

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

Onur Mutlu

[onur.mutlu@inf.ethz.ch](mailto:onur.mutlu@inf.ethz.ch)

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

April 21, 2017

CFAED Workshop on Resilient Systems Keynote

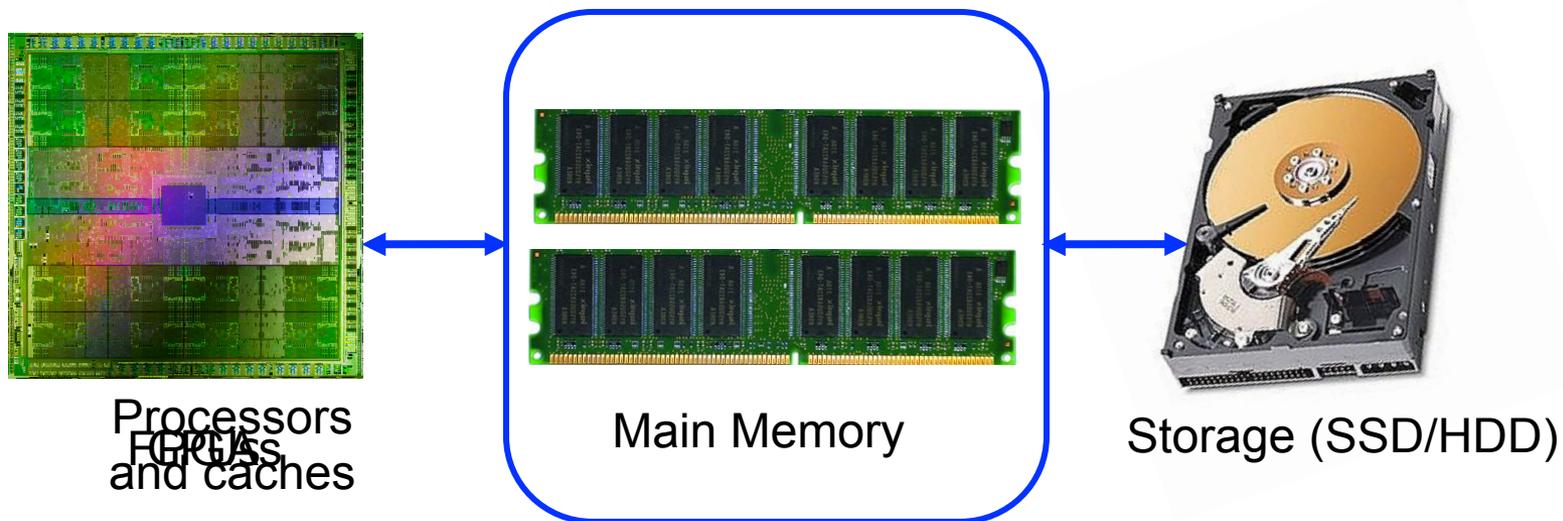


**SAFARI**

**ETH** zürich

# The Main Memory System

---

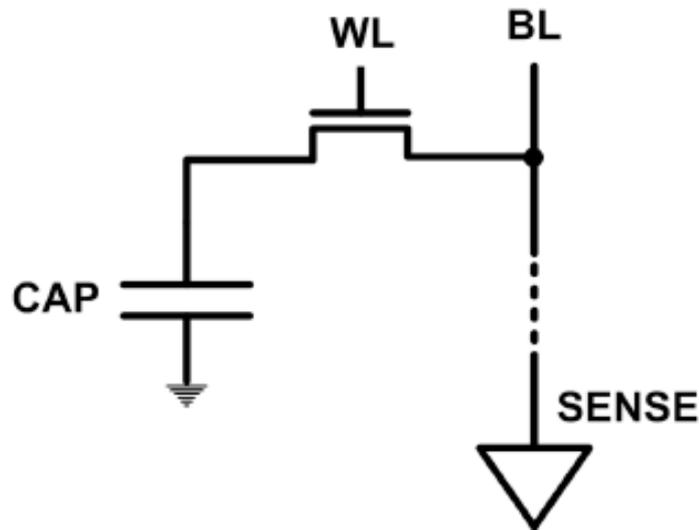


- Main memory is a critical component of all computing systems: server, mobile, embedded, desktop, sensor
- Main memory system must scale (in *size, technology, efficiency, cost, and management algorithms*) to maintain performance growth and technology scaling benefits

# The DRAM Scaling Problem

---

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



- As DRAM cell becomes smaller, it becomes more vulnerable

# Testing DRAM Scaling Issues ...



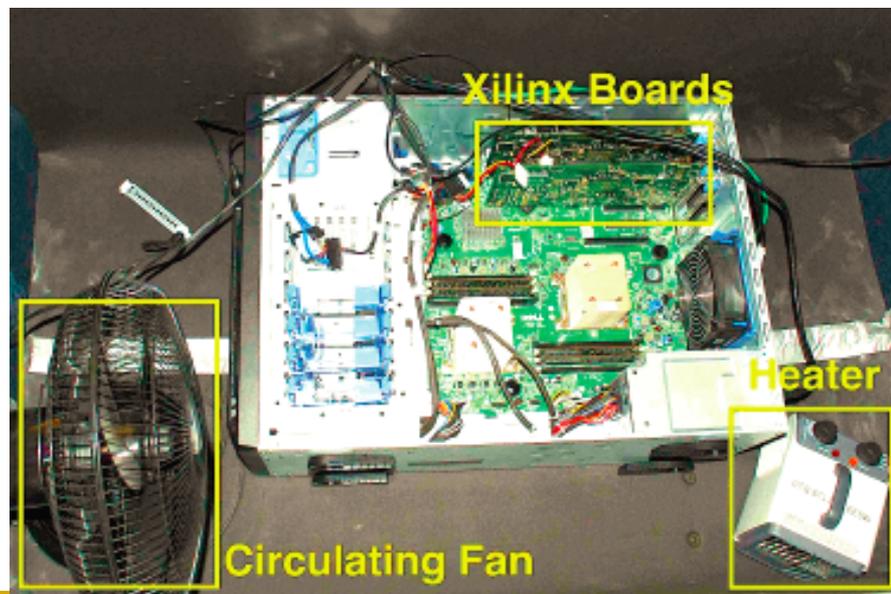
An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms (Liu et al., ISCA 2013)

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

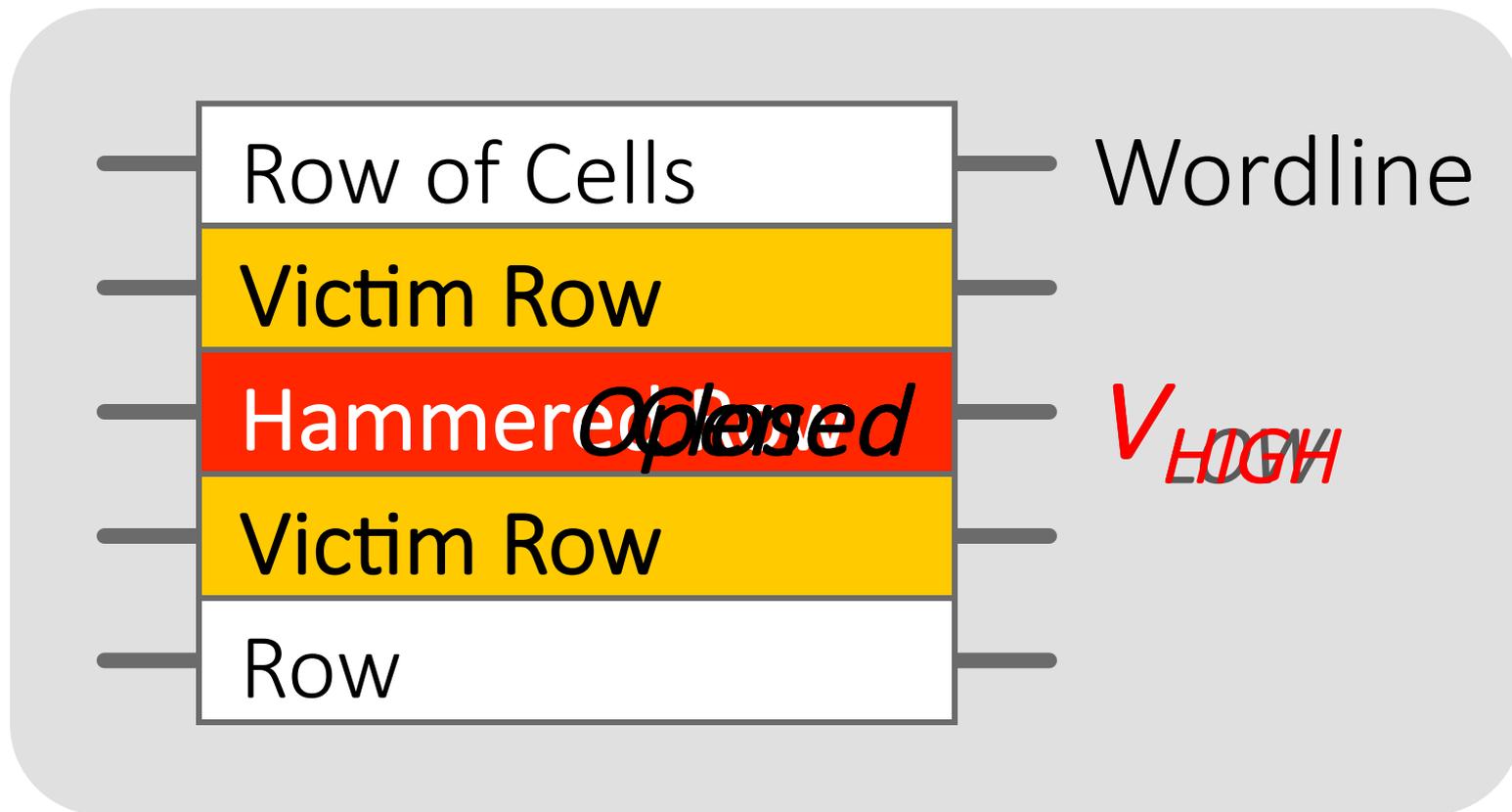
Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors (Kim et al., ISCA 2014)

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

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



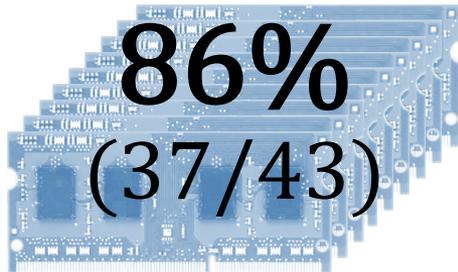
# Modern DRAM is Prone to Disturbance Errors



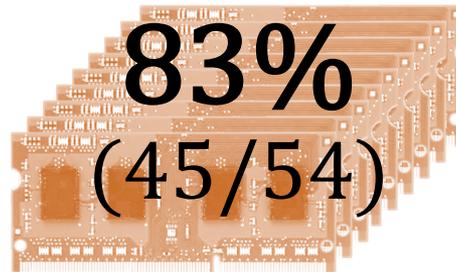
Repeatedly opening and closing a row enough times within a refresh interval induces **disturbance errors** in adjacent rows in **most real DRAM chips you can buy today**

# Most DRAM Modules Are Vulnerable

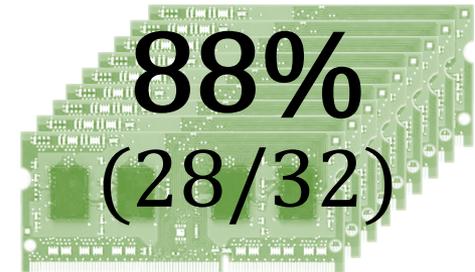
A company



B company



C company

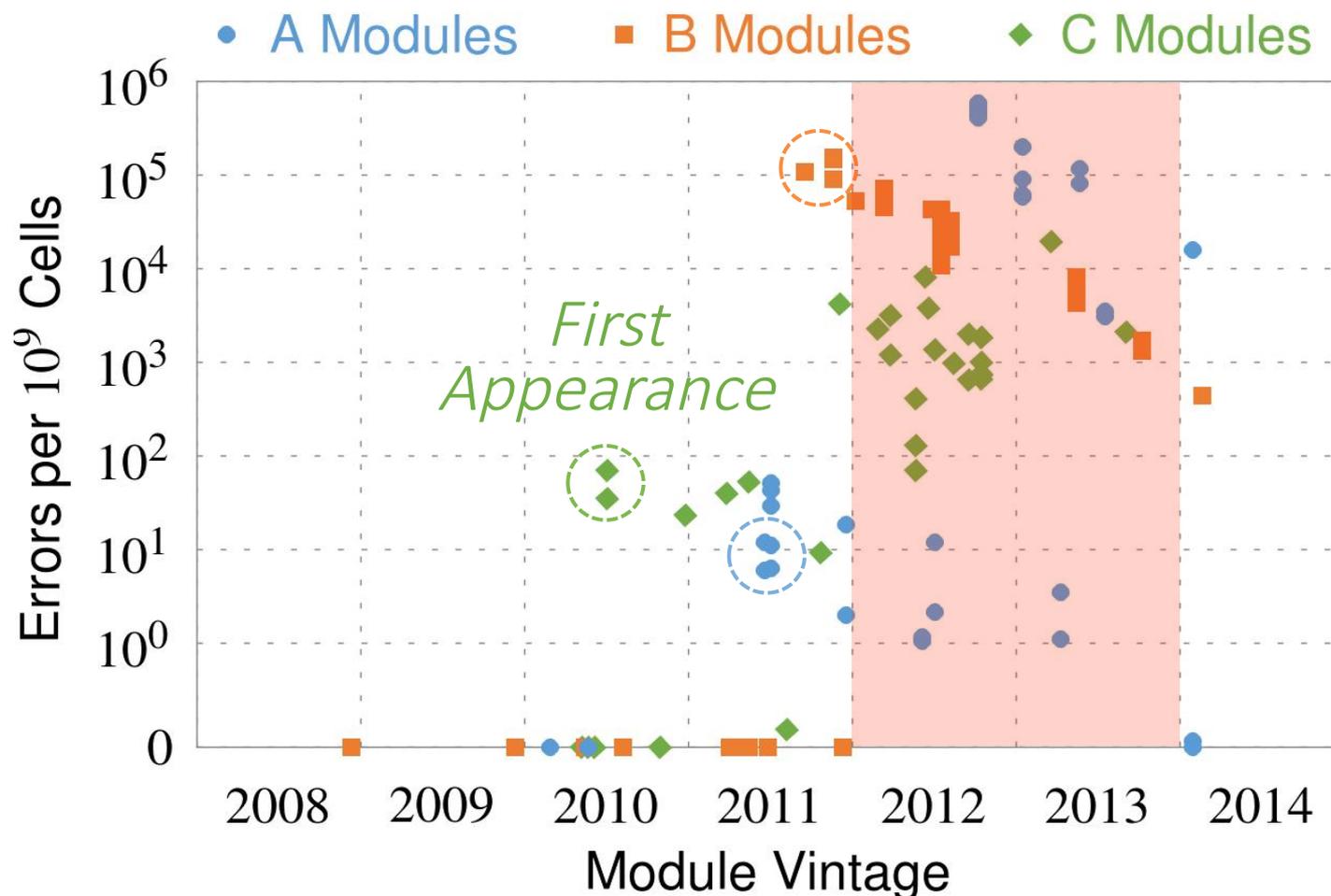


Up to  
 $1.0 \times 10^7$   
errors

Up to  
 $2.7 \times 10^6$   
errors

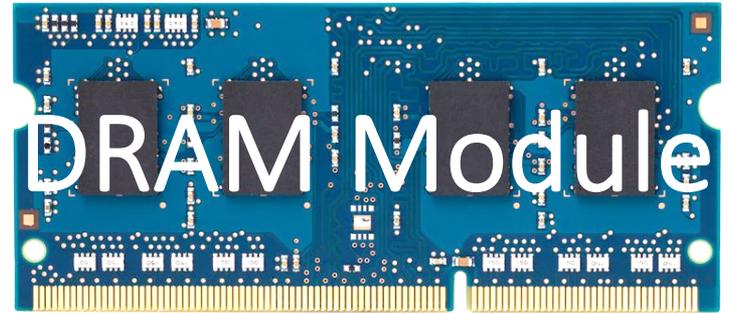
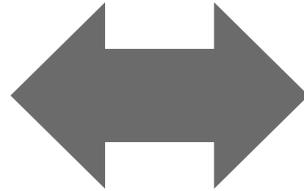
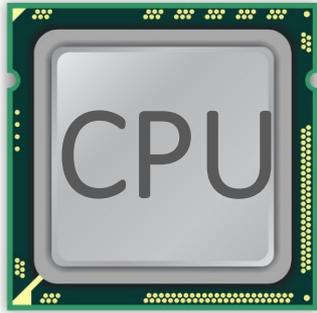
Up to  
 $3.3 \times 10^5$   
errors

# Recent DRAM Is More Vulnerable

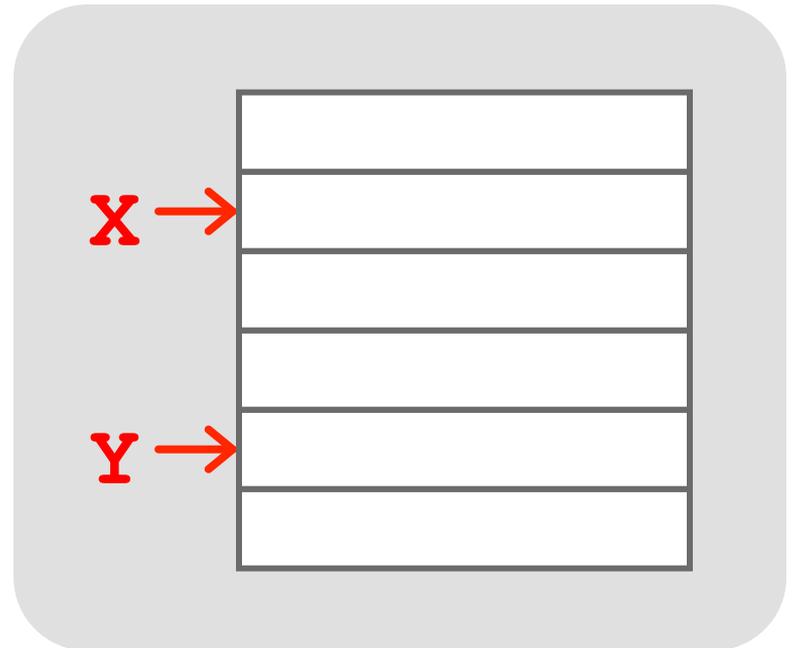


*All modules from 2012–2013 are vulnerable*

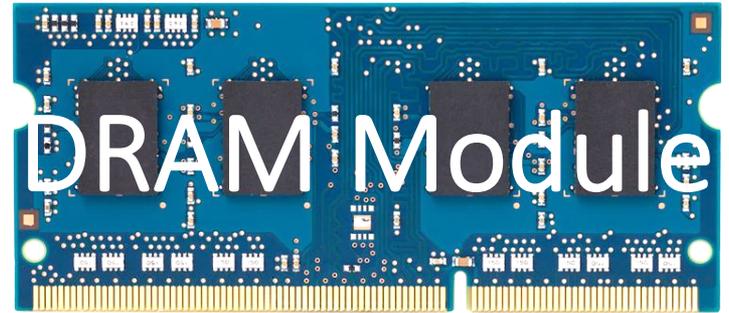
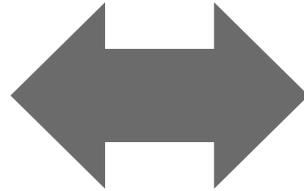
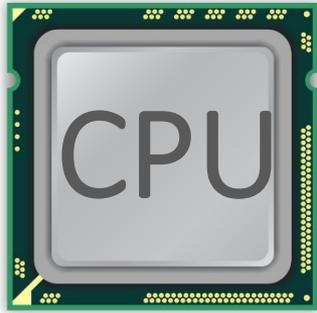
# A Simple Program Can Induce Many Errors



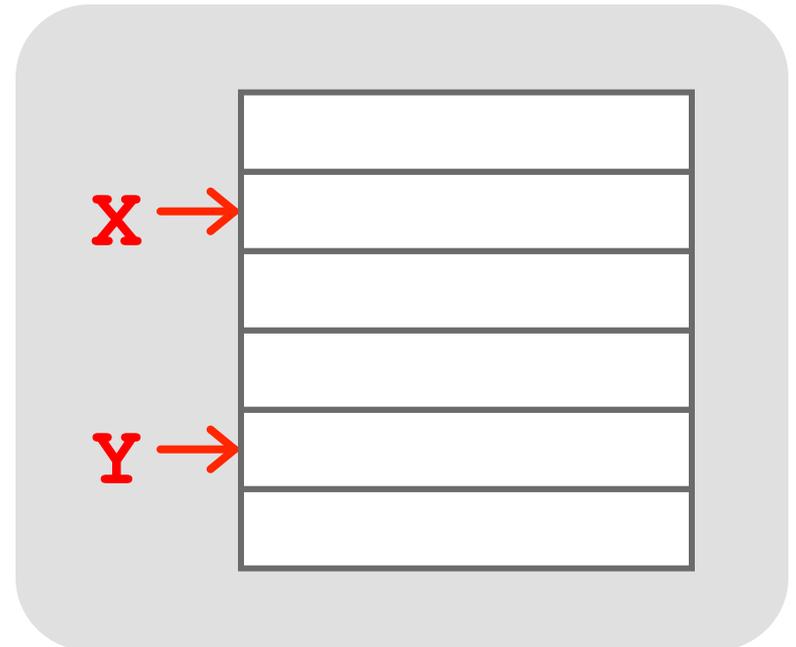
```
loop:  
  mov  (X), %eax  
  mov  (Y), %ebx  
  clflush (X)  
  clflush (Y)  
  mfence  
  jmp  loop
```



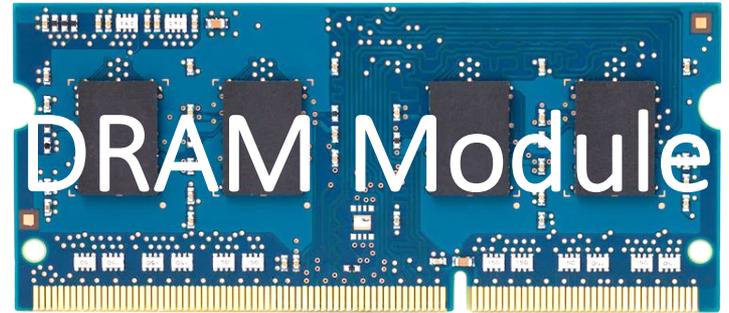
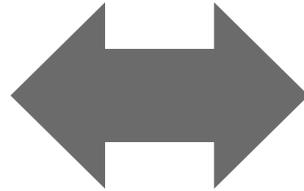
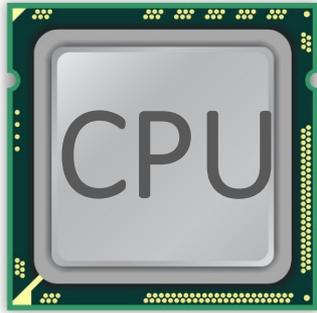
# A Simple Program Can Induce Many Errors



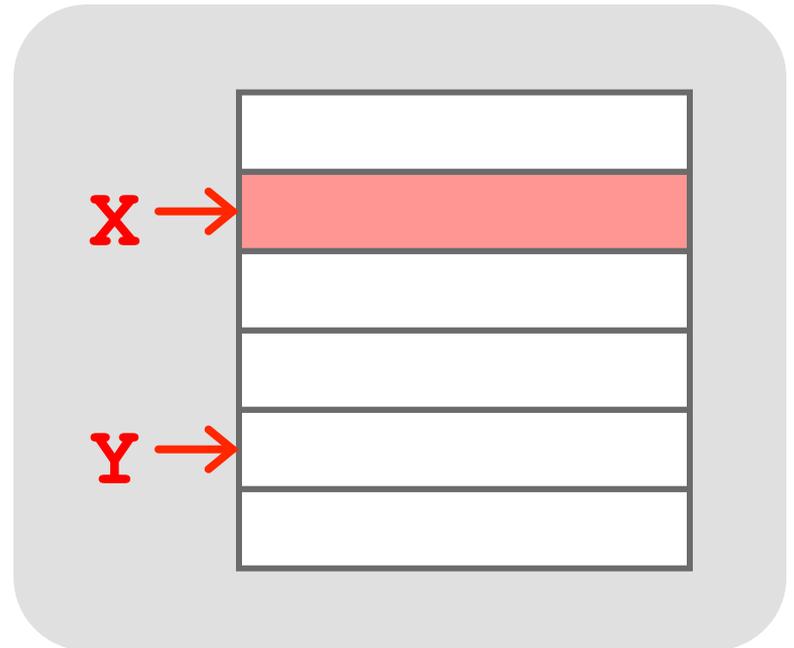
1. Avoid *cache hits*
  - Flush **X** from cache
2. Avoid *row hits* to **X**
  - Read **Y** in another row



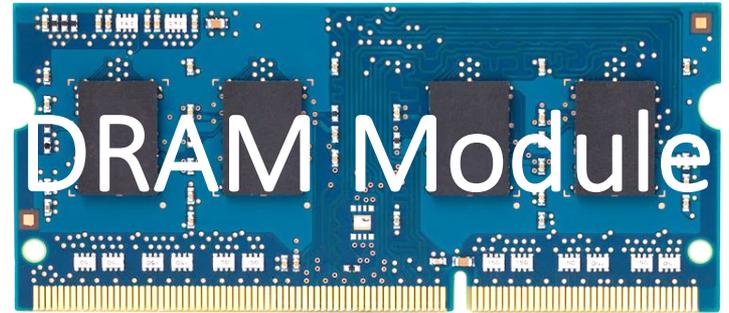
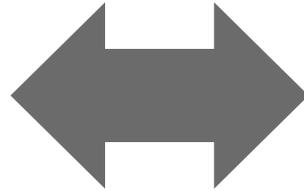
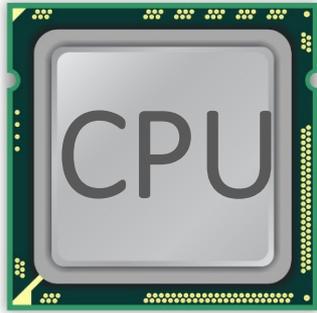
# A Simple Program Can Induce Many Errors



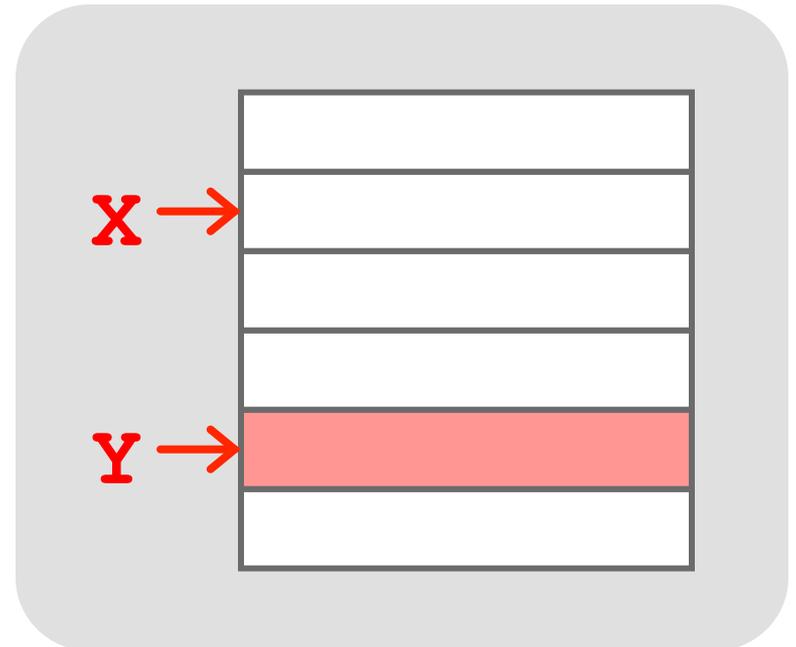
```
loop:  
  mov  (X), %eax  
  mov  (Y), %ebx  
  clflush (X)  
  clflush (Y)  
  mfence  
  jmp  loop
```



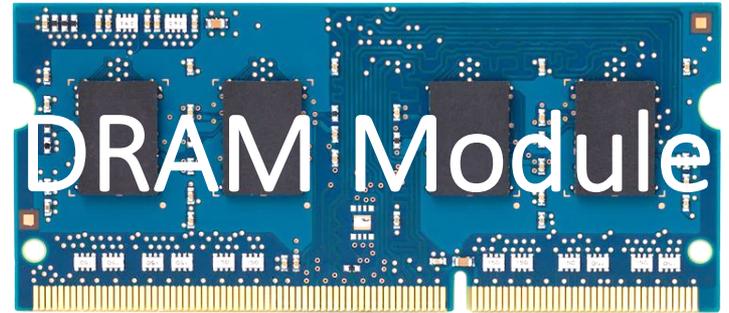
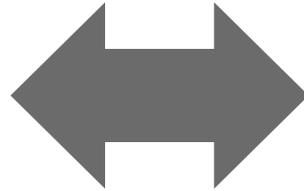
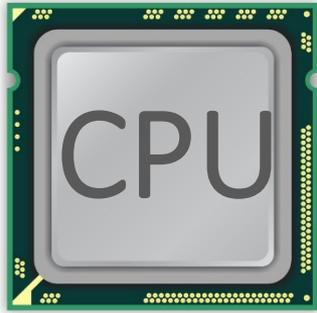
# A Simple Program Can Induce Many Errors



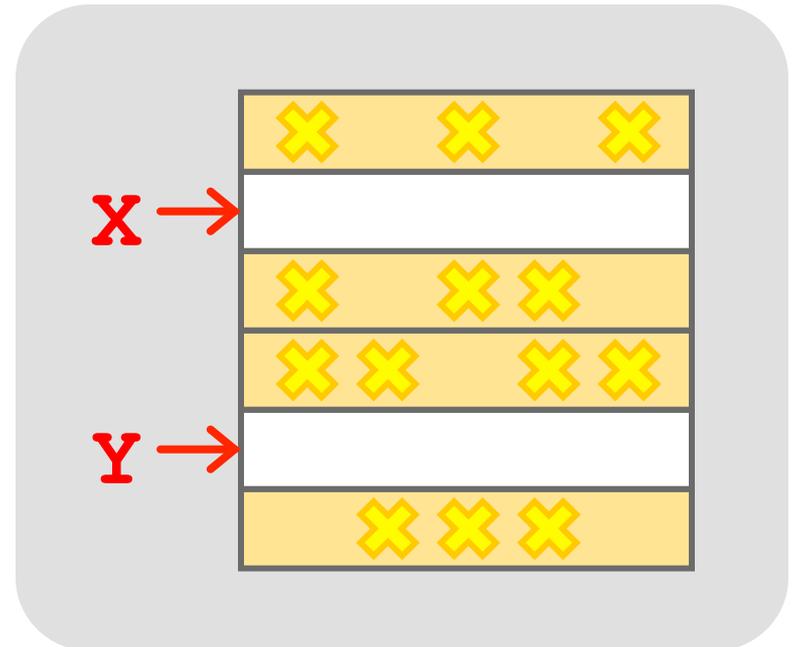
```
loop:  
  mov  (X), %eax  
  mov  (Y), %ebx  
  clflush (X)  
  clflush (Y)  
  mfence  
  jmp  loop
```



# A Simple Program Can Induce Many Errors



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



# Observed Errors in Real Systems

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

- *A real reliability & security issue*
- *In a more controlled environment, we can induce as many as **ten million** disturbance errors*

# One Can Take Over an Otherwise-Secure System

---

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

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

## Project Zero

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

News and updates from the Project Zero team at Google

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

Monday, March 9, 2015

Exploiting the DRAM rowhammer bug to gain kernel privileges

# RowHammer Security Attack Example

---

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

# Security Implications



# Rowhammer

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

# Selected Readings on RowHammer (I)

---

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

# Selected Readings on RowHammer (II)

---

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

# Selected Readings on RowHammer (III)

---

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

# More Security Implications

www.iaik.tugraz.at

Not there yet, but ...



ROOT privileges for web apps!

29

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



GATED  
COMMUNITIES

Rowhammer.js: A Remote Software-Induced Fault Attack in JavaScript

# More Security Implications

---



Drammer: Deterministic Rowhammer  
Attacks on Mobile Platforms

# More Security Implications?

---



# Root Causes of Disturbance Errors

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

*Confirmed by at least one manufacturer*

# Experimental DRAM Testing Infrastructure



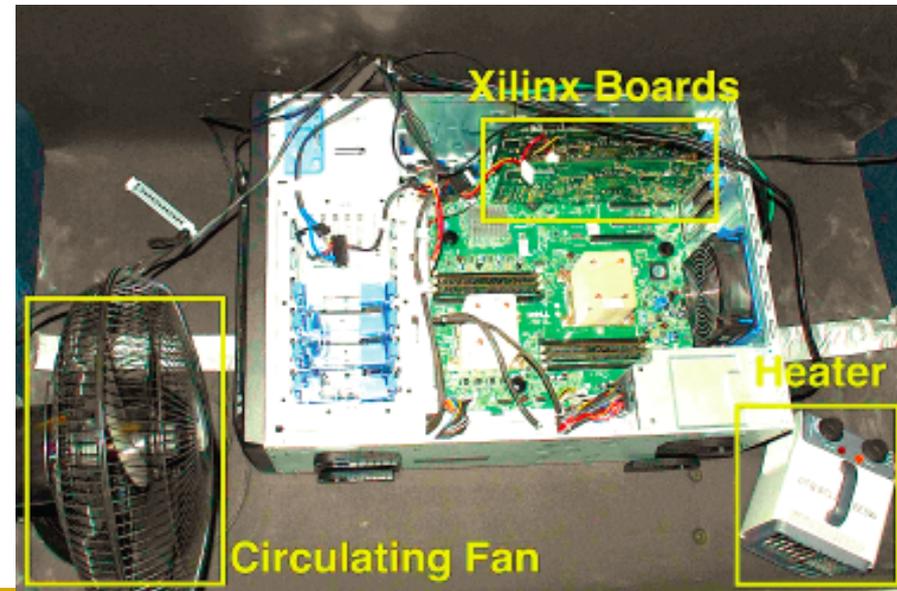
An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms (Liu et al., ISCA 2013)

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

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

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

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



# Experimental DRAM Testing Infrastructure

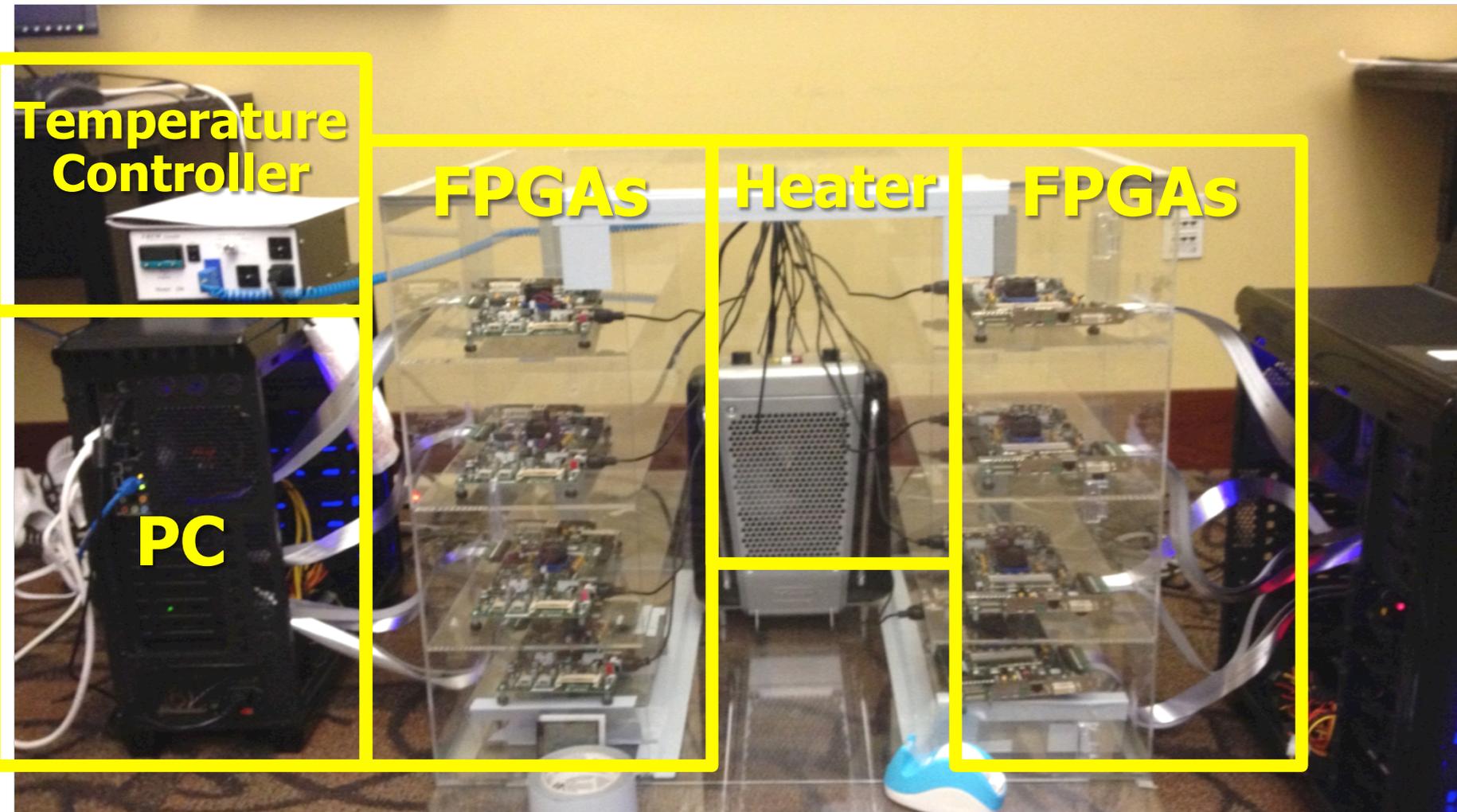
Temperature  
Controller

FPGAs

Heater

FPGAs

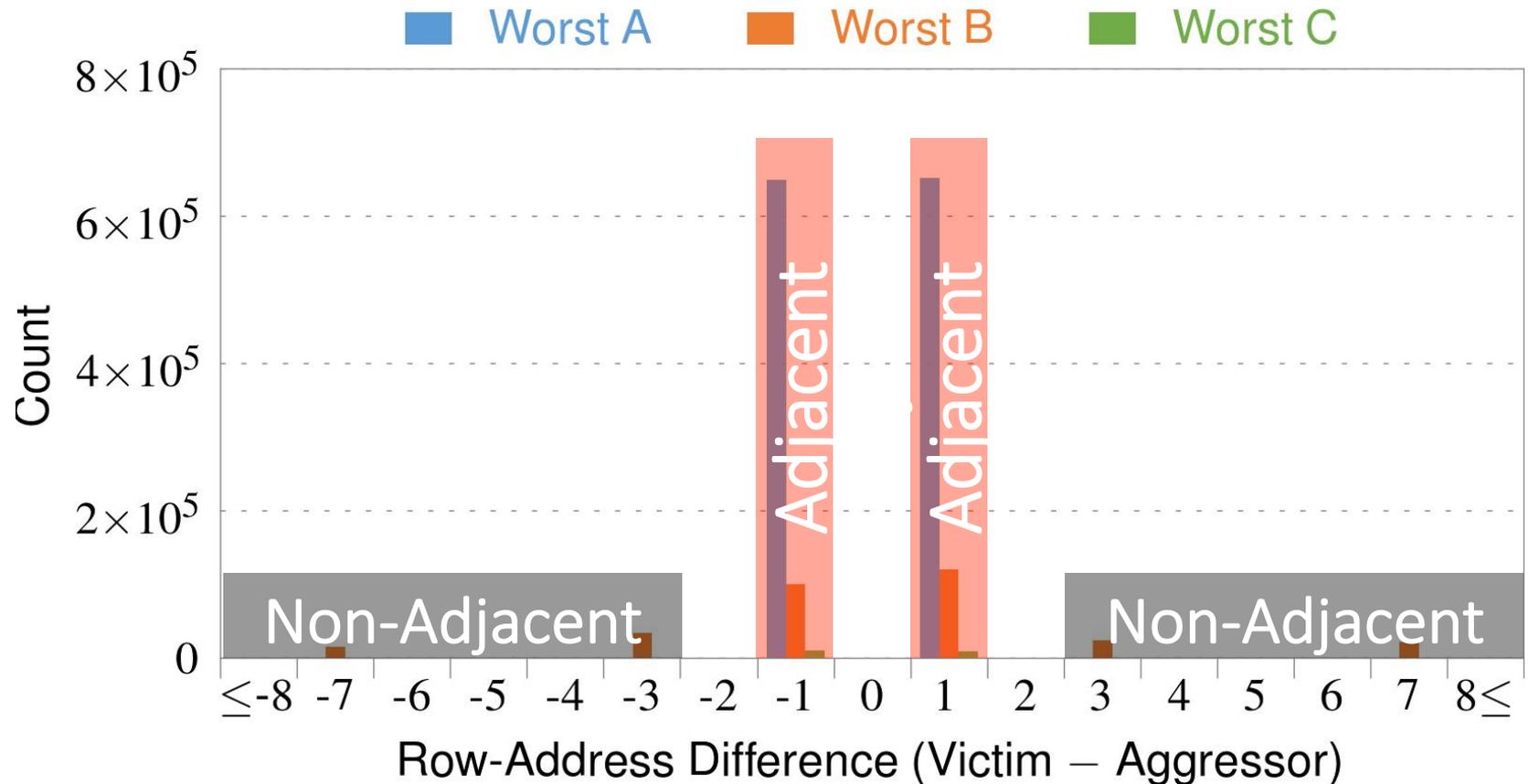
PC



# RowHammer Characterization Results

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

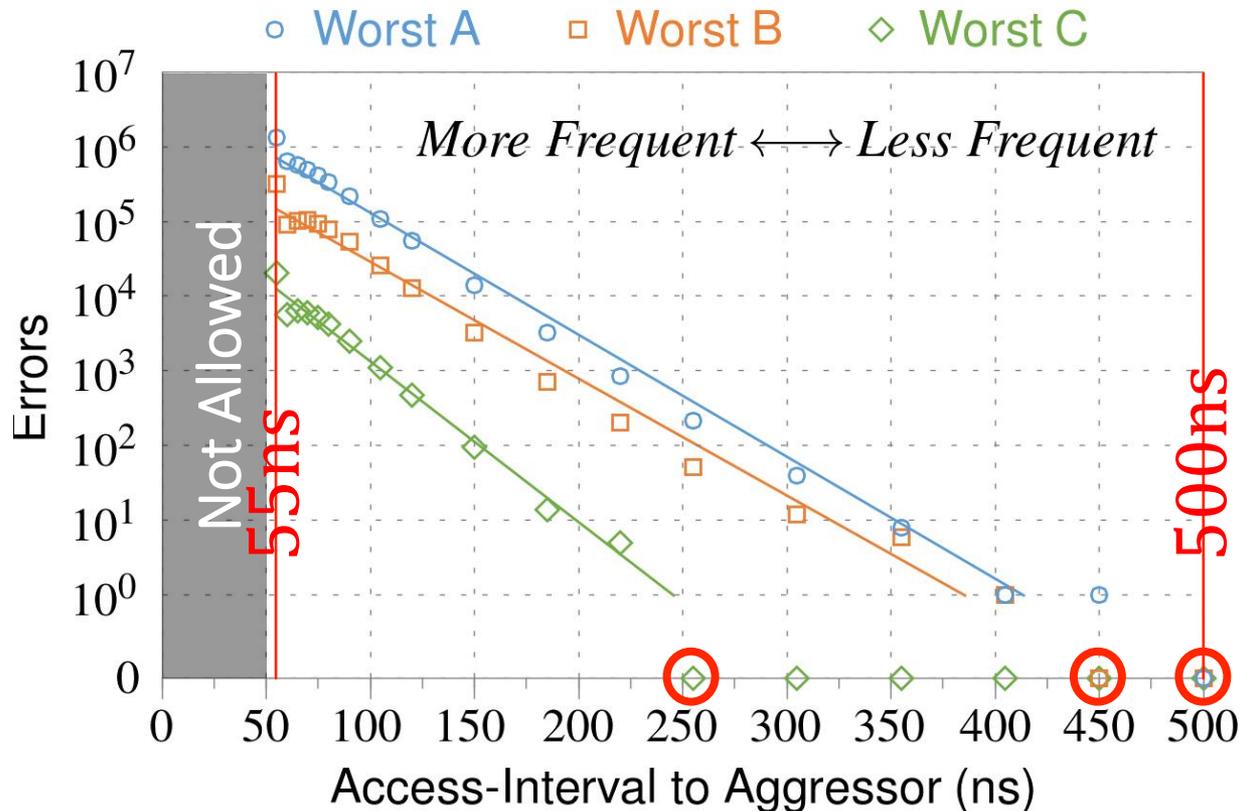
# 4. Adjacency: Aggressor & Victim



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

*Most aggressors & victims are adjacent*

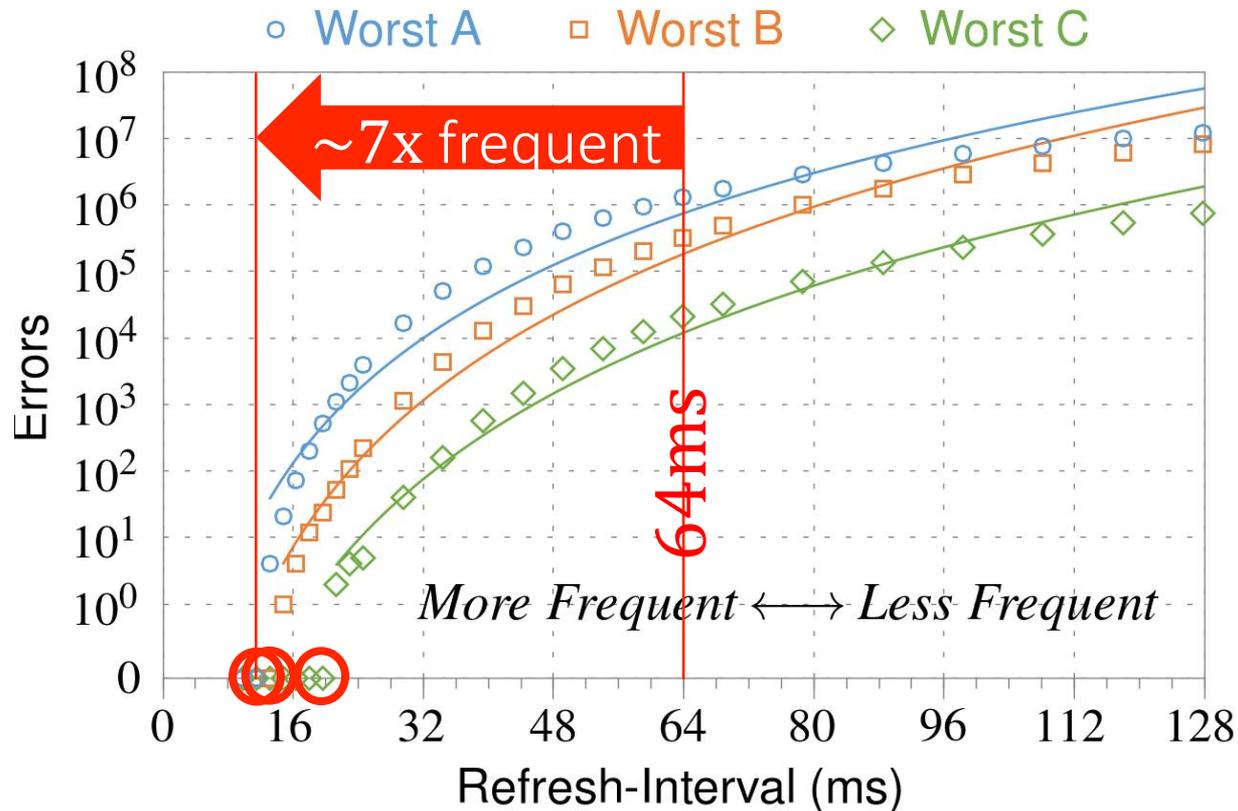
# ① Access Interval (Aggressor)



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

*Less frequent accesses → Fewer errors*

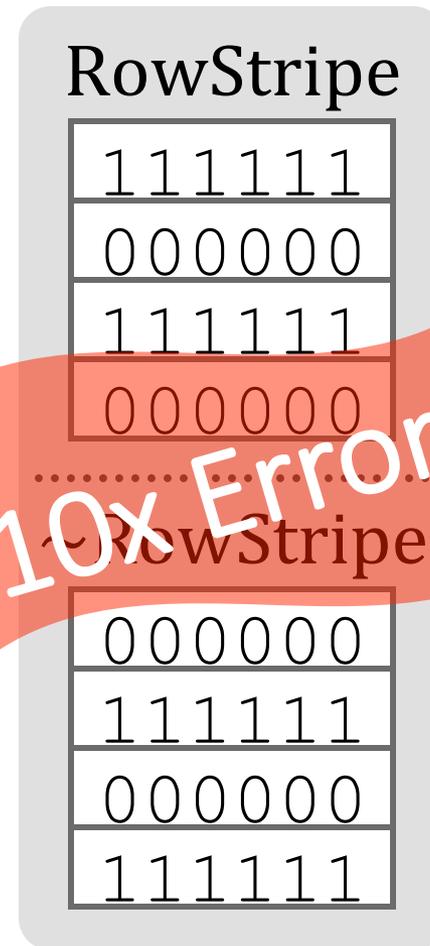
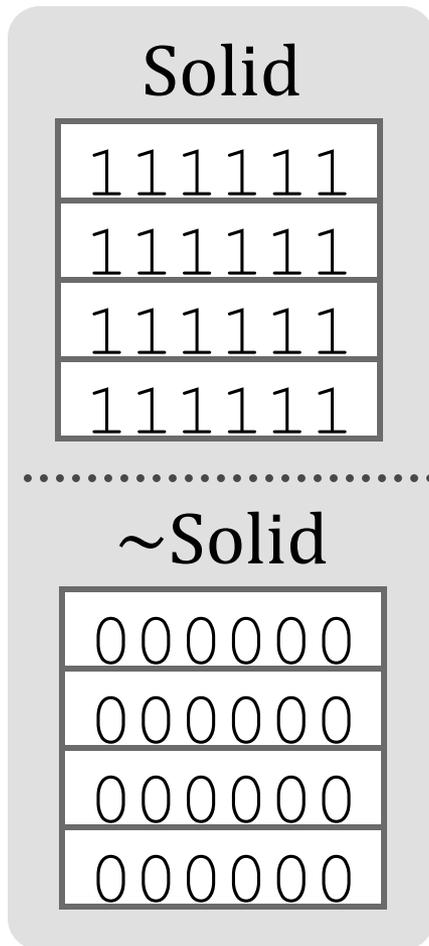
## ② Refresh Interval



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

**More frequent refreshes → Fewer errors**

# ③ Data Pattern



10x Errors

*Errors affected by data stored in other cells*

# 6. Other Results (in Paper)

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

# 6. Other Results (in Paper) cont'd

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

# Some Potential Solutions

---

- Make better DRAM chips

Cost

- Refresh frequently

Power, Performance

- Sophisticated ECC

Cost, Power

- Access counters

Cost, Power, Complexity

# Naive Solutions

## ① *Throttle accesses to same row*

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

## ② *Refresh more frequently*

- Shorten refresh-interval by  $\sim 7\text{x}$

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

# Apple's Patch for RowHammer

---

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

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

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

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

CVE-ID

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

HP and Lenovo released similar patches

---

# Our Solution

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

# Advantages of PARA

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

# Requirements for PARA

- If implemented in **DRAM chip**
  - Enough slack in timing 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.
- If implemented in **memory controller**
  - Better coordination between memory controller and DRAM
  - Memory controller should know which rows are physically adjacent

# More on RowHammer Analysis

---

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

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

<sup>1</sup>Carnegie Mellon University      <sup>2</sup>Intel Labs

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

Onur Mutlu  
ETH Zürich  
onur.mutlu@inf.ethz.ch  
<https://people.inf.ethz.ch/omutlu>

# Future of Main Memory

---

- DRAM is becoming less reliable → more vulnerable

# Large-Scale Failure Analysis of DRAM Chips

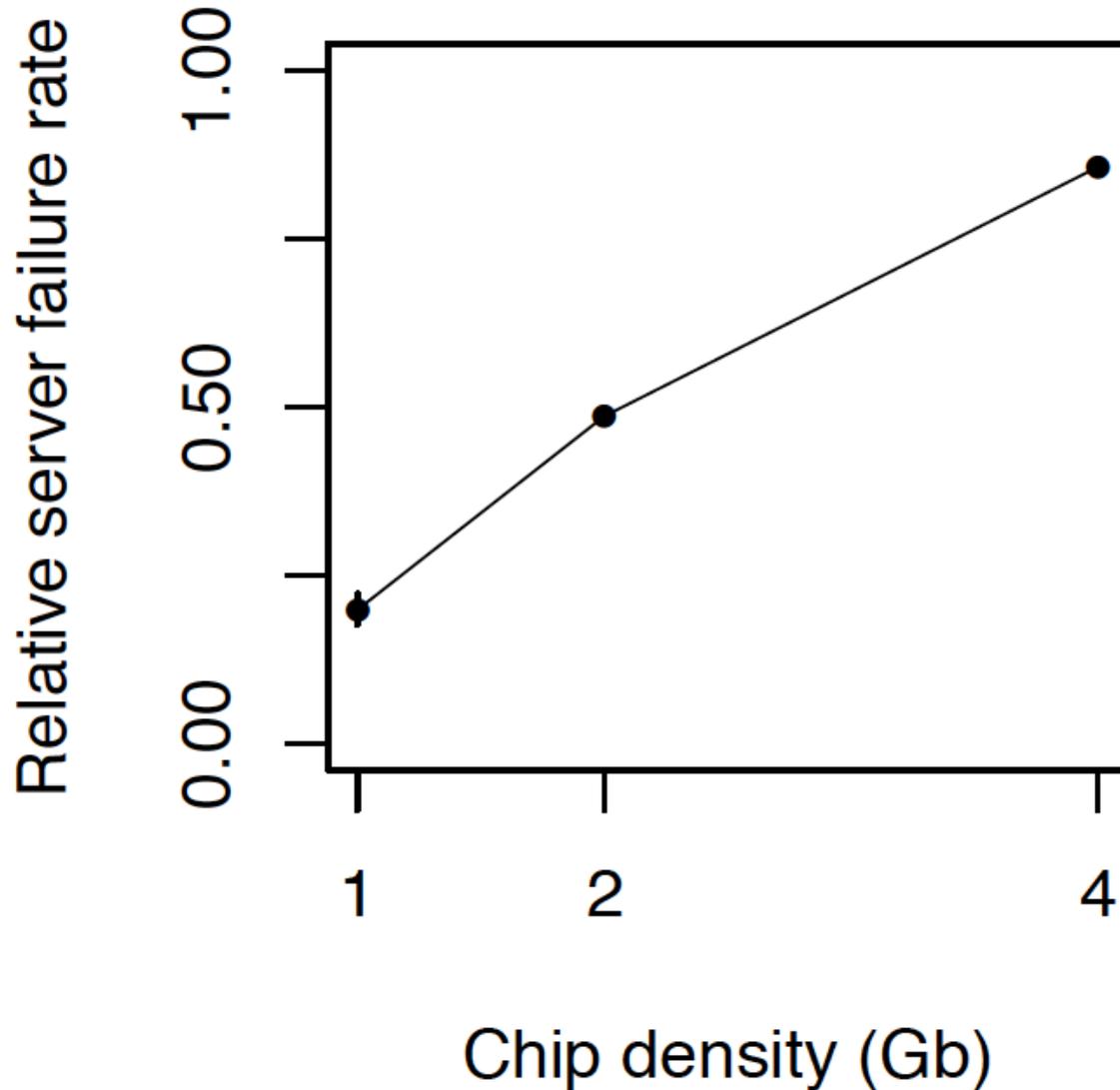
---

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

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

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

# DRAM Reliability Reducing



*Intuition:  
quadratic  
increase in  
capacity*

# Future of Main Memory

---

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

# DRAM Data Retention Time Failures

---

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

# Analysis of Data Retention Failures [ISCA'13]

---

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

Jamie Liu<sup>\*</sup>

Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213

[jamiel@alumni.cmu.edu](mailto:jamiel@alumni.cmu.edu)

Ben Jaiyen<sup>\*</sup>

Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213

[bjaiyen@alumni.cmu.edu](mailto:bjaiyen@alumni.cmu.edu)

Yoongu Kim

Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213

[yoonguk@ece.cmu.edu](mailto:yoonguk@ece.cmu.edu)

Chris Wilkerson

Intel Corporation  
2200 Mission College Blvd.  
Santa Clara, CA 95054  
[chris.wilkerson@intel.com](mailto:chris.wilkerson@intel.com)

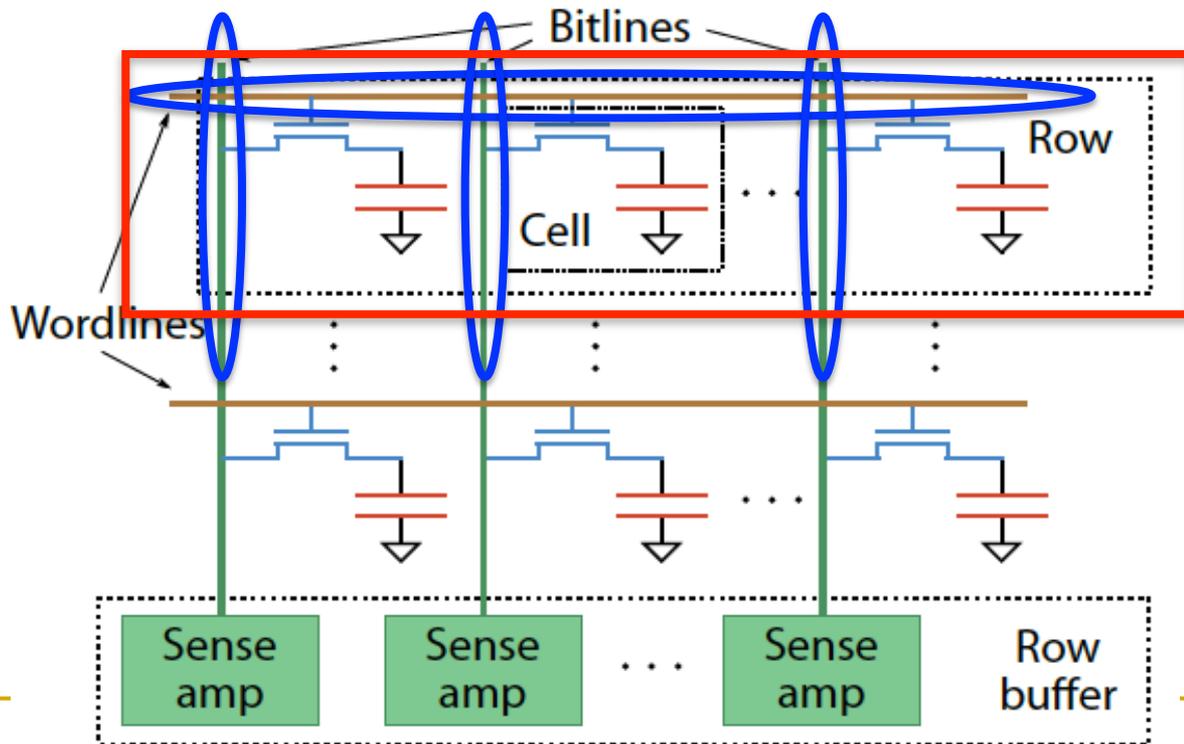
Onur Mutlu

Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213  
[onur@cmu.edu](mailto:onur@cmu.edu)



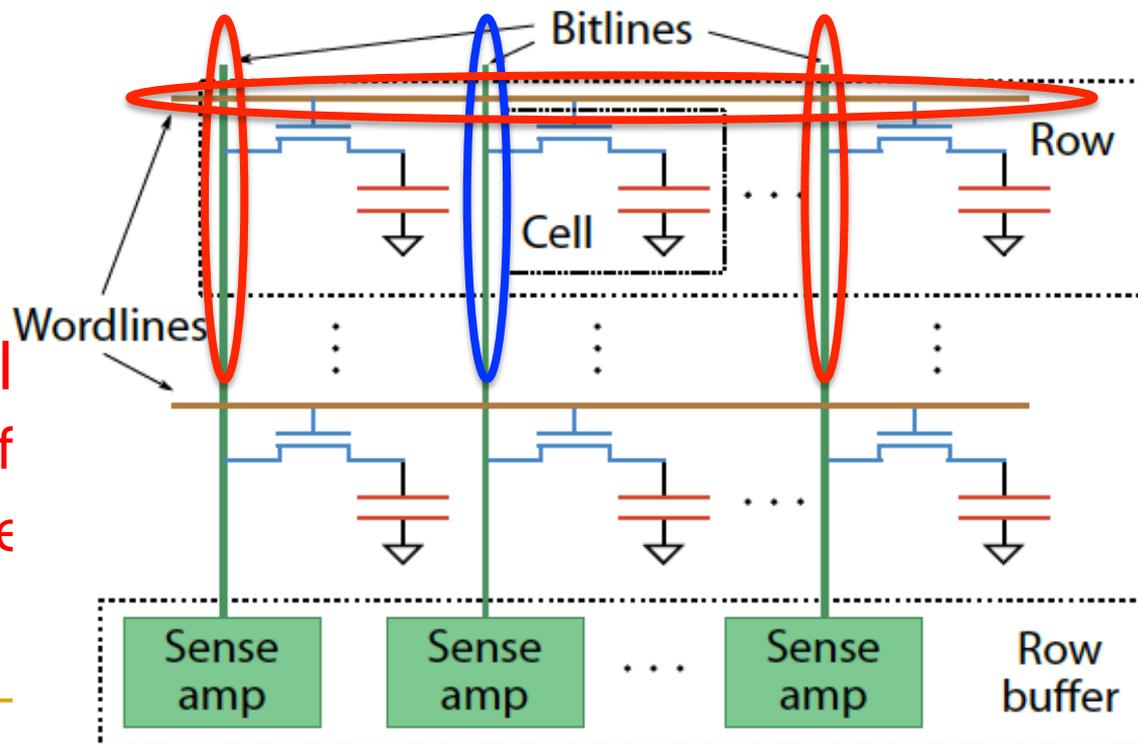
# Two Challenges to Retention Time Profiling

- **Challenge 1: Data Pattern Dependence (DPD)**
  - Retention time of a DRAM cell depends on its value and the values of cells nearby it
  - When a row is activated, all bitlines are perturbed simultaneously



# Data Pattern Dependence

- Electrical noise on the bitline affects reliable sensing of a DRAM cell
- The magnitude of this noise is affected by values of nearby cells via
  - Bitline-bitline coupling → electrical coupling between adjacent bitlines
  - Bitline-wordline coupling → electrical coupling between each bitline and the activated wordline

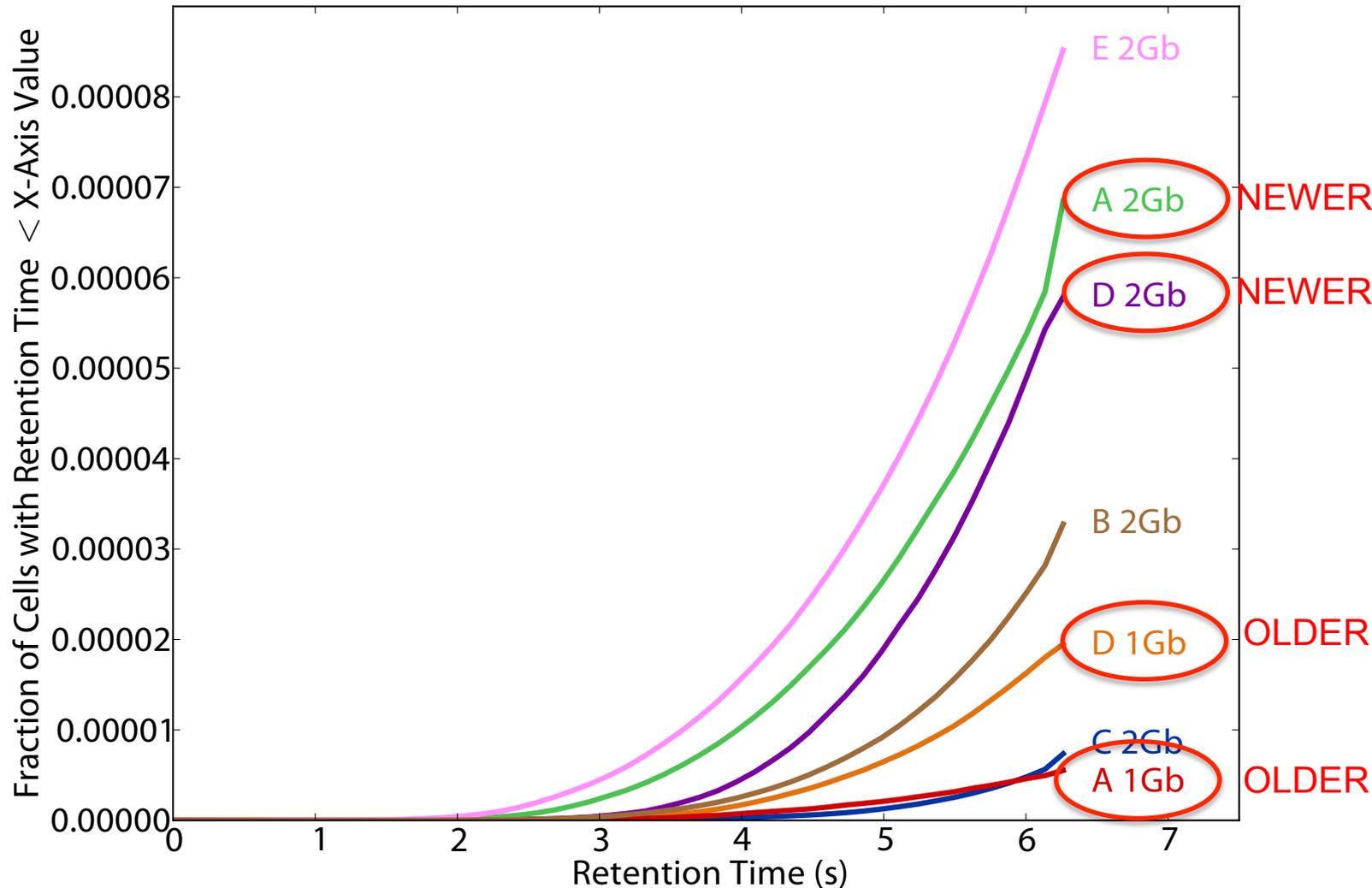


- Retention nearby cell  
→ need to f  
→ this patte

tored in  
attention time



# Modern DRAM Retention Time Distribution



**Newer device families have more weak cells than older ones  
Likely a result of technology scaling**

# Industry Is Writing Papers About It, Too

## DRAM Process Scaling Challenges

### ❖ Refresh

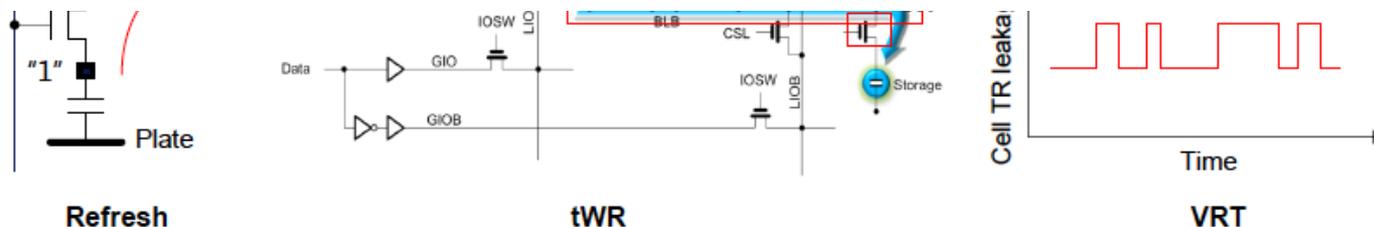
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance

THE MEMORY FORUM 2014

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

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

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



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

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

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

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

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

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

Onur Mutlu<sup>†</sup>  
onur@cmu.edu

<sup>†</sup>Carnegie Mellon University

<sup>\*</sup>Intel Labs

# Handling Variable Retention Time [DSN'15]

---

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

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

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

Samira Khan<sup>\*</sup>

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

Donghyuk Lee<sup>†‡</sup>

<sup>†</sup>Carnegie Mellon University

Onur Mutlu<sup>\*†</sup>

<sup>‡</sup>Nvidia

<sup>\*</sup>ETH Zürich

# How Do We Keep Memory Secure?

---

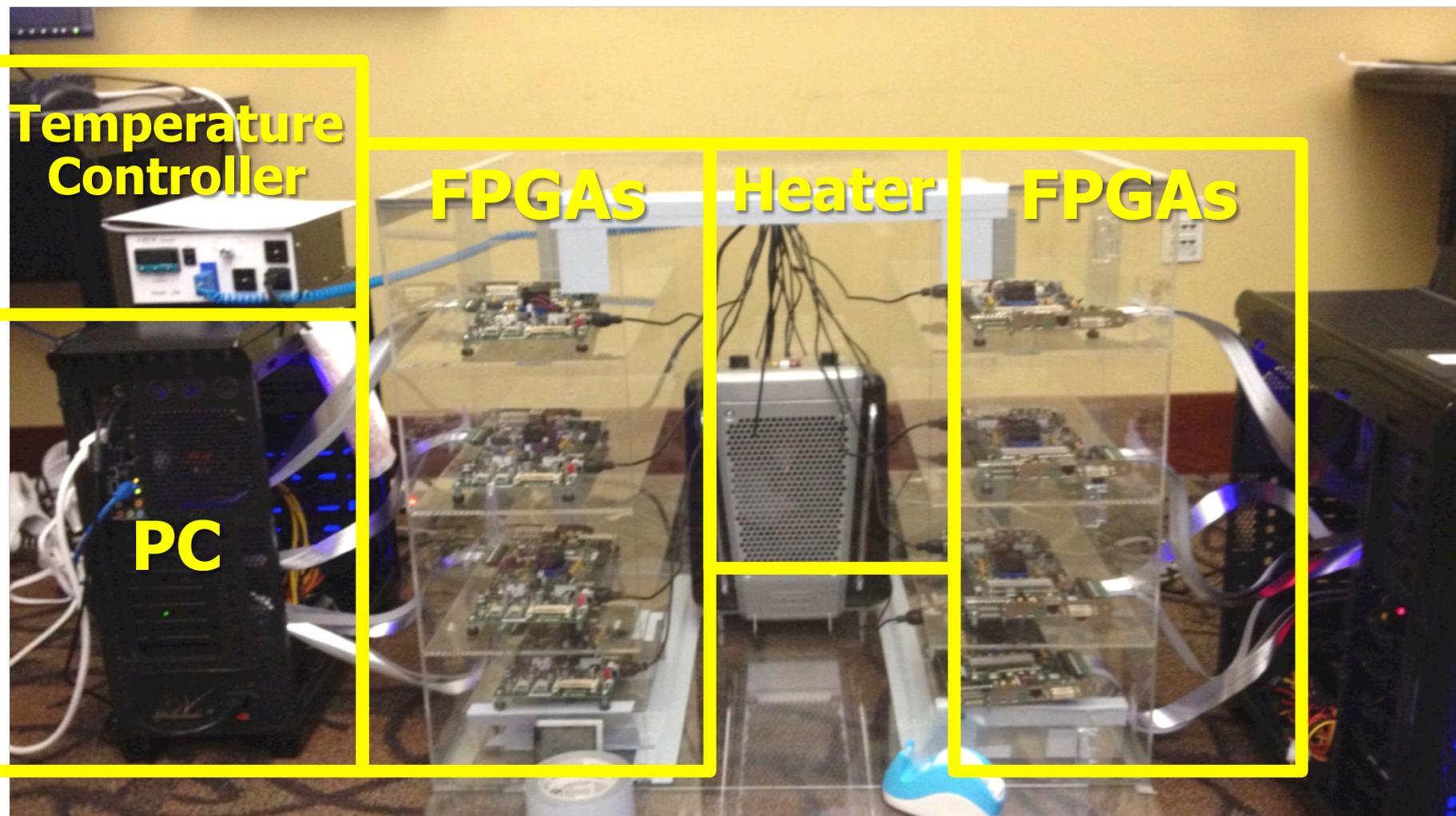
- DRAM
- Flash memory
- Emerging Technologies
  - Phase Change Memory
  - STT-MRAM
  - RRAM, memristors
  - ...

# How Do We Keep Memory Secure?

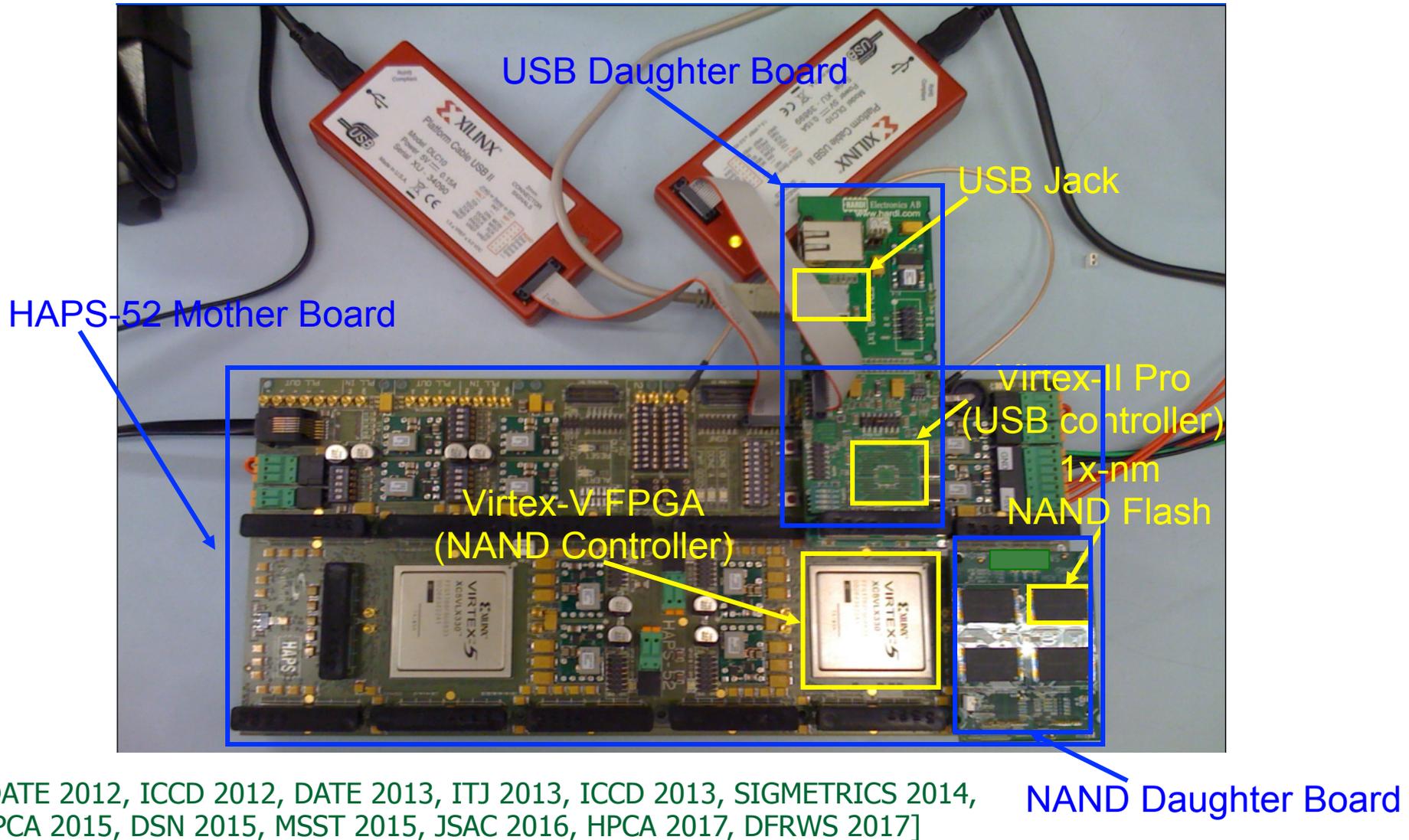
---

- **Understand:** Solid methodologies for failure modeling and discovery
  - Modeling based on real device data – small scale and large scale
- **Architect:** Principled co-architecting of system and memory
  - Good partitioning of duties across the stack
- **Design & Test:** Principled electronic design, automation, testing
  - High coverage and good interaction with system reliability methods

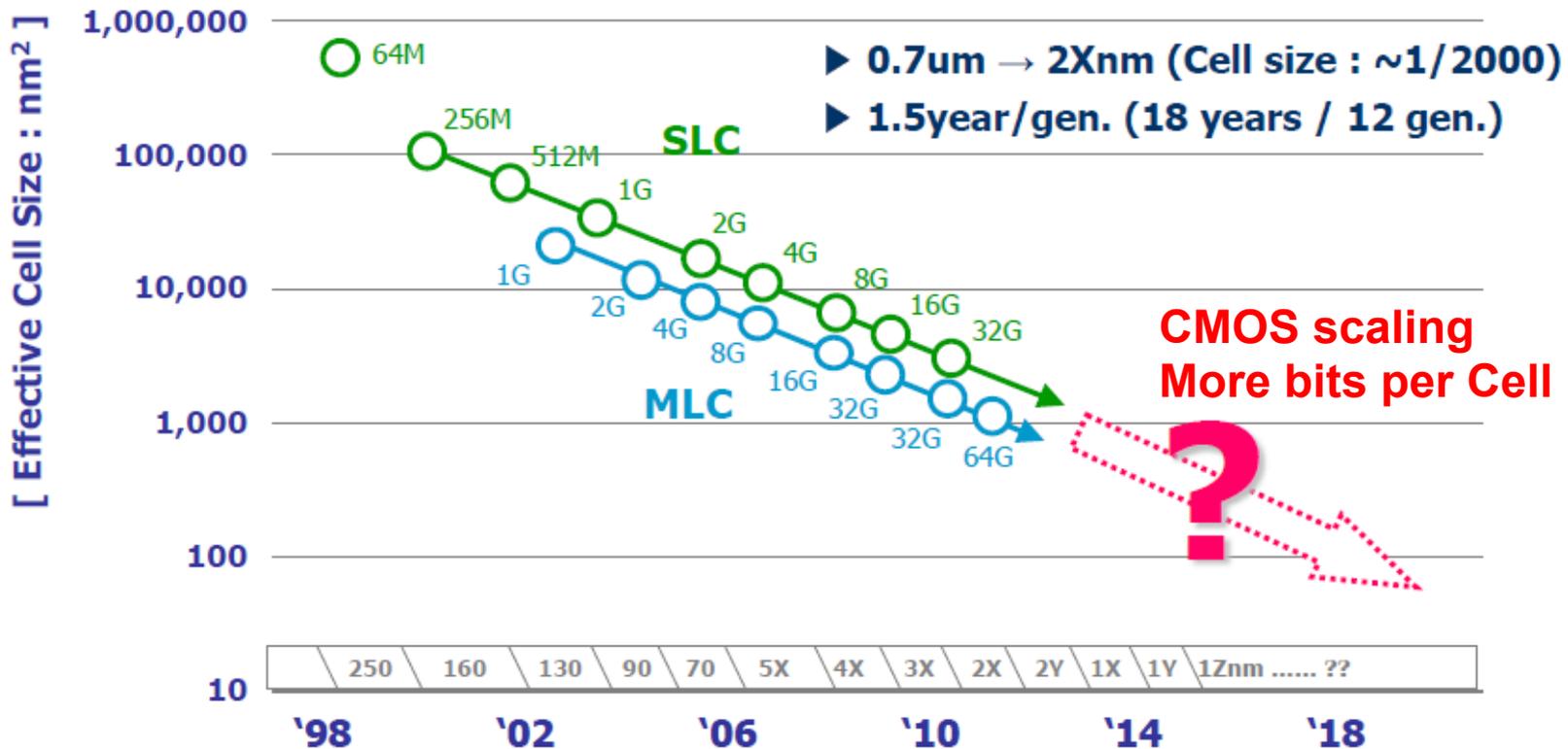
# Understand with Experiments (DRAM)



# Understand with Experiments (Flash)



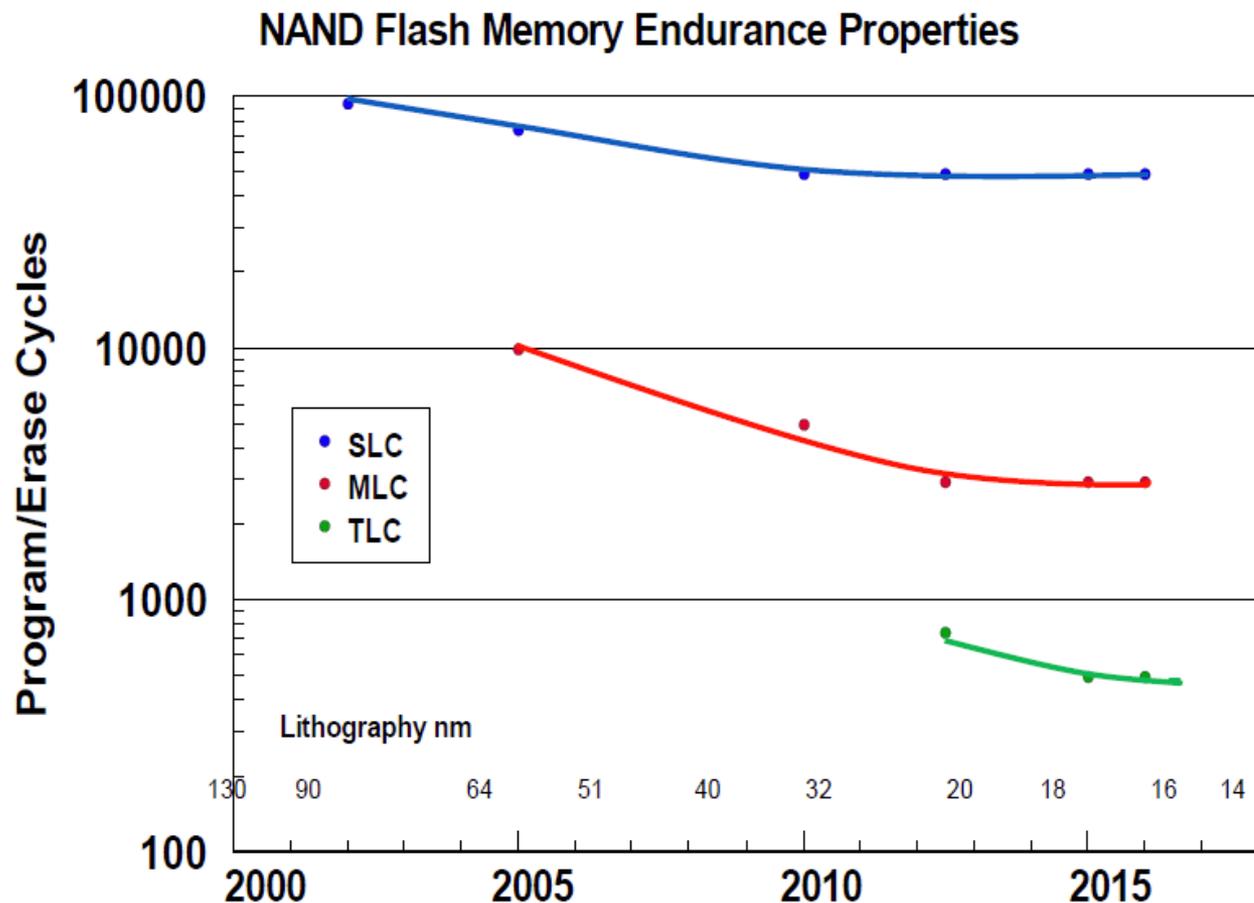
# Evolution of NAND Flash Memory



Seaung Suk Lee, "Emerging Challenges in NAND Flash Technology", Flash Summit 2011 (Hynix)

- Flash memory is widening its range of applications
  - Portable consumer devices, laptop PCs and enterprise servers

# Flash Challenges: Reliability and Endurance



- P/E cycles (provided)

A few thousand

- P/E cycles (required)

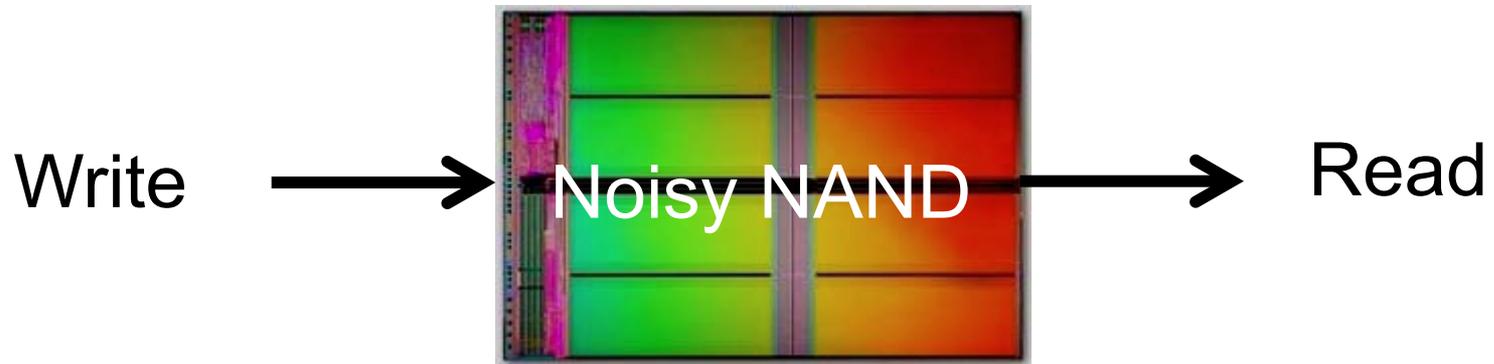
Writing the full capacity of the drive 10 times per day for 5 years (STEC)

> 50k P/E cycles

E. Grochowski et al., "Future technology challenges for NAND flash and HDD products", Flash Memory Summit 2012

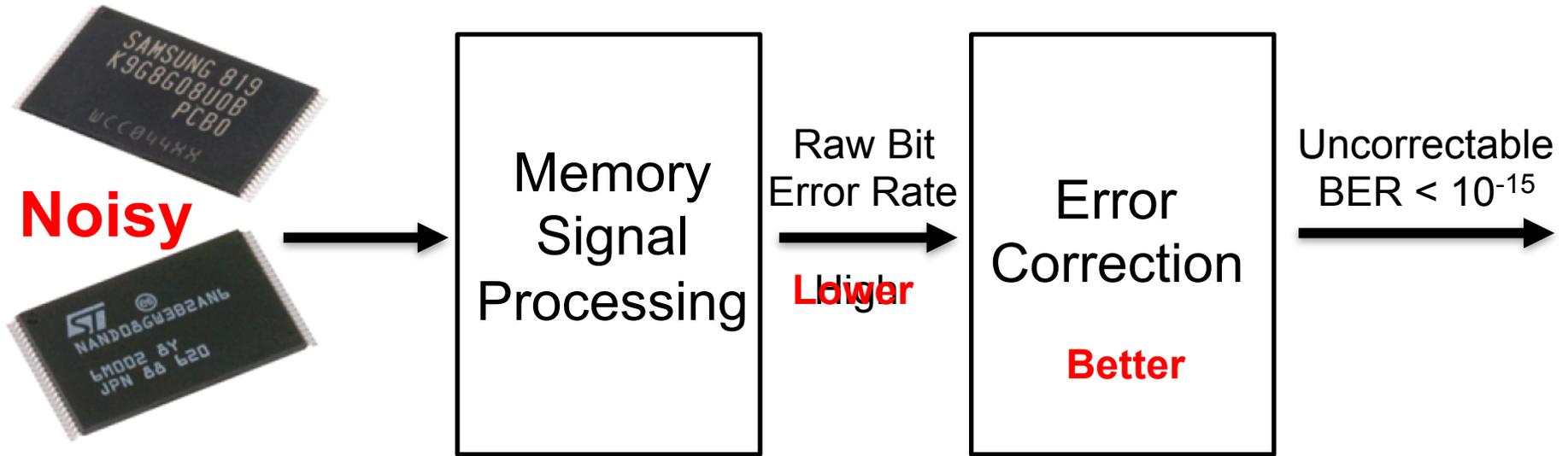
# NAND Flash Memory is Increasingly Noisy

---



# Future NAND Flash-based Storage Architecture

---



## Our Goals:

Build reliable error models for NAND flash memory

Design efficient reliability mechanisms based on the model

# NAND Flash Error Model



## Experimentally characterize and model dominant errors

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

Luo et al., "Enabling Accurate and Practical Online Flash Channel Modeling for Modern MLC NAND Flash Memory", **JSAC 2016**



Cai et al., "Threshold voltage distribution in MLC NAND Flash Memory: Characterization, Analysis, and Modeling", **DATE 2013**

Cai et al., "Vulnerabilities in MLC NAND Flash Memory Programming: Experimental Analysis, Exploits, and Mitigation Techniques", **HPCA 2017**

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

Cai et al., "Neighbor-Cell Assisted Error Correction in MLC NAND Flash Memories", **SIGMETRICS 2014**

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

Cai et al., "Flash Correct-and-Refresh: Retention-aware error management for increased flash memory lifetime", **ICCD 2012**

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

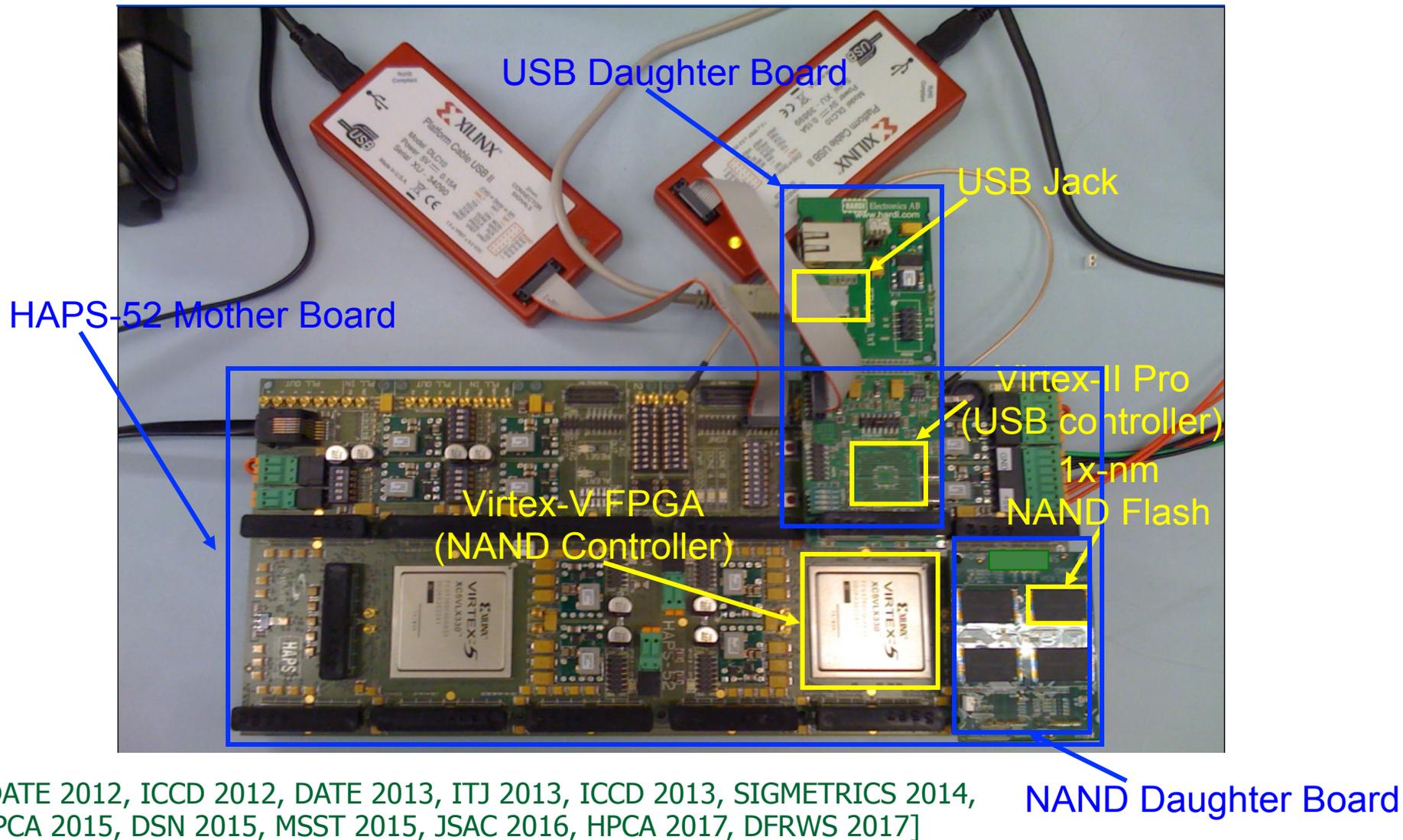
Cai et al., "Data Retention in MLC NAND Flash Memory: Characterization, Optimization and Recovery", **HPCA 2015**

# Our Goals and Approach

---

- Goals:
  - Understand error mechanisms and develop reliable predictive models for MLC NAND flash memory errors
  - Develop efficient error management techniques to mitigate errors and improve flash reliability and endurance
- Approach:
  - Solid experimental analyses of errors in real MLC NAND flash memory → drive the understanding and models
  - Understanding, models, and creativity → drive the new techniques

# Experimental Testing Platform

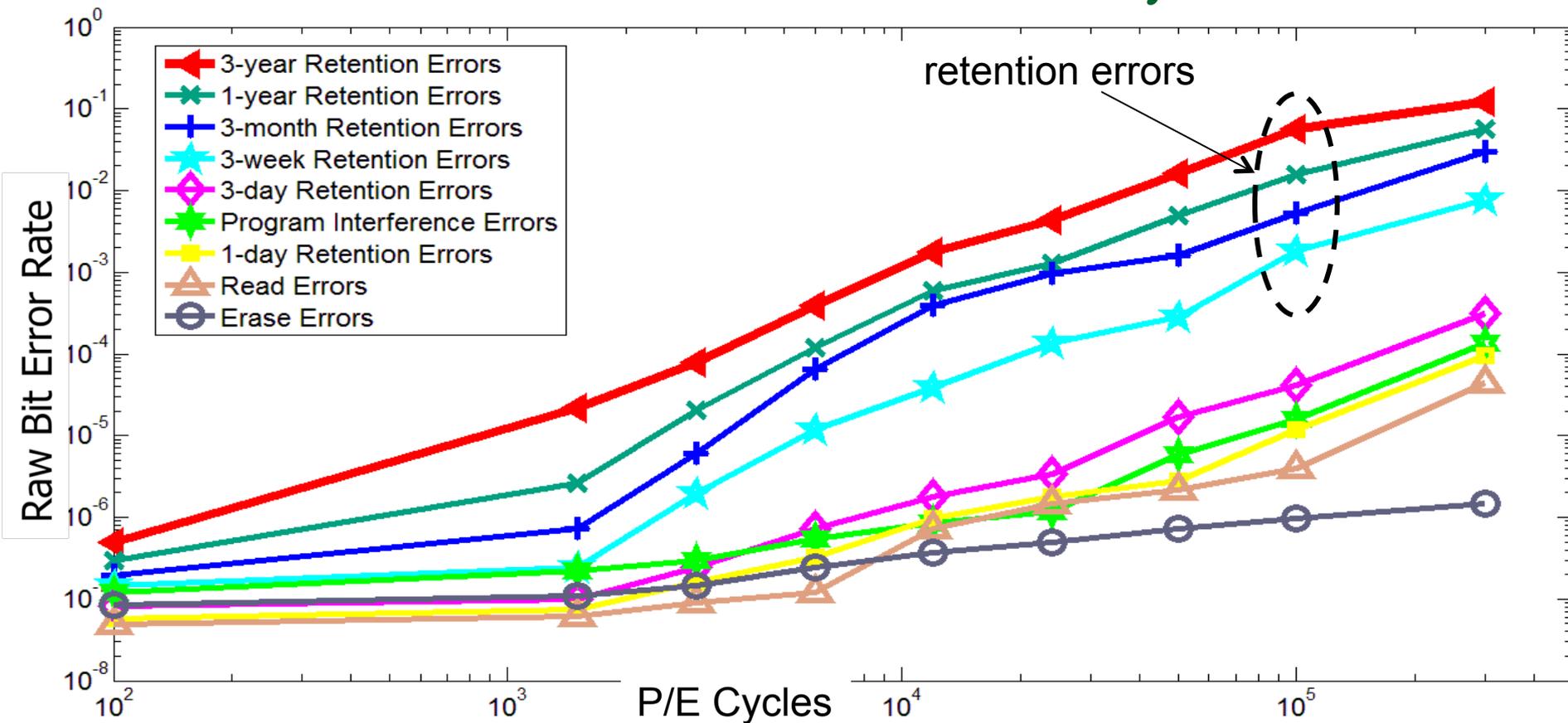


# NAND Flash Error Types

---

- Four types of errors [Cai+, DATE 2012]
- Caused by **common flash operations**
  - **Read** errors
  - **Erase** errors
  - **Program** (interference) errors
- Caused by flash **cell losing charge over time**
  - **Retention** errors
    - Whether an error happens depends on required retention time
    - Especially problematic in MLC flash because threshold voltage window to determine stored value is smaller

# Observations: Flash Error Analysis



- Raw bit error rate increases exponentially with P/E cycles
- Retention errors are dominant (>99% for 1-year ret. time)
- Retention errors increase with retention time requirement

# More on Flash Error Analysis

---

- Yu Cai, Erich F. Haratsch, Onur Mutlu, and Ken Mai, **"Error Patterns in MLC NAND Flash Memory: Measurement, Characterization, and Analysis"** *Proceedings of the Design, Automation, and Test in Europe Conference (DATE)*, Dresden, Germany, March 2012. [Slides \(ppt\)](#)

## **Error Patterns in MLC NAND Flash Memory: Measurement, Characterization, and Analysis**

Yu Cai<sup>1</sup>, Erich F. Haratsch<sup>2</sup>, Onur Mutlu<sup>1</sup> and Ken Mai<sup>1</sup>

<sup>1</sup>Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA

<sup>2</sup>LSI Corporation, 1110 American Parkway NE, Allentown, PA

<sup>1</sup>{yucai, onur, kenmai}@andrew.cmu.edu, <sup>2</sup>erich.haratsch@lsi.com

# Solution to Retention Errors

---

- Refresh periodically
- Change the period based on P/E cycle wearout
  - Refresh more often at higher P/E cycles
- Use a combination of **in-place** and **remapping-based** refresh

## Flash Correct-and-Refresh: Retention-Aware Error Management for Increased Flash Memory Lifetime

Yu Cai<sup>1</sup>, Gulay Yalcin<sup>2</sup>, Onur Mutlu<sup>1</sup>, Erich F. Haratsch<sup>3</sup>, Adrian Cristal<sup>2</sup>, Osman S. Unsal<sup>2</sup> and Ken Mai<sup>1</sup>

<sup>1</sup>DSSC, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA

<sup>2</sup>Barcelona Supercomputing Center, C/Jordi Girona 29, Barcelona, Spain

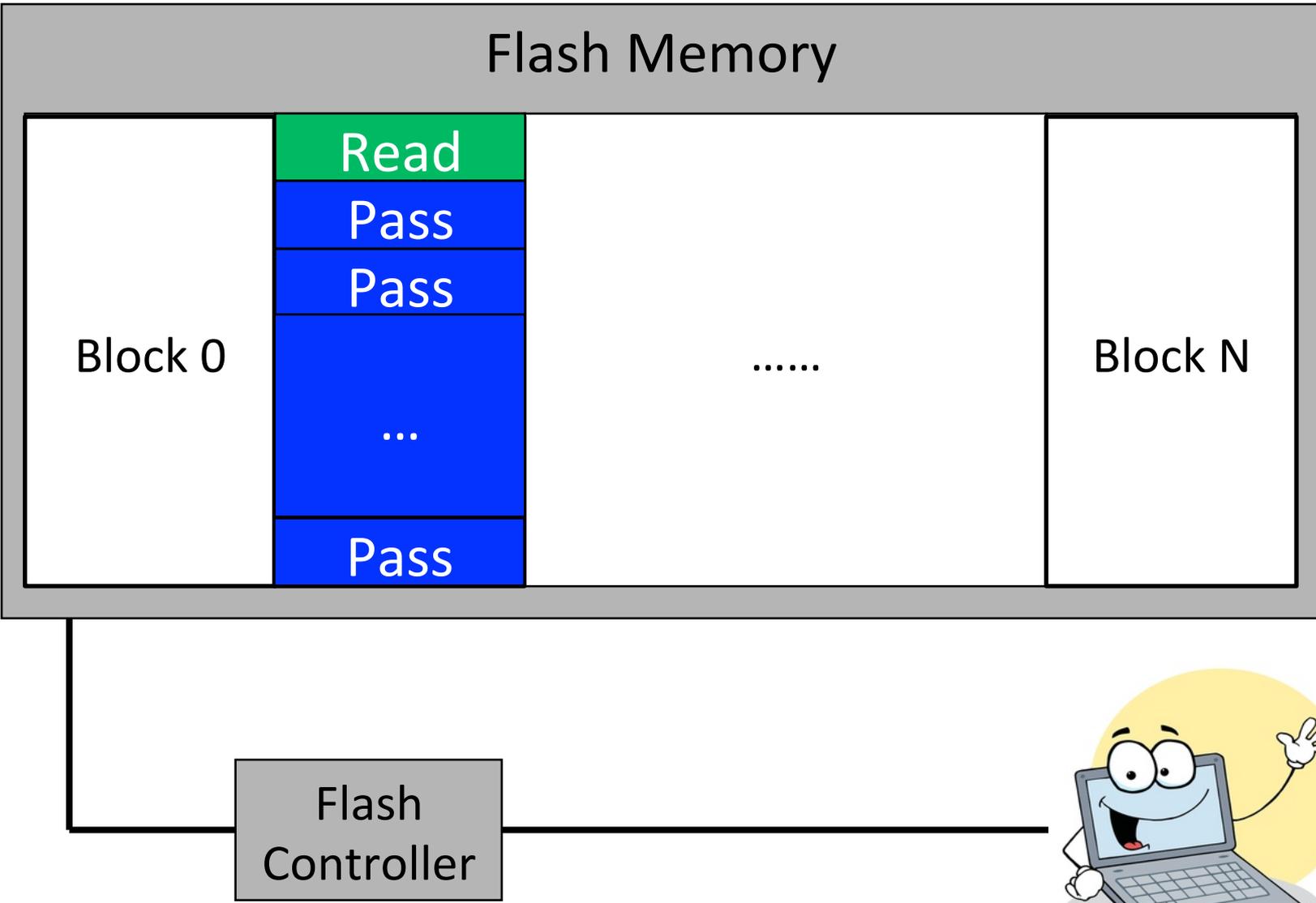
<sup>3</sup>LSI Corporation, 1110 American Parkway NE, Allentown, PA

# One Issue: Read Disturb in Flash Memory

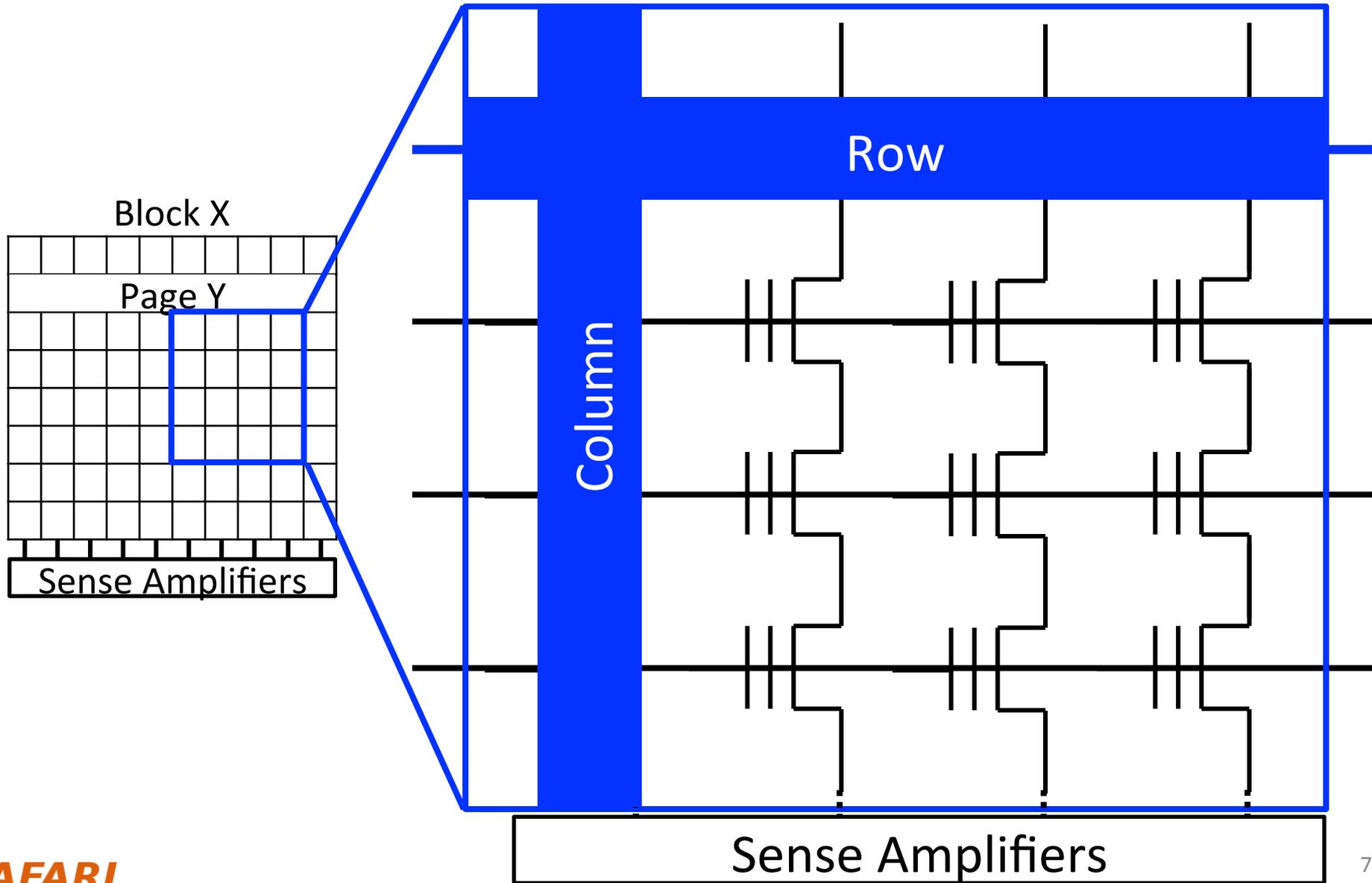
---

- All scaled memories are prone to read disturb errors

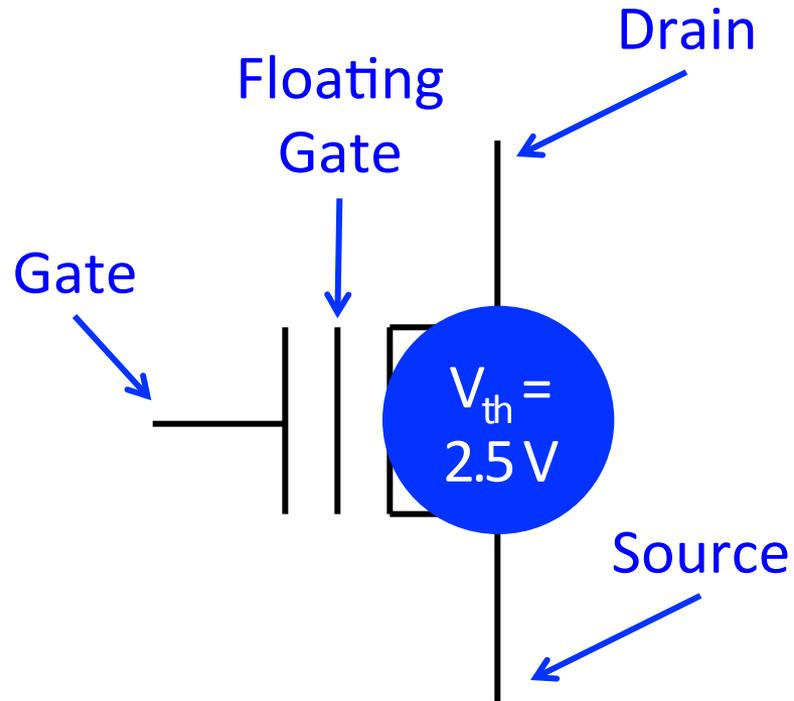
# NAND Flash Memory Background



# Flash Cell Array

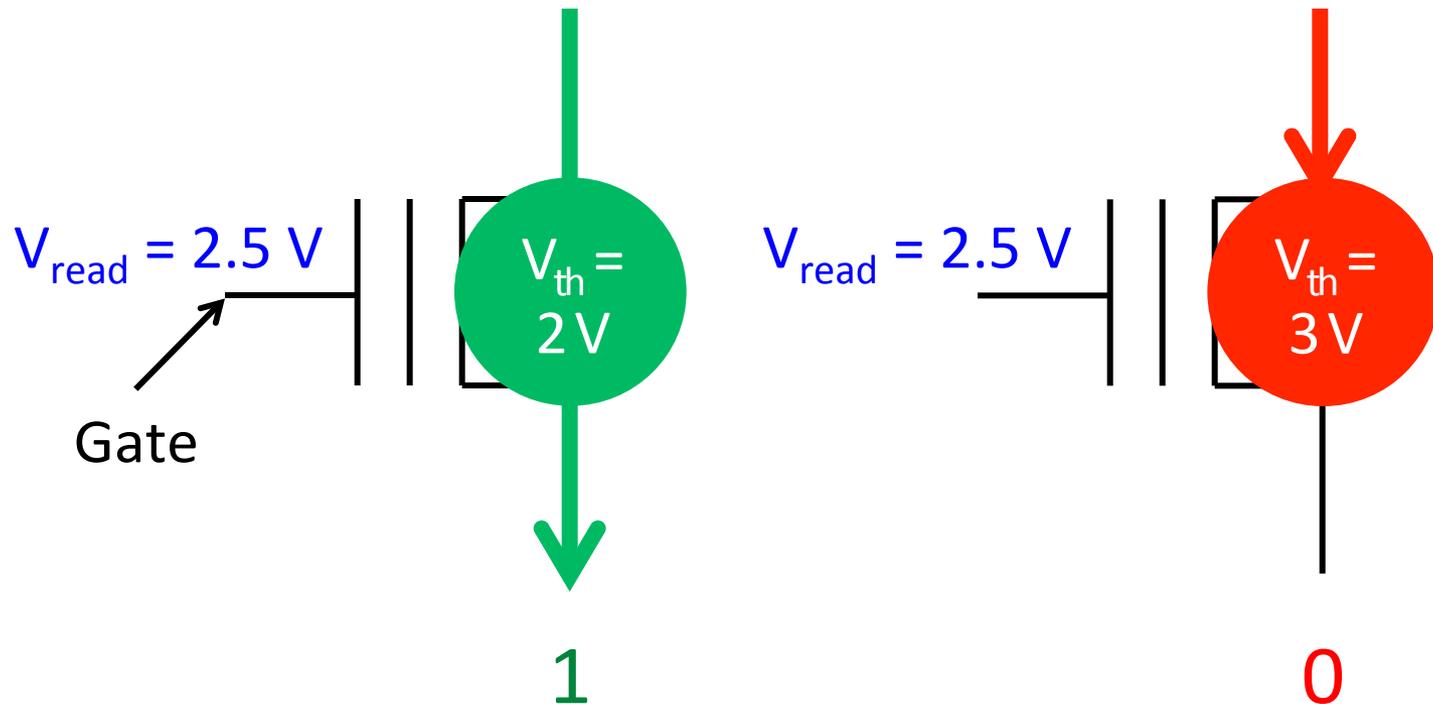


# Flash Cell

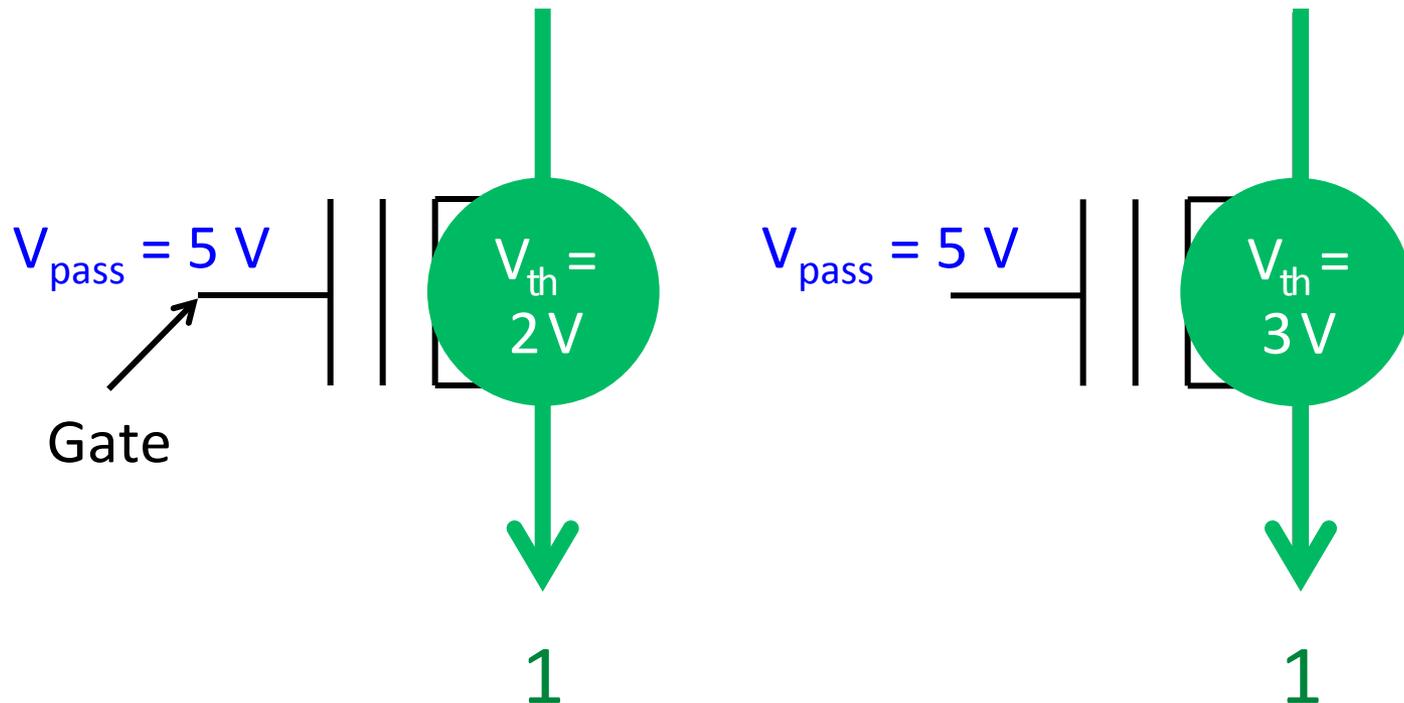


Floating Gate Transistor  
(Flash Cell)

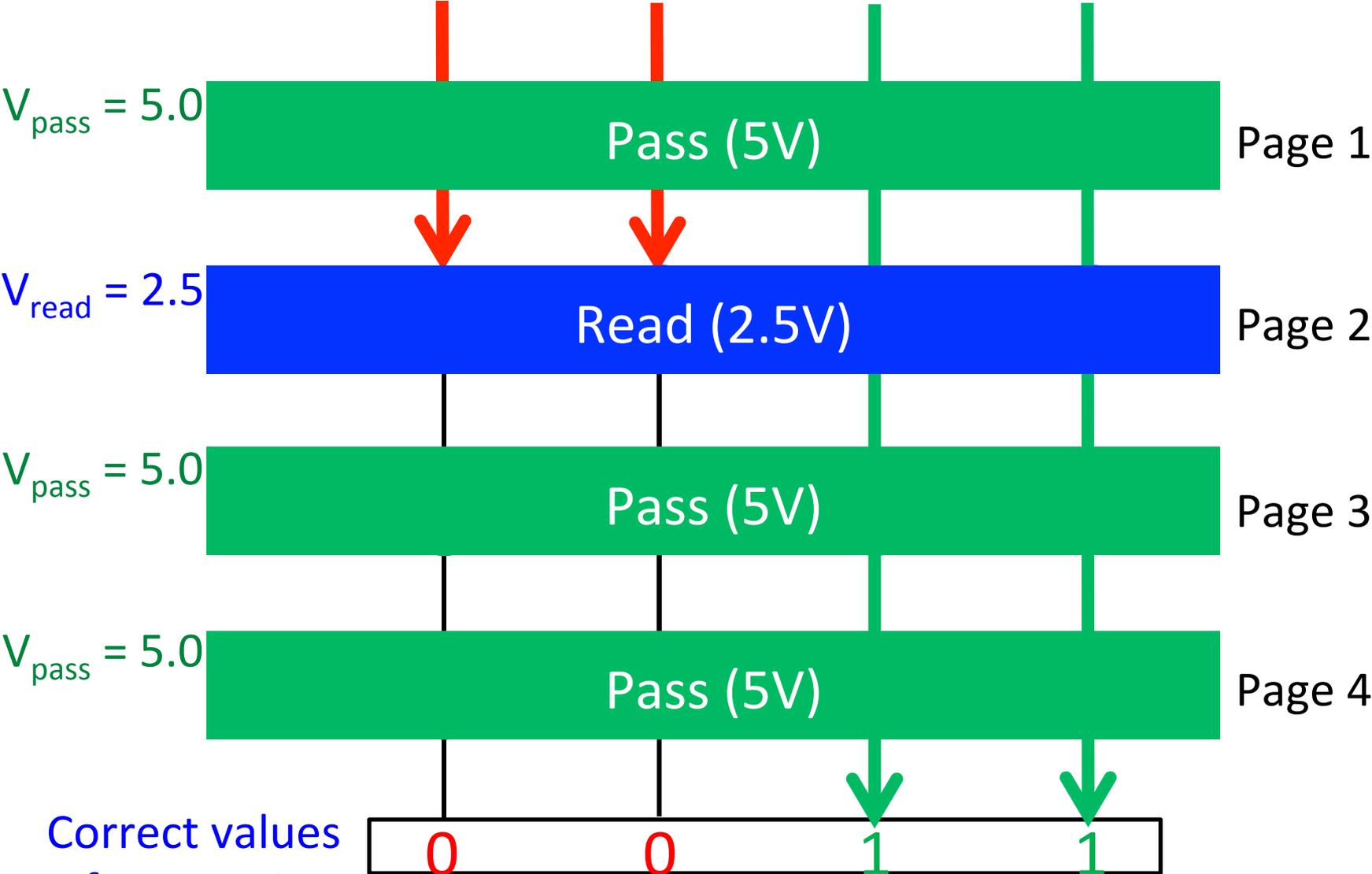
# Flash Read



# Flash Pass-Through

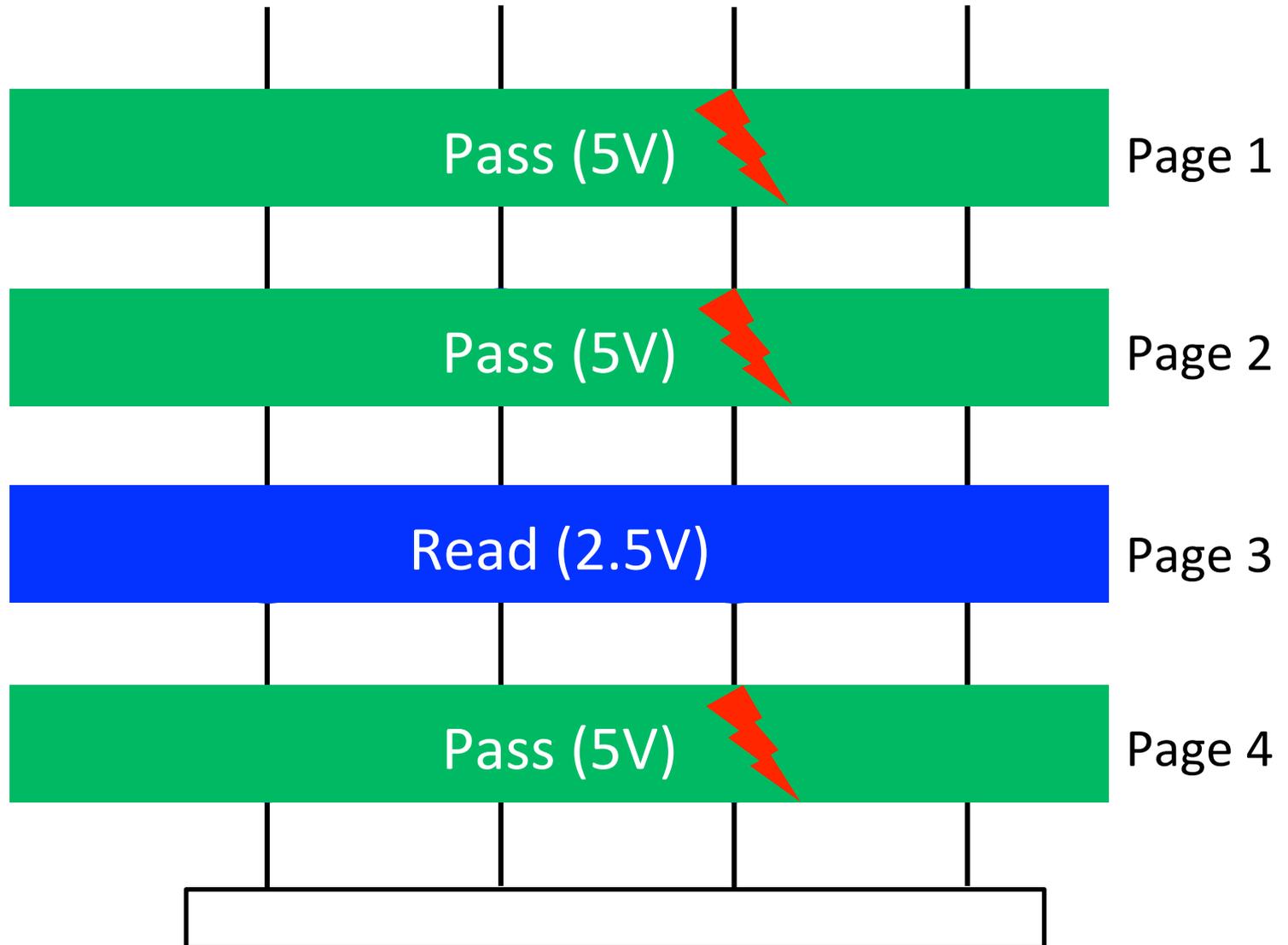


# Read from Flash Cell Array



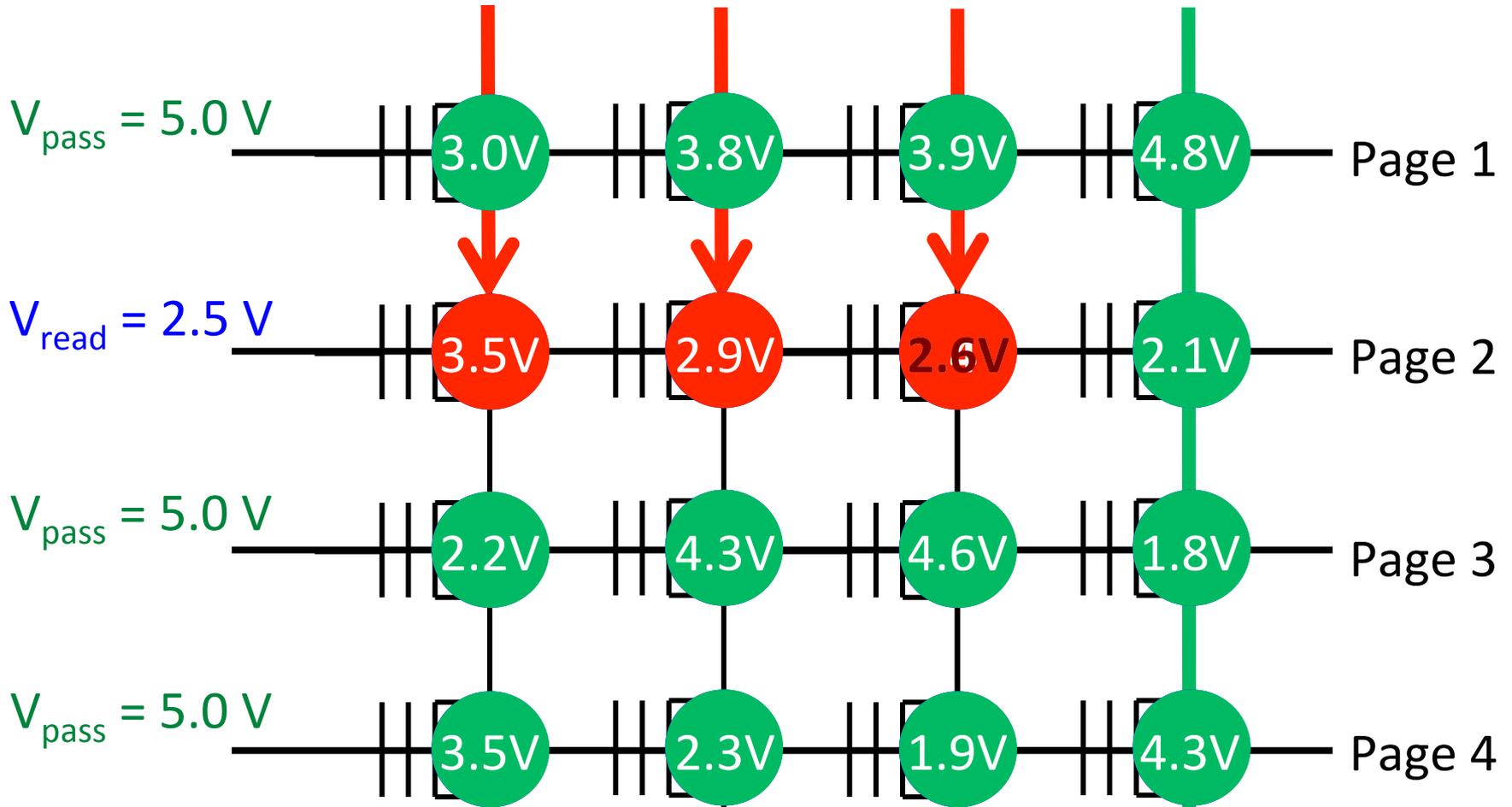
Correct values  
for page 2:

# Read Disturb Problem: “Weak Programming” Effect



**SAFARI** Repeatedly read page 3 (or any page other than page 2)

# Read Disturb Problem: "Weak Programming" Effect



Incorrect values  
from page 2:



High pass-through voltage induces "weak-programming" effect <sup>78</sup>

# Executive Summary



- **Read disturb errors** limit flash memory lifetime today
  - Apply a *high pass-through voltage* ( $V_{pass}$ ) to multiple pages on a read
  - Repeated application of  $V_{pass}$  can alter stored values in unread pages
- We **characterize read disturb** on real NAND flash chips
  - Slightly lowering  $V_{pass}$  greatly reduces read disturb errors
  - Some flash cells are more prone to read disturb
- **Technique 1: Mitigate** read disturb errors online
  - $V_{pass}$  **Tuning** dynamically finds and applies a lowered  $V_{pass}$  per block
  - Flash memory **lifetime improves by 21%**
- **Technique 2: Recover** after failure to prevent data loss
  - **Read Disturb Oriented Error Recovery** (RDR) selectively corrects cells more susceptible to read disturb errors
  - **Reduces raw bit error rate (RBER) by up to 36%**

# More on Flash Read Disturb Errors

---

- Yu Cai, Yixin Luo, Saugata Ghose, Erich F. Haratsch, Ken Mai, and Onur Mutlu,  
**"Read Disturb Errors in MLC NAND Flash Memory: Characterization and Mitigation"**  
*Proceedings of the  
45th Annual IEEE/IFIP International Conference on Dependable  
Systems and Networks (DSN), Rio de Janeiro, Brazil, June 2015.*

## Read Disturb Errors in MLC NAND Flash Memory: Characterization, Mitigation, and Recovery

Yu Cai, Yixin Luo, Saugata Ghose, Erich F. Haratsch\*, Ken Mai, Onur Mutlu  
Carnegie Mellon University, \*Seagate Technology  
[yucaicai@gmail.com](mailto:yucaicai@gmail.com), {[yixinluo](mailto:yixinluo@cmu.edu), [ghose](mailto:ghose@cmu.edu), [kenmai](mailto:kenmai@cmu.edu), [onur](mailto:onur@cmu.edu)}@cmu.edu

# Large-Scale Flash SSD Error Analysis

---

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

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

Justin Meza  
Carnegie Mellon University  
meza@cmu.edu

Qiang Wu  
Facebook, Inc.  
qw@fb.com

Sanjeev Kumar  
Facebook, Inc.  
skumar@fb.com

Onur Mutlu  
Carnegie Mellon University  
onur@cmu.edu

# Another Time: NAND Flash Vulnerabilities

---

- Onur Mutlu,  
**"Error Analysis and Management for MLC NAND Flash Memory"**  
*Technical talk at Flash Memory Summit 2014 (FMS), Santa Clara, CA, August 2014. Slides (ppt) (pdf)*

Cai+, "Error Patterns in MLC NAND Flash Memory: Measurement, Characterization, and Analysis," DATE 2012.

Cai+, "Flash Correct-and-Refresh: Retention-Aware Error Management for Increased Flash Memory Lifetime," ICCD 2012.

Cai+, "Threshold Voltage Distribution in MLC NAND Flash Memory: Characterization, Analysis and Modeling," DATE 2013.

Cai+, "Error Analysis and Retention-Aware Error Management for NAND Flash Memory," Intel Technology Journal 2013.

Cai+, "Program Interference in MLC NAND Flash Memory: Characterization, Modeling, and Mitigation," ICCD 2013.

Cai+, "Neighbor-Cell Assisted Error Correction for MLC NAND Flash Memories," SIGMETRICS 2014.

Cai+, "Data Retention in MLC NAND Flash Memory: Characterization, Optimization and Recovery," HPCA 2015.

Cai+, "Read Disturb Errors in MLC NAND Flash Memory: Characterization and Mitigation," DSN 2015.

Luo+, "WARM: Improving NAND Flash Memory Lifetime with Write-hotness Aware Retention Management," MSST 2015.

Meza+, "A Large-Scale Study of Flash Memory Errors in the Field," SIGMETRICS 2015.

Luo+, "Enabling Accurate and Practical Online Flash Channel Modeling for Modern MLC NAND Flash Memory," IEEE JSAC 2016.

Cai+, "Vulnerabilities in MLC NAND Flash Memory Programming: Experimental Analysis, Exploits, and Mitigation Techniques," HPCA 2017.

Fukami+, "Improving the Reliability of Chip-Off Forensic Analysis of NAND Flash Memory Devices," DFRWS EU 2017.

# Flash Memory Programming Vulnerabilities

---

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

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

[https://people.inf.ethz.ch/omutlu/pub/flash-memory-programming-vulnerabilities\\_hpca17.pdf](https://people.inf.ethz.ch/omutlu/pub/flash-memory-programming-vulnerabilities_hpca17.pdf)

# Summary

---

- **Memory reliability is reducing**
- Reliability issues open up security vulnerabilities
  - Very hard to defend against
- Rowhammer is an example
  - Its implications on system security research are tremendous & exciting
  
- **Good news: We have a lot more to do.**
- **Understand: Solid methodologies for failure modeling and discovery**
  - Modeling based on real device data – small scale and large scale
- **Architect: Principled co-architecting of system and memory**
  - Good partitioning of duties across the stack
- **Design & Test: Principled electronic design, automation, testing**
  - High coverage and good interaction with system reliability methods

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

Onur Mutlu

[onur.mutlu@inf.ethz.ch](mailto:onur.mutlu@inf.ethz.ch)

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

April 21, 2017

CFAED Workshop on Resilient Systems Keynote



**SAFARI**

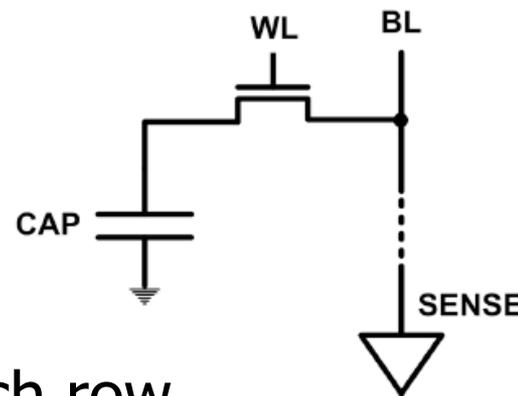
**ETH** zürich

# More on DRAM Data Retention

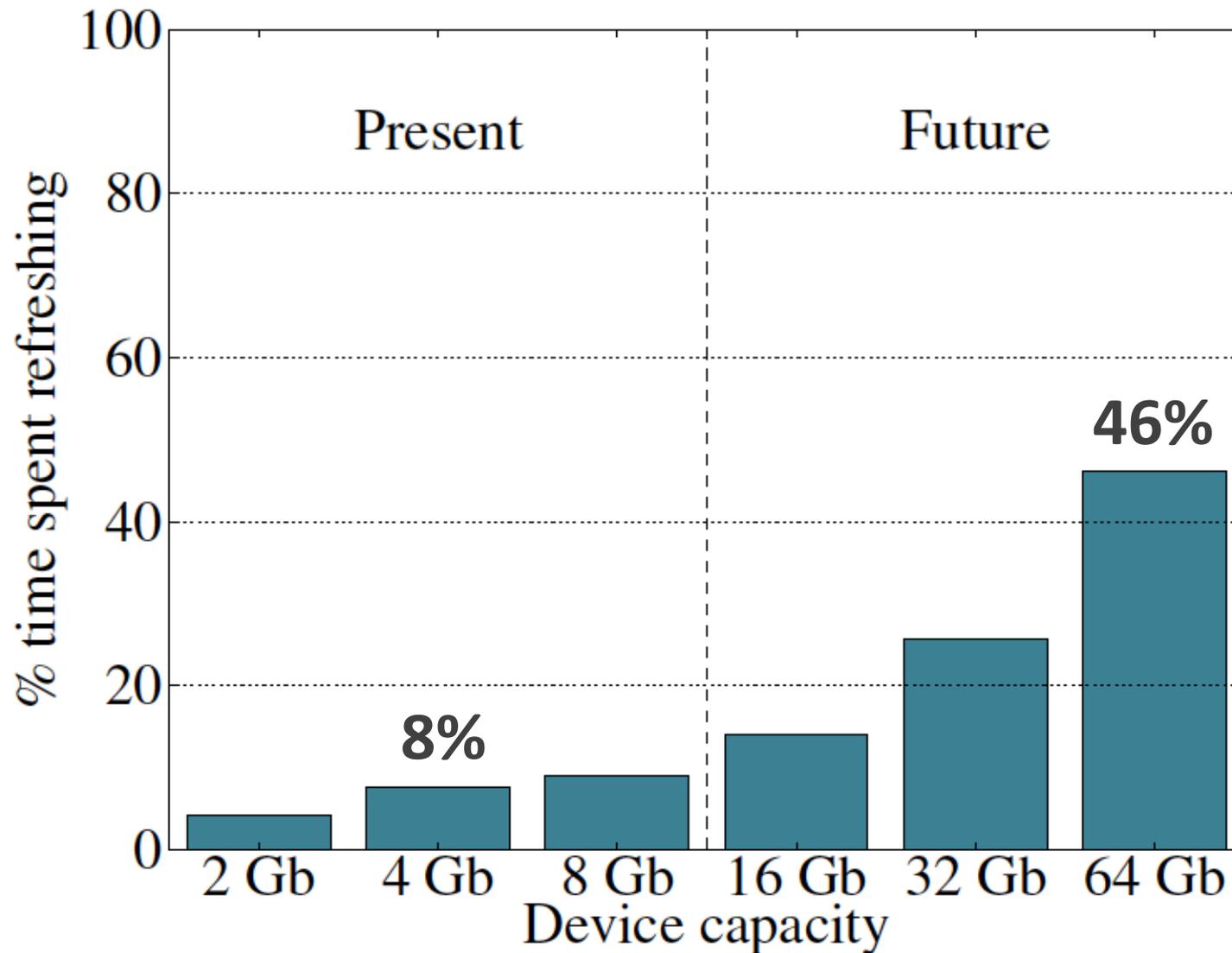
# DRAM Refresh

---

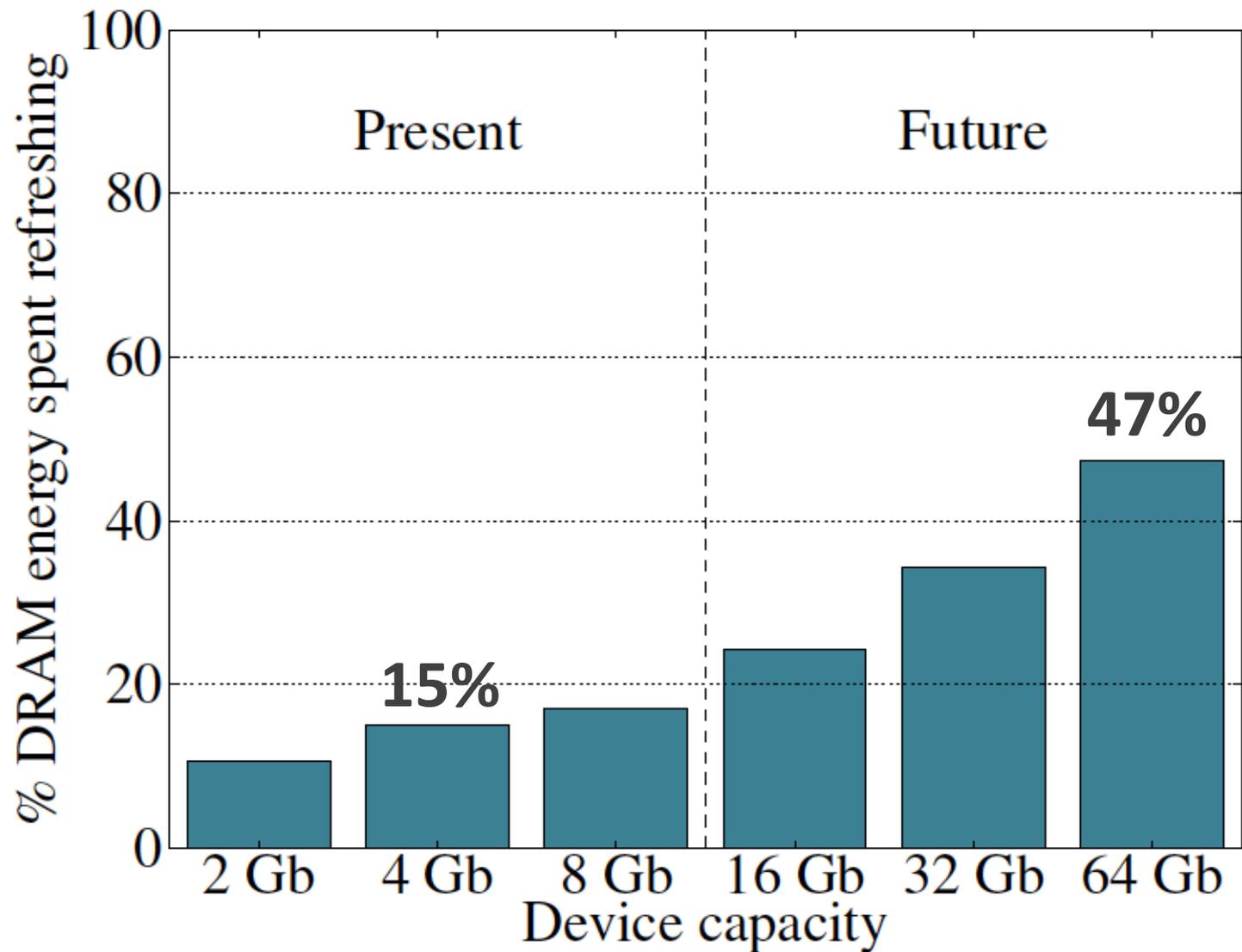
- DRAM capacitor charge leaks over time
- The memory controller needs to refresh each row periodically to restore charge
  - Activate each row every N ms
  - Typical N = 64 ms
- Downsides of refresh
  - **Energy consumption**: Each refresh consumes energy
  - **Performance degradation**: DRAM rank/bank unavailable while refreshed
  - **QoS/predictability impact**: (Long) pause times during refresh
  - **Refresh rate limits DRAM capacity scaling**



# Refresh Overhead: Performance

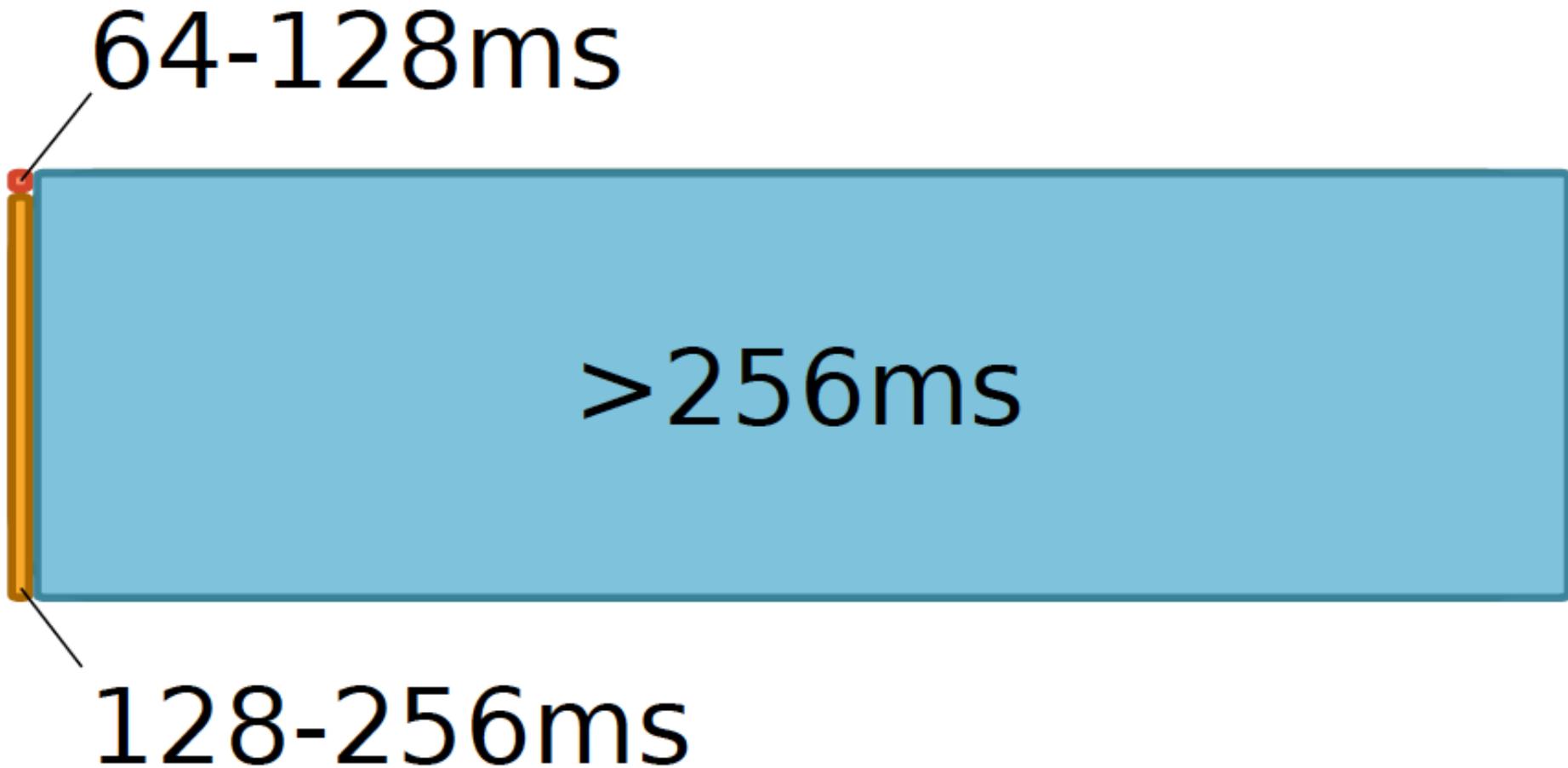


# Refresh Overhead: Energy



# Retention Time Profile of DRAM

---



# RAIDR: Eliminating Unnecessary Refreshes

■ Observation: Most DRAM rows can be refreshed much less often without losing data [Kim+, EDL'09][Liu+ ISCA'13]

■ Key idea: Refresh rows containing weak cells more frequently, other rows less frequently

1. Profiling: Profile retention time of all rows

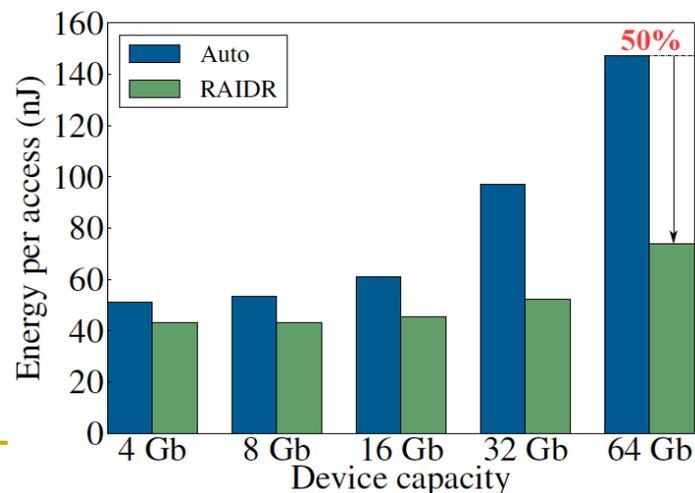
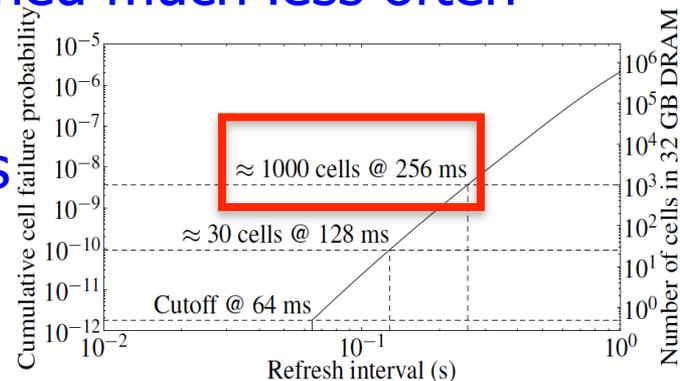
2. Binning: Store rows into bins by retention time in memory controller

*Efficient storage with Bloom Filters (only 1.25KB for 32GB memory)*

3. Refreshing: Memory controller refreshes rows in different bins at different rates

■ Results: 8-core, 32GB, SPEC, TPC-C, TPC-H

- 74.6% refresh reduction @ 1.25KB storage
- ~16%/20% DRAM dynamic/idle power reduction
- ~9% performance improvement
- Benefits increase with DRAM capacity



More Detail

# RowHammer in Popular Sites and Press

---

- [https://en.wikipedia.org/wiki/Row\\_hammer](https://en.wikipedia.org/wiki/Row_hammer)
- <https://twitter.com/hashtag/rowhammer?f=realtime>
- <http://www.rowhammer.com/>
- <http://www.zdnet.com/article/flipping-dram-bits-maliciously/>
- <http://www.infoworld.com/article/2894497/security/rowhammer-hardware-bug-threatens-to-smashnotebook->
- <http://www.zdnet.com/article/rowhammer-dram-flaw-could-be-widespread-says-google/>
- <http://arstechnica.com/security/2015/03/cutting-edge-hack-gives-super-user-status-by-exploiting-dramweakness/>
- <https://www.youtube.com/watch?v=H63dUfGBpxE>
- <http://www.wired.com/2015/03/google-hack-dram-memory-electric-leaks/>
- <https://www.grc.com/sn/sn-498-notes.pdf>

# Recap: The DRAM Scaling Problem

## DRAM Process Scaling Challenges

### ❖ Refresh

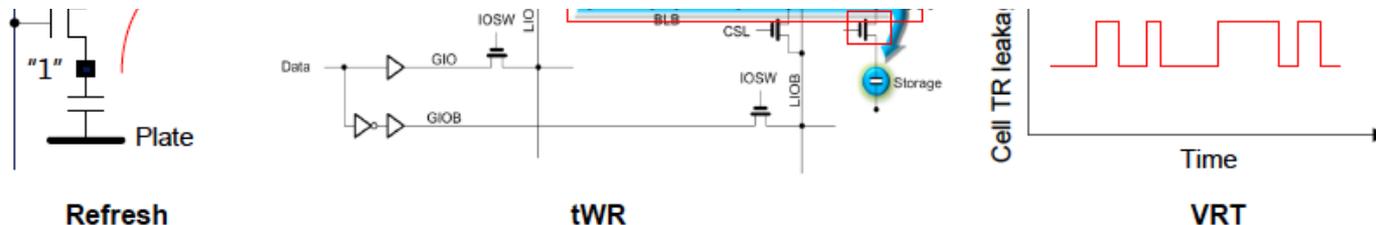
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance

THE MEMORY FORUM 2014

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

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

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



# DRAM Retention Failure Analysis

---

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

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

Jamie Liu\*

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

Ben Jaiyen\*

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

Yoongu Kim

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

Chris Wilkerson

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

Onur Mutlu

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

# Towards an Online Profiling System

## *Key Observations:*

- **Testing** alone **cannot detect** all possible failures
- **Combination** of ECC and other mitigation techniques is much more **effective**
  - **But degrades performance**
- **Testing** can help to reduce the **ECC strength**
  - Even when starting with a **higher strength ECC**

# Towards an Online Profiling System

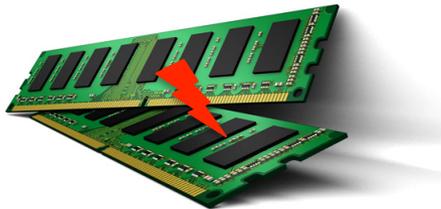
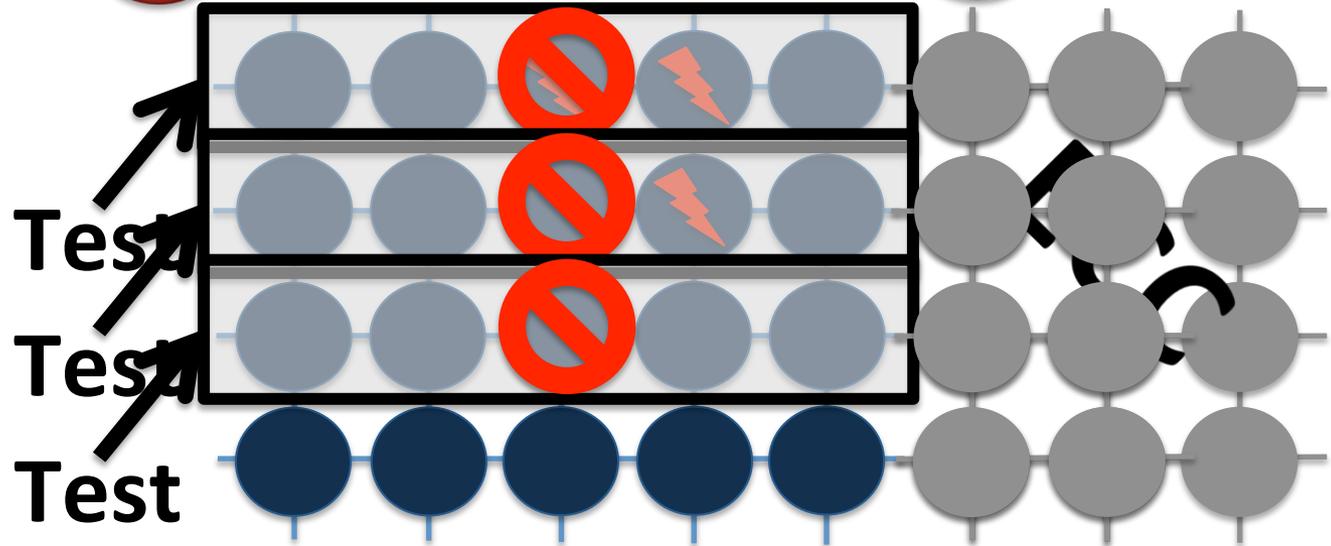
Initially Protect DRAM  
with Strong ECC

1



Periodically Test  
Parts of DRAM

2



Mitigate errors and  
reduce ECC

3

Run tests periodically after a short interval  
at smaller regions of memory

# Online Mitigating of DRAM Failures

---

- Samira Khan, Donghyuk Lee, Yoongu Kim, Alaa Alameldeen, Chris Wilkerson, and Onur Mutlu,  
**"The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study"**  
*Proceedings of the*  
*ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, Austin, TX, June 2014. [[Slides \(pptx\)](#)] [[pdf](#)] [[Poster \(pptx\)](#)] [[pdf](#)] [[Full data sets](#)]

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

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

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

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

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

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

Onur Mutlu<sup>†</sup>  
onur@cmu.edu

<sup>†</sup>Carnegie Mellon University

<sup>\*</sup>Intel Labs

# Memory Errors in Facebook Fleet

---

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

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

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

---

# Findings

Error/failure occurrence

*Page offlining  
at scale*

*Technology  
scaling*



New  
reliability  
trends

Modeling errors

*Architecture &  
workload*

# Findings

Error/failure occurrence

Page *2*

Errors follow a **power-law distribution** and a large number of errors occur due to **sockets/channels**

Modeling errors

*Architecture & workload*

# Findings

Error/failure occurrence

We find that ***newer*** cell fabrication technologies have ***higher failure rates***

*Technology scaling*

reliability trends

Modeling errors

*Architecture & workload*

# Findings

Error/failure occurrence

*Page* ***Chips per DIMM, transfer width, and workload type*** (not necessarily CPU/memory utilization) affect reliability *2*

trends

Modeling errors

*Architecture & workload*

# Findings

Error/failure occurrence

We have made publicly available a ***statistical model*** for assessing server memory reliability

Modeling errors

*Architecture & workload*

# Findings

Error/failure occurrence

*Page offlining  
at scale*

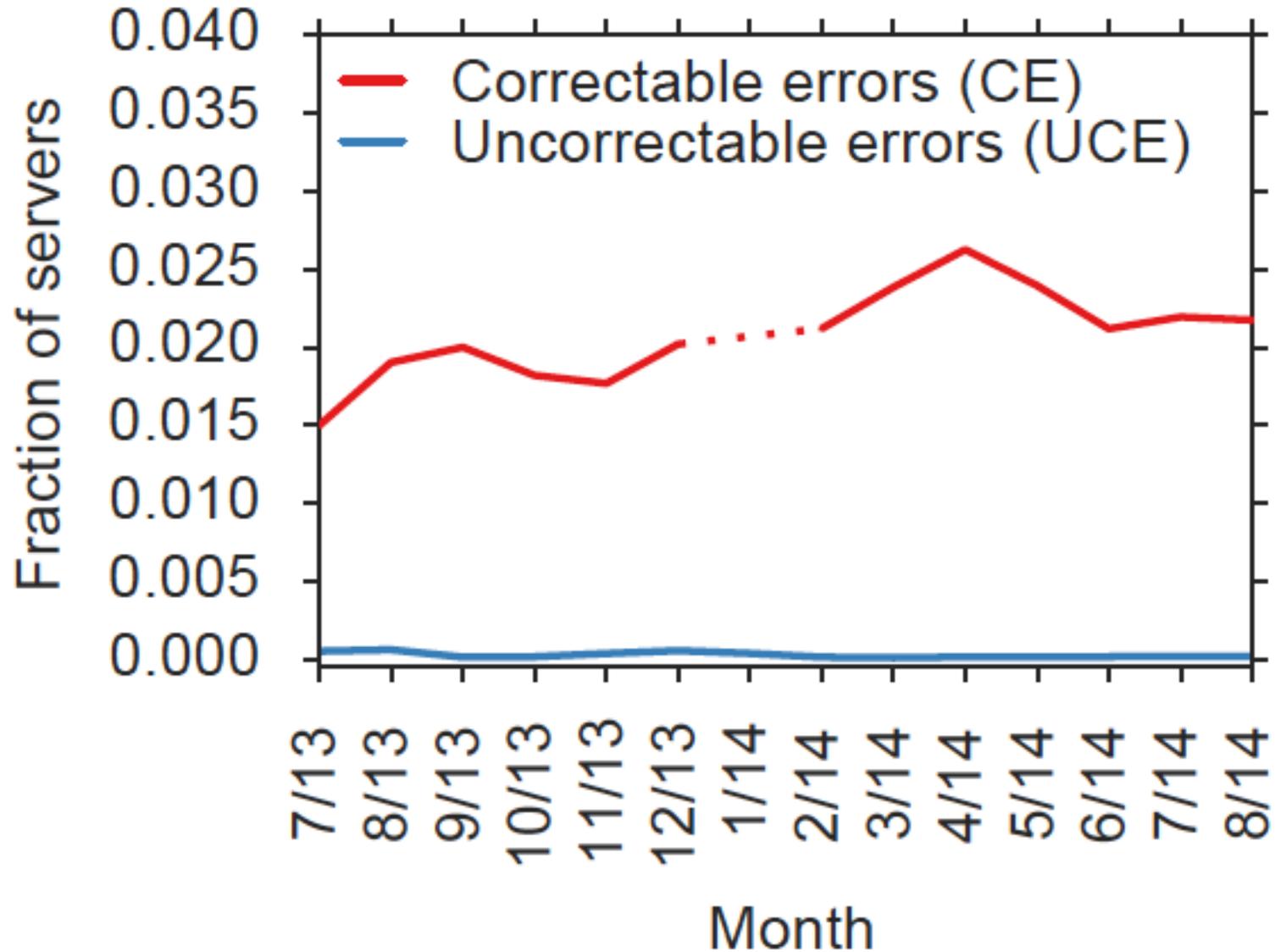
***First large-scale study*** of  
page offlining; real-world  
***limitations*** of technique

reliability  
trends

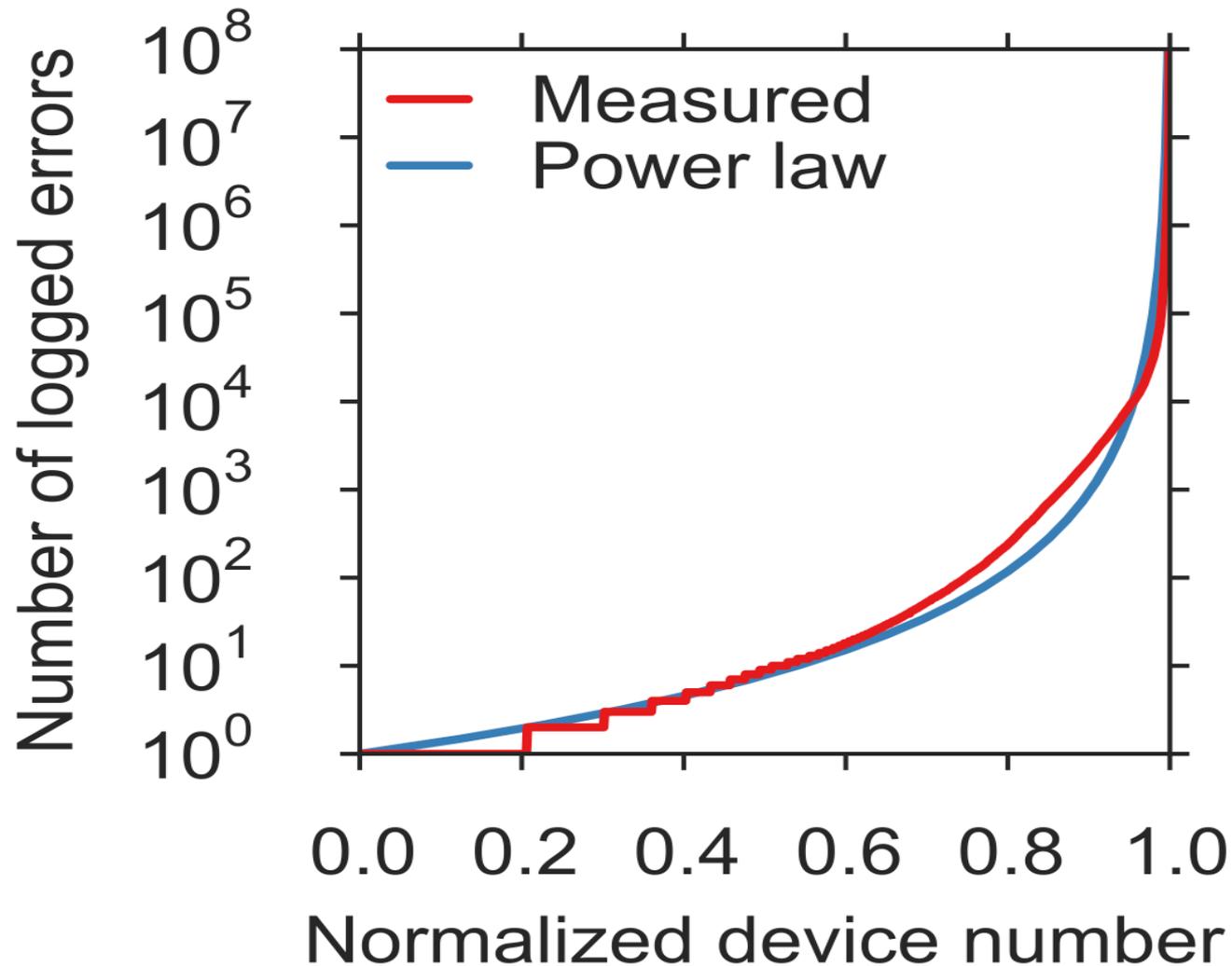
Modeling errors

*Architecture &  
workload*

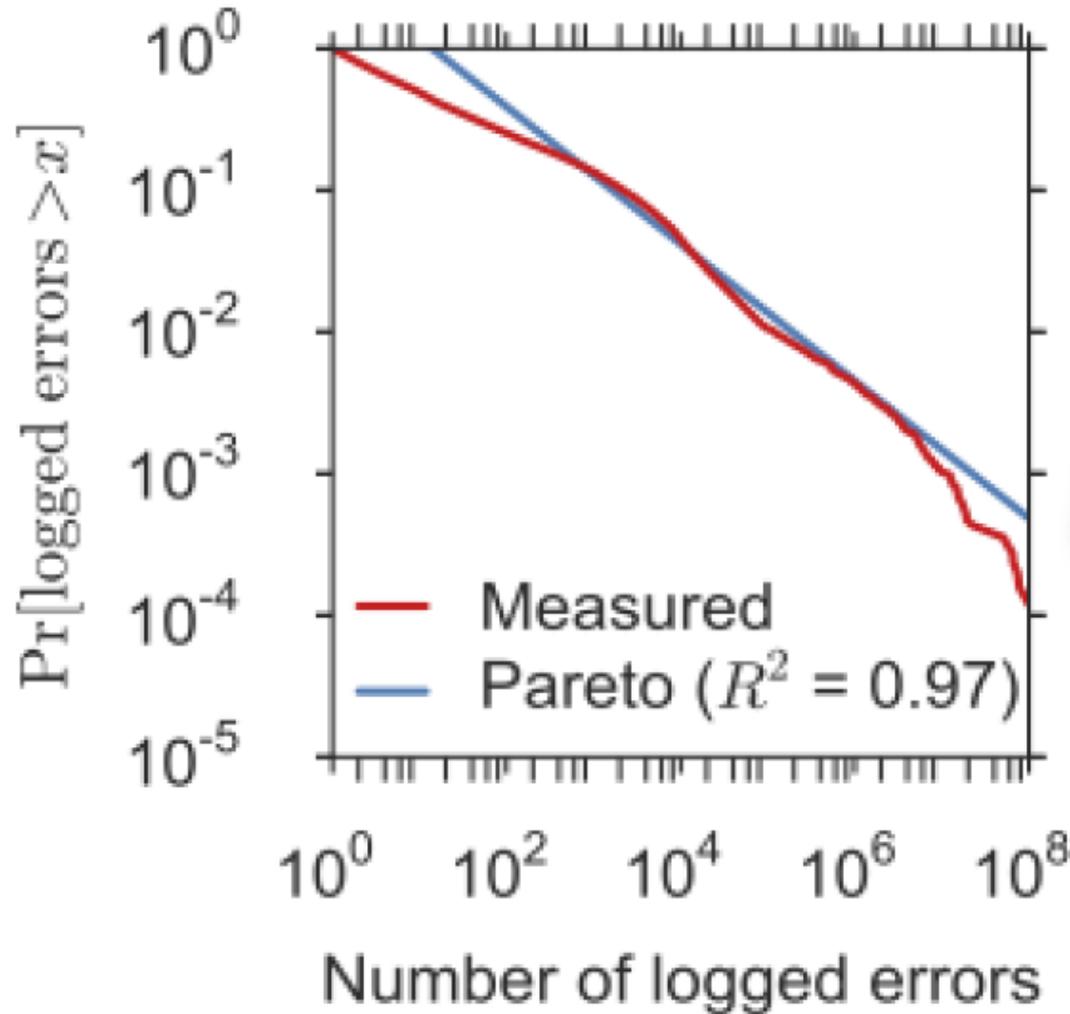
# Server error rate



# Memory error distribution



# Memory error distribution



*Decreasing  
hazard  
rate*

# Errors in Flash Memory (I)

## 1. Retention noise study and management

- 1) Yu Cai, Gulay Yalcin, Onur Mutlu, Erich F. Haratsch, Adrian Cristal, Osman Unsal, and Ken Mai,  
[Flash Correct-and-Refresh: Retention-Aware Error Management for Increased Flash Memory Lifetime](#), ICCD 2012.
- 2) Yu Cai, Yixin Luo, Erich F. Haratsch, Ken Mai, and Onur Mutlu,  
[Data Retention in MLC NAND Flash Memory: Characterization, Optimization and Recovery](#), HPCA 2015.
- 3) Yixin Luo, Yu Cai, Saugata Ghose, Jongmoo Choi, and Onur Mutlu,  
[WARM: Improving NAND Flash Memory Lifetime with Write-hotness Aware Retention Management](#), MSST 2015.

## 2. Flash-based SSD prototyping and testing platform

- 4) Yu Cai, Erich F. Haratsh, Mark McCartney, Ken Mai,  
[FPGA-based solid-state drive prototyping platform](#), FCCM 2011.

# Errors in Flash Memory (II)

## **3. Overall flash error analysis**

- 5) Yu Cai, Erich F. Haratsch, Onur Mutlu, and Ken Mai,  
[Error Patterns in MLC NAND Flash Memory: Measurement, Characterization, and Analysis](#), DATE 2012.
- 6) Yu Cai, Gulay Yalcin, Onur Mutlu, Erich F. Haratsch, Adrian Cristal, Osman Unsal, and Ken Mai,  
[Error Analysis and Retention-Aware Error Management for NAND Flash Memory](#), ITJ 2013.

## **4. Program and erase noise study**

- 7) Yu Cai, Erich F. Haratsch, Onur Mutlu, and Ken Mai,  
[Threshold Voltage Distribution in MLC NAND Flash Memory: Characterization, Analysis and Modeling](#), DATE 2013.

# Errors in Flash Memory (III)

## 5. Cell-to-cell interference characterization and tolerance

- 8) Yu Cai, Onur Mutlu, Erich F. Haratsch, and Ken Mai,  
[Program Interference in MLC NAND Flash Memory: Characterization, Modeling, and Mitigation](#), ICCD 2013.
- 9) Yu Cai, Gulay Yalcin, Onur Mutlu, Erich F. Haratsch, Osman Unsal, Adrian Cristal, and Ken Mai,  
[Neighbor-Cell Assisted Error Correction for MLC NAND Flash Memories](#), SIGMETRICS 2014.

## 6. Read disturb noise study

- 10) Yu Cai, Yixin Luo, Saugata Ghose, Erich F. Haratsch, Ken Mai, and Onur Mutlu,  
[Read Disturb Errors in MLC NAND Flash Memory: Characterization and Mitigation](#), DSN 2015.

## 7. Flash errors in the field

- 11) Justin Meza, Qiang Wu, Sanjeev Kumar, and Onur Mutlu,  
[A Large-Scale Study of Flash Memory Errors in the Field](#), SIGMETRICS 2015.

- Yu Cai, Yixin Luo, Erich F. Haratsch, Ken Mai, and Onur Mutlu, **"Data Retention in MLC NAND Flash Memory: Characterization, Optimization and Recovery"**  
*Proceedings of the 21st International Symposium on High-Performance Computer Architecture (HPCA)*, Bay Area, CA, February 2015.  
[[Slides \(pptx\)](#)] [[pdf](#)]

## Data Retention in MLC NAND Flash Memory: Characterization, Optimization, and Recovery

Yu Cai, Yixin Luo, Erich F. Haratsch\*, Ken Mai, Onur Mutlu  
Carnegie Mellon University, \*LSI Corporation

[yucaicai@gmail.com](mailto:yucaicai@gmail.com), [yixinluo@cs.cmu.edu](mailto:yixinluo@cs.cmu.edu), [erich.haratsch@lsi.com](mailto:erich.haratsch@lsi.com), {[kenmai](mailto:kenmai@ece.cmu.edu), [omutlu](mailto:omutlu@ece.cmu.edu)}@ece.cmu.edu

- Yu Cai, Yixin Luo, Saugata Ghose, Erich F. Haratsch, Ken Mai, and Onur Mutlu,  
**"Read Disturb Errors in MLC NAND Flash Memory: Characterization and Mitigation"**  
*Proceedings of the*  
*45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Rio de Janeiro, Brazil, June 2015.

## Read Disturb Errors in MLC NAND Flash Memory: Characterization, Mitigation, and Recovery

Yu Cai, Yixin Luo, Saugata Ghose, Erich F. Haratsch\*, Ken Mai, Onur Mutlu  
Carnegie Mellon University, \*Seagate Technology  
[yucaicai@gmail.com](mailto:yucaicai@gmail.com), {[yixinluo](mailto:yixinluo@cmu.edu), [ghose](mailto:ghose@cmu.edu), [kenmai](mailto:kenmai@cmu.edu), [onur](mailto:onur@cmu.edu)}@cmu.edu

- Yu Cai, Erich F. Haratsch, Onur Mutlu, and Ken Mai, **"Error Patterns in MLC NAND Flash Memory: Measurement, Characterization, and Analysis"**  
*Proceedings of the*  
*Design, Automation, and Test in Europe Conference*  
*(DATE)*, Dresden, Germany, March 2012. [Slides \(ppt\)](#)

## **Error Patterns in MLC NAND Flash Memory: Measurement, Characterization, and Analysis**

Yu Cai<sup>1</sup>, Erich F. Haratsch<sup>2</sup>, Onur Mutlu<sup>1</sup> and Ken Mai<sup>1</sup>

<sup>1</sup>Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA

<sup>2</sup>LSI Corporation, 1110 American Parkway NE, Allentown, PA

<sup>1</sup>{yucai, onur, kenmai}@andrew.cmu.edu, <sup>2</sup>erich.haratsch@lsi.com

# More Detail on Flash Error Analysis

---

- Yu Cai, Gulay Yalcin, Onur Mutlu, Erich F. Haratsch, Adrian Cristal, Osman Unsal, and Ken Mai,  
**"Error Analysis and Retention-Aware Error Management for NAND Flash Memory"**  
*Intel Technology Journal (ITJ) Special Issue on Memory Resiliency*, Vol. 17, No. 1, May 2013.

Intel® Technology Journal | Volume 17, Issue 1, 2013

ERROR ANALYSIS AND RETENTION-AWARE ERROR MANAGEMENT  
FOR NAND FLASH MEMORY

# Google's 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>

---

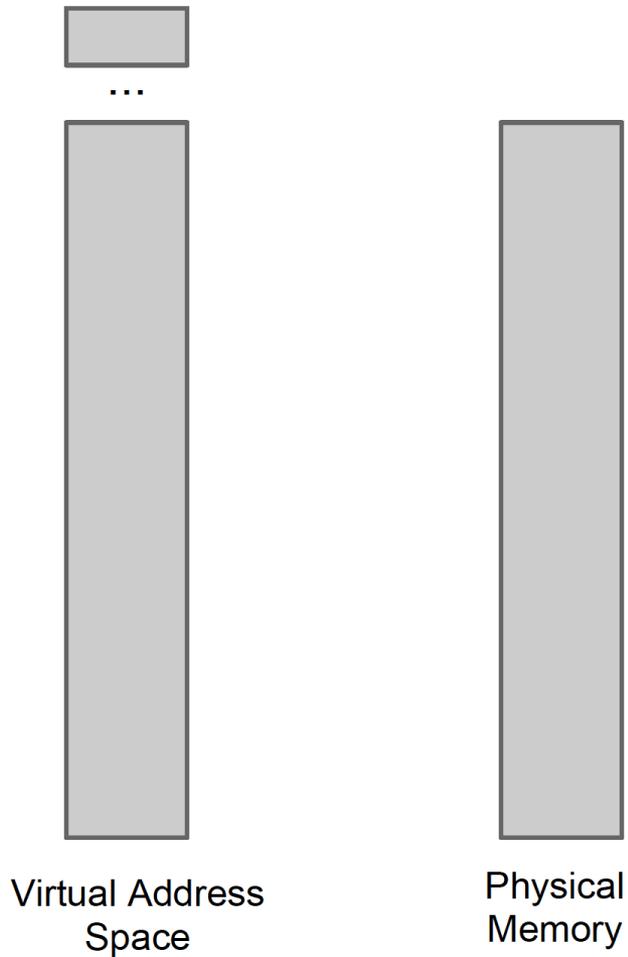
# Kernel exploit

- 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

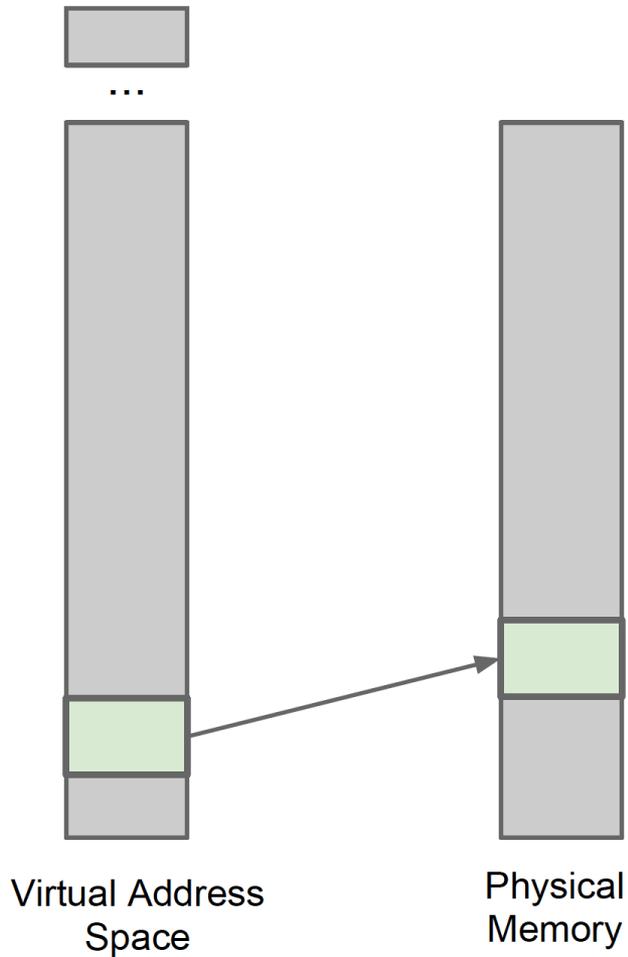
This slide is from Mark Seaborn and Thomas Dullien's BlackHat 2015 talk

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





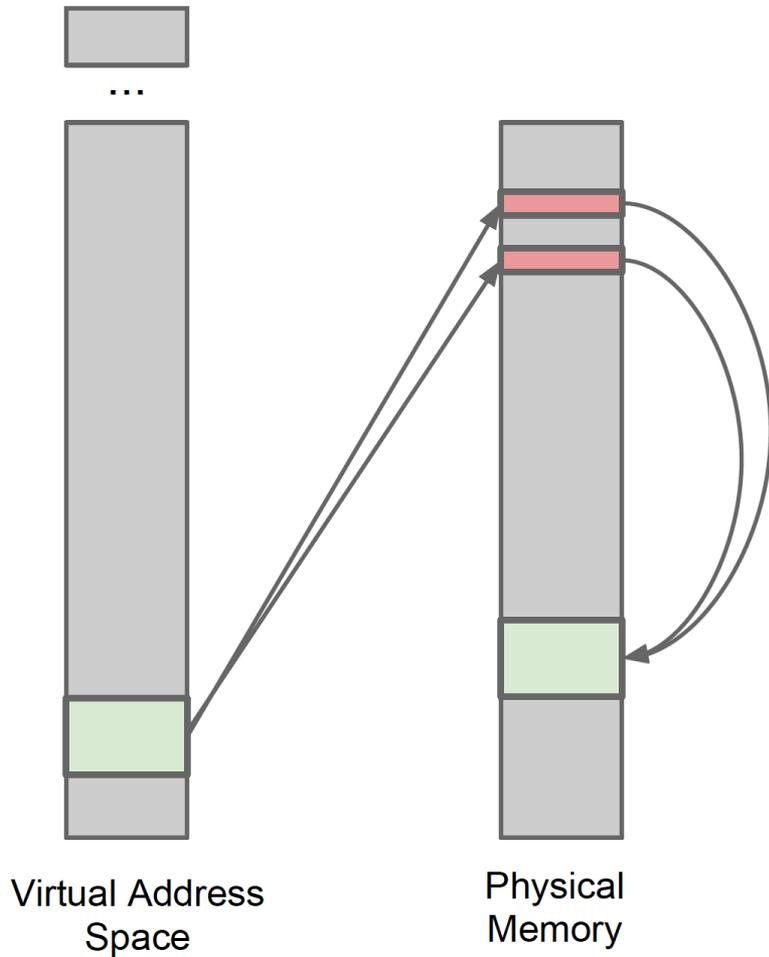
This slide is from Mark Seaborn and Thomas Dullien's BlackHat 2015 talk



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>

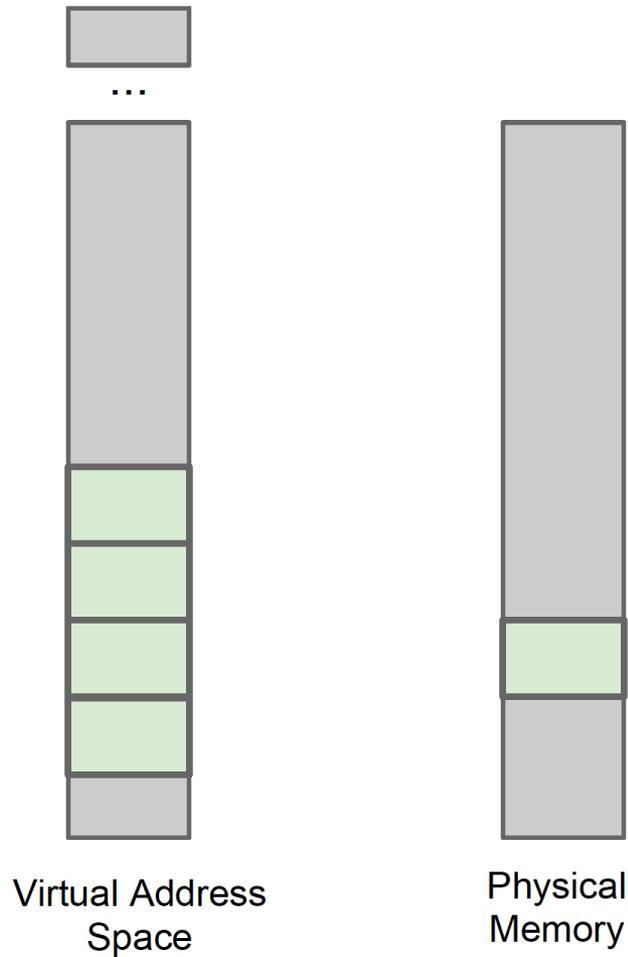


What happens when we map a file with read-write permissions? Indirection via page tables.

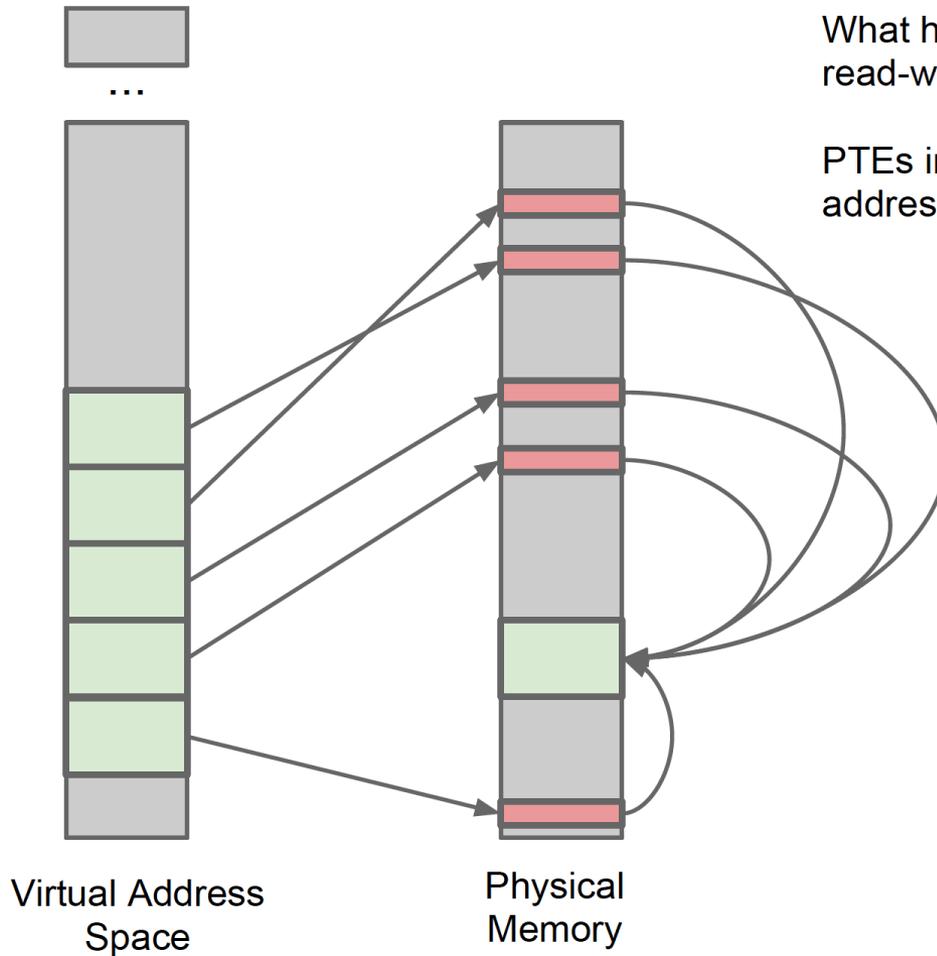
This slide is from Mark Seaborn and Thomas Dullien's BlackHat 2015 talk

<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?



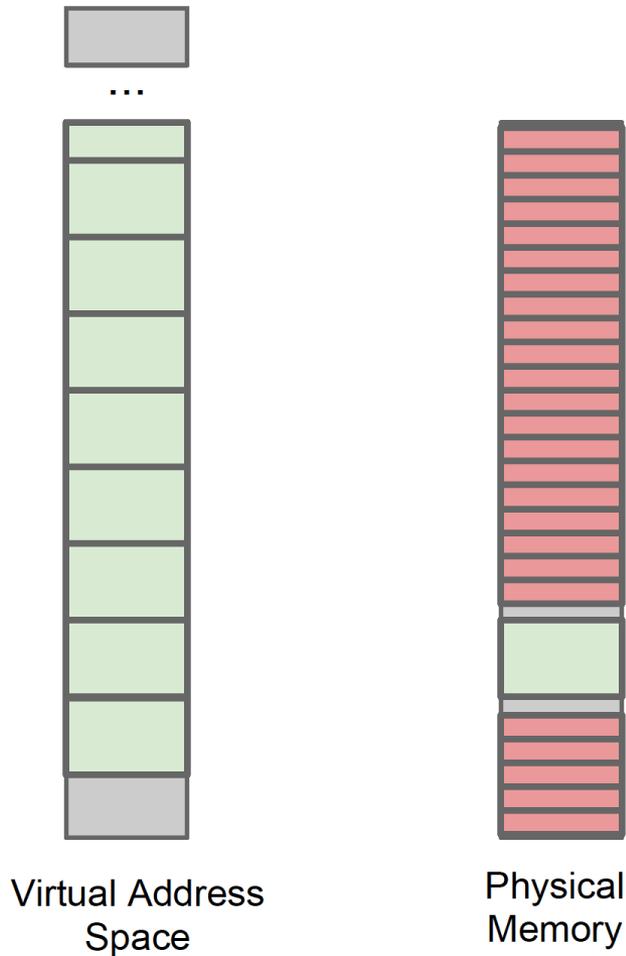
This slide is from Mark Seaborn and Thomas Dullien's BlackHat 2015 talk



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

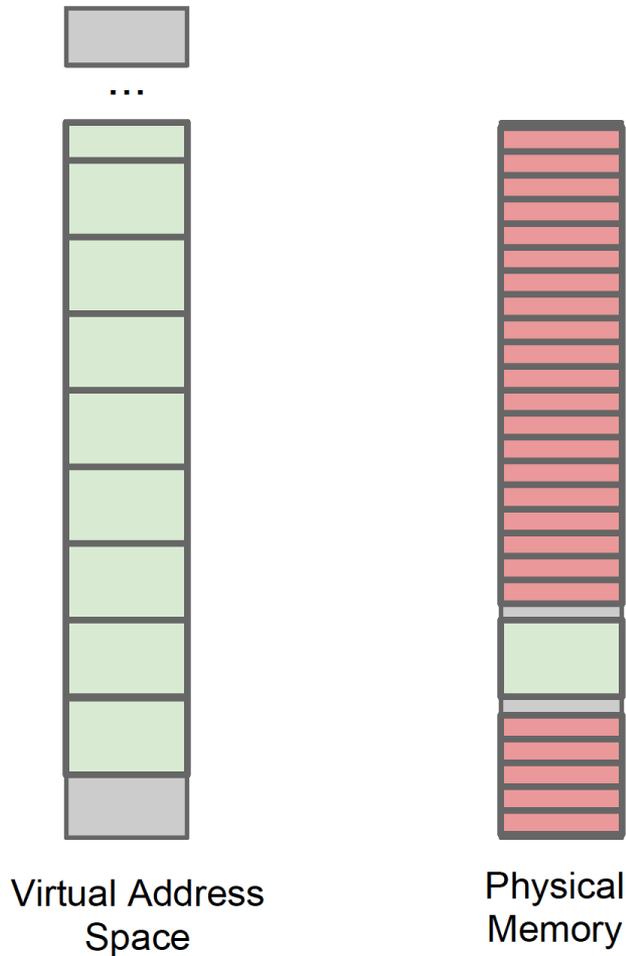


What happens when we repeatedly map a file with read-write permissions?

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



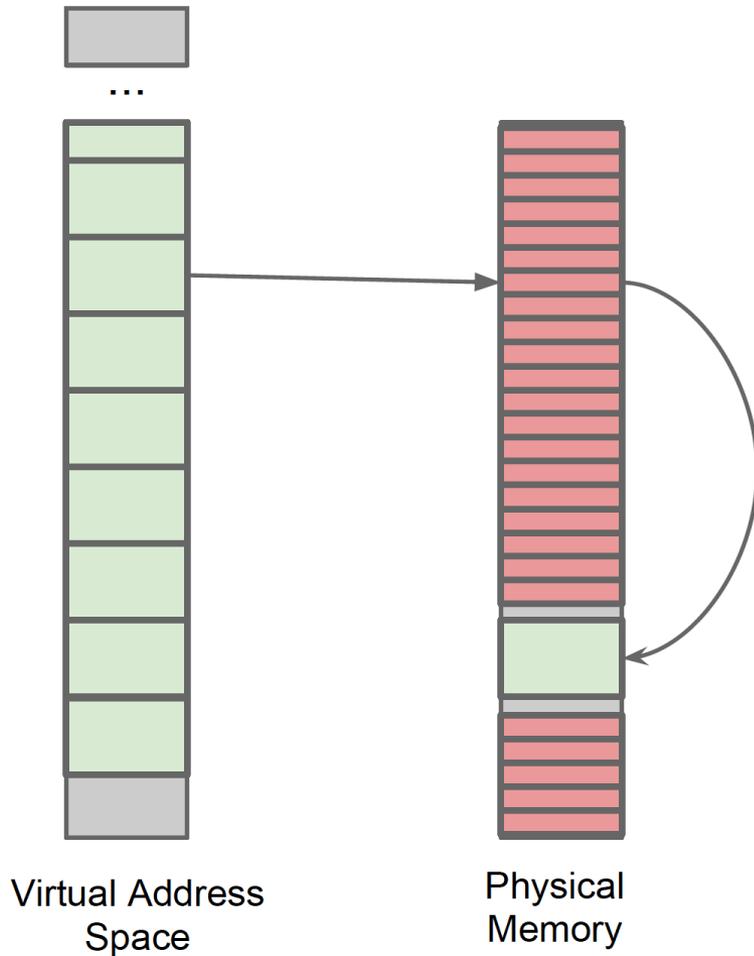
What happens when we repeatedly map a file with read-write permissions?

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



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

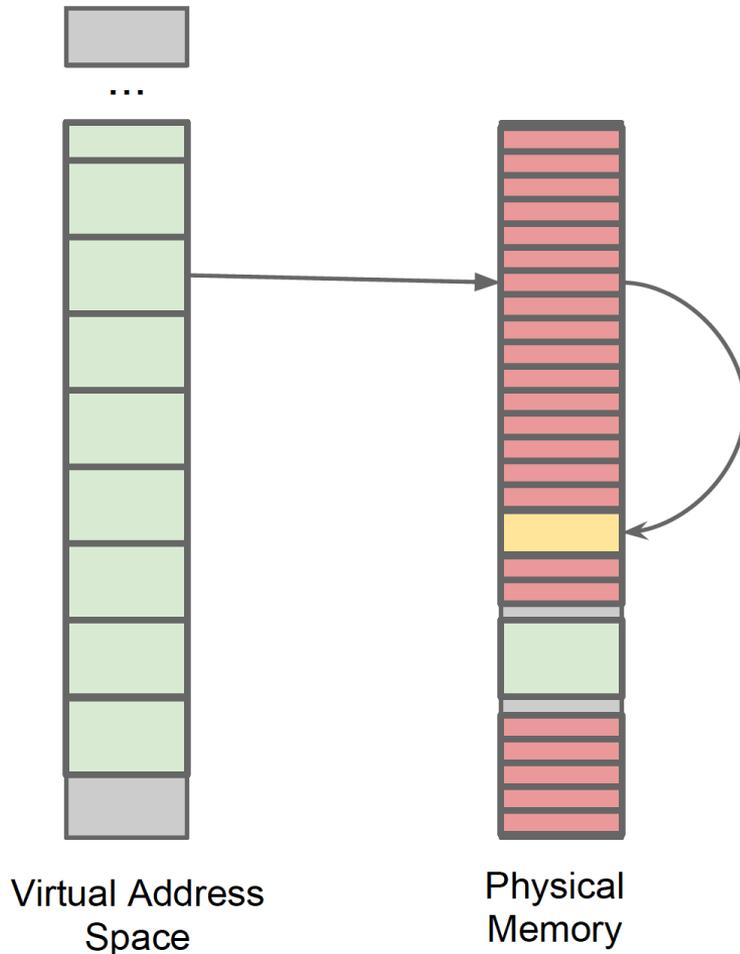
PTEs in physical memory help resolve virtual addresses to physical pages.

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



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

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

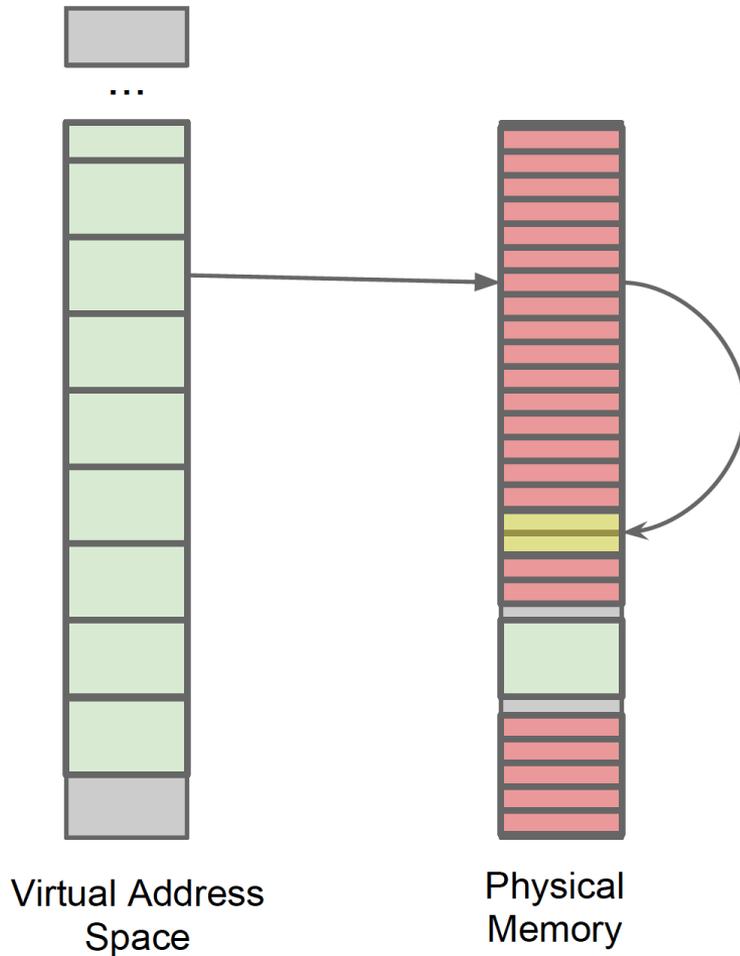
We can fill physical memory with PTEs.

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



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.

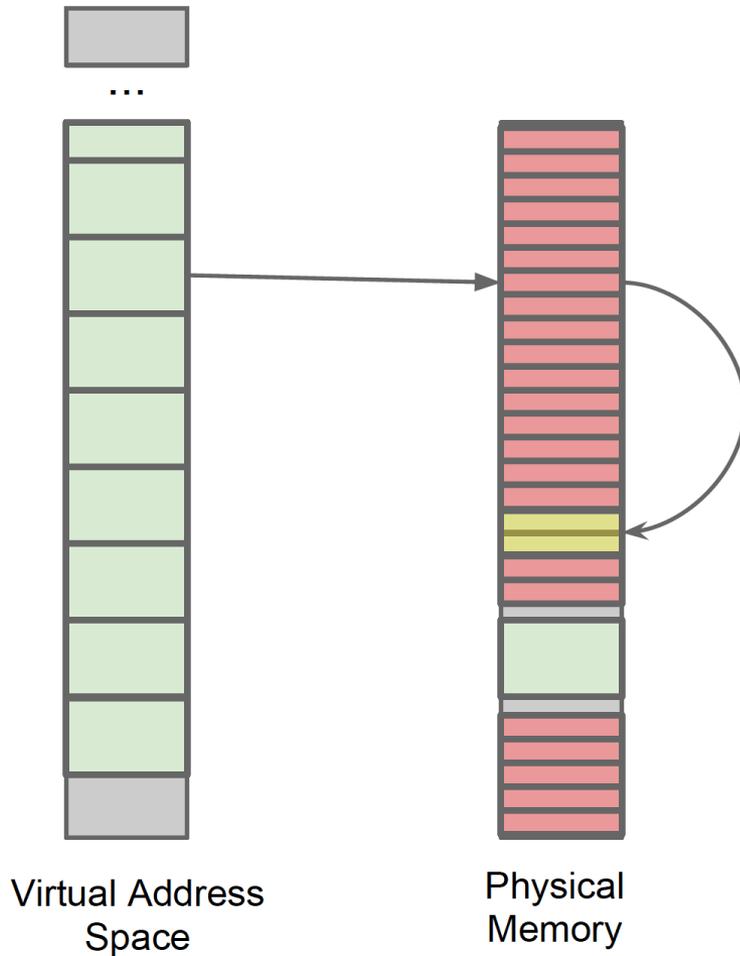
Each of them points to pages in the same physical file mapping.

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



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.

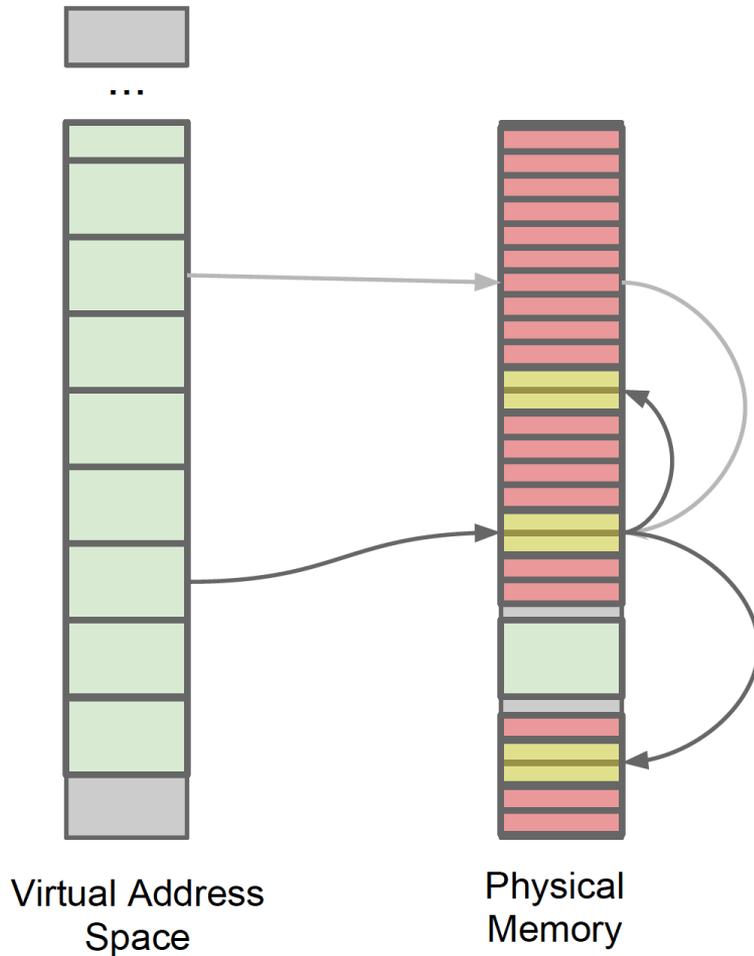
Each of them points to pages in the same physical file mapping.

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 ...



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

# 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>