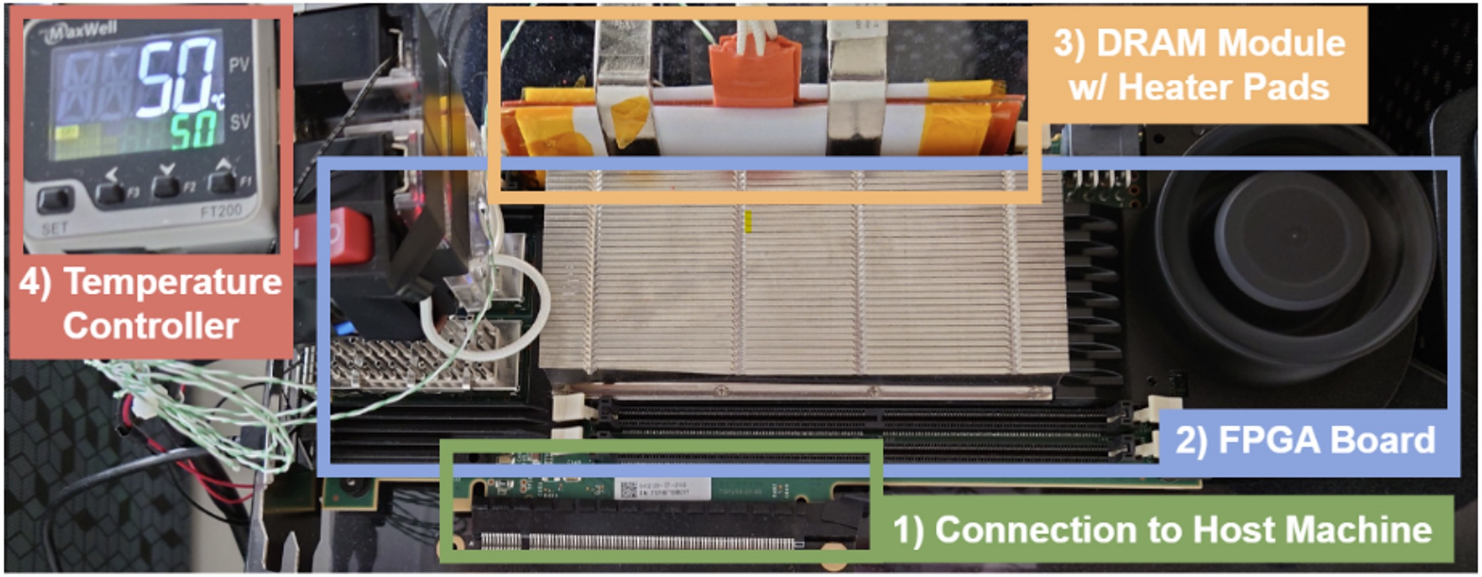


We observe bitflips even with **ONLY ONE activation** in extreme cases where the row stays open for 30ms

Real DRAM Chip Characterization (I)

FPGA-Based DDR4 Testing Infrastructure

- Based on SoftMC [Hassan+, HPCA'17] and DRAM Bender [Olgun+, TCAD'23]
- Fine-grained control over DRAM commands, timings, and temperature



Real DRAM Chip Characterization (II)

DRAM chips tested

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

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

Major Takeaways from Real DRAM Chips

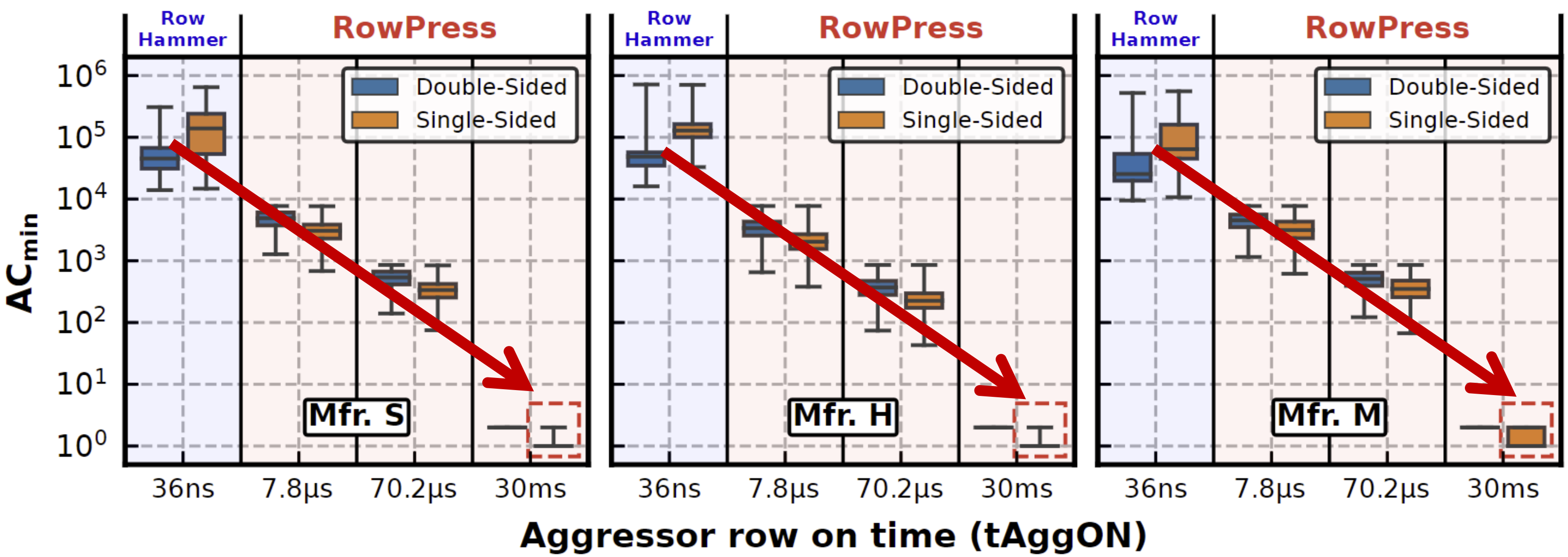
RowPress significantly **amplifies** DRAM's vulnerability to read disturbance

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

Key Characteristics of RowPress (I)

Amplifying Read Disturbance in DRAM

- Reduces the minimum number of row activations needed to induce a bitflip (**ACmin**) by **1-2 orders of magnitude**
- In extreme cases, activating a row **only once** induces bitflips



Key Characteristics of RowPress (II)

Amplifying Read Disturbance in DRAM

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

Different From RowHammer

- Affects a **different set of cells** compared to RowHammer and retention failures
- **Behaves differently** as access pattern and temperature changes compared to RowHammer

Real-System Demonstration (I)

Intel Core i5-10400
(Comet Lake)

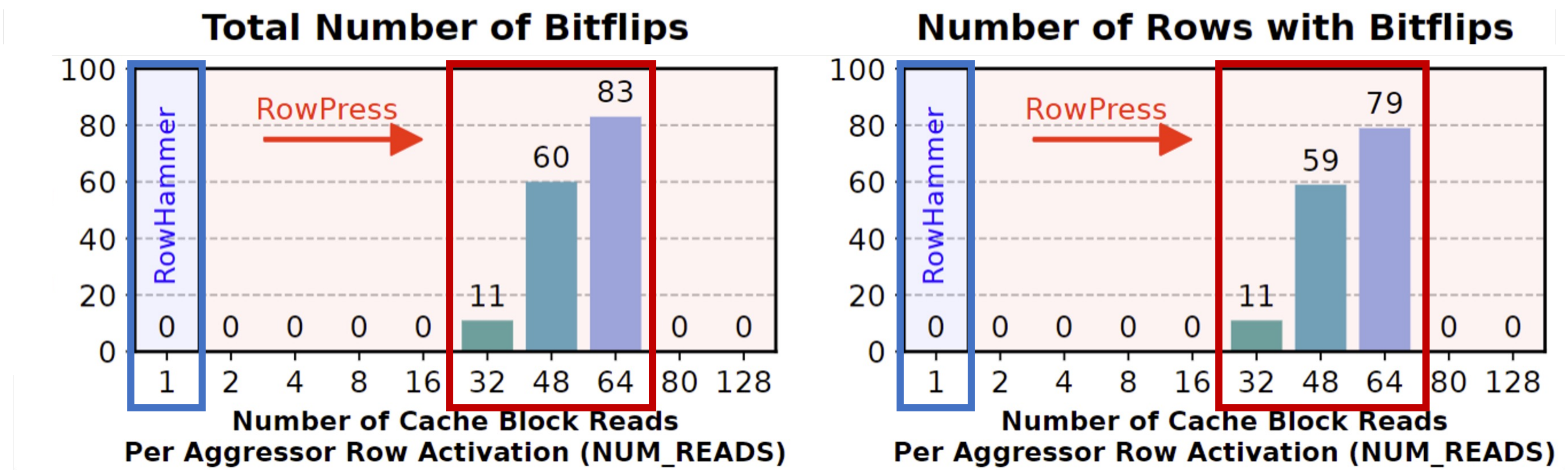
Samsung DDR4 Module
M378A2K43CB1-CTD
(Date Code: 20-10)
w/ TRR RowHammer Mitigation

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

```
// Sync with Refresh and Loop Below
  for (k = 0; k < NUM_AGGR_ACTS; k++)
    for (j = 0; j < NUM_READS; j++) *AGGRESSOR1[j];
    for (j = 0; j < NUM_READS; j++) *AGGRESSOR2[j];
    for (j = 0; j < NUM_READS; j++)
      clflushopt(AGGRESSOR1[j]);
      clflushopt(AGGRESSOR2[j]);
    mfence();
  activate_dummy_rows();
```

Number of Cache Blocks Accessed Per Aggressor Row ACT
(NUM_READS=1 is Rowhammer)

On 1500 victim rows



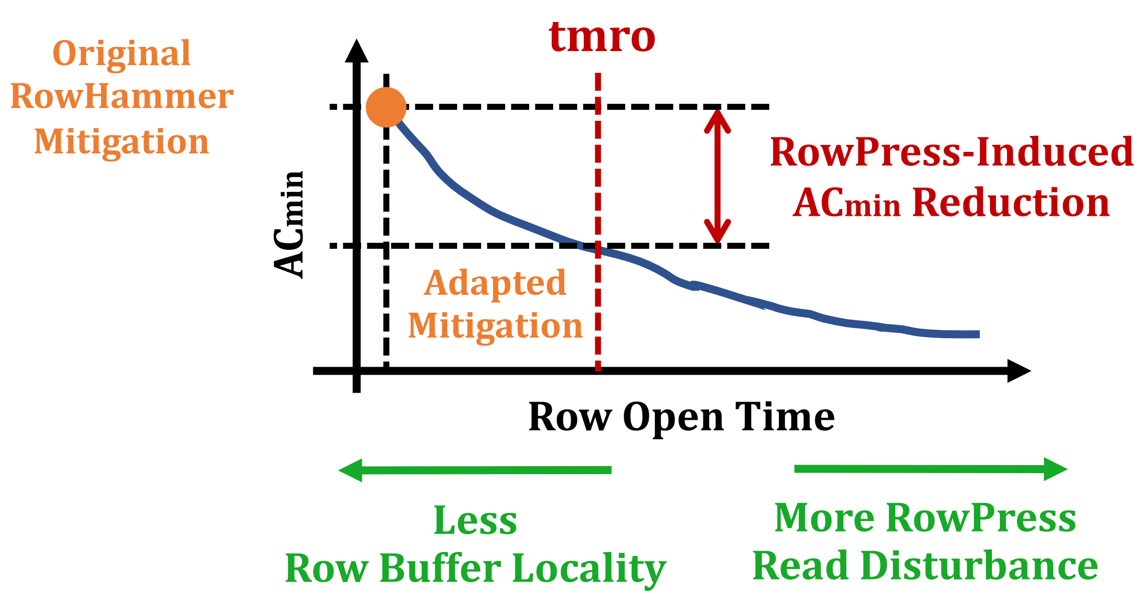
Leveraging RowPress, our user-level program induces bitflips when RowHammer cannot

Mitigating RowPress (I)

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

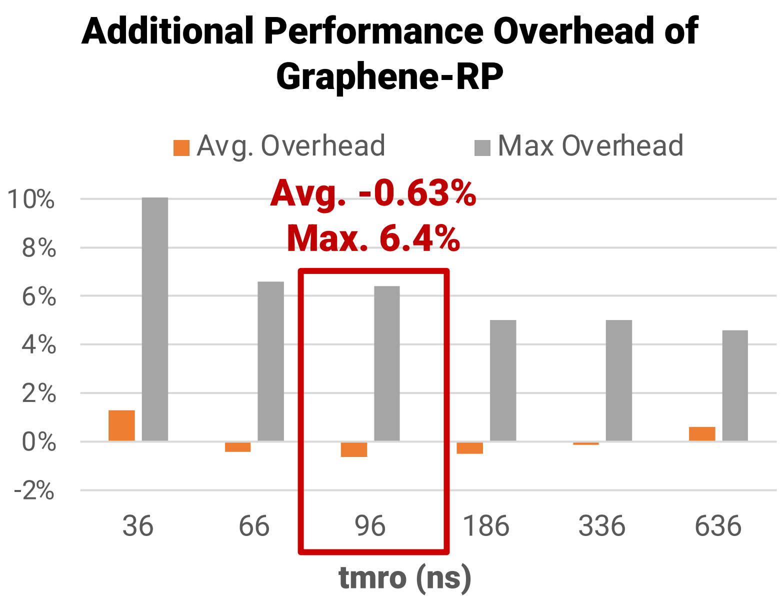
Key Idea:

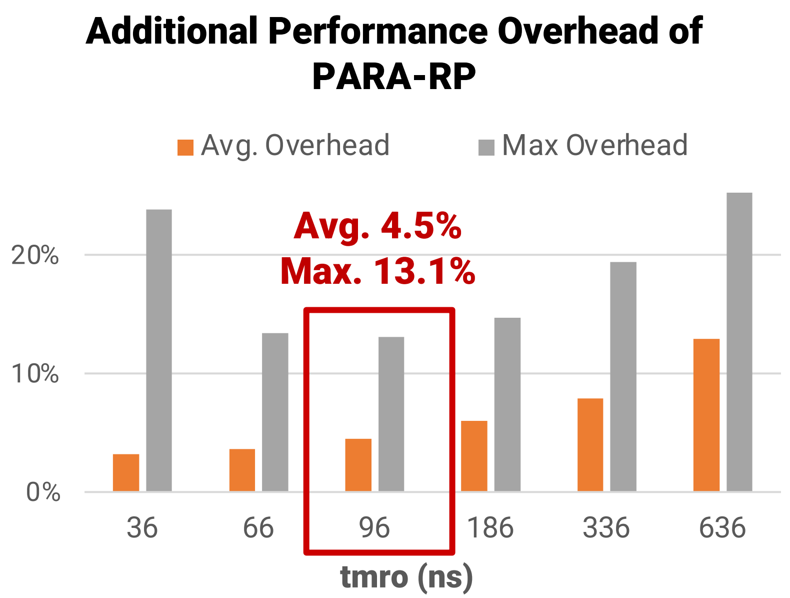
1. Limit the maximum row open time (**tmro**)
2. Configure the RowHammer mitigation to account for the **RowPress-induced reduction in AC_{min}**



Mitigating RowPress (II)

Key evaluation results





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

More Results & Source Code

Many more results & analyses in the paper

- 6 major takeaways
- 19 major empirical observations
- 3 more potential mitigations

Fully open source and artifact evaluated

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

RowPress [ISCA 2023]

- Haocong Luo, Ataberk Olgun, Giray Yaglikci, Yahya Can Tugrul, Steve Rhyner, M. Banu Cavlak, Joel Lindegger, Mohammad Sadrosadati, and Onur Mutlu,
"RowPress: Amplifying Read Disturbance in Modern DRAM Chips"
*Proceedings of the 50th International Symposium on Computer Architecture (**ISCA**)*, Orlando, FL, USA, June 2023.
[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Lightning Talk Video (3 minutes)]
[RowPress Source Code and Datasets (Officially Artifact Evaluated with All Badges)]
Officially artifact evaluated as available, reusable and reproducible.
Best artifact award at ISCA 2023. IEEE Micro Top Pick in 2024.

RowPress: Amplifying Read-Disturbance in Modern DRAM Chips

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

More to Come…

Two Major Directions

- **Understanding Bitflips (Hardware errors in general)**
  - Many effects on bitflips still need to be rigorously examined
    - Aging of DRAM Chips
    - Environmental Conditions (e.g., Process, Voltage, Temperature)
    - Memory Access Patterns
    - Memory Controller & System Design Decisions
    - …

- **Solving Bitflips (Hardware errors in general)**
  - Flexible and efficient solutions are necessary
    - In-field patchable / reconfigurable / programmable solutions
  - Co-architecting across the system stack/components is important
    - To avoid performance and denial-of-service problems

A RowHammer Survey: Recent Update

- Onur Mutlu, Ataberk Olgun, and A. Giray Yaglikci, **"Fundamentally Understanding and Solving RowHammer"** *Invited Special Session Paper at the 28th Asia and South Pacific Design Automation Conference (**ASP-DAC**)*, Tokyo, Japan, January 2023.
[arXiv version]
[Slides (pptx) (pdf)]
[Talk Video (26 minutes)]

Fundamentally Understanding and Solving RowHammer

Onur Mutlu
onur.mutlu@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

Ataberk Olgun
ataberk.olgun@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

A. Giray Yağlıkcı
giray.yaglikci@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

https://arxiv.org/pdf/2211.07613.pdf

Better Communication Between DRAM & Controller

A Case for Transparent Reliability in DRAM Systems

Minesh Patel[†] Taha Shahroodi[‡†] Aditya Manglik[†] A. Giray Yağlıkçı[†]
Ataberk Olgun[†] Haocong Luo[†] Onur Mutlu[†]

[†]*ETH Zürich* [‡]*TU Delft*

https://arxiv.org/pdf/2204.10378.pdf

Better Partitioning of DRAM & Controller

A Case for Self-Managing DRAM Chips: Improving Performance, Efficiency, Reliability, and Security via Autonomous in-DRAM Maintenance Operations

Hasan Hassan Ataberk Olgun A. Giray Yağlıkçı
Haocong Luo Onur Mutlu

ETH Zürich

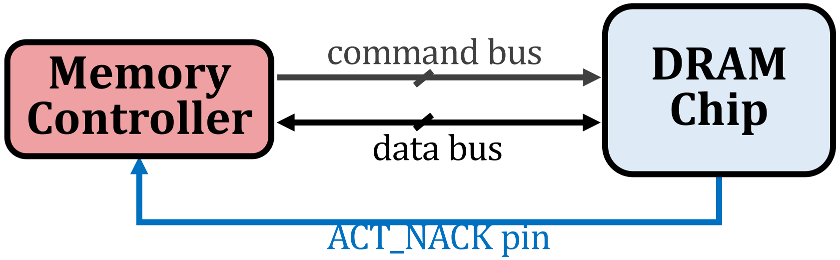
https://arxiv.org/pdf/2207.13358.pdf

Self-Managing DRAM (SMD)

enables autonomous in-DRAM maintenance operations

Key Idea:

Prevent the memory controller from accessing DRAM regions that are *under maintenance* by rejecting row activation (ACT) commands



Leveraging the ability to *reject an ACT*, a maintenance operation can be implemented *completely* within a DRAM chip

SMD-Based Maintenance Mechanisms

DRAM Refresh	**Fixed Rate (SMD-FR)** ***uniformly*** *refreshes* ***all*** *DRAM rows with a* ***fixed*** *refresh period*	**Variable Rate (SMD-VR)** ***skips*** *refreshing rows that can* ***retain their data for longer*** *than the default refresh period*
RowHammer Protection	**Probabilistic (SMD-PRP)** *Performs* ***neighbor row refresh*** *with* ***a small probability*** *on every row activation*	**Deterministic (SMD-DRP)** ***keeps track*** *of most* ***frequently activated*** *rows and performs* ***neighbor*** *row refresh when activation count threshold is exceeded*
Memory Scrubbing	**Periodic Scrubbing (SMD-MS)** *periodically* ***scans*** *the* ***entire*** *DRAM for errors and corrects them*	

Self-Managing DRAM: Summary

The three major DRAM maintenance operations:

- ❖Refresh
- ❖RowHammer Protection
- ❖Memory Scrubbing

Implementing new **maintenance mechanisms** often requires difficult-to-realize changes

Our Goal

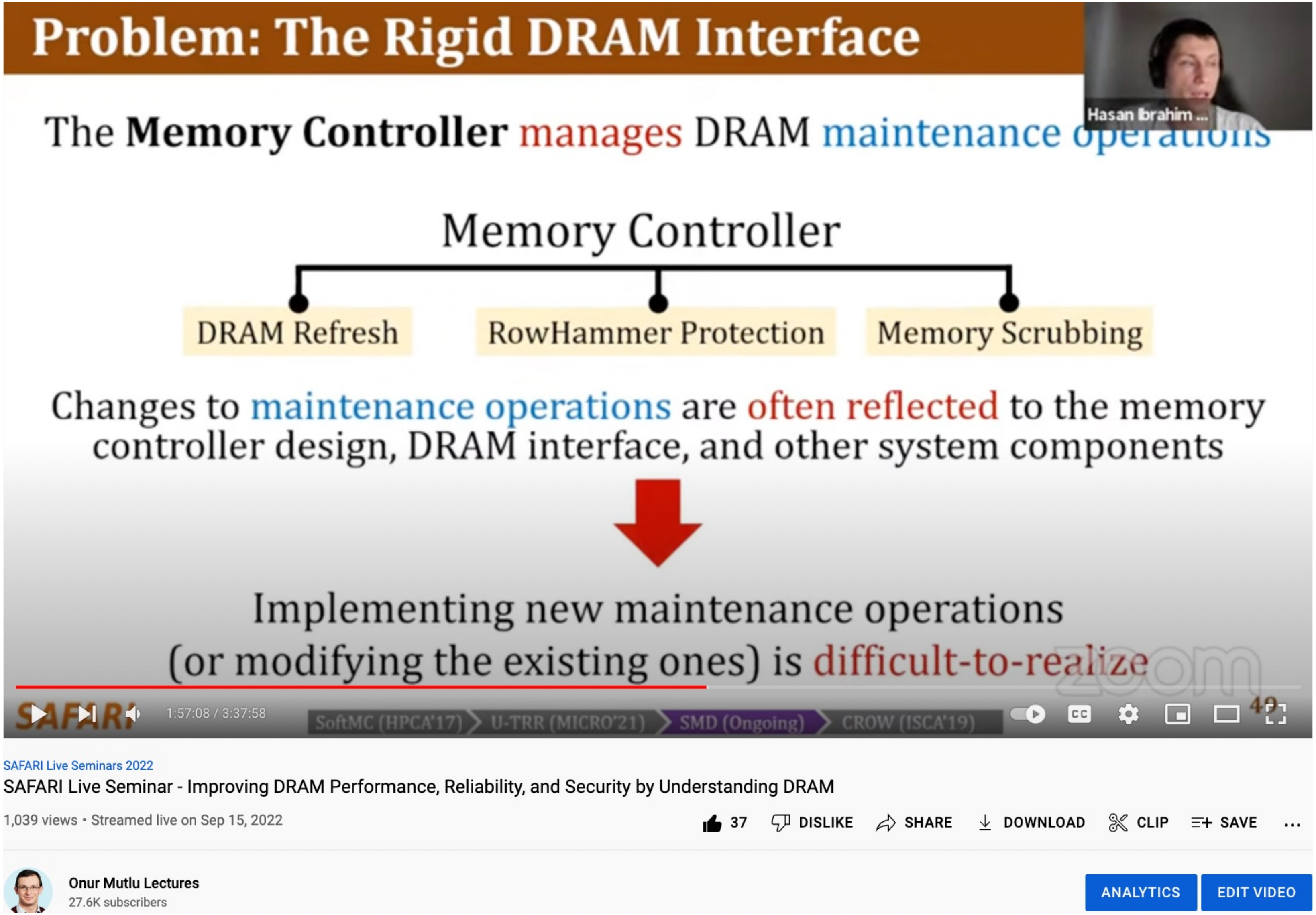
1. Ease the process of enabling new DRAM maintenance operations
2. Enable more efficient in-DRAM maintenance operations

Self-Managing DRAM (SMD)

Enables implementing new **in-DRAM** maintenance mechanisms
with **no further changes** in the *DRAM interface* and *memory controller*

SMD-based *refresh*, *RowHammer protection*, and *scrubbing* achieve
9.2% speedup and **6.2% lower DRAM energy** vs. conventional DRAM

Talk on Self-Managing DRAM



https://www.youtube.com/watch?v=mGa6-vpExbE

ABACuS: Another Intelligent Memory Controller

- Ataberk Olgun, Yahya Can Tugrul, Nisa Bostanci, Ismail Emir Yuksel, Haocong Luo, Steve Rhyner, Abdullah Giray Yaglikci, Geraldo F. Oliveira, and Onur Mutlu,
  "ABACuS: All-Bank Activation Counters for Scalable and Low Overhead RowHammer Mitigation"
  *To appear in Proceedings of the 33rd USENIX Security Symposium (**USENIX Security**)*, Philadelphia, PA, USA, August 2024.
  [arXiv version]
  [ABACuS Source Code]

ABACuS: All-Bank Activation Counters for Scalable and Low Overhead RowHammer Mitigation

Ataberk Olgun   Yahya Can Tugrul   Nisa Bostanci   Ismail Emir Yuksel
Haocong Luo   Steve Rhyner   Abdullah Giray Yaglikci   Geraldo F. Oliveira   Onur Mutlu

ETH Zurich

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

CoMeT: Another Intelligent Memory Controller

- **Appears at HPCA 2024**

CoMeT: Count-Min-Sketch-based Row Tracking to Mitigate RowHammer at Low Cost

F. Nisa Bostancı, İsmail Emir Yüksel, Ataberk Olgun, Konstantinos Kanellopoulos, Yahya Can Tuğrul, A. Giray Yağlıkçı, Mohammad Sadrosadati, Onur Mutlu

ETH Zürich

https://arxiv.org/pdf/2402.18769

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

SVaRD: Another Intelligent Memory Controller

- **Appears at HPCA 2024**

Spatial Variation-Aware Read Disturbance Defenses: Experimental Analysis of Real DRAM Chips and Implications on Future Solutions

Abdullah Giray Yağlıkçı   Yahya Can Tuğrul   Geraldo F. Oliveira
İsmail Emir Yüksel   Ataberk Olgun   Haocong Luo   Onur Mutlu
ETH Zürich

https://arxiv.org/pdf/2402.18652

RowHammer Defenses Can Cause Denial of Service

Leveraging Adversarial Detection to Enable Scalable and Low Overhead RowHammer Mitigations

Oğuzhan Canpolat[§†] A. Giray Yağlıkçı[§] Ataberk Olgun[§] İsmail Emir Yüksel[§] Yahya Can Tuğrul[§†]
Konstantinos Kanellopoulos[§] Oğuz Ergin[†] Onur Mutlu[§]

[§]*ETH Zürich* [†]*TOBB University of Economics and Technology* [*]*SAFARI Research Group*

https://arxiv.org/pdf/2404.13477

BreakHammer

- **Key Observation**: Mitigating DRAM read disturbance causes delays in memory accesses
- **Our Exploit**: Denial of memory service is possible via triggering mitigation mechanisms
- **Key Idea**: Throttling memory accesses of threads that trigger mitigation mechanisms repeatedly
- **BreakHammer**:
  - Detects the threads that repeatedly trigger the mitigation mechanisms
  - Limits their on-the-fly memory request counts and MSHRs
  - Near-zero area overhead and no additional memory access latency
- **Evaluation**:
  - Improves **system performance** by **48.7%** on average (**105.5%** max)
  - Reduces the **maximum slowdown** by **14.6%** on average

Oğuzhan Canpolat, A. Giray Yağlıkçı, Ataberk Olgun, İsmail Emir Yüksel, Yahya Can Tuğrul, Konstantinos Kanellopoulos, Oğuz Ergin, Onur Mutlu, "Leveraging Adversarial Detection to Enable Scalable and Low Overhead RowHammer Mitigations," [cs.CR] 2404.13477, 2024.

Some RowHammer Works in 2024

HPCA 2024
March 2 – 6, 2024

Session 5B: Rowhammer

Location: Sidlaw

Session Chair: TBD

10:00 AM – 10:20 AM

Spatial Variation-Aware Read Disturbance Defenses: Experimental Analysis of Real DRAM Chips and Implications on Future Solutions

Abdullah Giray Yaglikci, Geraldo Francisco de Oliveira Junior, Yahya Can Tugrul, Ismail Yuksel, Ataberk Olgun, Haocong Luo, Onur Mutlu

10:20 AM – 10:40 AM

START: Scalable Tracking for Any Rowhammer Threshold

Anish Saxena, Moinuddin Qureshi

10:40 AM – 11:00 AM

CoMeT: Count-Min Sketch-based Row Tracking to Mitigate RowHammer with Low Cost

Nisa Bostanci, Ismail Emir Yuksel, Ataberk Olgun, Konstantinos Kanellopoulos, Yahya Can Tuğrul, Giray Yaglikci, Mohammad Sadrosadati, Onur Mutlu

33RD USENIX SECURITY SYMPOSIUM

ABACuS: All-Bank Activation Counters for Scalable and Low Overhead RowHammer Mitigation
Ataberk Olgun, Yahya Can Tugrul, Nisa Bostanci, Ismail Emir Yuksel, Haocong Luo, Steve Rhyner, A
Zurich

Go Go Gadget Hammer: Flipping Nested Pointers for Arbitrary Data Leakage
Youssef Tobah, *University of Michigan;* Andrew Kwong, *UNC Chapel Hill;* Ingab Kang
Michigan

SledgeHammer: Amplifying Rowhammer via Bank-level Parallelism
Ingab Kang, *University of Michigan;* Walter Wang and Jason Kim, *Georgia Tech;* Step
Tech; Andrew Kwong, *UNC Chapel Hill;* Yuval Yarom, *Ruhr University Bochum*

ASPLOS 2024

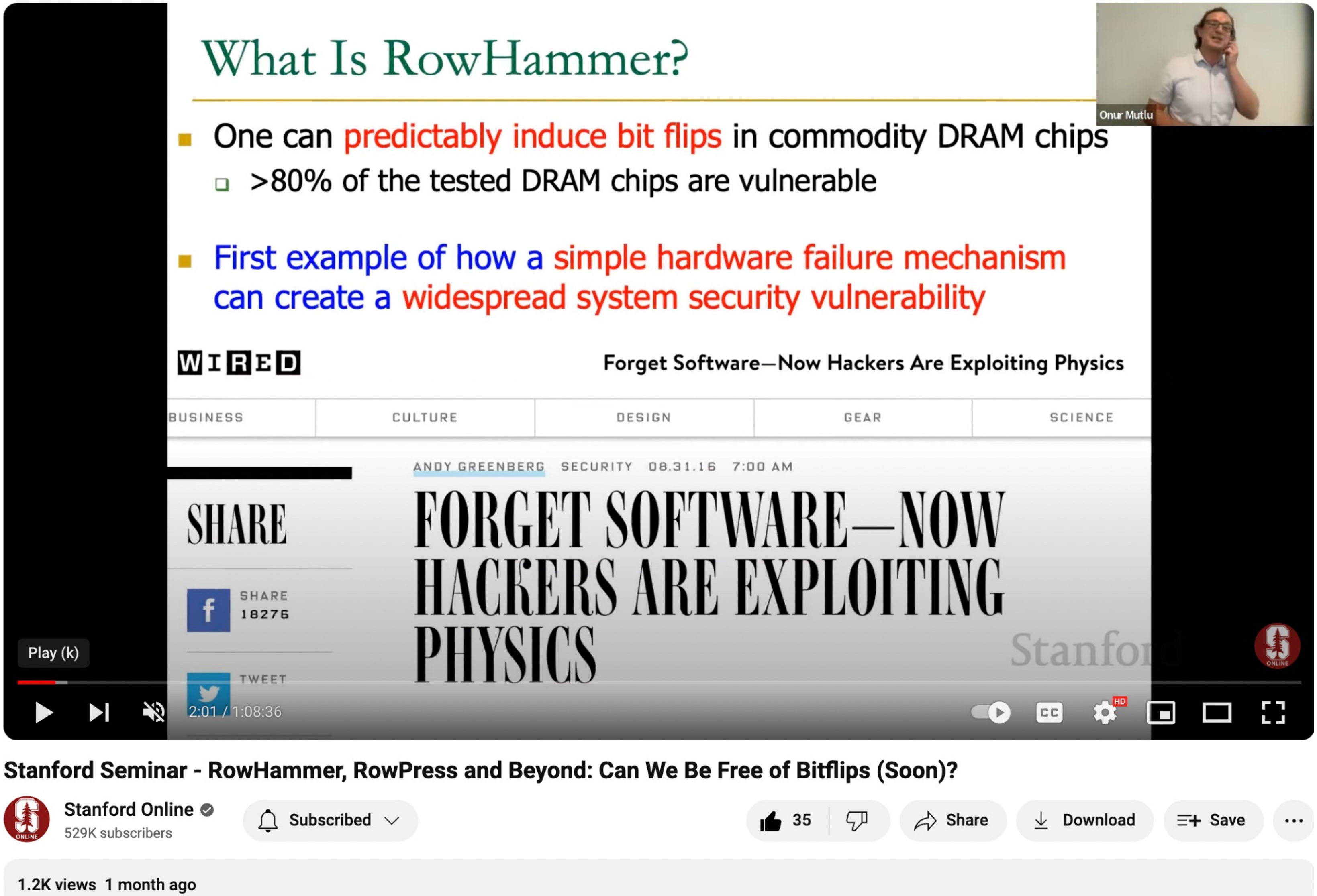
Rubix: Reducing the Overhead of Secure Rowhammer Mitigations via Randomized Line-to-Row Mapping

TAROT: A CXL SmartNIC-Based Defense Against Multi-bit Errors by Row-Hammer Attacks

ISCA 2024

PrIDE: Achieving Secure Rowhammer Mitigation with Low-Cost In-DRAM Trackers

A Recent Detailed RowHammer Lecture



https://www.youtube.com/watch?v=0W7YRRhnunw

Future Memory Robustness Challenges

Future of Main Memory Robustness

- 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

Future of Main Memory Robustness

- DRAM

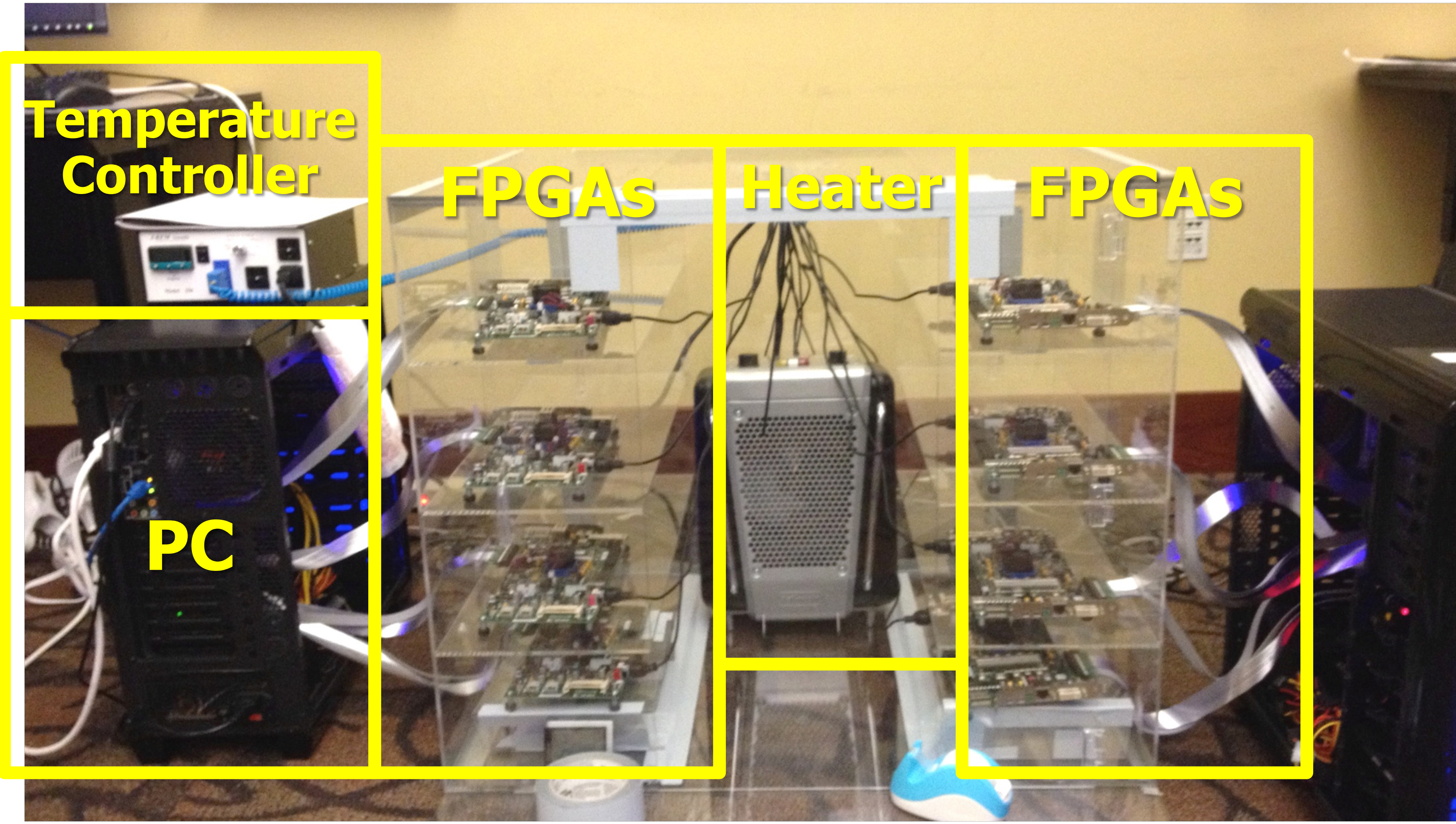
- Flash memory

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

Architecting Robust Memory Systems

- **Understand**: Methods for vulnerability modeling & discovery
  - Modeling and prediction based on real (device) data and analysis
- **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/safety/reliability
  - High coverage and good interaction with system reliability methods

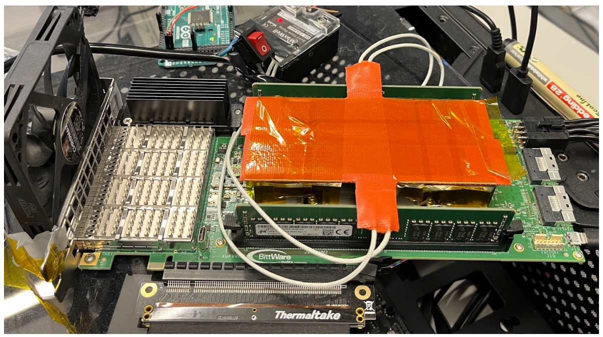
Understand and Model with Experiments (DRAM)

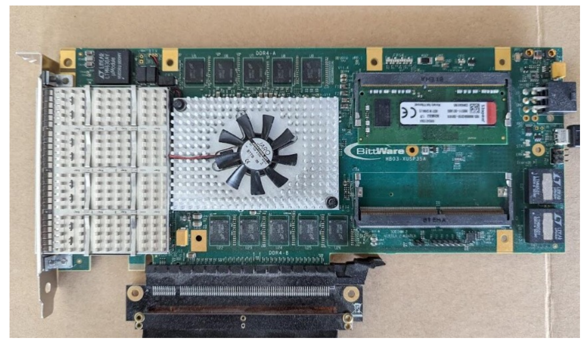


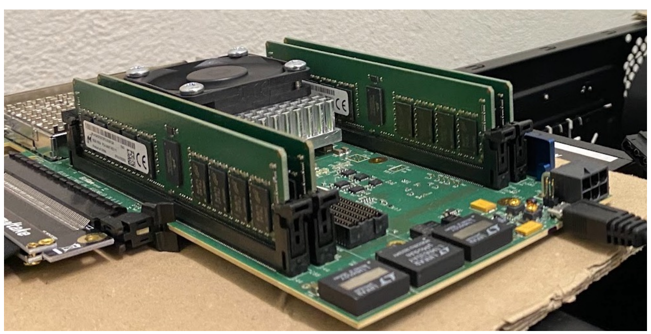
Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.

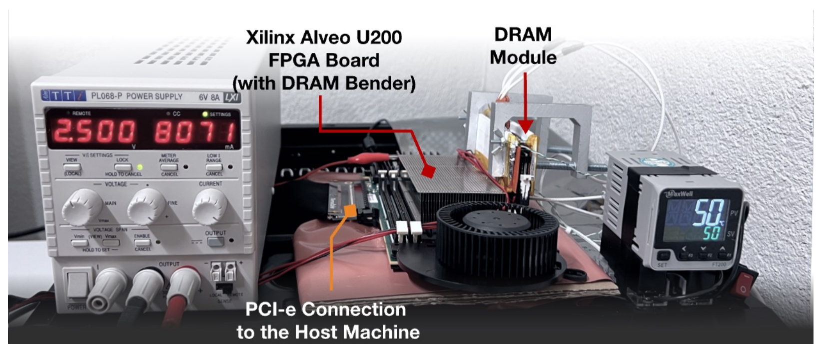
Understand and Model with Experiments (DRAM)

Five out of the box FPGA-based prototypes



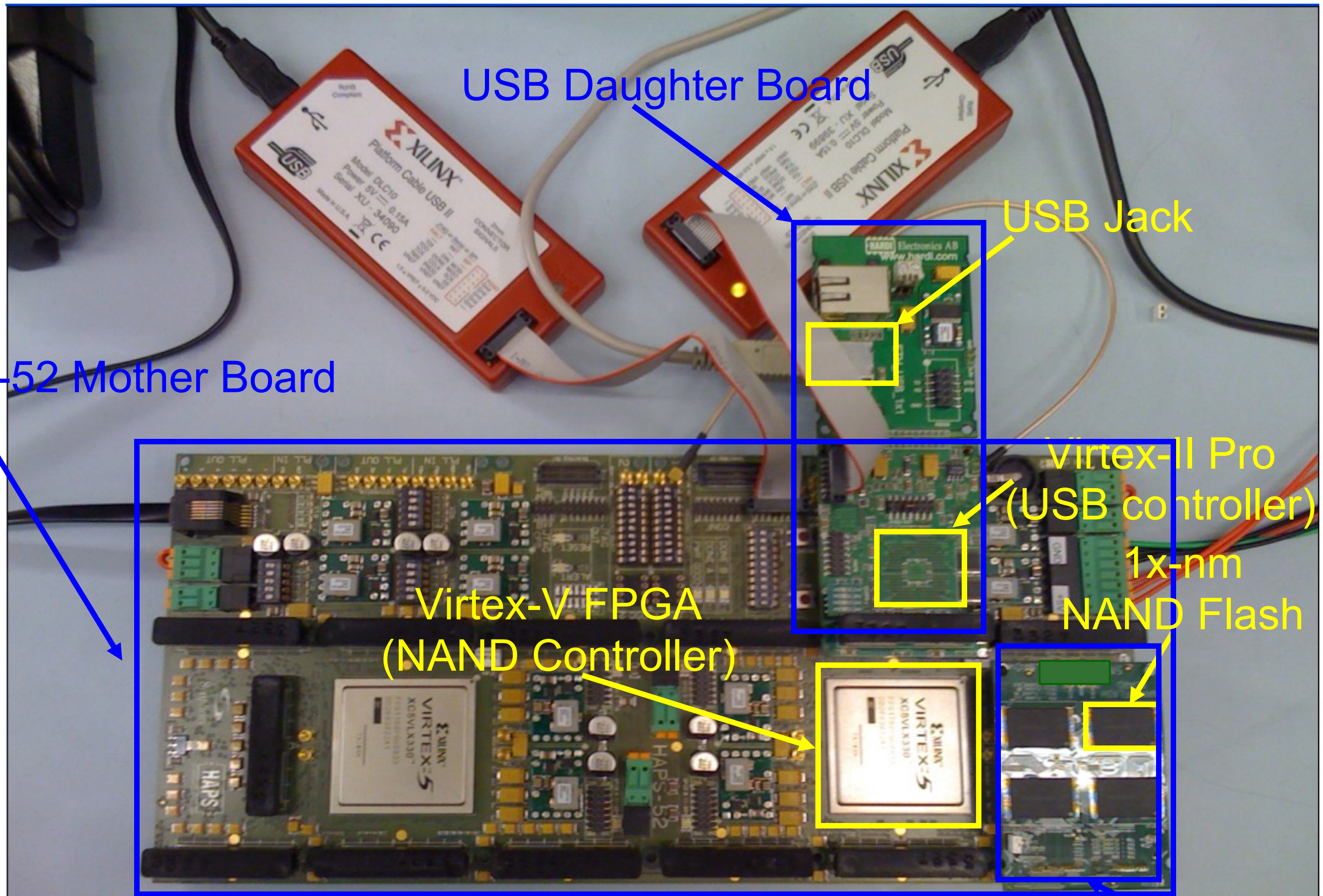






https://github.com/CMU-SAFARI/DRAM-Bender

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]

Cai+, "Error Characterization, Mitigation, and Recovery in Flash Memory Based Solid State Drives," Proc. IEEE 2017.

Collapse of the "Galloping Gertie" (1940)

Source: AP
http://www.wsdot.wa.gov/tnbhistory/connections/connections3.htm

Another Example (1994)

Source: By 최광모 - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=35197984

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)

Source: AFP / Valery HACHE, https://www.capitalfm.co.ke/news/2018/08/genoa-bridge-collapse-what-we-know/

A Most Recent Example (2022)

https://www.post-gazette.com/local/city/2022/01/28/pittsburgh-bridge-collapse-forbes-braddock-avenue-point-breeze-squirrel-hill/stories/202201280075

A Most Recent Example (2022)

https://www.reuters.com/world/us/biden-highlight-manufacturing-jobs-gdp-growth-pittsburgh-2022-01-28/

A Most Recent Example (2022)

https://usa.streetsblog.org/2022/01/28/pittsburgh-bridge-collapse-underscores-urgent-need-for-fix-it-first-policy/

A Most Recent Example (2022)

https://www.npr.org/2022/01/28/1076343656/pittsburgh-bridge-collapse-biden-visit

The Takeaway, Again

Intelligent Memory Controllers

Can Avoid Such Failures

A Takeaway

Main Memory Needs

Intelligent Controllers

for Security, Safety, Reliability, Scaling

Fundamentally
Robust
(Reliable, Secure, Safe)
Computing Architectures

Final Thoughts on RowHammer

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.[4]

IEEE S&P 2003

https://www.cs.princeton.edu/~appel/papers/memerr.pdf

Before RowHammer (II)

Using Memory Errors to Attack a Virtual Machine

Sudhakar Govindavajhala *   Andrew W. Appel
Princeton University
{sudhakar,appel}@cs.princeton.edu

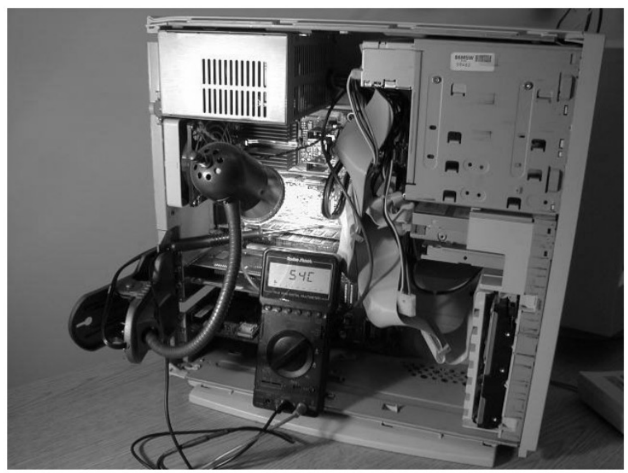


Figure 3. Experimental setup to induce memory errors, showing a PC built from surplus components, clip-on gooseneck lamp, 50-watt spotlight bulb, and digital thermometer. Not shown is the variable AC power supply for the lamp.

IEEE S&P 2003

https://www.cs.princeton.edu/~appel/papers/memerr.pdf

After RowHammer

A simple, exploitable memory error

can be induced by software

WIRED

Forget Software—Now Hackers Are Exploiting Physics

BUSINESS | CULTURE | DESIGN | GEAR | SCIENCE

ANDY GREENBERG SECURITY 08.31.16 7:00 AM

FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

SHARE

SHARE 18276

TWEET

After RowHammer

A simple, exploitable memory error

can be induced by software

ars TECHNICA   BIZ & IT   TECH   SCIENCE   POLICY   CARS   GAMING & CULTURE   ST

SON OF ROWHAMMER —

There’s a new way to flip bits in DRAM, and it works against the latest defenses

New technique produces lots of bitflips and could one day help form an attack.

DAN GOODIN - 10/19/2023, 5:30 AM

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 & bitflip attacks…
  - Tens of papers in top security, architecture, systems 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 HPCA/Usenix Sec 2024)
  - 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…**

Conclusion

Summary: RowHammer

- Memory reliability is reducing
- Reliability issues open up security and safety 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
  - Implications on system security & safety 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 defenses
  - We are developing principled models, methodologies, solutions

Acknowledgments

SAFARI

SAFARI Research Group

safari.ethz.ch

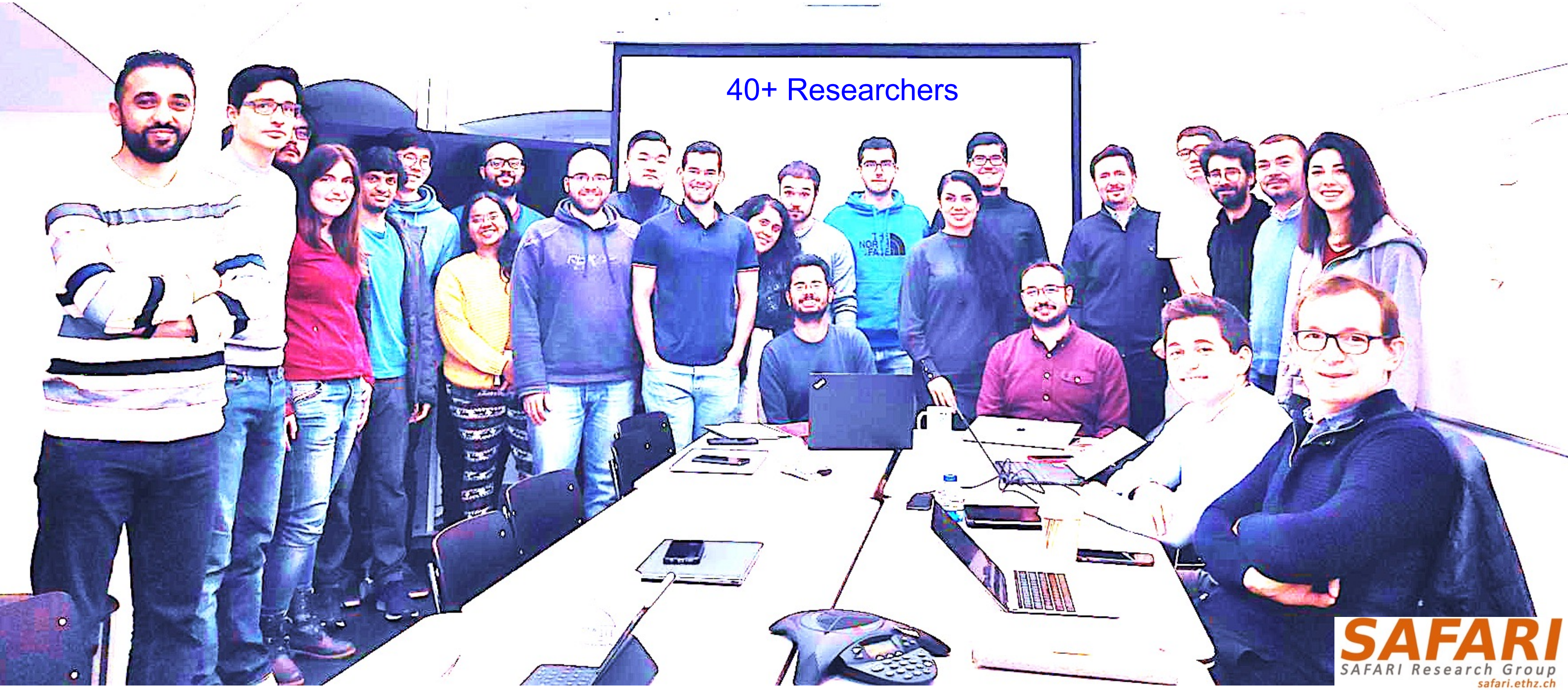
Think BIG, Aim HIGH!

https://safari.ethz.ch

SAFARI Research Group

Computer architecture, HW/SW, systems, bioinformatics, security, memory

https://safari.ethz.ch/safari-newsletter-january-2021/



Think BIG, Aim HIGH!

SAFARI

https://safari.ethz.ch

SAFARI Research Group: December 2021

- https://safari.ethz.ch/safari-newsletter-december-2021/

SAFARI Newsletter June 2023 Edition

- https://safari.ethz.ch/safari-newsletter-june-2023/

SAFARI
SAFARI Research Group

Think Big, Aim High

ETH zürich

View in your browser
June 2023

SAFARI Introduction & Research

Computer architecture, HW/SW, systems, bioinformatics, security, memory

SAFARI Research Group
safari.ethz.ch

Think BIG, Aim HIGH!

https://www.youtube.com/watch?v=mV2OuB2djEs

Referenced Papers, Talks, Artifacts

- All are available at

  https://people.inf.ethz.ch/omutlu/projects.htm

  https://www.youtube.com/onurmutlulectures

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

Open Source Tools: SAFARI GitHub

SAFARI Research Group at ETH Zurich and Carnegie Mellon University

Site for source code and tools distribution from SAFARI Research Group at ETH Zurich and Carnegie Mellon University.

440 followers | ETH Zurich and Carnegie Mellon U... | https://safari.ethz.ch/ | omutlu@gmail.com

Overview | Repositories 98 | Projects | Packages | People 13

ramulator Public

A Fast and Extensible DRAM Simulator, with built-in support for modeling many different DRAM technologies including DDRx, LPDDRx, GDDRx, WIOx, HBMx, and various academic proposals. Described in the...

C++ 519 206

prim-benchmarks Public

PrIM (Processing-In-Memory benchmarks) is the first benchmark suite for a real-world processing-in-memory (PIM) architecture. PrIM is developed to evaluate, analyze, and characterize the first publ...

C 125 45

MQSim Public

MQSim is a fast and accurate simulator modeling the performance of modern multi-queue (MQ) SSDs as well as traditional SATA based SSDs. MQSim faithfully models new high-bandwidth protocol implement...

C++ 265 144

rowhammer Public

Source code for testing the Row Hammer error mechanism in DRAM devices. Described in the ISCA 2014 paper by Kim et al. at http://users.ece.cmu.edu/~omutlu/pub/dram-row-hammer_isca14.pdf.

C 210 43

SoftMC Public

SoftMC is an experimental FPGA-based memory controller design that can be used to develop tests for DDR3 SODIMMs using a C++ based API. The design, the interface, and its capabilities and limitatio...

Verilog 120 27

Pythia Public

A customizable hardware prefetching framework using online reinforcement learning as described in the MICRO 2021 paper by Bera et al. (https://arxiv.org/pdf/2109.12021.pdf).

C++ 107 34

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

Ramulator 2.0

- Haocong Luo, Yahya Can Tugrul, F. Nisa Bostanci, Ataberk Olgun, A. Giray Yaglikci, and Onur Mutlu,
"Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator"
*Preprint on **arxiv***, August 2023.
[arXiv version]
[Ramulator 2.0 Source Code]

Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator

Haocong Luo, Yahya Can Tuğrul, F. Nisa Bostancı, Ataberk Olgun, A. Giray Yağlıkçı, and Onur Mutlu

https://arxiv.org/pdf/2308.11030.pdf

https://github.com/CMU-SAFARI/ramulator2

DRAM Bender

- Ataberk Olgun, Hasan Hassan, A Giray Yağlıkçı, Yahya Can Tuğrul, Lois Orosa, Haocong Luo, Minesh Patel, Oğuz Ergin, and Onur Mutlu,
"DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips"
IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (***TCAD***), 2023.
[Extended arXiv version]
[DRAM Bender Source Code]
[DRAM Bender Tutorial Video (43 minutes)]

DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips

Ataberk Olgun[§] Hasan Hassan[§] A. Giray Yağlıkçı[§] Yahya Can Tuğrul[§†]
Lois Orosa[§⊙] Haocong Luo[§] Minesh Patel[§] Oğuz Ergin[†] Onur Mutlu[§]
[§] *ETH Zürich* [†] *TOBB ETÜ* [⊙] *Galician Supercomputing Center*

https://github.com/CMU-SAFARI/DRAM-Bender

RowHammer & DRAM Exploration (Fall 2022)

- **Fall 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=softmc
- **Spring 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=softmc

- **Youtube Livestream (Spring 2022):**
  - https://www.youtube.com/watch?v=r5QxuoJWttg&list=PL5Q2soXY2Zi_1trfCckr6PTN8WR72icUO

- Bachelor's course
  - Elective at ETH Zurich
  - Introduction to DRAM organization & operation
  - Tutorial on using FPGA-based infrastructure
  - Verilog & C++
  - Potential research exploration

Lecture Video Playlist on YouTube

Lecture Playlist

2022 Meetings/Schedule (Tentative)

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W0	23.02 Wed.	YouTube Video	P&S SoftMC Tutorial	SoftMC Tutorial Slides (PDF) (PPT)	
W1	08.03 Tue.	YouTube Video	M1: Logistics & Intro to DRAM and SoftMC (PDF) (PPT)	Required Materials Recommended Materials	HW0
W2	15.03 Tue.	YouTube Video	M2: Revisiting RowHammer (PDF) (PPT)	(Paper PDF)	
W3	22.03 Tue.	YouTube Video	M3: Uncovering in-DRAM TRR & TRRespass (PDF) (PPT)		
W4	29.03 Tue.	YouTube Video	M4: Deeper Look Into RowHammer's Sensitivities (PDF) (PPT)		
W5	05.04 Tue.	YouTube Video	M5: QUAC-TRNG (PDF) (PPT)		
W6	12.04 Tue.	YouTube Video	M6: PiDRAM (PDF) (PPT)		

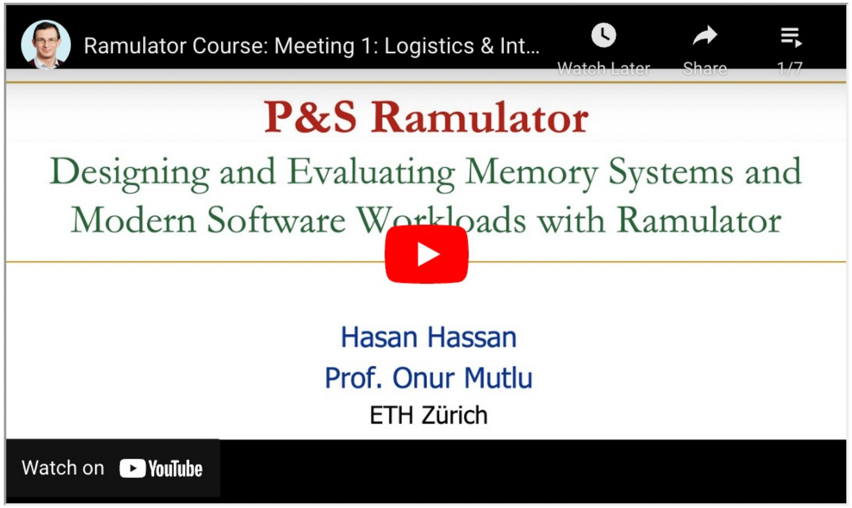
https://www.youtube.com/onurmutlulectures

Exploration of Emerging Memory Systems (Fall 2022)

- [Fall 2022 Edition:](https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=ramulator)
  - https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=ramulator
- [Spring 2022 Edition:](https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=ramulator)
  - https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=ramulator
- Youtube Livestream (Spring 2022):
  - https://www.youtube.com/watch?v=aM-lIXRQd3s&list=PL5Q2soXY2Zi_TlmLGw_Z8hBo2925ZApqV
- Bachelor's course
  - Elective at ETH Zurich
  - Introduction to memory system simulation
  - Tutorial on using Ramulator
  - C++
  - Potential research exploration

Lecture Video Playlist on YouTube

Lecture Playlist



2022 Meetings/Schedule (Tentative)

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	09.03 Wed.	YouTube Video	M1: Logistics & Intro to Simulating Memory Systems Using Ramulator (PDF) (PPT)		HW0
W2	16.03 Fri.	YouTube Video	M2: Tutorial on Using Ramulator (PDF) (PPT)		
W3	25.02 Fri.	YouTube Video	M3: BlockHammer (PDF) (PPT)		
W4	01.04 Fri.	YouTube Video	M4: CLR-DRAM (PDF) (PPT)		
W5	08.04 Fri.	YouTube Video	M5: SIMDRAM (PDF) (PPT)		
W6	29.04 Fri.	YouTube Video	M6: DAMOV (PDF) (PPT)		
W7	06.05 Fri.	YouTube Video	M7: Syncron (PDF) (PPT)		


Rowhammer

RowHammer, RowPress & Beyond

Can We Be Free of Bitflips (Soon)?

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu

13 June 2024

University of Colorado Boulder

SAFARI ETH zürich Carnegie Mellon

More RowHammer in 2020-2024

RowHammer in 2020 (I)

MICRO 2020 | Submit Work | Program | Atte

Session 1A: Security & Privacy I

5:00 PM CEST – 5:15 PM CEST

Graphene: Strong yet Lightweight Row Hammer Protection

Yeonhong Park, Woosuk Kwon, Eojin Lee, Tae Jun Ham, Jung Ho Ahn, Jae W. Lee (Seoul National University)

5:15 PM CEST – 5:30 PM CEST

Persist Level Parallelism: Streamlining Integrity Tree Updates for Secure Persistent Memory

Alexander Freij, Shougang Yuan, Huiyang Zhou (NC State University); Yan Solihin (University of Central Florida)

5:30 PM CEST – 5:45 PM CEST

PThammer: Cross-User-Kernel-Boundary Rowhammer through Implicit Accesses

Zhi Zhang (University of New South Wales and Data61, CSIRO, Australia); Yueqiang Cheng (Baidu Security); Dongxi Liu, Surya Nepal (Data61, CSIRO, Australia); Zhi Wang (Florida State University); Yuval Yarom (University of Adelaide and Data61, CSIRO, Australia)

RowHammer in 2020 (II)

S & P | Home | Program | Call For... | Attend | Workshops

Session #5: Rowhammer

Room 2

Session chair: Michael Franz (UC Irvine)

RAMBleed: Reading Bits in Memory Without Accessing Them
Andrew Kwong (University of Michigan), Daniel Genkin (University of Michigan), Daniel Gruss
Data61)

Are We Susceptible to Rowhammer? An End-to-End Methodology for Cloud Providers
Lucian Cojocar (Microsoft Research), Jeremie Kim (ETH Zurich, CMU), Minesh Patel (ETH Zu
(Microsoft Research), Onur Mutlu (ETH Zurich, CMU)

Leveraging EM Side-Channel Information to Detect Rowhammer Attacks
Zhenkai Zhang (Texas Tech University), Zihao Zhan (Vanderbilt University), Daniel Balasubrar
Peter Volgyesi (Vanderbilt University), Xenofon Koutsoukos (Vanderbilt University)

TRRespass: Exploiting the Many Sides of Target Row Refresh
Pietro Frigo (Vrije Universiteit Amsterdam, The Netherlands), Emanuele Vannacci (Vrije Univei
Veen (Qualcomm Technologies, Inc.), Onur Mutlu (ETH Zürich), Cristiano Giuffrida (Vrije Unive
The Netherlands), Kaveh Razavi (Vrije Universiteit Amsterdam, The Netherlands)

RowHammer in 2020 (III)

29TH USENIX SECURITY SYMPOSIUM

ATTEND PROGRAM PARTICIPATE SPONSORS ABOUT

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

Fan Yao, *University of Central Florida;* Adnan Siraj Rakin and Deliang Fan, *Arizona State University*

AVAILABLE MEDIA

Show details ▸

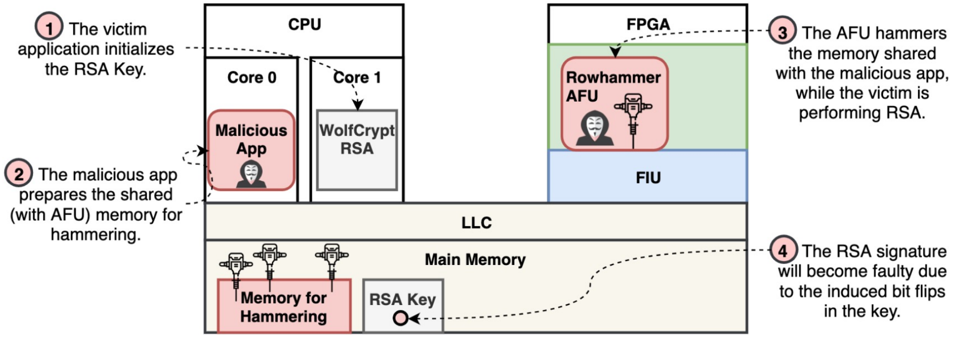
RowHammer in 2020 (IV)

- CHES 2020

JackHammer: Efficient Rowhammer on Heterogeneous FPGA-CPU Platforms

Zane Weissman[1], Thore Tiemann[2], Daniel Moghimi[1], Evan Custodio[3], Thomas Eisenbarth[2] and Berk Sunar[1]

[1] Worcester Polytechnic Institute, MA, USA
zweissman@wpi.edu, amoghimi@wpi.edu, sunar@wpi.edu
[2] University of Lübeck, Lübeck, Germany
thore.tiemann@student.uni-luebeck.de, thomas.eisenbarth@uni-luebeck.de
[3] Intel Corporation, Hudson, MA, USA
evan.custodio@intel.com



An **FPGA-based** RowHammer attack recovering **private keys** twice as fast compared to **CPU-based** attacks

RowHammer in 2021 (I)

HotOS XVIII

The 18th Workshop on Hot Topics in Operating Systems

~~31 May~~ 1 June–3 June 2021, Cyberspace, People's Couches, and Zoom

Stop! Hammer Time: Rethinking Our Approach to Rowhammer Mitigations

RowHammer in 2021 (II)

30TH USENIX SECURITY SYMPOSIUM

ATTEND PROGRAM PARTICIPATE SPONSORS ABOUT

SMASH: Synchronized Many-sided Rowhammer Attacks from JavaScript

RowHammer in 2021 (III)

Session 10A: Security & Privacy III

Session Chair: Hoda Naghibijouybari (Binghamton)

9:00 PM CEST – 9:15 PM CEST

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

Lois Orosa, Abdullah Giray Yaglikci, Haocong Luo (ETH Zurich); Ataberk Olgun (TOBB University of Economics and Technology); Jisung Park, Hasan Hassan, Minesh Patel, Jeremie S. Kim, Onur Mutlu (ETH Zurich)

Paper

9:15 PM CEST – 9:30 PM CEST

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

Hasan Hassan (ETH Zurich); Yahya Can Tugrul (TOBB University of Economics and Technology); Jeremie S. Kim (ETH Zurich); Victor van der Veen (Qualcomm); Kaveh Razavi, Onur Mutlu (ETH Zurich)

Paper

RowHammer in 2022 (I)

MAY 22-26, 2022 AT THE HYATT REGENCY, SAN FRANCISCO, CA

43rd IEEE Symposium on Security and Privacy

BLACKSMITH: Scalable Rowhammering in the Frequency Domain

SpecHammer: Combining Spectre and Rowhammer for New Speculative Attacks

PROTRR: Principled yet Optimal In-DRAM Target Row Refresh

DeepSteal: Advanced Model Extractions Leveraging Efficient Weight Stealing in Memories

RowHammer in 2022 (II)

Randomized Row-Swap: Mitigating Row Hammer by Breaking Spatial Correlation between Aggressor and Victim Rows

RowHammer in 2022 (III)

HPCA 2022

The 28th IEEE International Symposium on High-Performance Computer Architecture (HPCA-28), Seoul, South Korea

SafeGuard: Reducing the Security Risk from Row-Hammer via Low-Cost Integrity Protection

Mithril: Cooperative Row Hammer Protection on Commodity DRAM Leveraging Managed Refresh

RowHammer in 2022 (IV)

IRPS
International Reliability Physics Symposium

IRPS 2022

The Price of Secrecy: How Hiding Internal DRAM Topologies Hurts Rowhammer Defenses

Stefan Saroiu, Alec Wolman, Lucian Cojocar
Microsoft

RowHammer in 2022 (V)

31ST USENIX SECURITY SYMPOSIUM

AUGUST 10–12, 2022
BOSTON, MA, USA

Half-Double: Hammering From the Next Row Over

Andreas Kogler[1] Jonas Juffinger[1,2] Salman Qazi[3] Yoongu Kim[3] Moritz Lipp[4*]
Nicolas Boichat[3] Eric Shiu[5] Mattias Nissler[3] Daniel Gruss[1]

[1]*Graz University of Technology* [2]*Lamarr Security Research* [3]*Google*
[4]*Amazon Web Services* [5]*Rivos*

RowHammer in 2022 (VI)

HammerScope: Observing DRAM Power Consumption Using Rowhammer

When Frodo Flips:
End-to-End Key Recovery on FrodoKEM via Rowhammer

RowHammer in 2022 (VII)

AQUA: Scalable Rowhammer Mitigation by Quarantining Aggressor Rows at Runtime

Anish Saxena, Gururaj Saileshwar (Georgia Institute of Technology); Prashant J. Nair (University of British Columbia); Moinuddin Qureshi (Georgia Institute of Technology)

HiRA: Hidden Row Activation for Reducing Refresh Latency of Off-the-Shelf DRAM Chips

Abdullah Giray Yaglikci (ETH Zürich); Ataberk Olgun (TOBB University of Economics and Technology); Lois Orosa, Minesh Patel, Haocong Luo, Hasan Hassan (ETH Zürich); Oguz Ergin (TOBB University of Economics and Technology); Onur Mutlu (ETH Zürich)

RowHammer in 2022 (VII)

- A. Giray Yaglıkcı, Ataberk Olgun, Minesh Patel, Haocong Luo, Hasan Hassan, Lois Orosa, Oguz Ergin, and Onur Mutlu,
  "HiRA: Hidden Row Activation for Reducing Refresh Latency of Off-the-Shelf DRAM Chips"
  Proceedings of the 55th International Symposium on Microarchitecture (***MICRO***), Chicago, IL, USA, October 2022.
  [Slides (pptx) (pdf)]
  [Longer Lecture Slides (pptx) (pdf)]
  [Lecture Video (36 minutes)]
  [arXiv version]

HiRA: Hidden Row Activation for Reducing Refresh Latency of Off-the-Shelf DRAM Chips

A. Giray Yağlıkçı[1] Ataberk Olgun[1,2] Minesh Patel[1] Haocong Luo[1] Hasan Hassan[1]
Lois Orosa[1,3] Oğuz Ergin[2] Onur Mutlu[1]
[1]*ETH Zürich* [2]TOBB University of Economics and Technology [3]*Galicia Supercomputing Center (CESGA)*

https://arxiv.org/pdf/2209.10198.pdf

RowHammer in 2022 (VIII)

A Case for Transparent Reliability in DRAM Systems

Minesh Patel[†] Taha Shahroodi[‡†] Aditya Manglik[†] A. Giray Yağlıkçı[†]
Ataberk Olgun[†] Haocong Luo[†] Onur Mutlu[†]

[†]*ETH Zürich* [‡]*TU Delft*

https://arxiv.org/pdf/2204.10378.pdf

RowHammer in 2022 (IX)

A Case for Self-Managing DRAM Chips: Improving Performance, Efficiency, Reliability, and Security via Autonomous in-DRAM Maintenance Operations

Hasan Hassan Ataberk Olgun A. Giray Yağlıkçı
Haocong Luo Onur Mutlu

ETH Zürich

https://arxiv.org/pdf/2207.13358.pdf

RowHammer in 2023 (I)

MAY 22-26, 2023 AT THE HYATT REGENCY, SAN FRANCISCO, CA

44th IEEE Symposium on Security and Privacy

Session 6C: Rowhammer and spectre	Bayview AB	11:00 AM – 12:15 PM

Session Chair: Eyal Ronen

REGA: Scalable Rowhammer Mitigation with Refresh-Generating Activations
Michele Marazzi (ETH Zurich), Flavien Solt (ETH Zurich), Patrick Jattke (ETH Zurich), Kubo Takashi (Zentel Japan), Kaveh Razavi (ETH Zurich)

CSI:Rowhammer - Cryptographic Security and Integrity against Rowhammer
Jonas Juffinger (Lamarr Security Research, Graz University of Technology, Austria), Lukas Lamster (Graz University of Technology, Austria), Andreas Kogler (Graz University of Technology, Austria), Maria Eichlseder (Graz University of Technology, Austria), Moritz Lipp (Amazon Web Services, Austria), Daniel Gruss (Graz University of Technology, Austria)

Jolt: Recovering TLS Signing Keys via Rowhammer Faults
Koksal Mus (Worcester Polytechnic Institute), Yarkın Doröz (Worcester Polytechnic Institute), M. Caner Tol (Worcester Polytechnic Institute), Kristi Rahman (Worcester Polytechnic Institute), Berk Sunar (Worcester Polytechnic Institute)

RowHammer in 2023 (II)

HPCA 2023

The 29th IEEE International Symposium on High-Performance Computer Architecture (HPCA-29)

Scalable and Secure Row-Swap: Efficient and Safe Row Hammer Mitigation in Memory Systems
Jeonghyun Woo (University of British Columbia), Gururaj Saileshwar (Georgia Institute of Technology), Prashant J. Nair (University of British Columbia)

SHADOW: Preventing Row Hammer in DRAM with Intra-Subarray Row Shuffling
Minbok Wi (Seoul National University), Jaehyun Park (Seoul National University), Seoyoung Ko (Seoul National University), Michael Jaemin Kim (Seoul National University), Nam Sung Kim (UIUC), Eojin Lee (Inha University), Jung Ho Ahn (Seoul National University)

RowHammer in 2023 (III): SK Hynix

ISSCC 2023 / SESSION 28 / HIGH-DENSITY MEMORIES /

28.8 A 1.1V 16Gb DDR5 DRAM with Probabilistic-Aggressor Tracking, Refresh-Management Functionality, Per-Row Hammer Tracking, a Multi-Step Precharge, and Core-Bias Modulation for Security and Reliability Enhancement

Woongrae Kim, Chulmoon Jung, Seongnyuh Yoo, Duckhwa Hong, Jeongjin Hwang, Jungmin Yoon, Ohyong Jung, Joonwoo Choi, Sanga Hyun, Mankeun Kang, Sangho Lee, Dohong Kim, Sanghyun Ku, Donhyun Choi, Nogeun Joo, Sangwoo Yoon, Junseok Noh, Byeongyong Go, Cheolhoe Kim, Sunil Hwang, Mihyun Hwang, Seol-Min Yi, Hyungmin Kim, Sanghyuk Heo, Yeonsu Jang, Kyoungchul Jang, Shinho Chu, Yoonna Oh, Kwidong Kim, Junghyun Kim, Soohwan Kim, Jeongtae Hwang, Sangil Park, Junphyo Lee, Inchul Jeong, Joohwan Cho, Jonghwan Kim

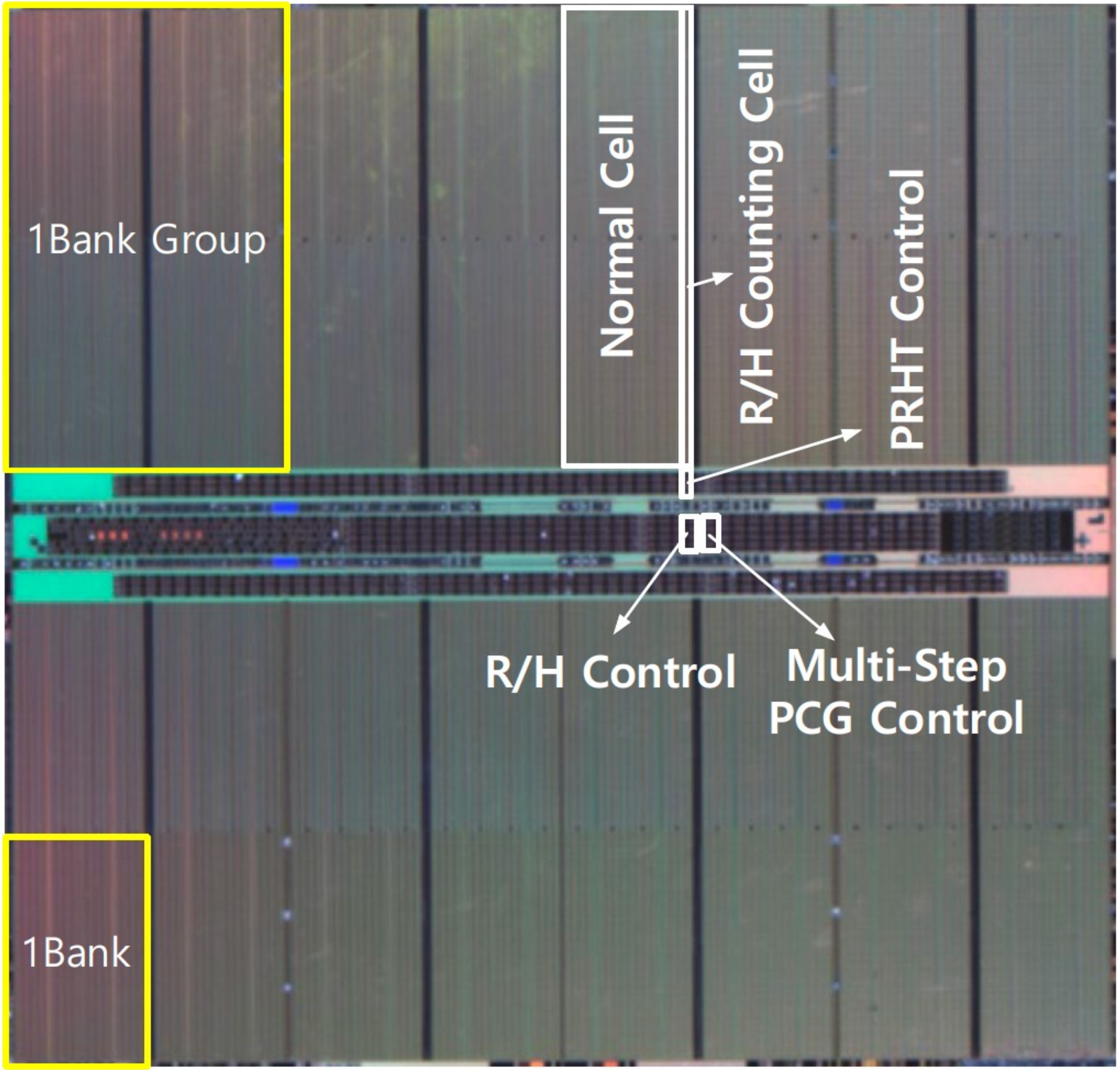
SK hynix Semiconductor, Icheon, Korea

Industry's RowHammer Solutions (I)

SK hynix Semiconductor, Icheon, Korea

DRAM products have been recently adopted in a wide range of high-performance computing applications: such as in cloud computing, in big data systems, and IoT devices. This demand creates larger memory capacity requirements, thereby requiring aggressive DRAM technology node scaling to reduce the cost per bit [1,2]. However, DRAM manufacturers are facing technology scaling challenges due to row hammer and refresh retention time beyond 1a-nm [2]. Row hammer is a failure mechanism, where repeatedly activating a DRAM row disturbs data in adjacent rows. Scaling down severely threatens reliability since a reduction of DRAM cell size leads to a reduction in the intrinsic row hammer tolerance [2,3]. To improve row hammer tolerance, there is a need to probabilistically activate adjacent rows with carefully sampled active addresses and to improve intrinsic row hammer tolerance [2]. In this paper, row-hammer-protection and refresh-management schemes are presented to guarantee DRAM security and reliability despite the aggressive scaling from 1a-nm to sub 10-nm nodes. The probabilistic-aggressor-tracking scheme with a refresh-management function (RFM) and per-row hammer tracking (PRHT) improve DRAM resilience. A multi-step precharge reinforces intrinsic row-hammer tolerance and a core-bias modulation improves retention time: even in the face of cell-transistor degradation due to technology scaling. This comprehensive scheme leads to a reduced probability of failure, due to row hammer attacks, by 93.1% and an improvement in retention time by 17%.

Industry's RowHammer Solutions (II)



ISSCC 2023 / SESSION 28 / HIGH-DENSITY MEMORIES /

28.8 A 1.1V 16Gb DDR5 DRAM with Probabilistic-Aggressor Tracking, Refresh-Management Functionality, Per-Row Hammer Tracking, a Multi-Step Precharge, and Core-Bias Modulation for Security and Reliability Enhancement

Woongrae Kim, Chulmoon Jung, Seongnyuh Yoo, Duckhwa Hong, Jeongjin Hwang, Jungmin Yoon, Ohyong Jung, Joonwoo Choi, Sanga Hyun, Mankeun Kang, Sangho Lee, Dohong Kim, Sanghyun Ku, Donhyun Choi, Nogeun Joo, Sangwoo Yoon, Junseok Noh, Byeongyong Go, Cheolhoe Kim, Sunil Hwang, Mihyun Hwang, Seol-Min Yi, Hyungmin Kim, Sanghyuk Heo, Yeonsu Jang, Kyoungchul Jang, Shinho Chu, Yoonna Oh, Kwidong Kim, Junghyun Kim, Soohwan Kim, Jeongtae Hwang, Sangil Park, Junphyo Lee, Inchul Jeong, Joohwan Cho, Jonghwan Kim

SK hynix Semiconductor, Icheon, Korea

RowHammer in 2023 (IV): Samsung

DSAC: Low-Cost Rowhammer Mitigation Using In-DRAM Stochastic and Approximate Counting Algorithm

Seungki Hong Dongha Kim Jaehyung Lee Reum Oh
Changsik Yoo Sangjoon Hwang Jooyoung Lee

DRAM Design Team, Memory Division, Samsung Electronics

https://arxiv.org/pdf/2302.03591v1.pdf

RowHammer in 2023 (V)

DSN 2023

[28 June, 14:30-16:00] RT-3: Memory 1 (Session Chair: TBD)

Compiler-Implemented Differential Checksums: Effective Detection and Correction of Transient and Permanent Memory Errors (REG)
C. Borchert; H. Schirmeier; O. Spinczyk

PT-Guard: Integrity-Protected Page Tables to Defend Against Breakthrough Rowhammer Attacks (REG)
A. Saxena; G. Saileshwar; J. Juffinger; A. Kogler; D. Gruss; M. Qureshi

Don't Knock! Rowhammer at the Backdoor of DNN Models (REG)
M. Tol; S. Islam; A. Adiletta; B. Sunar; Z. Zhang

[29 June, 16:00-17:30] DS23-4: Hardware Resilience and Human Factors (Session Chair: TBD)

An Experimental Analysis of RowHammer in HBM2 DRAM Chips
Ataberk Olgun, Majd Osseiran, Abdullah Giray Yaglikci, Yahya Can Tugrul, Juan Gomez Luna, Haocong Luo, Behzad Salami, Steve Rhyner and Onur Mutlu

RowHammer in 2023 (VI)

- SOSP 2023

Siloz: Leveraging DRAM Isolation Domains to Prevent Inter-VM Rowhammer

Kevin Loughlin
University of Michigan

Jonah Rosenblum
University of Michigan

Stefan Saroiu
Microsoft

Alec Wolman
Microsoft

Dimitrios Skarlatos
Carnegie Mellon University

Baris Kasikci
University of Washington and Google

RowHammer in 2023 (VII)

- IEEE Computer Architecture Letters, 2023

NoHammer: Preventing Row Hammer with Last-Level Cache Management

Seunghak Lee, Ki-Dong Kang, Gyeongseo Park, Nam Sung Kim, and Daehoon Kim

Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator

Haocong Luo, Yahya Can Tuğrul, F. Nisa Bostancı, Ataberk Olgun, A. Giray Yağlıkçı, and Onur Mutlu

- IEEE Embedded Systems Letters, 2023

Flipping Bits Like a Pro: Precise Rowhammering on Embedded Devices

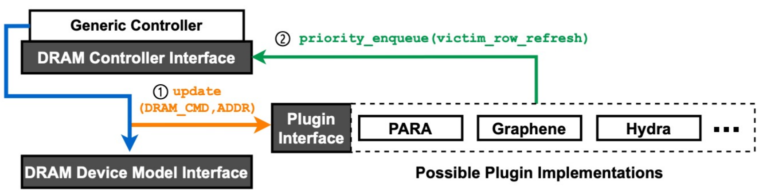
Anandpreet Kaur, Pravin Srivastav, Bibhas Ghoshal
Systems Lab, Indian Institute of Information Technology Allahabad (IIITA)

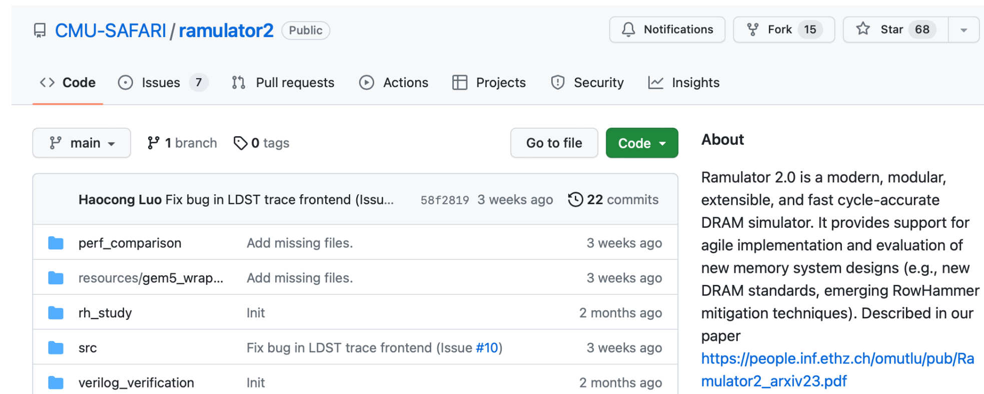
Ramulator 2.0

["Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator"]

IEEE Computer Architecture Letters, August 2023. (*Preprint on **arxiv***)

[arXiv version] [Ramulator 2.0 Source Code]





Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator

Haocong Luo, Yahya Can Tuğrul, F. Nisa Bostancı, Ataberk Olgun, A. Giray Yağlıkçı, and Onur Mutlu

RowHammer in 2023 (VIII)

- MEMSYS 2023

RAMPART: RowHammer Mitigation and Repair for Server Memory Systems

Steven C. Woo
Rambus Labs
Rambus Inc.
San Jose, CA
swoo@rambus.com

Wendy Elsasser
Rambus Labs
Rambus Inc.
San Jose, CA
welsasser@rambus.com

Mike Hamburg
Rambus Labs
Rambus Inc.
San Jose, CA
hamburg@rambus.com

Eric Linstadt
Rambus Labs
Rambus Inc.
San Jose, CA
elinstadt@rambus.com

Michael R. Miller
Rambus Labs
Rambus Inc.
San Jose, CA
michaelm@rambus.com

Taeksang Song
Rambus Labs
Rambus Inc.
San Jose, CA
tsong@rambus.com

James Tringali
Rambus Labs
Rambus Inc.
San Jose, CA
jamestr@rambus.com

- MICRO 2023

How to Kill the Second Bird with One ECC: The Pursuit of Row Hammer Resilient DRAM

Michael Jaemin Kim, Minbok Wi, Jaehyun Park, Seoyoung Ko, Jae Young Choi, Hwayoung Nam (Seoul National University); Nam Sung Kim (University of Illinois Urbana Champaign); Jung Ho Ahn (Seoul National University); Eojin Lee (Inha University)

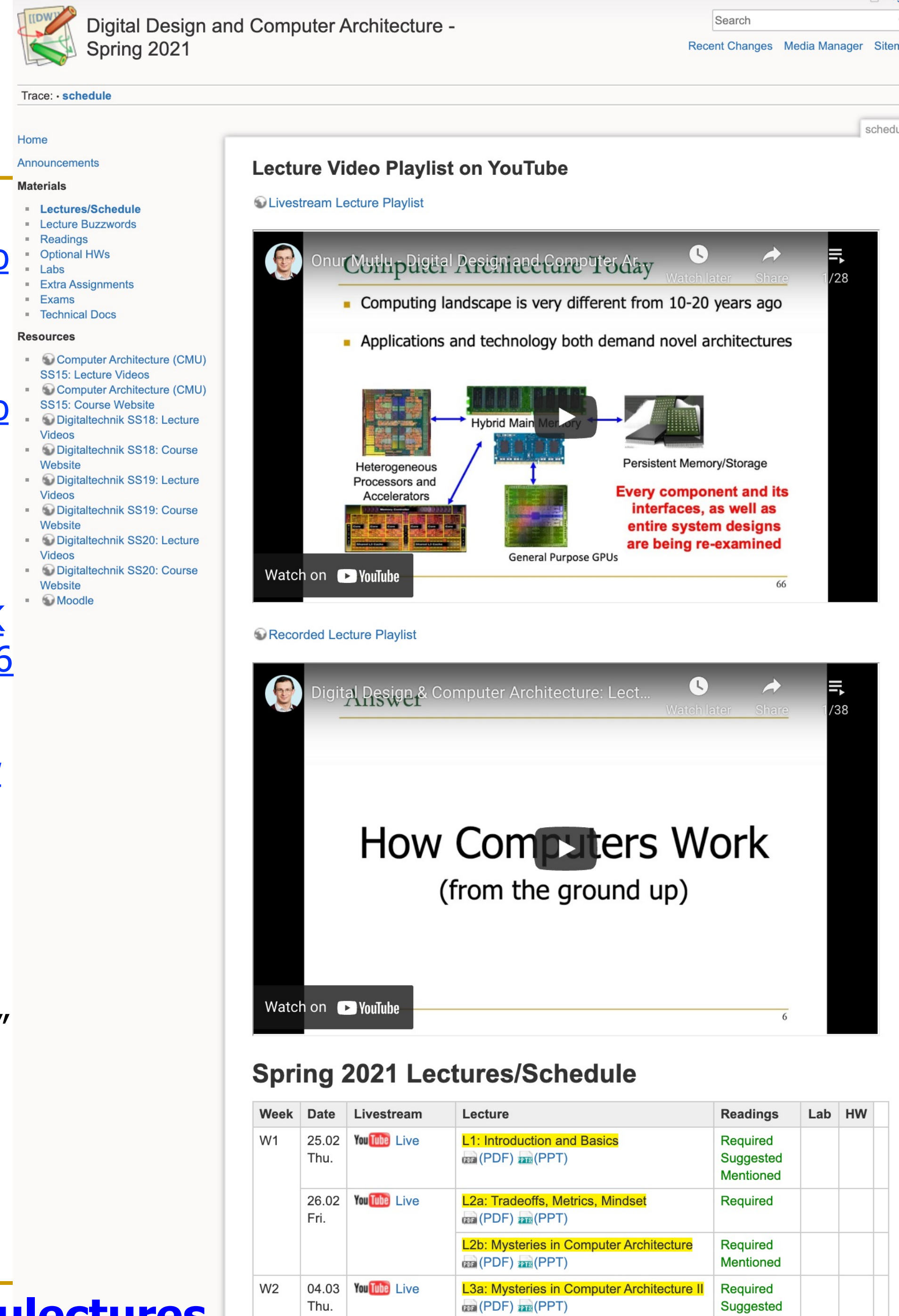
Related Courses

DDCA (Spring 2022)

- **[Spring 2022 Edition:](https://safari.ethz.ch/digitaltechnik/spring2022/doku.php?id=schedule)**
  - https://safari.ethz.ch/digitaltechnik/spring2022/doku.php?id=schedule
- **Spring 2021 Edition:**
  - https://safari.ethz.ch/digitaltechnik/spring2021/doku.php?id=schedule

- **Youtube Livestream (Spring 2022):**
  - https://www.youtube.com/watch?v=cpXdE3HwvK0&list=PL5Q2soXY2Zi97Ya5DEUpMpO2bbAoaG7c6
- **Youtube Livestream (Spring 2021):**
  - https://www.youtube.com/watch?v=LbC0EZY8yw4&list=PL5Q2soXY2Zi_uej3aY39YB5pfW4SJ7LlN

- Bachelor's course
  - 2nd semester at ETH Zurich
  - Rigorous introduction into "How Computers Work"
  - Digital Design/Logic
  - Computer Architecture
  - 10 FPGA Lab Assignments

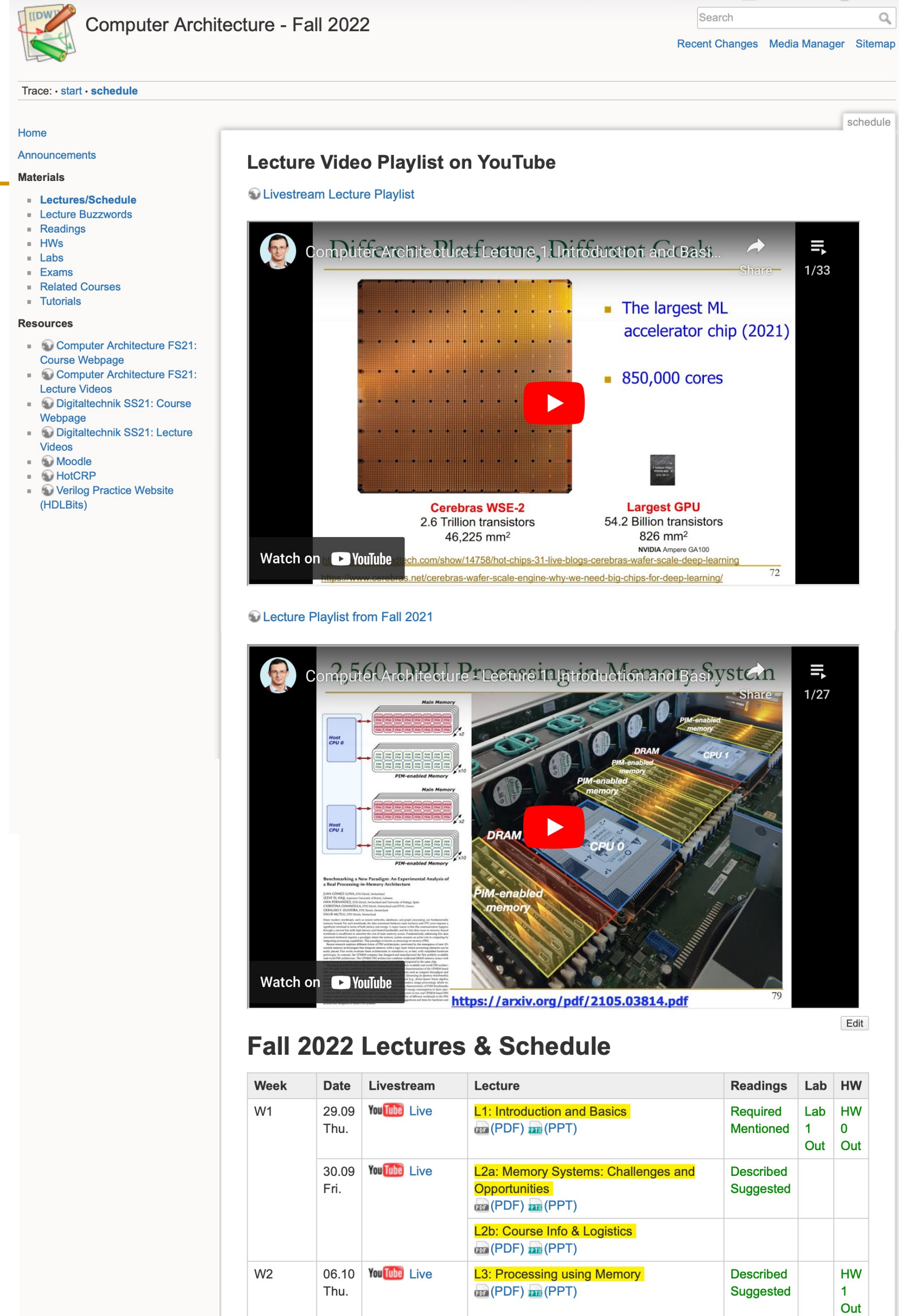


https://www.youtube.com/onurmutlulectures

Comp Arch (Fall 2022)

- **[Fall 2022 Edition:](https://safari.ethz.ch/architecture/fall2022/doku.php?id=schedule)**
  - https://safari.ethz.ch/architecture/fall2022/doku.php?id=schedule
- **[Fall 2021 Edition:](https://safari.ethz.ch/architecture/fall2021/doku.php?id=schedule)**
  - https://safari.ethz.ch/architecture/fall2021/doku.php?id=schedule
- **[Youtube Livestream (2022):](https://www.youtube.com/watch?v=4yfkM_5EFgo&list=PL5Q2soXY2Zi-Mnk1PxjEIG32HAGILkTOF)**
  - https://www.youtube.com/watch?v=4yfkM_5EFgo&list=PL5Q2soXY2Zi-Mnk1PxjEIG32HAGILkTOF
- **[Youtube Livestream (2021):](https://www.youtube.com/watch?v=4yfkM_5EFgo&list=PL5Q2soXY2Zi-Mnk1PxjEIG32HAGILkTOF)**
  - https://www.youtube.com/watch?v=4yfkM_5EFgo&list=PL5Q2soXY2Zi-Mnk1PxjEIG32HAGILkTOF

- Master's level course
  - Taken by Bachelor's/Masters/PhD students
  - Cutting-edge research topics + fundamentals in Computer Architecture
  - 5 Simulator-based Lab Assignments
  - Potential research exploration
  - Many research readings



RowHammer & DRAM Exploration (Fall 2022)

- **Fall 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=softmc
- **Spring 2022 Edition:**
  - https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=softmc
- **Youtube Livestream (Spring 2022):**
  - https://www.youtube.com/watch?v=r5QxuoJWttg&list=PL5Q2soXY2Zi_1trfCckr6PTN8WR72icUO
- Bachelor's course
  - Elective at ETH Zurich
  - Introduction to DRAM organization & operation
  - Tutorial on using FPGA-based infrastructure
  - Verilog & C++
  - Potential research exploration

Lecture Video Playlist on YouTube

Lecture Playlist

2022 Meetings/Schedule (Tentative)

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W0	23.02 Wed.	YouTube Video	P&S SoftMC Tutorial	SoftMC Tutorial Slides (PDF) (PPT)	
W1	08.03 Tue.	YouTube Video	M1: Logistics & Intro to DRAM and SoftMC (PDF) (PPT)	Required Materials Recommended Materials	HW0
W2	15.03 Tue.	YouTube Video	M2: Revisiting RowHammer (PDF) (PPT)	(Paper PDF)	
W3	22.03 Tue.	YouTube Video	M3: Uncovering in-DRAM TRR & TRRespass (PDF) (PPT)		
W4	29.03 Tue.	YouTube Video	M4: Deeper Look Into RowHammer's Sensitivities (PDF) (PPT)		
W5	05.04 Tue.	YouTube Video	M5: QUAC-TRNG (PDF) (PPT)		
W6	12.04 Tue.	YouTube Video	M6: PiDRAM (PDF) (PPT)		

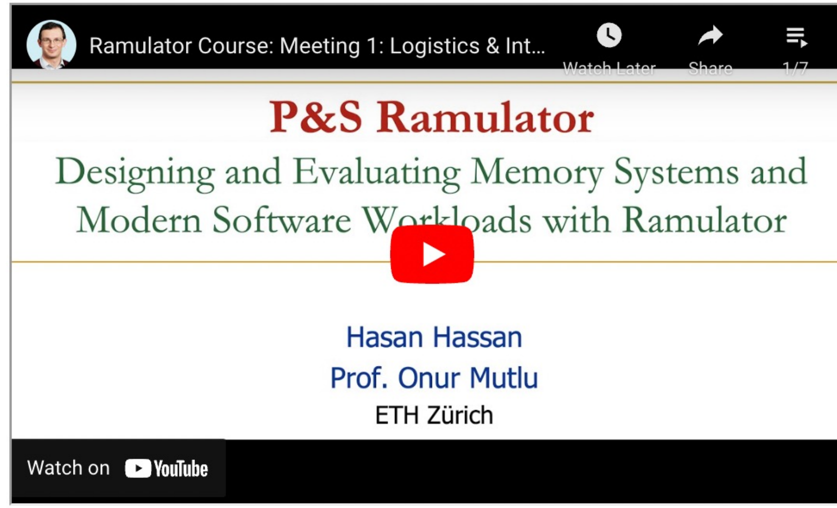
https://www.youtube.com/onurmutlulectures

Exploration of Emerging Memory Systems (Fall 2022)

- **[Fall 2022 Edition:](https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=ramulator)**
  - https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=ramulator
- **[Spring 2022 Edition:](https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=ramulator)**
  - https://safari.ethz.ch/projects_and_seminars/spring2022/doku.php?id=ramulator
- **Youtube Livestream (Spring 2022):**
  - https://www.youtube.com/watch?v=aM-IlXRQd3s&list=PL5Q2soXY2Zi_TlmLGw_Z8hBo2925ZApqV
- Bachelor's course
  - Elective at ETH Zurich
  - Introduction to memory system simulation
  - Tutorial on using Ramulator
  - C++
  - Potential research exploration

Lecture Video Playlist on YouTube

Lecture Playlist



2022 Meetings/Schedule (Tentative)

Week	Date	Livestream	Meeting	Learning Materials	Assignments
W1	09.03 Wed.	YouTube Video	M1: Logistics & Intro to Simulating Memory Systems Using Ramulator (PDF) (PPT)		HW0
W2	16.03 Fri.	YouTube Video	M2: Tutorial on Using Ramulator (PDF) (PPT)		
W3	25.02 Fri.	YouTube Video	M3: BlockHammer (PDF) (PPT)		
W4	01.04 Fri.	YouTube Video	M4: CLR-DRAM (PDF) (PPT)		
W5	08.04 Fri.	YouTube Video	M5: SIMDRAM (PDF) (PPT)		
W6	29.04 Fri.	YouTube Video	M6: DAMOV (PDF) (PPT)		
W7	06.05 Fri.	YouTube Video	M7: Syncron (PDF) (PPT)		

https://www.youtube.com/onurmutlulectures

An Early Proposal for Intelligent Controllers [IMW'13]

- Onur Mutlu,
  "Memory Scaling: A Systems Architecture Perspective"
  *Proceedings of the 5th International Memory Workshop (**IMW**)*, Monterey, CA, May 2013. Slides (pptx) (pdf)
  EETimes Reprint

Memory Scaling: A Systems Architecture Perspective

Onur Mutlu
Carnegie Mellon University
onur@cmu.edu
http://users.ece.cmu.edu/~omutlu/

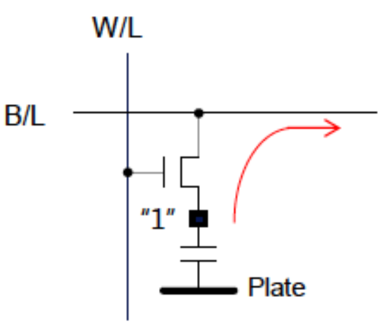
Industry Is Writing Papers About It, Too

DRAM Process Scaling Challenges

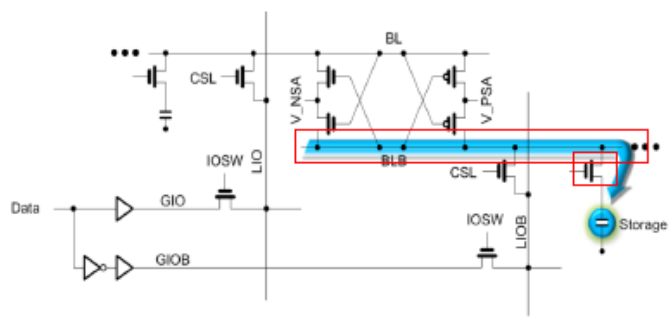
❖ **Refresh**
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance
- Leakage current of cell access transistors increasing

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

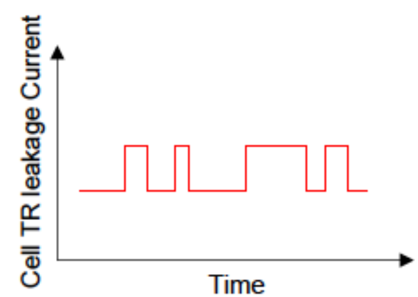
❖ **VRT**
- Occurring more frequently with cell capacitance decreasing



Refresh



tWR



VRT

Industry Is Writing Papers About It, Too

DRAM Process Scaling Challenges

❖ **Refresh**

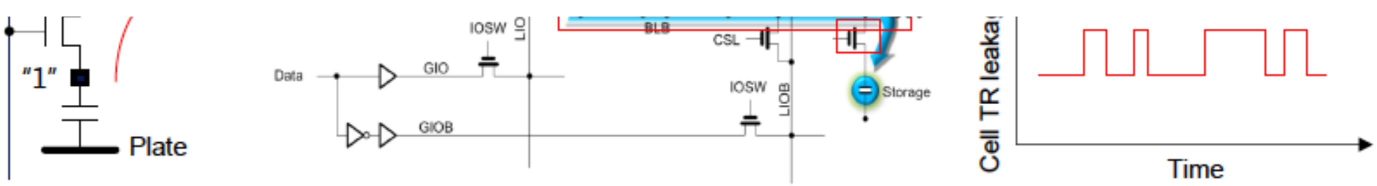
• Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance

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*



Refresh **tWR** **VRT**

The Memory Forum

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

ACM TOPLAS 1982

https://dl.acm.org/citation.cfm?id=357176

Funding Acknowledgments

- Alibaba, AMD, ASML, Google, Facebook, Hi-Silicon, HP Labs, Huawei, IBM, Intel, Microsoft, Nvidia, Oracle, Qualcomm, Rambus, Samsung, Seagate, VMware, Xilinx
- Microsoft Swiss JRC
- NSF
- NIH
- GSRC
- SRC
- CyLab
- EFCL

Thank you!

First RowHammer Analysis [ISCA 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] [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).
Selected to the ISCA-50 25-Year Retrospective Issue covering 1996-2020 in 2023 (Retrospective (pdf) Full Issue).

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

Yoongu Kim[1] Ross Daly* Jeremie Kim[1] Chris Fallin* Ji Hye Lee[1]
Donghyuk Lee[1] Chris Wilkerson[2] Konrad Lai Onur Mutlu[1]

[1]Carnegie Mellon University [2]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
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[§‡] Jeremie S. Kim[‡§]
[§]ETH Zürich [‡]Carnegie Mellon University

A RowHammer Survey: Recent Update

- Onur Mutlu, Ataberk Olgun, and A. Giray Yaglikci,
  "Fundamentally Understanding and Solving RowHammer"
  *Invited Special Session Paper at the 28th Asia and South Pacific Design Automation Conference (**ASP-DAC**)*, Tokyo, Japan, January 2023.
  [arXiv version]
  [Slides (pptx) (pdf)]
  [Talk Video (26 minutes)]

Fundamentally Understanding and Solving RowHammer

Onur Mutlu
onur.mutlu@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

Ataberk Olgun
ataberk.olgun@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

A. Giray Yağlıkcı
giray.yaglikci@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

https://arxiv.org/pdf/2211.07613.pdf

RowHammer & RowPress on HBM Chips

An Experimental Analysis of RowHammer in HBM2 DRAM Chips

Ataberk Olgun Majd Osseiran

A. Giray Yağlıkçı Yahya Can Tuğrul Haocong Luo Steve Rhyner

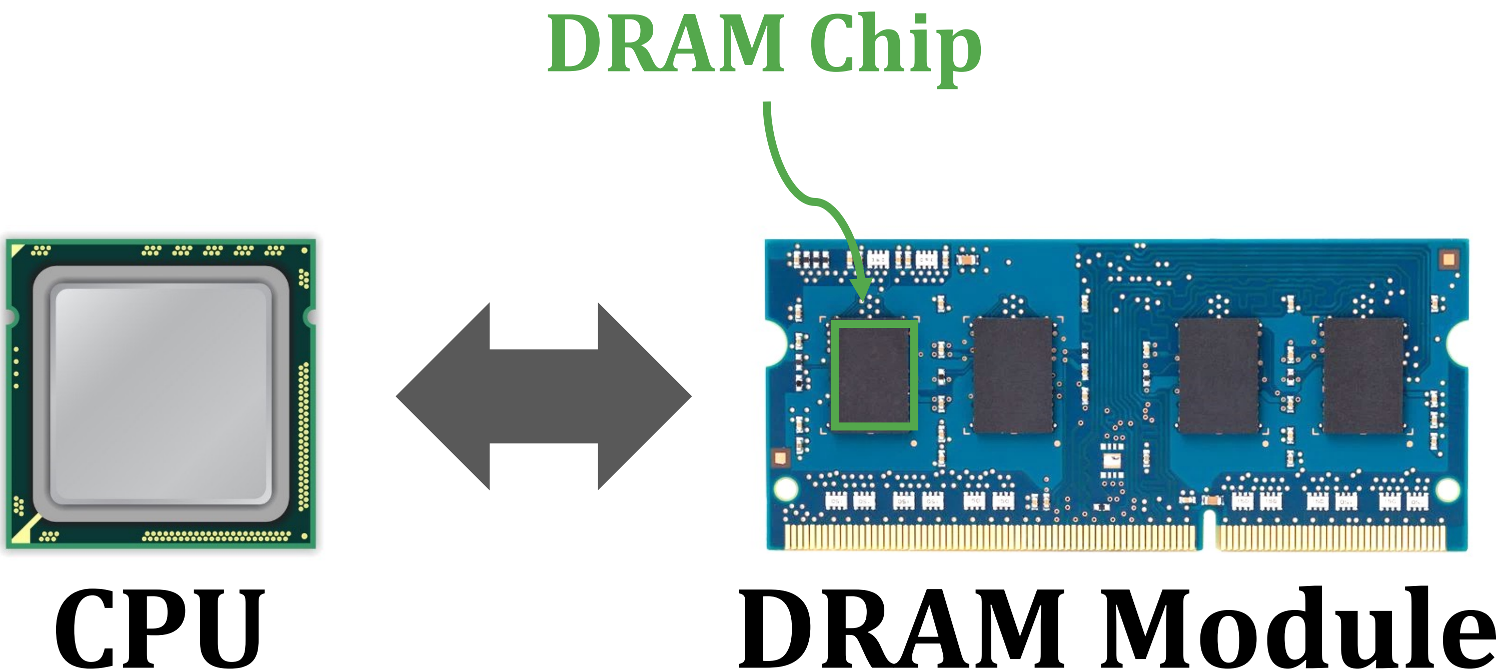
Behzad Salami Juan Gomez Luna Onur Mutlu

SAFARI

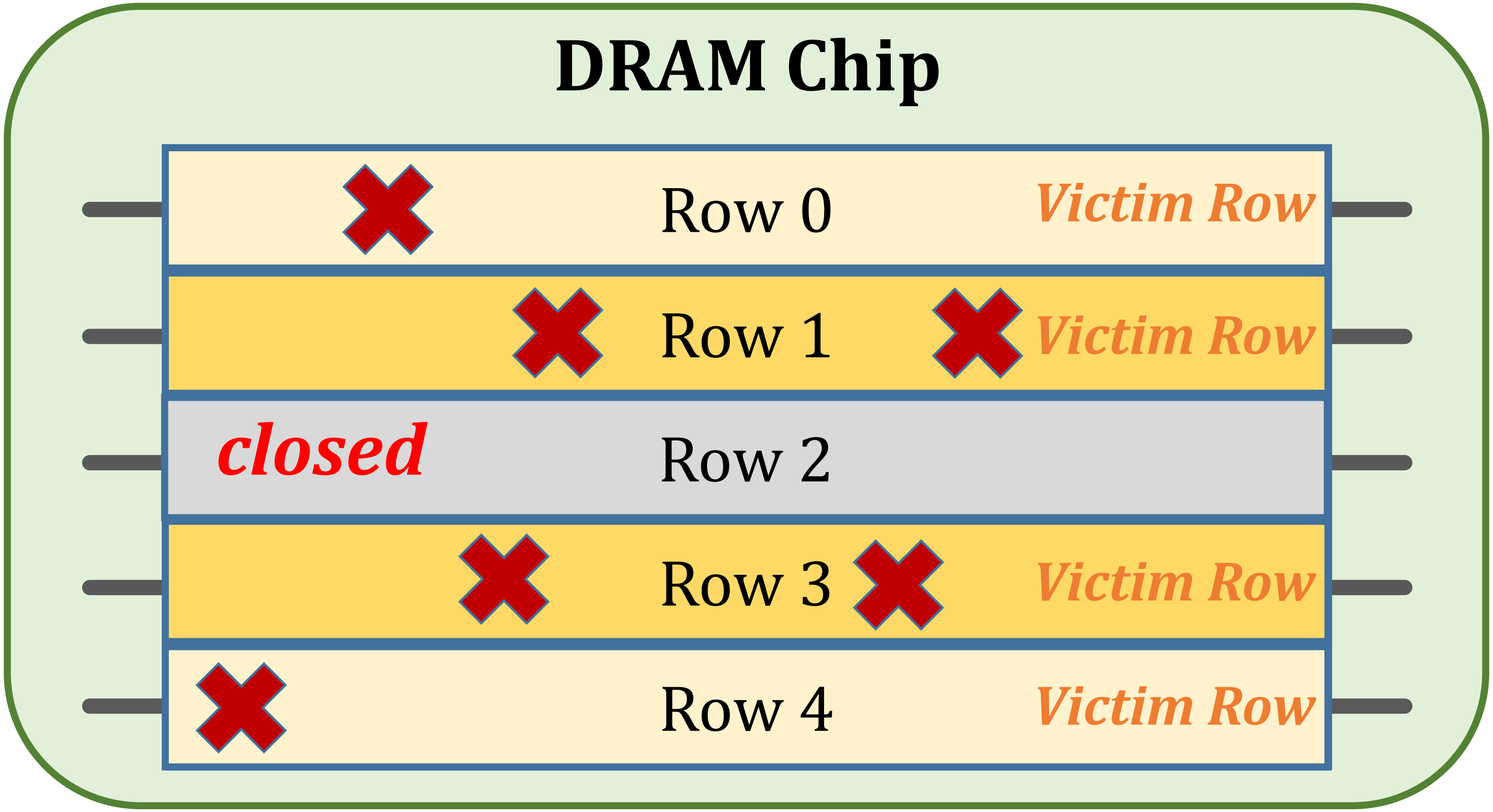
ETH zürich

AMERICAN UNIVERSITY OF BEIRUT

The RowHammer Vulnerability (I)

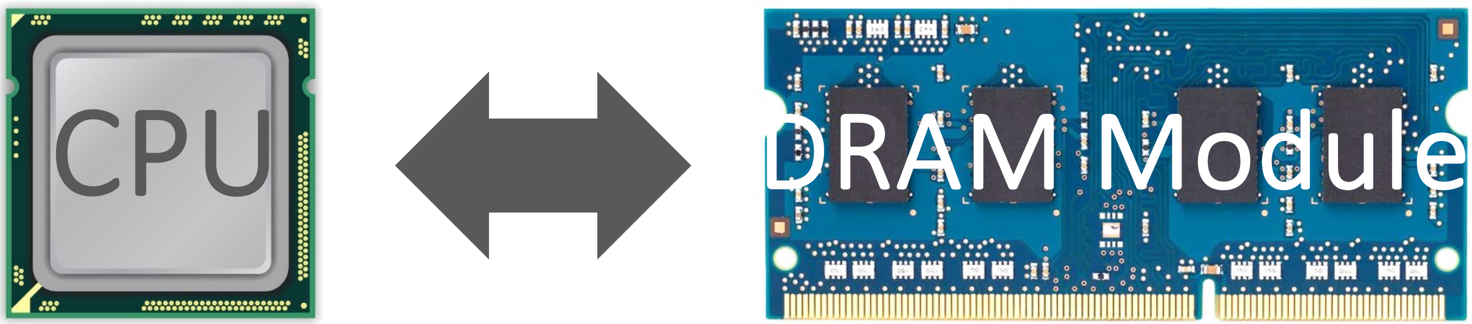


The RowHammer Vulnerability (II)

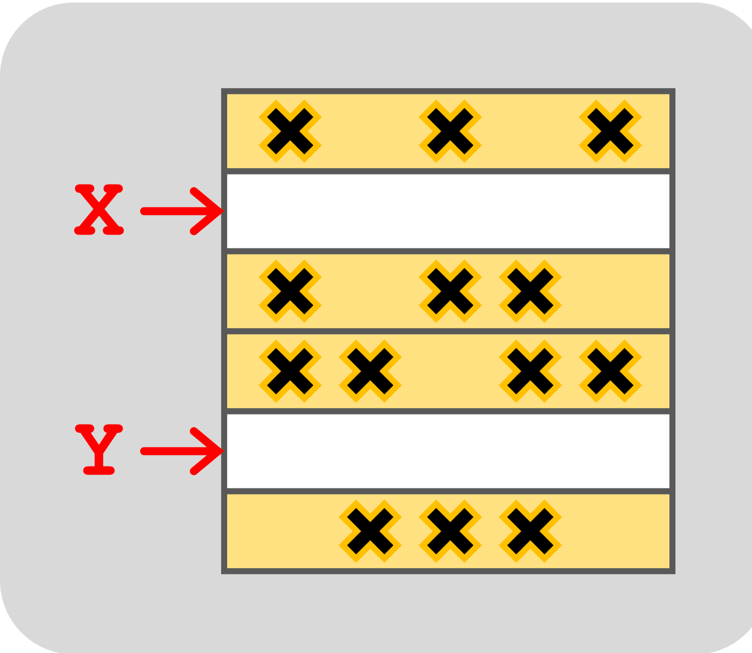


Repeatedly **opening** (activating) and **closing** (precharging) a DRAM row causes **RowHammer bit flips** in nearby rows

A Simple Program Can Induce Bitflips




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



One Can Take Over a System

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

Abstract. *$\boxed{\text{Memory isolation}}$ 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

Monday, March 9, 2015

Exploiting the DRAM rowhammer bug to gain kernel privileges

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

Most DRAM Modules Are Vulnerable (2020)

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

All tested DRAM types are susceptible to RowHammer bitflips

What about High Bandwidth Memory (HBM)?

Kim et al., "[Revisiting RowHammer: An Experimental Analysis of Modern DRAM Devices and Mitigation Techniques](#)," in ISCA, 2020.

Executive Summary

Motivation: HBM chips have new architectural characteristics (e.g., 3D-stacked dies) that might affect the RowHammer vulnerability in various ways

Understanding RowHammer enables designing effective and efficient solutions

Problem: No prior study demonstrates the RowHammer vulnerability in HBM

Goal: Experimentally analyze how vulnerable HBM DRAM chips are to RowHammer

Experimental Study: Detailed experimental characterization of RowHammer in a modern HBM2 DRAM chip. Our study provides two main findings:

1. Spatial variation of RowHammer vulnerability

- Different channels in a 3D-stacked HBM chip exhibit different RowHammer vulnerability
- DRAM rows near the end of a DRAM bank are more RowHammer resilient

2. On-DRAM-die RowHammer mitigations

- A modern HBM chip implements undisclosed on-DRAM-die RowHammer mitigation
- The mitigation refreshes a victim row after every 17 periodic refresh operations (e.g., similar to DDR4 chips)

Outline

1. HBM DRAM Organization & Operation

2. DRAM Cell Leakage & RowHammer

3. HBM DRAM Testing Methodology

4. RowHammer Spatial Variation Analysis

5. On-die RowHammer Mitigation Analysis

6. Conclusion

Outline

1. HBM DRAM Organization & Operation

2. DRAM Cell Leakage & RowHammer

3. HBM DRAM Testing Methodology

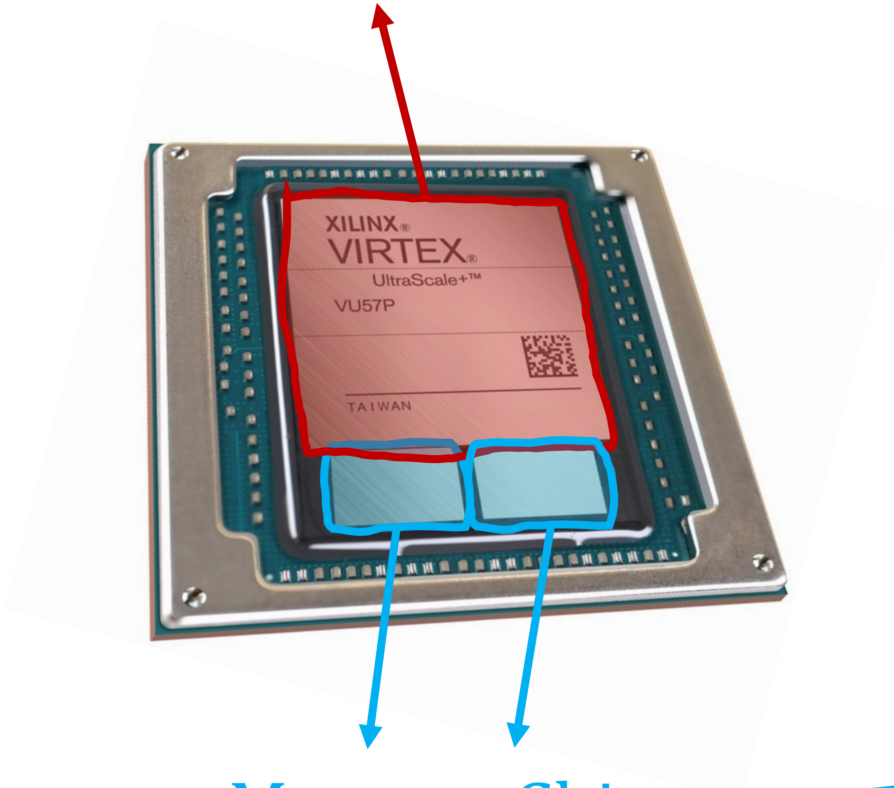
4. RowHammer Spatial Variation Analysis

5. On-die RowHammer Mitigation Analysis

6. Conclusion

System with High Bandwidth Memory

Compute Chip (e.g., FPGA)

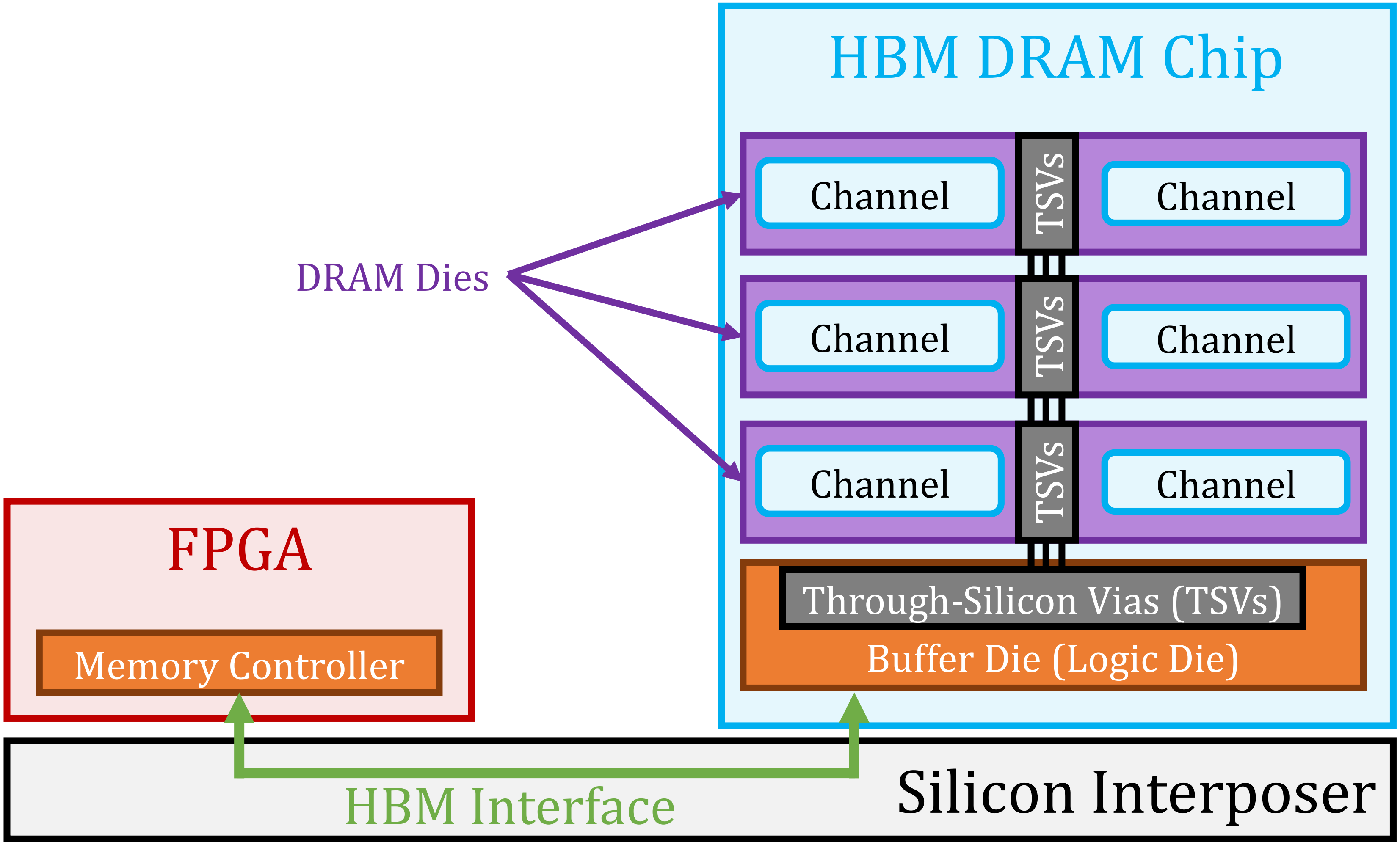


Inside one **package**

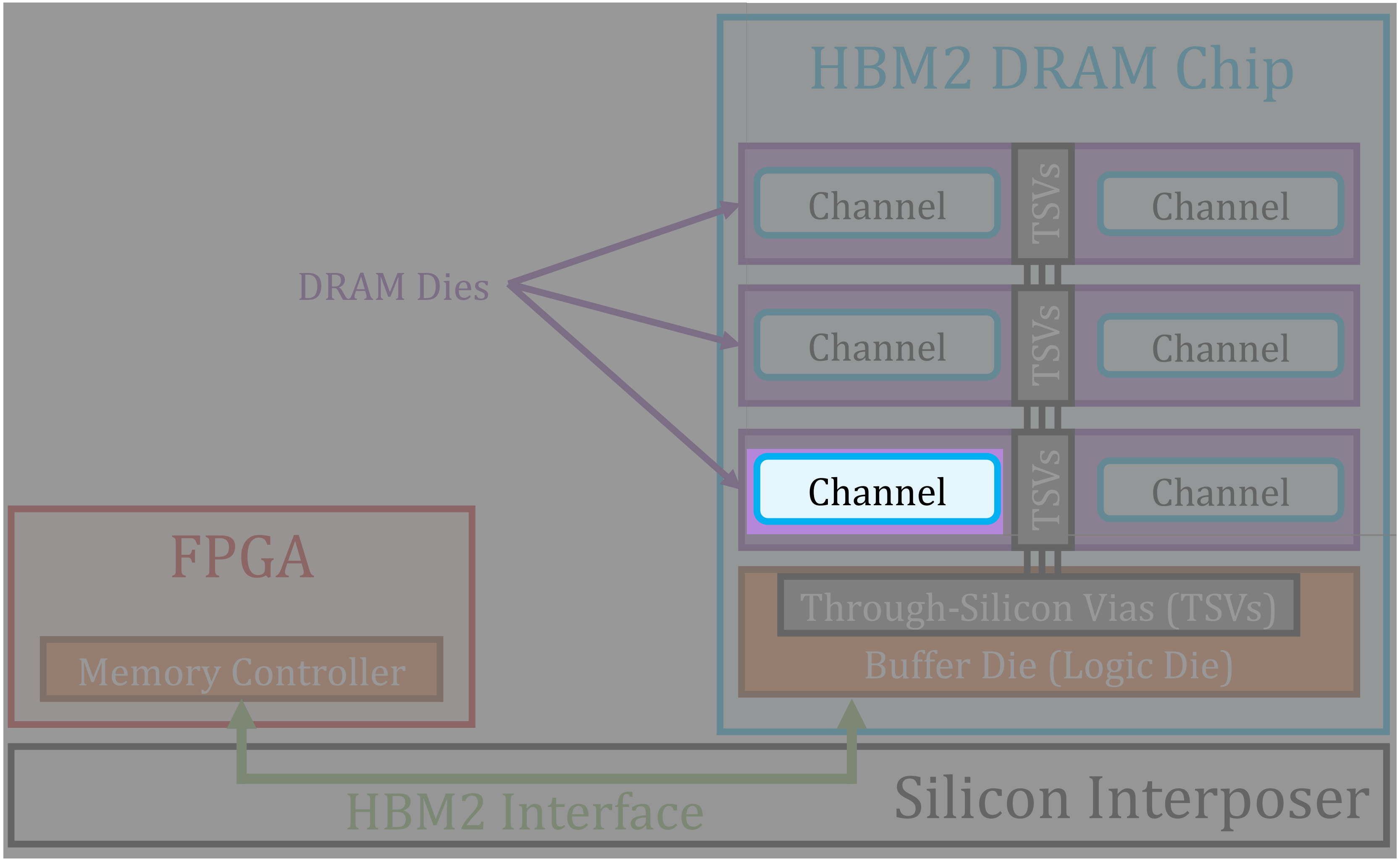
Memory Chip
(e.g., HBM DRAM)

https://www.hpcwire.com/off-the-wire/xilinx-adds-high-bandwidth-memory-capabilities-to-its-virtex-ultrascale-portfolio/

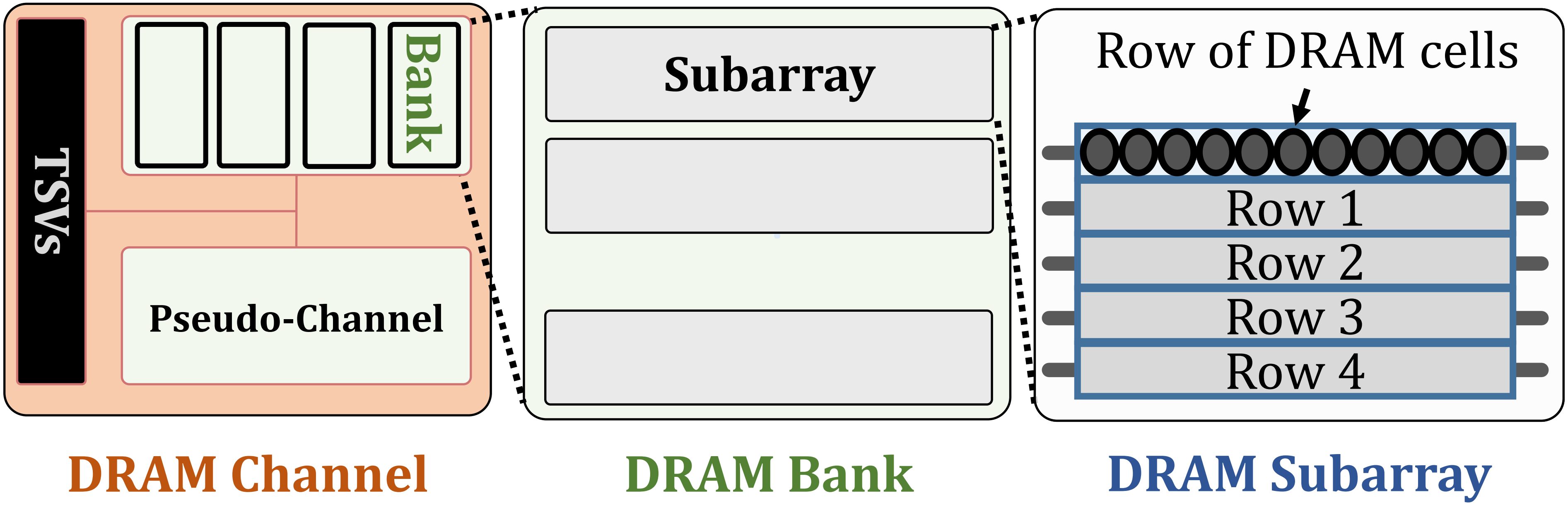
HBM DRAM Organization (I)



HBM DRAM Organization (I)



HBM DRAM Organization (II)



Outline

1. HBM DRAM Organization & Operation

2. DRAM Cell Leakage & RowHammer

3. HBM DRAM Testing Methodology

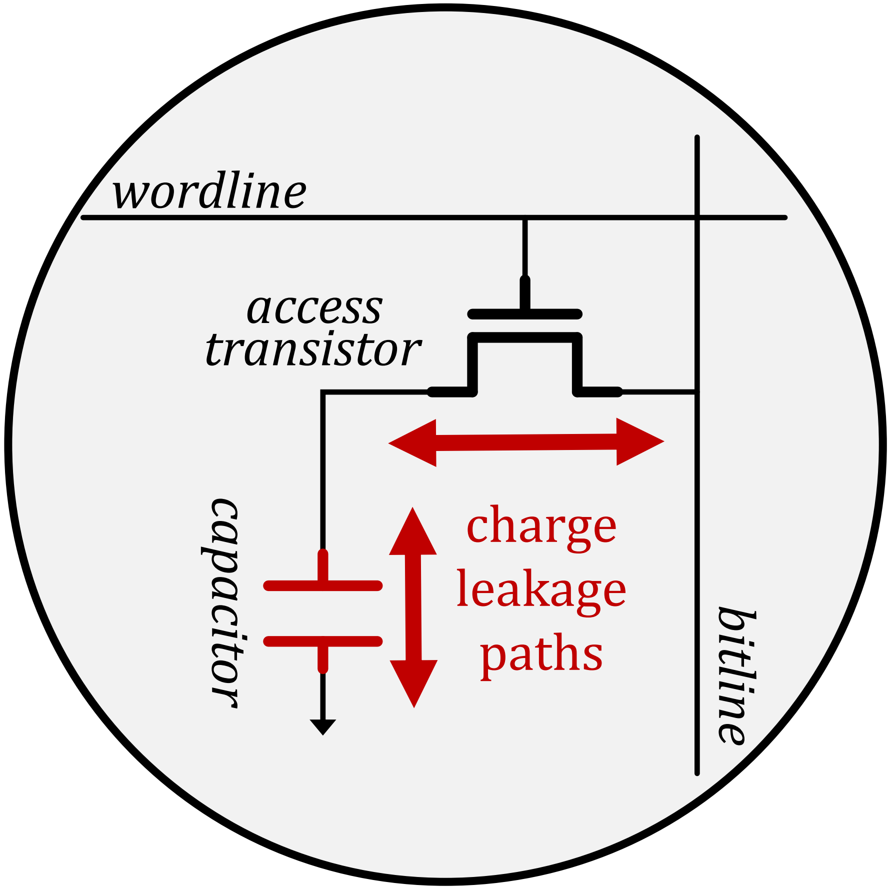
4. RowHammer Spatial Variation Analysis

5. On-die RowHammer Mitigation Analysis

6. Conclusion

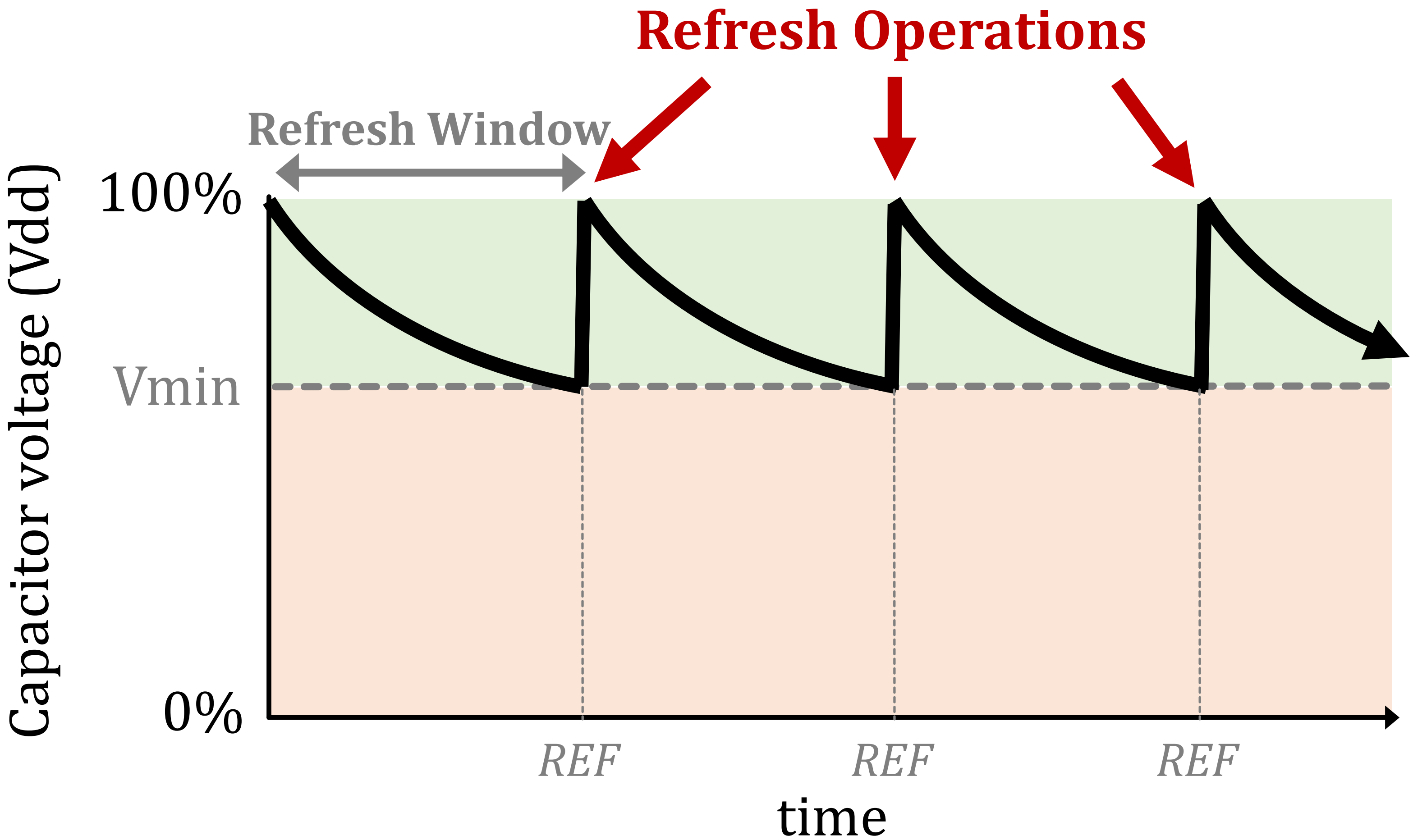
DRAM Cell Leakage

Each cell encodes information in **leaky** capacitors



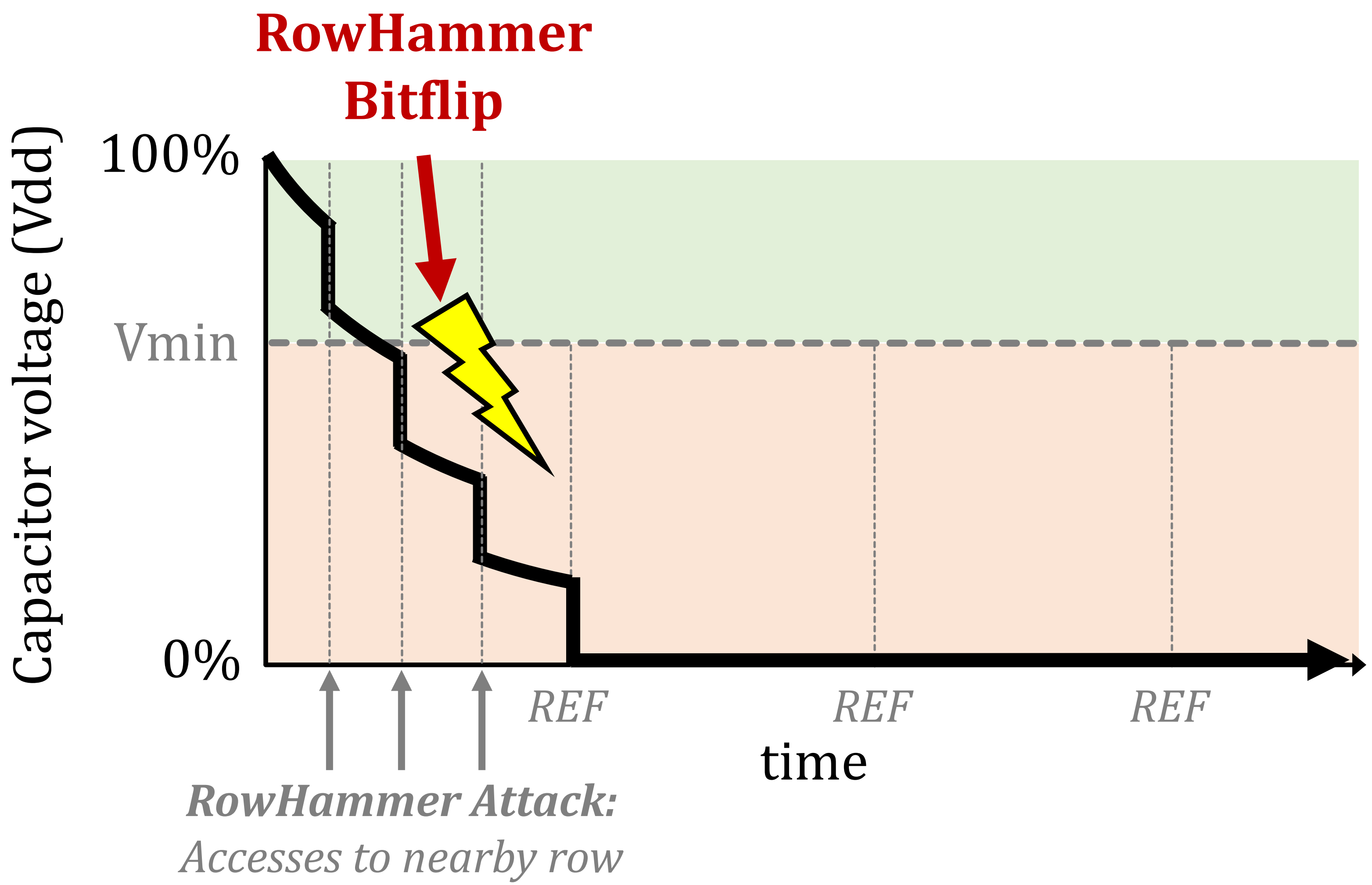
Stored data is **corrupted** if too much charge leaks (i.e., the capacitor voltage degrades too much)

DRAM Refresh



Periodic **refresh operations** preserve stored data

RowHammer Bitflips



Problem & Goal

Problem

No prior study demonstrates
the RowHammer vulnerability in high bandwidth memory

Our Goal

Experimentally analyze how vulnerable
real high bandwidth memory chips are to RowHammer

Outline

1. HBM DRAM Organization & Operation

2. DRAM Cell Leakage & RowHammer

3. HBM DRAM Testing Methodology

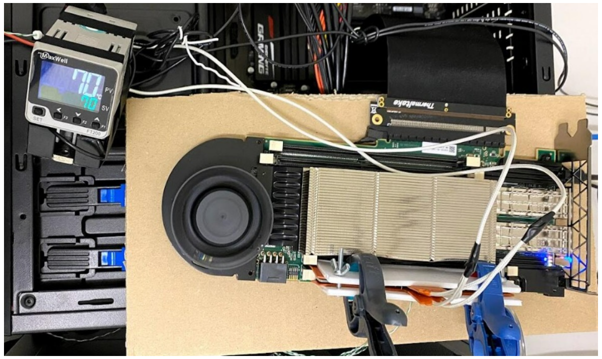
4. RowHammer Spatial Variation Analysis

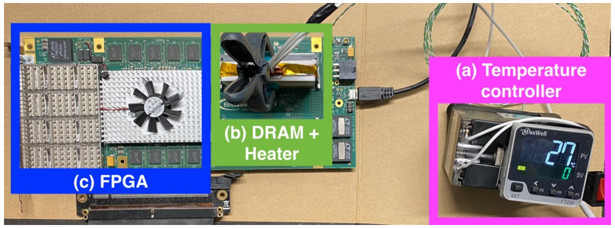
5. On-die RowHammer Mitigation Analysis

6. Conclusion

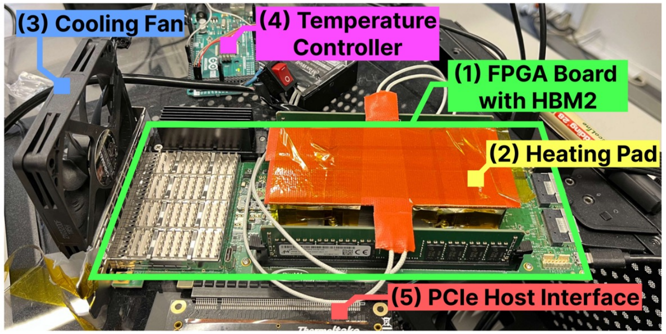
DRAM Testing Infrastructure

DRAM Bender DDR3/4 Testing Infrastructure

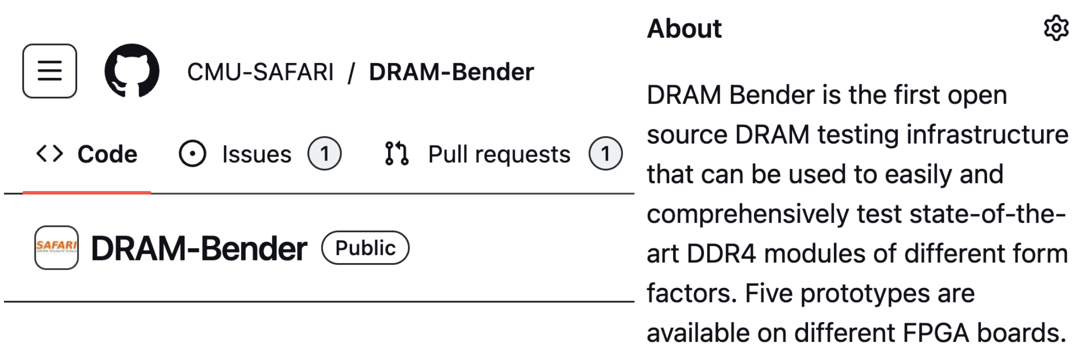




Adapt to work with HBM2 chips



https://github.com/CMU-SAFARI/DRAM-Bender


CMU-SAFARI / DRAM-Bender

Code Issues 1 Pull requests 1

DRAM-Bender Public

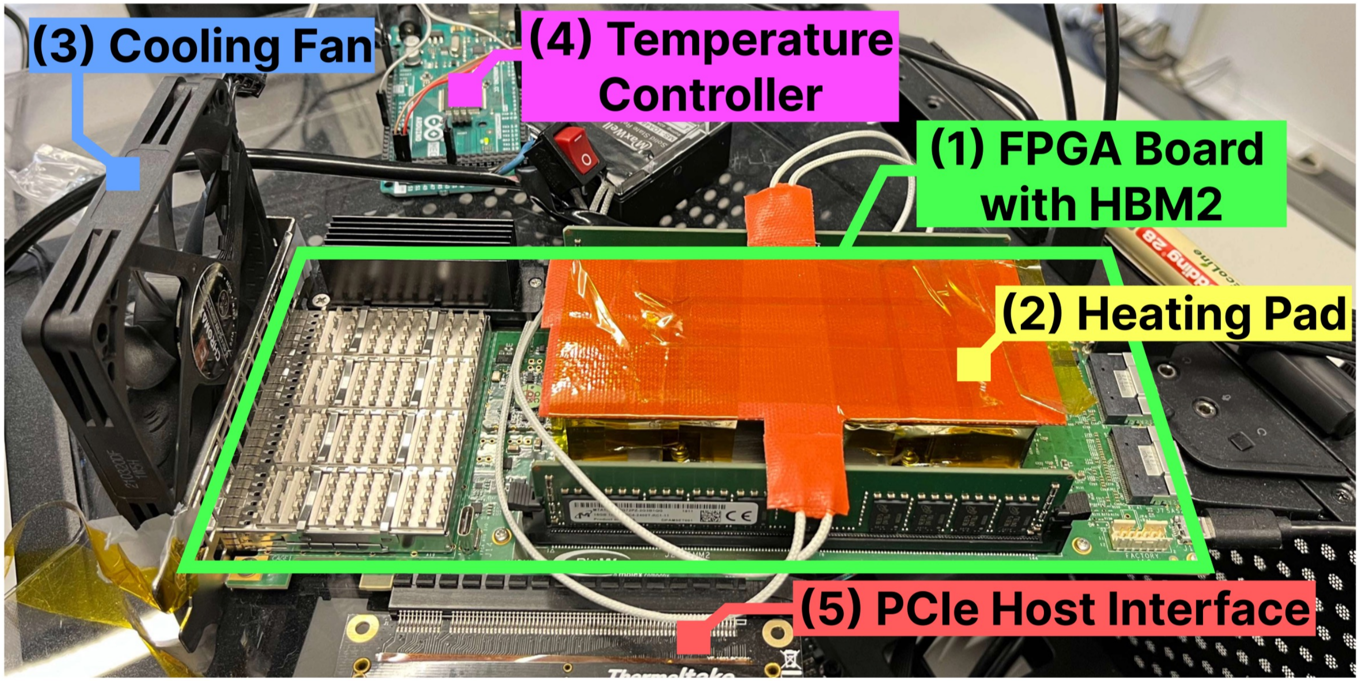
About

DRAM Bender is the first open source DRAM testing infrastructure that can be used to easily and comprehensively test state-of-the-art DDR4 modules of different form factors. Five prototypes are available on different FPGA boards.

Olgun et al., "DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips," in TCAD, 2023.

DRAM Testing Infrastructure

FPGA-based HBM2 Testing Setup (Bittware XUPVVH)



Fine-grained control over **DRAM commands**,
timing parameters (±1.66ns)

Olgun et al., "DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips," in TCAD, 2023.

RowHammer Testing Methodology (I)

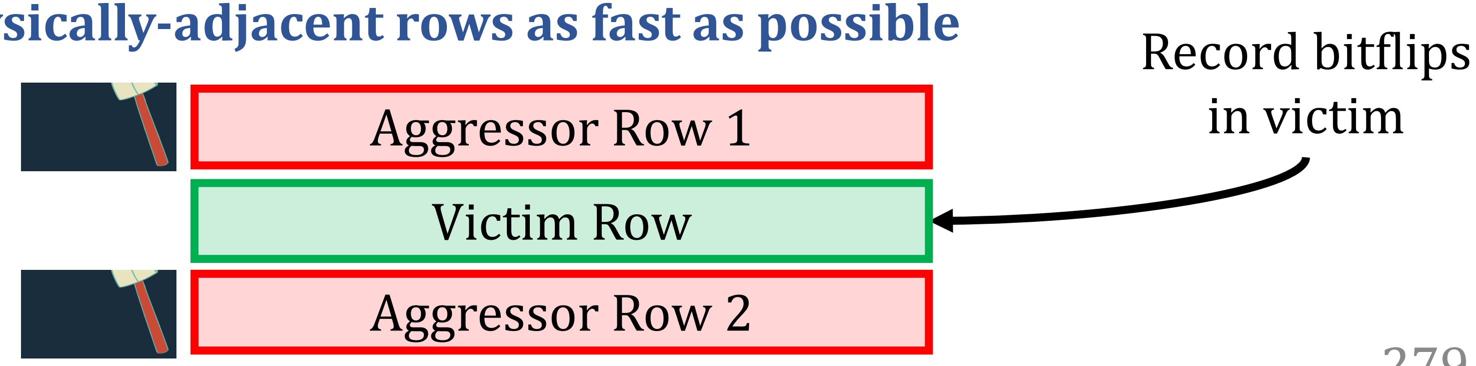
To characterize our DRAM chips at **worst-case** conditions:

1. Prevent sources of interference during core test loop

- **No DRAM refresh**: to avoid refreshing victim row
- **No RowHammer mitigation mechanisms**: to observe circuit-level effects
- Test for **less than a refresh window (32ms)** to avoid retention failures
- **Repeat tests** for five times

2. Worst-case RowHammer access sequence

- We use **worst-case** RowHammer access sequence based on prior works' observations
- Double-sided RowHammer: **repeatedly access the two physically-adjacent rows as fast as possible**



RowHammer Testing Methodology (II)

- Tested HBM2 chip's organization:
  - 8 channels
  - 2 pseudo-channels
  - 16 banks
  - 16384 rows (1 KiB each)

Xilinx FPGA
with HBM2 DRAM chips

- Test all channels, pseudo-channels, banks

- Test first, middle, and last 3K rows in a bank
  - 9K out of 16K (more than half)

- Keep HBM2 chip temperature at 85°C

Metrics

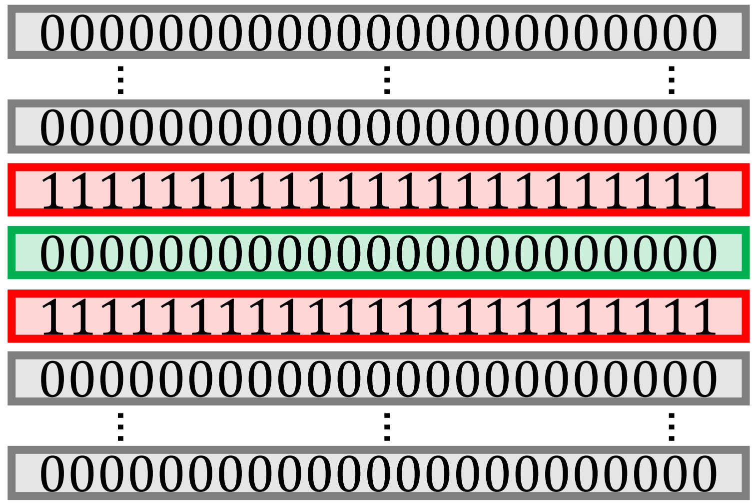
1. **Bit error rate (BER):**
   The fraction of DRAM cells in a row
   that experience a bitflip after 512K activations

   Higher is worse

2. **Hammer Count for the First Bitflip (HC_{first}):**
   Aggressor row activation count
   to cause the first bitflip in the victim row

   Lower is worse

Tested Data Patterns



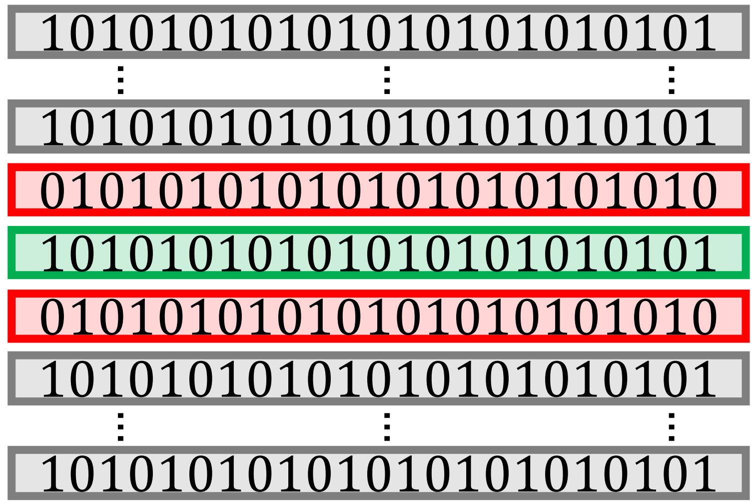
Row Addresses	***Rowstripe0***	***Rowstripe1***	***Checkered0***	***Checkered1***
Victim (V)	0x00	0xFF	0x55	0xAA
Aggressors ($V \pm 1$)	0xFF	0x00	0xAA	0x55
$V \pm [2:8]$	0x00	0xFF	0x55	0xAA

Tested Data Patterns

10101010101010101010101
⋮
10101010101010101010101
01010101010101010101010
10101010101010101010101
01010101010101010101010
10101010101010101010101
⋮
10101010101010101010101

Row Addresses	*Rowstripe0*	*Rowstripe1*	***Checkered0***	*Checkered1*
Victim (V)	0x00	0xFF	0x55	0xAA
Aggressors ($V \pm 1$)	0xFF	0x00	0xAA	0x55
$V \pm [2{:}8]$	0x00	0xFF	0x55	0xAA

Tested Data Patterns



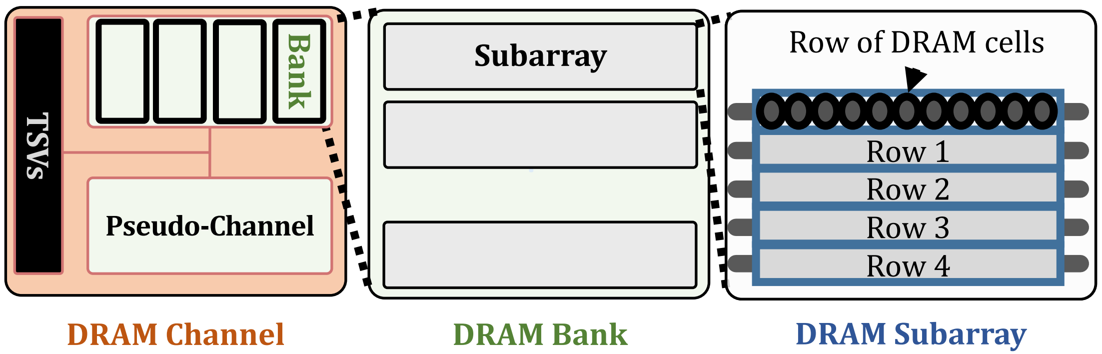
Row Addresses	***Rowstripe0***	***Rowstripe1***	***Checkered0***	***Checkered1***
Victim (V)	0x00	0xFF	0x55	0xAA
Aggressors ($V \pm 1$)	0xFF	0x00	0xAA	0x55
$V \pm [2:8]$	0x00	0xFF	0x55	0xAA

Worst-case data pattern (WCDP) of a row: Causes smallest HC_{first} for a row

Two Main Analyses

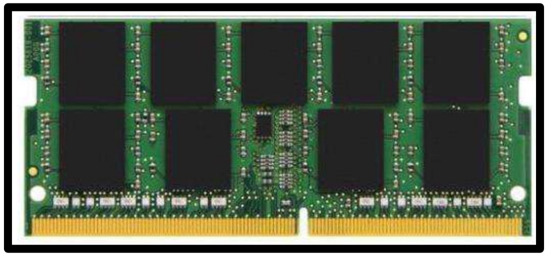
1. Spatial variation of RowHammer vulnerability

How does the RowHammer vulnerability change across channels, pseudo-channels, banks, rows in HBM?



2. On-DRAM-die RowHammer mitigations

Do real HBM chips implement undisclosed RowHammer mitigations resembling those that exist in DDR4?



Outline

1. HBM DRAM Organization & Operation

2. DRAM Cell Leakage & RowHammer

3. HBM DRAM Testing Methodology

4. RowHammer Spatial Variation Analysis

5. On-die RowHammer Mitigation Analysis

6. Conclusion

Key Takeaways from Spatial Variation Analysis

Takeaway 1

Different 3D-stacked HBM2 channels exhibit different RowHammer vulnerability

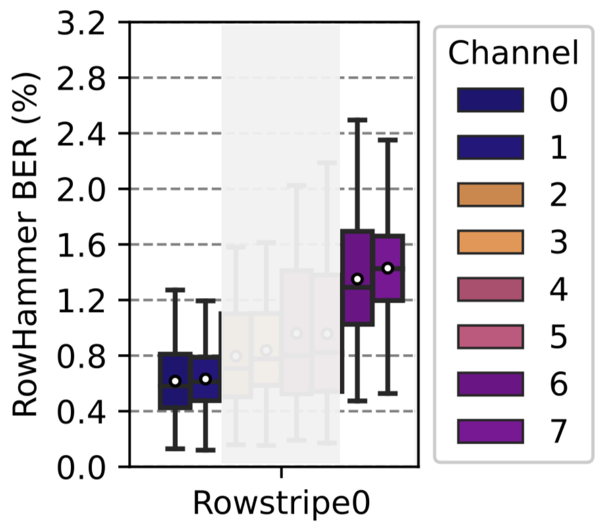
Takeaway 2

DRAM rows near the end of a DRAM bank
experience smaller bit error rate (BER) than others

Takeaway 3

Activation count needed to induce the first RowHammer bitflip (HC_{first})
changes with the data pattern and the physical location of the DRAM row

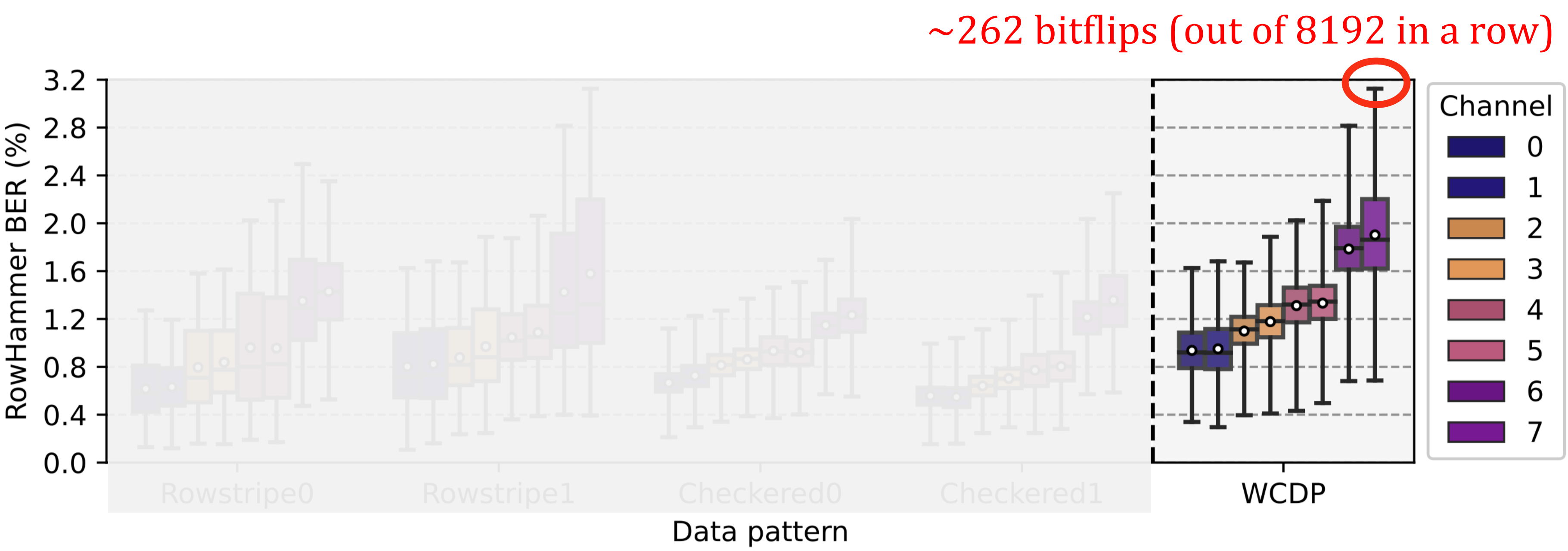
Spatial Distribution of BER (I)



There are bitflips in every tested DRAM row across all tested HBM2 channels

BER varies across channels: groups of two channels have different BERs

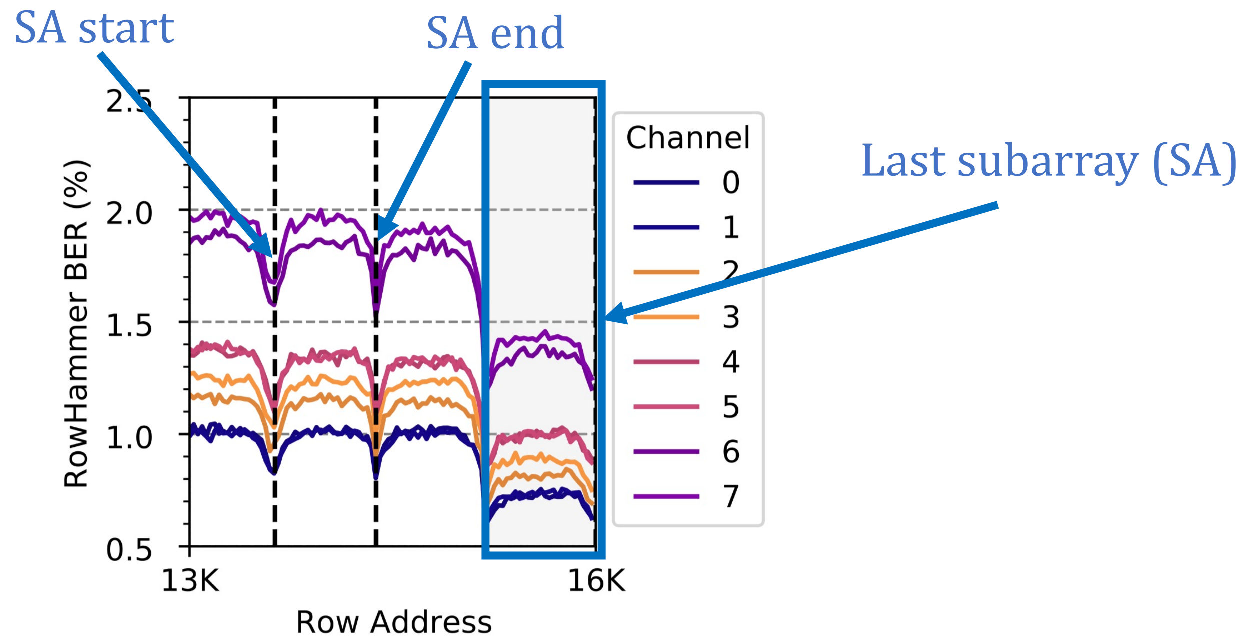
Spatial Distribution of BER (I)



The data pattern affects the BER distribution

Up to ~262 bitflips in a row of 8K bits
with 512K aggressor row activations

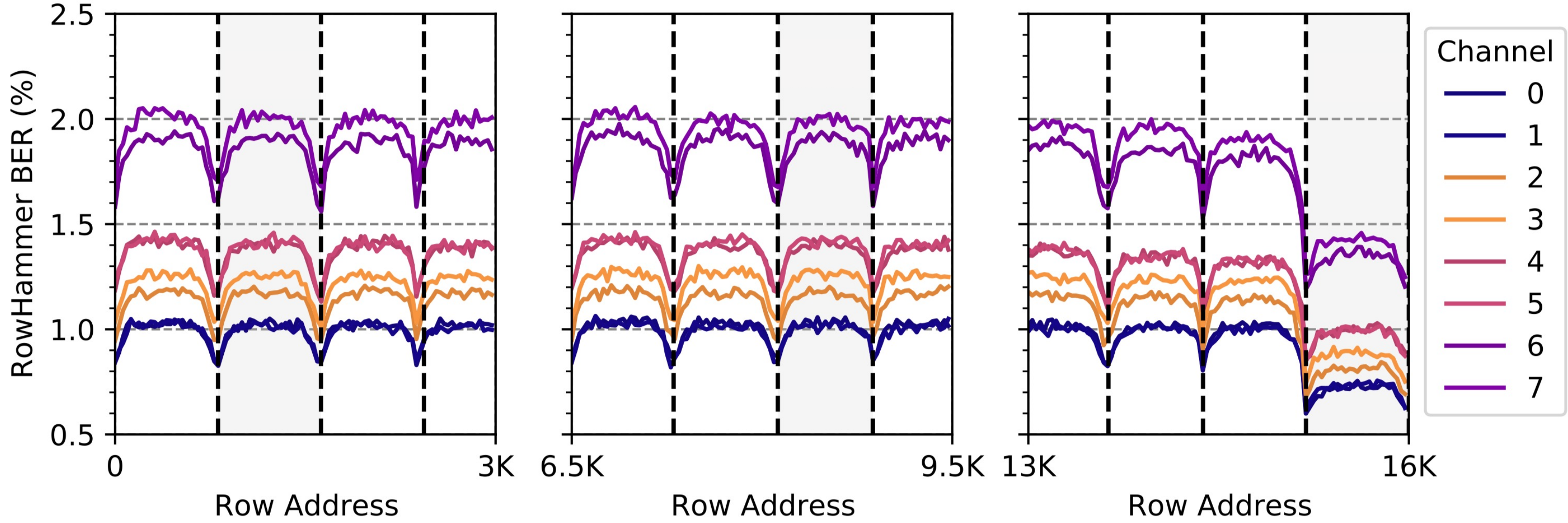
Spatial Distribution of BER (II)



BER is substantially smaller in the last subarray (i.e., last 832 rows)

BER periodically increases and decreases across rows:
BER is higher in the middle of a subarray

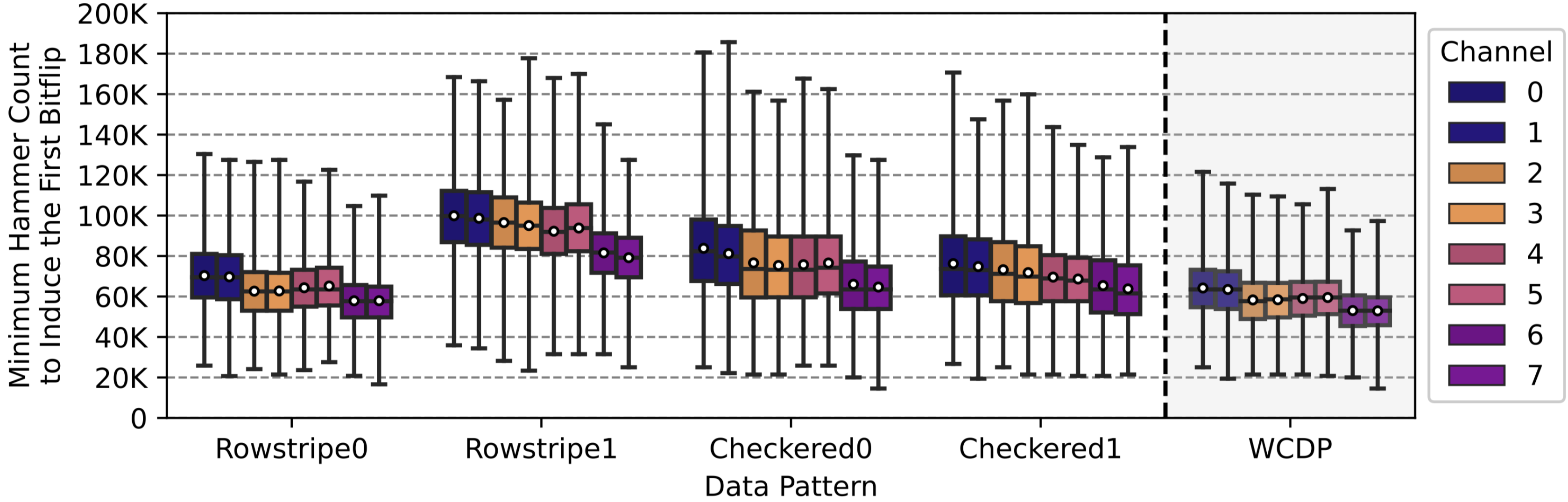
Spatial Distribution of BER (II)



BER is substantially smaller in the last subarray (i.e., last 832 rows)

BER periodically increases and decreases across rows:
BER is higher in the middle of a subarray

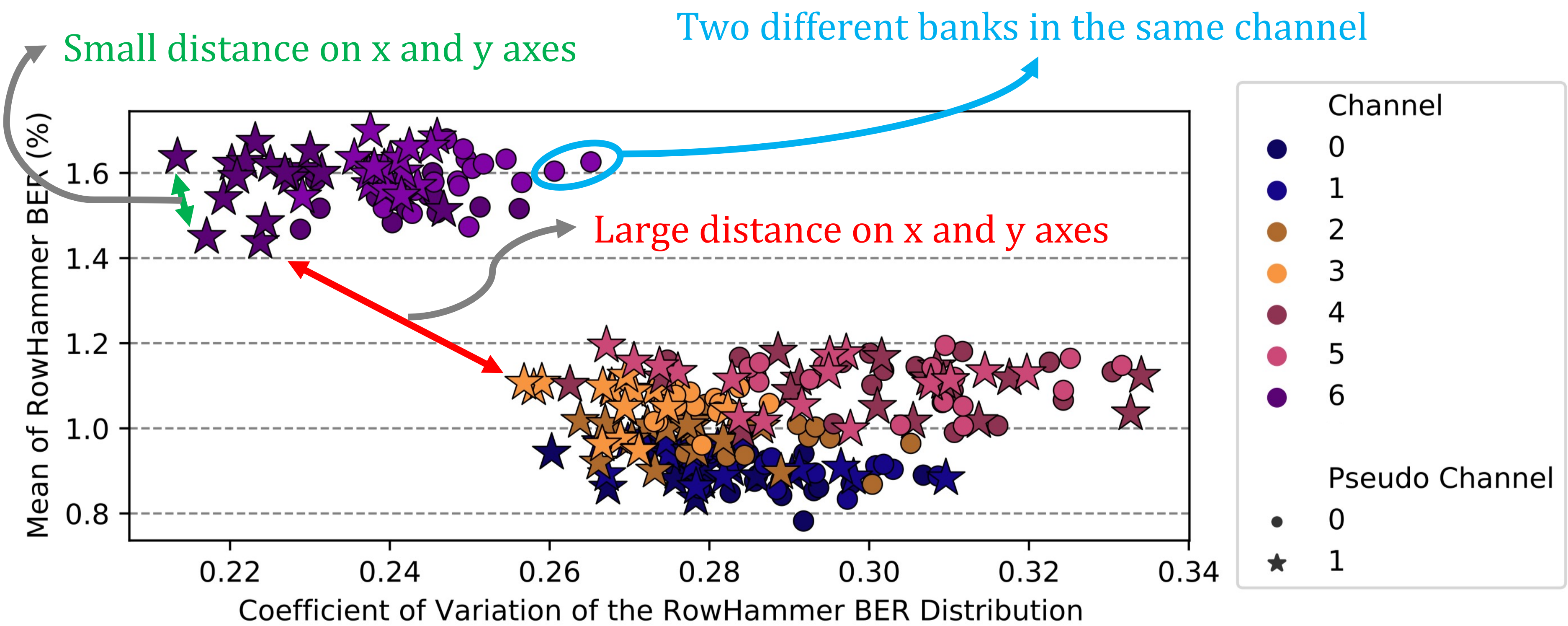
Spatial Distribution of HC_{first}



HC_{first} is as low as 14531 across all tested rows/channels:
Only ~1.3 ms to induce a RowHammer bitflip

HC_{first} distribution heavily depends on the data pattern

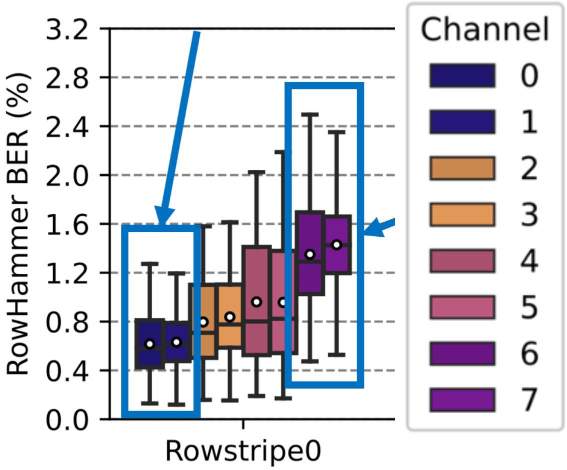
Variation in Bit Error Rate



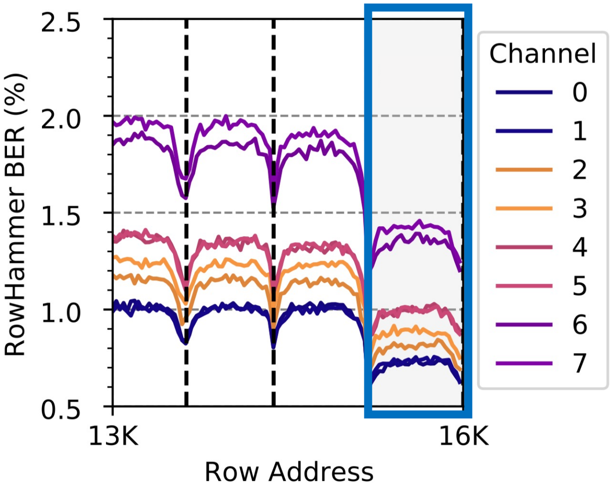
Banks in the same channel have similar variation in BER

Hypotheses from Characterization

1. Similar BER & HC_{first} within groups of two channels suggests these channels share DRAM dies



2. RowHammer BER changes with the row's proximity to sense amplifiers and bank I/O



Implications on Attacks and Mitigations

Key Observation: RowHammer BER and HC_{first} vary across channels

Two implications for RowHammer attacks and mitigations

A RowHammer attack can use the most-RH-vulnerable HBM2 channel to prepare for and perform the attack faster

A RowHammer mitigation can allocate fewer resources for RowHammer-resilient channels and more efficiently prevent RowHammer bitflips

Outline

1. HBM DRAM Organization & Operation

2. DRAM Cell Leakage & RowHammer

3. HBM DRAM Testing Methodology

4. RowHammer Spatial Variation Analysis

5. On-die RowHammer Mitigation Analysis

6. Conclusion

Key Takeaways from on-die Mitigation Analysis

Takeaway 1

A modern HBM2 chip implements an undisclosed on-DRAM-die RowHammer mitigation

Takeaway 2

This mitigation resembles the one in DDR4 chips from one major manufacturer as shown in prior work

Hassan et al., "Uncovering In-DRAM RowHammer Protection Mechanisms: A New Methodology, Custom RowHammer Patterns, and Implications," in MICRO, 2021.

On-Die RowHammer Mitigation Analysis (I)

HBM2 standard defines a "Target Row Refresh (TRR)-mode"

- Memory controller and DRAM cooperate to prevent RH bitflips

Real DDR4 chips implement on-die mitigation mechanisms

- Memory-controller-transparent, hidden behind periodic REF

Does a similar hidden mitigation mechanism exist in HBM2?

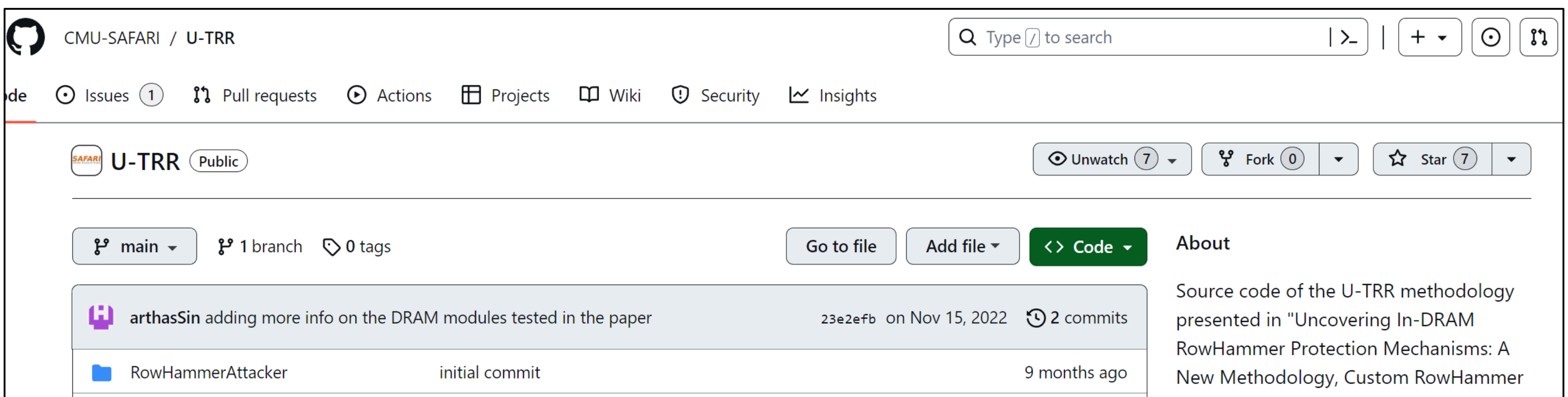
On-Die RowHammer Mitigation Analysis (II)

Hassan et al., "Uncovering In-DRAM RowHammer Protection Mechanisms: A New Methodology, Custom RowHammer Patterns, and Implications," in MICRO, 2021.

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

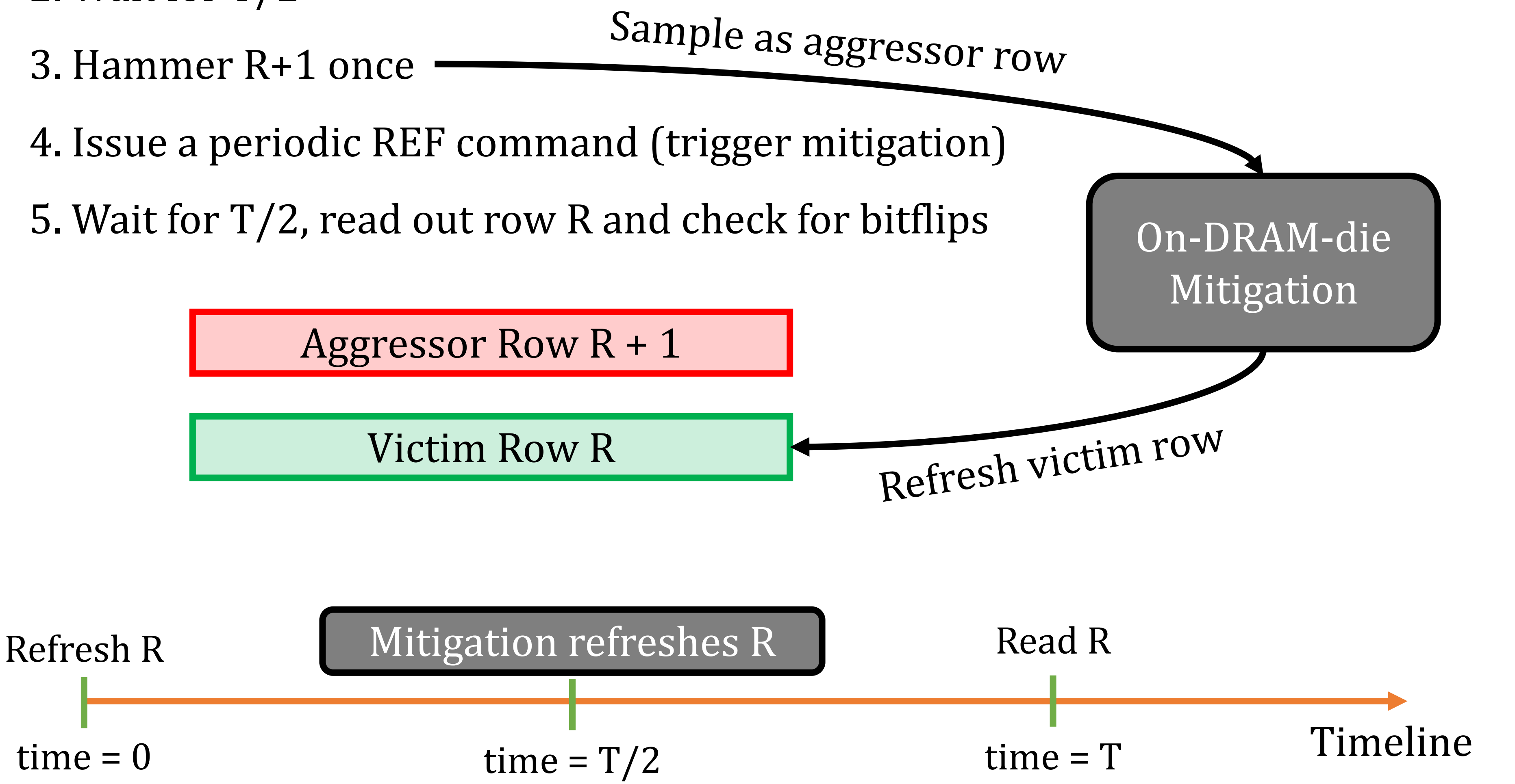
Hasan Hassan[†] Yahya Can Tuğrul[†‡] Jeremie S. Kim[†] Victor van der Veen[σ]
Kaveh Razavi[†] Onur Mutlu[†]
[†]*ETH Zürich* [‡]*TOBB University of Economics & Technology* [σ]*Qualcomm Technologies Inc.*

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



Experimental Methodology

1. Identify a row (R) with **T** retention time
2. Wait for T/2
3. Hammer R+1 once
4. Issue a periodic REF command (trigger mitigation)
5. Wait for T/2, read out row R and check for bitflips



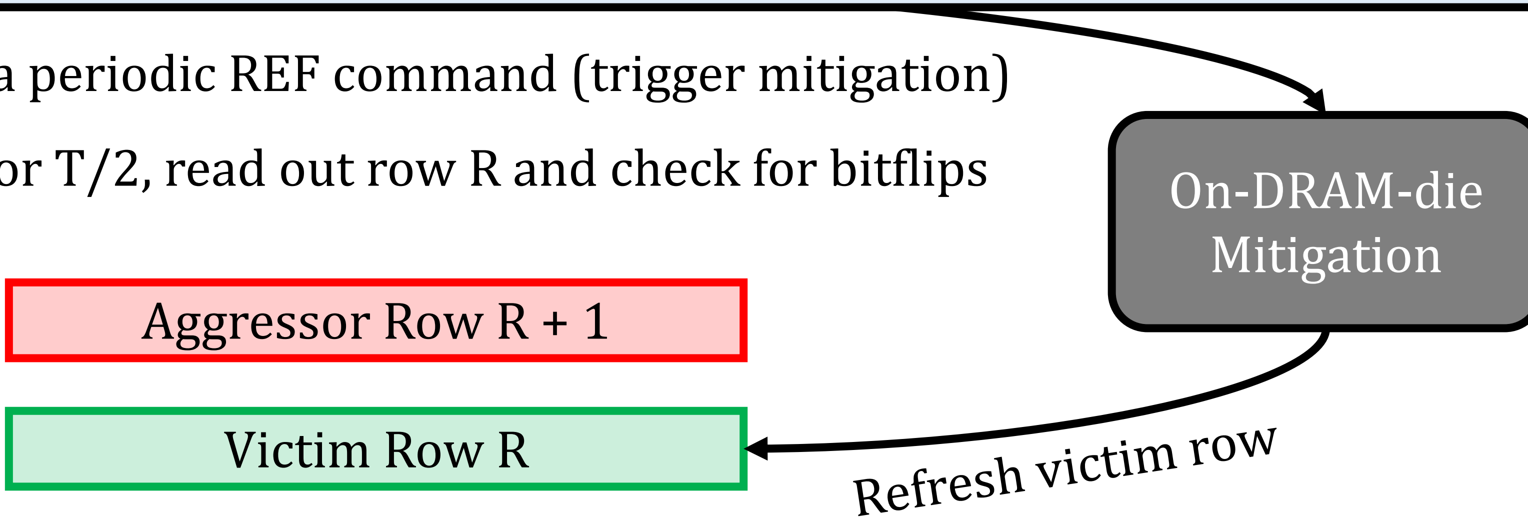
Experimental Methodology

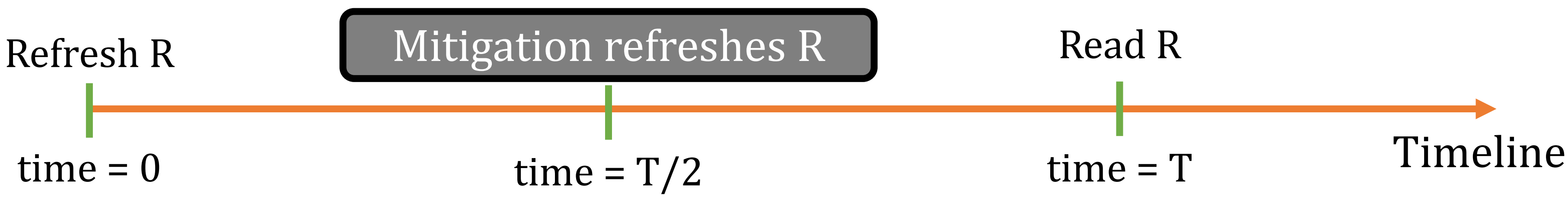
1. Identify a row (R) with **T** retention time

Row R experiences no bitflips only if on-DRAM-die mitigation exists

4. Issue a periodic REF command (trigger mitigation)

5. Wait for T/2, read out row R and check for bitflips





Experimental Methodology

1. Identify a row (R) with **T** retention time

Row R experiences no bitflips
only if on-DRAM-die mitigation exists

4. Issue a periodic REF command (trigger mitigation)

5. Wait for T/2, read out row R and check for bitflips

Row R experiences retention bitflips
if not refreshed at T/2

Victim Row R

Refresh victim row

[Hassan+, MICRO'21]

HBM2 DRAM Chips Implement Undisclosed TRR

The HBM2 chip implements an undisclosed on-die RowHammer mitigation mechanism

This mechanism performs a victim row refresh operation every 17 periodic refresh (REF) operations

This mitigation resembles the one in DDR4 chips from one major manufacturer

Outline

1. HBM DRAM Organization & Operation

2. DRAM Cell Leakage & RowHammer

3. HBM DRAM Testing Methodology

4. RowHammer Spatial Variation Analysis

5. On-die RowHammer Mitigation Analysis

6. Conclusion

Conclusion

We provide the first detailed experimental characterization of RowHammer in a modern HBM2 DRAM chip

Different channels in 3D-stacked HBM chips exhibit different RowHammer vulnerability

DRAM rows near the end of a DRAM bank are more RowHammer resilient

Two implications for RowHammer attacks and mitigations:

1. Faster and more effective attacks
2. More efficient mitigations

A modern HBM chip implements undisclosed on-DRAM-die RowHammer mitigation (e.g., similar to DDR4 chips)

Future Directions: To present more insights into how RowHammer behaves in HBM

1. Test more HBM DRAM chips, data patterns, at different temperature and voltage levels
2. Investigate read-disturb-based interference across different 3D-stacked HBM DRAM channels
3. Study the effects of the new read-disturb phenomenon, RowPress [Luo+, ISCA'23]

Luo et al., "RowPress: Amplifying Read Disturbance in Modern DRAM Chips," in ISCA, 2023.

Available on ArXiv

https://arxiv.org/abs/2305.17918

arXiv > cs > arXiv:2305.17918

Search... All fields Search

Help | Advanced Search

Computer Science > Cryptography and Security

[Submitted on 29 May 2023]

An Experimental Analysis of RowHammer in HBM2 DRAM Chips

Ataberk Olgun, Majd Osseiran, Abdullah Giray Ya{ğ}lık{c}ı, Yahya Can Tuğrul, Haocong Luo, Steve Rhyner, Behzad Salami, Juan Gomez Luna, Onur Mutlu

RowHammer (RH) is a significant and worsening security, safety, and reliability issue of modern DRAM chips that can be exploited to break memory isolation. Therefore, it is important to understand real DRAM chips' RH characteristics. Unfortunately, no prior work extensively studies the RH vulnerability of modern 3D-stacked high-bandwidth memory (HBM) chips, which are commonly used in modern GPUs.

In this work, we experimentally characterize the RH vulnerability of a real HBM2 DRAM chip. We show that 1) different 3D-stacked channels of HBM2 memory exhibit significantly different levels of RH vulnerability (up to 79% difference in bit error rate), 2) the DRAM rows at the end of a DRAM bank (rows with the highest addresses) exhibit significantly fewer RH bitflips than other rows, and 3) a modern HBM2 DRAM chip implements undisclosed RH defenses that are triggered by periodic refresh operations. We describe the implications of our observations on future RH attacks and defenses and discuss future work for understanding RH in 3D-stacked memories.

Comments: To appear at DSN Disrupt 2023

Subjects: **Cryptography and Security (cs.CR)**; Hardware Architecture (cs.AR)

Cite as: arXiv:2305.17918 **[cs.CR]**

(or arXiv:2305.17918v1 **[cs.CR]** for this version)

https://doi.org/10.48550/arXiv.2305.17918

Download:

- PDF
- Other formats

Current browse context:
cs.CR

< prev | next >

new | recent | 2305

Change to browse by:

cs

cs.AR

References & Citations

- NASA ADS
- Google Scholar
- Semantic Scholar

Export BibTeX Citation

Bookmark

An Experimental Analysis of RowHammer in HBM2 DRAM Chips

Link/QR code to full paper
https://arxiv.org/pdf/2305.17918

Ataberk Olgun   Majd Osseiran

A. Giray Yağlıkçı   Yahya Can Tuğrul   Haocong Luo   Steve Rhyner

Behzad Salami   Juan Gomez Luna   Onur Mutlu

ETH zürich   SAFARI   AMERICAN UNIVERSITY OF BEIRUT

Extended version submitted to HPCA

Understanding Read Disturbance in High Bandwidth Memory: An Experimental Analysis of Real HBM2 DRAM Chips

- Tests 5 more HBM2 chips
- Tests more DRAM components (e.g., banks and rows) per chip
- Analyzes hammer counts to induce more than one bitflip ($HC_{second,\ third, \ldots,\ tenth}$)
- Analyzes the RowPress vulnerability of HBM2 chips
- Further reverse engineers the on-DRAM-die RH defense mechanism

Methodology

TABLE II

TESTED DRAM COMPONENTS FOR EACH EXPERIMENT TYPE

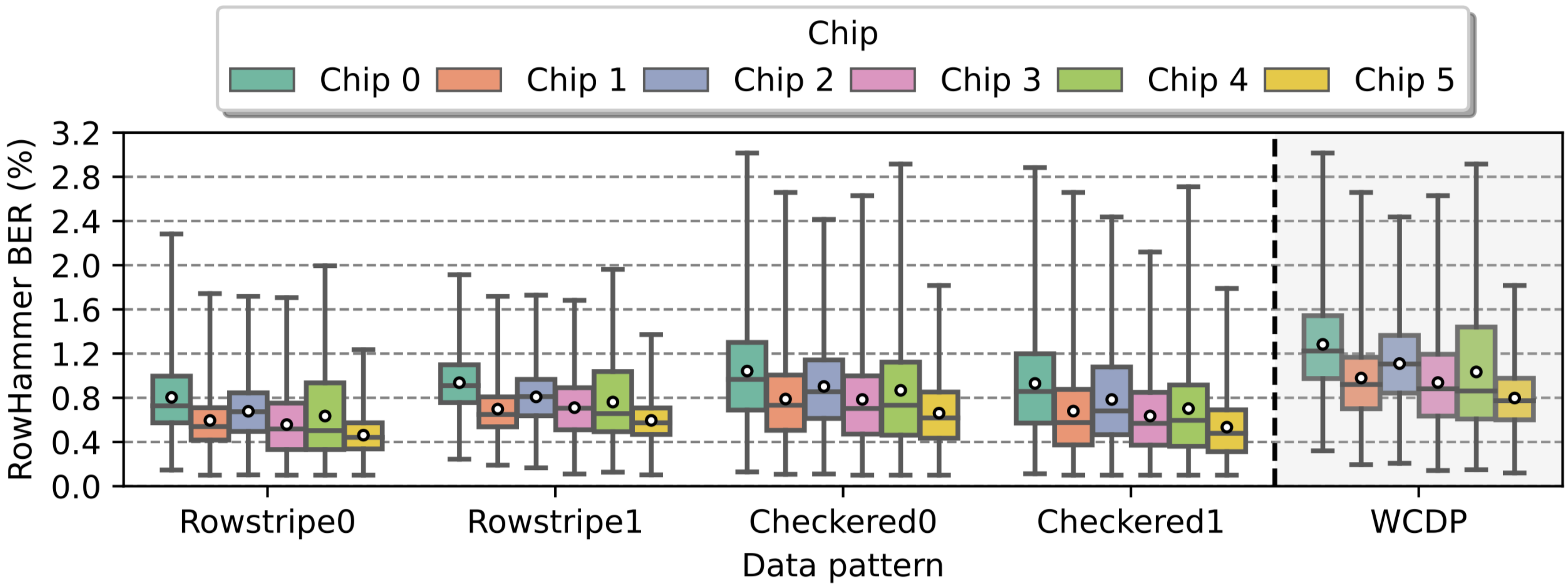
Experiment Type	Rows (Per Bank)	Banks	Pseudo Channels	Channels
RowHammer BER	16384	1	1	8
RowHammer HC_{first}	3072	3	2	8
RowPress BER	384	1	1	3
RowPress HC_{first}	384	1	1	3

TABLE III

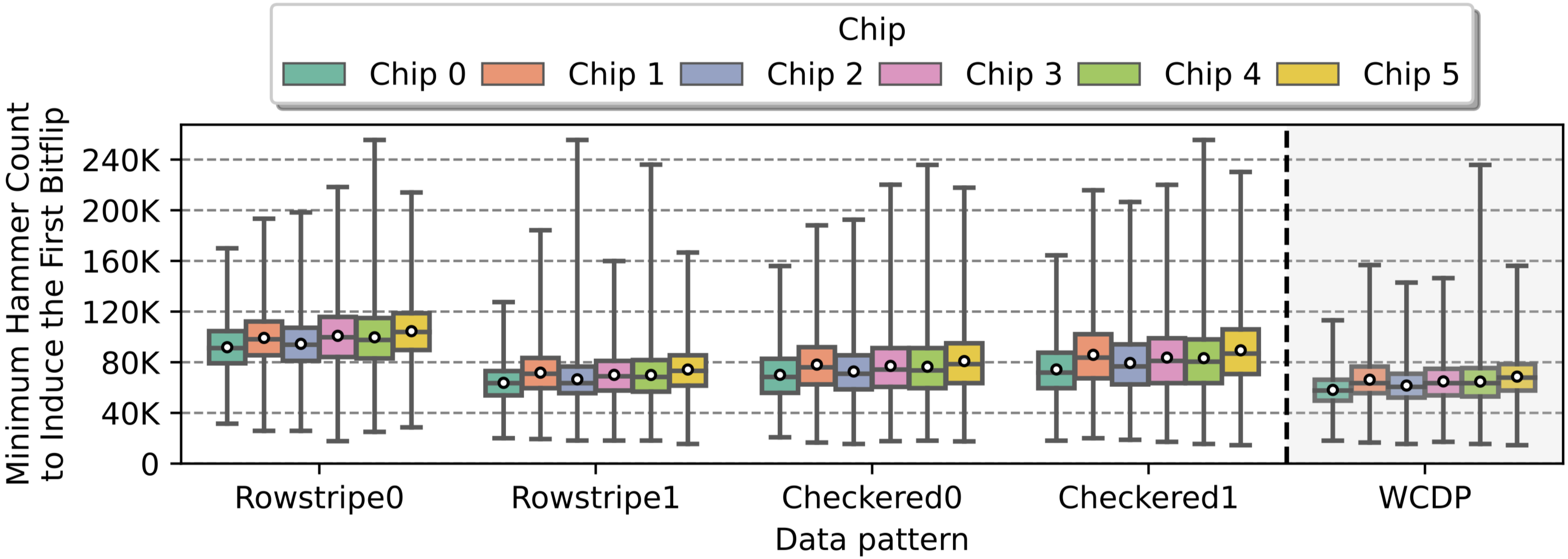
LABELS FOR THE HBM2 CHIPS IN EACH TESTED FPGA BOARD

FPGA Board	Chip Label
Bittware XUPVVH	Chip 0
AMD Xilinx Alveo U50	Chip 1, 2, 3, 4, 5

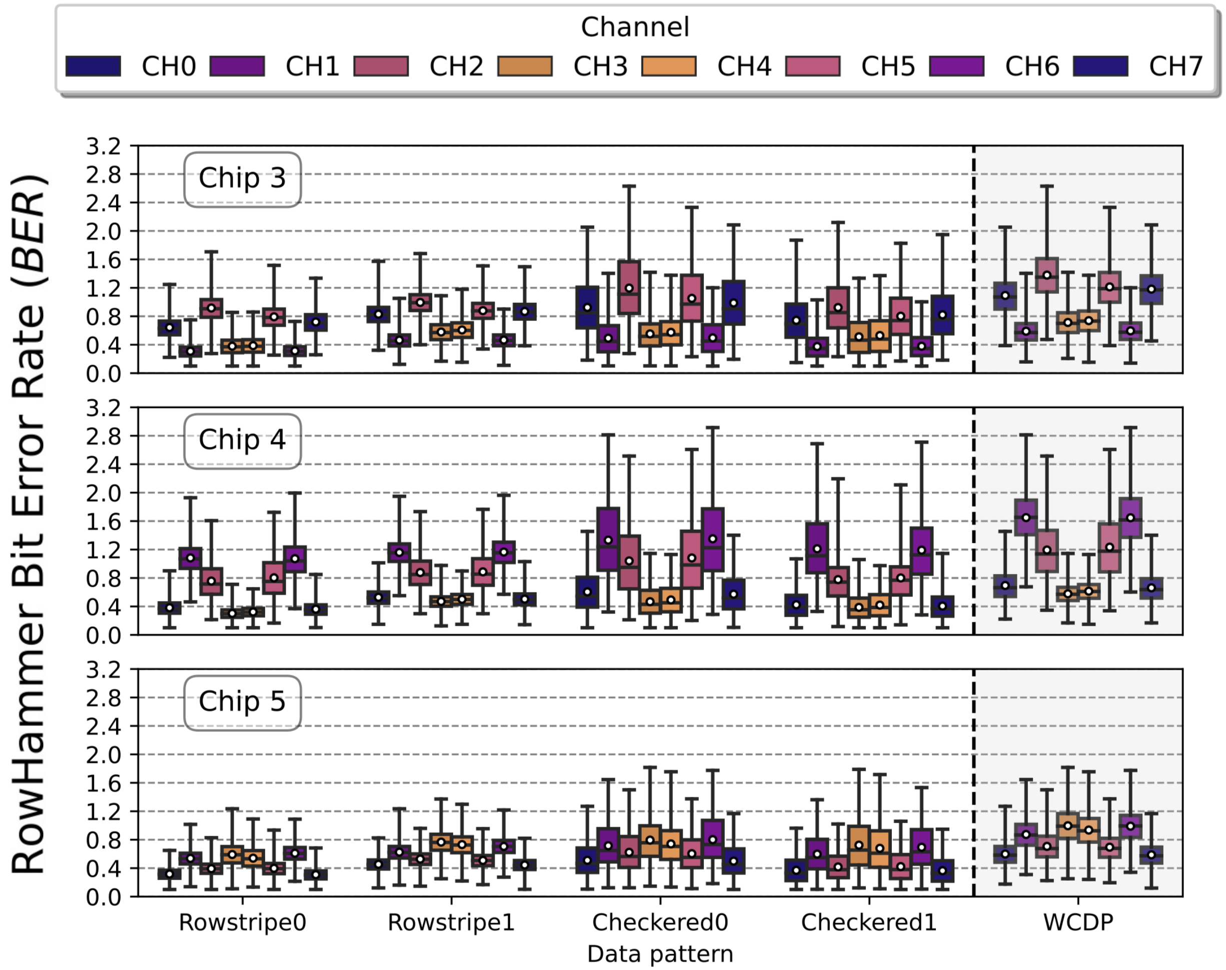
RowHammer BER Across Chips



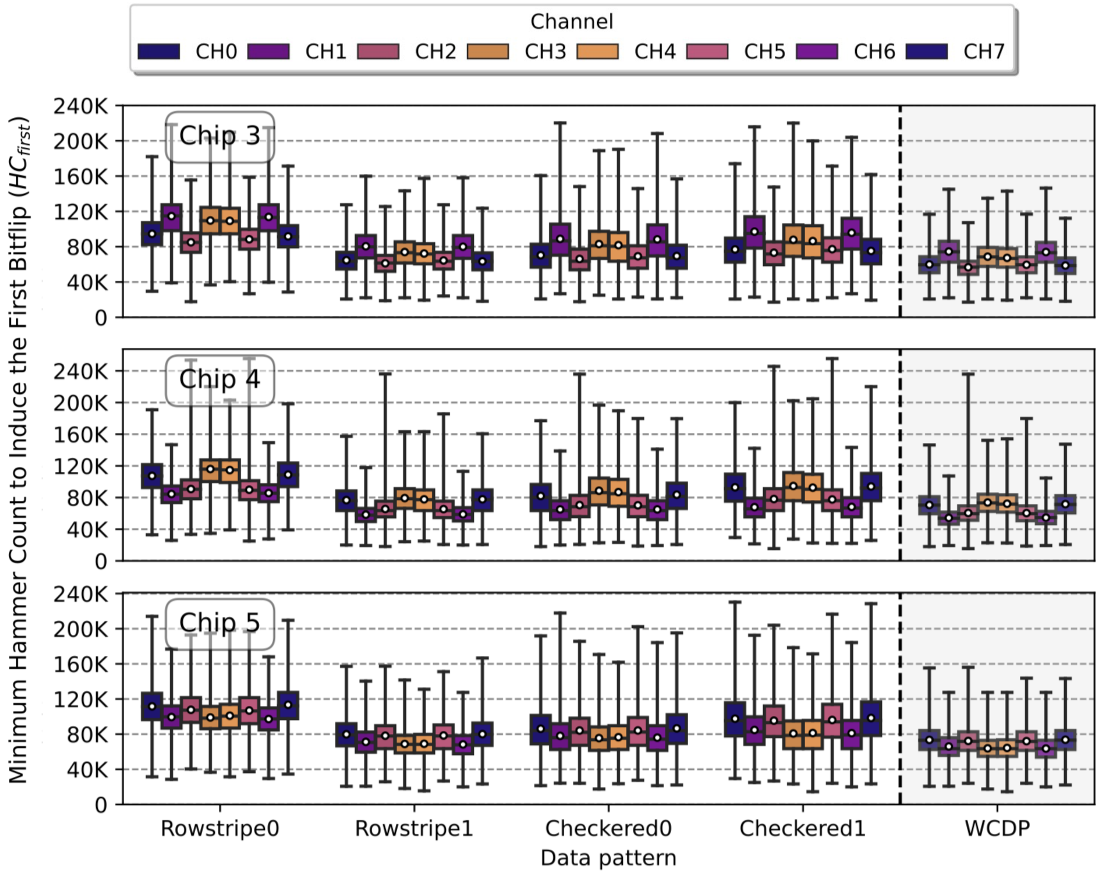
RowHammer HC_{first} Across Chips



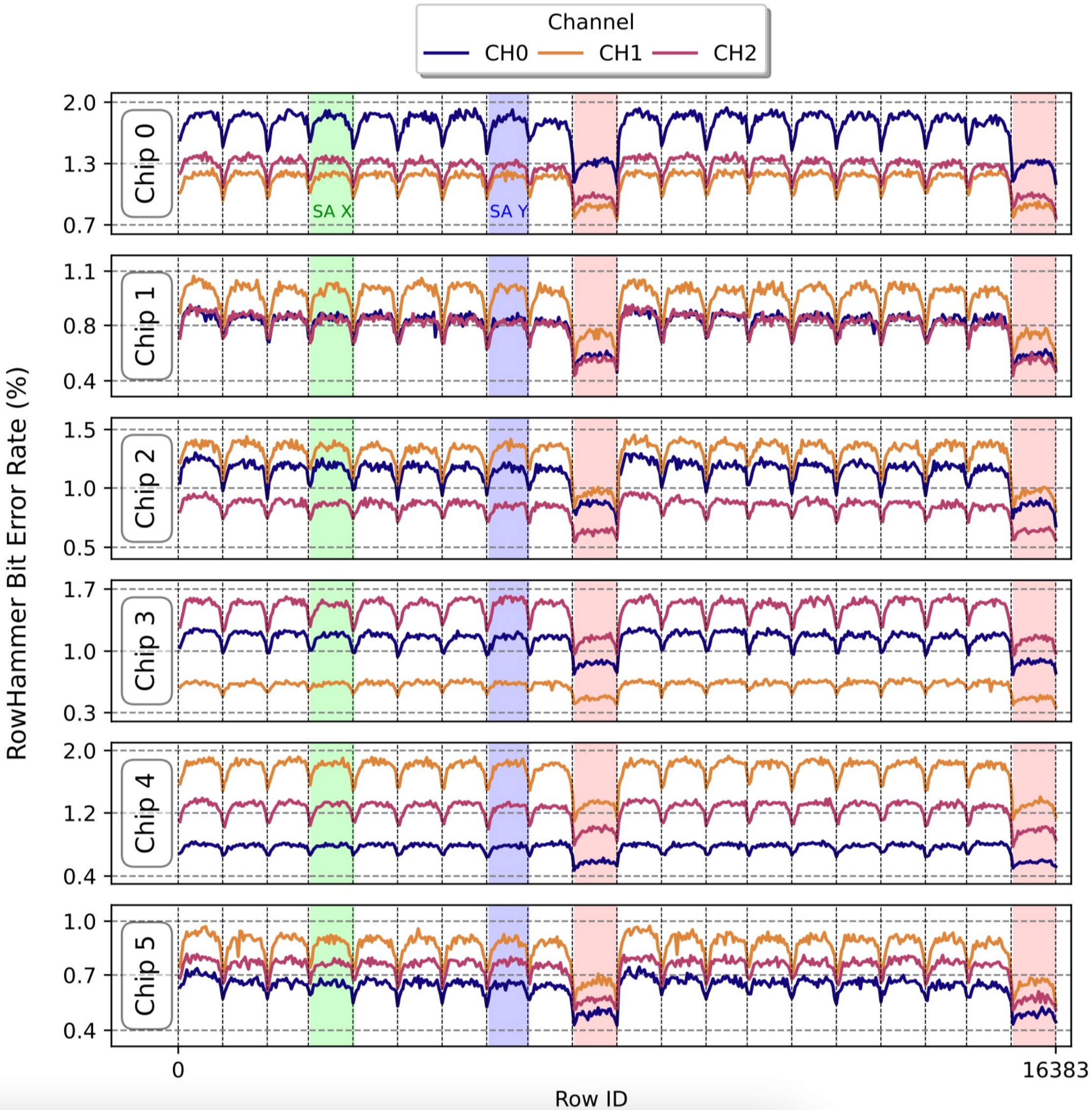
RowHammer BER Across Channels



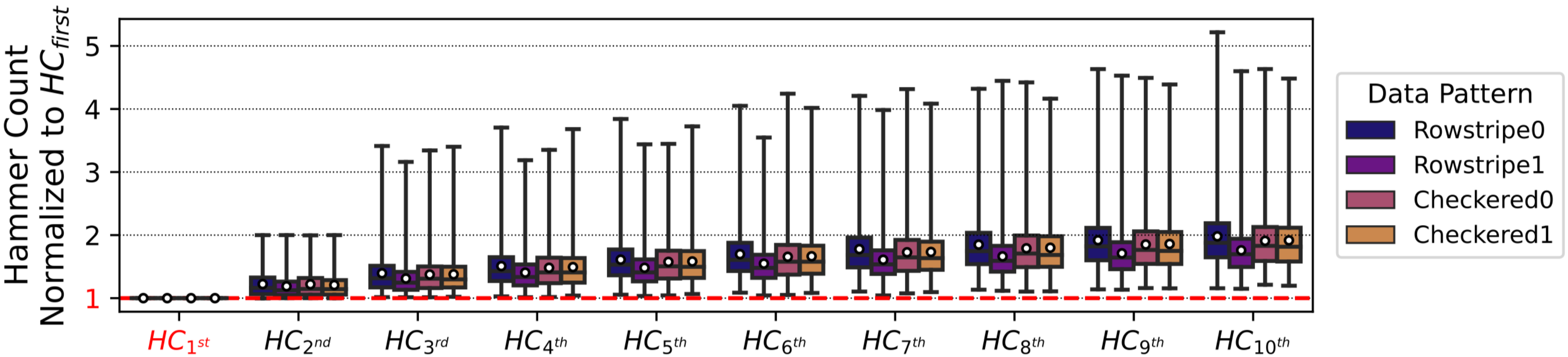
RowHammer HC_{first} Across Channels



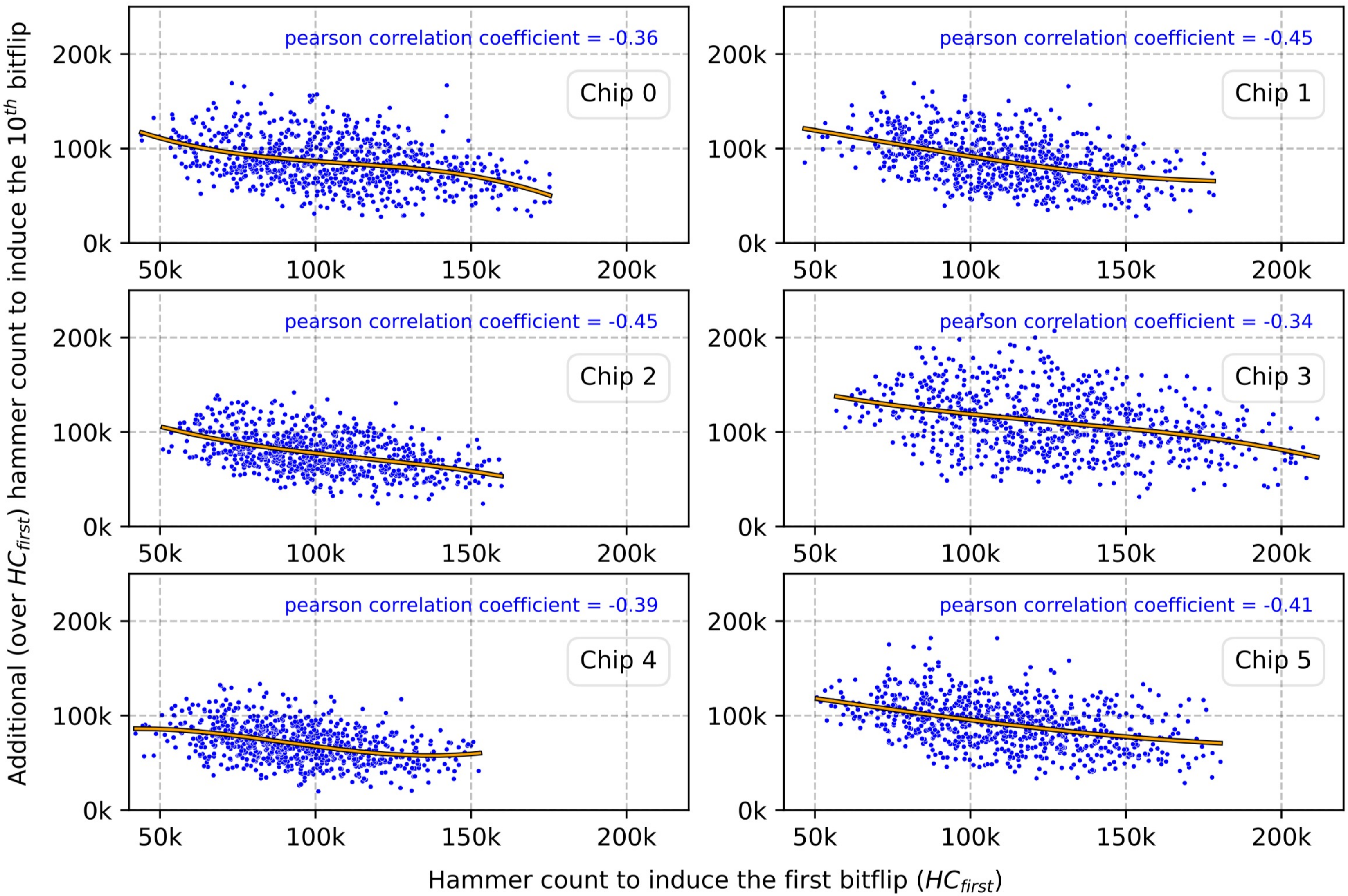
RowHammer BER Across Rows



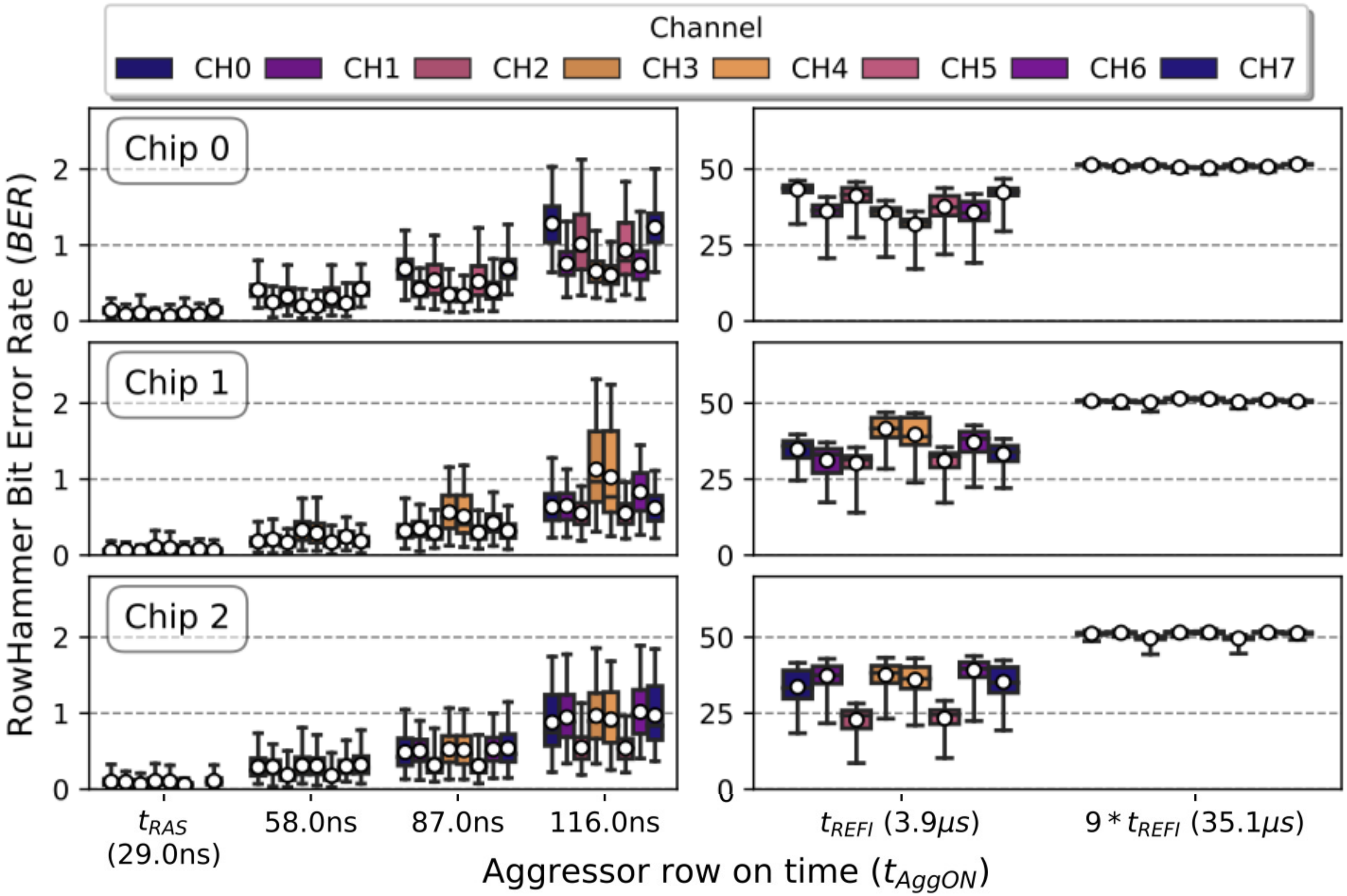
RowHammer's Sensitivity to Hammer Count (I)



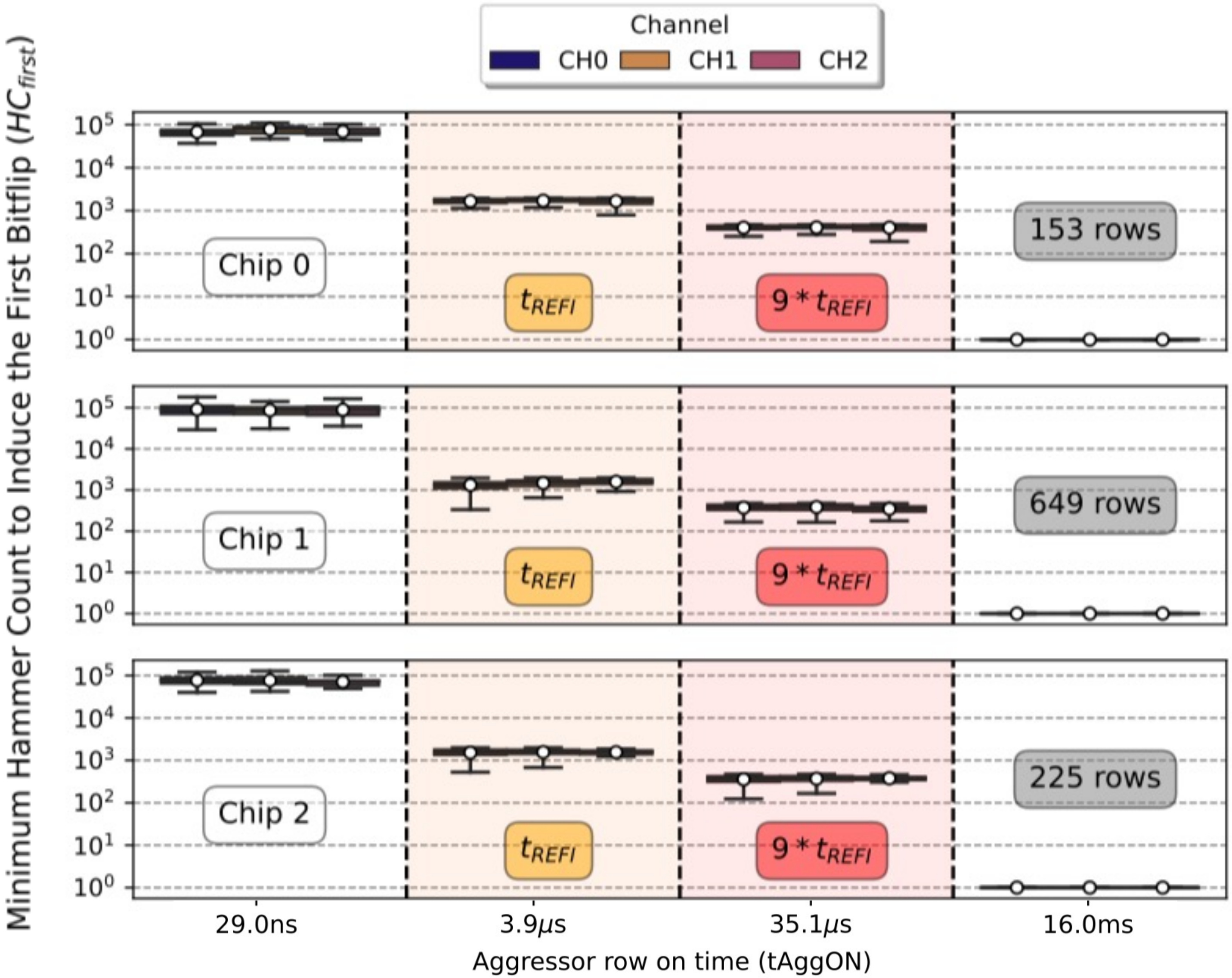
RowHammer's Sensitivity to Hammer Count (II)



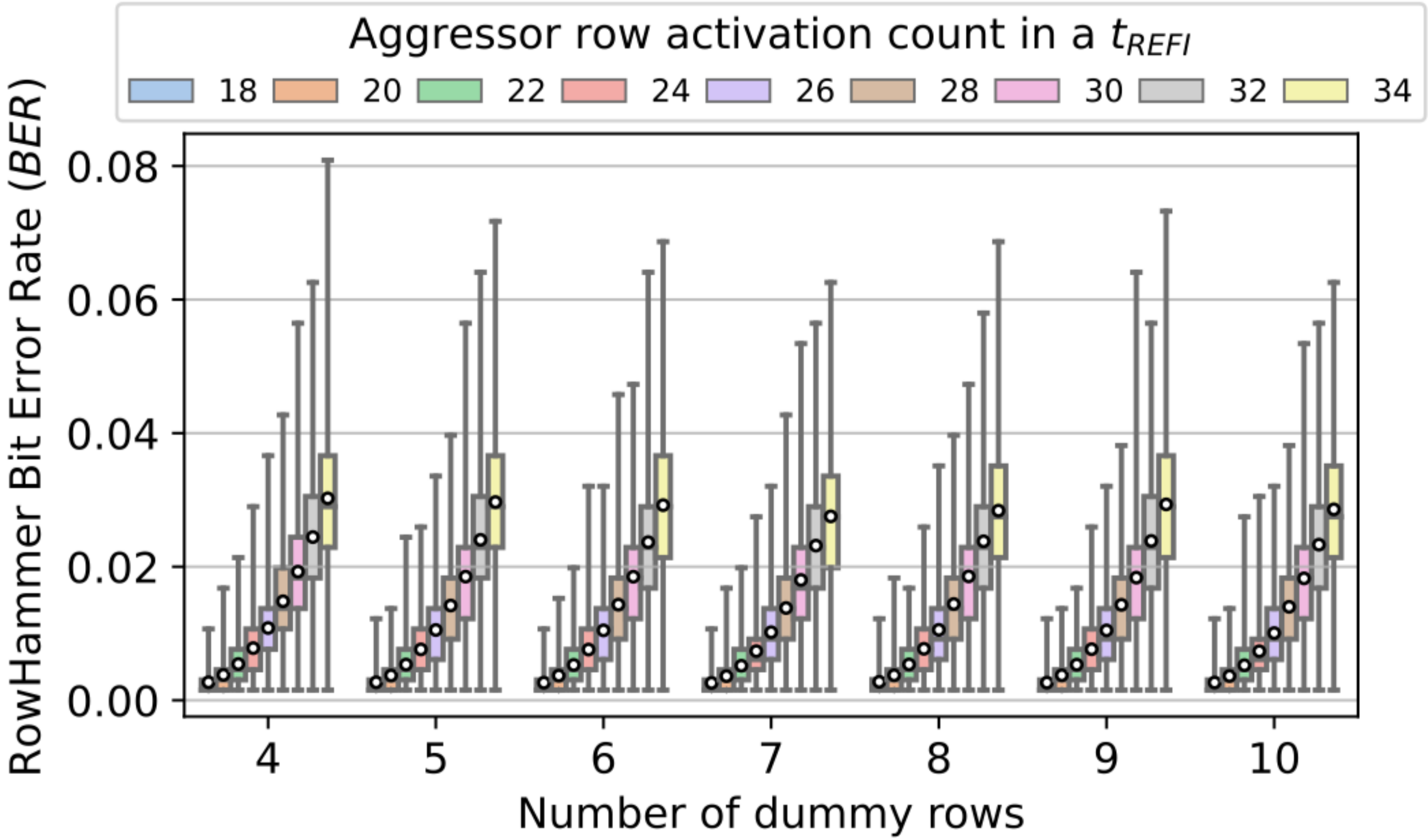
RowPress BER Across Channels



RowPress HC_{first} Across Channels



Attack Patterns to Bypass Undisclosed TRR



Silicon Level RowHammer and RowPress Mechanisms

Major RowHammer Silicon Mechanisms (I)

- There are two major silicon-level causes for RowHammer bitflips [1, 7].
- First, **capacitive coupling between the physically-adjacent aggressor and victim wordlines causes crosstalk.**
  - When the aggressor wordline is activated, the potential of the victim wordlines also increases [10], causing an increase in the access transistor subthreshold leakage of the victim cell [1, 4].
  - When the aggressor wordline is repeatedly activated many times, the accumulation of the increased subthreshold leakage causes bitflips.
- Second, **repeated switching of the channel of the access transistor of the aggressor cell injects electrons into the storage node of the victim cell, causing it to lose charge** [1 - 5].
  - The injected electrons mainly come from two sources.
  - First, when the aggressor access transistor is switched off, the diffused channel electrons are attracted to the storage node of the victim cell [2, 3, 6, 8]. This is because the victim cell's storage node has a higher potential compared to the bitline [2]. These electrons recombine with the stored charge in the victim cell, reducing the cell potential, and eventually causing a bitflip.
  - Second, the interface charge traps of the aggressor access transistor traps electrons during the activation of the victim row [4, 5, 8]. Later, when the aggressor row is closed, the trapped electrons are released and find their way to the storage node of the victim cell.

Major RowHammer Silicon Mechanisms (II)

- Although existing literature [1-6] suggests that both capacitive crosstalk between wordlines and electron migration and injection are the two fundamental silicon-level mechanisms for RowHammer bitflips, they do not quantitatively compare the contribution of these two mechanisms to make out a dominant cause for RowHammer.

- A recent work [9] investigates how each mechanism contributes to the significantly increased RowHammer vulnerability (i.e., requiring much less aggressor row activation to induce a bitflip) of the double-sided access pattern.

- The key takeaway of [9] is that the **trap-assisted electron migration & injection is the dominant mechanism for the increased vulnerability to double-sided RowHammer** (i.e., requiring fewer aggressor row activations to induce a bitflip) compared to single-sided, while **capacitive crosstalk is not a major factor in the increased vulnerability to double-sided RowHammer** compared to single-sided.

Silicon-Level RH: Pictorial Illustration (I)

- Figure 1 illustrates the physical layout of DRAM.
- Figure 2 shows how electrons are injected into the victim cell.

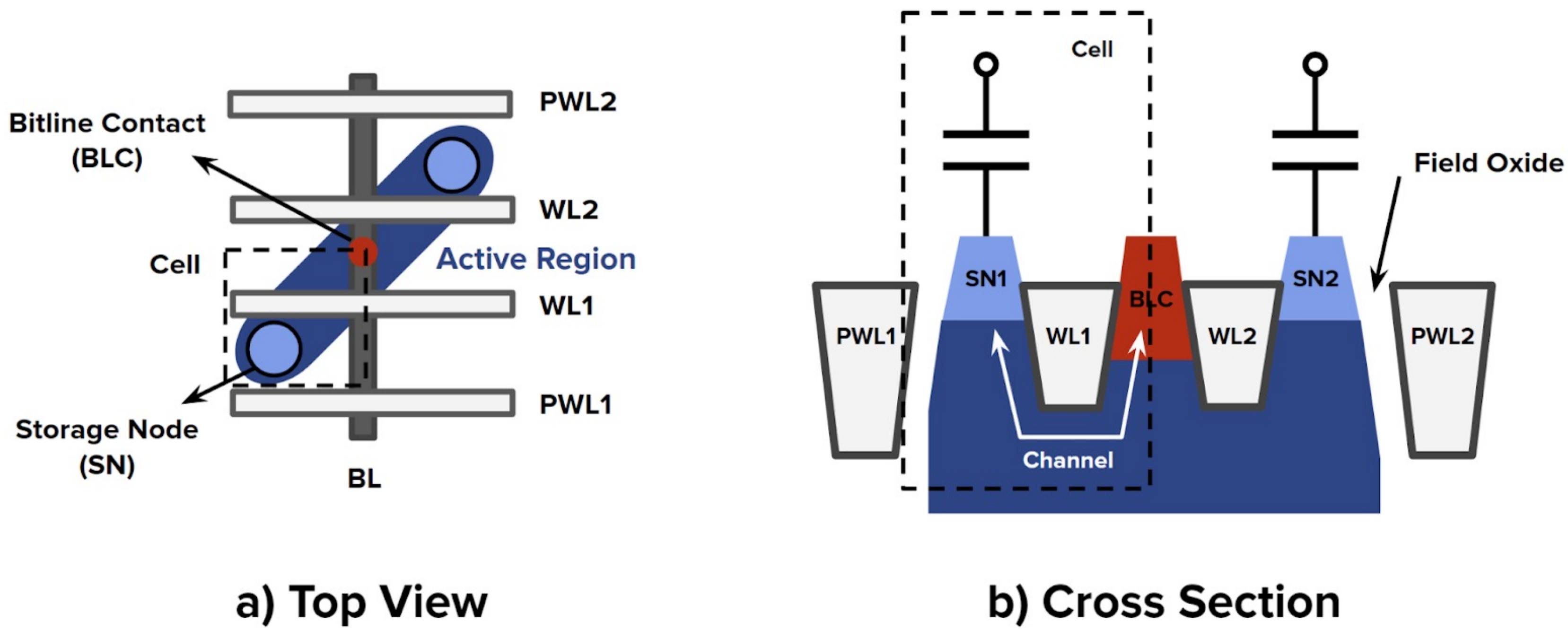


Figure 1. DRAM Physical Layout (Figurative)

Silicon-Level RH: Pictorial Illustration (II)

- Figure 2a) shows the initial state, where WL1 is the aggressor wordline and SN2 is the storage node of the victim cell, which is initially positively charged.
- When the aggressor wordline is open (Figure 2b), excessive electrons are concentrated in the aggressor's channel ① due to channel inversion and/or interface traps.
- When the aggressor wordline is closed, the channel inversion layer collapses (and/or the trapped electrons get released), and some of the excessive electrons can migrate and inject into the victim cell ②.

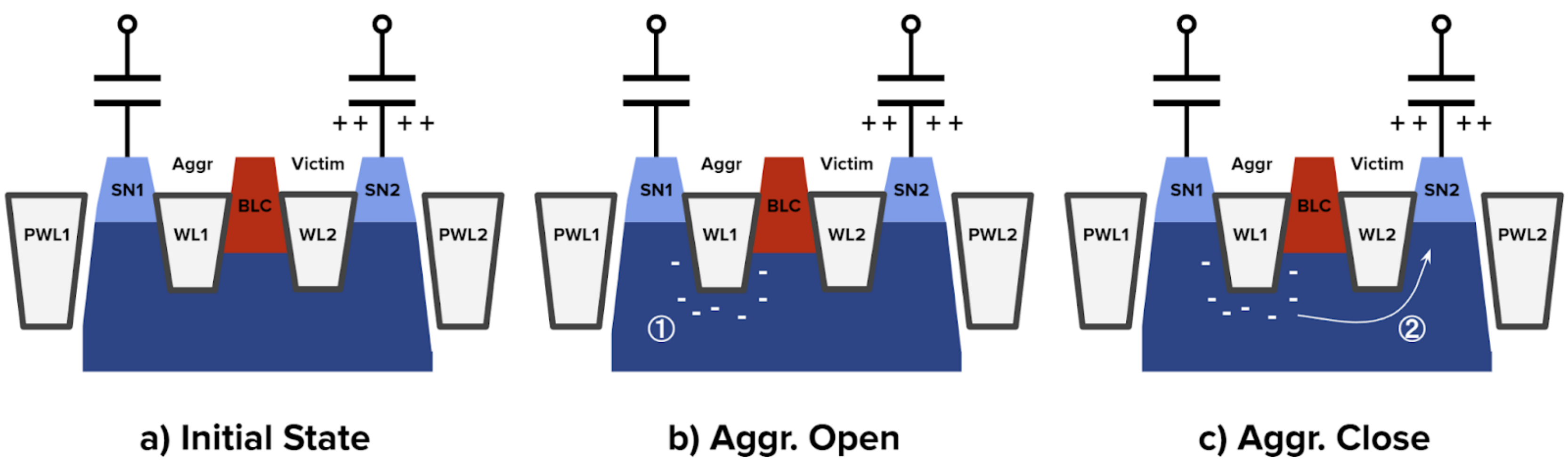


Figure 2. Electron Migration & Injection (Figurative)

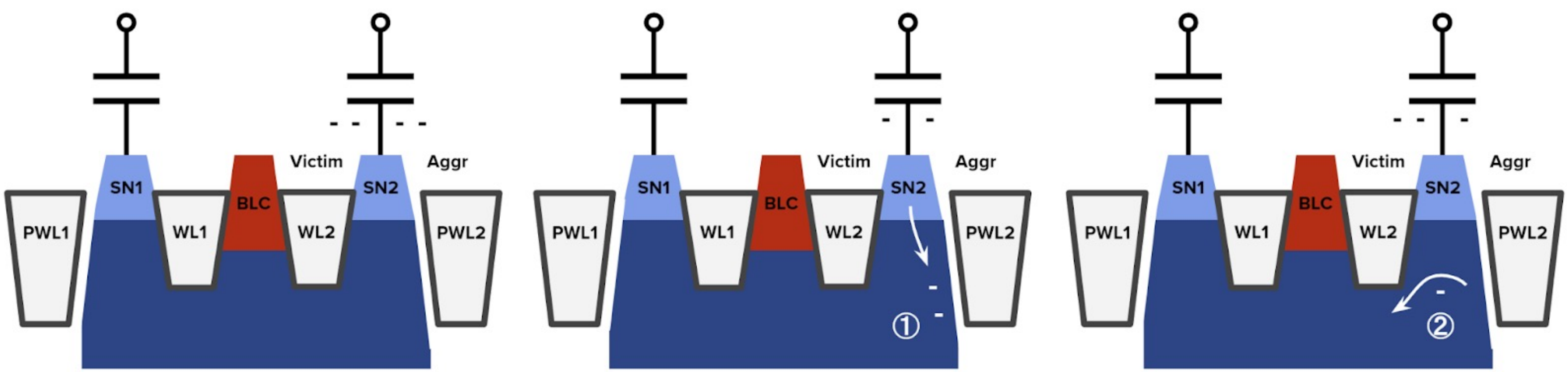
RH Silicon Mechanism References

[1] Walker et al., "On DRAM Rowhammer and the Physics of Insecurity," in IEEE Transactions on Electron Devices, 2021
[2] Park et al., "Experiments and root cause analysis for active-precharge hammering fault in DDR3 SDRAM under 3 × nm technology," in Microelectronics Reliability, 2016
[3] Yang et al., "Suppression of Row Hammer Effect by Doping Profile Modification in Saddle-Fin Array Devices for Sub-30-nm DRAM Technology," in IEEE Transactions on Device and Materials Reliability, 2016
[4] Ryu et al., "Overcoming the Reliability Limitation in the Ultimately scaled DRAM using Silicon Migration Technique by Hydrogen Annealing," in Technical Digest - International Electron Devices Meeting, IEDM, 2018
[5] Yang et al., "Trap-Assisted DRAM Row Hammer effect," in IEEE Electron Device Letters, 2019
[6] Gautam et al., "Row Hammering Mitigation Using Metal Nanowire in Saddle Fin DRAM," in IEEE Transactions on Electron Devices, 2019
[7] Han et al., "Surround Gate Transistor With Epitaxially Grown Si Pillar and Simulation Study on Soft Error and Rowhammer Tolerance for DRAM," in IEEE Transactions on Electron Devices, 2021
[8] Park et al., "Row Hammer Reduction Using a Buried Insulator in a Buried Channel Array Transistor," in IEEE Transactions on Electron Devices, 2021
[9] Zhou et al., "Double-sided Row Hammer Effect in Sub-20 nm DRAM: Physical Mechanism, Key Features and Mitigation," in IEEE International Reliability Physics Symposium (IRPS), 2023
[10] Redeker et al., "An Investigation into Crosstalk Noise in DRAM Structures," in Proceedings of the 2002 IEEE International Workshop on Memory Technology, Design and Testing (MTDT), 2002

Major RowPress Silicon Mechanism

- RowPress causes bitflips by keeping the aggressor row open for a long period of time. One silicon-level mechanism to explain RowPress is called **the passing gate effect** [11, 12].

- Figure 3 shows how the passing gate effect causes bitflips.
- In the initial state (Figure 3.a), SN2 is the victim and the passing wordline PWL2 is the aggressor. The victim cell is initially negatively charged.
- When PWL2 is kept open (Figure 3.b), it keeps attracting electrons from the victim cell ①.
- When PWL2 is closed (Figure 3.c), not all the attracted electrons will return to the victim cell, causing leakage.



a) Initial State b) Passing Aggr. Open C) Passing Aggr. Close

RP Silicon Mechanism References

[11] Hong et al., "DSAC: Low-Cost Rowhammer Mitigation Using In-DRAM Stochastic and Approximate Counting Algorithm," arXiv:2302.03591 [cs.CR]

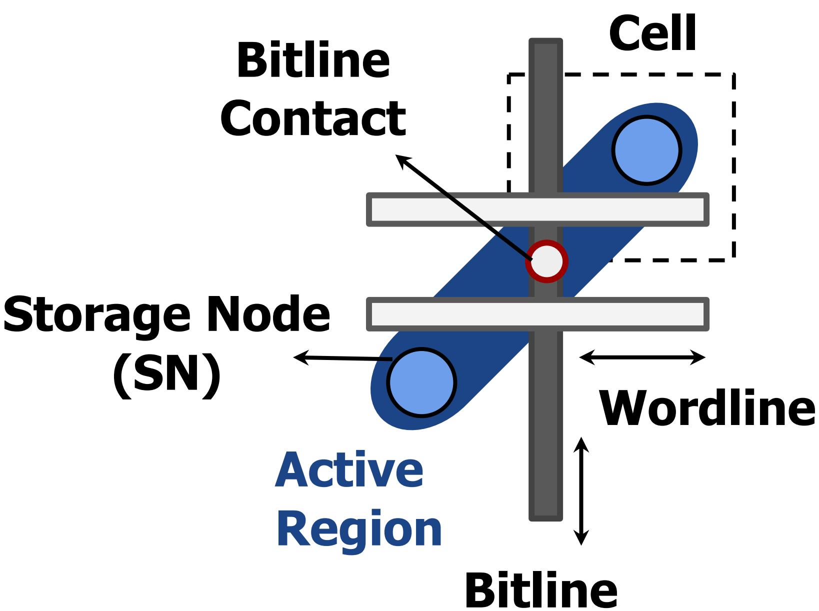
[12] Nam et al., "X-ray: Discovering DRAM Internal Structure and Error Characteristics by Issuing Memory Commands," in IEEE CAL, 2023

Illustrations

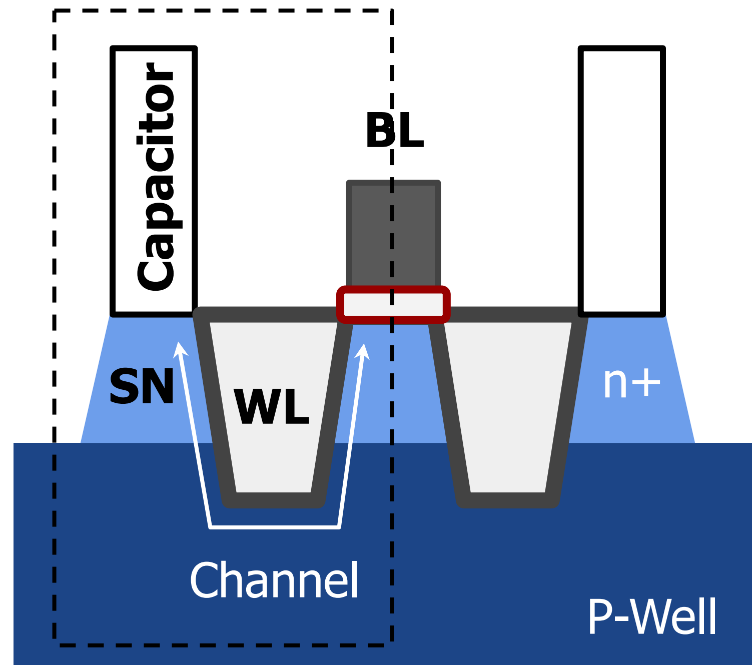
Silicon-Level Disturbance Mechanism

Electron Migration & Injection

Figurative illustration of the physical layout of a DRAM cell



Top View

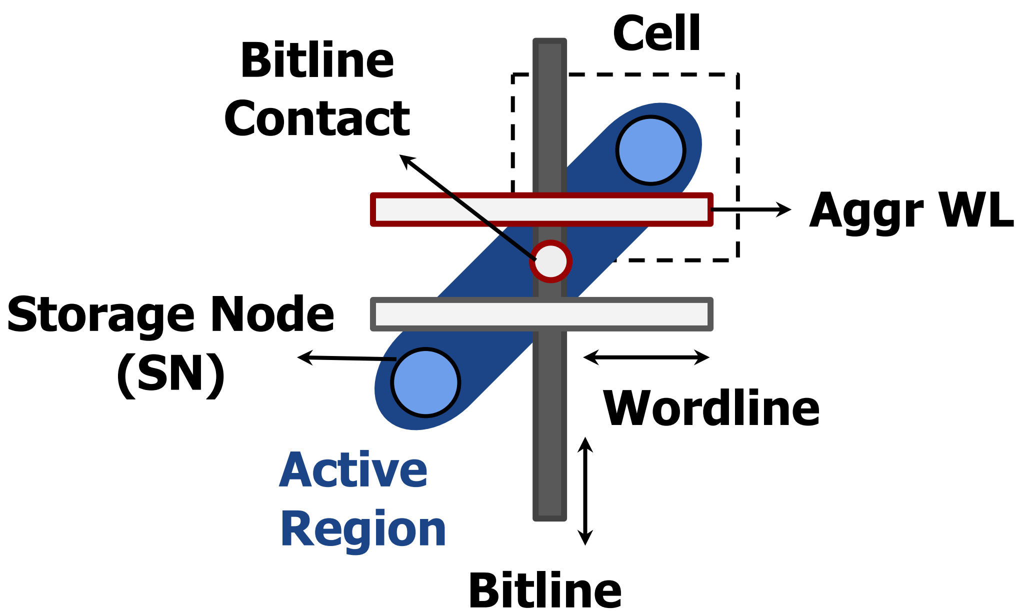


Cross Section

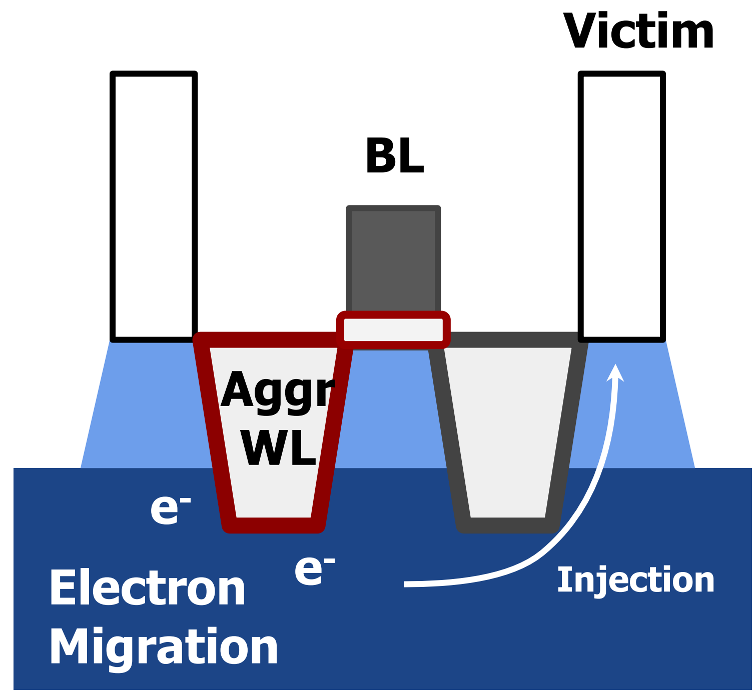
Silicon-Level Disturbance Mechanism

Electron Migration & Injection

High-level: Electrons migrate from the aggr channel to the victim node



Top View



Cross Section

Silicon-Level Disturbance Mechanism

Electron Migration & Injection

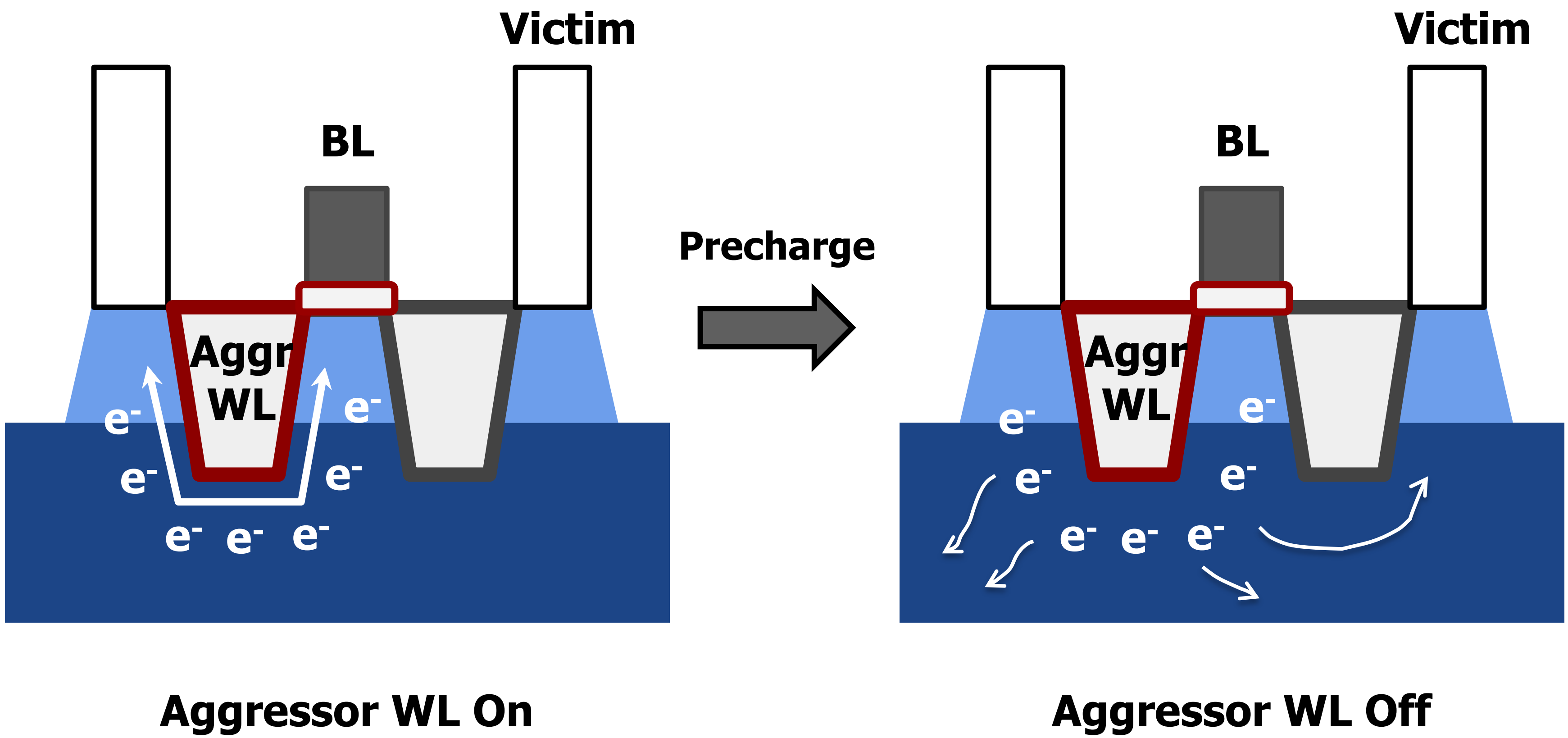
High-level: Electrons migrate from the aggr channel to the victim node

- **Sources of these electrons**
  - Collapse of the inversion layer of the aggressor row's access transistor channel when the aggressor WL is turned off
  - Interface traps at the aggressor WL that capture electrons when the aggressor WL is open, and release them when the aggressor WL is off

Silicon-Level Disturbance Mechanism

Electron Migration & Injection

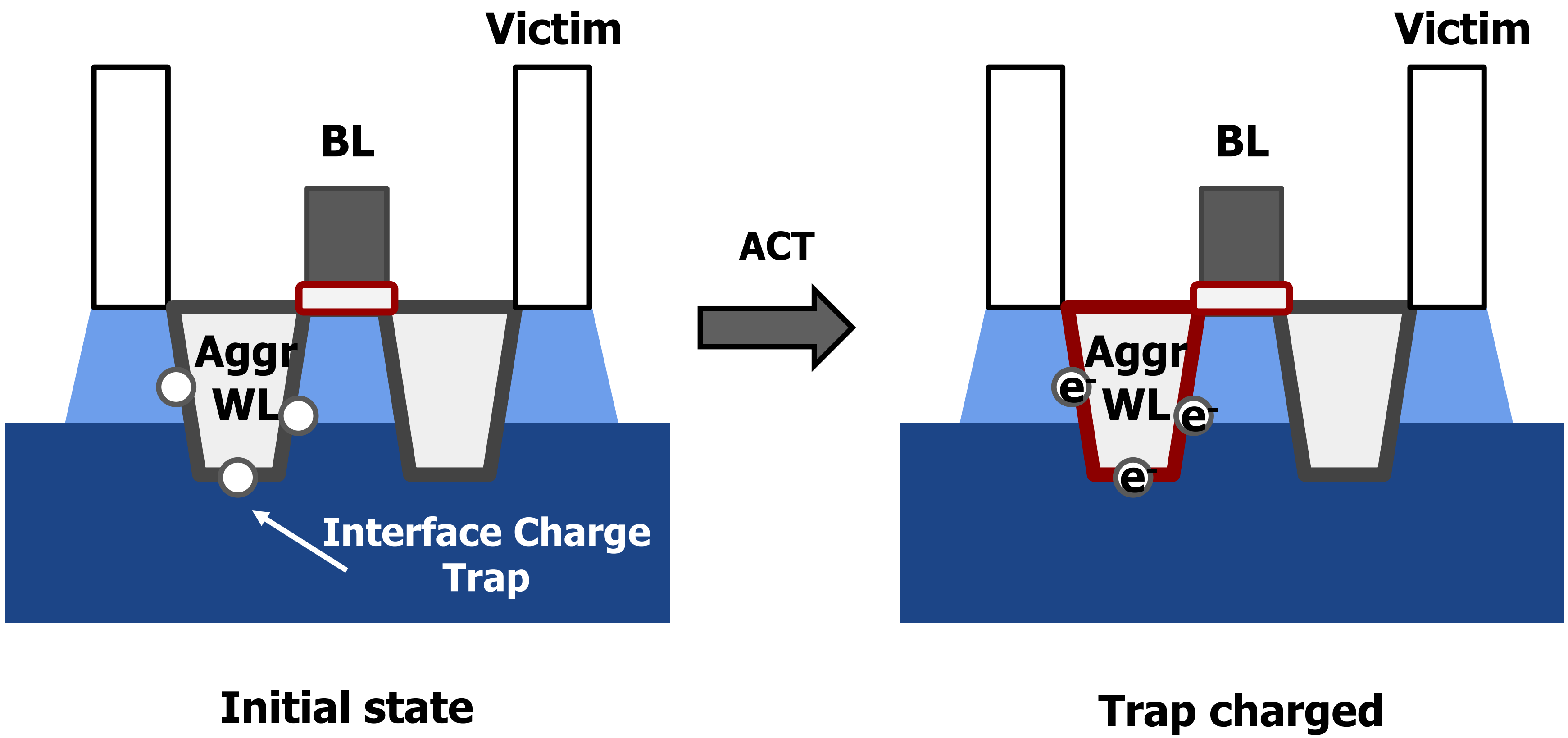
Inversion layer collapse in the aggr channel



Silicon-Level Disturbance Mechanism

Electron Migration & Injection

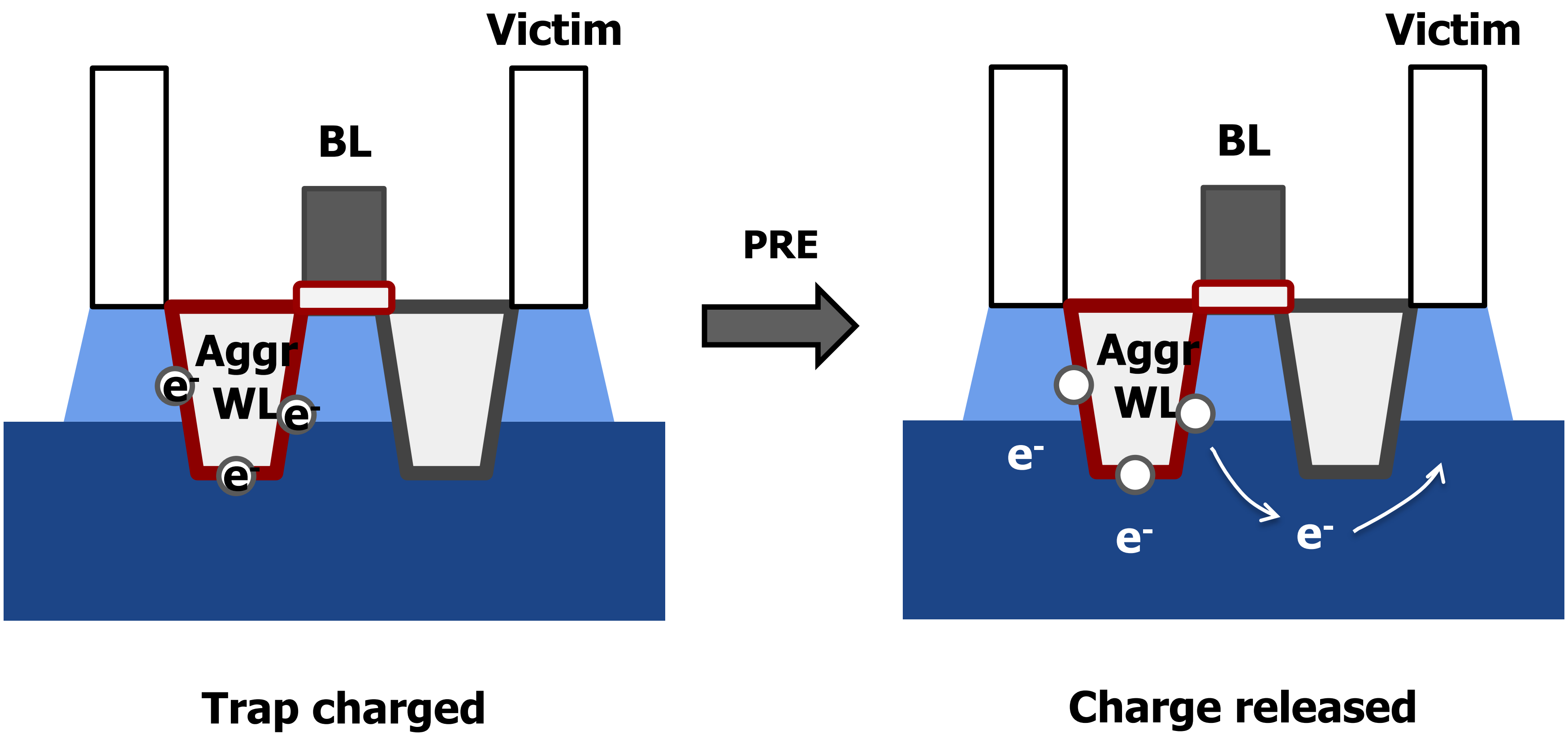
Interface charge trap



Silicon-Level Disturbance Mechanism

Electron Migration & Injection

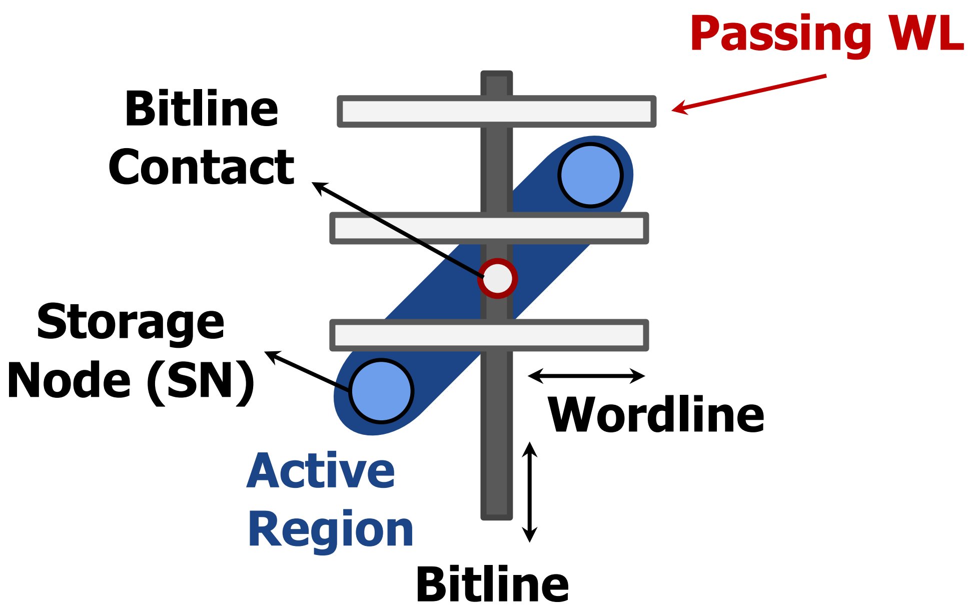
Interface charge trap



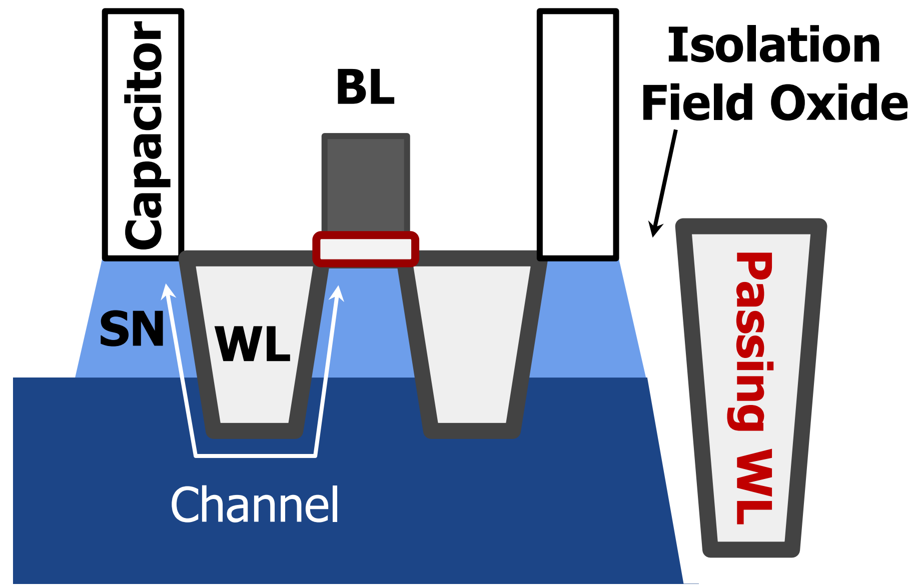
Silicon-Level Disturbance Mechanism

Passing Gate Effect

Figurative illustration of the physical layout of a DRAM cell



Top View

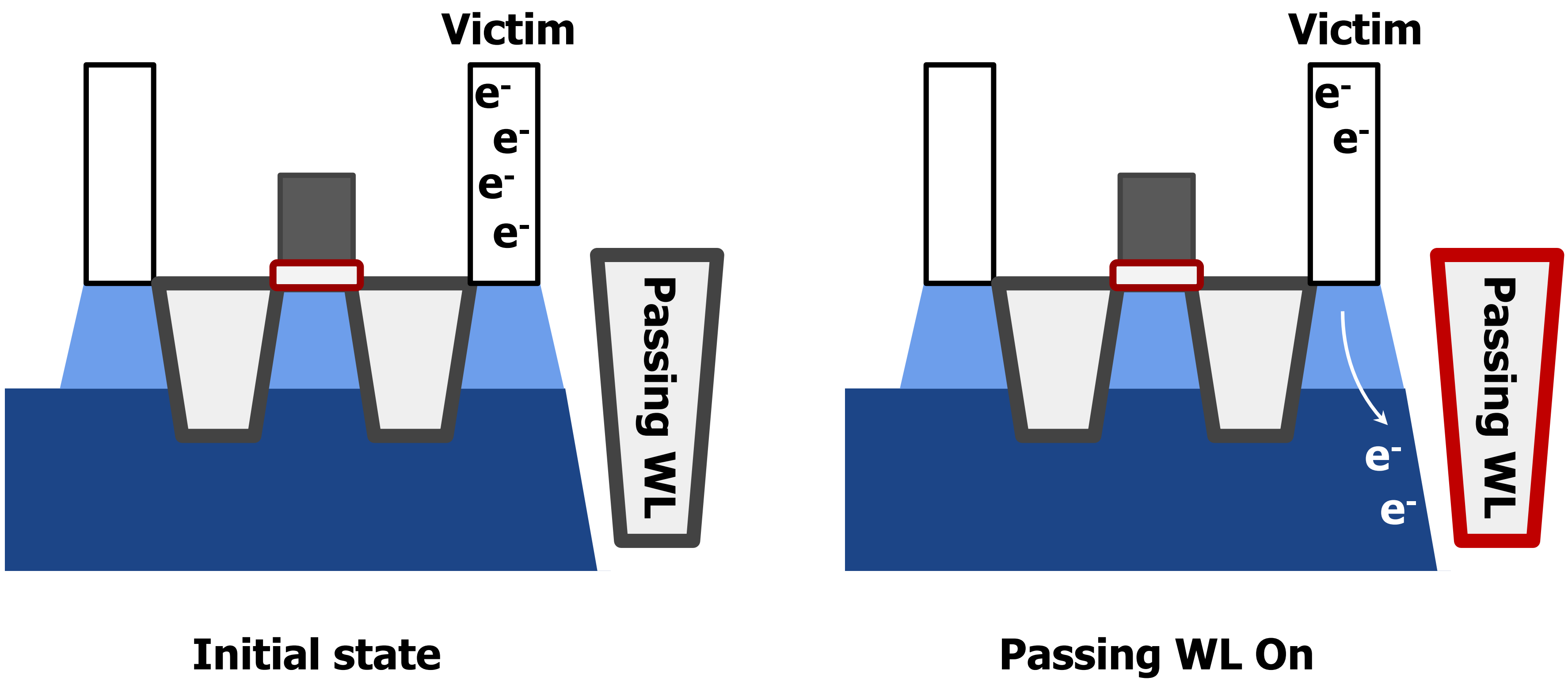


Cross Section

Silicon-Level Disturbance Mechanism

Passing Gate Effect

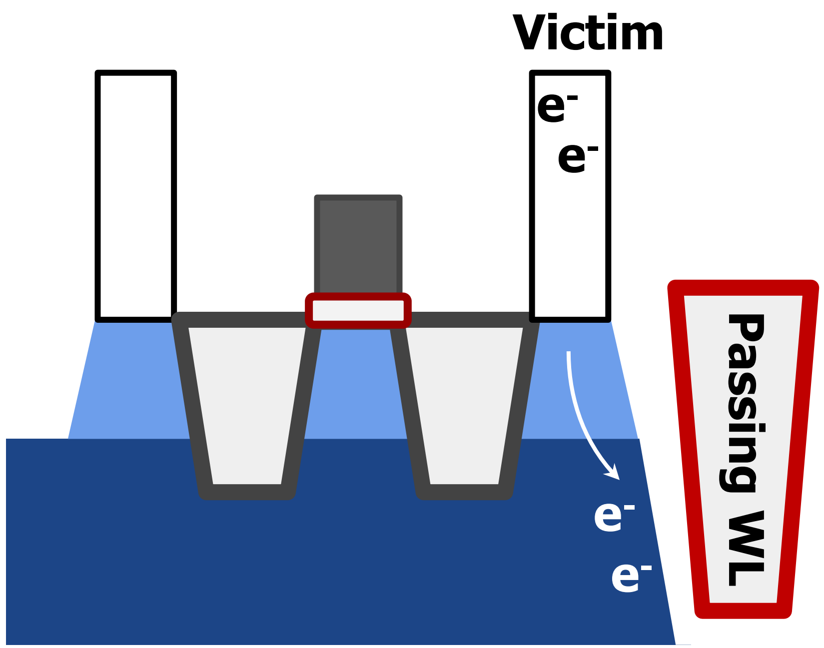
Attracts electrons from the victim



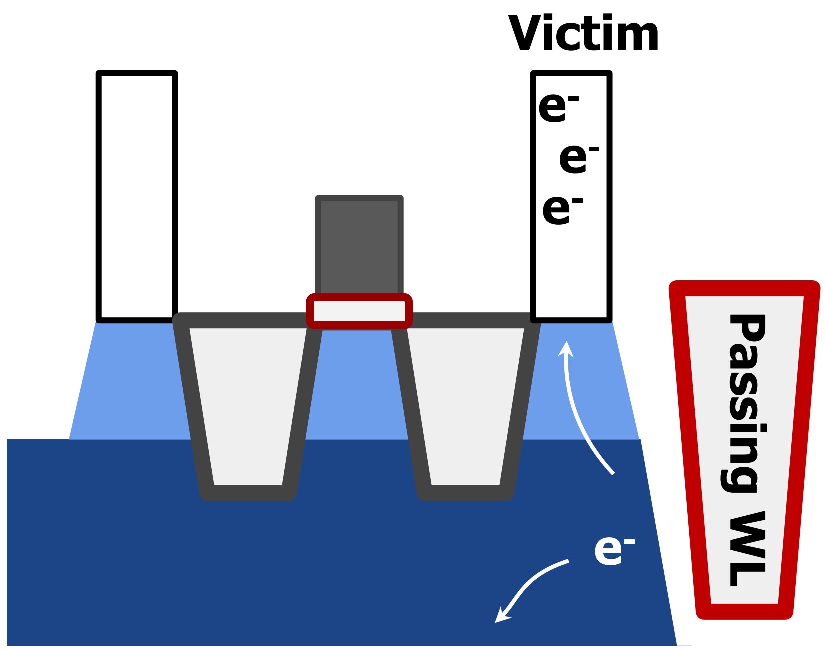
Silicon-Level Disturbance Mechanism

Passing Gate Effect

Attracts electrons from the victim



Passing WL On



Passing WL Off
(Some electrons do not return to the victim)

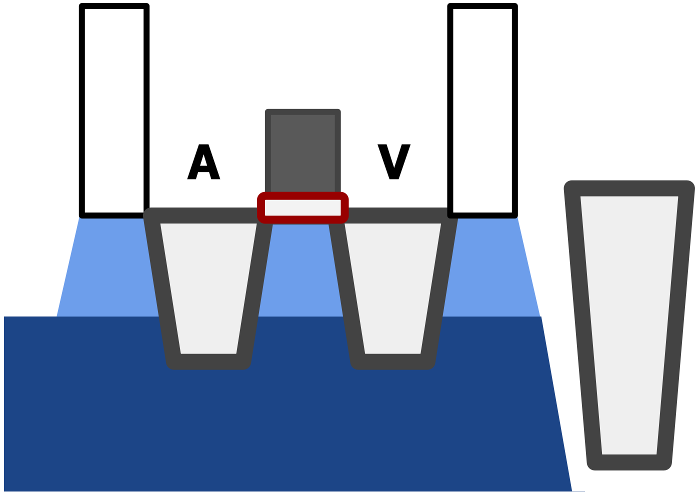
Silicon-Level Disturbance Mechanism

Passing Gate Effect

- The longer the passing WL is open, the more electrons it can attract from the victim.
- Major contributor to the RowPress vulnerability.

Access Pattern - RowHammer

Single-Sided - Case 1

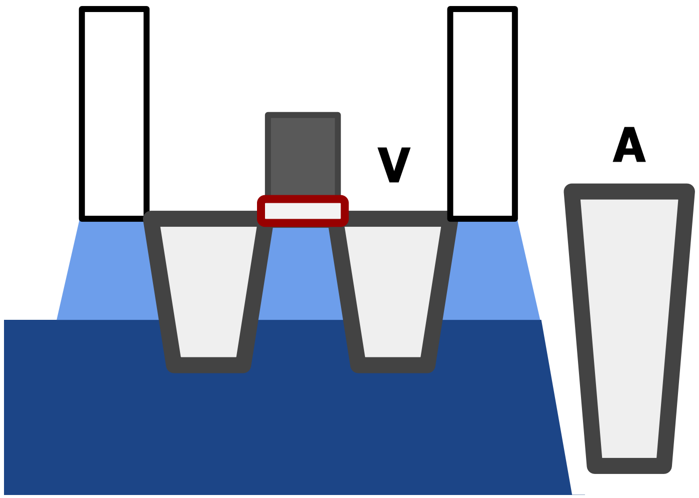


Contributing mechanisms

1. Increased subthreshold leakage due to AWL-VWL crosstalk
2. Electron migration and injection from aggressor channel to victim node
3. "Normal" leakage as time passes by

Access Pattern - RowHammer

Single-Sided - Case 2

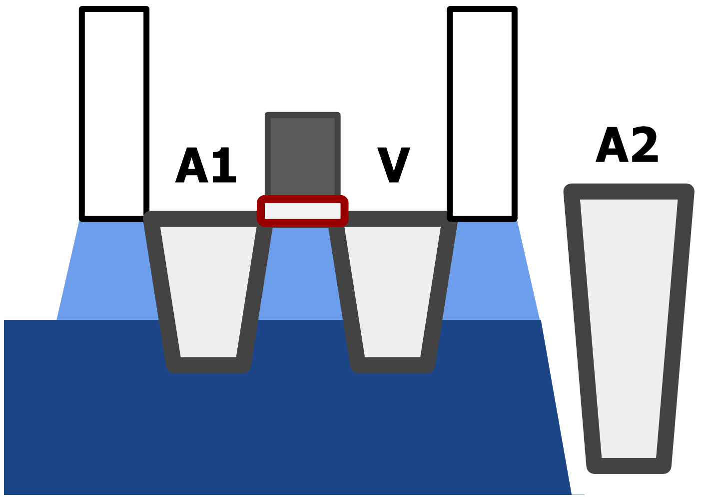


Contributing mechanisms

1. Increased subthreshold leakage due to AWL-VWL crosstalk (?)
2. Passing gate effect
3. "Normal" leakage as time passes by

Access Pattern - RowHammer

Double-Sided



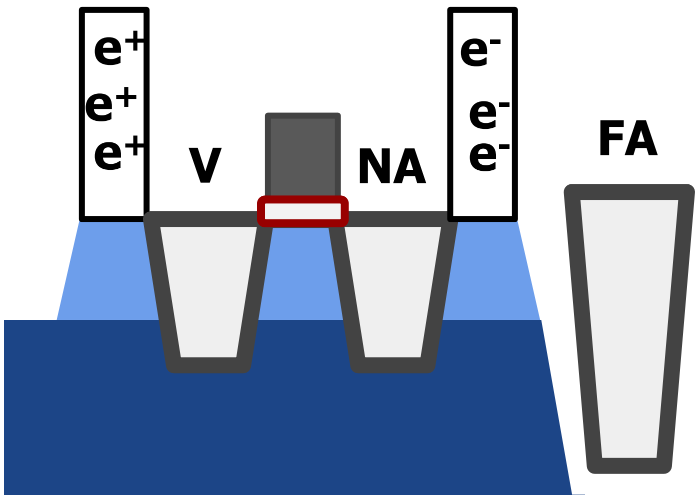
Contributing mechanisms

1. Increased subthreshold leakage due to AWL-VWL crosstalk
2. Electron migration & injection from A1
3. Passing gate effect from A2
4. "Normal" leakage as time passes by

Access Pattern - RowHammer

Half-Double

Many Far Aggressor (FA) activations followed by only a few
Near Aggressor (NA) activations causes bitflips in the Victim (V)



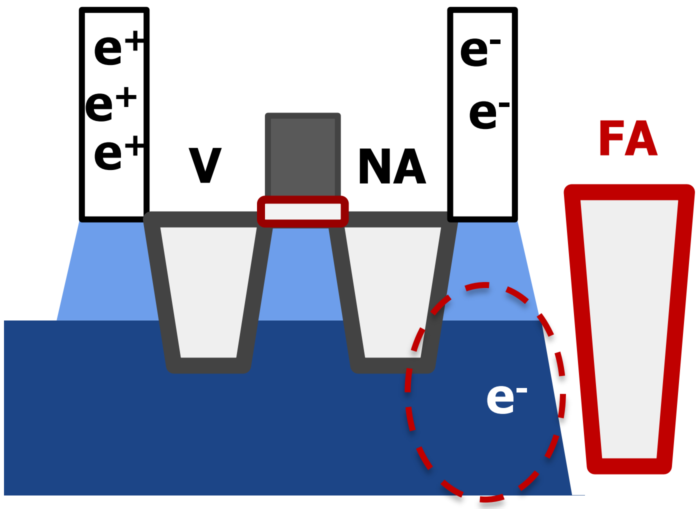
Hypothesized mechanisms

1. Frequent FA activation accumulates electrons near the NA side
2. Few NA activations causes those electrons to migrate and inject into V

Access Pattern - RowHammer

Half-Double

Many Far Aggressor (FA) activations followed by only a few Near Aggressor (NA) activations causes bitflips in the Victim (V)



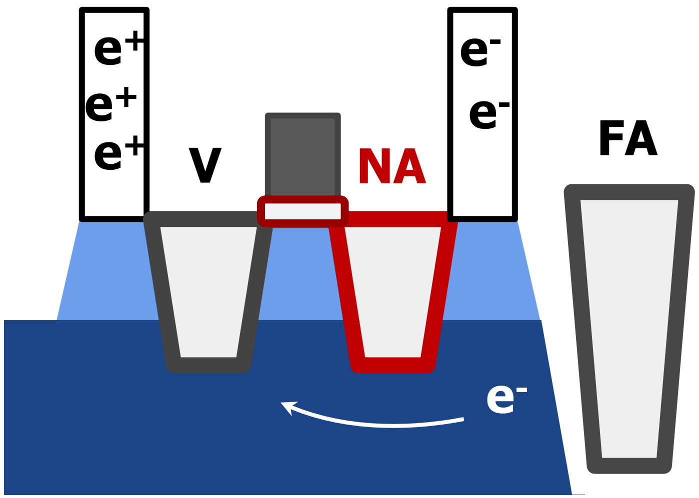
Hypothesized mechanisms

1. Frequent FA activation accumulates electrons near the NA side
2. Few NA activations causes those electrons to migrate and inject into V

Access Pattern - RowHammer

Half-Double

Many Far Aggressor (FA) activations followed by only a few
Near Aggressor (NA) activations causes bitflips in the Victim (V)



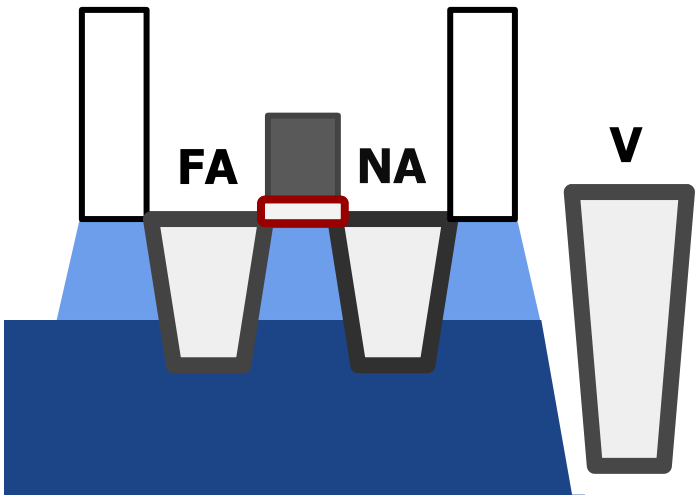
Hypothesized mechanisms

1. Frequent FA activation accumulates electrons near the NA side
2. Few NA activations causes those electrons to migrate and inject into V

Access Pattern - RowHammer

Half-Double

Another case…?



No hypothesized mechanisms so far…

Ongoing Works

ABACuS:
All-Bank Activation Counters for Scalable and Low Overhead RowHammer Mitigation

USENIX Security 2024

Ataberk Olgun

21.09.2023

SAFARI

ETH zürich

Executive Summary

Problem: RowHammer vulnerability worsens as DRAM becomes denser

- Existing defenses become more costly
- Benign workloads frequently trigger performance-degrading RowHammer mitigations

Goal: Prevent RowHammer bitflips at low performance, energy, and area cost

Key Observation: Workloads tend to access the same row in all DRAM banks at around the same time

Key Idea: Use one hardware counter to keep track of activation counts of the same row across all banks

- Make high-performance, area-hungry counter-based mechanisms practical

Key Results: Memory system simulations using 62 single core and 62 8-core workloads

At all tested RowHammer thresholds (1000, 500, 250 125):

Faster than the lowest-area-cost counter-based defense mechanism
Smaller than the lowest-performance-overhead counter-based defense mechanism

0.59% avg. performance overhead (single-core) at a future RowHammer threshold (1K)

- Only 9.79 KiB on-chip storage per DRAM rank (0.02% of a Xeon processor)

1.52% avg. performance overhead (single-core) at an ultra-low threshold (125)

- 75.70 KiB on-chip storage per DRAM rank (0.11% of the Xeon processor)

RowHammer in HBM Chips (2023)

- Ataberk Olgun, Majd Osserian, A. Giray Yağlıkçı, Yahya Can Tugrul, Haocong Luo, Steve Rhyner, Behzad Salami, Juan Gomez-Luna, and Onur Mutlu,
  "An Experimental Analysis of RowHammer in HBM2 DRAM Chips"
  *Proceedings of the 53nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Disrupt Track (**DSN Disrupt**)*, Porto, Portugal, June 2023.
  [arXiv version]
  [Slides (pptx) (pdf)]
  [Talk Video (24 minutes, including Q&A)]

An Experimental Analysis of RowHammer in HBM2 DRAM Chips

Ataberk Olgun[1] Majd Osseiran[1,2] A. Giray Yağlıkçı[1] Yahya Can Tuğrul[1]
Haocong Luo[1] Steve Rhyner[1] Behzad Salami[1] Juan Gomez Luna[1] Onur Mutlu[1]
[1]*SAFARI Research Group, ETH Zürich* [2]*American University of Beirut*

https://arxiv.org/pdf/2305.17918.pdf

Executive Summary

Motivation: HBM chips have new architectural characteristics (e.g., 3D-stacked dies) that might affect the RowHammer vulnerability in various ways

Understanding RowHammer enables designing effective and efficient solutions

Problem: No prior study demonstrates the RowHammer vulnerability in HBM

Goal: Experimentally analyze how vulnerable HBM DRAM chips are to RowHammer

Experimental Study: Detailed experimental characterization of RowHammer in a modern HBM2 DRAM chip. Our study provides two main findings:

1. Spatial variation of RowHammer vulnerability

- Different channels in a 3D-stacked HBM chip exhibit different RowHammer vulnerability
- DRAM rows near the end of a DRAM bank are more RowHammer resilient

2. On-DRAM-die RowHammer mitigations

- A modern HBM chip implements undisclosed on-DRAM-die RowHammer mitigation
- The mitigation refreshes a victim row after every 17 periodic refresh operations (e.g., similar to DDR4 chips)

Ongoing and Future Work

Discover New Bitflips

Fundamentally Fix Them

To Build More Robust Systems for Future

Removed Slides

How Reliable/Secure/Safe is This Bridge?

Source: http://www.technologystudent.com/struct1/tacom1.png

Collapse of the "Galloping Gertie"

Source: AP
http://www.wsdot.wa.gov/tnbhistory/connections/connections3.htm

How Safe & Secure Are These People?

Security is about preventing unforeseen consequences

Source: https://s-media-cache-ak0.pinimg.com/originals/48/09/54/4809543a9c7700246a0cf8acdae27abf.jpg

How Safe & Secure Are Our Platforms?

Security is about preventing unforeseen consequences

Source: https://taxistartup.com/wp-content/uploads/2015/03/UK-Self-Driving-Cars.jpg

Collapse of the "Galloping Gertie" (1940)

Source: AP
http://www.wsdot.wa.gov/tnbhistory/connections/connections3.htm

Another Example (1994)

Source: By 최광모 - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=35197984

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)

Source: AFP / Valery HACHE, https://www.capitalfm.co.ke/news/2018/08/genoa-bridge-collapse-what-we-know/

A Most Recent Example (2022)

https://www.post-gazette.com/local/city/2022/01/28/pittsburgh-bridge-collapse-forbes-braddock-avenue-point-breeze-squirrel-hill/stories/202201280075

A Most Recent Example (2022)

https://www.reuters.com/world/us/biden-highlight-manufacturing-jobs-gdp-growth-pittsburgh-2022-01-28/

A Most Recent Example (2022)

https://usa.streetsblog.org/2022/01/28/pittsburgh-bridge-collapse-underscores-urgent-need-for-fix-it-first-policy/

A Most Recent Example (2022)

https://www.npr.org/2022/01/28/1076343656/pittsburgh-bridge-collapse-biden-visit