Memory Systems and Memory-Centric Computing

Lecture 5: Memory Robustness II

Onur Mutlu

omutlu@gmail.com

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

19 July 2024

HiPEAC ACACES Summer School 2024

SAFARI ETH zürich

Agenda For Today

- Memory Systems and Memory-Centric Computing
  - July 15-19, 2024

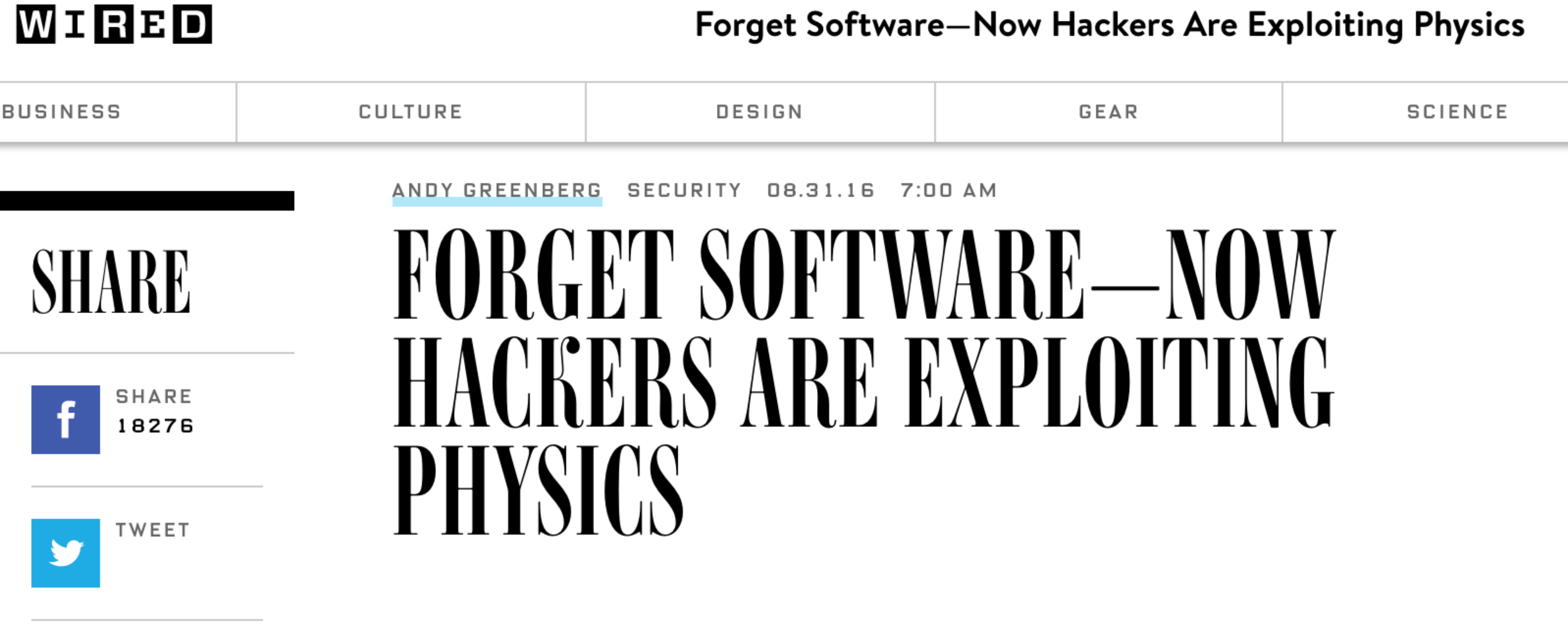
- Topic 1: Memory Trends, Challenges, Opportunities, Basics
- Topic 2: Memory-Centric Computing
- Topic 3: Memory Robustness: RowHammer, RowPress & Beyond
- Topic 4: Machine Learning Driven Memory Systems
- Topic 5 (another course): Architectures for Genomics and ML
- Topic 6 (unlikely): Non-Volatile Memories and Storage
- Topic 7 (unlikely): Memory Latency, Predictability & QoS

- Major Overview Reading:
  - Mutlu et al., "A Modern Primer on Processing in Memory," Book Chapter on Emerging Computing and Devices, 2022.

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

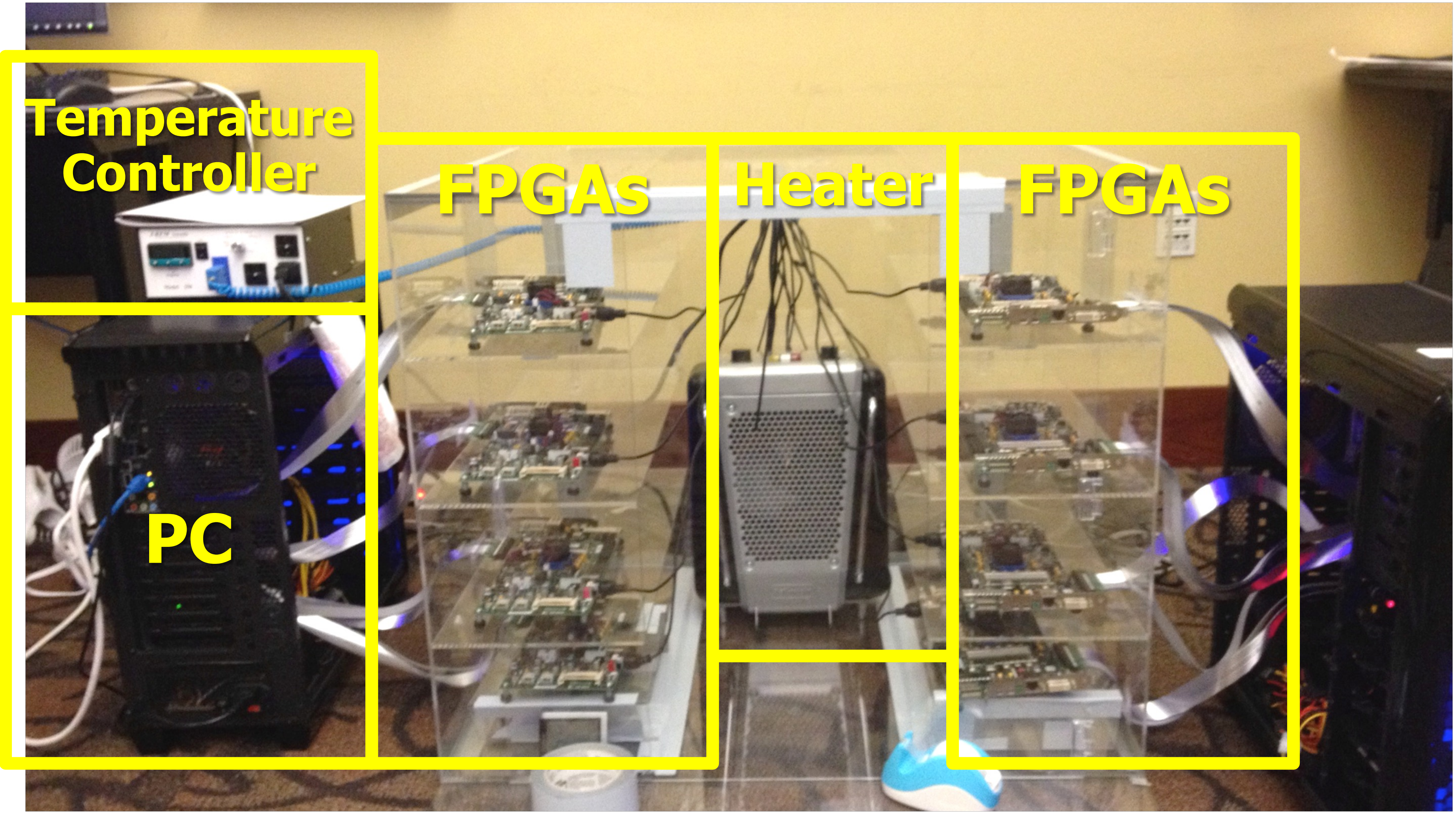


Forget Software—Now Hackers Are Exploiting Physics

ANDY GREENBERG SECURITY 08.31.16 7:00 AM

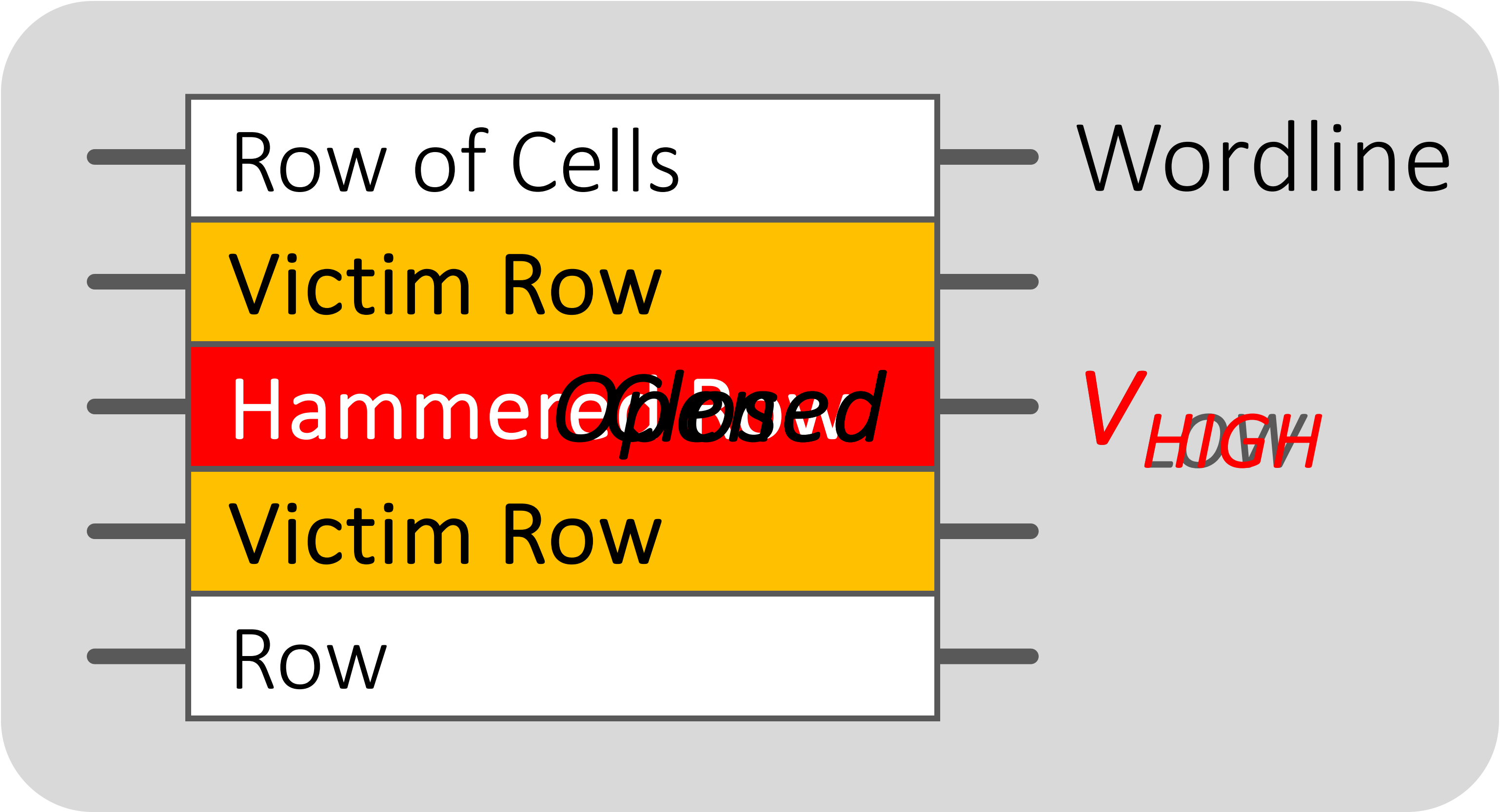
FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

Infrastructures to Understand Such Issues



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

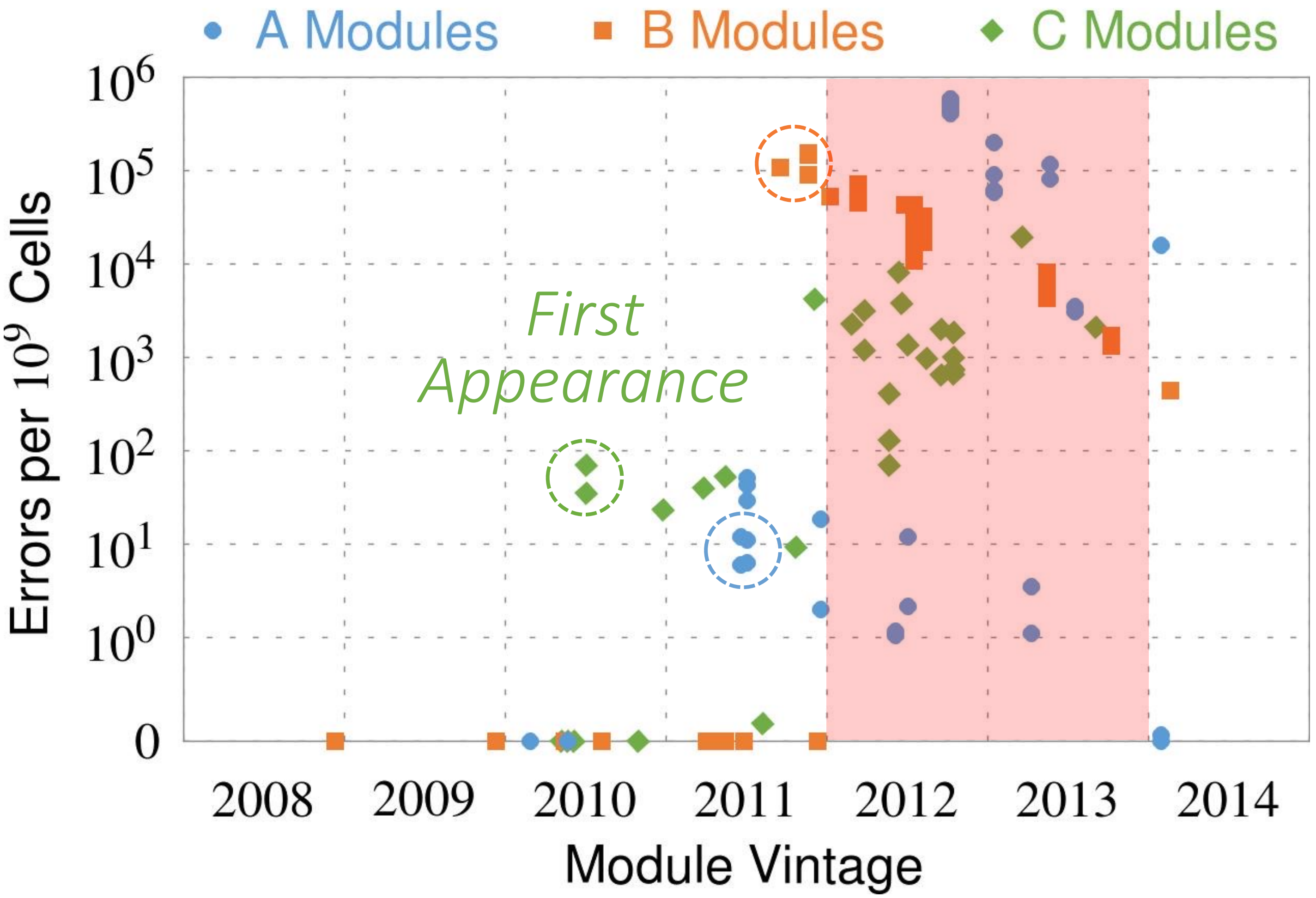
Modern DRAM is Prone to Disturbance Errors



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)

Recent DRAM Is More Vulnerable



All modules from 2012–2013 are vulnerable

Why Is This Happening?

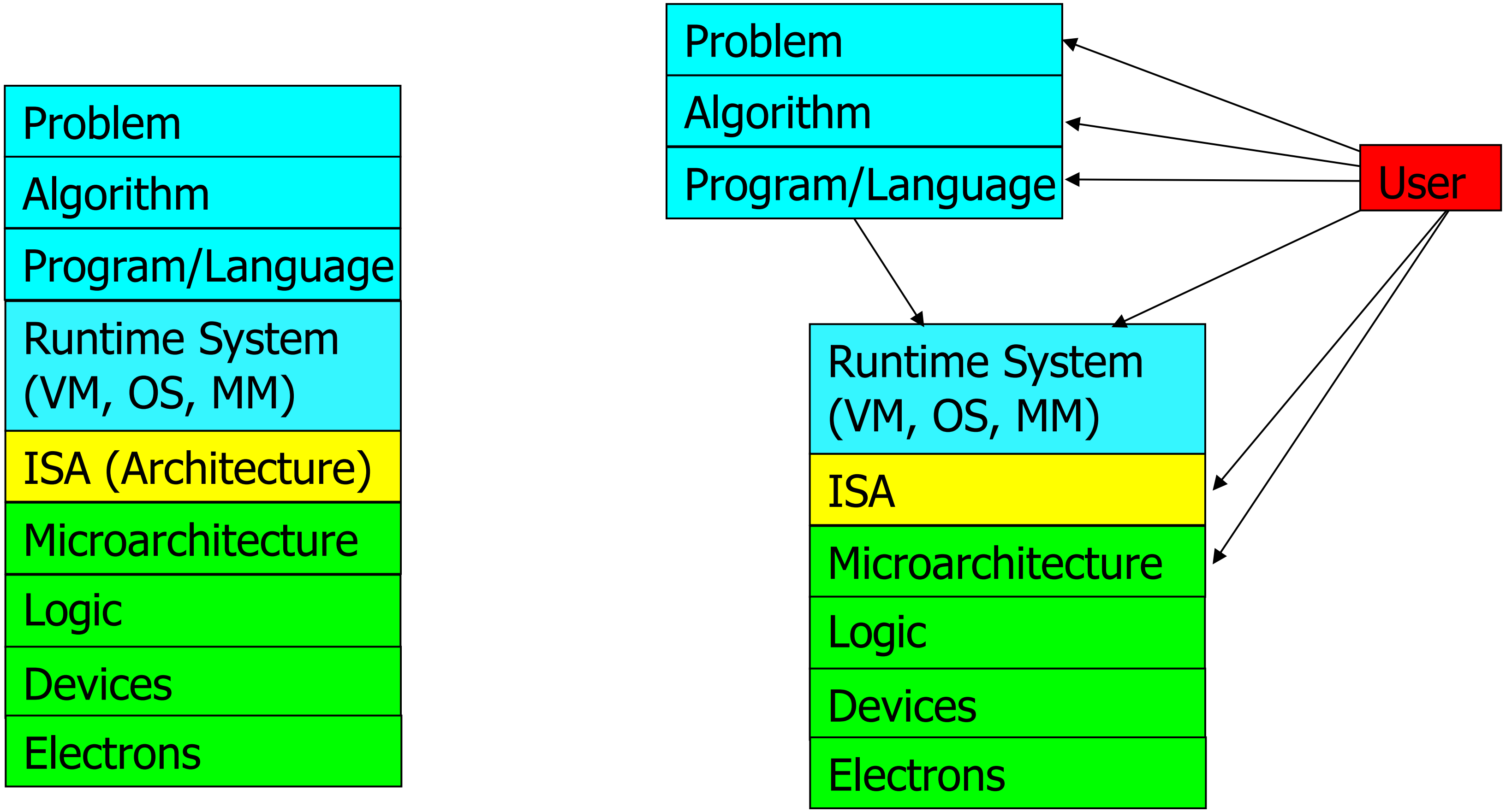
- DRAM cells are too close to each other!
  - They are not electrically isolated from each other

- 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

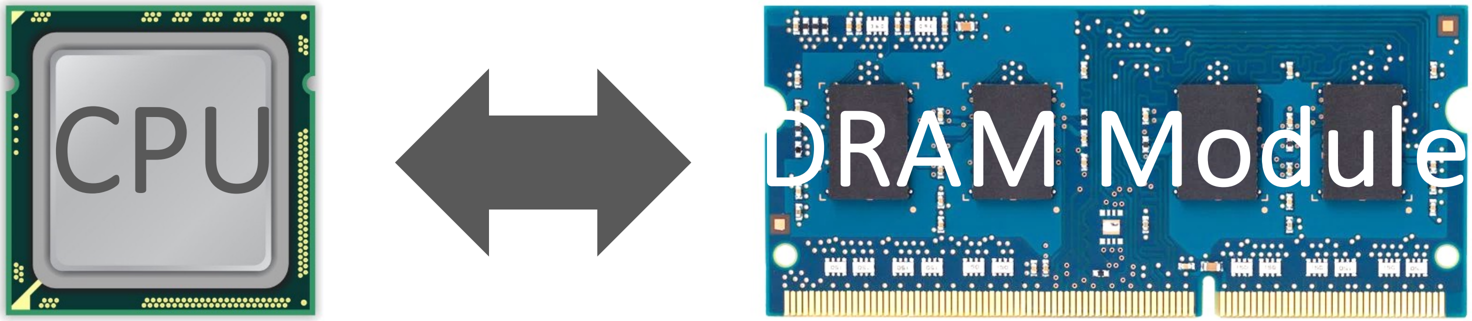
- Example: When we activate (apply high voltage) to a row, an adjacent row gets slightly activated as well
  - Vulnerable cells in that slightly-activated row lose a little bit of charge
  - If RowHammer happens enough times, charge in such cells gets drained

Higher-Level Implications

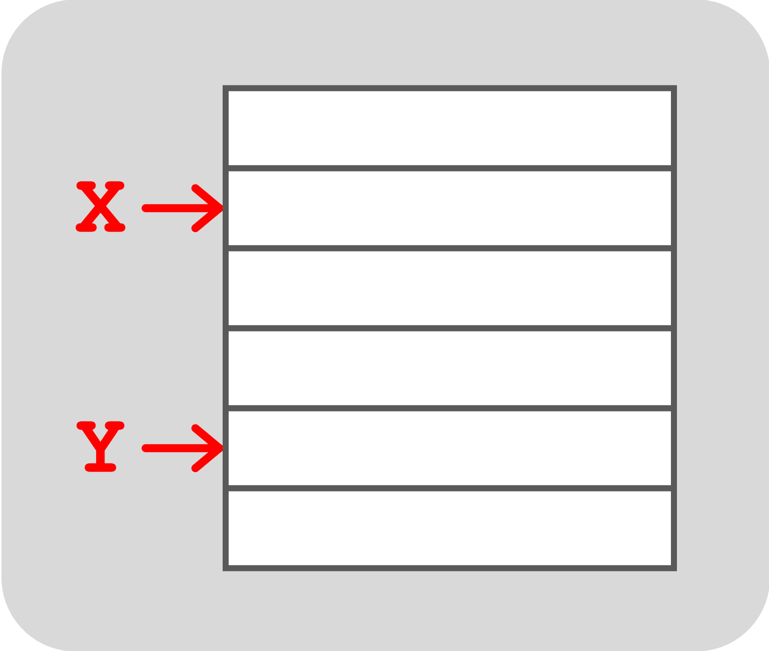
- This simple circuit level failure mechanism has enormous implications on upper layers of the transformation hierarchy



A Simple Program Can Induce Many Errors

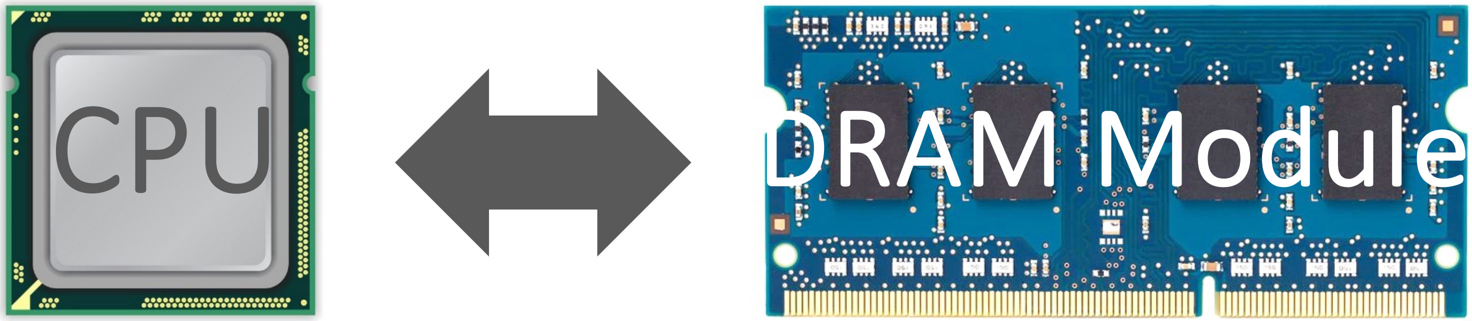


```
loop:
  mov (X), %eax
  mov (Y), %ebx
  clflush (X)
  clflush (Y)
  mfence
  jmp loop
```



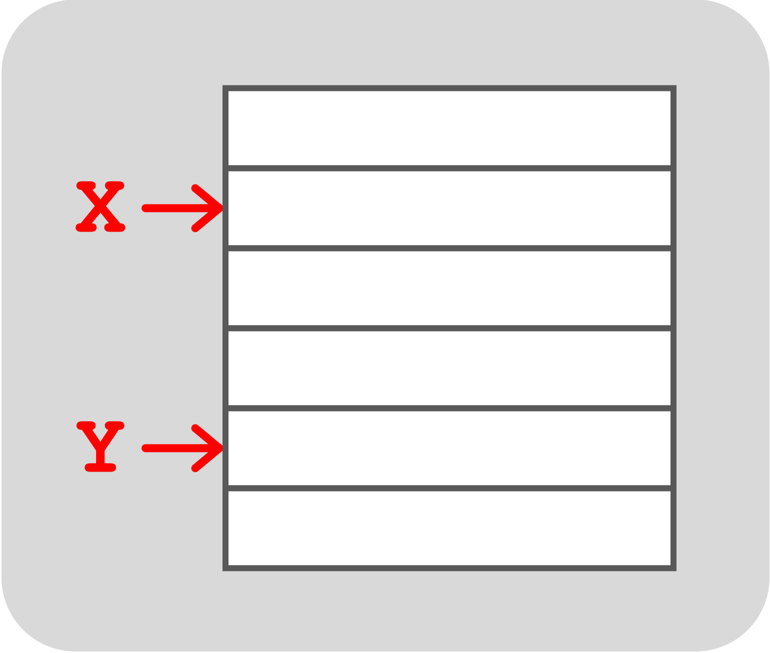
Download from: https://github.com/CMU-SAFARI/rowhammer

A Simple Program Can Induce Many Errors



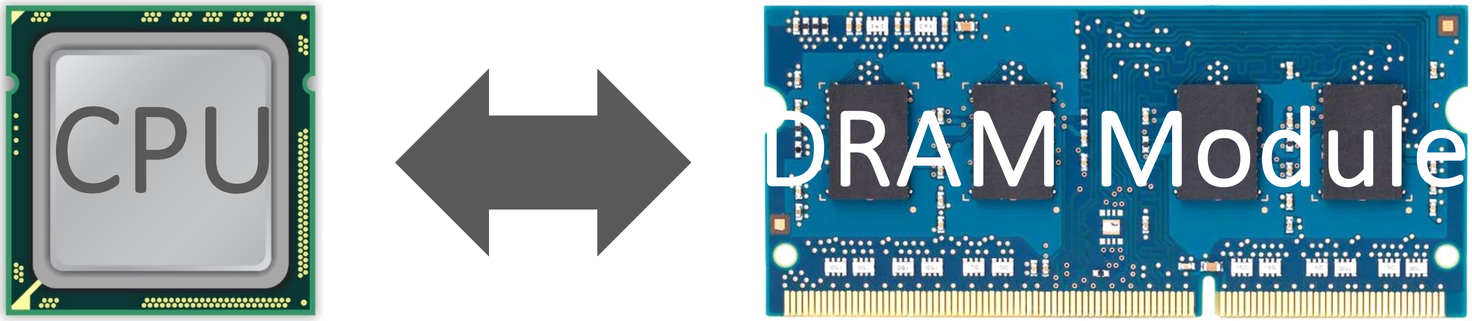
1. Avoid ***cache hits***
   – Flush **X** from cache

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

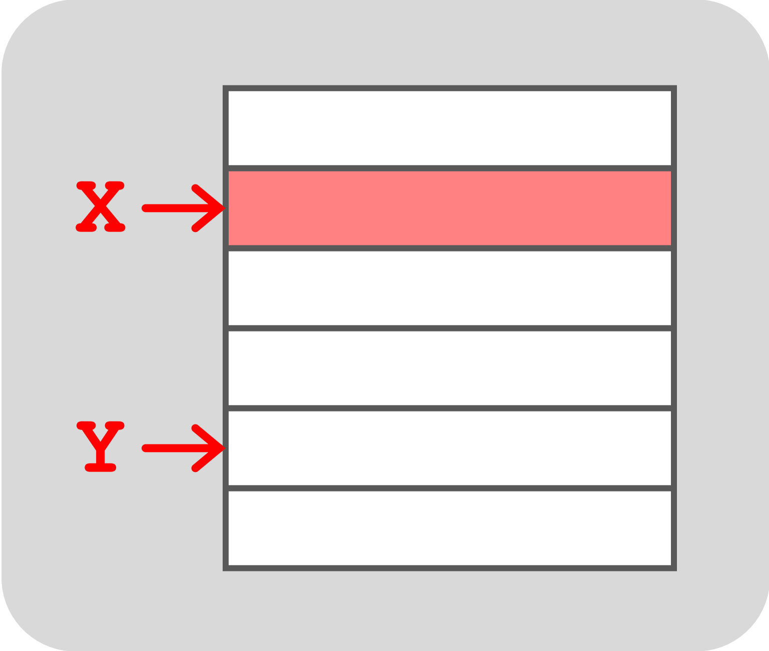


Download from: https://github.com/CMU-SAFARI/rowhammer

A Simple Program Can Induce Many Errors

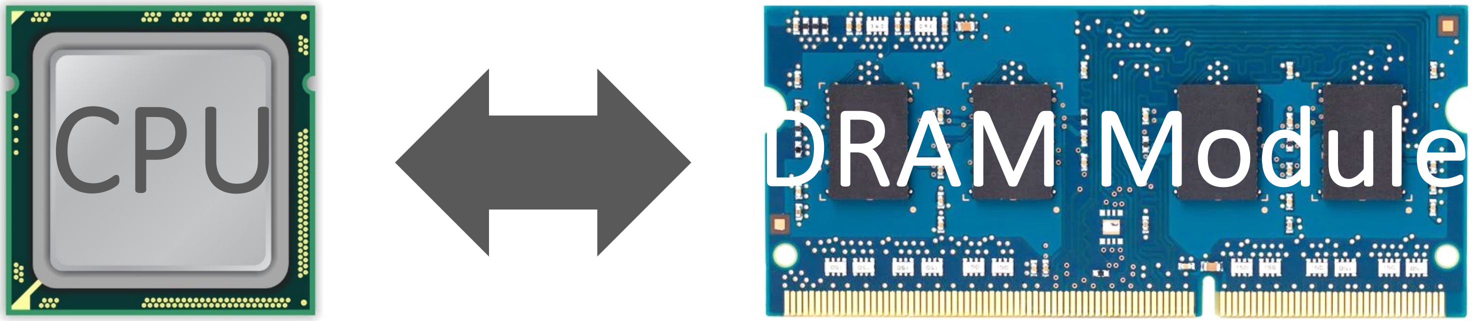


```
loop:
   mov  (X), %eax
   mov  (Y), %ebx
   clflush (X)
   clflush (Y)
   mfence
   jmp loop
```

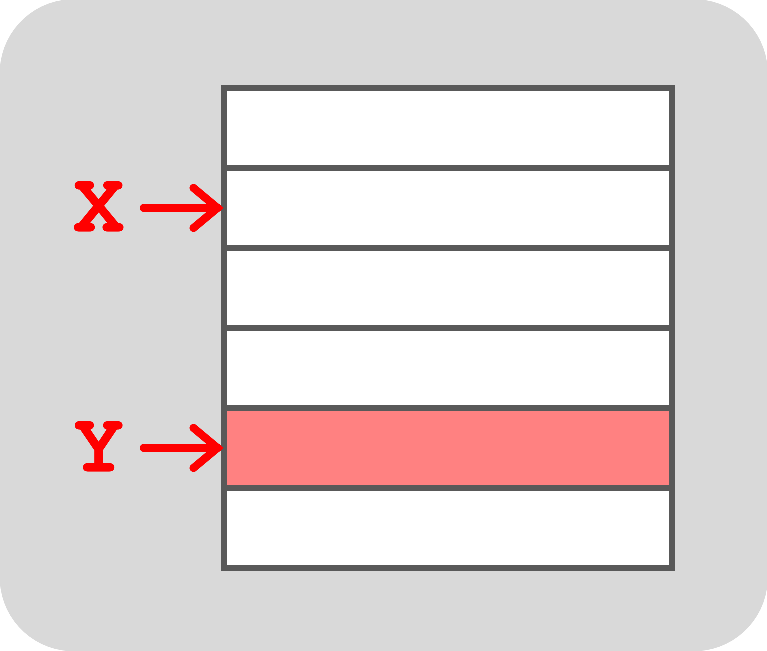


Download from: https://github.com/CMU-SAFARI/rowhammer

A Simple Program Can Induce Many Errors

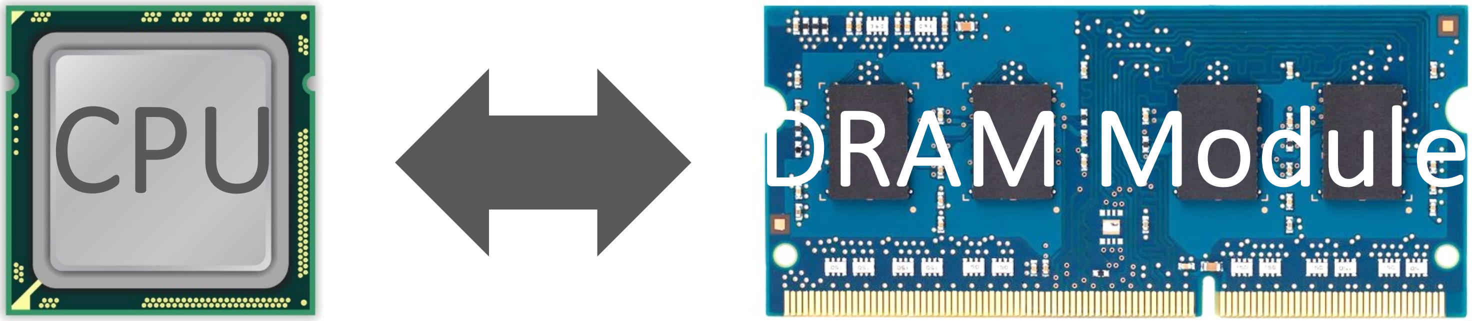


```
loop:
   mov (X), %eax
   mov (Y), %ebx
   clflush (X)
   clflush (Y)
   mfence
   jmp loop
```

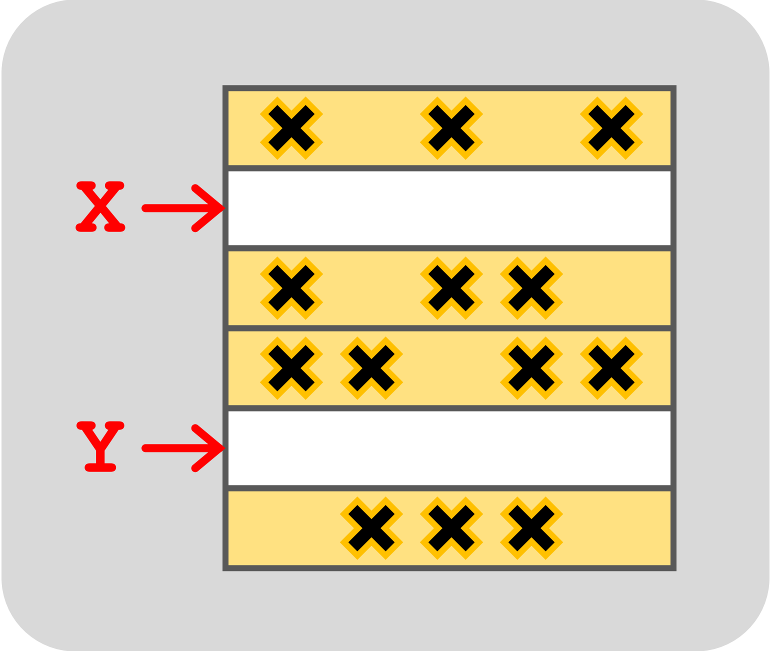


Download from: https://github.com/CMU-SAFARI/rowhammer

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

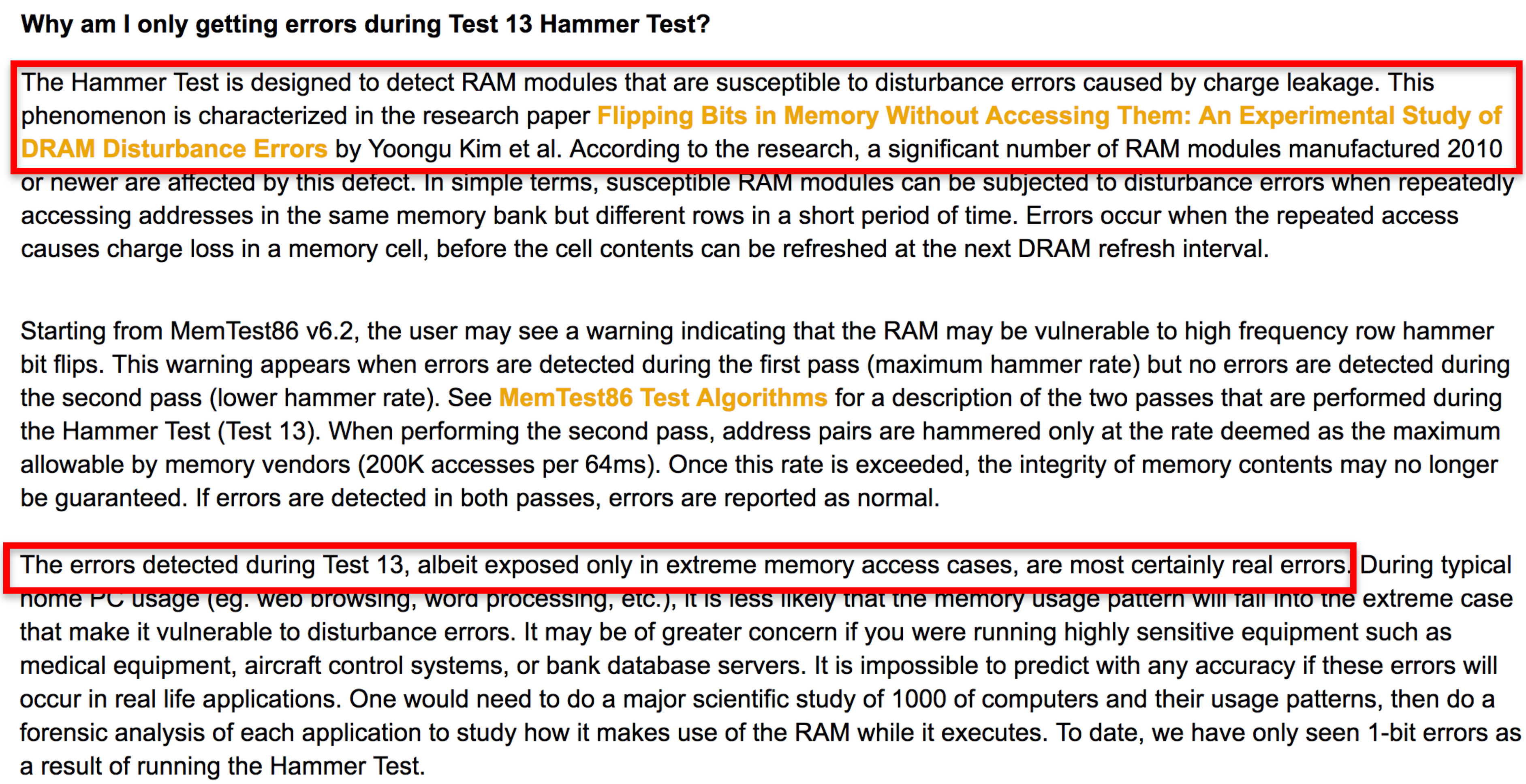
A real robustness issue

(including reliability, security, safety)

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

First Adopters: Memory Testing Software

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



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

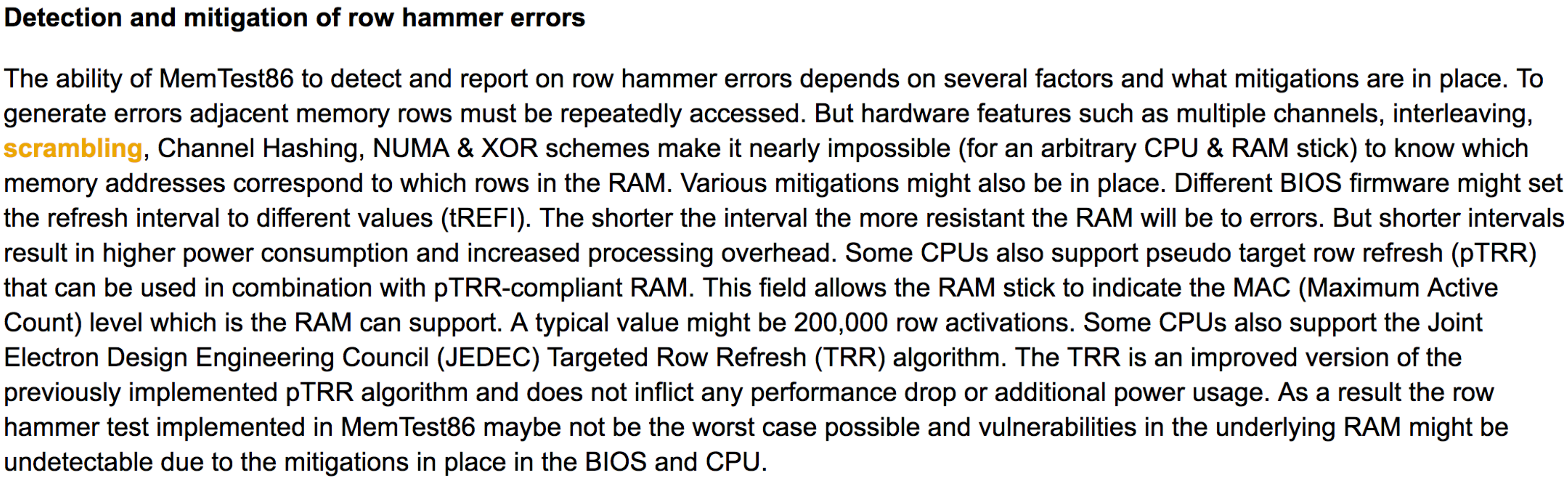
The Hammer Test is designed to detect RAM modules that are susceptible to disturbance errors caused by charge leakage. This phenomenon is characterized in the research paper **Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors** by Yoongu Kim et al. According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

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

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

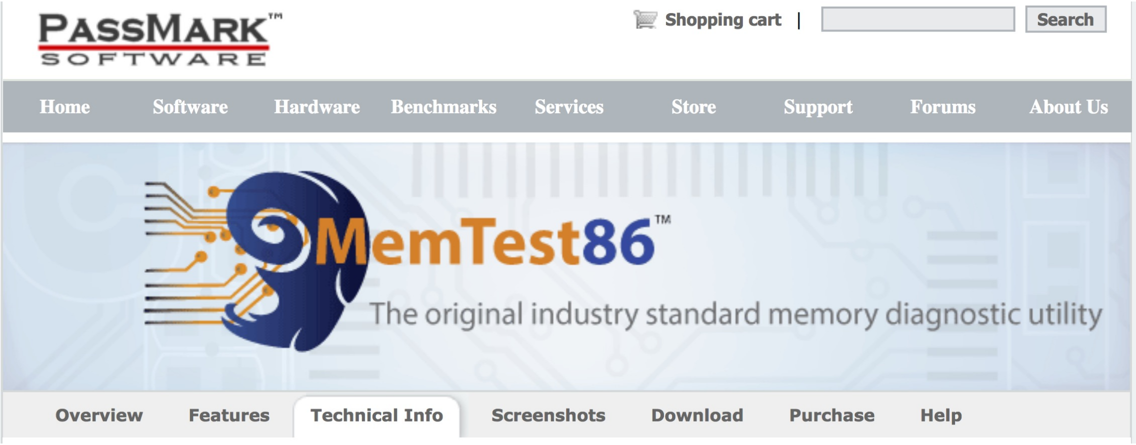
First Adopters: Memory Testing Software

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

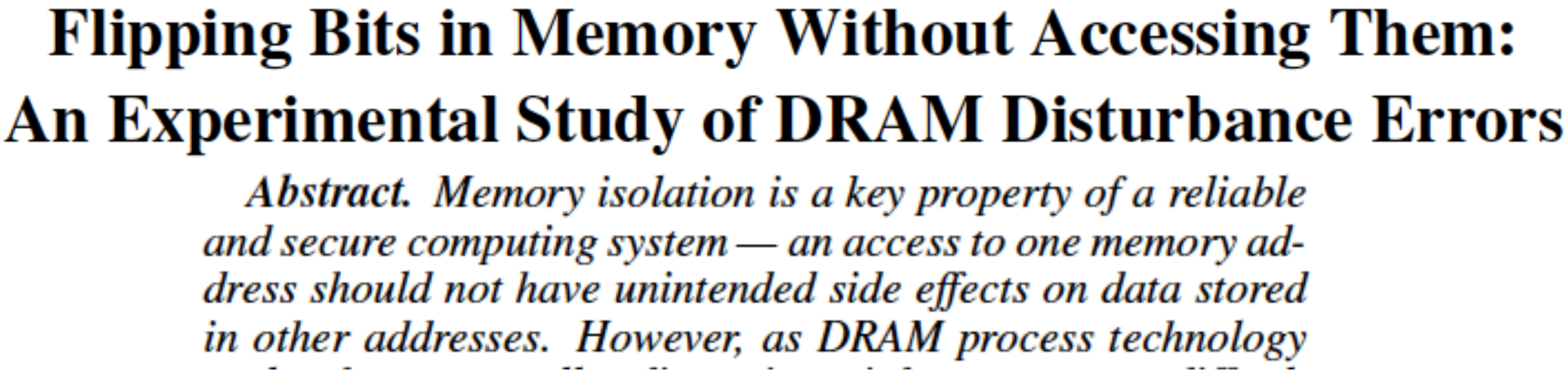


Detection and mitigation of row hammer errors

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

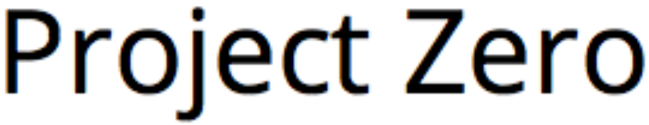


One Can Take Over an Otherwise-Secure System



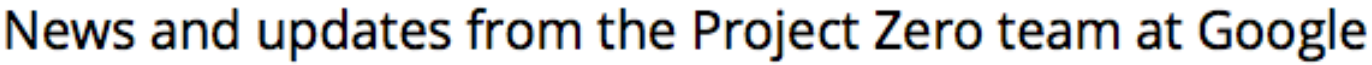
Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors

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*



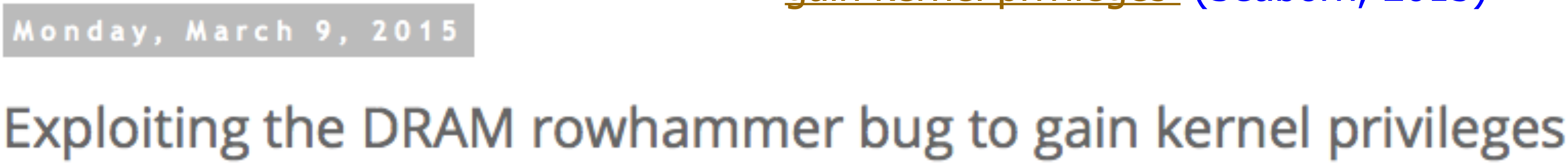
Project Zero

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

Many RowHammer Security Exploits

- One can exploit RowHammer to
- Take over a system
- Read data they do not have access to
- Break out of virtual machine sandboxes
- Corrupt important data → e.g., render ML inference useless
- Steal secret data (e.g., crypto keys & ML model parameters)

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)

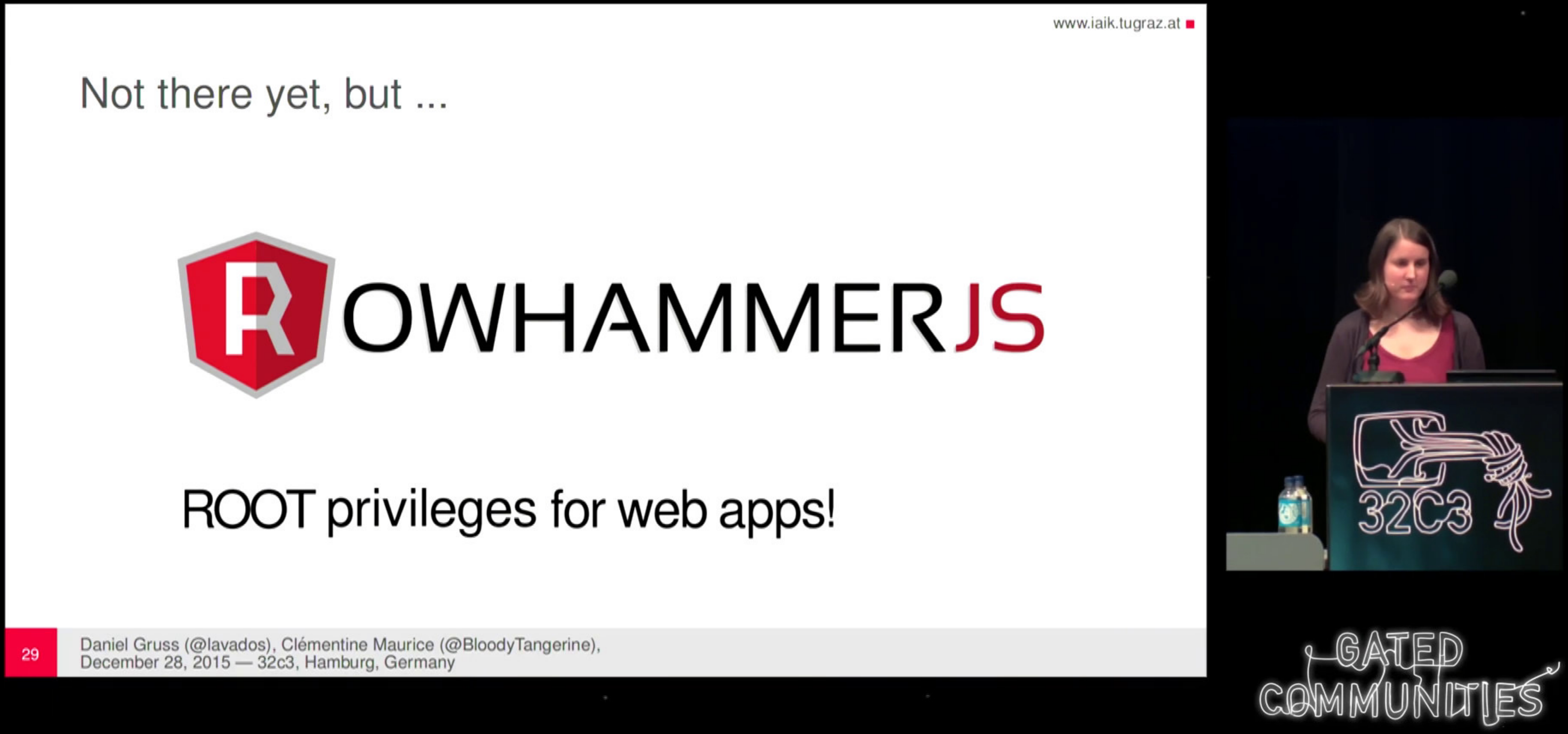
Security Implications

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.”



www.iaik.tugraz.at

Not there yet, but ...

ROWHAMMERJS

ROOT privileges for web apps!

29

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

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

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

More Security Implications (II)

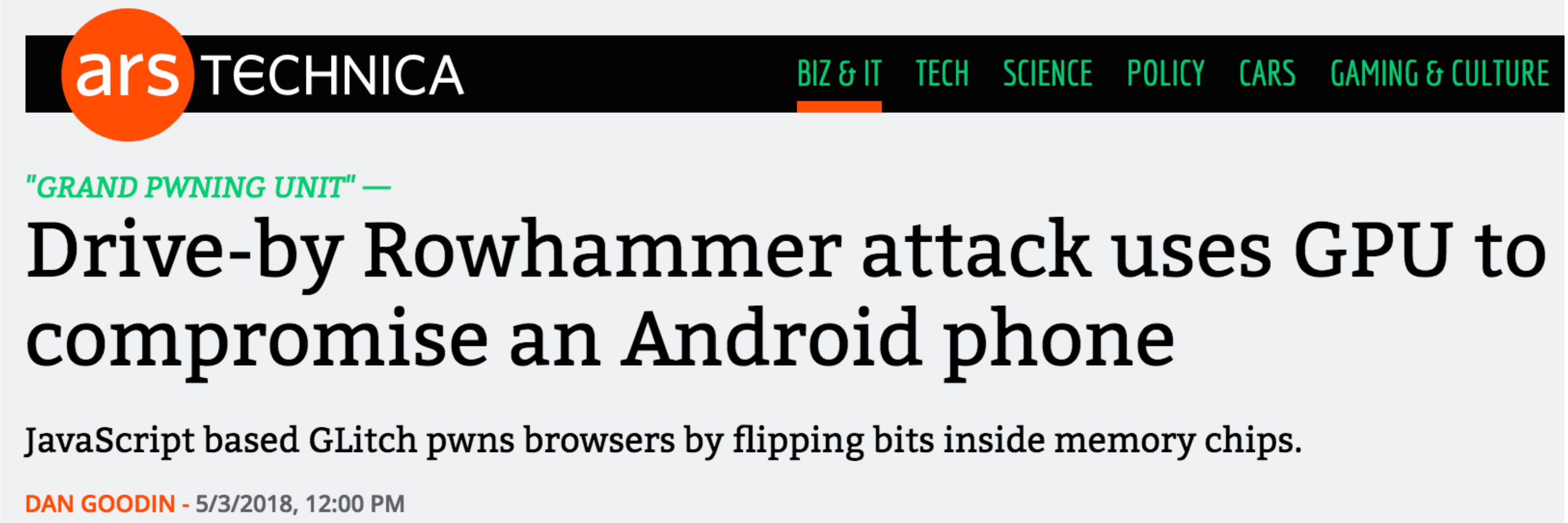
"Can gain control of a smart phone deterministically"

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

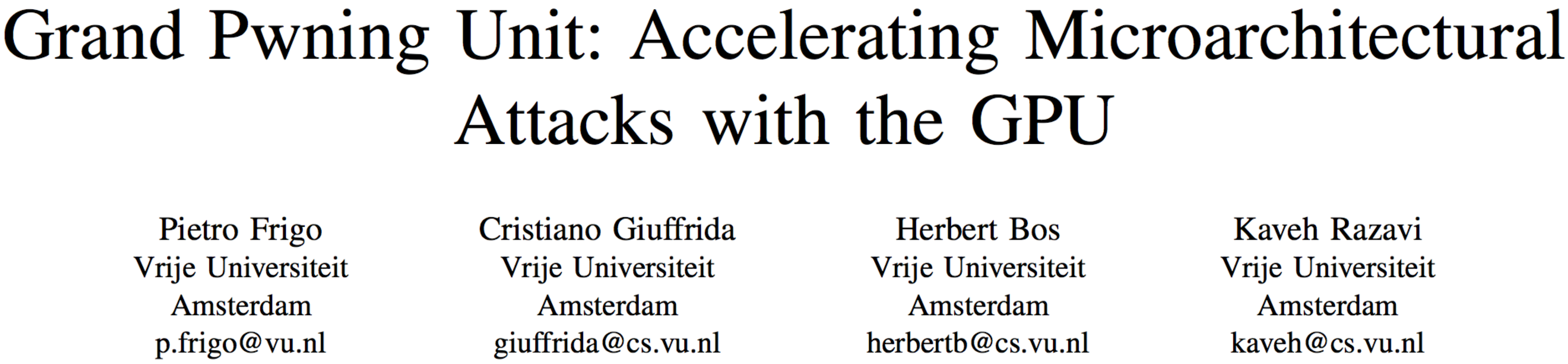


"GRAND PWNING UNIT" —

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

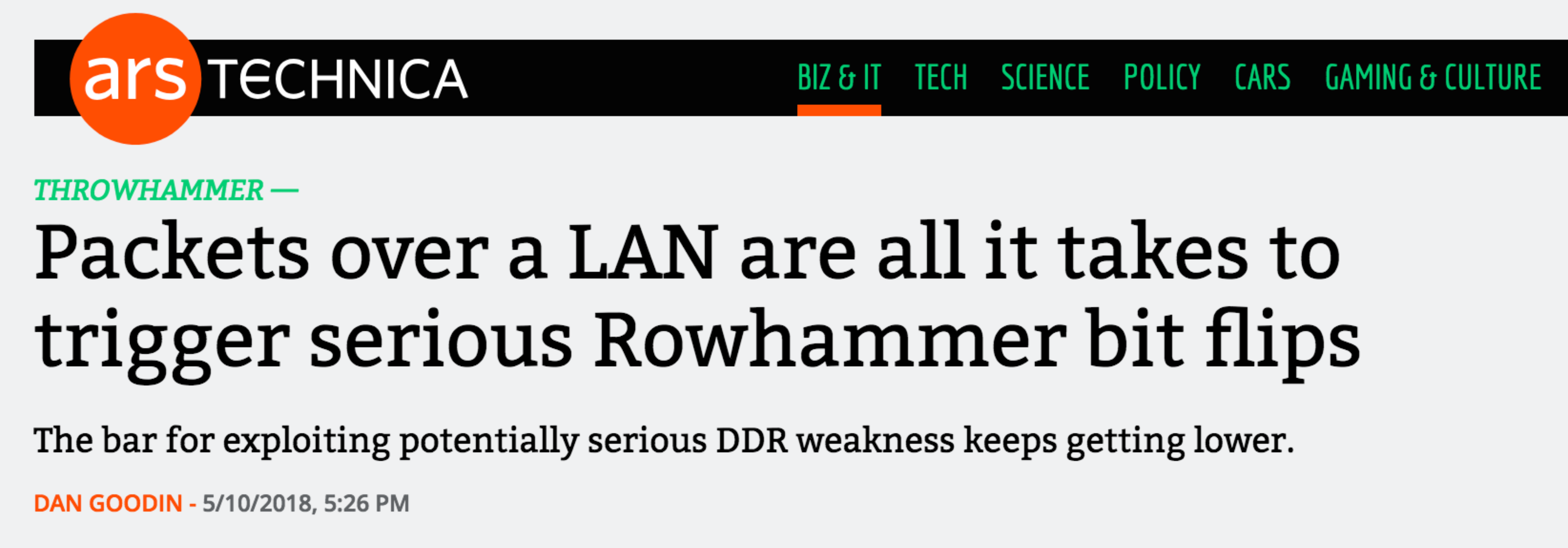
Cristiano Giuffrida
Vrije Universiteit Amsterdam
giuffrida@cs.vu.nl

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

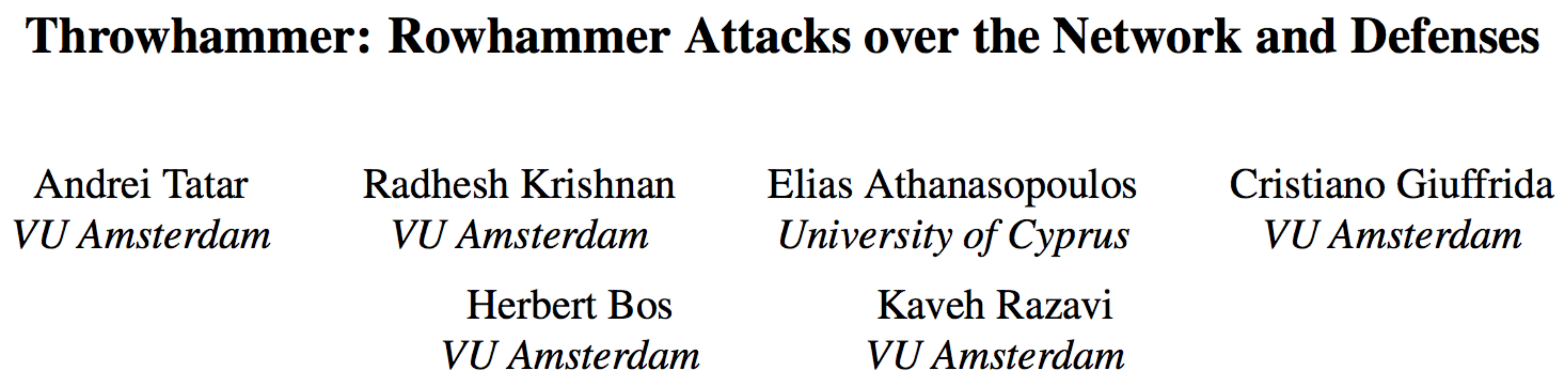


THROWHAMMER —

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*

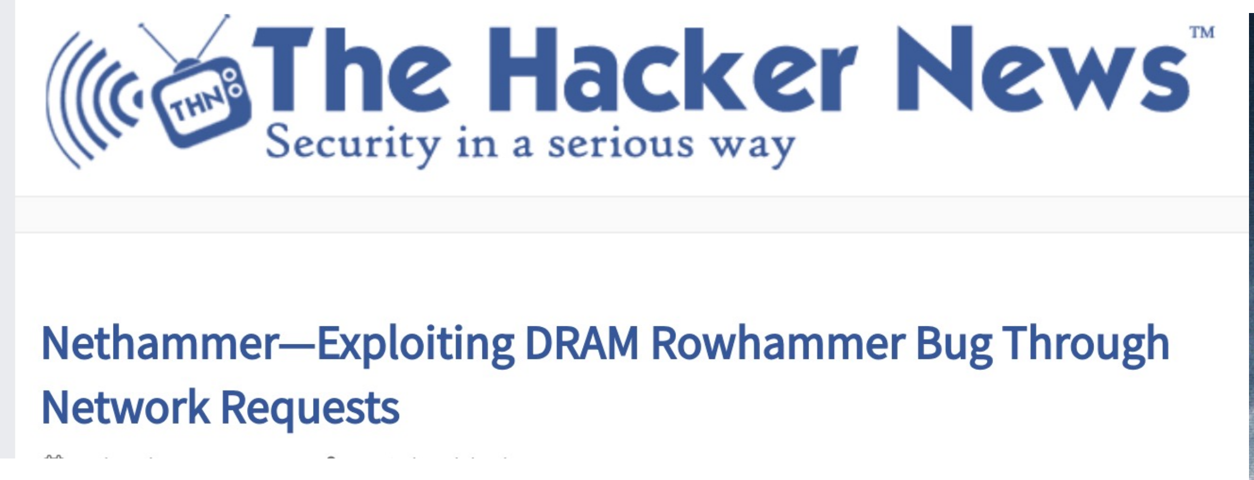
Cristiano Giuffrida *VU Amsterdam*

Herbert Bos *VU Amsterdam*

Kaveh Razavi *VU Amsterdam*

More Security Implications (V)

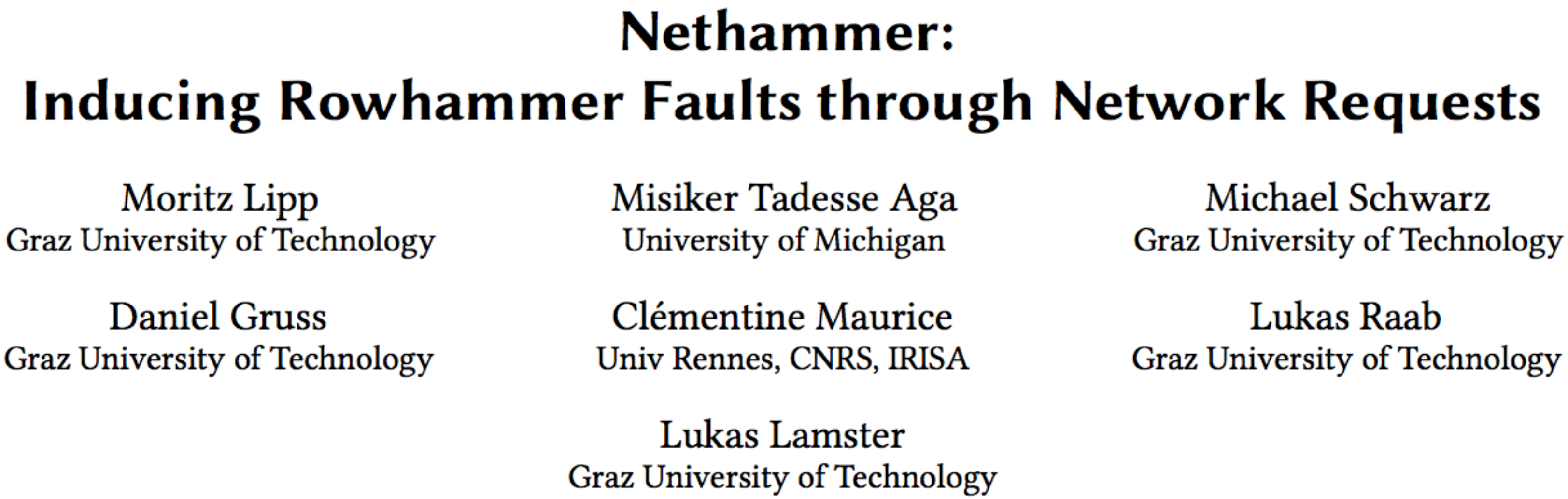
- Rowhammer over RDMA (II)



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

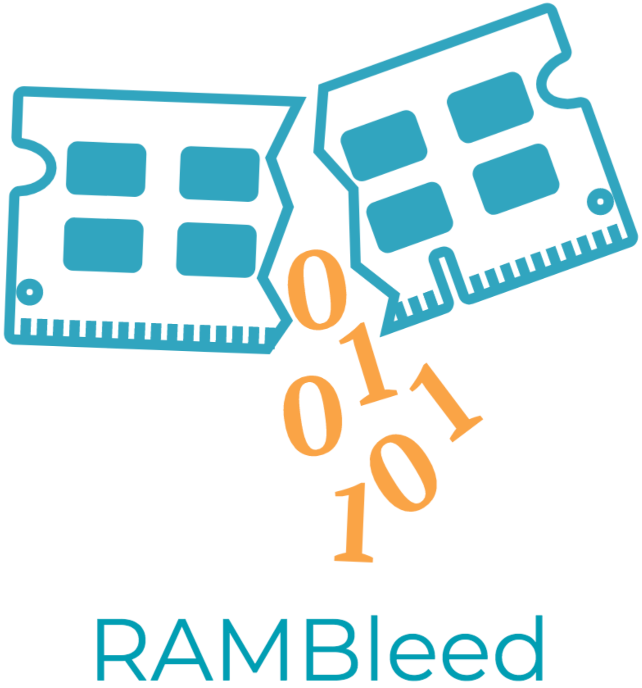
Clémentine Maurice
Univ Rennes, CNRS, IRISA

Lukas Raab
Graz University of Technology

Lukas Lamster
Graz University of Technology

More Security Implications (VI)

- IEEE S&P 2020



RAMBleed: Reading Bits in Memory Without Accessing Them

Andrew Kwong
University of Michigan
ankwong@umich.edu

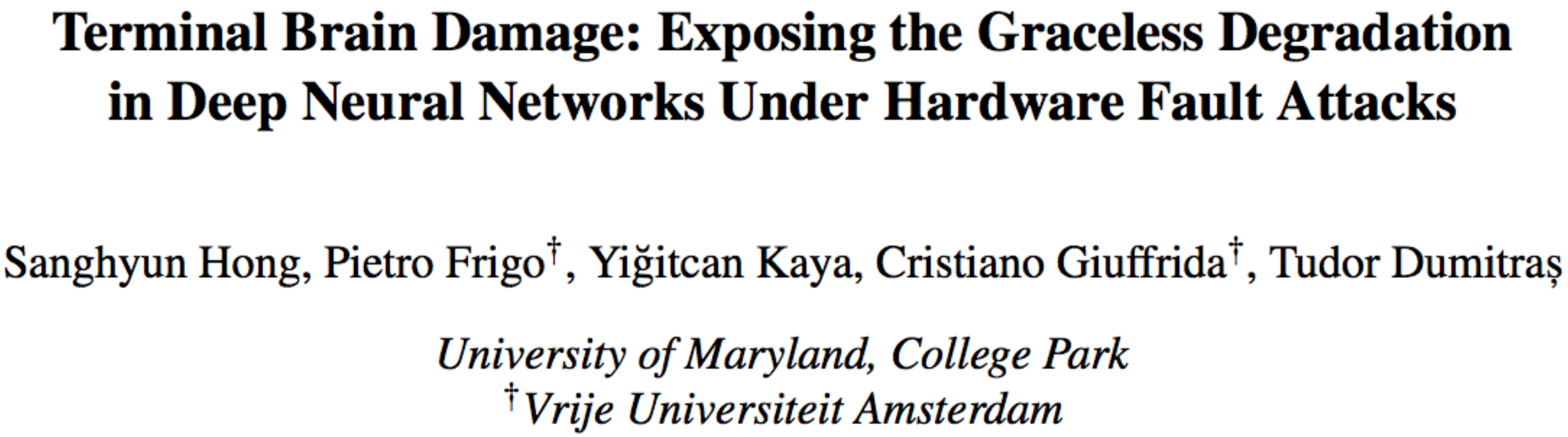
Daniel Genkin
University of Michigan
genkin@umich.edu

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)

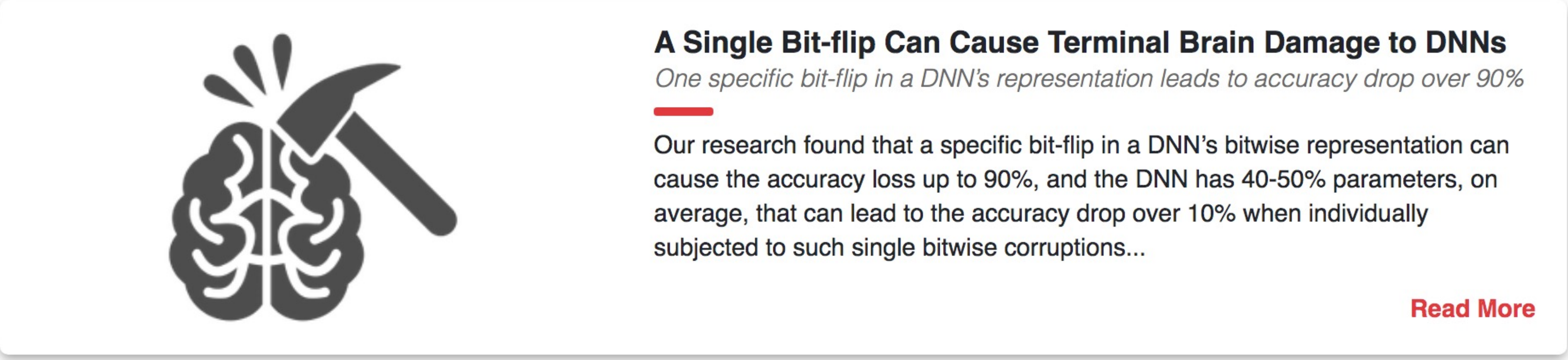
- USENIX Security 2019



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

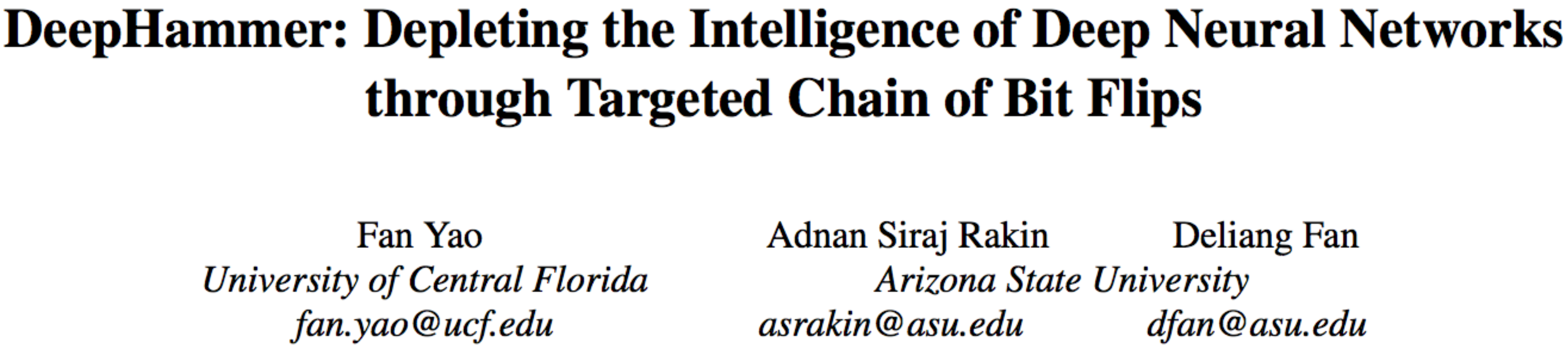
One specific bit-flip in a DNN's representation leads to accuracy drop over 90%

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

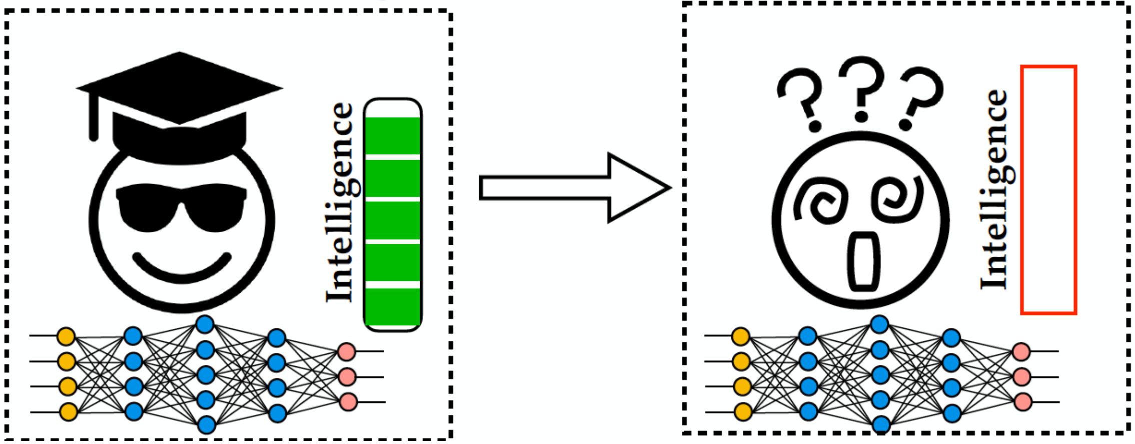
Fan Yao
University of Central Florida
fan.yao@ucf.edu

Adnan Siraj Rakin
Deliang Fan
Arizona State University
asrakin@asu.edu
dfan@asu.edu

Degrade the **inference accuracy** to the level of **Random Guess**

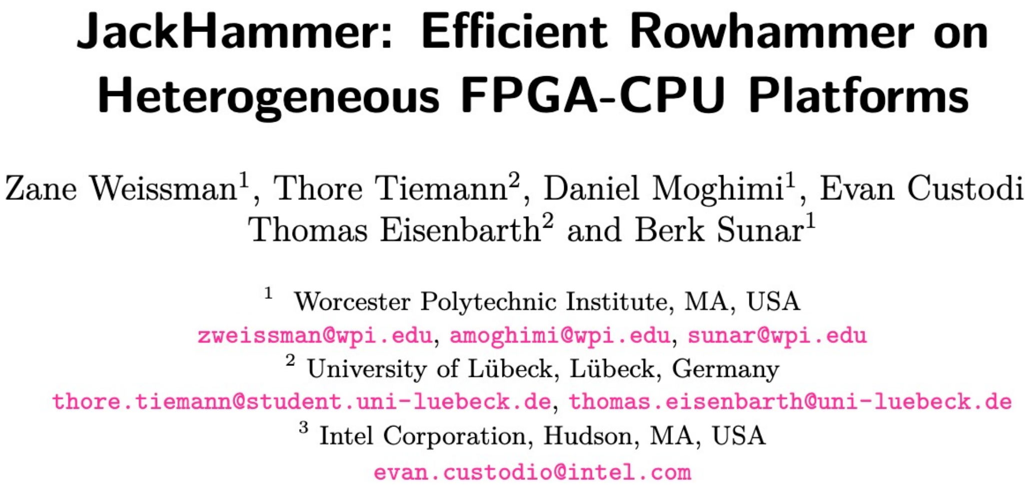
Example: ResNet-20 for CIFAR-10, **10** output classes

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



More Security Implications (IX)

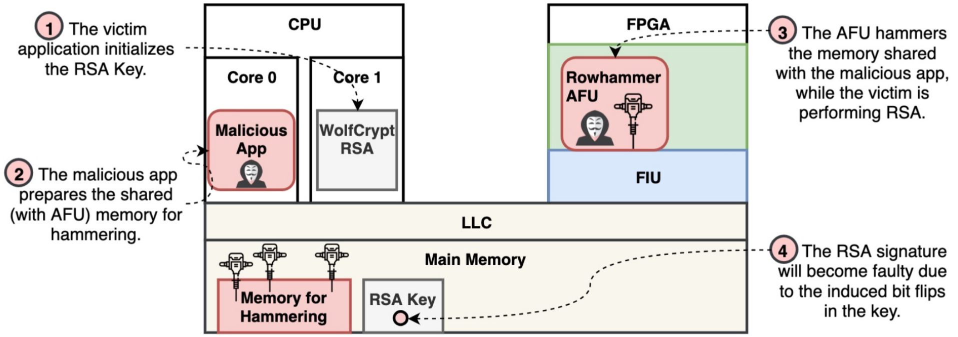
- CHES 2020



JackHammer: Efficient Rowhammer on Heterogeneous FPGA-CPU Platforms

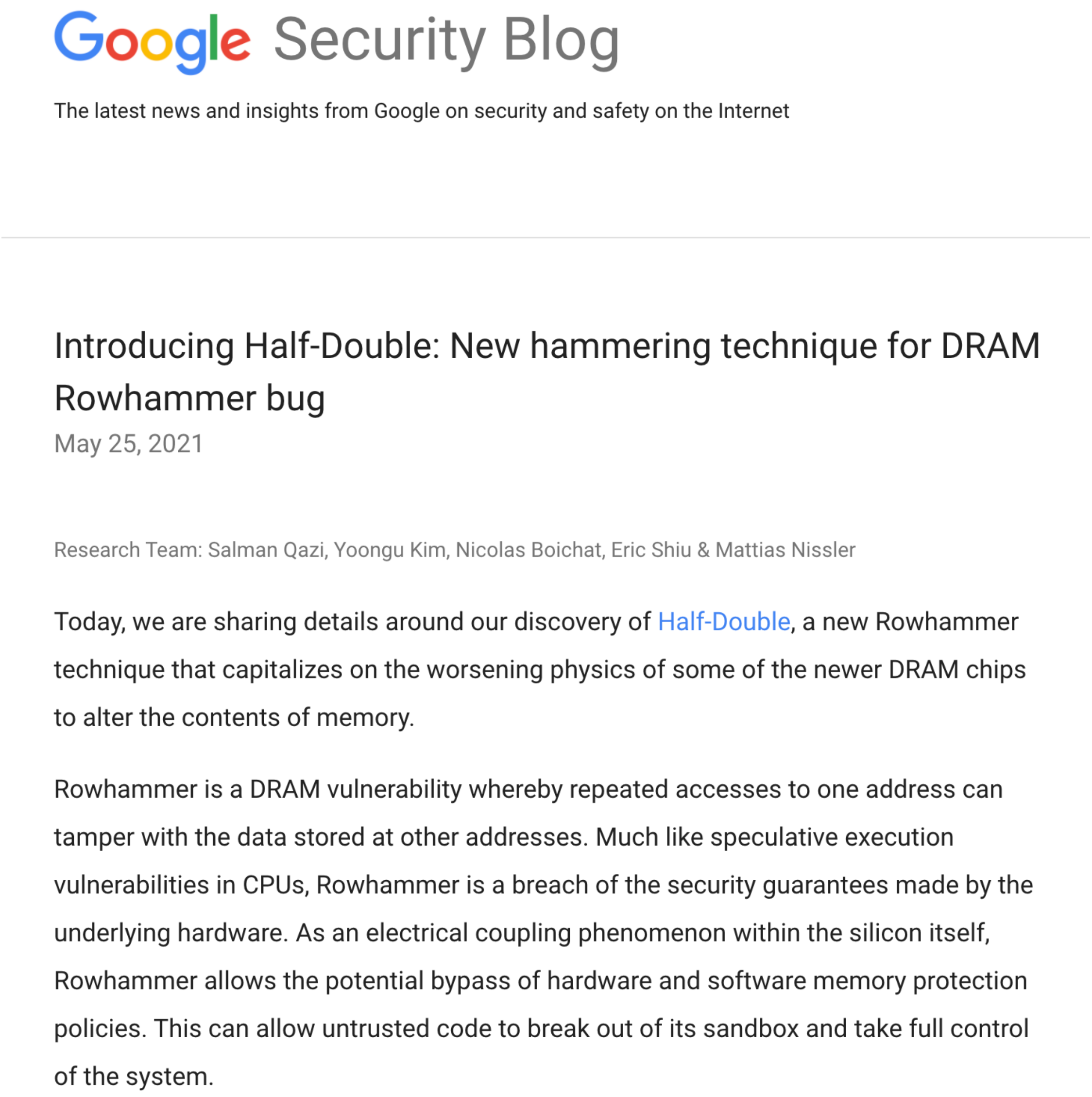
Zane Weissman[1], Thore Tiemann[2], Daniel Moghimi[1], Evan Custodio[3],
Thomas Eisenbarth[2] and Berk Sunar[1]

[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


1 The victim application initializes the RSA Key.
2 The malicious app prepares the shared (with AFU) memory for hammering.
CPU
Core 0
Core 1
Malicious App
WolfCrypt RSA
FPGA
Rowhammer AFU
FIU
LLC
Main Memory
Memory for Hammering
RSA Key
3 The AFU hammers the memory shared with the malicious app, while the victim is performing RSA.
4 The RSA signature will become faulty due to the induced bit flips in the key.

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

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.

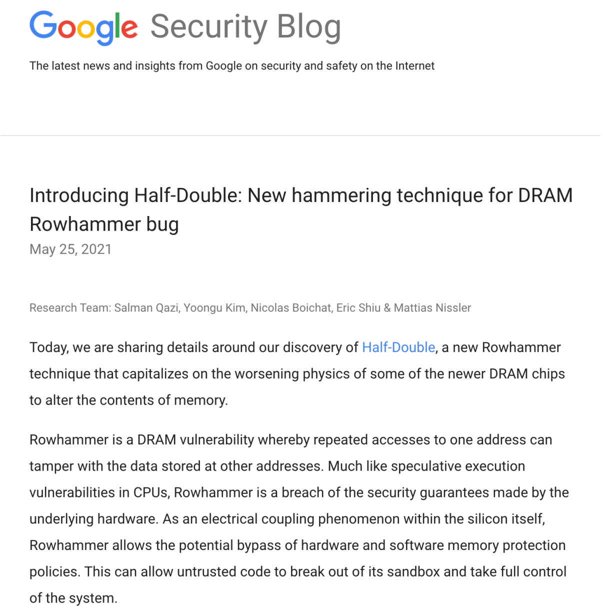
Rowhammer is a DRAM vulnerability whereby repeated accesses to one address can tamper with the data stored at other addresses. Much like speculative execution vulnerabilities in CPUs, Rowhammer is a breach of the security guarantees made by the underlying hardware. As an electrical coupling phenomenon 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 (X)

- **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

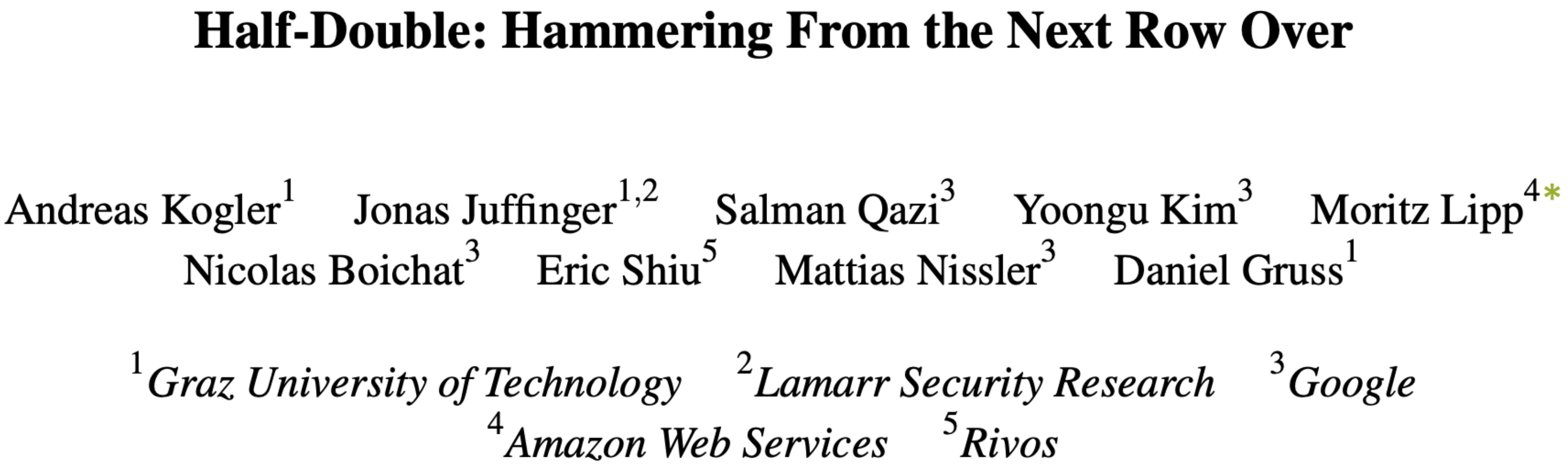
Introducing Half-Double: New hammering technique for DRAM Rowhammer bug

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]

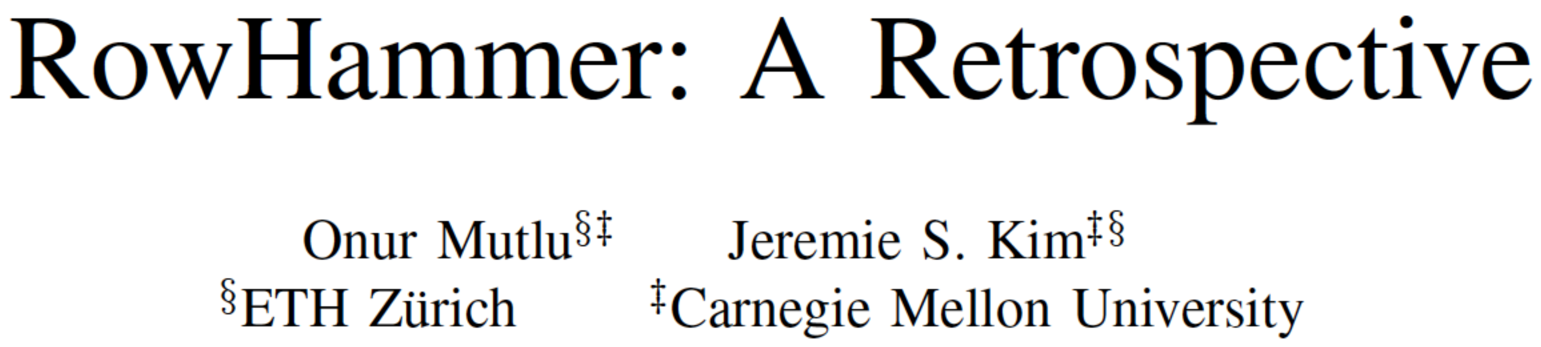
[1]*Graz University of Technology* [2]*Lamarr Security Research* [3]*Google*
[4]*Amazon Web Services* [5]*Rivos*

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

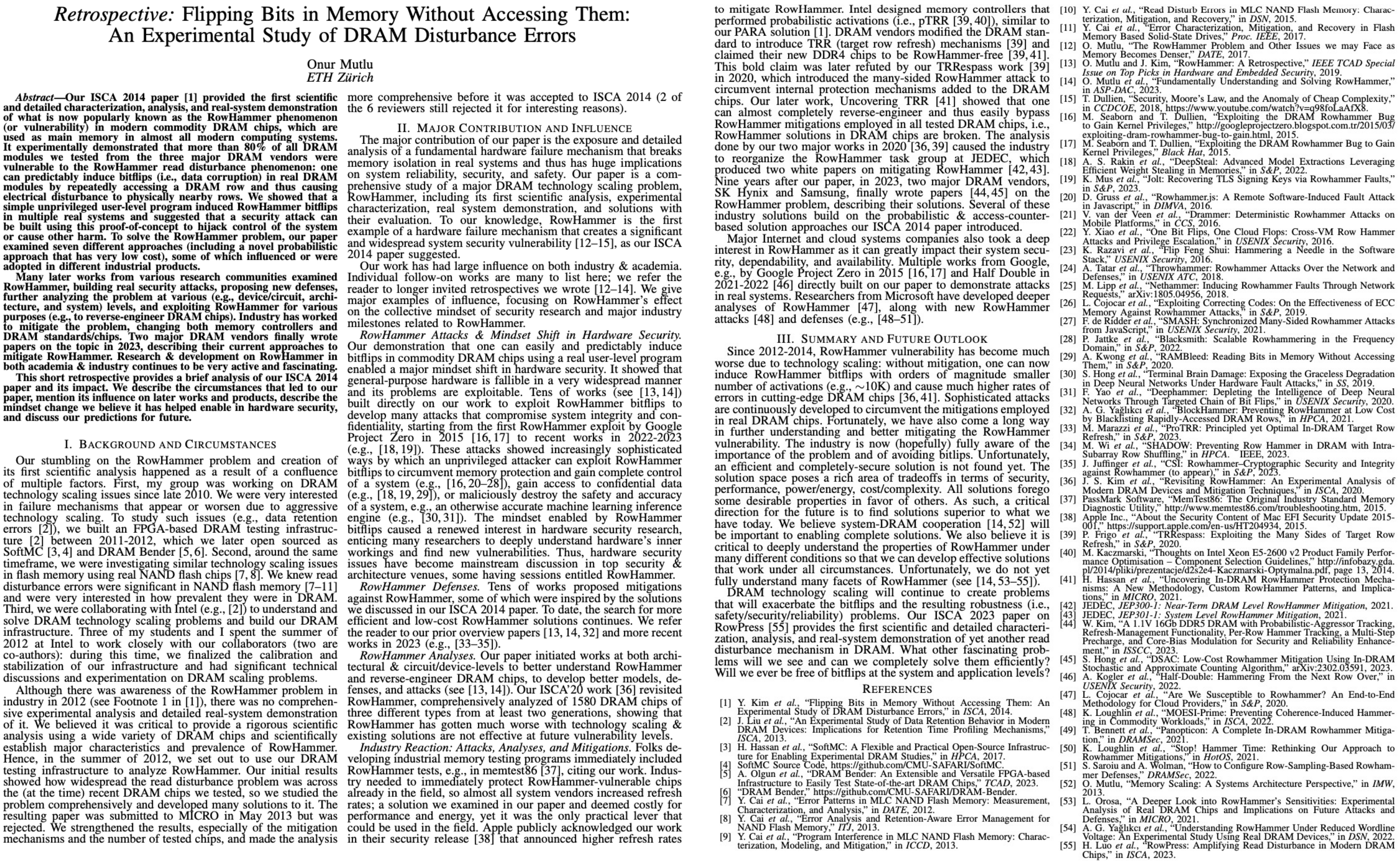
Onur Mutlu
onur.mutlu@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

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

A Short Retrospective @ 50 Years of ISCA



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

Onur Mutlu
ETH Zürich

***Abstract*—Our ISCA 2014 paper [1] provided the first scientific and detailed characterization, analysis, and real-system demonstration of what is now popularly known as the RowHammer phenomenon (or vulnerability) in modern commodity DRAM chips, which are used as main memory in almost all modern computing systems. It experimentally demonstrated that more than 80% of all DRAM modules we tested from the three major DRAM vendors were vulnerable to the RowHammer read disturbance phenomenon: one can predictably induce bitflips (i.e., data corruption) in real DRAM modules by repeatedly accessing a DRAM row and thus causing electrical disturbance to physically nearby rows. We showed that a simple unprivileged user-level program induced RowHammer bitflips in multiple real systems and suggested that a security attack can be built using this proof-of-concept to hijack control of the system or cause other harm. To solve the RowHammer problem, our paper examined seven different approaches (including a novel probabilistic approach that has very low cost), some of which influenced or were adopted in different industrial products.**

Many later works from various research communities examined RowHammer, building real security attacks, proposing new defenses, further analyzing the problem at various (e.g., device/circuit, architecture, and system) levels, and exploiting RowHammer for various purposes (e.g., to reverse-engineer DRAM chips). Industry has worked to mitigate the problem, changing both memory controllers and DRAM standards/chips. Two major DRAM vendors finally wrote papers on the topic in 2023, describing their current approaches to mitigate RowHammer. Research & development on RowHammer in both academia & industry continues to be very active and fascinating.

This short retrospective provides a brief analysis of our ISCA 2014 paper and its impact. We describe the circumstances that led to our paper, mention its influence on later works and products, describe the mindset change we believe it has helped enable in hardware security, and discuss our predictions for future.

I. Background and Circumstances

Our stumbling on the RowHammer problem and creation of its first scientific analysis happened as a result of a confluence of multiple factors. First, my group was working on DRAM technology scaling issues since late 2010. We were very interested in failure mechanisms that appear or worsen due to aggressive technology scaling. To study such issues (e.g., data retention errors [2]), we built an FPGA-based DRAM testing infrastructure [2] between 2011-2012, which we later open sourced as SoftMC [3,4] and DRAM Bender [5,6]. Second, around the same timeframe, we were investigating similar technology scaling issues in flash memory using real NAND flash chips [7,8]. We knew read disturbance errors were significant in NAND flash memory [7–11] and were very interested in how prevalent they were in DRAM. Third, we were collaborating with Intel (e.g., [2]) to understand and solve DRAM technology scaling problems and build our DRAM infrastructure. Three of my students and I spent the summer of 2012 at Intel to work closely with our collaborators (two are co-authors): during this time, we finalized the calibration and stabilization of our infrastructure and had significant technical discussions and experimentation on DRAM scaling problems.

Although there was awareness of the RowHammer problem in industry in 2012 (see Footnote 1 in [1]), there was no comprehensive experimental analysis and detailed real-system demonstration of it. We believed it was critical to provide a rigorous scientific analysis using a wide variety of DRAM chips and scientifically establish major characteristics and prevalence of RowHammer. Hence, in the summer of 2012, we set out to use our DRAM testing infrastructure to analyze RowHammer. Our initial results showed how widespread the read disturbance problem was across the (at the time) recent DRAM chips we tested, so we studied the problem comprehensively and developed many solutions to it. The resulting paper was submitted to MICRO in May 2013 but was rejected. We strengthened the results, especially of the mitigation mechanisms and the number of tested chips, and made the analysis more comprehensive before it was accepted to ISCA 2014 (2 of the 6 reviewers still rejected it for interesting reasons).

II. Major Contribution and Influence

The major contribution of our paper is the exposure and detailed analysis of a fundamental hardware failure mechanism that breaks memory isolation in real systems and thus has huge implications on system reliability, security, and safety. Our paper is a comprehensive study of a major DRAM technology scaling problem, RowHammer, including its first scientific analysis, experimental characterization, real system demonstration, and solutions with their evaluation. To our knowledge, RowHammer is the first example of a hardware failure mechanism that creates a significant and widespread system security vulnerability [12–15], as our ISCA 2014 paper suggested.

Our work has had large influence on both industry & academia. Individual follow-on works are many to list here; we refer the reader to longer invited retrospectives we wrote [12–14]. We give major examples of influence, focusing on RowHammer's effect on the collective mindset of security research and major industry milestones related to RowHammer.

RowHammer Attacks & Mindset Shift in Hardware Security. Our demonstration that one can easily and predictably induce bitflips in commodity DRAM chips using a real user-level program enabled a major mindset shift in hardware security. It showed that general-purpose hardware is fallible in a very widespread manner and its problems are exploitable. Tens of works (see [13,14]) built directly on our work to exploit RowHammer bitflips to develop many attacks that compromise system integrity and confidentiality, starting from the first RowHammer exploit by Google Project Zero in 2015 [16,17] to recent works in 2022-2023 (e.g., [18,19]). These attacks showed increasingly sophisticated ways by which an unprivileged attacker can exploit RowHammer bitflips to circumvent memory protection and gain complete control of a system (e.g., [16,20–28]), gain access to confidential data (e.g., [18,19,29]), or maliciously destroy the safety and accuracy of a system, e.g., an otherwise accurate machine learning inference engine (e.g., [30,31]). The mindset enabled by RowHammer bitflips caused a renewed interest in hardware security research, enticing many researchers to deeply understand hardware's inner workings and find new vulnerabilities. Thus, hardware security issues have become mainstream discussion in top security & architecture venues, some having sessions entitled RowHammer.

RowHammer Defenses. Tens of works proposed mitigations against RowHammer, some of which were inspired by the solutions we discussed in our ISCA 2014 paper. To date, the search for more efficient and low-cost RowHammer solutions continues. We refer the reader to our prior overview papers [13,14,32] and more recent works in 2023 (e.g., [33–35]).

RowHammer Analyses. Our paper initiated works at both architectural & circuit/device-levels to better understand RowHammer and reverse-engineer DRAM chips, to develop better models, defenses, and attacks (see [13,14]). Our ISCA'20 work [36] revisited RowHammer, comprehensively analyzed of 1580 DRAM chips of three different types from at least two generations, showing that RowHammer has gotten much worse with technology scaling & existing solutions are not effective at future vulnerability levels.

Industry Reaction: Attacks, Analyses, and Mitigations. Folks developing industrial memory testing programs immediately included RowHammer tests, e.g., in memtest86 [37], citing our work. Industry needed to immediately protect RowHammer-vulnerable chips already in the field, so almost all system vendors increased refresh rates; a solution we examined in our paper and deemed costly for performance and energy, yet it was the only practical lever that could be used in the field. Apple publicly acknowledged our work in their security release [38] that announced higher refresh rates to mitigate RowHammer. Intel designed memory controllers that performed probabilistic activations (i.e., pTRR [39,40]), similar to our PARA solution [1]. DRAM vendors modified the DRAM standard to introduce TRR (target row refresh) mechanisms [39] and claimed their new DDR4 chips to be RowHammer-free [39,41]. This bold claim was later refuted by our TRRespass work [39] in 2020, which introduced the many-sided RowHammer attack to circumvent internal protection mechanisms added to the DRAM chips. Our later work, Uncovering TRR [41] showed that one can almost completely reverse-engineer and thus easily bypass RowHammer mitigations employed in all tested DRAM chips, i.e., RowHammer solutions in DRAM chips are broken. The analysis done by our two major works in 2020 [36,39] caused the industry to reorganize the RowHammer task group at JEDEC, which produced two white papers on mitigating RowHammer [42,43]. Nine years after our paper, in 2023, two major DRAM vendors, SK Hynix and Samsung, finally wrote papers [44,45] on the RowHammer problem, describing their solutions. Several of these industry solutions build on the probabilistic & access-counter-based solution approaches our ISCA 2014 paper introduced.

Major Internet and cloud systems companies also took a deep interest in RowHammer as it can greatly impact their system security, dependability, and availability. Multiple works from Google, e.g., by Google Project Zero in 2015 [16,17] and Half Double in 2021-2022 [46] directly built on our paper to demonstrate attacks in real systems. Researchers from Microsoft have developed deeper analyses of RowHammer [47], along with new RowHammer attacks [48] and defenses (e.g., [48–51]).

III. Summary and Future Outlook

Since 2012-2014, RowHammer vulnerability has become much worse due to technology scaling: without mitigation, one can now induce RowHammer bitflips with orders of magnitude smaller number of activations (e.g., ∼10K) and cause much higher rates of errors in cutting-edge DRAM chips [36,41]. Sophisticated attacks are continuously developed to circumvent the mitigations employed in real DRAM chips. Fortunately, we have also come a long way in further understanding and better mitigating the RowHammer vulnerability. The industry is now (hopefully) fully aware of the importance of the problem and of avoiding bitlips. Unfortunately, an efficient and completely-secure solution is not found yet. The solution space poses a rich area of tradeoffs in terms of security, performance, power/energy, cost/complexity. All solutions forego some desirable properties in favor of others. As such, a critical direction for the future is to find solutions superior to what we have today. We believe system-DRAM cooperation [14,52] will be important to enabling complete solutions. We also believe it is critical to deeply understand the properties of RowHammer under many different conditions so that we can develop effective solutions that work under all circumstances. Unfortunately, we do not yet fully understand many facets of RowHammer (see [14,53–55]).

DRAM technology scaling will continue to create problems that will exacerbate the bitflips and the resulting robustness (i.e., safety/security/reliability) problems. Our ISCA 2023 paper on RowPress [55] provides the first scientific and detailed characterization, analysis, and real-system demonstration of yet another read disturbance mechanism in DRAM. What other fascinating problems will we see and can we completely solve them efficiently? Will we ever be free of bitflips at the system and application levels?

References

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

[2] J. Liu *et al.*, "An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms," *ISCA*, 2013.

[3] H. Hassan *et al.*, "SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies," in *HPCA*, 2017.

[4] SoftMC Source Code, https://github.com/CMU-SAFARI/SoftMC.

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

[6] "DRAM Bender," https://github.com/CMU-SAFARI/DRAM-Bender.

[7] Y. Cai *et al.*, "Error Patterns in MLC NAND Flash Memory: Measurement, Characterization, and Analysis," in *DATE*, 2012.

[8] Y. Cai *et al.*, "Error Analysis and Retention-Aware Error Management for NAND Flash Memory," *ITJ*, 2013.

[9] Y. Cai *et al.*, "Program Interference in MLC NAND Flash Memory: Characterization, Modeling, and Mitigation," in *ICCD*, 2013.

[10] Y. Cai *et al.*, "Read Disturb Errors in MLC NAND Flash Memory: Characterization, Mitigation, and Recovery," in *DSN*, 2015.

[11] Y. Cai *et al.*, "Error Characterization, Mitigation, and Recovery in Flash Memory Based Solid-State Drives," *Proc. IEEE*, 2017.

[12] O. Mutlu, "The RowHammer Problem and Other Issues we may Face as Memory Becomes Denser," *DATE*, 2017.

[13] O. Mutlu and J. Kim, "RowHammer: A Retrospective," *IEEE TCAD Special Issue on Top Picks in Hardware and Embedded Security*, 2019.

[14] O. Mutlu *et al.*, "Fundamentally Understanding and Solving RowHammer," in *ASP-DAC*, 2023.

[15] T. Dullien, "Security, Moore's Law, and the Anomaly of Cheap Complexity," in *CCDCOE*, 2018, https://www.youtube.com/watch?v=q98foLaAfX8.

[16] M. Seaborn and T. Dullien, "Exploiting the DRAM Rowhammer Bug to Gain Kernel Privileges," http://googleprojectzero.blogspot.com.tr/2015/03/exploiting-dram-rowhammer-bug-to-gain.html, 2015.

[17] M. Seaborn and T. Dullien, "Exploiting the DRAM Rowhammer Bug to Gain Kernel Privileges," *Black Hat*, 2015.

[18] A. S. Rakin *et al.*, "DeepSteal: Advanced Model Extractions Leveraging Efficient Weight Stealing in Memories," in *S&P*, 2022.

[19] K. Mus *et al.*, "Jolt: Recovering TLS Signing Keys via Rowhammer Faults," in *S&P*, 2023.

[20] D. Gruss *et al.*, "Rowhammer.js: A Remote Software-Induced Fault Attack in Javascript," in *DIMVA*, 2016.

[21] V. van der Veen *et al.*, "Drammer: Deterministic Rowhammer Attacks on Mobile Platforms," in *CCS*, 2016.

[22] Y. Xiao *et al.*, "One Bit Flips, One Cloud Flops: Cross-VM Row Hammer Attacks and Privilege Escalation," in *USENIX Security*, 2016.

[23] K. Razavi *et al.*, "Flip Feng Shui: Hammering a Needle in the Software Stack," *USENIX Security*, 2016.

[24] A. Tatar *et al.*, "Throwhammer: Rowhammer Attacks Over the Network and Defenses," in *USENIX ATC*, 2018.

[25] M. Lipp *et al.*, "Nethammer: Inducing Rowhammer Faults Through Network Requests," arXiv:1805.04956, 2018.

[26] L. Cojocar *et al.*, "Exploiting Correcting Codes: On the Effectiveness of ECC Memory Against Rowhammer Attacks," in *S&P*, 2019.

[27] F. de Ridder *et al.*, "SMASH: Synchronized Many-Sided Rowhammer Attacks from JavaScript," in *USENIX Security*, 2021.

[28] P. Jattke *et al.*, "Blacksmith: Scalable Rowhammering in the Frequency Domain," in *S&P*, 2022.

[29] A. Kwong *et al.*, "RAMBleed: Reading Bits in Memory Without Accessing Them," in *S&P*, 2020.

[30] S. Hong *et al.*, "Terminal Brain Damage: Exposing the Graceless Degradation in Deep Neural Networks Under Hardware Fault Attacks," in *SS*, 2019.

[31] F. Yao *et al.*, "Deephammer: Depleting the Intelligence of Deep Neural Networks Through Targeted Chain of Bit Flips," in *USENIX Security*, 2020.

[32] A. G. Yağlıkcı *et al.*, "BlockHammer: Preventing RowHammer at Low Cost by Blacklisting Rapidly-Accessed DRAM Rows," in *HPCA*, 2021.

[33] M. Marazzi *et al.*, "ProTRR: Principled yet Optimal In-DRAM Target Row Refresh," in *S&P*, 2023.

[34] M. Wi *et al.*, "SHADOW: Preventing Row Hammer in DRAM with Intra-Subarray Row Shuffling," in *HPCA*. IEEE, 2023.

[35] J. Juffinger *et al.*, "CSI: Rowhammer–Cryptographic Security and Integrity against Rowhammer (to appear)," in *S&P*, 2023.

[36] J. S. Kim *et al.*, "Revisiting RowHammer: An Experimental Analysis of Modern DRAM Devices and Mitigation Techniques," in *ISCA*, 2020.

[37] PassMark Software, "MemTest86: The Original Industry Standard Memory Diagnostic Utility," http://www.memtest86.com/troubleshooting.htm, 2015.

[38] Apple Inc., "About the Security Content of Mac EFI Security Update 2015-001," https://support.apple.com/en-us/HT204934, 2015.

[39] P. Frigo *et al.*, "TRRespass: Exploiting the Many Sides of Target Row Refresh," in *S&P*, 2020.

[40] M. Kaczmarski, "Thoughts on Intel Xeon E5-2600 v2 Product Family Performance Optimisation – Component Selection Guidelines," http://infobazy.gda.pl/2014/pliki/prezentacje/d2s2e4-Kaczmarski-Optymalna.pdf, page 13, 2014.

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

[42] JEDEC, *JEP300-1: Near-Term DRAM Level RowHammer Mitigation*, 2021.

[43] JEDEC, *JEP301-1: System Level RowHammer Mitigation*, 2021.

[44] W. Kim, "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," in *ISSCC*, 2023.

[45] S. Hong *et al.*, "DSAC: Low-Cost Rowhammer Mitigation Using In-DRAM Stochastic and Approximate Counting Algorithm," arXiv:2302.03591, 2023.

[46] A. Kogler *et al.*, "Half-Double: Hammering From the Next Row Over," in *USENIX Security*, 2022.

[47] L. Cojocar *et al.*, "Are We Susceptible to Rowhammer? An End-to-End Methodology for Cloud Providers," in *S&P*, 2020.

[48] K. Loughlin *et al.*, "MOESI-Prime: Preventing Coherence-Induced Hammering in Commodity Workloads," in *ISCA*, 2022.

[49] T. Bennett *et al.*, "Panopticon: A Complete In-DRAM Rowhammer Mitigation," in *DRAMSec*, 2021.

[50] K. Loughlin *et al.*, "Stop! Hammer Time: Rethinking Our Approach to Rowhammer Mitigations," in *HotOS*, 2021.

[51] S. Saroiu and A. Wolman, "How to Configure Row-Sampling-Based Rowhammer Defenses," *DRAMSec*, 2022.

[52] O. Mutlu, "Memory Scaling: A Systems Architecture Perspective," in *IMW*, 2013.

[53] L. Orosa, "A Deeper Look into RowHammer's Sensitivities: Experimental Analysis of Real DRAM Chips and Implications on Future Attacks and Defenses," in *MICRO*, 2021.

[54] A. G. Yağlıkcı *et al.*, "Understanding RowHammer Under Reduced Wordline Voltage: An Experimental Study Using Real DRAM Devices," in *DSN*, 2022.

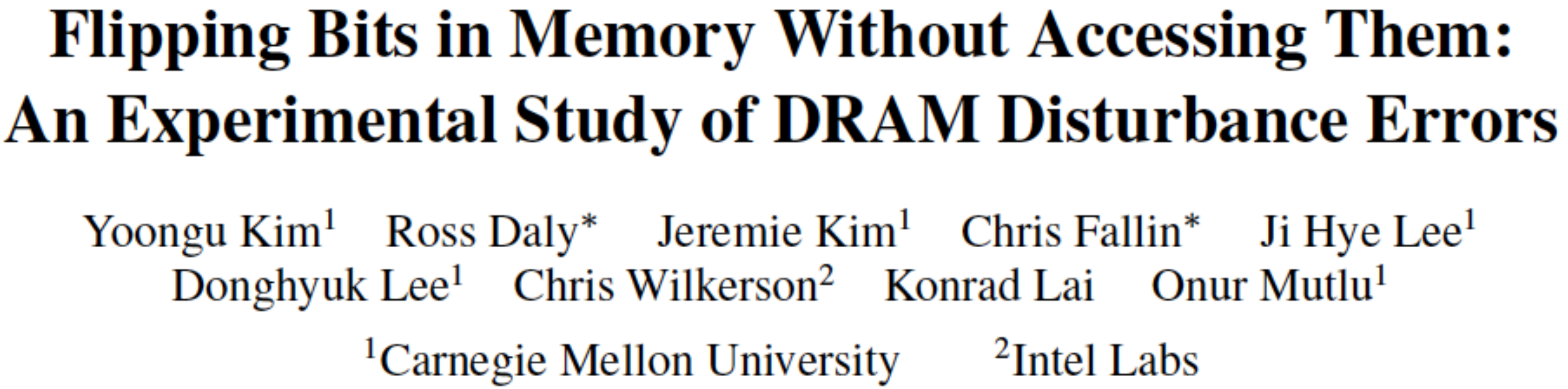
[55] H. Luo *et al.*, "RowPress: Amplifying Read Disturbance in Modern DRAM Chips," in *ISCA*, 2023.

https://arxiv.org/pdf/2306.16093

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). Winner of the 2024 IFIP Jean-Claude Laprie Award in dependable computing (link).

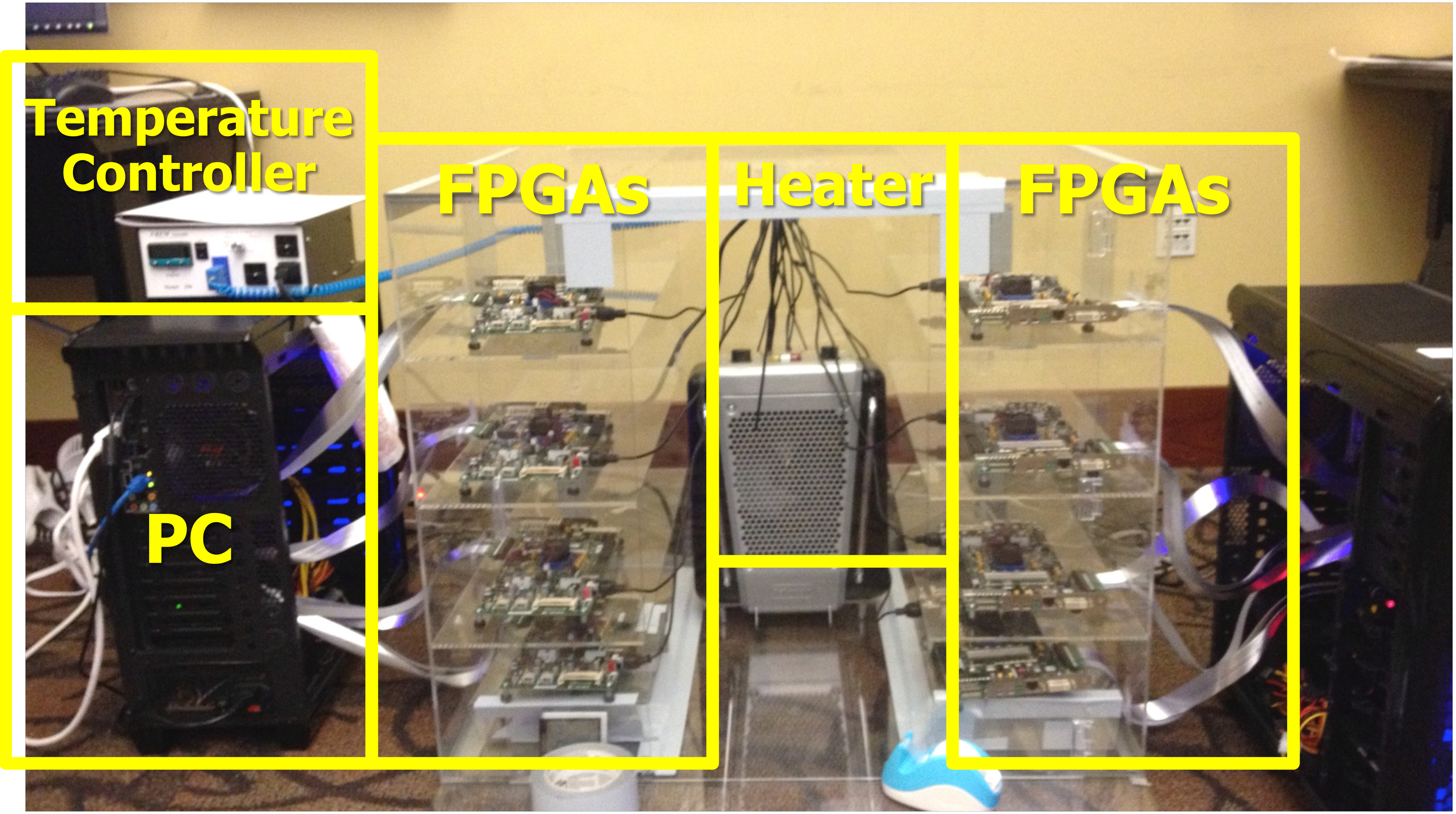


Flipping Bits in Memory Without Accessing Them:
An Experimental Study of DRAM Disturbance Errors

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)

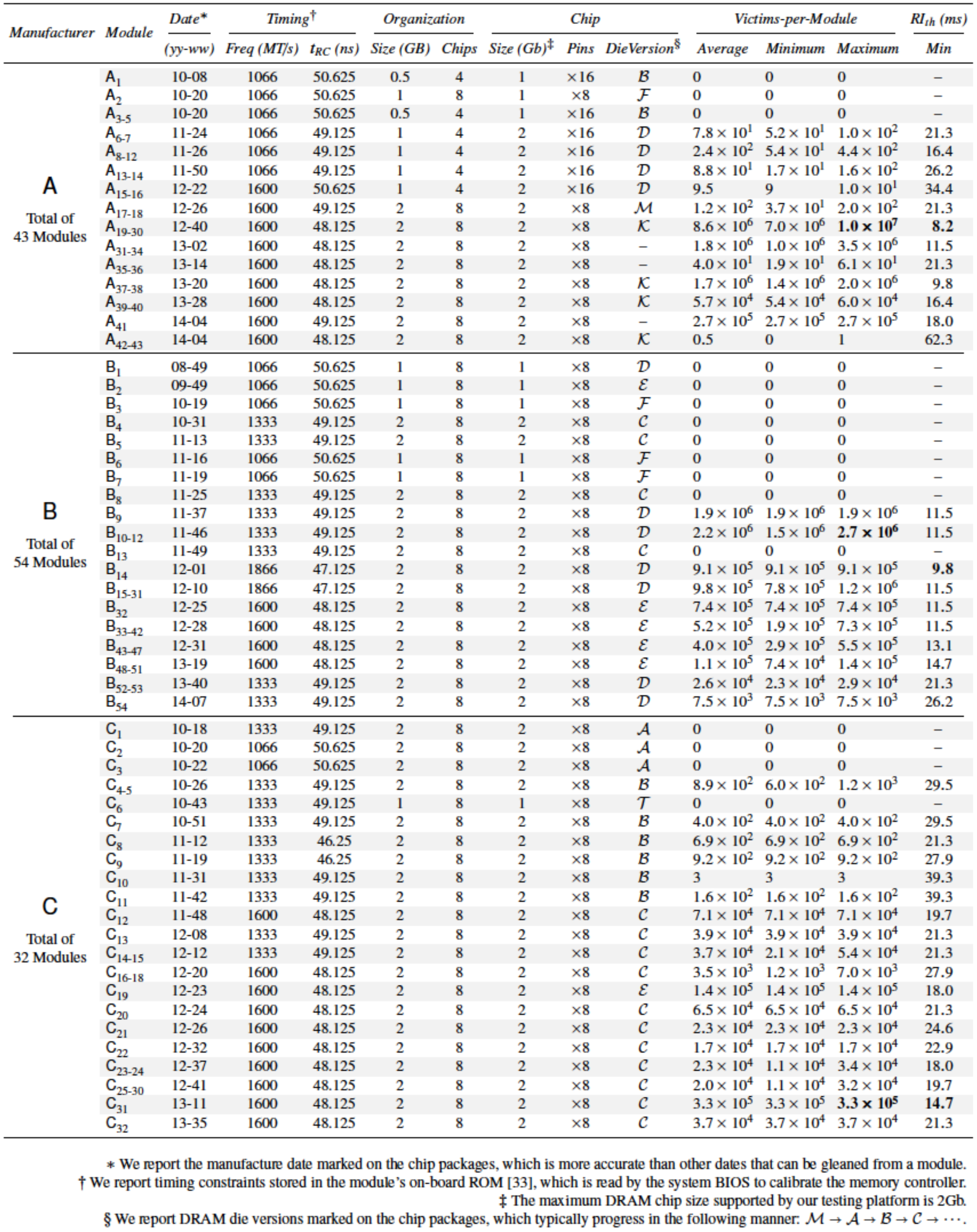


SAFARI

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*	Timing†		Organization		Chip			Victims-per-Module			RI_{th} (ms)
		(yy-ww)	Freq (MT/s)	t_{RC} (ns)	Size (GB)	Chips	Size (Gb)‡	Pins	DieVersion§	Average	Minimum	Maximum	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	**1.0×10^7**	**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	**2.7×10^6**	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	**3.3×10^5**	**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: $\mathcal{M} \to \mathcal{A} \to \mathcal{B} \to \mathcal{C} \to \cdots$.

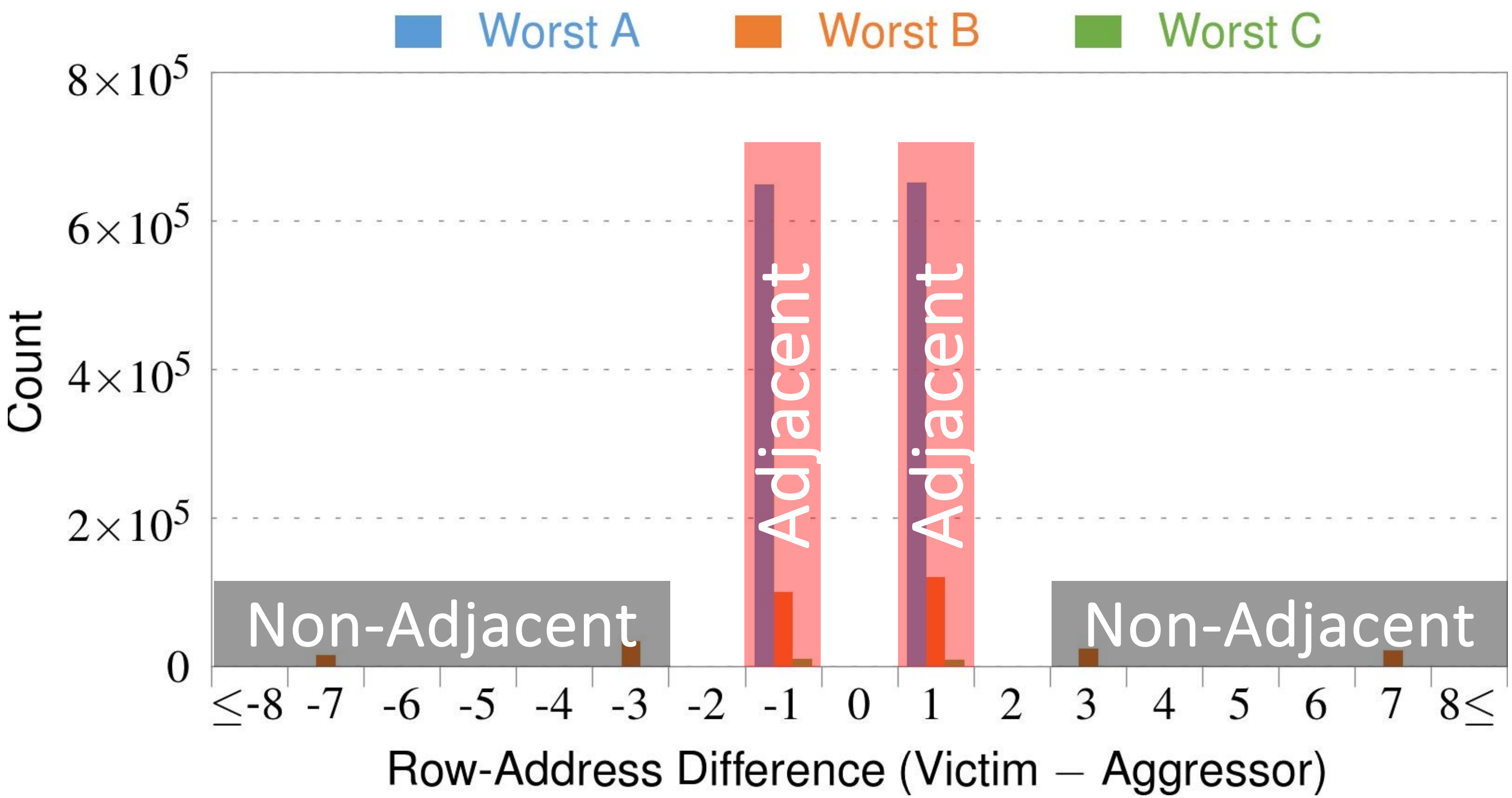
Table 3. Sample population of 129 DDR3 DRAM modules, categorized by manufacturer and sorted by manufacture date

RowHammer Characterization Results

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

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

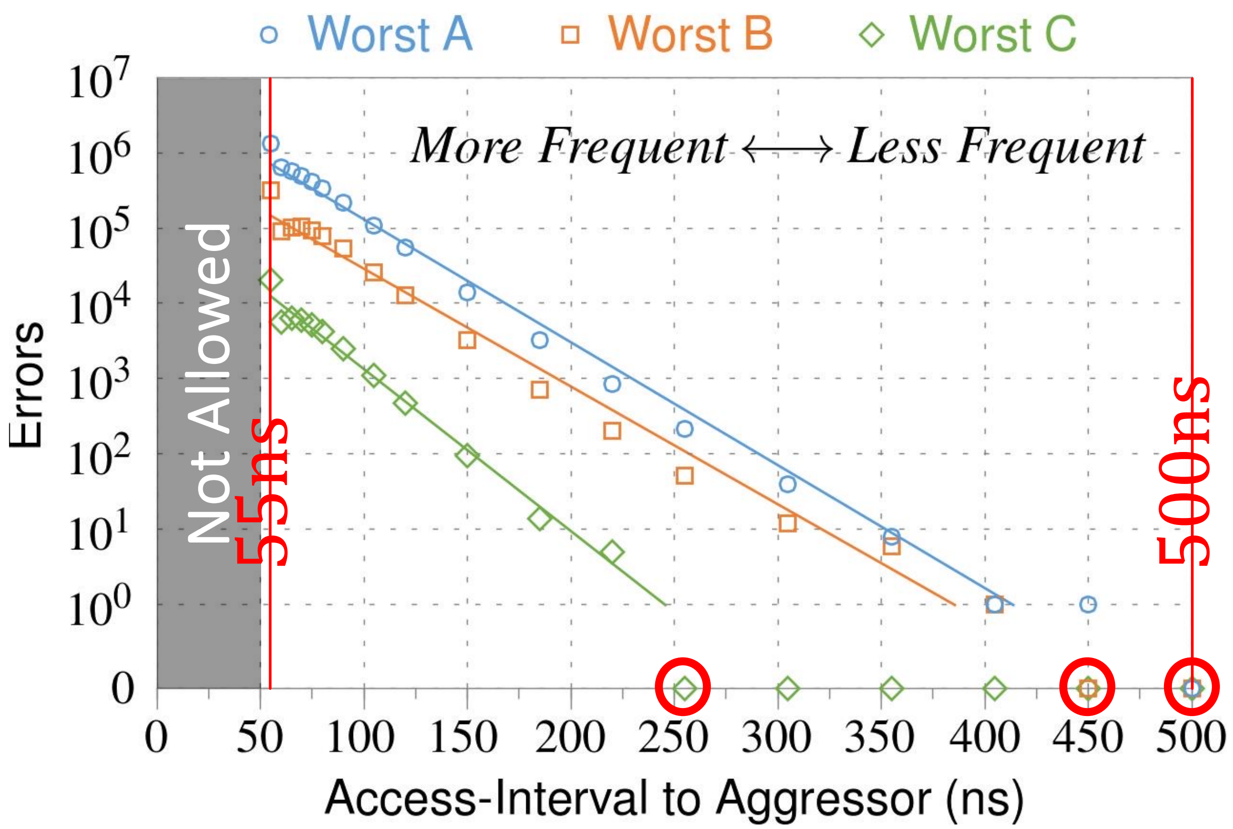
4. Adjacency: Aggressor & Victim



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

Most aggressors & victims are adjacent

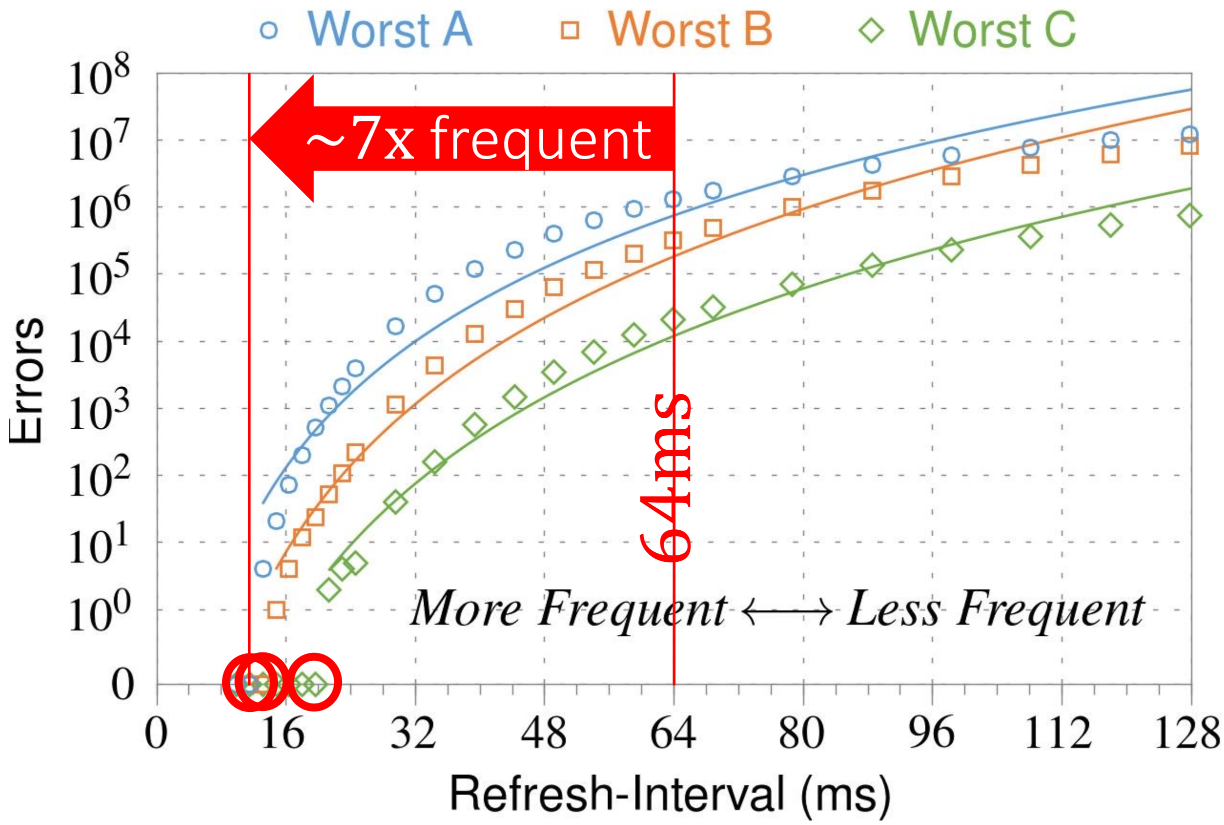
❶ Access Interval (Aggressor)



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

Less frequent accesses → Fewer errors

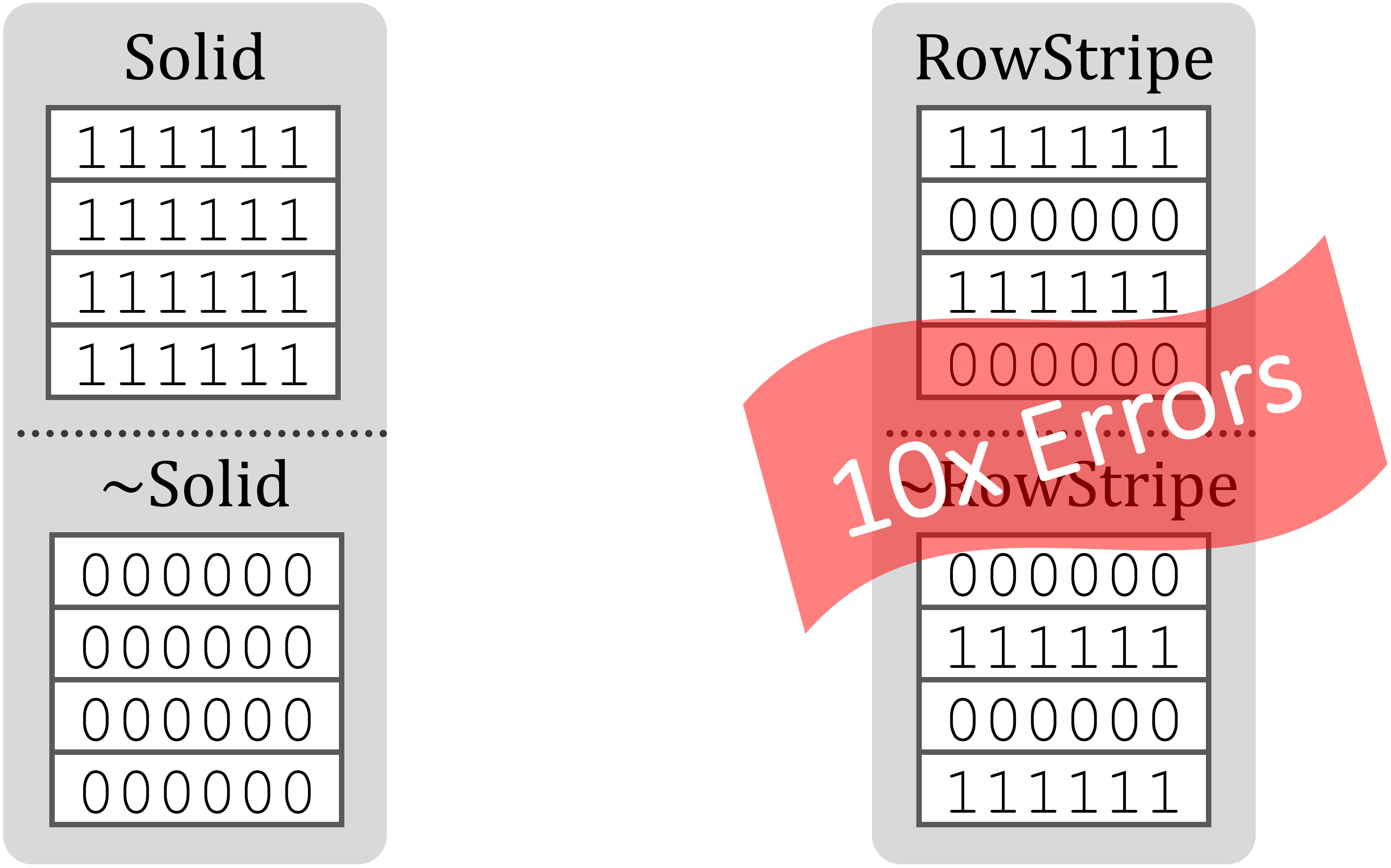
❷ Refresh Interval



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

More frequent refreshes → Fewer errors

❸ Data Pattern



Solid

111111
111111
111111
111111

~Solid

000000
000000
000000
000000

RowStripe

111111
000000
111111
000000

~RowStripe

000000
111111
000000
111111

10x Errors

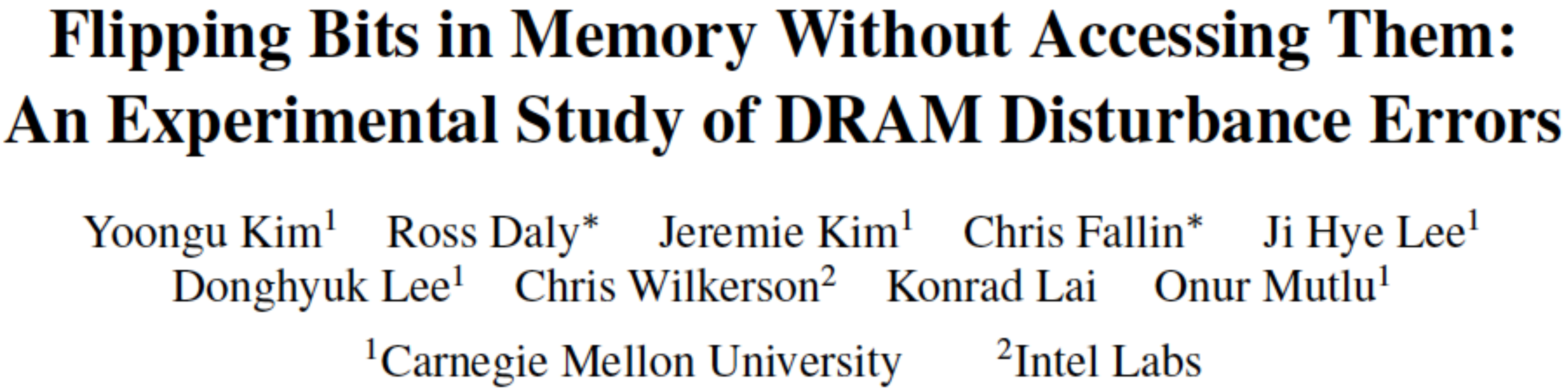
Errors affected by data stored in other cells

6. Other Key Observations [ISCA'14]

- *Victim Cells ≠ 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). Winner of the 2024 IFIP Jean-Claude Laprie Award in dependable computing (link).



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*

Hard to Guarantee RowHammer-Free Chips (2020)

- Lucian Cojocar, Jeremie Kim, Minesh Patel, Lillian Tsai, Stefan Saroiu, Alec Wolman, and Onur Mutlu,
  "Are We Susceptible to Rowhammer? An End-to-End Methodology for Cloud Providers"
  *Proceedings of the 41st IEEE Symposium on Security and Privacy (**S&P**)*, San Francisco, CA, USA, May 2020.
  [Slides (pptx) (pdf)]
  [Talk Video (17 minutes)]

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

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

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

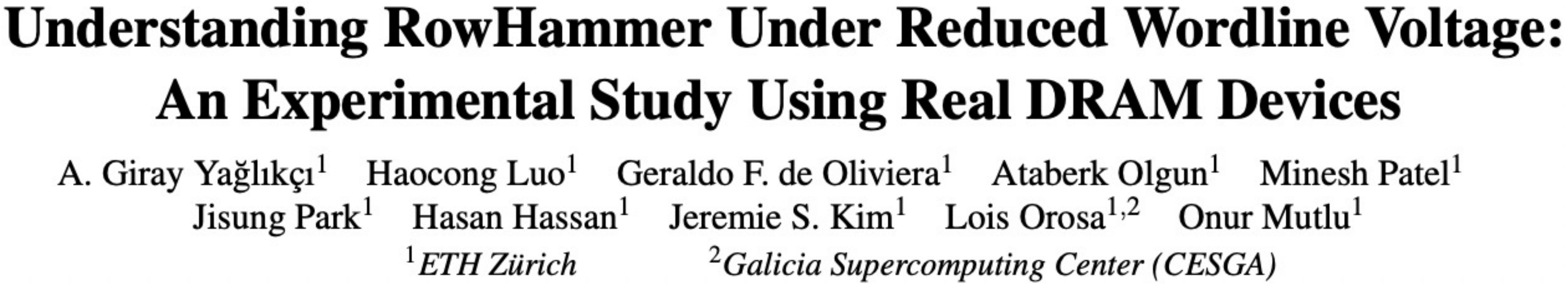
Minesh Patel ETH Zürich

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)]

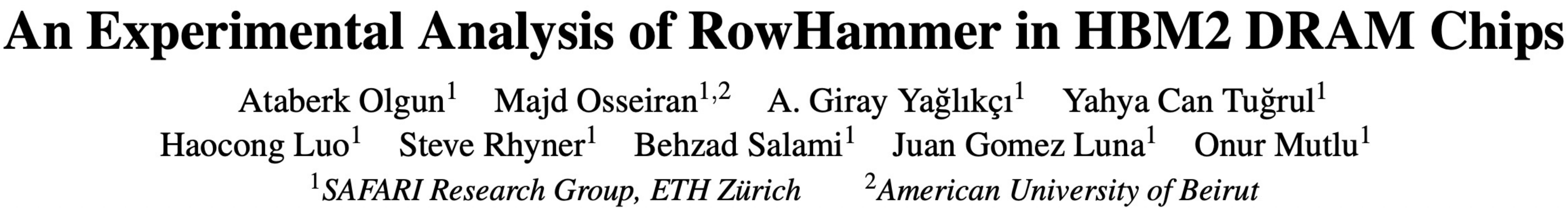


Understanding RowHammer Under Reduced Wordline Voltage: An Experimental Study Using Real DRAM Devices

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)]



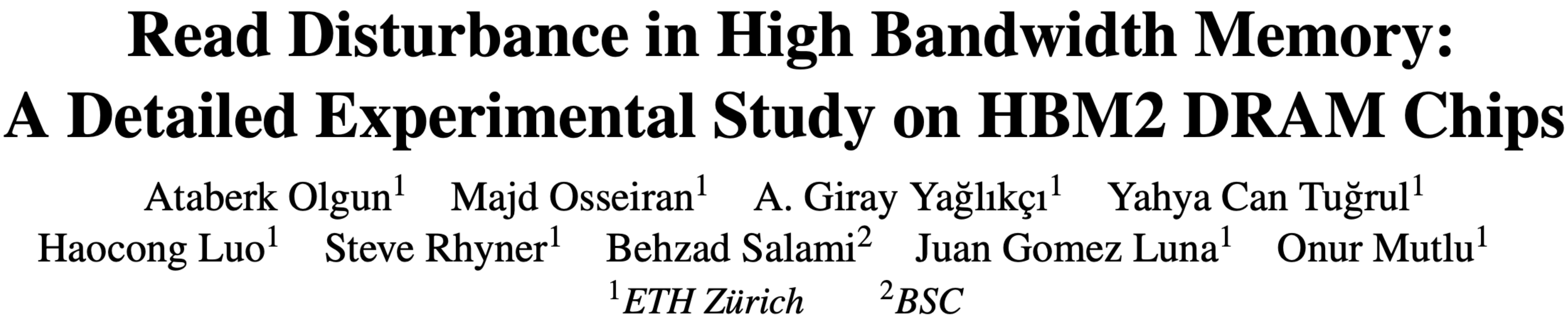
An Experimental Analysis of RowHammer in HBM2 DRAM Chips

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**



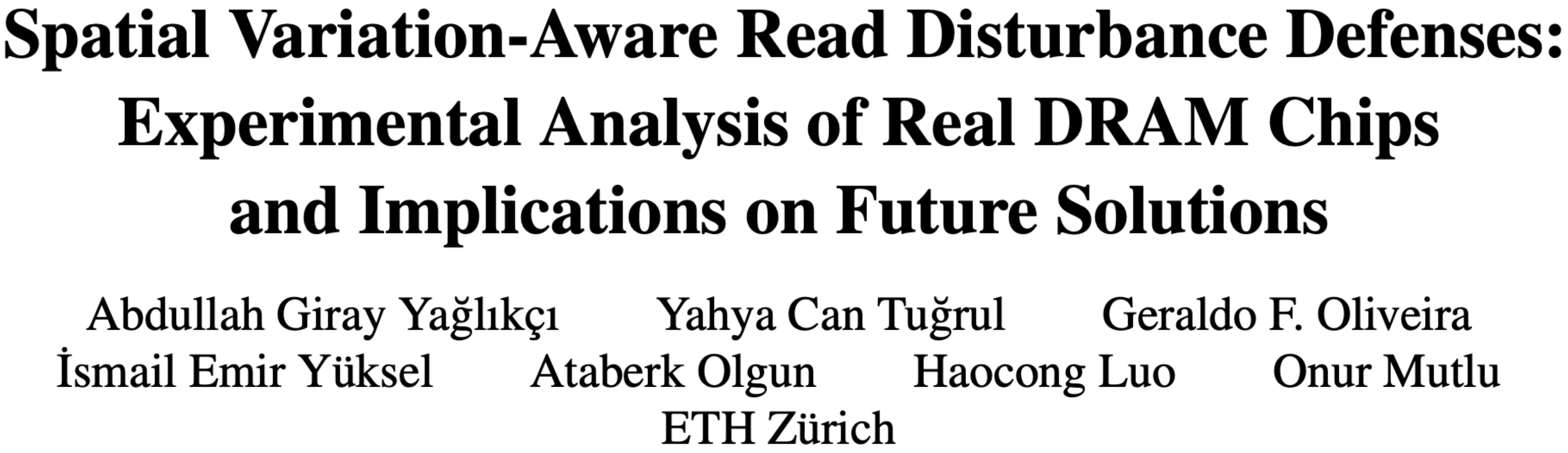
Read Disturbance in High Bandwidth Memory:
A Detailed Experimental Study on HBM2 DRAM Chips

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

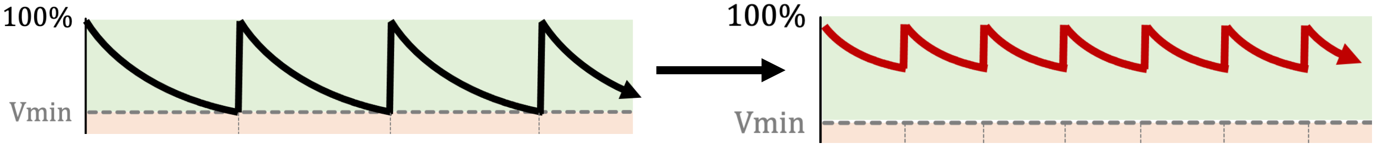
- Immediate
  - To protect the vulnerable DRAM chips in the field
  - Limited possibilities

- 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

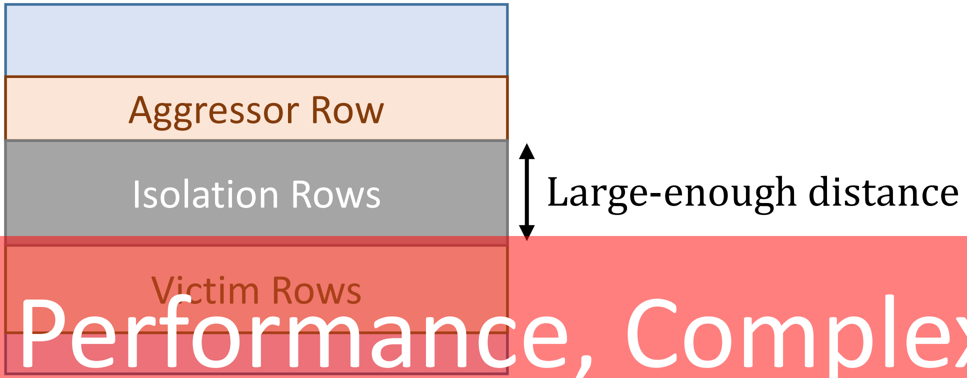
RowHammer Solution Approaches

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



Fewer activations possible
in a refresh interval

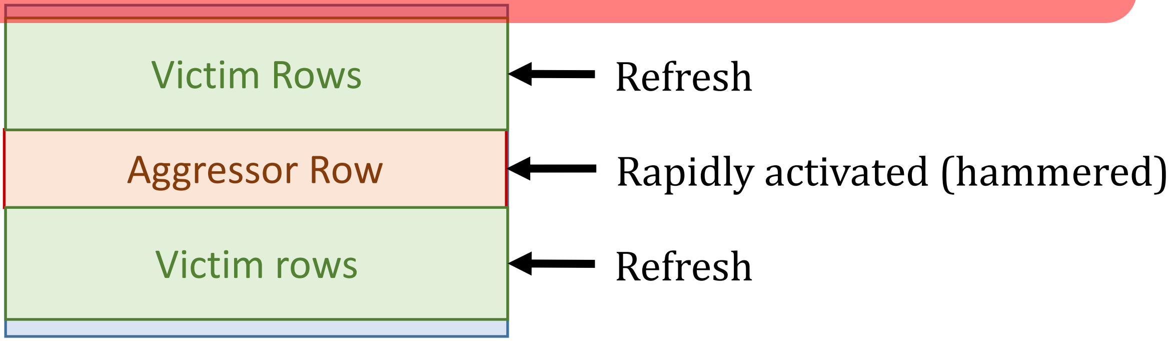
- Physical isolation



Large-enough distance

Cost, Power, Performance, Complexity

- Reactive refresh

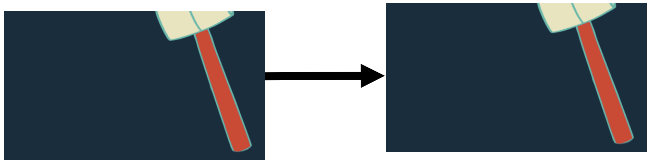


Refresh

Rapidly activated (hammered)

Refresh

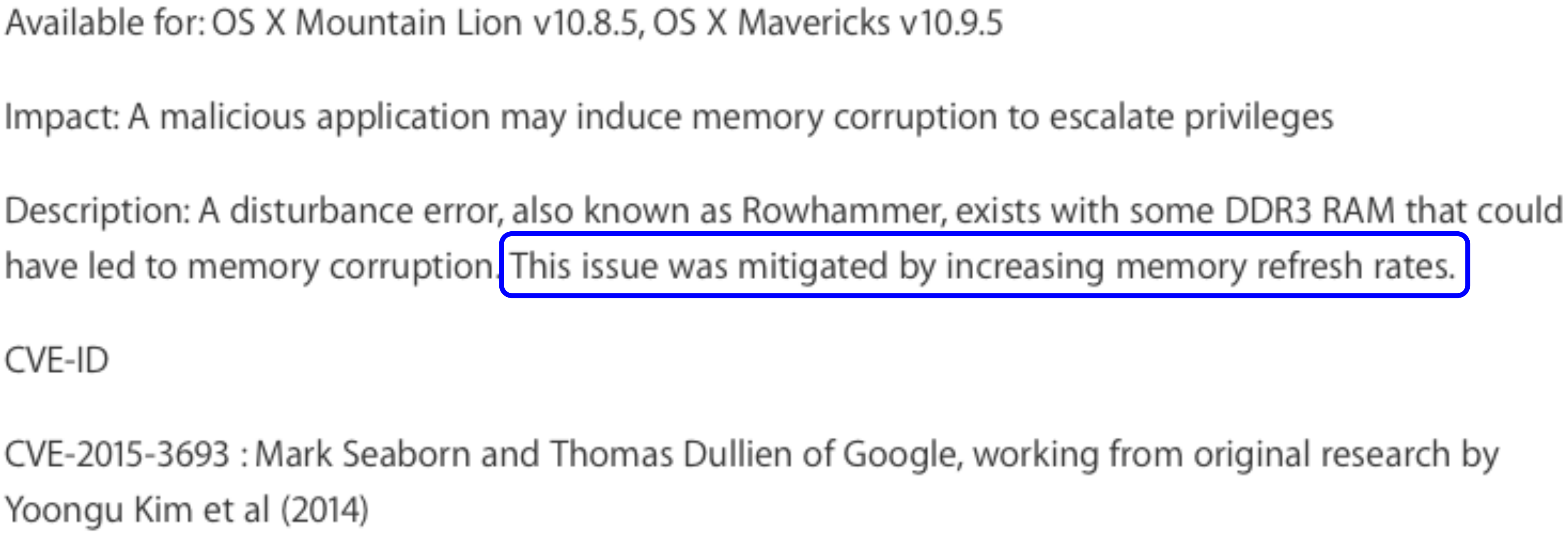
- Proactive throttling



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

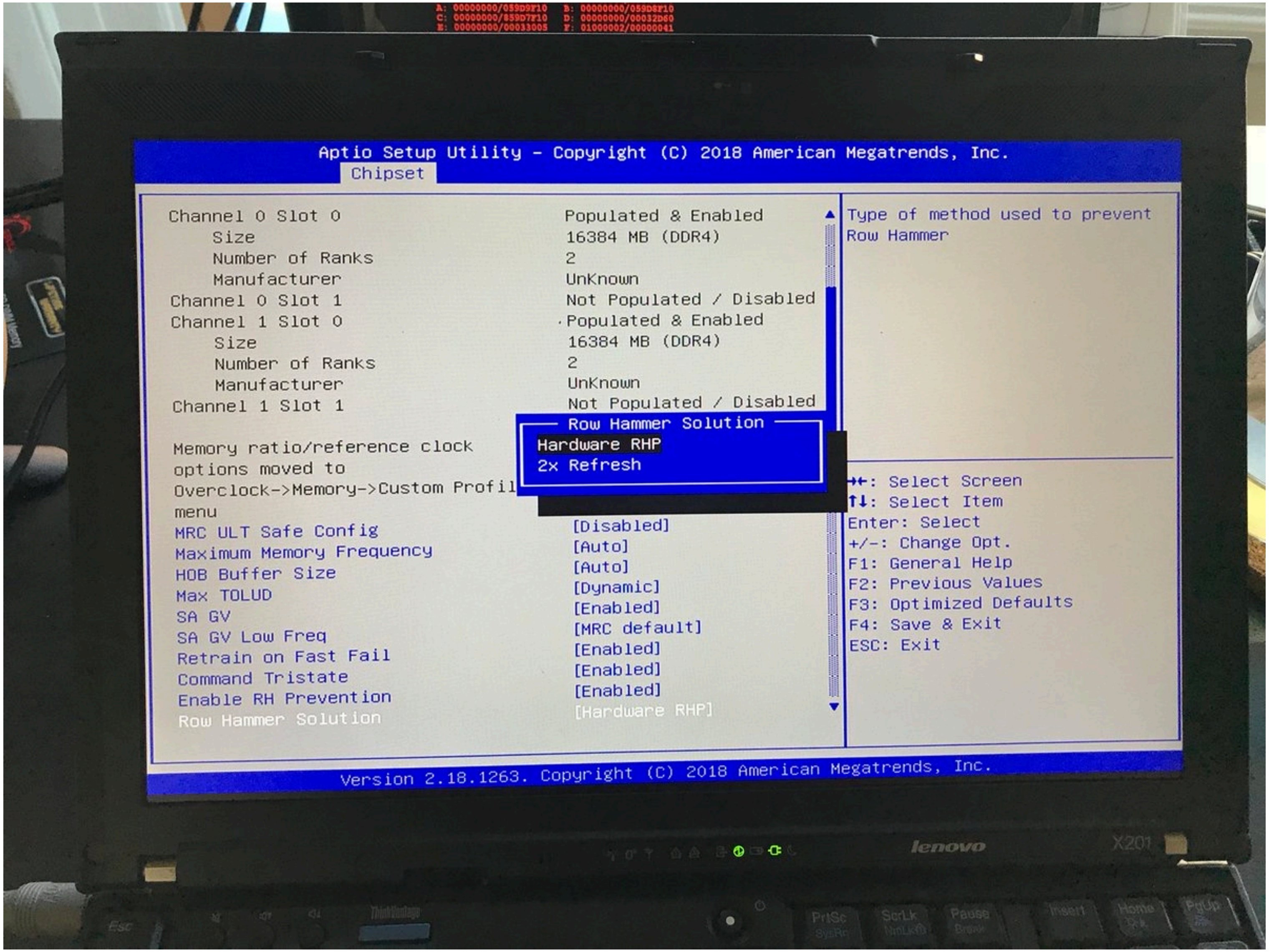
- *PARA is an effective and low-overhead solution to prevent disturbance errors*

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)

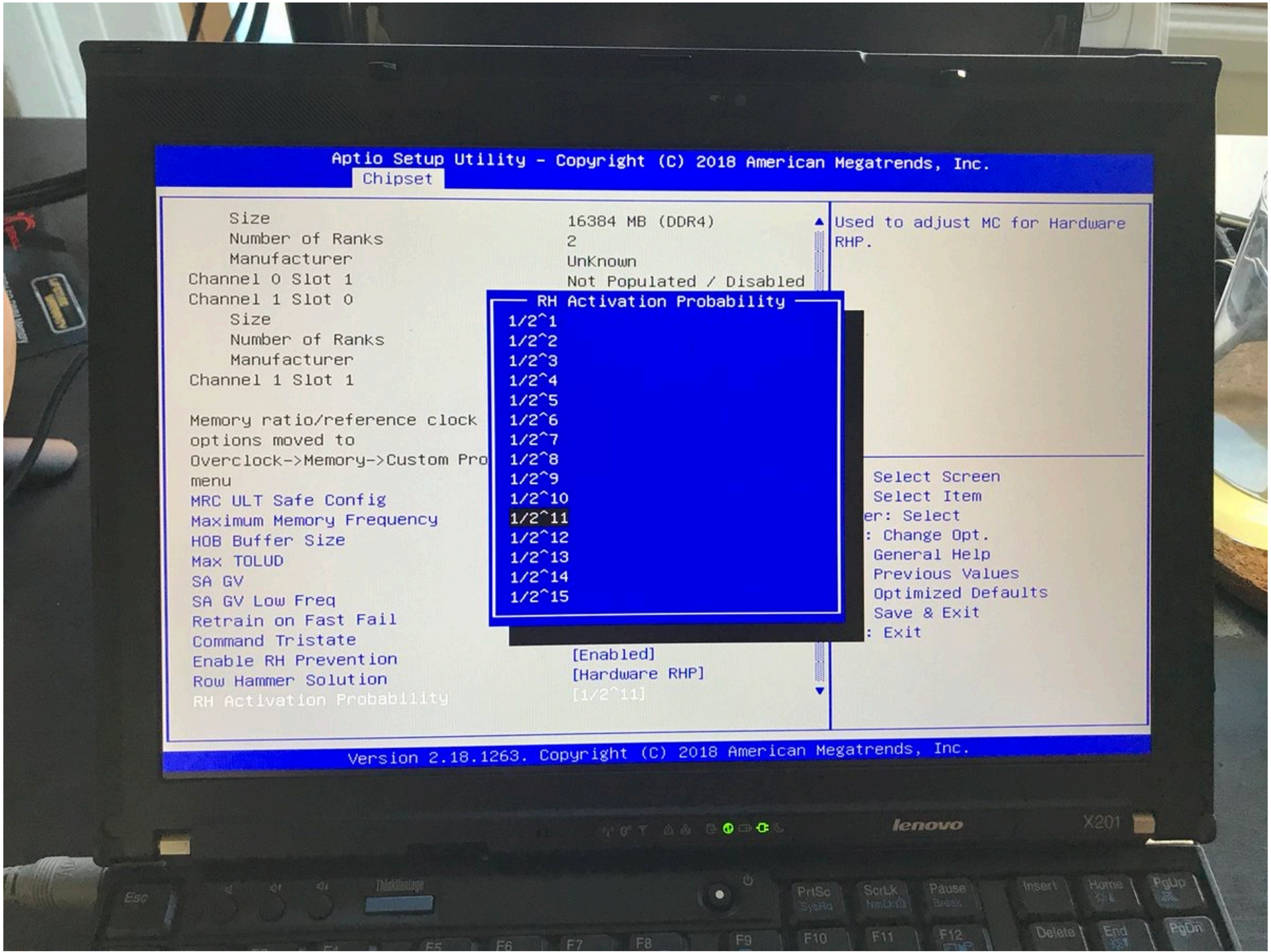


```
A: 00000000/059D9F10   B: 00000000/059D8F10
C: 00000000/859D7F10   D: 00000000/00032D60
E: 00000000/00033005   F: 01000002/00000041
```

```
        Aptio Setup Utility - Copyright (C) 2018 American Megatrends, Inc.
              Chipset
 Channel 0 Slot 0                          Populated & Enabled       | Type of method used to prevent
     Size                                  16384 MB (DDR4)           | Row Hammer
     Number of Ranks                       2                         |
     Manufacturer                          UnKnown                   |
 Channel 0 Slot 1                          Not Populated / Disabled  |
 Channel 1 Slot 0                          Populated & Enabled       |
     Size                                  16384 MB (DDR4)           |
     Number of Ranks                       2                         |
     Manufacturer                          UnKnown                   |
 Channel 1 Slot 1                          Not Populated / Disabled  |
                                  ---- Row Hammer Solution ----      |
 Memory ratio/reference clock     Hardware RHP                       |
 options moved to                 2x Refresh                         | ++: Select Screen
 Overclock->Memory->Custom Profil                                    | ↑↓: Select Item
 menu                                                                | Enter: Select
 MRC ULT Safe Config                        [Disabled]               | +/-: Change Opt.
 Maximum Memory Frequency                   [Auto]                   | F1: General Help
 HOB Buffer Size                            [Auto]                   | F2: Previous Values
 Max TOLUD                                  [Dynamic]                | F3: Optimized Defaults
 SA GV                                      [Enabled]                | F4: Save & Exit
 SA GV Low Freq                             [MRC default]            | ESC: Exit
 Retrain on Fast Fail                       [Enabled]                |
 Command Tristate                           [Enabled]                |
 Enable RH Prevention                       [Enabled]                |
 Row Hammer Solution                        [Hardware RHP]           |
          Version 2.18.1263. Copyright (C) 2018 American Megatrends, Inc.
```

https://twitter.com/isislovecruft/status/1021939922754723841

Probabilistic Activation in Real Life (II)

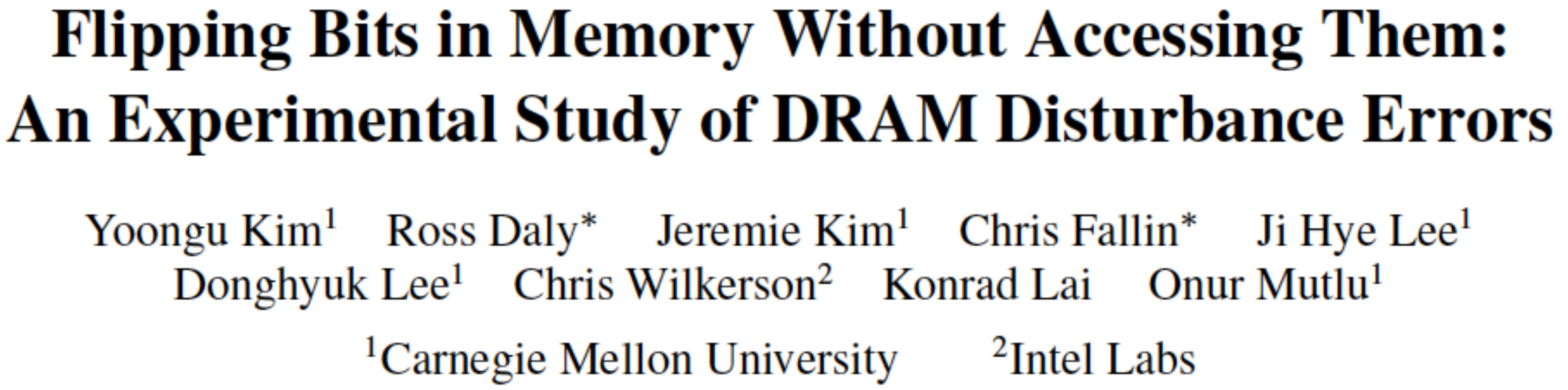


```
        Aptio Setup Utility - Copyright (C) 2018 American Megatrends, Inc.
                     Chipset

     Size                                    16384 MB (DDR4)         Used to adjust MC for Hardware
     Number of Ranks                         2                       RHP.
     Manufacturer                            UnKnown
 Channel 0 Slot 1                            Not Populated / Disabled
 Channel 1 Slot 0                  RH Activation Probability
     Size                          1/2^1
     Number of Ranks               1/2^2
     Manufacturer                  1/2^3
 Channel 1 Slot 1                  1/2^4
                                   1/2^5
 Memory ratio/reference clock      1/2^6
 options moved to                  1/2^7
 Overclock->Memory->Custom Pro     1/2^8
 menu                              1/2^9                              Select Screen
 MRC ULT Safe Config               1/2^10                             Select Item
 Maximum Memory Frequency          1/2^11                           er: Select
 HOB Buffer Size                   1/2^12                           : Change Opt.
 Max TOLUD                         1/2^13                             General Help
 SA GV                             1/2^14                             Previous Values
 SA GV Low Freq                    1/2^15                             Optimized Defaults
 Retrain on Fast Fail                                                 Save & Exit
 Command Tristate                                                   : Exit
 Enable RH Prevention                        [Enabled]
 Row Hammer Solution                         [Hardware RHP]
 RH Activation Probability                   [1/2^11]

          Version 2.18.1263. Copyright (C) 2018 American Megatrends, Inc.
```

https://twitter.com/isislovecruft/status/1021939922754723841

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). Winner of the 2024 IFIP Jean-Claude Laprie Award in dependable computing (link).



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

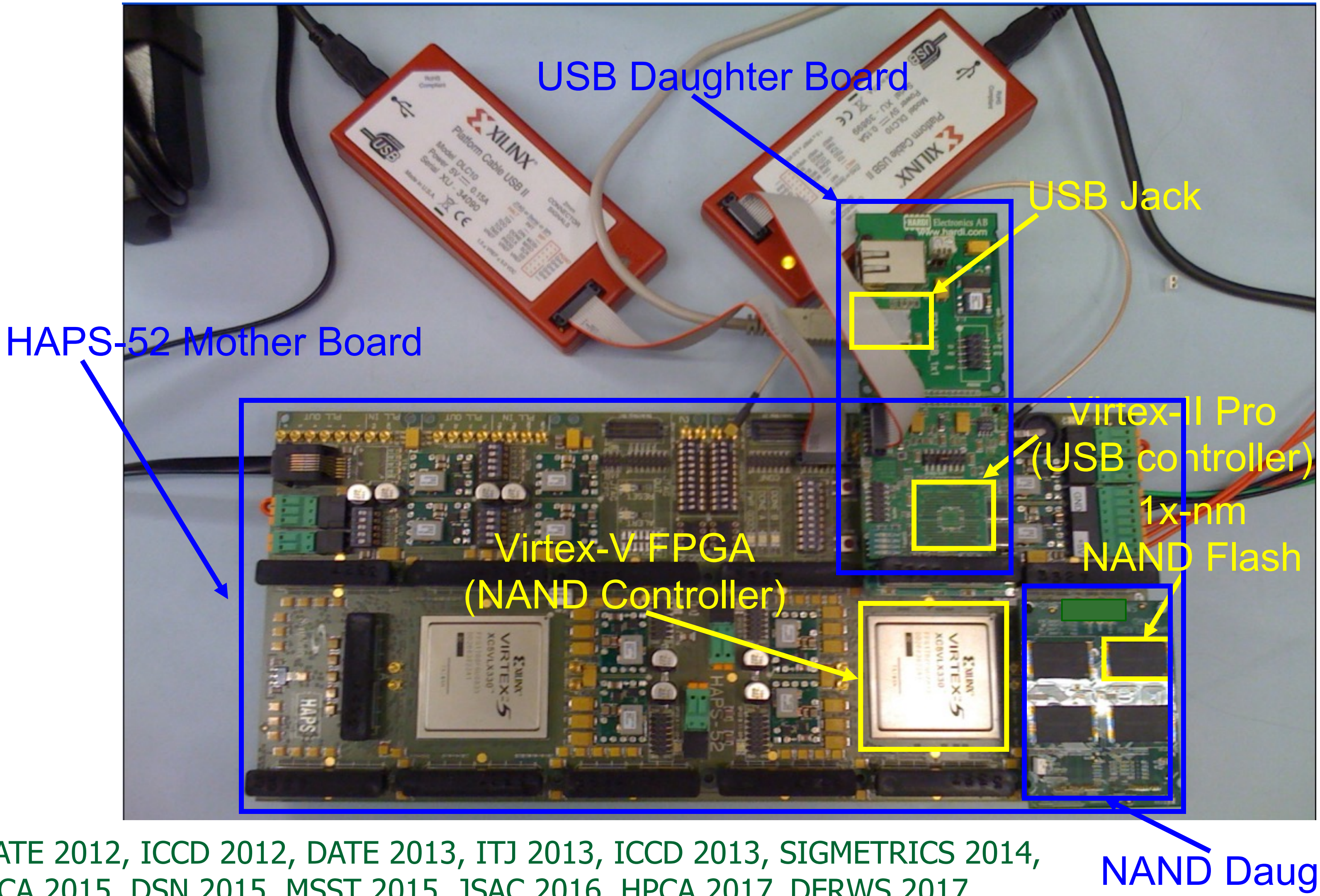
A Takeaway

Main Memory Needs

Intelligent Controllers

for Security, Safety, Reliability, Scaling

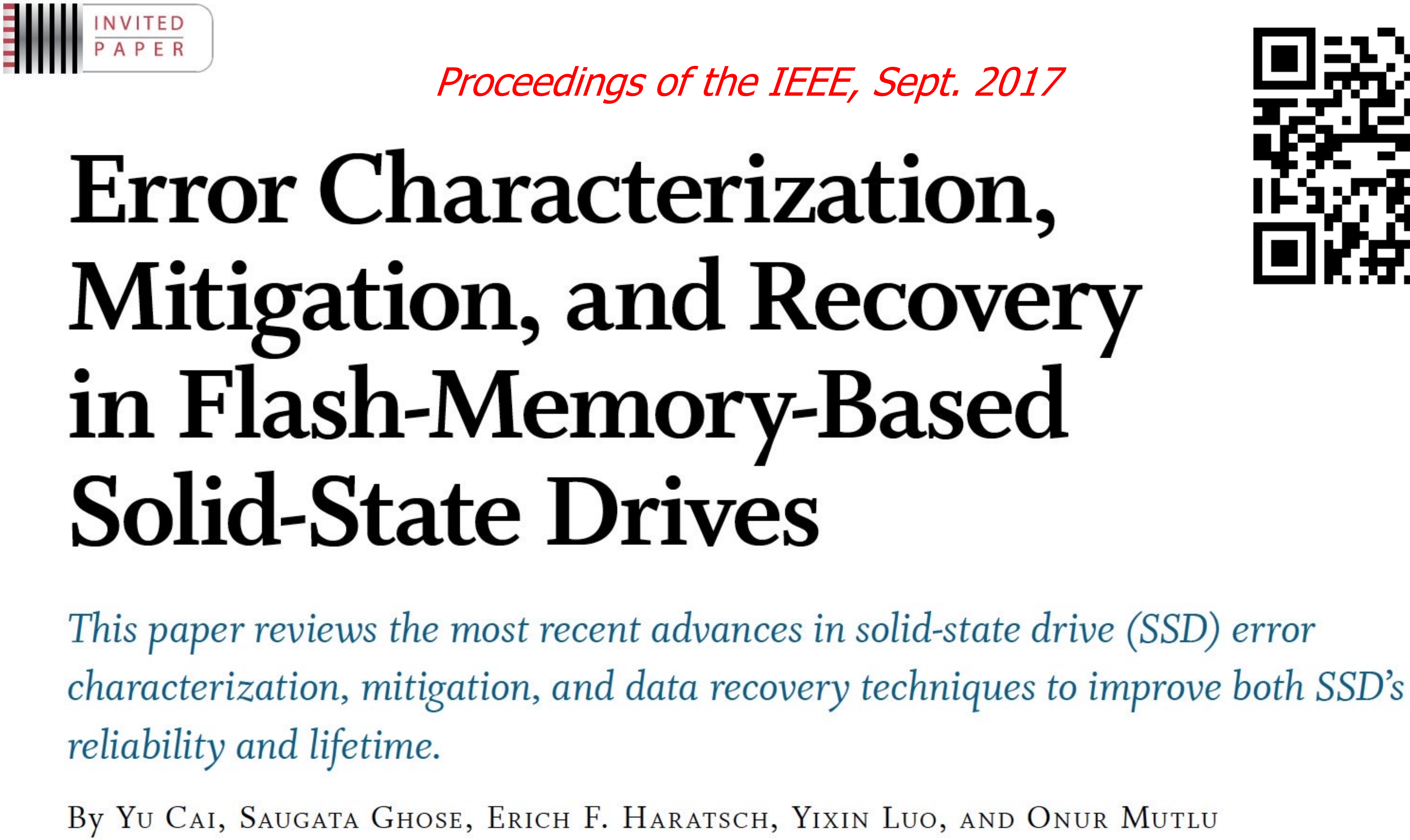
Aside: Intelligent Controller for NAND Flash



[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

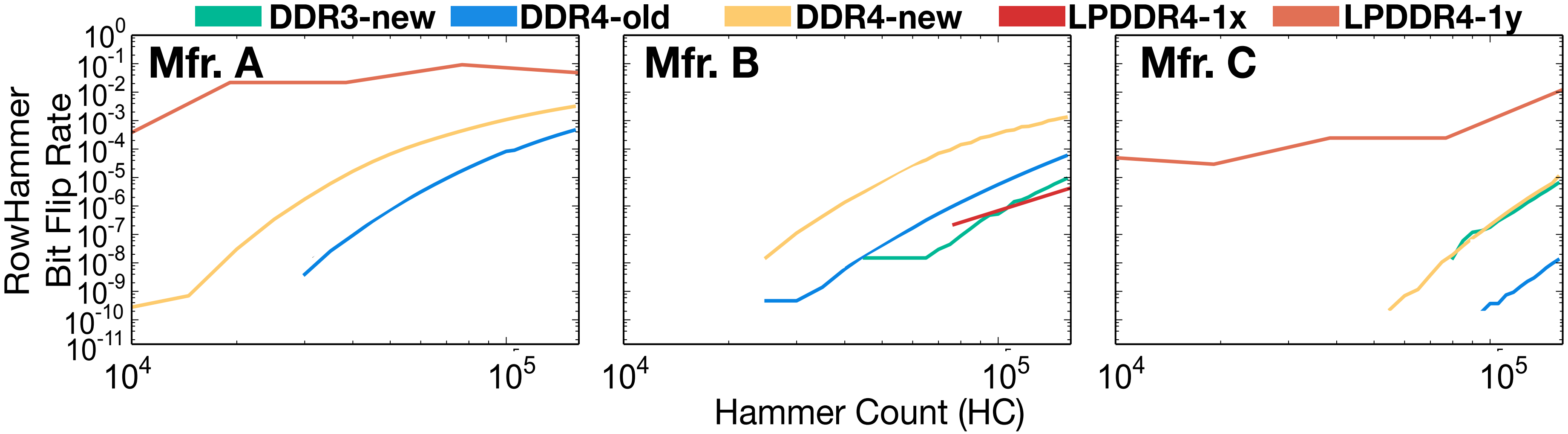
DRAM type-node	Number of Chips (Modules) Tested			
	Mfr. A	*Mfr. B*	*Mfr. C*	*Total*
DDR3-old	56 (10)	88 (11)	28 (7)	**172 (28)**
DDR3-new	80 (10)	52 (9)	104 (13)	**236 (32)**
DDR4-old	112 (16)	24 (3)	128 (18)	**264 (37)**
DDR4-new	264 (43)	16 (2)	108 (28)	**388 (73)**
LPDDR4-1x	12 (3)	180 (45)	N/A	**192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

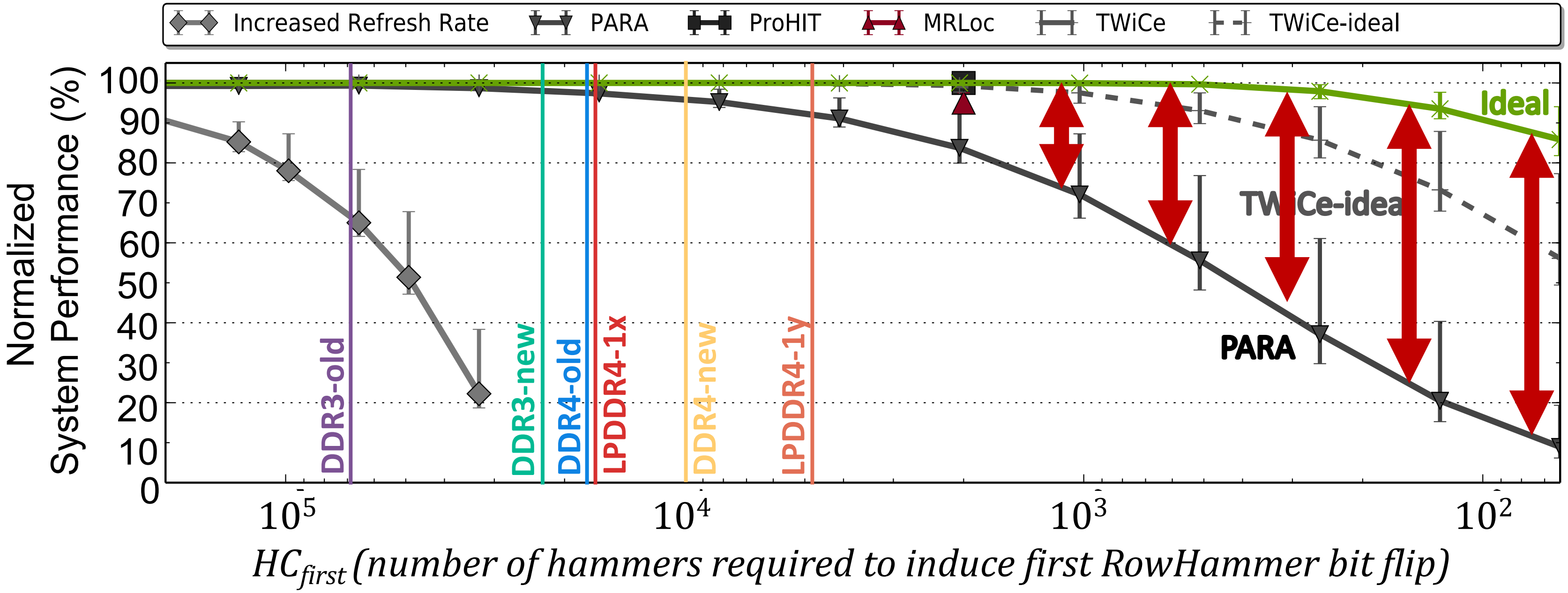
3. Hammer Count (HC) Effects



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}

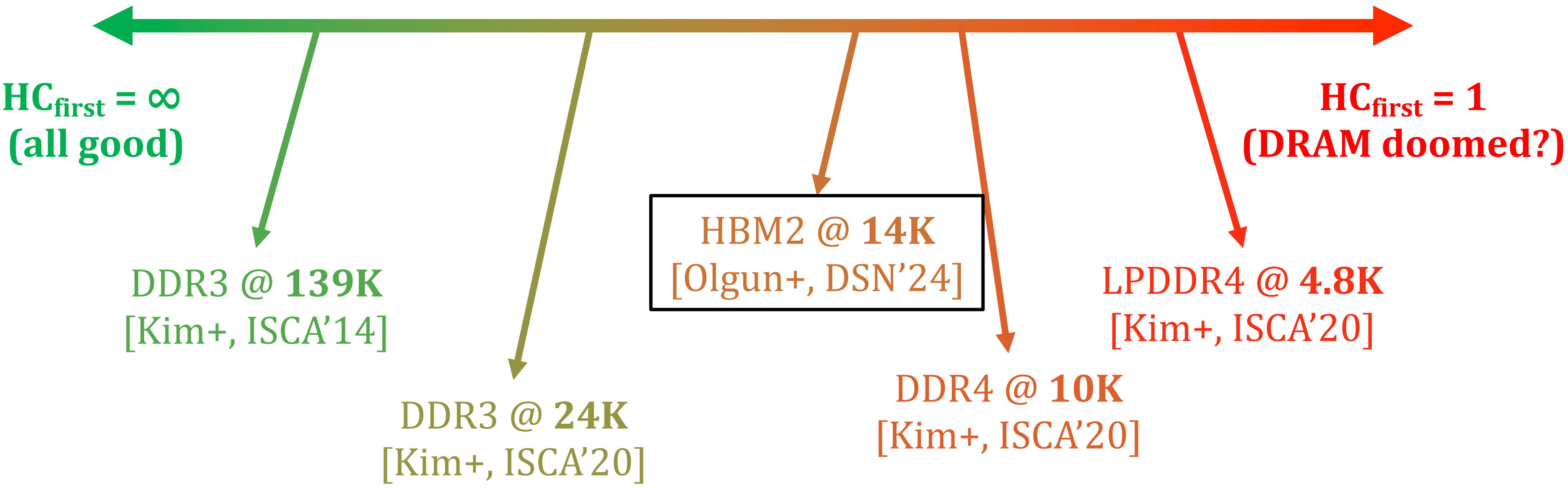
Key Takeaways

RowHammer is a technology scaling problem

Finding a good solution to RowHammer is difficult (and will become more so)

Reported HC_{first} Values (2012 – Now)

HC_{first} : Number of hammer for the first bitflip



$\text{HC}_{\text{first}} = \infty$
(all good)

DDR3 @ **139K**
[Kim+, ISCA’14]

DDR3 @ **24K**
[Kim+, ISCA’20]

HBM2 @ **14K**
[Olgun+, DSN’24]

DDR4 @ **10K**
[Kim+, ISCA’20]

LPDDR4 @ **4.8K**
[Kim+, ISCA’20]

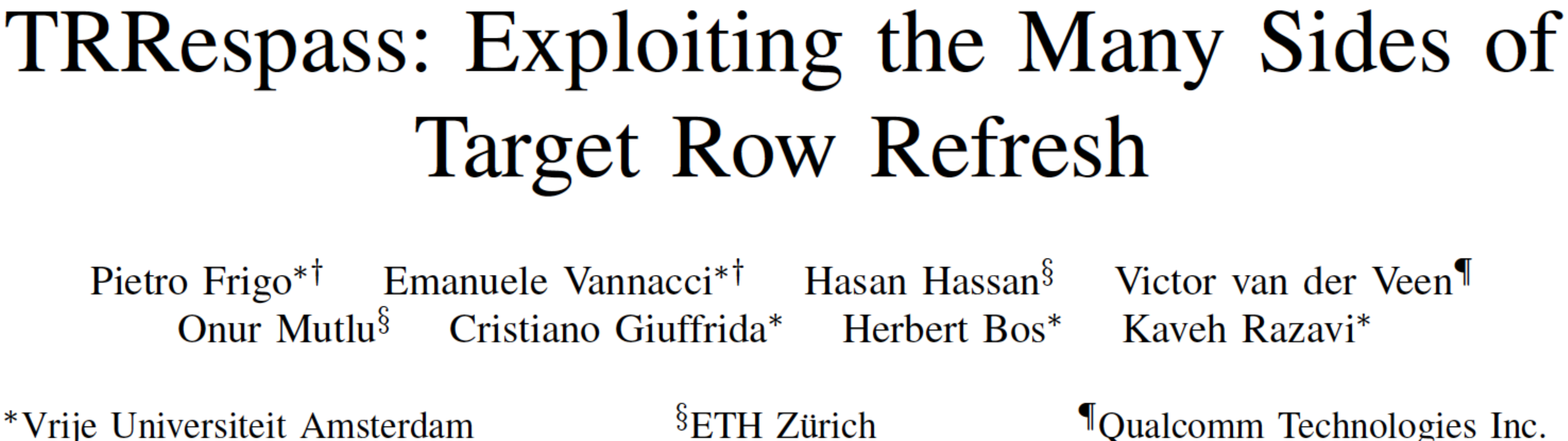
$\text{HC}_{\text{first}} = 1$
(DRAM doomed?)

*Not shown: Significant variance in HC_{first} across vendors and die variations

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

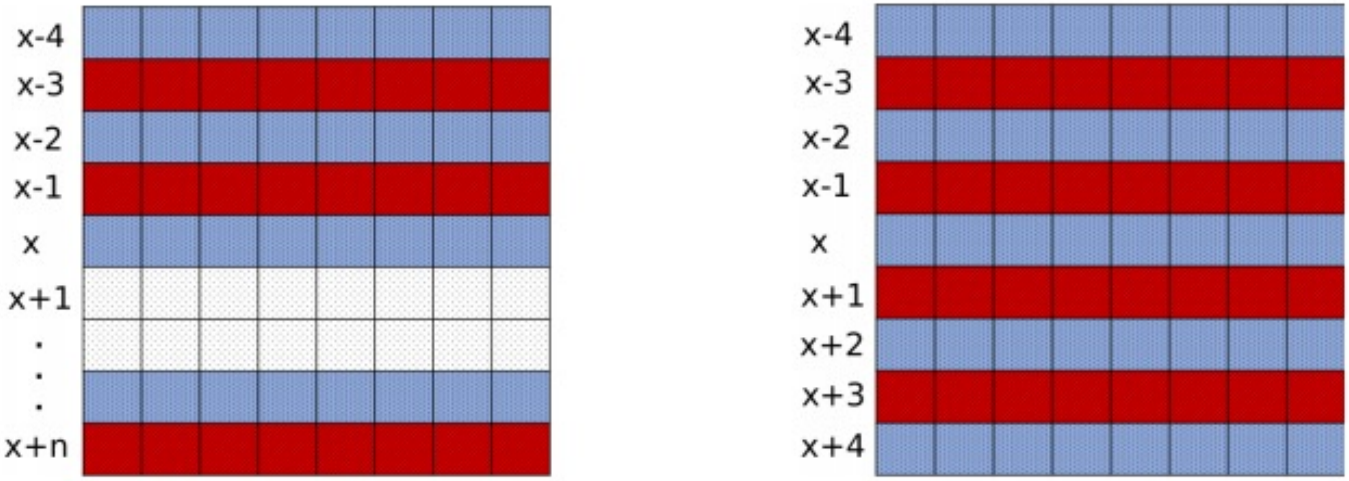
- First work to show that TRR-protected DRAM chips are vulnerable to RowHammer in the field
  - Mitigations advertised as secure are not secure

- 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



(a) Assisted double-sided **(b)** 4-sided

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

BitFlips vs. Number of Aggressor Rows

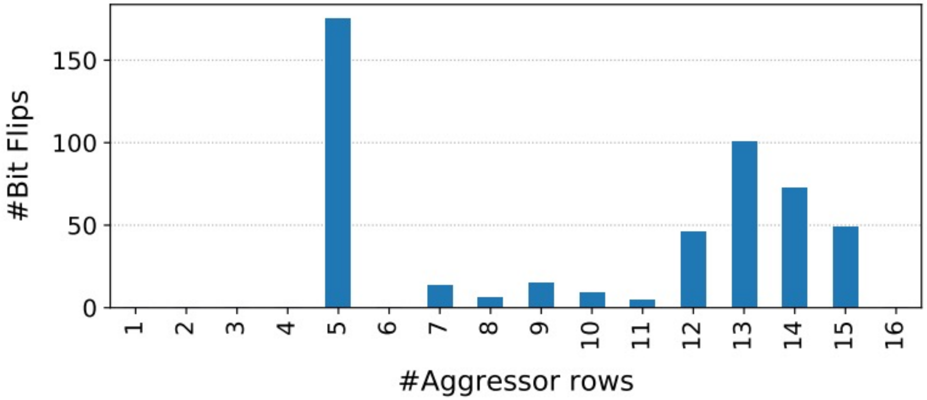


Fig. 10: Bit flips vs. number of aggressor rows. Module $\mathcal{C}_{12}$: Number of bit flips in bank 0 as we vary the number of aggressor rows. Using SoftMC, we refresh DRAM with standard `tREFI` and run the tests until each aggressor rows is hammered 500K times.

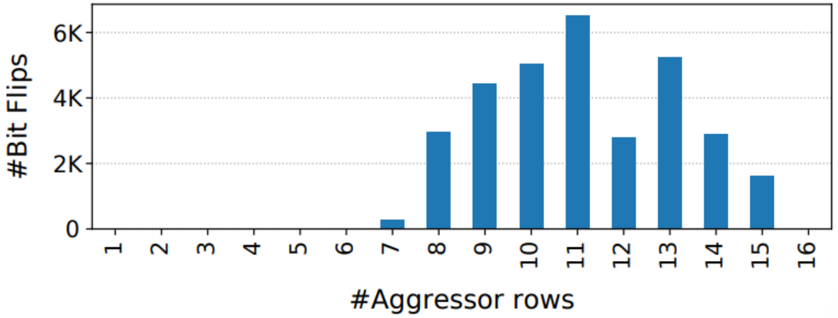


Fig. 11: Bit flips vs. number of aggressor rows. Module $\mathcal{A}_{15}$: Number of bit flips in bank 0 as we vary the number of aggressor rows. Using SoftMC, we refresh DRAM with standard `tREFI` and run the tests until each aggressor rows is hammered 500K times.

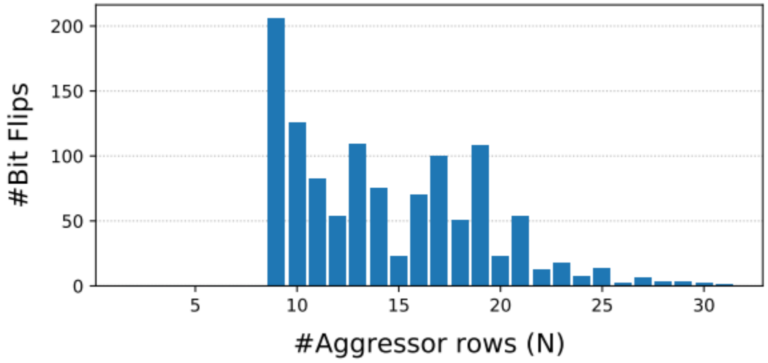


Fig. 13: Bit flips vs. number of aggressor rows. Module $\mathcal{A}_{10}$: Number of bit flips triggered with *N-sided* RowHammer for varying number of *N* on Intel Core i7-7700K. Each aggressor row is one row away from the closest aggressor row (i.e., `VAVAVA...` configuration) and aggressor rows are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

TRRespass Vulnerable DRAM Modules

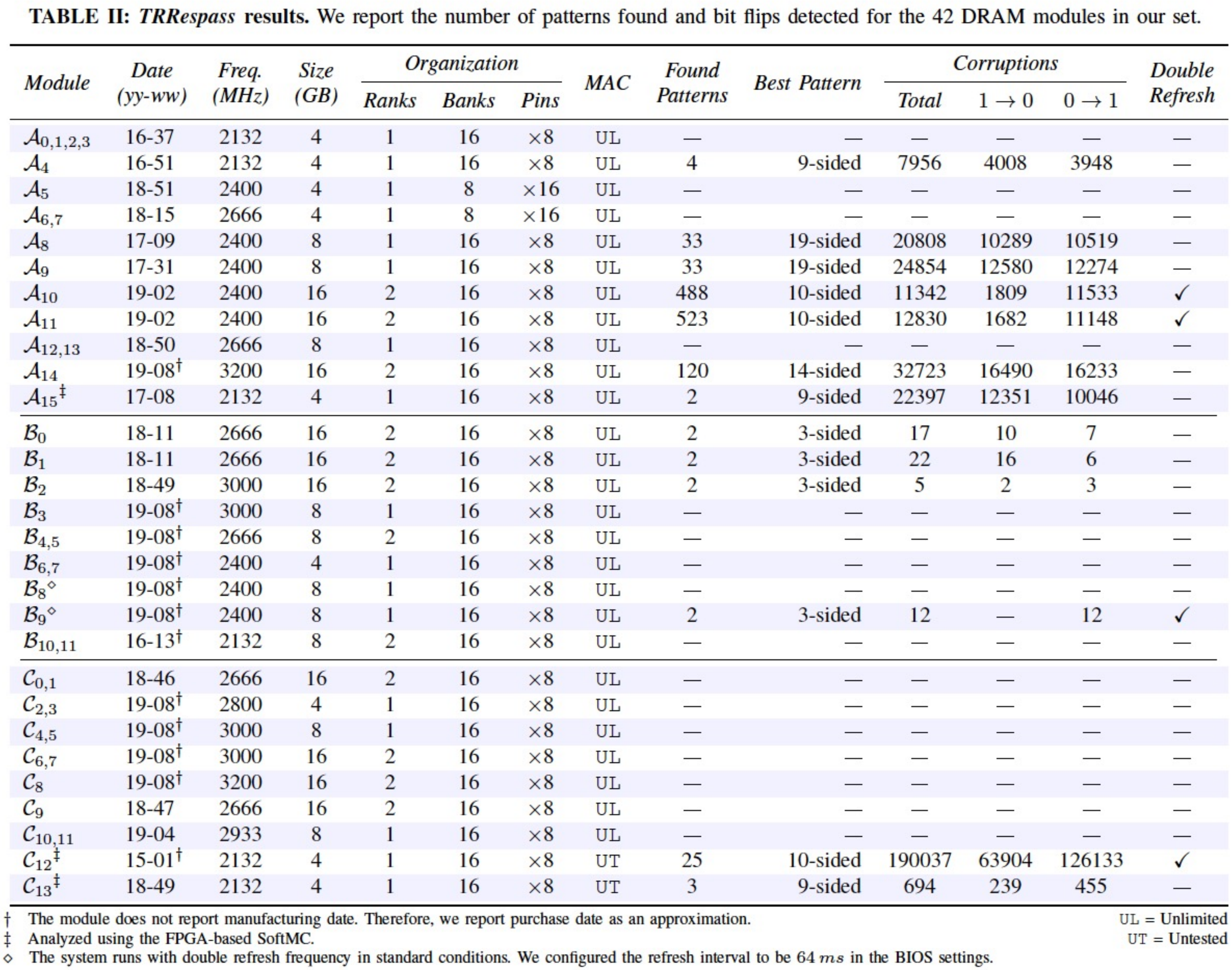


TABLE II: *TRRespass* results. We report the number of patterns found and bit flips detected for the 42 DRAM modules in our set.

Module	*Date (yy-ww)*	*Freq. (MHz)*	*Size (GB)*	*Organization*			*MAC*	*Found Patterns*	*Best Pattern*	*Corruptions*			*Double Refresh*
				Ranks	*Banks*	*Pins*				*Total*	1 → 0	0 → 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

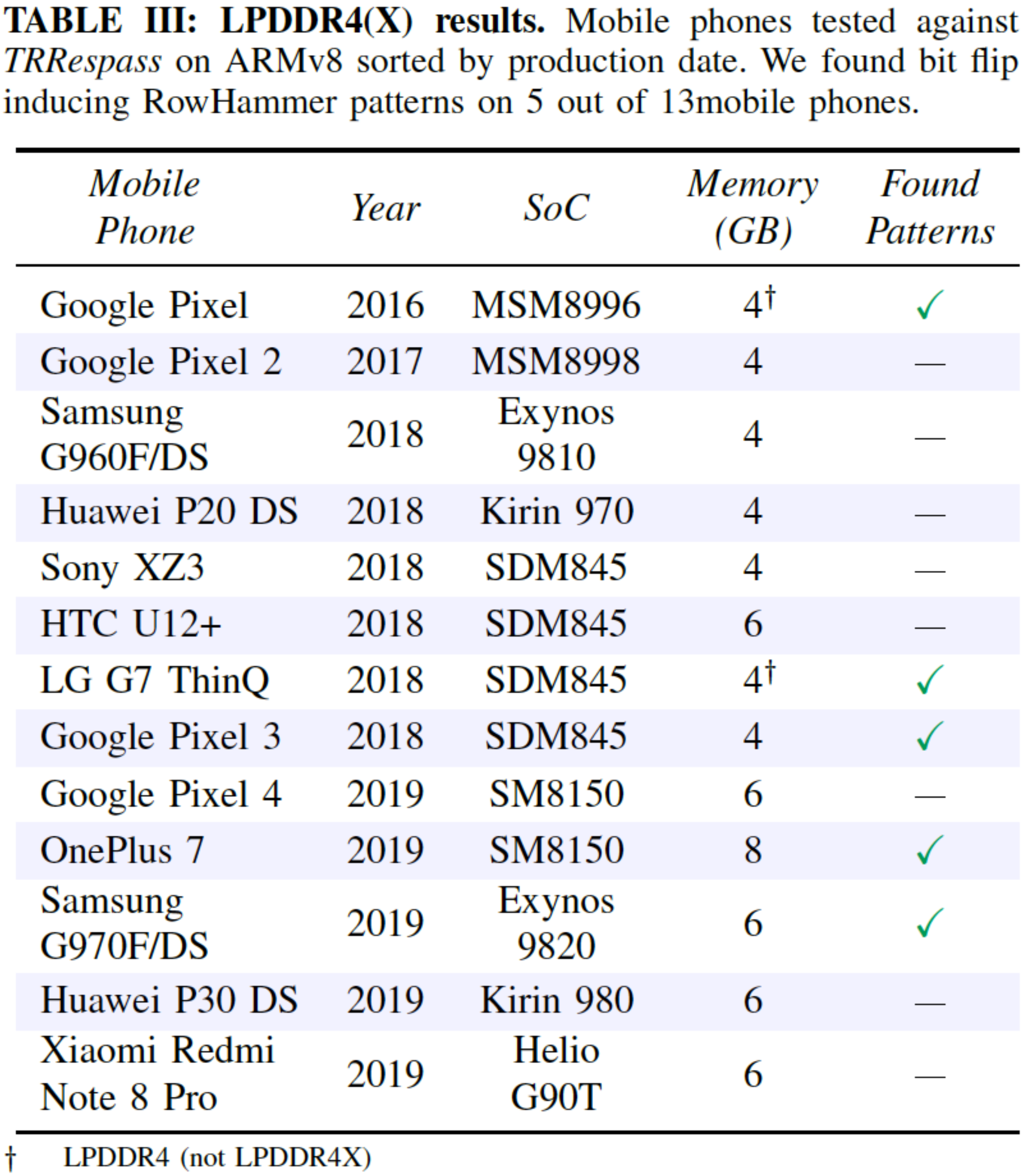


TABLE III: LPDDR4(X) results. Mobile phones tested against *TRRespass* on ARMv8 sorted by production date. We found bit flip inducing RowHammer patterns on 5 out of 13mobile phones.

Mobile Phone	*Year*	*SoC*	*Memory (GB)*	*Found Patterns*
Google Pixel	2016	MSM8996	4†	✓
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†	✓
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

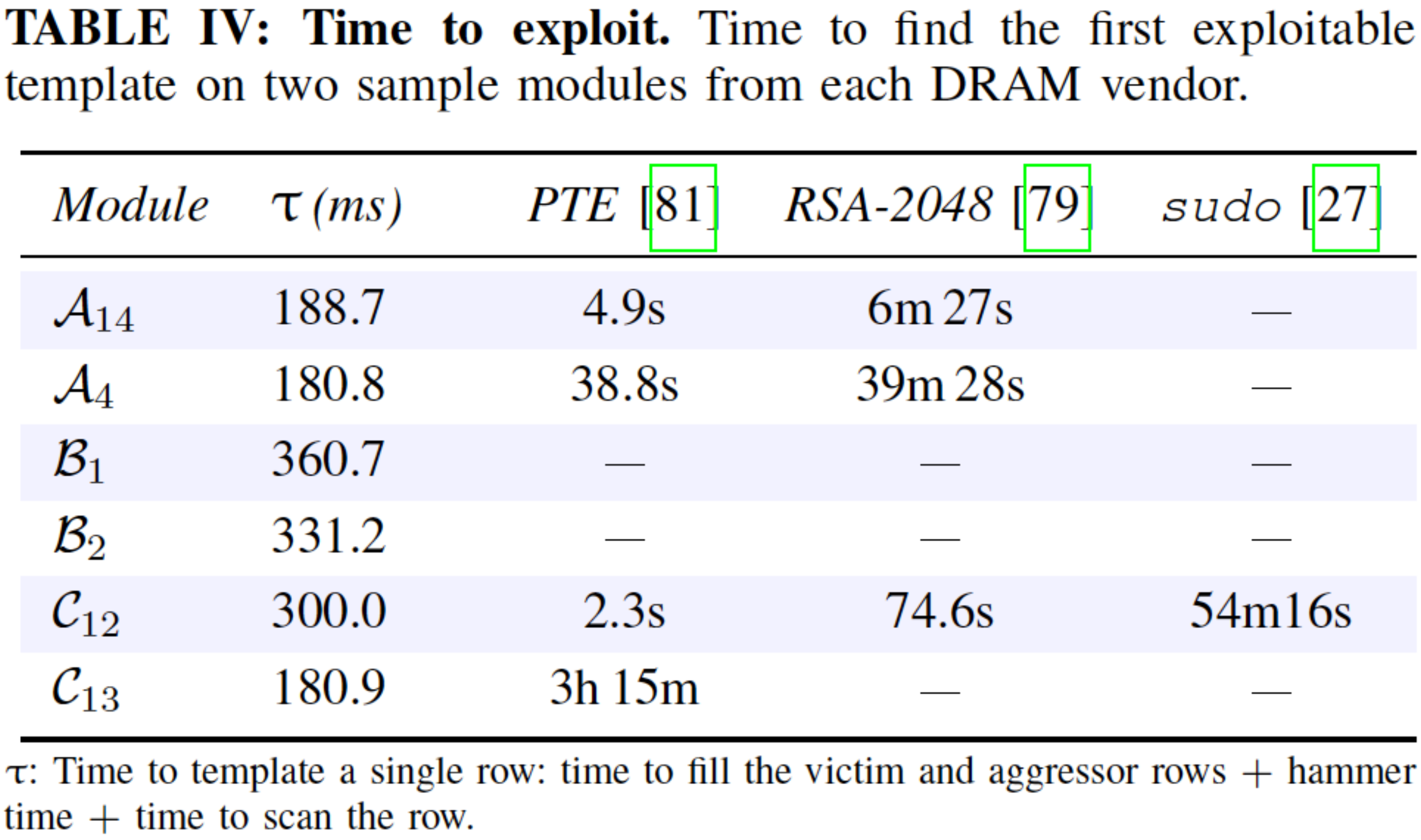


TABLE IV: Time to exploit. Time to find the first exploitable template on two sample modules from each DRAM vendor.

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

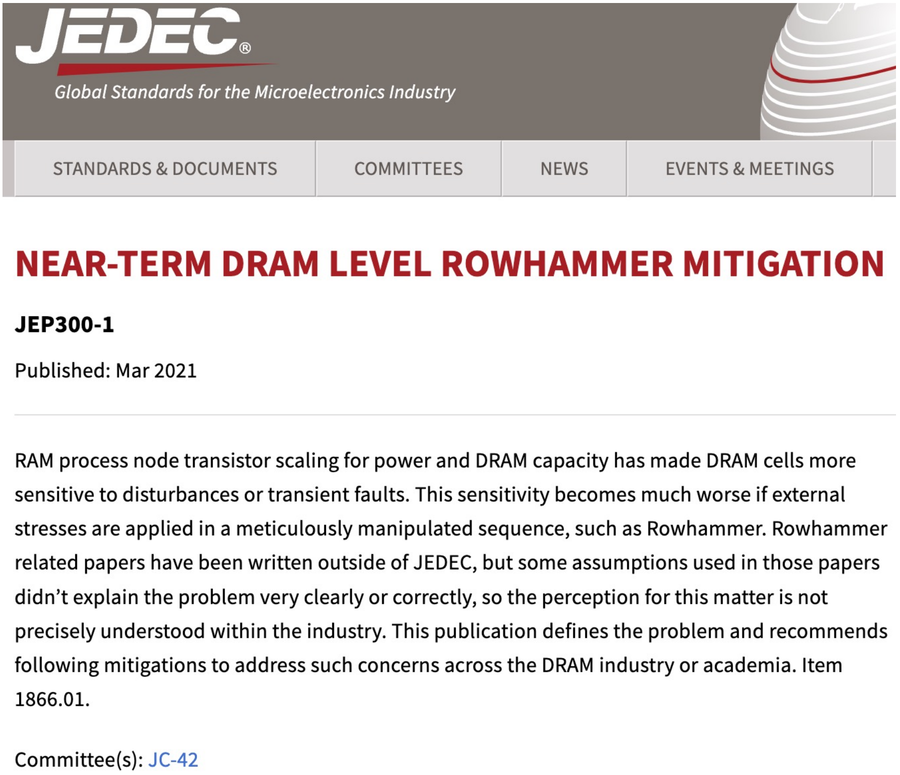
- These results are scratching the surface
  - TRRespass tool is not exhaustive
  - There is a lot of room for uncovering more vulnerable chips and phones

TRRespass Key Takeaways

RowHammer is still
an open problem

Security by obscurity
is likely not a good solution

Improvements in JEDEC (2020-2021)



NEAR-TERM DRAM LEVEL ROWHAMMER MITIGATION

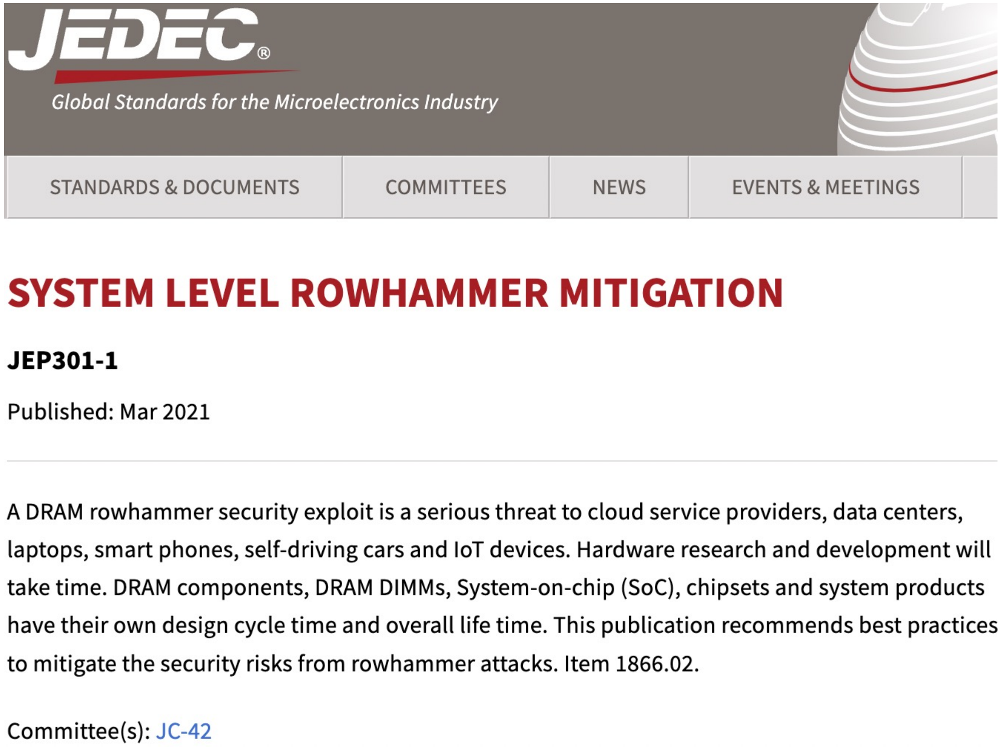
JEP300-1

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



SYSTEM LEVEL ROWHAMMER MITIGATION

JEP301-1

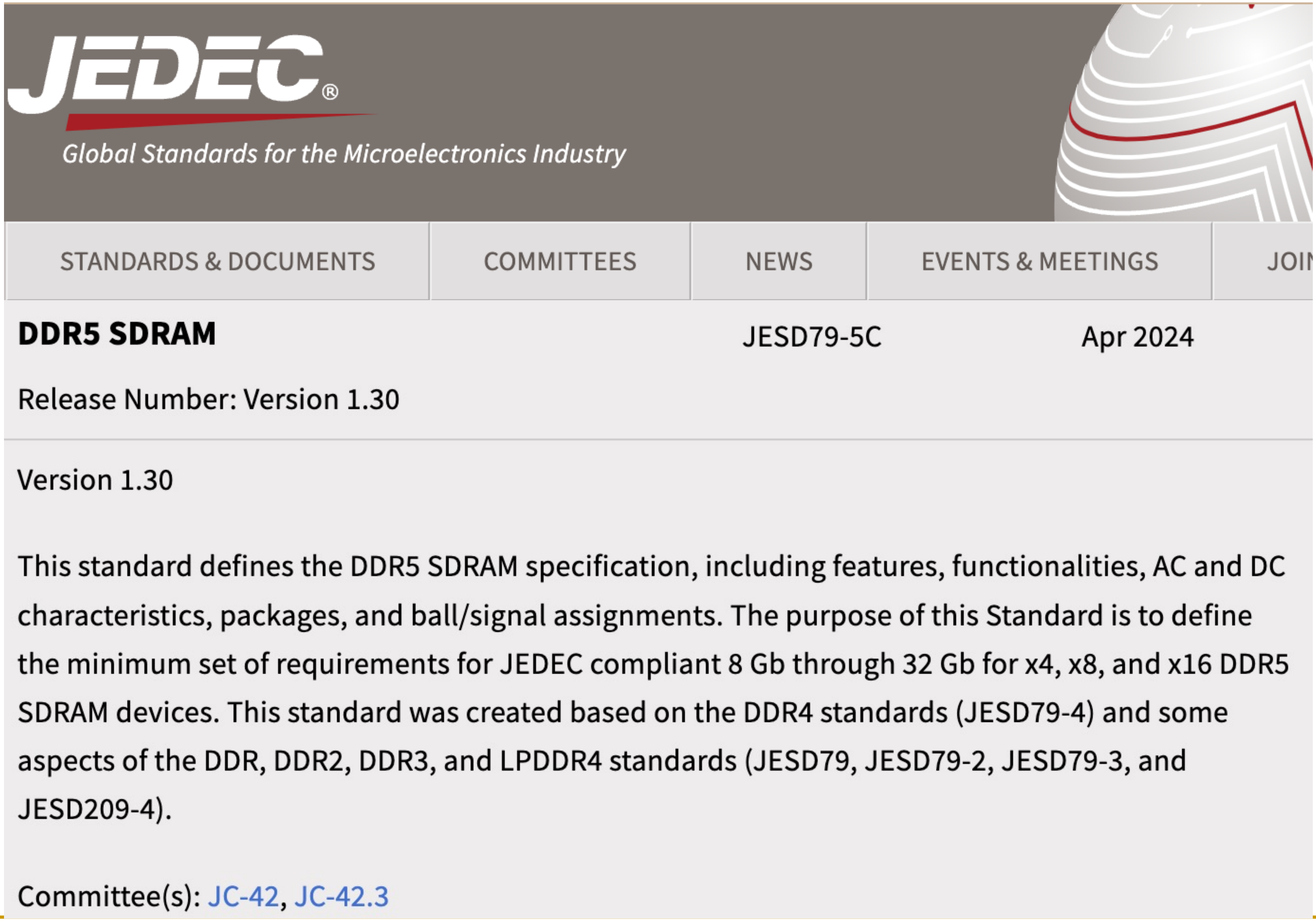
Published: Mar 2021

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)



DDR5 SDRAM JESD79-5C Apr 2024

Release Number: Version 1.30

Version 1.30

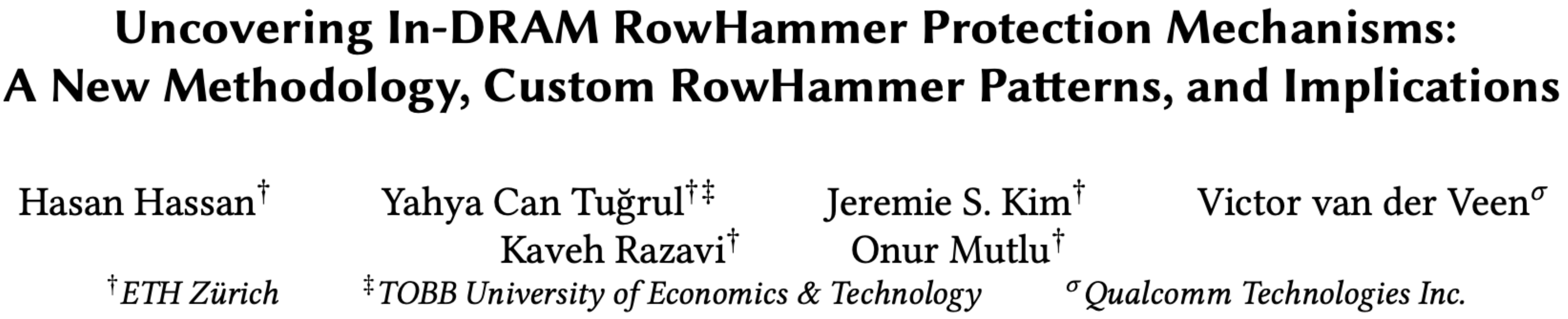
This standard defines the DDR5 SDRAM specification, including features, functionalities, AC and DC characteristics, packages, and ball/signal assignments. The purpose of this Standard is to define the minimum set of requirements for JEDEC compliant 8 Gb through 32 Gb for x4, x8, and x16 DDR5 SDRAM devices. 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

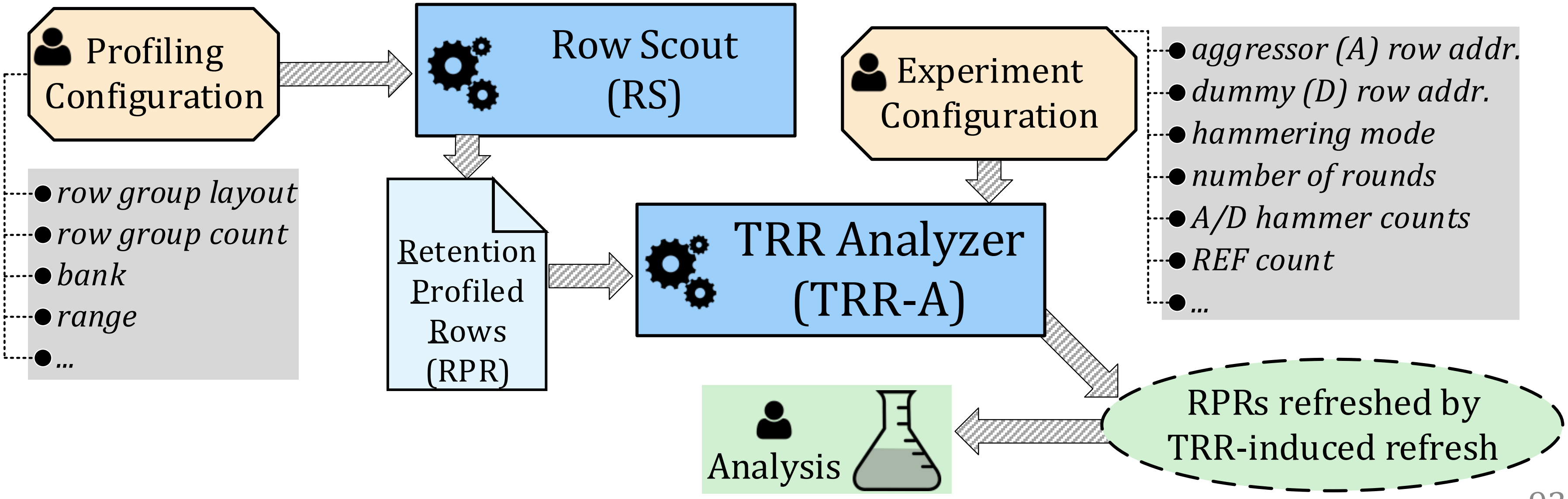
Hasan Hassan[†] Yahya Can Tuğrul[†‡] Jeremie S. Kim[†] Victor van der Veen[σ]
Kaveh Razavi[†] Onur Mutlu[†]
[†] *ETH Zürich* [‡] *TOBB University of Economics & Technology* [σ] *Qualcomm Technologies Inc.*

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



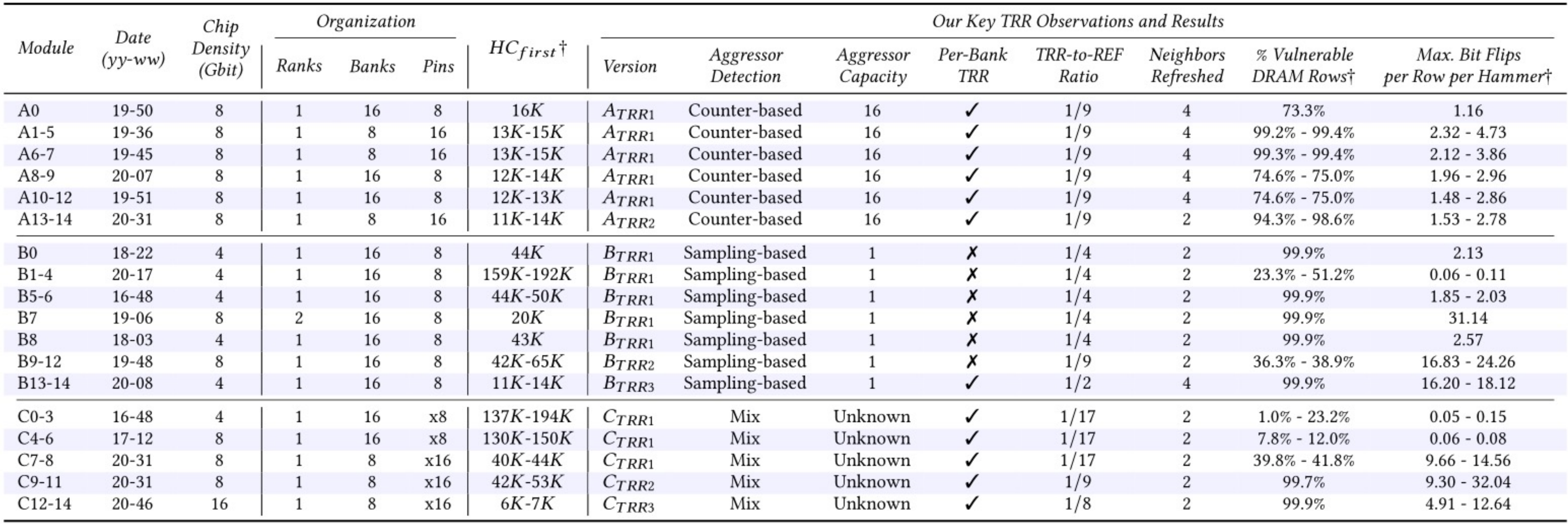
https://github.com/CMU-SAFARI/U-TRR

Key Takeaways

All 45 modules we test are vulnerable

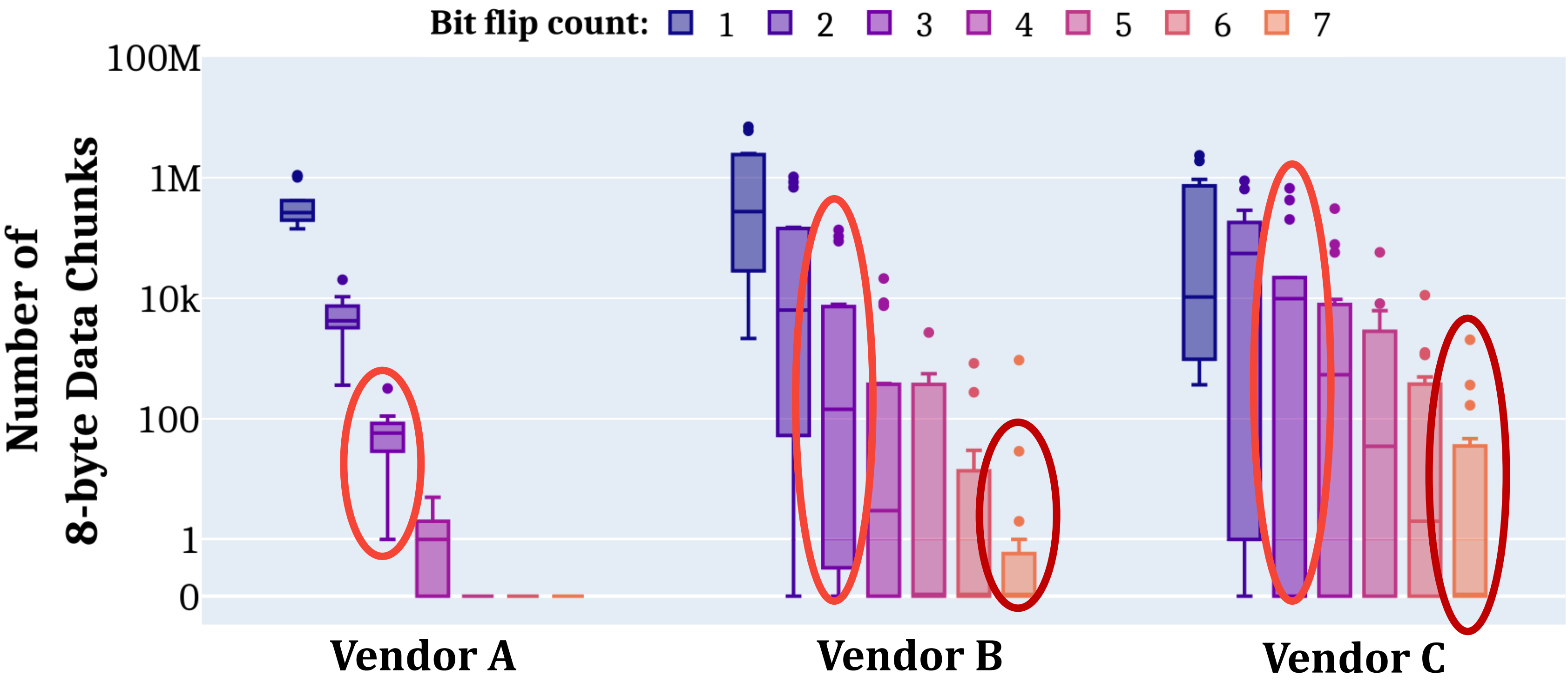
99.9% of rows in a DRAM bank
experience **at least one RowHammer bit flip**

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

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

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**

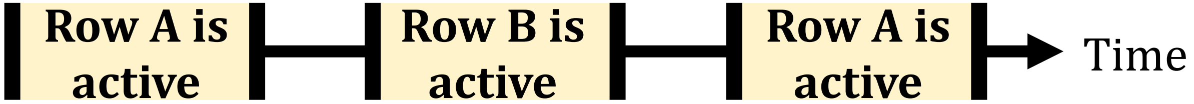
- **6 major takeaways** from **16 novel observations**

- 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

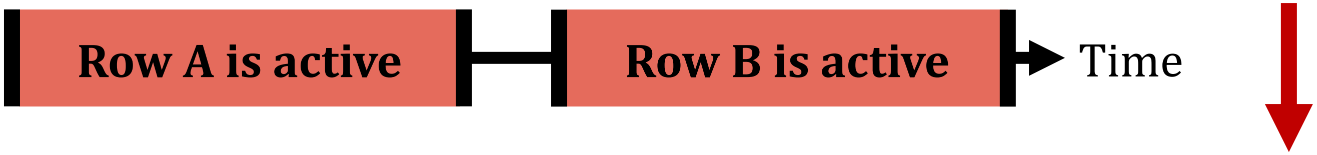
- Our novel observations can inspire and aid future work
  - Craft **more effective attacks**
  - Design **more effective and efficient defenses**

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:



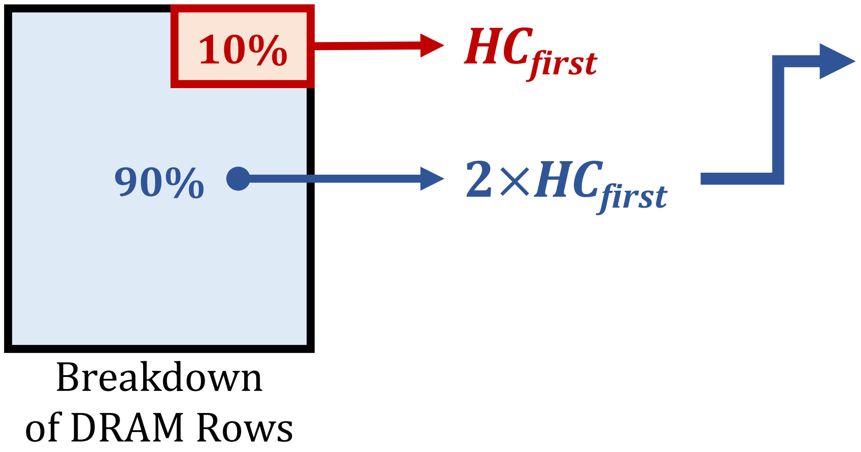
36% reduction in *HC_{first}*

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**

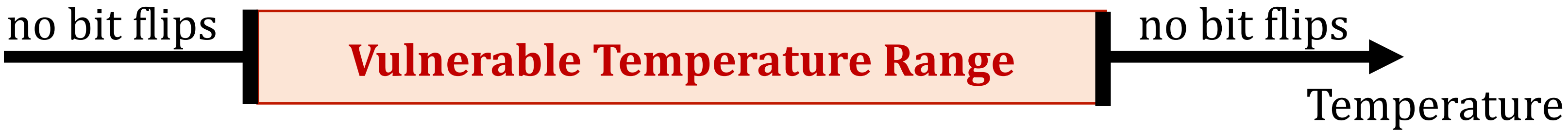


Aggressiveness can be reduced:

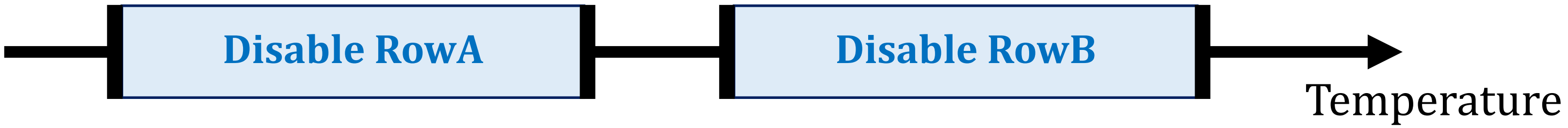
33% area reduction
for BlockHammer [Yağlıkçı+, HPCA'21]

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

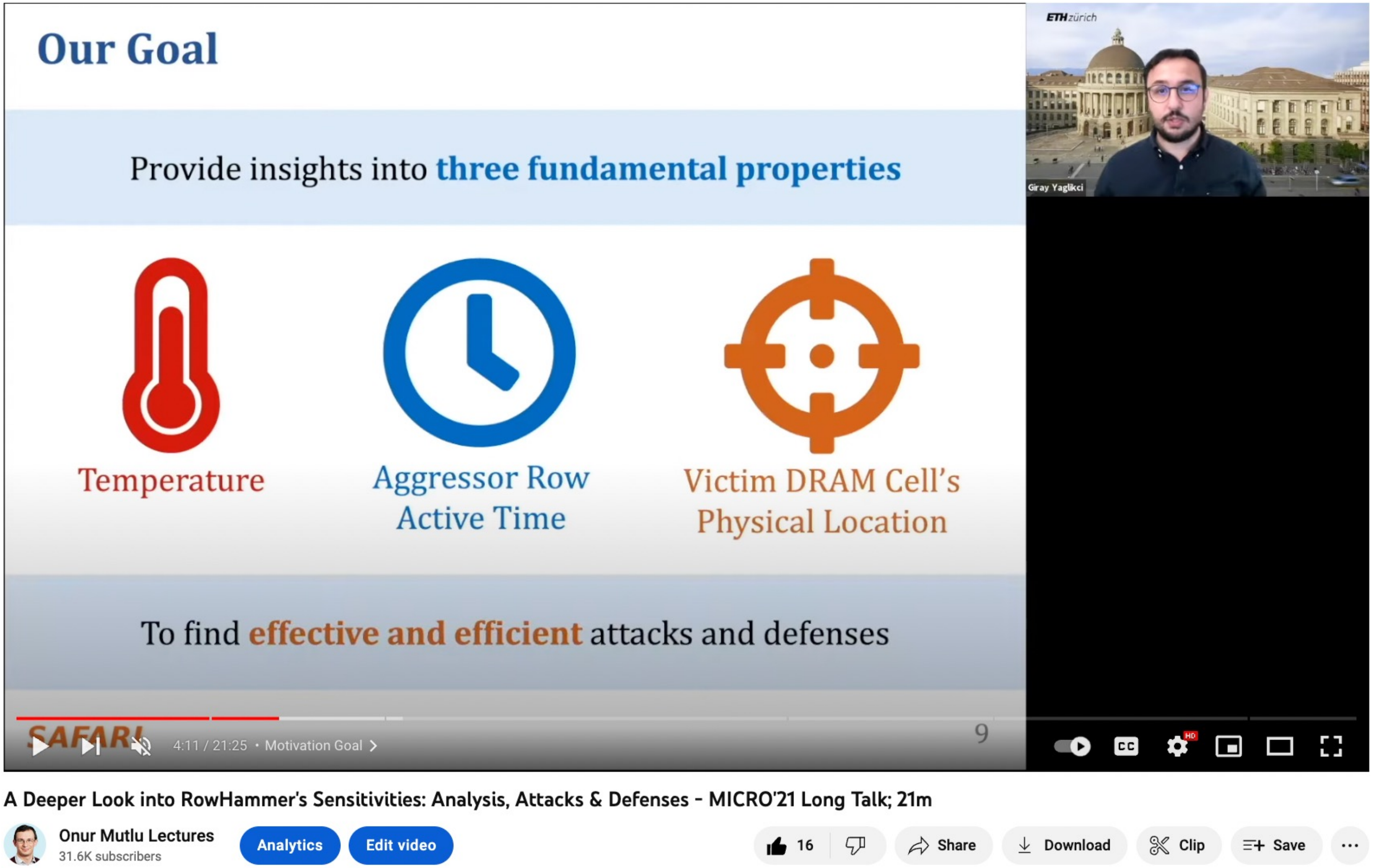
- **Example 2: Leveraging variation with temperature**
  - A DRAM cell experiences **bit flips** within **a bounded temperature range**



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



Deeper Look into RowHammer: Talk Video



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

SAFARI 9

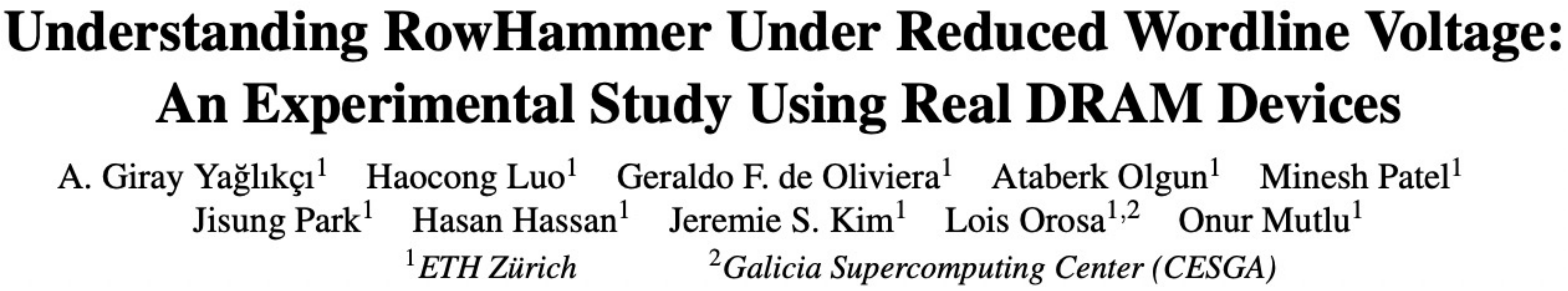
A Deeper Look into RowHammer's Sensitivities: Analysis, Attacks & Defenses - MICRO'21 Long Talk; 21m

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)]

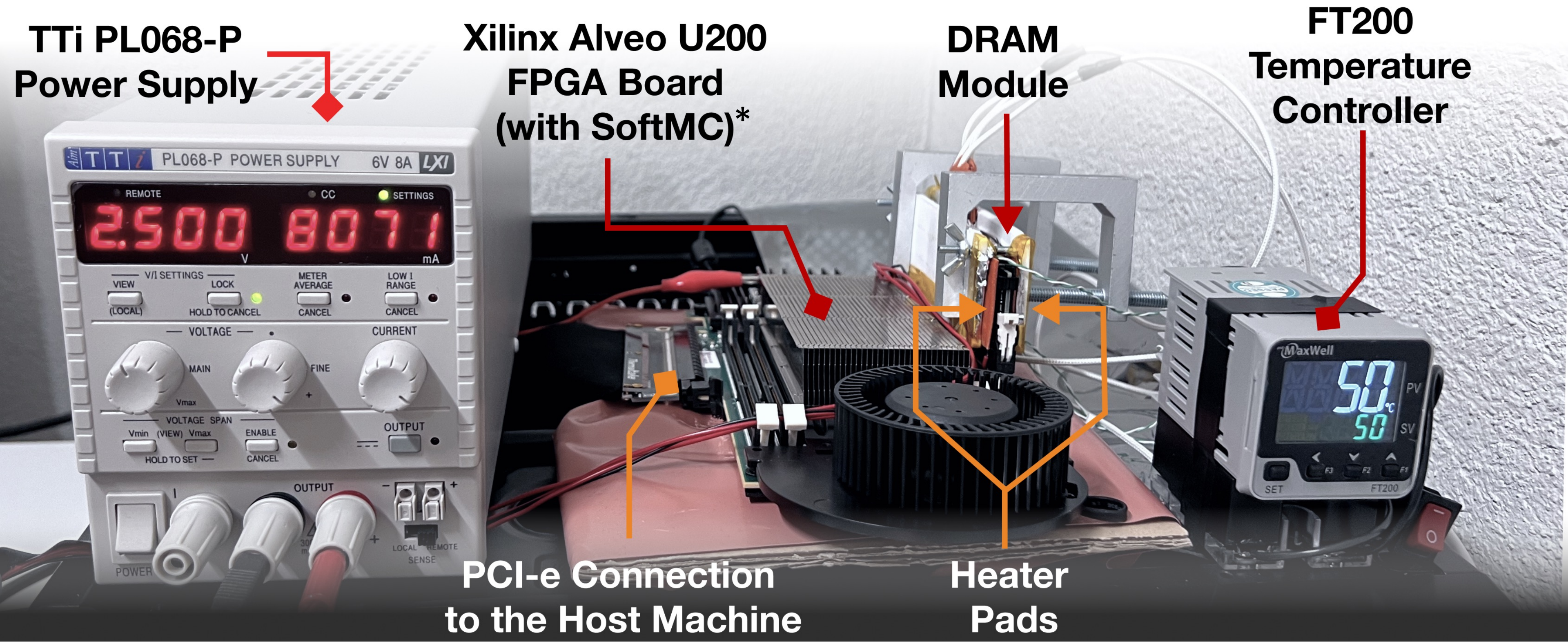


Understanding RowHammer Under Reduced Wordline Voltage: An Experimental Study Using Real DRAM Devices

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)**

SAFARI

*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

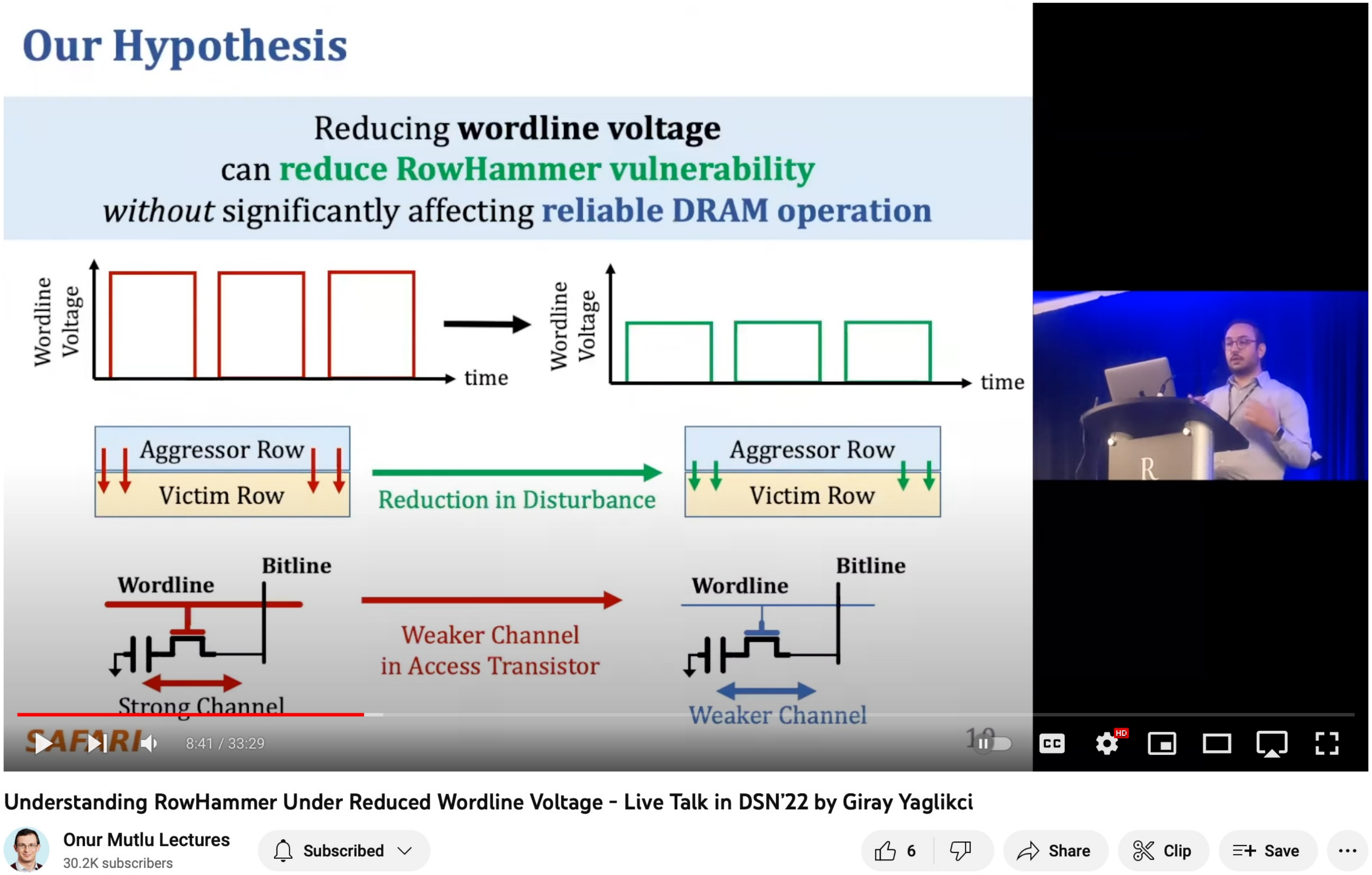
- More than **76%** of the tested DRAM chips **reliably operate** using **nominal** timing parameters
- Remaining **24%** **reliably operate** with **increased** (up to 24ns) row activation latency

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



Our Hypothesis

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

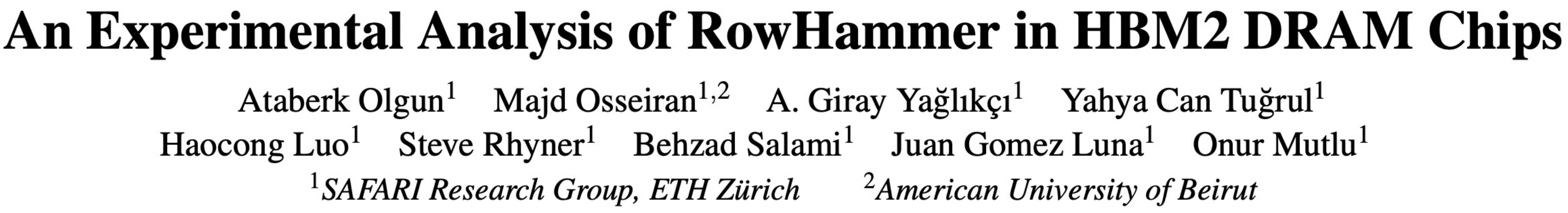
Understanding RowHammer Under Reduced Wordline Voltage - Live Talk in DSN'22 by Giray Yaglikci

Onur Mutlu Lectures
30.2K subscribers

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)]



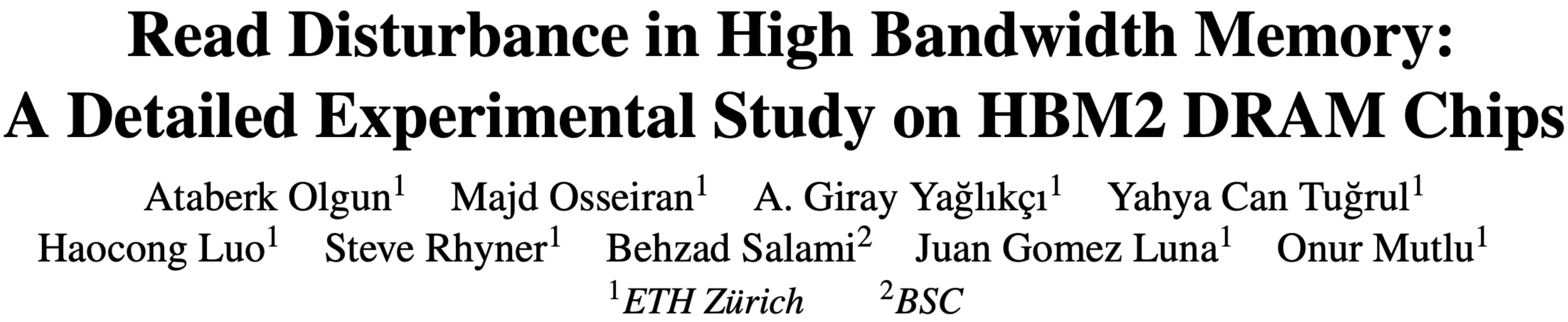
An Experimental Analysis of RowHammer in HBM2 DRAM Chips

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**



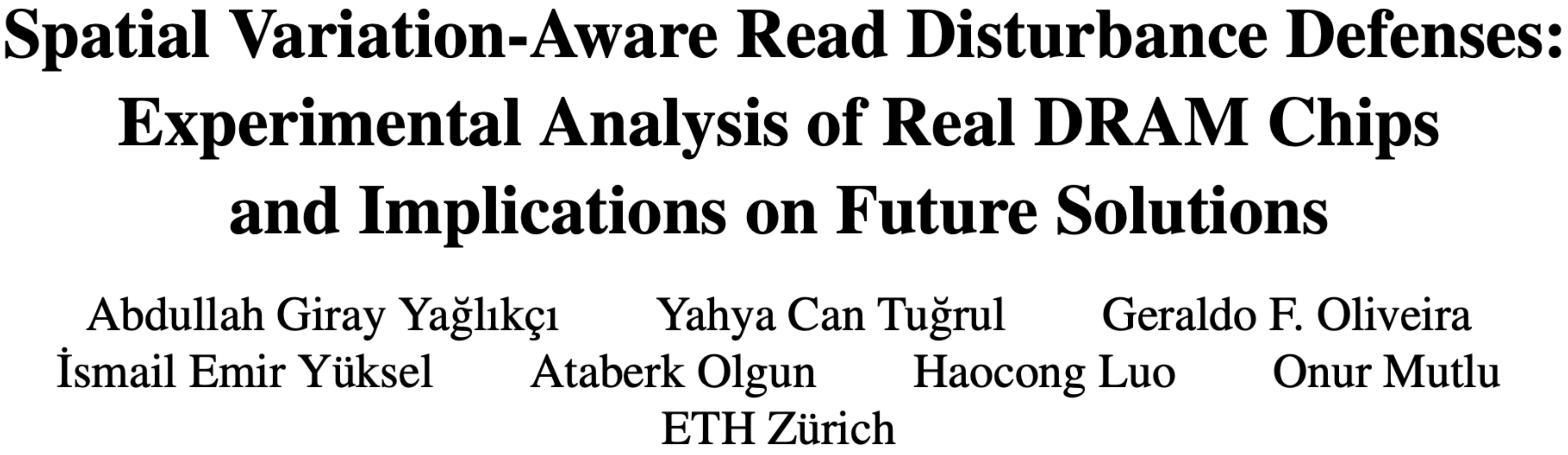
Read Disturbance in High Bandwidth Memory: A Detailed Experimental Study on HBM2 DRAM Chips

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**



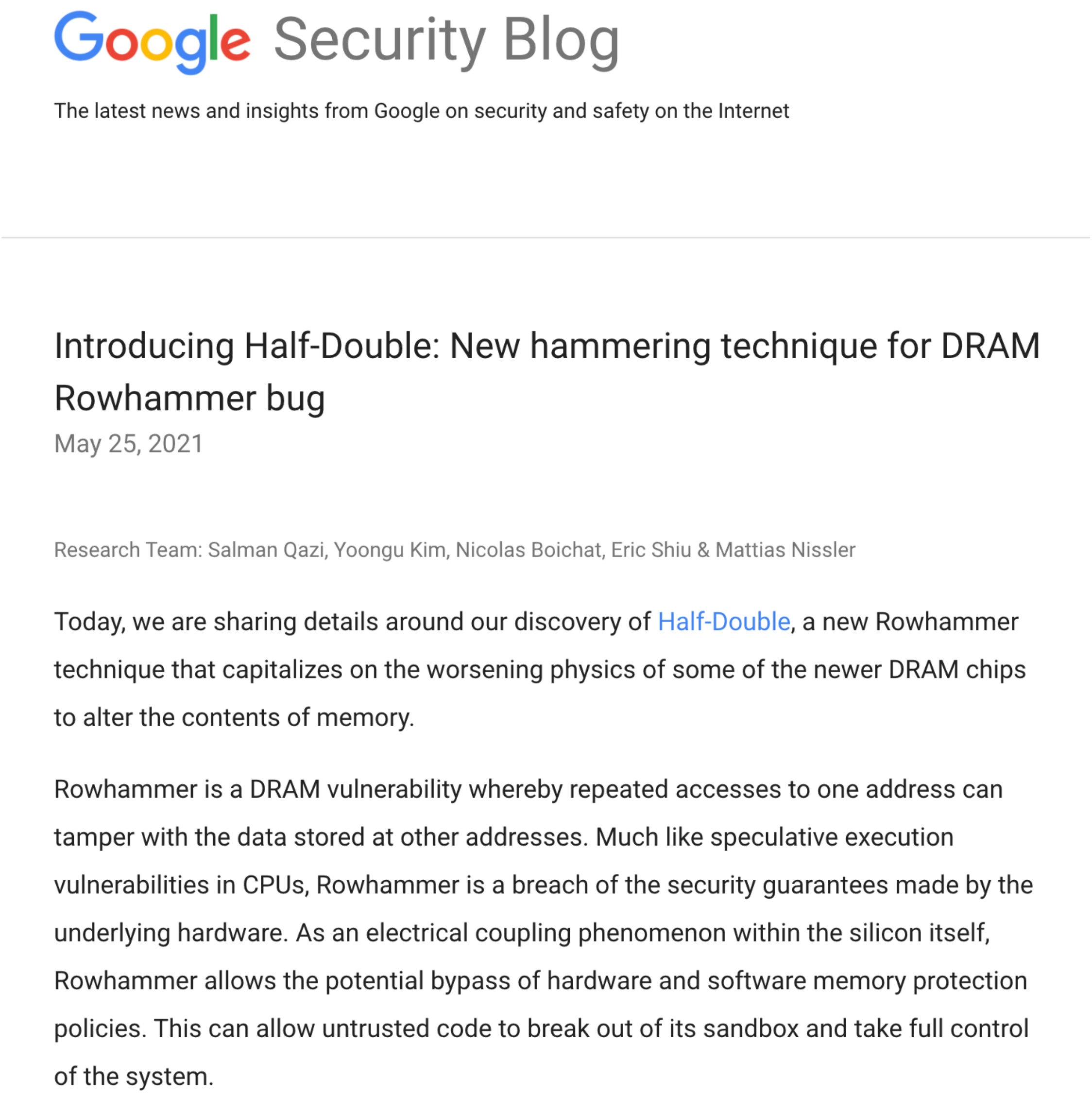
Spatial Variation-Aware Read Disturbance Defenses: Experimental Analysis of Real DRAM Chips and Implications on Future Solutions

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 Attacks & Solutions

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

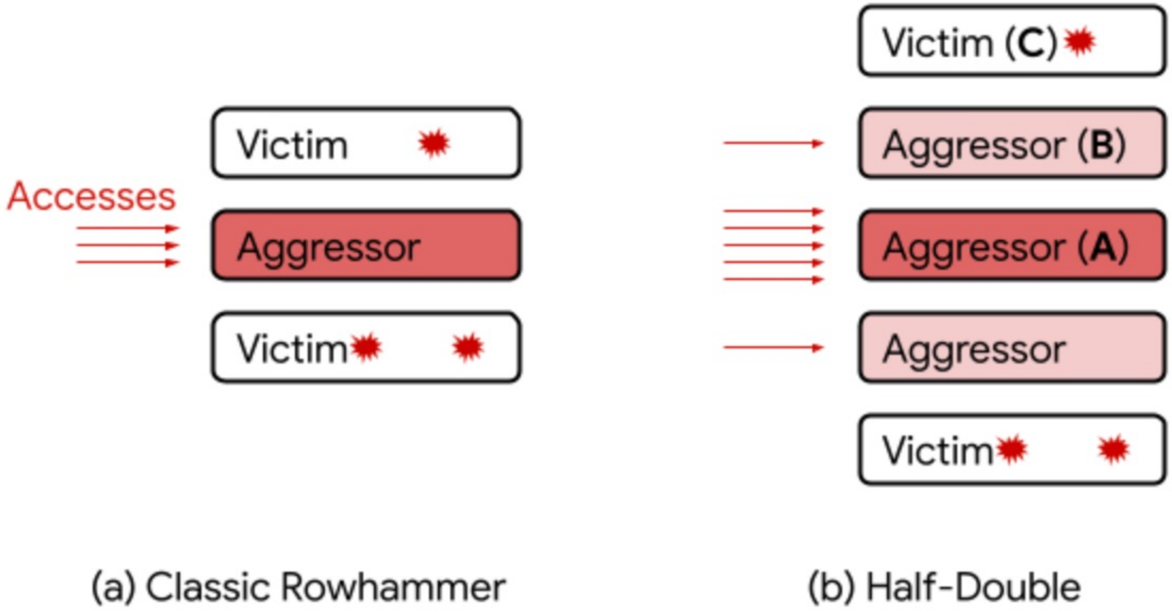
Research Team: Salman Qazi, Yoongu Kim, Nicolas Boichat, Eric Shiu & Mattias Nissler

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

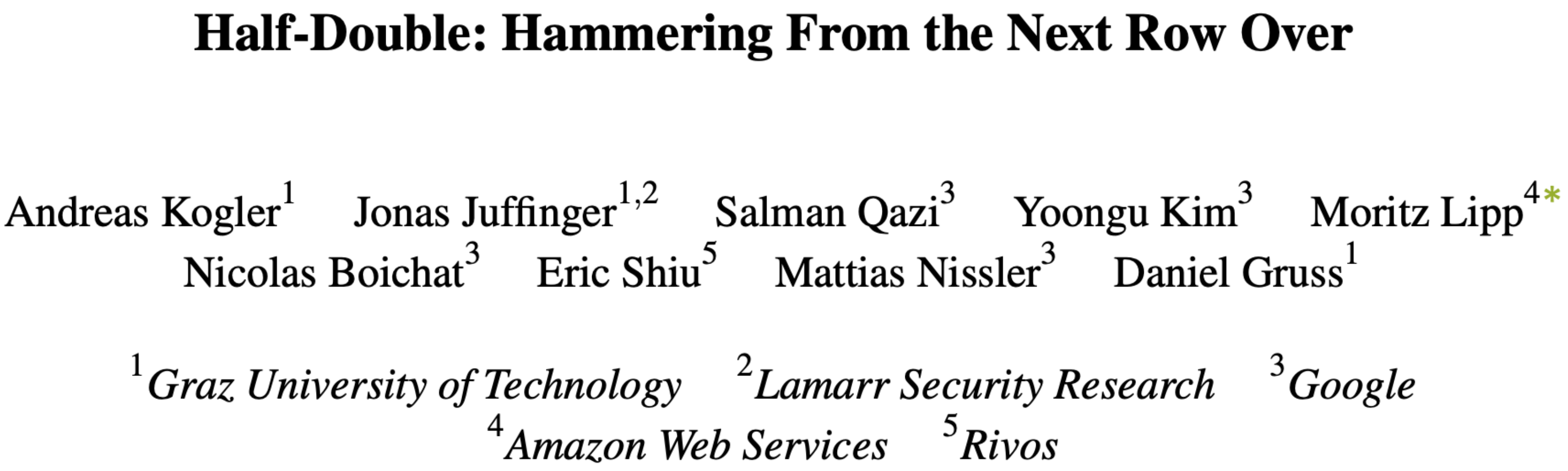
(b) Half-Double

- 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.

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

Google's Half-Double RowHammer Attack

- **Appears at USENIX Security 2022**



Half-Double: Hammering From the Next Row Over

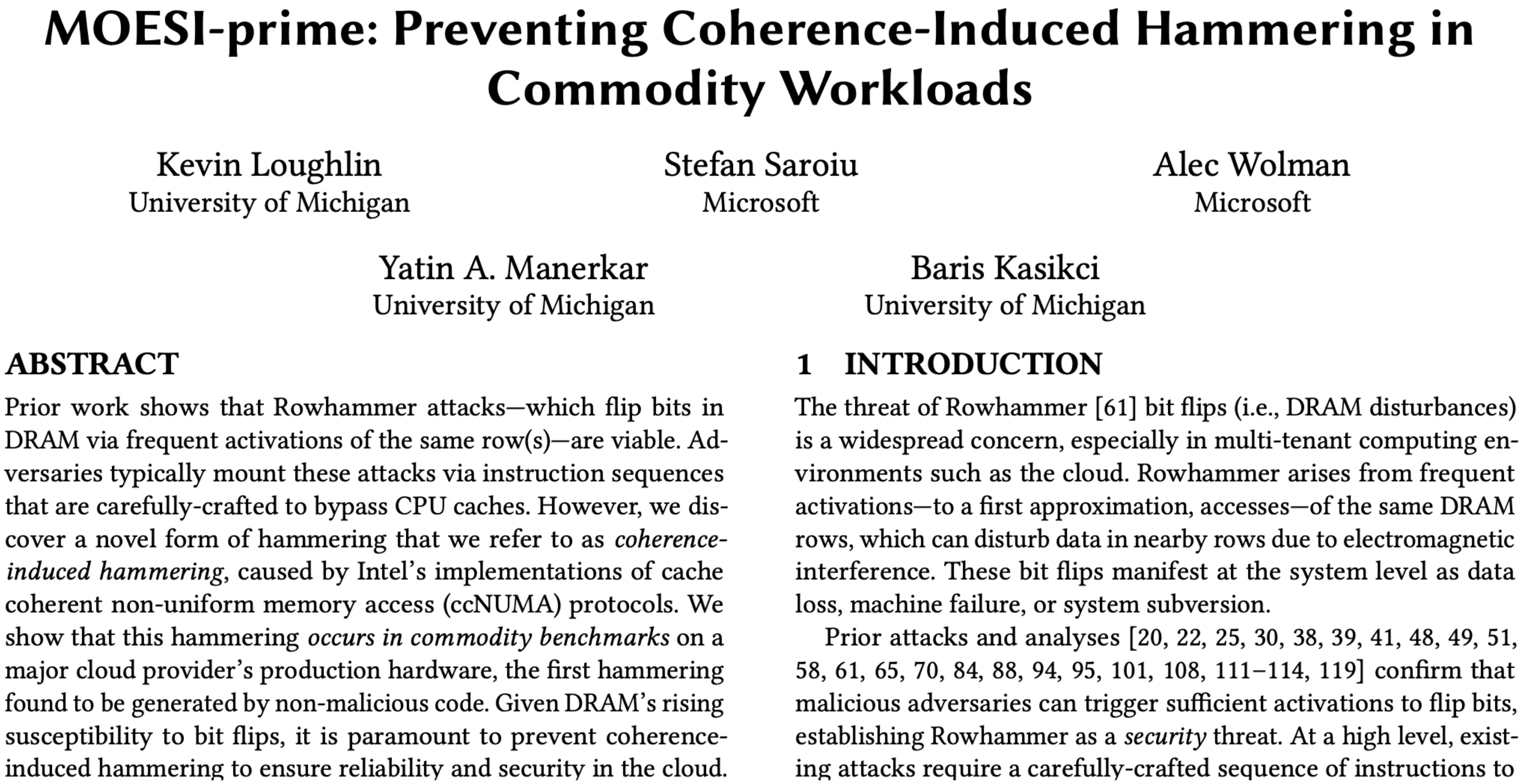
Andreas Kogler[1] Jonas Juffinger[1,2] Salman Qazi[3] Yoongu Kim[3] Moritz Lipp[4]*
Nicolas Boichat[3] Eric Shiu[5] Mattias Nissler[3] Daniel Gruss[1]

[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

Microsoft's MOESI-prime Work **[ISCA'22]**

- Introduces coherence-induced hammering
- Hammering in commodity workloads (non-malicious code)



MOESI-prime: Preventing Coherence-Induced Hammering in Commodity Workloads

Kevin Loughlin
University of Michigan

Stefan Saroiu
Microsoft

Alec Wolman
Microsoft

Yatin A. Manerkar
University of Michigan

Baris Kasikci
University of Michigan

ABSTRACT

Prior work shows that Rowhammer attacks—which flip bits in DRAM via frequent activations of the same row(s)—are viable. Adversaries typically mount these attacks via instruction sequences that are carefully-crafted to bypass CPU caches. However, we discover a novel form of hammering that we refer to as *coherence-induced hammering*, caused by Intel's implementations of cache coherent non-uniform memory access (ccNUMA) protocols. We show that this hammering *occurs in commodity benchmarks* on a major cloud provider's production hardware, the first hammering found to be generated by non-malicious code. Given DRAM's rising susceptibility to bit flips, it is paramount to prevent coherence-induced hammering to ensure reliability and security in the cloud.

1 INTRODUCTION

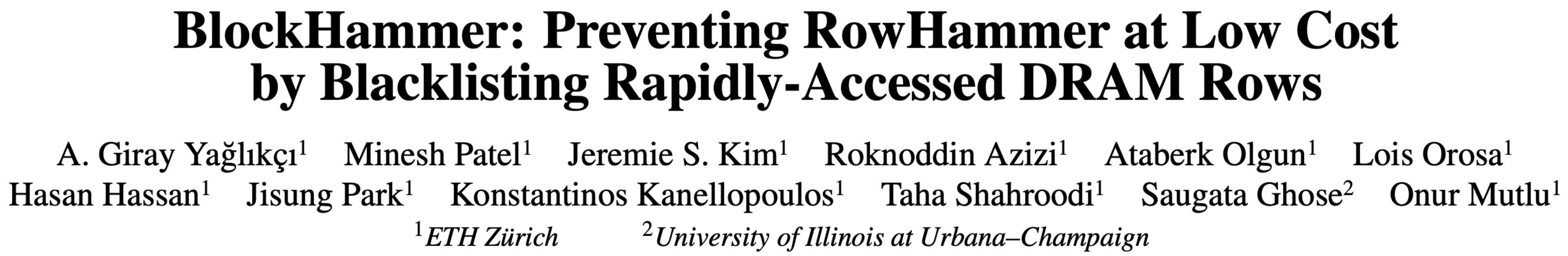
The threat of Rowhammer [61] bit flips (i.e., DRAM disturbances) is a widespread concern, especially in multi-tenant computing environments such as the cloud. Rowhammer arises from frequent activations—to a first approximation, accesses—of the same DRAM rows, which can disturb data in nearby rows due to electromagnetic interference. These bit flips manifest at the system level as data loss, machine failure, or system subversion.

Prior attacks and analyses [20, 22, 25, 30, 38, 39, 41, 48, 49, 51, 58, 61, 65, 70, 84, 88, 94, 95, 101, 108, 111–114, 119] confirm that malicious adversaries can trigger sufficient activations to flip bits, establishing Rowhammer as a *security* threat. At a high level, existing attacks require a carefully-crafted sequence of instructions to

https://www.kevinloughlin.org/moesi-prime.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*

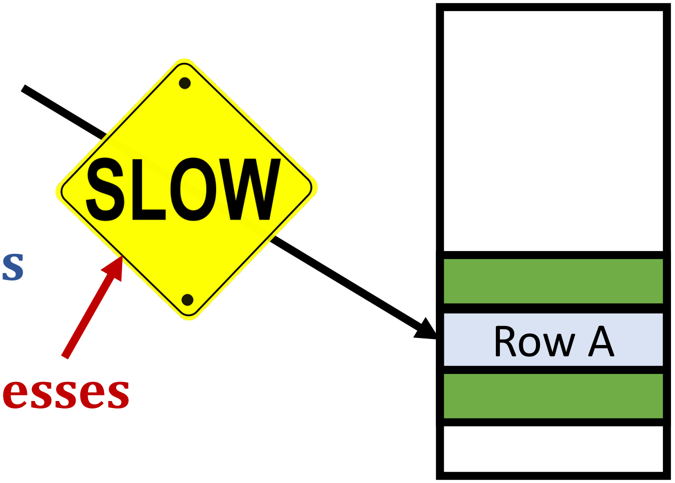
Two Key Challenges

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

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

A Takeaway

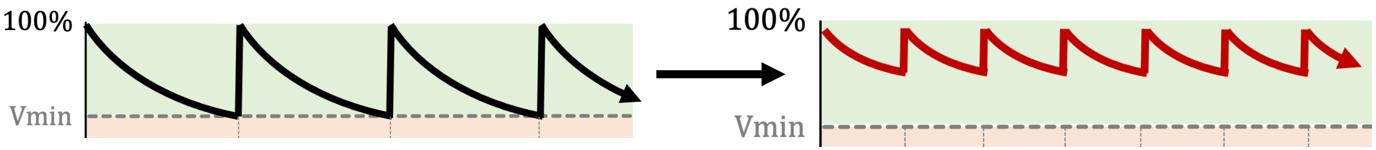
Main Memory Needs

Intelligent Controllers

for Security, Safety, Reliability, Scaling

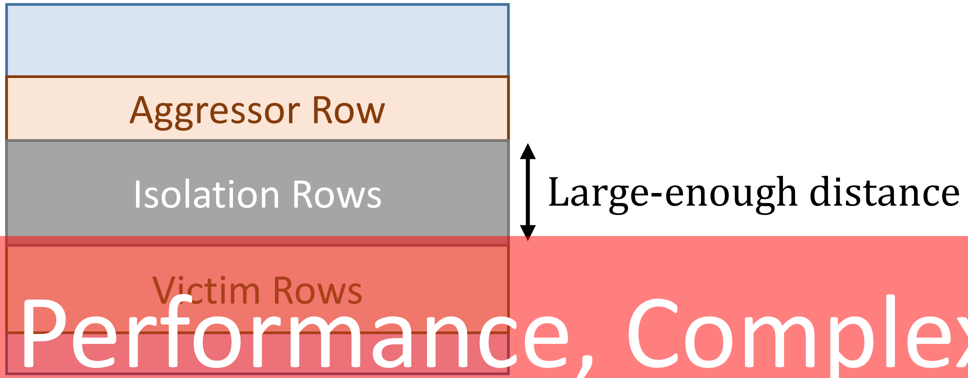
RowHammer Solution Approaches

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



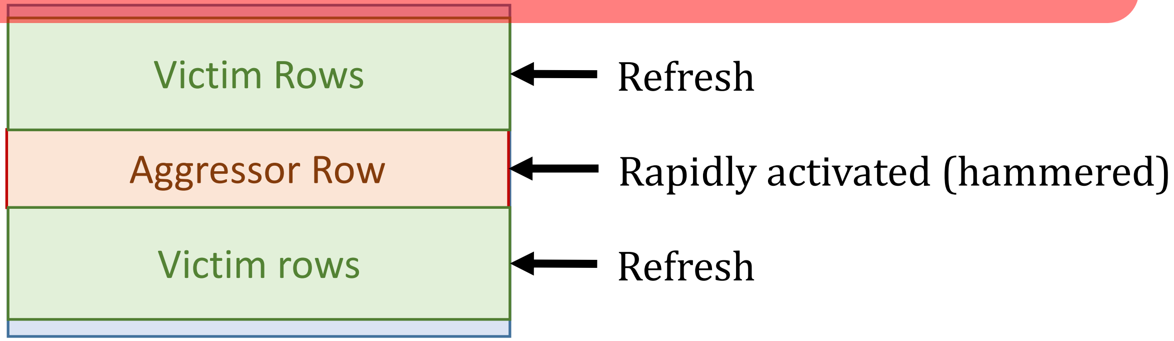
Fewer activations possible
in a refresh interval

- Physical isolation

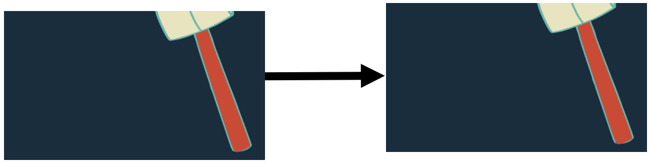


Cost, Power, Performance, Complexity

- Reactive refresh

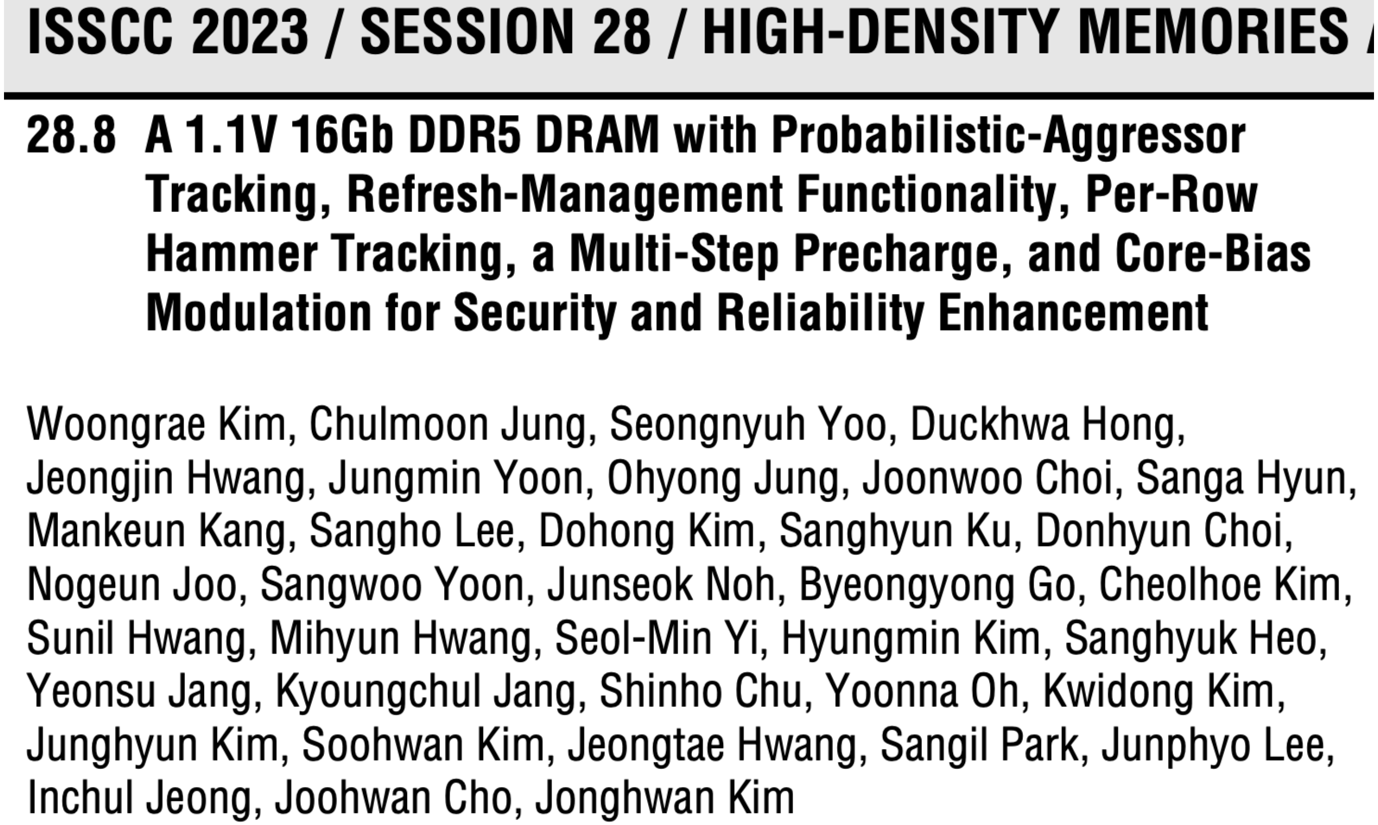


- Proactive throttling



Fewer activations allowed for aggressive applications

RowHammer in 2023: SK Hynix



ISSCC 2023 / SESSION 28 / HIGH-DENSITY MEMORIES /

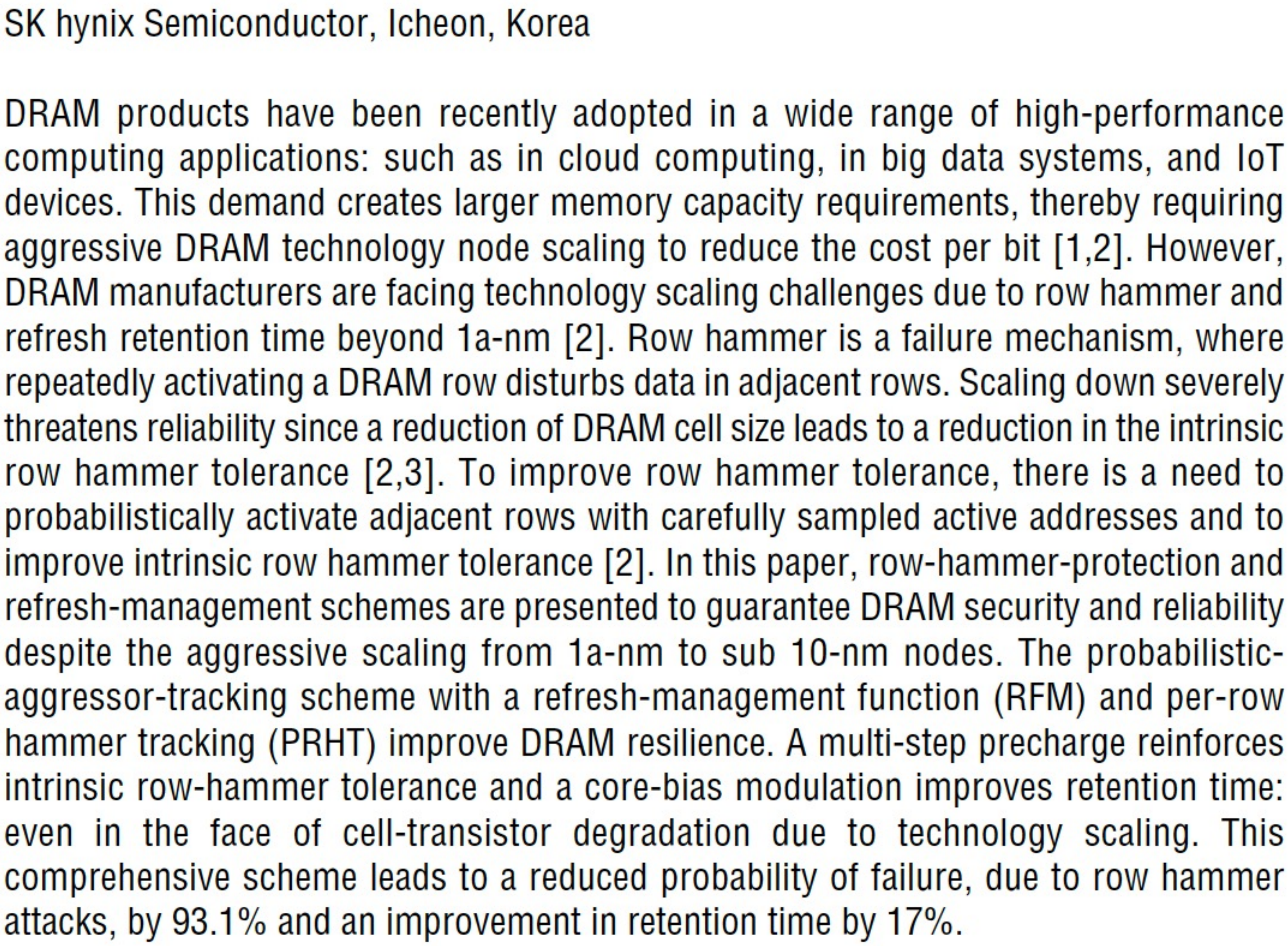
28.8 A 1.1V 16Gb DDR5 DRAM with Probabilistic-Aggressor Tracking, Refresh-Management Functionality, Per-Row Hammer Tracking, a Multi-Step Precharge, and Core-Bias Modulation for Security and Reliability Enhancement

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

ISSCC 2023
2023 International Solid-State Circuits Conference
February 19-23, 2023 San Francisco, CA

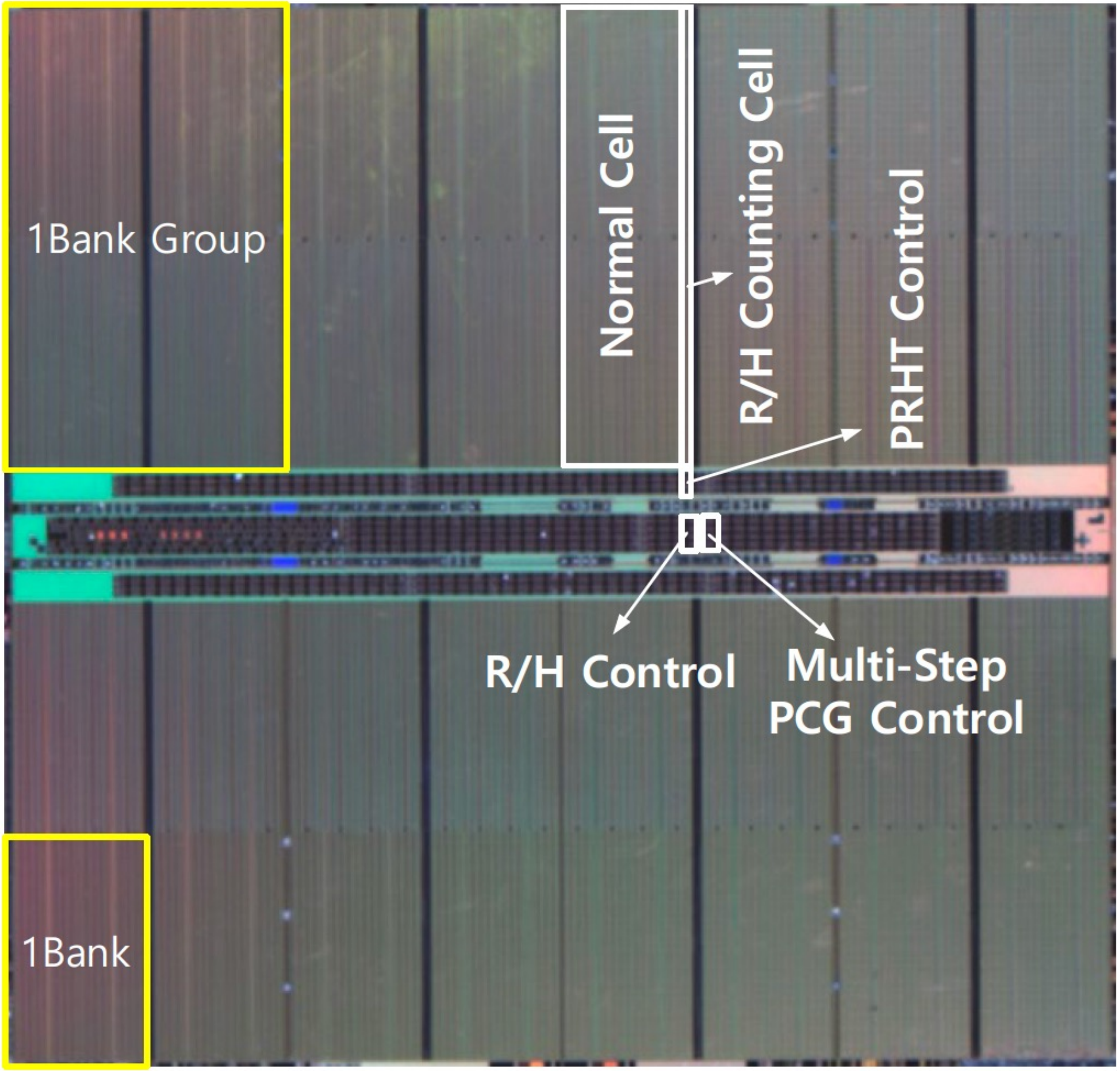
Industry's RowHammer Solutions (I)

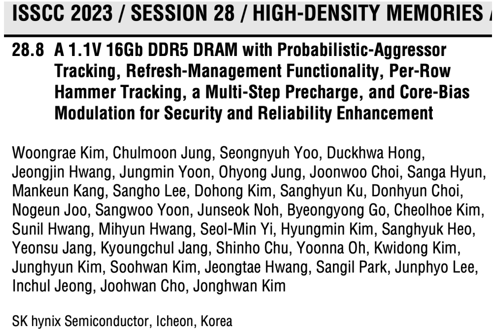


SK hynix Semiconductor, Icheon, Korea

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)





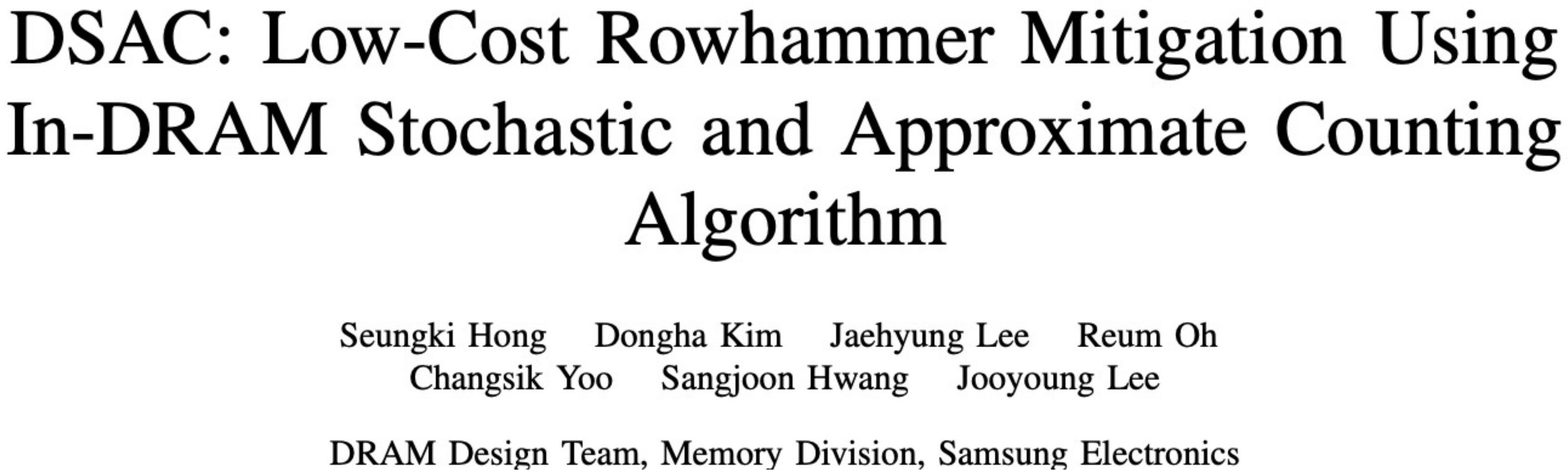
ISSCC 2023 / SESSION 28 / HIGH-DENSITY MEMORIES /

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



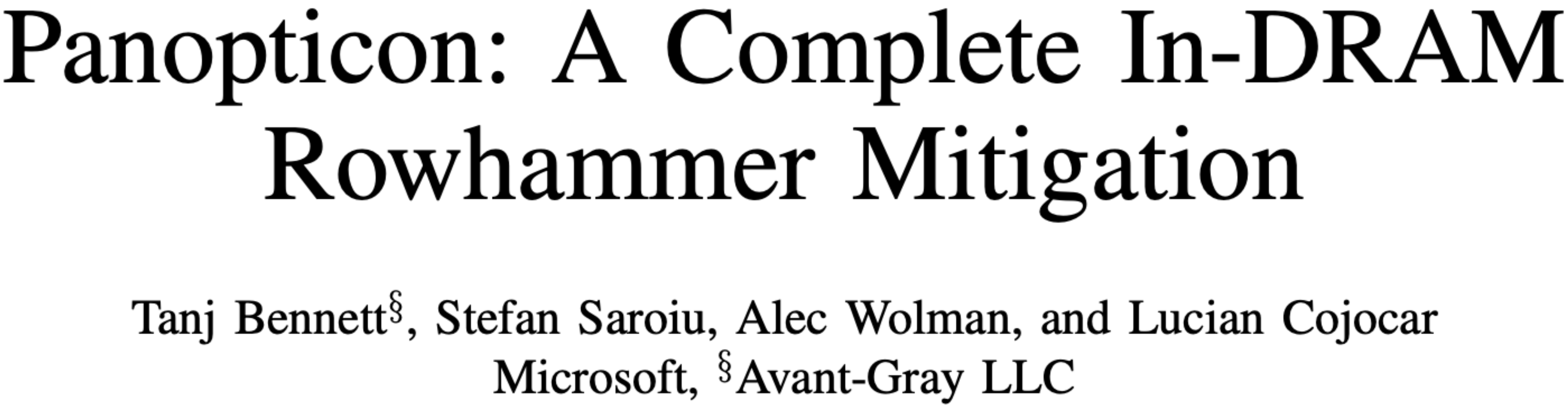
DSAC: Low-Cost Rowhammer Mitigation Using In-DRAM Stochastic and Approximate Counting Algorithm

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

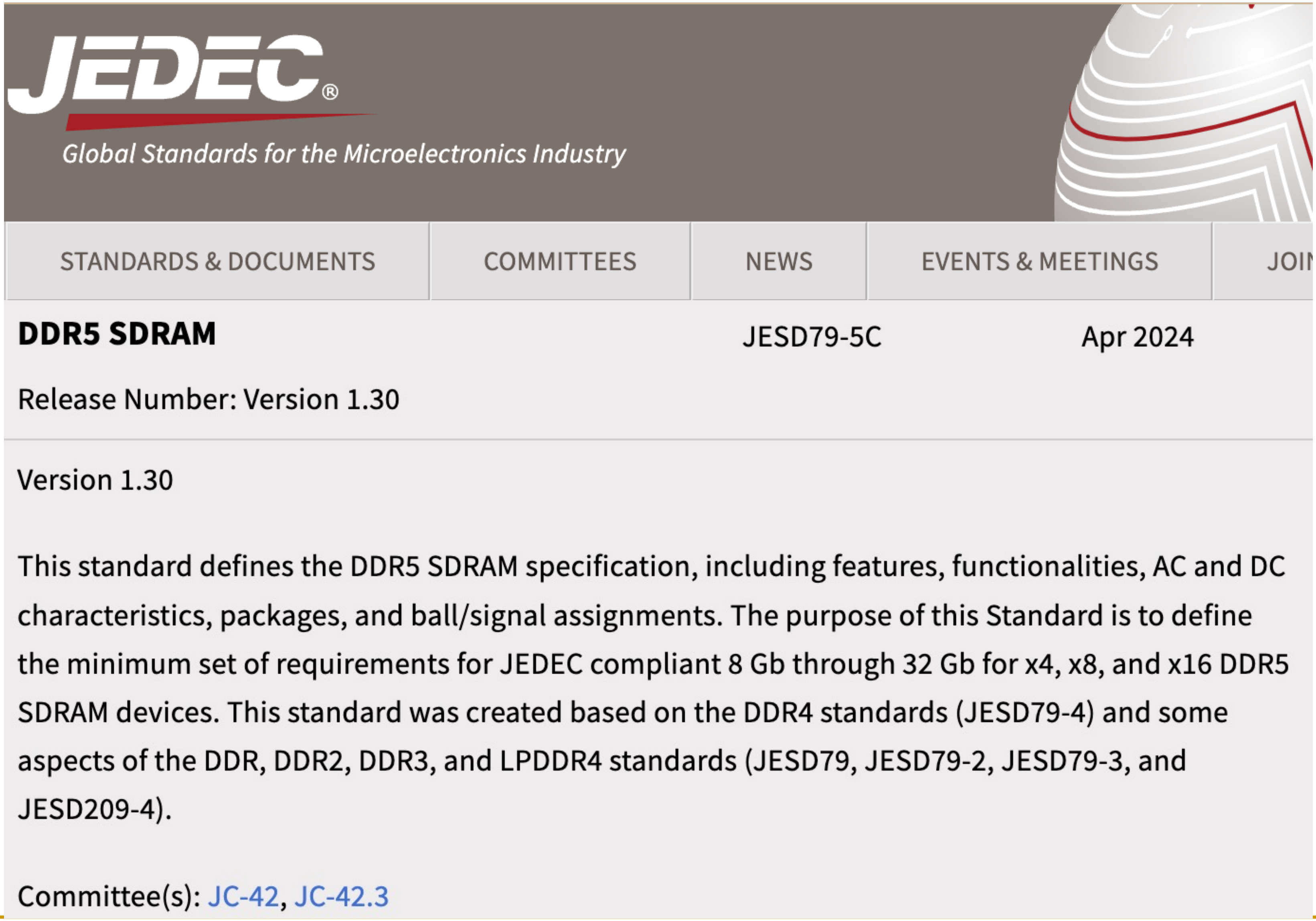


Panopticon: A Complete In-DRAM Rowhammer Mitigation

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)



DDR5 SDRAM JESD79-5C Apr 2024

Release Number: Version 1.30

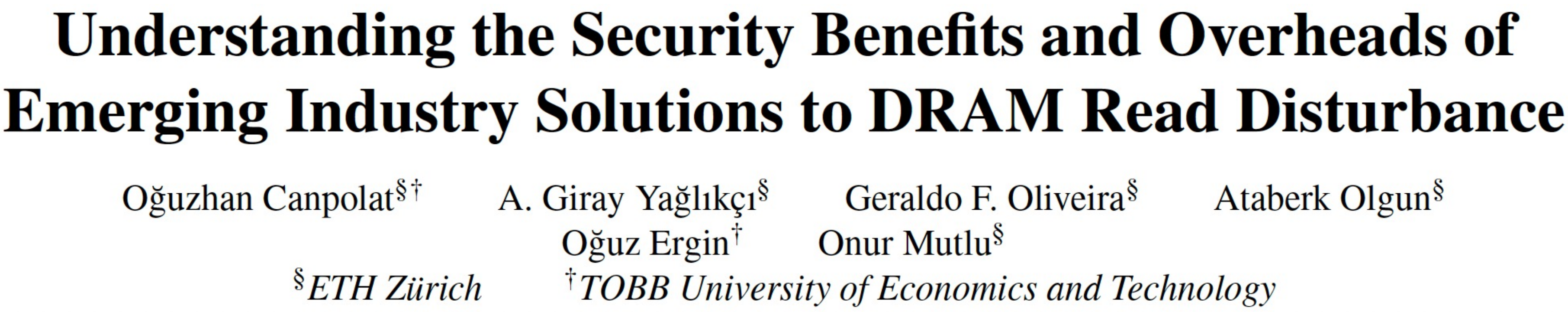
Version 1.30

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

Evaluation of Industry’s Recent Solutions

- **Appears at DRAMSec 2024**



Understanding the Security Benefits and Overheads of Emerging Industry Solutions to DRAM Read Disturbance

Oğuzhan Canpolat[§†] A. Giray Yağlıkçı[§] Geraldo F. Oliveira[§] Ataberk Olgun[§]
Oğuz Ergin[†] Onur Mutlu[§]
[§]*ETH Zürich* [†]*TOBB University of Economics and Technology*

https://arxiv.org/pdf/2406.19094

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

Question

Are we now

RowHammer-free

in 2024 and Beyond?

Are We Now BitFlip Free?

- **Appeared at ISCA in June 2023**

What if there is another phenomenon that
does NOT require high row activation count?

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

Artifacts Available V1.1 Artifacts Evaluated Functional V1.1 Results Reproduced V1.1

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**

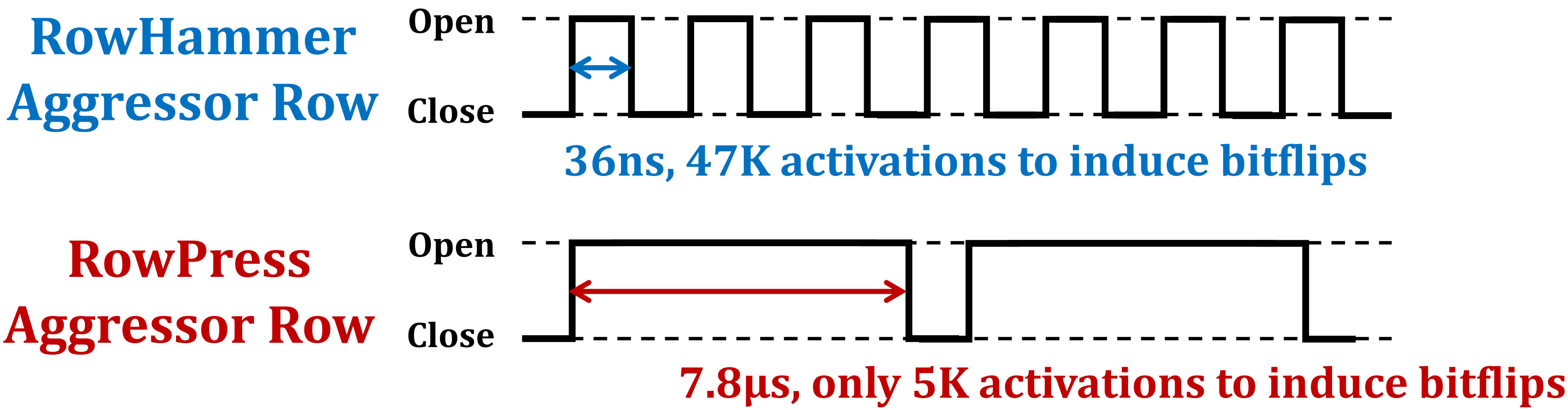
What if there is another read-disturb phenomenon that **does NOT rely on high row activation count**?

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

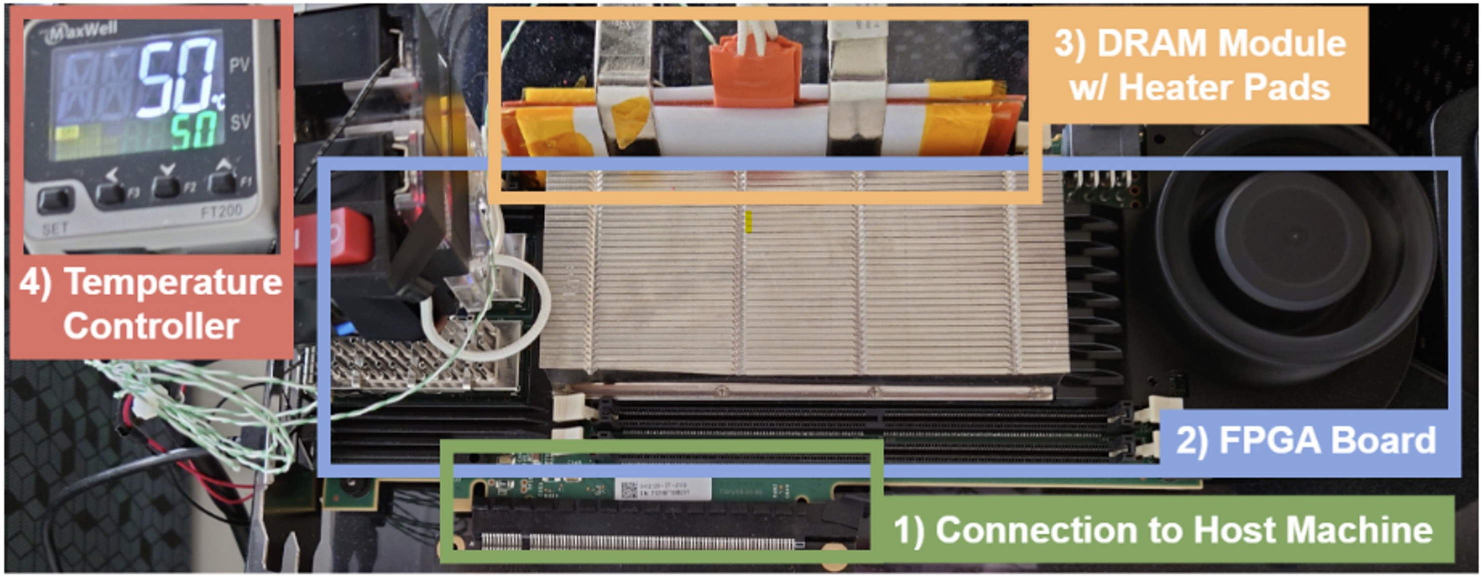

RowHammer
Aggressor Row
Open
Close
36ns, 47K activations to induce bitflips
RowPress
Aggressor Row
Open
Close
7.8µs, only 5K activations to induce bitflips

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

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


3) DRAM Module w/ Heater Pads
4) Temperature Controller
2) FPGA Board
1) Connection to Host Machine
MaxWell
50
PV
SV
FT200
SET

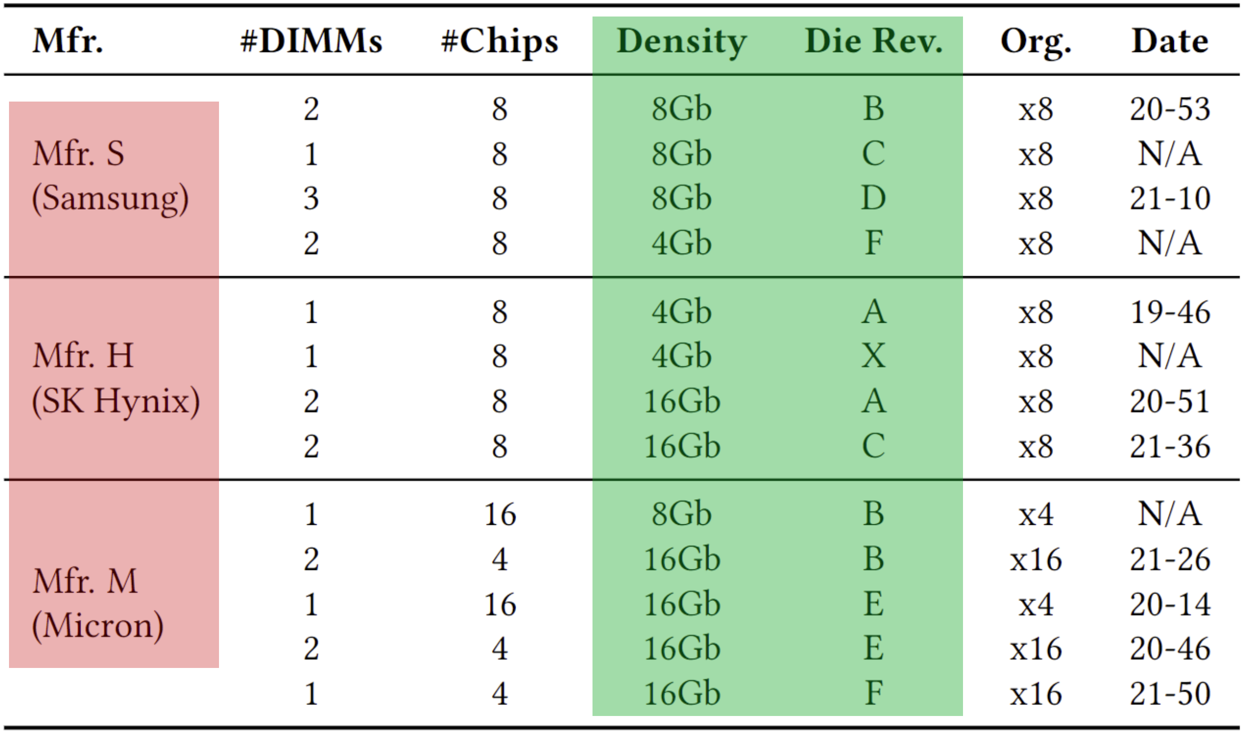
SAFARI

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

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

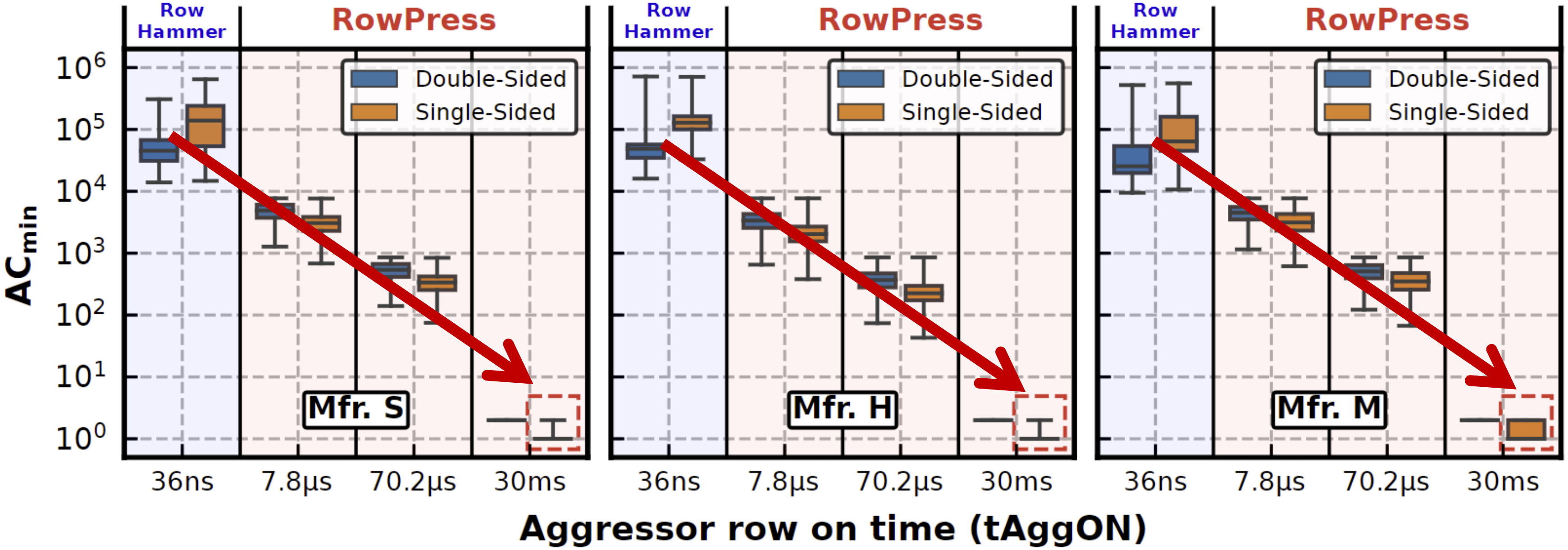
RowPress has a **different** underlying error **mechanism** from RowHammer

Key Characteristics of RowPress (I)

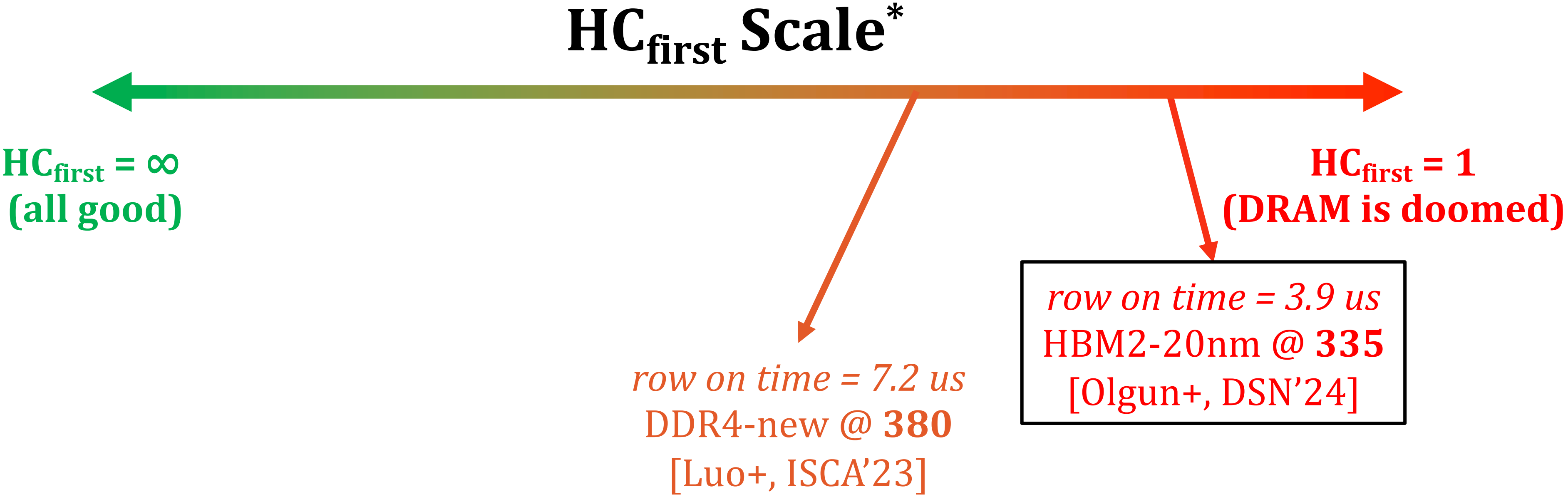
Amplifying Read Disturbance in DRAM

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

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



RowPress at t_{AggON} = Refresh Interval



HC_{first} Scale*

HC_{first} = ∞
(all good)

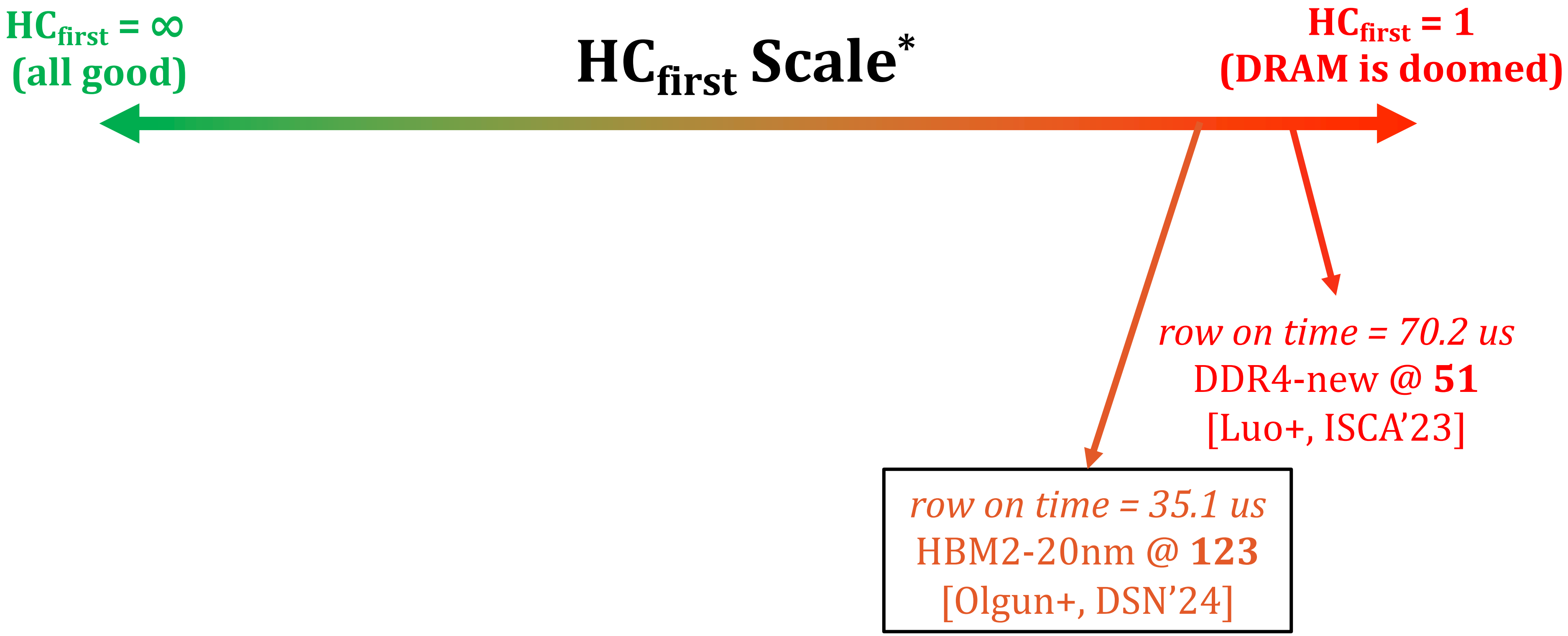
HC_{first} = 1
(DRAM is doomed)

row on time = 7.2 us
DDR4-new @ **380**
[Luo+, ISCA'23]

row on time = 3.9 us
HBM2-20nm @ **335**
[Olgun+, DSN'24]

*Not shown: Significant variance in HC_{first} across vendors and die variations

RowPress at t_{AggON} = 9 * Refresh Interval



HC_{first} = ∞
(all good)

HC_{first} Scale*

HC_{first} = 1
(DRAM is doomed)

row on time = 70.2 us
DDR4-new @ **51**
[Luo+, ISCA'23]

row on time = 35.1 us
HBM2-20nm @ **123**
[Olgun+, DSN'24]

*Not shown: Significant variance in HC_{first} across vendors and die variations

Key Characteristics of RowPress (II)

Amplifying Read Disturbance in DRAM

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

Different From RowHammer

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

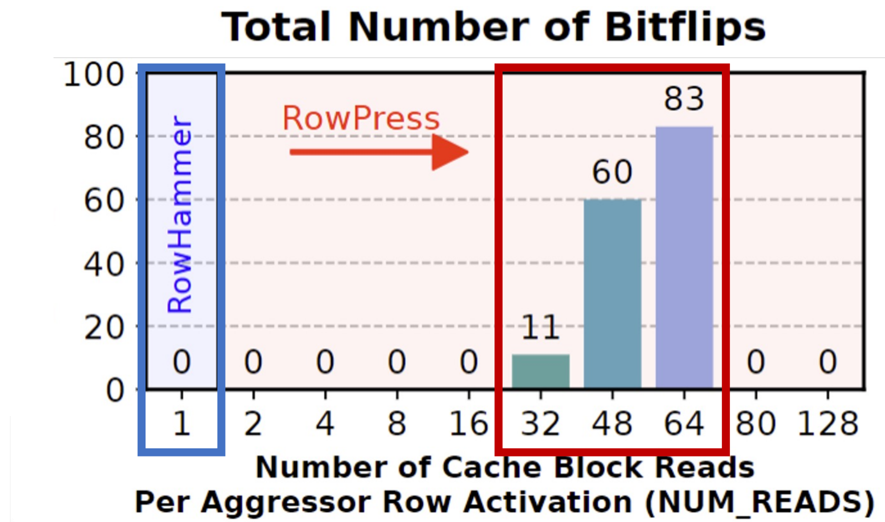
```
// Sync with Refresh and Loop Below
 for (k = 0; k < NUM_AGGR_ACTS; k++)
  for (j = 0; j < NUM_READS; j++) *AGGRESSOR1[j];
  for (j = 0; j < NUM_READS; j++) *AGGRESSOR2[j];
  for (j = 0; j < NUM_READS; j++)
   clflushopt(AGGRESSOR1[j]);
   clflushopt(AGGRESSOR2[j]);
  mfence();
 activate_dummy_rows();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

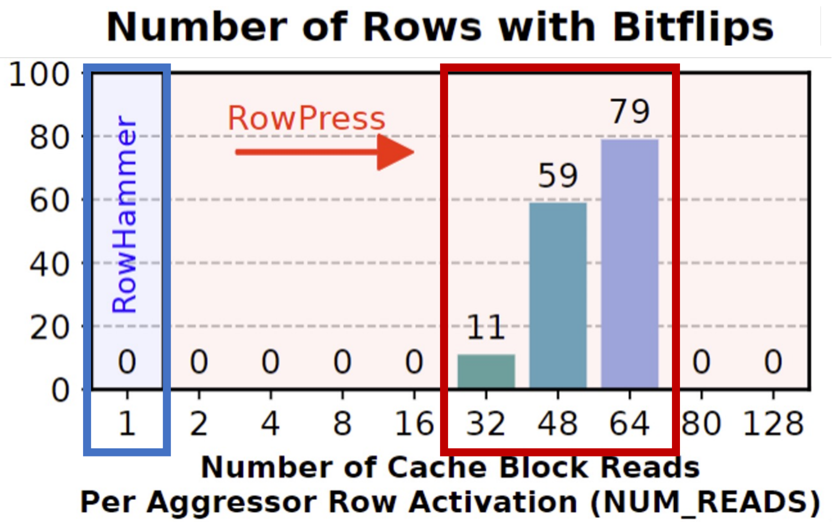
Number of Cache Blocks Accessed Per Aggressor Row ACT
(NUM_READS=1 is Rowhammer)

SAFARI

Real-System Demonstration (II)

On 1500 victim rows





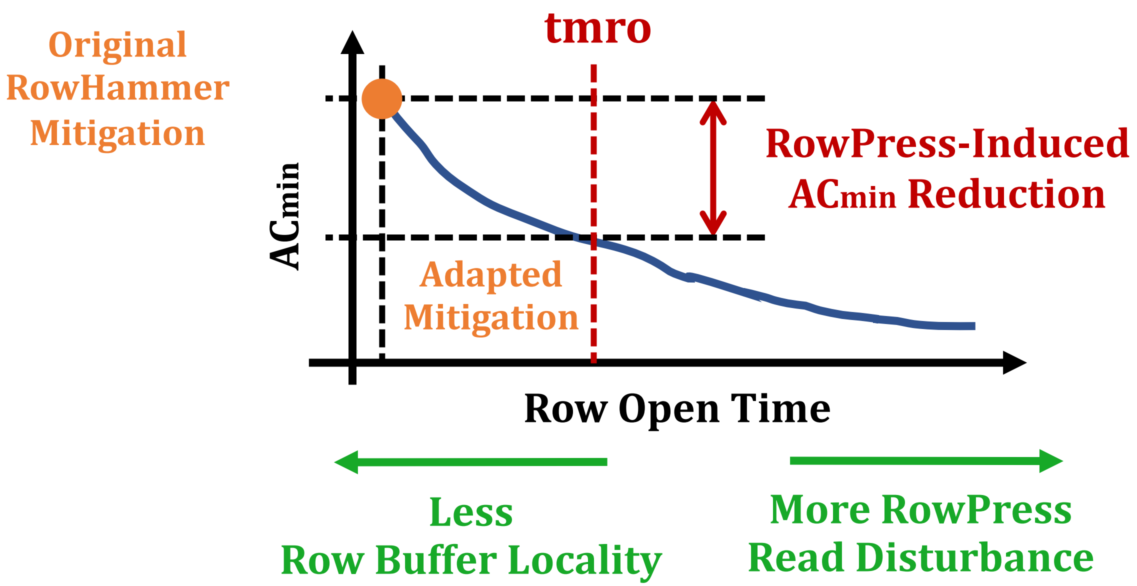
Leveraging RowPress, our user-level program induces bitflips when RowHammer cannot

Mitigating RowPress (I)

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

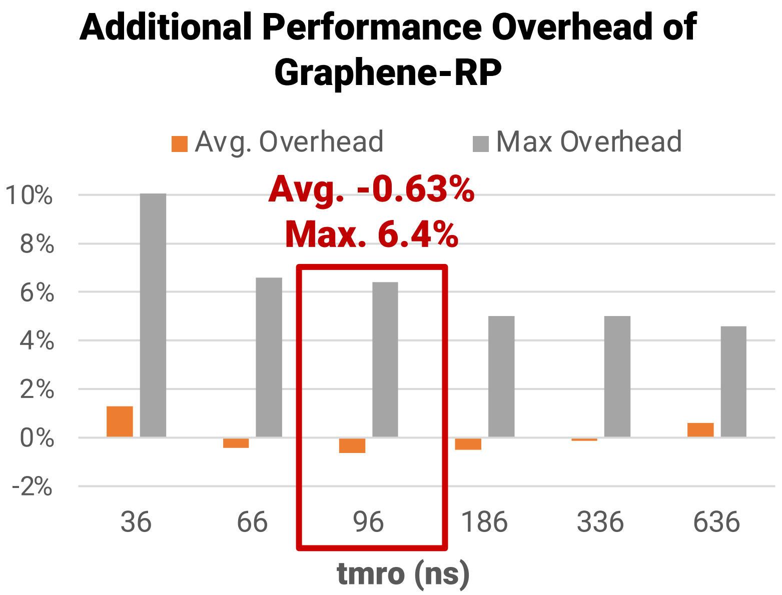
Key Idea:

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

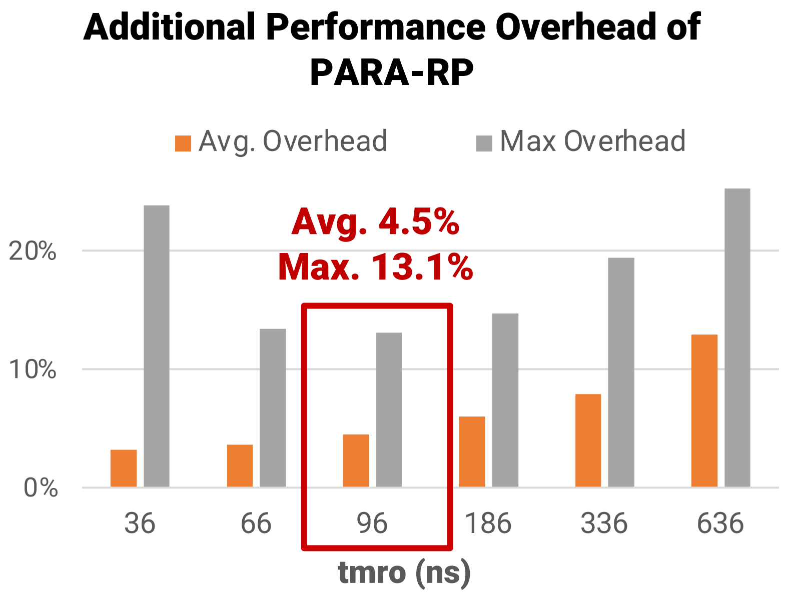


Mitigating RowPress (II)

Key evaluation results



Avg. -0.63%
Max. 6.4%



Avg. 4.5%
Max. 13.1%

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

Artifacts Available acm V1.1

Artifacts Evaluated acm Functional V1.1

Results Reproduced acm V1.1

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

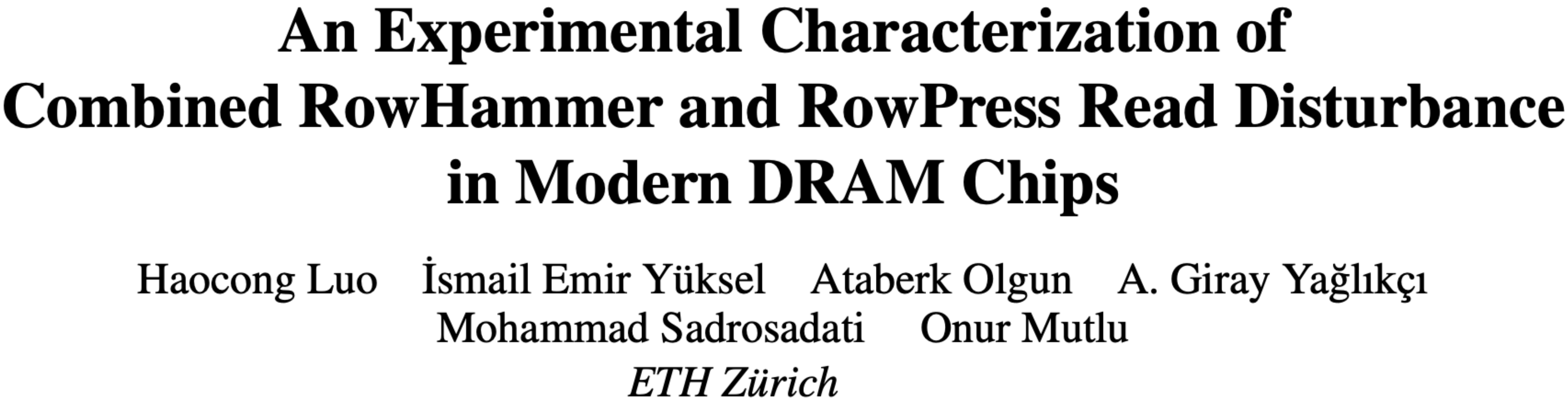
Ataberk Olgun
ataberk.olgun@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

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

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

Combining RowHammer and RowPress

- **Appears at DSN Disrupt 2024**

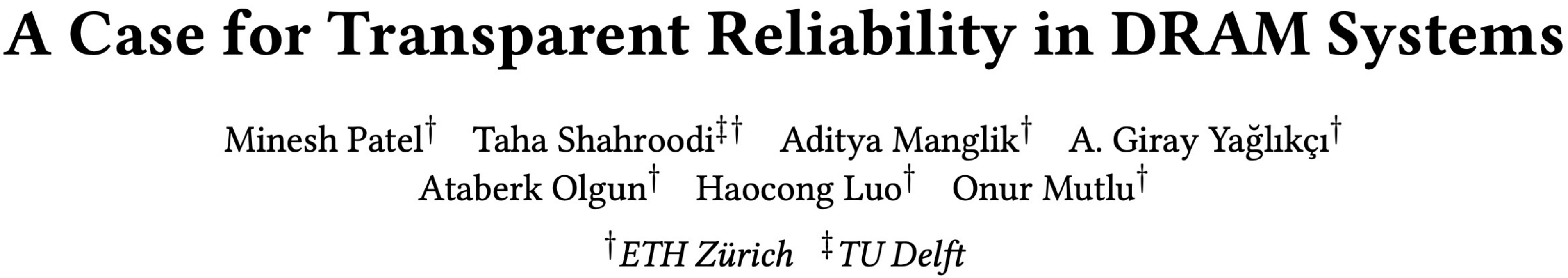


An Experimental Characterization of Combined RowHammer and RowPress Read Disturbance in Modern DRAM Chips

Haocong Luo İsmail Emir Yüksel Ataberk Olgun A. Giray Yağlıkçı
Mohammad Sadrosadati Onur Mutlu
ETH Zürich

https://arxiv.org/pdf/2406.13080v2

Better Communication Between DRAM & Controller



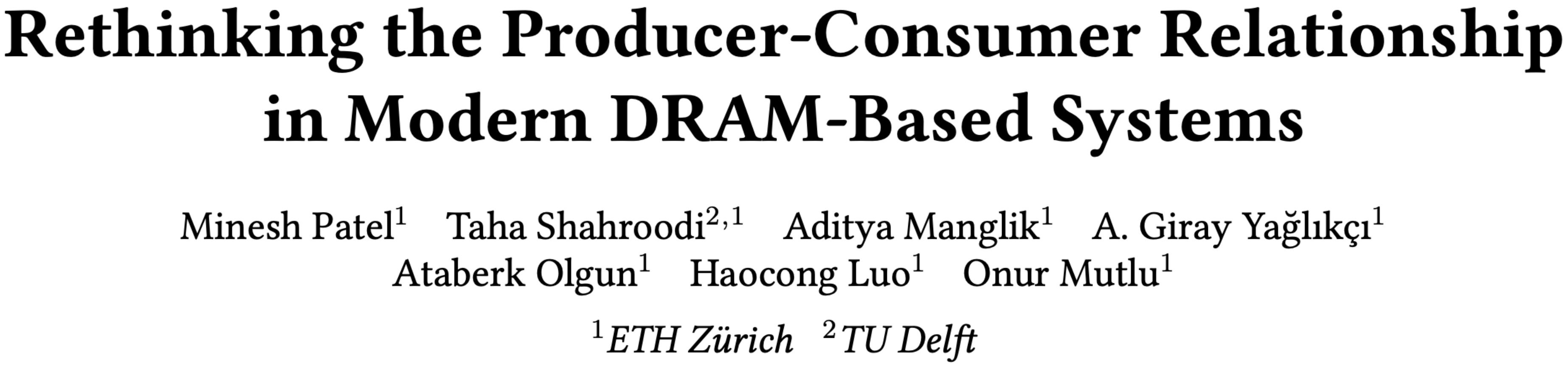
A Case for Transparent Reliability in DRAM Systems

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 Communication Between DRAM & Controller



Rethinking the Producer-Consumer Relationship in Modern DRAM-Based Systems

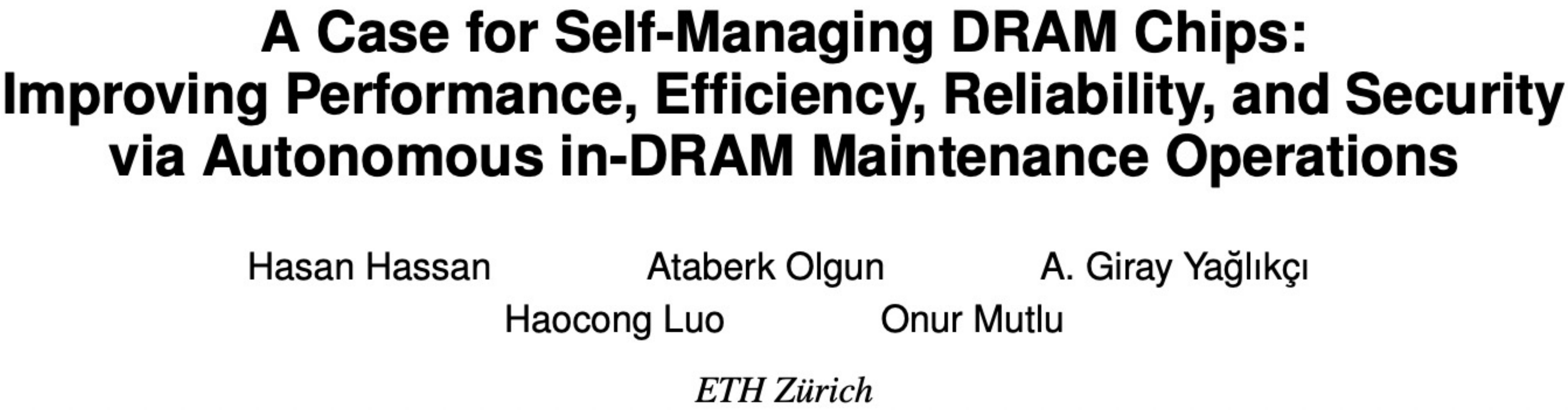
Minesh Patel[1] Taha Shahroodi[2,1] Aditya Manglik[1] A. Giray Yağlıkçı[1]
Ataberk Olgun[1] Haocong Luo[1] Onur Mutlu[1]

[1] *ETH Zürich* [2] *TU Delft*

https://arxiv.org/pdf/2401.16279

Better Partitioning of DRAM & Controller

- **To Appear at MICRO 2024**



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

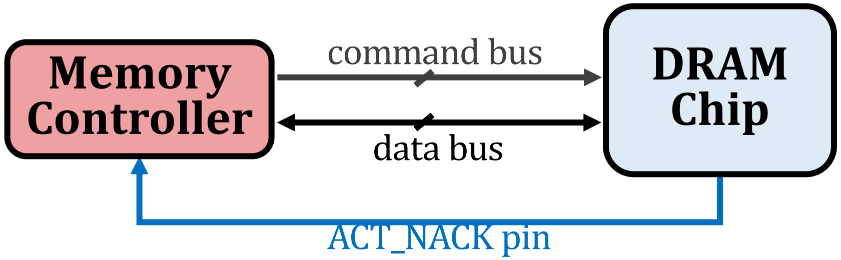
Self-Managing DRAM: Overview

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

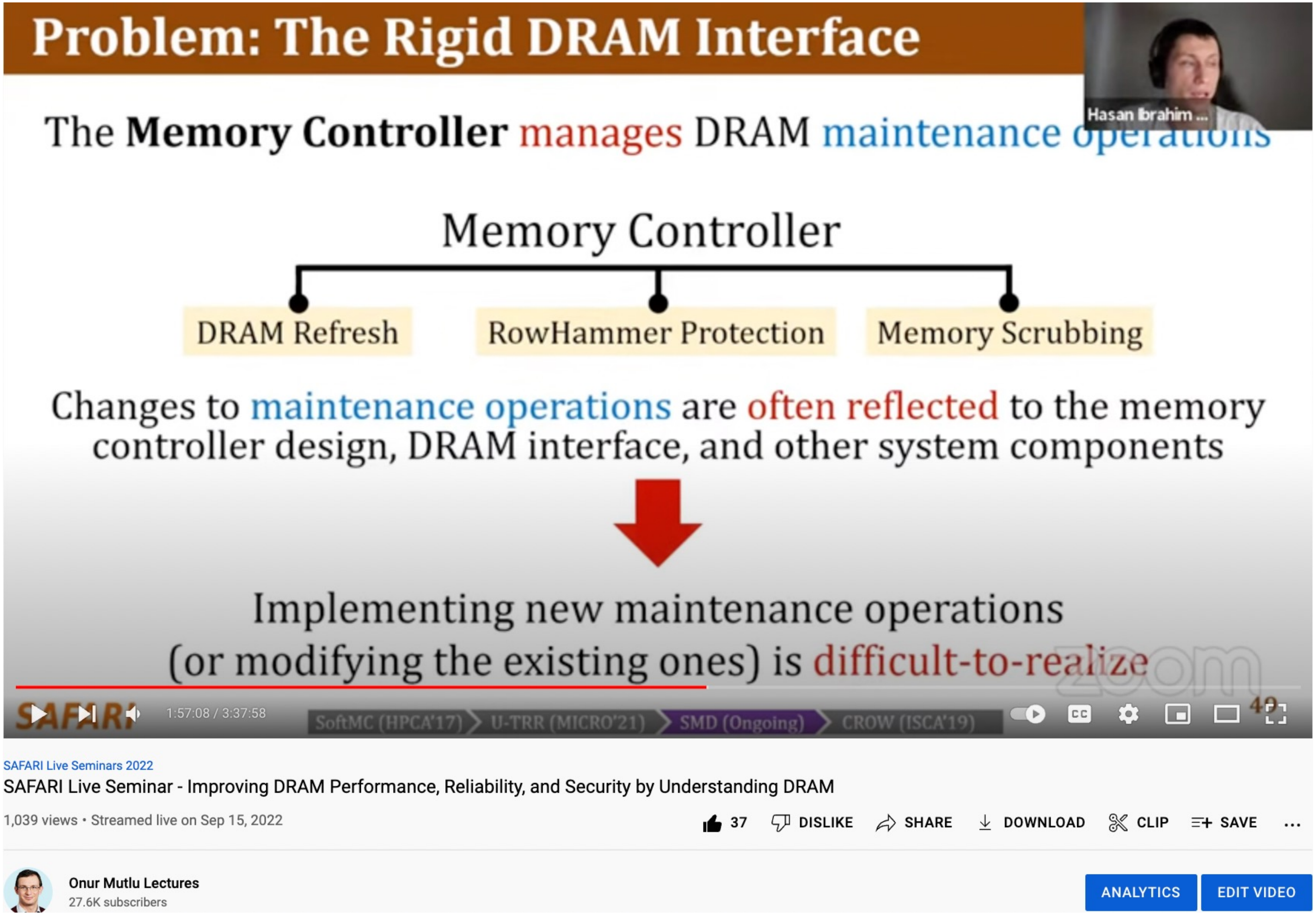
Self-Managing DRAM (SMD)

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

SAFARI

Talk on Self-Managing DRAM



Problem: The Rigid DRAM Interface

The **Memory Controller** manages DRAM maintenance operations

Memory Controller

DRAM Refresh | RowHammer Protection | Memory Scrubbing

Changes to maintenance operations are often reflected to the memory controller design, DRAM interface, and other system components

Implementing new maintenance operations (or modifying the existing ones) is difficult-to-realize

SAFARI Live Seminars 2022

SAFARI Live Seminar - Improving DRAM Performance, Reliability, and Security by Understanding DRAM

1,039 views • Streamed live on Sep 15, 2022

Onur Mutlu Lectures

27.6K subscribers

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

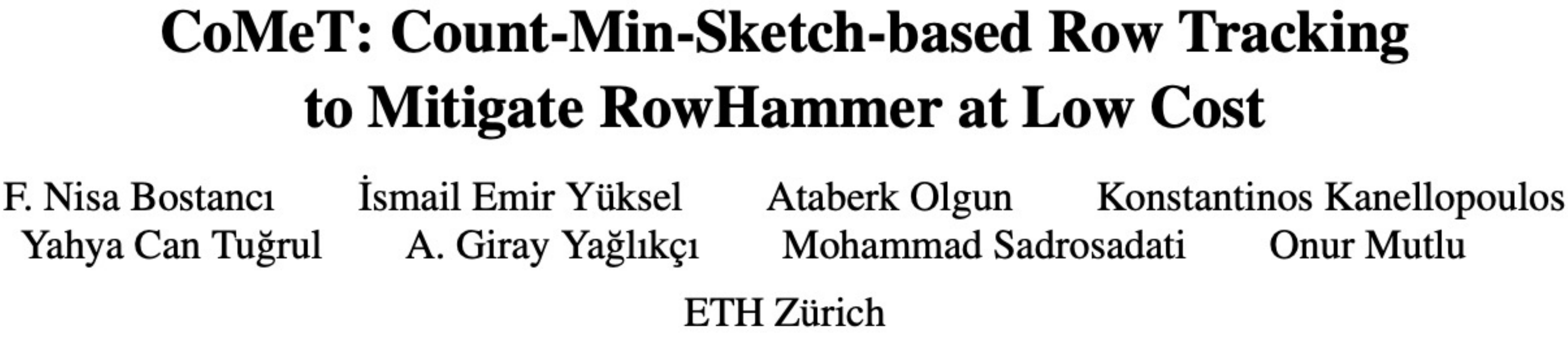
Ataberk Olgun Yahya Can Tugrul Nisa Bostanci Ismail Emir Yuksel
Haocong Luo Steve Rhyner Abdullah Giray Yaglikci Geraldo F. Oliveira Onur Mutlu

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

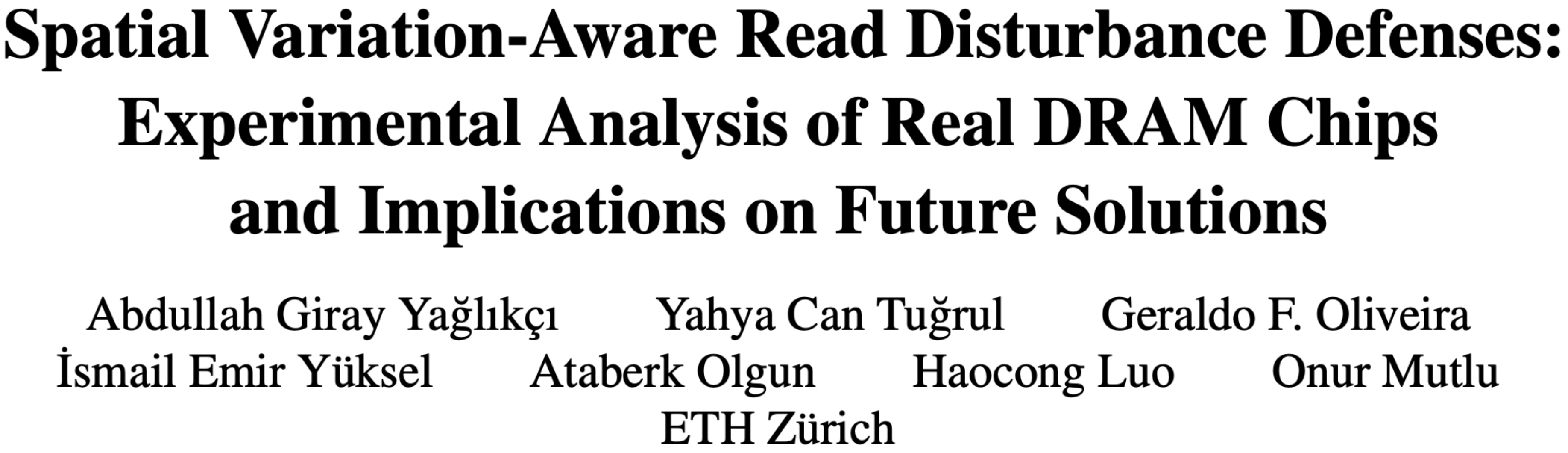
F. Nisa Bostancı İsmail Emir Yüksel Ataberk Olgun Konstantinos Kanellopoulos
Yahya Can Tuğrul A. Giray Yağlıkçı Mohammad Sadrosadati Onur Mutlu
ETH Zürich

https://arxiv.org/pdf/2402.18769

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

SVaRD: Another Intelligent Memory Controller

- **Appears at HPCA 2024**



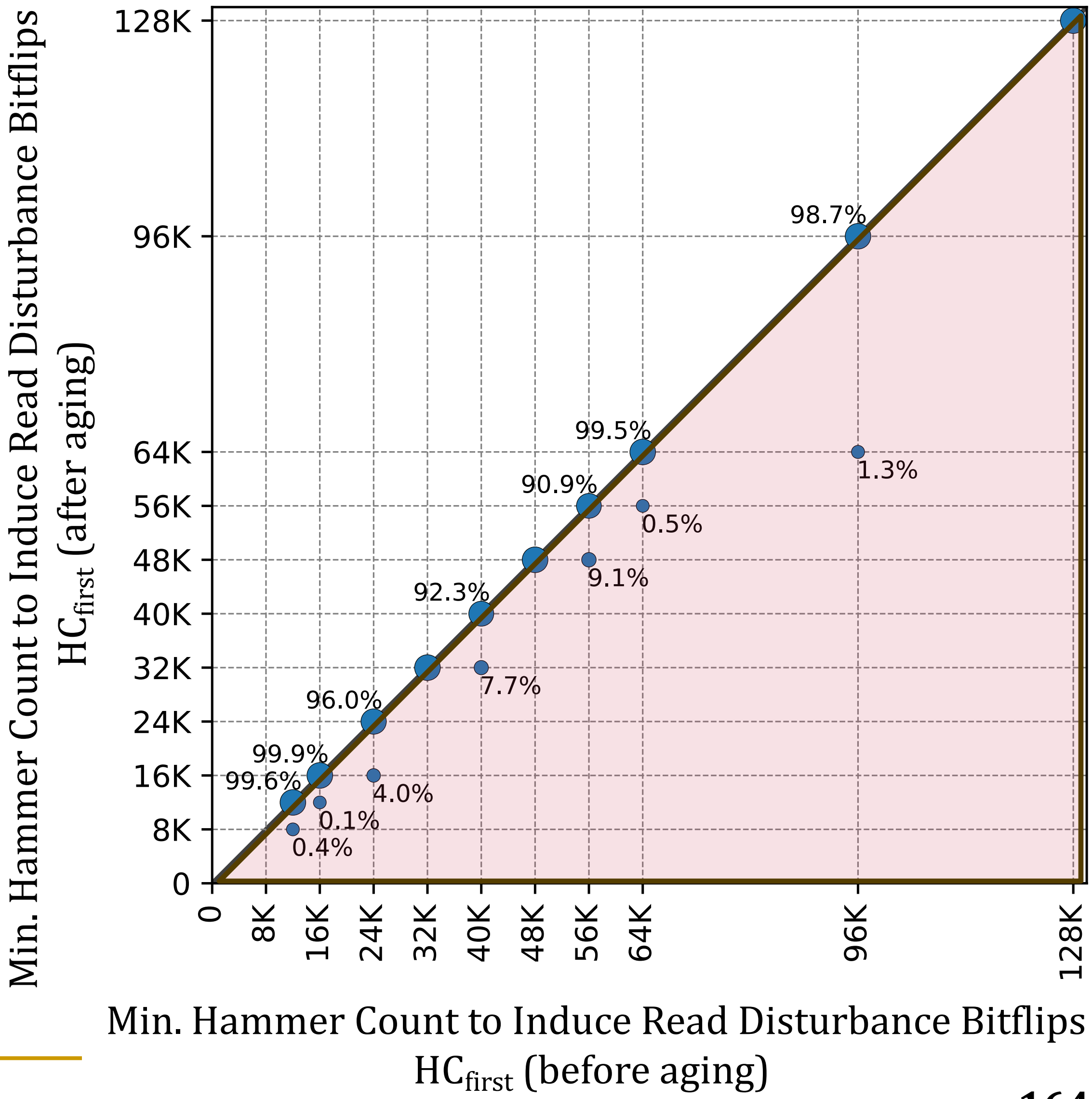
Spatial Variation-Aware Read Disturbance Defenses: Experimental Analysis of Real DRAM Chips and Implications on Future Solutions

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

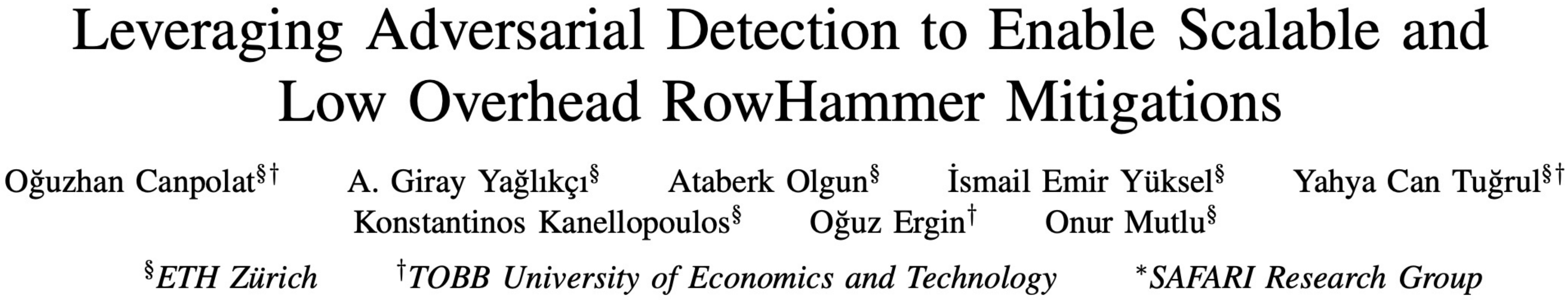
Effect of Aging on RowHammer BitFlips

Aging can lead to RowHammer bitflips at **smaller** hammer counts



RowHammer Defenses Can Cause Denial of Service

- **To Appear at MICRO 2024**



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 (miss buffers)
  - 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.

Industry's Recent Solutions Are Vulnerable

- **Appears at DRAMSec 2024**

Understanding the Security Benefits and Overheads of Emerging Industry Solutions to DRAM Read Disturbance

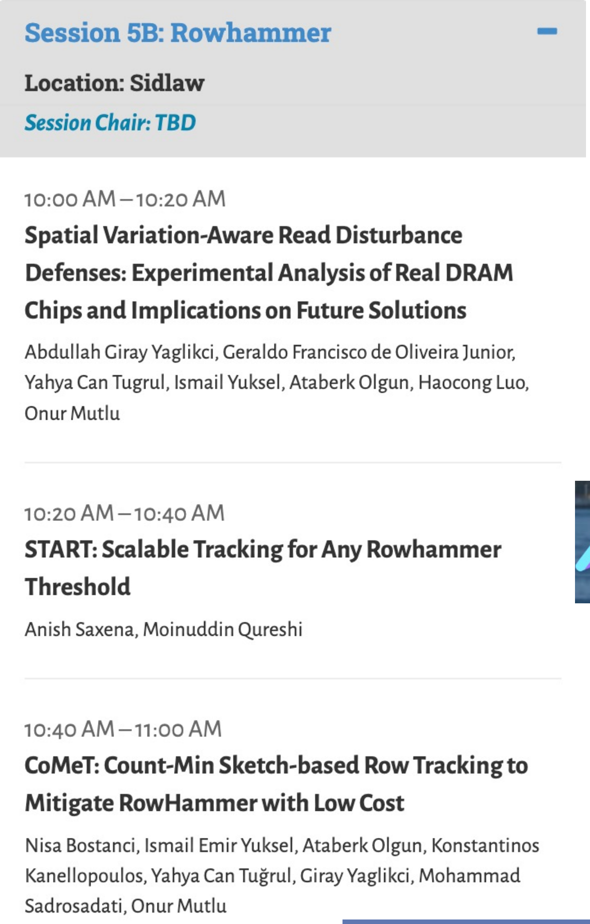
Oğuzhan Canpolat[§†] A. Giray Yağlıkçı[§] Geraldo F. Oliveira[§] Ataberk Olgun[§]
Oğuz Ergin[†] Onur Mutlu[§]
[§]*ETH Zürich* [†]*TOBB University of Economics and Technology*

https://arxiv.org/pdf/2406.19094

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

Some RowHammer Works in 2024 (I)

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

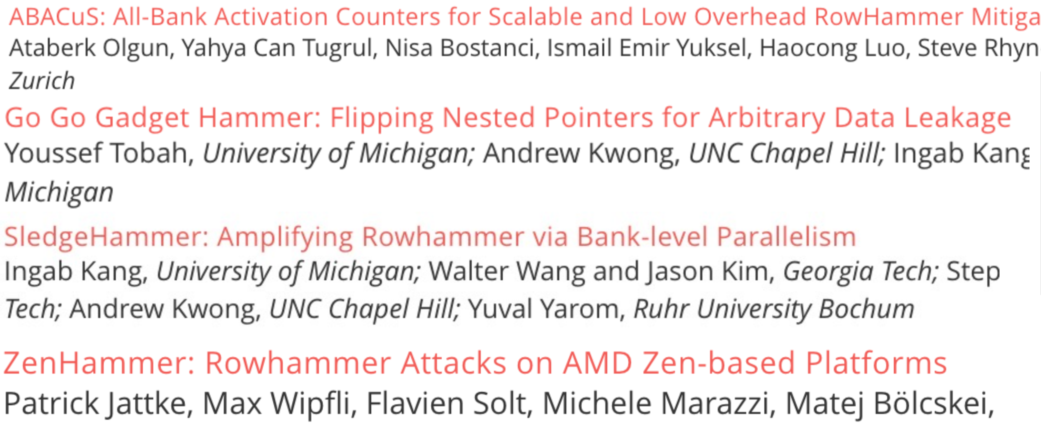
Anish Saxena, Moinuddin Qureshi

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

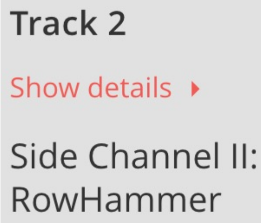


ABACuS: All-Bank Activation Counters for Scalable and Low Overhead RowHammer Mitigation
Ataberk Olgun, Yahya Can Tugrul, Nisa Bostanci, Ismail Emir Yuksel, Haocong Luo, Steve Rhyner, A
Zurich

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*

ZenHammer: Rowhammer Attacks on AMD Zen-based Platforms
Patrick Jattke, Max Wipfli, Flavien Solt, Michele Marazzi, Matej Bölcskei,

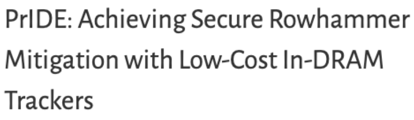


Track 2

Show details ▸

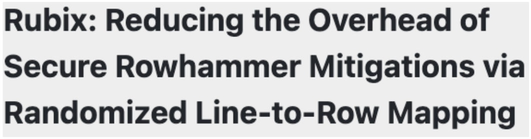
Side Channel II: RowHammer

ISCA 2024



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

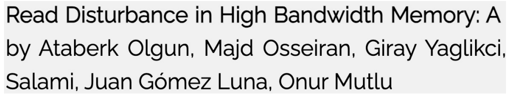
ASPLOS 2024



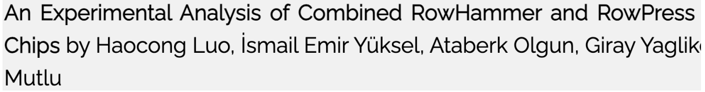
Rubix: Reducing the Overhead of Secure Rowhammer Mitigations via Randomized Line-to-Row Mapping

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

DSN 2024



Read Disturbance in High Bandwidth Memory: A
by Ataberk Olgun, Majd Osseiran, Giray Yaglikci,
Salami, Juan Gómez Luna, Onur Mutlu

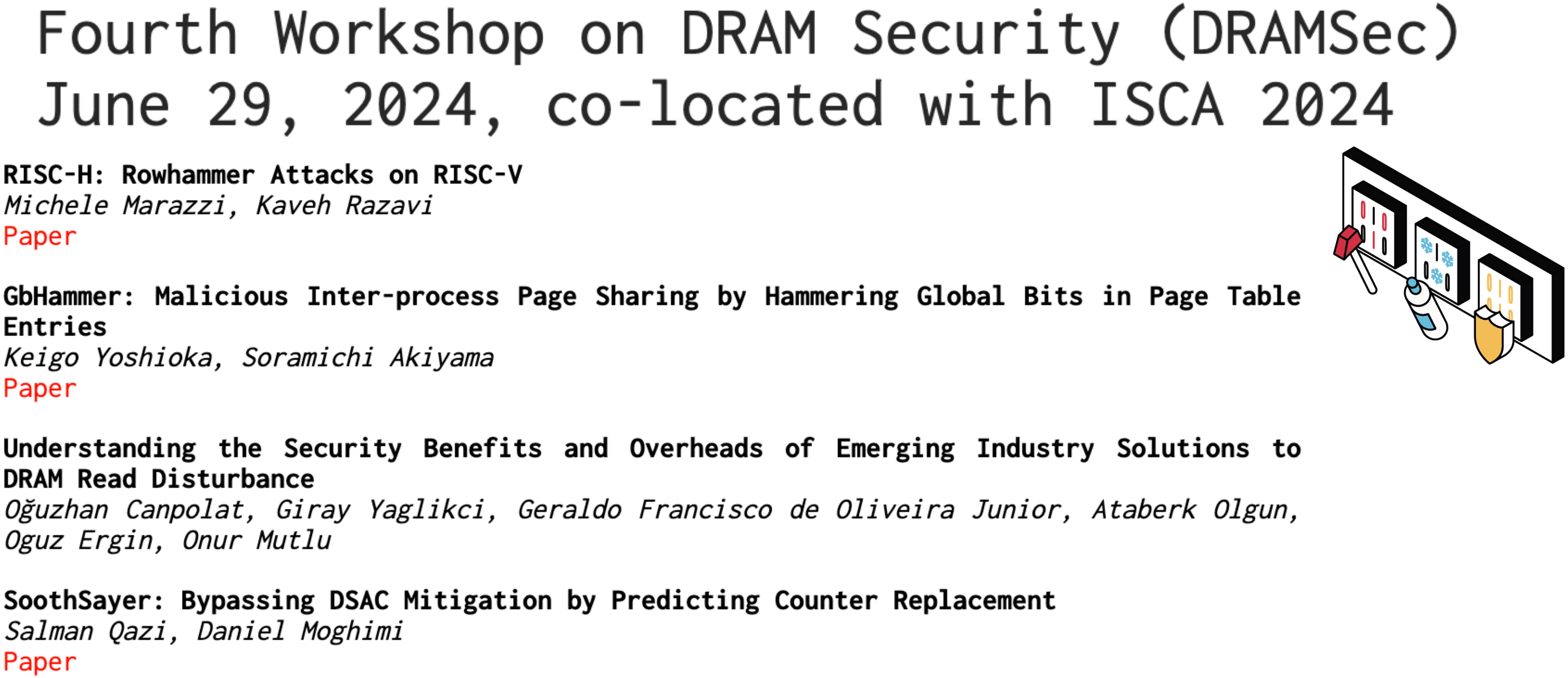


An Experimental Analysis of Combined RowHammer and RowPress Chips by Haocong Luo, İsmail Emir Yüksel, Ataberk Olgun, Giray Yaglik
Mutlu

IRPS
International Reliability Physics Symposium

Understanding the Physical Mechanism of RowPress at the Device-Level in Sub-20 nm DRAM

Some RowHammer Works in 2024 (II)



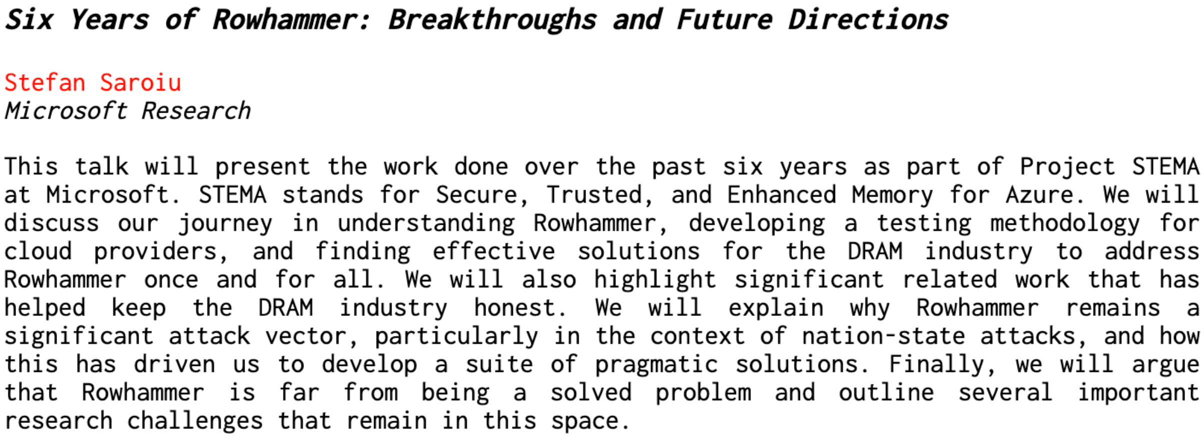
Fourth Workshop on DRAM Security (DRAMSec) June 29, 2024, co-located with ISCA 2024

RISC-H: Rowhammer Attacks on RISC-V
Michele Marazzi, Kaveh Razavi
Paper

GbHammer: Malicious Inter-process Page Sharing by Hammering Global Bits in Page Table Entries
Keigo Yoshioka, Soramichi Akiyama
Paper

Understanding the Security Benefits and Overheads of Emerging Industry Solutions to DRAM Read Disturbance
Oğuzhan Canpolat, Giray Yaglikci, Geraldo Francisco de Oliveira Junior, Ataberk Olgun, Oguz Ergin, Onur Mutlu

SoothSayer: Bypassing DSAC Mitigation by Predicting Counter Replacement
Salman Qazi, Daniel Moghimi
Paper

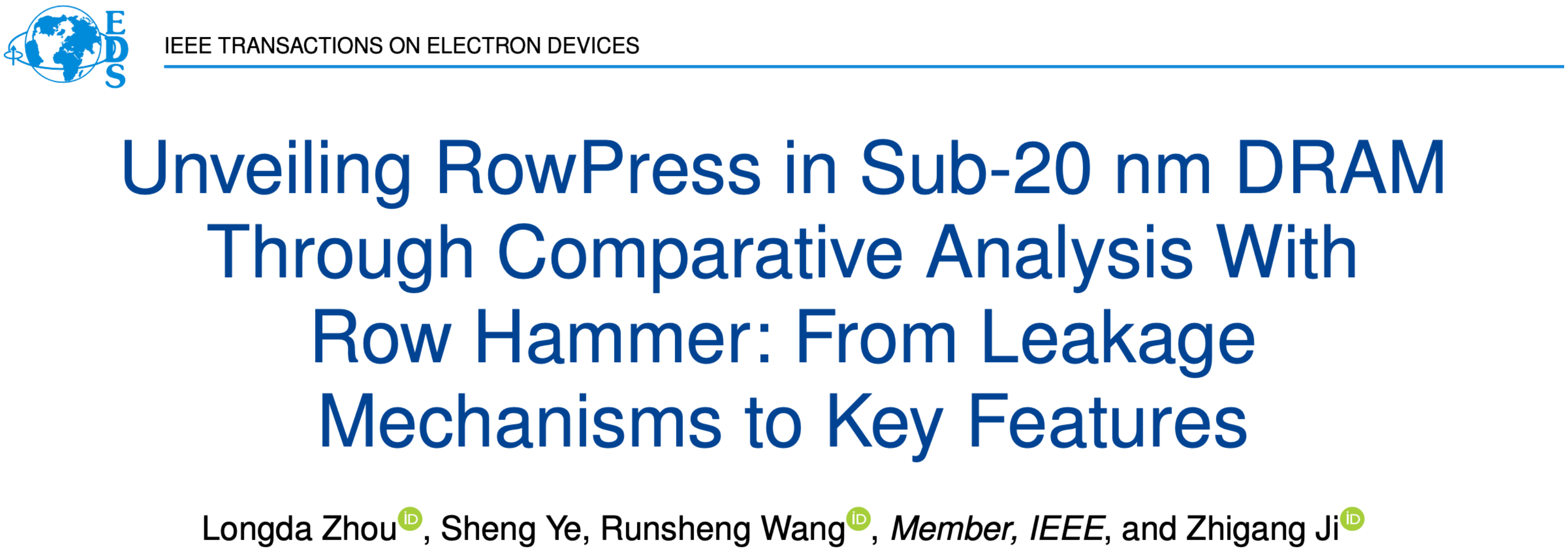


Six Years of Rowhammer: Breakthroughs and Future Directions

Stefan Saroiu
Microsoft Research

This talk will present the work done over the past six years as part of Project STEMA at Microsoft. STEMA stands for Secure, Trusted, and Enhanced Memory for Azure. We will discuss our journey in understanding Rowhammer, developing a testing methodology for cloud providers, and finding effective solutions for the DRAM industry to address Rowhammer once and for all. We will also highlight significant related work that has helped keep the DRAM industry honest. We will explain why Rowhammer remains a significant attack vector, particularly in the context of nation-state attacks, and how this has driven us to develop a suite of pragmatic solutions. Finally, we will argue that Rowhammer is far from being a solved problem and outline several important research challenges that remain in this space.

Some RowHammer Works in 2024 (III)



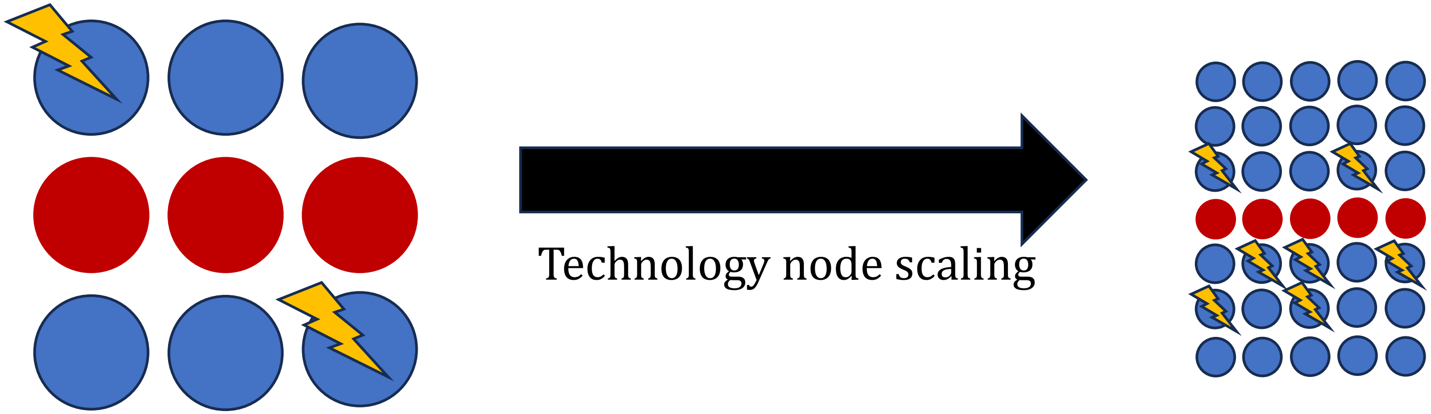
IEEE TRANSACTIONS ON ELECTRON DEVICES

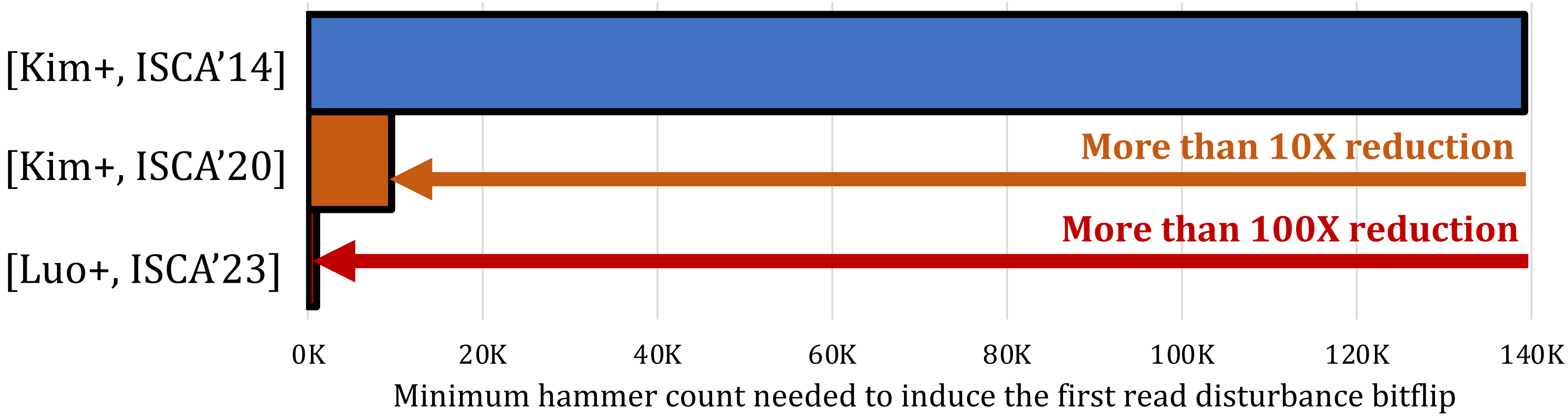
Unveiling RowPress in Sub-20 nm DRAM Through Comparative Analysis With Row Hammer: From Leakage Mechanisms to Key Features

Longda Zhou, Sheng Ye, Runsheng Wang, *Member, IEEE*, and Zhigang Ji

Future Memory Robustness Challenges

Technology Scaling Worsens Vulnerability





DRAM cells become **increasingly**
more vulnerable (to **read disturbance**)

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

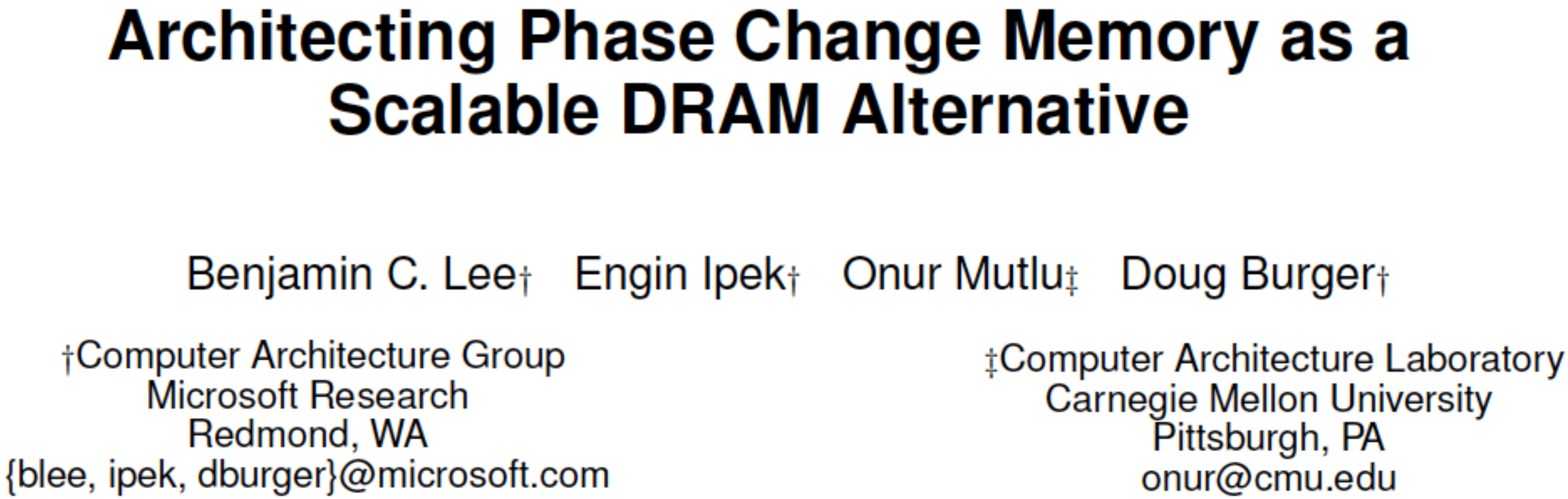
- DRAM

- Flash memory

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

Emerging Memories Also Need Intelligent Controllers

- Benjamin C. Lee, Engin Ipek, Onur Mutlu, and Doug Burger, **"Architecting Phase Change Memory as a Scalable DRAM Alternative"** *Proceedings of the 36th International Symposium on Computer Architecture* (***ISCA***), pages 2-13, Austin, TX, June 2009. Slides (pdf)
One of the 13 computer architecture papers of 2009 selected as Top Picks by IEEE Micro. Selected as a CACM Research Highlight.
2022 Persistent Impact Prize.



Architecting Phase Change Memory as a Scalable DRAM Alternative

Benjamin C. Lee† Engin Ipek† Onur Mutlu‡ Doug Burger†

†Computer Architecture Group
Microsoft Research
Redmond, WA
{blee, ipek, dburger}@microsoft.com

‡Computer Architecture Laboratory
Carnegie Mellon University
Pittsburgh, PA
onur@cmu.edu

The Takeaway

Intelligent

Memory Controllers

Enhance Robustness

& Enable Better Scaling

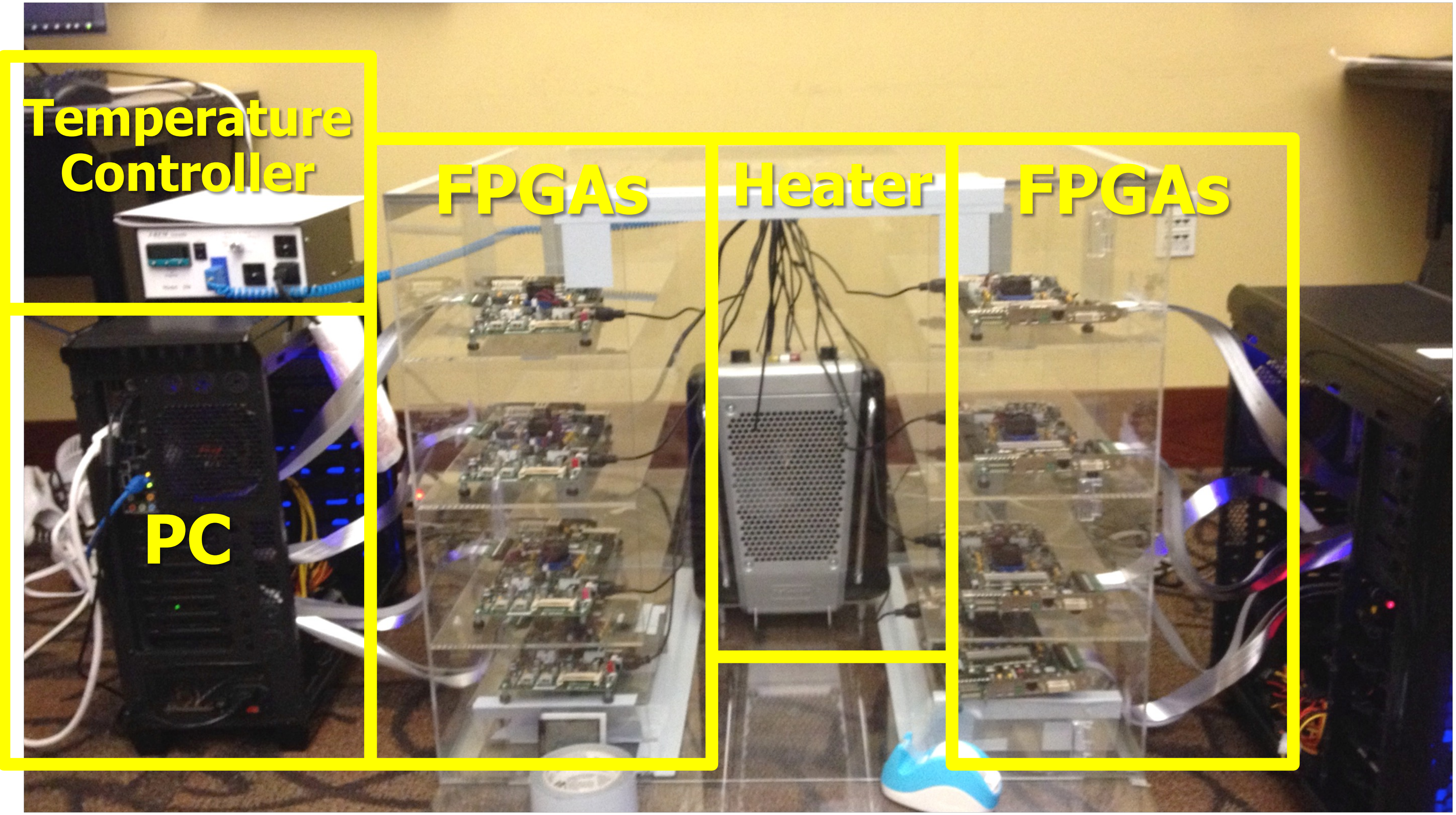
Architecting Robust Memory Systems

- **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

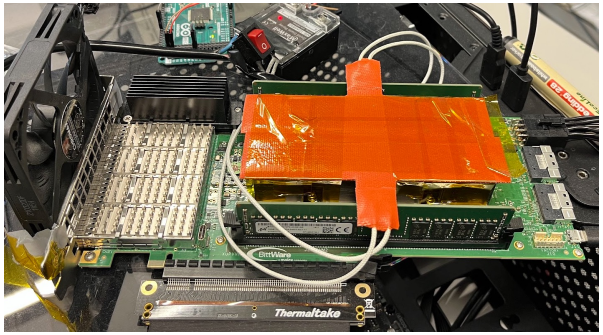
Understand and Model with Experiments (DRAM)

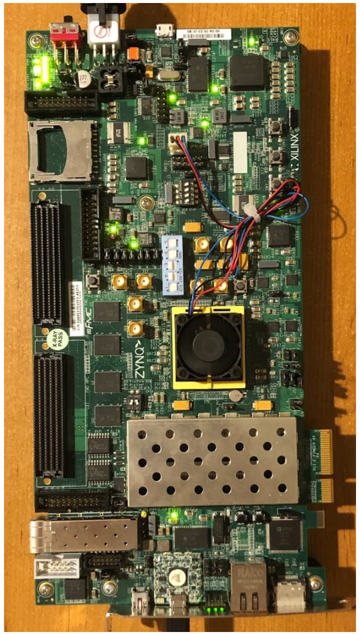


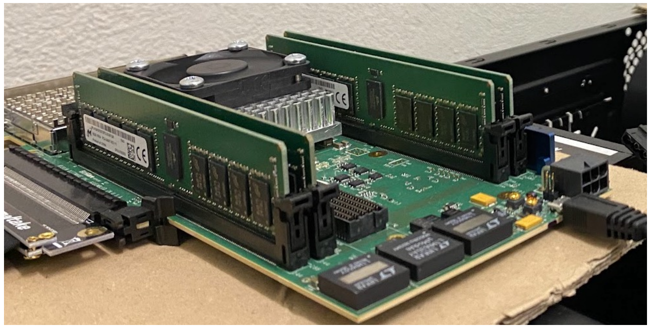
Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.

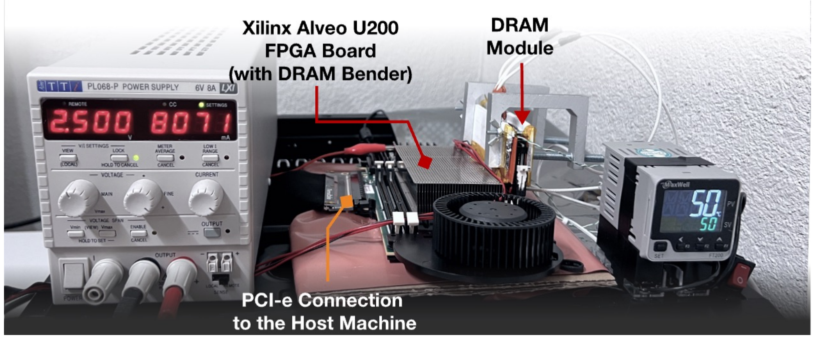
Understand and Model with Experiments (DRAM)

Five out of the box FPGA-based prototypes



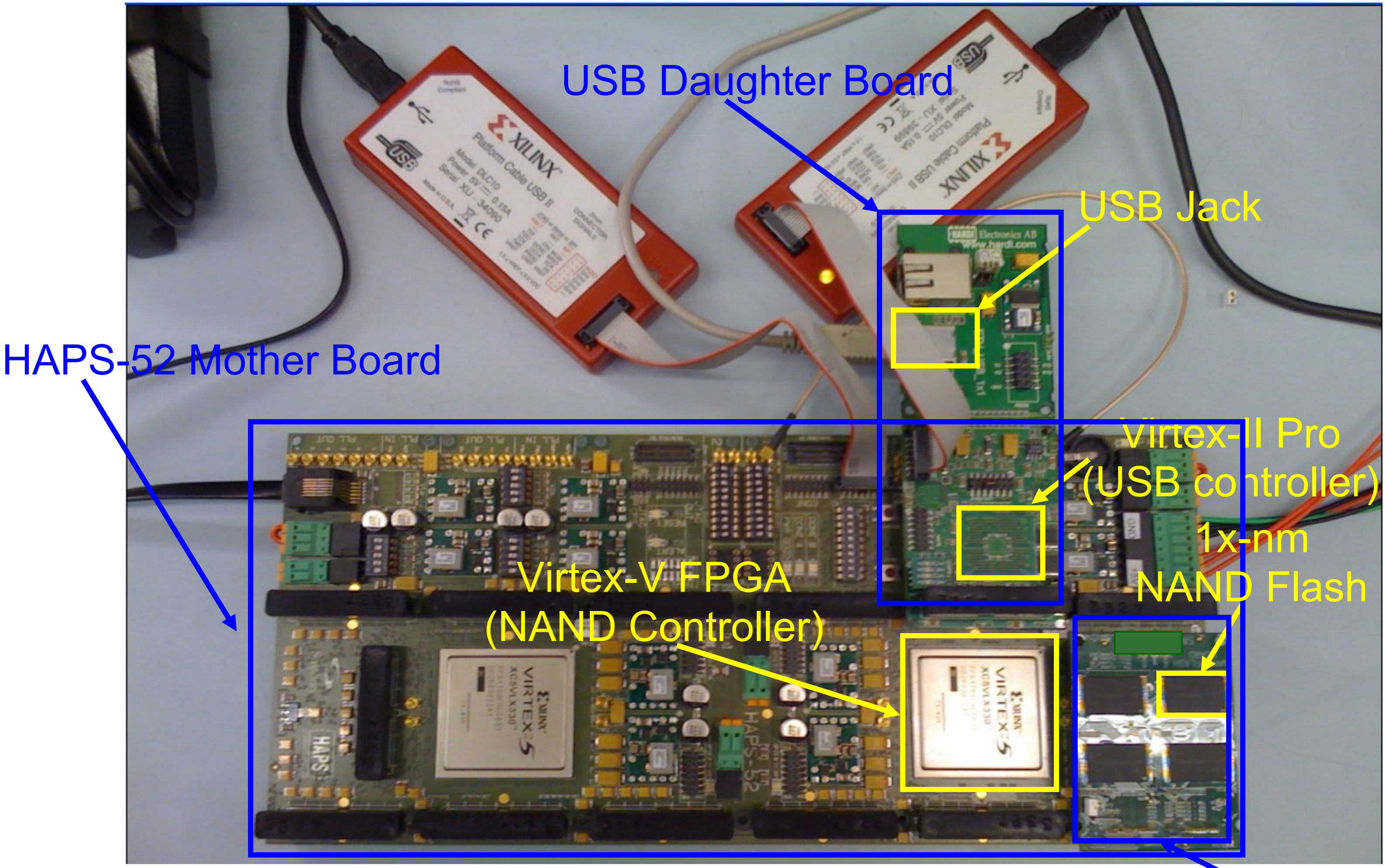






https://github.com/CMU-SAFARI/DRAM-Bender

Understand and Model with Experiments (Flash)



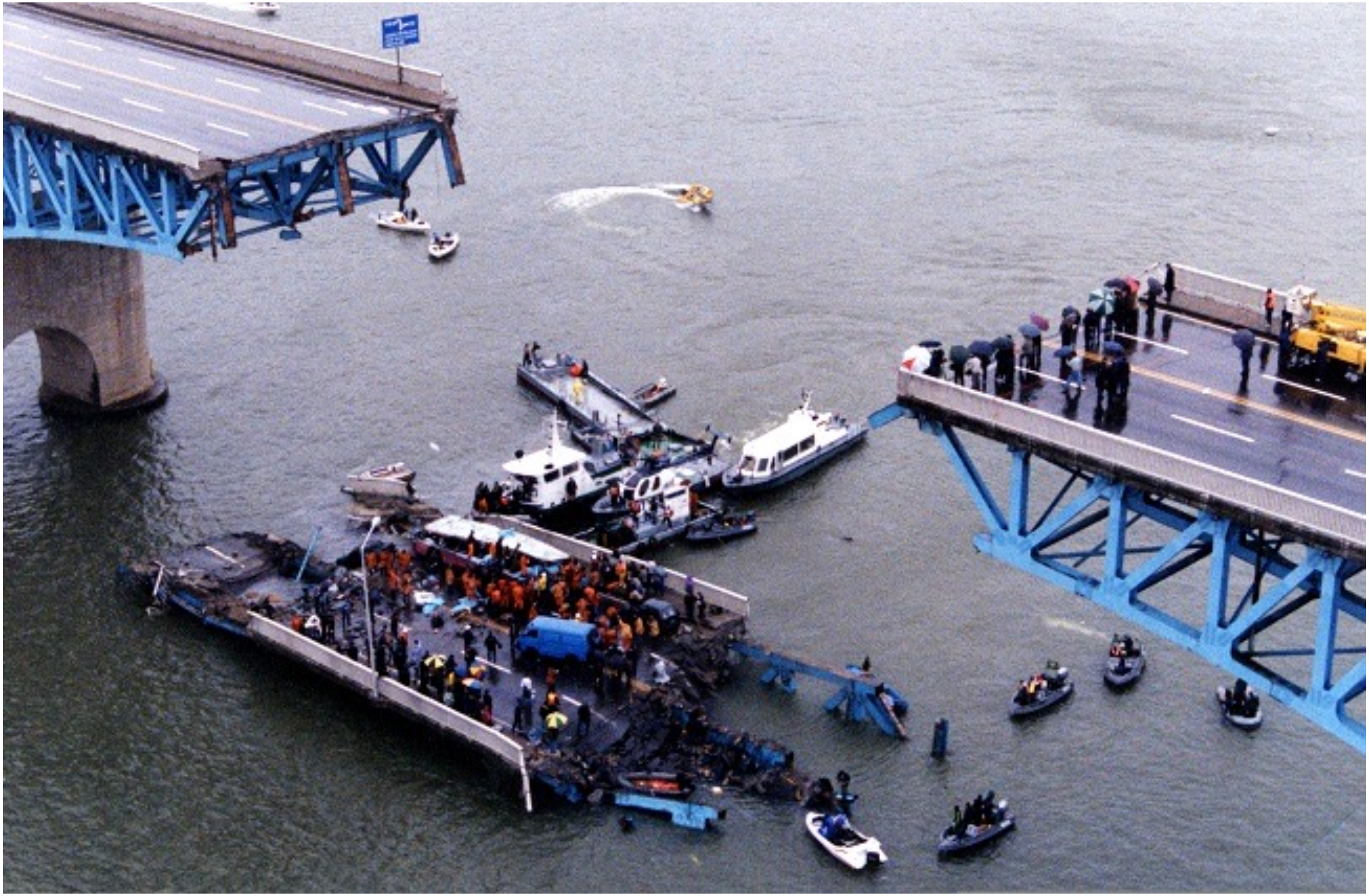
[DATE 2012, ICCD 2012, DATE 2013, ITJ 2013, ICCD 2013, SIGMETRICS 2014, HPCA 2015, DSN 2015, MSST 2015, JSAC 2016, HPCA 2017, DFRWS 2017, PIEEE 2017, HPCA 2018, SIGMETRICS 2018]

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

Collapse of the "Galloping Gertie" (1940)

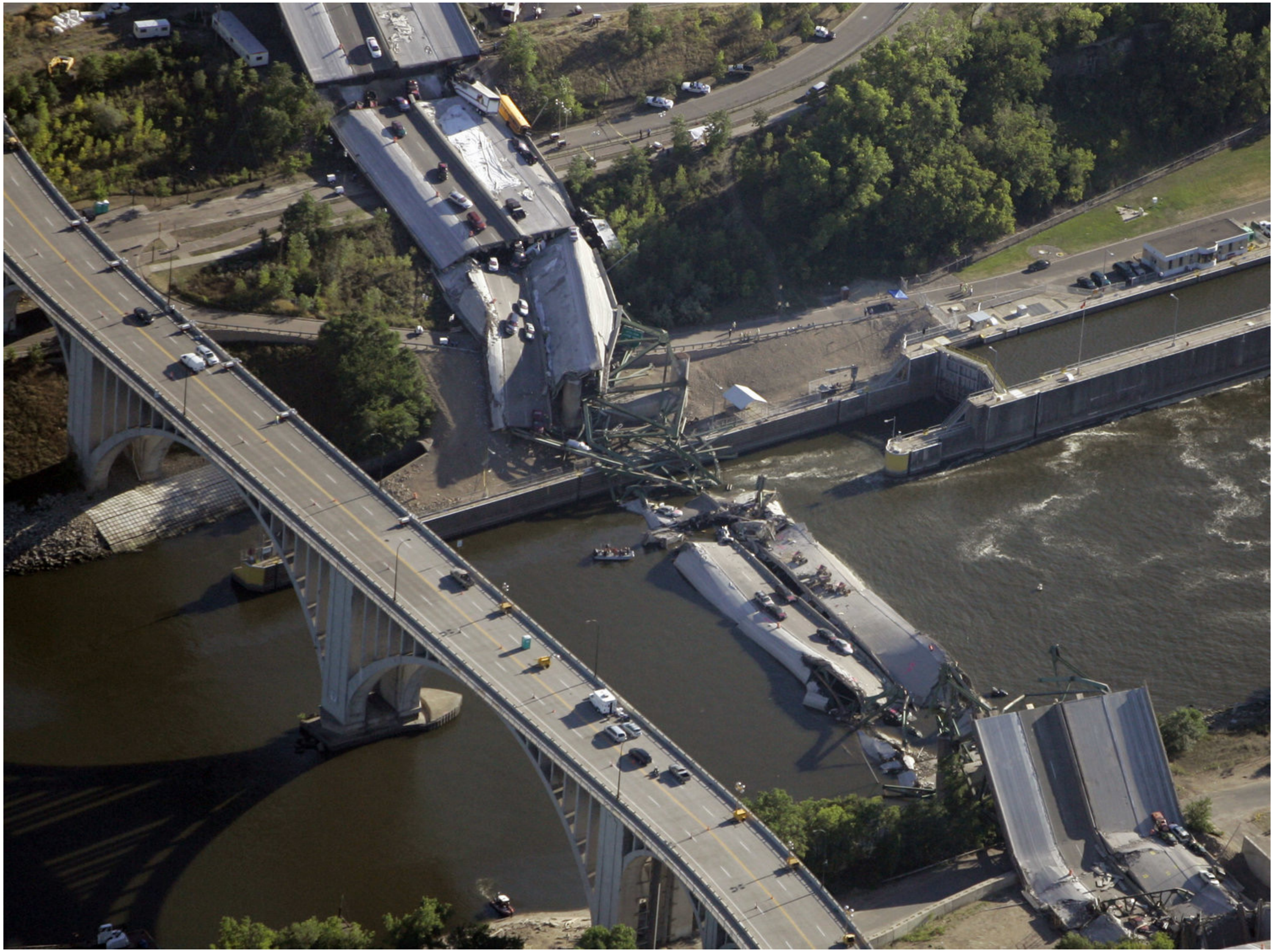
Source: AP
http://www.wsdot.wa.gov/tnbhistory/connections/connections3.htm

Another Example (1994)



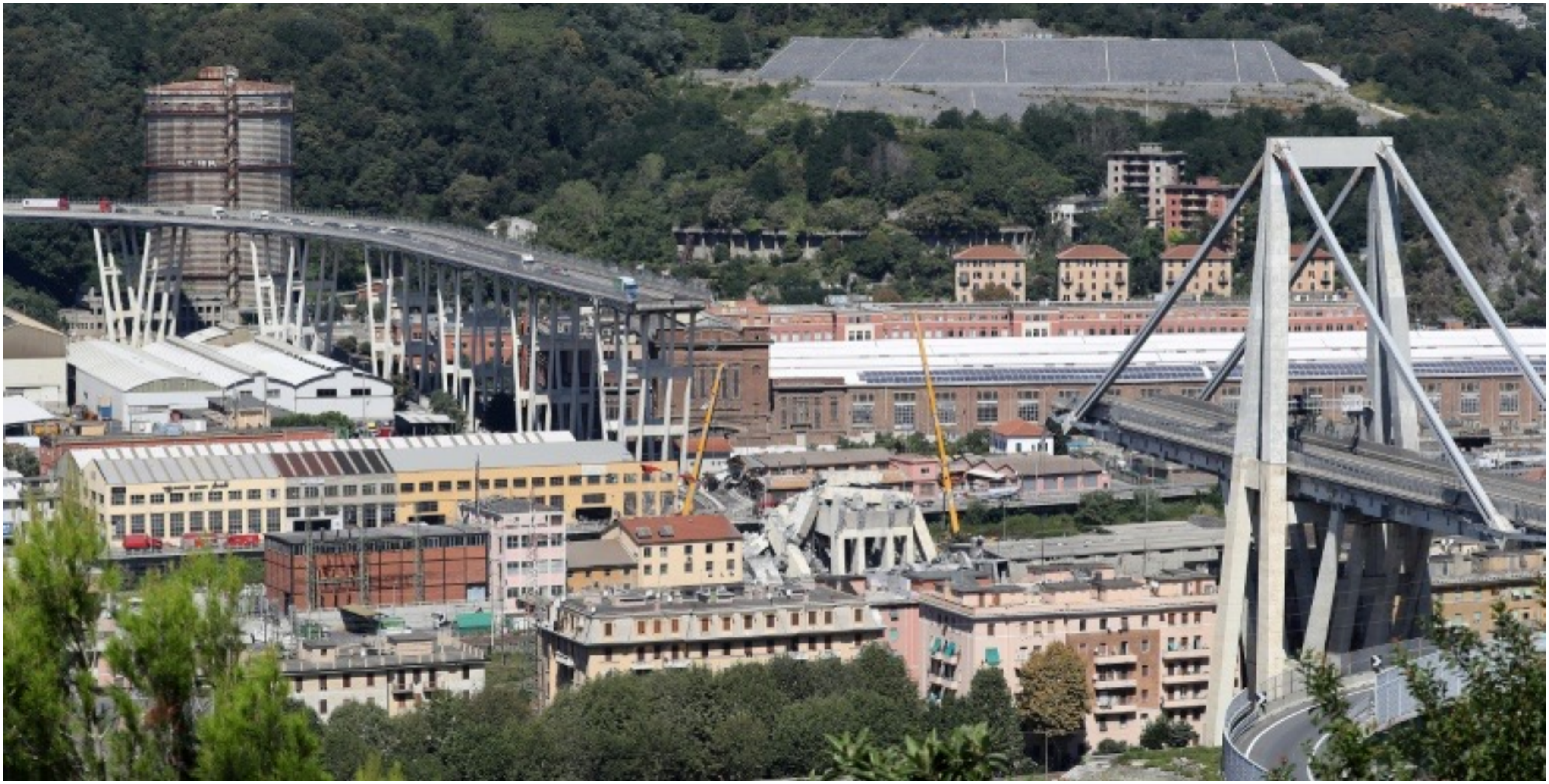
Source: By 최광모 - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=35197984

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

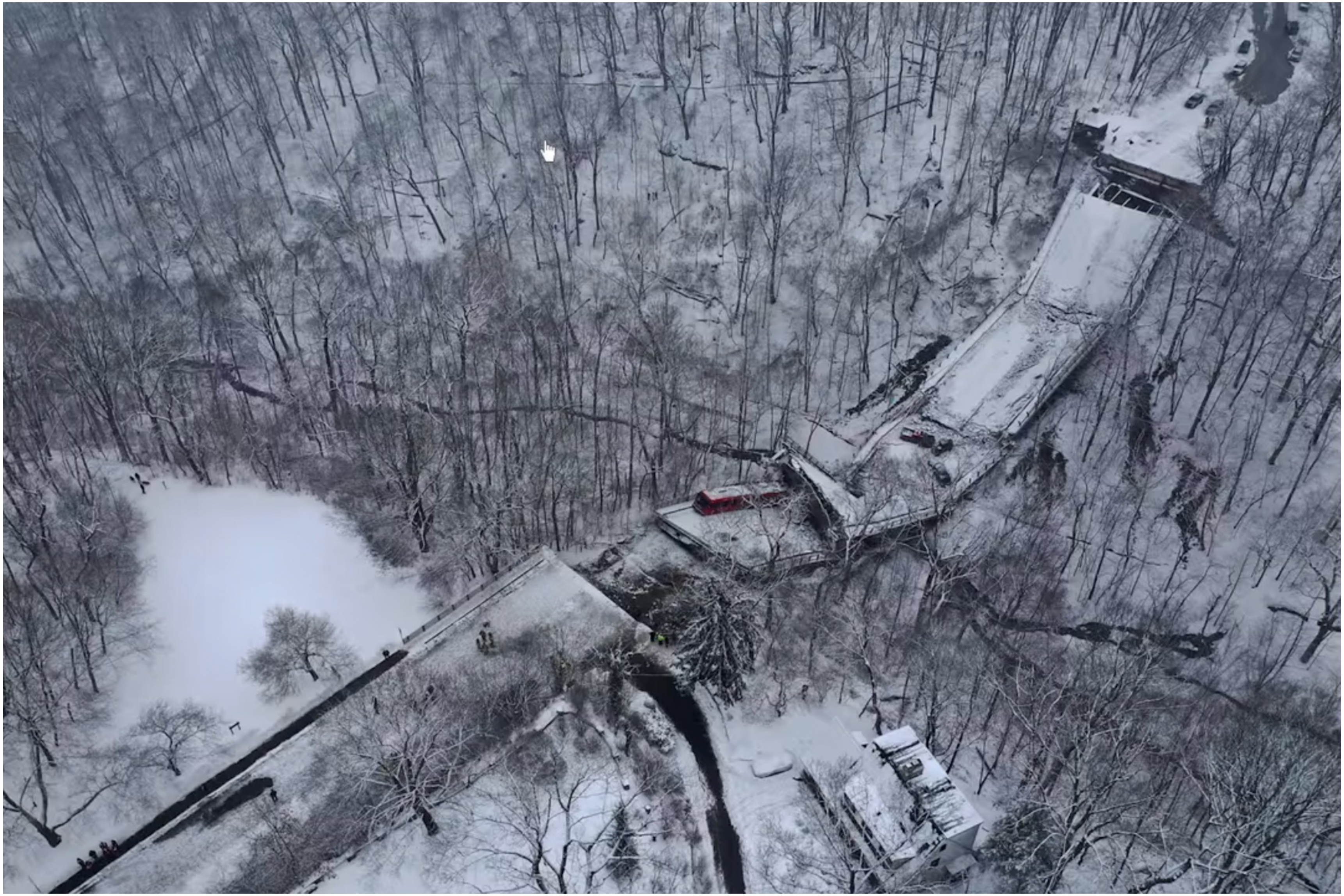
A More Recent Example (2018)



SAFARI

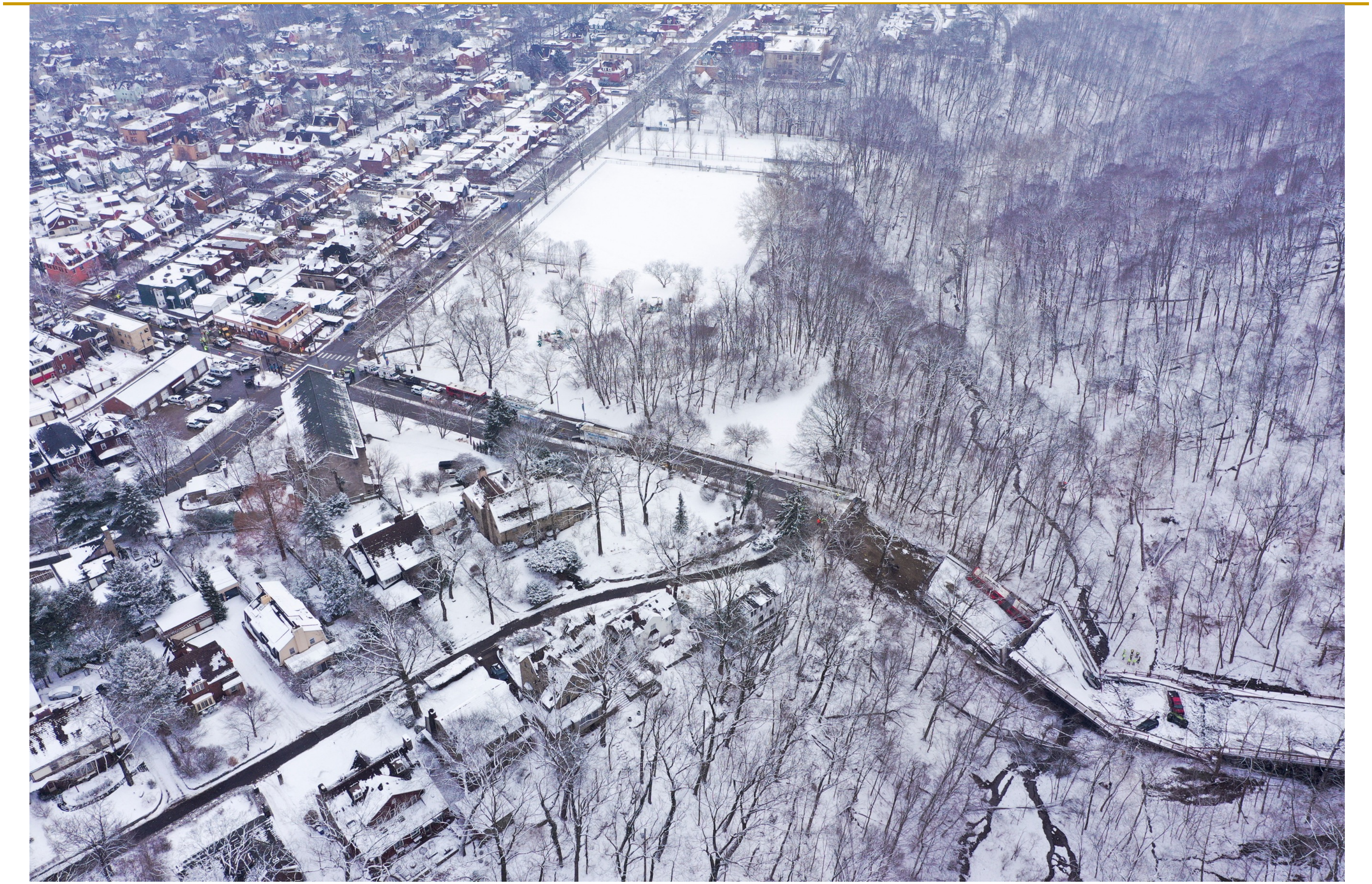
Source: AFP / Valery HACHE, https://www.capitalfm.co.ke/news/2018/08/genoa-bridge-collapse-what-we-know/

A Most Recent Example (2022)



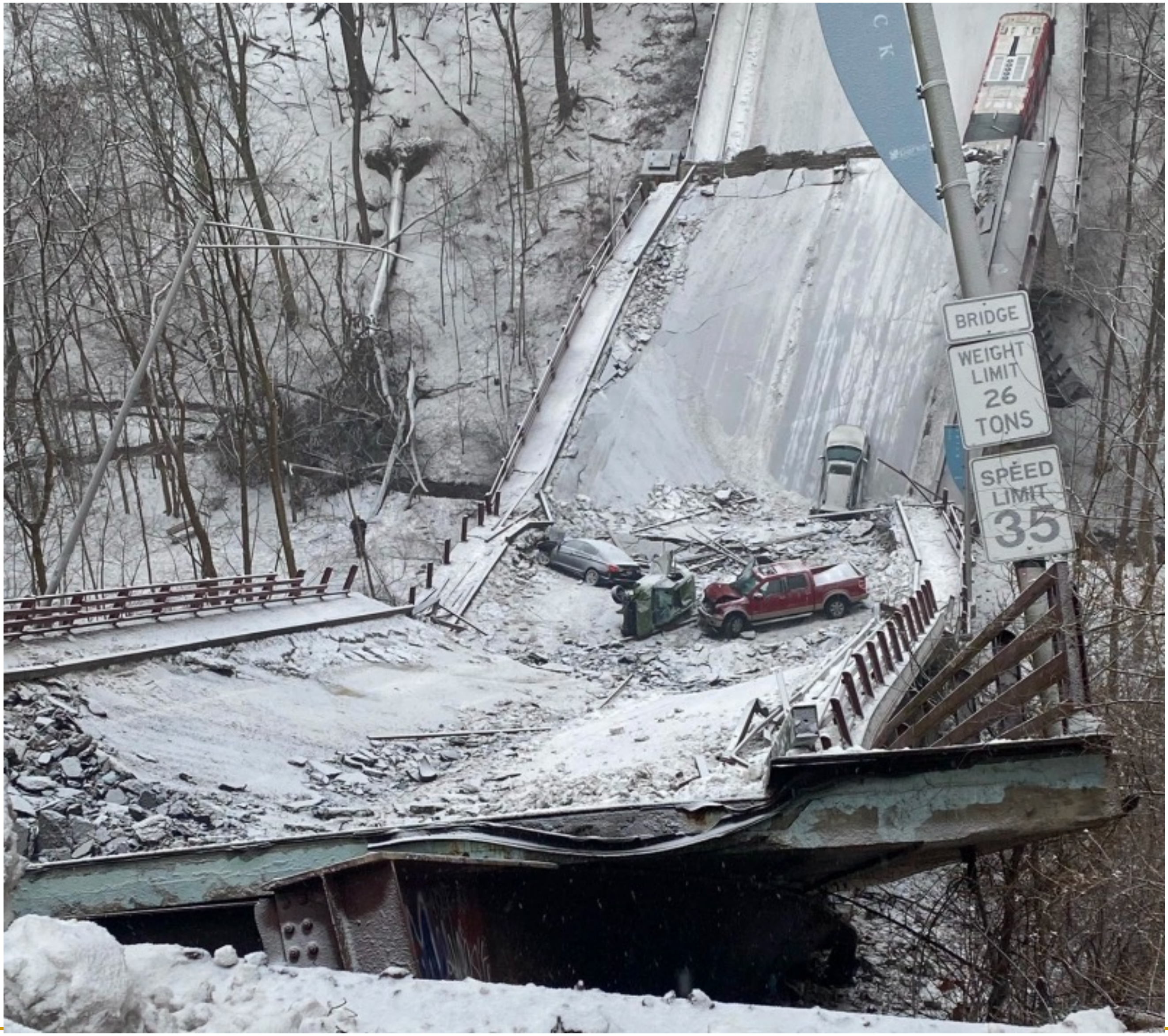
https://www.post-gazette.com/local/city/2022/01/28/pittsburgh-bridge-collapse-forbes-braddock-avenue-point-breeze-squirrel-hill/stories/202201280075

A Most Recent Example (2022)



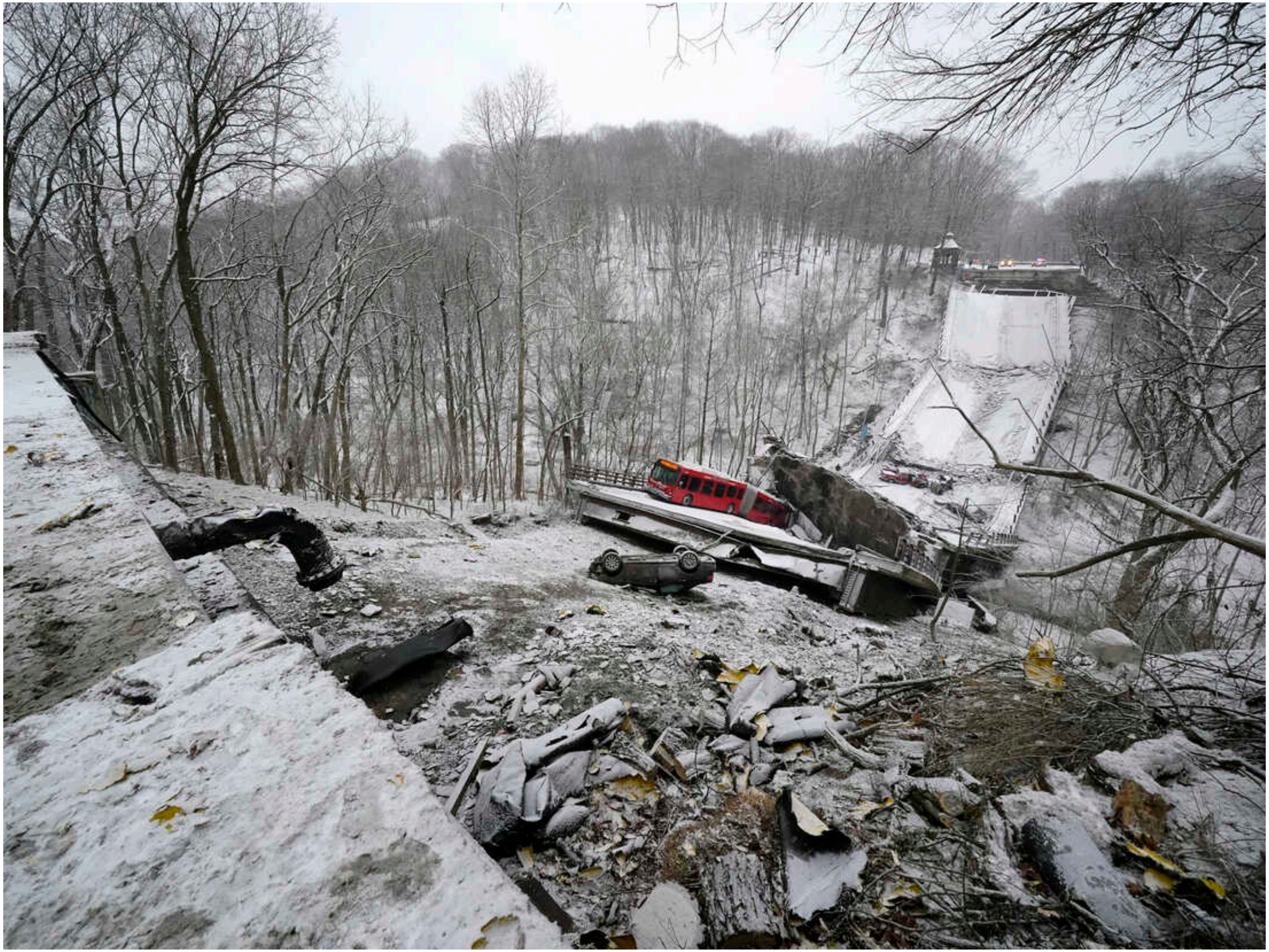
https://www.reuters.com/world/us/biden-highlight-manufacturing-jobs-gdp-growth-pittsburgh-2022-01-28/

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

Challenge and Opportunity for Future

Fundamentally

Robust

(Reliable, Secure, Safe)

Computing Architectures

Final Thoughts on RowHammer

Aside: Byzantine Failures

- This class of failures is known as Byzantine failures

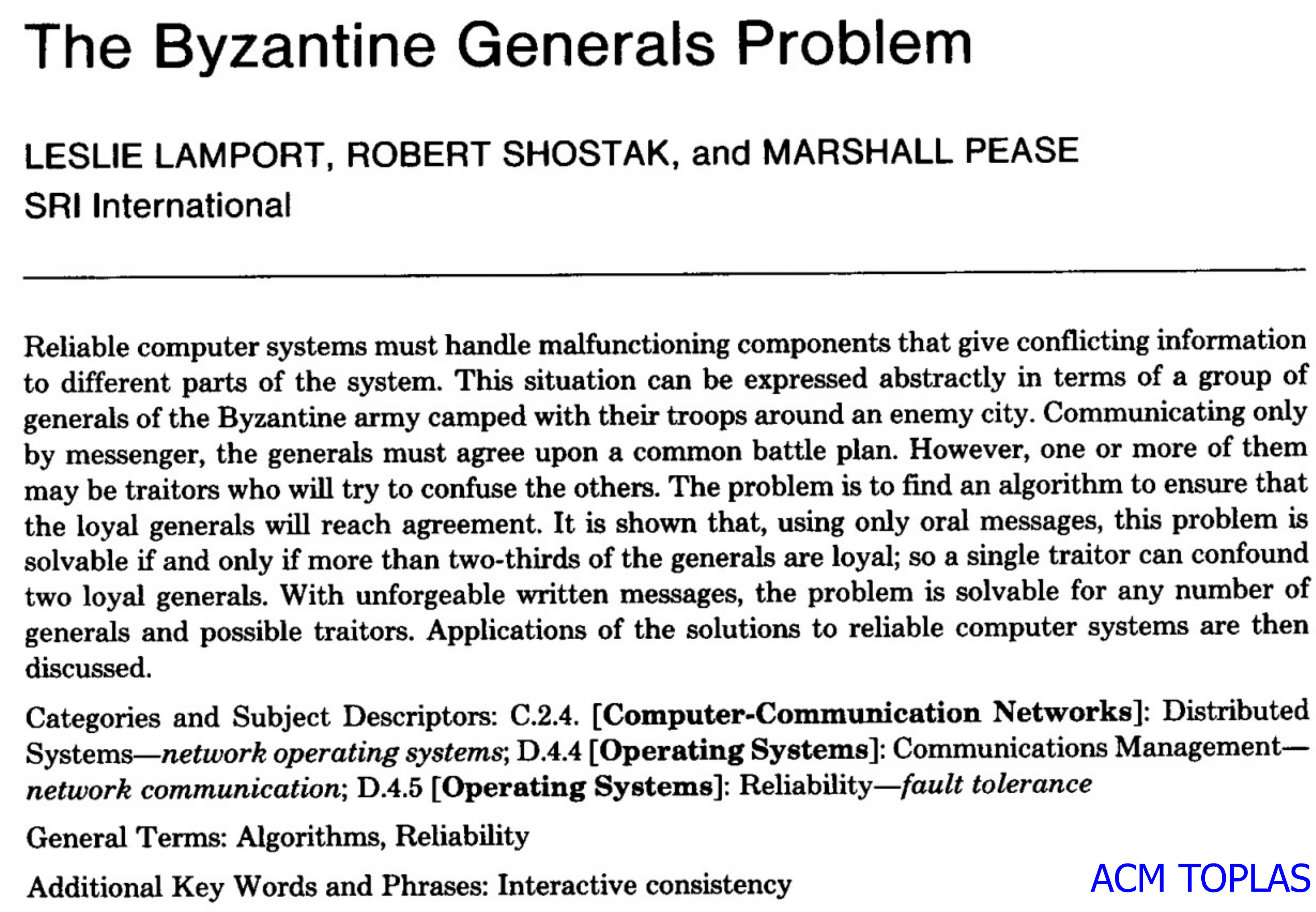
- Characterized by
  - Undetected erroneous computation
  - Opposite of "fail fast (with an error or no result)"

- "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.

Slide credit: Mahadev Satyanarayanan, CMU, 15-440, Spring 2015

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*

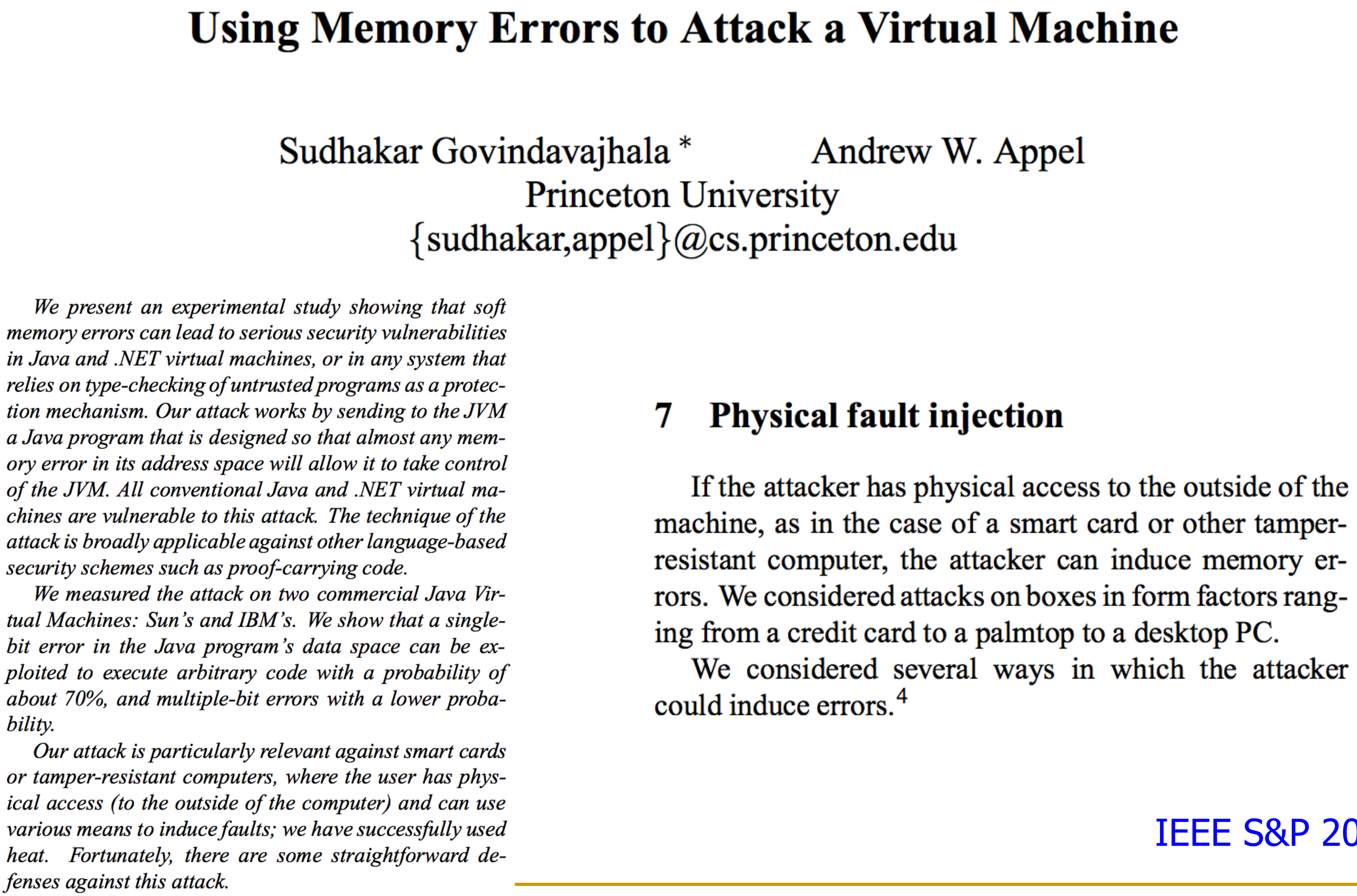
General Terms: Algorithms, Reliability

Additional Key Words and Phrases: Interactive consistency

ACM TOPLAS 1982

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

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

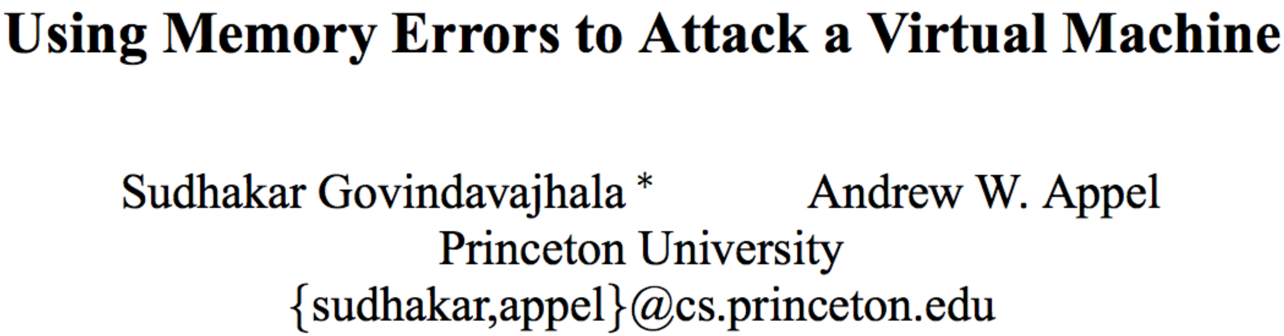
If the attacker has physical access to the outside of the machine, as in the case of a smart card or other tamper-resistant computer, the attacker can induce memory errors. We considered attacks on boxes in form factors ranging from a credit card to a palmtop to a desktop PC.

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

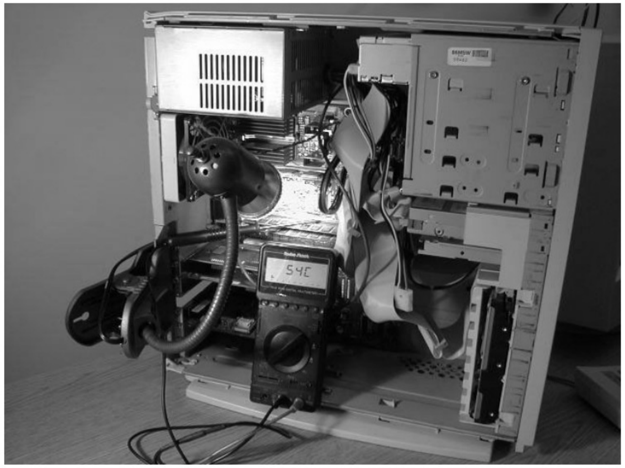


Figure 3. Experimental setup to induce memory errors, showing a PC built from surplus components, clip-on gooseneck lamp, 50-watt spotlight bulb, and digital thermometer. Not shown is the variable AC power supply for the lamp.

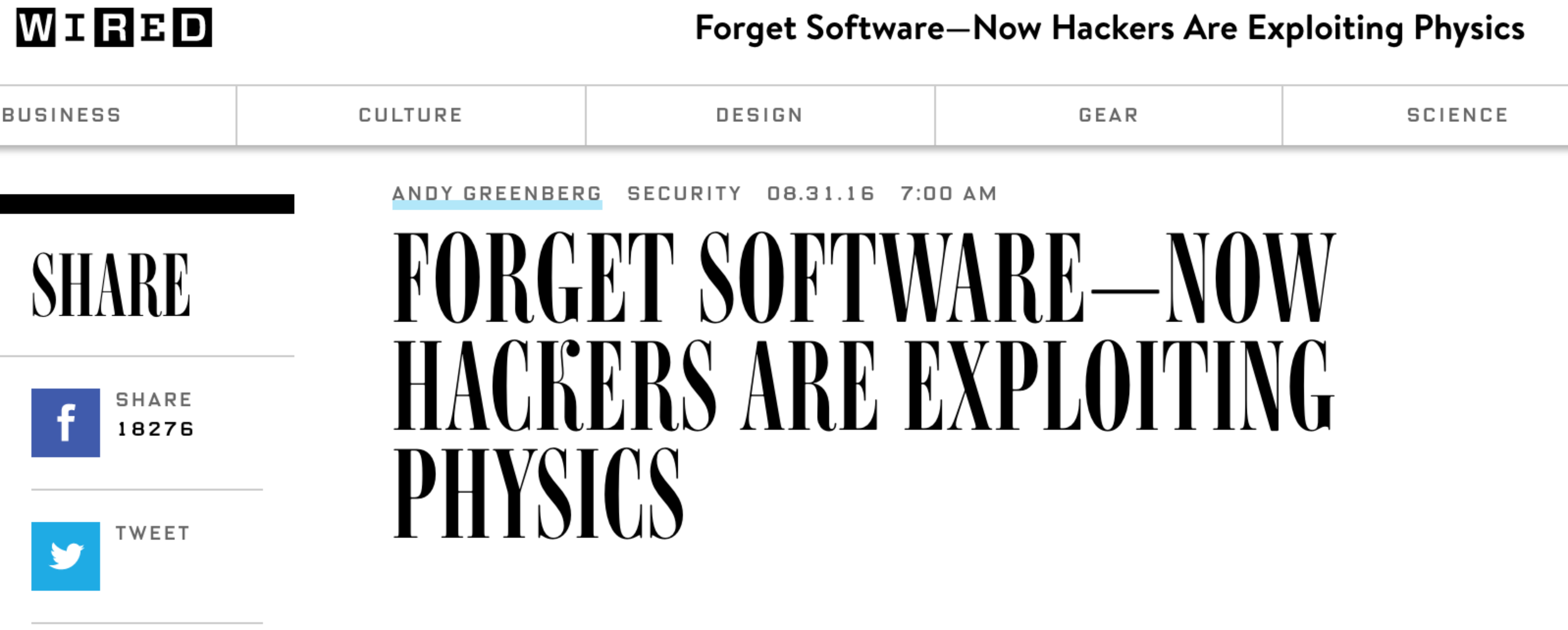
IEEE S&P 2003

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

After RowHammer

A simple, exploitable memory error

can be induced by software



Forget Software—Now Hackers Are Exploiting Physics

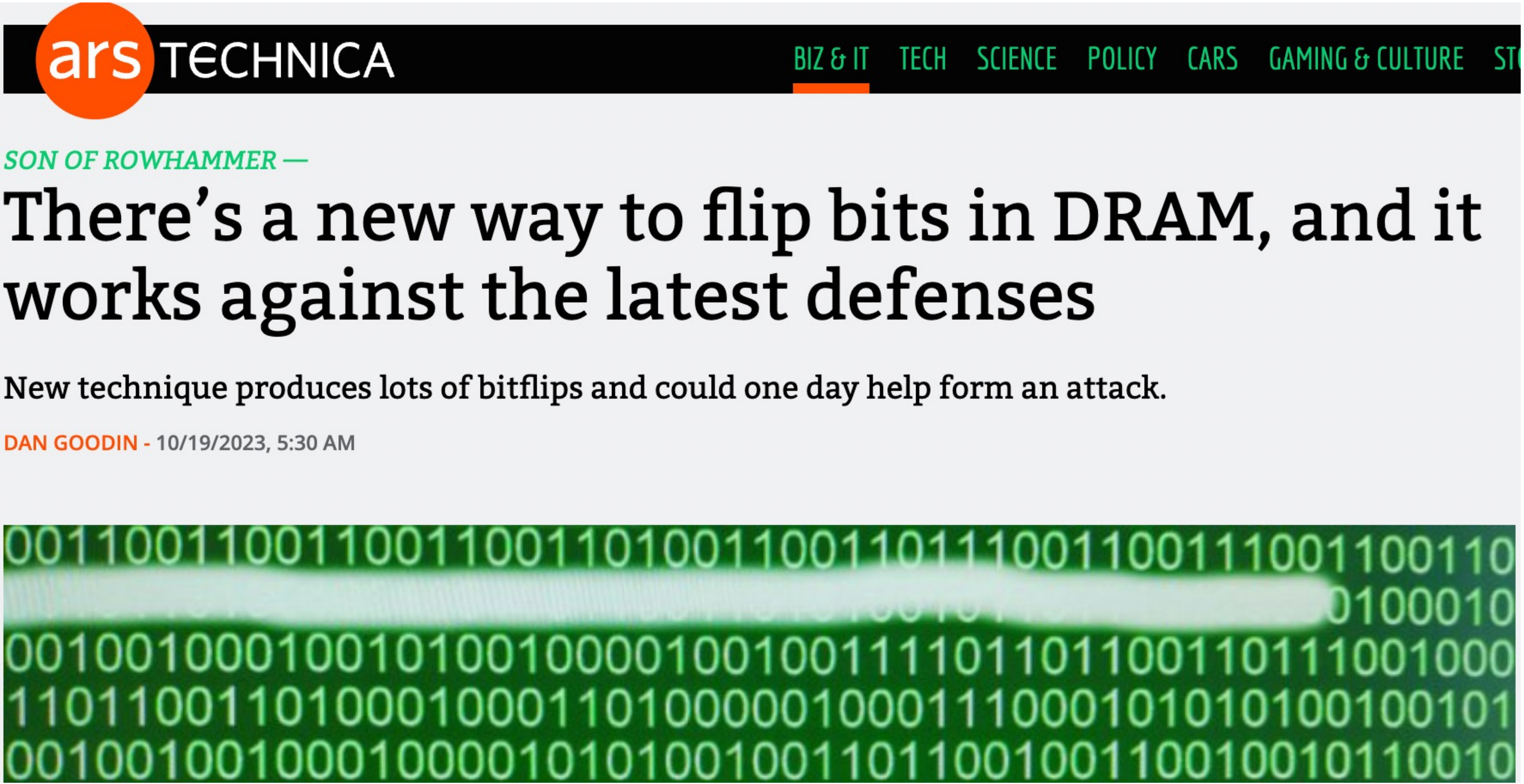
ANDY GREENBERG SECURITY 08.31.16 7:00 AM

FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

After RowHammer

A simple, exploitable memory error

can be induced by software



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 Usenix Security 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

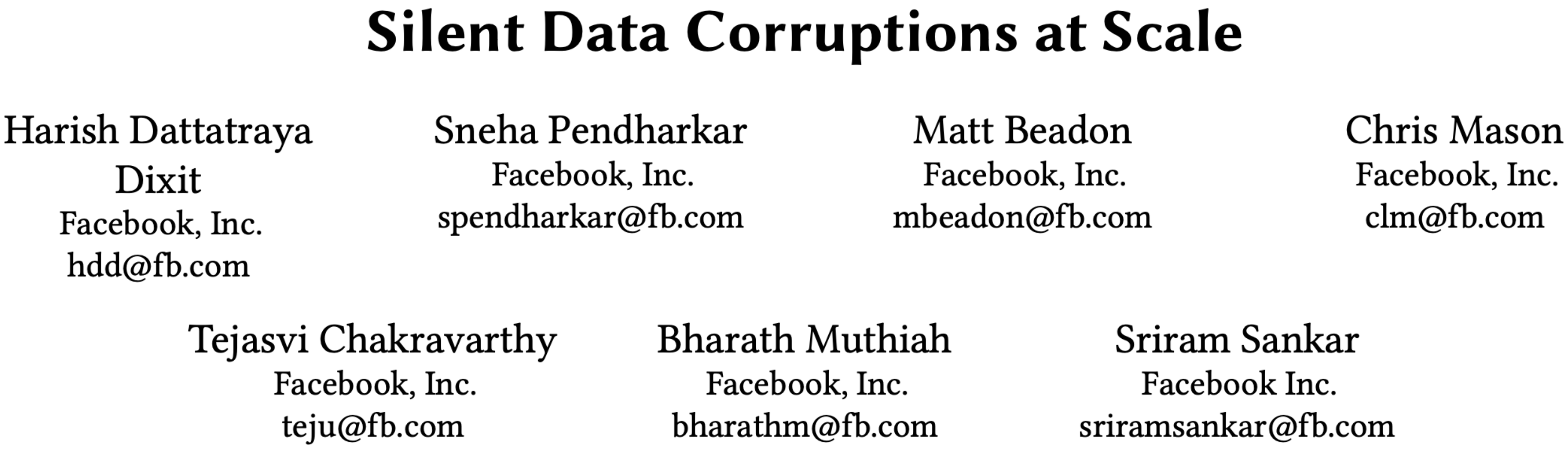
- **Good news: We have a lot more to do**
  - We are now fully aware hardware is easily fallible
  - We are developing both attacks and defenses
  - We are developing principled models, methodologies, solutions

(Silent) Data Corruption in Logic

Data Corruption is in Logic, Too

- Intermittent defects can cause silent data corruption
- They may be hard to detect or replicate
- They may be exploitable

Silent Data Corruption in Logic (2021)



Silent Data Corruptions at Scale

Harish Dattatraya Dixit
Facebook, Inc.
hdd@fb.com

Sneha Pendharkar
Facebook, Inc.
spendharkar@fb.com

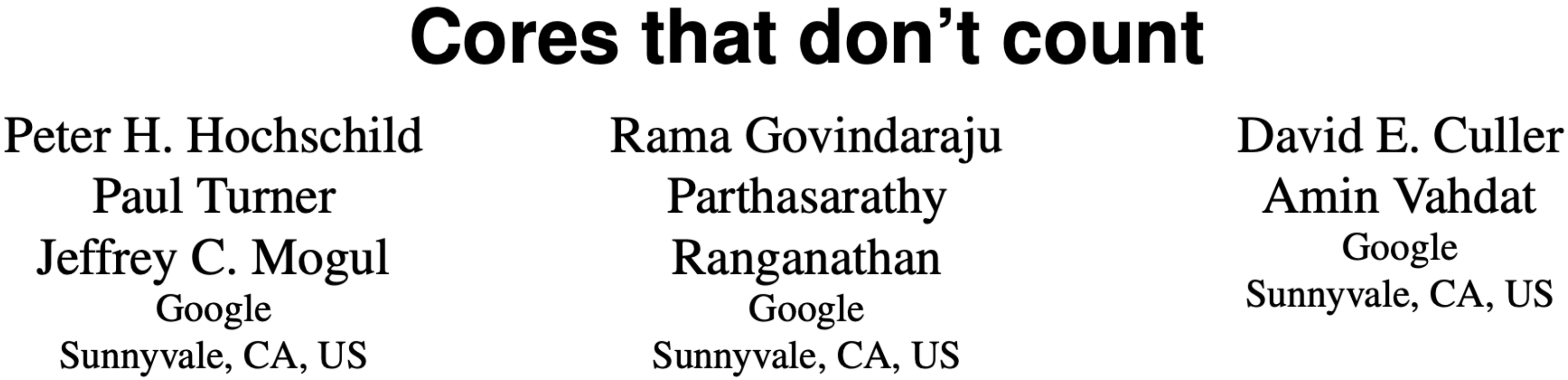
Matt Beadon
Facebook, Inc.
mbeadon@fb.com

Chris Mason
Facebook, Inc.
clm@fb.com

Tejasvi Chakravarthy
Facebook, Inc.
teju@fb.com

Bharath Muthiah
Facebook, Inc.
bharathm@fb.com

Sriram Sankar
Facebook Inc.
sriramsankar@fb.com



Cores that don’t count

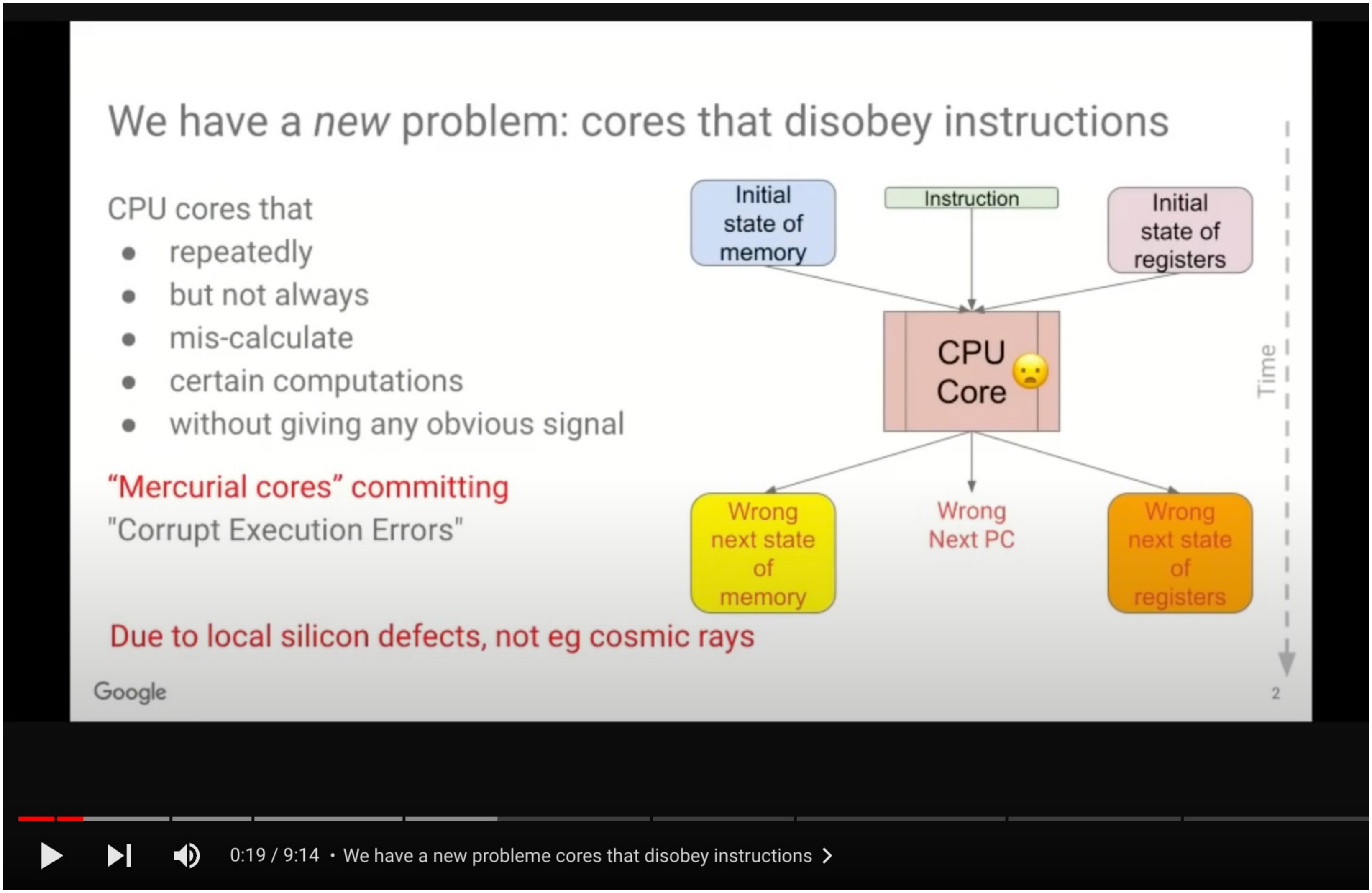
Peter H. Hochschild
Paul Turner
Jeffrey C. Mogul
Google
Sunnyvale, CA, US

Rama Govindaraju
Parthasarathy
Ranganathan
Google
Sunnyvale, CA, US

David E. Culler
Amin Vahdat
Google
Sunnyvale, CA, US

https://www.youtube.com/watch?v=QMF3rqhjYuM

Silent Data Corruption In-the-Field (2021)



We have a *new* problem: cores that disobey instructions

CPU cores that

- repeatedly
- but not always
- mis-calculate
- certain computations
- without giving any obvious signal

"Mercurial cores" committing

"Corrupt Execution Errors"

Due to local silicon defects, not eg cosmic rays

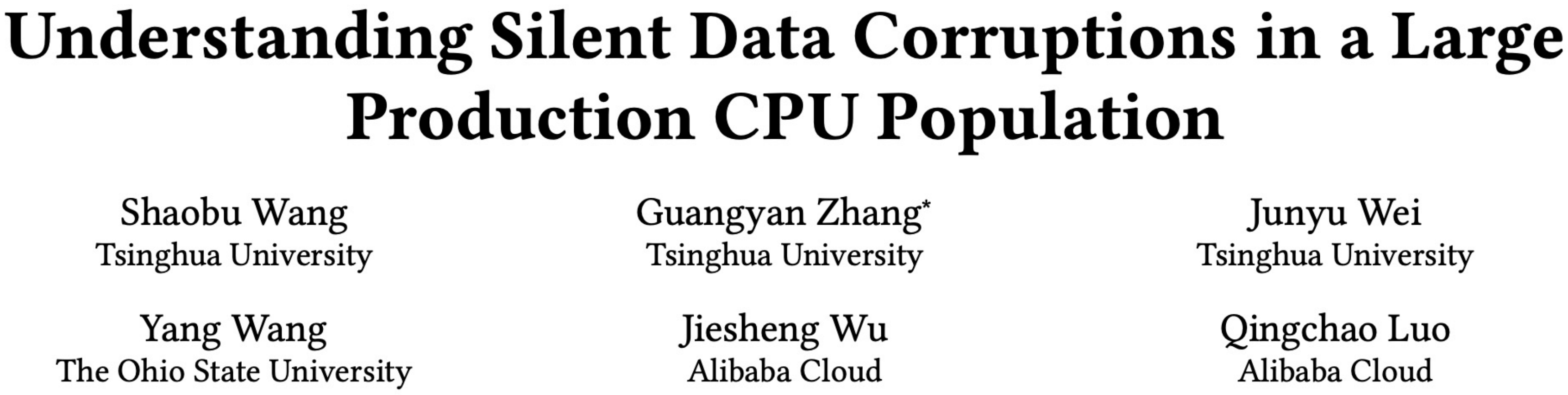
Initial state of memory
Instruction
Initial state of registers
CPU Core
Time
Wrong next state of memory
Wrong Next PC
Wrong next state of registers
Google
2

0:19 / 9:14 • We have a new probleme cores that disobey instructions

HotOS 2021: Cores That Don't Count (Fun Hardware)

https://www.youtube.com/watch?v=QMF3rqhjYuM

Silent Data Corruption in Logic (2023)



Understanding Silent Data Corruptions in a Large Production CPU Population

Shaobu Wang
Tsinghua University

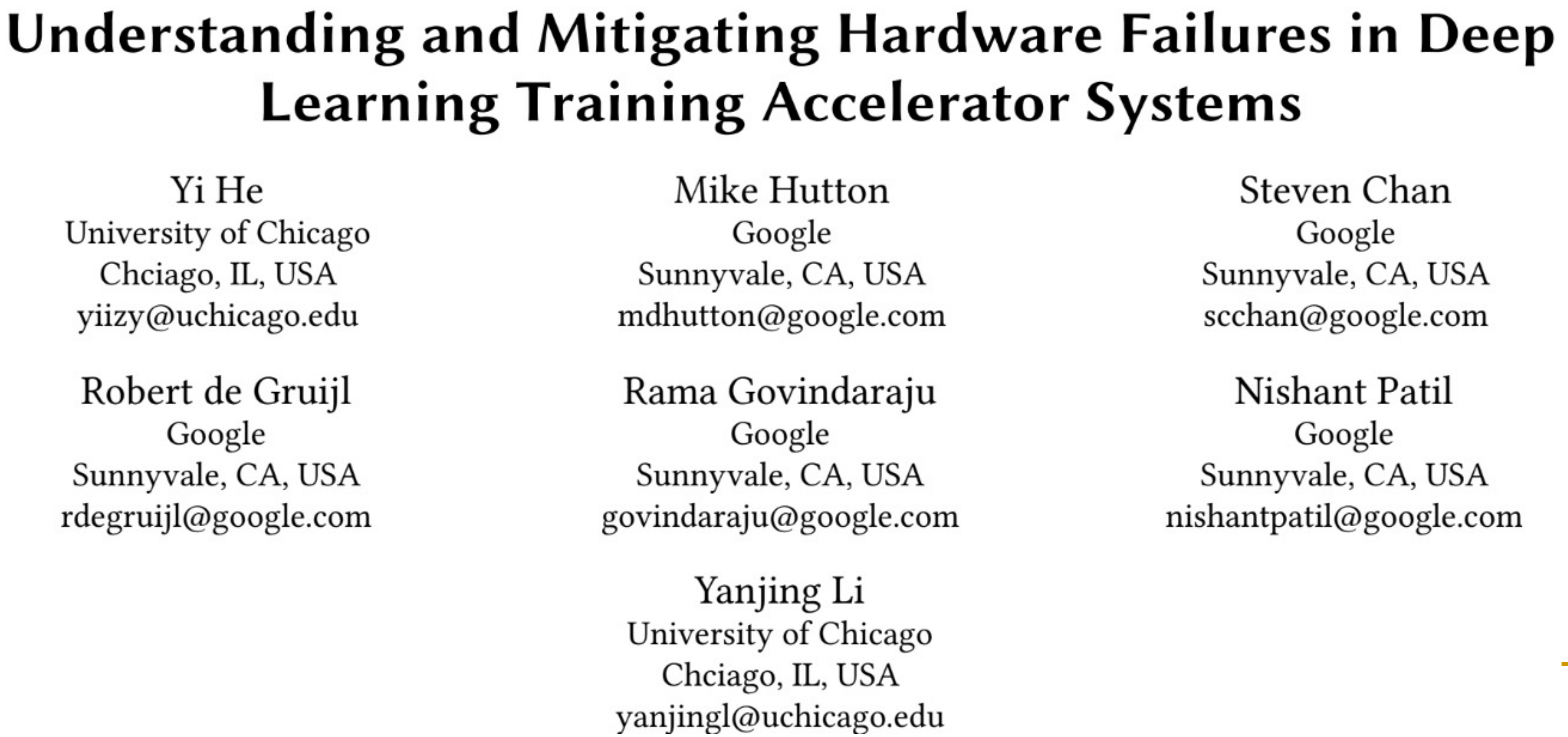
Guangyan Zhang*
Tsinghua University

Junyu Wei
Tsinghua University

Yang Wang
The Ohio State University

Jiesheng Wu
Alibaba Cloud

Qingchao Luo
Alibaba Cloud



Understanding and Mitigating Hardware Failures in Deep Learning Training Accelerator Systems

Yi He
University of Chicago
Chciago, IL, USA
yiizy@uchicago.edu

Mike Hutton
Google
Sunnyvale, CA, USA
mdhutton@google.com

Steven Chan
Google
Sunnyvale, CA, USA
scchan@google.com

Robert de Gruijl
Google
Sunnyvale, CA, USA
rdegruijl@google.com

Rama Govindaraju
Google
Sunnyvale, CA, USA
govindaraju@google.com

Nishant Patil
Google
Sunnyvale, CA, USA
nishantpatil@google.com

Yanjing Li
University of Chicago
Chciago, IL, USA
yanjingl@uchicago.edu

Takeaways

- Both memory and logic errors will become worse with technology scaling

- Hardware errors will create worse robustness problems

- **We cannot afford to ignore data corruption**

Acknowledgments

Acknowledgments

SAFARI

SAFARI Research Group

safari.ethz.ch

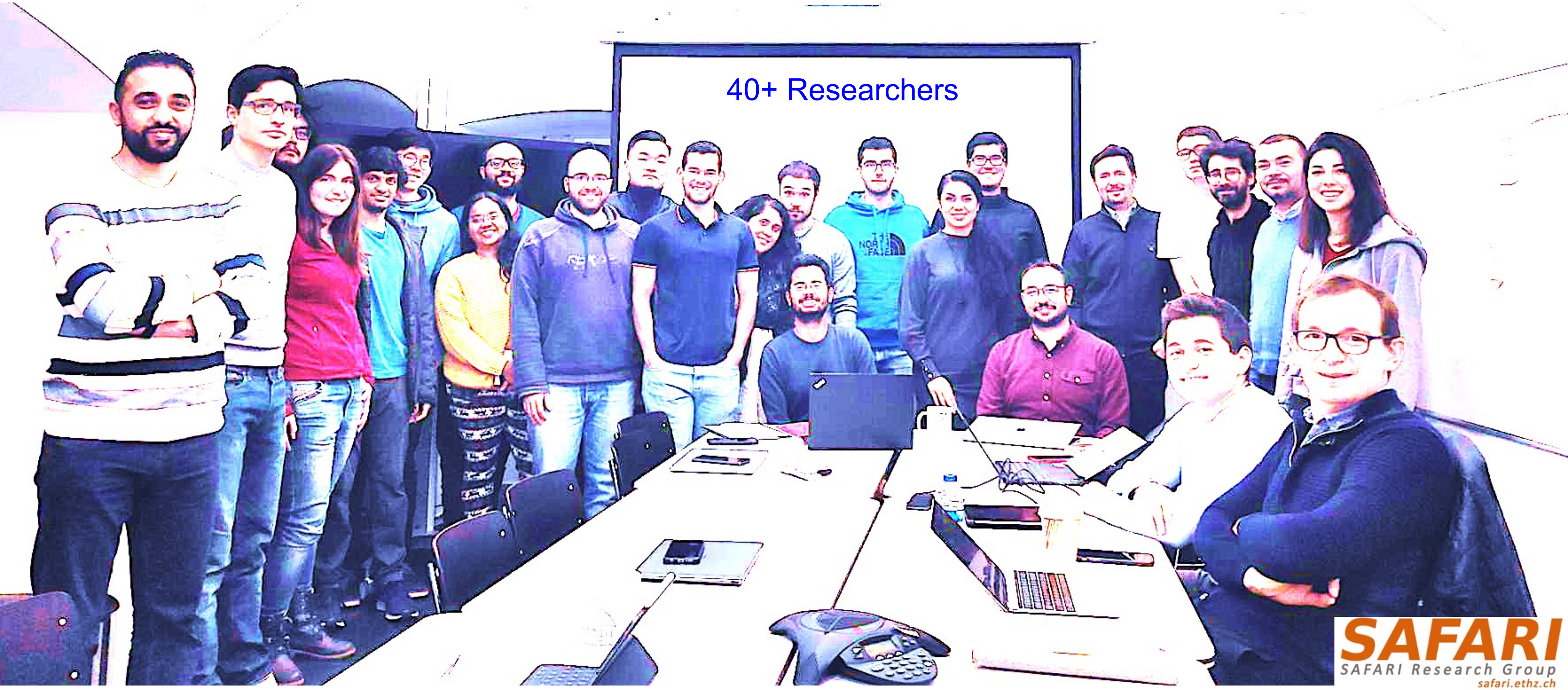
Think BIG, Aim HIGH!

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!

https://safari.ethz.ch

SAFARI Research Group: December 2021

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

SAFARI

SAFARI Research Group

Think Big, Aim High

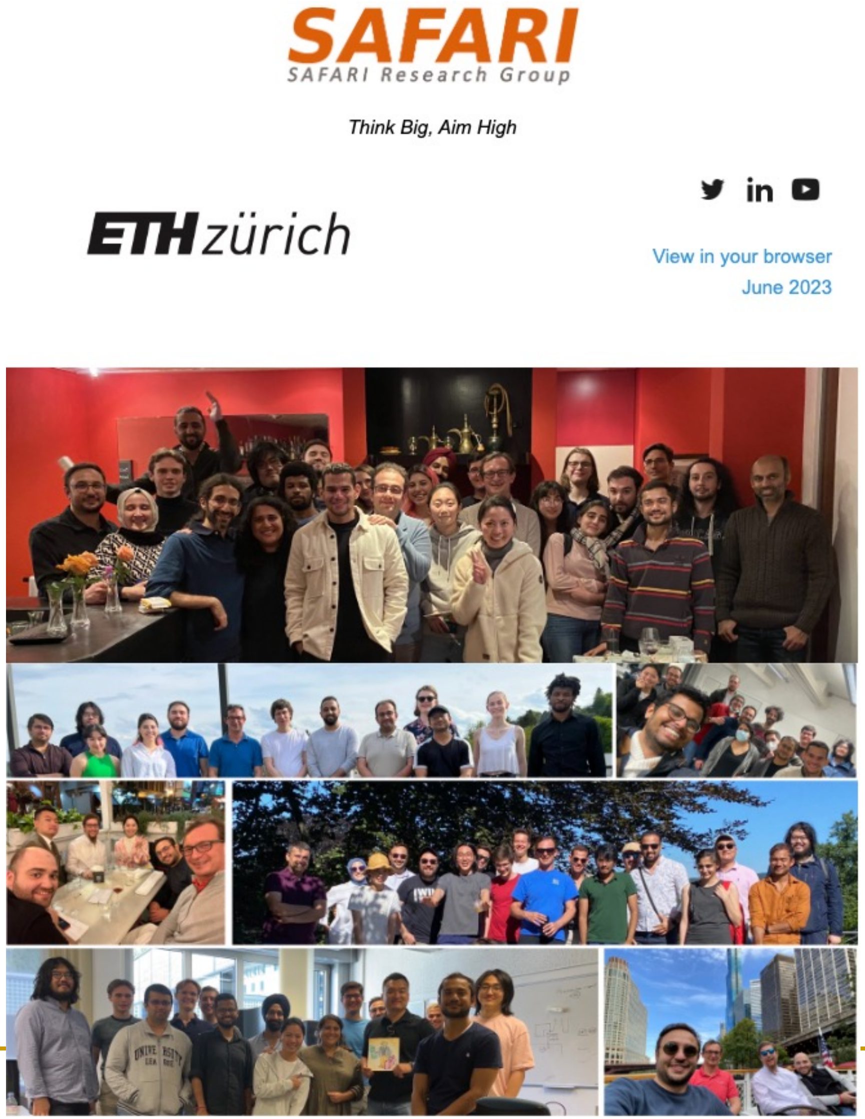
ETH zürich

View in your browser

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

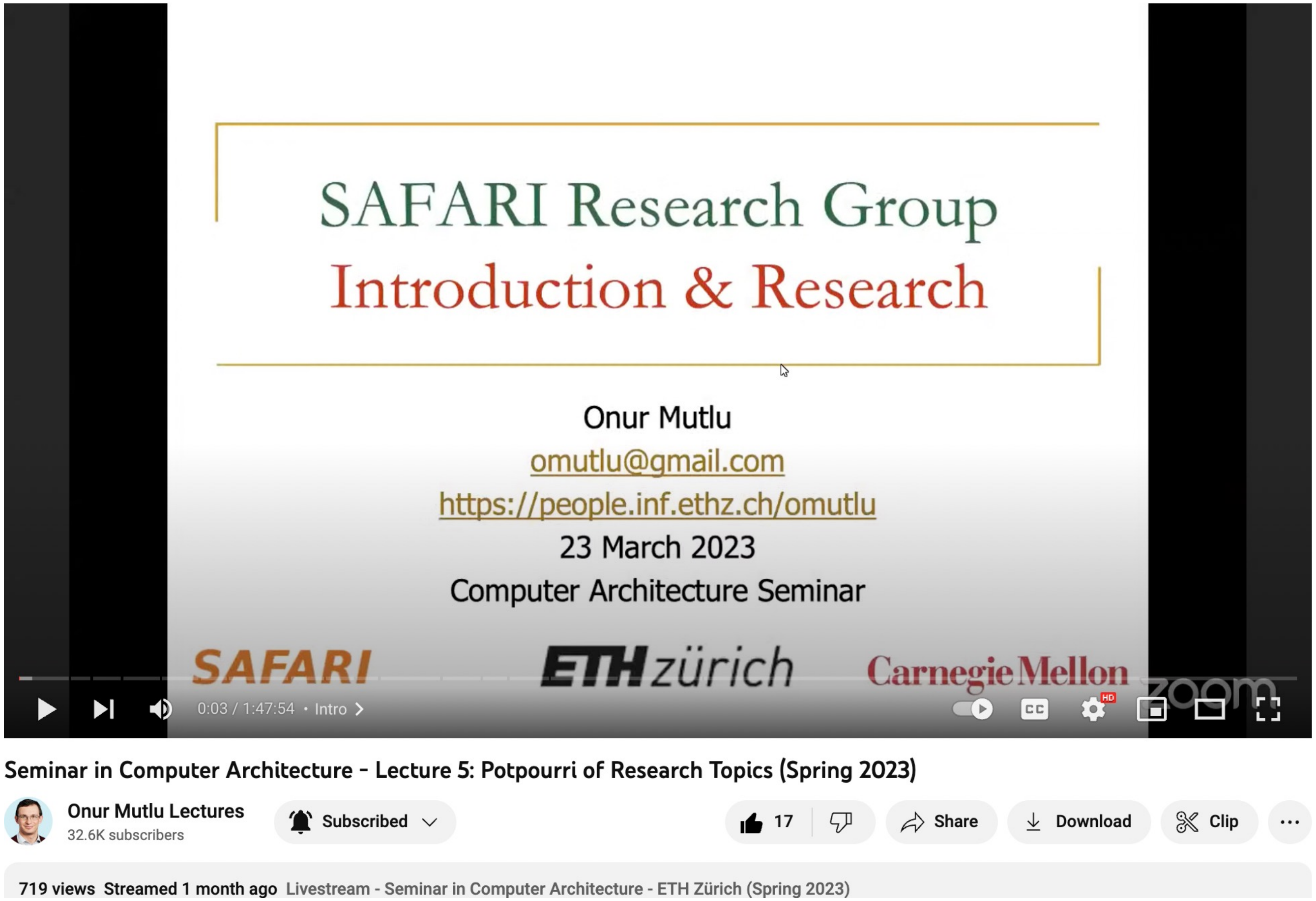
June 2023

SAFARI Newsletter July 2024 Edition

- **https://safari.ethz.ch/safari-newsletter-july-2024/**

SAFARI Introduction & Research

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



SAFARI Research Group

Introduction & Research

Onur Mutlu

omutlu@gmail.com

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

23 March 2023

Computer Architecture Seminar

Seminar in Computer Architecture - Lecture 5: Potpourri of Research Topics (Spring 2023)

719 views Streamed 1 month ago Livestream - Seminar in Computer Architecture - ETH Zürich (Spring 2023)

SAFARI

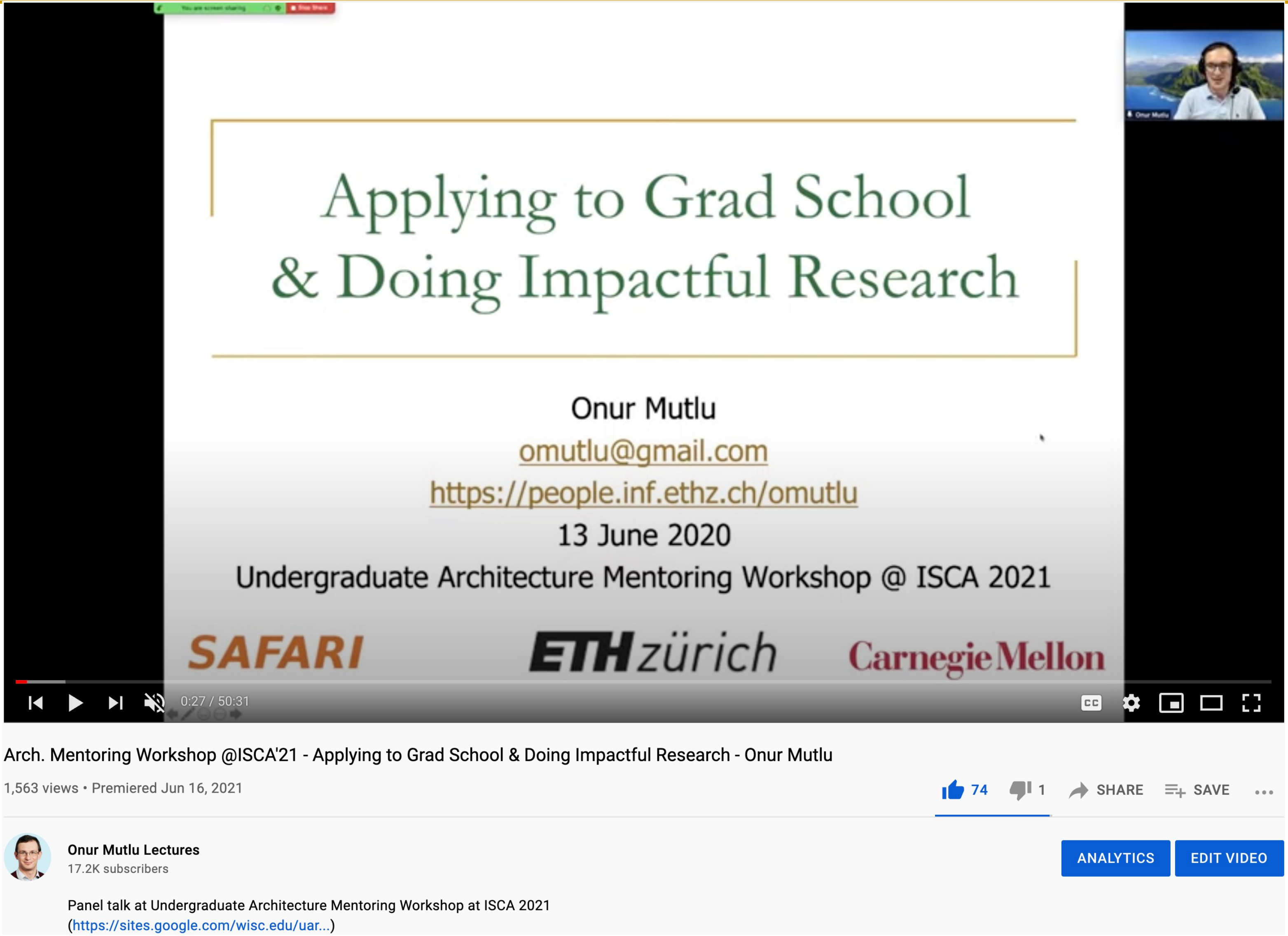
SAFARI Research Group

safari.ethz.ch

Think BIG, Aim HIGH!

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

Suggestions on Research, Education, PhD



Applying to Grad School & Doing Impactful Research

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
13 June 2020
Undergraduate Architecture Mentoring Workshop @ ISCA 2021

Arch. Mentoring Workshop @ISCA'21 - Applying to Grad School & Doing Impactful Research - Onur Mutlu

1,563 views • Premiered Jun 16, 2021

Panel talk at Undergraduate Architecture Mentoring Workshop at ISCA 2021 (https://sites.google.com/wisc.edu/uar...)

https://www.youtube.com/watch?v=83tlorht7Mc&list=PL5Q2soXY2Zi8D_5MGV6EnXEJHnV2YFBJl&index=54

Funding Acknowledgments

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

Thank you!

Referenced Papers, Talks, Artifacts

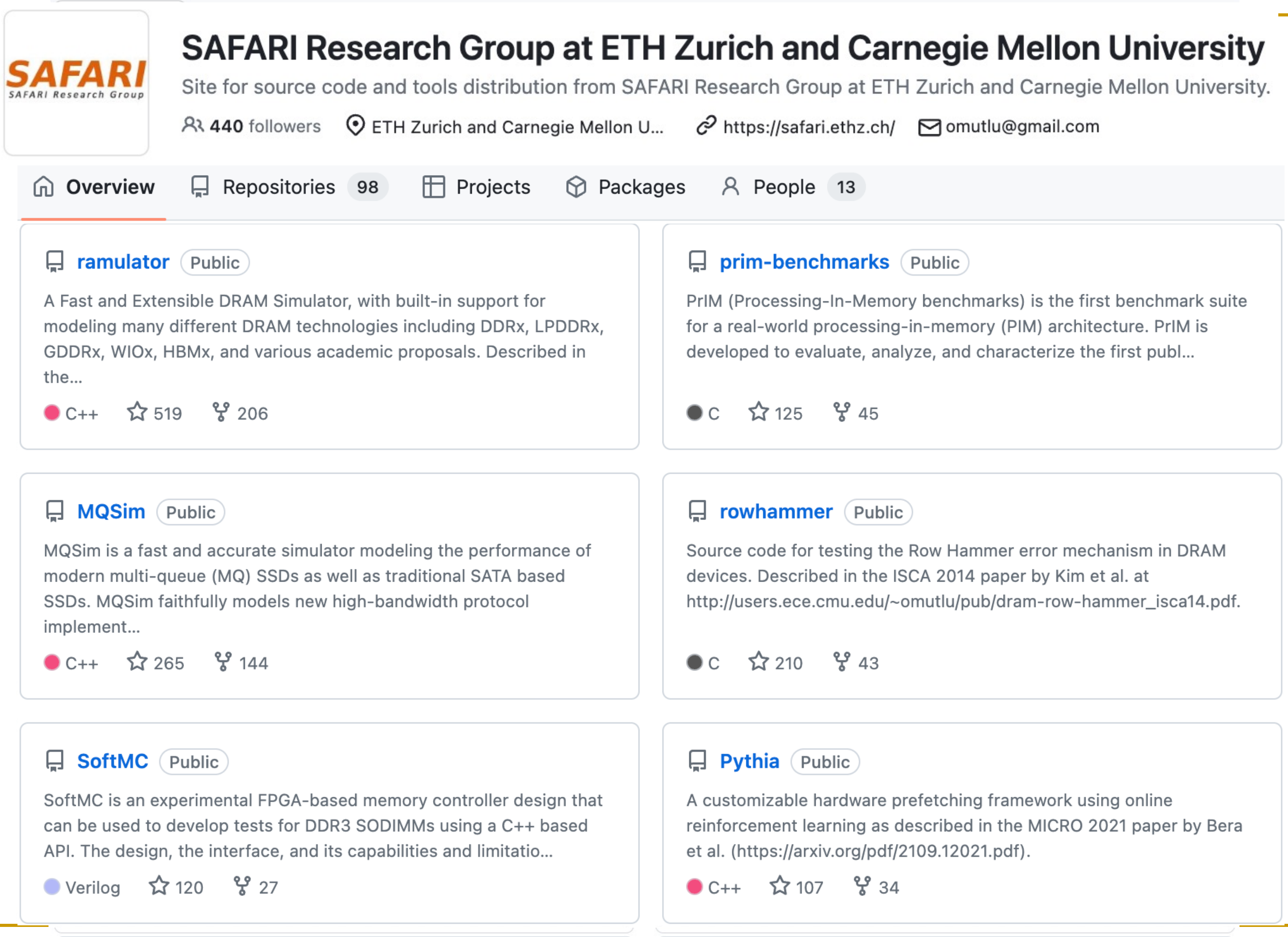
- All are available at

  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

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 Forks 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 Forks 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 Forks 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 Forks 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 Forks 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 Forks 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

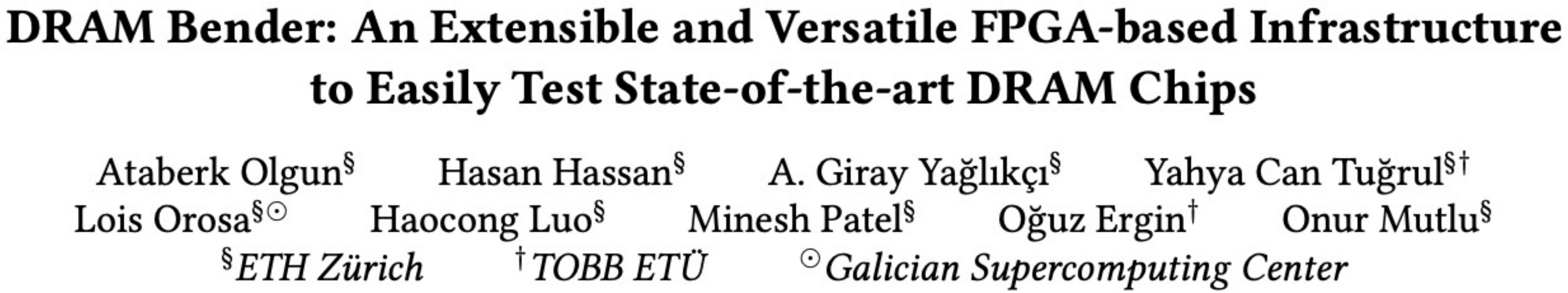
Haocong Luo, Yahya Can Tuğrul, F. Nisa Bostancı, Ataberk Olgun, A. Giray Yağlıkçı, and Onur Mutlu

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)]



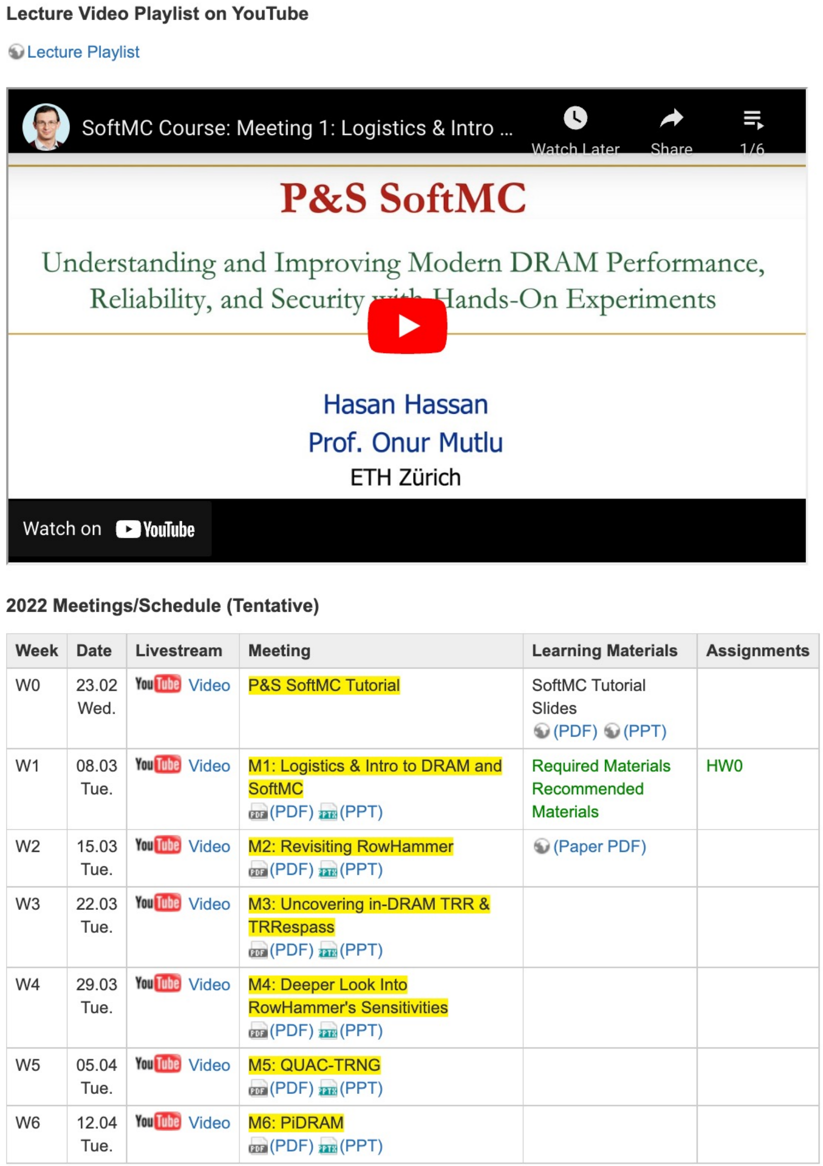
DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips

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

P&S SoftMC

Understanding and Improving Modern DRAM Performance, Reliability, and Security with Hands-On Experiments

Hasan Hassan
Prof. Onur Mutlu
ETH Zürich

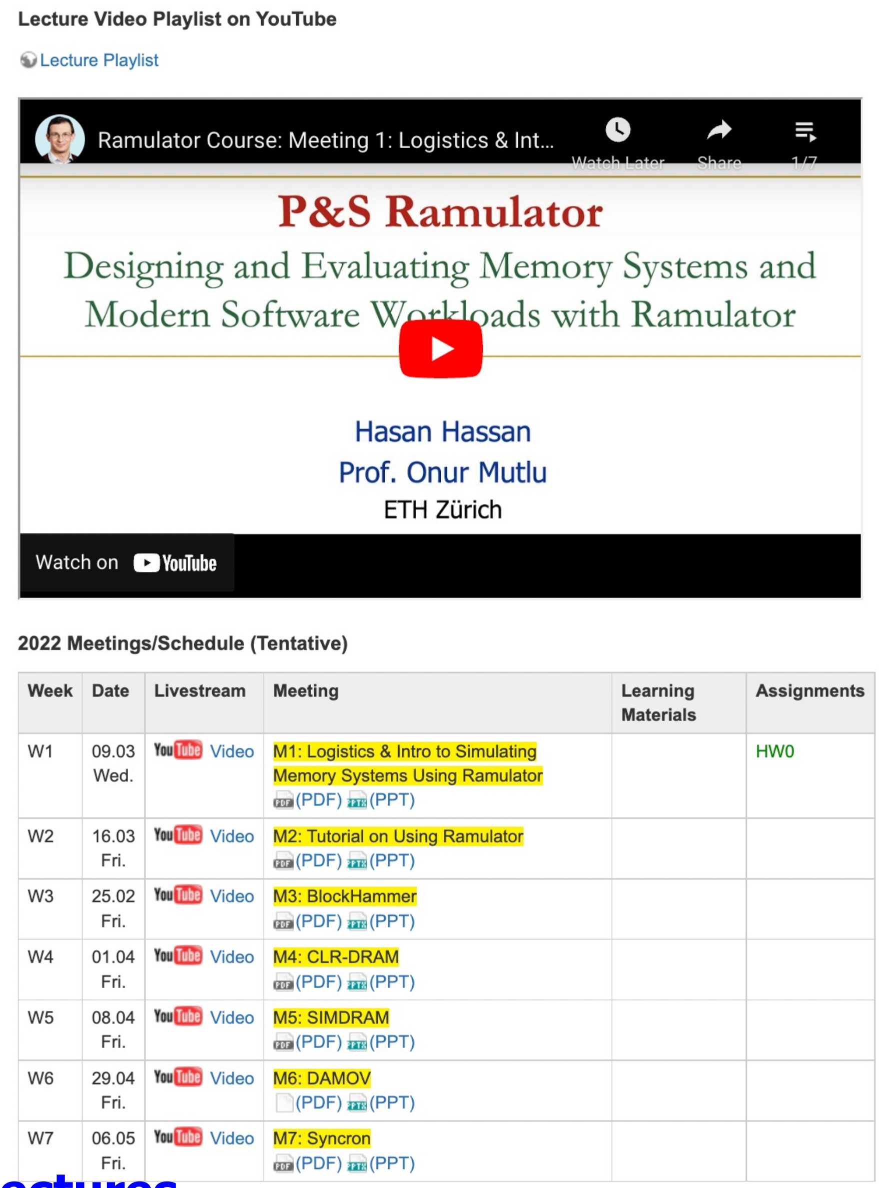
2022 Meetings/Schedule (Tentative)

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

Ramulator Course: Meeting 1: Logistics & Int...

P&S Ramulator

Designing and Evaluating Memory Systems and Modern Software Workloads with Ramulator

Hasan Hassan

Prof. Onur Mutlu

ETH Zürich

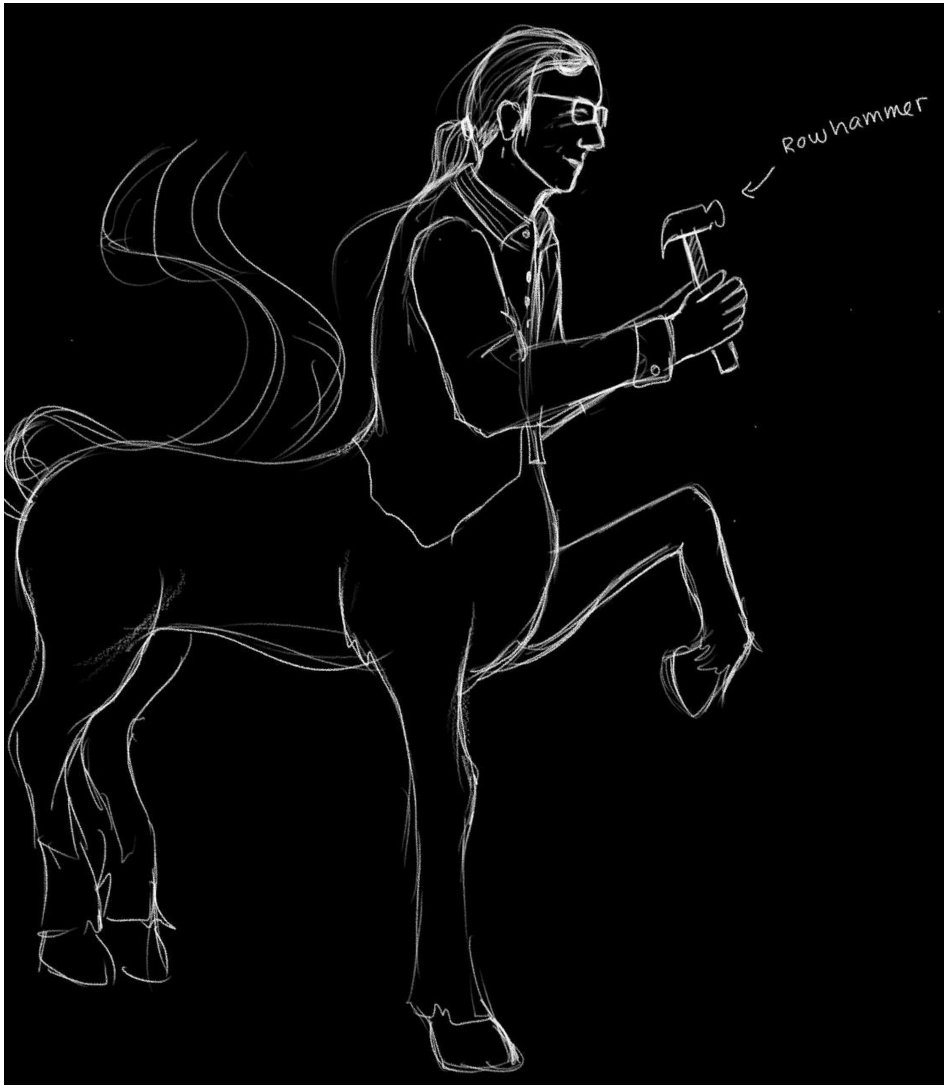
2022 Meetings/Schedule (Tentative)

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

Rowhammer

I Talk A Lot About RowHammer



Art credit: Malti Redeker (https://www.instagram.com/malti.red/)

Memory Systems and Memory-Centric Computing

Lecture 5: Memory Robustness II

Onur Mutlu

omutlu@gmail.com

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

19 July 2024

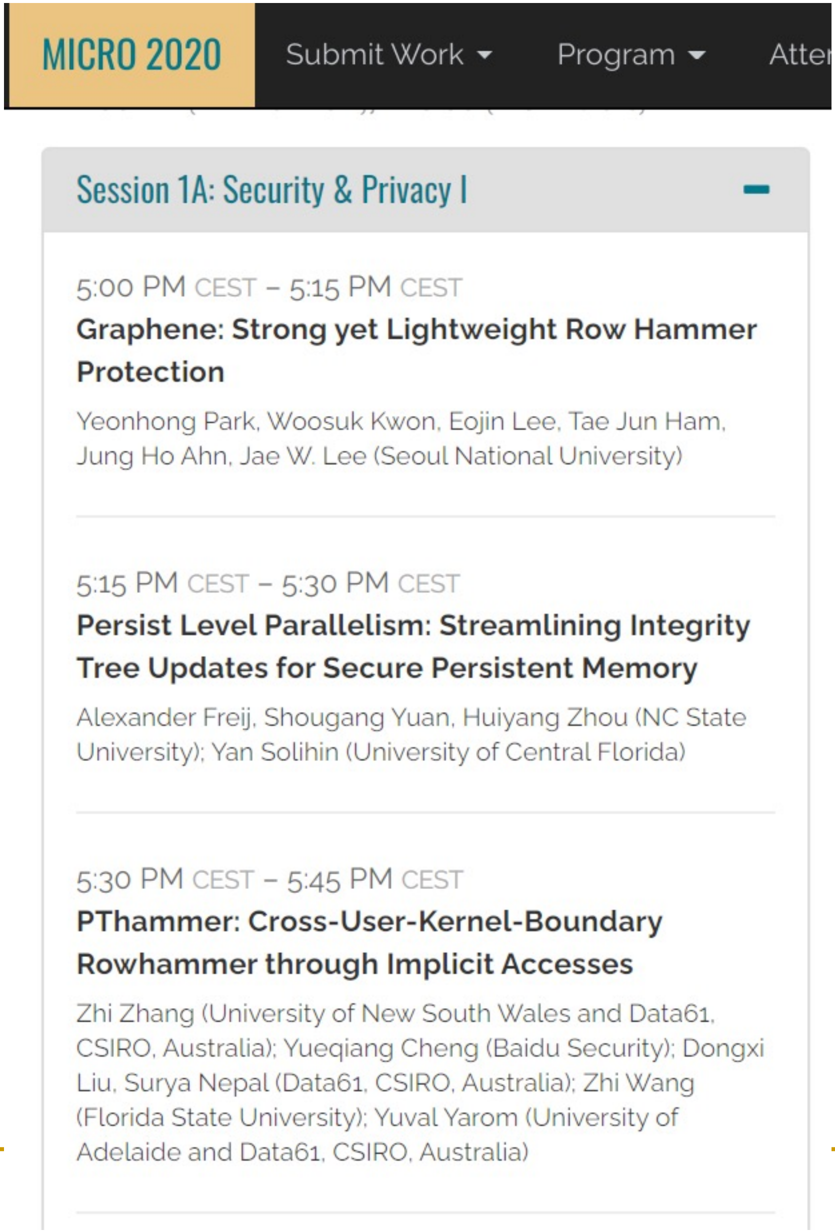
HiPEAC ACACES Summer School 2024

SAFARI

ETH zürich

More RowHammer in 2020-2024

RowHammer in 2020 (I)



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

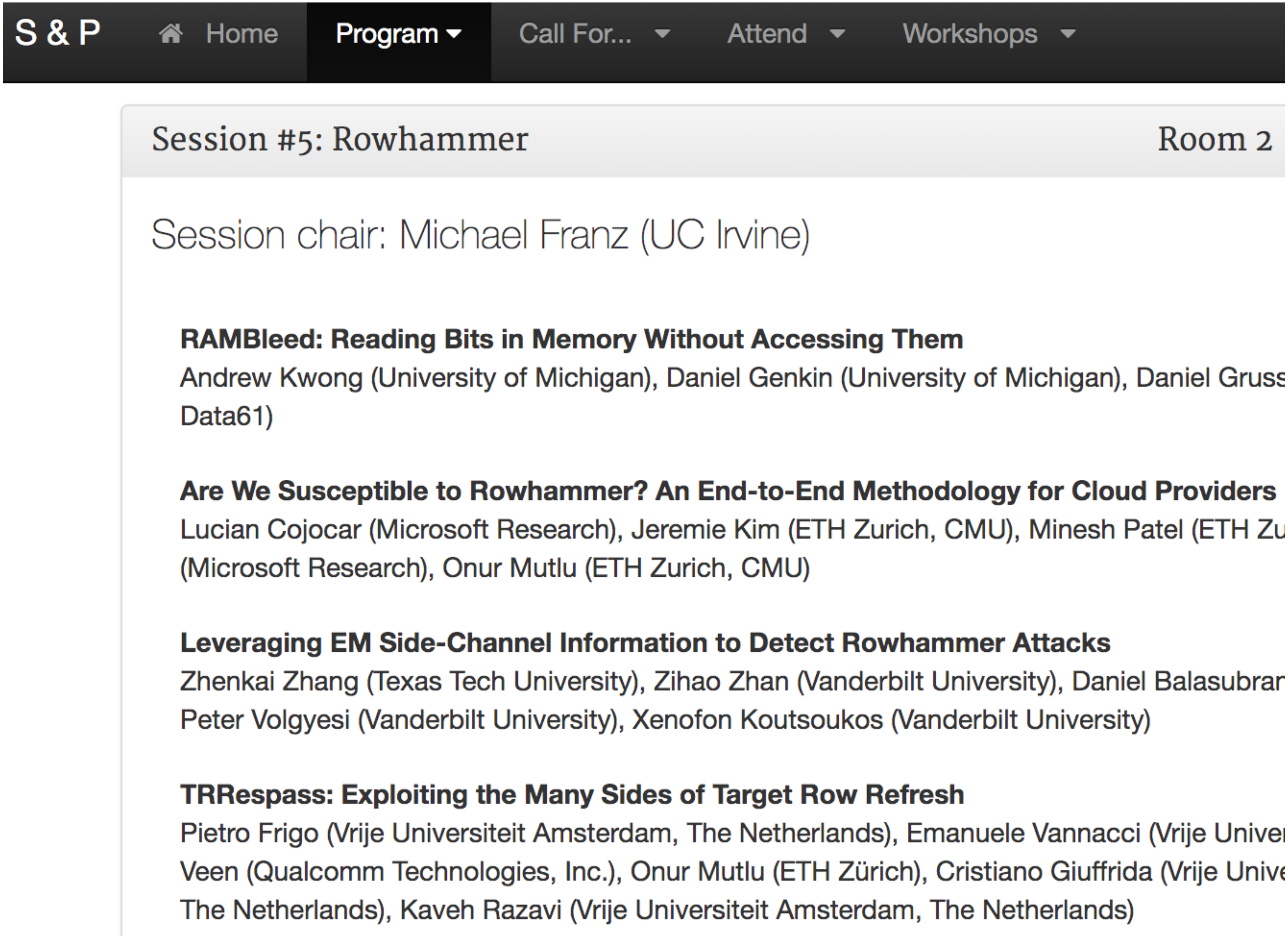
Alexander Freij, Shougang Yuan, Huiyang Zhou (NC State University); Yan Solihin (University of Central Florida)

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)



Session #5: Rowhammer Room 2

Session chair: Michael Franz (UC Irvine)

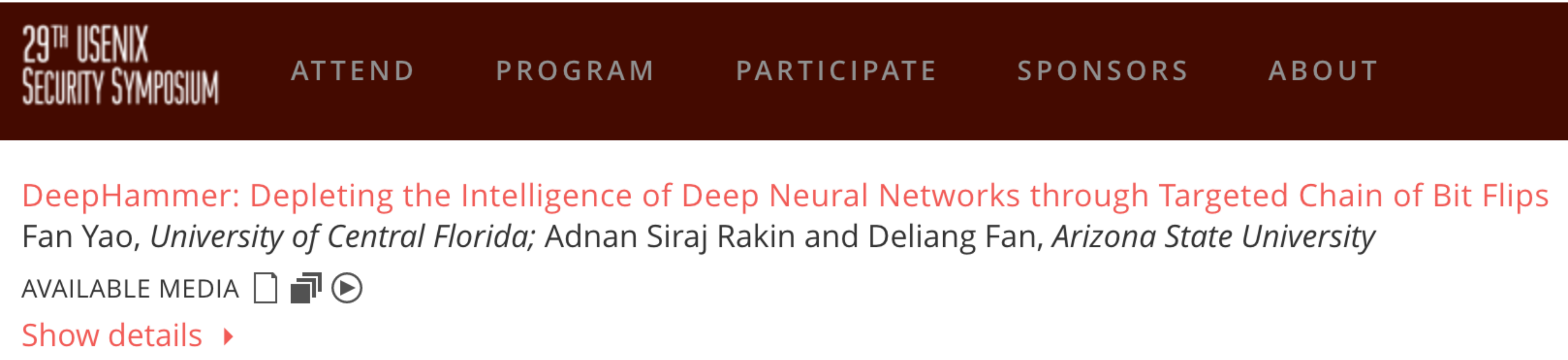
RAMBleed: Reading Bits in Memory Without Accessing Them
Andrew Kwong (University of Michigan), Daniel Genkin (University of Michigan), Daniel Gruss
Data61)

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)

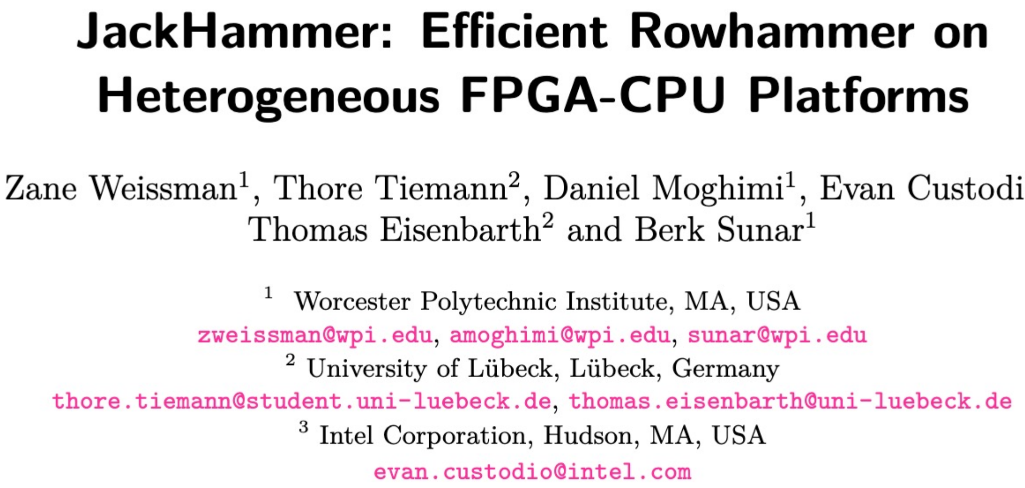


DeepHammer: Depleting the Intelligence of Deep Neural Networks through Targeted Chain of Bit Flips

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

RowHammer in 2020 (IV)

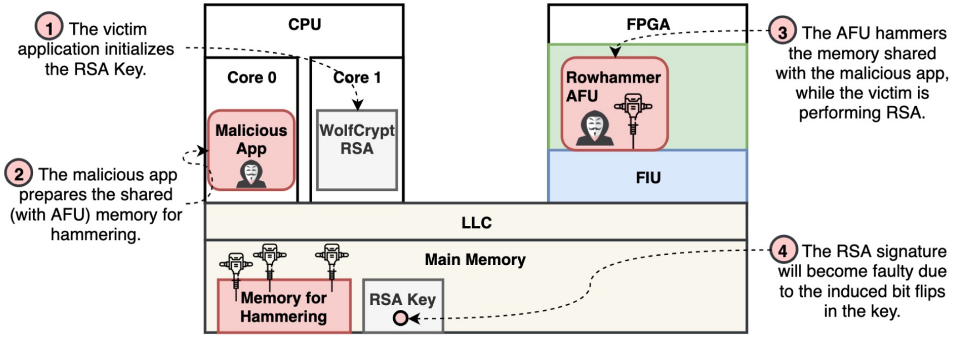
- CHES 2020



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


1 The victim application initializes the RSA Key.
2 The malicious app prepares the shared (with AFU) memory for hammering.
CPU
Core 0
Core 1
Malicious App
WolfCrypt RSA
FPGA
Rowhammer AFU
FIU
3 The AFU hammers the memory shared with the malicious app, while the victim is performing RSA.
LLC
Main Memory
Memory for Hammering
RSA Key
4 The RSA signature will become faulty due to the induced bit flips in the key.

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

RowHammer in 2021 (I)

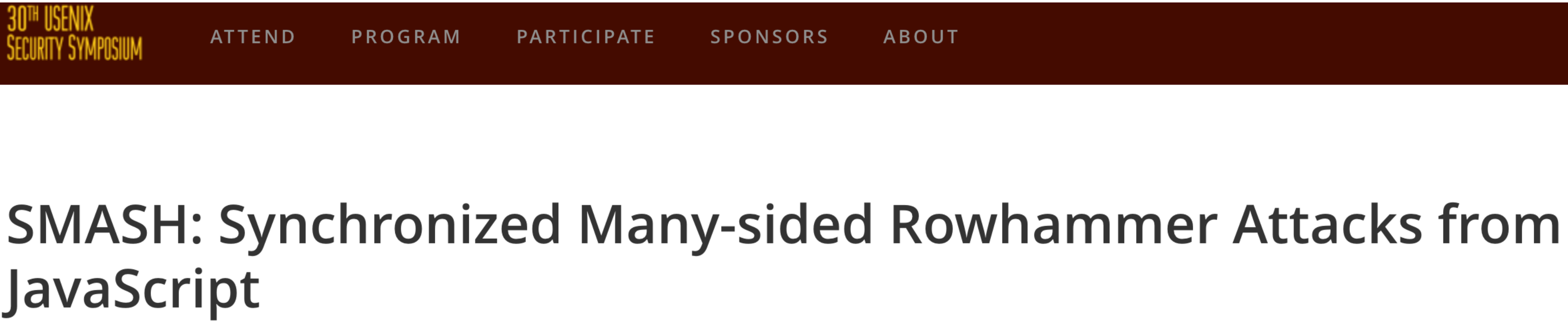
HotOS XVIII

The 18th Workshop on Hot Topics in Operating Systems

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

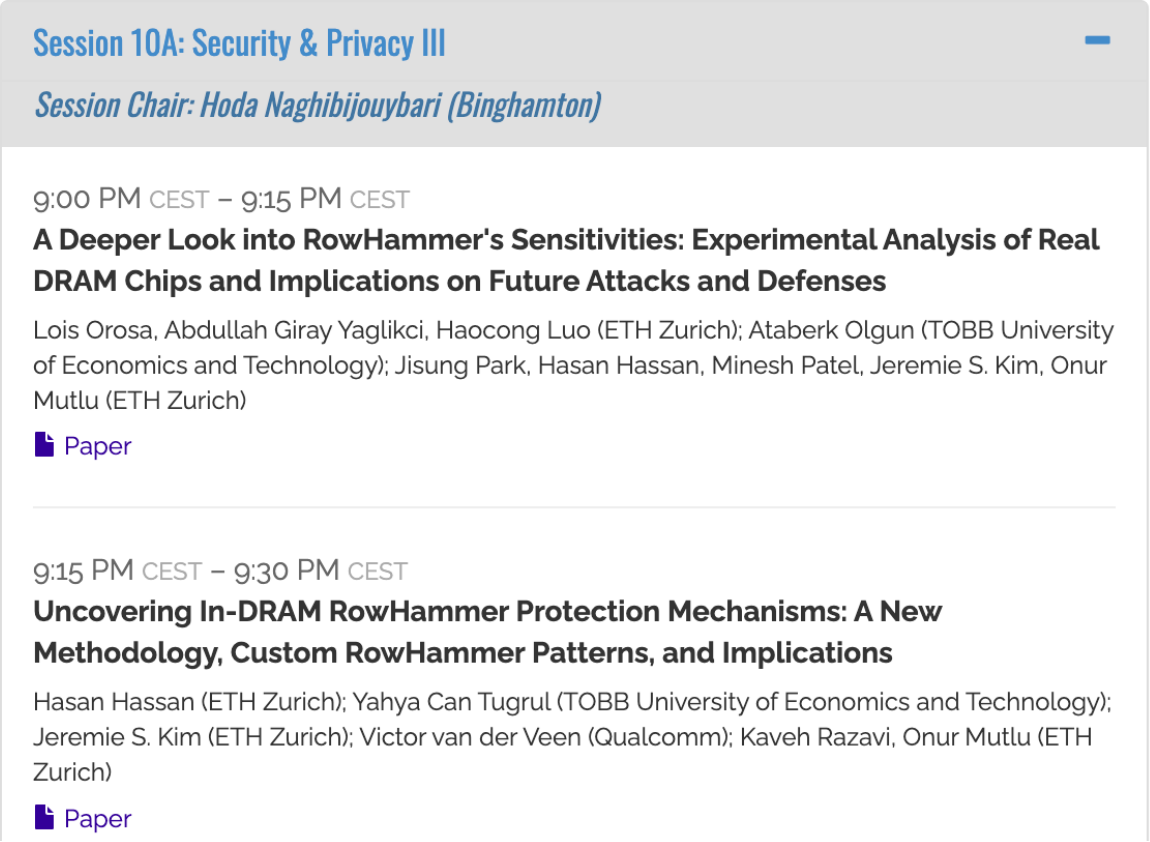
Stop! Hammer Time: Rethinking Our Approach to Rowhammer Mitigations

RowHammer in 2021 (II)



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

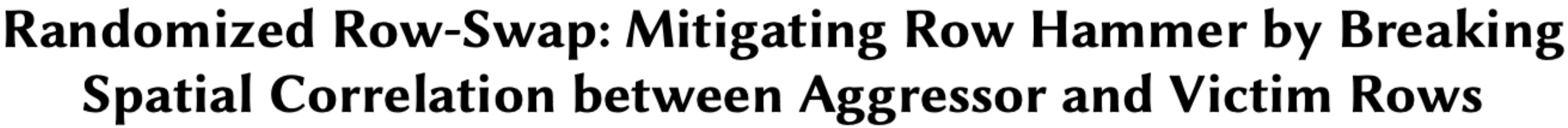
BLACKSMITH: Scalable Rowhammering in the Frequency Domain

SpecHammer: Combining Spectre and Rowhammer for New Speculative Attacks

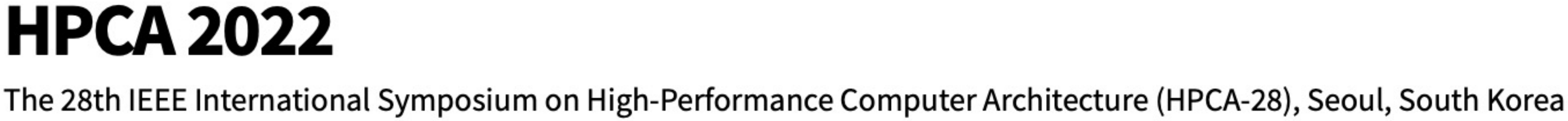
PROTRR: Principled yet Optimal In-DRAM Target Row Refresh

DeepSteal: Advanced Model Extractions Leveraging Efficient Weight Stealing in Memories

RowHammer in 2022 (II)

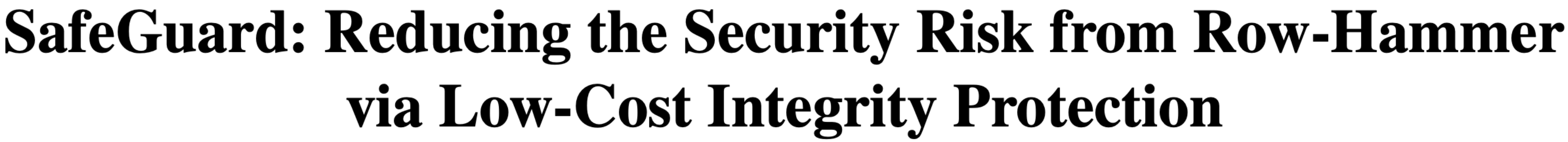

Randomized Row-Swap: Mitigating Row Hammer by Breaking Spatial Correlation between Aggressor and Victim Rows

RowHammer in 2022 (III)

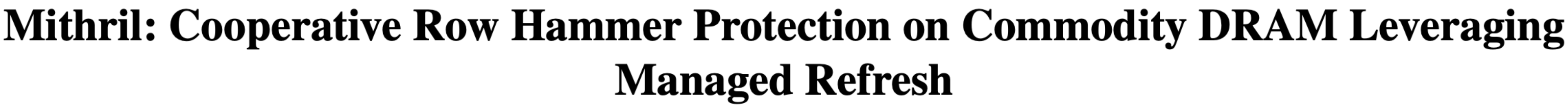


HPCA 2022

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



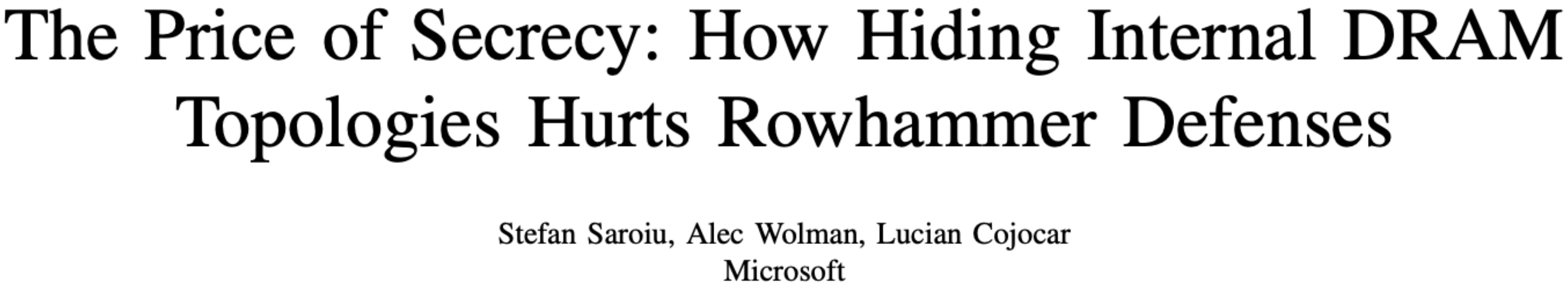
Mithril: Cooperative Row Hammer Protection on Commodity DRAM Leveraging Managed Refresh

RowHammer in 2022 (IV)

IRPS

International Reliability Physics Symposium

IRPS 2022



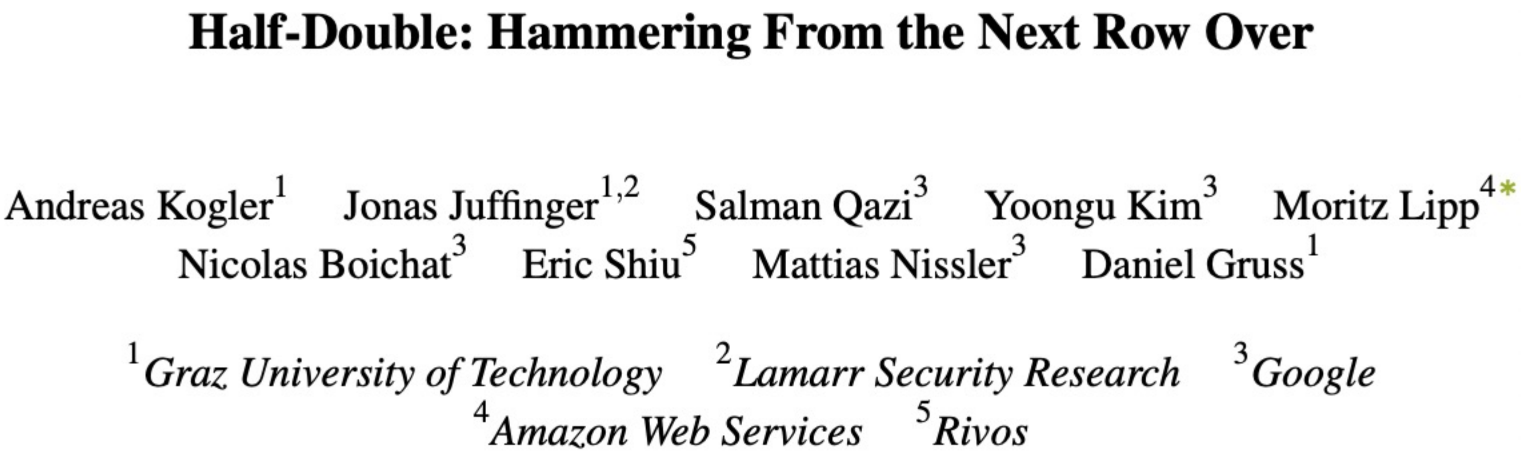
The Price of Secrecy: How Hiding Internal DRAM Topologies Hurts Rowhammer Defenses

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)

ACM CCS 2022

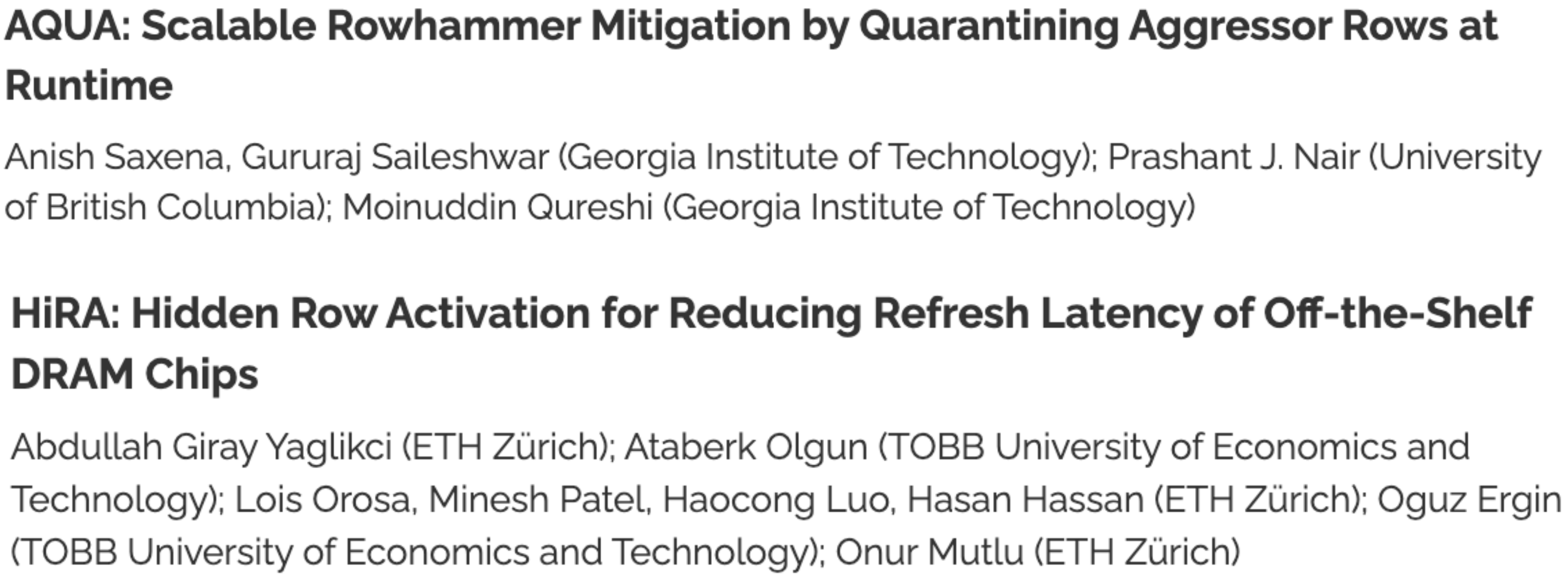
November 7-11, 2022

Los Angeles, U.S.A.

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

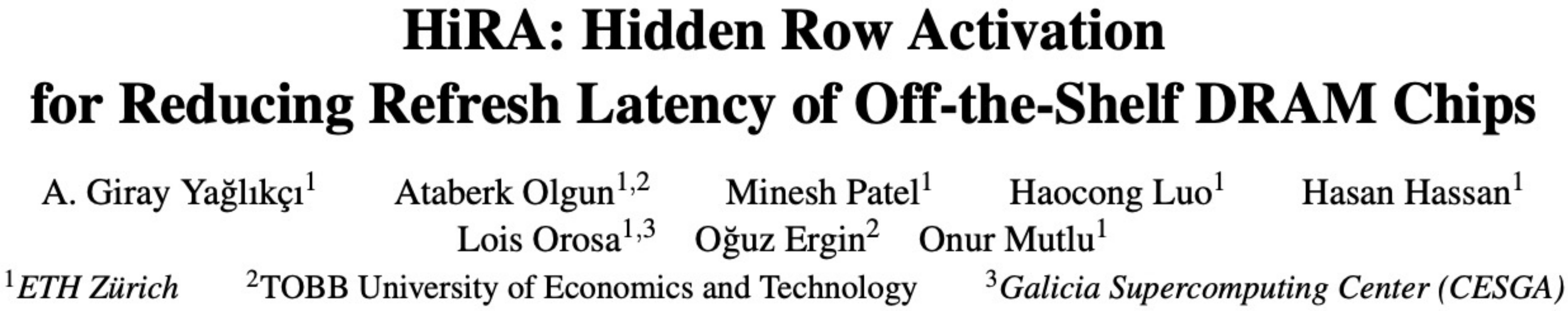
Anish Saxena, Gururaj Saileshwar (Georgia Institute of Technology); Prashant J. Nair (University of British Columbia); Moinuddin Qureshi (Georgia Institute of Technology)

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]



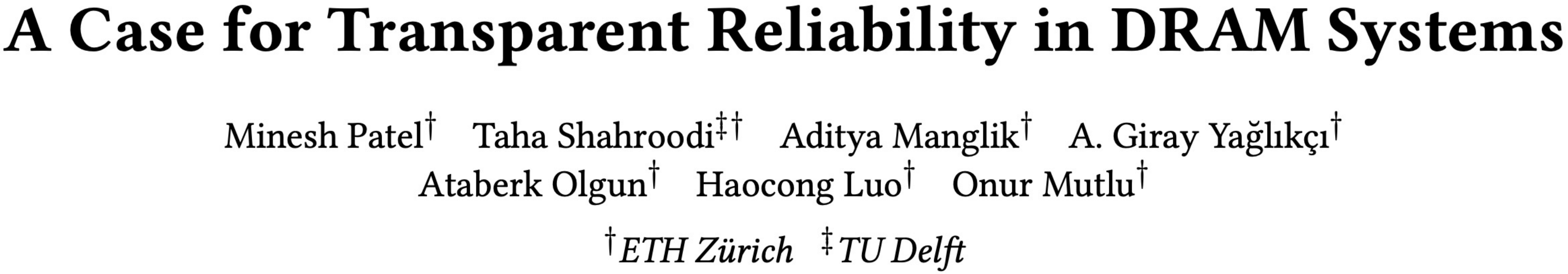
HiRA: Hidden Row Activation for Reducing Refresh Latency of Off-the-Shelf DRAM Chips

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)



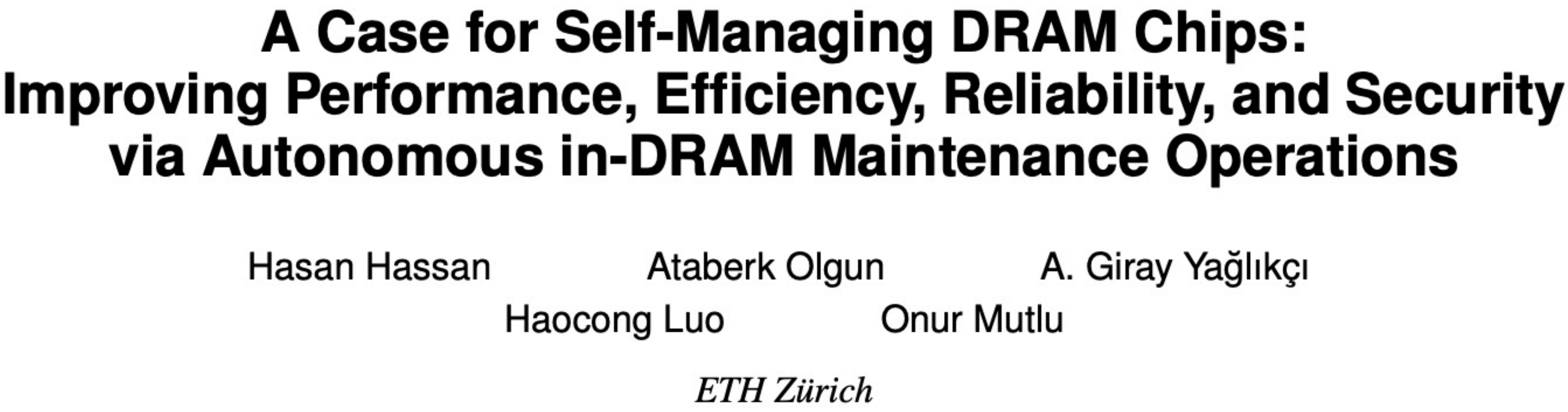
A Case for Transparent Reliability in DRAM Systems

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)



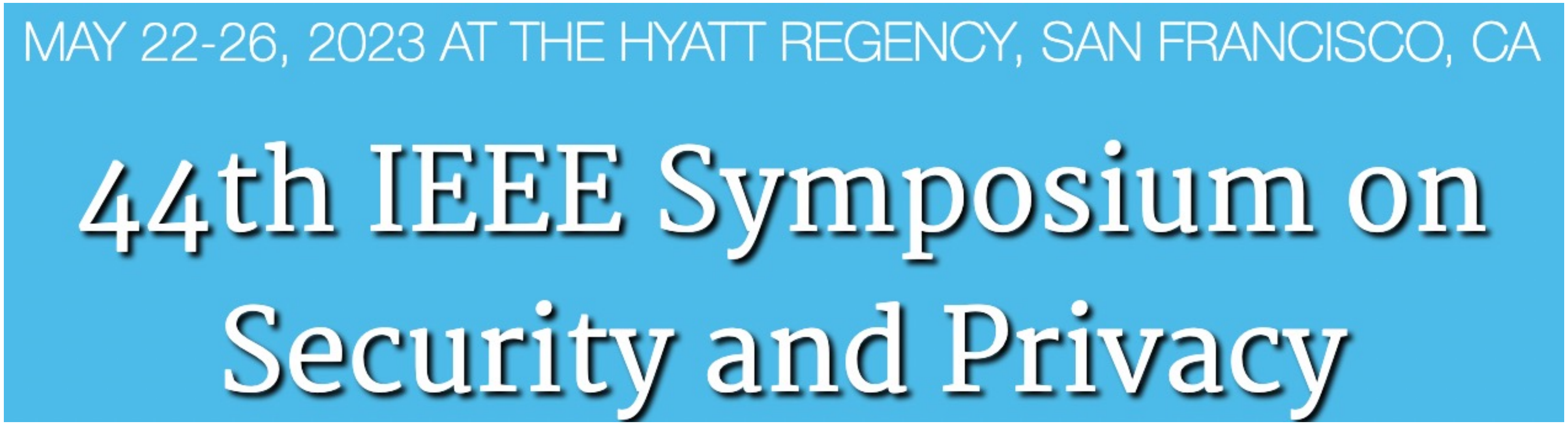
A Case for Self-Managing DRAM Chips: Improving Performance, Efficiency, Reliability, and Security via Autonomous in-DRAM Maintenance Operations

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

ETH Zürich

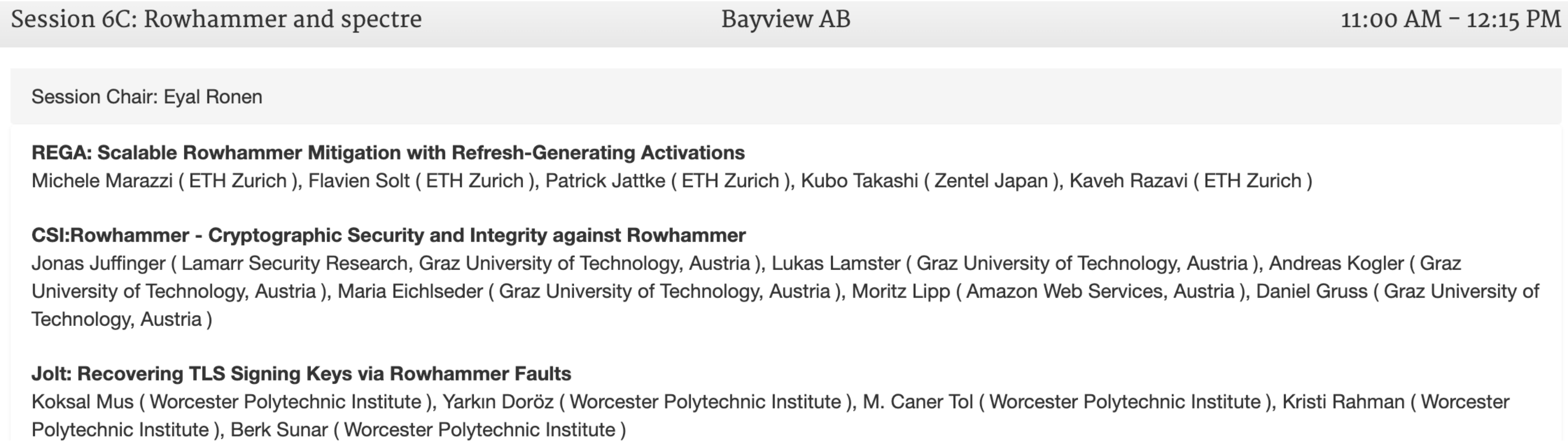
https://arxiv.org/pdf/2207.13358.pdf

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

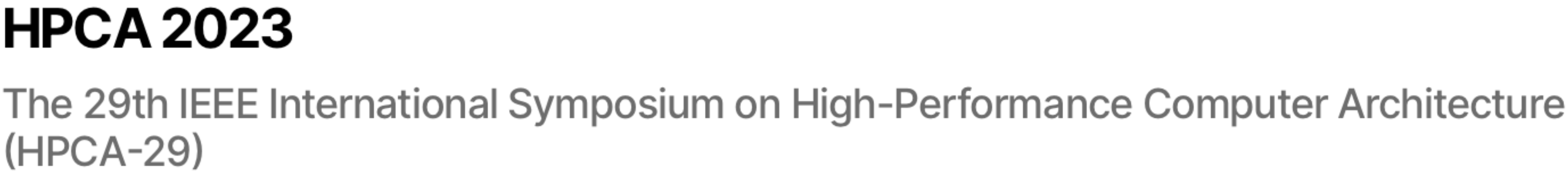
Session Chair: Eyal Ronen

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)

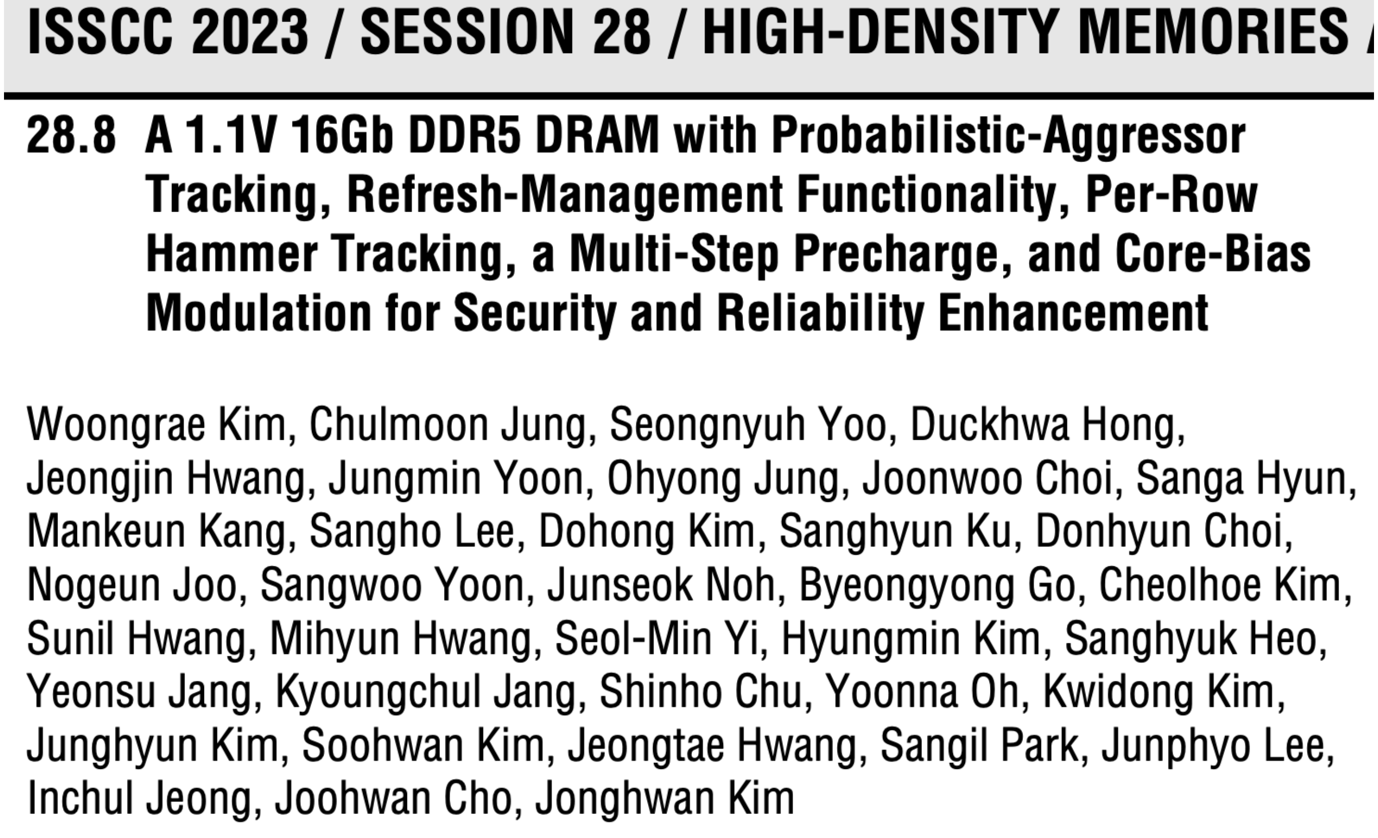
Scalable and Secure Row-Swap: Efficient and Safe Row Hammer Mitigation in Memory Systems

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 /

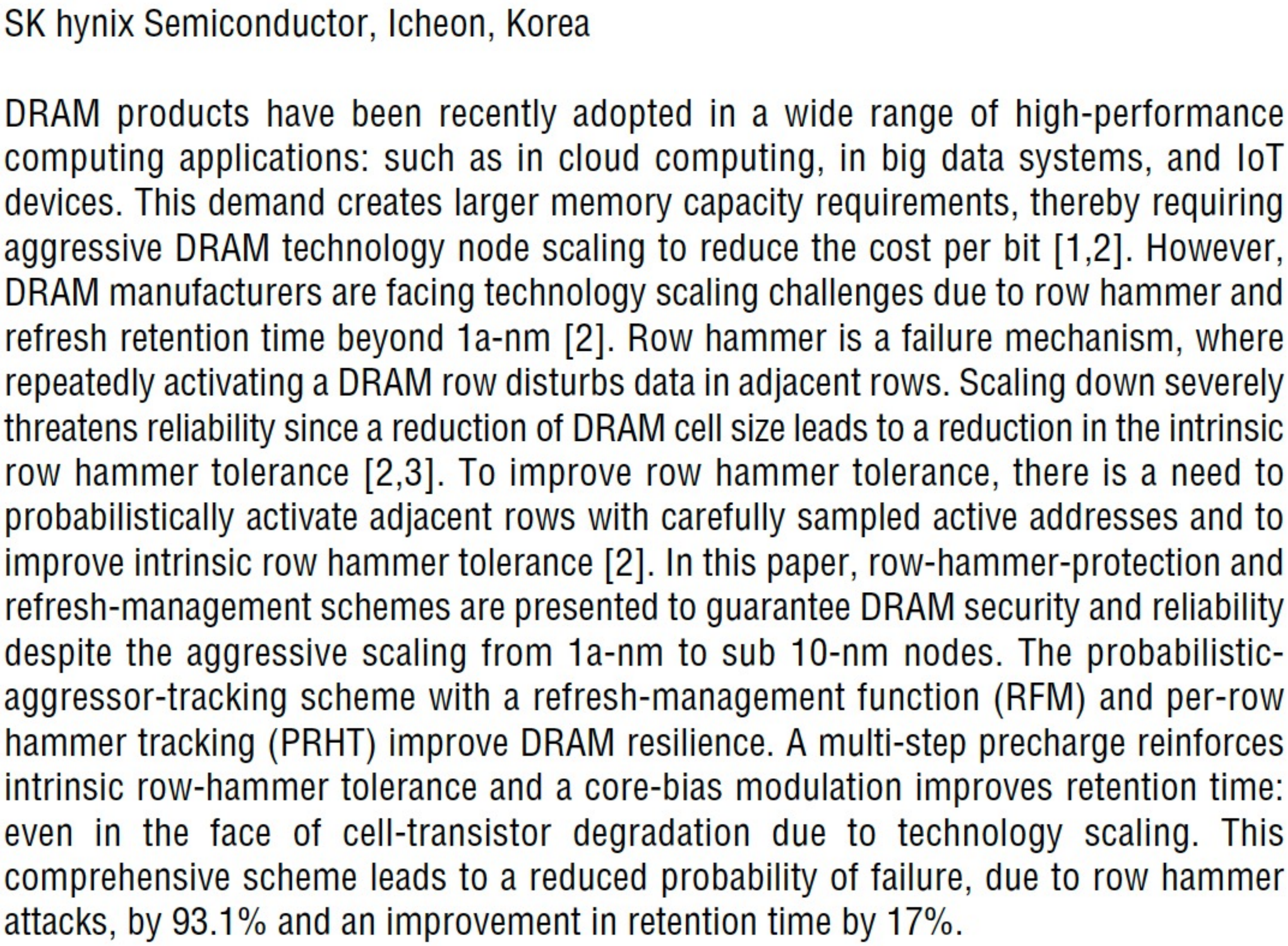
28.8 A 1.1V 16Gb DDR5 DRAM with Probabilistic-Aggressor Tracking, Refresh-Management Functionality, Per-Row Hammer Tracking, a Multi-Step Precharge, and Core-Bias Modulation for Security and Reliability Enhancement

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

ISSCC 2023
2023 International Solid-State Circuits Conference
February 19-23, 2023 San Francisco, CA

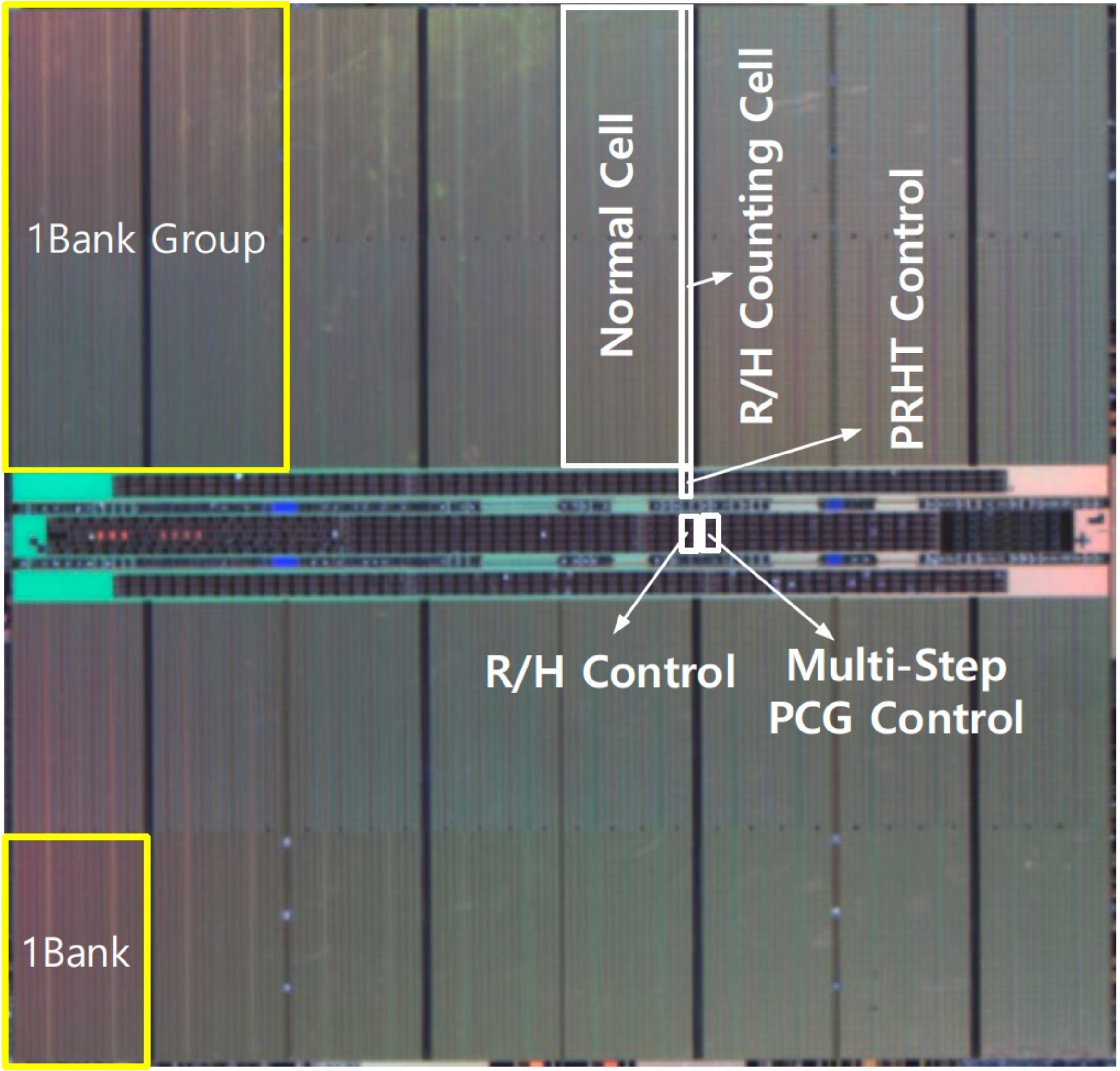
Industry's RowHammer Solutions (I)

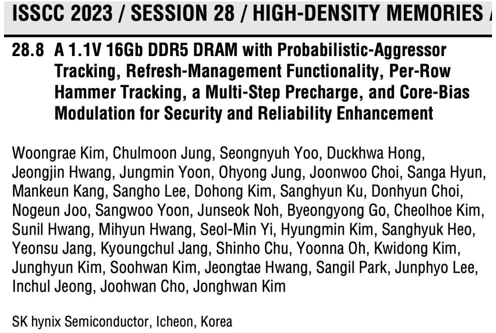


SK hynix Semiconductor, Icheon, Korea

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)



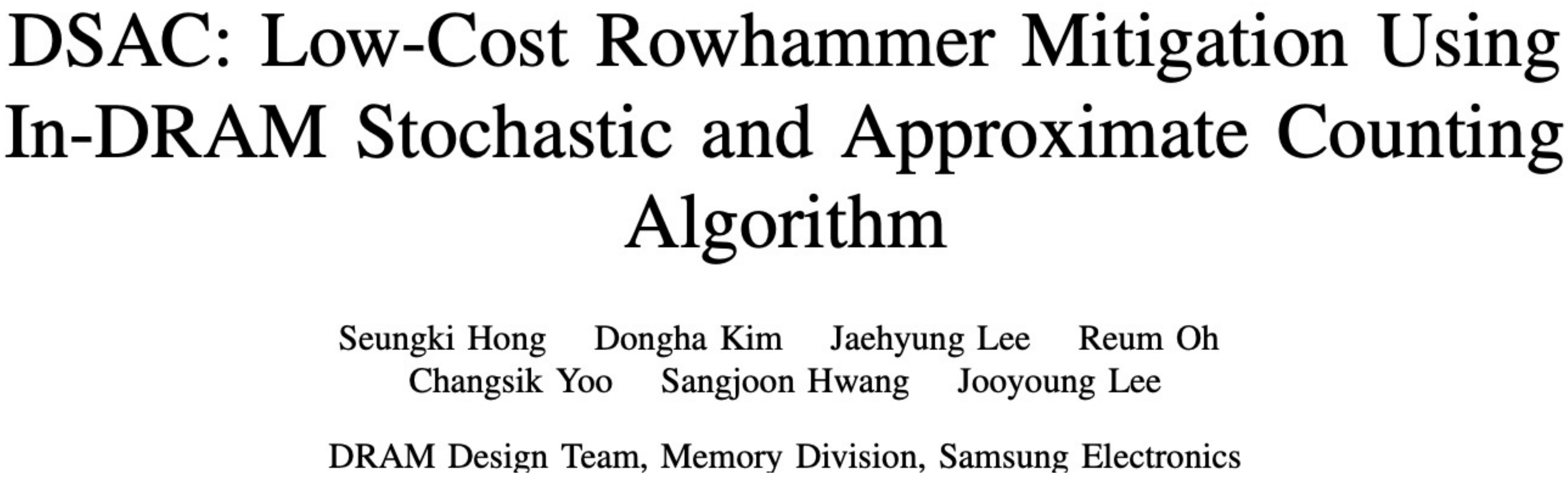

ISSCC 2023 / SESSION 28 / HIGH-DENSITY MEMORIES /

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

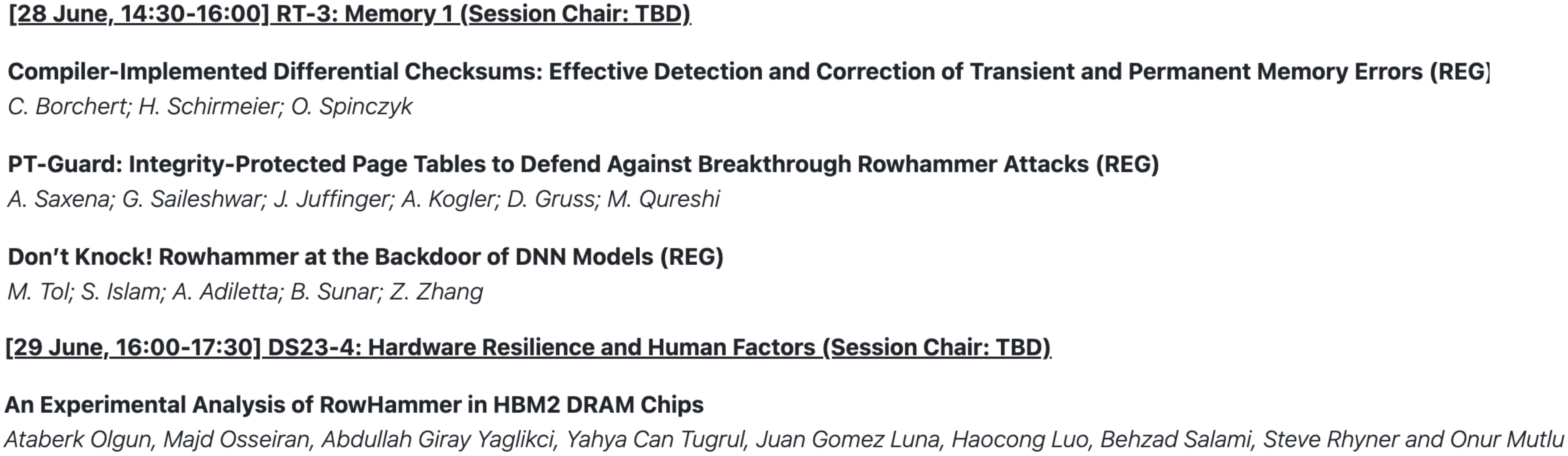
https://arxiv.org/pdf/2302.03591v1.pdf

RowHammer in 2023 (V)

DSN 2023

Welcome to DSN 2023

Porto, Portugal, June 27-30, 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

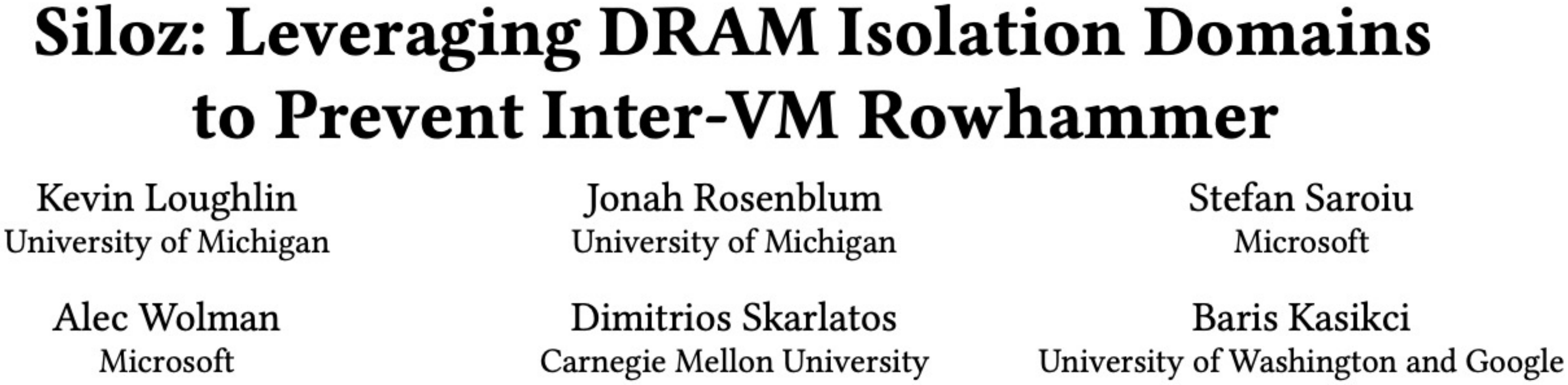
RowHammer in 2023 (VI)

- SOSP 2023

SOSP 2023

The 29th ACM Symposium on Operating Systems Principles

October 23-26, 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

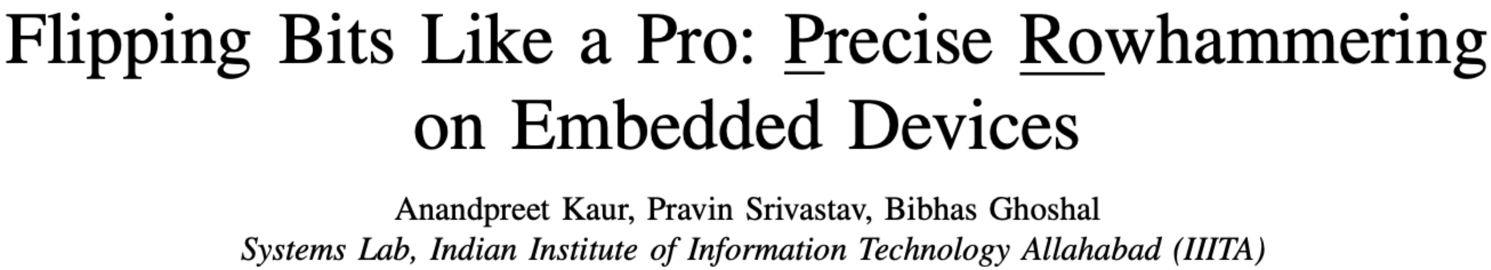
NoHammer: Preventing Row Hammer with Last-Level Cache Management

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

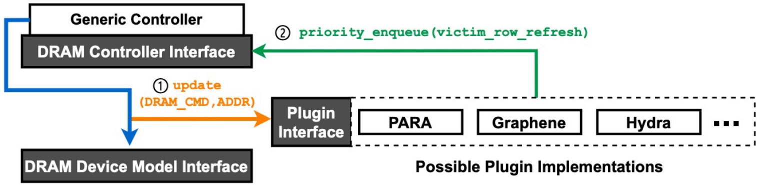
Anandpreet Kaur, Pravin Srivastav, Bibhas Ghoshal
Systems Lab, Indian Institute of Information Technology Allahabad (IIITA)

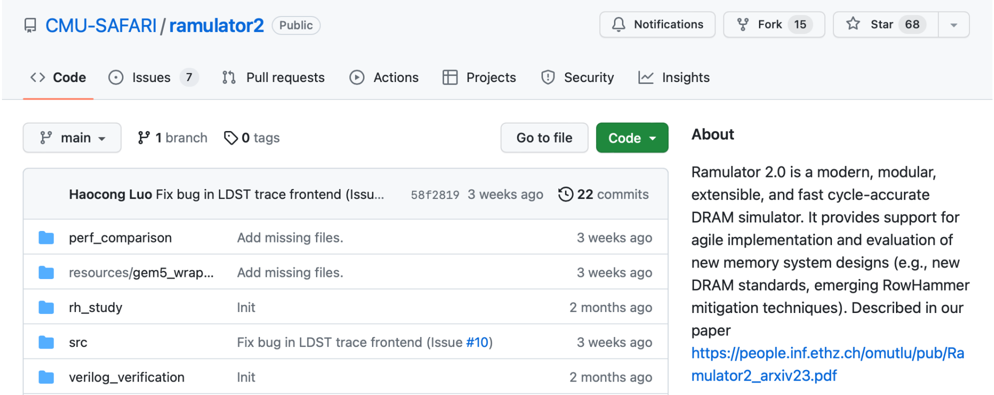
Ramulator 2.0

"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]


Generic Controller
DRAM Controller Interface
② priority_enqueue(victim_row_refresh)
① update (DRAM_CMD,ADDR)
Plugin Interface
PARA
Graphene
Hydra
...
DRAM Device Model Interface
Possible Plugin Implementations



Haocong Luo Fix bug in LDST trace frontend (Issu... 58f2819 3 weeks ago 22 commits

perf_comparison	Add missing files.	3 weeks ago
resources/gem5_wrap...	Add missing files.	3 weeks ago
rh_study	Init	2 months ago
src	Fix bug in LDST trace frontend (Issue #10)	3 weeks ago
verilog_verification	Init	2 months ago

About

Ramulator 2.0 is a modern, modular, extensible, and fast cycle-accurate DRAM simulator. It provides support for agile implementation and evaluation of new memory system designs (e.g., new DRAM standards, emerging RowHammer mitigation techniques). Described in our paper https://people.inf.ethz.ch/omutlu/pub/Ramulator2_arxiv23.pdf

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

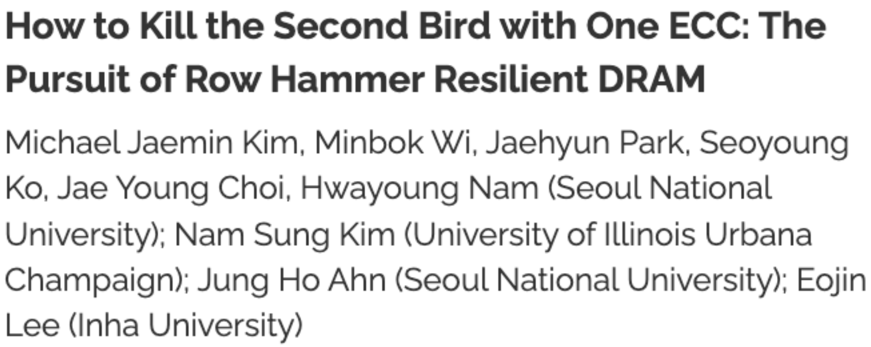
Eric Linstadt
Rambus Labs
Rambus Inc.
San Jose, CA
elinstadt@rambus.com

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)

Google's Original RowHammer Attack

The following slides are from Mark Seaborn and Thomas Dullien's BlackHat 2015 talk

https://www.blackhat.com/docs/us-15/materials/us-15-Seaborn-Exploiting-The-DRAM-Rowhammer-Bug-To-Gain-Kernel-Privileges.pdf

https://www.youtube.com/watch?v=0U7511Fb4to

Kernel exploit

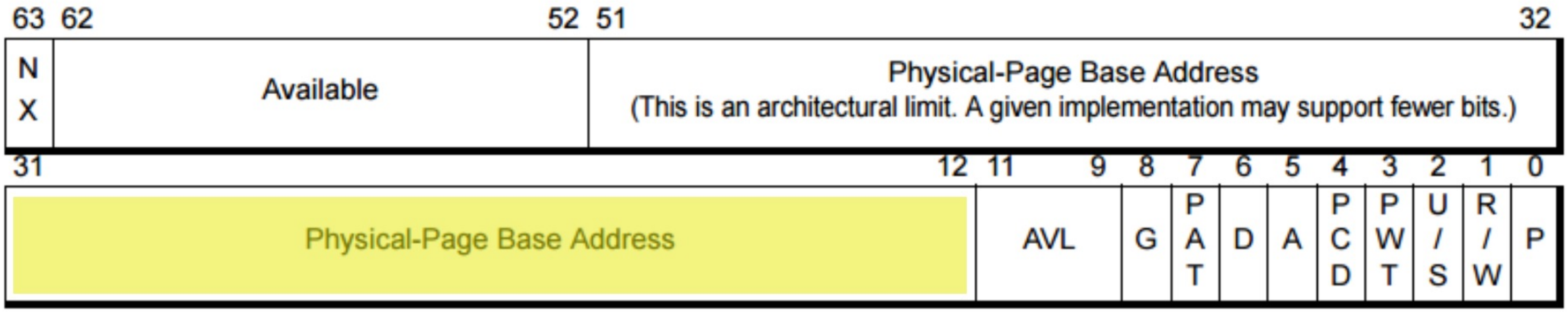
- x86 page tables entries (PTEs) are **dense and trusted**
  - They control access to physical memory
  - A bit flip in a PTE's physical page number can give a process access to a different physical page
- Aim of exploit: Get access to a page table
  - Gives access to all of physical memory
- Maximise chances that a bit flip is useful:
  - Spray physical memory with page tables
  - Check for useful, repeatable bit flip first

This slide is from Mark Seaborn and Thomas Dullien's BlackHat 2015 talk

https://www.blackhat.com/docs/us-15/materials/us-15-Seaborn-Exploiting-The-DRAM-Rowhammer-Bug-To-Gain-Kernel-Privileges.pdf

x86-64 Page Table Entries (PTEs)

- Page table is a 4k page containing array of 512 PTEs
- Each PTE is 64 bits, containing:



63	62 52	51 32
NX	Available	Physical-Page Base Address (This is an architectural limit. A given implementation may support fewer bits.)

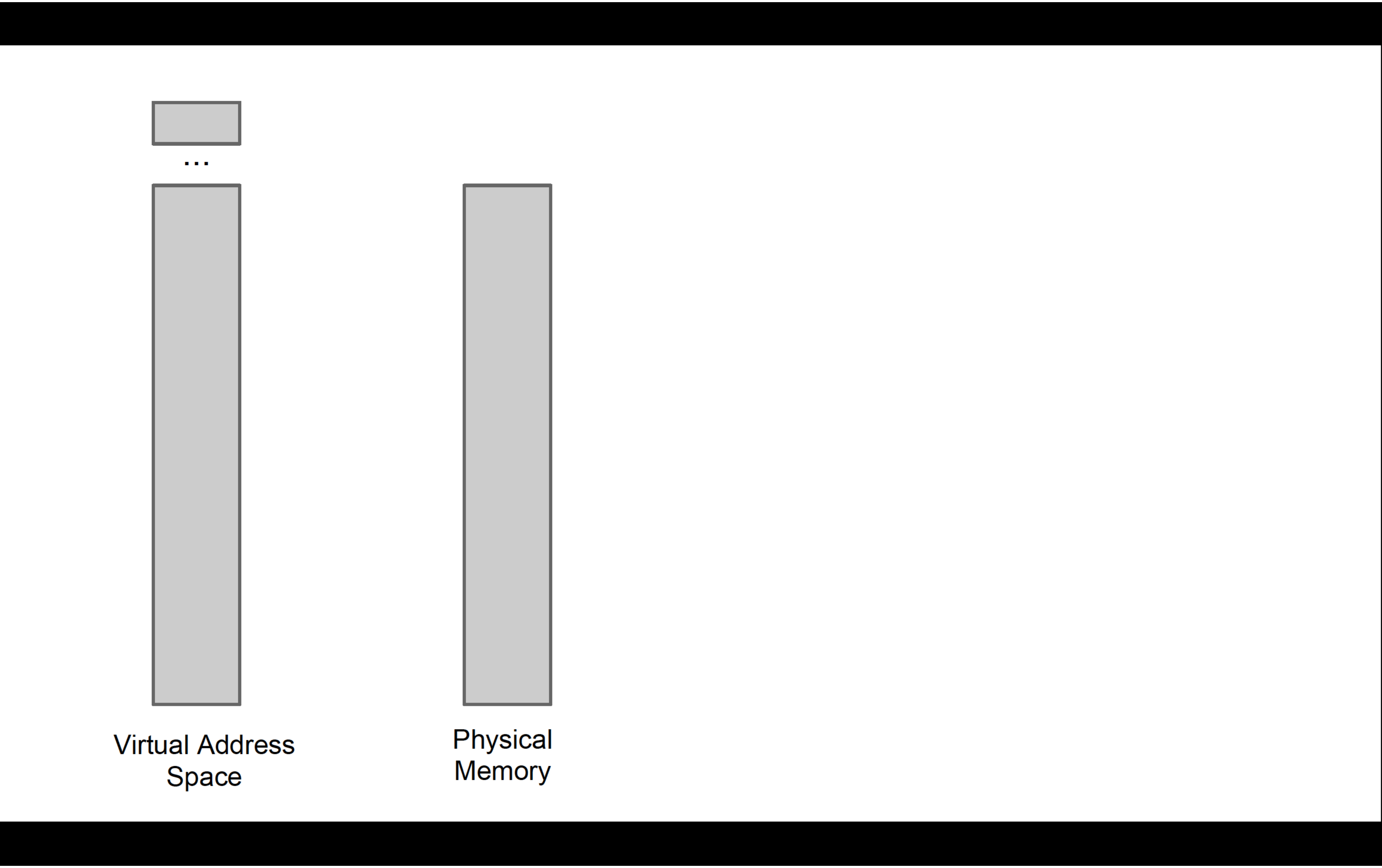
31 12	11 9	8	7	6	5	4	3	2	1	0
Physical-Page Base Address	AVL	G	PAT	D	A	PCD	PWT	U/S	R/W	P

Figure 5-21. 4-Kbyte PTE—Long Mode

- Could flip:
    - "Writable" permission bit (RW): 1 bit → 2% chance
    - Physical page number: 20 bits on 4GB system → 31% chance

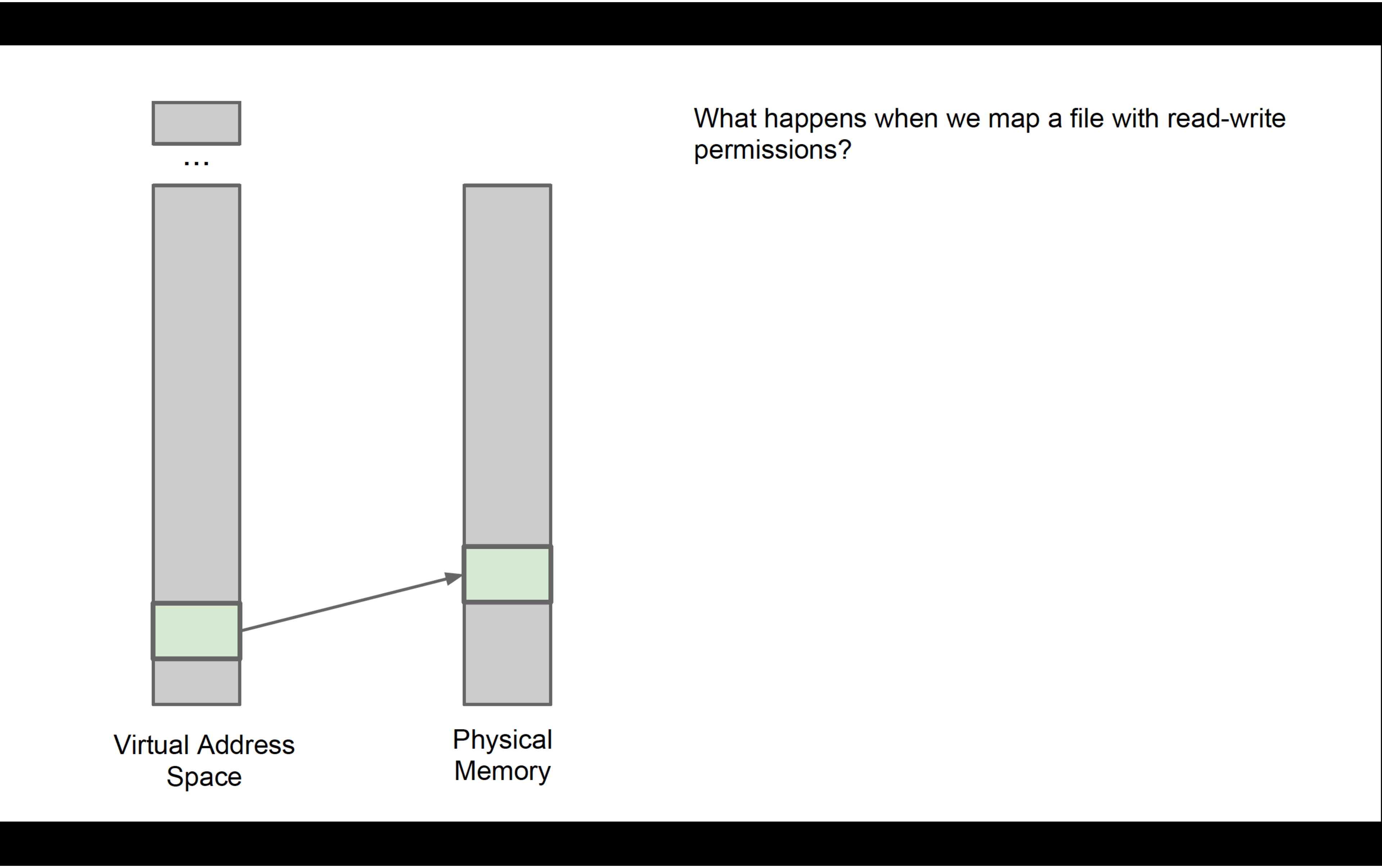
This slide is from Mark Seaborn and Thomas Dullien's BlackHat 2015 talk

https://www.blackhat.com/docs/us-15/materials/us-15-Seaborn-Exploiting-The-DRAM-Rowhammer-Bug-To-Gain-Kernel-Privileges.pdf



This slide is from Mark Seaborn and Thomas Dullien’s BlackHat 2015 talk

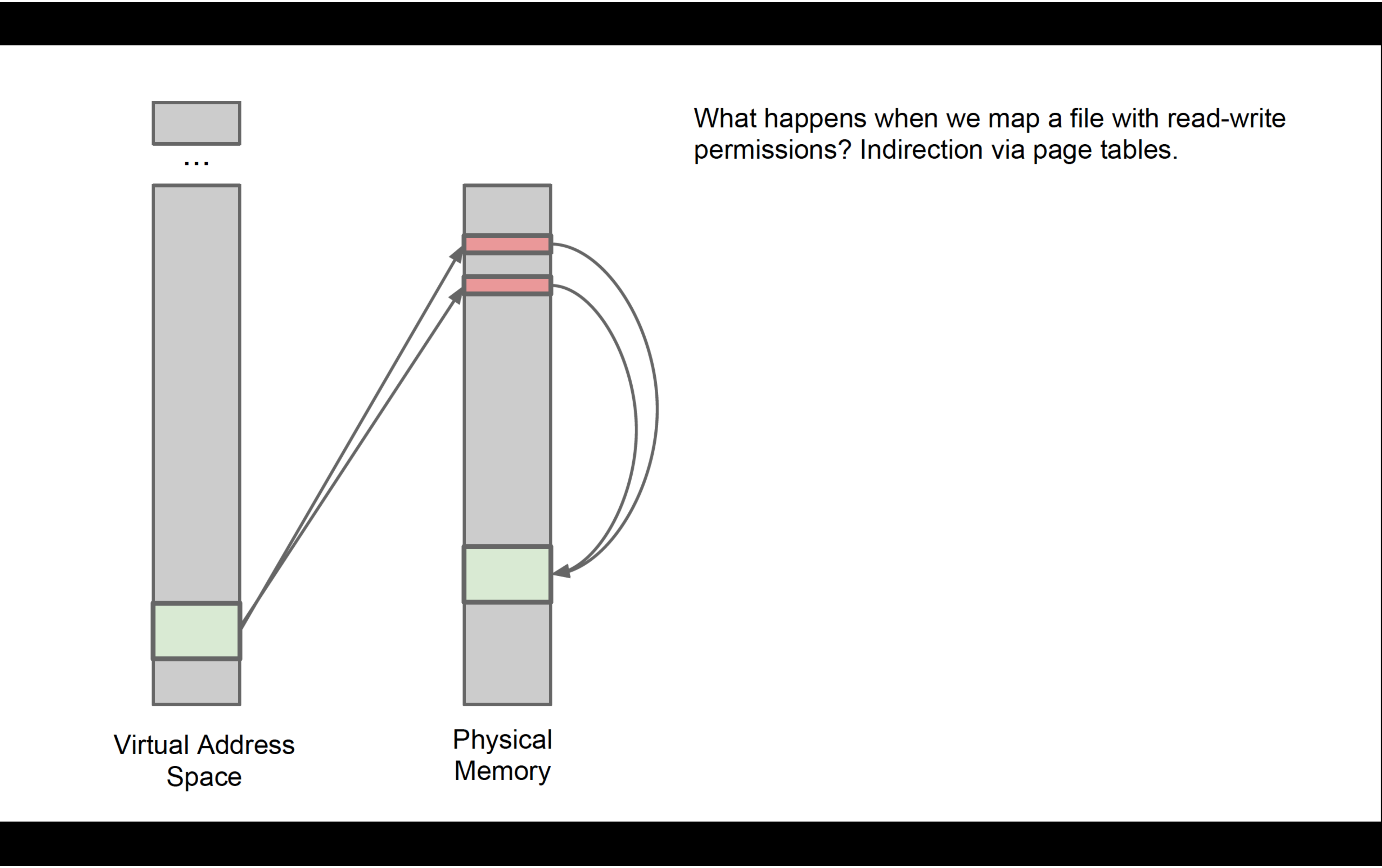
https://www.blackhat.com/docs/us-15/materials/us-15-Seaborn-Exploiting-The-DRAM-Rowhammer-Bug-To-Gain-Kernel-Privileges.pdf



What happens when we map a file with read-write permissions?

This slide is from Mark Seaborn and Thomas Dullien's BlackHat 2015 talk

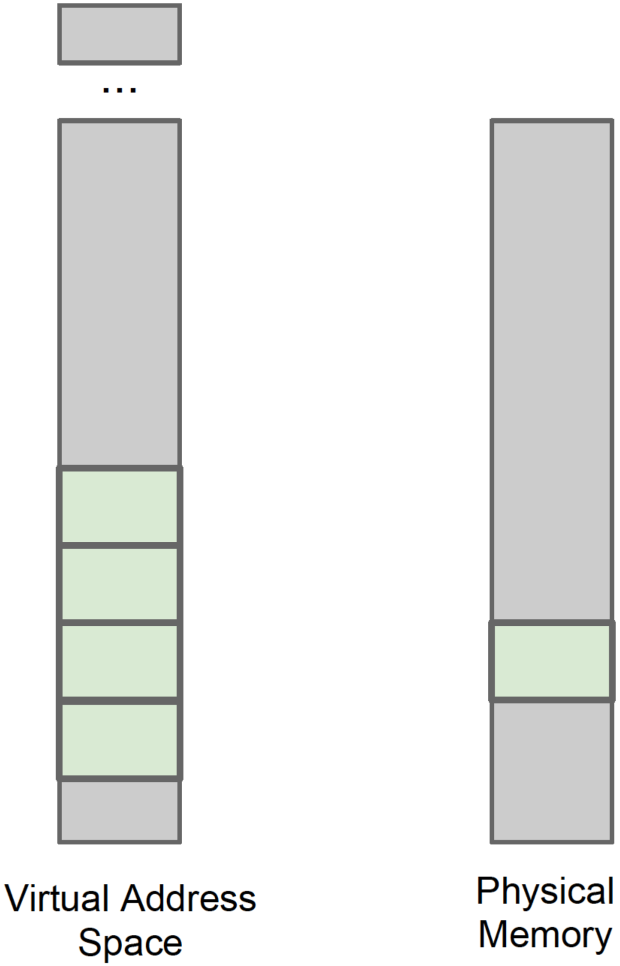
https://www.blackhat.com/docs/us-15/materials/us-15-Seaborn-Exploiting-The-DRAM-Rowhammer-Bug-To-Gain-Kernel-Privileges.pdf



What happens when we map a file with read-write permissions? Indirection via page tables.

This slide is from Mark Seaborn and Thomas Dullien's BlackHat 2015 talk

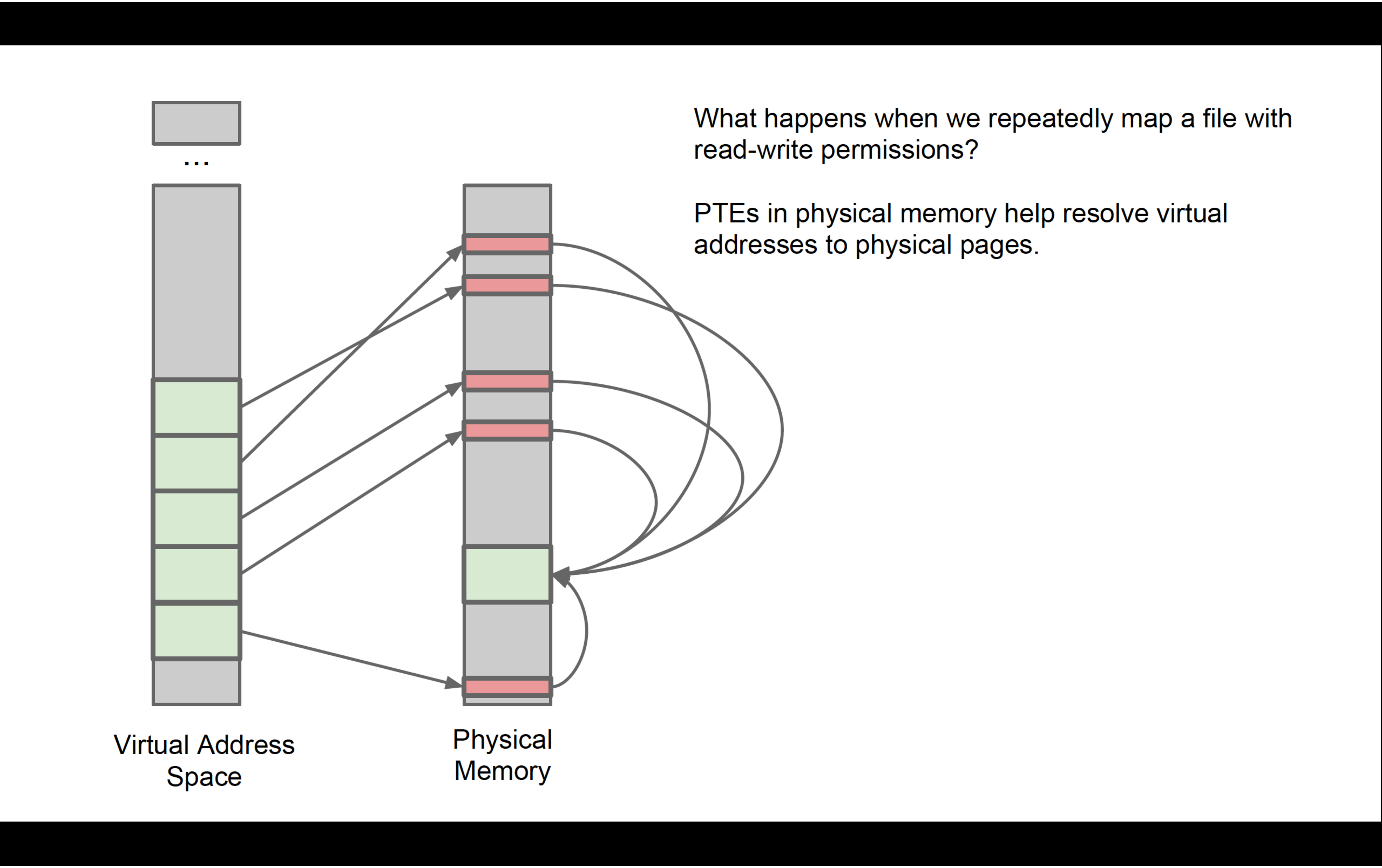
https://www.blackhat.com/docs/us-15/materials/us-15-Seaborn-Exploiting-The-DRAM-Rowhammer-Bug-To-Gain-Kernel-Privileges.pdf



What happens when we repeatedly map a file with read-write permissions?

This slide is from Mark Seaborn and Thomas Dullien’s BlackHat 2015 talk

https://www.blackhat.com/docs/us-15/materials/us-15-Seaborn-Exploiting-The-DRAM-Rowhammer-Bug-To-Gain-Kernel-Privileges.pdf

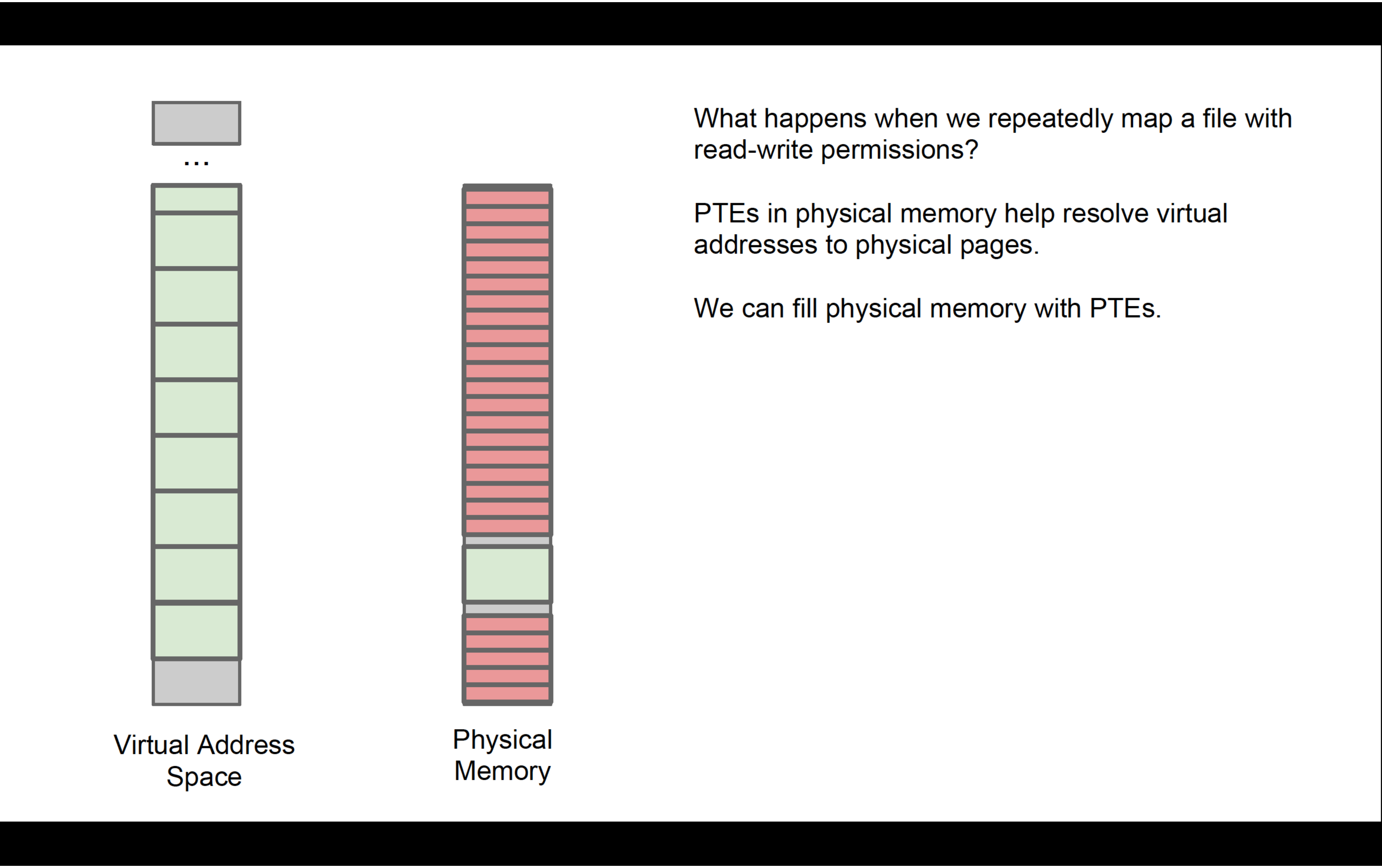


What happens when we repeatedly map a file with read-write permissions?

PTEs in physical memory help resolve virtual addresses to physical pages.

This slide is from Mark Seaborn and Thomas Dullien's BlackHat 2015 talk

https://www.blackhat.com/docs/us-15/materials/us-15-Seaborn-Exploiting-The-DRAM-Rowhammer-Bug-To-Gain-Kernel-Privileges.pdf



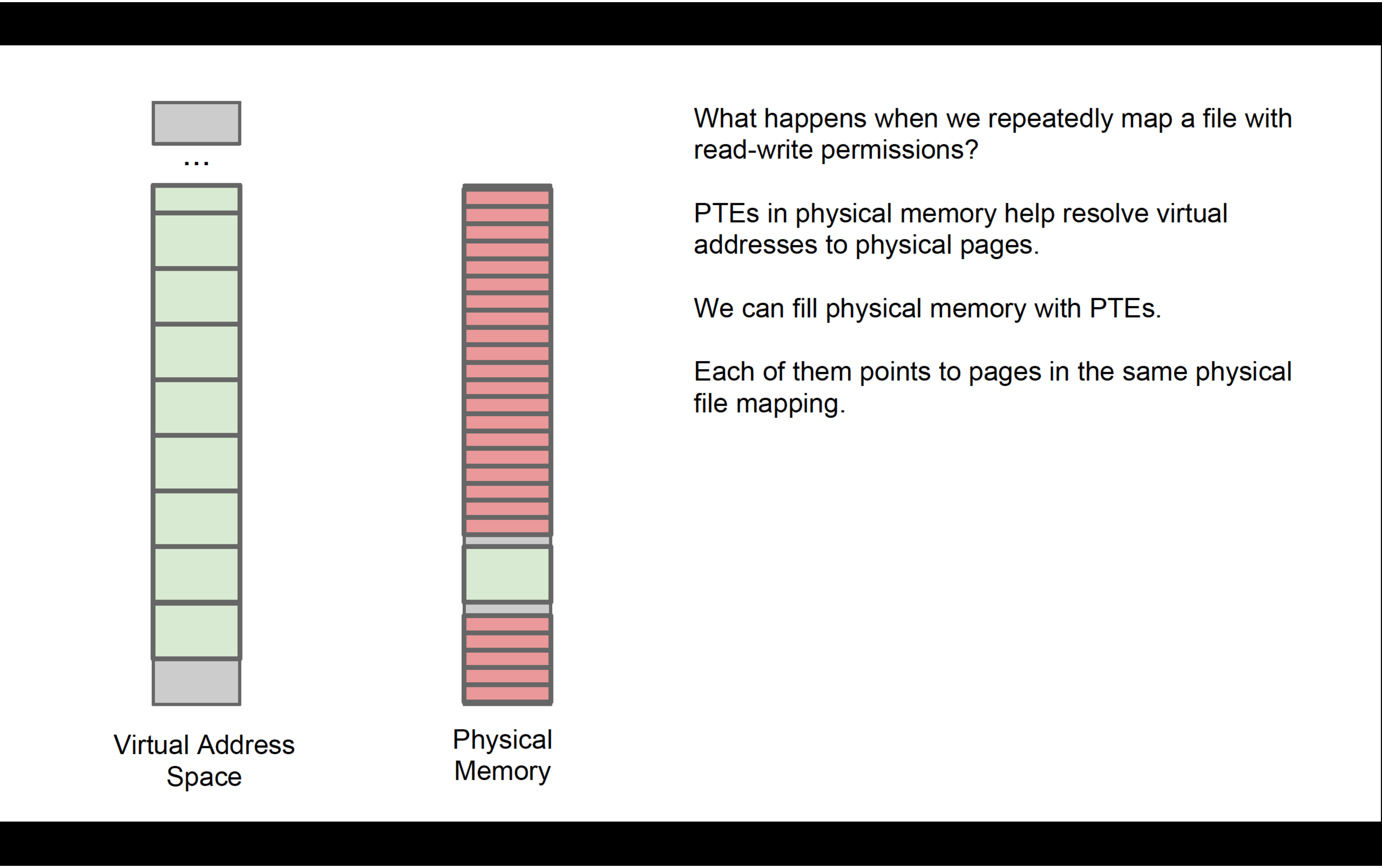
What happens when we repeatedly map a file with read-write permissions?

PTEs in physical memory help resolve virtual addresses to physical pages.

We can fill physical memory with PTEs.

This slide is from Mark Seaborn and Thomas Dullien’s BlackHat 2015 talk

https://www.blackhat.com/docs/us-15/materials/us-15-Seaborn-Exploiting-The-DRAM-Rowhammer-Bug-To-Gain-Kernel-Privileges.pdf



What happens when we repeatedly map a file with read-write permissions?

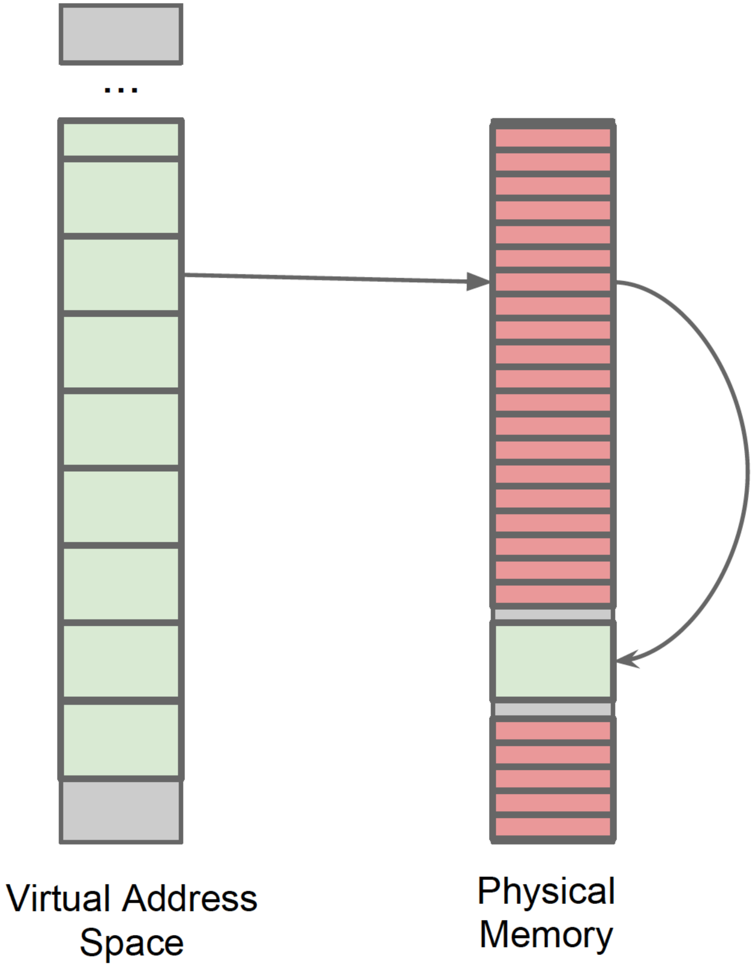
PTEs in physical memory help resolve virtual addresses to physical pages.

We can fill physical memory with PTEs.

Each of them points to pages in the same physical file mapping.

This slide is from Mark Seaborn and Thomas Dullien’s BlackHat 2015 talk

https://www.blackhat.com/docs/us-15/materials/us-15-Seaborn-Exploiting-The-DRAM-Rowhammer-Bug-To-Gain-Kernel-Privileges.pdf



What happens when we repeatedly map a file with read-write permissions?

PTEs in physical memory help resolve virtual addresses to physical pages.

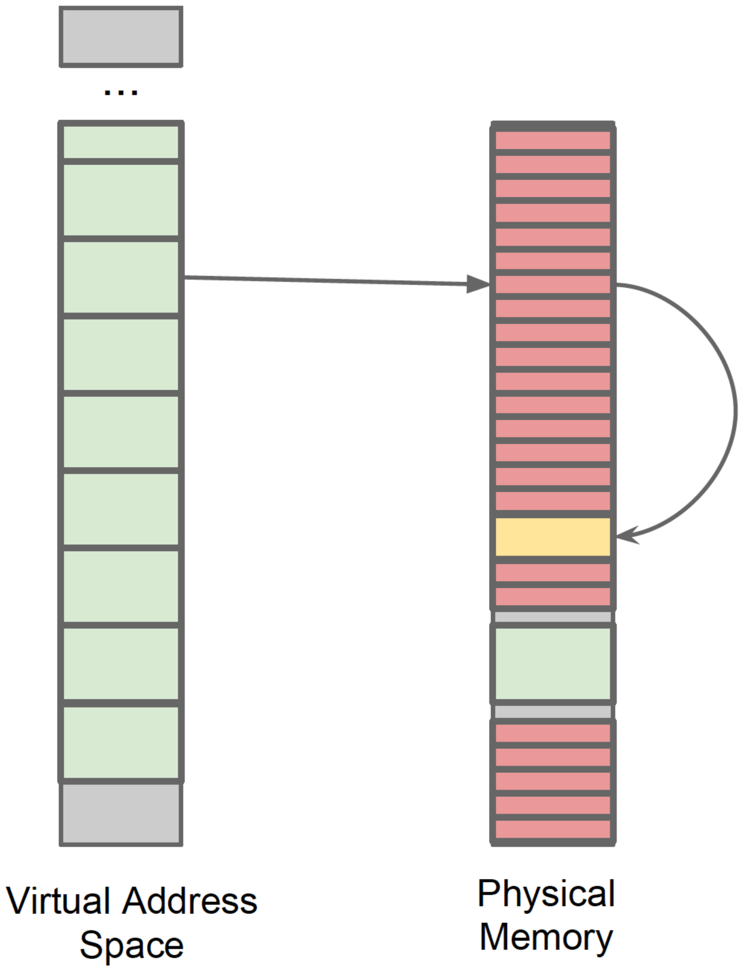
We can fill physical memory with PTEs.

Each of them points to pages in the same physical file mapping.

If a bit in the right place in the PTE flips ...

This slide is from Mark Seaborn and Thomas Dullien's BlackHat 2015 talk

https://www.blackhat.com/docs/us-15/materials/us-15-Seaborn-Exploiting-The-DRAM-Rowhammer-Bug-To-Gain-Kernel-Privileges.pdf



What happens when we repeatedly map a file with read-write permissions?

PTEs in physical memory help resolve virtual addresses to physical pages.

We can fill physical memory with PTEs.

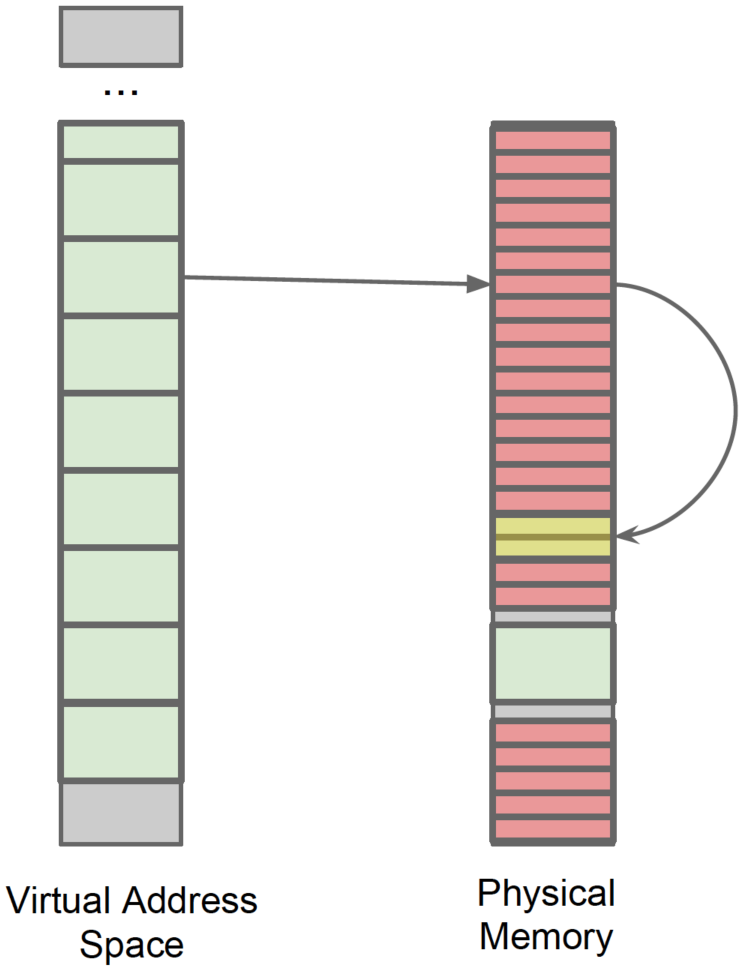
Each of them points to pages in the same physical file mapping.

If a bit in the right place in the PTE flips …

… the corresponding virtual address now points to a wrong physical page - with RW access.

This slide is from Mark Seaborn and Thomas Dullien’s BlackHat 2015 talk

https://www.blackhat.com/docs/us-15/materials/us-15-Seaborn-Exploiting-The-DRAM-Rowhammer-Bug-To-Gain-Kernel-Privileges.pdf



What happens when we repeatedly map a file with read-write permissions?

PTEs in physical memory help resolve virtual addresses to physical pages.

We can fill physical memory with PTEs.

Each of them points to pages in the same physical file mapping.

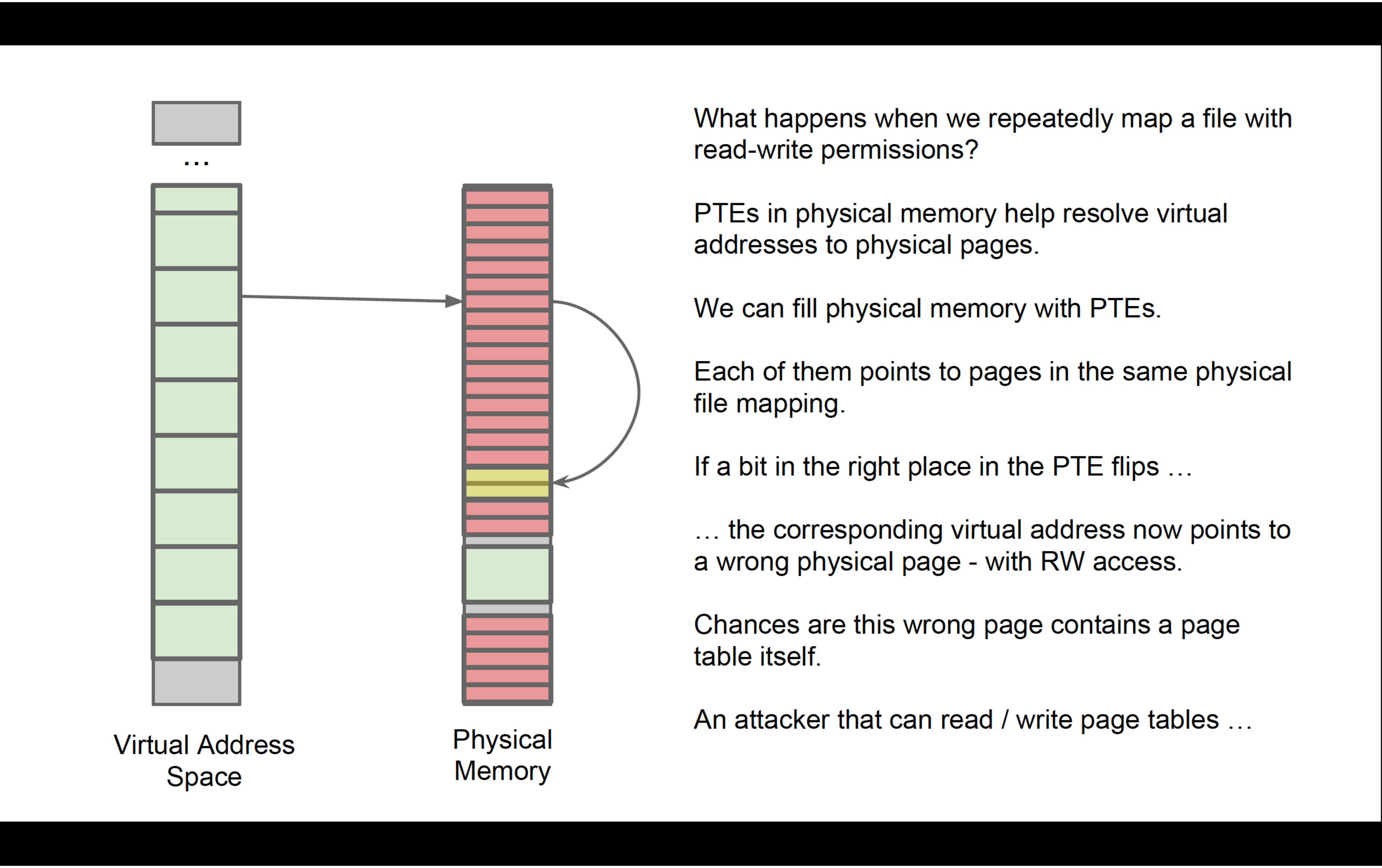
If a bit in the right place in the PTE flips …

… the corresponding virtual address now points to a wrong physical page - with RW access.

Chances are this wrong page contains a page table itself.

This slide is from Mark Seaborn and Thomas Dullien's BlackHat 2015 talk

https://www.blackhat.com/docs/us-15/materials/us-15-Seaborn-Exploiting-The-DRAM-Rowhammer-Bug-To-Gain-Kernel-Privileges.pdf



What happens when we repeatedly map a file with read-write permissions?

PTEs in physical memory help resolve virtual addresses to physical pages.

We can fill physical memory with PTEs.

Each of them points to pages in the same physical file mapping.

If a bit in the right place in the PTE flips …

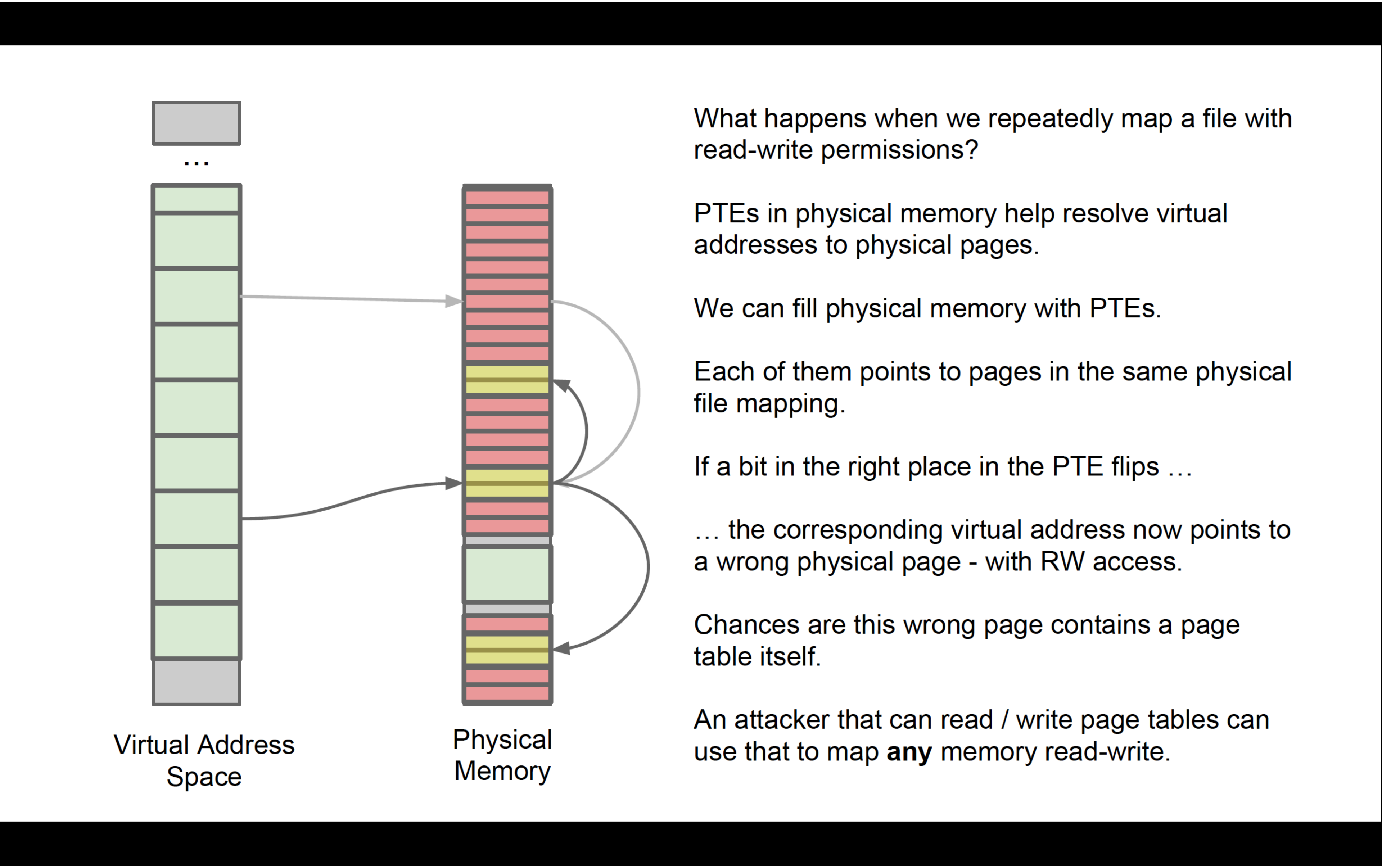
… the corresponding virtual address now points to a wrong physical page - with RW access.

Chances are this wrong page contains a page table itself.

An attacker that can read / write page tables …

This slide is from Mark Seaborn and Thomas Dullien’s BlackHat 2015 talk

https://www.blackhat.com/docs/us-15/materials/us-15-Seaborn-Exploiting-The-DRAM-Rowhammer-Bug-To-Gain-Kernel-Privileges.pdf



What happens when we repeatedly map a file with read-write permissions?

PTEs in physical memory help resolve virtual addresses to physical pages.

We can fill physical memory with PTEs.

Each of them points to pages in the same physical file mapping.

If a bit in the right place in the PTE flips …

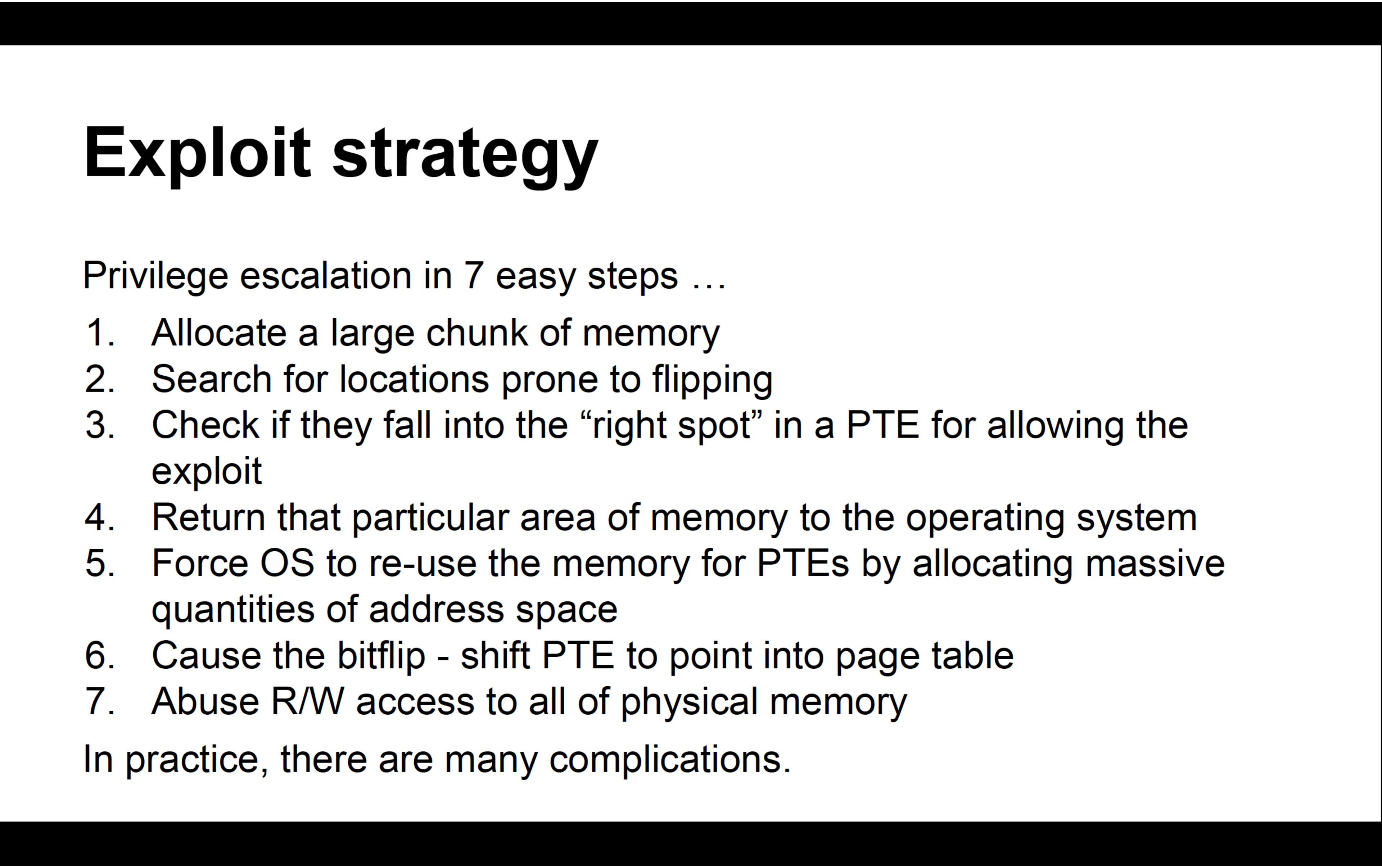
… the corresponding virtual address now points to a wrong physical page - with RW access.

Chances are this wrong page contains a page table itself.

An attacker that can read / write page tables can use that to map **any** memory read-write.

This slide is from Mark Seaborn and Thomas Dullien’s BlackHat 2015 talk

https://www.blackhat.com/docs/us-15/materials/us-15-Seaborn-Exploiting-The-DRAM-Rowhammer-Bug-To-Gain-Kernel-Privileges.pdf



Exploit strategy

Privilege escalation in 7 easy steps …

1. Allocate a large chunk of memory
2. Search for locations prone to flipping
3. Check if they fall into the “right spot” in a PTE for allowing the exploit
4. Return that particular area of memory to the operating system
5. Force OS to re-use the memory for PTEs by allocating massive quantities of address space
6. Cause the bitflip - shift PTE to point into page table
7. Abuse R/W access to all of physical memory

In practice, there are many complications.

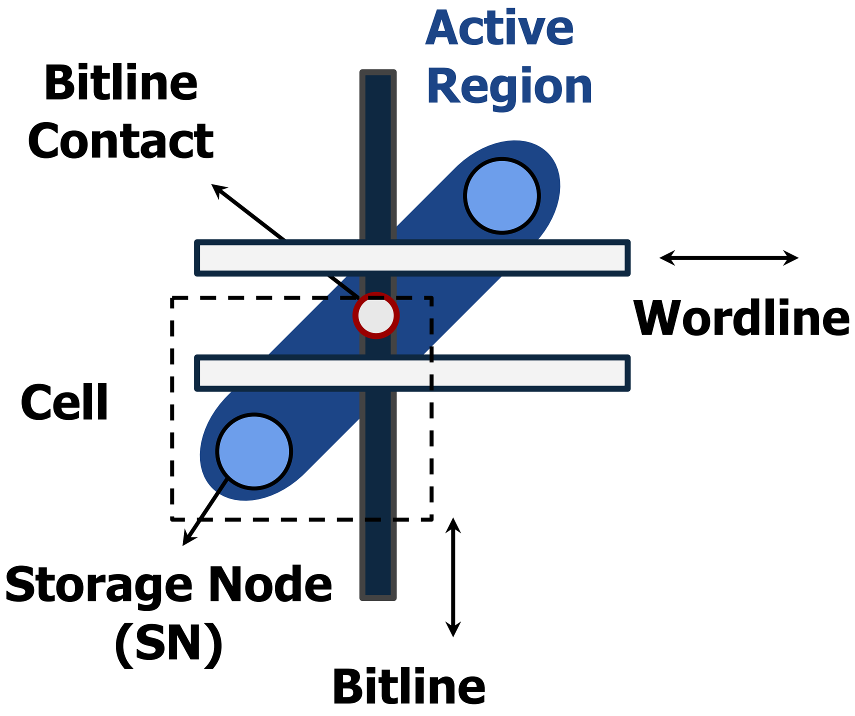
This slide is from Mark Seaborn and Thomas Dullien’s BlackHat 2015 talk

https://www.blackhat.com/docs/us-15/materials/us-15-Seaborn-Exploiting-The-DRAM-Rowhammer-Bug-To-Gain-Kernel-Privileges.pdf

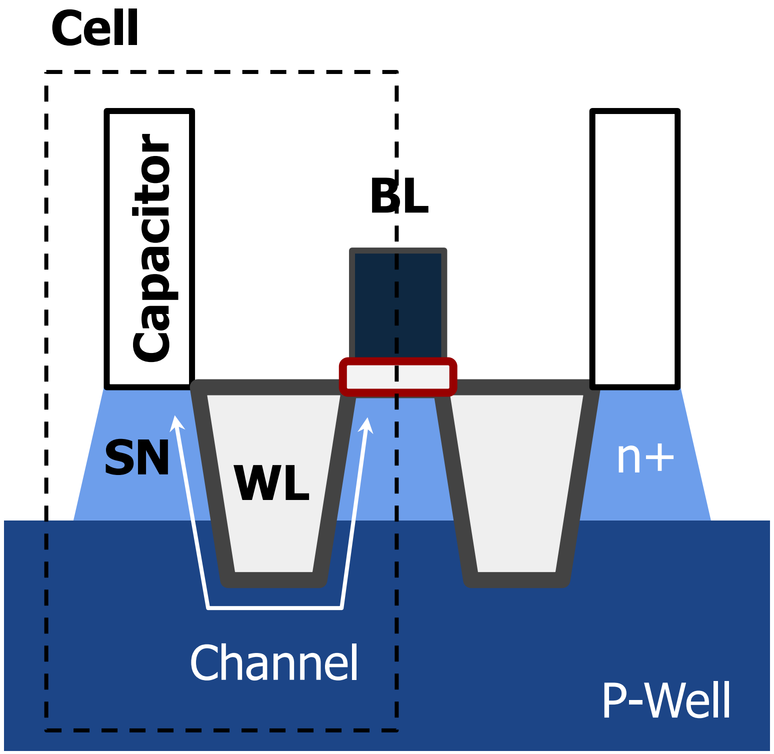
Device-Level RowHammer Mechanisms

DRAM Array Layout

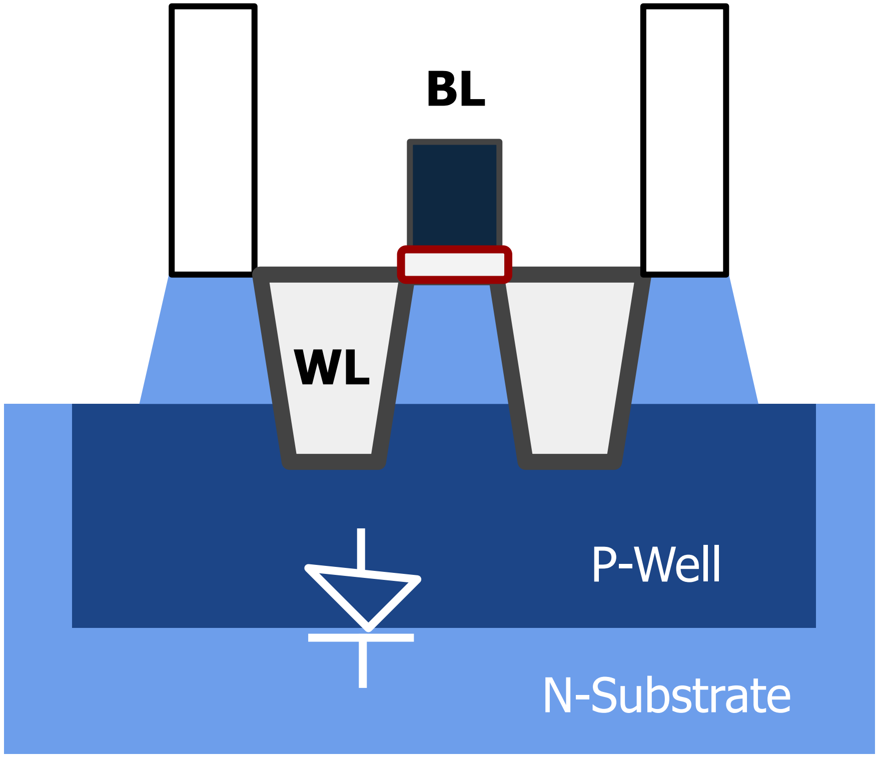
Top View



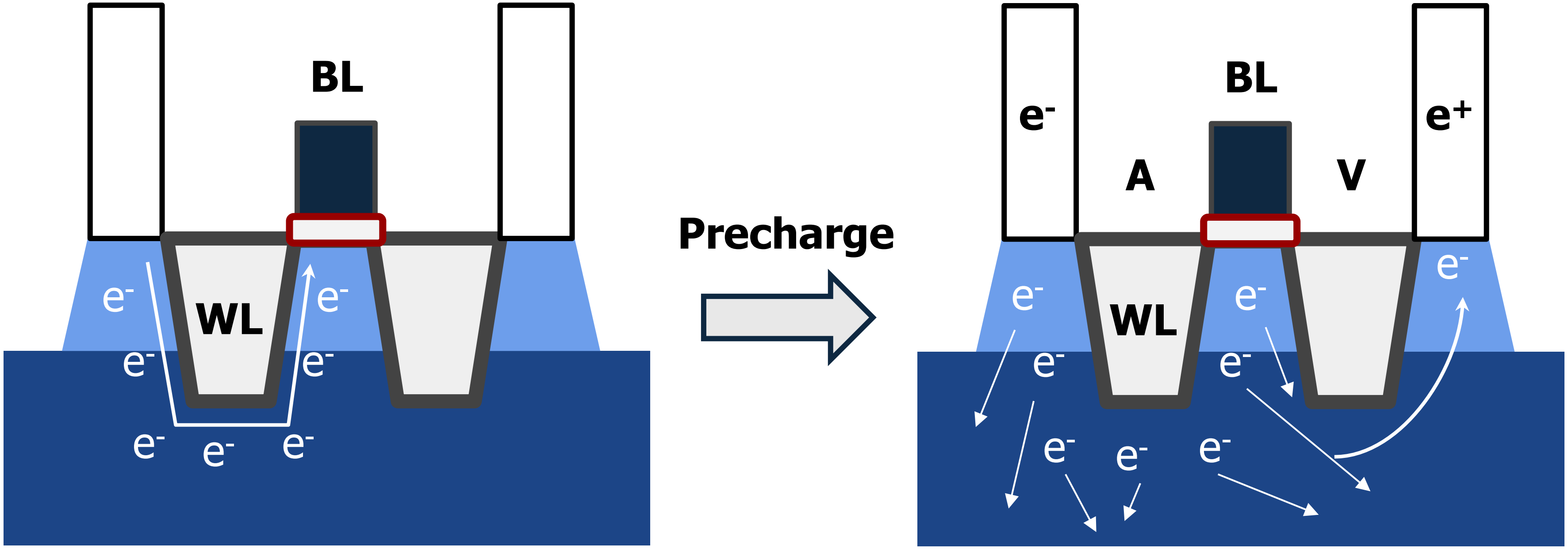
Cross Section



Mechanism 0: Reflecting Electric Field



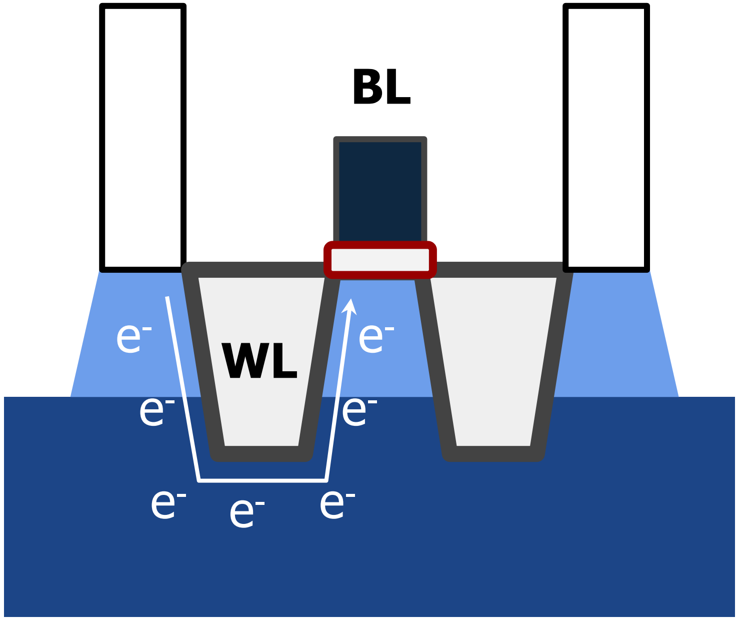
Mechanism 1: Electron Injection



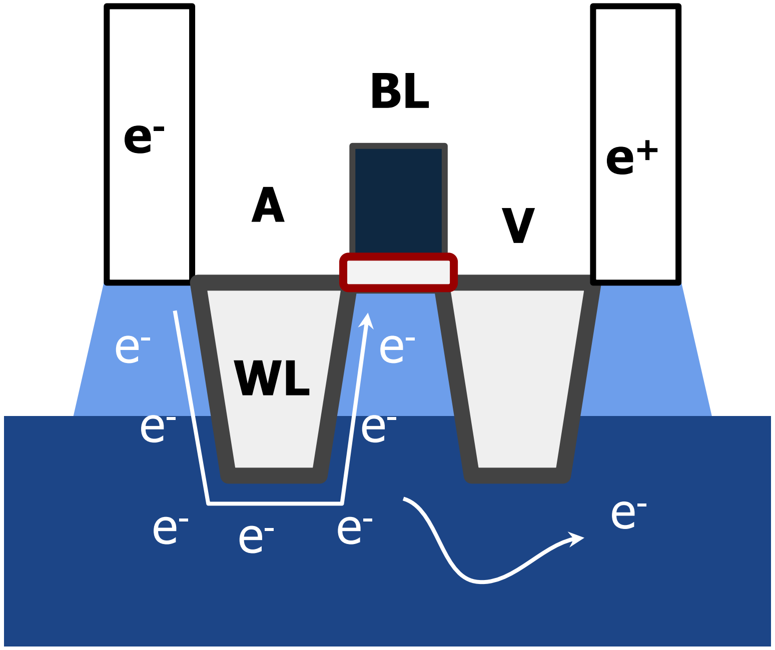
Aggressor ACT

Recombination

Mechanism 2: Electron Drift



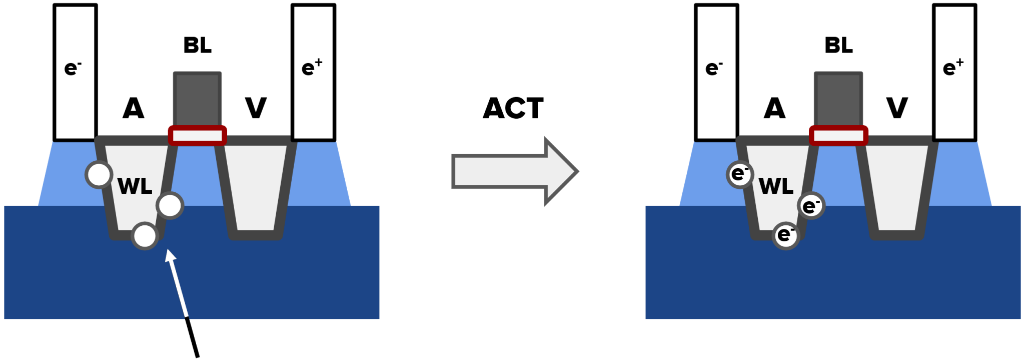
Aggressor ACT



Electron Drift

More

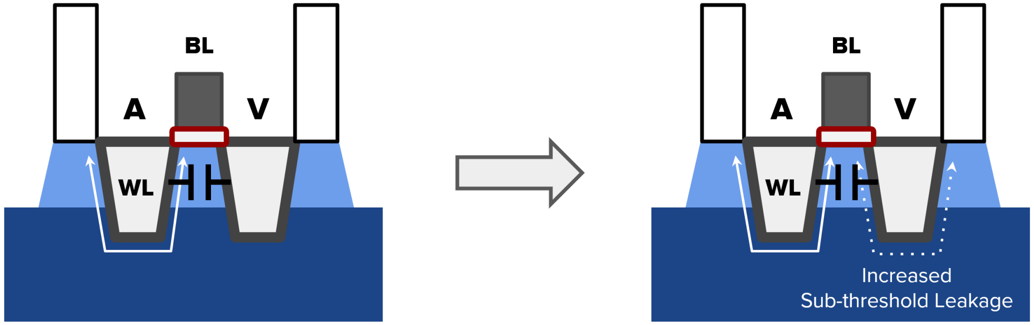
- Charge traps



Interface Charge Trap

Trap Charged

- Wordline Crosstalk



Aggressor ACT

Victim Leakage

Virtual Memory

Access Control & Protection Mechanisms

- Are based on virtual memory (VM), invented in 1950s

- VM has not changed much even after decades of technology scaling and memory system improvements

- VM causes large performance problems and is responsible for large complexity, power, energy
- VM is poor for fine-grained security and access control
- VM hinders innovation in heterogeneous (accelerator) systems and new architectures (e.g., processing near data)

- **It is time to rethink virtual memory**

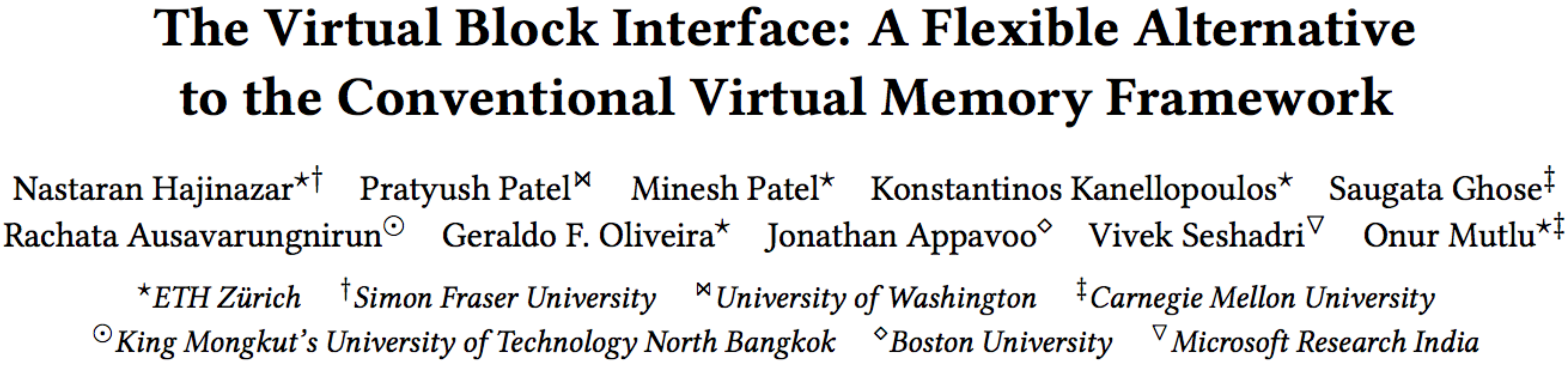
Virtual Memory: Parting Thoughts

- Virtual Memory is one of the most successful examples of
  - architectural support for programmers
  - how to partition work between hardware and software
  - hardware/software cooperation
  - programmer/architect tradeoff

- Going forward: How does virtual memory fare and scale into the future? Five key trends:
  - Increasing, huge physical memory sizes (local & remote)
  - Hybrid physical memory systems (DRAM + NVM + SSD)
  - Many accelerators in the system accessing physical memory
  - Virtualized systems (hypervisors, software virtualization, local and remote memories)
  - Processing in memory systems – near-data accelerators

Rethinking Virtual Memory

Nastaran Hajinazar, Pratyush Patel, Minesh Patel, Konstantinos Kanellopoulos, Saugata Ghose, Rachata Ausavarungnirun, Geraldo Francisco de Oliveira Jr., Jonathan Appavoo, Vivek Seshadri, and Onur Mutlu,
"The Virtual Block Interface: A Flexible Alternative to the Conventional Virtual Memory Framework"
Proceedings of the 47th International Symposium on Computer Architecture (***ISCA***), Virtual, June 2020.
[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[ARM Research Summit Poster (pptx) (pdf)]
[Talk Video (26 minutes)]
[Lightning Talk Video (3 minutes)]
[Lecture Video (43 minutes)]



The Virtual Block Interface: A Flexible Alternative to the Conventional Virtual Memory Framework

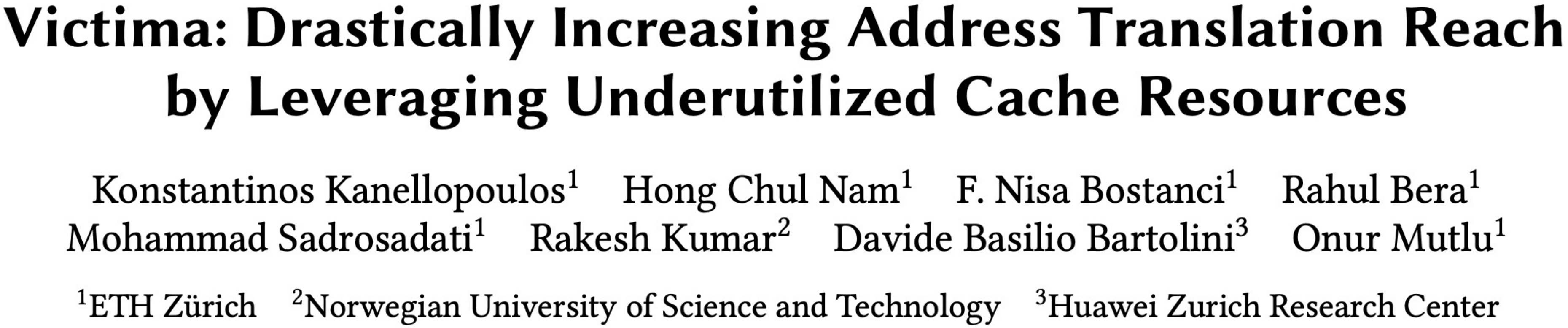
Nastaran Hajinazar[⋆†] Pratyush Patel[⋈] Minesh Patel[⋆] Konstantinos Kanellopoulos[⋆] Saugata Ghose[‡]
Rachata Ausavarungnirun[⊙] Geraldo F. Oliveira[⋆] Jonathan Appavoo[◇] Vivek Seshadri[▽] Onur Mutlu[⋆‡]

[⋆]*ETH Zürich* [†]*Simon Fraser University* [⋈]*University of Washington* [‡]*Carnegie Mellon University*
[⊙]*King Mongkut's University of Technology North Bangkok* [◇]*Boston University* [▽]*Microsoft Research India*

https://people.inf.ethz.ch/omutlu/pub/VBI-virtual-block-interface_isca20.pdf

Better Virtual Memory (I)

Konstantinos Kanellopoulos, Hong Chul Nam, F. Nisa Bostanci, Rahul Bera, Mohammad Sadrosadati, Rakesh Kumar, Davide Basilio Bartolini, and Onur Mutlu,
"Victima: Drastically Increasing Address Translation Reach by Leveraging Underutilized Cache Resources"
Proceedings of the 56th International Symposium on Microarchitecture (***MICRO***), Toronto, ON, Canada, November 2023.
[Slides (pptx) (pdf)]
[arXiv version]
[Victima Source Code (Officially Artifact Evaluated with All Badges)]
Officially artifact evaluated as available, functional, reusable and reproducible.
Distinguished artifact award at MICRO 2023.



Victima: Drastically Increasing Address Translation Reach by Leveraging Underutilized Cache Resources

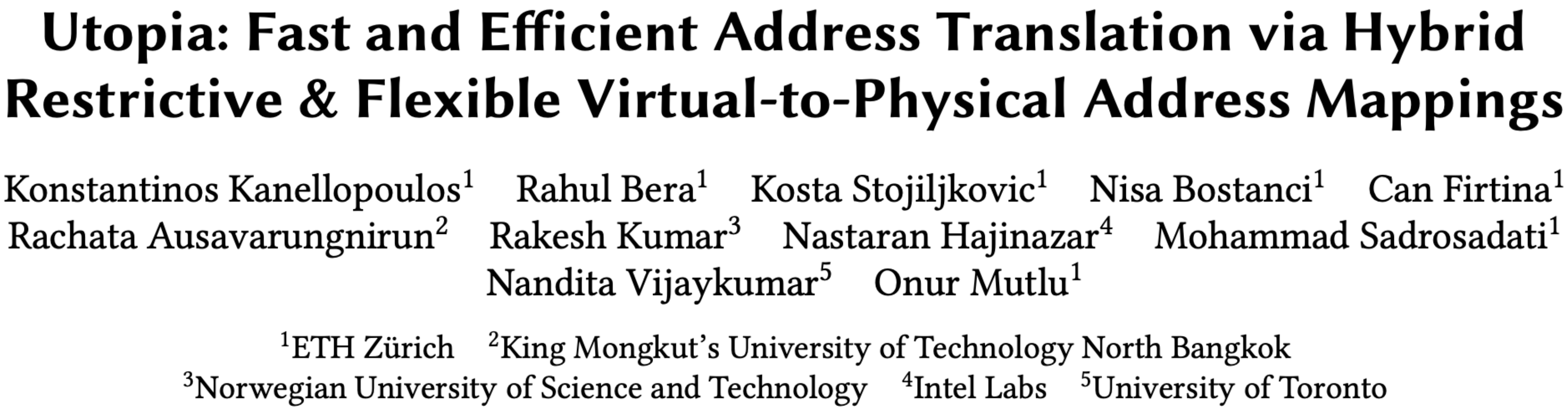
Konstantinos Kanellopoulos[1] Hong Chul Nam[1] F. Nisa Bostanci[1] Rahul Bera[1]
Mohammad Sadrosadati[1] Rakesh Kumar[2] Davide Basilio Bartolini[3] Onur Mutlu[1]

[1]ETH Zürich [2]Norwegian University of Science and Technology [3]Huawei Zurich Research Center

https://arxiv.org/pdf/2310.04158

Better Virtual Memory (II)

Konstantinos Kanellopoulos, Rahul Bera, Kosta Stojiljkovic, Nisa Bostanci, Can Firtina, Rachata Ausavarungnirun, Rakesh Kumar, Nastaran Hajinazar, Mohammad Sadrosadati, Nandita Vijaykumar, and Onur Mutlu,
"Utopia: Fast and Efficient Address Translation via Hybrid Restrictive & Flexible Virtual-to-Physical Address Mappings"
Proceedings of the 56th International Symposium on Microarchitecture (***MICRO***), Toronto, ON, Canada, November 2023.
[Slides (pptx) (pdf)]
[arXiv version]
[Utopia Source Code]



Utopia: Fast and Efficient Address Translation via Hybrid Restrictive & Flexible Virtual-to-Physical Address Mappings

Konstantinos Kanellopoulos[1] Rahul Bera[1] Kosta Stojiljkovic[1] Nisa Bostanci[1] Can Firtina[1]
Rachata Ausavarungnirun[2] Rakesh Kumar[3] Nastaran Hajinazar[4] Mohammad Sadrosadati[1]
Nandita Vijaykumar[5] Onur Mutlu[1]

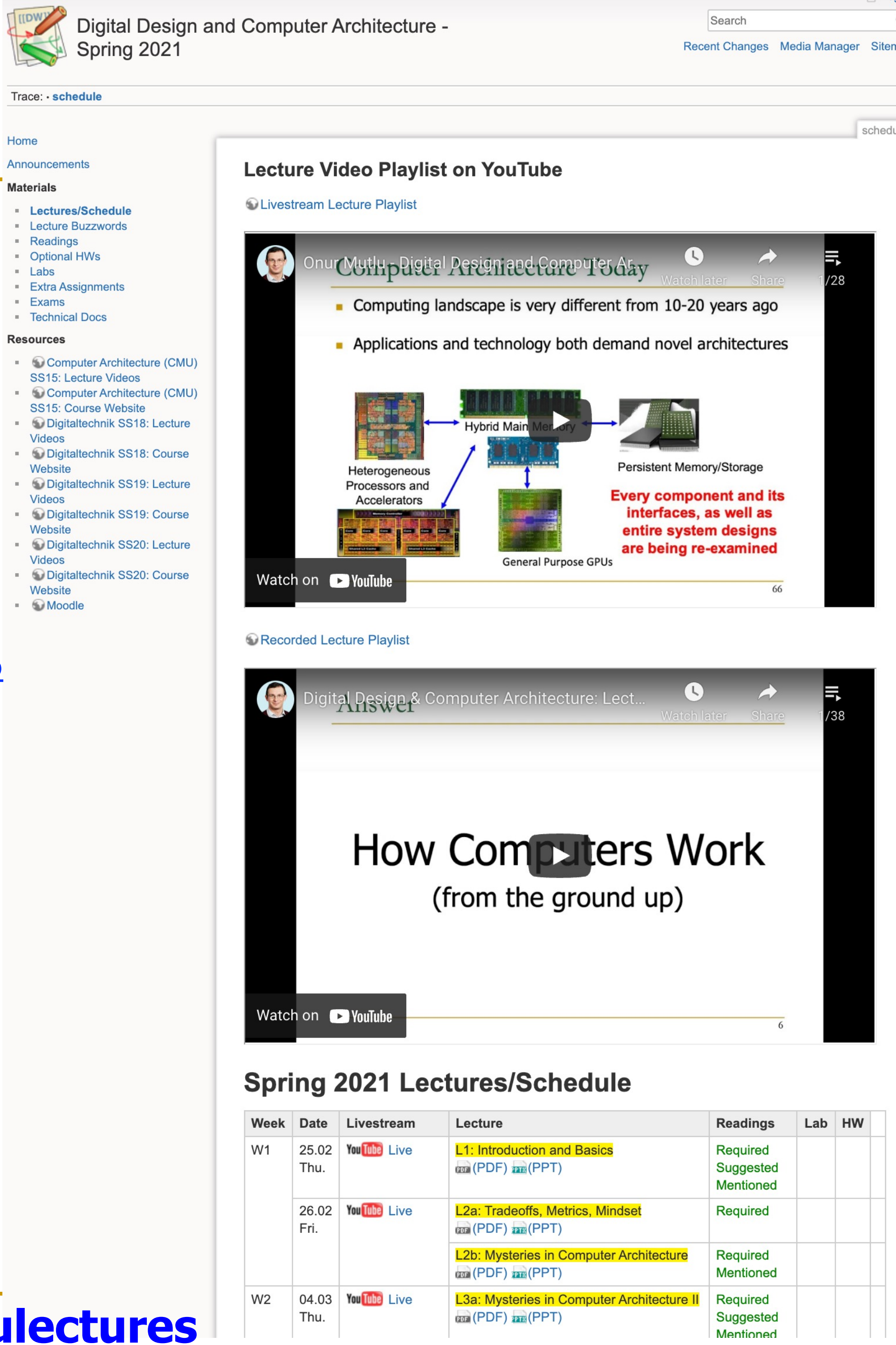
[1]ETH Zürich [2]King Mongkut's University of Technology North Bangkok
[3]Norwegian University of Science and Technology [4]Intel Labs [5]University of Toronto

https://arxiv.org/abs/2211.12205

Related Courses

DDCA (Spring 2022)

- **Spring 2022 Edition:**
  - 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



Digital Design and Computer Architecture - Spring 2021

Search

Recent Changes Media Manager Siten

Trace: · schedule

schedu

Home

Announcements

Materials

- Lectures/Schedule
- Lecture Buzzwords
- Readings
- Optional HWs
- Labs
- Extra Assignments
- Exams
- Technical Docs

Resources

- Computer Architecture (CMU) SS15: Lecture Videos
- Computer Architecture (CMU) SS15: Course Website
- Digitaltechnik SS18: Lecture Videos
- Digitaltechnik SS18: Course Website
- Digitaltechnik SS19: Lecture Videos
- Digitaltechnik SS19: Course Website
- Digitaltechnik SS20: Lecture Videos
- Digitaltechnik SS20: Course Website
- Moodle

Lecture Video Playlist on YouTube

Livestream Lecture Playlist

Onur Mutlu - Digital Design and Computer Ar...
Computer Architecture Today
Watch later
Share
1/28
Computing landscape is very different from 10-20 years ago
Applications and technology both demand novel architectures
Hybrid Main Memory
Heterogeneous Processors and Accelerators
Persistent Memory/Storage
Every component and its interfaces, as well as entire system designs are being re-examined
General Purpose GPUs
Watch on YouTube
66

Recorded Lecture Playlist

Spring 2021 Lectures/Schedule

Week	Date	Livestream	Lecture	Readings	Lab	HW
W1	25.02 Thu.	YouTube Live	L1: Introduction and Basics (PDF) (PPT)	Required Suggested Mentioned		
	26.02 Fri.	YouTube Live	L2a: Tradeoffs, Metrics, Mindset (PDF) (PPT)	Required		
			L2b: Mysteries in Computer Architecture (PDF) (PPT)	Required Mentioned		
W2	04.03 Thu.	YouTube Live	L3a: Mysteries in Computer Architecture II (PDF) (PPT)	Required Suggested Mentioned		

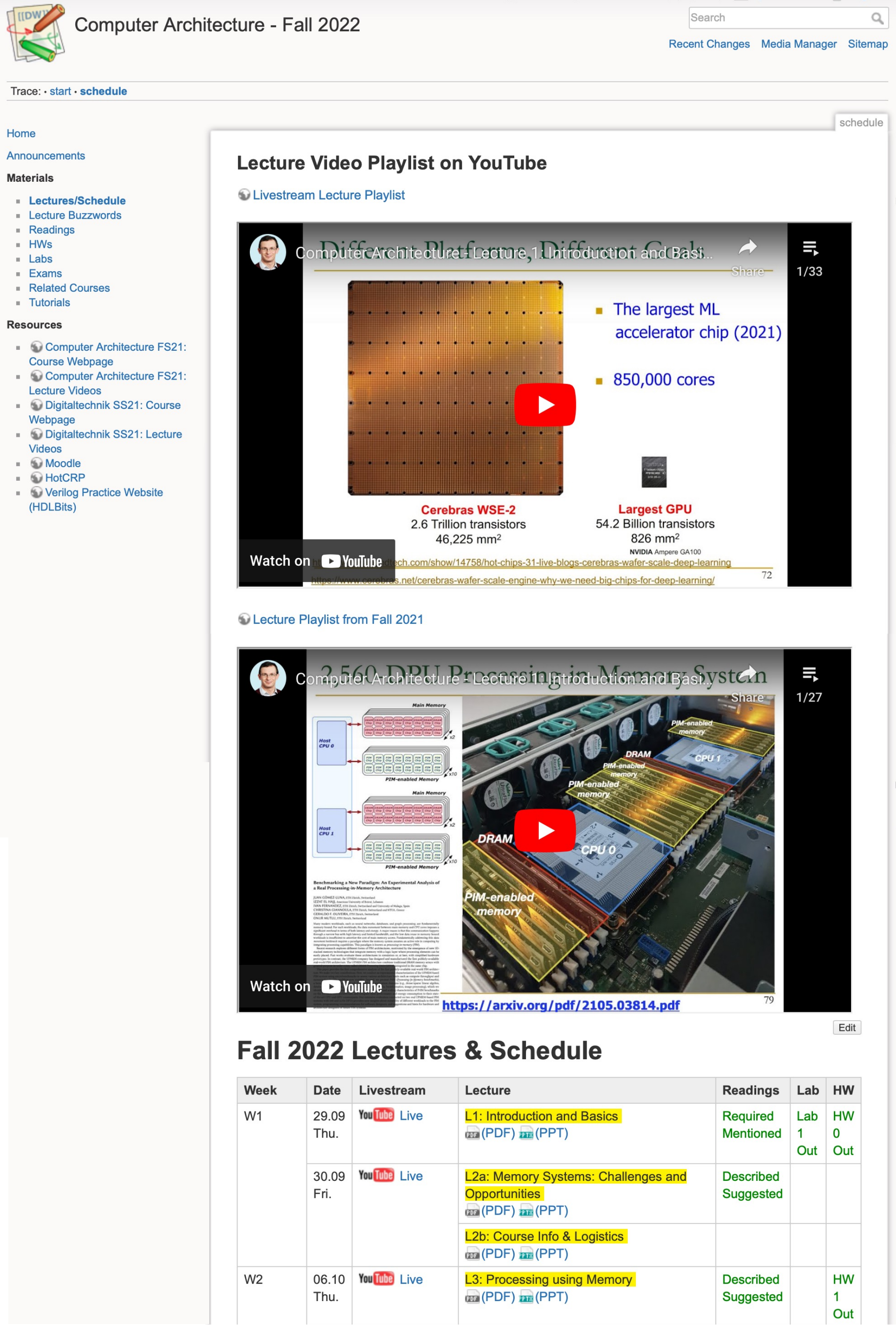
https://www.youtube.com/onurmutlulectures

Comp Arch (Fall 2022)

- **Fall 2022 Edition:**
  - https://safari.ethz.ch/architecture/fall2022/doku.php?id=schedule
- **Fall 2021 Edition:**
  - https://safari.ethz.ch/architecture/fall2021/doku.php?id=schedule

- **Youtube Livestream (2022):**
  - 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

- 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



Computer Architecture - Fall 2022

Trace: · start · schedule

Materials

- Lectures/Schedule
- Lecture Buzzwords
- Readings
- HWs
- Labs
- Exams
- Related Courses
- Tutorials

Resources

- Computer Architecture FS21: Course Webpage
- Computer Architecture FS21: Lecture Videos
- Digitaltechnik SS21: Course Webpage
- Digitaltechnik SS21: Lecture Videos
- Moodle
- HotCRP
- Verilog Practice Website (HDLBits)

Lecture Video Playlist on YouTube

Livestream Lecture Playlist

Lecture Playlist from Fall 2021

Computer Architecture - Lecture 1: Introduction and Basi...
Share
1/27
2,560-DPU Processing-in-Memory System
Host CPU 0
Host CPU 1
Main Memory
PIM-enabled Memory
DRAM
CPU 0
CPU 1
PIM-enabled memory
79
Watch on YouTube
https://arxiv.org/pdf/2105.03814.pdf

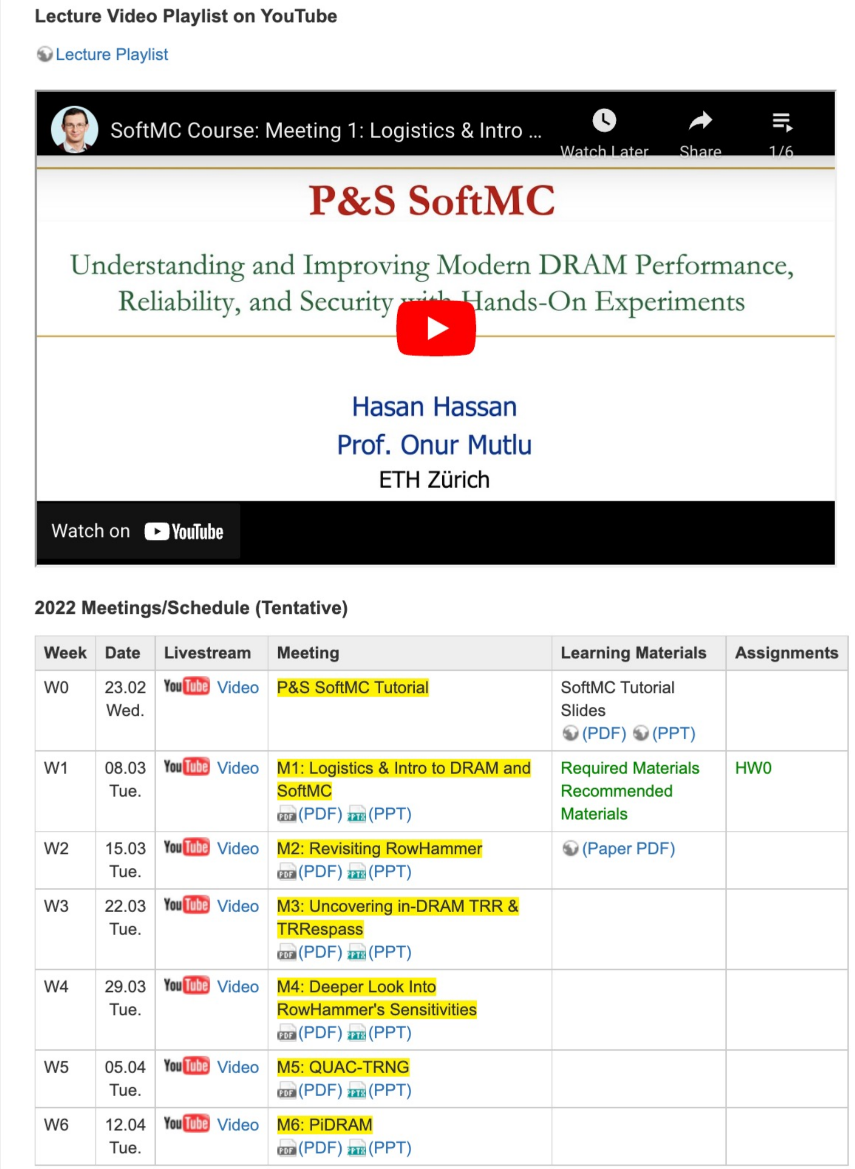
Fall 2022 Lectures & Schedule

Week	Date	Livestream	Lecture	Readings	Lab	HW
W1	29.09 Thu.	YouTube Live	L1: Introduction and Basics (PDF) (PPT)	Required Mentioned	Lab 1 Out	HW 0 Out
	30.09 Fri.	YouTube Live	L2a: Memory Systems: Challenges and Opportunities (PDF) (PPT)	Described Suggested		
			L2b: Course Info & Logistics (PDF) (PPT)			
W2	06.10 Thu.	YouTube Live	L3: Processing using Memory (PDF) (PPT)	Described Suggested		HW 1 Out

https://www.youtube.com/onurmutlulectures

RowHammer & DRAM Exploration (Fall 2022)

- [Fall 2022 Edition:](https://safari.ethz.ch/projects_and_seminars/fall2022/doku.php?id=softmc)
  - 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)
  - 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)
  - 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

SoftMC Course: Meeting 1: Logistics & Intro ...

P&S SoftMC

Understanding and Improving Modern DRAM Performance, Reliability, and Security with Hands-On Experiments

Hasan Hassan
Prof. Onur Mutlu
ETH Zürich

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)		

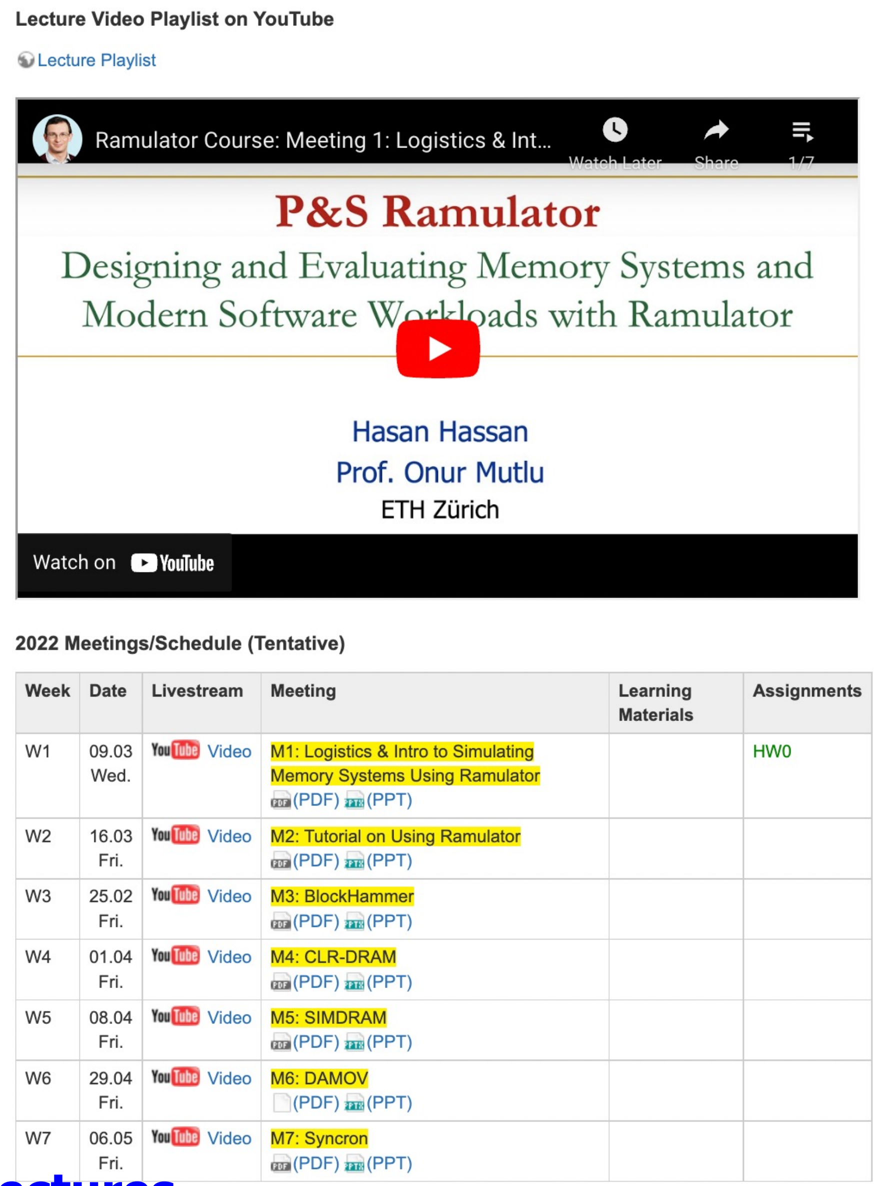
https://www.youtube.com/onurmutlulectures

Exploration of Emerging Memory Systems (Fall 2022)

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

- **Youtube Livestream (Spring 2022):**
  - https://www.youtube.com/watch?v=aM-llXRQd3s&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

P&S Ramulator

Designing and Evaluating Memory Systems and Modern Software Workloads with Ramulator

Hasan Hassan

Prof. Onur Mutlu

ETH Zürich

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