Computer Architecture

Lecture 2: RowHammer and Beyond

Prof. Onur Mutlu
ETH Zürich
Fall 2018
20 September 2018
We Will Dig Deeper More
In This Course

“Good ideas are a dime a dozen”

“Making them work is oftentimes the real contribution”
Another Example

- **DRAM Row Hammer (or, DRAM Disturbance Errors)**

- How a simple hardware failure mechanism can create a widespread system security vulnerability
Maslow’s (Human) Hierarchy of Needs


- We need to start with reliability and security...
How Reliable/Secure/Safe is This Bridge?

Source: http://www.technologystudent.com/struct1/tacom1.png
Collapse of the “Galloping Gertie”
How Secure Are These People?

Security is about preventing unforeseen consequences
The DRAM Scaling Problem

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

- DRAM capacity, cost, and energy/power hard to scale
As Memory Scales, It Becomes Unreliable

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

![Graph showing the relationship between chip density (Gb) and relative server failure rate.](image)

**Intuition:** quadratic increase in capacity
Large-Scale Failure Analysis of DRAM Chips

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

- Justin Meza, Qiang Wu, Sanjeev Kumar, and Onur Mutlu, "Revisiting Memory Errors in Large-Scale Production Data Centers: Analysis and Modeling of New Trends from the Field" Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Rio de Janeiro, Brazil, June 2015.

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

Justin Meza  Qiang Wu*  Sanjeev Kumar*  Onur Mutlu
Carnegie Mellon University  *Facebook, Inc.
Infrastructures to Understand Such Issues

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)
Infrastructures to Understand Such Issues

SoftMC: Open Source DRAM Infrastructure


- Flexible
- Easy to Use (C++ API)
- Open-source

github.com/CMU-SAFARI/SoftMC
SoftMC

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

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

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

\textsuperscript{1}ETH Zürich \textsuperscript{2}TOBB University of Economics & Technology \textsuperscript{3}Carnegie Mellon University \textsuperscript{4}University of Virginia \textsuperscript{5}Microsoft Research \textsuperscript{6}NVIDIA Research
Data Retention in Memory [Liu et al., ISCA 2013]

- Retention Time Profile of DRAM looks like this:

- Location dependent
- Stored value pattern dependent
- Time dependent

Retention Time Profile:
- 64-128ms
- >256ms
- 128-256ms
One can predictably induce errors in most DRAM memory chips.
DRAM RowHammer

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

- A Modules
- B Modules
- C Modules

Errors per $10^9$ Cells

Module Vintage

2008 2009 2010 2011 2012 2013 2014
Recent DRAM Is More Vulnerable

![Graph showing error rates and module vintage]

- **A Modules**
- **B Modules**
- **C Modules**

**First Appearance**

Errors per $10^9$ Cells

Module Vintage

Recent DRAM Is More Vulnerable

All modules from 2012–2013 are vulnerable
Why Is This Happening?

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

- Access to one cell affects the value in nearby cells
  - due to electrical interference between
    - the cells
    - wires used for accessing the cells
  - Also called cell-to-cell coupling/interference

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

- This simple circuit level failure mechanism has enormous implications on upper layers of the transformation hierarchy.
A Simple Program Can Induce Many Errors

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

Download from: https://github.com/CMU-SAFARI/rowhammer
A Simple Program Can Induce Many Errors

1. Avoid *cache hits*
   – Flush \( X \) from cache

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

Download from: [https://github.com/CMU-SAFARI/rowhammer](https://github.com/CMU-SAFARI/rowhammer)
A Simple Program Can Induce Many Errors

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

Download from: https://github.com/CMU-SAFARI/rowhammer
A Simple Program Can Induce Many Errors

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

Download from: https://github.com/CMU-SAFARI/rowhammer
A Simple Program Can Induce Many Errors

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

Download from: https://github.com/CMU-SAFARI/rowhammer
# Observed Errors in Real Systems

<table>
<thead>
<tr>
<th>CPU Architecture</th>
<th>Errors</th>
<th>Access-Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Haswell (2013)</td>
<td>22.9K</td>
<td>12.3M/sec</td>
</tr>
<tr>
<td>Intel Ivy Bridge (2012)</td>
<td>20.7K</td>
<td>11.7M/sec</td>
</tr>
<tr>
<td>Intel Sandy Bridge (2011)</td>
<td>16.1K</td>
<td>11.6M/sec</td>
</tr>
<tr>
<td>AMD Piledriver (2012)</td>
<td>59</td>
<td>6.1M/sec</td>
</tr>
</tbody>
</table>

A real reliability & security issue

One Can Take Over an Otherwise-Secure System

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

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

Project Zero

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
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

Rowhammer
Security Implications

It’s like breaking into an apartment by repeatedly slamming a neighbor’s door until the vibrations open the door you were after.
Selected Readings on RowHammer (I)

- Our first detailed study: Rowhammer analysis and solutions (June 2014)

- Our Source Code to Induce Errors in Modern DRAM Chips (June 2014)
  - https://github.com/CMU-SAFARI/rowhammer

- Google Project Zero’s Attack to Take Over a System (March 2015)
  - Exploiting the DRAM rowhammer bug to gain kernel privileges (Seaborn+, 2015)
  - https://github.com/google/rowhammer-test
  - Double-sided Rowhammer
Selected Readings on RowHammer (II)

- **Remote RowHammer Attacks via JavaScript** (July 2015)
  - [https://github.com/IAIK/rowhammerjs](https://github.com/IAIK/rowhammerjs)
  - Gruss et al., DIMVA 2016.
  - **CLFLUSH-free Rowhammer**
    - “A fully automated attack that requires nothing but a website with JavaScript to trigger faults on remote hardware.”
    - “We can gain unrestricted access to systems of website visitors.”

- **ANVIL: Software-Based Protection Against Next-Generation Rowhammer Attacks** (March 2016)
  - [http://dl.acm.org/citation.cfm?doid=2872362.2872390](http://dl.acm.org/citation.cfm?doid=2872362.2872390)
  - Aweke et al., ASPLOS 2016
  - **CLFLUSH-free Rowhammer**
  - Software based monitoring for rowhammer detection
Selected Readings on RowHammer (III)

- Dedup Est Machina: Memory Deduplication as an Advanced Exploitation Vector (May 2016)
  - Exploits Rowhammer and Memory Deduplication to overtake a browser
  - “We report on the first reliable remote exploit for the Rowhammer vulnerability running entirely in Microsoft Edge.”
  - “[an attacker] ... can reliably “own” a system with all defenses up, even if the software is entirely free of bugs.”
Selected Readings on RowHammer (IV)

- **Flip Feng Shui: Hammering a Needle in the Software Stack** (August 2016)
  - Razavi et al., USENIX Security 2016.
  - Combines memory deduplication and RowHammer
  - “A malicious VM can gain unauthorized access to a co-hosted VM running OpenSSH.”
  - Breaks OpenSSH public key authentication

- **Drammer: Deterministic Rowhammer Attacks on Mobile Platforms** (October 2016)
  - [http://dl.acm.org/citation.cfm?id=2976749.2978406](http://dl.acm.org/citation.cfm?id=2976749.2978406)
  - Van Der Veen et al., CCS 2016
  - **Can take over an ARM-based Android system deterministically**
  - Exploits predictable physical memory allocator behavior
    - Can deterministically place security-sensitive data (e.g., page table) in an attacker-chosen, vulnerable location in memory
Selected Readings on RowHammer (V)

- **Grand Pwning Unit: Accelerating Microarchitectural Attacks with the GPU** (May 2018)
  - The first end-to-end remote Rowhammer exploit on mobile platforms that use our GPU-based primitives in orchestration to **compromise browsers on mobile devices in under two minutes**.

- **Throwhammer: Rowhammer Attacks over the Network and Defenses** (July 2018)
  - Tatar et al., USENIX ATC 2018.
  - “[We] show that an attacker can trigger and exploit Rowhammer bit flips directly from a remote machine by only sending network packets.”
Selected Readings on RowHammer (VI)

- Nethammer: Inducing Rowhammer Faults through Network Requests (July 2018)
  - Lipp et al., arxiv.org 2018.
  - “Nethammer is the first truly remote Rowhammer attack, without a single attacker-controlled line of code on the targeted system.”
More Security Implications (I)

“We can gain unrestricted access to systems of website visitors.”

ROOT privileges for web apps!

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

Source: https://lab.dsst.io/32c3-slides/7197.html
More Security Implications (II)

“Can gain control of a smart phone deterministically”

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

Drammer: Deterministic Rowhammer Attacks on Mobile Platforms, CCS’16
More Security Implications (III)

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

"GRAND PWNING UNIT" —

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

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

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

Grand Pwning Unit: Accelerating Microarchitectural Attacks with the GPU

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

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

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

Kaveh Razavi
Vrije Universiteit
Amsterdam
kaveh@cs.vu.nl
More Security Implications (IV)

- Rowhammer over RDMA (I)

THROWHAMMER —

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

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

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

Throwhammer: Rowhammer Attacks over the Network and Defenses

Andrei Tatar  
VU Amsterdam

Radhesh Krishnan  
VU Amsterdam

Elias Athanasopoulos  
University of Cyprus

Cristiano Giuffrida  
VU Amsterdam

Herbert Bos  
VU Amsterdam

Kaveh Razavi  
VU Amsterdam
More Security Implications (V)

- Rowhammer over RDMA (II)

The Hacker News™
Security in a serious way

Nethammer—Exploiting DRAM Rowhammer Bug Through Network Requests

Nethammer:
Inducing Rowhammer Faults through Network Requests

Moritz Lipp
Graz University of Technology

Misiker Tadesse Aga
University of Michigan

Michael Schwarz
Graz University of Technology

Daniel Gruss
Graz University of Technology

Clémentine Maurice
Univ Rennes, CNRS, IRISA

Lukas Raab
Graz University of Technology

Lukas Lamster
Graz University of Technology
More Security Implications?
Where RowHammer Was Discovered...

Understanding RowHammer
Root Causes of Disturbance Errors

• **Cause 1: Electromagnetic coupling**
  – Toggling the wordline voltage briefly increases the voltage of adjacent wordlines
  – Slightly opens adjacent rows → Charge leakage

• **Cause 2: Conductive bridges**

• **Cause 3: Hot-carrier injection**

Confirmed by at least one manufacturer
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)
RowHammer Characterization Results

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

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

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

Most aggressors & victims are adjacent
1. Access Interval (Aggressor)

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

Less frequent accesses ➔ Fewer errors
Note: Using three modules with the most errors (only first bank)

More frequent refreshes → Fewer errors
### Data Pattern

<table>
<thead>
<tr>
<th>Solid</th>
<th>RowStripe</th>
</tr>
</thead>
<tbody>
<tr>
<td>111111</td>
<td>111111</td>
</tr>
<tr>
<td>111111</td>
<td>000000</td>
</tr>
<tr>
<td>111111</td>
<td>111111</td>
</tr>
<tr>
<td>111111</td>
<td>000000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>~Solid</th>
<th>~RowStripe</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000</td>
<td>000000</td>
</tr>
<tr>
<td>000000</td>
<td>111111</td>
</tr>
<tr>
<td>000000</td>
<td>000000</td>
</tr>
<tr>
<td>000000</td>
<td>111111</td>
</tr>
</tbody>
</table>

*Errors affected by data stored in other cells*
6. Other Results (in Paper)

- **Victim Cells ≠ Weak Cells (i.e., leaky cells)**
  - Almost no overlap between them

- **Errors not strongly affected by temperature**
  - Default temperature: 50°C
  - At 30°C and 70°C, number of errors changes <15%

- **Errors are repeatable**
  - Across ten iterations of testing, >70% of victim cells had errors in every iteration
6. Other Results (in Paper) cont’d

• As many as 4 errors per cache-line
  – Simple ECC (e.g., SECDED) cannot prevent all errors

• Number of cells & rows affected by aggressor
  – Victims cells per aggressor: ≤110
  – Victims rows per aggressor: ≤9

• Cells affected by two aggressors on either side
  – Very small fraction of victim cells (<100) have an error when either one of the aggressors is toggled
More on RowHammer Analysis


[Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)] [Source Code and Data]
Retrospective on RowHammer & Future

Onur Mutlu,
"The RowHammer Problem and Other Issues We May Face as Memory Becomes Denser"
[Slides (pptx) (pdf)]
RowHammer Solutions
Some Potential Solutions

- Make better DRAM chips  
  **Cost**

- Refresh frequently  
  **Power, Performance**

- Sophisticated ECC  
  **Cost, Power**

- Access counters  
  **Cost, Power, Complexity**
Naive Solutions

1. *Throttle accesses to same row*
   - Limit access-interval: $\geq 500\text{ns}$
   - Limit number of accesses: $\leq 128\text{K} (=64\text{ms}/500\text{ns})$

2. *Refresh more frequently*
   - Shorten refresh-interval by $\sim 7x$

Both naive solutions introduce significant overhead in performance and power
Apple’s Patch for RowHammer


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

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

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

CVE-ID

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

HP, Lenovo, and other vendors released similar patches
Our Solution to RowHammer

- **PARA:** _Probabilistic Adjacent Row Activation_

- **Key Idea**
  - After closing a row, we activate (i.e., refresh) one of its neighbors with a low probability: \( p = 0.005 \)

- **Reliability Guarantee**
  - When \( p=0.005 \), errors in one year: \( 9.4 \times 10^{-14} \)
  - By adjusting the value of \( p \), we can vary the strength of protection against errors
Advantages of PARA

• **PARA refreshes rows infrequently**
  – Low power
  – Low performance-overhead
    • Average slowdown: 0.20% (for 29 benchmarks)
    • Maximum slowdown: 0.75%

• **PARA is stateless**
  – Low cost
  – Low complexity

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

• If implemented in **DRAM chip**
  – Enough slack in timing parameters
  – Plenty of slack today:
    • Chang et al., “Understanding Latency Variation in Modern DRAM Chips,” SIGMETRICS 2016.
    • Lee et al., “Design-Induced Latency Variation in Modern DRAM Chips,” SIGMETRICS 2017.
    • Chang et al., “Understanding Reduced-Voltage Operation in Modern DRAM Devices,” SIGMETRICS 2017.

• If implemented in **memory controller**
  – Better coordination between memory controller and DRAM
  – Memory controller should know which rows are physically adjacent
Probabilistic Activation in Real Life (I)

https://twitter.com/isislovecruft/status/1021939922754723841
More on RowHammer Analysis

- Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors"
  [Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)] [Source Code and Data]
Onur Mutlu,
"The RowHammer Problem and Other Issues We May Face as Memory Becomes Denser"
[Slides (pptx) (pdf)]
Industry Is Writing Papers About It, Too

DRAM Process Scaling Challenges

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

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

- **VRT**
  - Occurring more frequently with cell capacitance decreasing
Call for Intelligent Memory Controllers

DRAM Process Scaling Challenges

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

THE MEMORY FORUM 2014

Co-Architecting Controllers and DRAM to Enhance DRAM Process Scaling

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

Samsung Electronics, Hwasung, Korea / *Samsung Electronics, San Jose / **Intel
Aside: Intelligent Controller for NAND Flash


Aside: Intelligent Controller for NAND Flash

Proceedings of the IEEE, Sept. 2017

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

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

By Yu Cai, Saugata Ghose, Erich F. Haratsch, Yixin Luo, and Onur Mutlu

https://arxiv.org/pdf/1706.08642
A Key Takeaway

Main Memory Needs

Intelligent Controllers
Challenge and Opportunity for Future

Fundamentally Secure, Reliable, Safe Computing Architectures
Some Thoughts on RowHammer

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

- How to exploit and fix the vulnerability requires a strong understanding across the transformation layers
  - And, a strong understanding of tools available to you

- Fixing needs to happen for two types of chips
  - Existing chips (already in the field)
  - Future chips

- Mechanisms for fixing are different between the two types
Aside: Byzantine Failures

- This class of failures is known as Byzantine failures

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

- “erroneous” can be “malicious” (intent is the only distinction)

- Very difficult to detect and confine Byzantine failures

- Do all you can to avoid them

More on RowHammer (I)

- **Our first detailed study: Rowhammer analysis and solutions** (June 2014)
  - Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu,
  "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors"

- **Our Source Code to Induce Errors in Modern DRAM Chips** (June 2014)
  - [https://github.com/CMU-SAFARI/rowhammer](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](https://github.com/google/rowhammer-test)
  - Double-sided Rowhammer
More on RowHammer (II)

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

Yoongu Kim¹ Ross Daly* Jeremie Kim¹ Chris Fallin* Ji Hye Lee¹ Donghyuk Lee¹ Chris Wilkerson² Konrad Lai Onur Mutlu¹

¹Carnegie Mellon University ²Intel Labs

One potential reading for your Homework 1 assignment
Retrospective on RowHammer & Future

- Onur Mutlu,
  "The RowHammer Problem and Other Issues We May Face as Memory Becomes Denser"
  [Slides (pptx) (pdf)]

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

Onur Mutlu
ETH Zürich
onur.mutlu@inf.ethz.ch
https://people.inf.ethz.ch/omutlu
Future Memory
Reliability/Security Challenges
Future of Main Memory

- DRAM is becoming less reliable $\rightarrow$ more vulnerable
Large-Scale Failure Analysis of DRAM Chips

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

- Justin Meza, Qiang Wu, Sanjeev Kumar, and Onur Mutlu, "Revisiting Memory Errors in Large-Scale Production Data Centers: Analysis and Modeling of New Trends from the Field" Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Rio de Janeiro, Brazil, June 2015.

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

![Graph showing relative server failure rate vs. chip density (Gb)].

**Intuition:** quadratic increase in capacity

Meza+, “Revisiting Memory Errors in Large-Scale Production Data Centers,” DSN’15.
Future of Main Memory

- DRAM is becoming less reliable → more vulnerable
- Due to difficulties in DRAM scaling, other problems may also appear (or they may be going unnoticed)

- Some errors may already be slipping into the field
  - Read disturb errors (Rowhammer)
  - Retention errors
  - Read errors, write errors
  - ...

- These errors can also pose security vulnerabilities
DRAM Data Retention Time Failures

- Determining the data retention time of a cell/row is getting more difficult

- Retention failures may already be slipping into the field
Analysis of Data Retention Failures [ISCA’13]


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

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

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

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

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

Onur Mutlu
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213
onur@cmu.edu
Two Challenges to Retention Time Profiling

- Data Pattern Dependence (DPD) of retention time
- Variable Retention Time (VRT) phenomenon
Two Challenges to Retention Time Profiling

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

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

- Bitline-bitline coupling → electrical coupling between adjacent bitlines
- Bitline-wordline coupling → electrical coupling between each bitline and the activated wordline

Retention time of a cell depends on data patterns stored in nearby cells:

→ need to find the worst data pattern to find worst-case retention time
→ this pattern is location dependent
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
  - Worst-case retention time depends on a random process
    → need to find the worst case despite this

[Kim et al., IEEE TED’11]
Newer device families have more weak cells than older ones. Likely a result of technology scaling.
An Example VRT Cell

A cell from E 2Gb chip family
Variable Retention Time

Many failing cells jump from very high retention time to very low.

Most failing cells exhibit VRT.

Min ret time = Max ret time
Expected if no VRT

A 2Gb chip family
Industry Is Writing Papers About It, Too

**DRAM Process Scaling Challenges**

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

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

- **VRT**
  - Occurring more frequently with cell capacitance decreasing
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
Mitigation of Retention Issues [SIGMETRICS’14]

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

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

Key Observations:

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

1. Initially Protect DRAM with Strong ECC
2. Periodically Test Parts of DRAM
3. Mitigate errors and reduce ECC

Run tests periodically after a short interval at smaller regions of memory
Handling Variable Retention Time [DSN’15]

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

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

Moinuddin K. Qureshi†  Dae-Hyun Kim†  Samira Khan‡  Prashant J. Nair†  Onur Mutlu‡
†Georgia Institute of Technology
‡Carnegie Mellon University
{moin, dhkim, pnair6}@ece.gatech.edu
{samirakhan, onur}@cmu.edu
Samira Khan, Donghyuk Lee, and Onur Mutlu, "PARBOR: An Efficient System-Level Technique to Detect Data-Dependent Failures in DRAM"


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

Samira Khan*  Donghyuk Lee†‡  Onur Mutlu*†
*University of Virginia  †Carnegie Mellon University  ‡Nvidia  *ETH Zürich
Detecting and Mitigating Data-Dependent DRAM Failures by Exploiting Current Memory Content

Samira Khan*  Chris Wilkerson†  Zhe Wang†  Alaa R. Alameldeen†  Donghyuk Lee‡  Onur Mutlu*

*University of Virginia  †Intel Labs  ‡Nvidia Research  *ETH Zürich
Handling Both DPD and VRT [ISCA'17]

- Minesh Patel, Jeremie S. Kim, and Onur Mutlu,
"The Reach Profiler (REAPER): Enabling the Mitigation of DRAM Retention Failures via Profiling at Aggressive Conditions"
[Slides (pptx) (pdf)]
[Lightning Session Slides (pptx) (pdf)]

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

The Reach Profiler (REAPER):
Enabling the Mitigation of DRAM Retention Failures via Profiling at Aggressive Conditions
Minesh Patel$\&\#\#$, Jeremie S. Kim$\&\#\#\&\#\#$, Onur Mutlu$\&\#\#$
$\&$ETH Zürich  $\&\#\#$Carnegie Mellon University
The DRAM Latency PUF:
Quickly Evaluating Physical Unclonable Functions by Exploiting the Latency-Reliability Tradeoff in Modern Commodity DRAM Devices

Jeremie S. Kim  Minesh Patel
Hasan Hassan  Onur Mutlu

QR Code for the paper

HPCA 2018
Keeping Future Memory Secure
How Do We Keep Memory Secure?

- DRAM

- Flash memory

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

SAFARI 109
Design fundamentally secure computing architectures

Predict and prevent such safety issues
Recall: Collapse of the “Galloping Gertie”
Another Example (1994)

Source: By 최광모 - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=35197984
Yet Another Example (2007)

A More Recent Example (2018)

[Image of the Genoa bridge collapse]

Architecting for Security

- **Understand**: Methods for vulnerability modeling & discovery
  - Modeling and prediction based on real (device) data and analysis
  - Understanding vulnerabilities
  - Developing reliable metrics

- **Architect**: Principled architectures with security as key concern
  - Good partitioning of duties across the stack
  - Cannot give up performance and efficiency
  - Patch-ability in the field

- **Design & Test**: Principled design, automation, (online) testing
  - Design for security
  - High coverage and good interaction with system reliability methods
Understand and Model with Experiments (DRAM)

Understand and Model with Experiments (Flash)

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

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

By Yu Cai, Saugata Ghose, Erich F. Haratsch, Yixin Luo, and Onur Mutlu

https://arxiv.org/pdf/1706.08642
Understanding Flash Memory Reliability

- Justin Meza, Qiang Wu, Sanjeev Kumar, and Onur Mutlu, "A Large-Scale Study of Flash Memory Errors in the Field" Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS), Portland, OR, June 2015. [Slides (pptx) (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
Vulnerabilities in MLC NAND Flash Memory Programming: Experimental Analysis, Exploits, and Mitigation Techniques

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

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

In this paper, we demonstrate that two-step programming exposes new reliability and security vulnerabilities. We expe-
HeatWatch: Improving 3D NAND Flash Memory Device Reliability by Exploiting Self-Recovery and Temperature Awareness

Yixin Luo†, Saugata Ghose†, Yu Cai‡, Erich F. Haratsch‡, and Onur Mutlu§†

†Carnegie Mellon University  ‡Seagate Technology  §ETH Zürich

Abstract

Yixin Luo†, Saugata Ghose†, Yu Cai†, Erich F. Haratsch‡, Onur Mutlu§†

†Carnegie Mellon University ‡Seagate Technology §ETH Zürich
Another Talk: NAND Flash Memory Robustness

Yu Cai, Saugata Ghose, Erich F. Haratsch, Yixin Luo, and Onur Mutlu,
"Error Characterization, Mitigation, and Recovery in Flash Memory Based Solid State Drives"


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

There are Two Other Solution Directions

- **New Technologies:** Replace or (more likely) augment DRAM with a different technology
  - Non-volatile memories

- **Embracing Un-reliability:**
  Design memories with different reliability and store data intelligently across them
  [Luo+ DSN 2014]

- ...
Exploiting Memory Error Tolerance with Hybrid Memory Systems

Vulnerable data

Tolerant data

Reliable memory

Low-cost memory

On Microsoft’s Web Search workload
Reduces server hardware cost by 4.7 %
Achieves single server availability target of 99.90 %

Heterogeneous-Reliability Memory [DSN 2014]
More on Heterogeneous-Reliability Memory

- Yixin Luo, Sriram Govindan, Bikash Sharma, Mark Santaniello, Justin Meza, Aman Kansal, Jie Liu, Badriddine Khessib, Kushagra Vaid, and Onur Mutlu, "Characterizing Application Memory Error Vulnerability to Optimize Data Center Cost via Heterogeneous-Reliability Memory"

Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Atlanta, GA, June 2014. [Summary] [Slides (pptx) (pdf)] [Coverage on ZDNet]
RowHammer Conclusion
Summary: Memory Reliability and Security

- Memory reliability is reducing
- Reliability issues open up security vulnerabilities
  - Very hard to defend against
- Rowhammer is an example
  - Its implications on system security research are tremendous & exciting

- **Good news: We have a lot more to do.**
- **Understand:** Solid methodologies for failure modeling and discovery
  - Modeling based on real device data – small scale and large scale
- **Architect:** Principled co-architecting of system and memory
  - Good partitioning of duties across the stack
- **Design & Test:** Principled electronic design, automation, testing
  - High coverage and good interaction with system reliability methods
Fundamentally Secure, Reliable, Safe Computing Architectures
One Important Takeaway

Main Memory Needs
Intelligent Controllers
Computer Architecture

Lecture 2: RowHammer and Beyond

Prof. Onur Mutlu
ETH Zürich
Fall 2018
20 September 2018
Flash Memory
Reliability and Security
Limits of Charge Memory

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

- Reliable sensing becomes difficult as charge storage unit size reduces
Evolution of NAND Flash Memory

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

Seaung Suk Lee, “Emerging Challenges in NAND Flash Technology”, Flash Summit 2011 (Hynix)
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
Our Goals:
Build reliable error models for NAND flash memory
Design efficient reliability mechanisms based on the model
NAND Flash Error Model

Experimentally characterize and model dominant errors

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

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., “Error Analysis and Retention-Aware Error Management for NAND Flash Memory”, ITJ 2013
Cai et al., “Data Retention in MLC NAND Flash Memory: Characterization, Optimization and Recovery”, HPCA 2015
Our Goals and Approach

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

**Approach:**
- Solid experimental analyses of errors in real MLC NAND flash memory → drive the understanding and models
- Understanding, models, and creativity → drive the new techniques
Experimental Testing Platform


NAND Flash Error Types

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

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

- Caused by flash cell losing charge over time
  - Retention errors

  - Whether an error happens depends on required retention time
  - Especially problematic in MLC flash because threshold voltage window to determine stored value is smaller
Observations: Flash Error Analysis

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

Yu Cai, Erich F. Haratsch, Onur Mutlu, and Ken Mai, "Error Patterns in MLC NAND Flash Memory: Measurement, Characterization, and Analysis"

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

Yu Cai¹, Erich F. Haratsch², Onur Mutlu¹ and Ken Mai¹
¹Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA
²LSI Corporation, 1110 American Parkway NE, Allentown, PA
¹{yucai, onur, kenmai}@andrew.cmu.edu, ²erich.haratsch@lsi.com

SAFARI
Solution to Retention Errors

- Refresh periodically

- Change the period based on P/E cycle wearout
  - Refresh more often at higher P/E cycles

- Use a combination of in-place and remapping-based refresh

Flash Correct-and-Refresh [ICCD’12]

- 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 Cai¹, Gulay Yalcin², Onur Mutlu¹, Erich F. Haratsch³, Adrian Cristal², Osman S. Unsal² and Ken Mai¹

¹DSSC, Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA  
²Barcelona Supercomputing Center, C/Jordi Girona 29, Barcelona, Spain  
³LSI Corporation, 1110 American Parkway NE, Allentown, PA
## Many Errors and Their Mitigation [PIEEE’17]

### Table 3 List of Different Types of Errors Mitigated by NAND Flash Error Mitigation Mechanisms

<table>
<thead>
<tr>
<th>Mitigation Mechanism</th>
<th>Error Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shadow Program Sequencing [35,40] (Section V-A)</td>
<td></td>
</tr>
<tr>
<td>Neighbor-Cell Assisted Error Correction [36] (Section V-B)</td>
<td></td>
</tr>
<tr>
<td>Refresh [34,39,67,68] (Section V-C)</td>
<td></td>
</tr>
<tr>
<td>Read-Retry [33,72,107] (Section V-D)</td>
<td>X</td>
</tr>
<tr>
<td>Voltage Optimization [37,38,74] (Section V-E)</td>
<td></td>
</tr>
<tr>
<td>Hot Data Management [41,63,70] (Section V-F)</td>
<td>X</td>
</tr>
<tr>
<td>Adaptive Error Mitigation [43,65,77,78,82] (Section V-G)</td>
<td>X</td>
</tr>
</tbody>
</table>

Many Errors and Their Mitigation [PIEEE’17]

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

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

By Yu Cai, Saugata Ghose, Erich F. Haratsch, Yixin Luo, and Onur Mutlu

https://arxiv.org/pdf/1706.08642
One Issue: Read Disturb in Flash Memory

- All scaled memories are prone to read disturb errors
  - DRAM
  - SRAM
- Hard Disks: Adjacent Track Interference
- NAND Flash
NAND Flash Memory Background

Flash Memory

Block 0

Read
Pass
Pass
...
Pass

......

Block N

Flash Controller
Flash Cell Array

Block X
Page Y
Sense Amplifiers

Row
Column
Sense Amplifiers
Flash Cell

Floating Gate Transistor (Flash Cell)

\[ V_{th} = 2.5 \text{ V} \]
Flash Read

$V_{\text{read}} = 2.5 \text{ V}$

$V_{\text{th}} = 2 \text{ V}$

Gate

1

$V_{\text{read}} = 2.5 \text{ V}$

$V_{\text{th}} = 3 \text{ V}$

0
Flash Pass-Through

$V_{pass} = 5\ V$

$V_{th} = 2\ V$

$V_{pass} = 5\ V$

$V_{th} = 3\ V$
Read from Flash Cell Array

\[ V_{\text{pass}} = 5.0 \]

Pass (5V)

\[ V_{\text{read}} = 2.5 \]

Read (2.5V)

\[ V_{\text{pass}} = 5.0 \]

Pass (5V)

\[ V_{\text{pass}} = 5.0 \]

Pass (5V)

Correct values for page 2:

0 0 1 1
Read Disturb Problem: “Weak Programming” Effect

Repeatedly read page 3 (or any page other than page 2)
Read Disturb Problem: “Weak Programming” Effect

High pass-through voltage induces “weak-programming” effect.

Incorrect values from page 2:

High pass-through voltage induces “weak-programming” effect.
Executive Summary [DSN’15]

- **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%
Read Disturb Prone vs. Resistant Cells

PDF

Disturb-Resistant

Disturb-Prone

N read disturbs

N read disturbs

Normalized $V_{th}$
Observation 2: Some Flash Cells Are More Prone to Read Disturb

After 250K read disturbs:

Disturb-prone cells have higher threshold voltages

Disturb-resistant cells have lower threshold voltages

Disturb-prone $\rightarrow$ ER state

Disturb-resistant $\rightarrow$ P1 state
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 ($V_{th} \neq V_{ref} - \sigma < V_{th} < V_{ref} - \sigma$)
  – Predict among these susceptible cells
    • Cells with more $V_{th}$ shifts are disturb-prone $\rightarrow$ Higher $V_{th}$ state
    • Cells with less $V_{th}$ shifts are disturb-resistant $\rightarrow$ Lower $V_{th}$ state

Reduces total error count by up to 36% @ 1M read disturbs
ECC can be used to correct the remaining errors
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 SSD Error Analysis [SIGMETRICS’15]

- First large-scale field study of flash memory errors

  [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
Another Lecture: NAND Flash Reliability

- Yu Cai, Saugata Ghose, Erich F. Haratsch, Yixin Luo, and Onur Mutlu, "Error Characterization, Mitigation, and Recovery in Flash Memory Based Solid State Drives" 

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


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

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

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

In this paper, we demonstrate that two-step programming exposes new reliability and security vulnerabilities. We experi-
Error Characterization, Mitigation, and Recovery in Flash-Memory-Based Solid-State Drives

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

By Yu Cai, Saugata Ghose, Erich F. Haratsch, Yixin Luo, and Onur Mutlu

https://arxiv.org/pdf/1706.08642
Summary: Memory Reliability and Security

- Memory reliability is reducing
- Reliability issues open up security vulnerabilities
  - Very hard to defend against
- Rowhammer is an example
  - Its implications on system security research are tremendous & exciting

**Good news: We have a lot more to do.**

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

Experimentally characterize and model dominant errors

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


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., “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

Cai et al., “Vulnerabilities in MLC NAND Flash Memory Programming: Experimental Analysis, Exploits, and Mitigation Techniques”, HPCA 2017
Threshold Voltage Distribution in MLC NAND Flash Memory: Characterization, Analysis, and Modeling

Yu Cai, Erich F. Haratsch, Onur Mutlu, and Ken Mai,
"Threshold Voltage Distribution in MLC NAND Flash Memory: Characterization, Analysis and Modeling"
Program Interference and Vref Prediction


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

Yu Cai¹, Onur Mutlu¹, Erich F. Haratsch² and Ken Mai¹
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"
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"
  [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
Flash Memory Programming Vulnerabilities

- Yu Cai, Saugata Ghose, Yixin Luo, Ken Mai, Onur Mutlu, and Erich F. Haratsch,
  "Vulnerabilities in MLC NAND Flash Memory Programming: Experimental Analysis, Exploits, and Mitigation Techniques"
  [Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)]

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

Yu Cai†  Saugata Ghose†  Yixin Luo††  Ken Mai†  Onur Mutlu§†  Erich F. Haratsch‡
†Carnegie Mellon University  ‡Seagate Technology  §ETH Zürich
Accurate and Online Channel Modeling

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

[Abstract]