2022 William Carter Award
Recognition and Ceremony

Minesh Patel
DSN’22, Baltimore, MD
28 June 2022
Honored to be a Recipient

• Recognition in honor of William Carter

• I am proud to build upon the direction he set
Thank You!

My Ph.D. adviser

Onur Mutlu
Thank You!

My Ph.D. adviser

Onur Mutlu

Defense committee

Mattan Erez
Moinuddin Qureshi
Vilas Sridharan
Christian Weis

Additional letter writers

Stefan Saroiu
Jared Zerbe
Thank You!

Award sponsors

IEEE TC-FTG
IFIP WG 10.4

Organizers

Conference chairs
Selection committee
Thank You!

Friends
Colleagues
Mentors

SAFARI group
Internships
School (CMU, ETH)
Research community

Family

Parents (Alpa, Hamen)
Sister (Shreya)
Self-Introduction

• Originally from **Houston, TX**
  • Bachelors at UT Austin in 2015 (EE + Physics)

• Ph.D. with **Onur Mutlu** from **ETH Zürich**
  • Started at CMU in 2015
  • Defended October 1, 2021 (graduated April 2022)
  • Focused on **memory systems reliability**

• Currently exploring options for what comes next
  • Broadly interested in **architecture/systems topics**
  • Thinking about work in **industry research**
Dissertation Overview

“Enabling Effective Error Mitigation in Modern Memory Chips that Use On-Die ECC”
Defended Oct. 2021 (ETH Zürich)
Deposited Apr. 2022 (DOI 10.3929/ethz-b-000542542)

Advisor:
Onur Mutlu (ETH Zürich)

Co-Examiners:
Mattan Erez (UT Austin)
Moinuddin Qureshi (Georgia Tech)
Vilas Sridharan (AMD)
Christian Weis (TU Kaiserslautern)
“Separation of Concerns”
between manufacturers

Standardized Interface

Processor

Main Memory (DRAM)

Enables each party to solve their own design challenges
Challenge:
DRAM suffers from errors that cause data loss or system failure if ignored.
Manufacturers’ primary goal is to increase storage density, but this **exacerbates** errors

1. **Increases costs** for manufacturers and consumers
2. **Limits** systems’ overall potential for growth
Solution: Error Mitigation Techniques

Recently, DRAM manufacturers started using on-die error-correcting codes (on-die ECC)

Proprietary and self-contained; Invisible to the processor

Hides the most common errors from the processor (e.g., random single-bit errors)
To Processor

Addressing all errors in DRAM is very expensive

Hides the most common errors from the processor (e.g., random single-bit errors)

Proprietary and self-contained; Invisible to the processor

Simple and low-cost
Preserves trade secrets of DRAM manufacturers
Convenient for many commodity systems

Limited error correction capability for low cost
Partial error-correction can complicate system design and test
Problems Introduced by On-Die ECC

On-die ECC negatively impacts system design and test efforts

Predictable and/or well-understood errors due to physical processes

Unknown filtration from on-die ECC partially correcting the errors

Unpredictable, obfuscated errors that are hard to understand or reason about
Parties Impacted by Obfuscated Errors

• **Anyone** who must understand error characteristics in the course of their work is potentially affected

**Error-Mitigation Designers**
Forced to make **limiting assumptions** (e.g., worst-case behavior) that lead to **inefficient designs**

**Third-Party Testers**
Hard to **debug observed errors** because on-die ECC conceals the **underlying cause**

**Research Scientists**
**Experimental studies** of DRAM technology characteristics polluted by **on-die ECC artifacts**
Thesis Statement

Exploit the interaction between on-die ECC and the statistical characteristics of memory errors.

We can use new memory testing techniques to recover the error characteristics that on-die ECC obfuscates.

Enable scientists and engineers to make informed decisions towards building robust systems.
1.3 Thesis Statement

Our approach is encompassed by the following thesis statement:

The error characteristics that on-die ECC obfuscates can be recovered using new memory testing techniques that exploit the interaction between on-die ECC and the statistical characteristics of memory error mechanisms to expose physical cell behavior, thereby enabling scientists and engineers to make informed decisions towards building smarter and more robust systems.
Core Contributions

1. REAPER (ISCA’17)
   Understand the basic properties of DRAM data-retention errors

2. EIN (DSN’19, best paper)
   Understand and recover the error characteristics beneath on-die ECC

3. BEER (MICRO’20, best paper)
   Determine exactly how on-die ECC obfuscates error characteristics

4. HARP (MICRO’21)
   Understand how errors appear and how to identify at-risk bits

5. Recommendations
   Arguing for increased transparency of DRAM reliability characteristics
Core Contributions

1. **REAPER (ISCA’17)**
   - Understand the basic properties of DRAM data-retention errors

2. **EIN (DSN’19, best paper)**
   - Understand and recover the error characteristics beneath on-die ECC

3. **BEER (MICRO’20, best paper)**
   - Determine exactly how on-die ECC obfuscates error characteristics

4. **HARP (MICRO’21)**
   - Understand how errors appear and how to identify at-risk bits

5. **Recommendations (arXiv’22)**
   - Arguing for increased transparency of DRAM reliability characteristics
DRAM Cell

- **access transistor**
- **storage capacitor**

stores one bit of data

Data Encoding

- "charged" = 1 or 0
  - design-dependent
- "discharged" = 0 or 1

REAPER (ISCA’17)  EIN (DSN’19)  BEER (MICRO’20)  HARP (MICRO’21)  Recs (arXiv’22)
DRAM cells **leak charge** over time

**Fully charged**

**Data-retention error**

**DRAM Refresh**

Periodically restores the charge of all cells to prevent data-retention errors

**Significant performance and energy overhead**
Making Refresh More Efficient

Only a few cells require frequent refreshing

- Fast-leaking
- Slow-leaking

Hard to identify:
1. Process, voltage, temperature
2. Variable retention time
3. Data pattern dependence

Goal: quickly and efficiently identify the error-prone cells
Experimental Error Characterization

• We study the **data-retention error characteristics** in 368 real LPDDR4 DRAM chips

1. Cells are **more likely** to fail at an **increased**
   (1) refresh interval; or (2) temperature

2. Profiling involves a complex **tradeoff space**: 
   (1) **speed**; (2) **coverage**; and (3) **false positives**
Reach Profiling

- Faster
- More reliable
- False positives possible

operate here

profile here

refresh interval

temperature
Evaluating Reach Profiling

1. **2.5x faster** than the state-of-the-art baseline for 99% coverage and a 50% false positive rate
   - *Even faster* (>3.5x) with more false positives (>100%)

2. Enables operating at **longer refresh intervals** by reducing the overall profiling overhead
   - 16.3% end-to-end performance improvement
   - 36.4% DRAM power reduction
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

ABSTRACT
Modern DRAM-based systems suffer from significant energy and latency penalties due to conservative DRAM refresh standards. Volatile DRAM cells can retain information across a wide distribution of times ranging from milliseconds to many minutes, but each cell is currently refreshed every 64ms to account for the extreme tail end of the retention time distribution, leading to a high refresh overhead. Due to poor DRAM technology scaling, this problem is expected to get worse in future device generations. Hence, the current approach of refreshing all cells with the worst-case refresh rate must be replaced with a more intelligent design.

Many prior works propose reducing the refresh overhead by extending the default refresh interval to a higher value, which we refer to as the target refresh interval, across parts or all of a DRAM chip. These proposals handle the small set of failing cells that cannot retain data throughout the entire extended refresh interval via regular refreshes. However, the target refresh interval is set to the maximum one available in the current refresh interval, which results in a large number of regular refreshes to ensure data correctness. This approach not only reduces the overall refresh overhead but also incurs large performance penalties due to the high energy consumption during refresh.

KEYWORDS
DRAM, refresh, retention failures, reliability, testing, memory

1 INTRODUCTION
DRAM stores data in volatile capacitors that constantly leak charge and therefore requires periodic charge restoration to maintain data correctness. As cell capacitor sizes decrease with process scaling and the total number of cells per chip increases each device generation [36], the total amount of time and energy required to restore all cells to their correct value, a process known as DRAM refresh, scales unfavorably [23, 41, 63]. The periodic refresh of DRAM cell capacitors consumes up to 50% of total DRAM power [63] and incurs large performance penalties as DRAM cells are unavailable during refresh [23, 63, 73, 75].

Minesh Patel, Jeremie S. Kim, and Onur Mutlu,
"The Reach Profiler (REAPER): Enabling the Mitigation of DRAM Retention Failures via Profiling at Aggressive Conditions"
Core Contributions

Recommendations

5. Arguing for increased transparency of DRAM reliability characteristics

1. Understand the basic properties of DRAM data-retention errors

2. Understand and recover the error characteristics beneath on-die ECC

3. Determine exactly how on-die ECC obfuscates error characteristics

4. Understand how errors appear and how to identify at-risk bits

5. REAPER (ISCA'17)

HARP (MICRO’21)

EIN (DSN’19, best paper)

BEER (MICRO’20, best paper)

Data Store

To processor

On-Die ECC Logic

DRAM Chip
Third-Party DRAM Users

Study DRAM errors to understand a DRAM chip’s reliability characteristics

- Expected error rates?
- ‘Weak’ cell locations?
- Inter-chip variation?
- Temperature dependence?
- Statistical error properties?
- Minimum operating timings?

System Architects
Test Engineers
Research Scientists

REAPER (ISCA’17)  EIN (DSN’19)  BEER (MICRO’20)  HARP (MICRO’21)  Recs (arXiv’22)
Third-Party DRAM Users

Study DRAM errors to understand a DRAM chip’s reliability characteristics

Gain *exploitable insights* to improve performance, reliability, etc.

- REAPER (ISCA’17)
- EIN (DSN’19)
- BEER (MICRO’20)
- HARP (MICRO’21)
- Recs (arXiv’22)
On-Die ECC Interferes with Studying Errors

No-ECC DRAMs

- $P_0$
- $P_1$
- $P_2$

Test Routine

On-Die ECC DRAMs

- $E_0$
- $E_1$
- $E_2$

Well-Understood Error Distributions
- Based on physical properties of DRAM
- Easy to reason about and understand

Unpredictable Error Distributions
- Dependent on ECC implementation
- Hard to reason about and predict
On-Die ECC Interferes with Studying Errors

No-ECC DRAMs

P0  P1  P2

Test Routine

On-Die ECC DRAMs

E0  E1  E2

Our goal: Recover the error characteristics that on-die ECC obfuscates
Key Idea: Statistical Inference

Known
- Directly observable

Inferable
- By using statistical methods

Unknown
- But predictable based on well-understood DRAM error mechanisms

Diagram:
- CPU
- DRAM Chip
- On-Die ECC
- Error-Prone Data Store

- REAPER (ISCA’17)
- EIN (DSN’19)
- BEER (MICRO’20)
- HARP (MICRO’21)
- Recs (arXiv’22)
EIN: Error Inference Methodology

1. Define Experimental Setup
   e.g., testing parameters, DRAM chips

2. Simulate Suspected ECCs
   e.g., Hamming, BCH, etc.

3. Real-Chip Experiments
   with unknown ECC scheme

Monte-Carlo Simulation
https://github.com/CMU-SAFARI/EINSim

4. Perform Inference
   Maximum-a-priori (MAP) estimation

Most Likely ECC Scheme

Credits:
- REAPER (ISCA’17)
- EIN (DSN’19)
- BEER (MICRO’20)
- HARP (MICRO’21)
- Recs (arXiv’22)
Applying EIN to Real Chips

• Apply EIN to 314 real LPDDR4 DRAM chips

• Show that EIN can infer both:
  • The ECC scheme to be a (136, 128) Hamming code
  • Raw bit error rates of data-retention errors

• EIN works without:
  • Visibility into the ECC mechanism
  • Disabling ECC
  • Tampering with the hardware
Minesh Patel, Jeremie S. Kim, Hasan Hassan, and Onur Mutlu, "Understanding and Modeling On-Die Error Correction in Modern DRAM: An Experimental Study Using Real Devices"

https://github.com/CMU-SAFARI/EINSim
Core Contributions

5 Recommendations
Arguing for increased transparency of DRAM reliability characteristics

1 REAPER (ISCA’17)
Understand the basic properties of DRAM data-retention errors

3 BEER (MICRO’20, best paper)
Determine exactly how on-die ECC obfuscates error characteristics

2 EIN (DSN’19, best paper)
Understand and recover the error characteristics beneath on-die ECC

4 HARP (MICRO’21)
Understand how errors appear and how to identify at-risk bits
Our goal: Determine exactly how on-die ECC obfuscates errors (i.e., its parity-check matrix)

- **BEER**: Reveals how on-die ECC scrambles errors
- **BEEP**: Enables inferring raw bit error locations
Key idea: disabling DRAM refresh induces data-retention errors only in CHARGED cells
Key idea: disabling DRAM refresh induces data-retention errors only in CHARGED cells

We can selectively induce errors by controlling bit-flip directions.
BEER Testing Methodology

1. Induce **uncorrectable data-retention** errors by disabling **DRAM refresh** operations.

2. Identify which **uncorrectable errors** are and are not possible.

3. Solve for the **parity-check matrix** using a **SAT solver**.
Using BEER in Practice

• BEER determines the parity-check matrix **without:**
  1. hardware support or tools
  2. prior knowledge about on-die ECC
  3. access to ECC metadata (e.g., syndromes)

• Open-source C++ tool on GitHub
  
  https://github.com/CMU-SAFARI/BEER
Experimental demonstration
80 LPDDR4 DRAM chips
(3 major manufacturers)

Two-Part Evaluation

Simulation of correctness and practicality
Over 100,000 representative ECC codes
of varying word lengths (4 – 247 bits)
1. Different manufacturers appear to use different parity-check matrices

2. Chips of the same model appear to use identical parity-check matrices

Two-Part Evaluation

1. BEER works for all simulated test cases

2. BEER is practical in both runtime and memory usage
Bit-Exact ECC Recovery (BEER): Determining DRAM On-Die ECC Functions by Exploiting DRAM Data Retention Characteristics

Minesh Patel†  Jeremie S. Kim‡†  Taha Shahroodi†  Hasan Hassan†  Onur Mutlu‡†
†ETH Zürich  ‡Carnegie Mellon University

Increasing single-cell DRAM error rates have pushed DRAM manufacturers to adopt on-die error-correction coding (ECC), which operates entirely within a DRAM chip to improve factory yield. The on-die ECC function and its effects on DRAM reliability are considered trade secrets, so only the manufacturer knows precisely how on-die ECC alters the externally-visible reliability characteristics. Consequently, on-die ECC obstructs third-party DRAM customers (e.g., test engineers, experimental researchers), who typically design, test, and validate systems based on these characteristics.

Entirely within the DRAM chip [39, 76, 120, 129, 138]. On-die ECC is completely invisible outside of the DRAM chip, so ECC metadata (i.e., parity-check bits, error syndromes) that is used to correct errors is hidden from the rest of the system.

Prior works [60, 97, 98, 120, 129, 133, 138, 147] indicate that existing on-die ECC codes are 64- or 128-bit single-error correction (SEC) Hamming codes [44]. However, each DRAM manufacturer considers their on-die ECC mechanism’s design and implementation to be highly proprietary and ensures not to reveal its details in any public documentation, including DRAM

Minesh Patel, Jeremie S. Kim, Taha Shahroodi, Hasan Hassan, and Onur Mutlu, "Bit-Exact ECC Recovery (BEER): Determining DRAM On-Die ECC Functions by Exploiting DRAM Data Retention Characteristics"
https://github.com/CMU-SAFARI/BEER
Core Contributions

5. Recommendations
   Arguing for increased transparency of DRAM reliability characteristics

1. REAPER (ISCA’17)
   Understand the basic properties of DRAM data-retention errors

3. BEER (MICRO’20, best paper)
   Determine exactly how on-die ECC obfuscates error characteristics

2. EIN (DSN’19, best paper)
   Understand and recover the error characteristics beneath on-die ECC

4. HARP (MICRO’21)
   Understand how errors appear and how to identify at-risk bits

To processor

DRAM Chip

On-Die ECC Logic

Data Store

REAPER (ISCA’17)  EIN (DSN’19)  BEER (MICRO’20)  HARP (MICRO’21)  Recs (arXiv’22)
Profiling a Memory Chip with On-Die ECC

Unreliable Memory

Profiler

On-Die ECC

Data Store

Which bits are at risk of error?

On-die ECC changes how errors appear to the profiler

Goal: understand and address any challenges that on-die ECC introduces for error profiling
Challenges Introduced by On-Die ECC

1. Exponentially increases the total number of at-risk bits

2. Makes it harder to identify individual at-risk bits

3. Interferes with commonly-used data patterns for memory testing
Key Observation: Two Sources of Errors

1. Direct error
   Due to errors in the storage array

2. Indirect error
   Artifact of the on-die ECC algorithm

Upper-bounded by the ECC algorithm
Key Observation: Two Sources of Errors

1. Direct error
   Due to errors in the storage array

2. Indirect error
   Artifact of the on-die ECC algorithm

Key Idea: decouple profiling for direct and indirect errors

Upper-bounded by the ECC algorithm
Hybrid Active-Reactive Profiling (HARP)

1. **Active Profiling**
   - Quickly identifies all direct errors with existing profiling techniques using an on-die ECC bypass path.

2. **Reactive Profiling**
   - Safely identifies indirect errors using secondary ECC at least as strong as on-die ECC.
Hybrid Active-Reactive Profiling (HARP)

1. Active Profiling
Quickly identifies direct errors with existing profiling techniques using an on-die ECC bypass path

2. Reactive Profiling
Safely identifies indirect errors using secondary ECC at least as strong as on-die ECC

HARP reduces the problem of profiling with on-die ECC to profiling without on-die ECC
Evaluations

1. HARP improves **coverage** and **performance** relative to two state-of-the-art baseline profiling algorithms
   
   • E.g., **20.6-62.1% faster** to achieve 99\(^{th}\)-percentile coverage for 2-5 raw-bit errors per on-die ECC word

2. HARP **outperforms** the best-performing baseline in a case study of mitigating data-retention errors
   
   • E.g., **3.7x faster** given a per-bit error probability of 0.75

**We conclude that HARP **overcomes** all three profiling challenges**
https://github.com/CMU-SAFARI/HARP
Core Contributions

REAPER (ISCA’17)
Understand the basic properties of DRAM data-retention errors

EIN (DSN’19, best paper)
Understand and recover the error characteristics beneath on-die ECC

BEER (MICRO’20, best paper)
Determine exactly how on-die ECC obfuscates error characteristics

HARP (MICRO’21)
Understand how errors appear and how to identify at-risk bits

Recommendations
Arguing for increased transparency of DRAM reliability characteristics
Many Ways to Exploit Commodity DRAM

Processor

Software

< Optimize Here >

Reduce timing/voltage margins
  e.g., Access and refresh timings

Use system-level error mitigations
  e.g., ECC, redundancy, replication

Use security enhancements
  e.g., RowHammer and Cold-Boot defenses

Cost
Security
Reliability
Performance
Energy/Power

REAPER (ISCA’17)  EIN (DSN’19)  BEER (MICRO’20)  HARP (MICRO’21)  Recs (arXiv’22)
Many Ways to Exploit Commodity DRAM

Unfortunately, adopting these proposals typically relies on unavailable information about DRAM reliability characteristics (e.g., design characteristics, testing practices, error behavior).

- Use system-level error mitigations e.g., ECC, redundancy, replication
- Use security enhancements e.g., RowHammer and Cold-Boot defenses
- Reduce timing/voltage margins e.g., Access and refresh timings
- Use security enhancements e.g., RowHammer and Cold-Boot defenses
Source of the Problem

• Commodity DRAM specifications do not provide this information by design

However, the **opportunity cost** of preserving this status quo is **increasing**

• Technology scaling **exacerbates** refresh, RowHammer, etc.
• Many old and new proposals for **leveraging this opportunity**
Source of the Problem

- Commodity DRAM specifications do not provide this information by design

Proposal: revisit DRAM specifications to improve information transparency
Two-Step Plan for Transparency

• **No change** to DRAM hardware or design
  • Just provide **information** so that system designers can make better **informed decisions** and **reason about** their designs

1. **Short-term: convey basic information**
   • Whatever the manufacturers feel is practical to do so
   • Possibly develop a crowdsourced database
   • E.g., basic design properties that can be reverse-engineered

2. **Long-term: rethink DRAM standards**
   • Incorporate transparency of reliability-related topics
   • E.g., error models, testing guidelines
A Case for Transparent Reliability in DRAM Systems

Minesh Patel† Taha Shahroodi†† Aditya Manglik† A. Giray Yağlıkçı†
Ataberk Olgun† Haocong Luo† Onur Mutlu†
†ETH Zürich ‡TU Delft

Mass-produced commodity DRAM is the preferred choice of main memory for a broad range of computing systems due to its favorable cost-per-bit. However, today’s systems have diverse system-specific needs (e.g., performance, energy, reliability) that are difficult to address using one-size-fits-all general-purpose DRAM. Unfortunately, although system designers can theoretically adapt commodity DRAM chips to meet their particular design goals (e.g., by exploiting slack in access timings to improve performance, or implementing system-level RowHammer mitigations), we observe that designers today lack the necessary insight into commodity DRAM chips’ reliability characteristics to implement these techniques in practice.

To ensure that system designers can integrate commodity DRAM chips from any manufacturer, the DRAM interface and operating characteristics have long been standardized by the JEDEC consortium [8]. JEDEC maintains a limited set of DRAM standards for commodity DRAM chips with different target applications, e.g., general-purpose DDRn [9–11], bandwidth-

Core Contributions

1. **REAPER (ISCA’17)**
   - Understand the basic properties of DRAM data-retention errors

2. **EIN (DSN’19, best paper)**
   - Understand and recover the error characteristics beneath on-die ECC

3. **BEER (MICRO’20, best paper)**
   - Determine exactly how on-die ECC obfuscates error characteristics

4. **HARP (MICRO’21)**
   - Understand how errors appear and how to identify at-risk bits

5. **Recommendations (Ongoing)**
   - Arguing for increased transparency of DRAM reliability characteristics

Diagram:
- DRAM Chip
  - On-Die ECC Logic
  - Data Store

Flow:
- To processor
- From processor
In a general sense in the recommendations, we can use new memory testing techniques to recover the error characteristics that on-die ECC obfuscates.
Future Research Directions

• Extending the techniques that we propose
  • Other ECC types and error mechanisms
  • Emerging ECC architectures and memory technologies

• Using the information that our techniques reveal
  • Improved system-level error mitigation mechanisms
  • Better diagnostic techniques for errors in the field

• Devising alternatives to on-die ECC
  • Different on-die ECC architectures
  • Cooperative on-die ECC and secondary error mitigation

• Improving transparency of DRAM reliability
## Other Significant Works

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Near-Data Processing</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Boroumand+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Hajinazar+</td>
</tr>
<tr>
<td></td>
<td>CAL’16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ASPLOS’21</td>
</tr>
<tr>
<td><strong>Reducing Memory Access Latency</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Kim+</td>
<td>Wang+</td>
<td>Hassan+</td>
<td>Wang+</td>
<td>Luo+</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ICCD’18</td>
<td>MICRO’18</td>
<td>ISCA’19</td>
<td>MICRO’20</td>
<td>ISCA’20</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Security Topics</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Kim+</td>
<td>Kim+</td>
<td>Orosa+</td>
<td>Olgun+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>HPCA’18</td>
<td>HPCA’19</td>
<td>ISCA’21</td>
<td>ISCA’21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Virtual Memory</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Hajinazar+</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ISCA’20</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Understanding RowHammer</strong></td>
<td></td>
<td></td>
<td></td>
<td>Cojocar+</td>
<td></td>
<td></td>
<td>Yağlıkçı+</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>S&amp;P'20</td>
<td>Kim+</td>
<td>Orosa+</td>
<td>DSN’22</td>
</tr>
<tr>
<td><strong>Mitigating RowHammer</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yağlıkçı+</td>
<td>HPCA'21</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2022 William Carter Award Recognition and Ceremony

Minesh Patel
DSN’22, Baltimore, MD
28 June 2022


