Reliability (and Security) Issues of DRAM and NAND Flash Scaling

Onur Mutlu

omutlu@gmail.com

http://users.ece.cmu.edu/~omutlu/

HPCA Memory Reliability Workshop

March 13, 2016

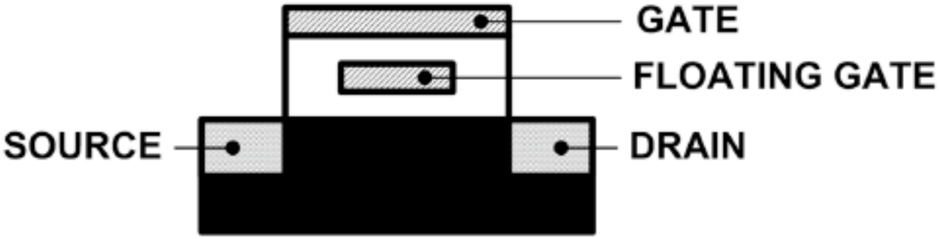
Carnegie Mellon

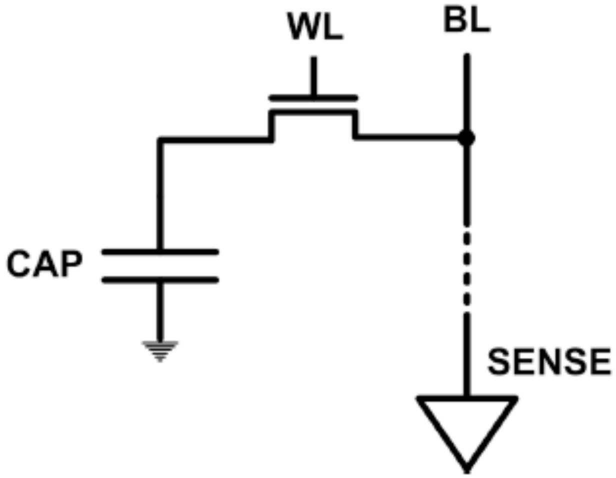
SAFARI

Limits of Charge Memory

- Difficult charge placement and control
  - Flash: floating gate charge
  - DRAM: capacitor charge, transistor leakage

- Reliable sensing, data retention, and charge control become more difficult as charge storage unit size reduces





Agenda

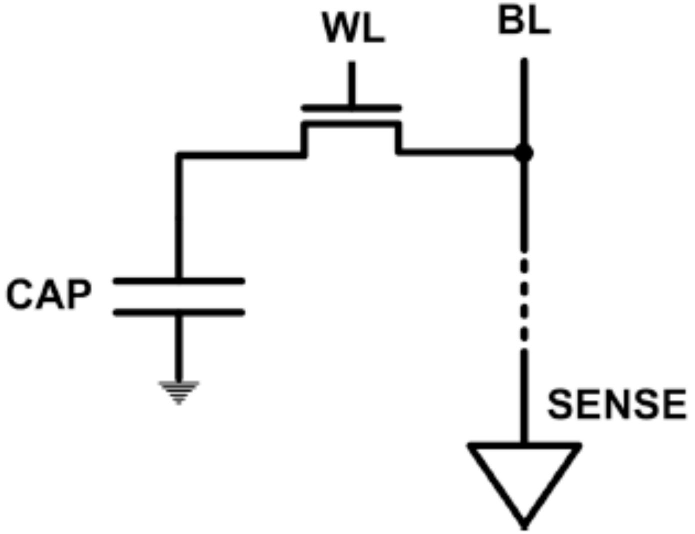
- DRAM Scaling Issues
  - DRAM RowHammer Problem
  - Some Other DRAM Reliability Studies

- NAND Flash Scaling Issues
  - Some NAND Flash Reliability Studies
  - Read Disturb Errors in NAND Flash Memory

- Summary and Discussion

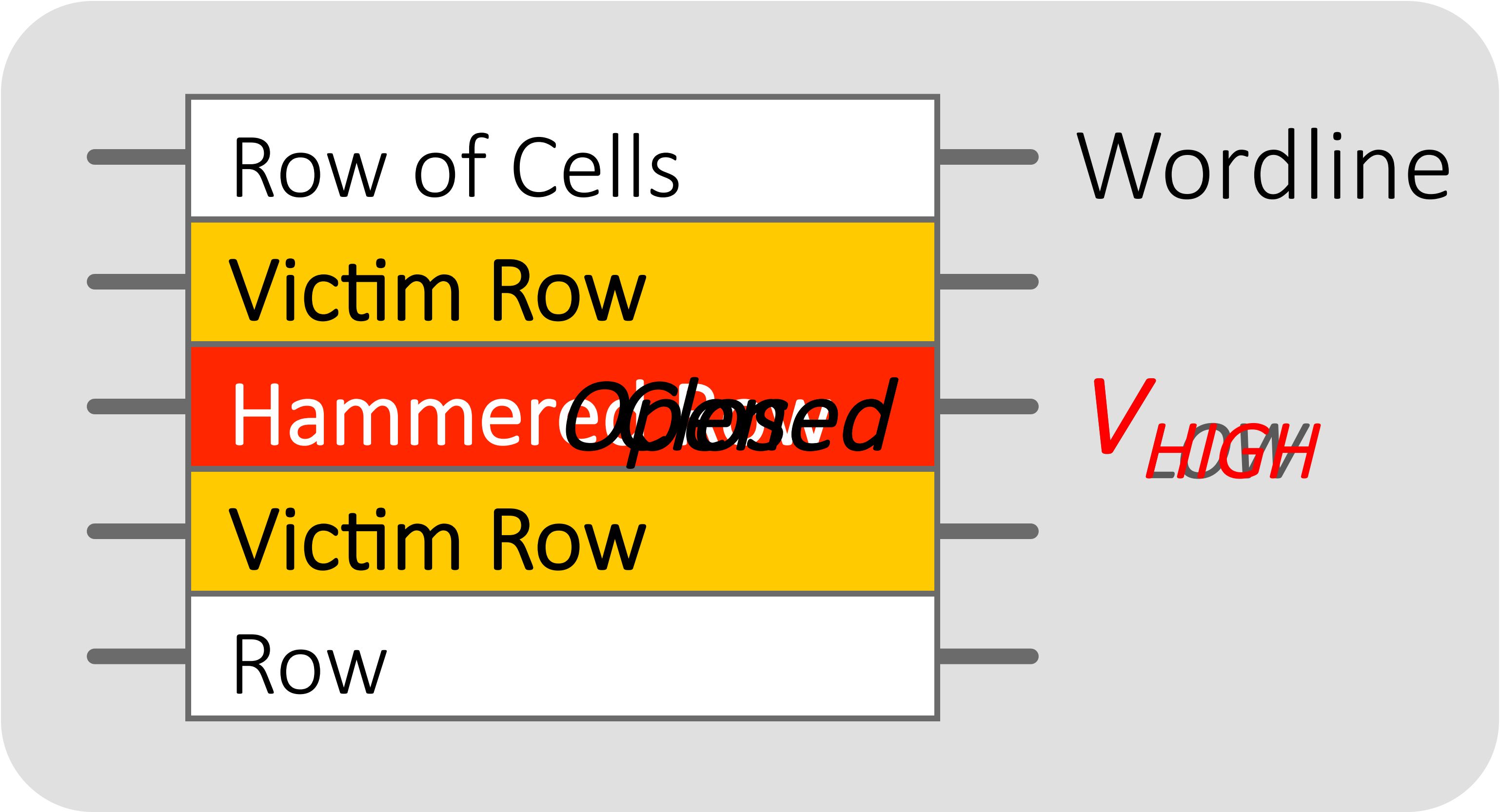
The DRAM Scaling Problem

- 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

An Example of the DRAM Scaling Problem

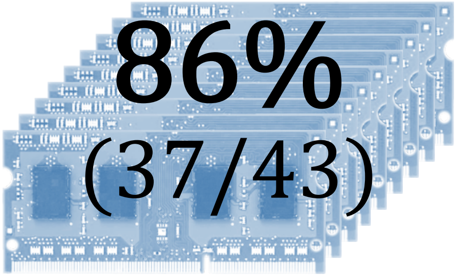


Repeatedly opening and closing a row enough times within a refresh interval induces disturbance errors in adjacent rows in most real DRAM chips you can buy today

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

Most DRAM Modules Are Vulnerable

A company



86%

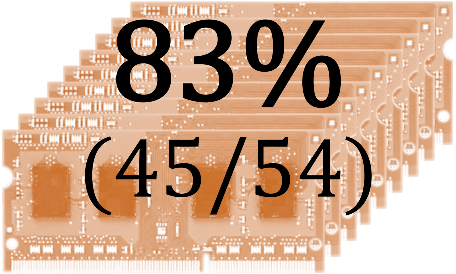
(37/43)

Up to

1.0×10^{7}

errors

B company



83%

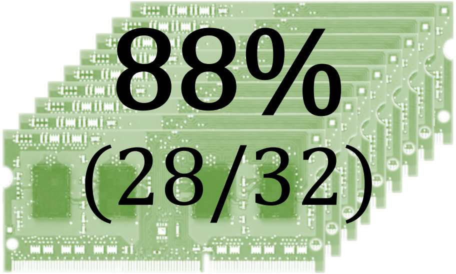
(45/54)

Up to

2.7×10^{6}

errors

C company



88%

(28/32)

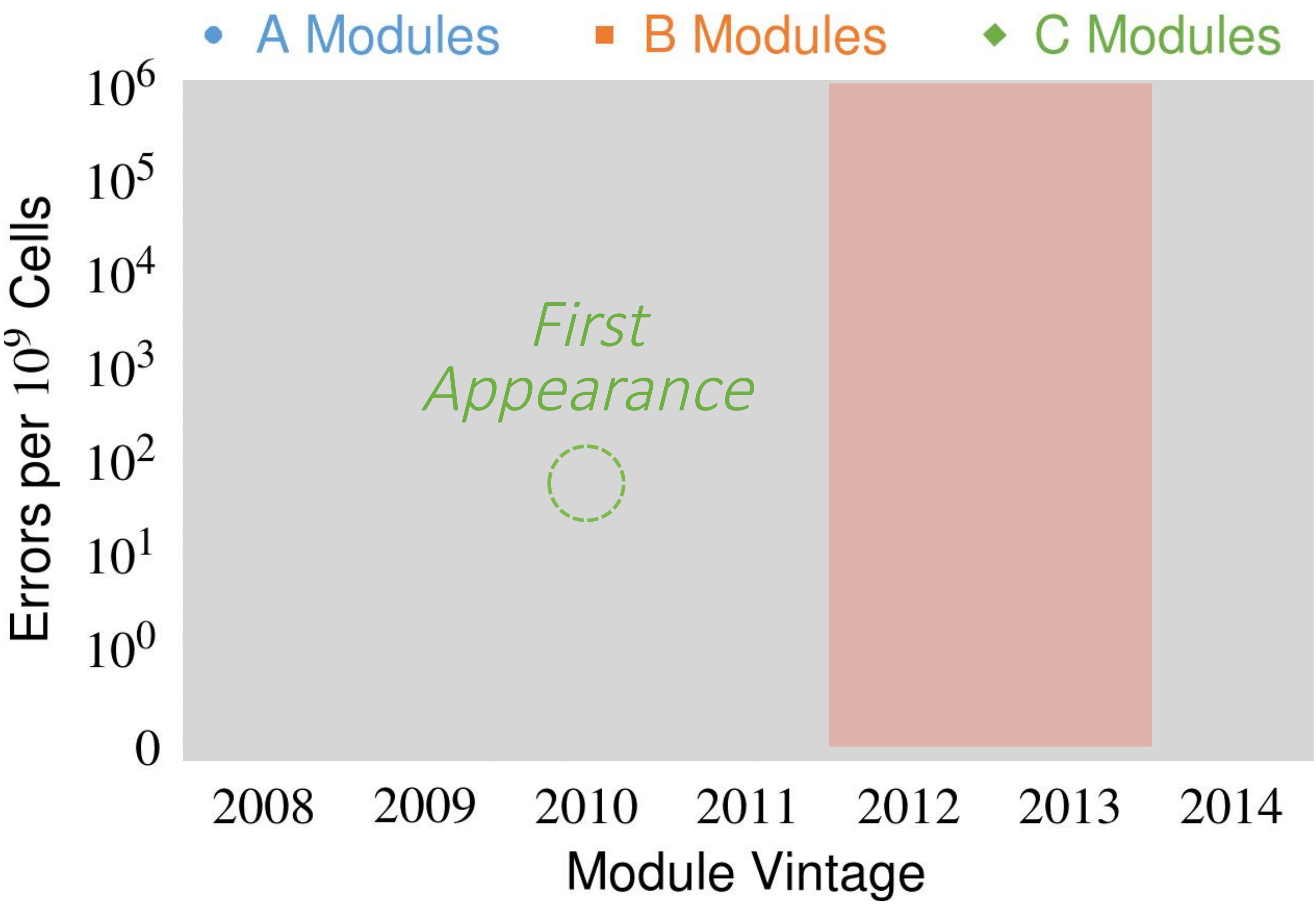
Up to

3.3×10^{5}

errors

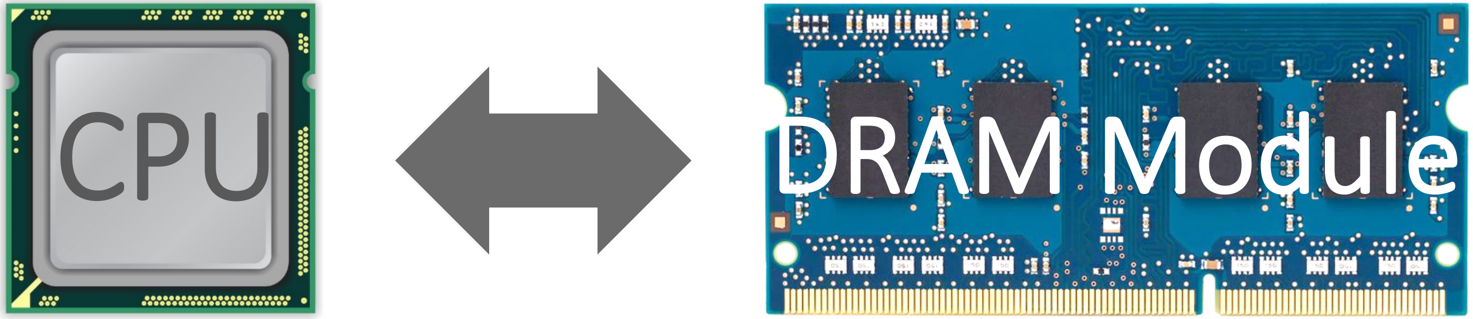
Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors, (Kim et al., ISCA 2014)

Recent DRAM Is More Vulnerable

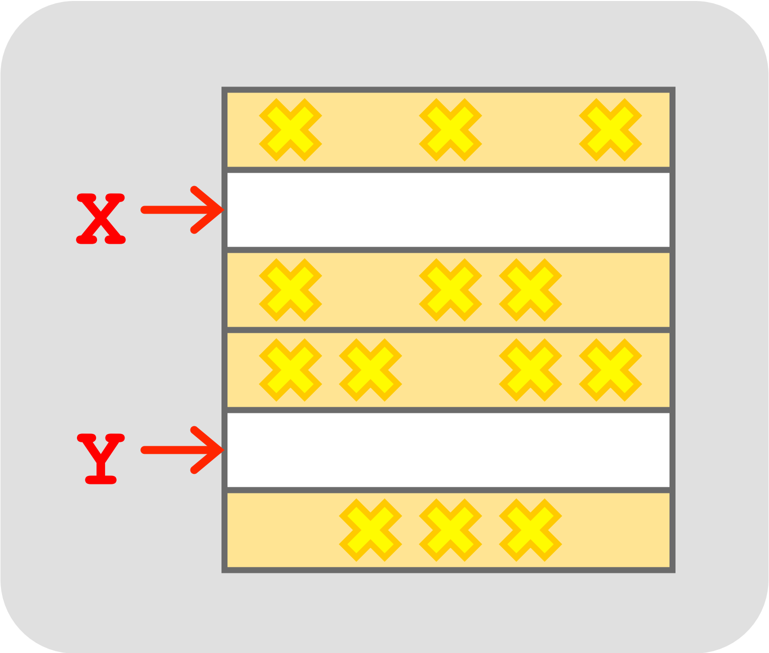


All modules from 2012–2013 are vulnerable

A Simple Program Can Induce Many Errors



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



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

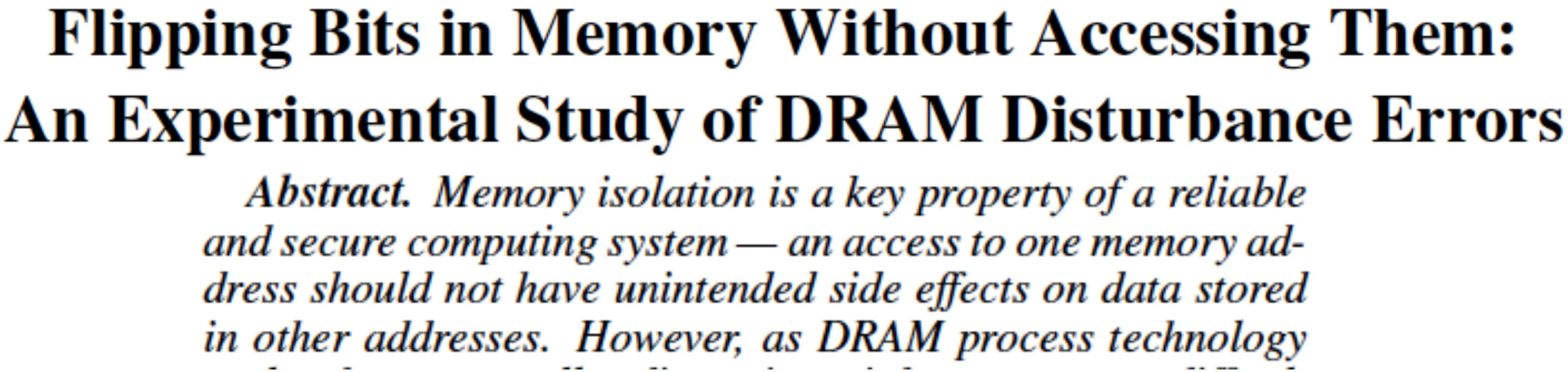
Observed Errors in Real Systems

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

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

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

One Can Take Over an Otherwise-Secure System



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

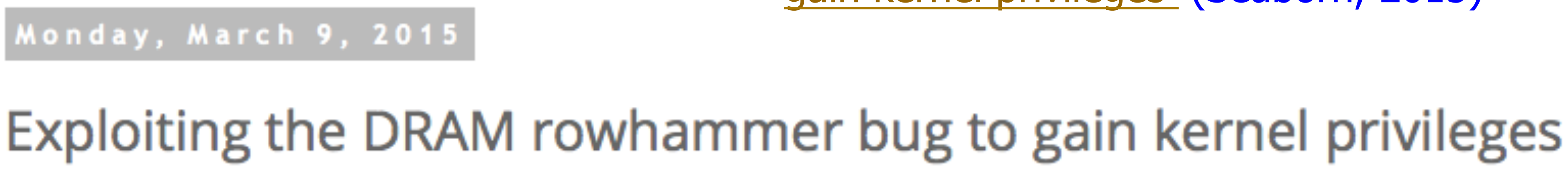
Abstract. *Memory isolation is a key property of a reliable and secure computing system — an access to one memory address should not have unintended side effects on data stored in other addresses. However, as DRAM process technology*

Project Zero

News and updates from the Project Zero team at Google

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

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



Monday, March 9, 2015

Exploiting the DRAM rowhammer bug to gain kernel privileges

RowHammer Security Attack Example

- "Rowhammer" is a problem with some recent DRAM devices in which repeatedly accessing a row of memory can cause bit flips in adjacent rows (Kim et al., ISCA 2014).
  - Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors (Kim et al., ISCA 2014)

- We tested a selection of laptops and found that a subset of them exhibited the problem.
- We built two working privilege escalation exploits that use this effect.
  - Exploiting the DRAM rowhammer bug to gain kernel privileges (Seaborn, 2015)

- One exploit uses rowhammer-induced bit flips to gain kernel privileges on x86-64 Linux when run as an unprivileged userland process.
- When run on a machine vulnerable to the rowhammer problem, the process was able to induce bit flips in page table entries (PTEs).
- It was able to use this to gain write access to its own page table, and hence gain read-write access to all of physical memory.

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

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

Selected Readings on RowHammer

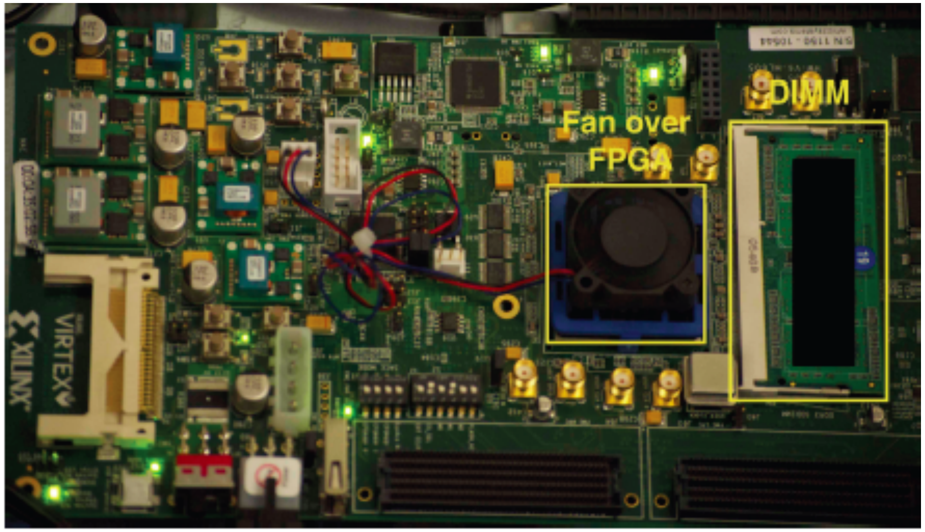
- Our first detailed study: Rowhammer analysis and solutions
  - Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu, **"Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors"** *Proceedings 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
  - 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
- Remote RowHammer Attacks via JavaScript (July 2015)
  - http://arxiv.org/abs/1507.06955
  - https://github.com/IAIK/rowhammerjs

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



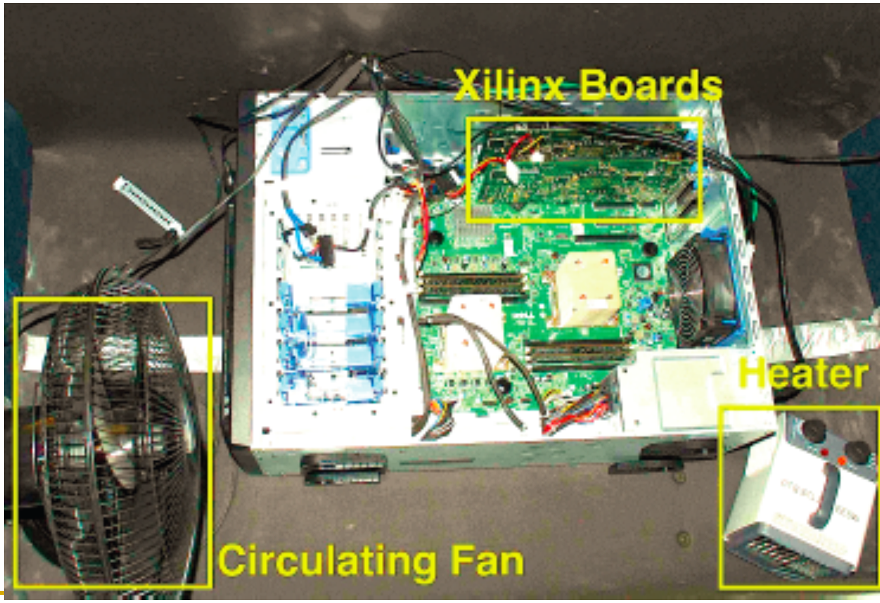
An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms (Liu et al., ISCA 2013)

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

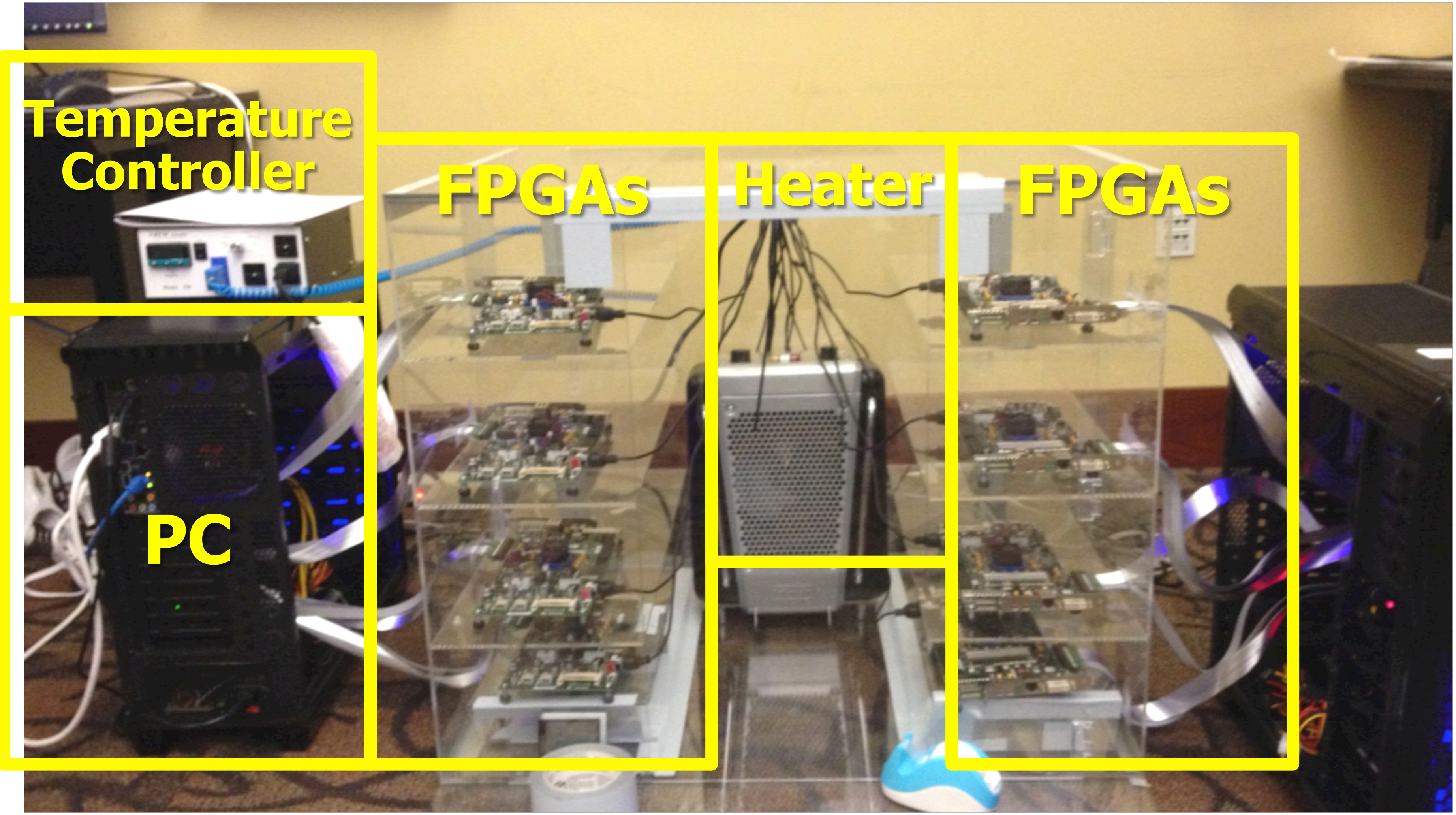
Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors (Kim et al., ISCA 2014)

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 Infrastructure (DRAM)



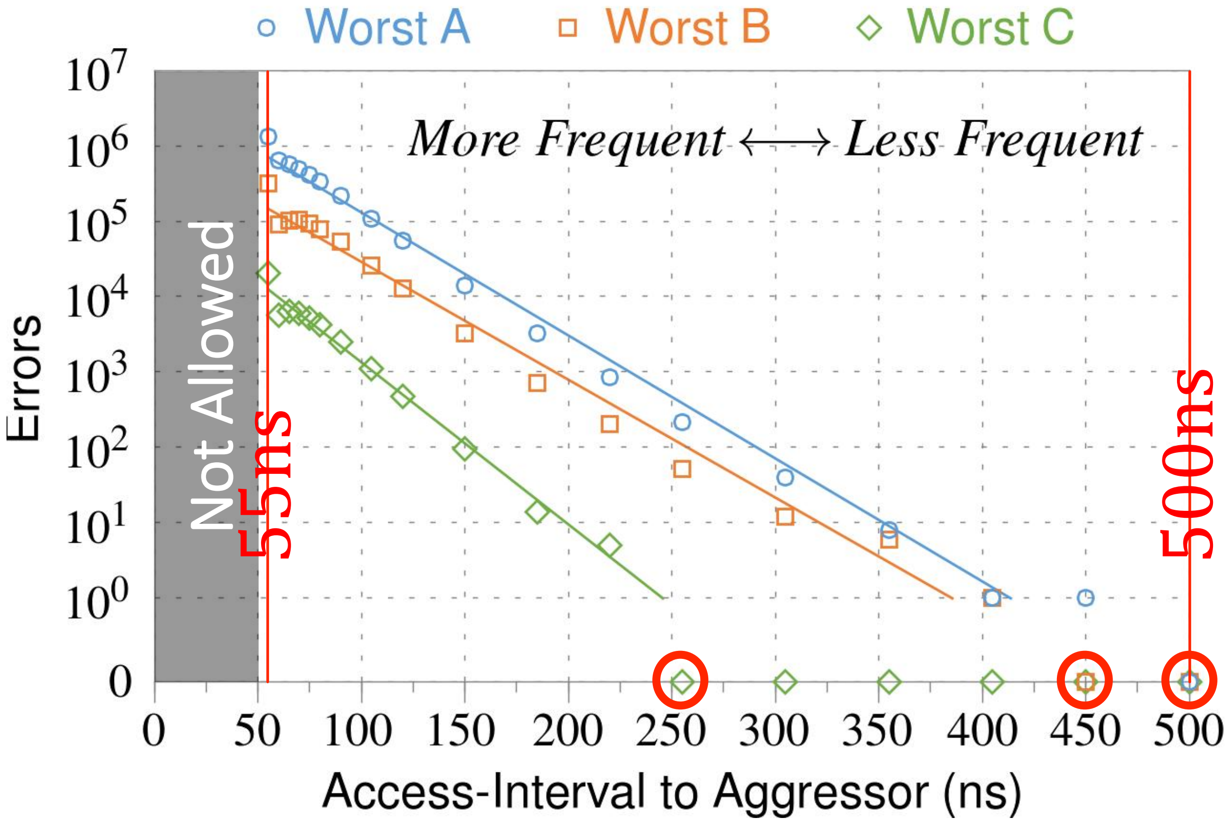
Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.

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)

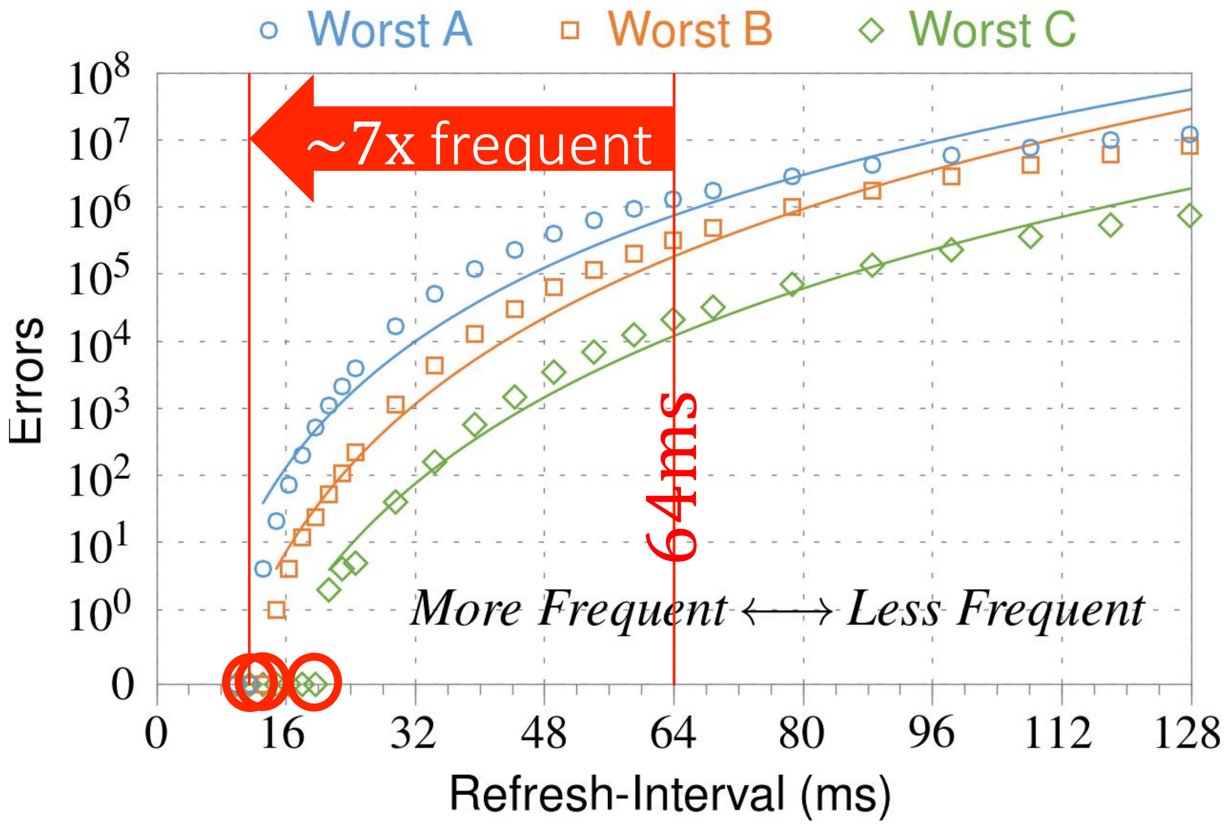
➊ Access Interval (Aggressor)



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

Less frequent accesses → Fewer errors

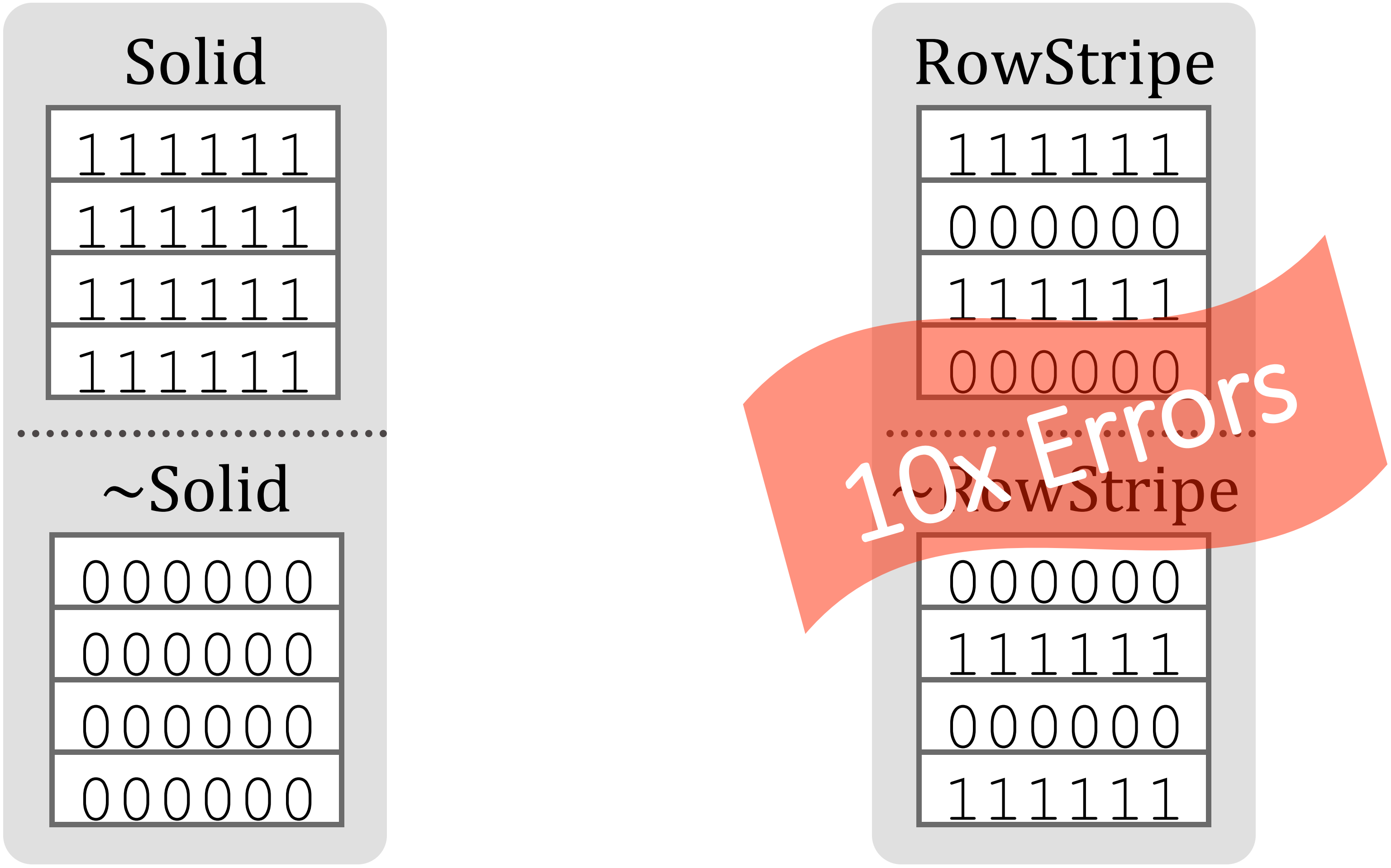
❷ Refresh Interval



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

Naive Solutions

❶ *Throttle accesses to same row*

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

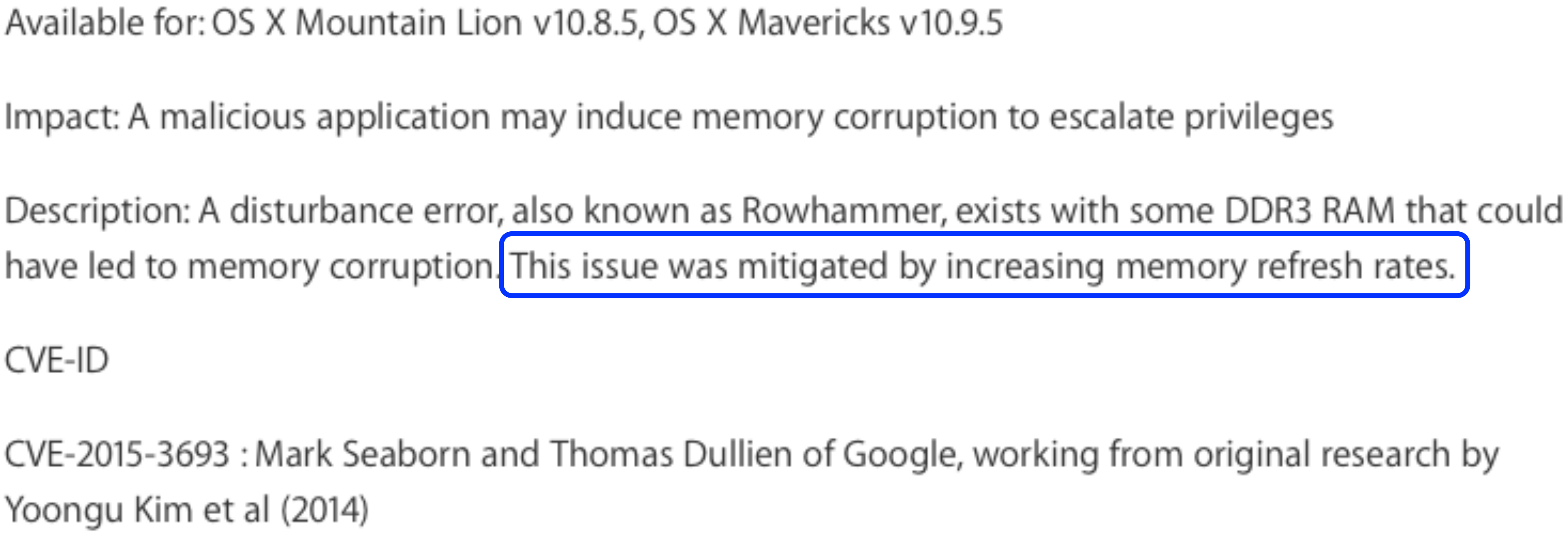
❷ *Refresh more frequently*

- 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: $\mathrm{p} = 0.005$

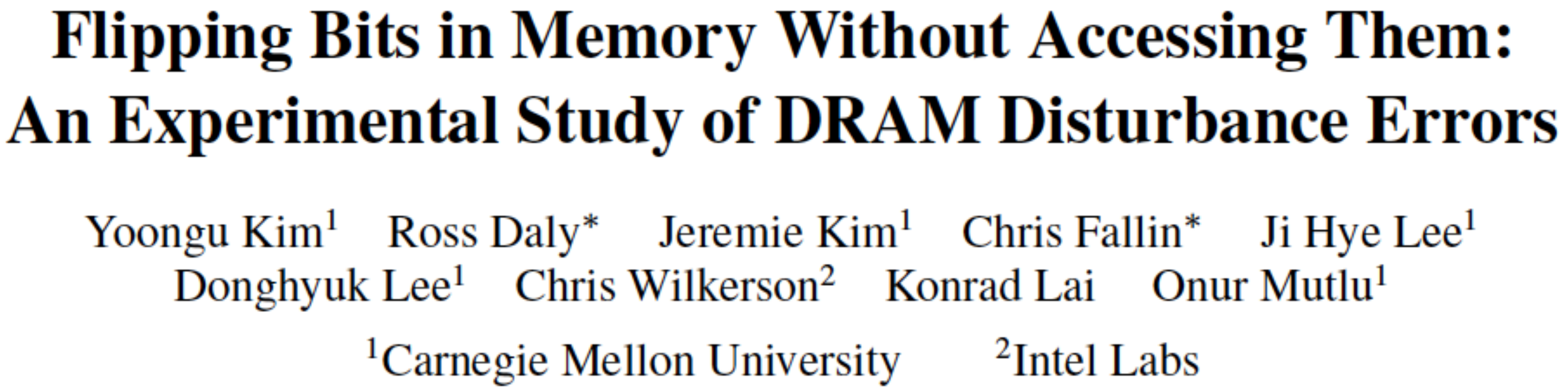
- Reliability Guarantee
  - When $\mathrm{p}{=}0.005$, errors in one year: 9.4×10^{-14}
  - By adjusting the value of p, we can provide an arbitrarily strong protection against errors

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*

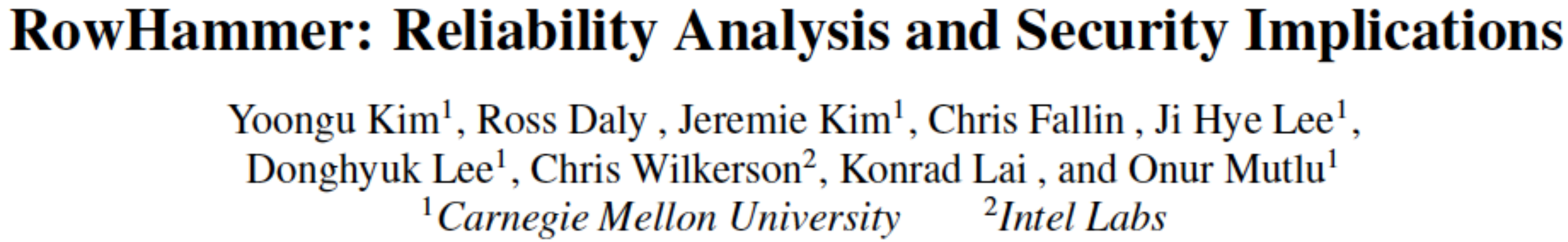
More on RowHammer Analysis



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: Reliability Analysis and Security Implications

Yoongu Kim[1], Ross Daly , Jeremie Kim[1], Chris Fallin , Ji Hye Lee[1],
Donghyuk Lee[1], Chris Wilkerson[2], Konrad Lai , and Onur Mutlu[1]
[1]*Carnegie Mellon University* [2]*Intel Labs*

Future of Main Memory

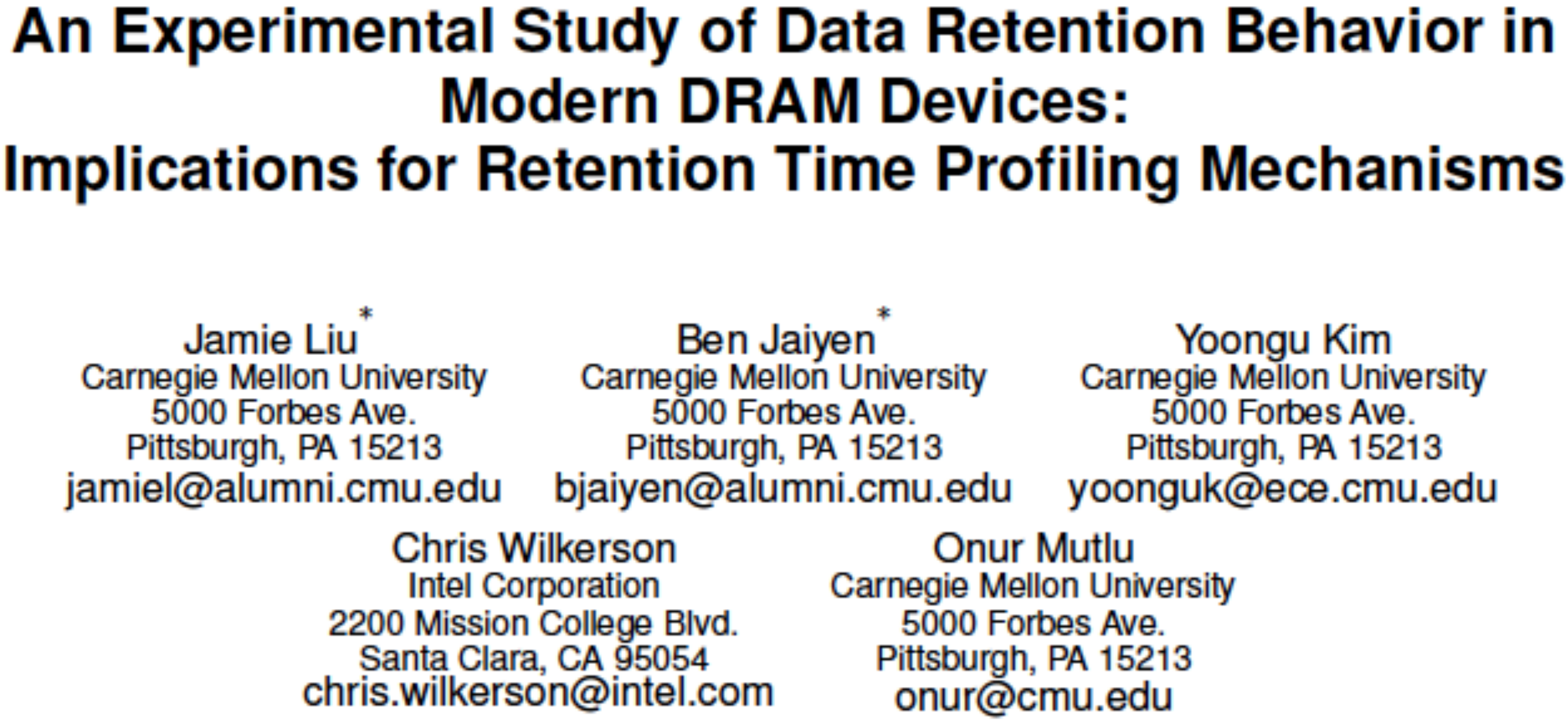
- DRAM is becoming less reliable → more vulnerable

- Due to difficulties in DRAM scaling, unexpected types of failures may appear

- And, they may already be slipping into the field
  - Read disturb errors (Rowhammer)
  - Retention errors
  - Read errors, write errors
  - …

- These failures can also pose security vulnerabilities

Analysis of Retention Failures **[ISCA'13]**



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

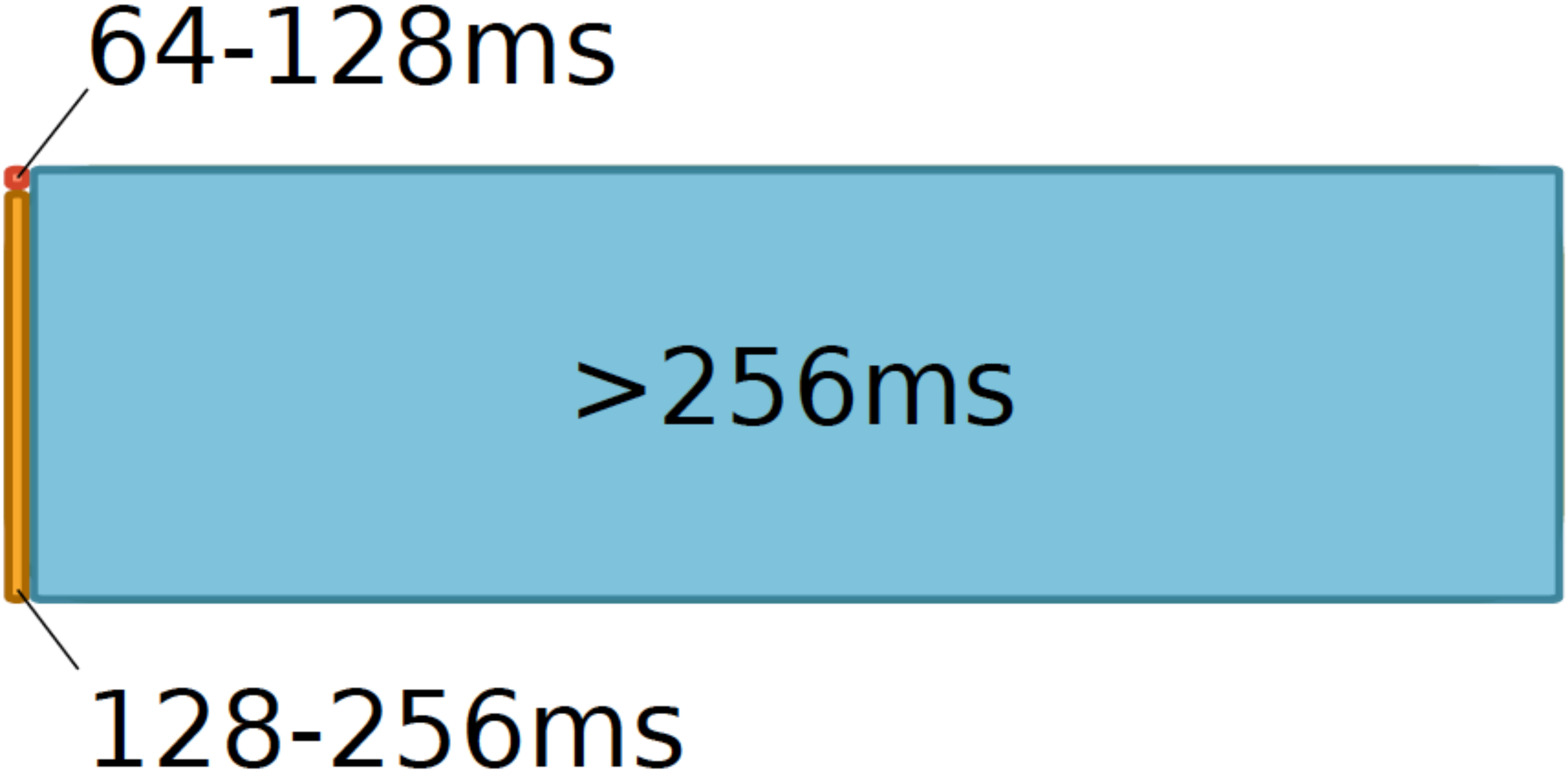
Ben Jaiyen[*]
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
bjaiyen@alumni.cmu.edu

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

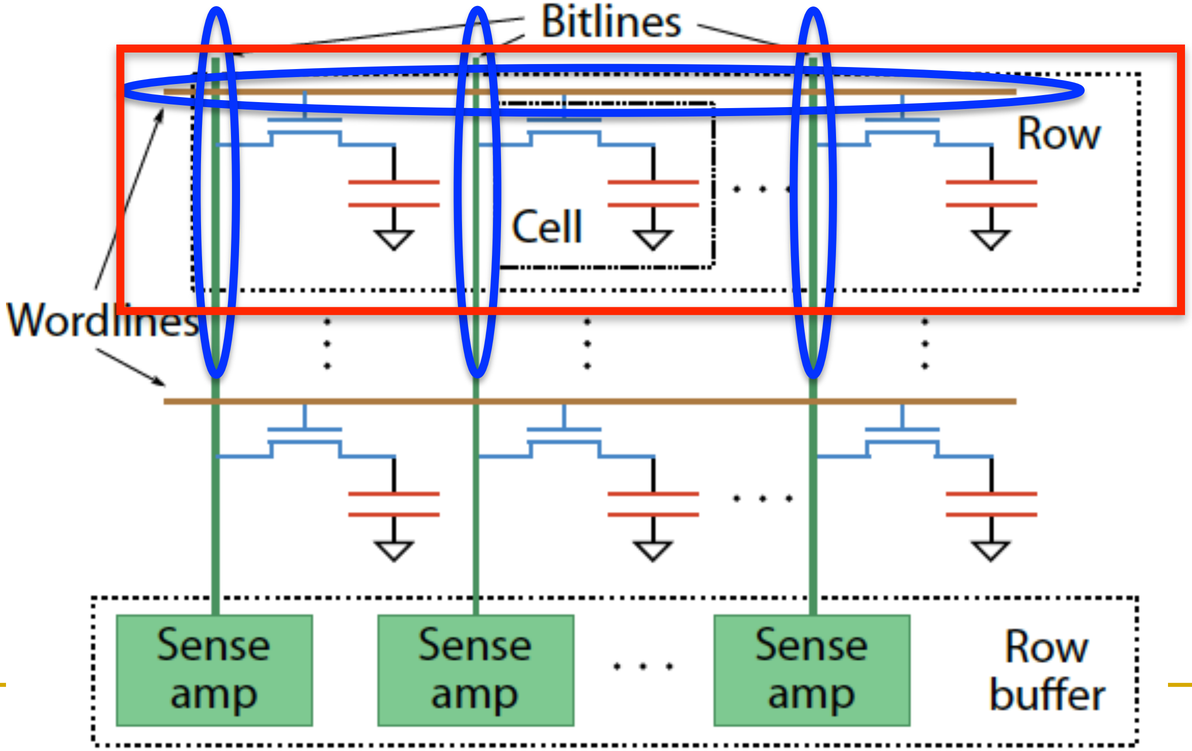
Onur Mutlu
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
onur@cmu.edu

Can We Exploit the DRAM Retention Time Profile?



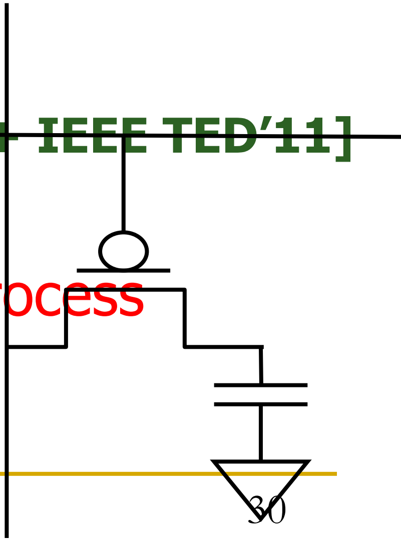
Two Challenges to Retention Time Profiling

- 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

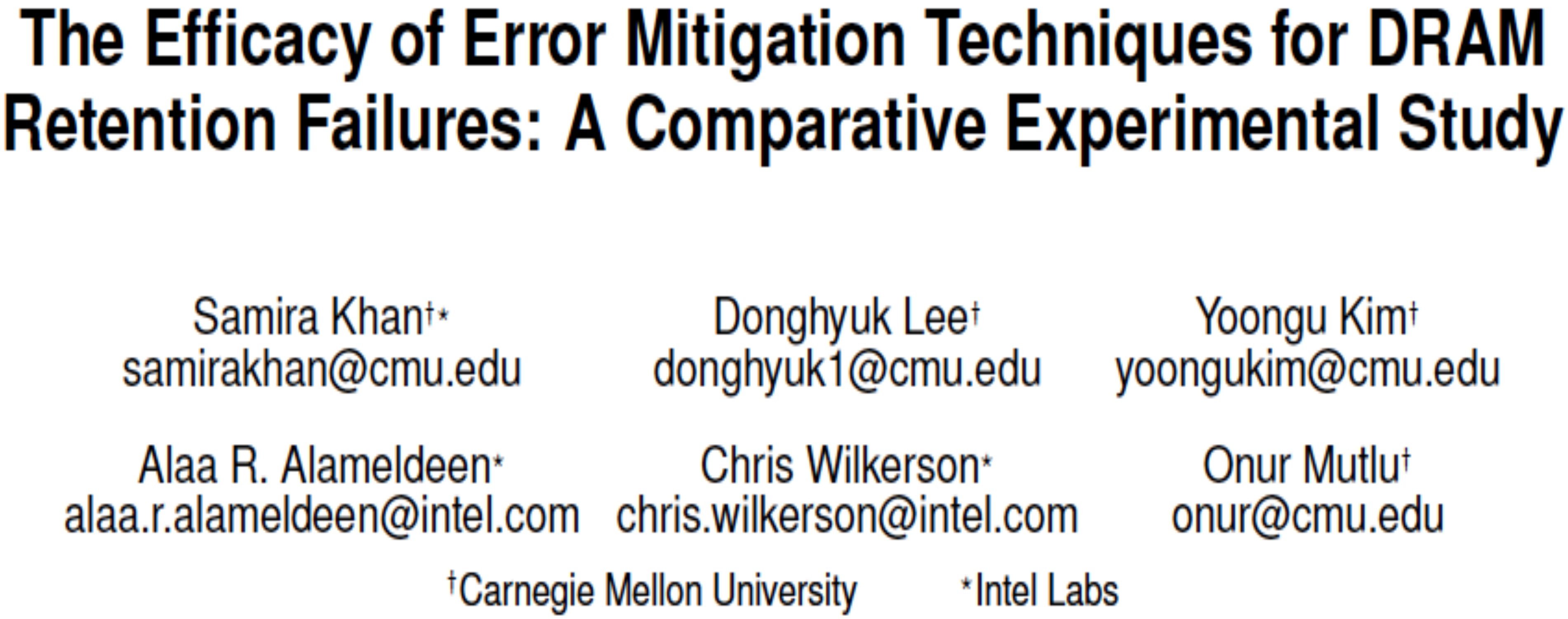


Two Challenges to Retention Time Profiling

- Challenge 2: Variable Retention Time (VRT)
  - Retention time of a DRAM cell changes randomly over time
    - a cell alternates between multiple retention time states
  - Leakage current of a cell changes sporadically due to a charge trap in the gate oxide of the DRAM cell access transistor
  - When the trap becomes occupied, charge leaks more readily from the transistor's drain, leading to a short retention time
    - Called *Trap-Assisted Gate-Induced Drain Leakage*
  - This process appears to be a random process **[Kim+ IEEE TED'11]**
  - Worst-case retention time depends on a random process → need to find the worst case despite this



Mitigation of Retention Issues [SIGMETRICS'14]



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

Samira Khan[†][*]
samirakhan@cmu.edu

Donghyuk Lee[†]
donghyuk1@cmu.edu

Yoongu Kim[†]
yoongukim@cmu.edu

Alaa R. Alameldeen[*]
alaa.r.alameldeen@intel.com

Chris Wilkerson[*]
chris.wilkerson@intel.com

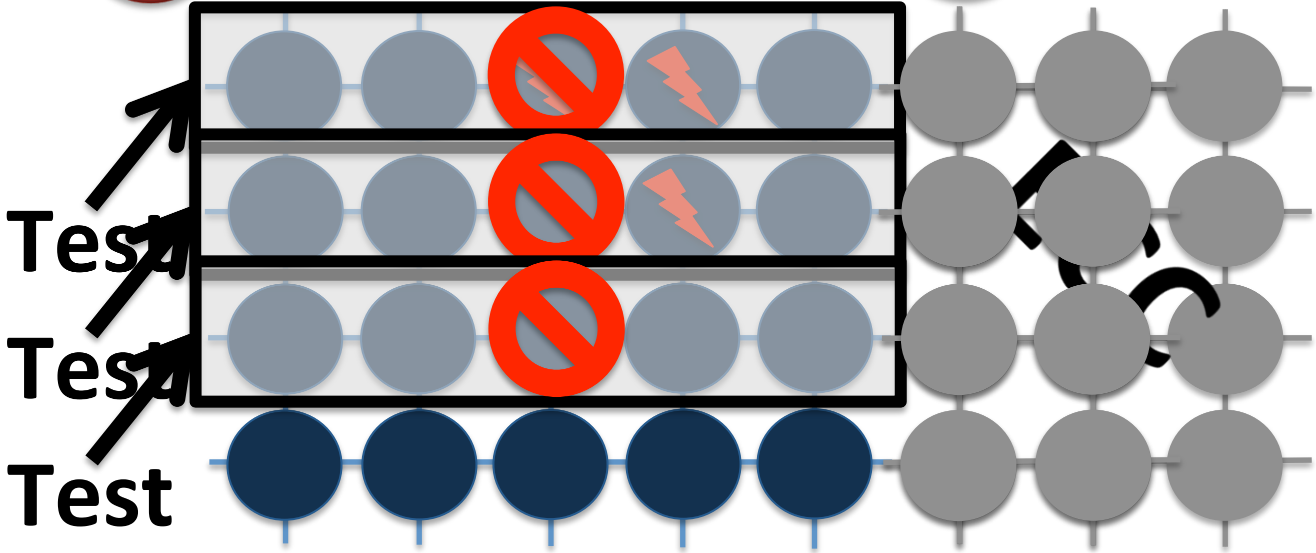
Onur Mutlu[†]
onur@cmu.edu

[†]Carnegie Mellon University [*]Intel Labs

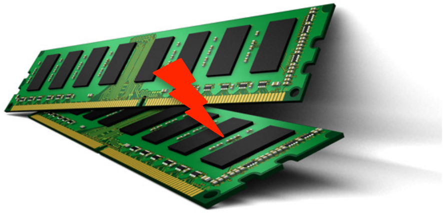
Online Profiling of DRAM In the Field

Initially protect DRAM with ECC 1

Periodically test parts of DRAM 2



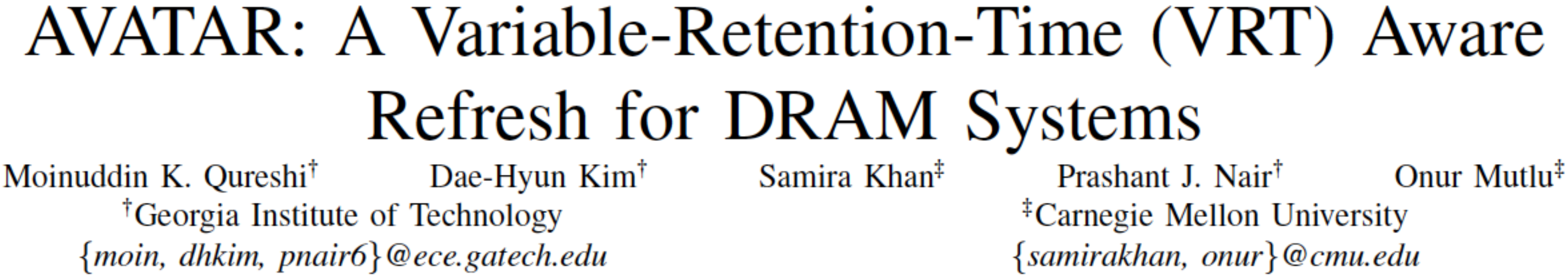
Adjust refresh rate and reduce ECC 3



Optimize DRAM and mitigate errors online without disturbing the system and applications

Multi-Rate Refresh with Online Profiling & ECC

- Moinuddin Qureshi, Dae Hyun Kim, Samira Khan, Prashant Nair, and Onur Mutlu,
  "AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems"
  Proceedings of the
  45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (***DSN***), Rio de Janeiro, Brazil, June 2015.
  [Slides (pptx) (pdf)]



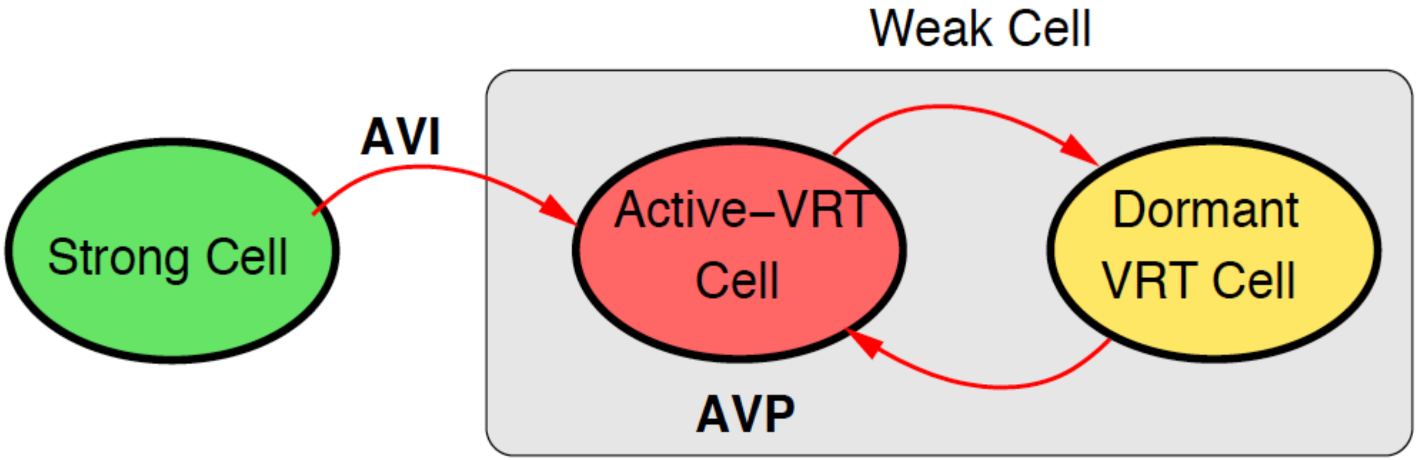
AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems

Moinuddin K. Qureshi[†] Dae-Hyun Kim[†] Samira Khan[‡] Prashant J. Nair[†] Onur Mutlu[‡]

[†]Georgia Institute of Technology
{moin, dhkim, pnair6}@ece.gatech.edu

[‡]Carnegie Mellon University
{samirakhan, onur}@cmu.edu

ARCHITECTURE MODEL FOR CELL UNDER VRT



Two key parameters:

Active-VRT Pool (AVP): How many VRT cells in this period?

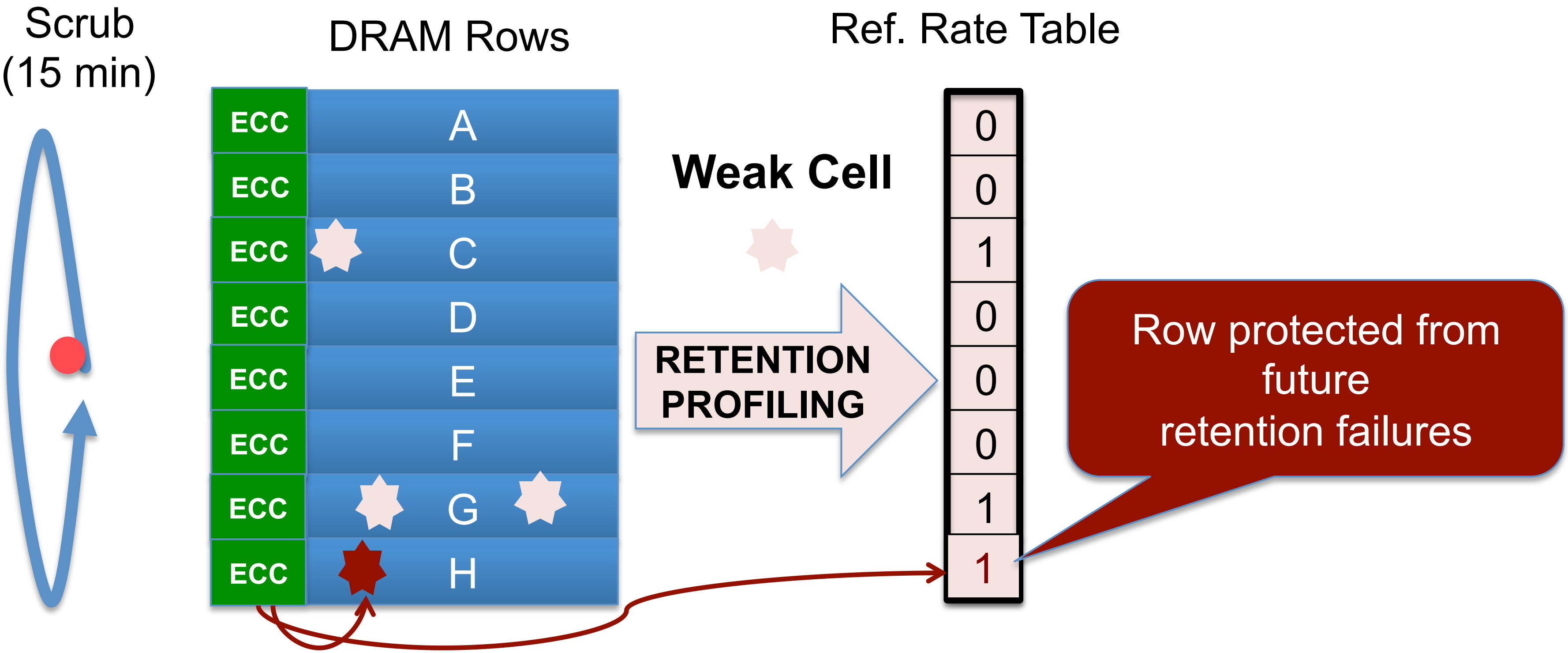
Active-VRT Injection (AVI): How many new (previously undiscovered) cells became weak in this period?

Model has two parameters: AVP and AVI

AVATAR

Insight: Avoid forming Active VRT Pool ➔ Upgrade on ECC error

Observation: Rate of VRT >> Rate of soft error (50x-2500x)

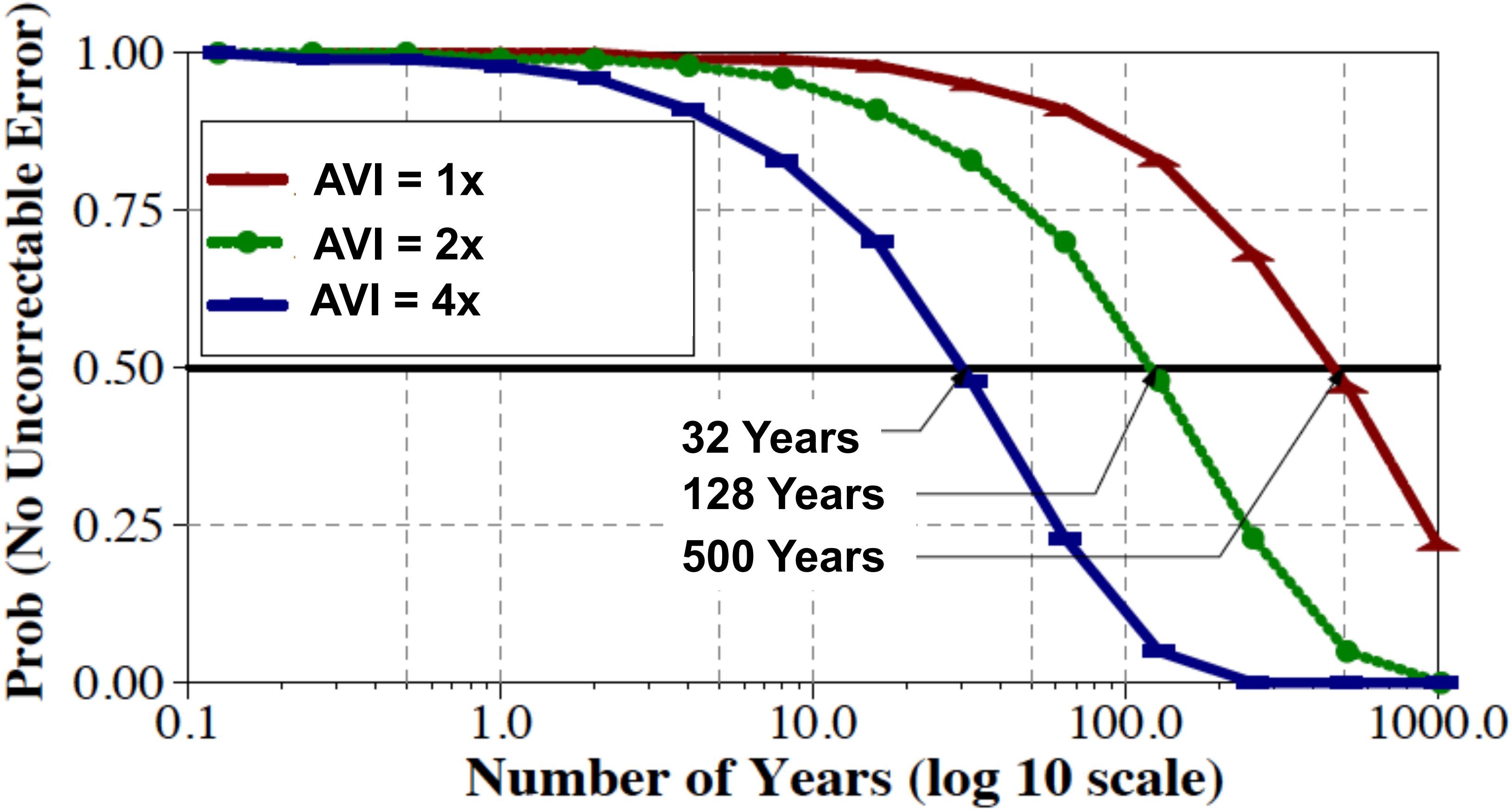


Row protected from future retention failures

AVATAR mitigates VRT by breaking AVP Pool

AVATAR: TIME TO FAILURE

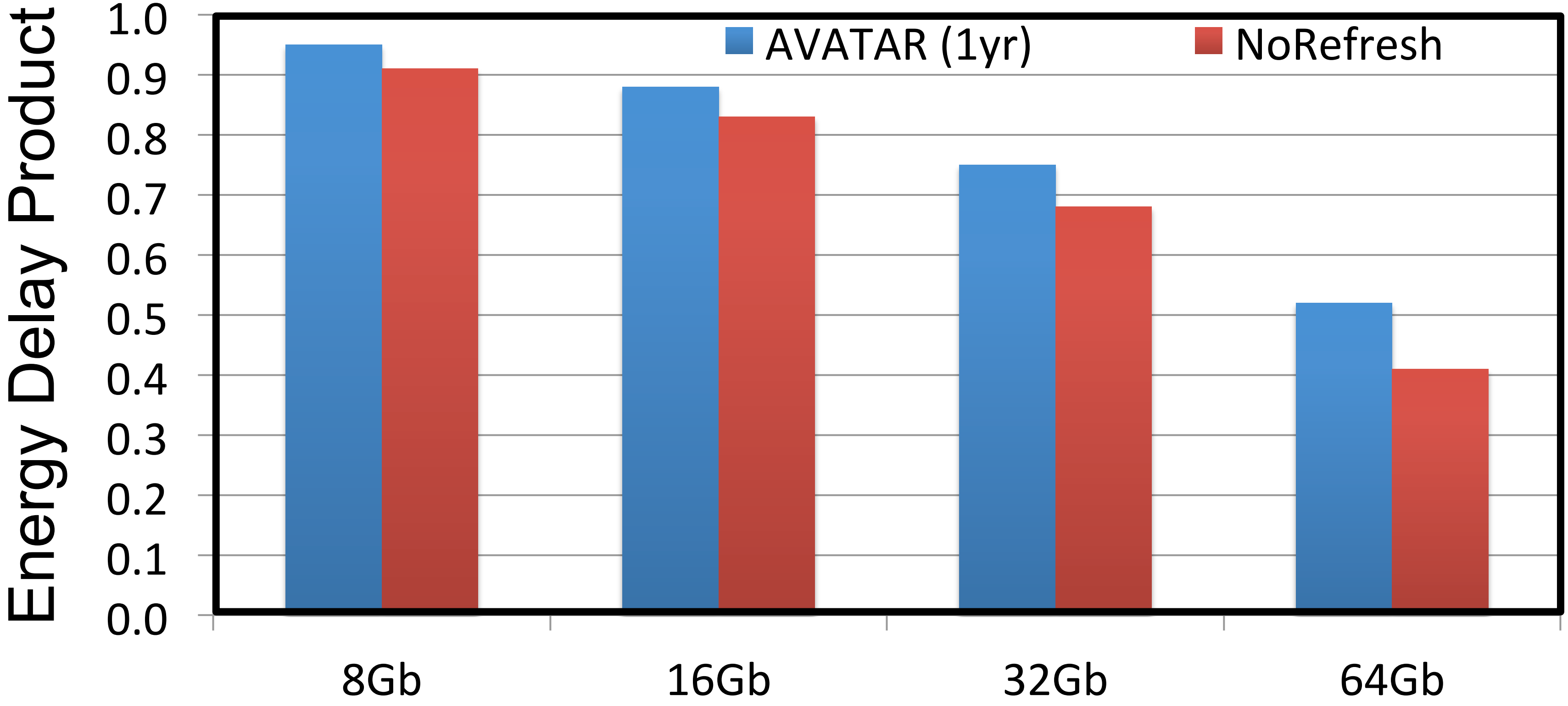
System: Four channels, each with 8GB DIMM



AVATAR increases time to failure to 10s of years

* We include the effect of soft error in the above lifetime analysis (details in the paper)

ENERGY DELAY PRODUCT

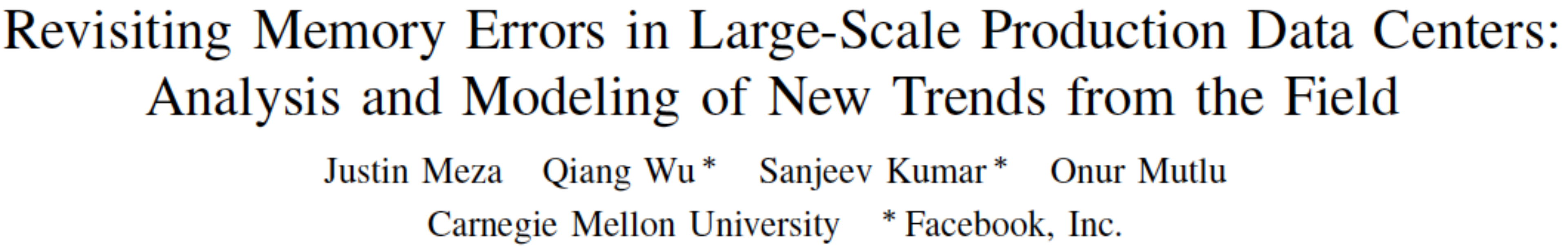


AVATAR reduces EDP,
Significant reduction at higher capacity nodes

Large-Scale Failure Analysis of DRAM Chips

- Analysis and modeling of memory errors found in all of Facebook’s server fleet

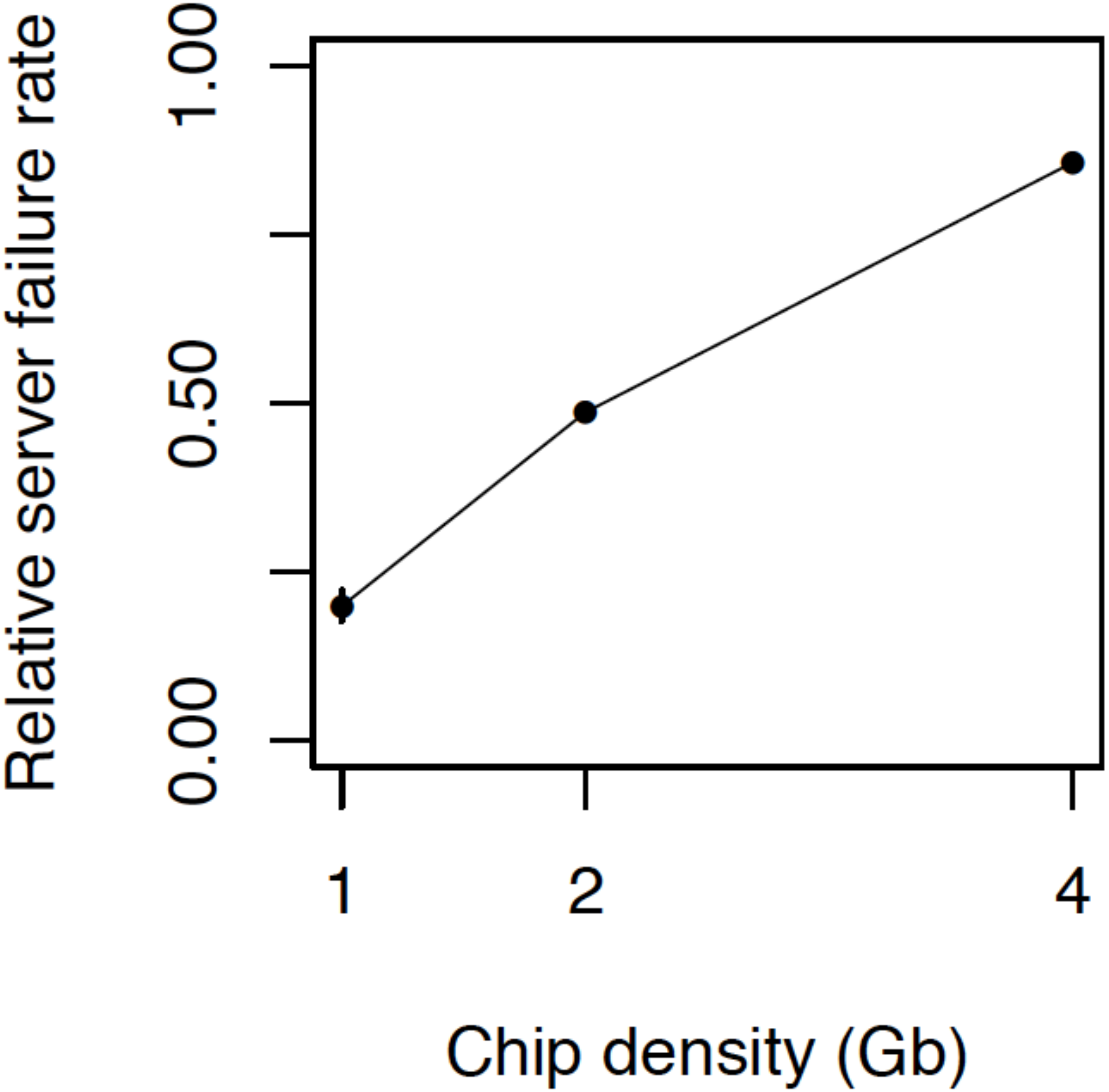
- Justin Meza, Qiang Wu, Sanjeev Kumar, and Onur Mutlu, **"Revisiting Memory Errors in Large-Scale Production Data Centers: Analysis and Modeling of New Trends from the Field"** *Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks* (***DSN***), Rio de 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

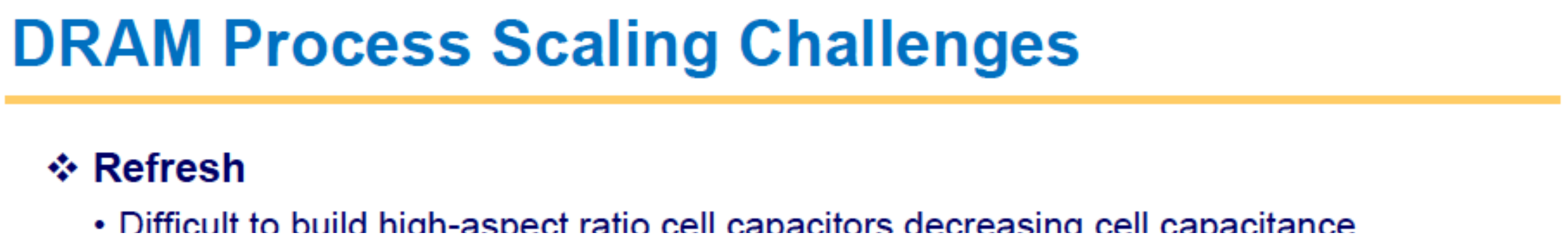
Justin Meza Qiang Wu * Sanjeev Kumar * Onur Mutlu
Carnegie Mellon University * Facebook, Inc.

DRAM Reliability Reducing



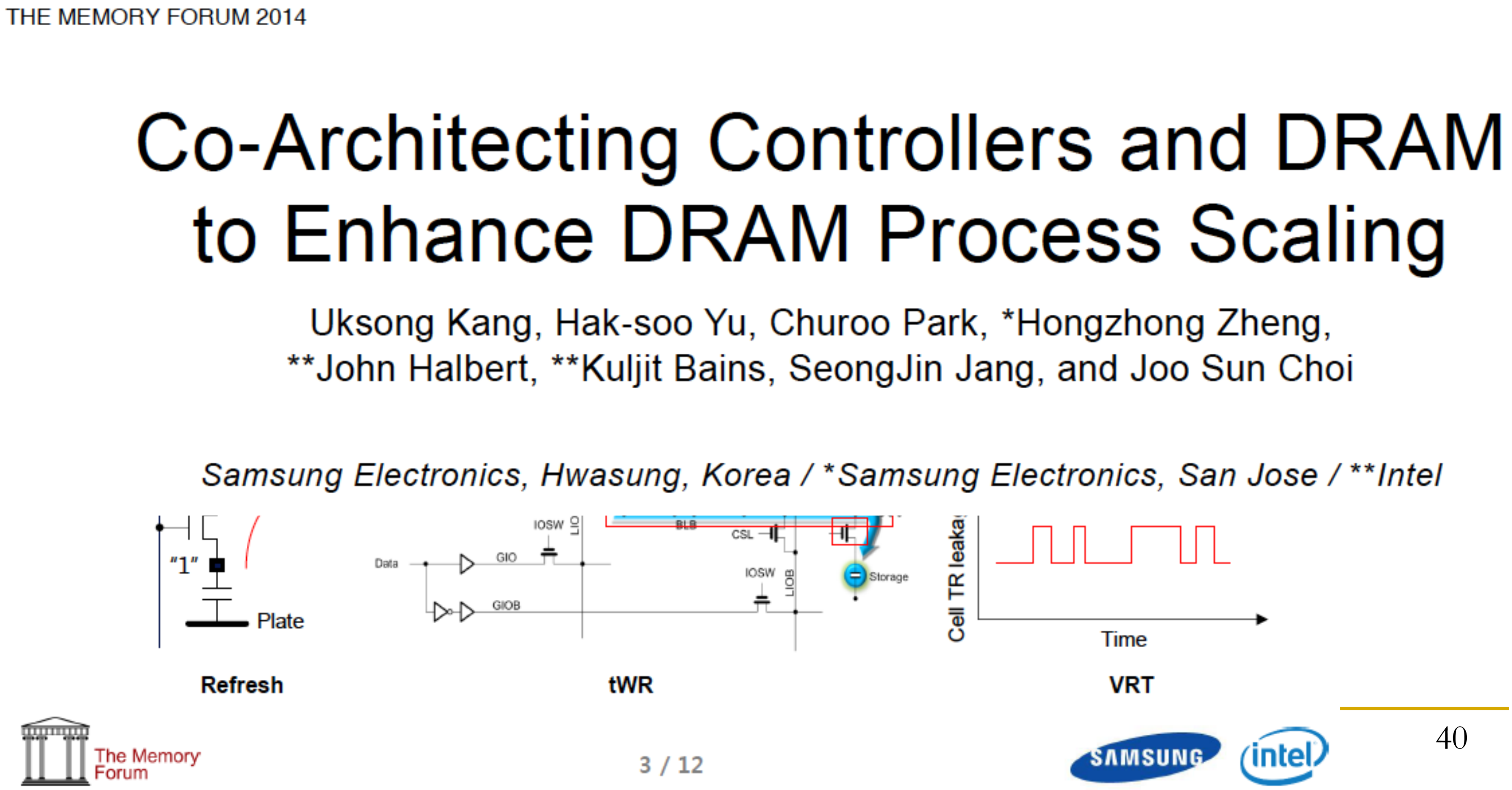
Intuition: quadratic increase in capacity

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*

Refresh

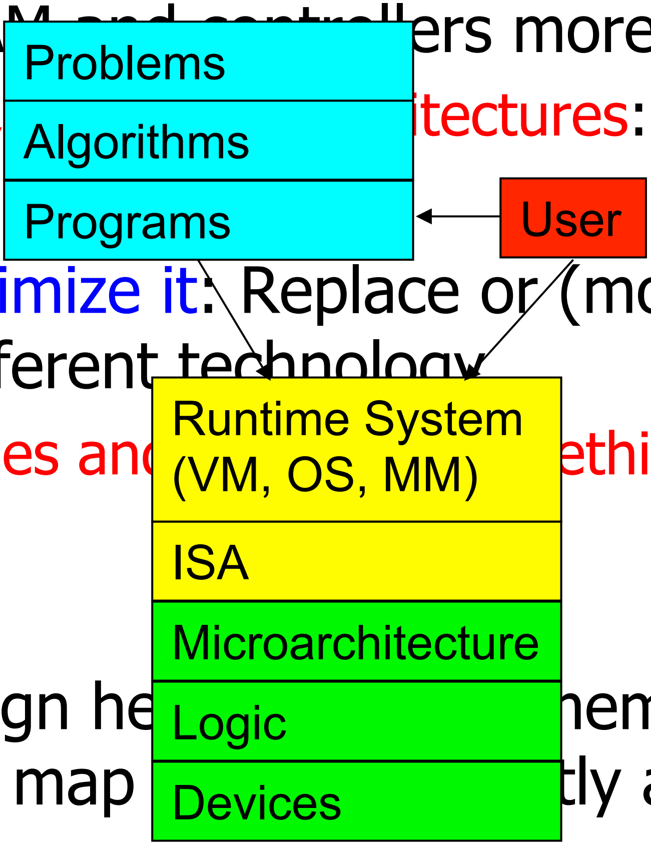
tWR

VRT

3 / 12

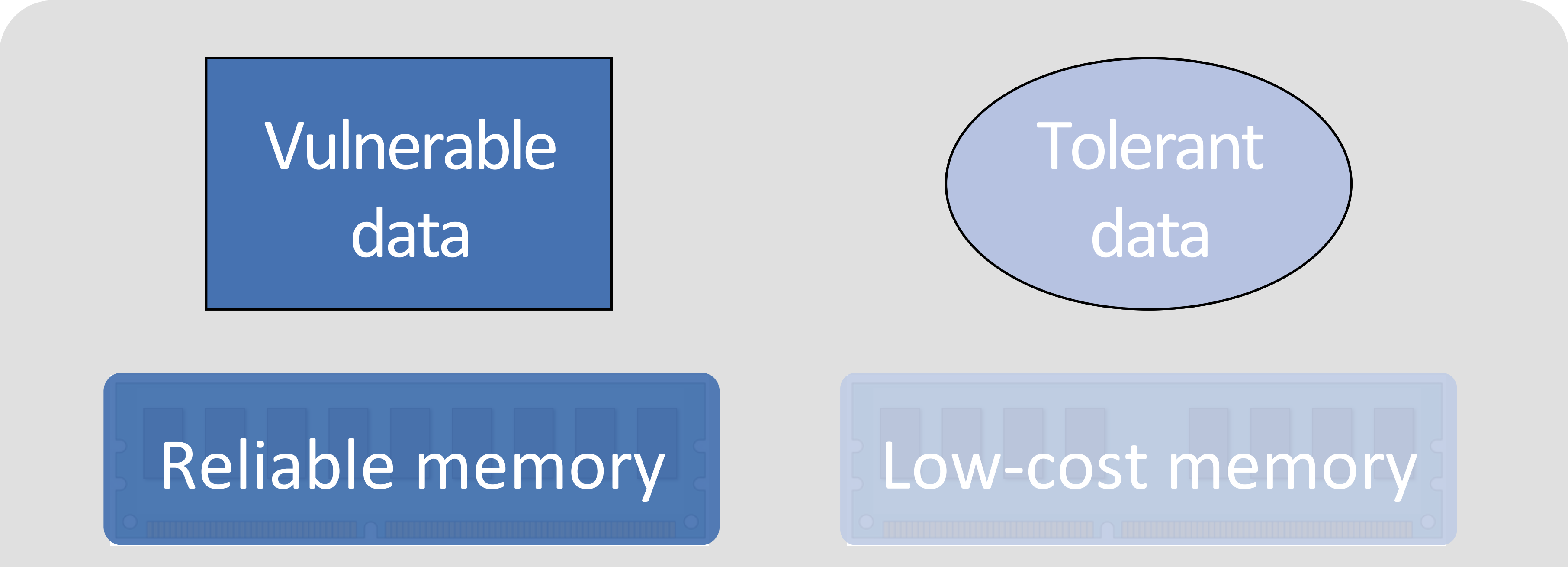
How Do We Solve The Problem?

- Fix it: Make DRAM and controllers more intelligent
  - New interfaces, itectures: system-DRAM codesign
- Eliminate or minimize it: Replace or (more likely) augment DRAM with a different technology
  - New technologies an ethinking of memory & storage
- Embrace it: Design he nemories (none of which are perfect) and map tly across them
  - New models for data management and maybe usage



Solutions (to memory scaling) require software/hardware/device cooperation

Exploiting Memory Error Tolerance with Hybrid Memory Systems



On Microsoft's Web Search workload

Reduces server hardware cost by 4.7 %

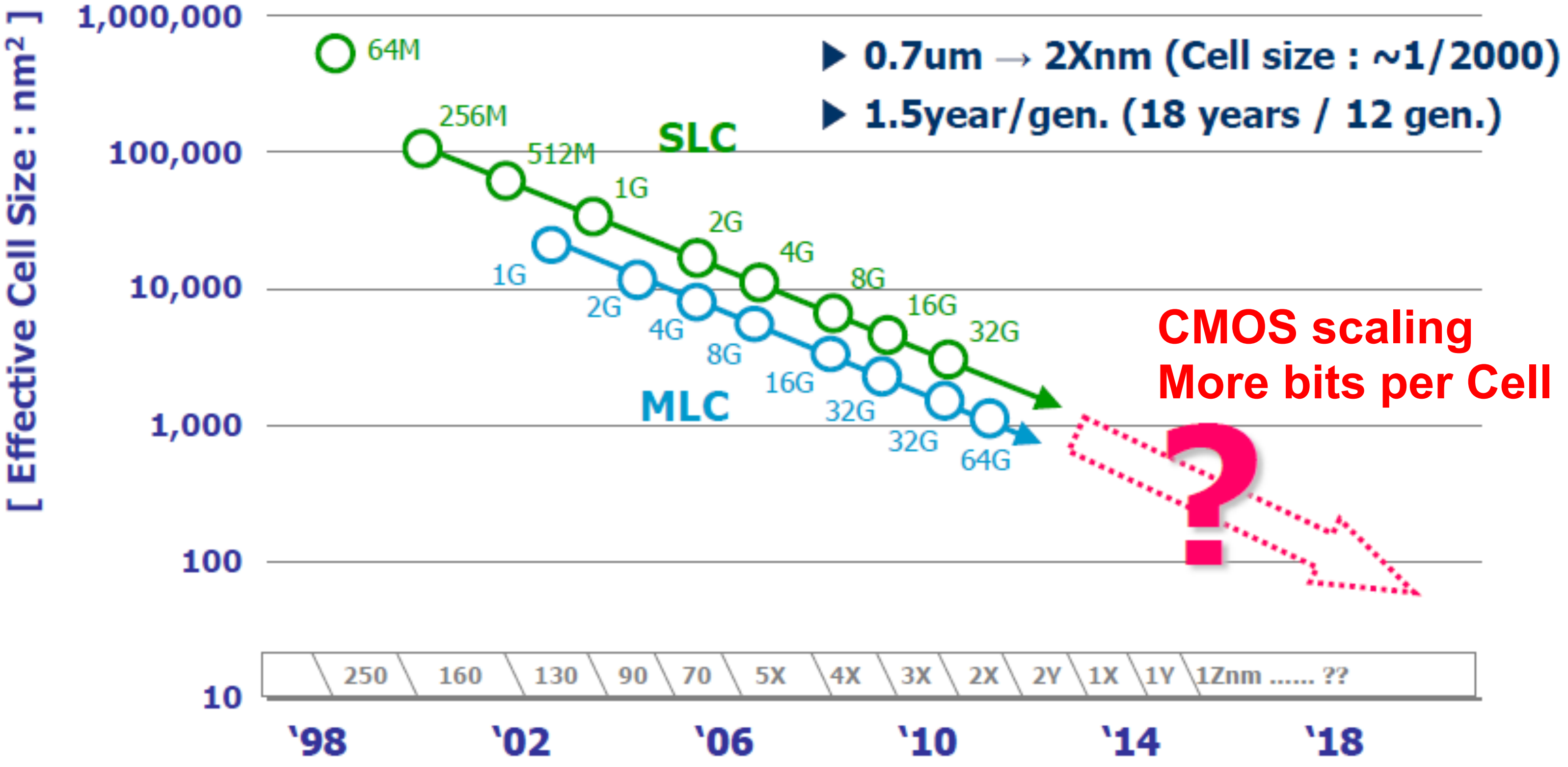
Achieves single server availability target of 99.90 %

Heterogeneous-**R**eliability **M**emory [DSN 2014]

Agenda

- DRAM Scaling Issues
  - DRAM RowHammer Problem
  - Some Other DRAM Reliability Studies
- NAND Flash Scaling Issues
  - Some NAND Flash Reliability Studies
  - Read Disturb Errors in NAND Flash Memory
- Summary and Discussion

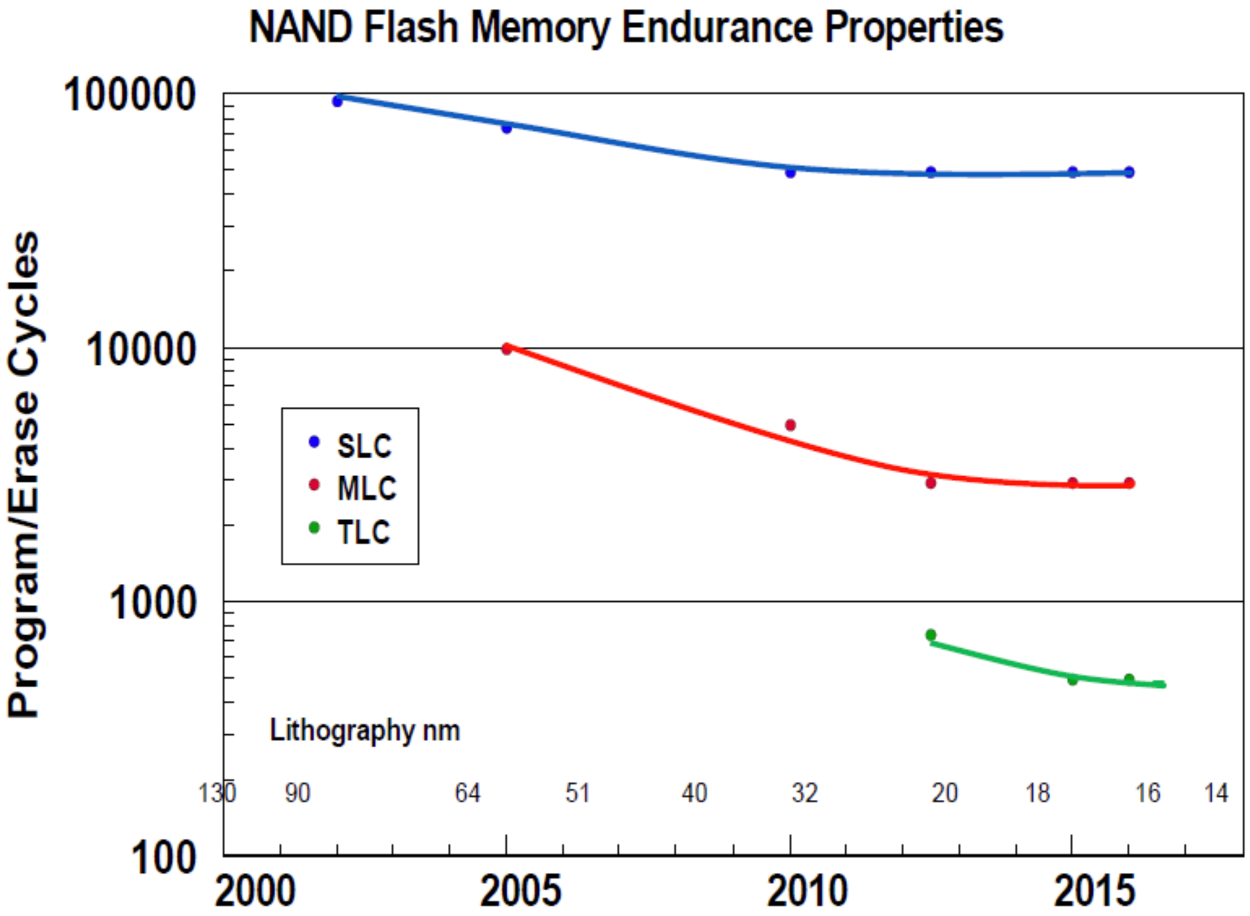
Evolution of NAND Flash Memory



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



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

- **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

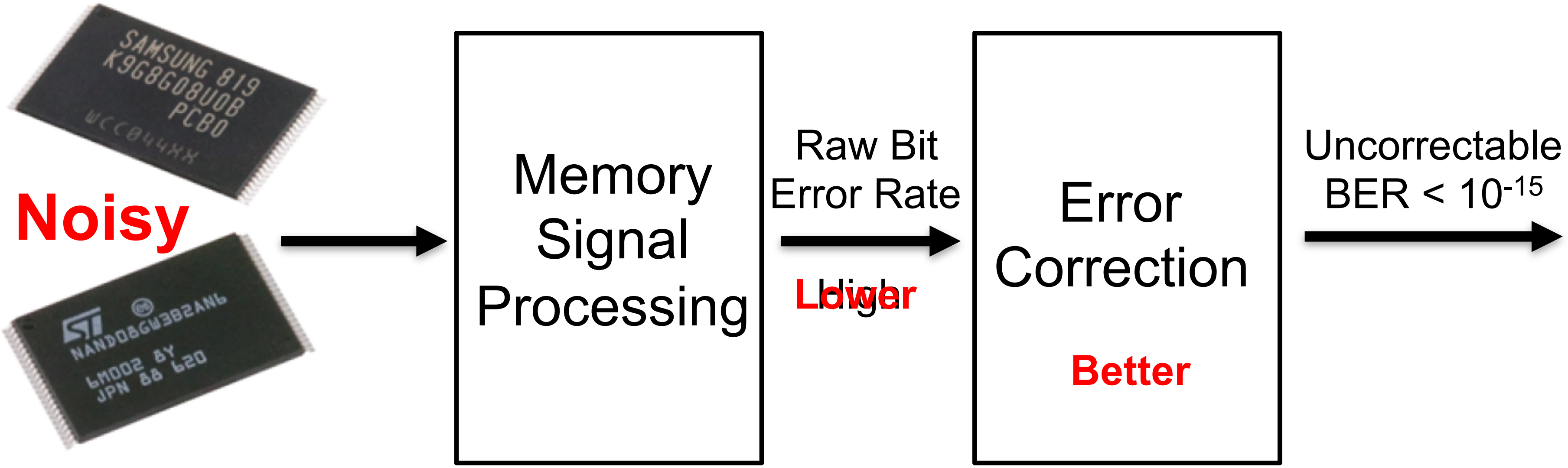
NAND Flash Memory is Increasingly Noisy

Write

Noisy NAND

Read

Future NAND Flash-based Storage Architecture



Noisy

Memory Signal Processing

Raw Bit Error Rate

Higher

Lower

Error Correction

Better

Uncorrectable BER < 10^{-15}

Our Goals:

Build reliable error models for NAND flash memory

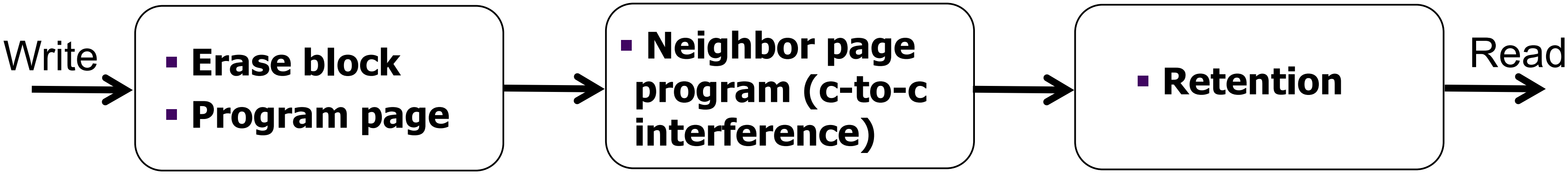
Design efficient reliability mechanisms based on the model

NAND Flash Error Model

Write
Noisy NAND
Read

Experimentally characterize and model dominant errors

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



Write

- **Erase block**
- **Program page**

- **Neighbor page program (c-to-c interference)**

- **Retention**

Read

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

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**

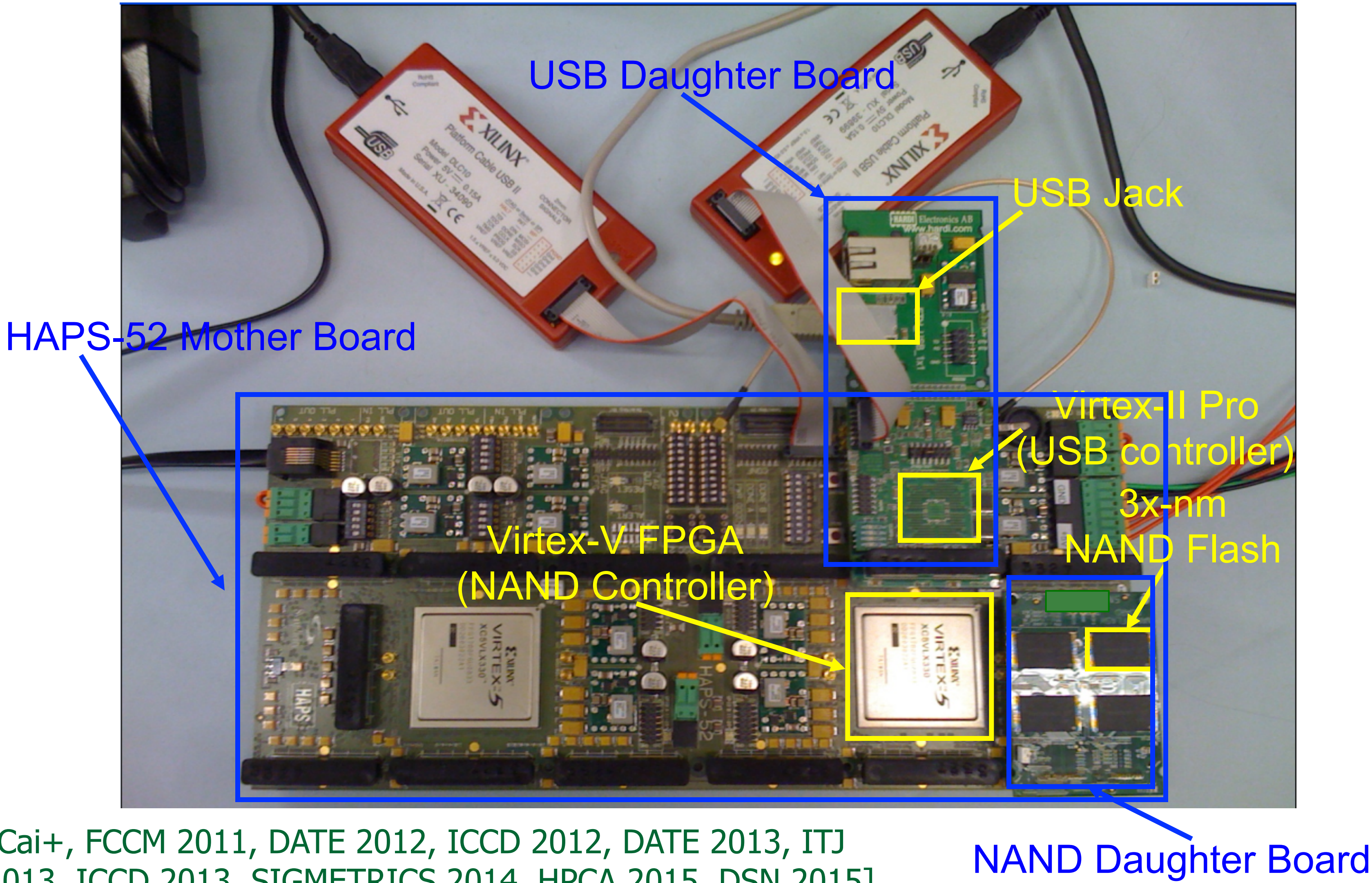
Cai et al**.**, “Data Retention in MLC NAND Flash Memory: Characterization, Optimization and Recovery" **, HPCA 2015**

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

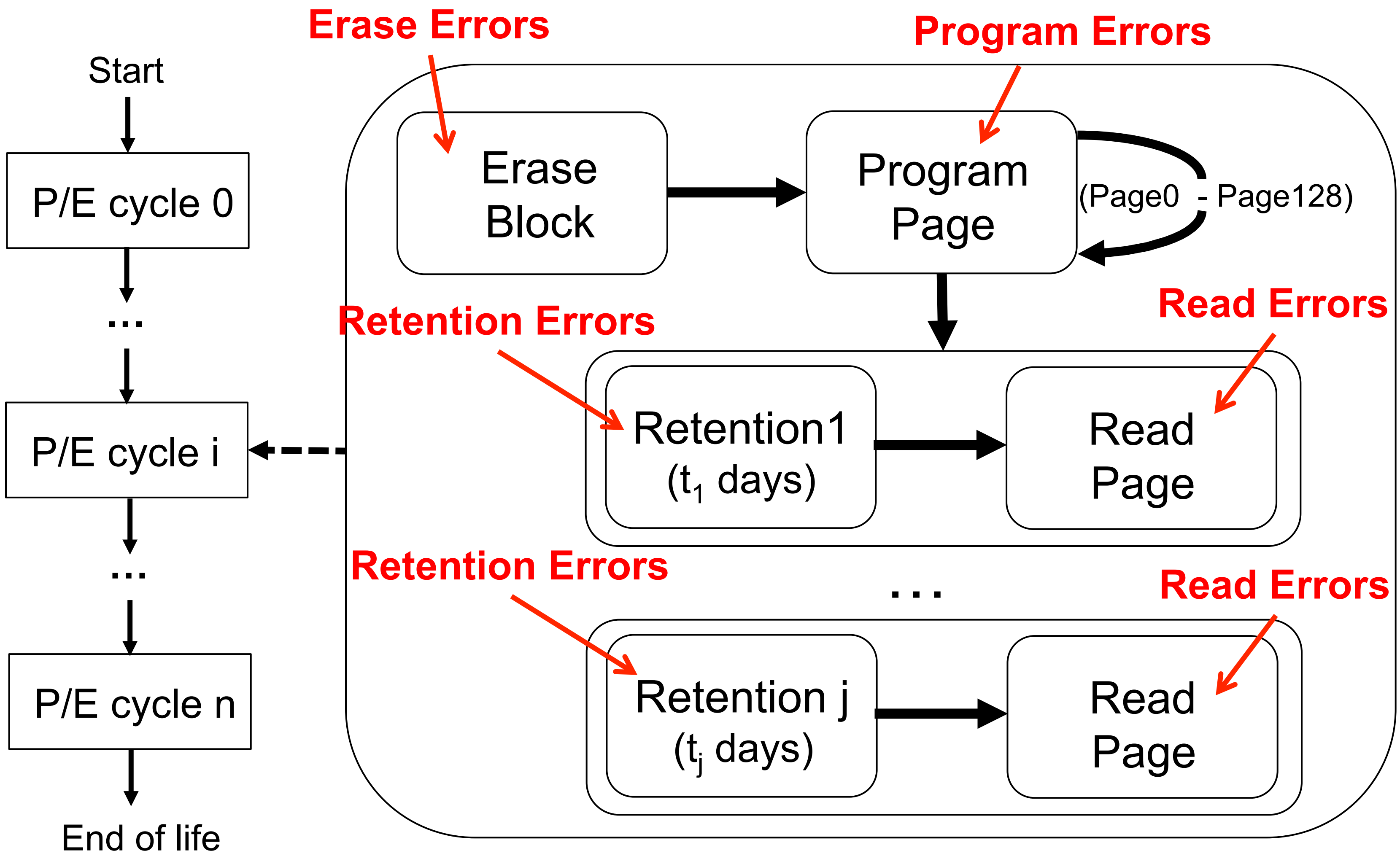
Experimental Testing Platform



[Cai+, FCCM 2011, DATE 2012, ICCD 2012, DATE 2013, ITJ 2013, ICCD 2013, SIGMETRICS 2014, HPCA 2015, DSN 2015]

Cai et al., FPGA-based Solid-State Drive prototyping platform, FCCM 2011.

NAND Flash Usage and Error Model



Methodology: Error and ECC Analysis

- Characterized errors and error rates of 3x and 2y-nm MLC NAND flash using an experimental FPGA-based platform
  - **[Cai+, DATE'12, ICCD'12, DATE'13, ITJ'13, ICCD'13, SIGMETRICS'14]**

- Quantified Raw Bit Error Rate (RBER) at a given P/E cycle
  - Raw Bit Error Rate: Fraction of erroneous bits without any correction

- Quantified error correction capability (and area and power consumption) of various BCH-code implementations
  - Identified how much RBER each code can tolerate

    → how many P/E cycles (flash lifetime) each code can sustain

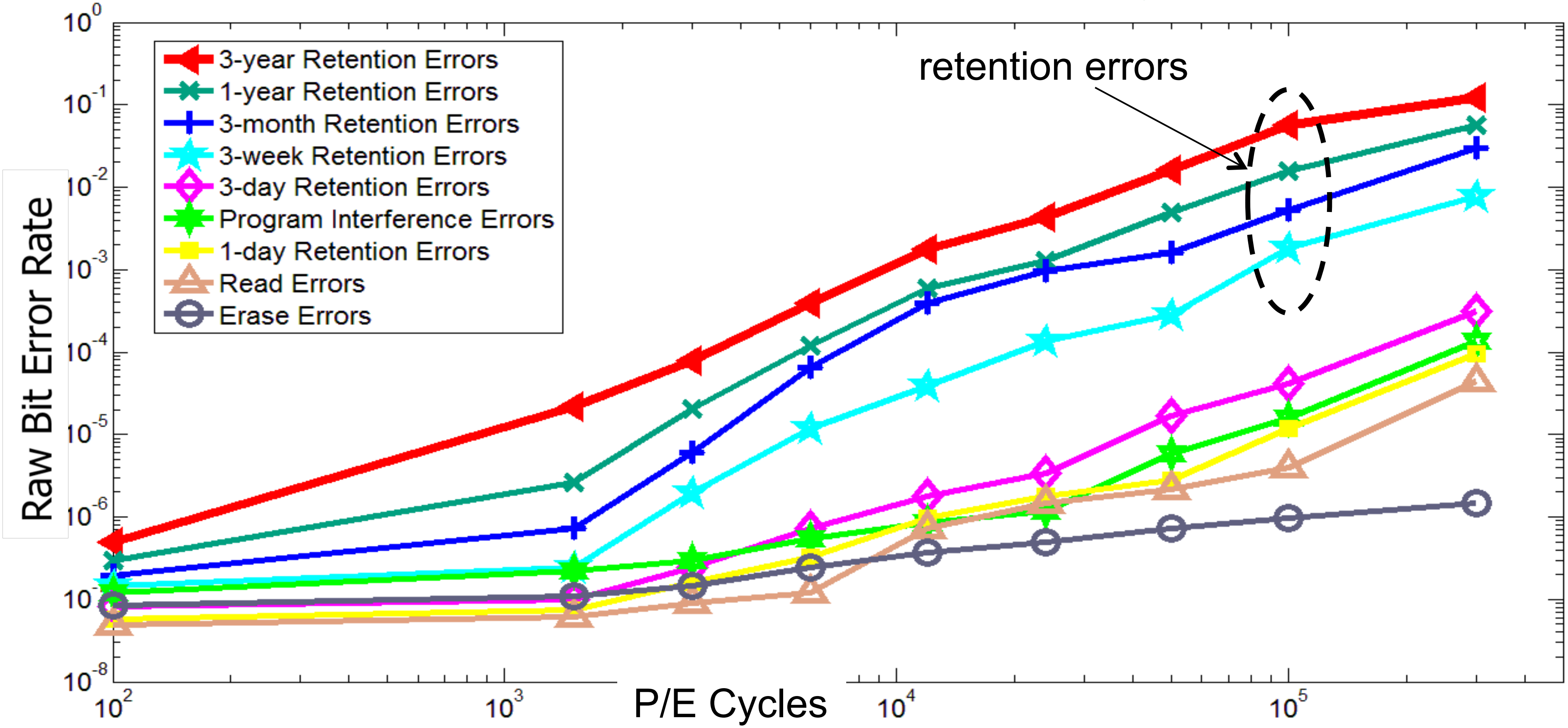
NAND Flash Error Types

- 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

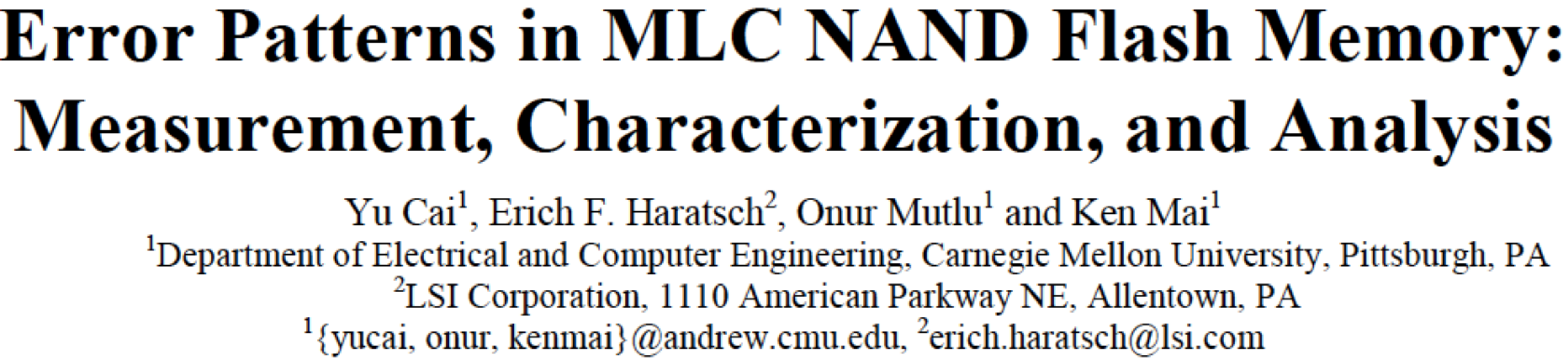


- Raw bit error rate increases exponentially with P/E cycles
- Retention errors are dominant (>99% for 1-year ret. time)
- Retention errors increase with retention time requirement

Cai et al., Error Patterns in MLC NAND Flash Memory, DATE 2012.

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)

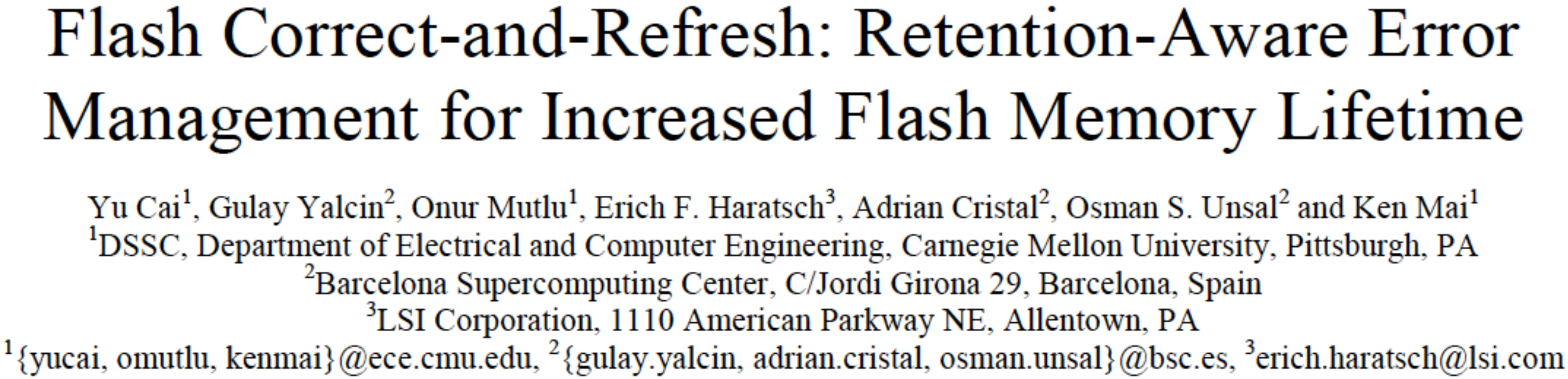


Error Patterns in MLC NAND Flash Memory: Measurement, Characterization, and Analysis

Yu Cai[1], Erich F. Haratsch[2], Onur Mutlu[1] and Ken Mai[1]
[1]Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA
[2]LSI Corporation, 1110 American Parkway NE, Allentown, PA
[1]{yucai, onur, kenmai}@andrew.cmu.edu, [2]erich.haratsch@lsi.com

Flash Correct and Refresh

- 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"** *Proceedings of the 30th IEEE International Conference on Computer Design* (***ICCD***), Montreal, Quebec, Canada, September 2012. Slides (ppt) (pdf)

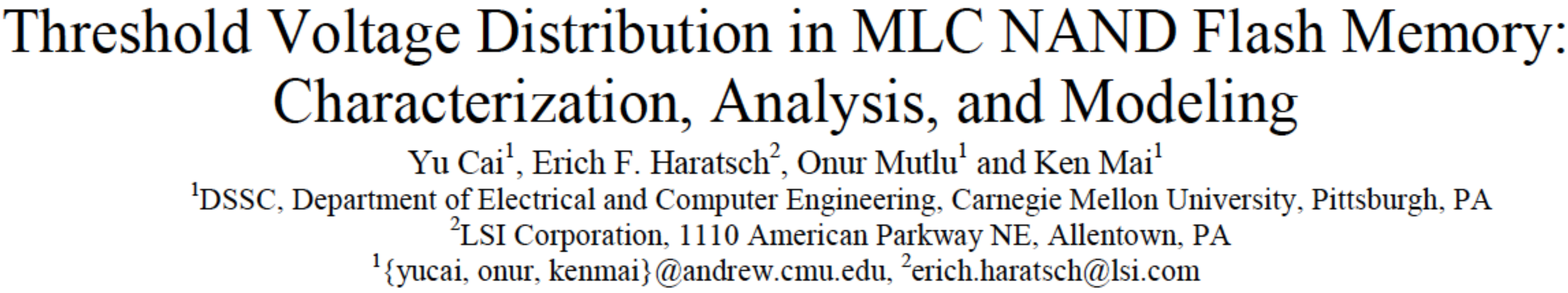


Flash Correct-and-Refresh: Retention-Aware Error Management for Increased Flash Memory Lifetime

Yu Cai[1], Gulay Yalcin[2], Onur Mutlu[1], Erich F. Haratsch[3], Adrian Cristal[2], Osman S. Unsal[2] and Ken Mai[1]
[1]DSSC, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA
[2]Barcelona Supercomputing Center, C/Jordi Girona 29, Barcelona, Spain
[3]LSI Corporation, 1110 American Parkway NE, Allentown, PA
[1]{yucai, omutlu, kenmai}@ece.cmu.edu, [2]{gulay.yalcin, adrian.cristal, osman.unsal}@bsc.es, [3]erich.haratsch@lsi.com

Threshold Voltage Modeling

- Yu Cai, Erich F. Haratsch, Onur Mutlu, and Ken Mai, **"Threshold Voltage Distribution in MLC NAND Flash Memory: Characterization, Analysis and Modeling"** *Proceedings of the Design, Automation, and Test in Europe Conference (**DATE**)*, Grenoble, France, March 2013. Slides (ppt)

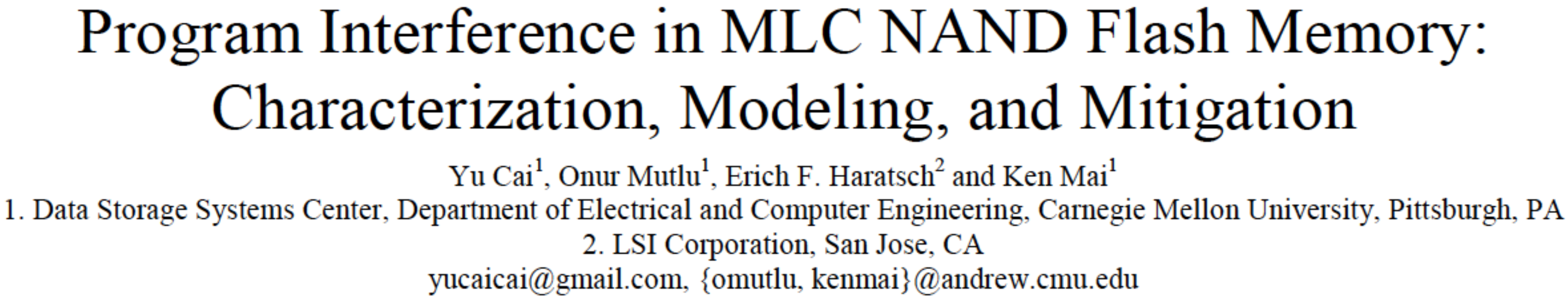


Threshold Voltage Distribution in MLC NAND Flash Memory: Characterization, Analysis, and Modeling

Yu Cai[1], Erich F. Haratsch[2], Onur Mutlu[1] and Ken Mai[1]
[1]DSSC, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA
[2]LSI Corporation, 1110 American Parkway NE, Allentown, PA
[1]{yucai, onur, kenmai}@andrew.cmu.edu, [2]erich.haratsch@lsi.com

Program Interference Modeling

- Yu Cai, Onur Mutlu, Erich F. Haratsch, and Ken Mai, **"Program Interference in MLC NAND Flash Memory: Characterization, Modeling, and Mitigation"** *Proceedings of the 31st IEEE International Conference on Computer Design* (***ICCD***), Asheville, NC, October 2013. Slides (pptx) (pdf) Lightning Session Slides (pdf)

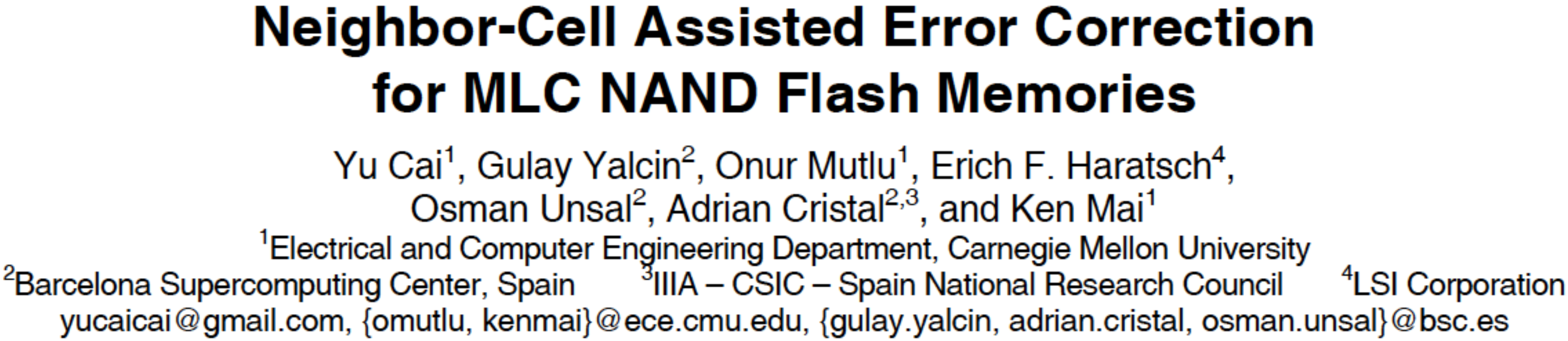


Program Interference in MLC NAND Flash Memory: Characterization, Modeling, and Mitigation

Yu Cai[1], Onur Mutlu[1], Erich F. Haratsch[2] and Ken Mai[1]
1. Data Storage Systems Center, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA
2. LSI Corporation, San Jose, CA
yucaicai@gmail.com, {omutlu, kenmai}@andrew.cmu.edu

Neighbor-Assisted Error Correction

- Yu Cai, Gulay Yalcin, Onur Mutlu, Eric Haratsch, Osman Unsal, Adrian Cristal, and Ken Mai, **"Neighbor-Cell Assisted Error Correction for MLC NAND Flash Memories"** *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (**SIGMETRICS**)*, Austin, TX, June 2014. Slides (ppt) (pdf)

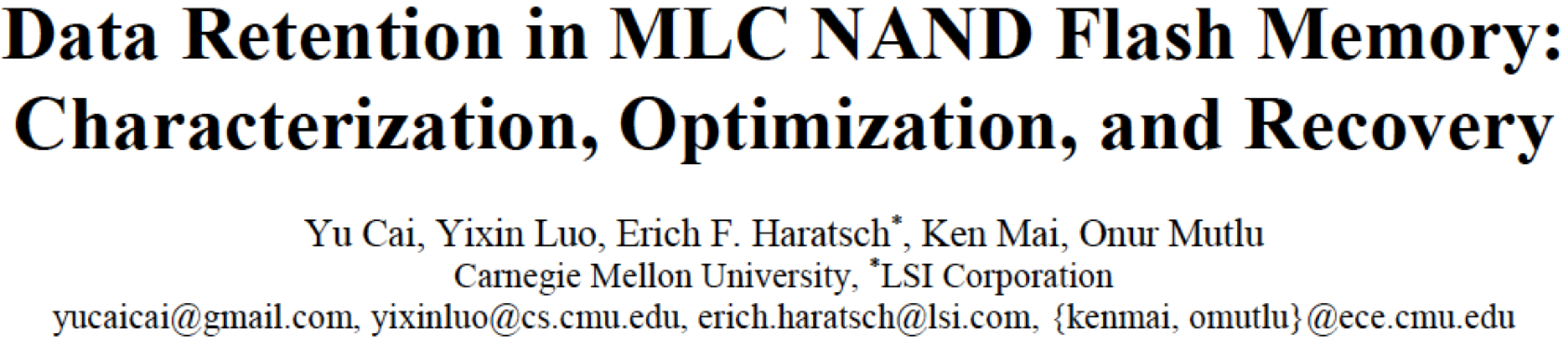


Neighbor-Cell Assisted Error Correction for MLC NAND Flash Memories

Yu Cai[1], Gulay Yalcin[2], Onur Mutlu[1], Erich F. Haratsch[4],
Osman Unsal[2], Adrian Cristal[2,3], and Ken Mai[1]
[1]Electrical and Computer Engineering Department, Carnegie Mellon University
[2]Barcelona Supercomputing Center, Spain [3]IIIA – CSIC – Spain National Research Council [4]LSI Corporation
yucaicai@gmail.com, {omutlu, kenmai}@ece.cmu.edu, {gulay.yalcin, adrian.cristal, osman.unsal}@bsc.es

Data Retention Analysis & Recovery

- 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, yixinluo@cs.cmu.edu, erich.haratsch@lsi.com, {kenmai, omutlu}@ece.cmu.edu

Agenda

- DRAM Scaling Issues
  - DRAM RowHammer Problem
  - Some Other DRAM Reliability Studies
- NAND Flash Scaling Issues
  - Some NAND Flash Reliability Studies
  - Read Disturb Errors in NAND Flash Memory
- Summary and Discussion

Read Disturb Errors in Flash Memory

Paper

- Presented at IEEE/IFIP DSN 2015 Conference in June 2015.
- Full paper for details:
  - 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.
  - http://users.ece.cmu.edu/~omutlu/pub/flash-read-disturb-errors_dsn15.pdf

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%

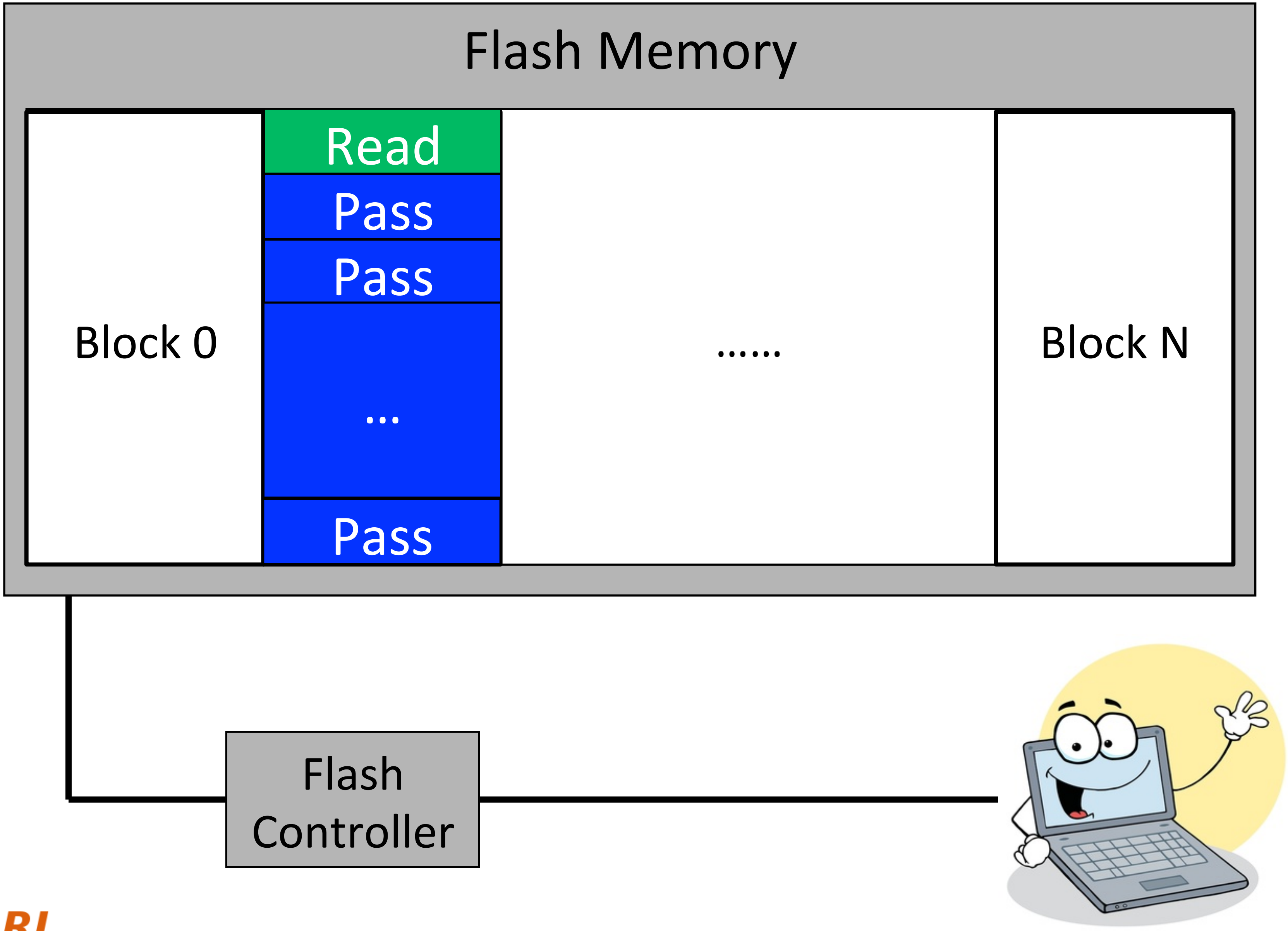
Outline

- Background (Problem and Goal)
- Key Experimental Observations
- Mitigation: V_{pass} Tuning
- Recovery: Read Disturb Oriented Error Recovery
- Conclusion

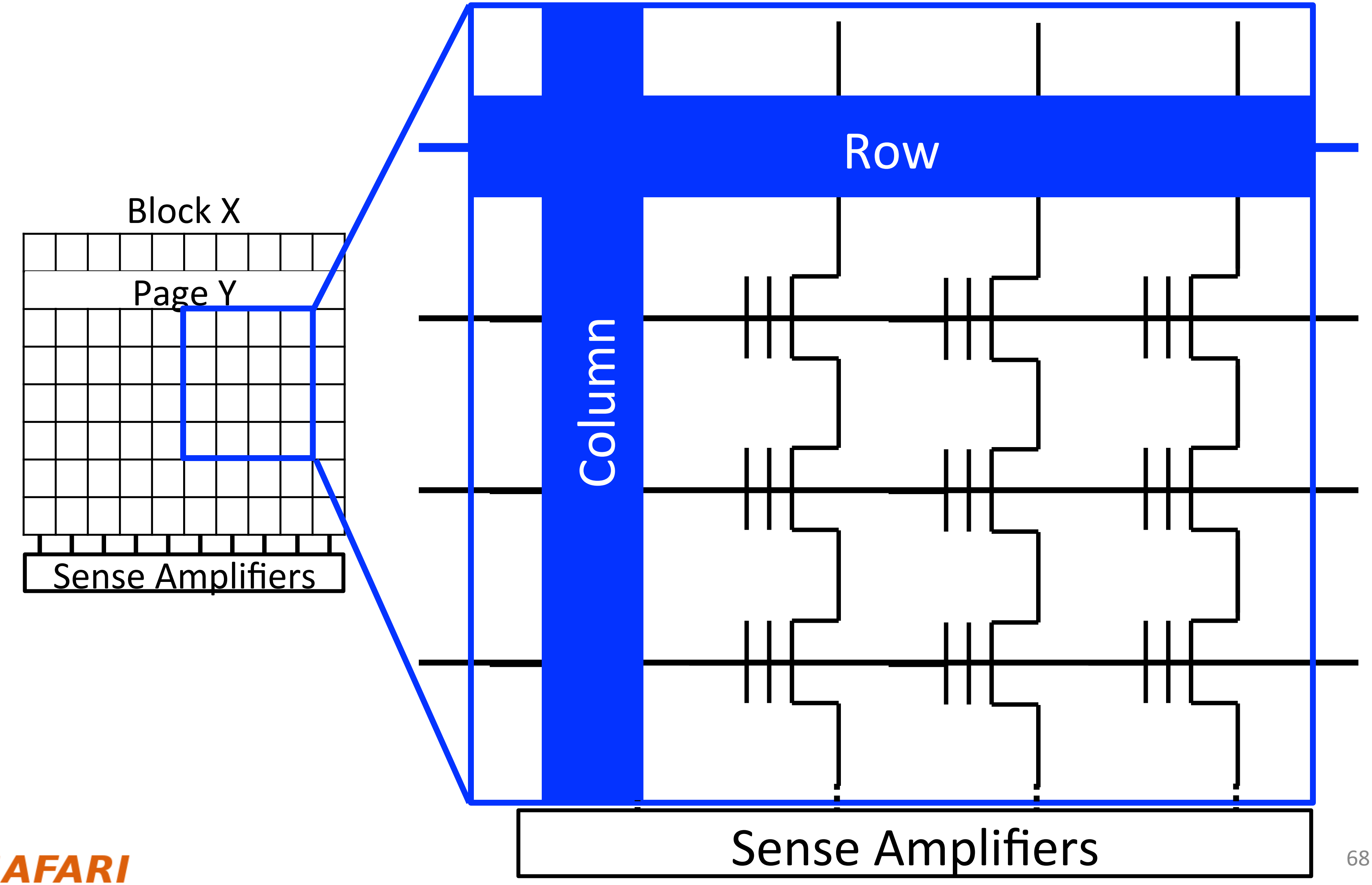
Outline

- Background (Problem and Goal)
- Key Experimental Observations
- Mitigation: V_{pass} Tuning
- Recovery: Read Disturb Oriented Error Recovery
- Conclusion

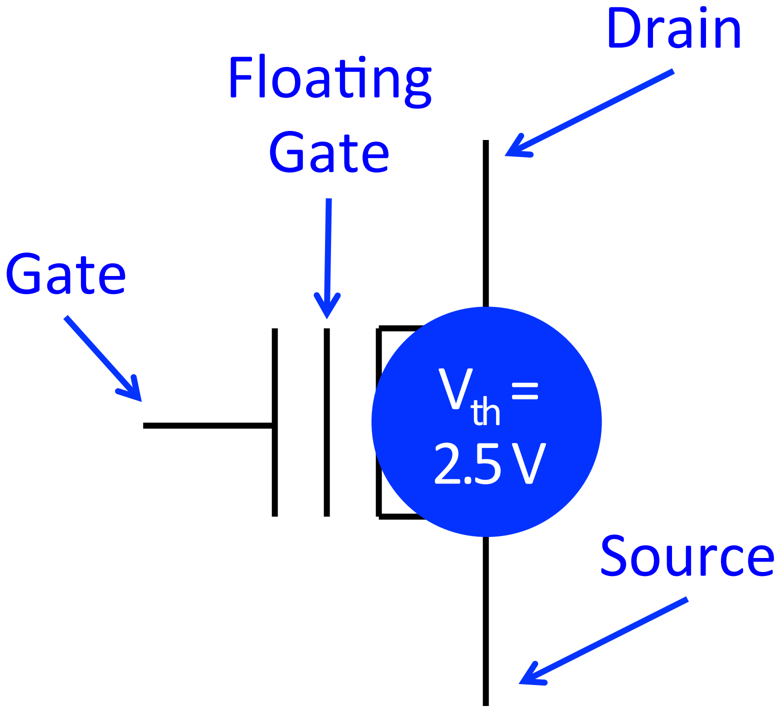
NAND Flash Memory Background



Flash Cell Array

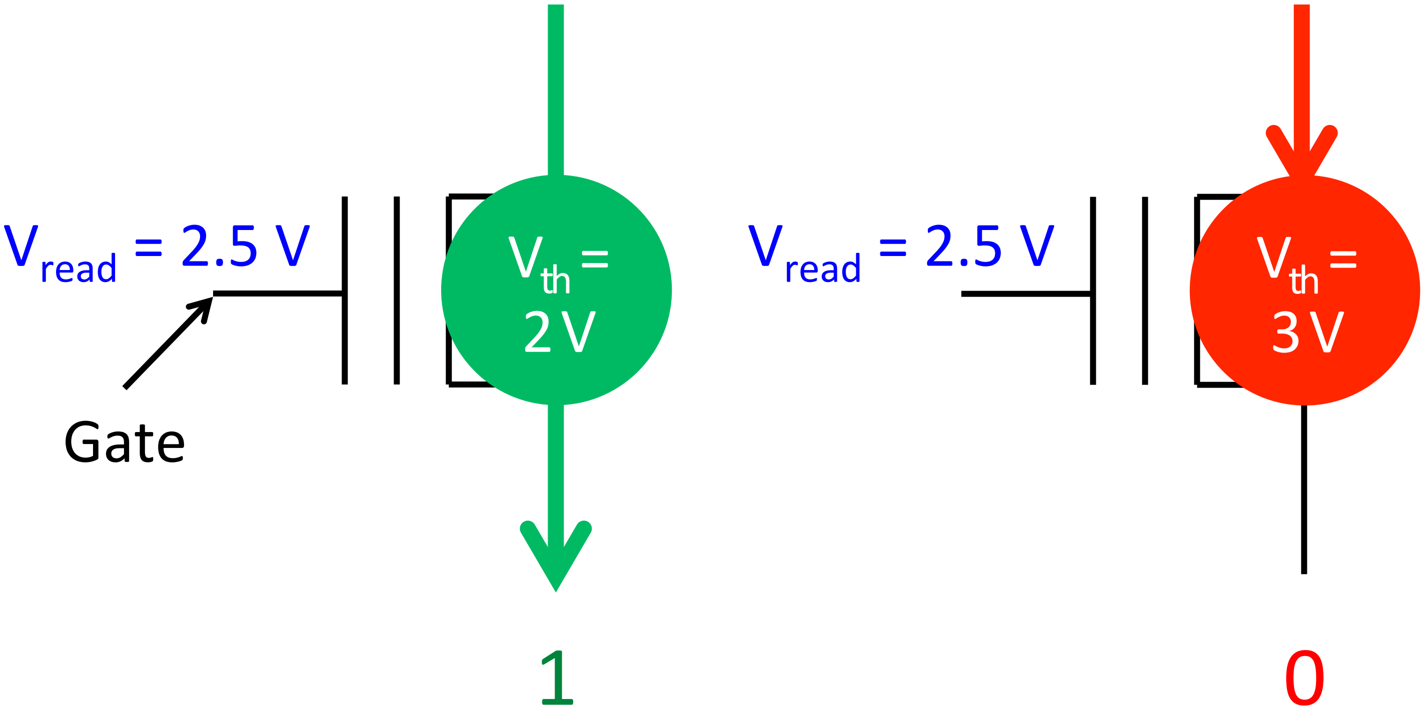


Flash Cell

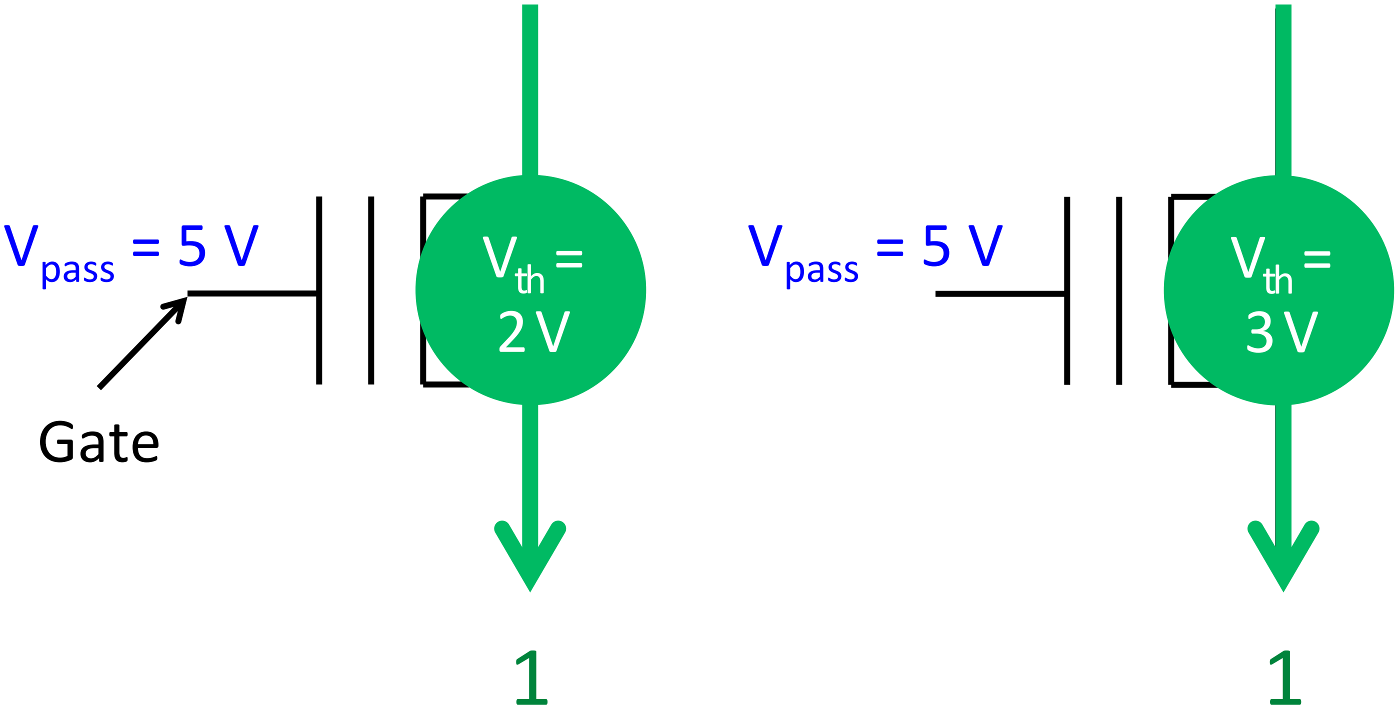


Floating Gate Transistor
(Flash Cell)

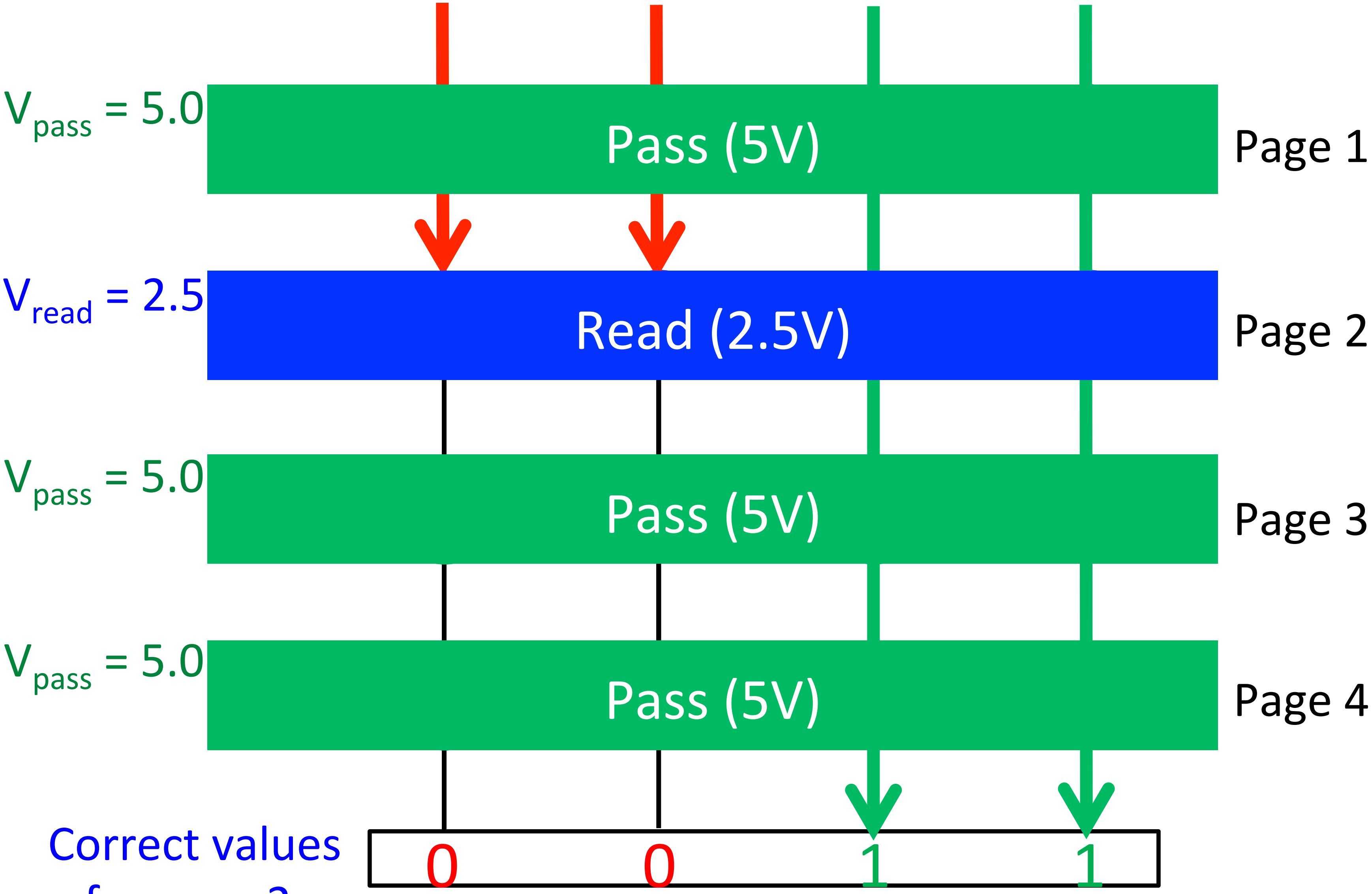
Flash Read



Flash Pass-Through

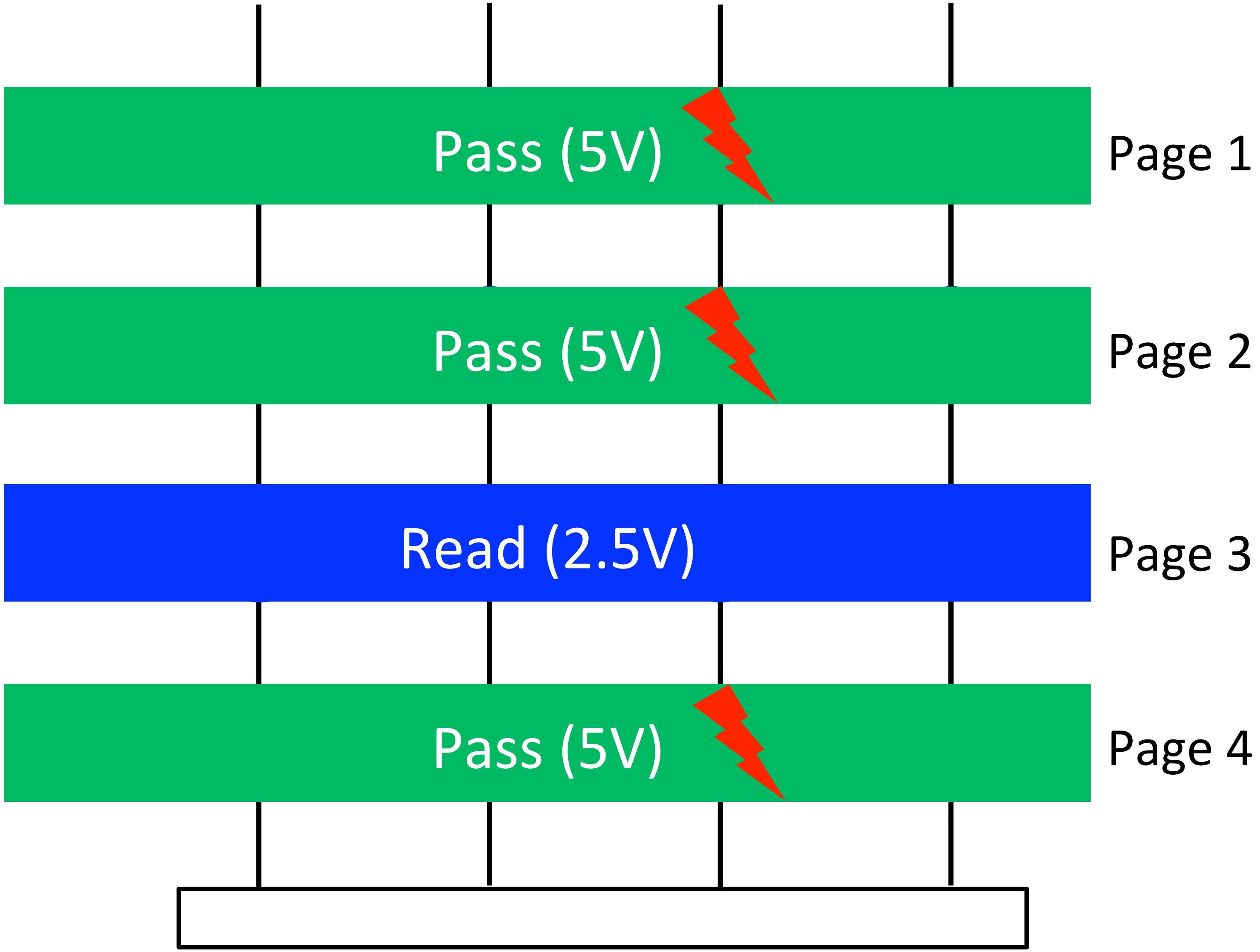


Read from Flash Cell Array



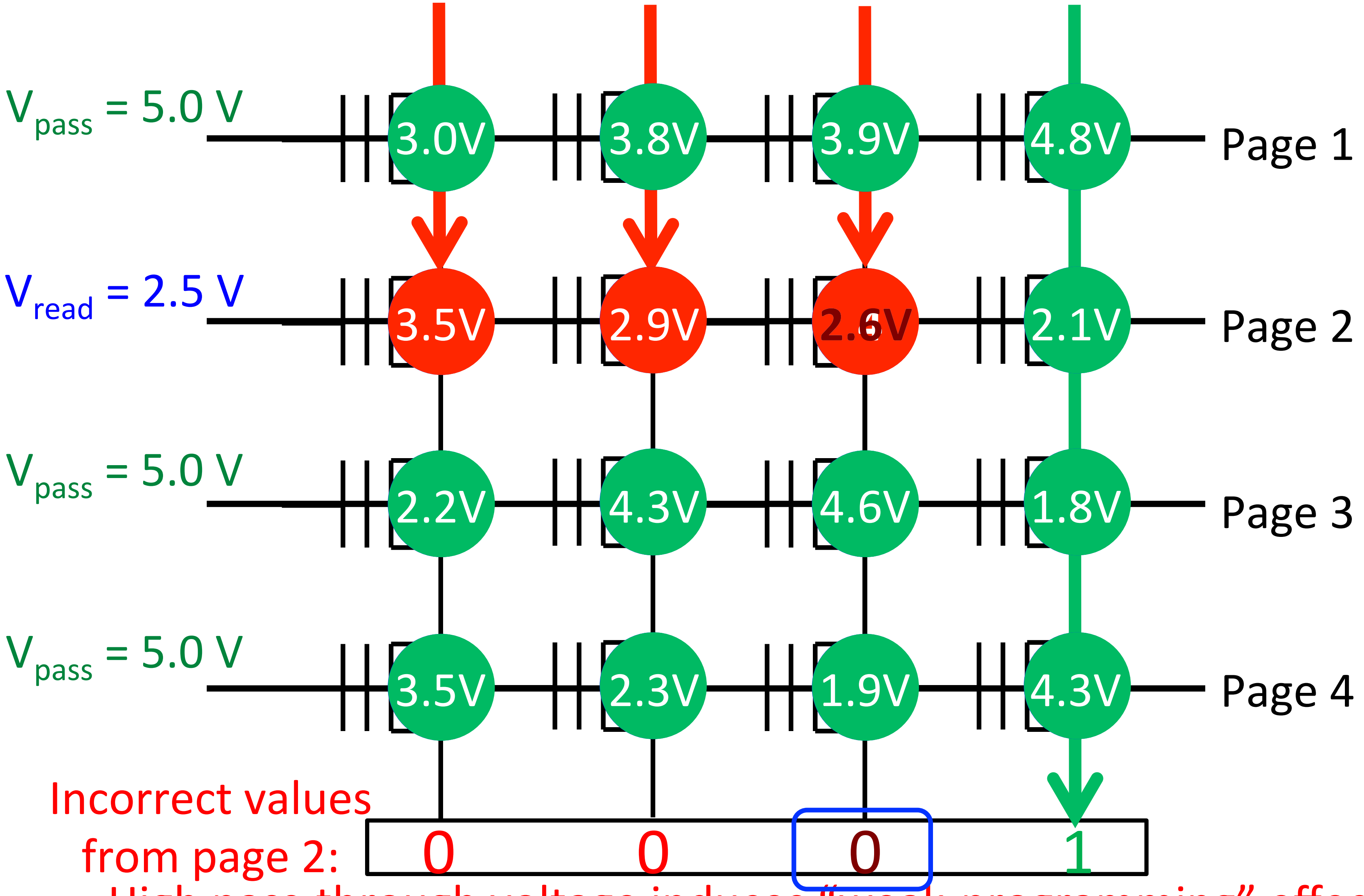
Correct values for page 2:

Read Disturb Problem: “Weak Programming” Effect



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

Read disturb errors: Reading from one page can alter the values stored in other unread pages

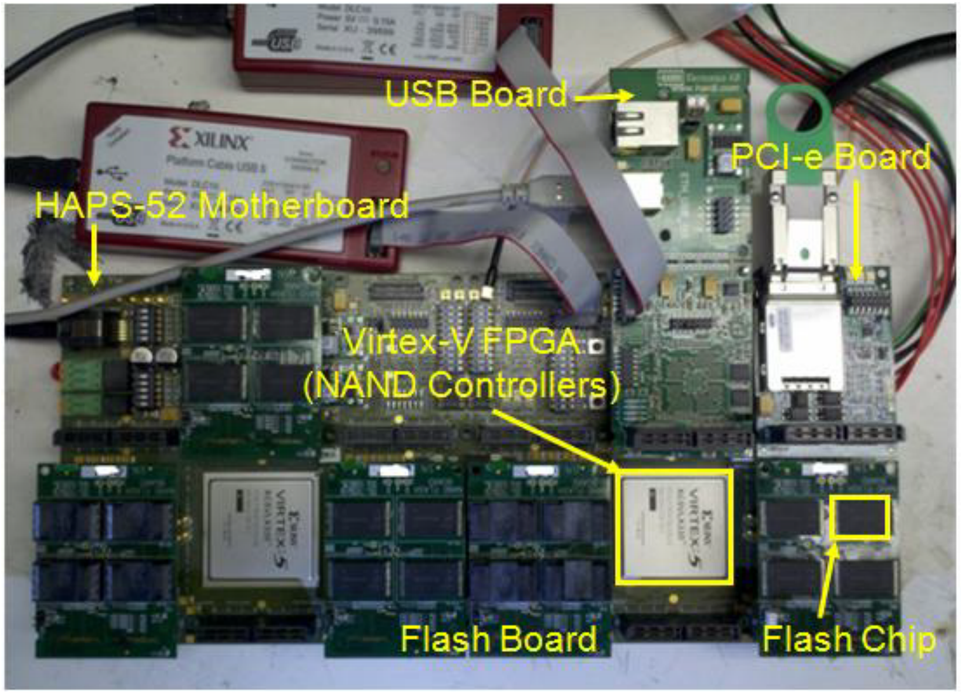
Goal: Mitigate and Recover Read Disturb Errors

Outline

- Background (Problem and Goal)
- Key Experimental Observations
- Mitigation: V_{pass} Tuning
- Recovery: Read Disturb Oriented Error Recovery
- Conclusion

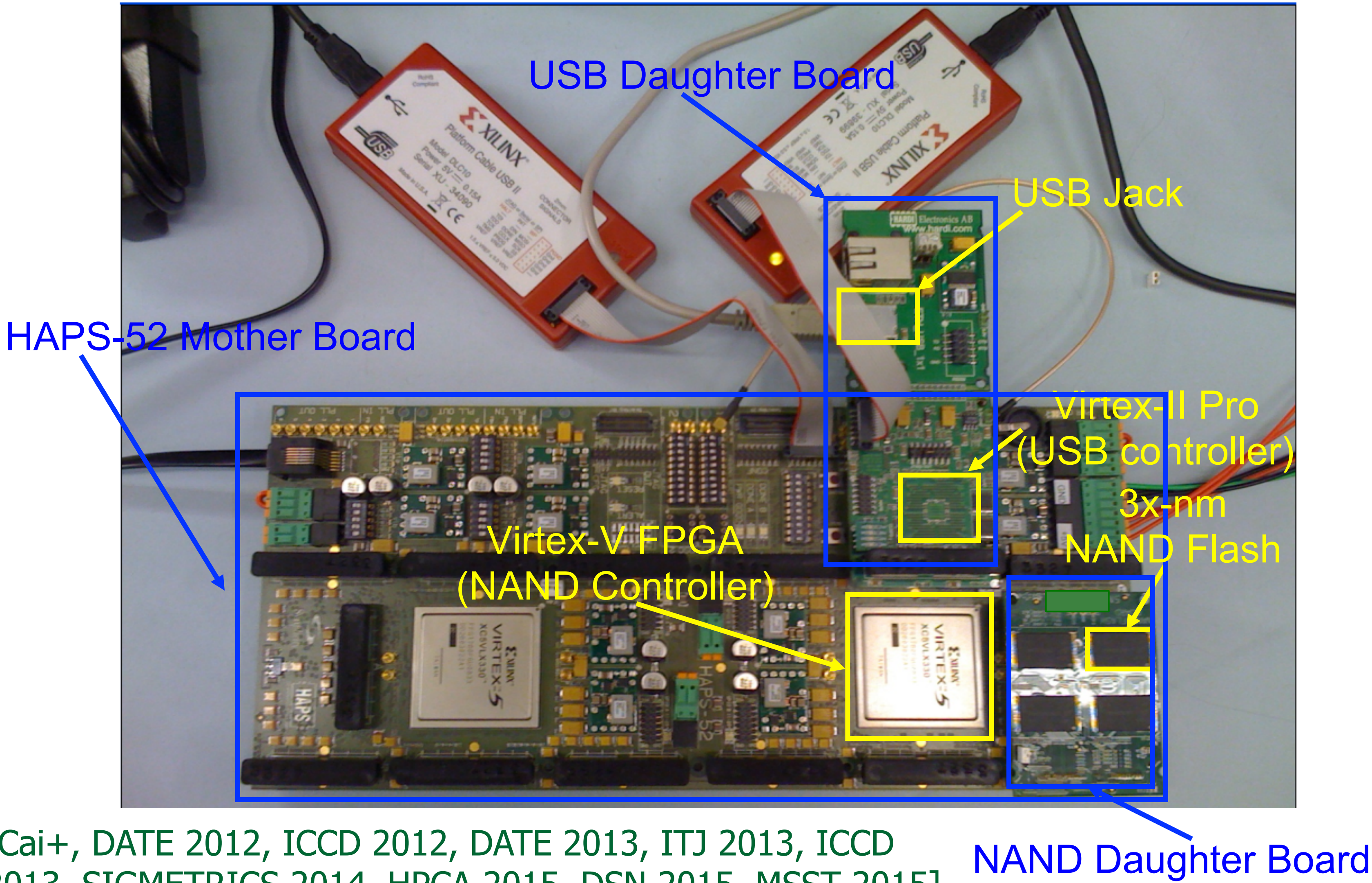
Methodology

- FPGA-based flash memory testing platform [Cai+, FCCM ‘11]



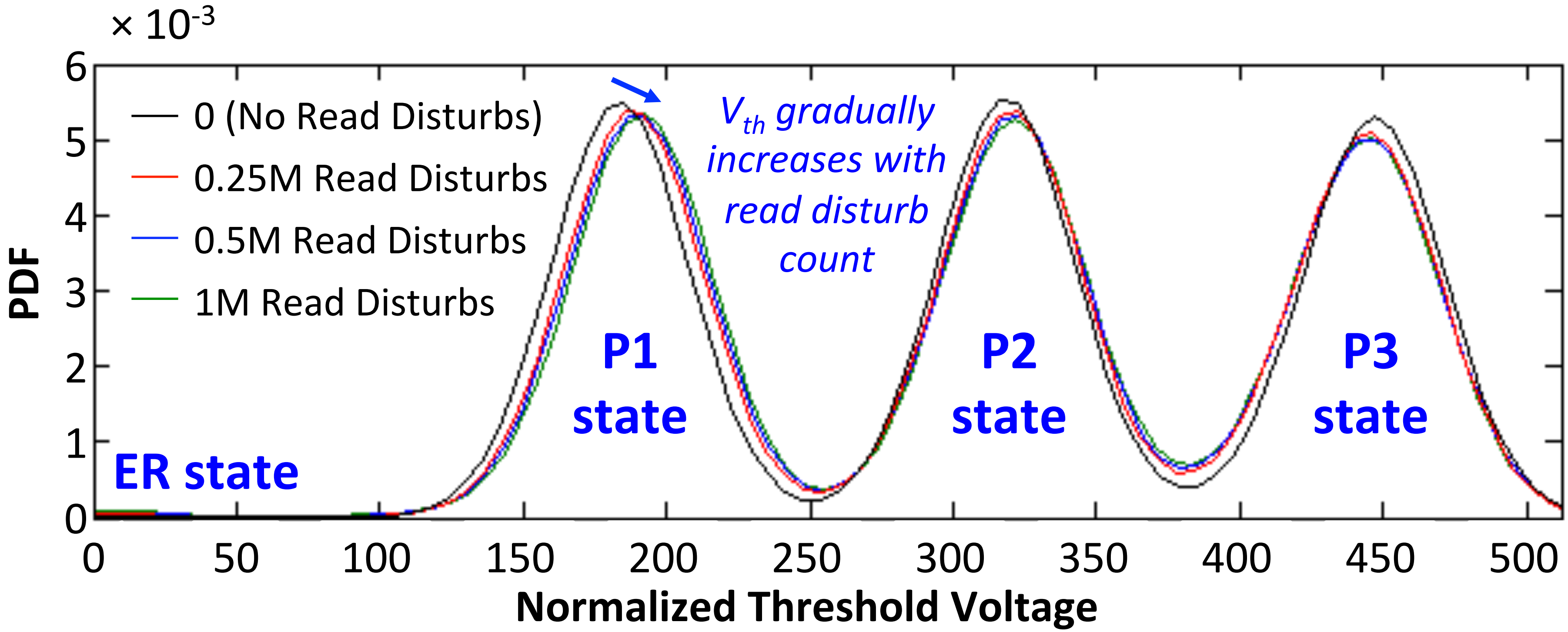
- Real 20- to 24-nm MLC NAND flash chips
- 0 to 1M read disturbs
- 0 to 15K Program/Erase Cycles (PEC)

Experimental Infrastructure



[Cai+, DATE 2012, ICCD 2012, DATE 2013, ITJ 2013, ICCD 2013, SIGMETRICS 2014, HPCA 2015, DSN 2015, MSST 2015]

Read Disturb Effect on V_{th} Distribution



V_{th} gradually increases with read disturb count

Other Experimental Observations

- Lower threshold voltage states are affected more by read disturb
- Wear-out increases read disturb effect

Key Observation 1: Slightly lowering V_{pass} greatly reduces read disturb errors

1400

1300

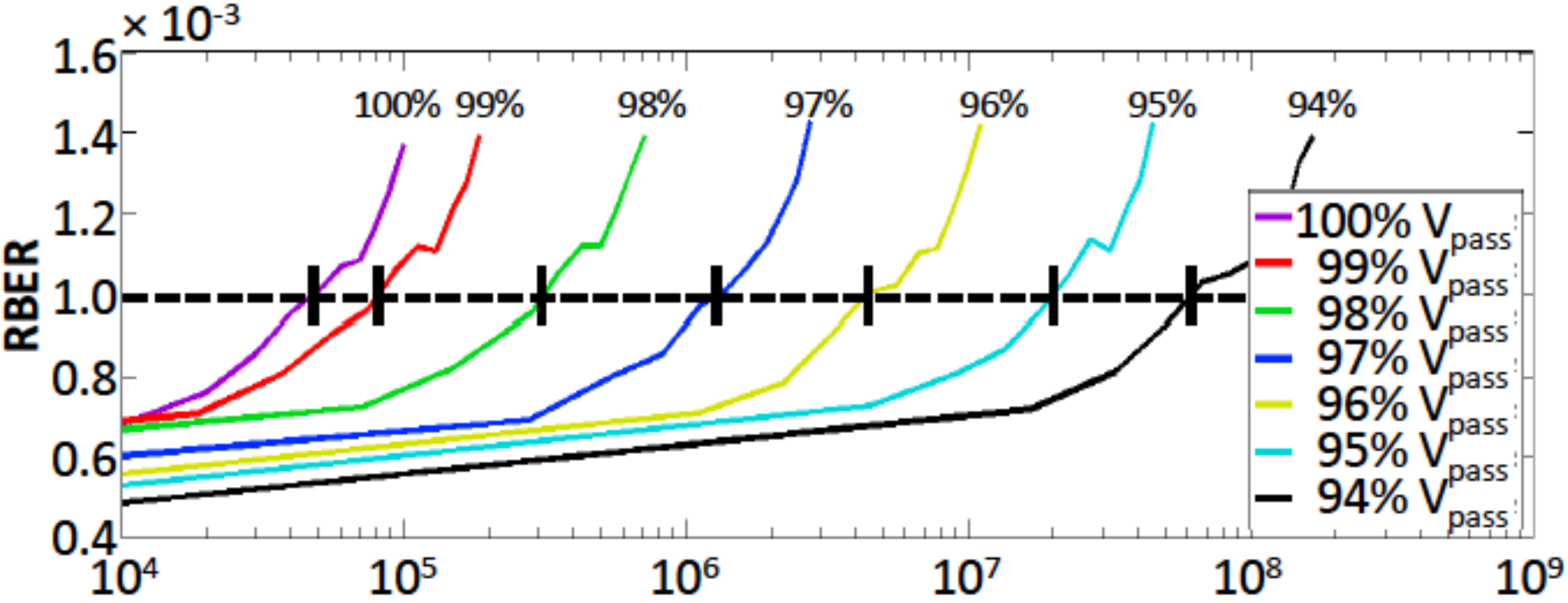


Fig. 11. Raw bit error rate vs. read disturb count for different V_{pass} values, for flash memory under 8K P/E cycles of wear.

Percentage of Vpass Reduction

Outline

- Background (Problem and Goal)
- Key Experimental Observations
- Mitigation: V_{pass} Tuning
- Recovery: Read Disturb Oriented Error Recovery
- Conclusion

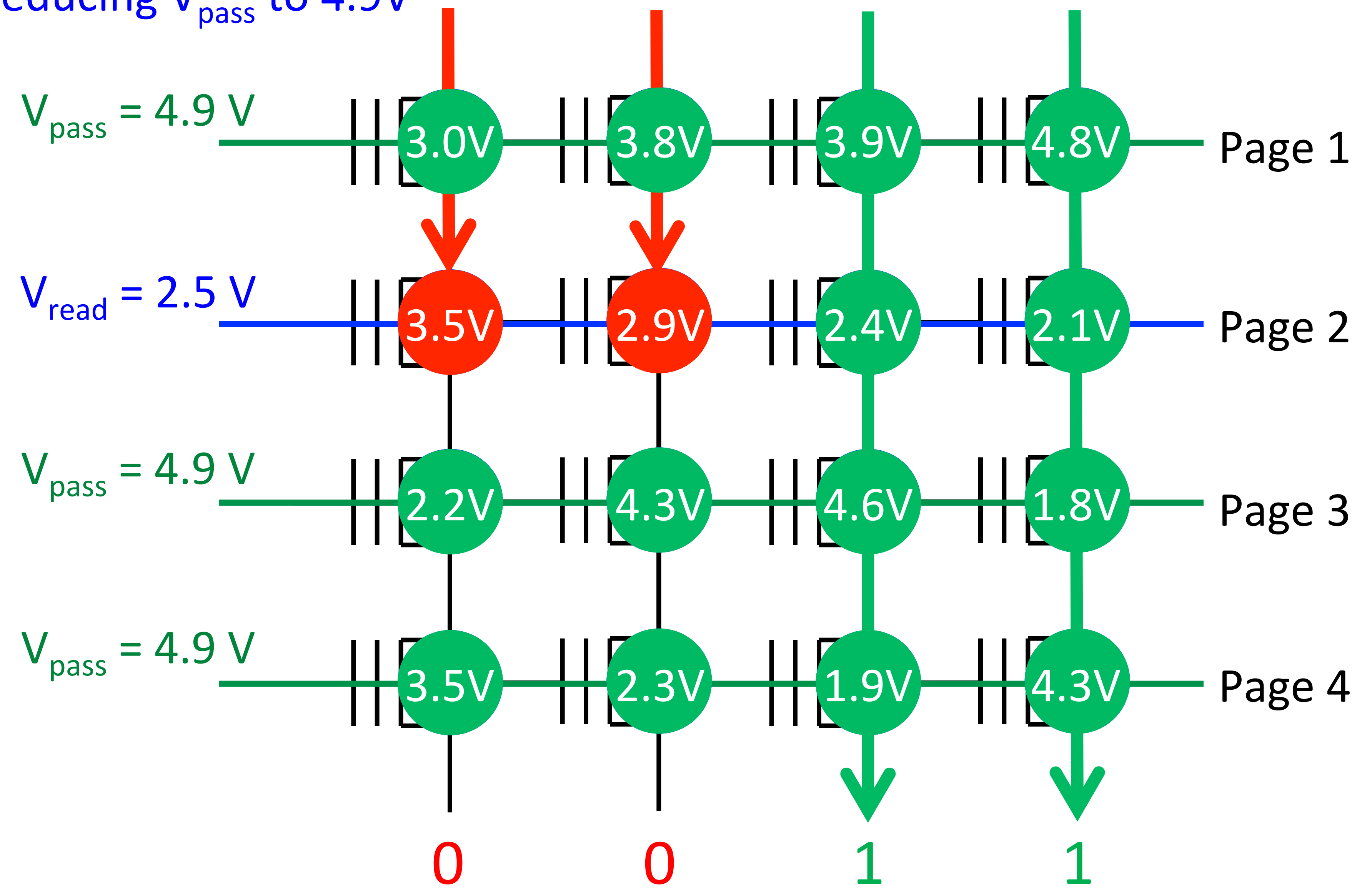
Read Disturb Mitigation: V_{pass} Tuning

- Key Idea: Dynamically find and apply a lowered V_{pass}

- Trade-off for lowering V_{pass}
  + Allows more read disturbs
  – Induces more read errors

Read Errors Induced by V_{pass} Reduction

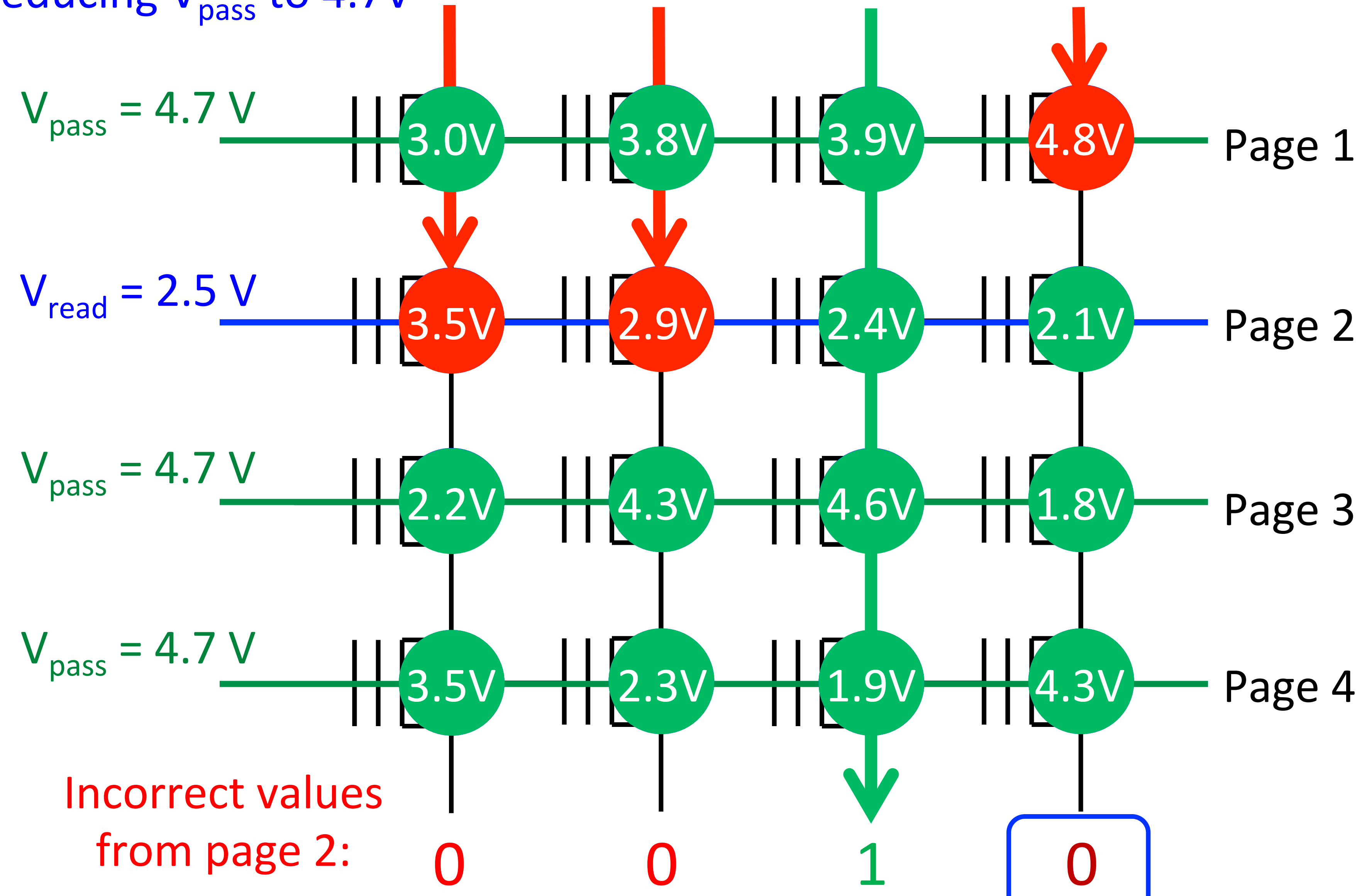
Reducing V_{pass} to 4.9V



SAFARI

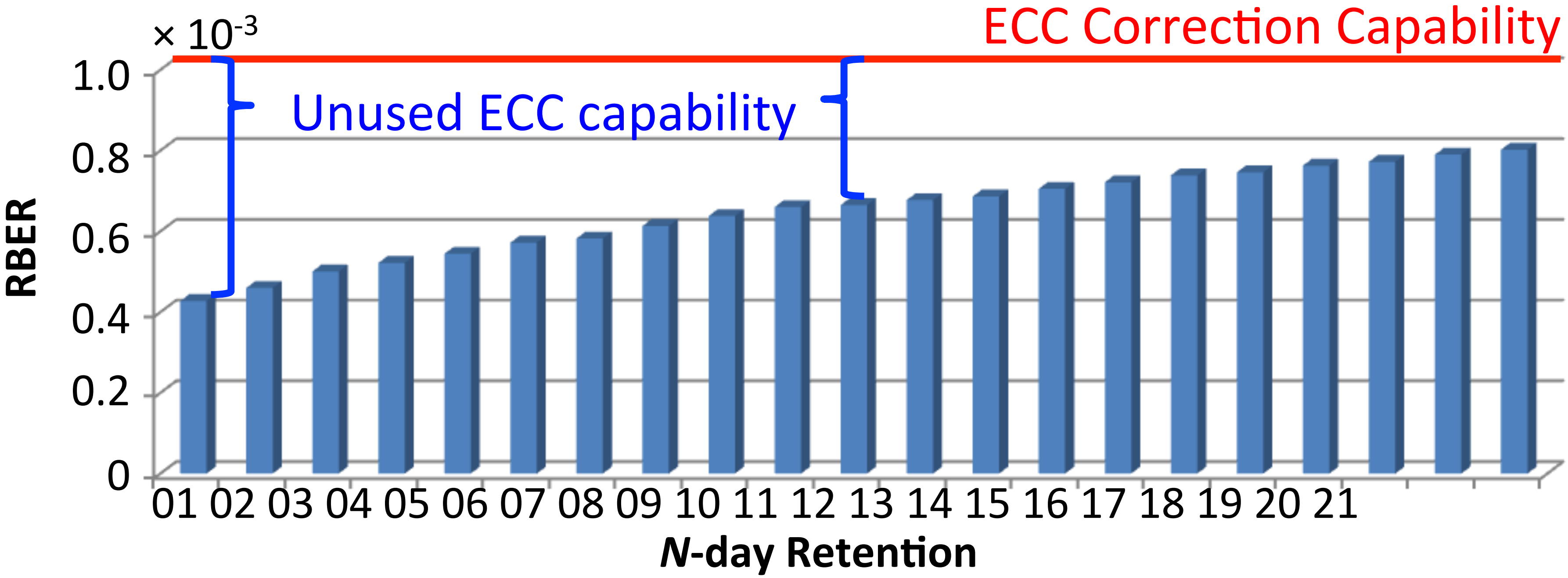
Read Errors Induced by V_{pass} Reduction

Reducing V_{pass} to 4.7V



Incorrect values from page 2:

Utilizing the Unused ECC Capability



1. ECC provisioned for high retention "age"
2. Unused ECC capability can be used to fix read errors
3. Unused ECC capability decreases over retention age

Dynamically adjust V_{pass} so that read errors fully utilize the unused ECC capability

V_{pass} Reduction Trade-Off Summary

- Today: Conservatively set V_{pass} to a high voltage
  – Accumulates more read disturb errors at the end of each refresh interval
  + No read errors

- Idea: Dynamically adjust V_{pass} to unused ECC capability
  + Minimize read disturb errors
  o Control read errors to be tolerable by ECC
  o If read errors exceed ECC capability, read again with a higher V_{pass} to correct read errors

V_{pass} Tuning Steps

- Perform once for each block every day:

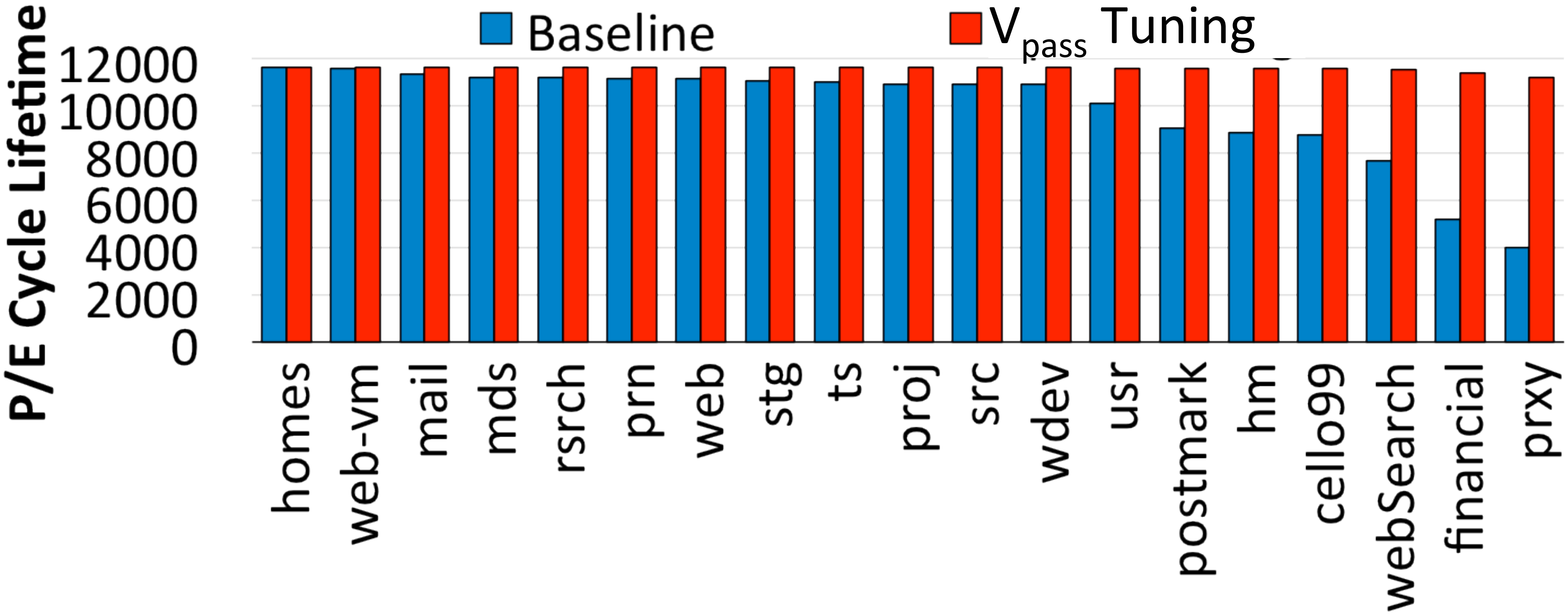
1. Estimate *unused ECC capability (using retention age)*
2. Aggressively reduce V_{pass} until *read errors exceed ECC capability*
3. Gradually increase V_{pass} until read errors become just less than ECC capability

Evaluation of V_{pass} Tuning

- 19 real workload I/O traces
- Assume 7-day refresh period
- Similar methodology as before to determine acceptable V_{pass} reduction

- Overhead for a 512 GB flash drive:
  - 128 KB storage overhead for per-block V_{pass} setting and worst-case page
  - 24.34 sec/day average V_{pass} Tuning overhead

V_{pass} Tuning Lifetime Improvements



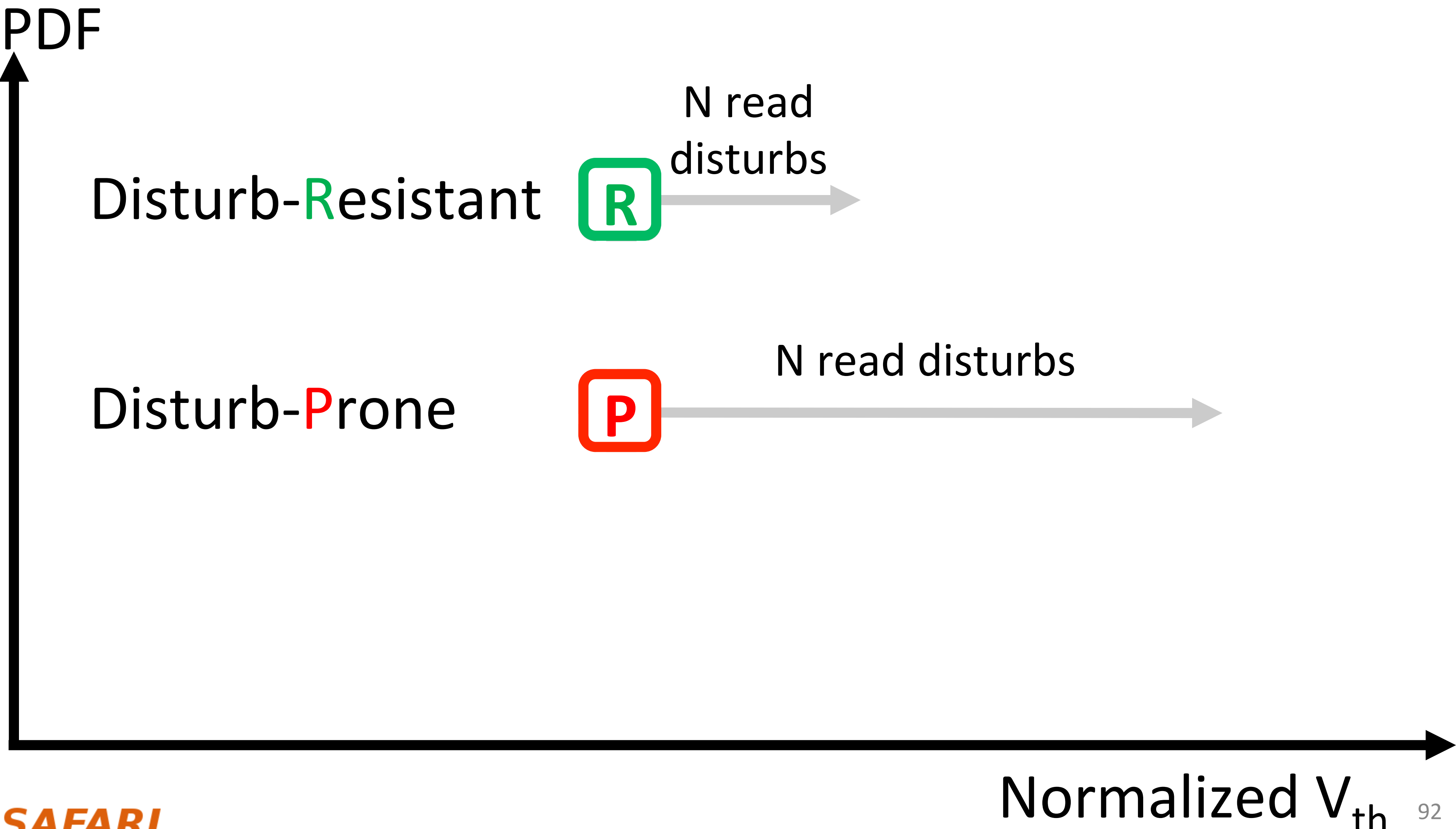
Average lifetime improvement: 21.0%

SAFARI

Outline

- Background (Problem and Goal)
- Key Experimental Observations
- Mitigation: V_{pass} Tuning
- Recovery: Read Disturb Oriented Error Recovery
- Conclusion

Read Disturb Resistance

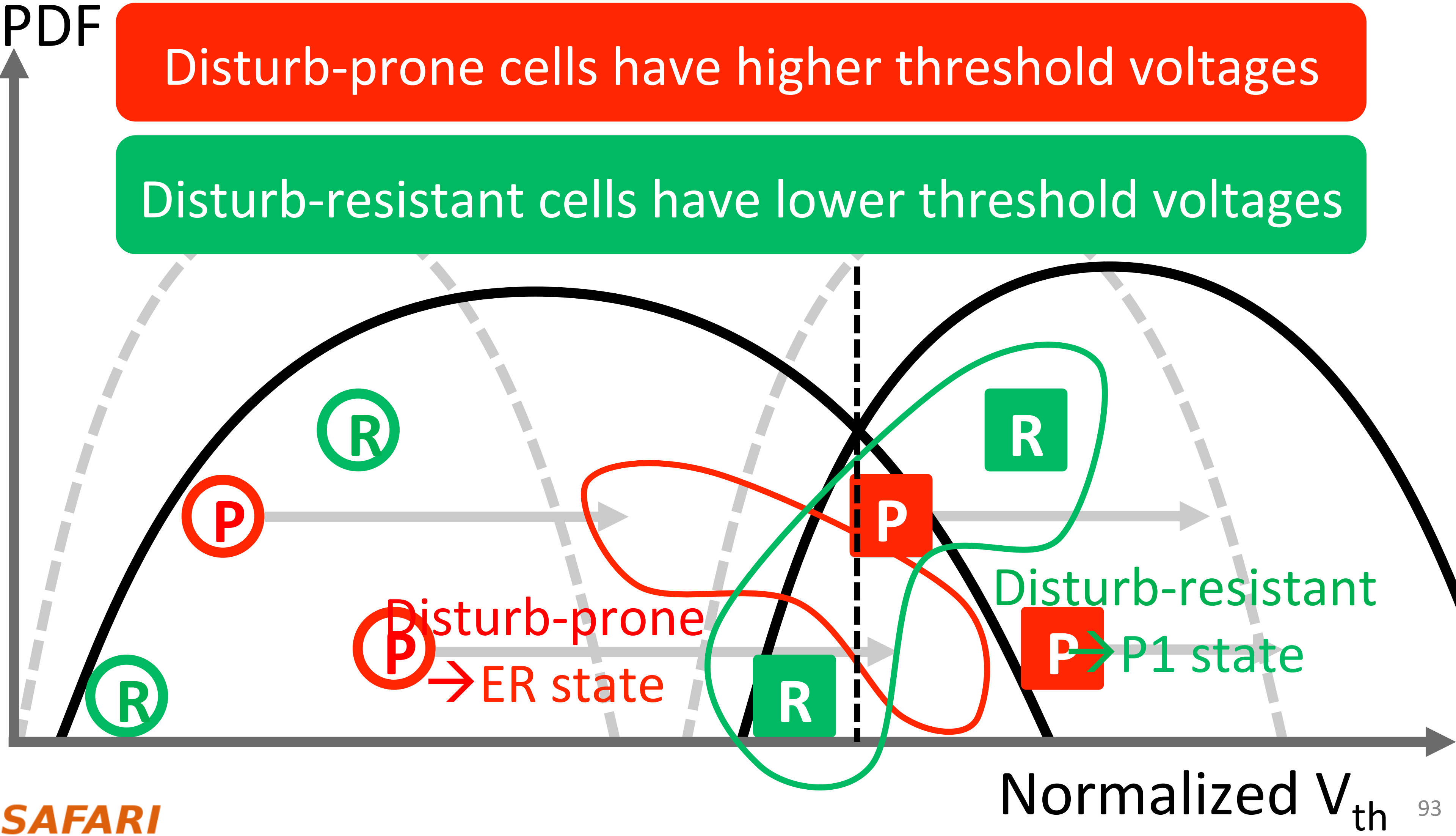


Observation 2: Some Flash Cells Are More Prone to Read Disturb

After 250K read disturb:

Disturb-prone cells have higher threshold voltages

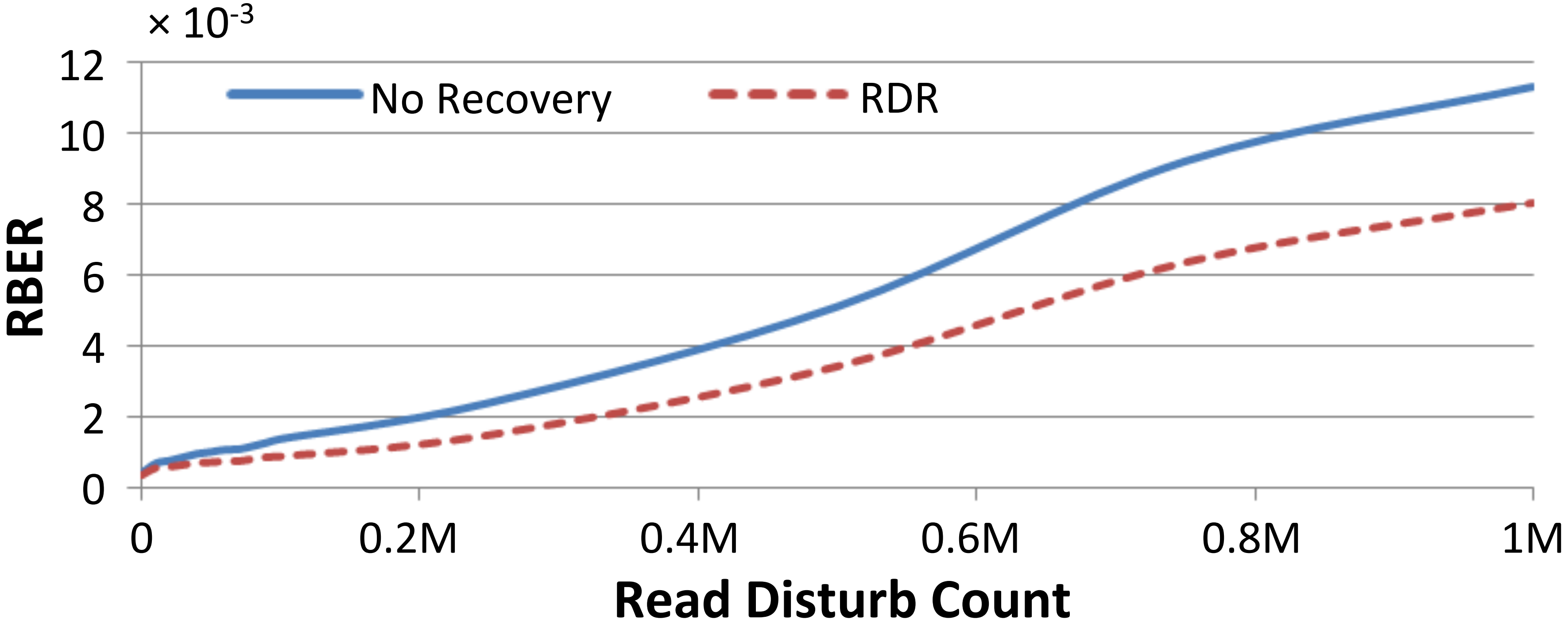
Disturb-resistant cells have lower threshold voltages



Read Disturb Oriented Error Recovery (RDR)

- Triggered by an uncorrectable flash error
  - Back up all valid data in the faulty block
  - Disturb the faulty page 100K times (more)
  - Compare V_{th}'s before and after read disturb
  - Select cells susceptible to flash errors ($V_{ref}-\sigma<V_{th}<V_{ref}-\sigma$)
  - Predict among these susceptible cells
    - Cells with more V_{th} shifts are disturb-prone → Lower V_{th} state
    - Cells with less V_{th} shifts are disturb-resistant → Higher V_{th} state

RDR Evaluation



Reduces total error counts by up to 36% @ 1M read disturbs
ECC can be used to correct the remaining errors

Outline

- Background (Problem and Goal)
- Key Experimental Observations
- Mitigation: V_{pass} Tuning
- Recovery: Read Disturb Oriented Error Recovery
- Conclusion

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, {yixinluo, ghose, kenmai, onur}@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

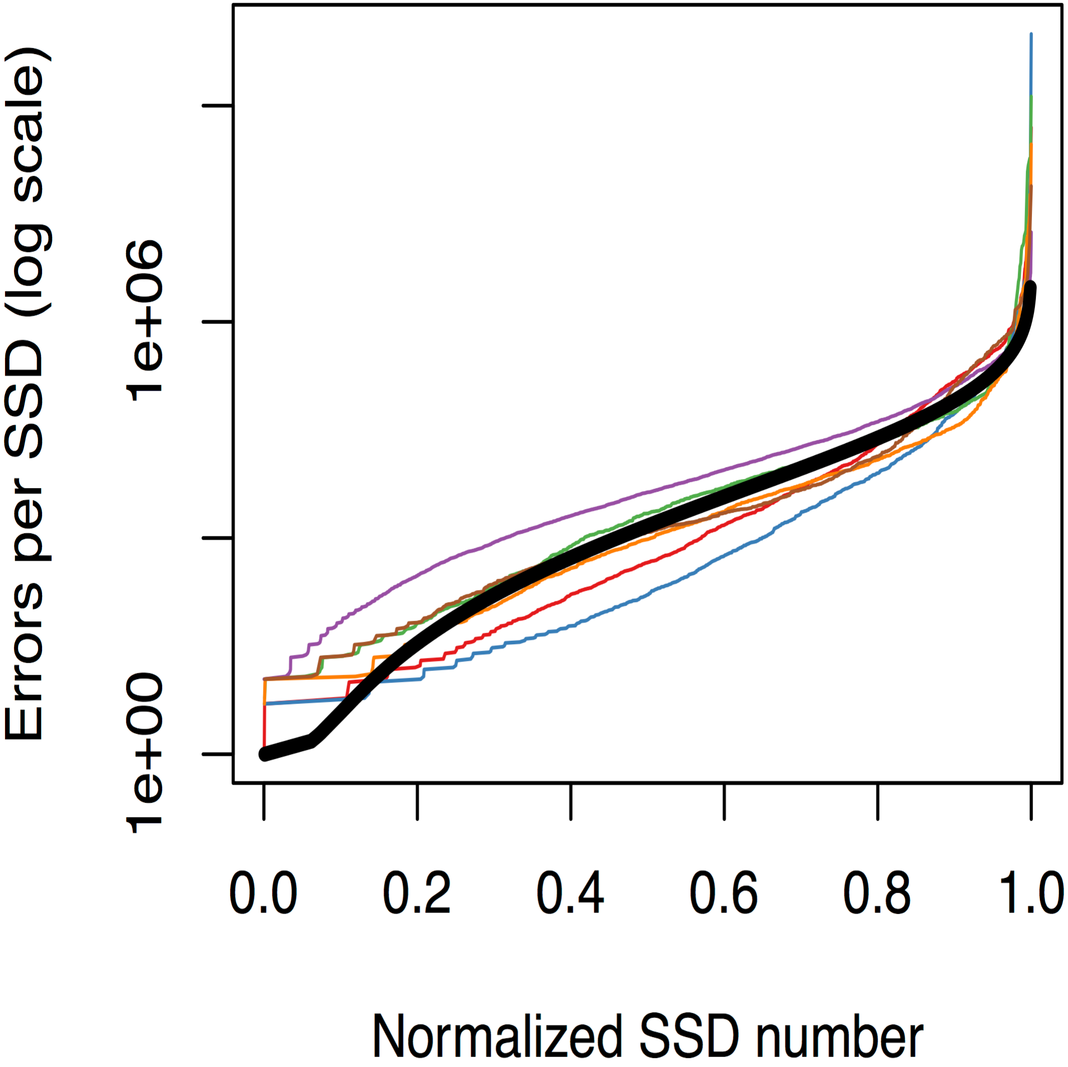
Justin Meza
Carnegie Mellon University
meza@cmu.edu

Qiang Wu
Facebook, Inc.
qwu@fb.com

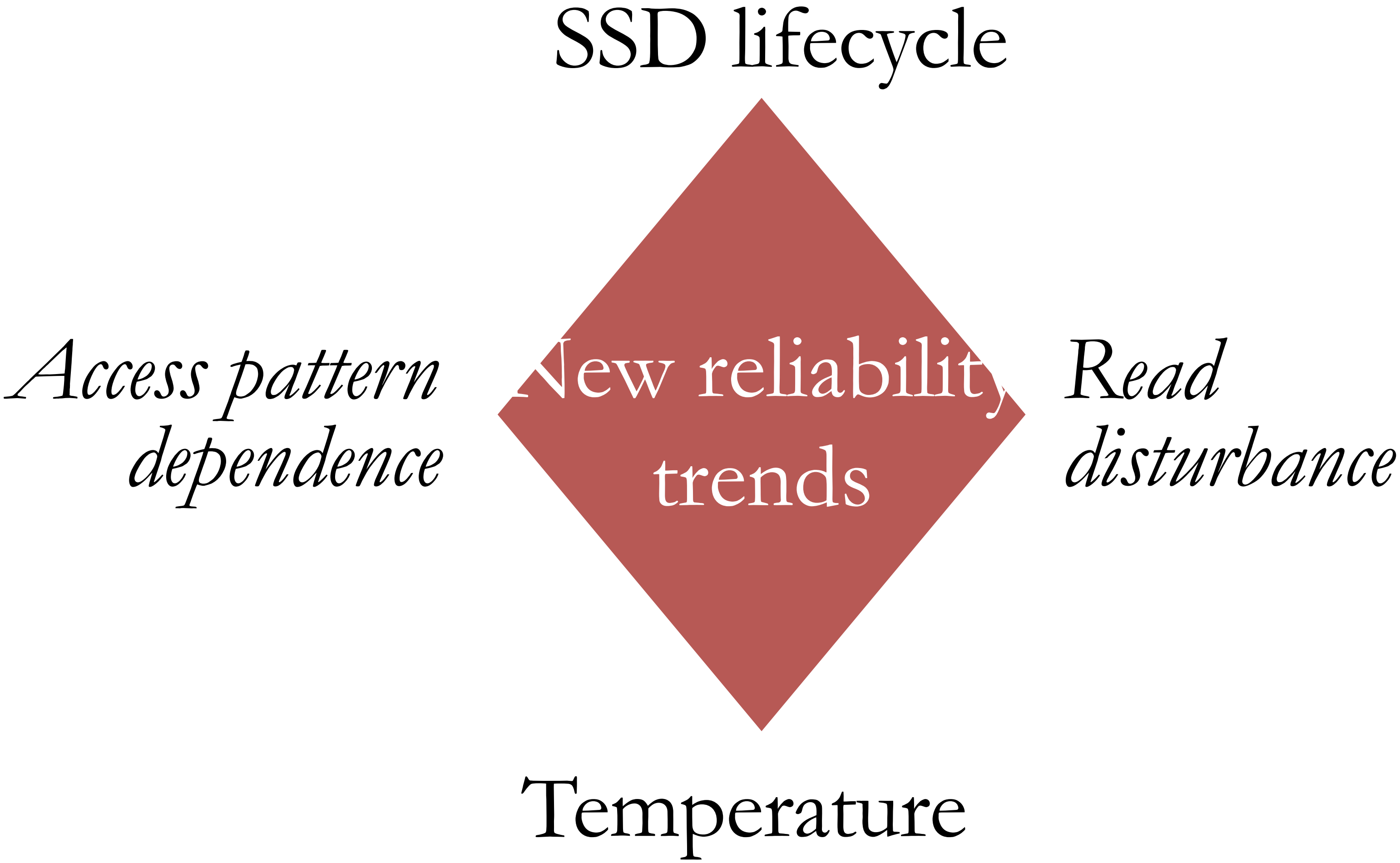
Sanjeev Kumar
Facebook, Inc.
skumar@fb.com

Onur Mutlu
Carnegie Mellon University
onur@cmu.edu

A few SSDs cause most errors



Summary



SSD lifecycle

Access pattern dependence

New reliability trends

Read disturbance

Temperature

Agenda

- DRAM Scaling Issues
  - DRAM RowHammer Problem
  - Some Other DRAM Reliability Studies

- NAND Flash Scaling Issues
  - Some NAND Flash Reliability Studies
  - Read Disturb Errors in NAND Flash Memory

- Summary and Discussion

Summary

- DRAM and Flash Scaling Challenges are real and critical
  - They lead to many reliability (and security) challenges

- We need to understand various reliability issues with both
  - Small-scale experimental studies (FPGA-based testing platforms)
  - Large-scale experimental studies (data centers and clusters)

- We need to innovate at all levels
  - DRAM and Flash architecture and controllers
  - Hardware, software, devices

- There are many problems to solve
  - Industry-academia cooperation is much needed and welcome

Reliability (and Security) Issues of DRAM and NAND Flash Scaling

Onur Mutlu

omutlu@gmail.com

http://users.ece.cmu.edu/~omutlu/

HPCA Memory Reliability Workshop

March 13, 2016

Carnegie Mellon

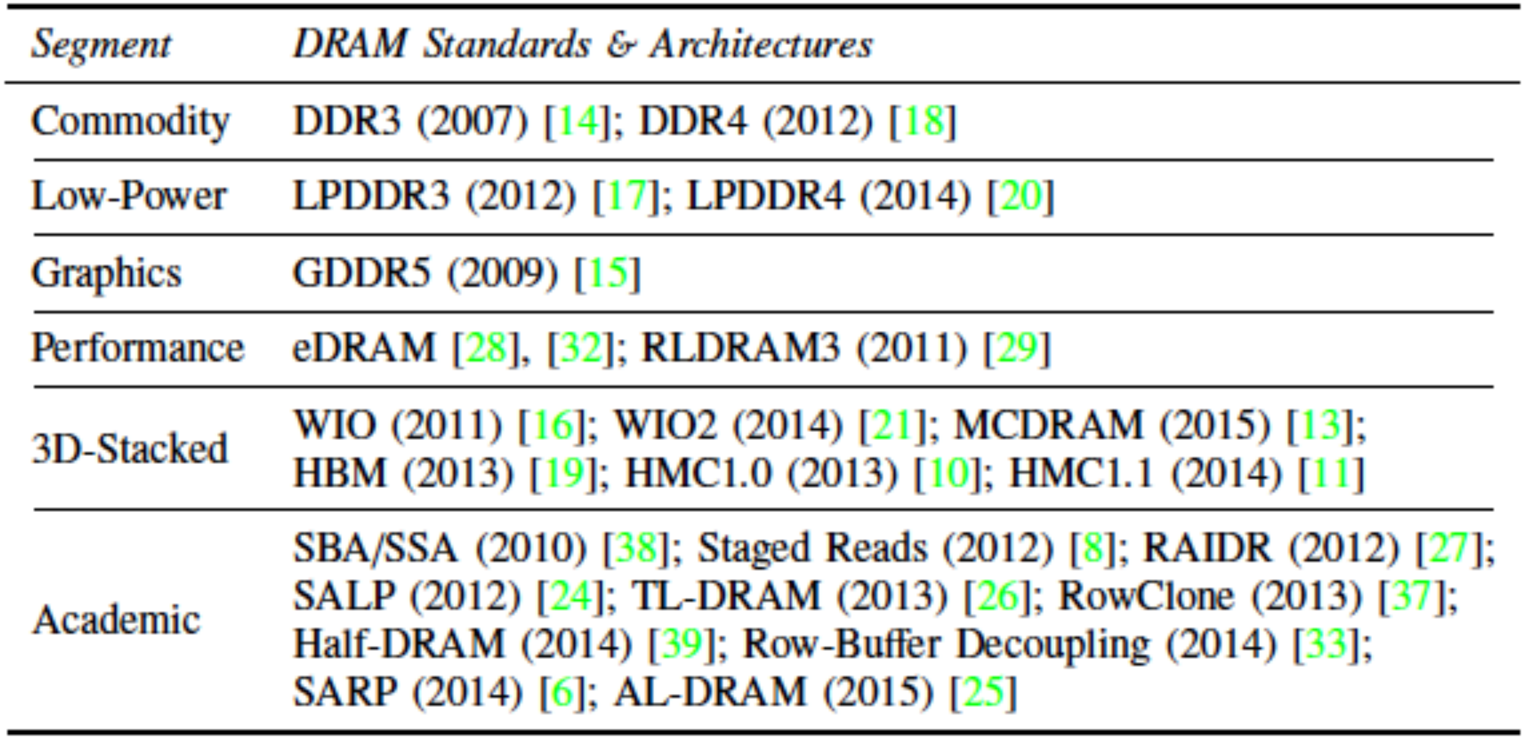
SAFARI

Ramulator: A Fast and Extensible DRAM Simulator

[IEEE Comp Arch Letters'15]

Ramulator Motivation

- DRAM and Memory Controller landscape is changing
- Many new and upcoming standards
- Many new controller designs
- A fast and easy-to-extend simulator is very much needed

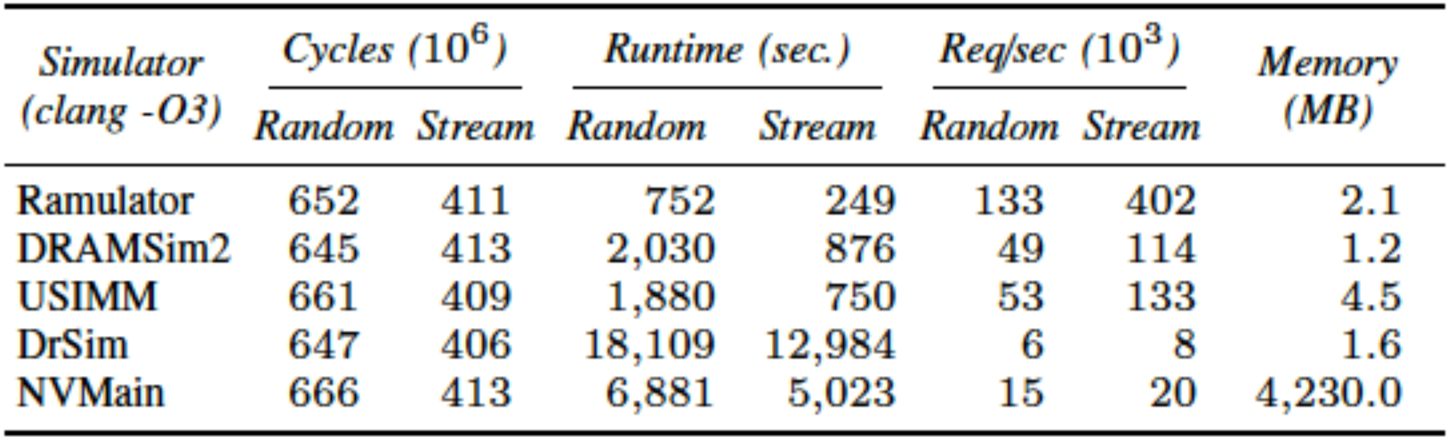


Segment	*DRAM Standards & Architectures*
Commodity	DDR3 (2007) [14]; DDR4 (2012) [18]
Low-Power	LPDDR3 (2012) [17]; LPDDR4 (2014) [20]
Graphics	GDDR5 (2009) [15]
Performance	eDRAM [28], [32]; RLDRAM3 (2011) [29]
3D-Stacked	WIO (2011) [16]; WIO2 (2014) [21]; MCDRAM (2015) [13]; HBM (2013) [19]; HMC1.0 (2013) [10]; HMC1.1 (2014) [11]
Academic	SBA/SSA (2010) [38]; Staged Reads (2012) [8]; RAIDR (2012) [27]; SALP (2012) [24]; TL-DRAM (2013) [26]; RowClone (2013) [37]; Half-DRAM (2014) [39]; Row-Buffer Decoupling (2014) [33]; SARP (2014) [6]; AL-DRAM (2015) [25]

Table 1. Landscape of DRAM-based memory

Ramulator

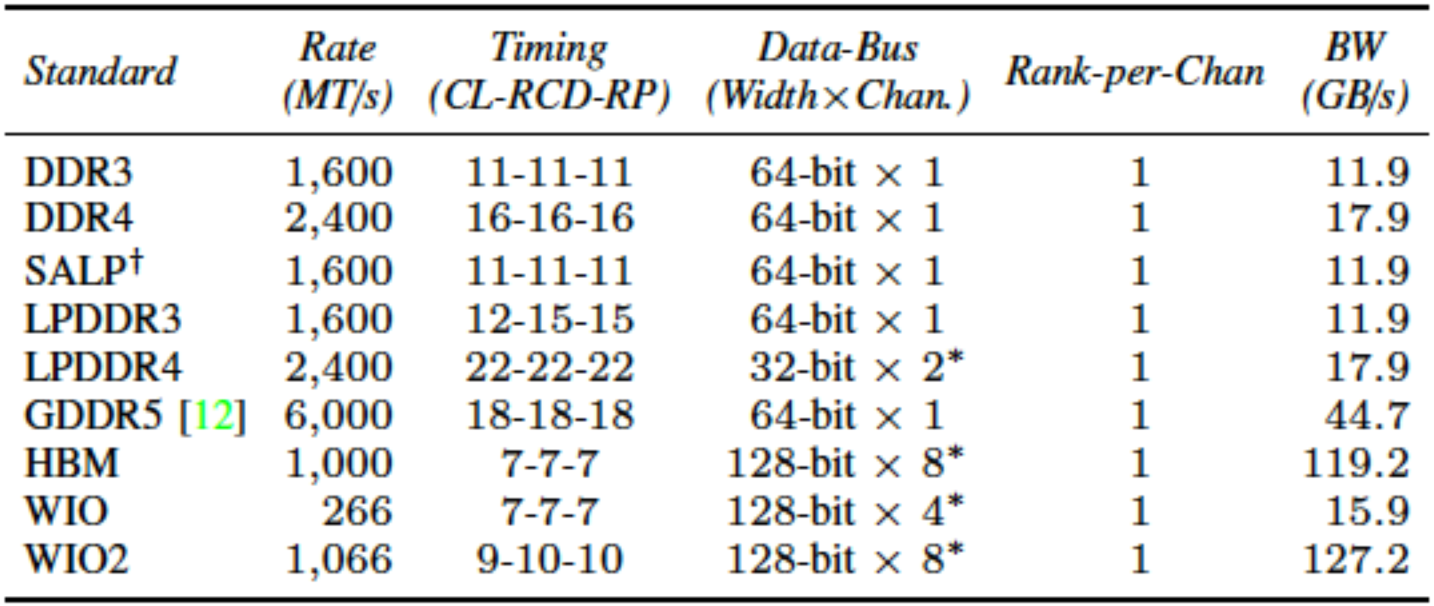
- Provides out-of-the box support for many DRAM standards:
  - DDR3/4, LPDDR3/4, GDDR5, WIO1/2, HBM, plus new proposals (SALP, AL-DRAM, TLDRAM, RowClone, and SARP)
- ~2.5X faster than fastest open-source simulator
- Modular and extensible to different standards



Simulator (clang -O3)	*Cycles (10^6)*		*Runtime (sec.)*		*Req/sec (10^3)*		*Memory (MB)*
	Random	*Stream*	*Random*	*Stream*	*Random*	*Stream*	
Ramulator	652	411	752	249	133	402	2.1
DRAMSim2	645	413	2,030	876	49	114	1.2
USIMM	661	409	1,880	750	53	133	4.5
DrSim	647	406	18,109	12,984	6	8	1.6
NVMain	666	413	6,881	5,023	15	20	4,230.0

Table 3. Comparison of five simulators using two traces

Case Study: Comparison of DRAM Standards



Standard	*Rate (MT/s)*	*Timing (CL-RCD-RP)*	*Data-Bus (Width×Chan.)*	*Rank-per-Chan*	*BW (GB/s)*
DDR3	1,600	11-11-11	64-bit × 1	1	11.9
DDR4	2,400	16-16-16	64-bit × 1	1	17.9
SALP†	1,600	11-11-11	64-bit × 1	1	11.9
LPDDR3	1,600	12-15-15	64-bit × 1	1	11.9
LPDDR4	2,400	22-22-22	32-bit × 2*	1	17.9
GDDR5 [12]	6,000	18-18-18	64-bit × 1	1	44.7
HBM	1,000	7-7-7	128-bit × 8*	1	119.2
WIO	266	7-7-7	128-bit × 4*	1	15.9
WIO2	1,066	9-10-10	128-bit × 8*	1	127.2

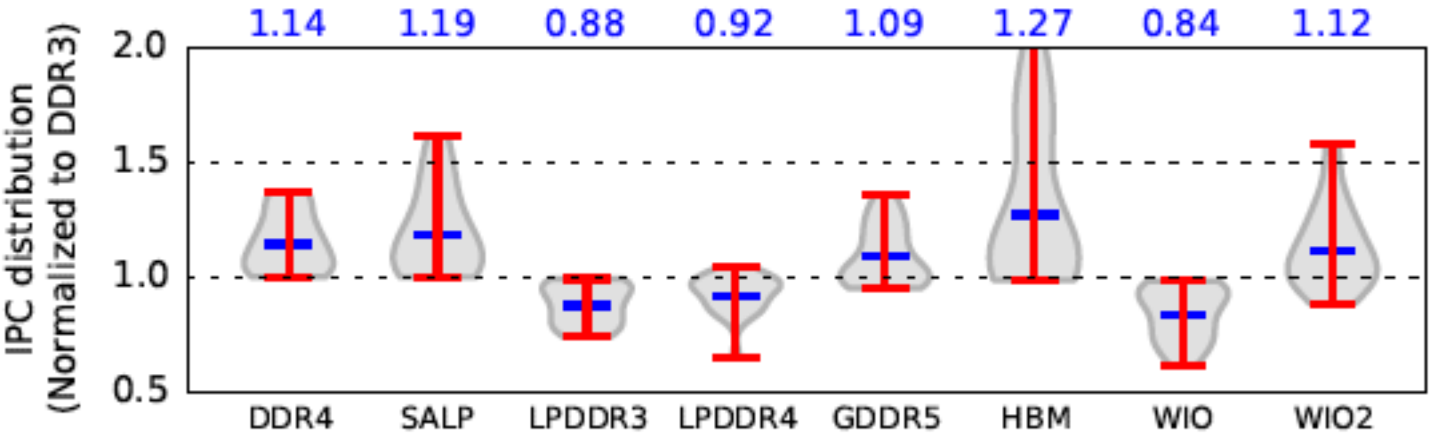


Figure 2. Performance comparison of DRAM standards

Across 22 workloads, simple CPU model

Ramulator Paper and Source Code

- Yoongu Kim, Weikun Yang, and Onur Mutlu,
  "Ramulator: A Fast and Extensible DRAM Simulator"
  IEEE Computer Architecture Letters (***CAL***), March 2015.
  [Source Code]

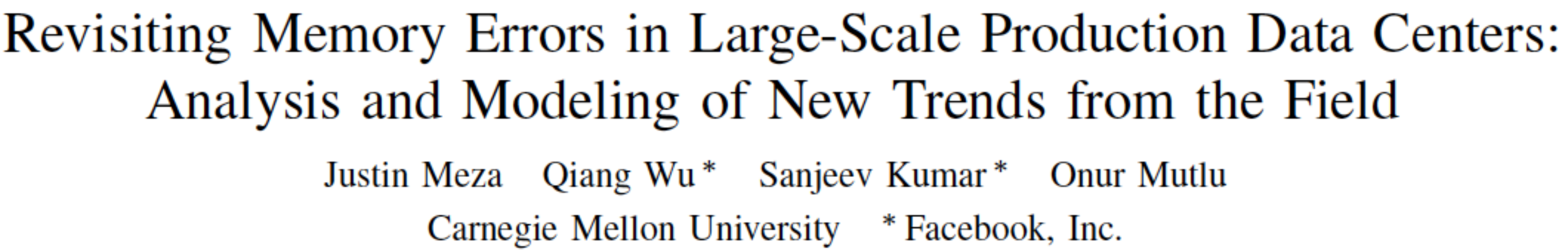
- Source code is released under the liberal MIT License
  - https://github.com/CMU-SAFARI/ramulator

More Detail on DRAM Errors

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]

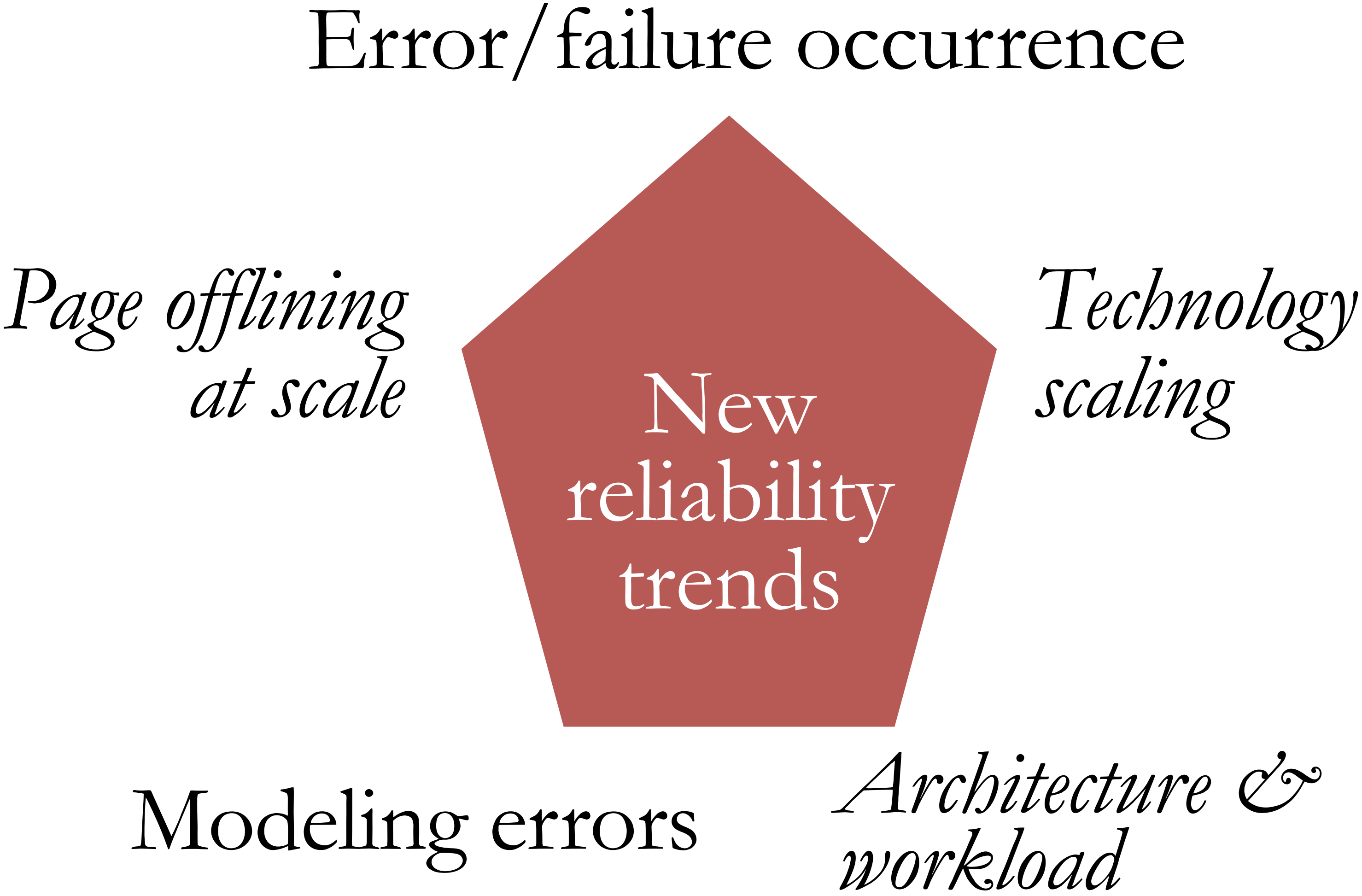


Revisiting Memory Errors in Large-Scale Production Data Centers: Analysis and Modeling of New Trends from the Field

Justin Meza Qiang Wu * Sanjeev Kumar * Onur Mutlu

Carnegie Mellon University * Facebook, Inc.

Findings



Error/failure occurrence

Page offlining at scale

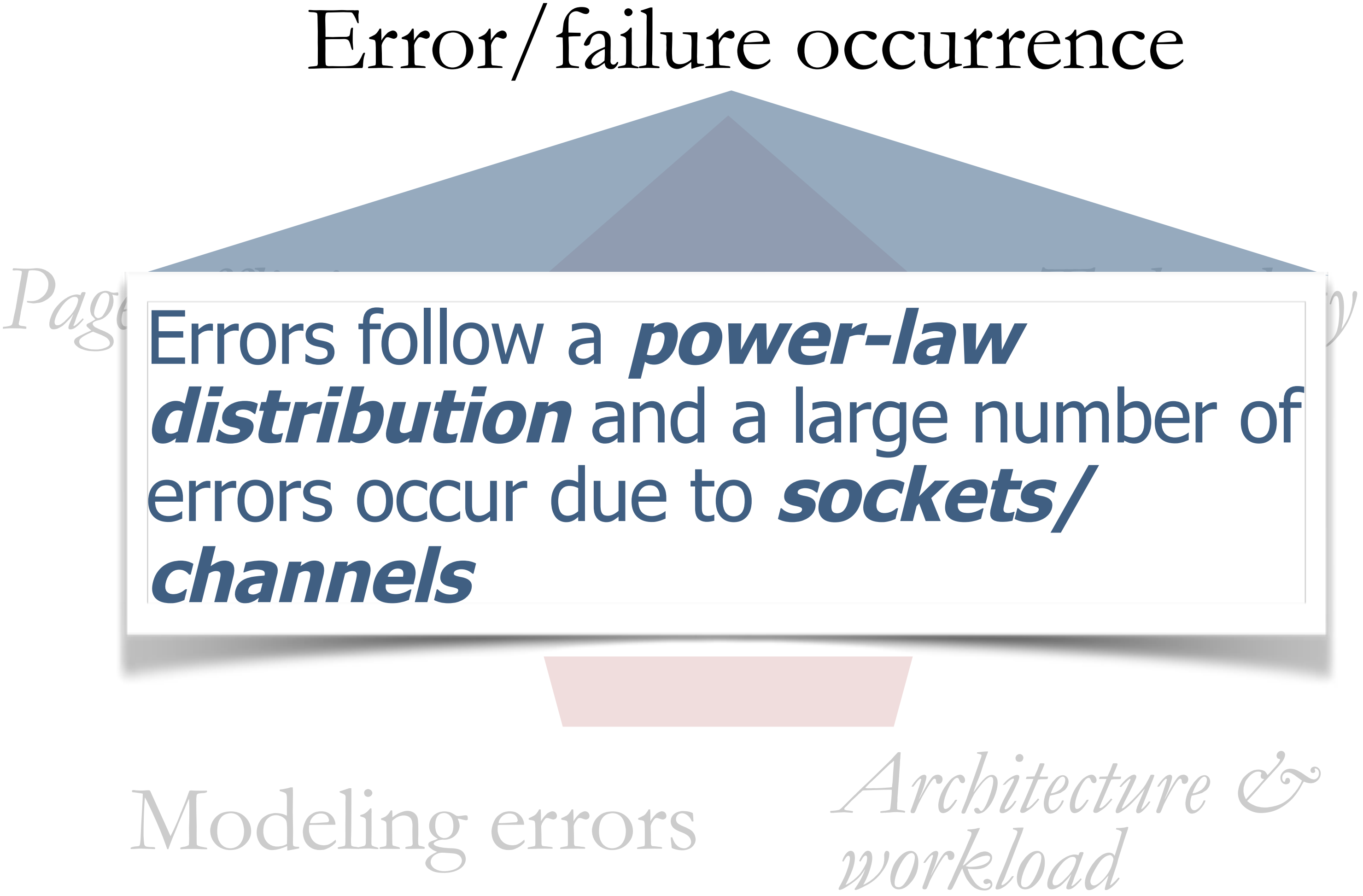
New reliability trends

Technology scaling

Modeling errors

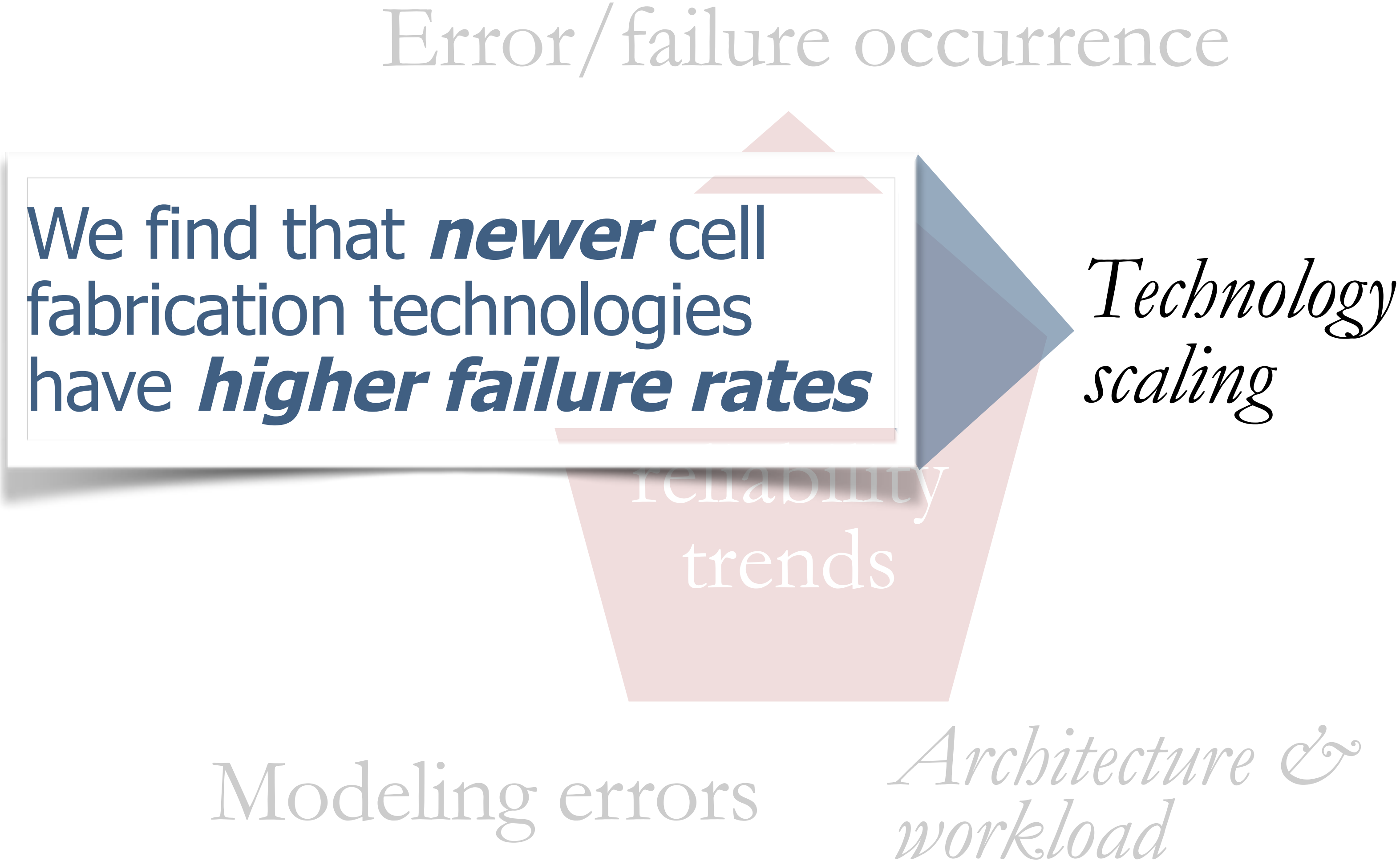
Architecture & workload

Findings



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

Findings



Error/failure occurrence

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

Technology scaling

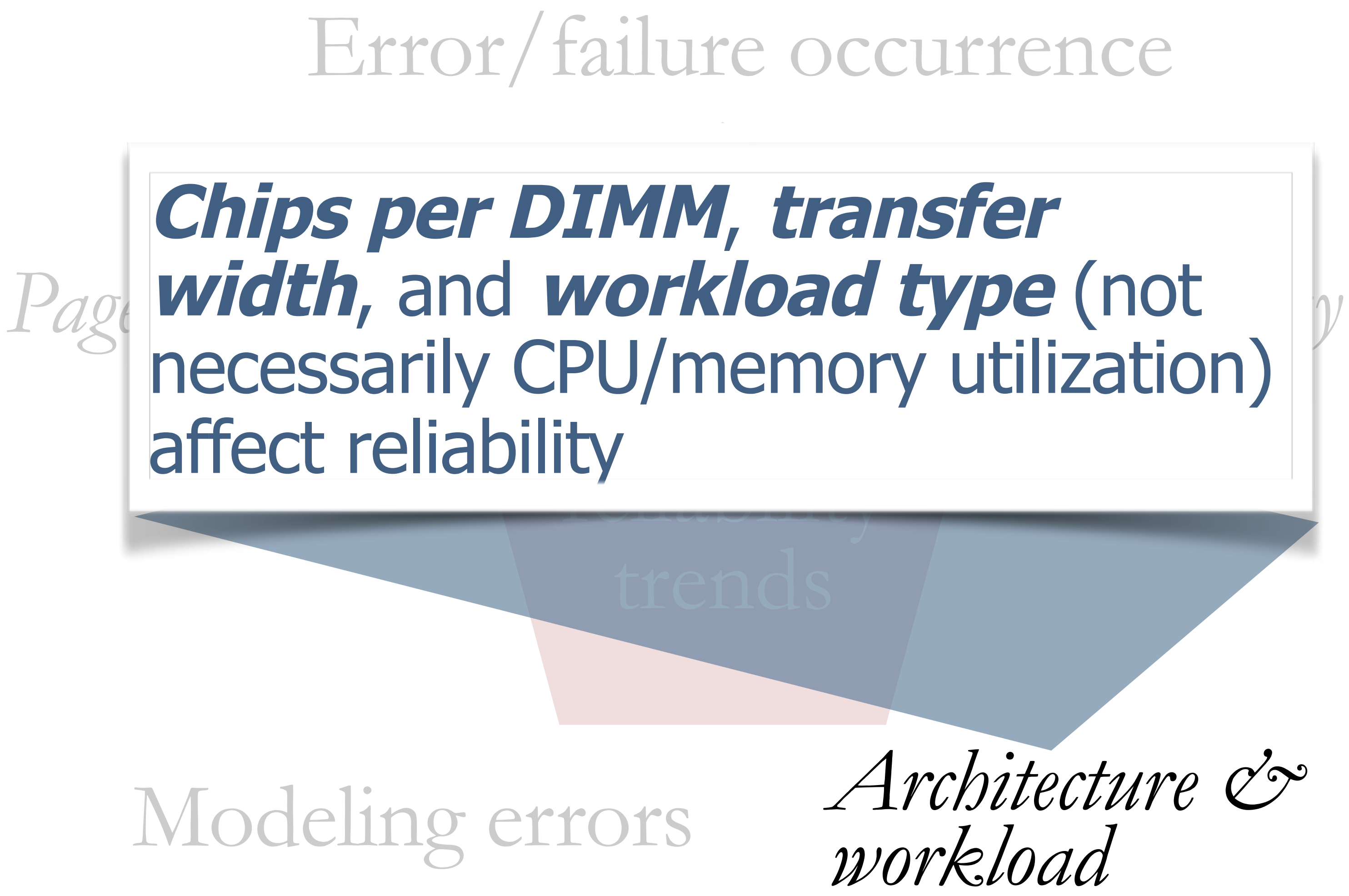
reliability trends

Modeling errors

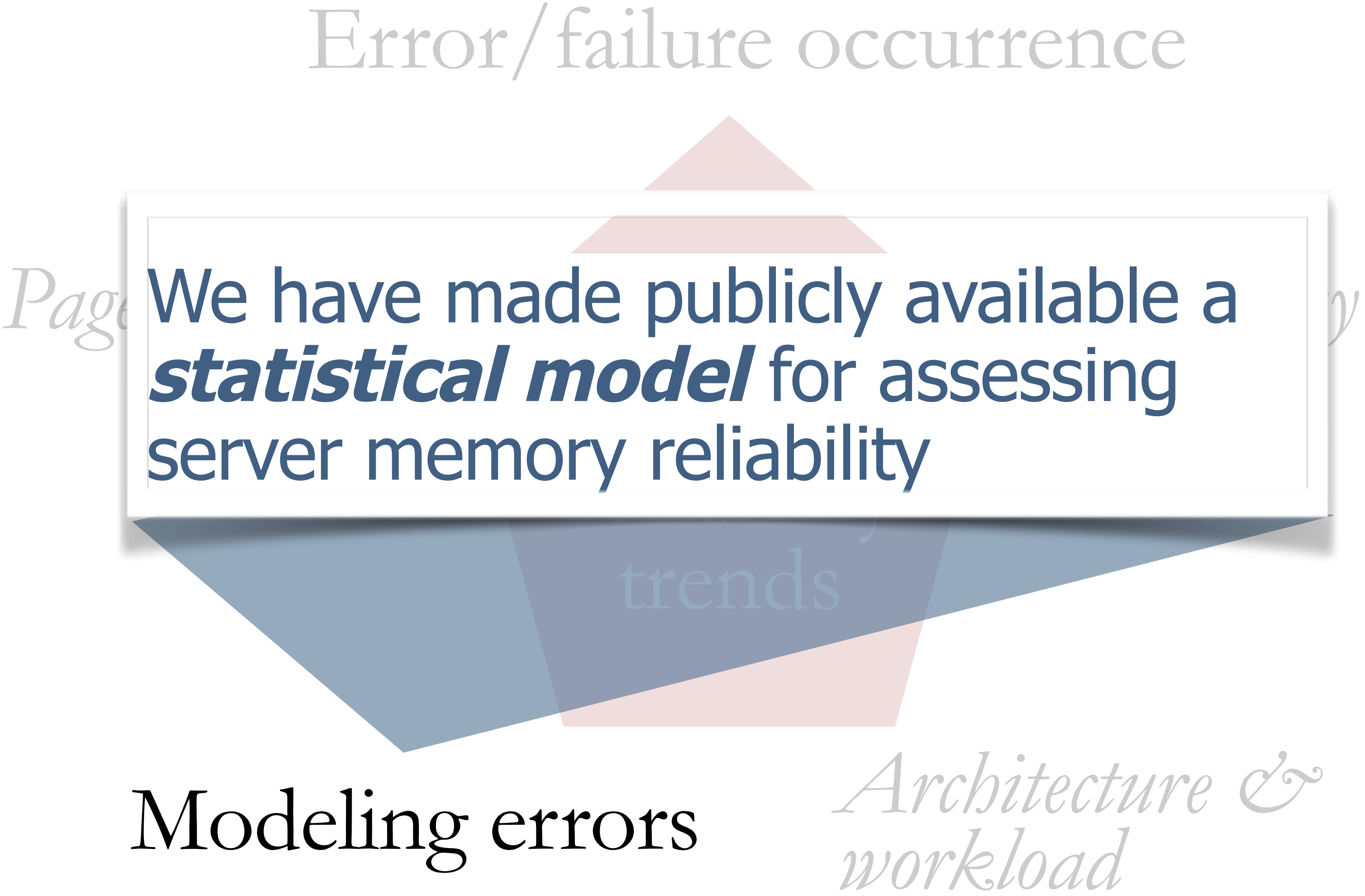
Architecture & workload

Findings

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

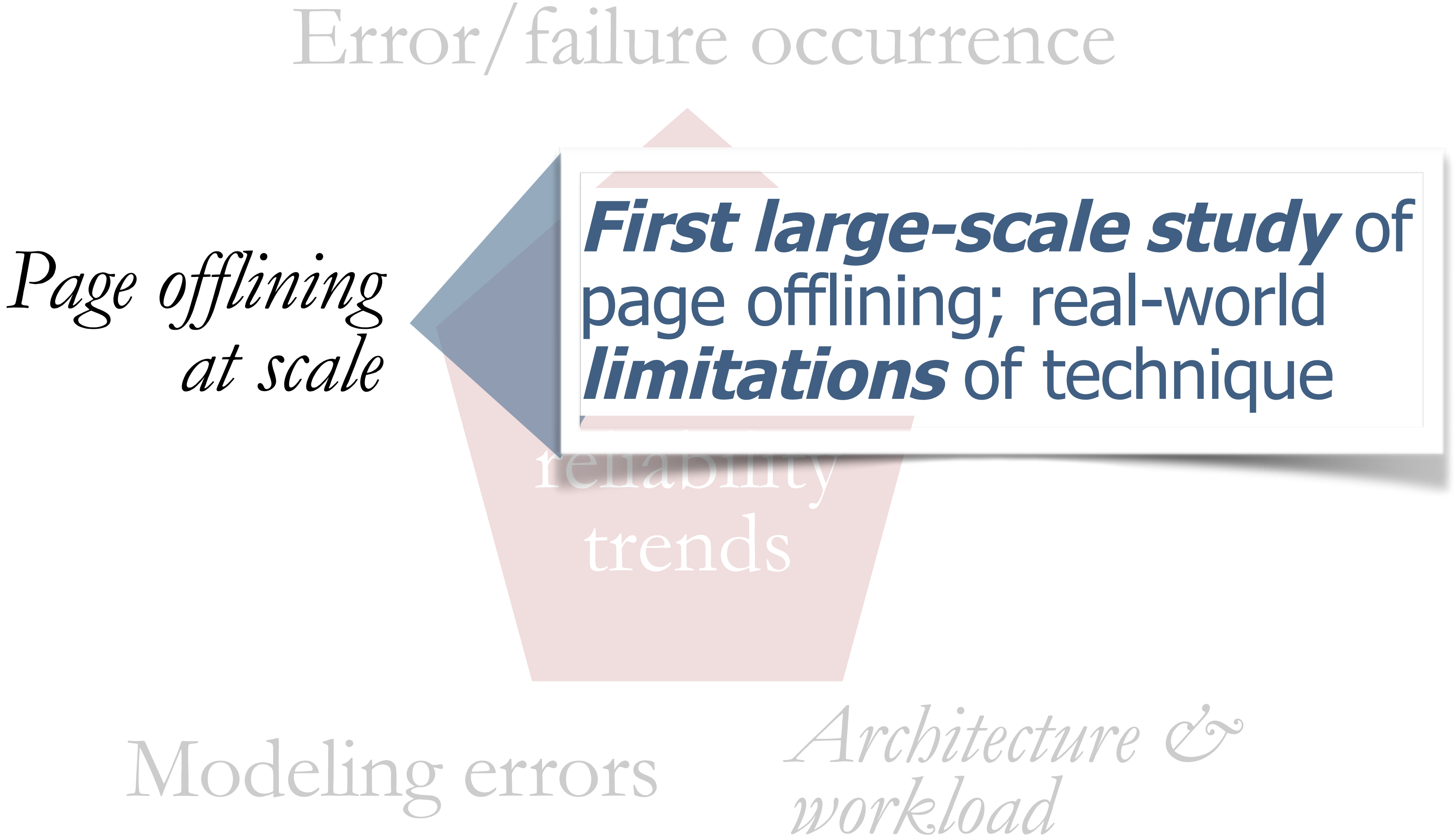


Findings



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

Findings



Error/failure occurrence

Page offlining at scale

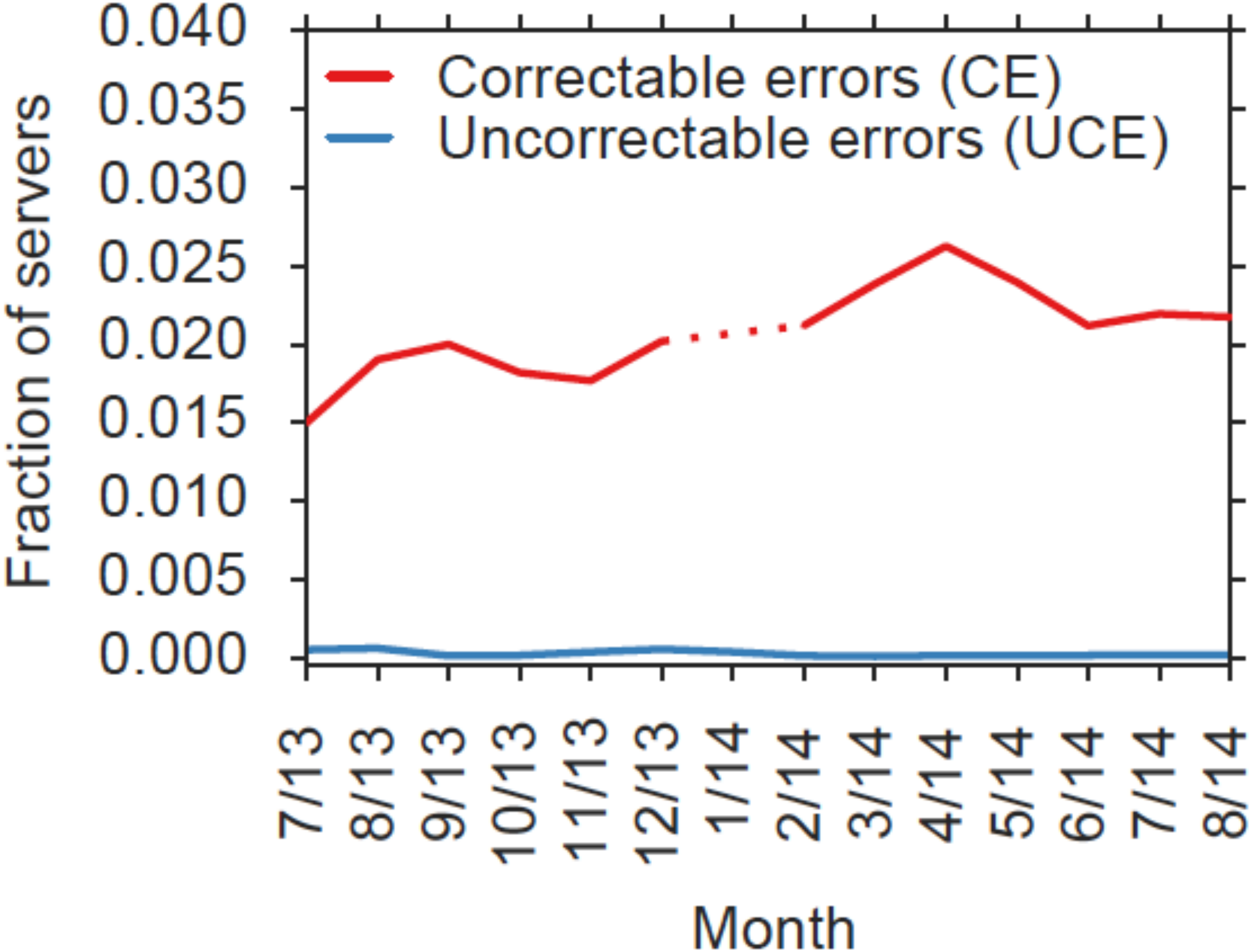
First large-scale study of page offlining; real-world ***limitations*** of technique

reliability trends

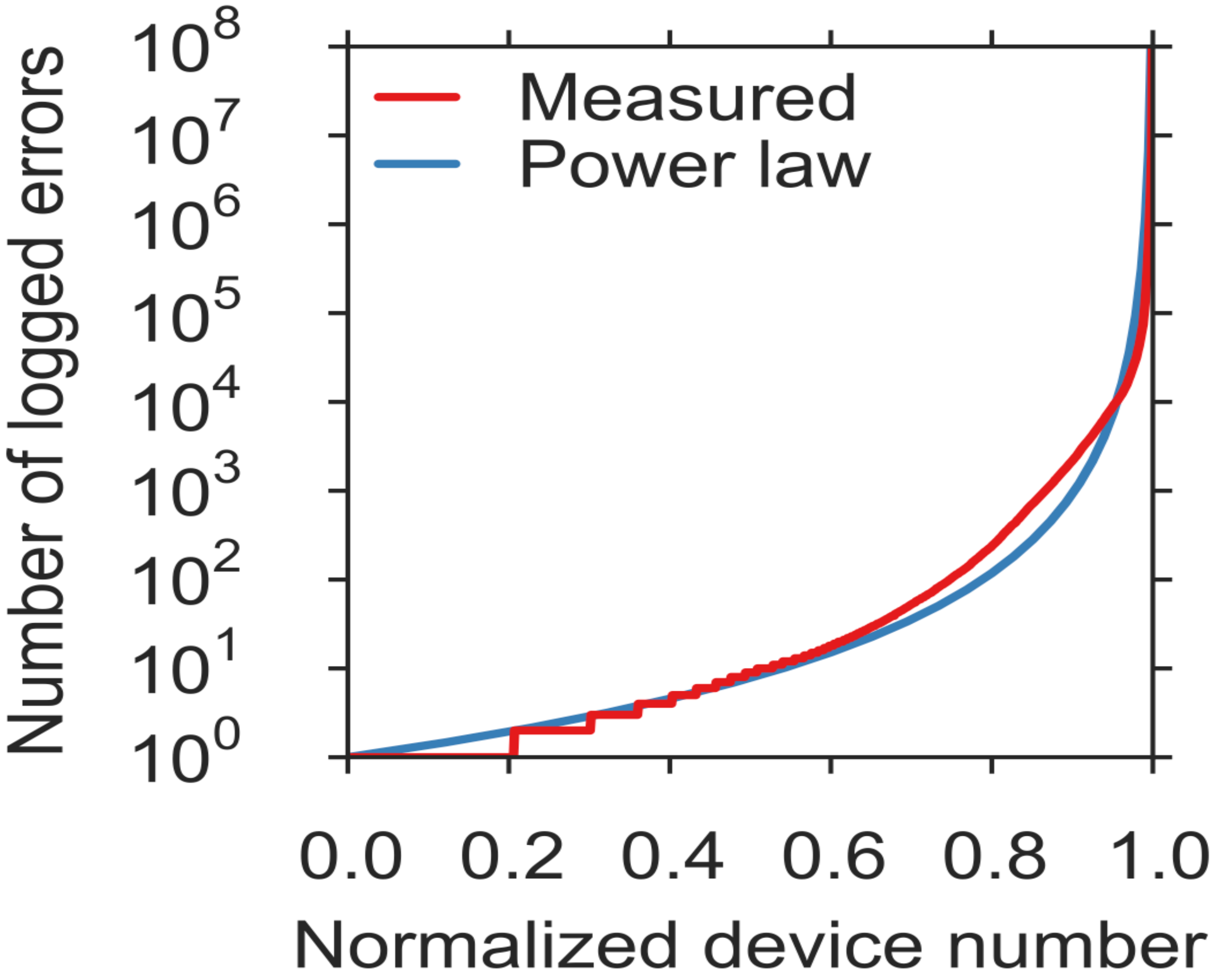
Modeling errors

Architecture & workload

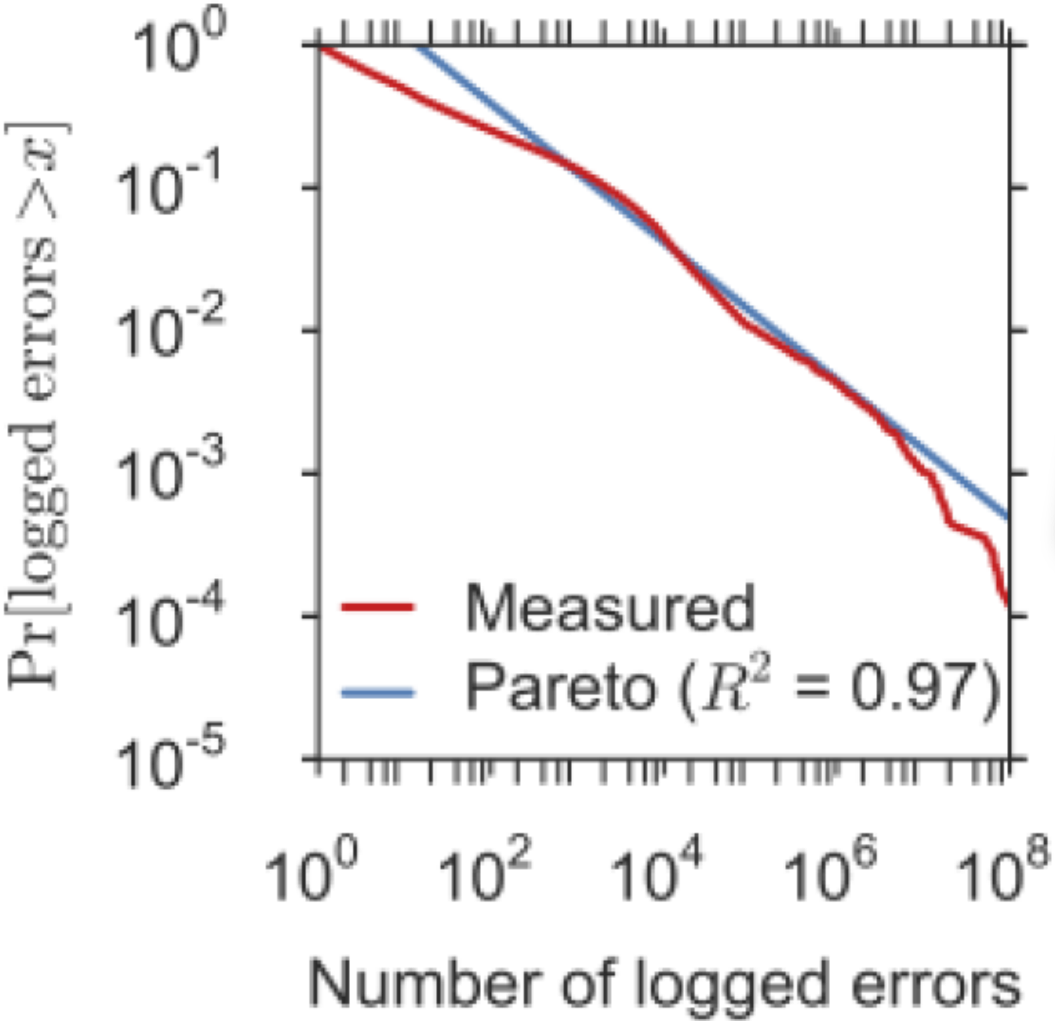
Server error rate



Memory error distribution



Memory error distribution

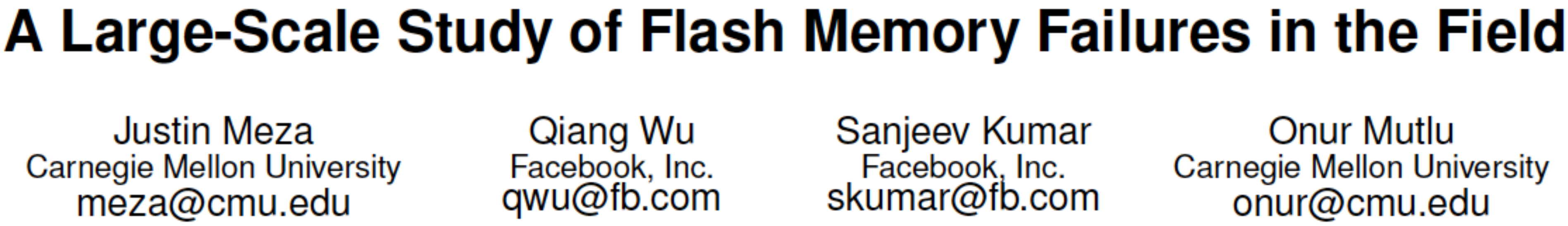


Decreasing hazard rate

Large Scale Field Analysis of Flash Memory Errors

SSD Error Analysis of Facebook Systems

- 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

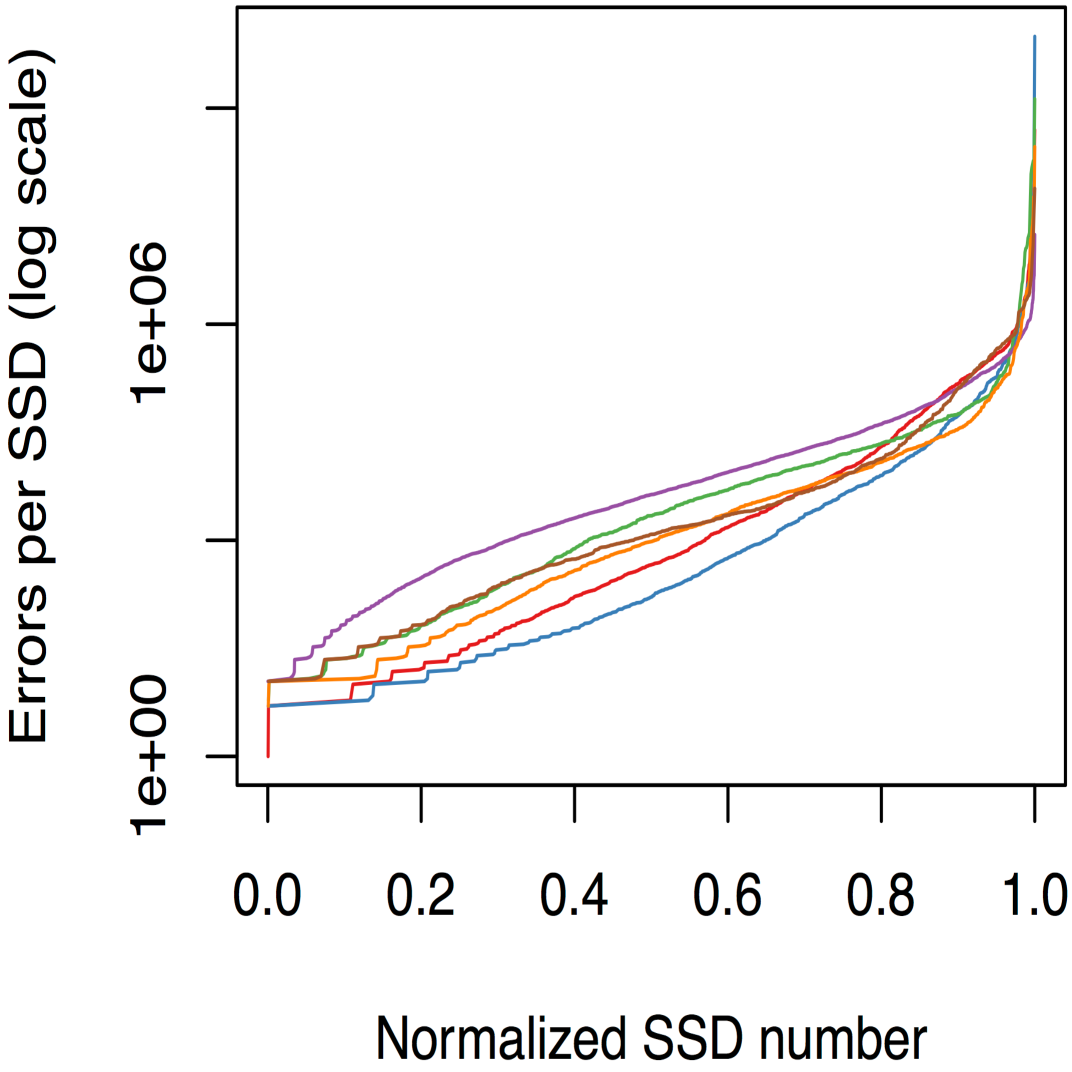
Justin Meza
Carnegie Mellon University
meza@cmu.edu

Qiang Wu
Facebook, Inc.
qwu@fb.com

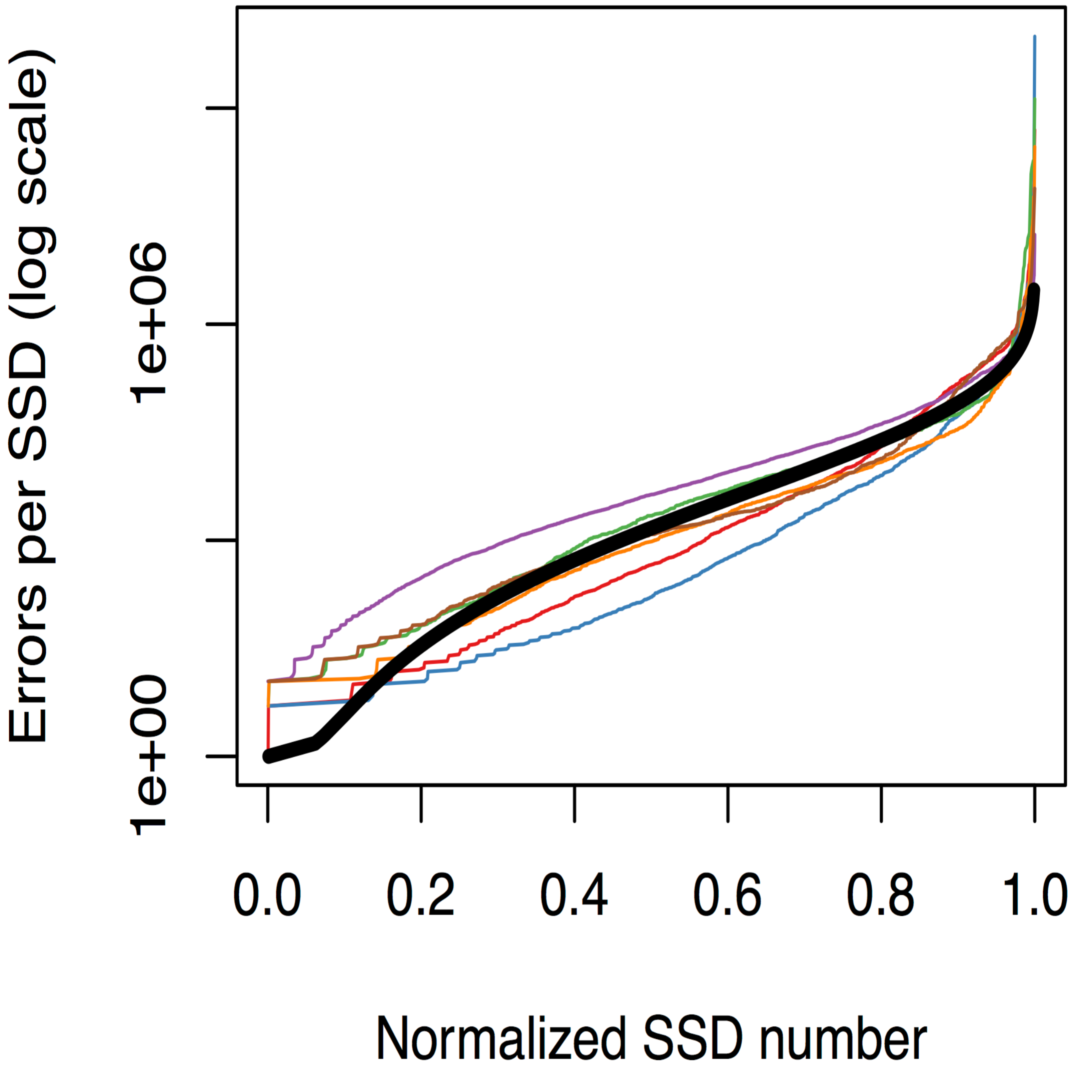
Sanjeev Kumar
Facebook, Inc.
skumar@fb.com

Onur Mutlu
Carnegie Mellon University
onur@cmu.edu

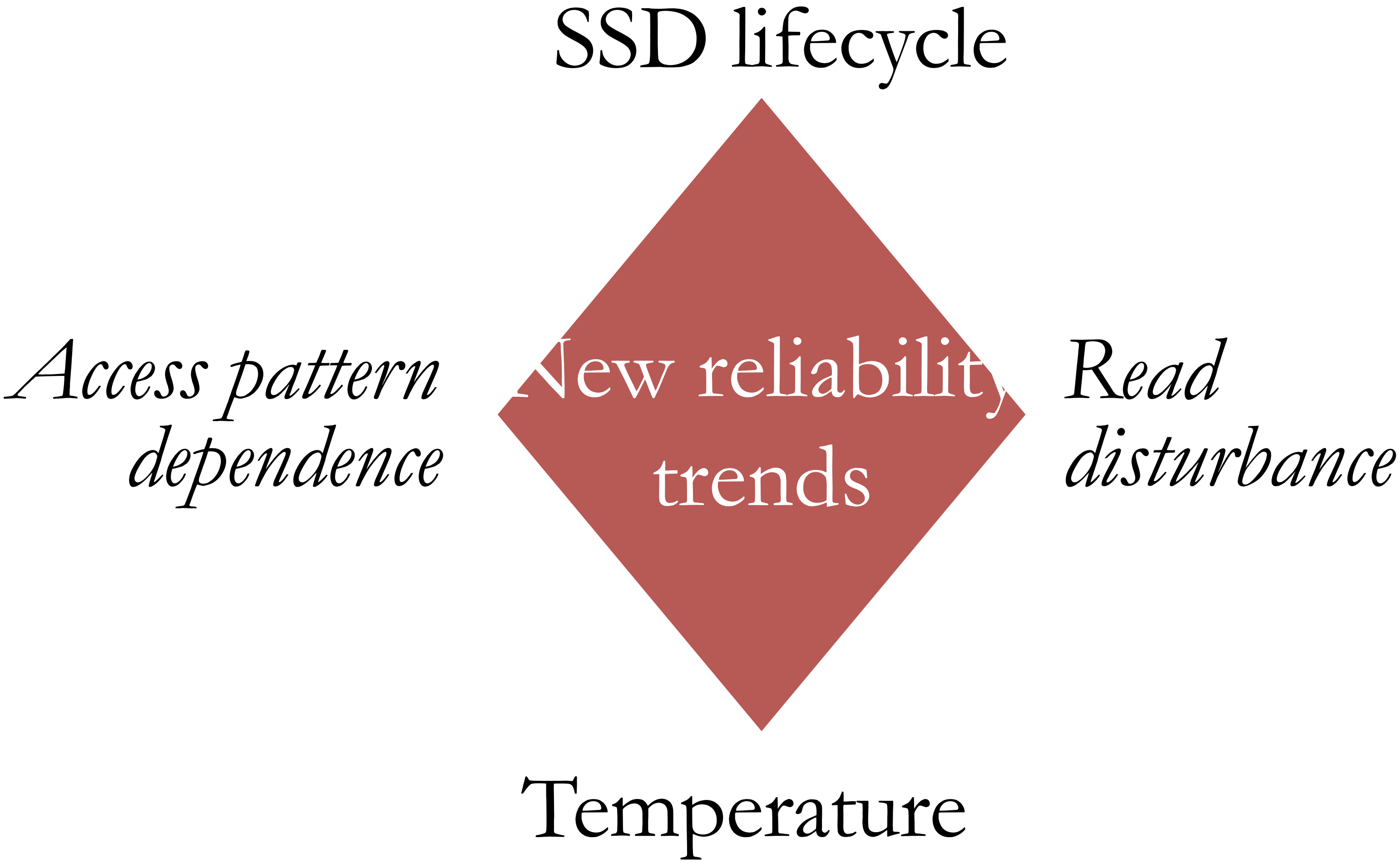
A few SSDs cause most errors



A few SSDs cause most errors



Summary



SSD lifecycle

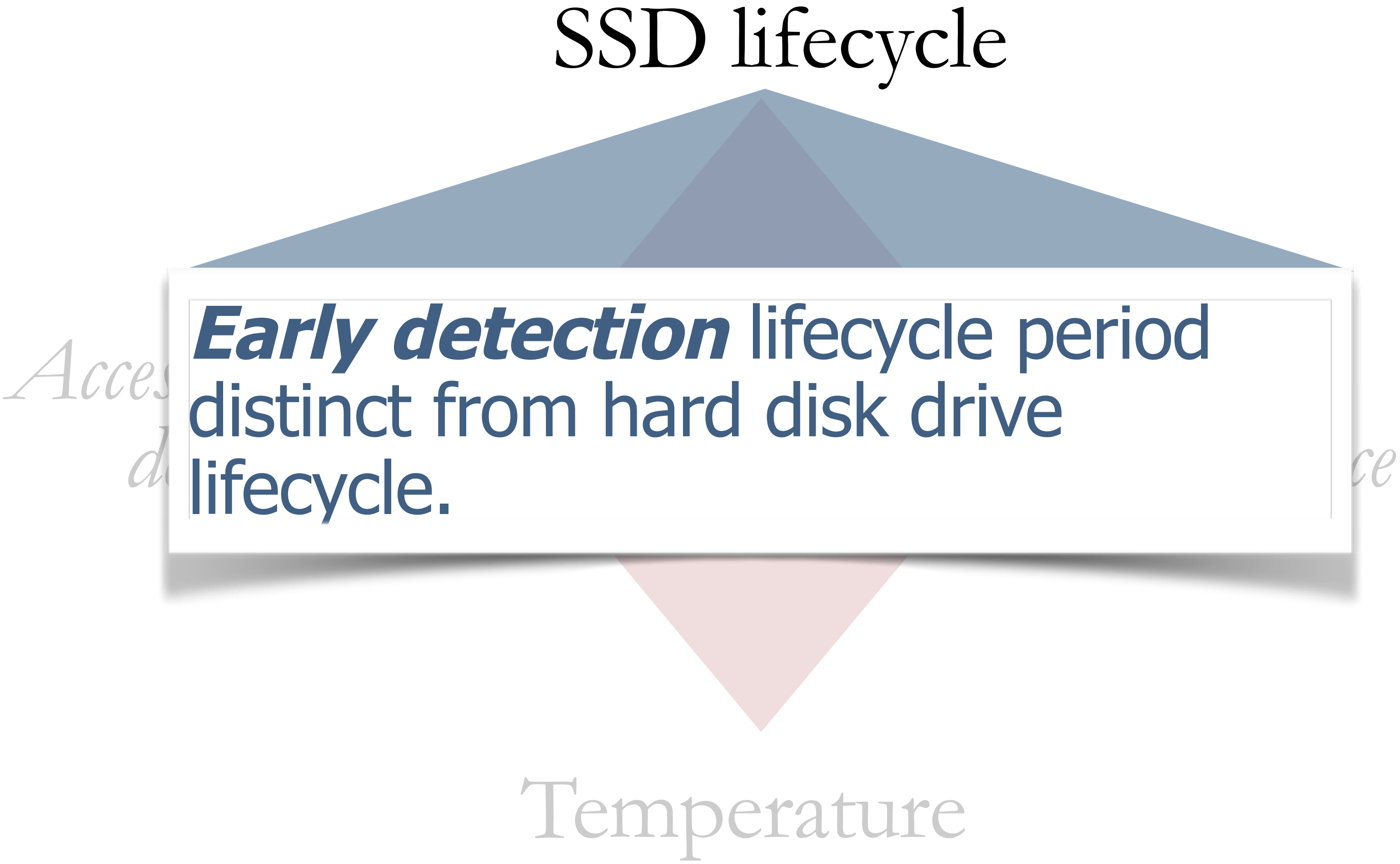
Access pattern dependence

New reliability trends

Read disturbance

Temperature

Summary



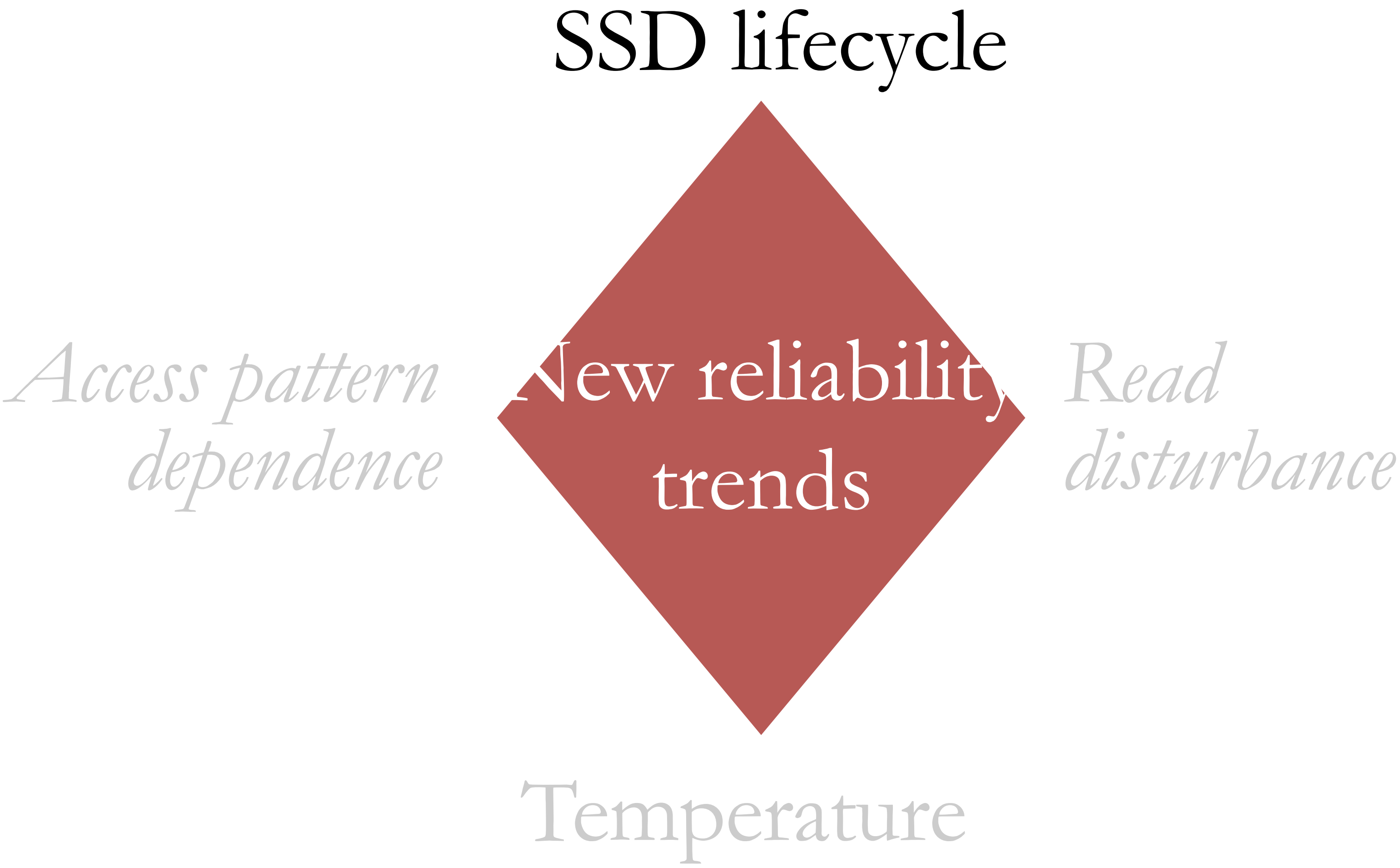
SSD lifecycle

Acces
d

ce

Temperature

Early detection lifecycle period distinct from hard disk drive lifecycle.



SSD lifecycle

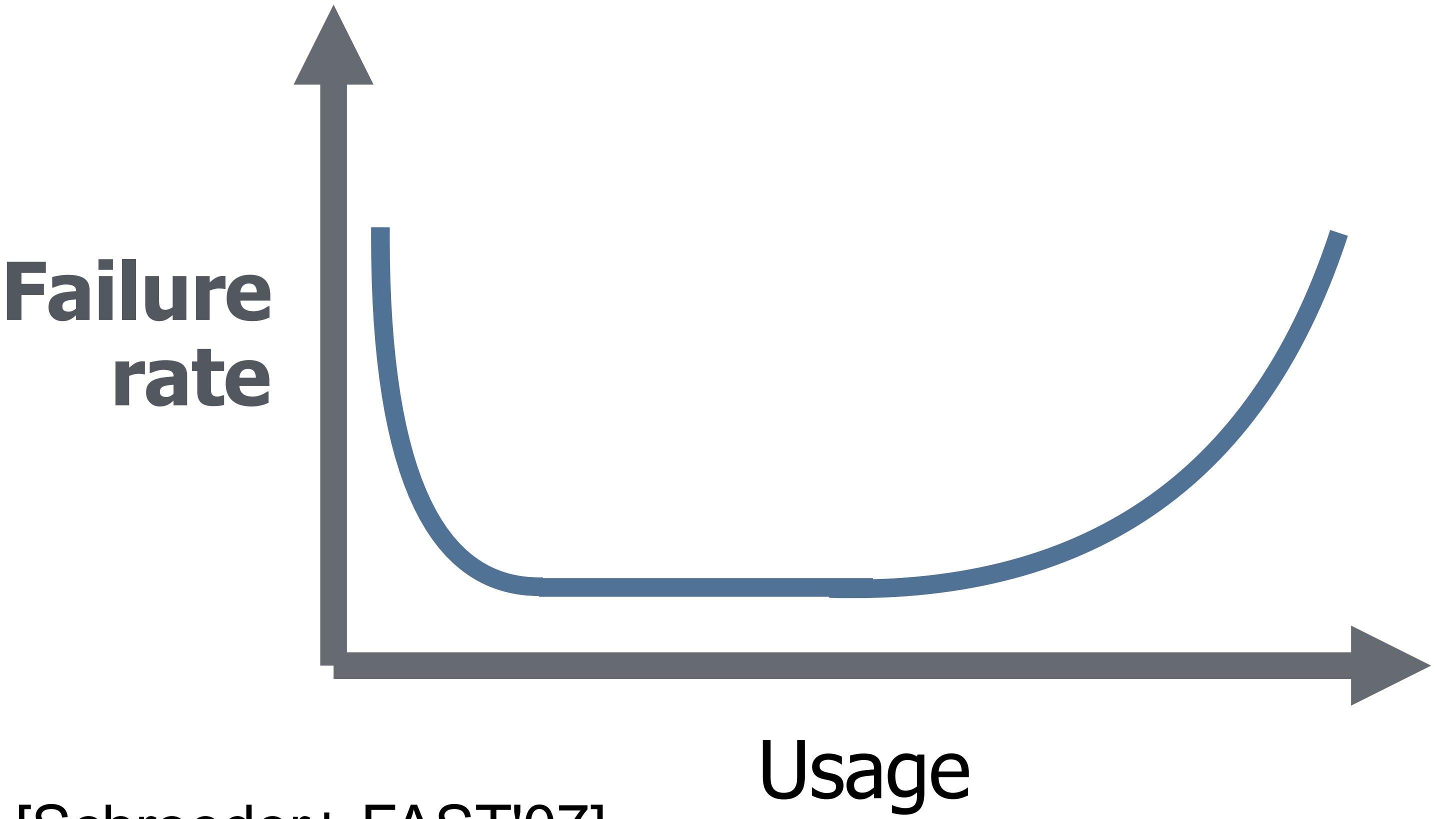
Access pattern dependence

New reliability trends

Read disturbance

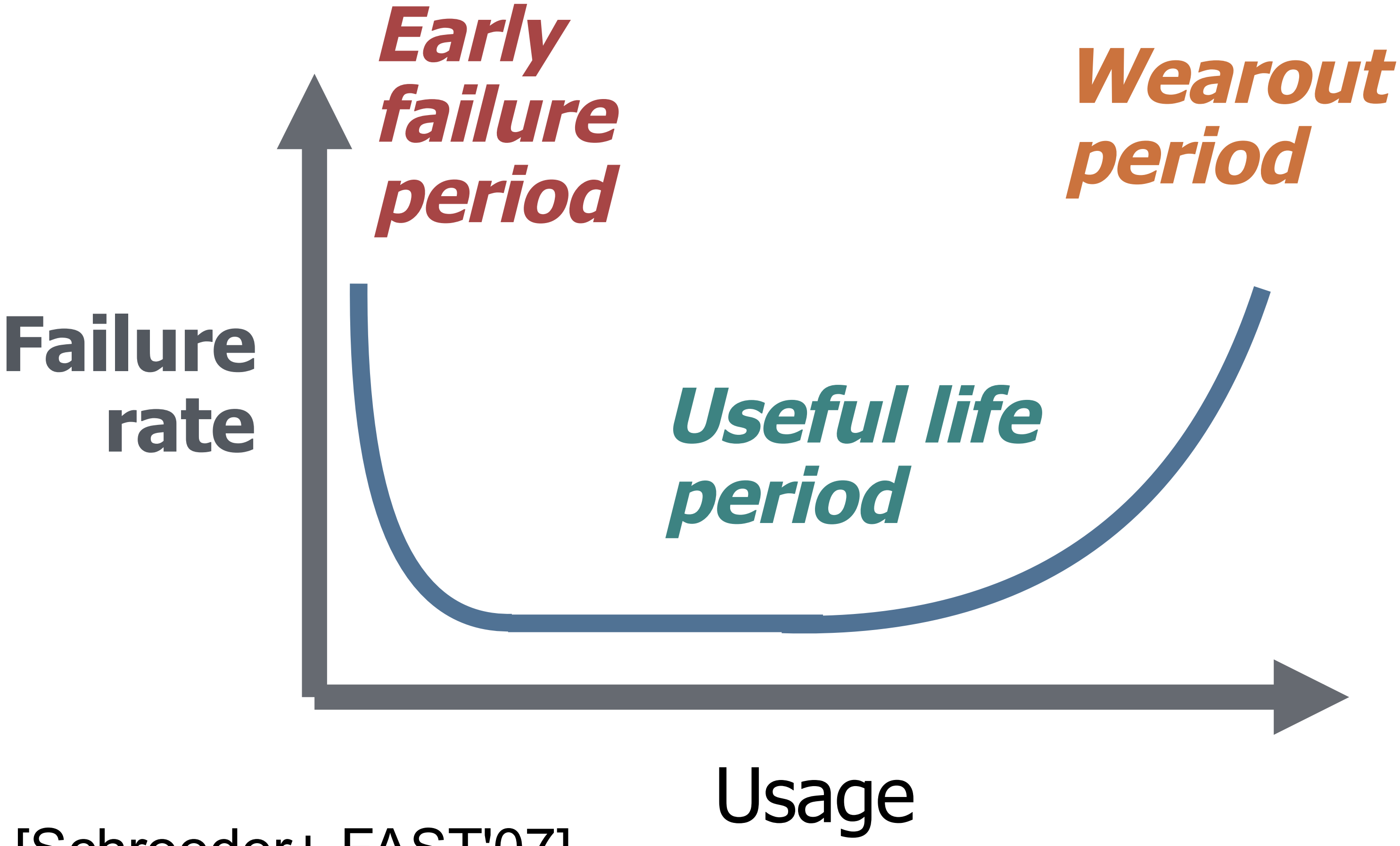
Temperature

Storage lifecycle background: the bathtub curve for disk drives



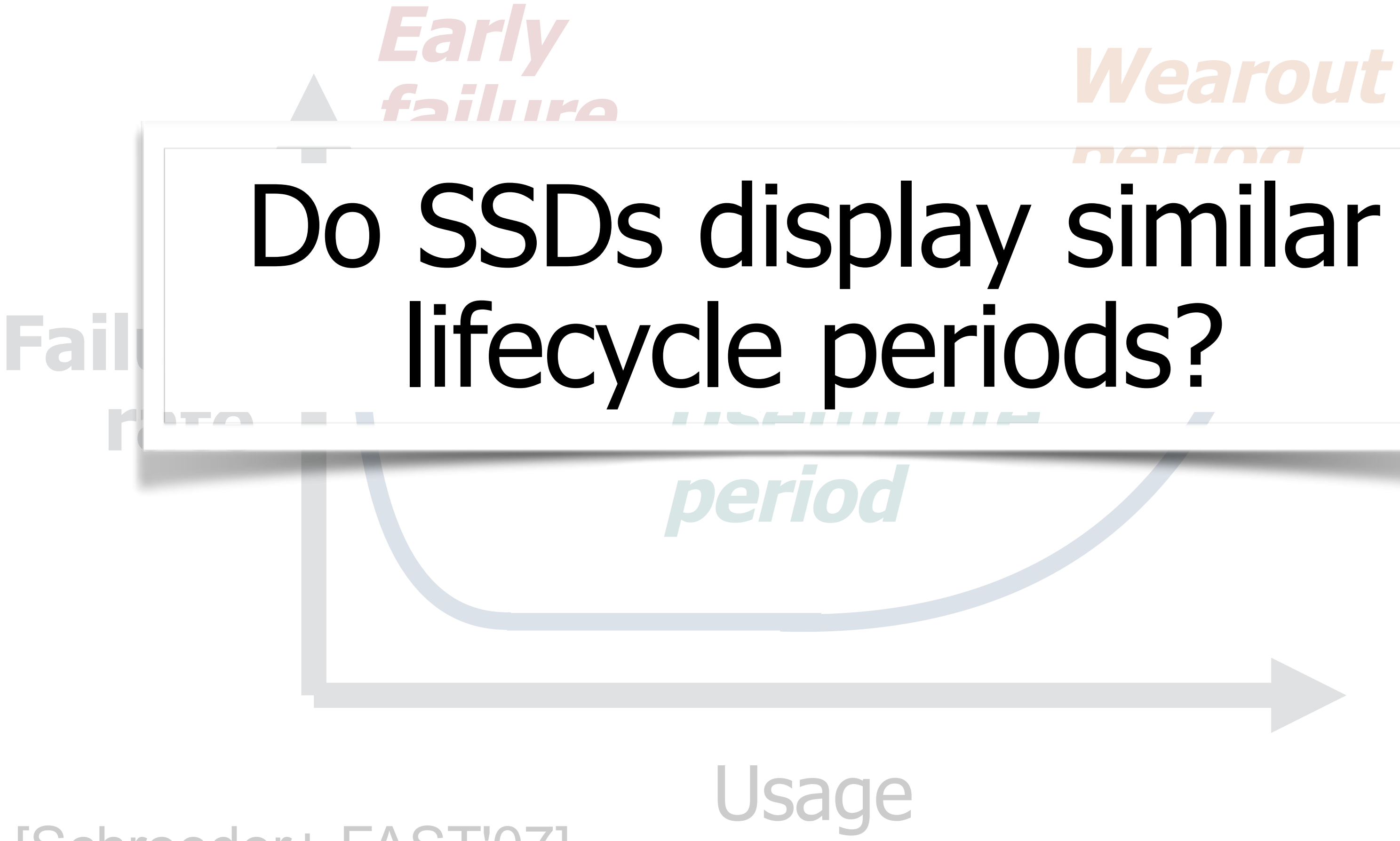
[Schroeder+,FAST'07]

Storage lifecycle background: the bathtub curve for disk drives



[Schroeder+,FAST'07]

Storage lifecycle background: the bathtub curve for disk drives



Do SSDs display similar lifecycle periods?

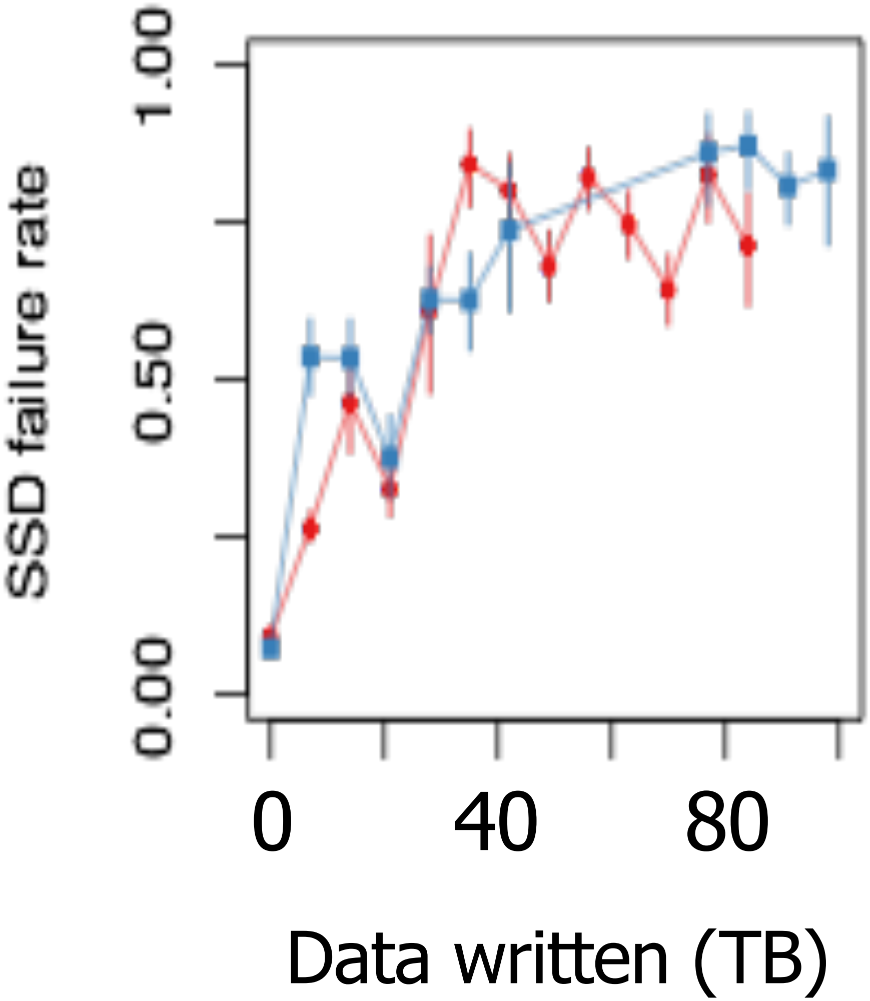
[Schroeder+,FAST'07]

Use data written to flash

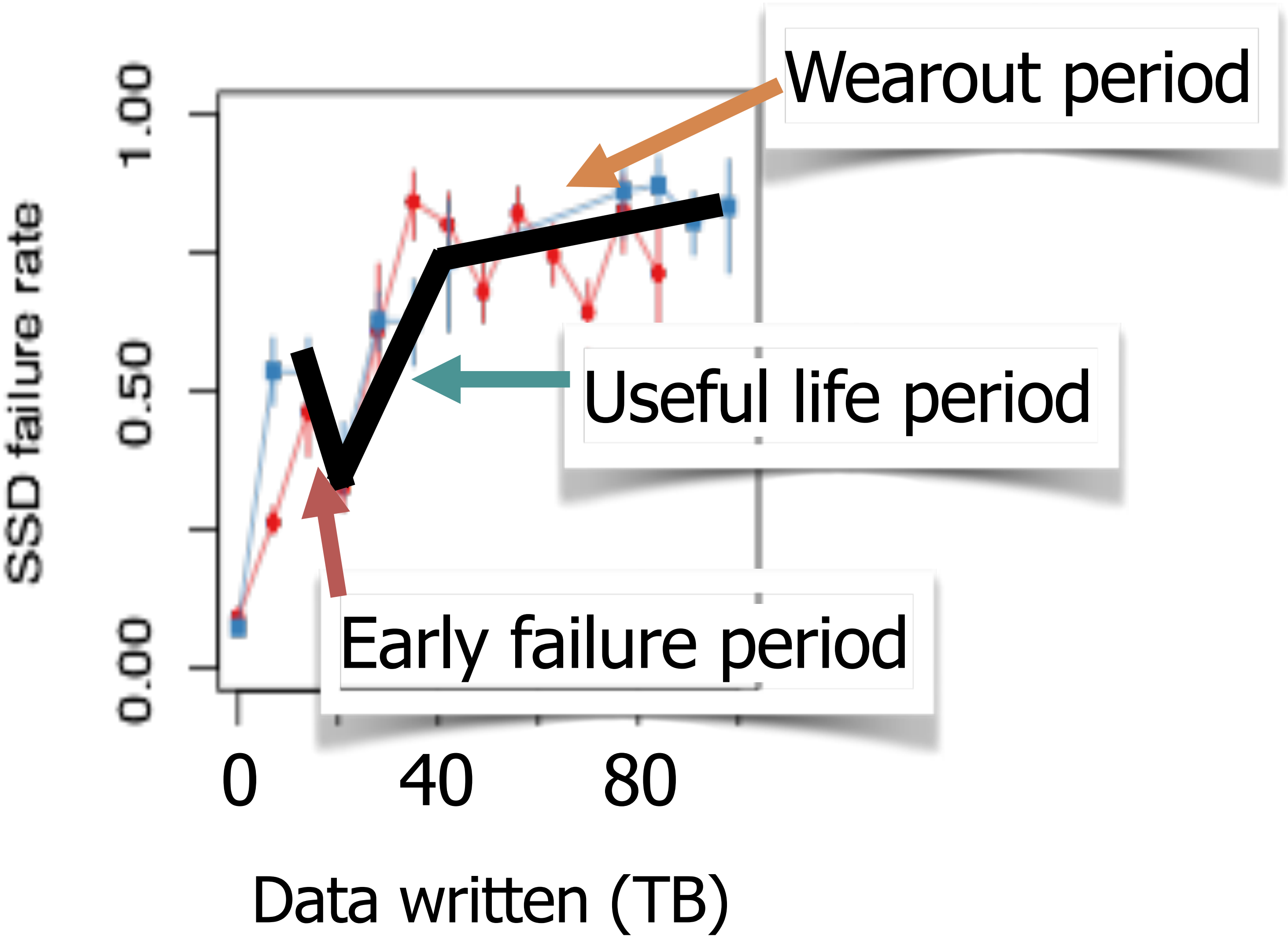
to examine SSD lifecycle

(time-independent utilization metric)

720GB, 1 SSD   720GB, 2 SSDs



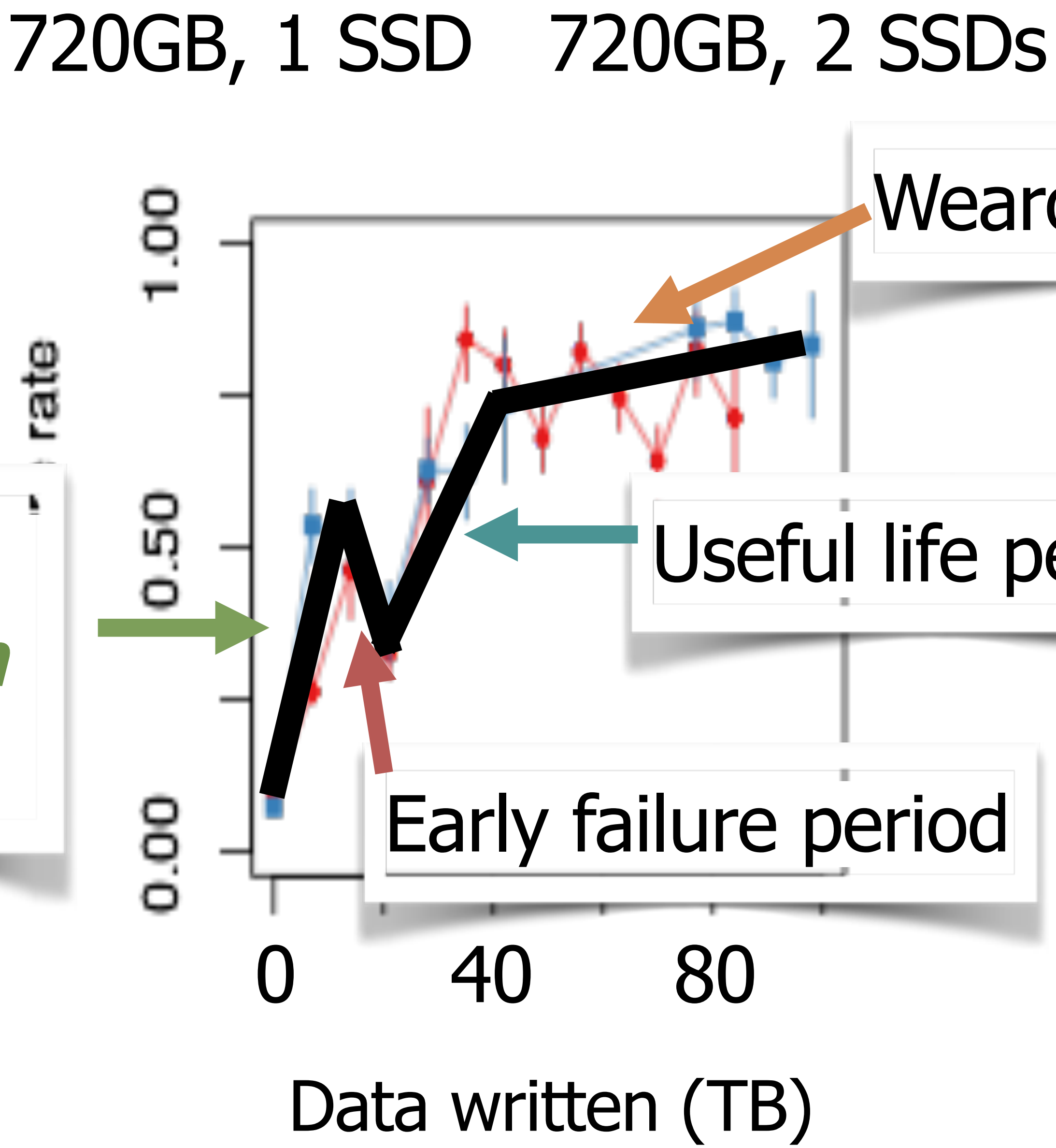
720GB, 1 SSD 720GB, 2 SSDs



Wearout period

Useful life period

Early failure period



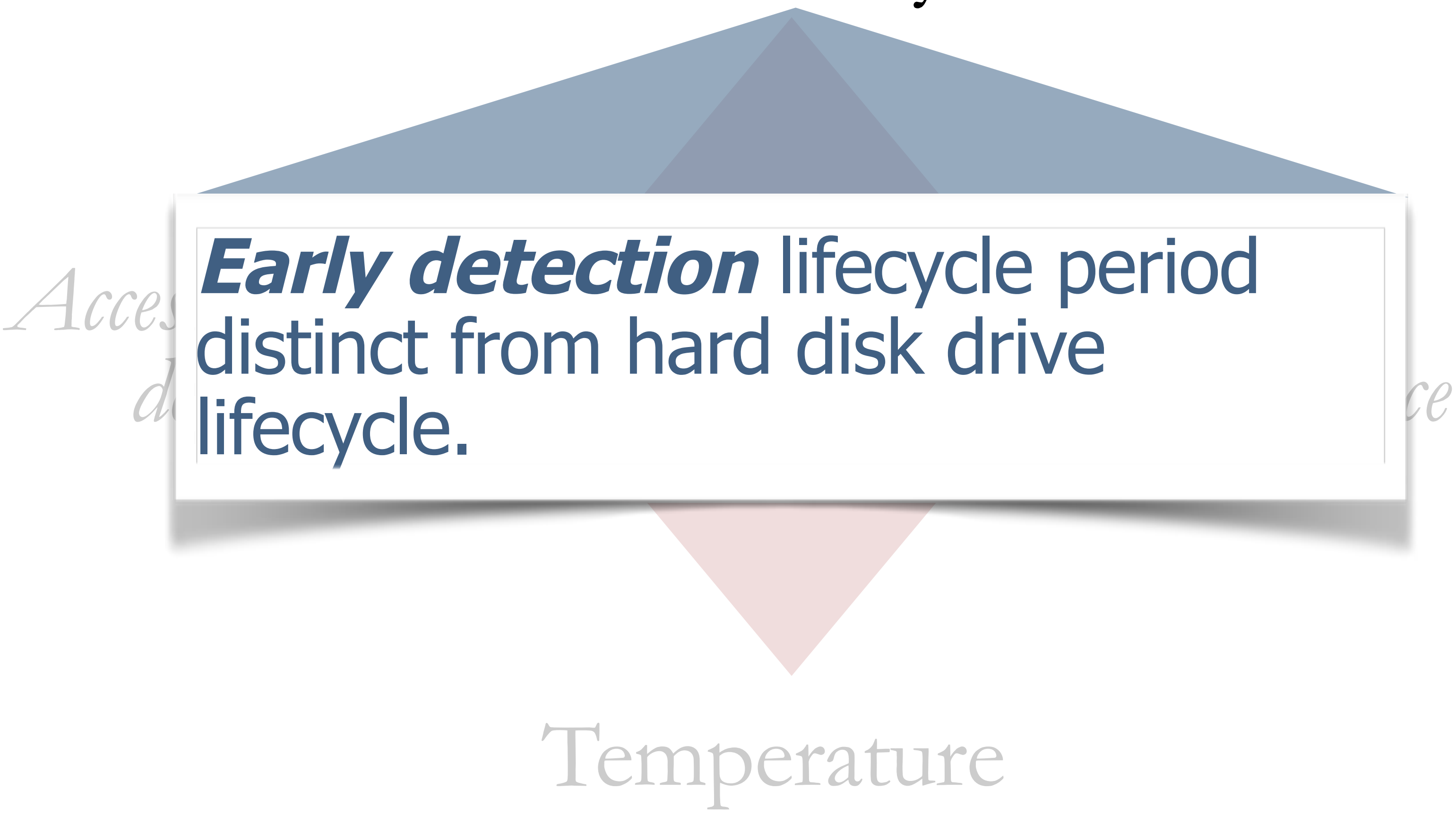
Wearout period

Useful life period

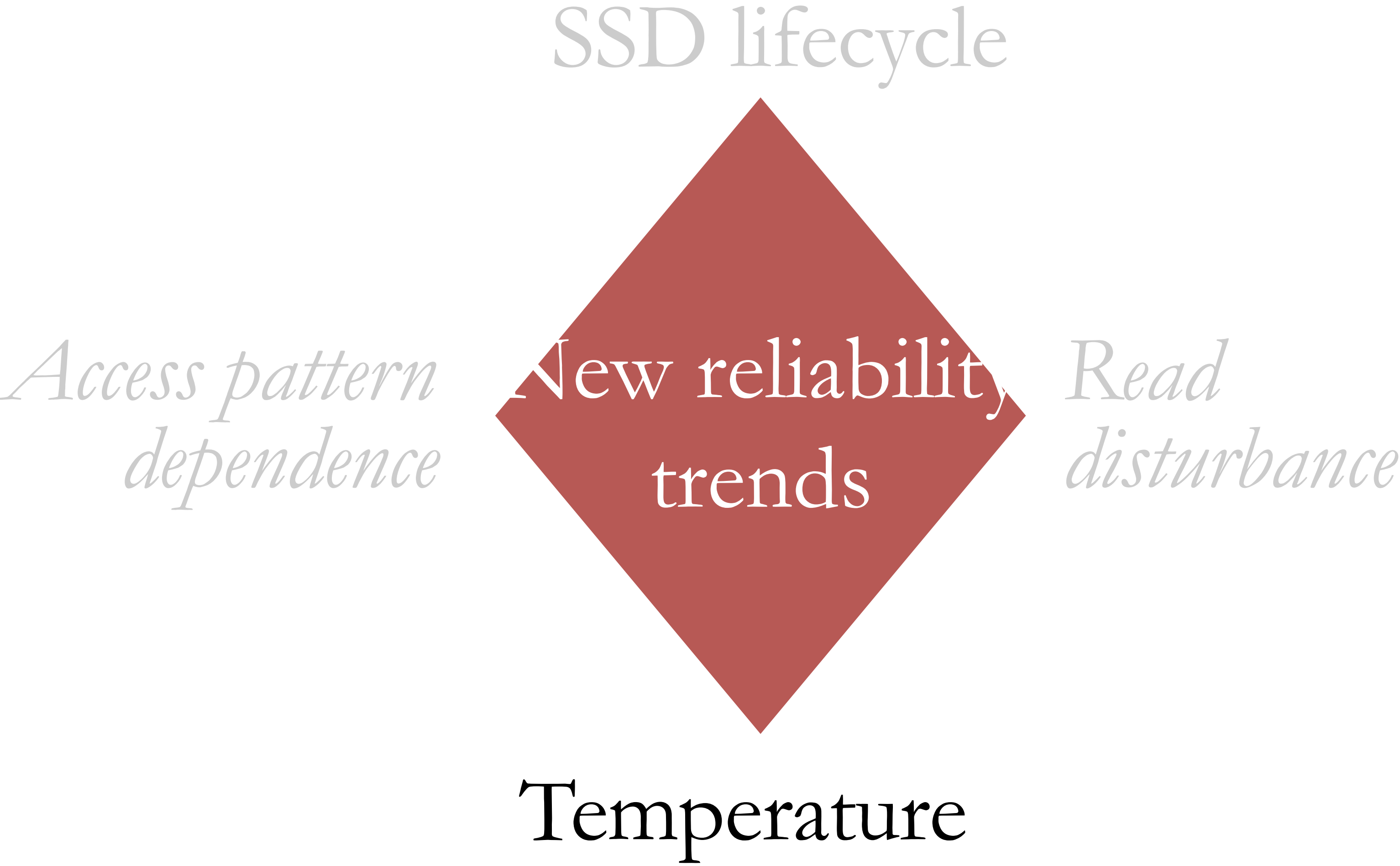
Early detection period

Early failure period

SSD lifecycle



Early detection lifecycle period distinct from hard disk drive lifecycle.



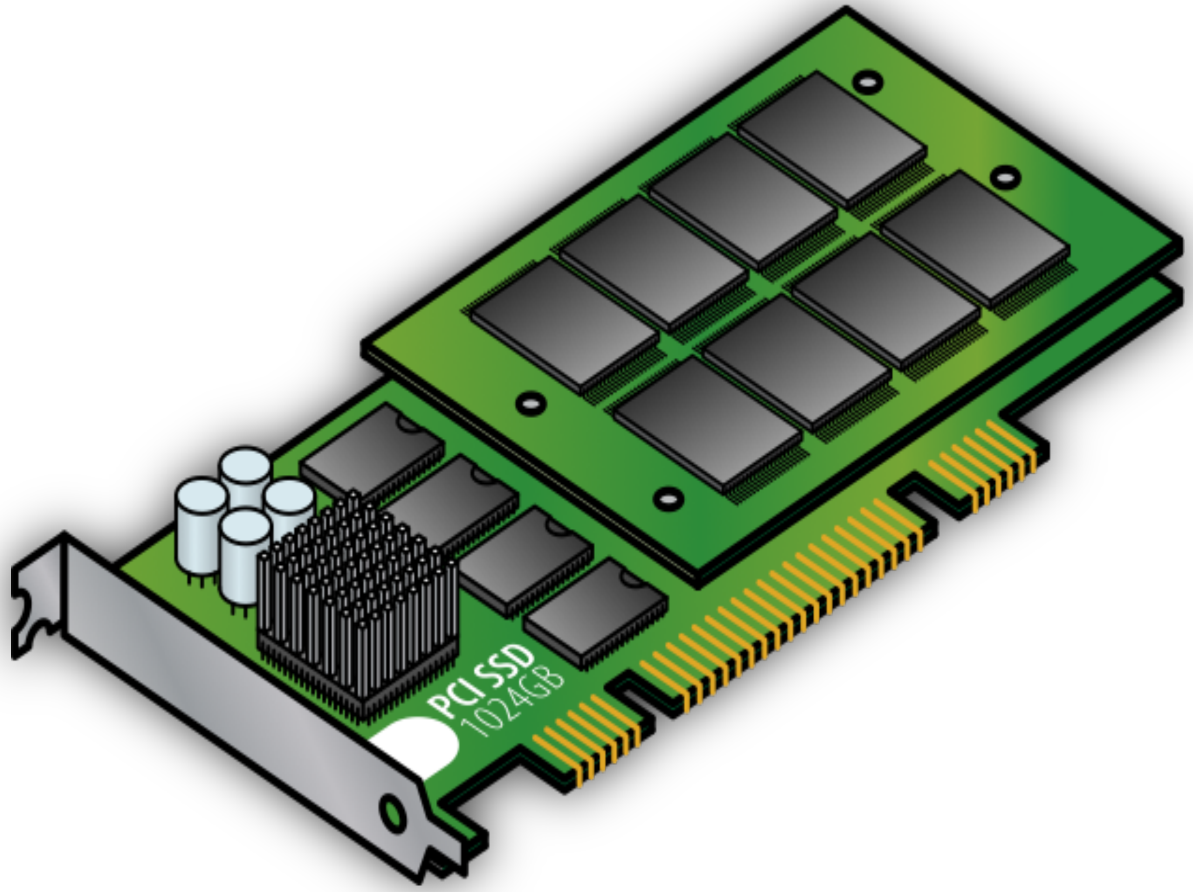
SSD lifecycle

Access pattern dependence

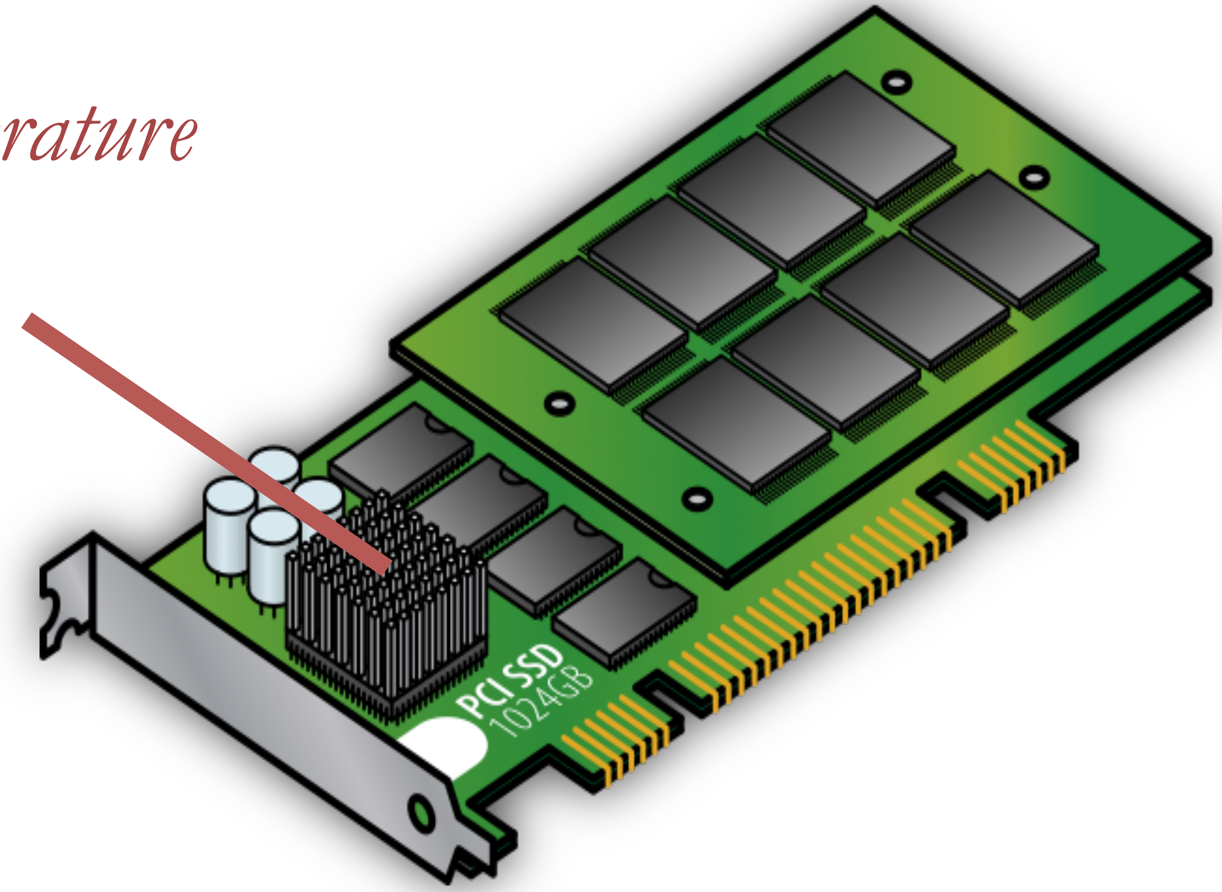
New reliability trends

Read disturbance

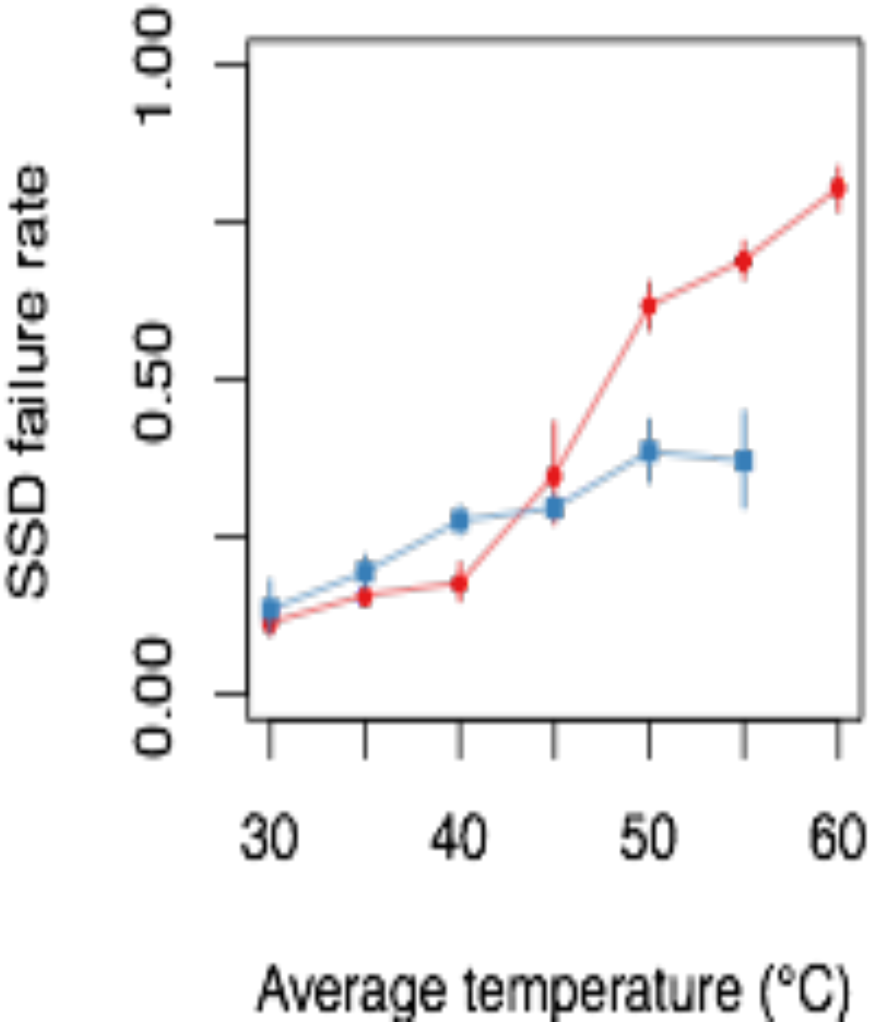
Temperature


PCI SSD
1024GB

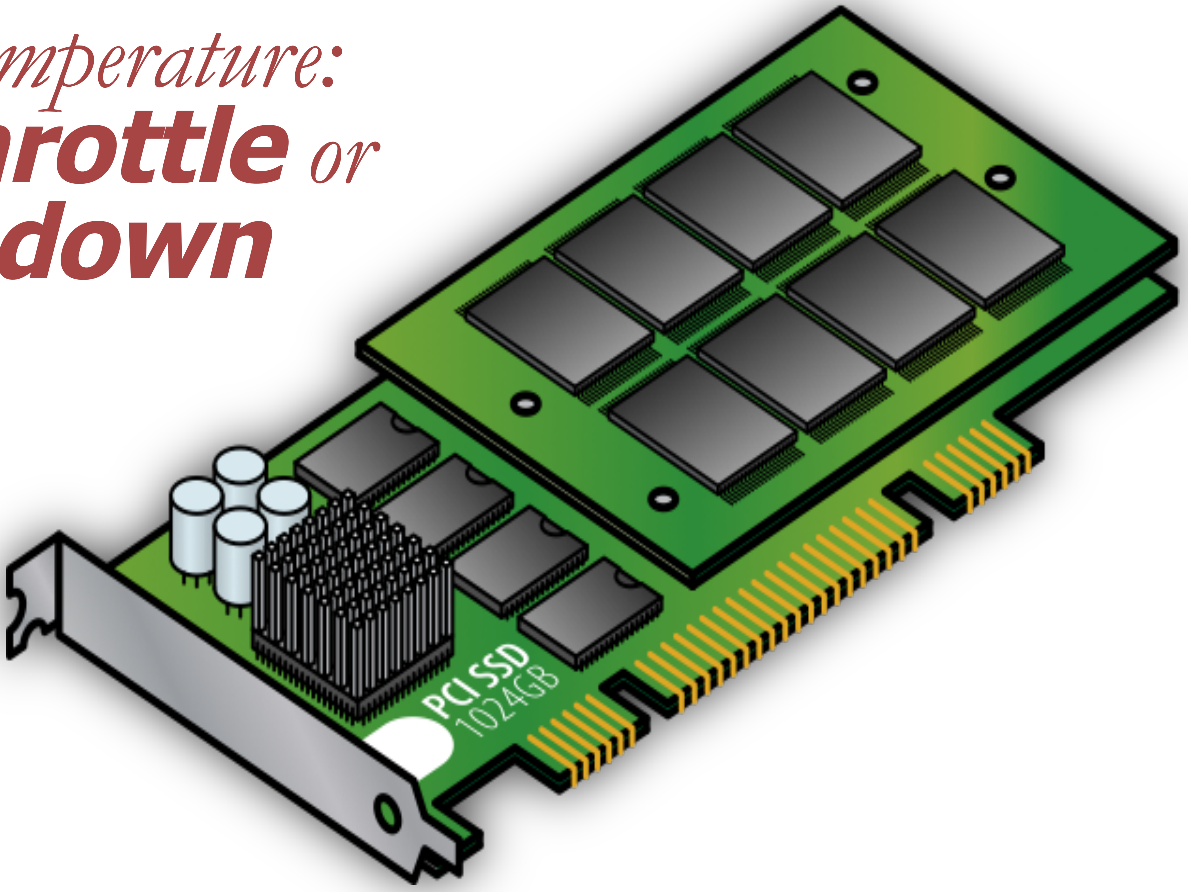
Temperature sensor



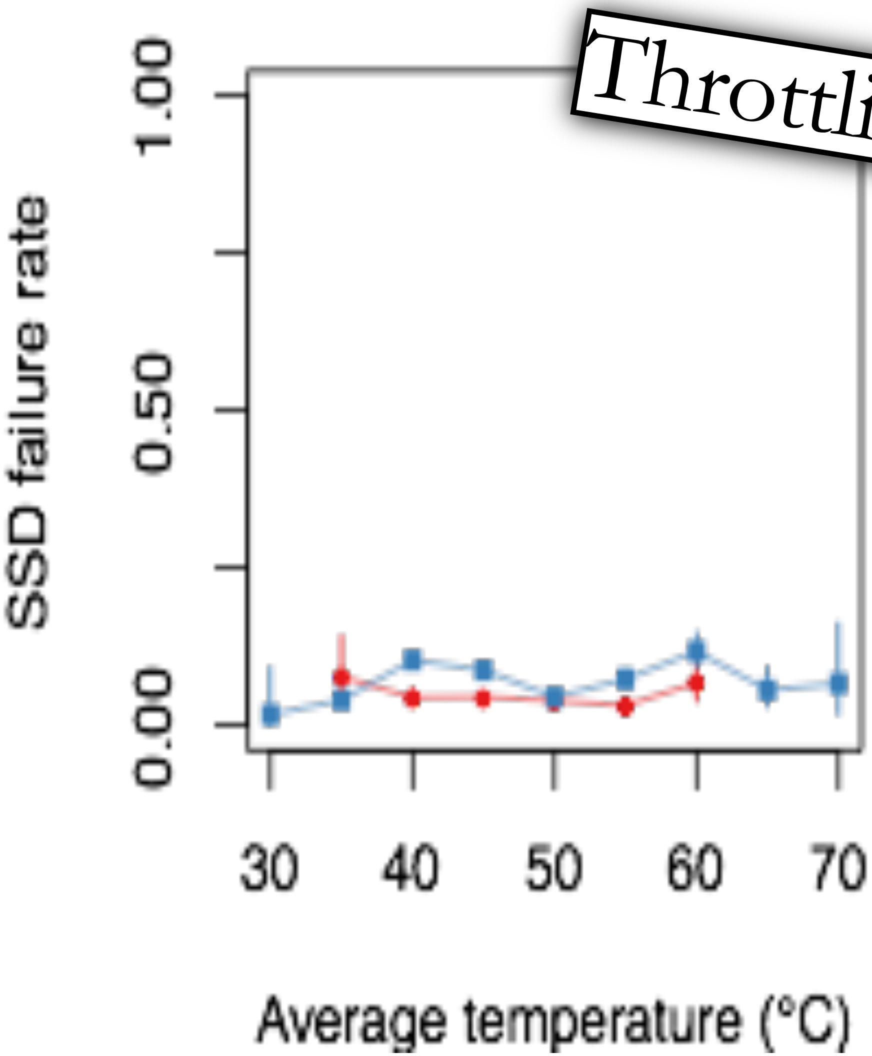
720GB, 1 SSD 720GB, 2 SSDs



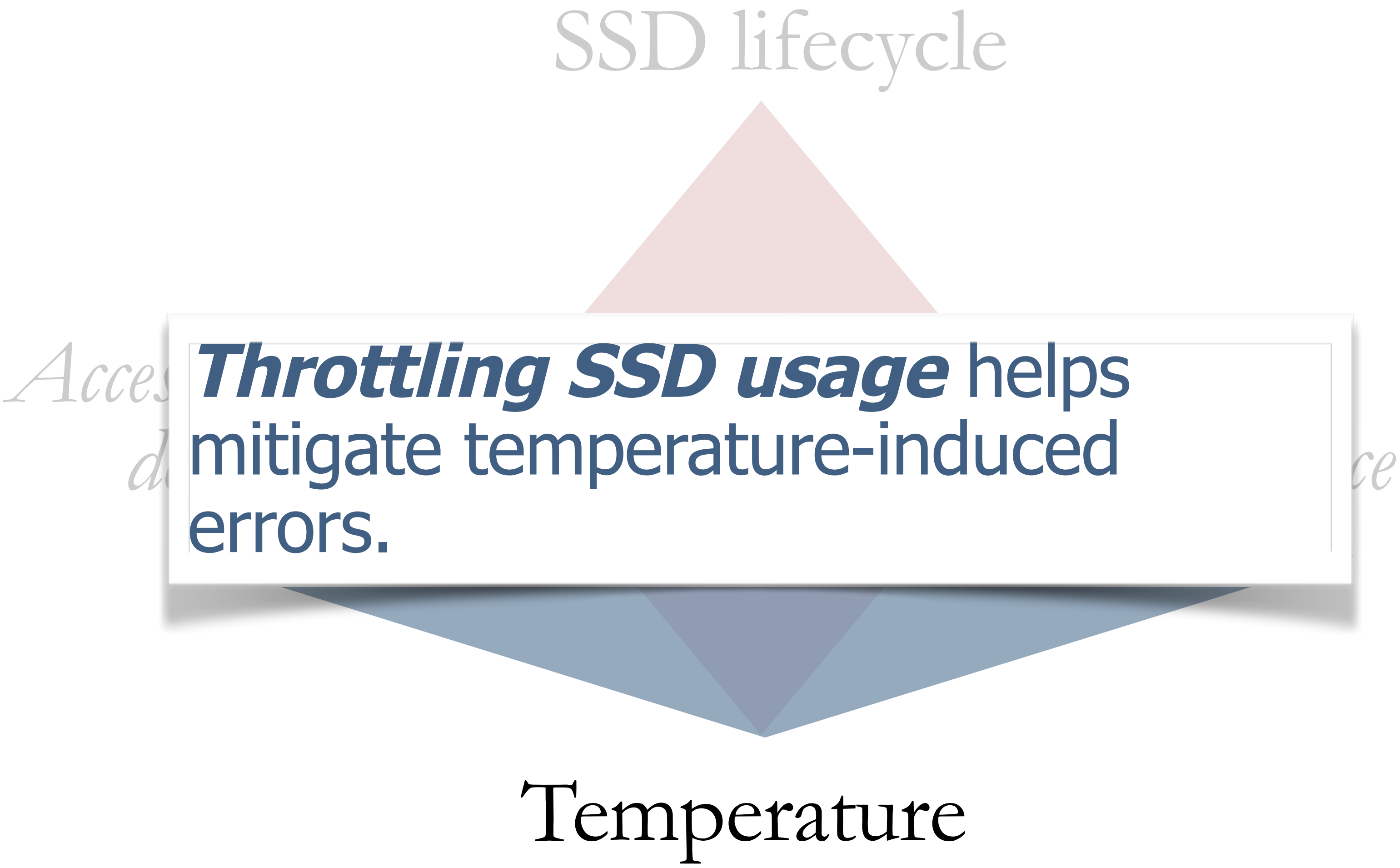
High temperature:
may **throttle** *or*
shut down



1.2TB, 1 SSD 3.2TB, 1 SSD

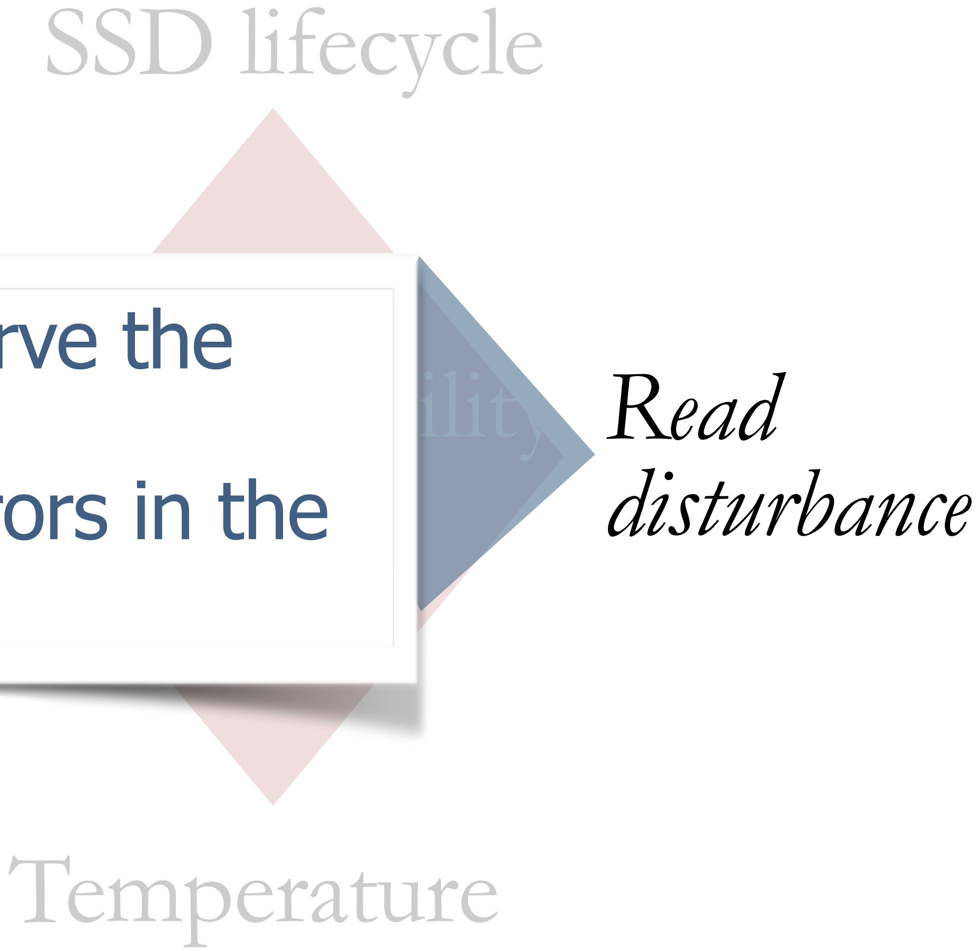


Throttling



Throttling SSD usage helps mitigate temperature-induced errors.

Summary



SSD lifecycle

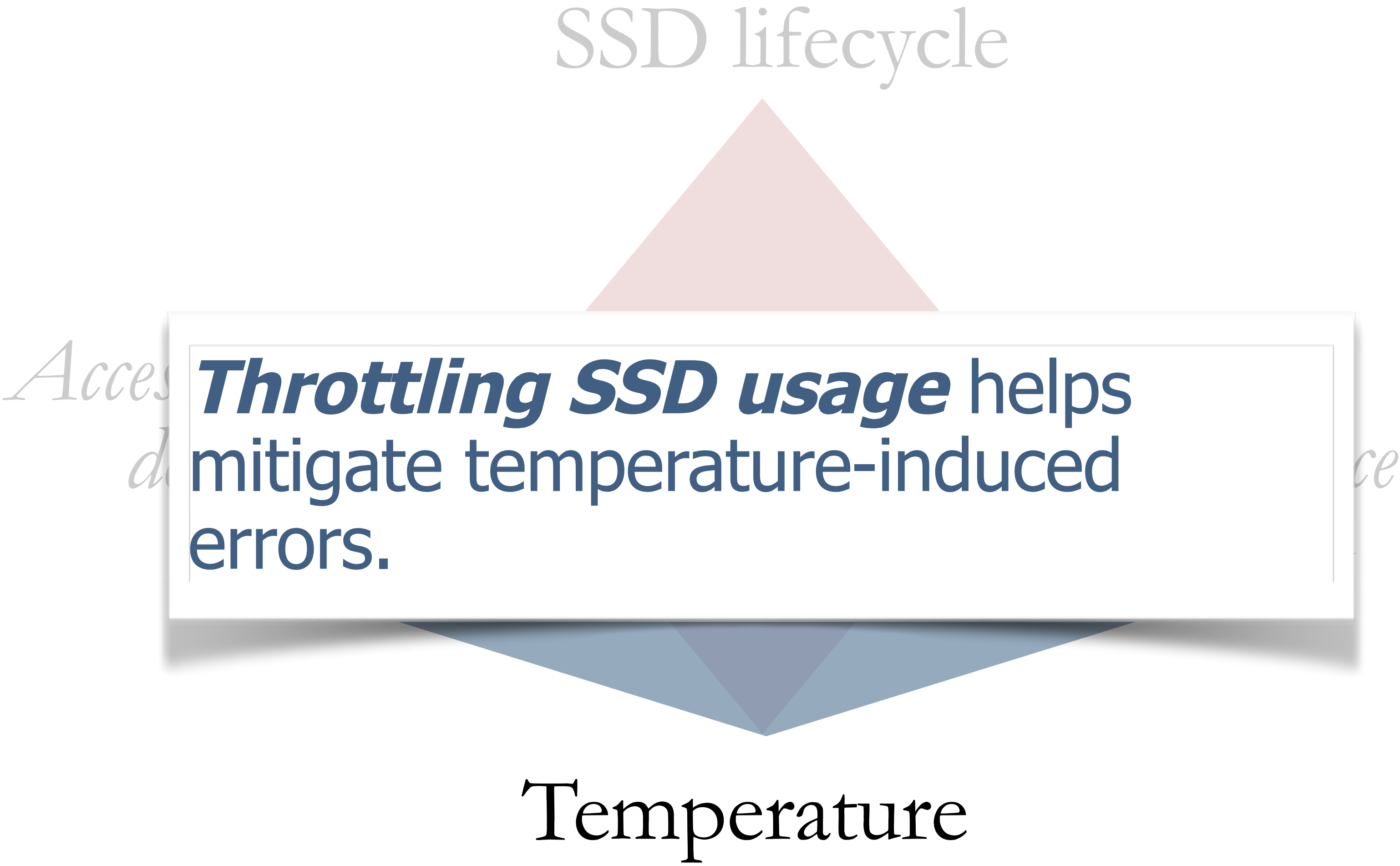
ility

Read disturbance

Temperature

We ***do not*** observe the effects of ***read disturbance*** errors in the field.

Summary



Throttling SSD usage helps mitigate temperature-induced errors.

Summary

SSD lifecycle

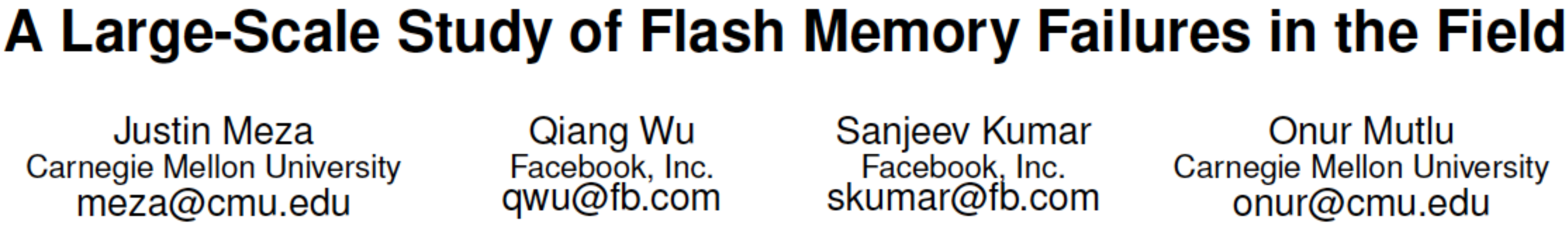
Access pattern dependence

We quantify the effects of the ***page cache*** and ***write amplification*** in the field.

Temperature

More on SSD Error Analysis in the Field

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

Sanjeev Kumar
Facebook, Inc.
skumar@fb.com

Onur Mutlu
Carnegie Mellon University
onur@cmu.edu

NAND Flash Memory Readings

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.

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

Error Analysis and Management for MLC NAND Flash Memory

Onur Mutlu
onur@cmu.edu

(joint work with Yu Cai, Gulay Yalcin, Erich Haratsch, Ken Mai, Adrian Cristal, Osman Unsal)

August 7, 2014

Flash Memory Summit 2014, Santa Clara, CA

Executive Summary

- Problem: MLC NAND flash memory reliability/endurance is a key challenge for satisfying future storage systems' requirements

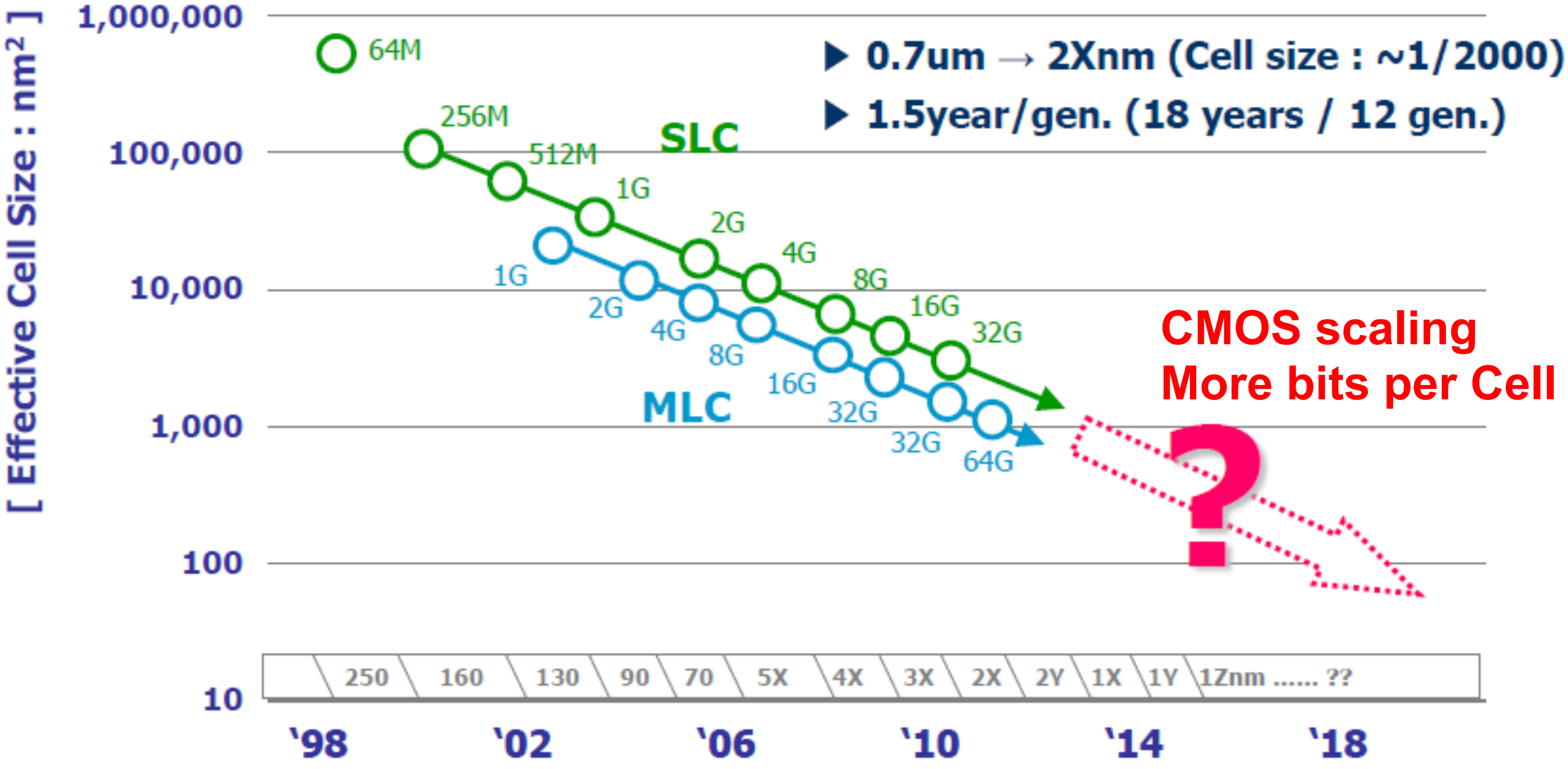
- Our Goals: (1) Build reliable error models for NAND flash memory via experimental characterization, (2) Develop efficient techniques to improve reliability and endurance

- This talk provides a "flash" summary of our recent results published in the past 3 years:
    - Experimental error and threshold voltage characterization **[DATE'12&13]**
    - Retention-aware error management **[ICCD'12]**
    - Program interference analysis and read reference V prediction **[ICCD'13]**
    - Neighbor-assisted error correction **[SIGMETRICS'14]**

Agenda

- Background, Motivation and Approach
- Experimental Characterization Methodology
- Error Analysis and Management
  - Characterization Results
  - Retention-Aware Error Management
  - Threshold Voltage and Program Interference Analysis
  - Read Reference Voltage Prediction
  - Neighbor-Assisted Error Correction
- Summary

Evolution of NAND Flash Memory



▶ 0.7um → 2Xnm (Cell size : ~1/2000)

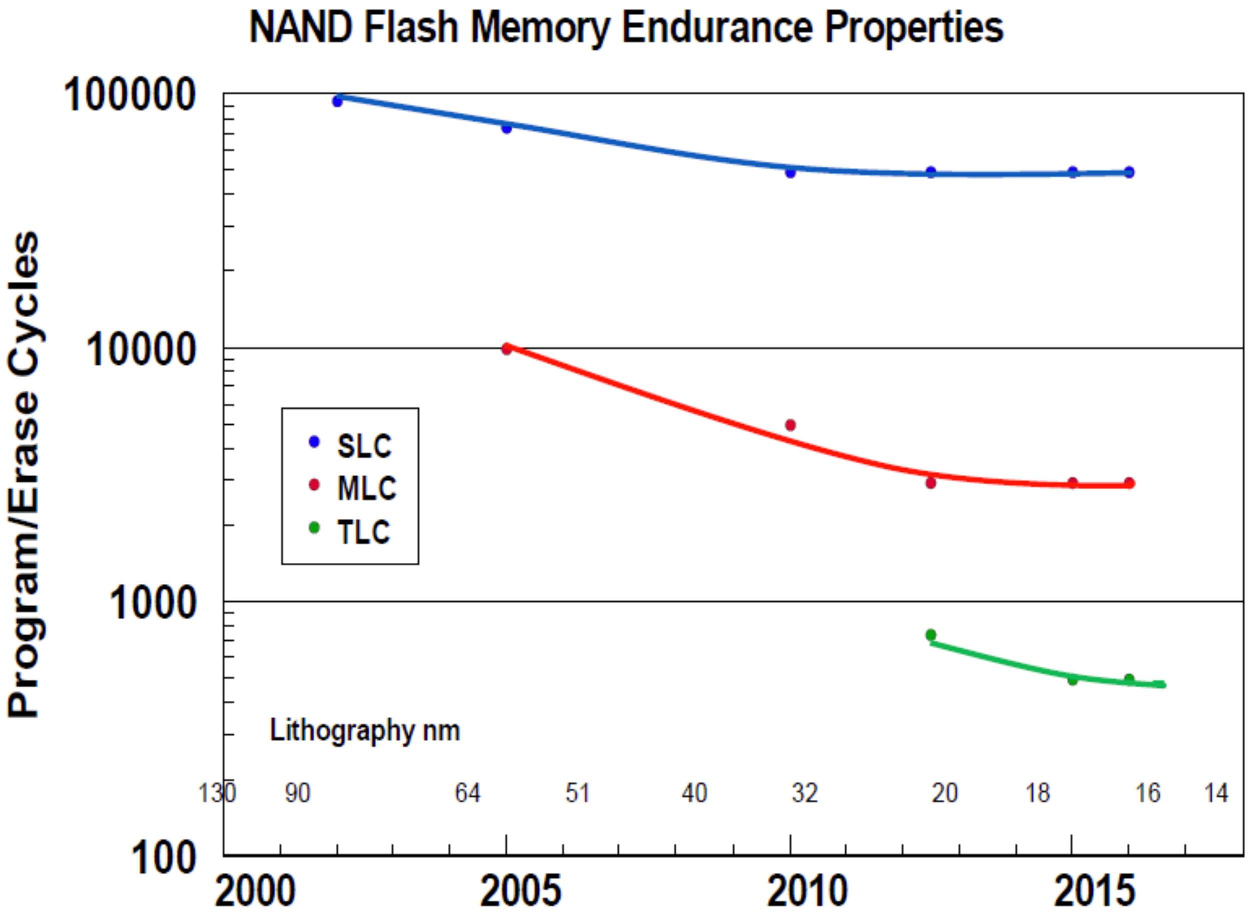
▶ 1.5year/gen. (18 years / 12 gen.)

CMOS scaling
More bits per Cell

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



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

- **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

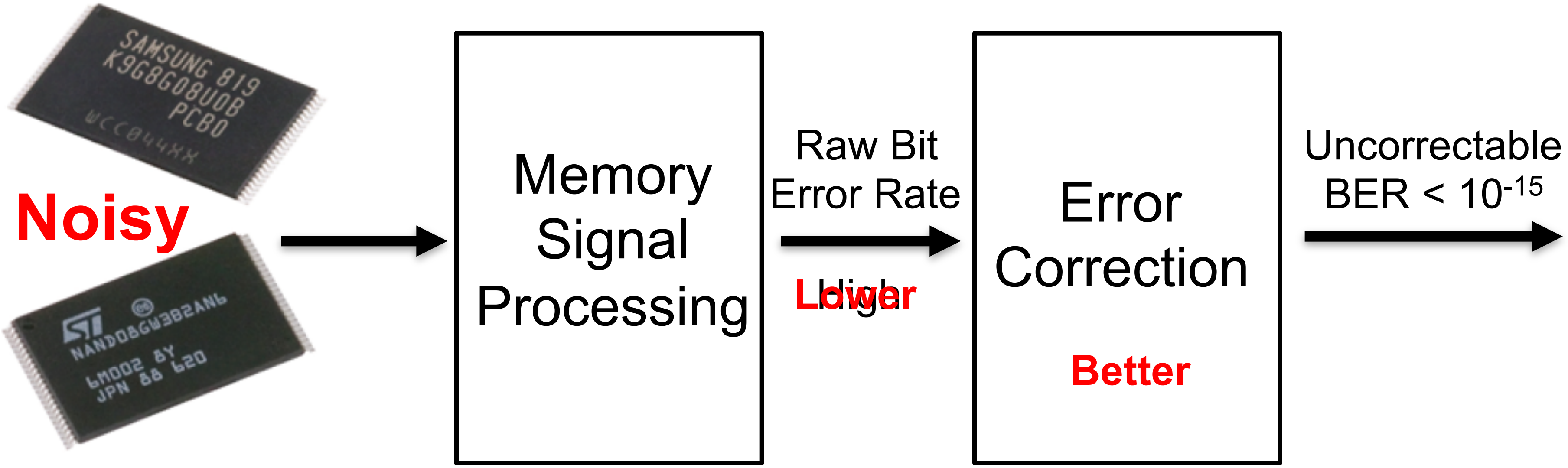
NAND Flash Memory is Increasingly Noisy

Write

Noisy NAND

Read

Future NAND Flash-based Storage Architecture



Our Goals:

Build reliable error models for NAND flash memory

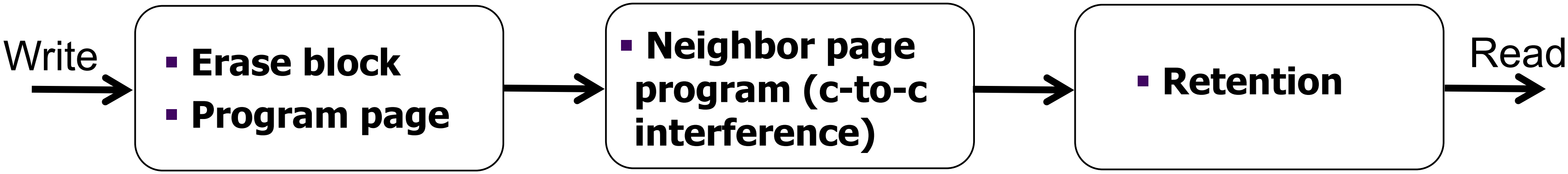
Design efficient reliability mechanisms based on the model

NAND Flash Error Model

Write
Noisy NAND
Read

Experimentally characterize and model dominant errors

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


Write

- **Erase block**
- **Program page**

- **Neighbor page program (c-to-c interference)**

- **Retention**

Read

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

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., “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**

Our Goals and Approach

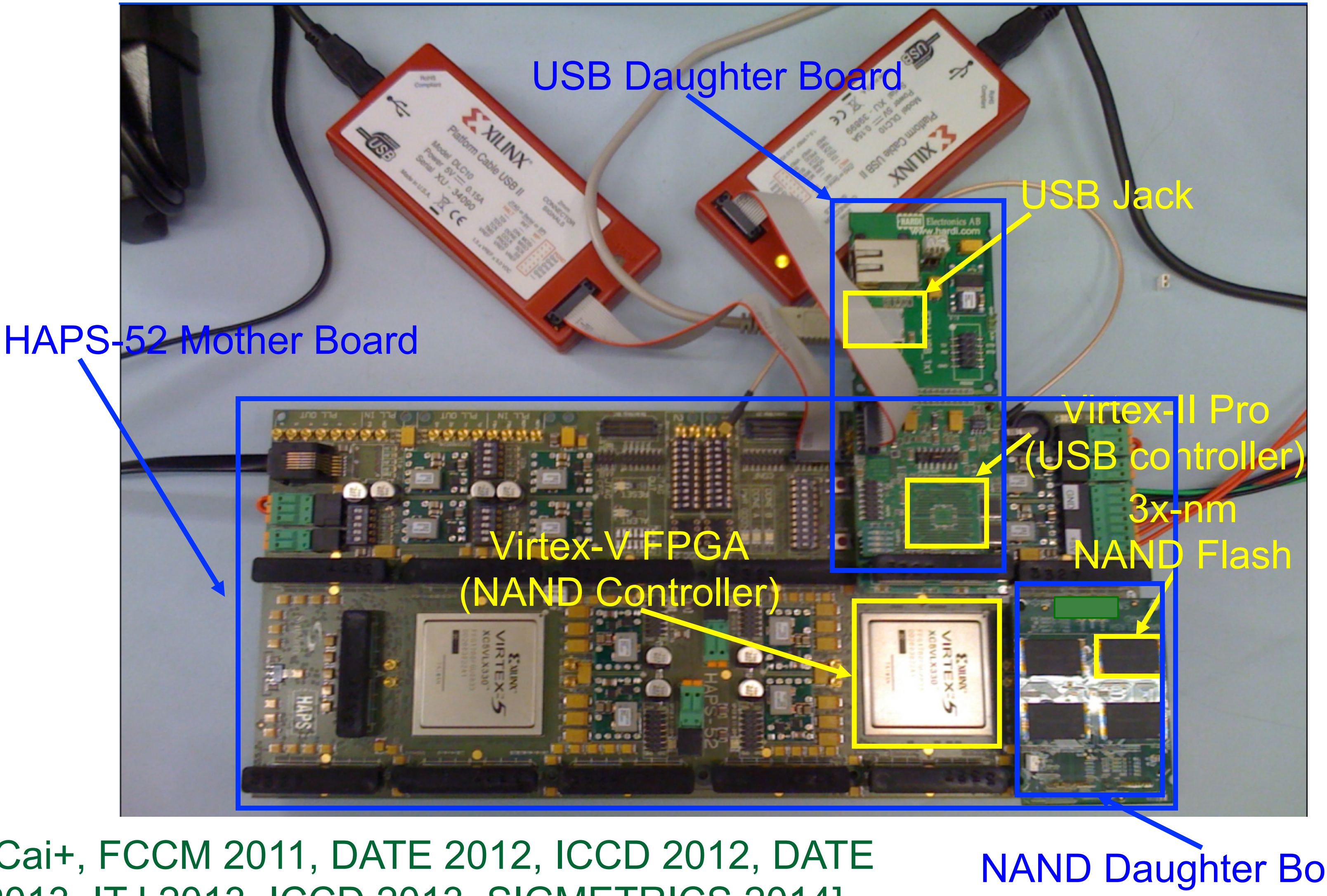
- Goals:
  - Understand error mechanisms and develop reliable predictive models for MLC NAND flash memory errors
  - Develop efficient error management techniques to mitigate errors and improve flash reliability and endurance

- 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

Agenda

- Background, Motivation and Approach
- Experimental Characterization Methodology
- Error Analysis and Management
    - Main Characterization Results
    - Retention-Aware Error Management
    - Threshold Voltage and Program Interference Analysis
    - Read Reference Voltage Prediction
    - Neighbor-Assisted Error Correction
- Summary

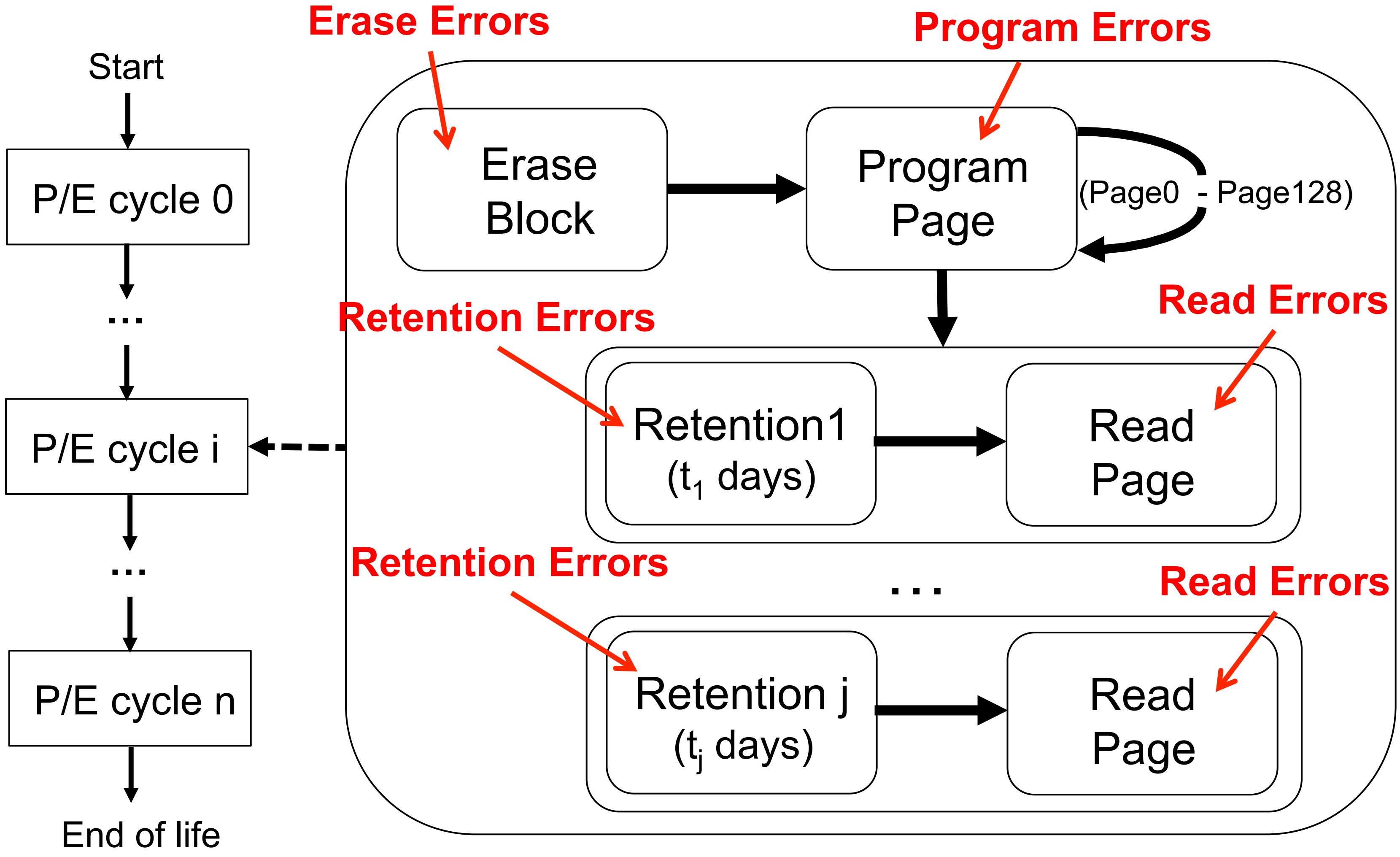
Experimental Testing Platform



[Cai+, FCCM 2011, DATE 2012, ICCD 2012, DATE 2013, ITJ 2013, ICCD 2013, SIGMETRICS 2014]

Cai et al., FPGA-based Solid-State Drive prototyping platform, FCCM 2011.

NAND Flash Usage and Error Model



Methodology: Error and ECC Analysis

- Characterized errors and error rates of 3x and 2y-nm MLC NAND flash using an experimental FPGA-based platform
  - **[Cai+, DATE'12, ICCD'12, DATE'13, ITJ'13, ICCD'13, SIGMETRICS'14]**

- Quantified Raw Bit Error Rate (RBER) at a given P/E cycle
  - Raw Bit Error Rate: Fraction of erroneous bits without any correction

- Quantified error correction capability (and area and power consumption) of various BCH-code implementations
  - Identified how much RBER each code can tolerate

    → how many P/E cycles (flash lifetime) each code can sustain

NAND Flash Error Types

- Four types of errors [Cai+, DATE 2012]

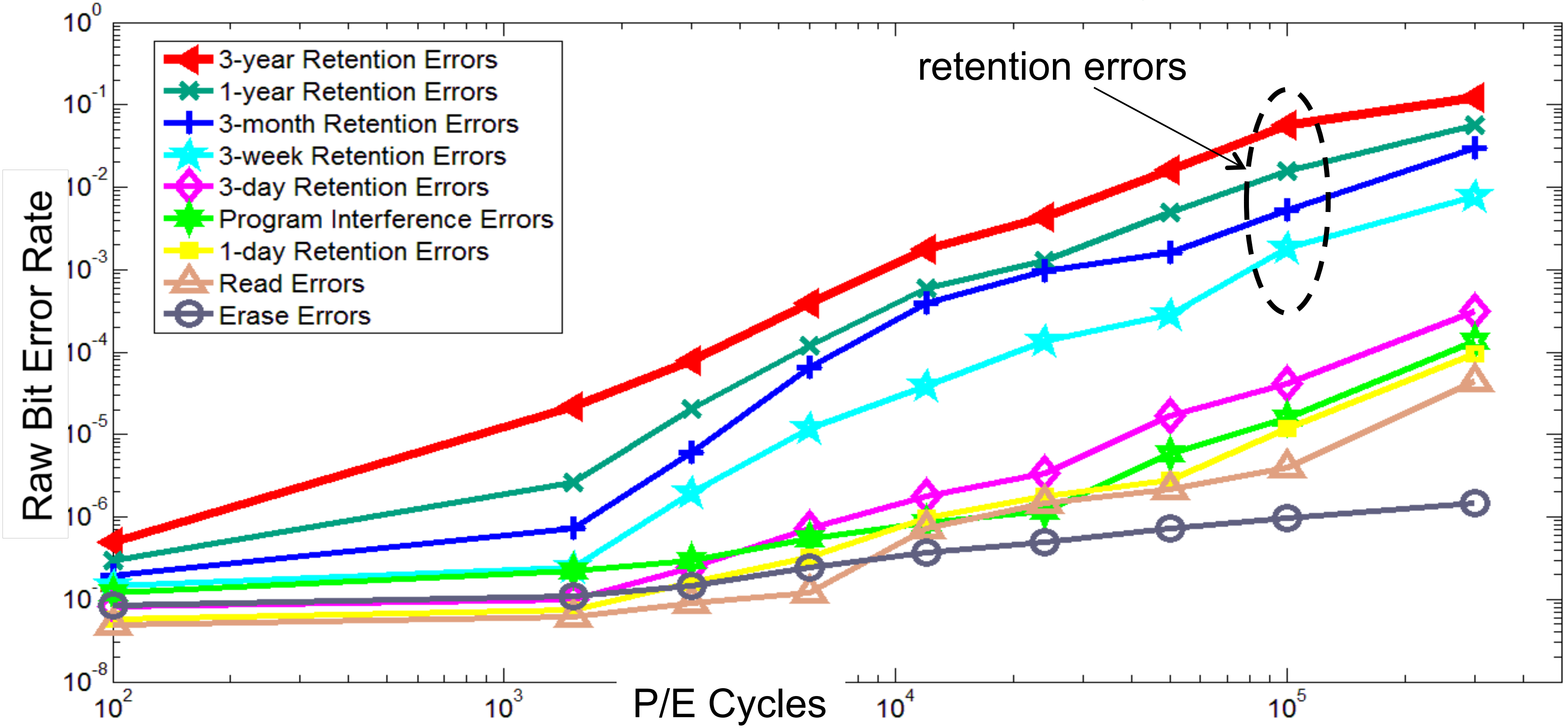
- Caused by common flash operations
  - Read errors
  - Erase errors
  - Program (interference) errors

- 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

Agenda

- Background, Motivation and Approach
- Experimental Characterization Methodology
- Error Analysis and Management
  - Main Characterization Results
  - Retention-Aware Error Management
  - Threshold Voltage and Program Interference Analysis
  - Read Reference Voltage Prediction
  - Neighbor-Assisted Error Correction
- Summary

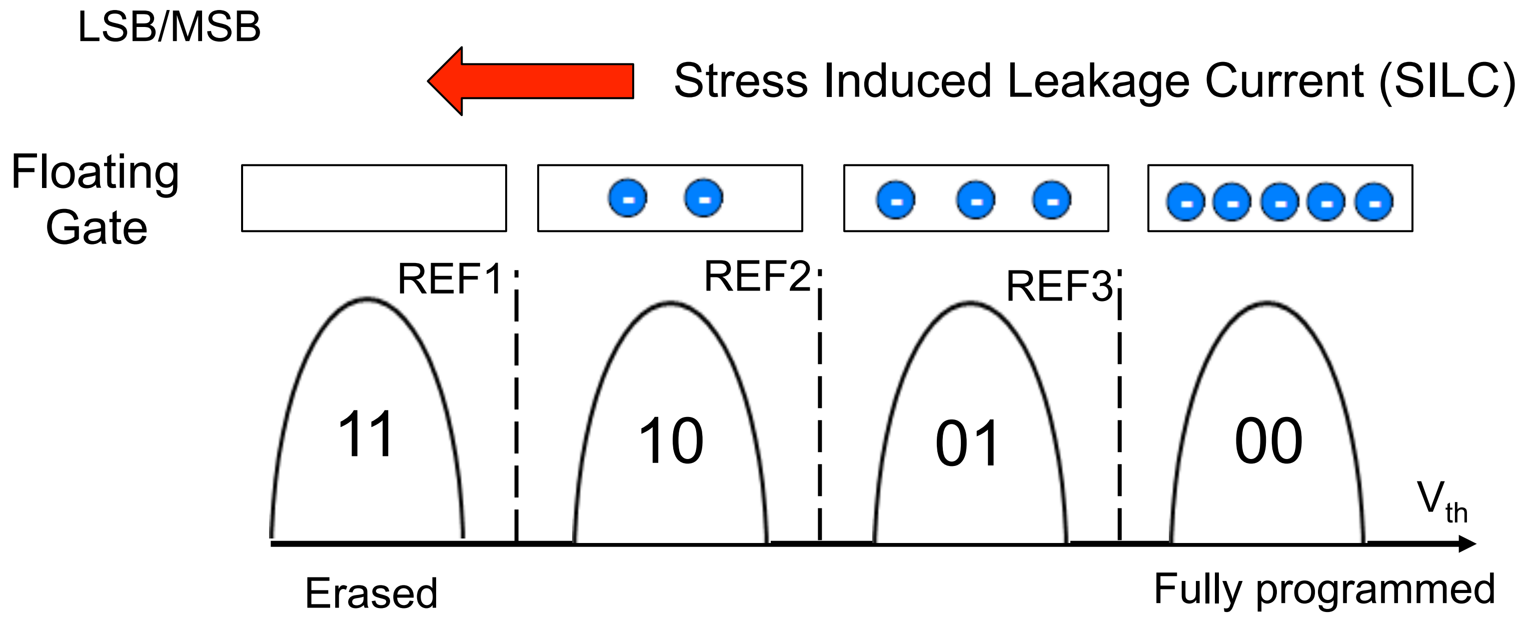
Observations: Flash Error Analysis



- 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

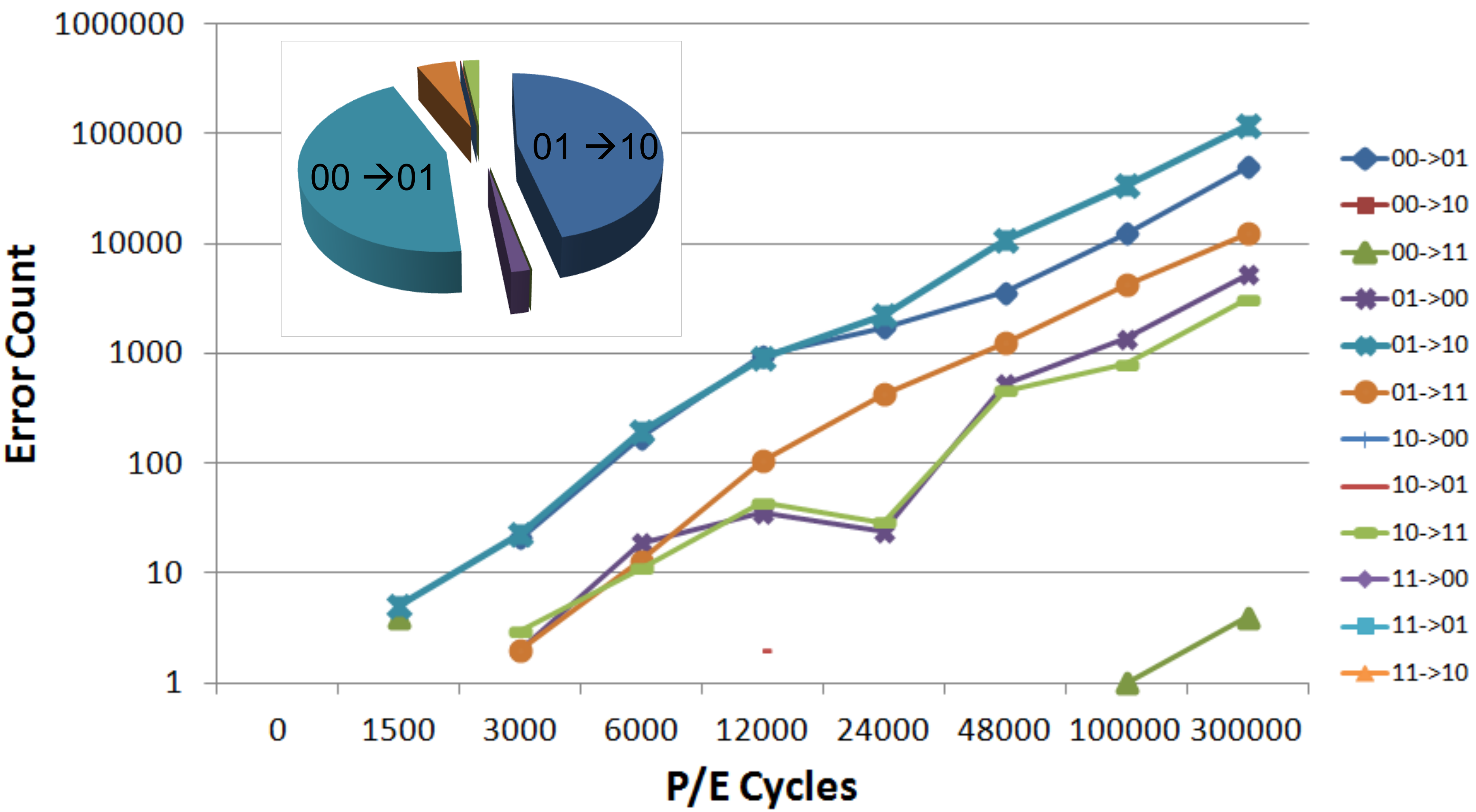
SAFARI Cai et al., Error Patterns in MLC NAND Flash Memory, DATE 2012.

Retention Error Mechanism



- Electron loss from the floating gate causes retention errors
  - Cells with more programmed electrons suffer more from retention errors
  - Threshold voltage is more likely to shift by one window than by multiple

Retention Error Value Dependency



- Cells with more programmed electrons tend to suffer more from retention noise (i.e. 00 and 01)

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)

Agenda

- Background, Motivation and Approach
- Experimental Characterization Methodology
- Error Analysis and Management
  - Main Characterization Results
  - Retention-Aware Error Management
  - Threshold Voltage and Program Interference Analysis
  - Read Reference Voltage Prediction
  - Neighbor-Assisted Error Correction
- Summary

Flash Correct-and-Refresh (FCR)

- Key Observations:
  - Retention errors are the dominant source of errors in flash memory **[Cai+ DATE 2012][Tanakamaru+ ISSCC 2011]**
    → limit flash lifetime as they increase over time
  - Retention errors can be corrected by "refreshing" each flash page periodically

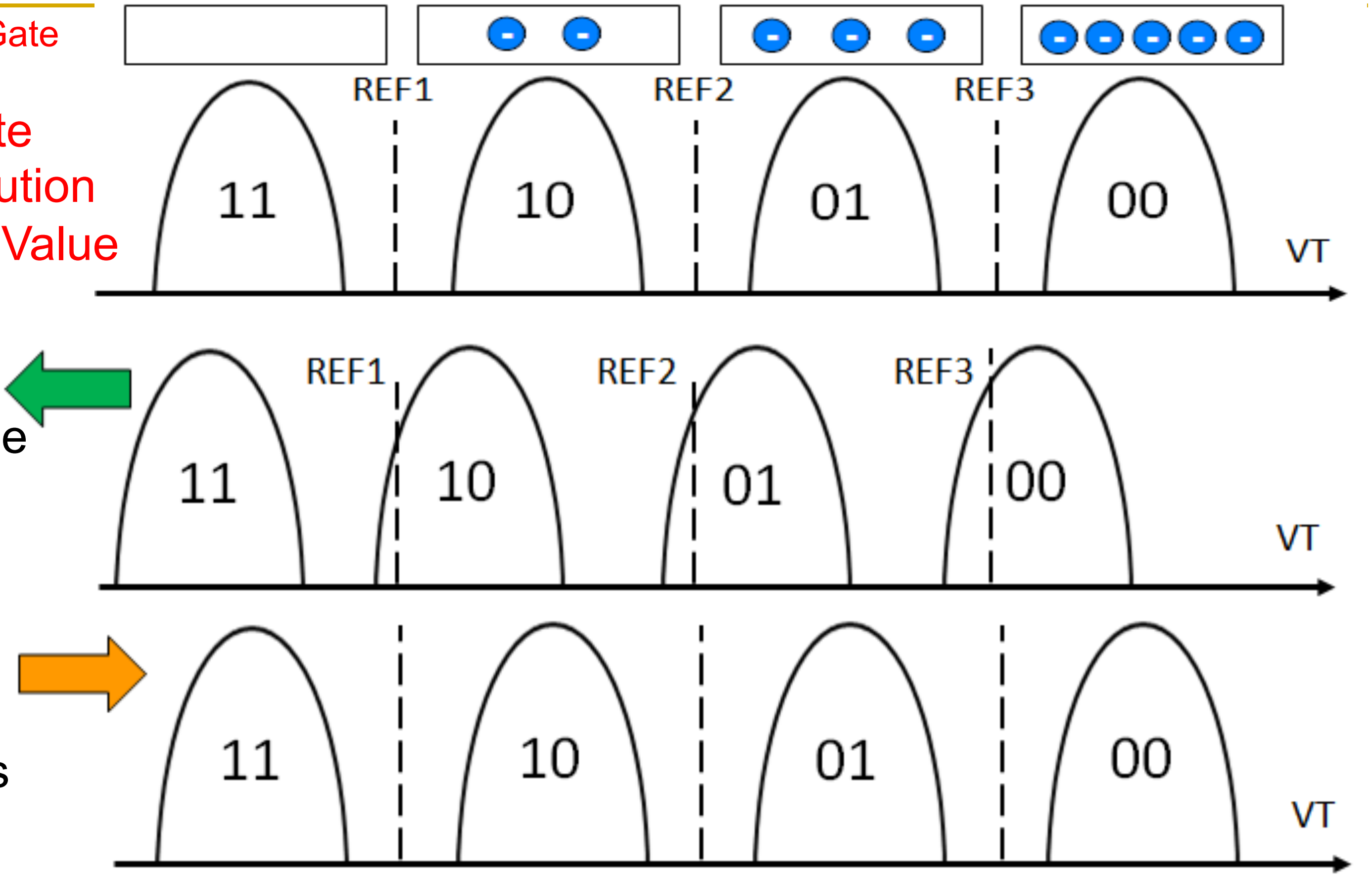
- Key Idea:
  - Periodically read each flash page,
  - Correct its errors using "weak" ECC, and
  - Either remap it to a new physical page or reprogram it in-place,
  - Before the page accumulates more errors than ECC-correctable
  - Optimization: Adapt refresh rate to endured P/E cycles

Cai et al., Flash Correct and Refresh, ICCD 2012.

FCR: Two Key Questions

- How to refresh?
  - Remap a page to another one
  - Reprogram a page (in-place)
  - Hybrid of remap and reprogram
- When to refresh?
  - Fixed period
  - Adapt the period to retention error severity

In-Place Reprogramming of Flash Cells



Floating Gate

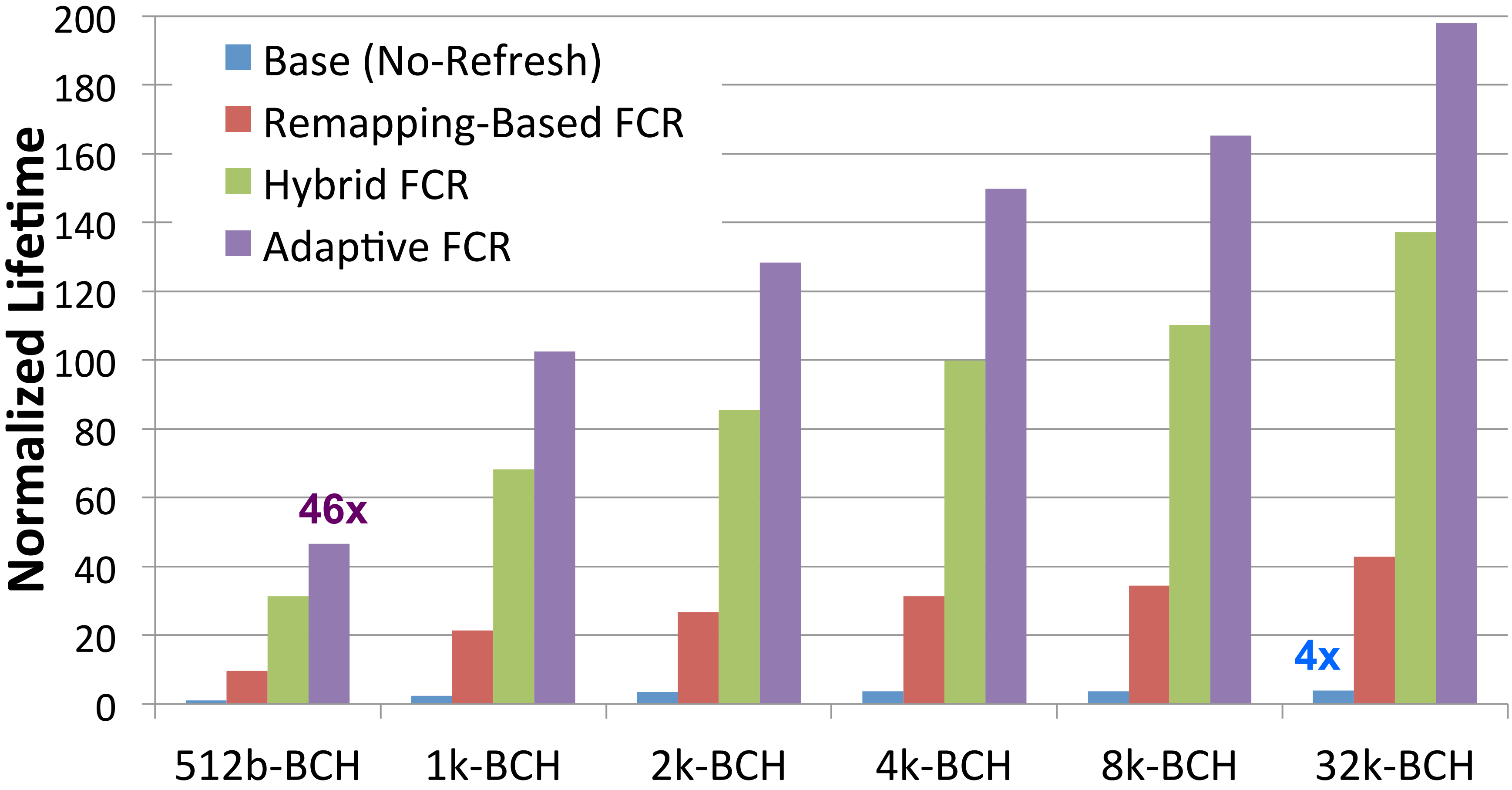
Floating Gate Voltage Distribution for each Stored Value

Retention errors are caused by cell voltage shifting to the left

ISPP moves cell voltage to the right; fixes retention errors

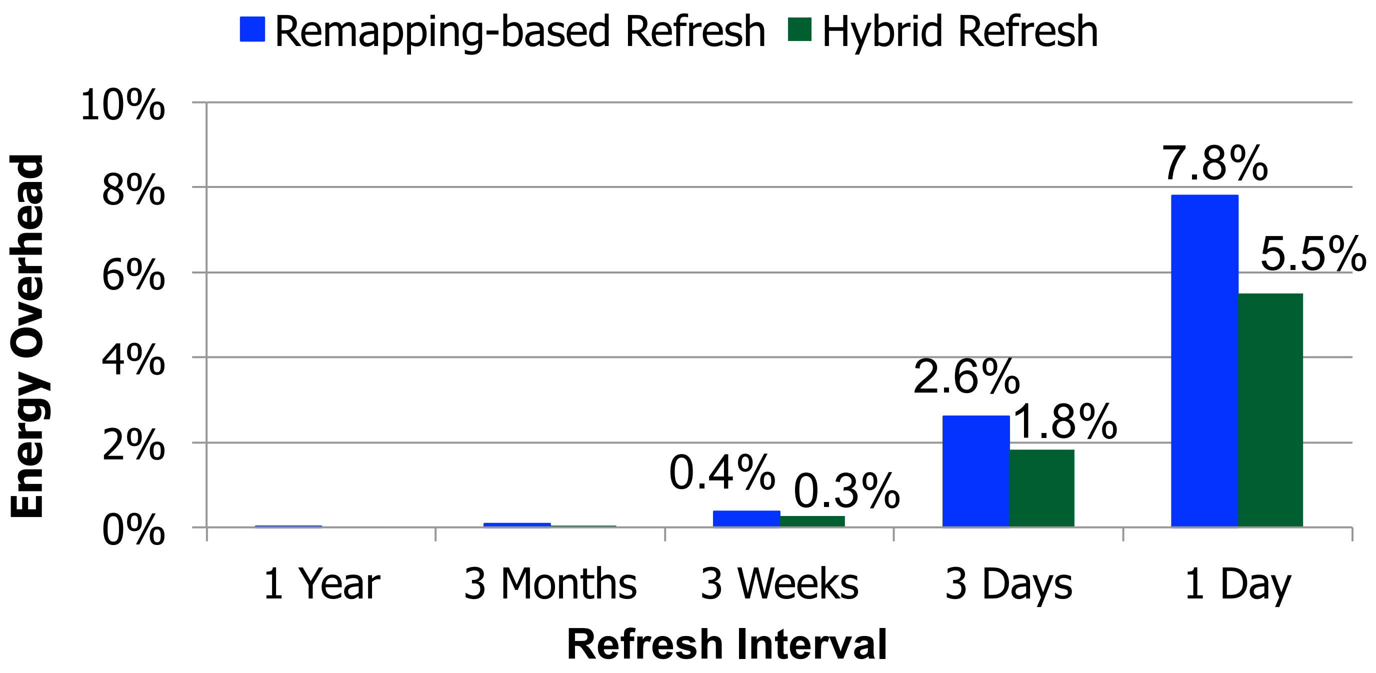
- Pro: No remapping needed → no additional erase operations
- Con: Increases the occurrence of program errors

Normalized Flash Memory Lifetime



Lifetime of FCR much higher than lifetime of stronger ECC

Energy Overhead



- Adaptive-rate refresh: <1.8% energy increase until daily refresh is triggered

More Detail and Analysis on FCR

- 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"** *Proceedings of the 30th IEEE International Conference on Computer Design (**ICCD**)*, Montreal, Quebec, Canada, September 2012. Slides (ppt) (pdf)

Agenda

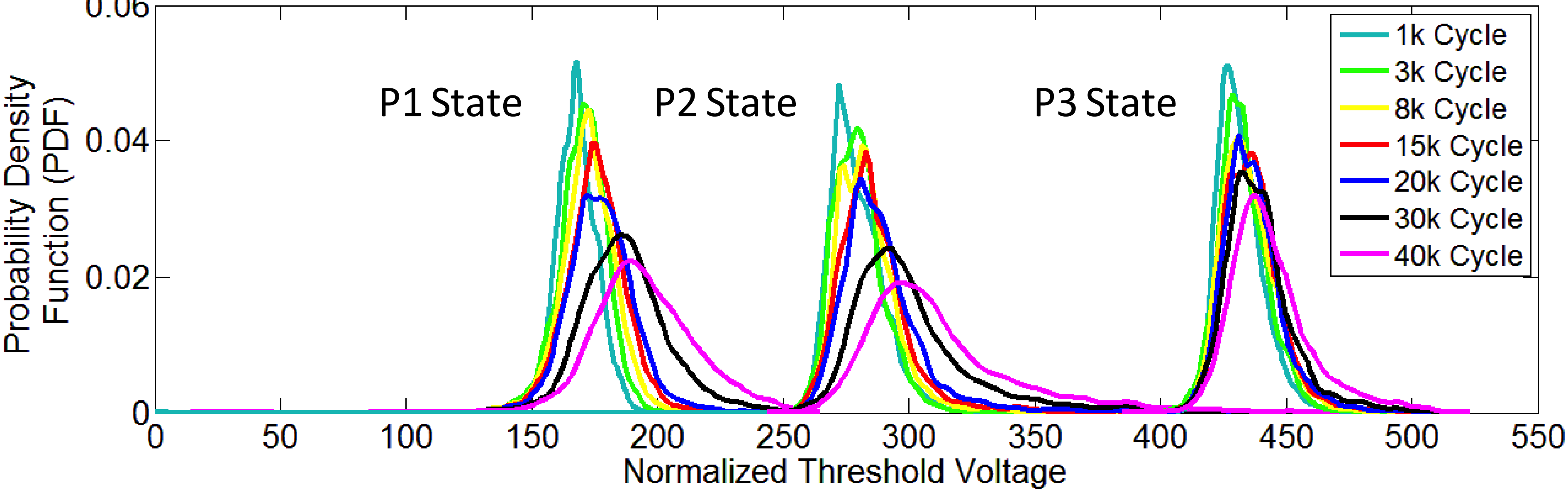
- Background, Motivation and Approach
- Experimental Characterization Methodology
- Error Analysis and Management
    - Main Characterization Results
    - Retention-Aware Error Management
    - Threshold Voltage and Program Interference Analysis
    - Read Reference Voltage Prediction
    - Neighbor-Assisted Error Correction
- Summary

Key Questions

- How does threshold voltage (Vth) distribution of different programmed states change over flash lifetime?
- Can we model it accurately and predict the Vth changes?
- Can we build mechanisms that can correct for Vth changes? (thereby reducing read error rates)

Threshold Voltage Distribution Model

Characterized on 2Y-nm chips using the read-retry feature



Gaussian distribution with additive white noise

As P/E cycles increase ...

- Distribution shifts to the right
- Distribution becomes wider

Cai et al., Threshold Voltage Distribution in MLC NAND Flash Memory, DATE 2013.

Threshold Voltage Distribution Model

- Vth distribution can be modeled with ~95% accuracy as a Gaussian distribution with additive white noise

- Distortion in Vth over P/E cycles can be modeled and predicted as an exponential function of P/E cycles
  - With more than 95% accuracy

More Detail on Threshold Voltage Model

- Yu Cai, Erich F. Haratsch, Onur Mutlu, and Ken Mai, **"Threshold Voltage Distribution in MLC NAND Flash Memory: Characterization, Analysis and Modeling"** *Proceedings of the Design, Automation, and Test in Europe Conference (**DATE**)*, Grenoble, France, March 2013. Slides (ppt)

Program Interference Errors

- When a cell is being programmed, voltage level of a neighboring cell changes (unintentionally) due to parasitic capacitance coupling

  → can change the data value stored

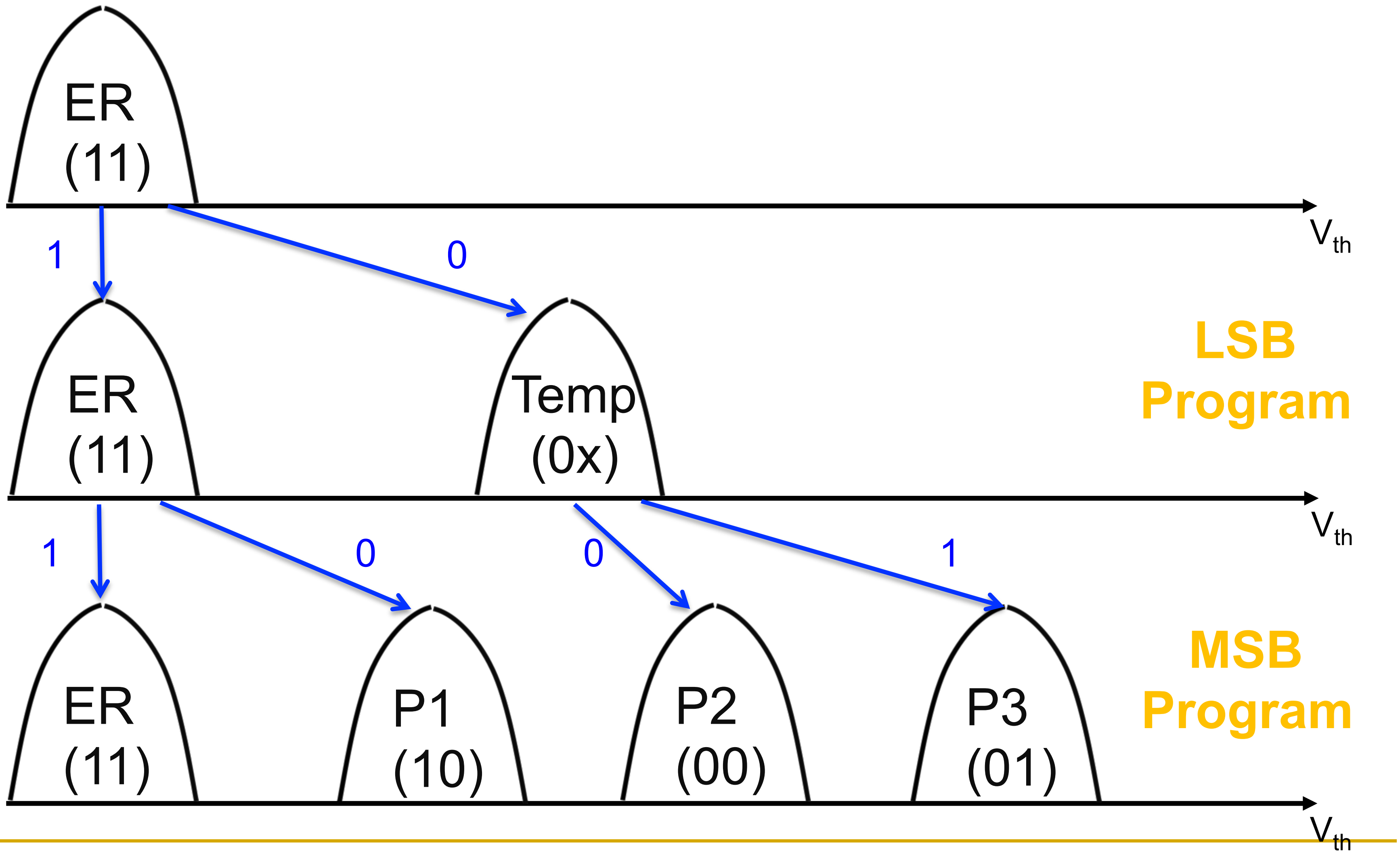
- Also called program interference error

- Causes neighboring cell voltage to increase (shift right)

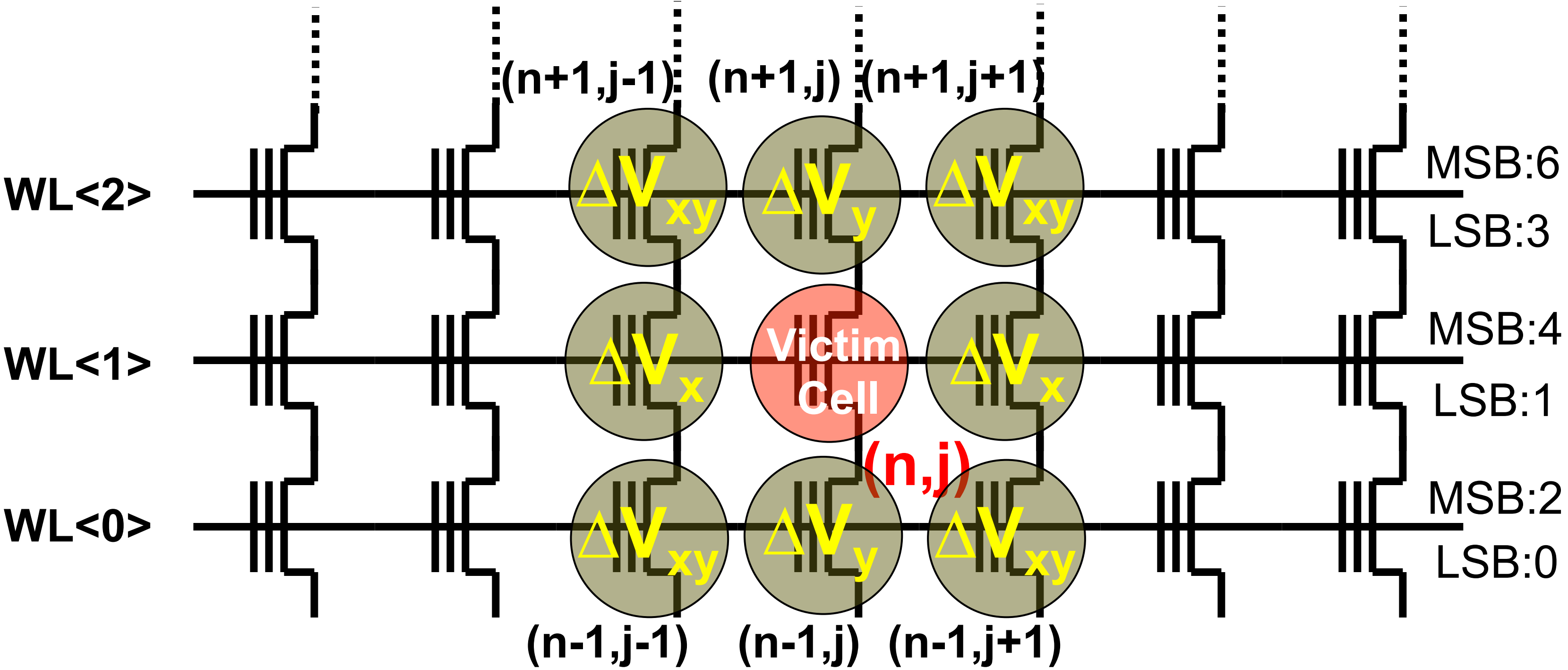
- Once retention errors are minimized, these errors can become dominant

How Current Flash Cells are Programmed

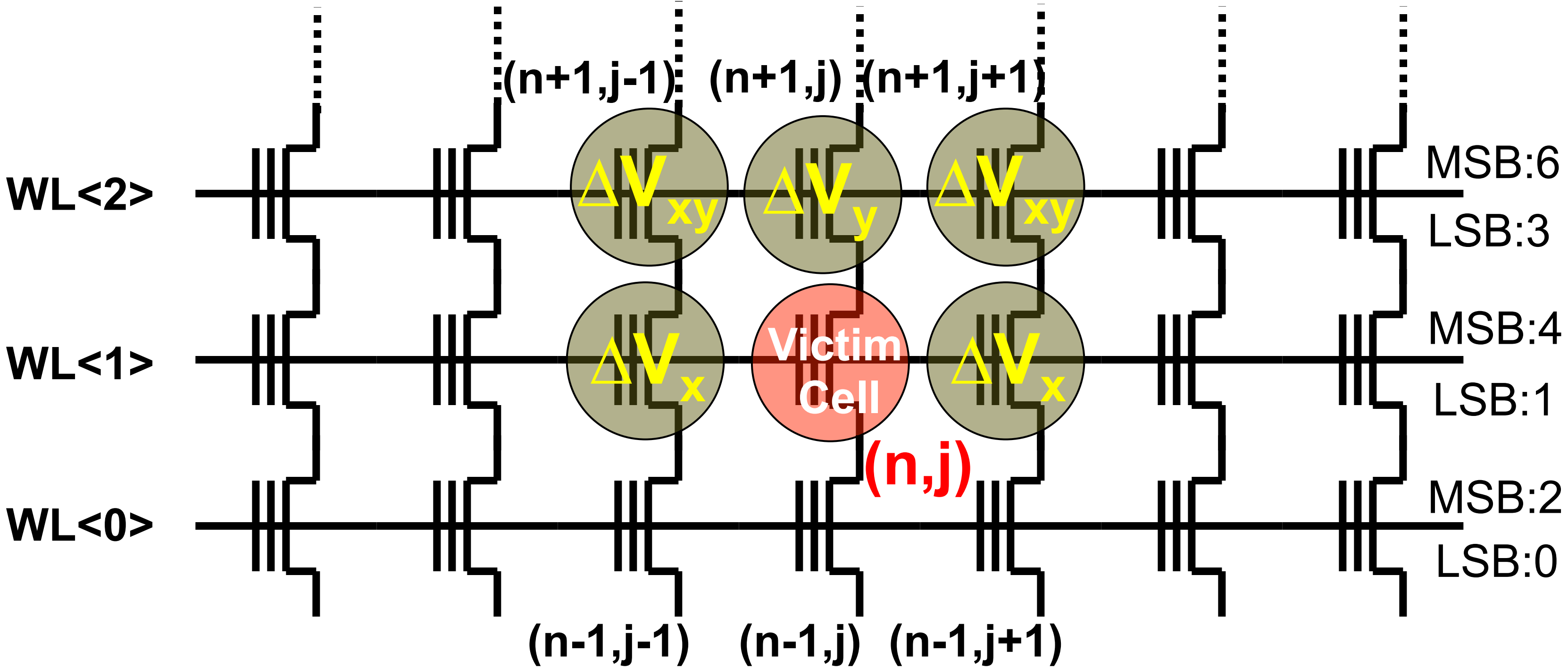
- Programming 2-bit MLC NAND flash memory in two steps



Basics of Program Interference



Traditional Model for Vth Change



- Traditional model for victim cell threshold voltage change

$$\Delta V_{victim} = (2C_x \Delta V_x + C_y \Delta V_y + 2C_{xy} \Delta V_{xy}) / C_{total}$$

Not accurate and requires knowledge of coupling caps!

Our Goal and Idea

- Develop a new, more accurate and easier to implement model for program interference

- Idea:
  - Empirically characterize and model the effect of neighbor cell Vth changes on the Vth of the victim cell
  - Fit neighbor Vth change to a linear regression model and find the coefficients of the model via empirical measurement

$$\Delta V_{victim}(n,j) = \sum_{y=j-K}^{j+K} \sum_{x=n+1}^{n=M} \alpha(x,y) \boxed{\Delta V_{neighbor}(x,y)} + \alpha_0 \boxed{V_{victim}^{before}(n,j)}$$

Can be measured

Developing a New Model via Empirical Measurement

- Feature extraction for V_{th} changes based on characterization
  - Threshold voltage changes on aggressor cell
  - Original state of victim cell

- Enhanced linear regression model

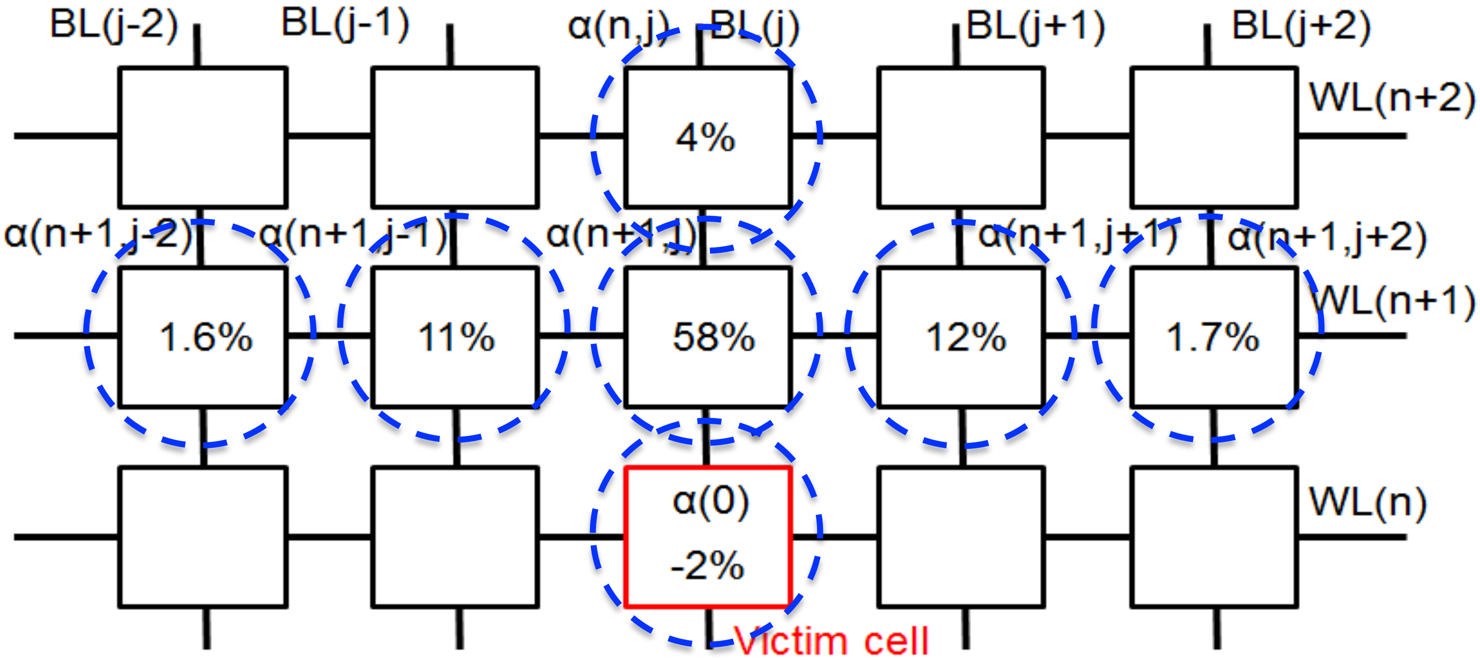
$$\Delta V_{victim}(n,j) = \sum_{y=j-K}^{j+K} \sum_{x=n+1}^{n=M} \alpha(x,y)\Delta V_{neighbor}(x,y) + \alpha_0 V_{victim}^{before}(n,j)$$

$$Y = X\alpha + \varepsilon \longleftarrow \text{(vector expression)}$$

- Maximum likelihood estimation of the model coefficients

$$\arg\min_{\alpha}(\|X \times \alpha - Y\|_2^2 + \lambda\|\alpha\|_1)$$

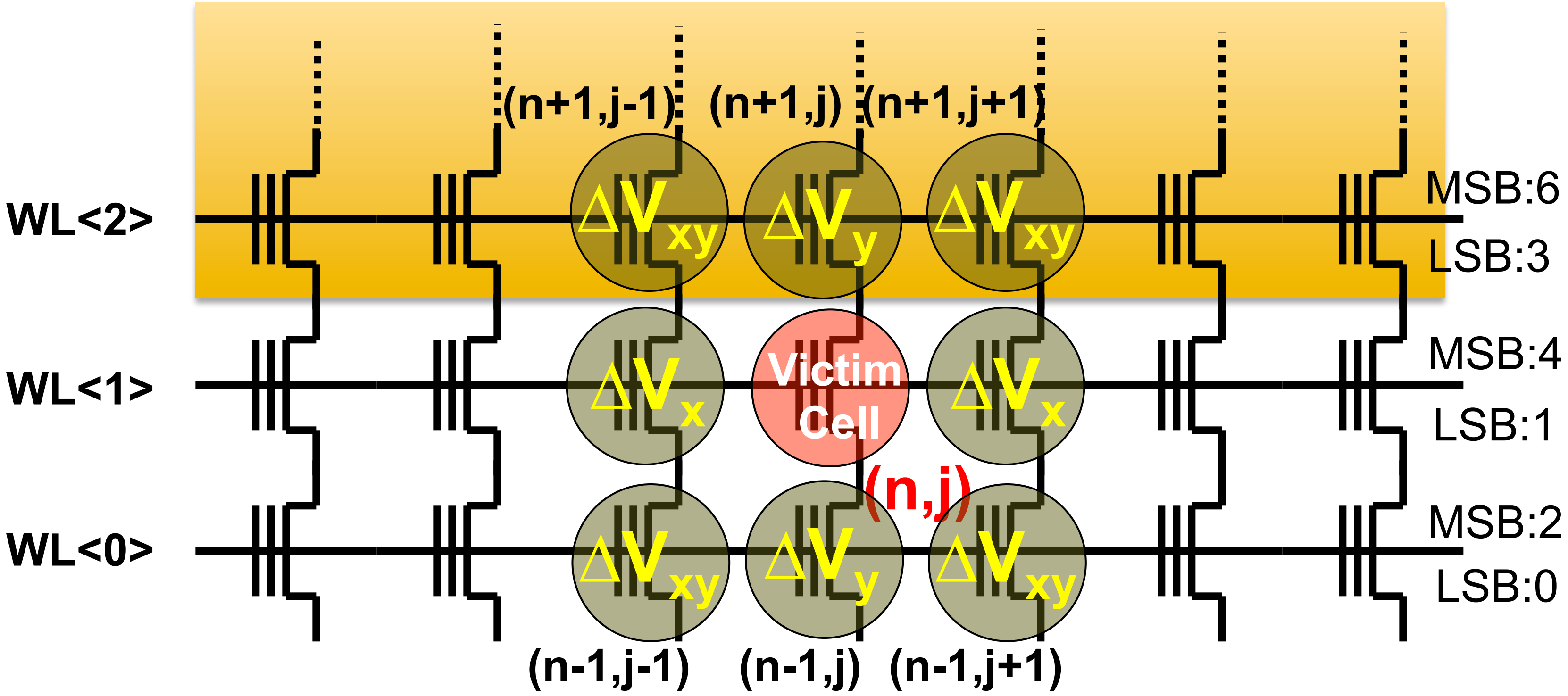
Effect of Neighbor Voltages on the Victim



- Immediately-above cell interference is dominant
- Immediately-diagonal neighbor is the second dominant
- Far neighbor cell interference exists
- Victim cell's Vth has negative effect on interference

Cai et al., Program Interference in MLC NAND Flash Memory, ICCD 2013

New Model for Program Interference

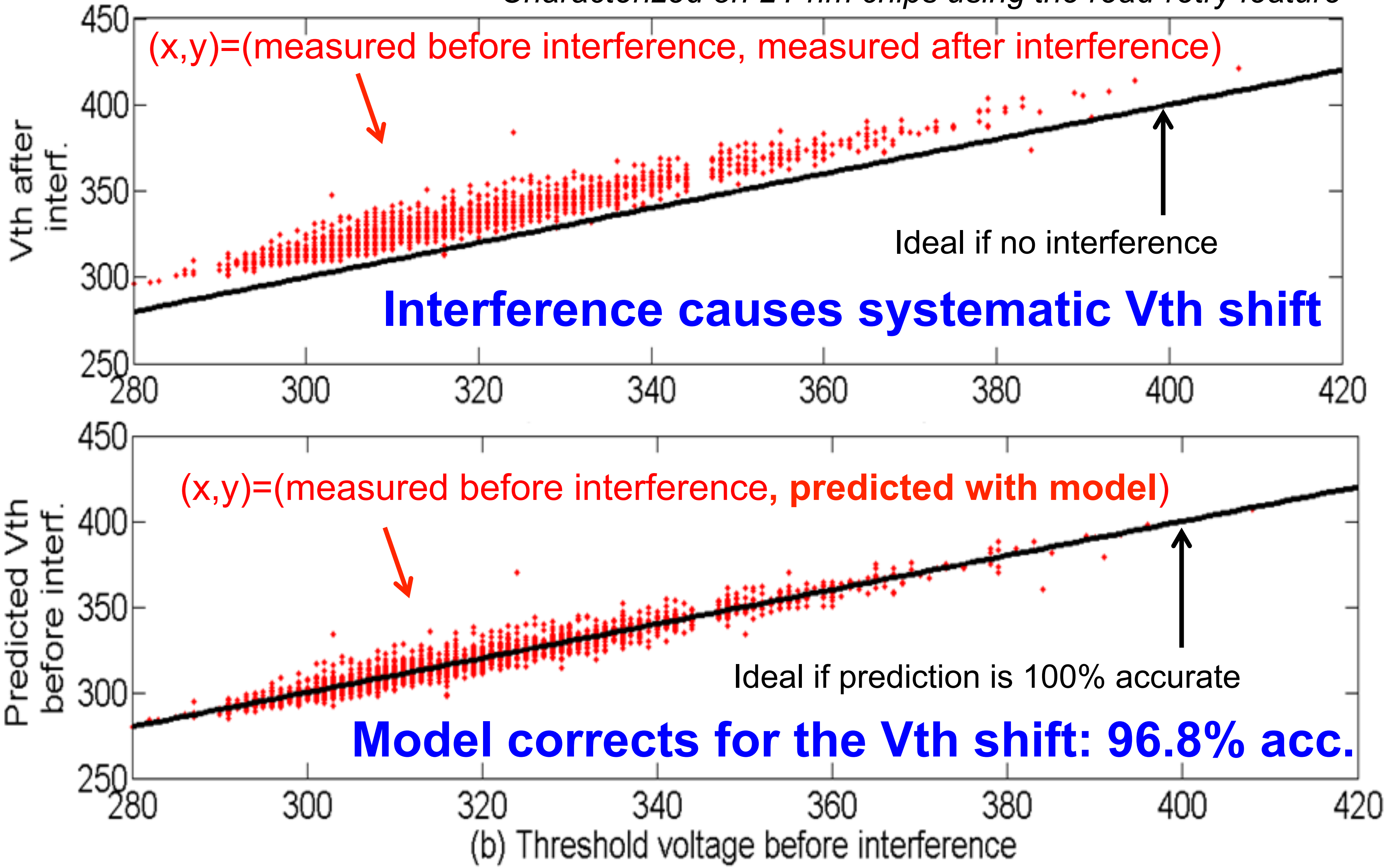


$$\Delta V_{victim}(n,j) = \sum_{y=j-K}^{j+K} \sum_{x=n+1}^{n+M} \alpha(x,y)\Delta V_{neighbor}(x,y) + \alpha_0 V_{victim}^{before}(n,j)$$

Cai et al., Program Interference in MLC NAND Flash Memory, ICCD 2013

Model Accuracy

Characterized on 2Y-nm chips using the read-retry feature



(x,y)=(measured before interference, measured after interference)

Ideal if no interference

Interference causes systematic Vth shift

(x,y)=(measured before interference, **predicted with model**)

Ideal if prediction is 100% accurate

Model corrects for the Vth shift: 96.8% acc.

Many Other Results in the Paper

- Yu Cai, Onur Mutlu, Erich F. Haratsch, and Ken Mai, **"Program Interference in MLC NAND Flash Memory: Characterization, Modeling, and Mitigation"** *Proceedings of the 31st IEEE International Conference on Computer Design* (***ICCD***), Asheville, NC, October 2013. Slides (pptx) (pdf) Lightning Session Slides (pdf)

Agenda

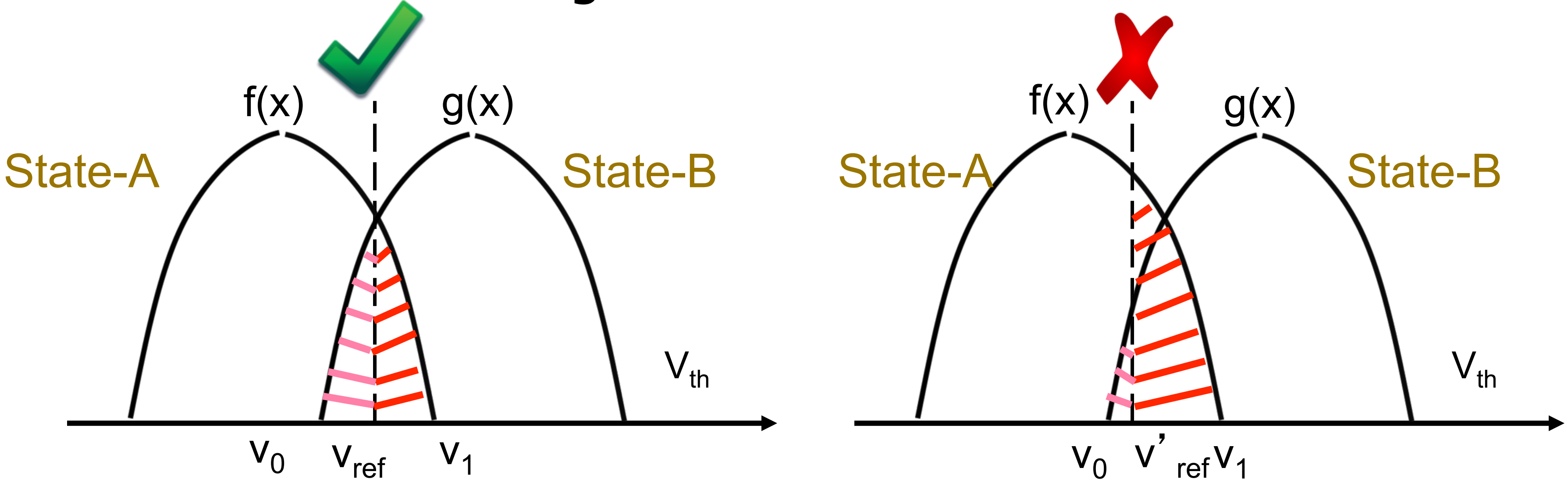
- Background, Motivation and Approach
- Experimental Characterization Methodology
- Error Analysis and Management
  - Main Characterization Results
  - Retention-Aware Error Management
  - Threshold Voltage and Program Interference Analysis
  - Read Reference Voltage Prediction
  - Neighbor-Assisted Error Correction
- Summary

Mitigation: Applying the Model

- So, what can we do with the model?
- Goal: Mitigate the effects of program interference caused voltage shifts

Optimum Read Reference for Flash Memory

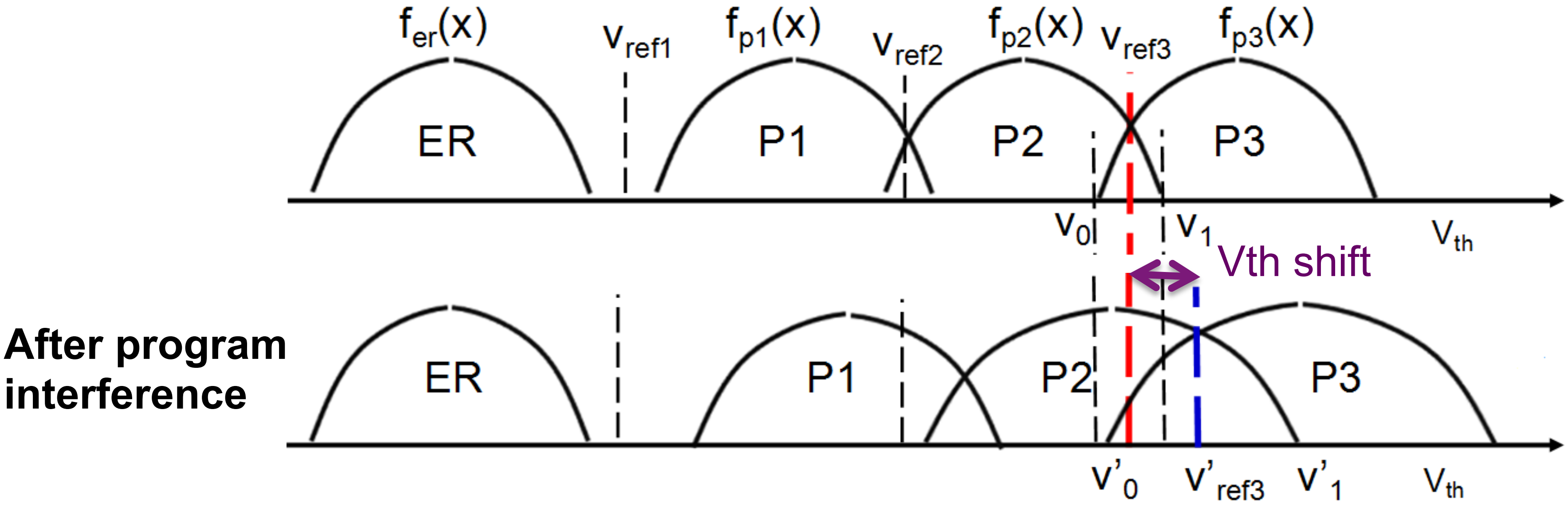
- Read reference voltage affects the raw bit error rate



$$BER1 = \int_{v_{ref}}^{+\infty} f(x)dx + \int_{-\infty}^{v_{ref}} g(x)dx \qquad BER2 = \int_{v'_{ref}}^{+\infty} f(x)dx + \int_{-\infty}^{v'_{ref}} g(x)dx$$

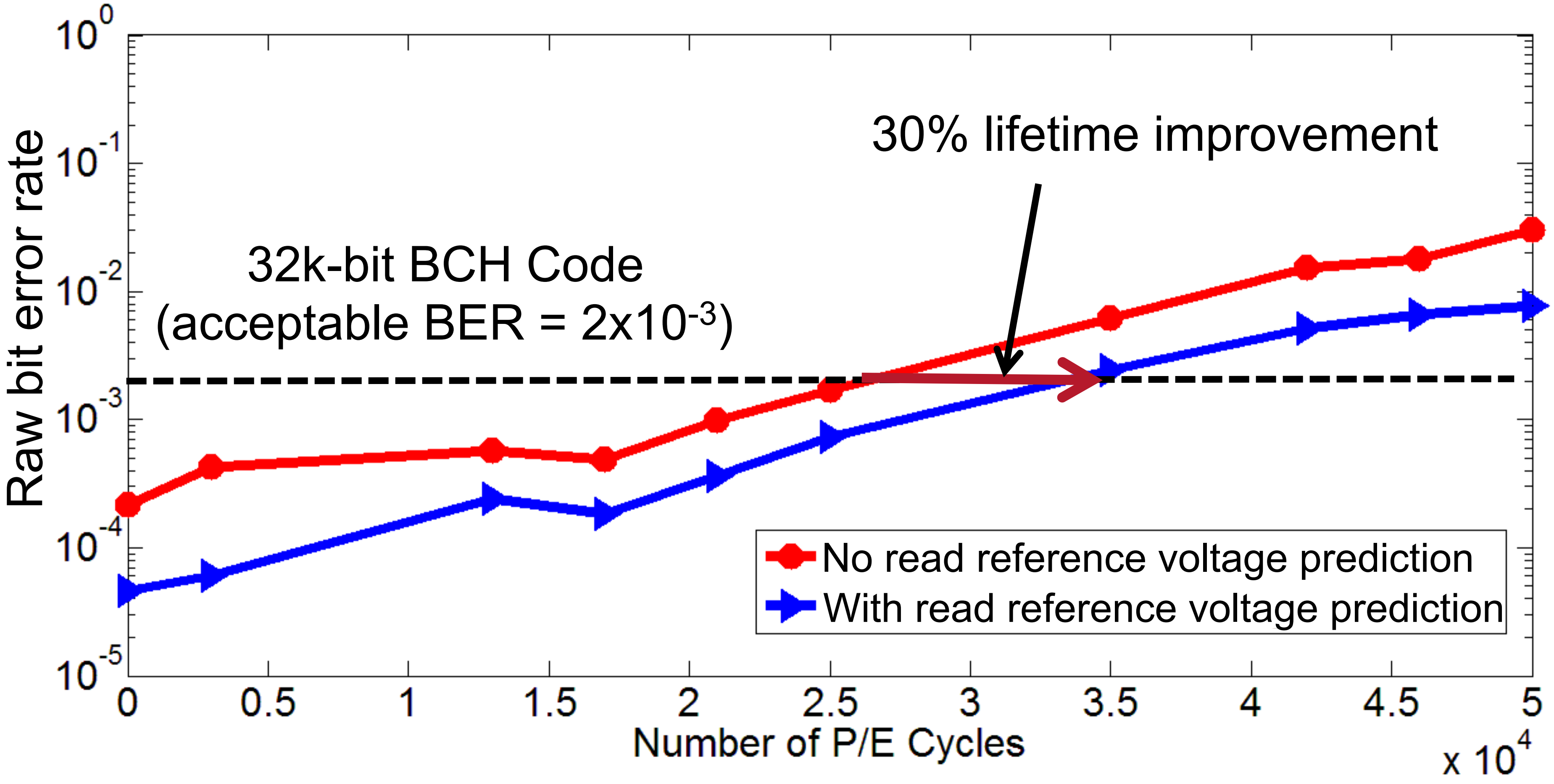
- There exists an optimal read reference voltage
  - Predictable if the statistics (i.e. mean, variance) of threshold voltage distributions are characterized and modeled

Optimum Read Reference Voltage Prediction



- Vth shift learning (done every ~1k P/E cycles)
  - Program sample cells with known data pattern and test Vth
  - Program aggressor neighbor cells and test victim Vth after interference
  - Characterize the mean shift in Vth (i.e., program interference noise)
- Optimum read reference voltage prediction
  - Default read reference voltage + Predicted mean Vth shift by model

Effect of Read Reference Voltage Prediction



- Read reference voltage prediction reduces raw BER (by 64%) and increases the P/E cycle lifetime (by 30%)

More on Read Reference Voltage Prediction

- Yu Cai, Onur Mutlu, Erich F. Haratsch, and Ken Mai, **"Program Interference in MLC NAND Flash Memory: Characterization, Modeling, and Mitigation"** *Proceedings of the 31st IEEE International Conference on Computer Design (**ICCD**)*, Asheville, NC, October 2013. Slides (pptx) (pdf) Lightning Session Slides (pdf)

Agenda

- Background, Motivation and Approach
- Experimental Characterization Methodology
- Error Analysis and Management
  - Main Characterization Results
  - Retention-Aware Error Management
  - Threshold Voltage and Program Interference Analysis
  - Read Reference Voltage Prediction
  - Neighbor-Assisted Error Correction
- Summary

Goal

- Develop a better error correction mechanism for cases where ECC fails to correct a page

Observations So Far

- Immediate neighbor cell has the most effect on the victim cell when programmed

- A single set of read reference voltages is used to determine the value of the (victim) cell

- The set of read reference voltages is determined based on the ***overall threshold voltage distribution of all cells*** in flash memory

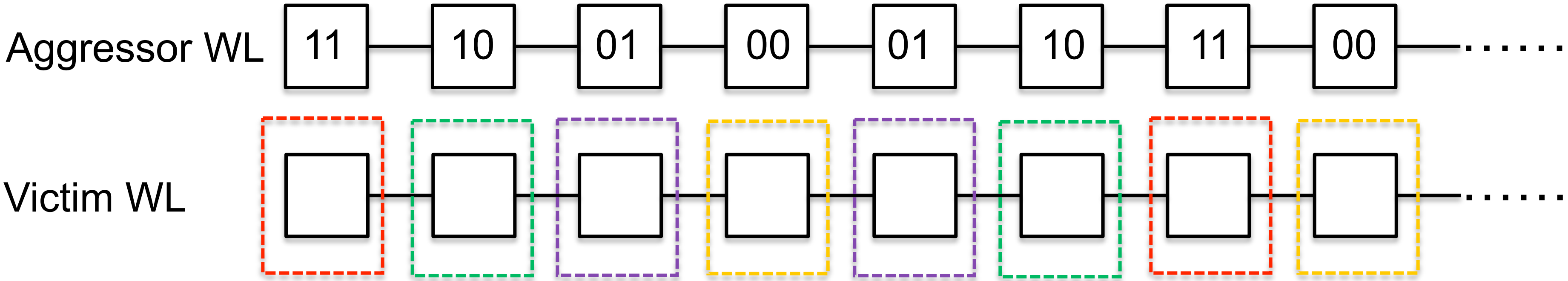
New Observations [Cai+ SIGMETRICS'14]

- Vth distributions of **cells with different-valued immediate-neighbor cells** are significantly different
  - Because neighbor value affects the amount of Vth shift

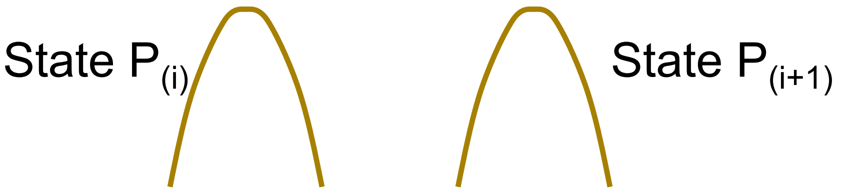
- Corollary: If we know the value of the immediate-neighbor, we can find a more accurate set of read reference voltages based on the "conditional" threshold voltage distribution

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

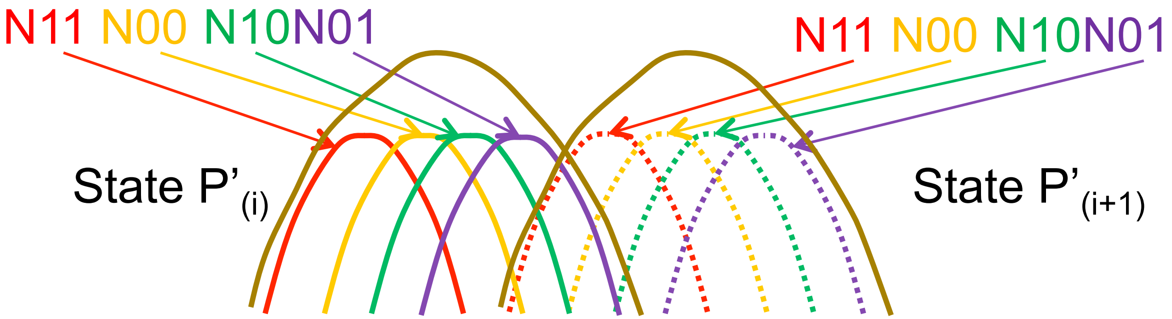
Secrets of Threshold Voltage Distributions



Victim WL **before** MSB page of aggressor WL are programmed



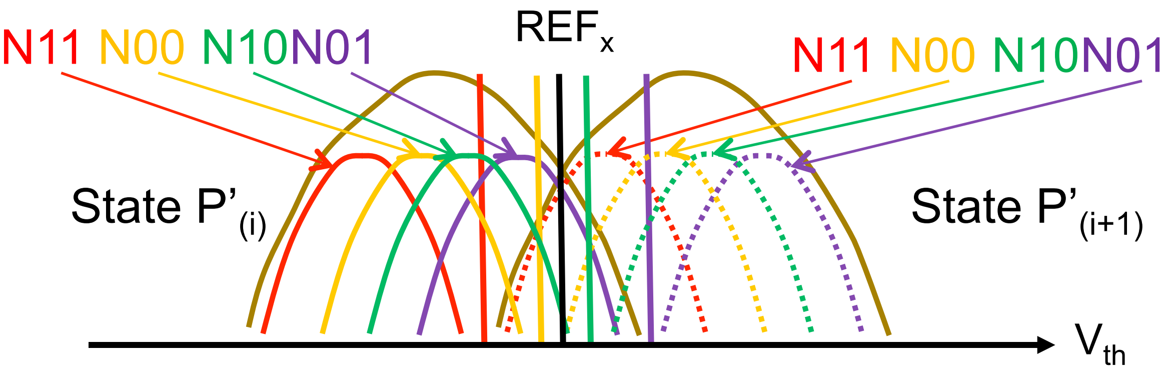
Victim WL **after** MSB page of aggressor WL are programmed



If We Knew the Immediate Neighbor …

- Then, we could choose a different read reference voltage to more accurately read the “victim” cell

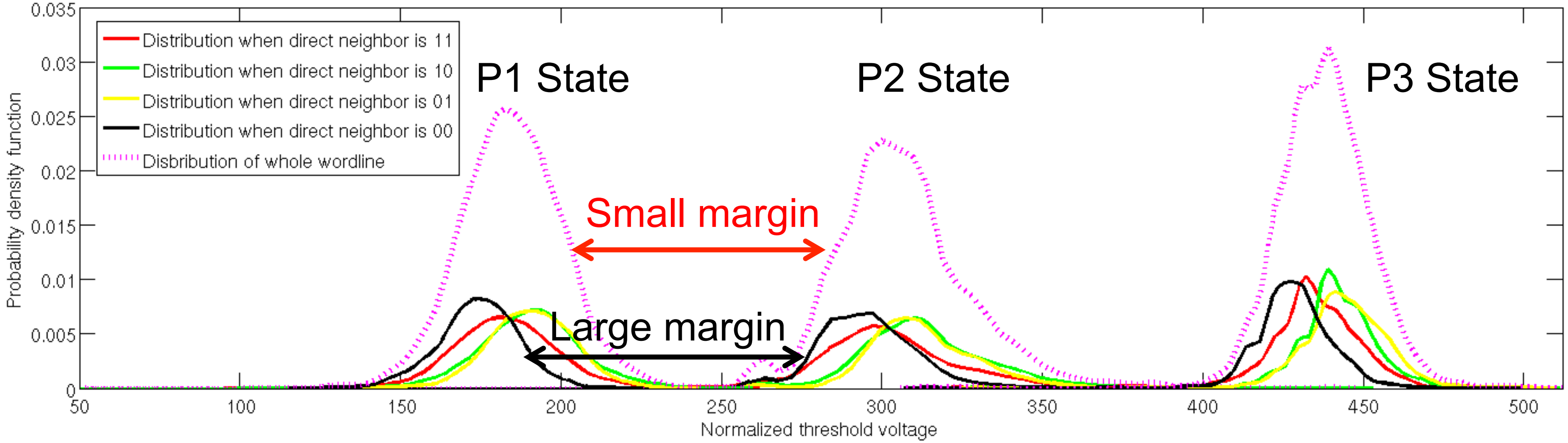
Overall vs Conditional Reading

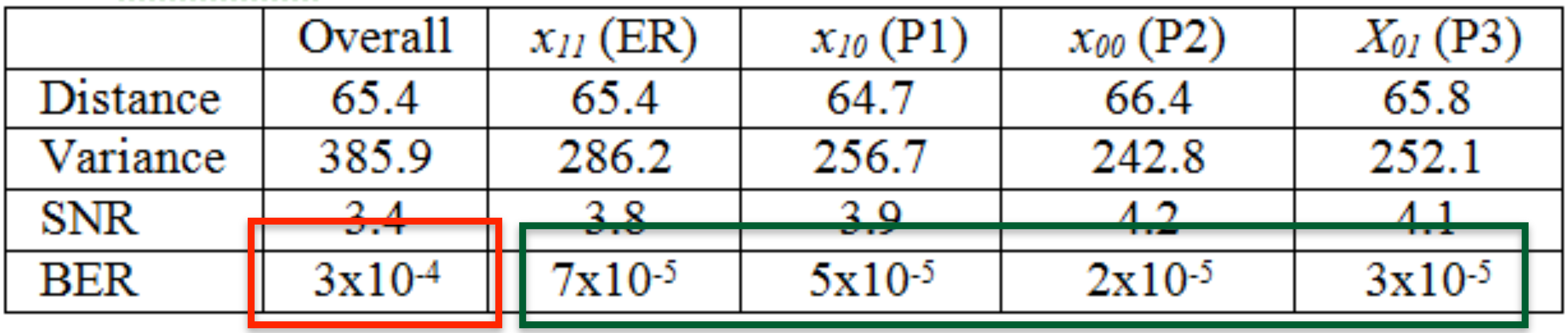


- Using the optimum read reference voltage based on the overall distribution leads to more errors

- Better to use the optimum read reference voltage based on the conditional distribution (i.e., value of the neighbor)
  - Conditional distributions of two states are farther apart from each other

Measurement Results





	Overall	x_{11} (ER)	x_{10} (P1)	x_{00} (P2)	X_{01} (P3)
Distance	65.4	65.4	64.7	66.4	65.8
Variance	385.9	286.2	256.7	242.8	252.1
SNR	3.4	3.8	3.9	4.2	4.1
BER	3x10^{-4}	7x10^{-5}	5x10^{-5}	2x10^{-5}	3x10^{-5}

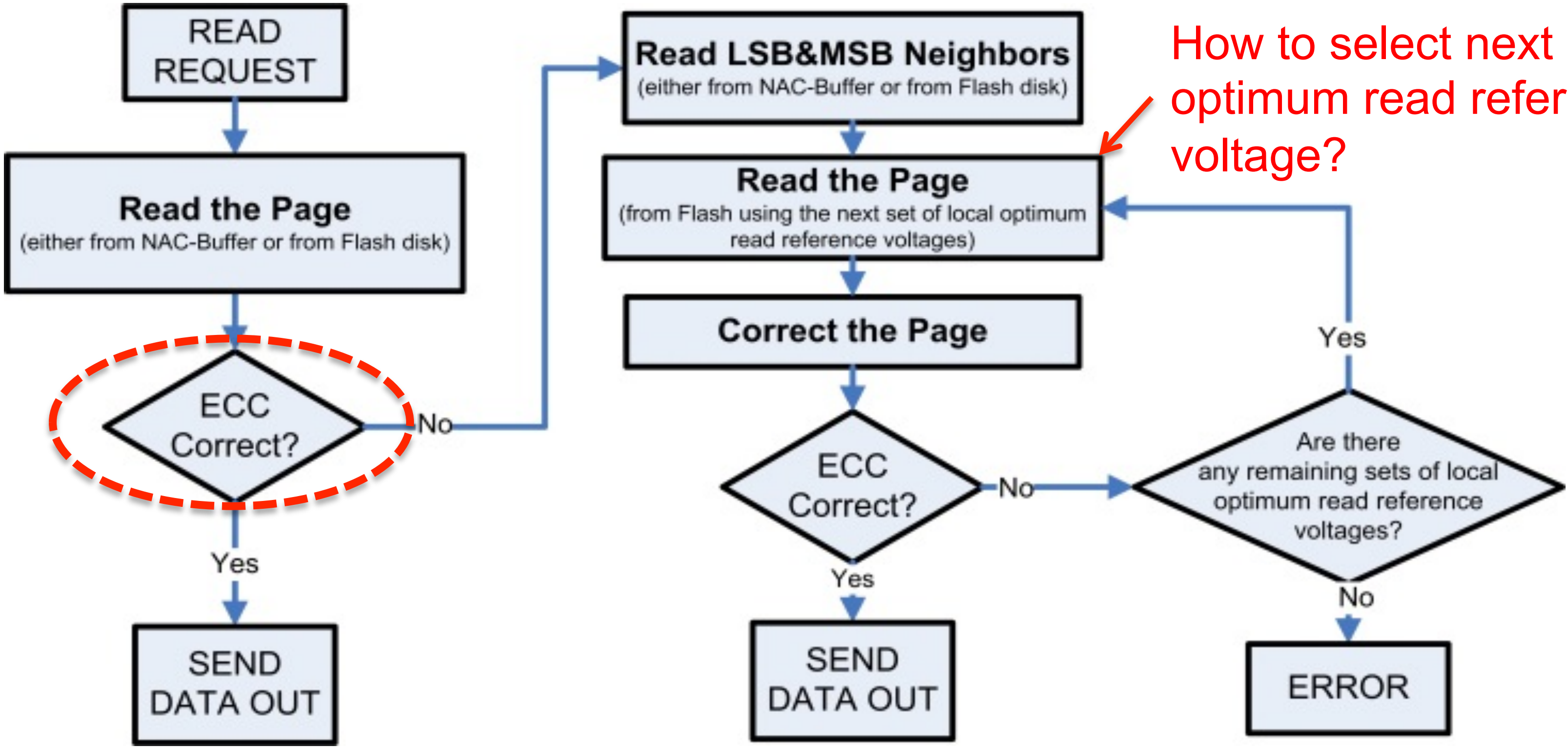
Raw BER of conditional reading is much smaller than overall reading

Idea: Neighbor Assisted Correction (NAC)

- Read a page with the read reference voltages based on overall Vth distribution (same as today) and buffer it

- If ECC fails:
  - Read the immediate-neighbor page
  - Re-read the page using the read reference voltages corresponding to the voltage distribution assuming a particular immediate-neighbor value
  - Replace the buffered values of the cells with that particular immediate-neighbor cell value
  - Apply ECC again

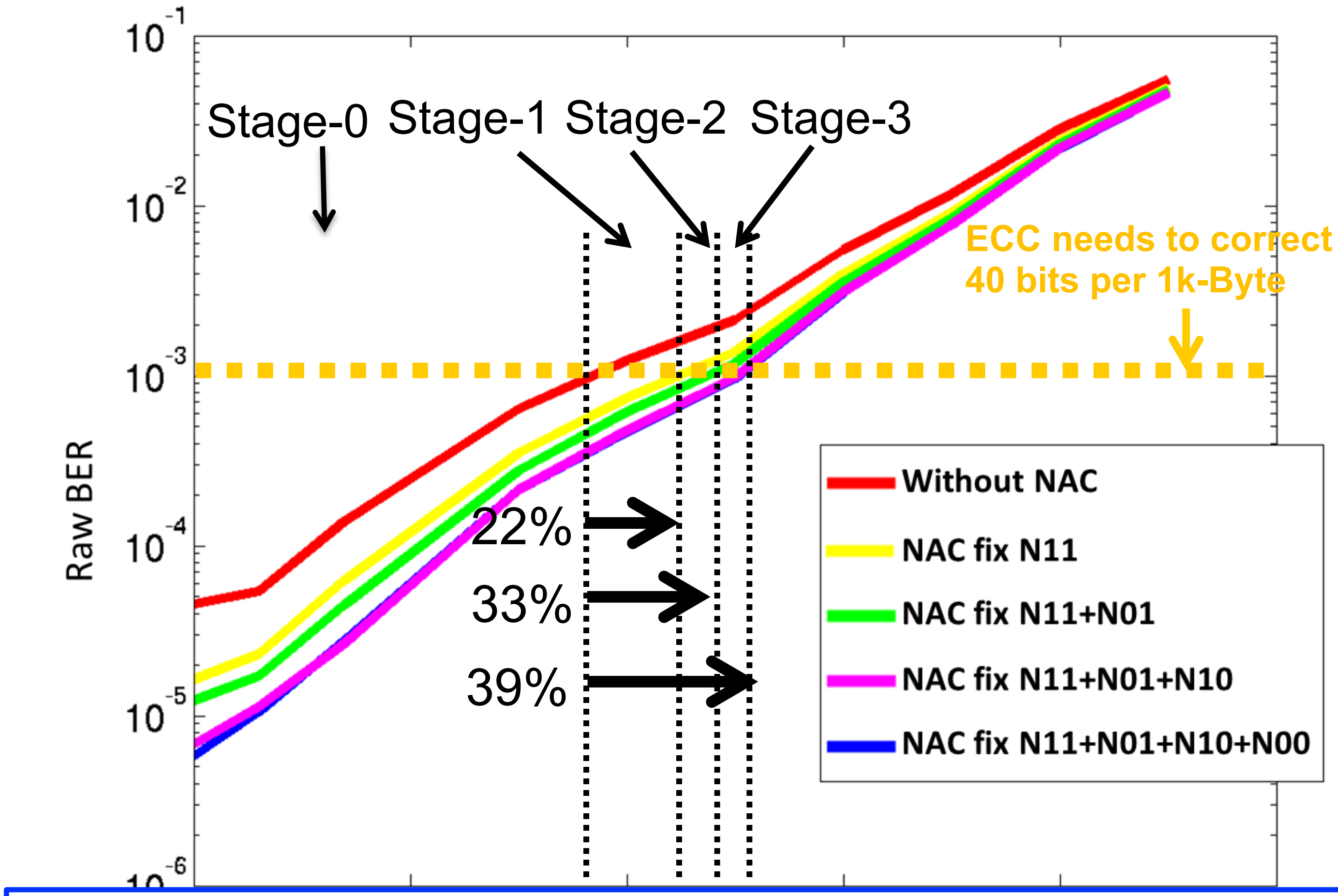
Neighbor Assisted Correction Flow


READ REQUEST
Read the Page
(either from NAC-Buffer or from Flash disk)
ECC Correct?
No
Yes
SEND DATA OUT
Read LSB&MSB Neighbors
(either from NAC-Buffer or from Flash disk)
Read the Page
(from Flash using the next set of local optimum read reference voltages)
Correct the Page
ECC Correct?
No
Yes
SEND DATA OUT
Are there any remaining sets of local optimum read reference voltages?
Yes
No
ERROR

How to select next local optimum read reference voltage?

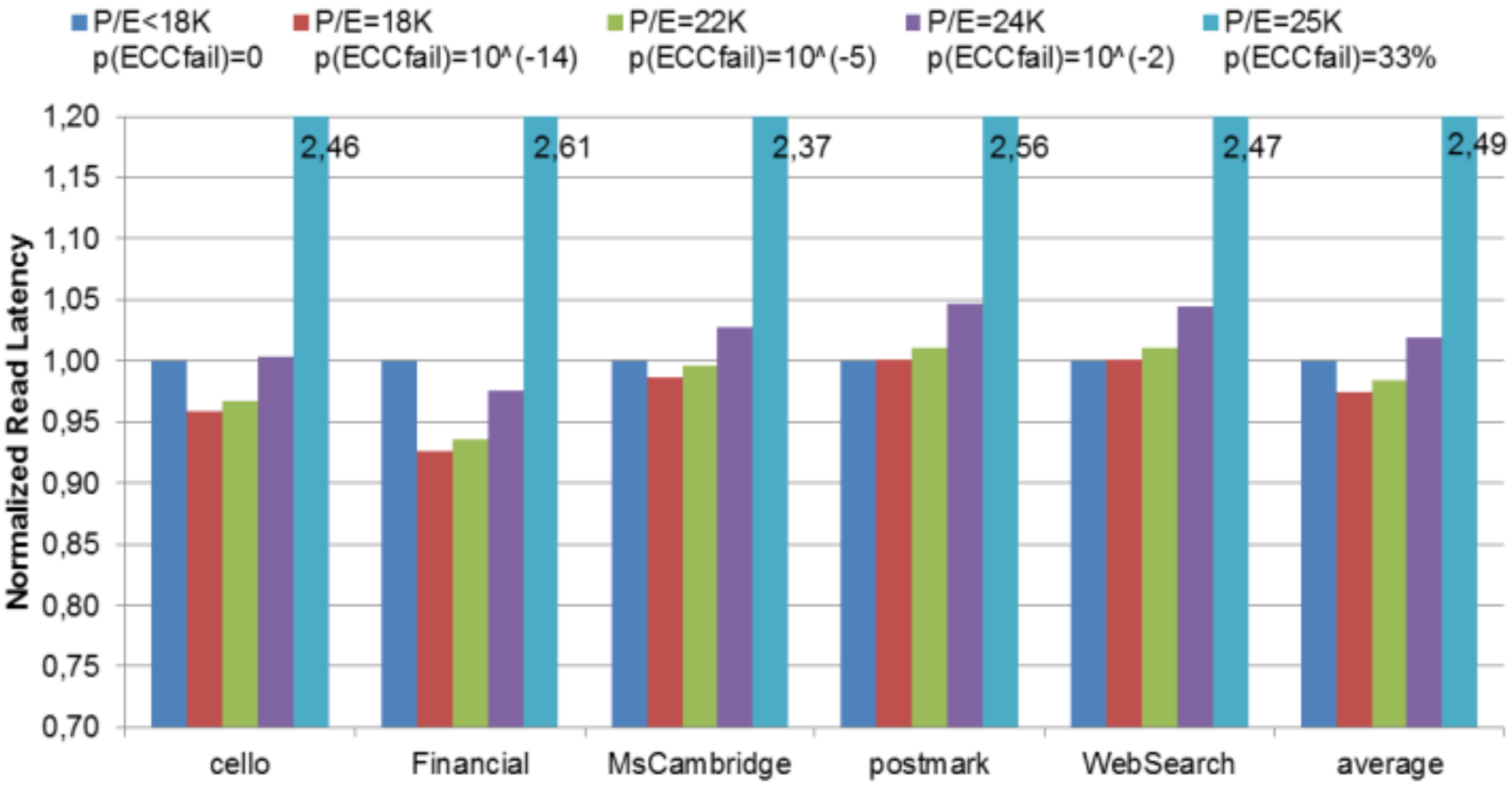
- Trigger neighbor-assisted reading only when ECC fails
- Read neighbor values and use corresponding read reference voltages in a prioritized order until ECC passes

Lifetime Extension with NAC



33% lifetime improvement at no performance loss

Performance Analysis of NAC



No performance loss within nominal lifetime and with reasonable (1%) ECC fail rates

More on Neighbor-Assisted Correction

- Yu Cai, Gulay Yalcin, Onur Mutlu, Eric Haratsch, Osman Unsal, Adrian Cristal, and Ken Mai, **"Neighbor-Cell Assisted Error Correction for MLC NAND Flash Memories"** *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems* (***SIGMETRICS***), Austin, TX, June 2014. Slides (ppt) (pdf)

Agenda

- Background, Motivation and Approach
- Experimental Characterization Methodology
- Error Analysis and Management
  - Main Characterization Results
  - Retention-Aware Error Management
  - Threshold Voltage and Program Interference Analysis
  - Read Reference Voltage Prediction
  - Neighbor-Assisted Error Correction
- Summary

Executive Summary

- Problem: MLC NAND flash memory reliability/endurance is a key challenge for satisfying future storage systems' requirements

- We are: (1) Building reliable error models for NAND flash memory via experimental characterization, (2) Developing efficient techniques to improve reliability and endurance

- This talk provided a "flash" summary of our recent results published in the past 3 years:
  - Experimental error and threshold voltage characterization **[DATE'12&13]**
  - Retention-aware error management **[ICCD'12]**
  - Program interference analysis and read reference V prediction **[ICCD'13]**
  - Neighbor-assisted error correction **[SIGMETRICS'14]**

Readings (I)

- 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)

- 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"
  Proceedings of the 30th IEEE International Conference on Computer Design (***ICCD***), Montreal, Quebec, Canada, September 2012. Slides (ppt) (pdf)

- Yu Cai, Erich F. Haratsch, Onur Mutlu, and Ken Mai,
  "Threshold Voltage Distribution in MLC NAND Flash Memory: Characterization, Analysis and Modeling"
  Proceedings of the Design, Automation, and Test in Europe Conference (***DATE***), Grenoble, France, March 2013. Slides (ppt)

Readings (II)

- 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.

- Yu Cai, Onur Mutlu, Erich F. Haratsch, and Ken Mai,
  "Program Interference in MLC NAND Flash Memory: Characterization, Modeling, and Mitigation"
  *Proceedings of the 31st IEEE International Conference on Computer Design (**ICCD**)*, Asheville, NC, October 2013. Slides (pptx) (pdf) Lightning Session Slides (pdf)

- Yu Cai, Gulay Yalcin, Onur Mutlu, Eric Haratsch, Osman Unsal, Adrian Cristal, and Ken Mai,
  "Neighbor-Cell Assisted Error Correction for MLC NAND Flash Memories"
  *Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (**SIGMETRICS**)*, Austin, TX, June 2014. Slides (ppt) (pdf)

Referenced Papers

- All are available at http://users.ece.cmu.edu/~omutlu/projects.htm

Related Videos and Course Materials

- Computer Architecture Lecture Videos on Youtube
  - https://www.youtube.com/playlist?list=PL5PHm2jkkXmidJOd59REog9jDnPDTG6IJ

- Computer Architecture Course Materials
  - http://www.ece.cmu.edu/~ece447/s13/doku.php?id=schedule

- Advanced Computer Architecture Course Materials
  - http://www.ece.cmu.edu/~ece740/f13/doku.php?id=schedule

- Advanced Computer Architecture Lecture Videos on Youtube
  - https://www.youtube.com/playlist?list=PL5PHm2jkkXmgDN1PLwOY_tGtUlynnyV6D

Thank you.

Feel free to email me with any questions & feedback

onur@cmu.edu
http://users.ece.cmu.edu/~omutlu/

Error Analysis and Management for MLC NAND Flash Memory

Onur Mutlu
onur@cmu.edu

(joint work with Yu Cai, Gulay Yalcin, Eric Haratsch, Ken Mai, Adrian Cristal, Osman Unsal)

August 7, 2014

Flash Memory Summit 2014, Santa Clara, CA