Enabling Effective Error Mitigation in Modern Memory Chips that Use On-Die ECC

Minesh Patel

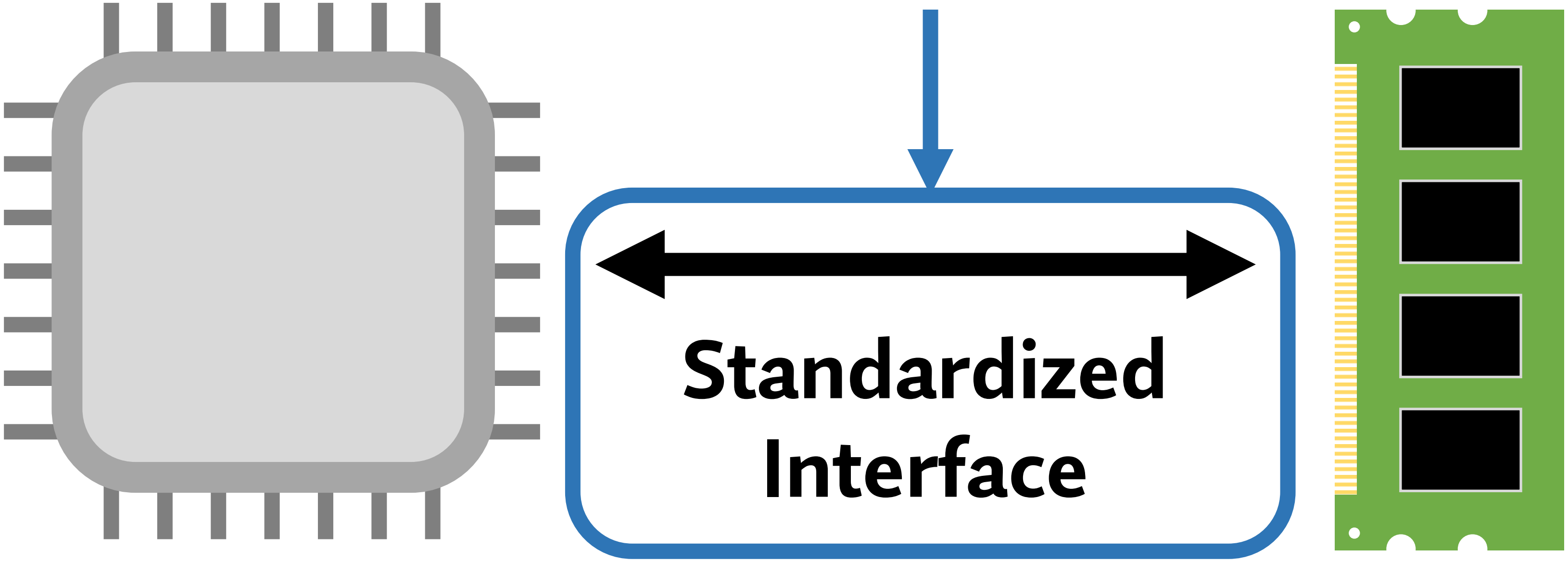
Doctoral Examination

1 October 2021

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

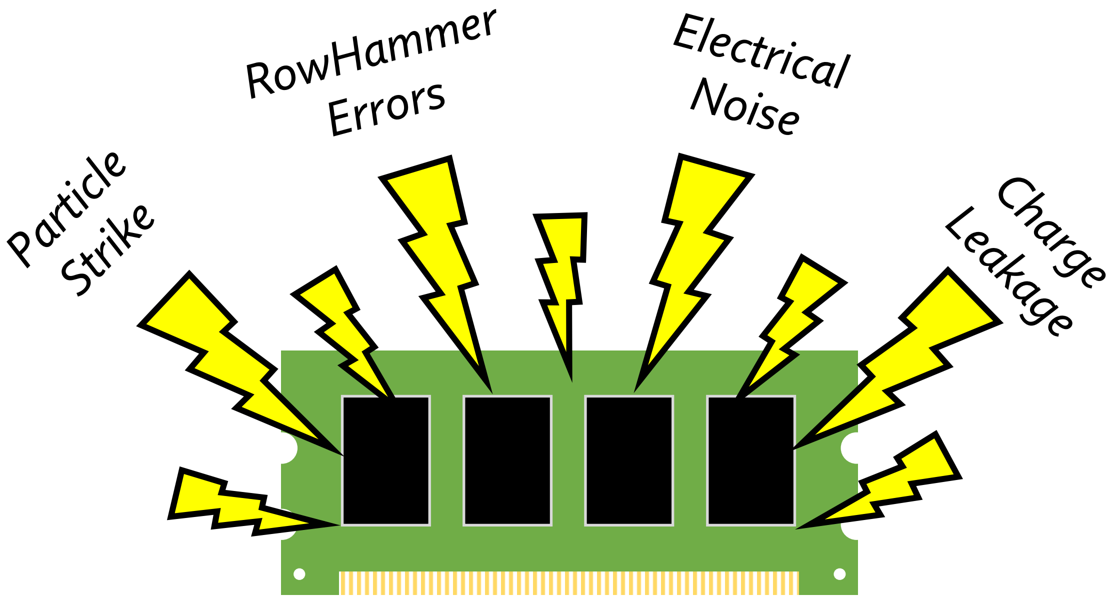


Standardized Interface

Processor

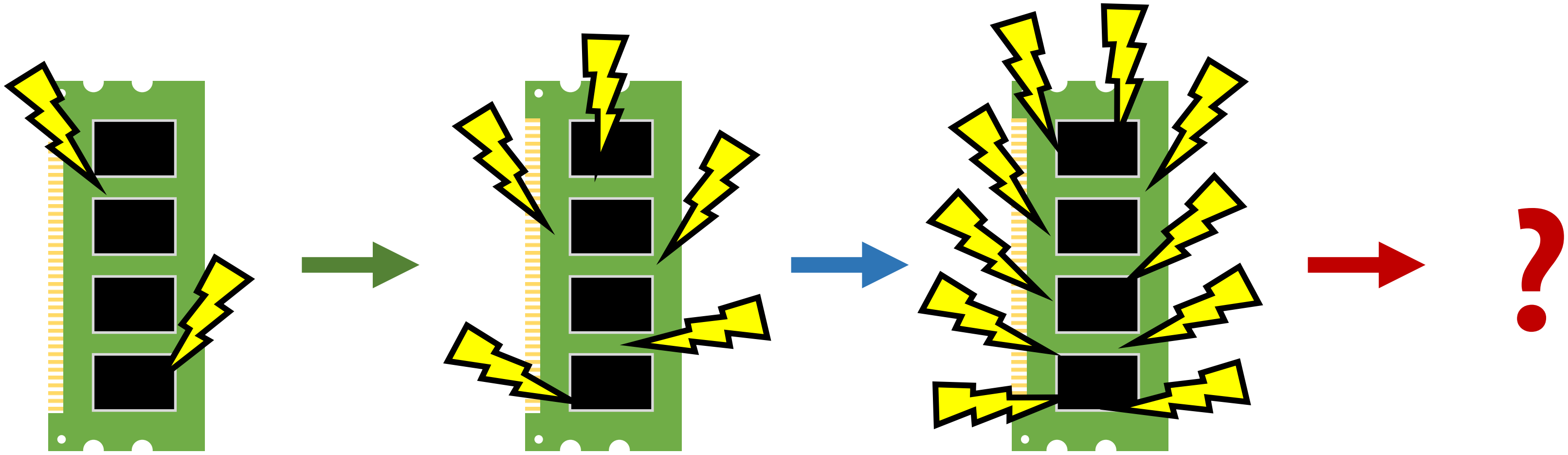
Main Memory (DRAM)

Enables each party to **solve** their own **design challenges**



Challenge:

DRAM suffers from **errors** that cause **data loss** or **system failure** if ignored



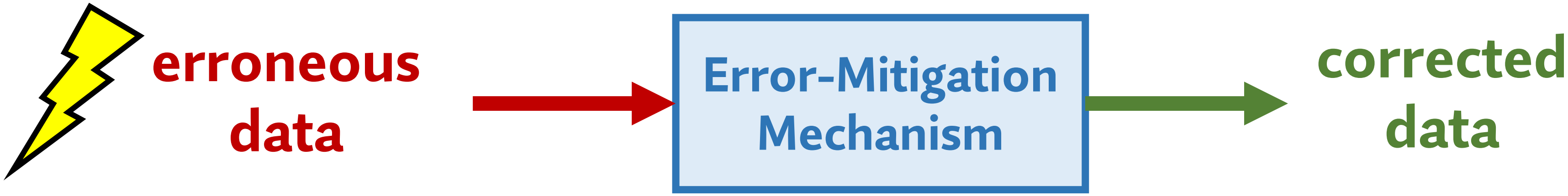
1 Mb *1985* → **512 Mb** *2000* → **32 Gb** *2020* → ?

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

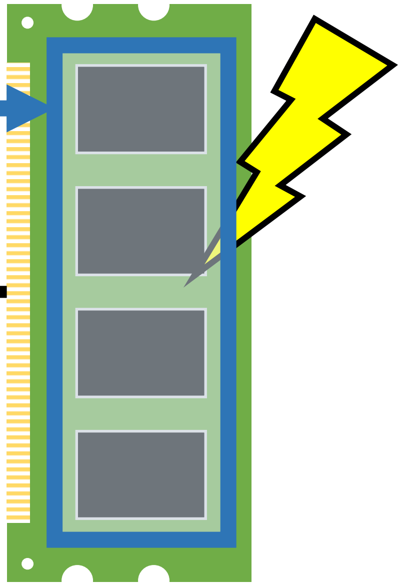


Recently, DRAM manufacturers started using on-die error-correcting codes **(on-die ECC)**

Proprietary *and* ***self-contained****;*
Invisible *to the processor*

To Processor

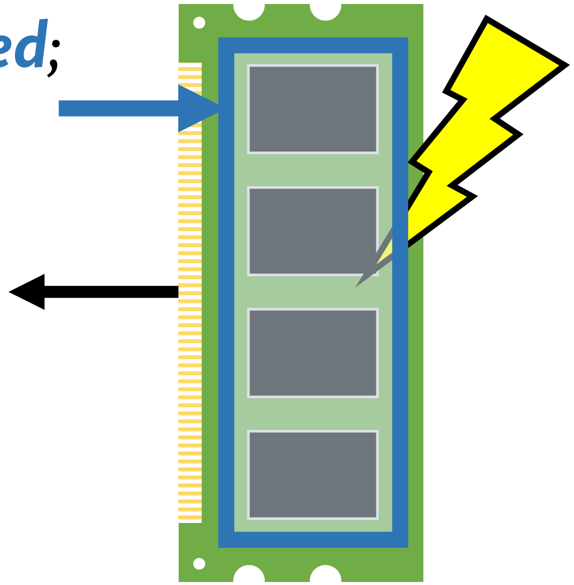
Hides the ***most common*** *errors from the processor*
(e.g., random single-bit errors)



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

- ✗ Maintaining low costs means a **limited** error correction capability
- ✗ Partial error-correction can **complicate** system design and test

Proprietary *and* ***self-contained***;
Invisible *to the processor*

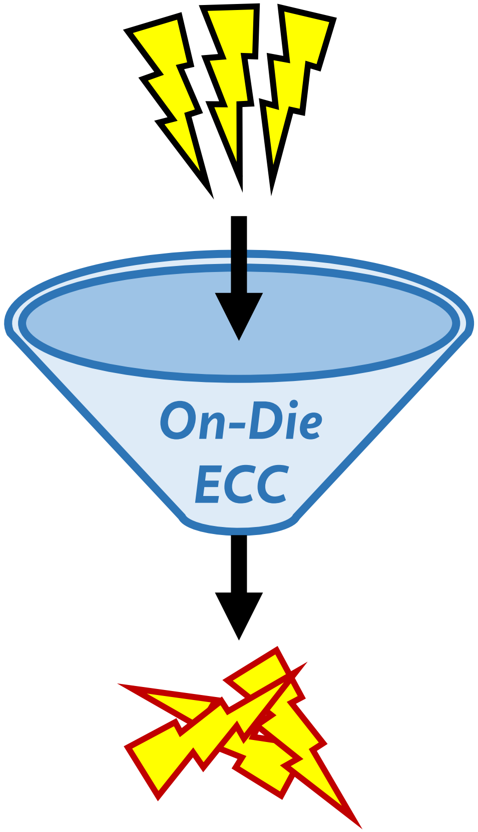


Addressing **all errors**
in DRAM is **very expensive**

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

Problems Introduced by On-Die ECC

On-die ECC negatively impacts system **design** and **test** efforts



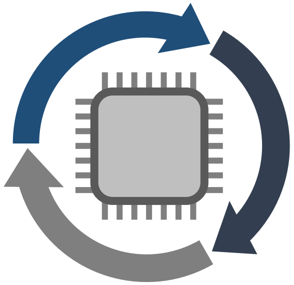
Predictable and/or **well-understood** errors due to **physical processes**

Unknown filtration from on-die ECC partially correcting the errors

Unpredictable, **obfuscated** errors that are hard to **understand** or **reason about**

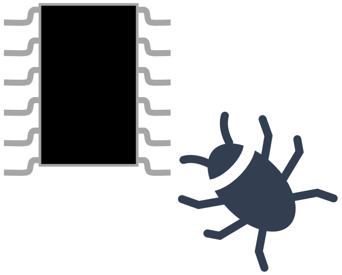
Parties Impacted by Obfuscated Errors

- **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**



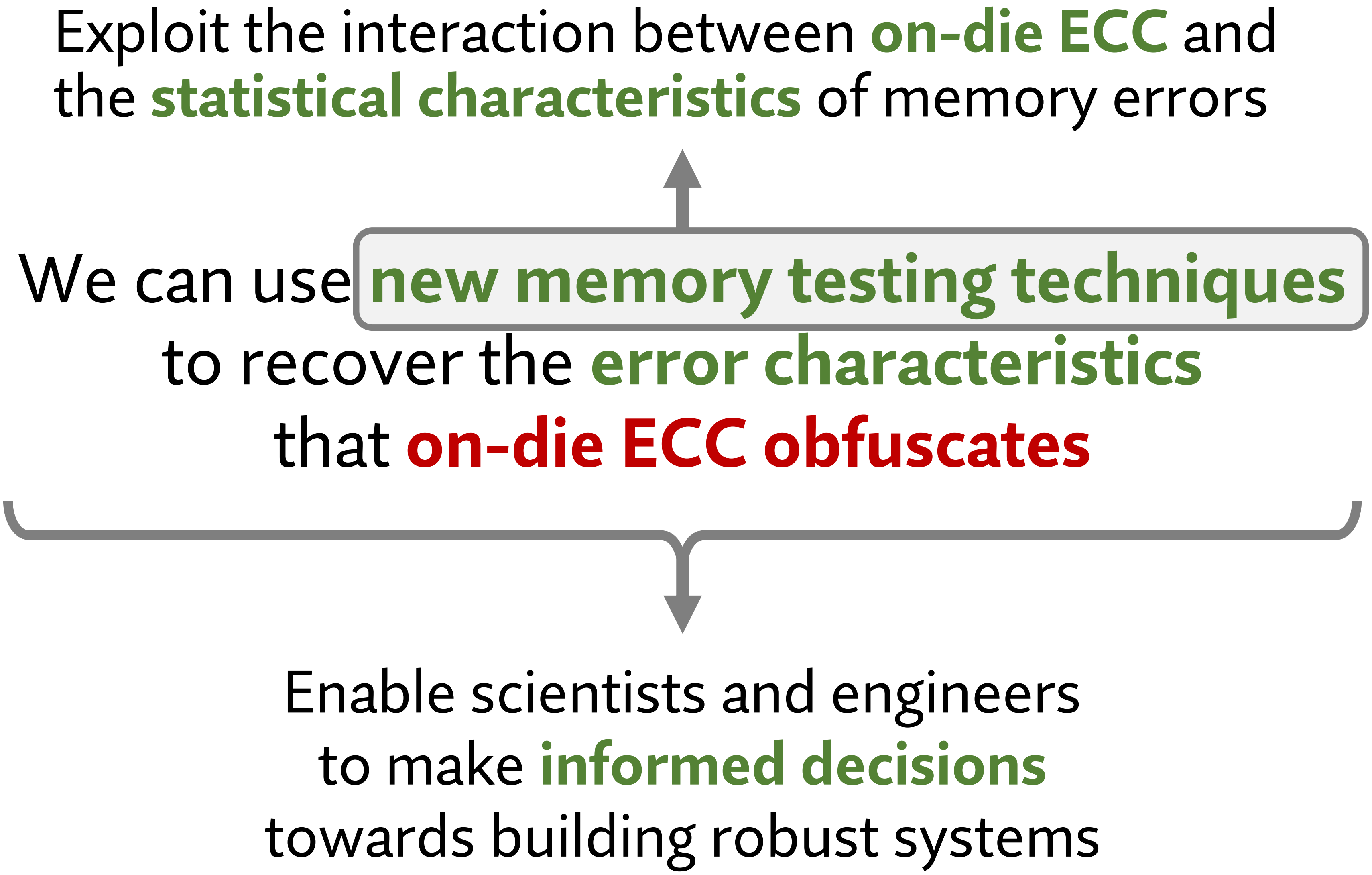
Third-Party Testers

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

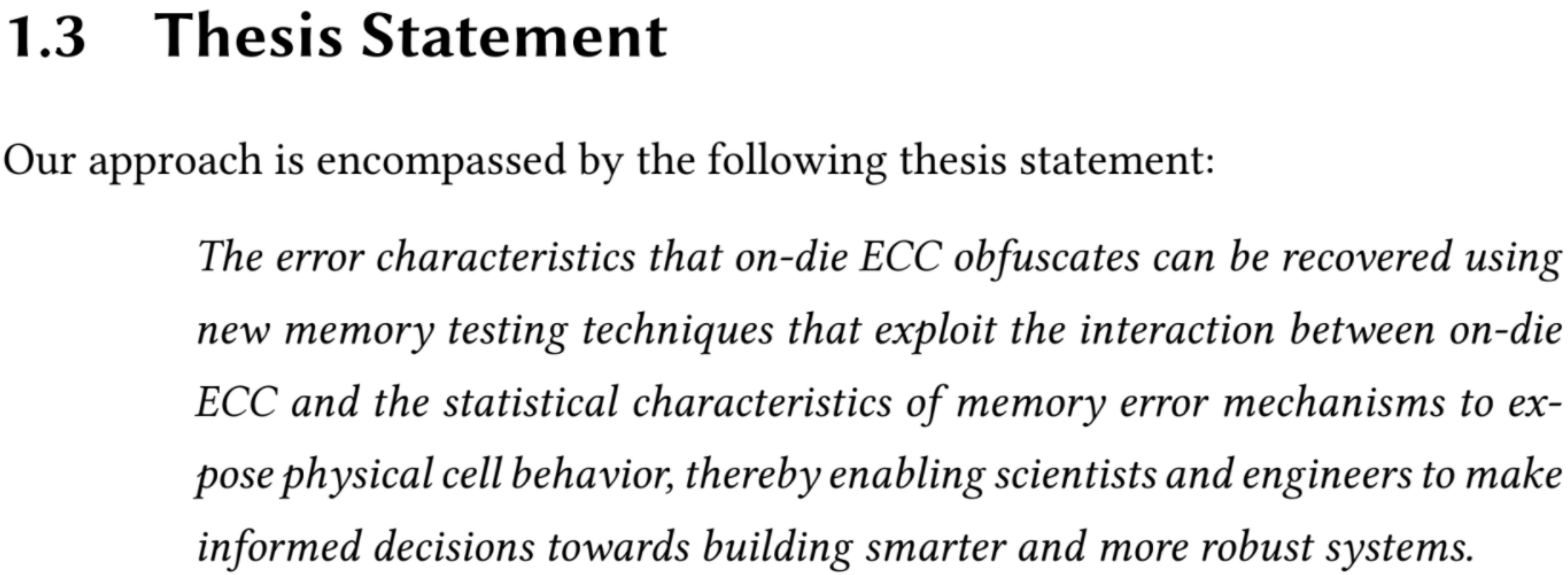


Exploit the interaction between **on-die ECC** and the **statistical characteristics** of memory errors

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

Thesis Statement (Verbatim)



1.3 Thesis Statement

Our approach is encompassed by the following thesis statement:

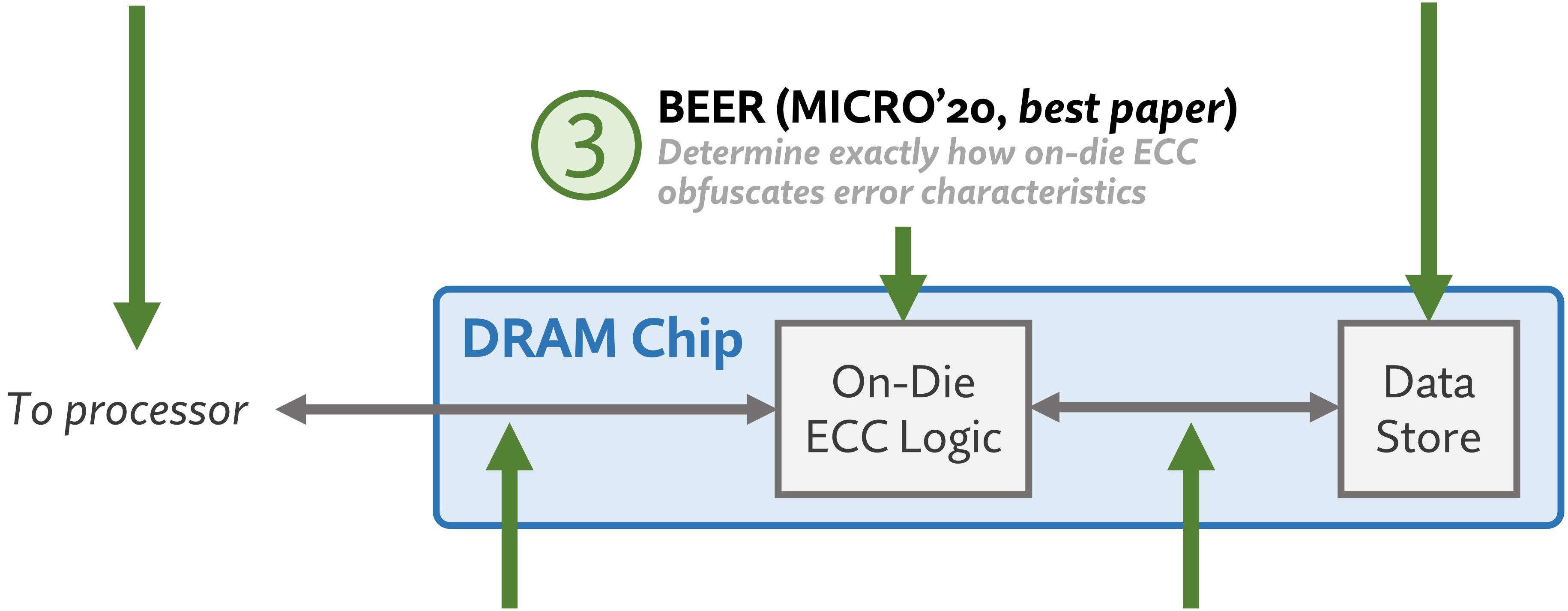
> *The error characteristics that on-die ECC obfuscates can be recovered using new memory testing techniques that exploit the interaction between on-die ECC and the statistical characteristics of memory error mechanisms to expose physical cell behavior, thereby enabling scientists and engineers to make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

Core Contributions

5 **Recommendations**
Arguing for increased transparency of DRAM reliability characteristics

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

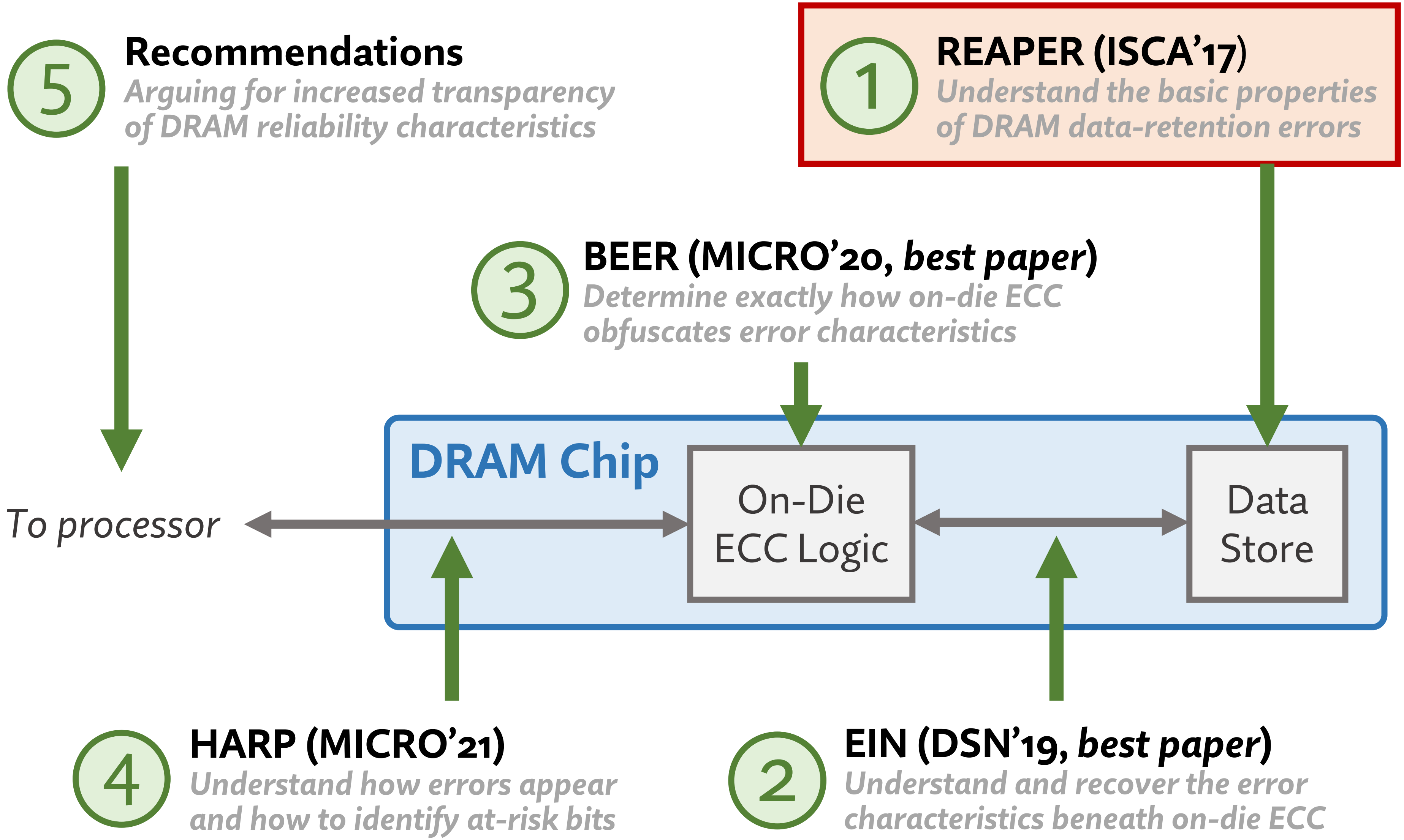
3 **BEER (MICRO’20, *best paper*)**
Determine exactly how on-die ECC obfuscates error characteristics



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

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

Core Contributions



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

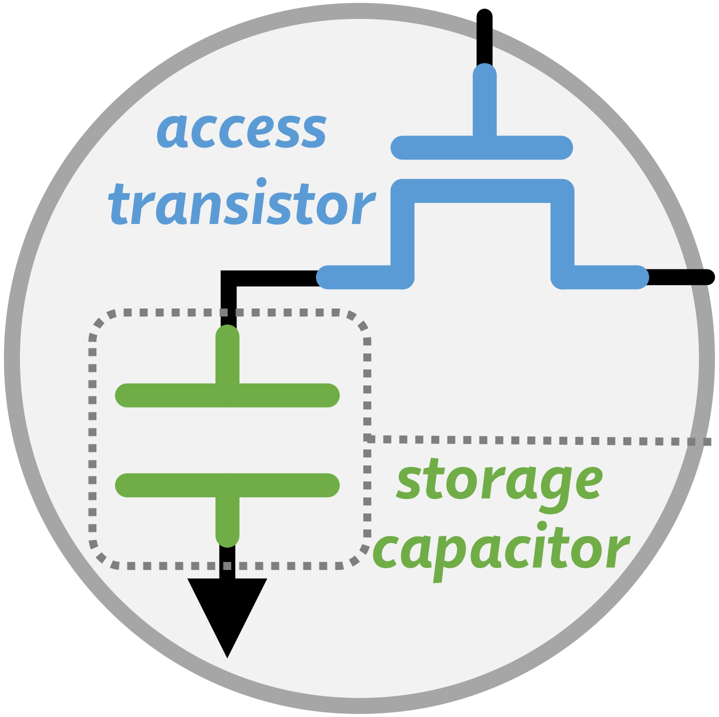
2 **EIN (DSN'19, *best paper*)**
Understand and recover the error characteristics beneath on-die ECC

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

DRAM Cell



stores one bit of data

Data Encoding

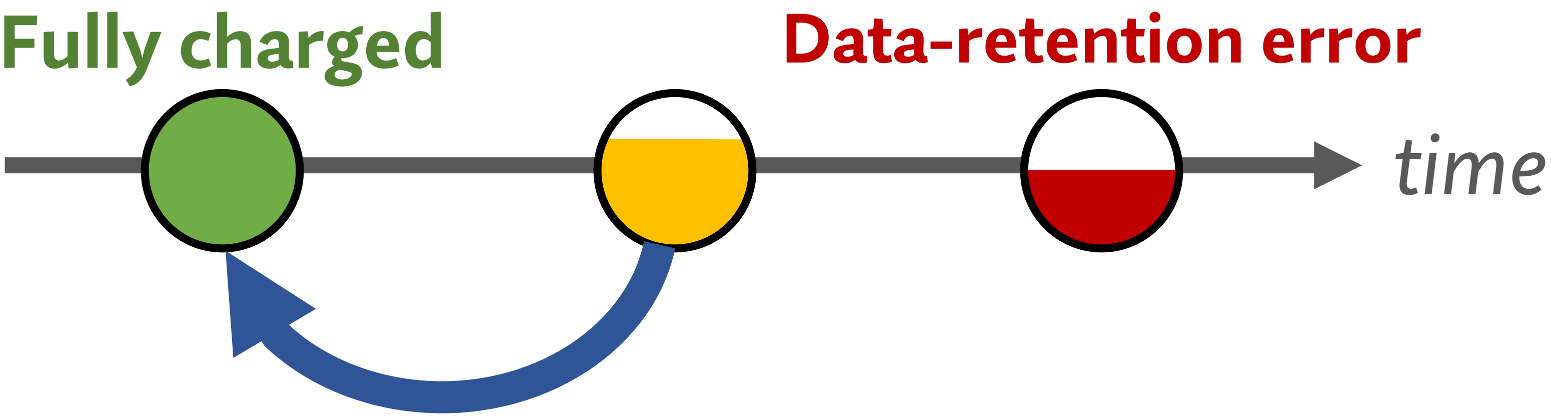
“charged”

● = $\underbrace{\texttt{1 or 0}}_{\textit{design-dependent}}$

“discharged”

○ = `0 or 1`

DRAM cells **leak charge** over time



DRAM Refresh

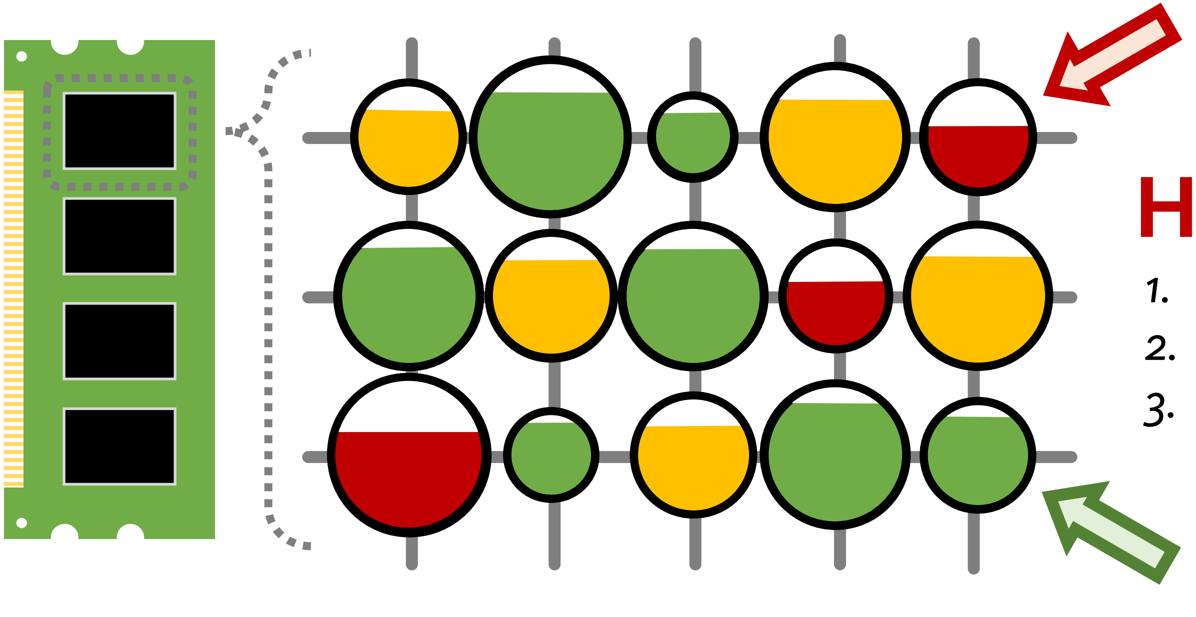
Periodically restores the charge of all cells
to prevent data-retention errors

⇩

Significant performance
and energy overhead

Making Refresh More Efficient

Only a **few cells** require frequent refreshing



fast-leaking

Hard to identify

1. *Process, voltage, temperature*
2. *Variable retention time*
3. *Data pattern dependence*

slow-leaking

Goal: quickly and **efficiently** identify the **error-prone** cells

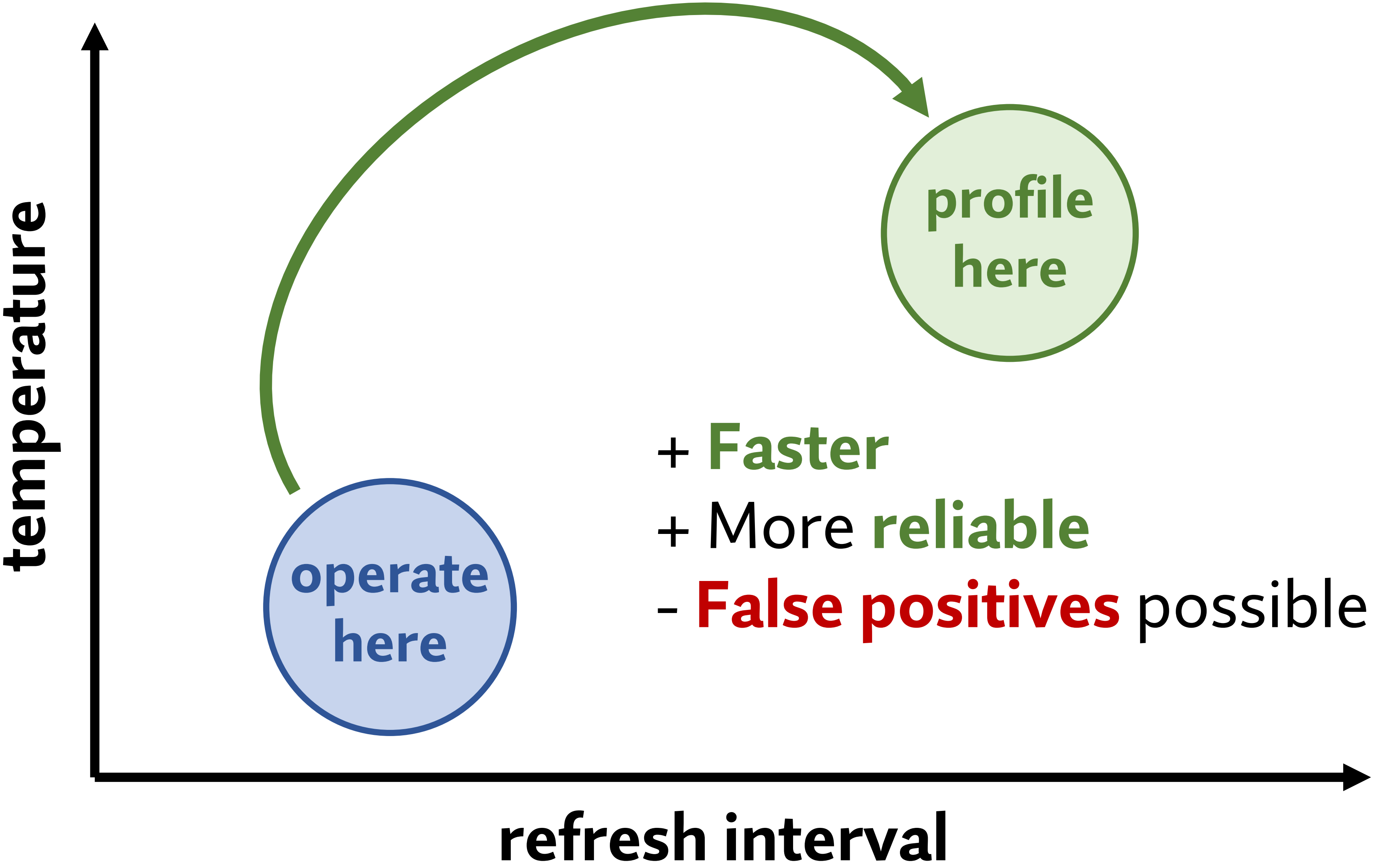
Experimental Error Characterization

- 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**

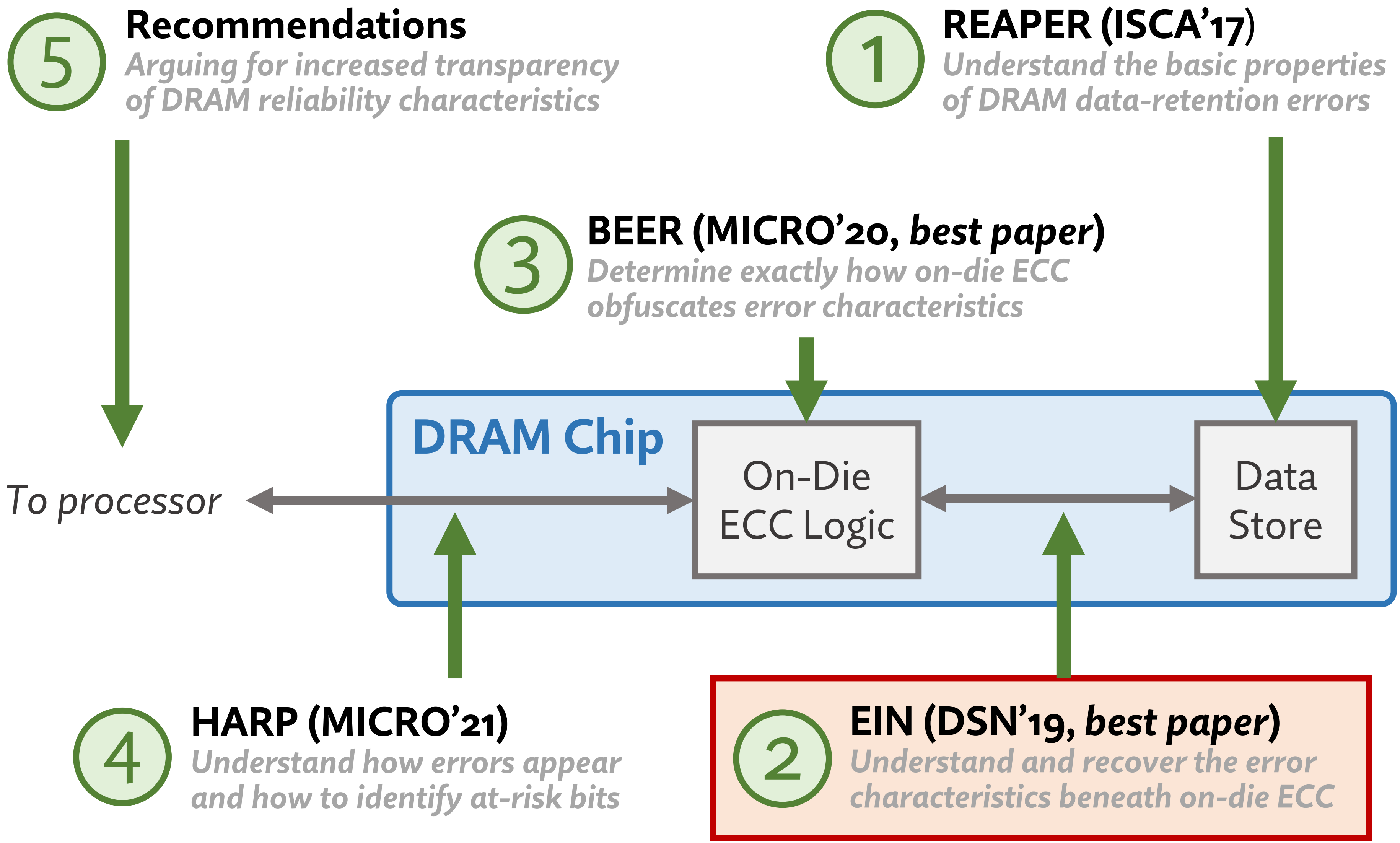
- **False positives** possible

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

Core Contributions



5 **Recommendations**
Arguing for increased transparency of DRAM reliability characteristics

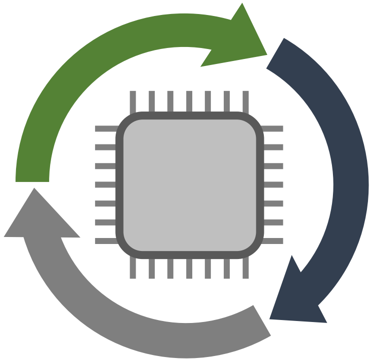
1 **REAPER (ISCA’17)**
Understand the basic properties of DRAM data-retention errors

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

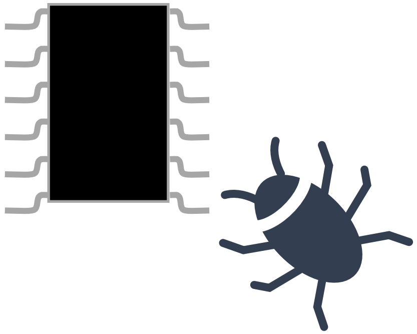
4 **HARP (MICRO’21)**
Understand how errors appear and how to identify at-risk bits

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

Third-Party DRAM Users



System Architects



Test Engineers

Research Scientists

Study **DRAM errors** to understand a DRAM chip's **reliability characteristics**

Expected error rates?

'Weak' cell locations?

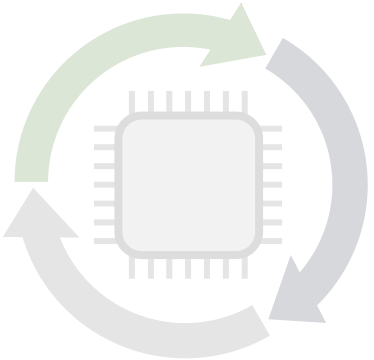
Inter-chip variation?

Temperature dependence?

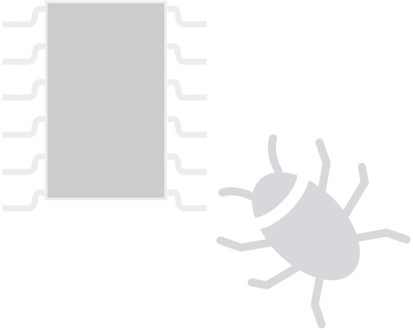
Statistical error properties?

Minimum operating timings?

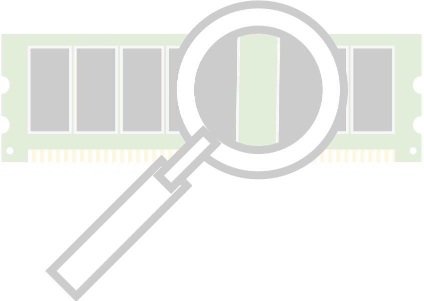
Third-Party DRAM Users



System Architects



Test Engineers

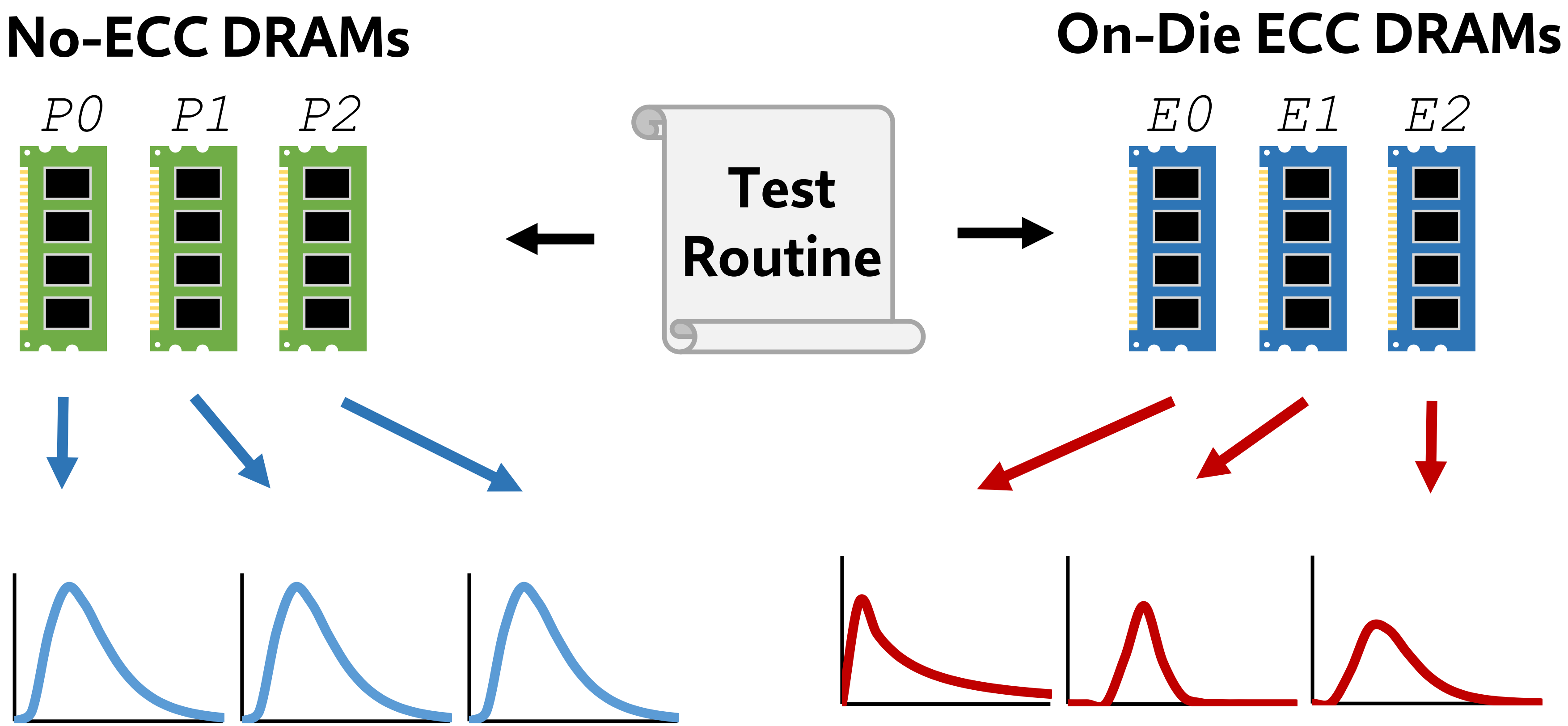


Research Scientists

Study **DRAM errors** to understand a DRAM chip's **reliability characteristics**

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

On-Die ECC Interferes with Studying Errors



No-ECC DRAMs

On-Die ECC DRAMs

Test Routine

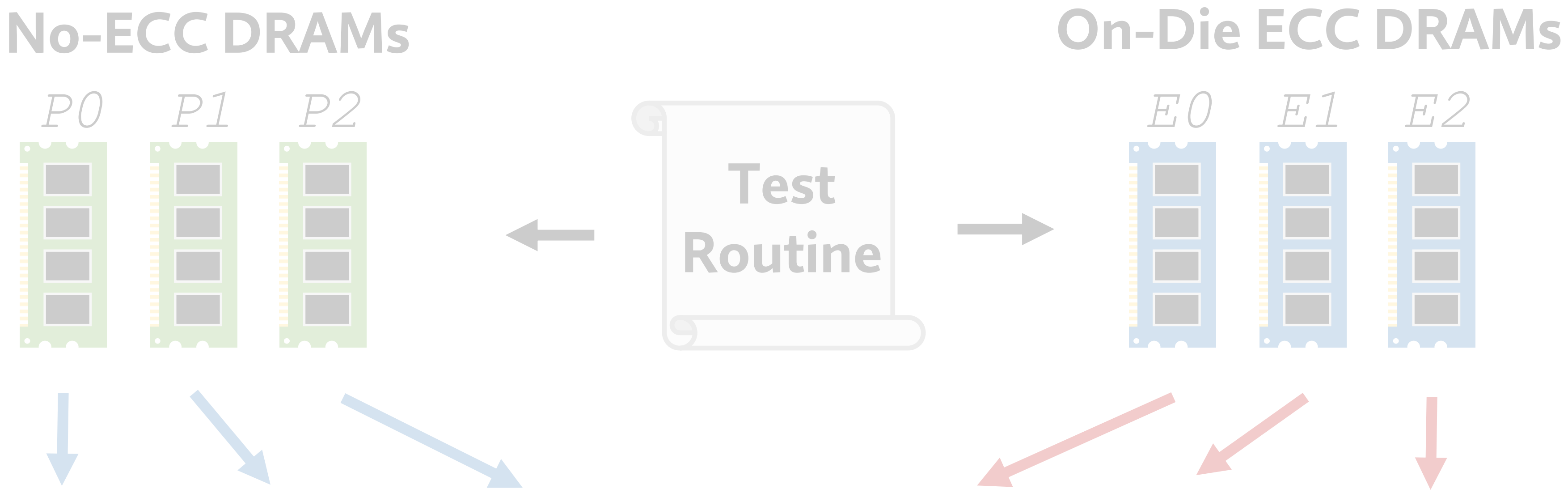
Well-Understood Error Distributions

- *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*

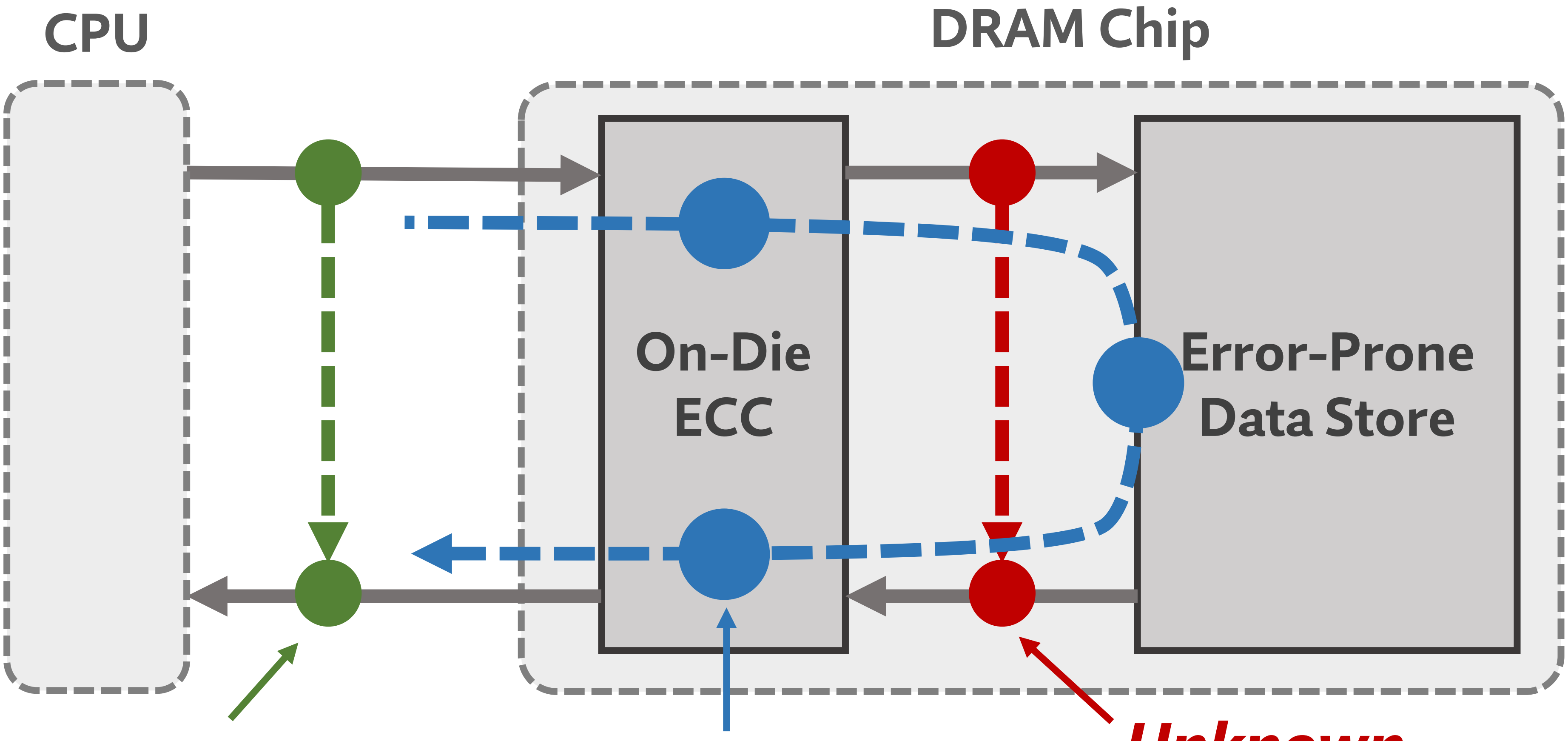
On-Die ECC Interferes with Studying Errors



Our goal:

Recover the **error characteristics** that on-die ECC **obfuscates**

Key Idea: Statistical Inference

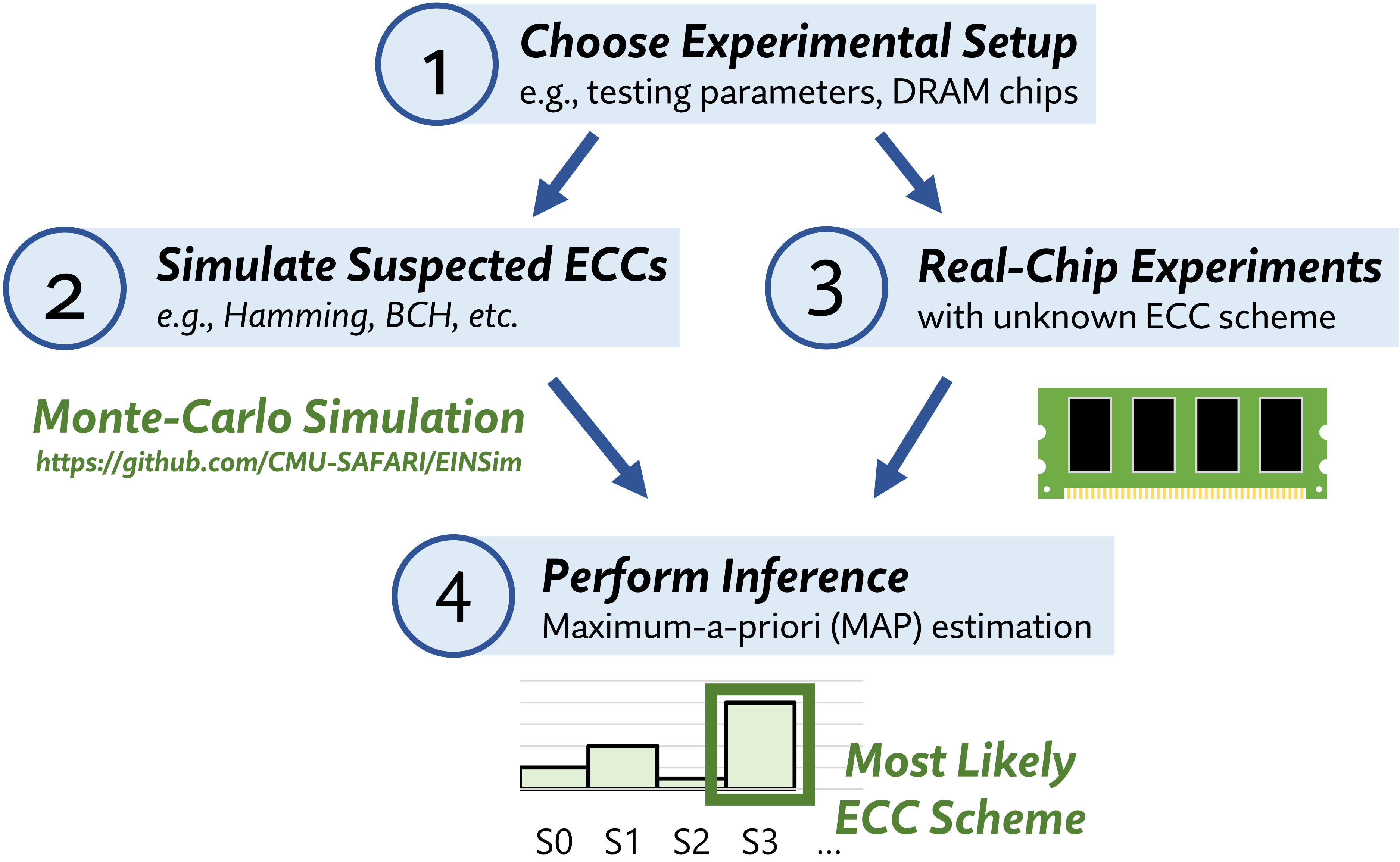


Known
Directly observable

Inferable
By using
statistical methods

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

EIN: Error Inference Methodology



1 **Choose 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

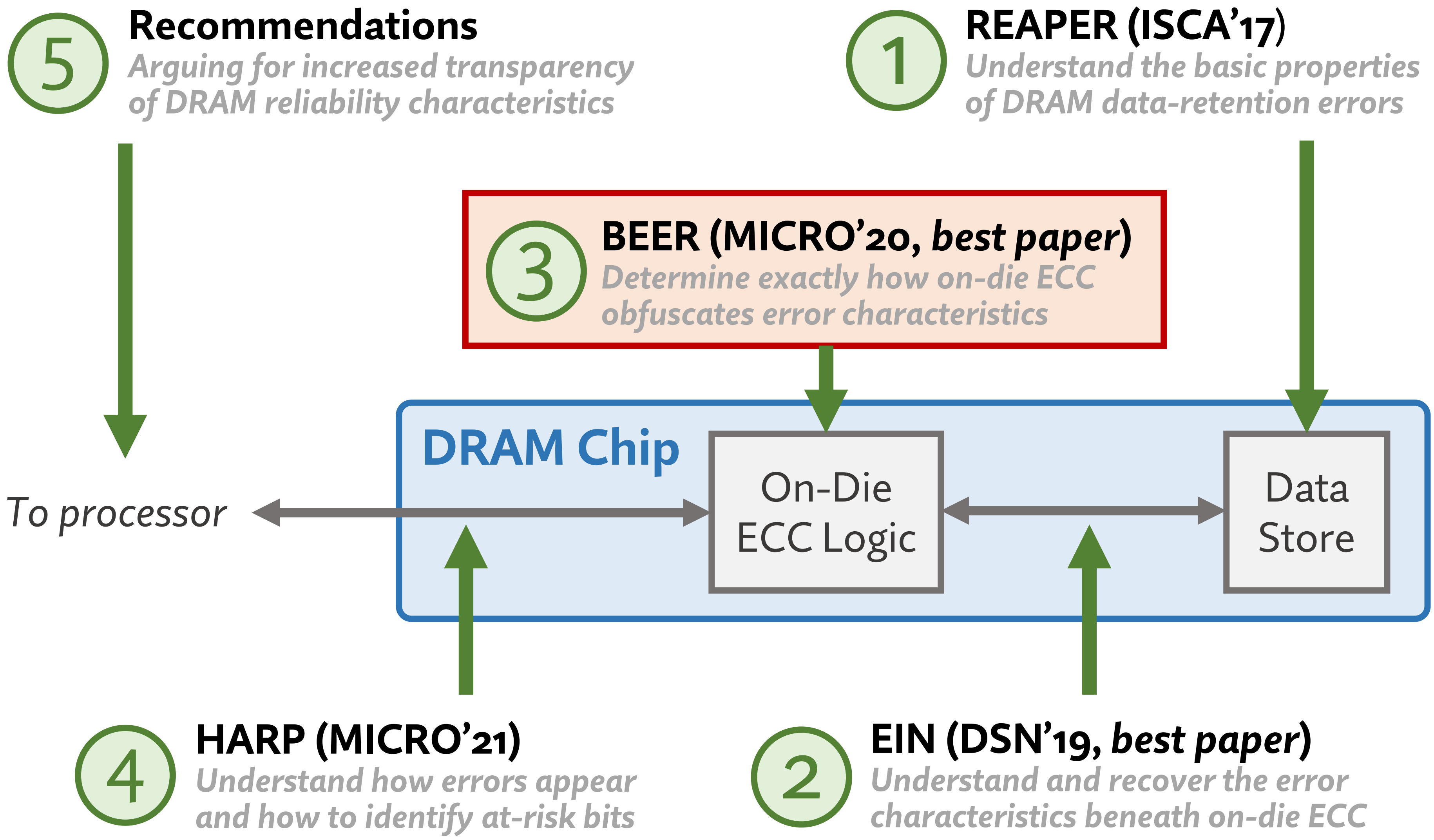
Applying EIN to Real Chips

- 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

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

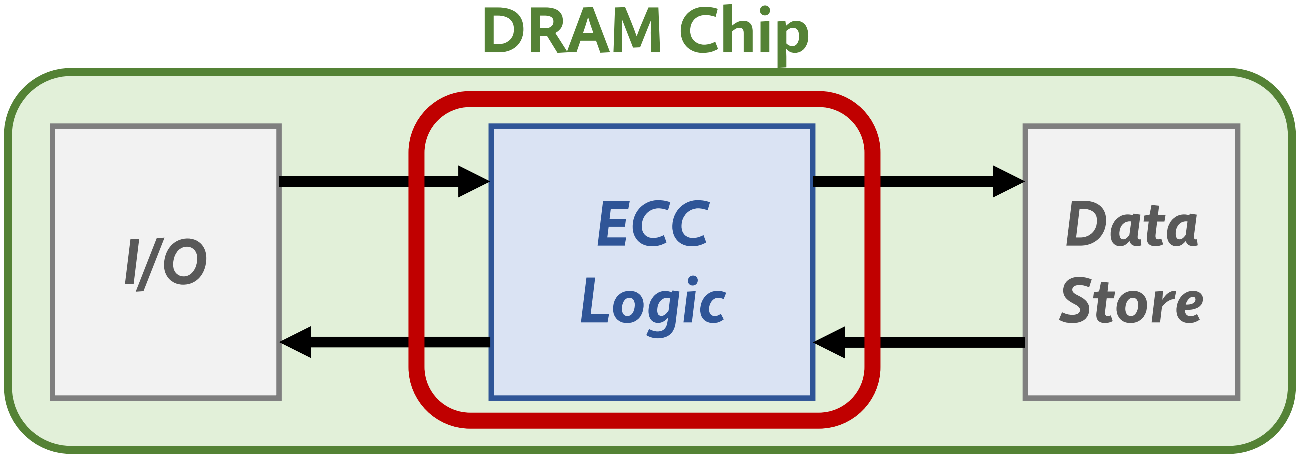
3 **BEER (MICRO’20, *best paper*)**
Determine exactly how on-die ECC obfuscates error characteristics

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

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

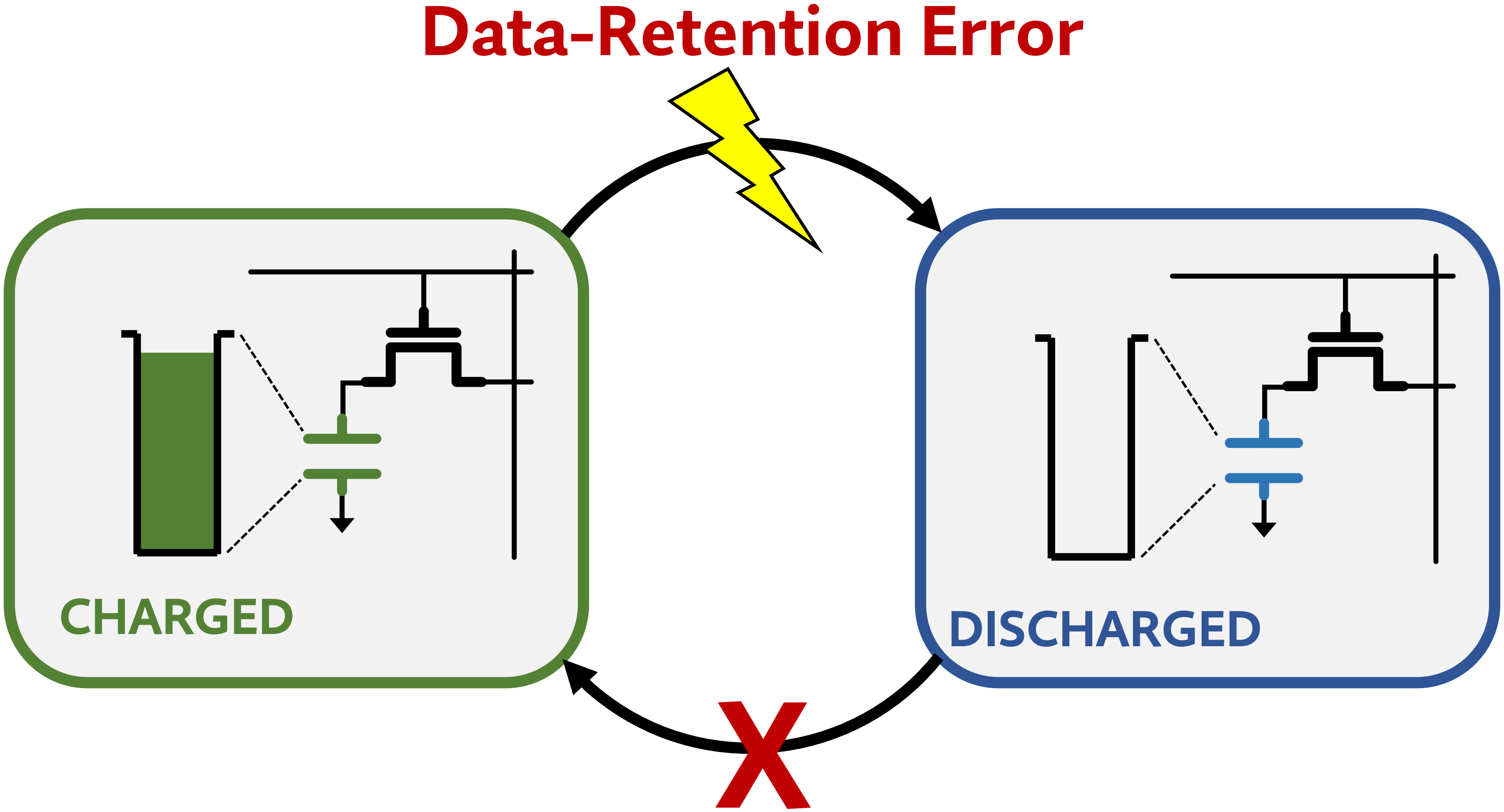
Our goal:

Determine **exactly how** on-die ECC obfuscates errors (i.e., its parity-check matrix)

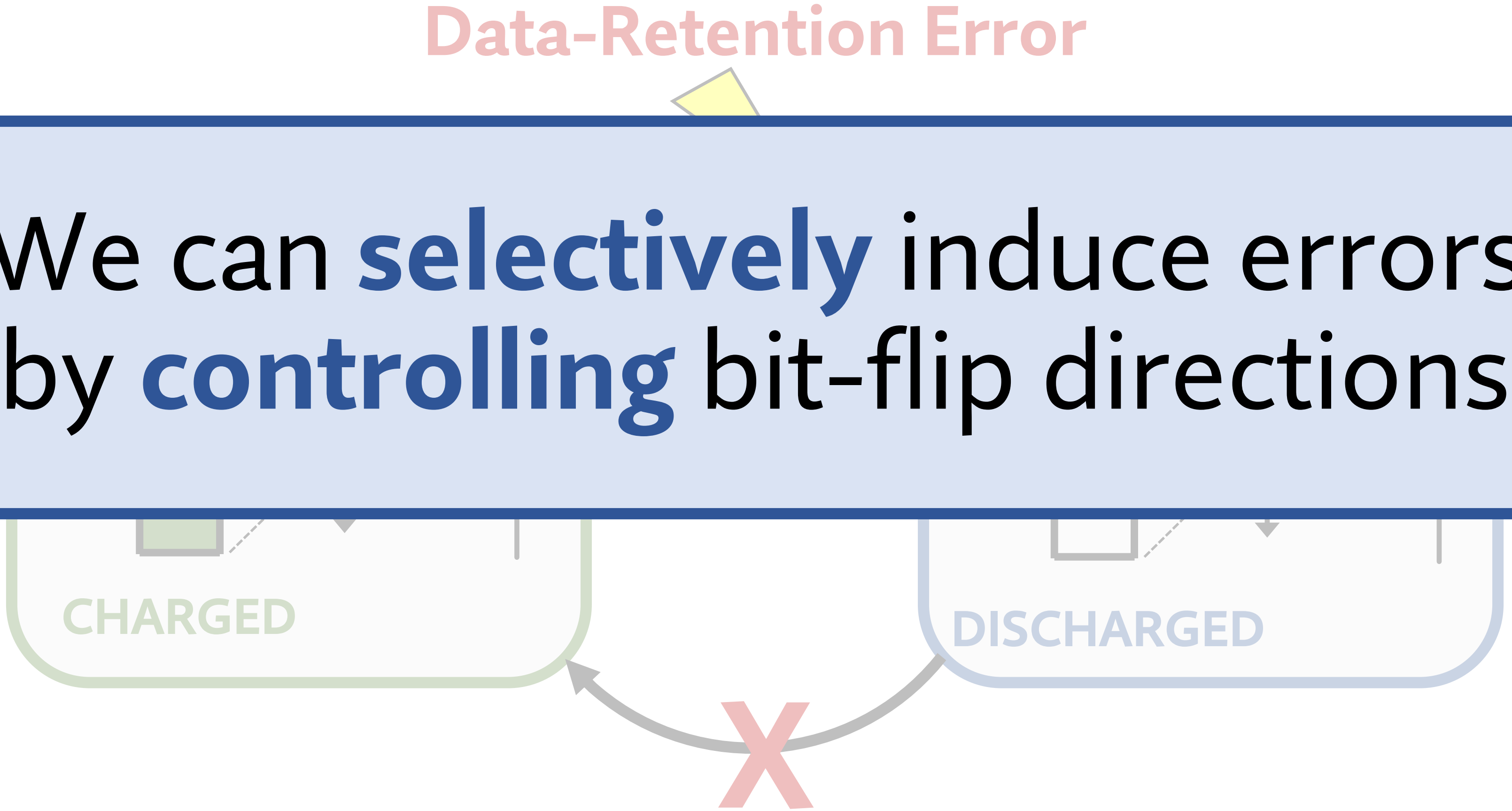


- **BEER:** Reveals **how** on-die ECC scrambles errors
- **BEEP:** Enables **inferring** raw bit error locations

Key idea: disabling DRAM refresh induces data-retention errors **only** in CHARGED cells

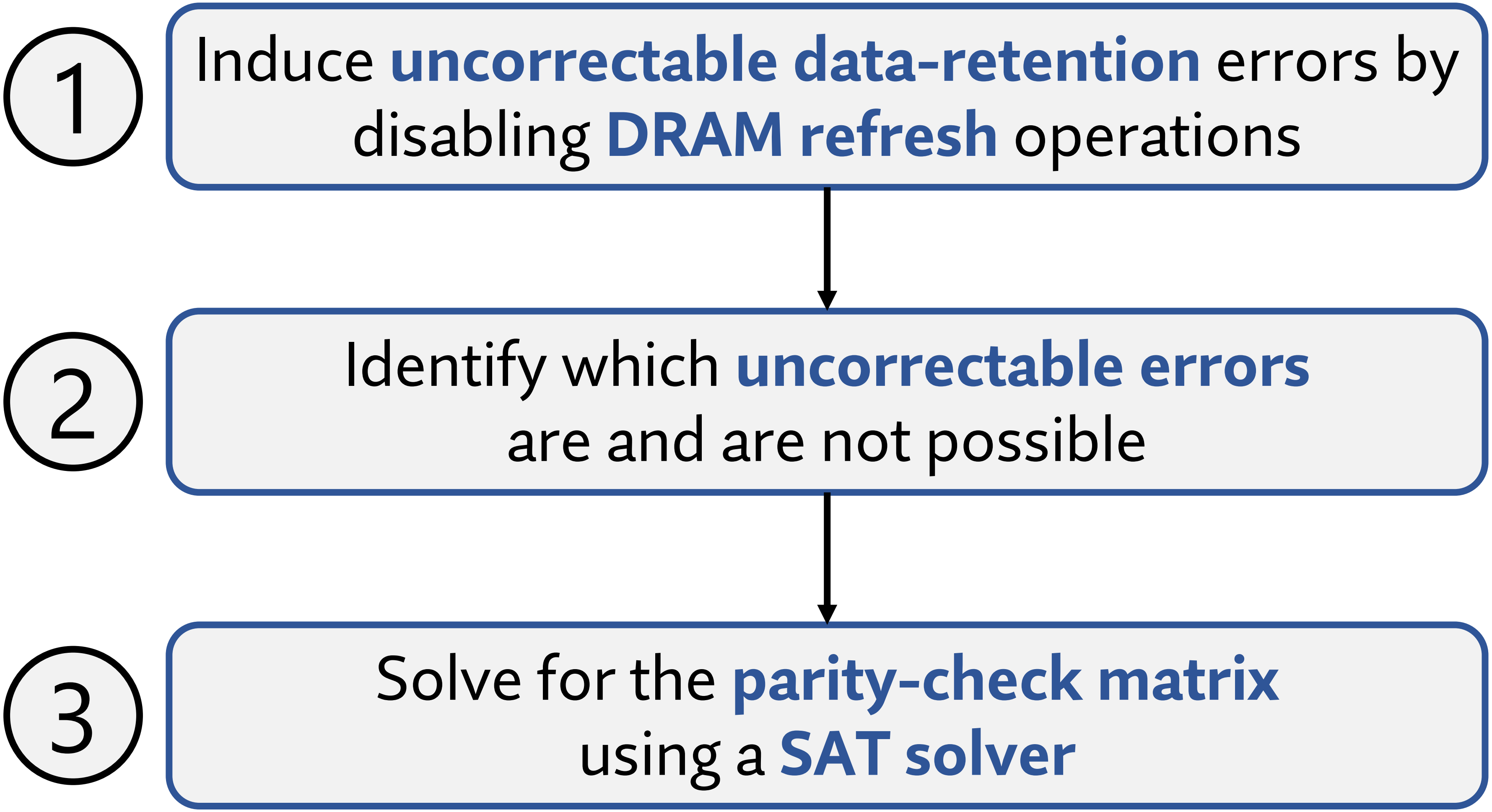


Key idea: disabling DRAM refresh induces data-retention errors **only** in CHARGED cells



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

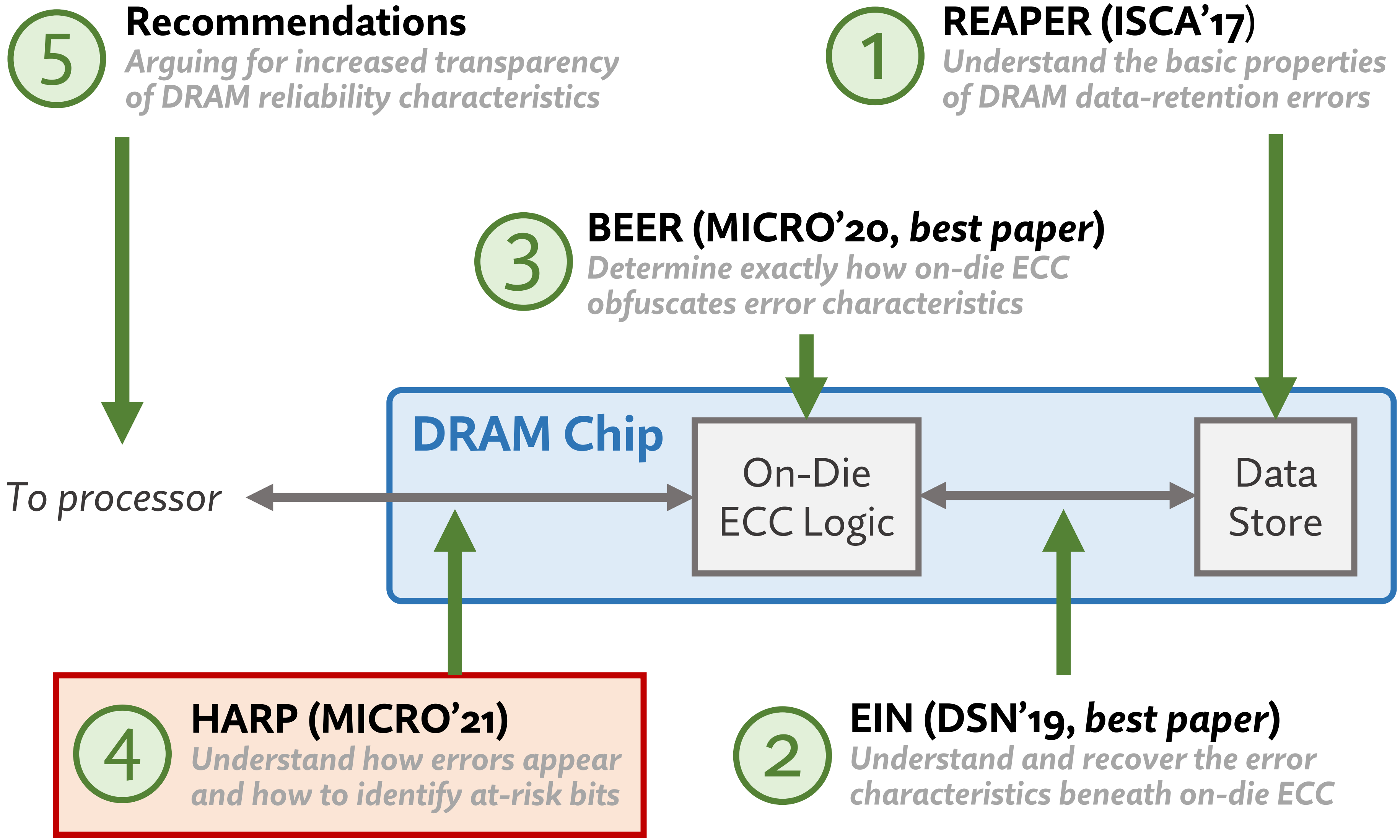
Over 100,000 representative ECC codes
of varying word lengths (4 – 247 bits)

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

Core Contributions



5 **Recommendations**
Arguing for increased transparency of DRAM reliability characteristics

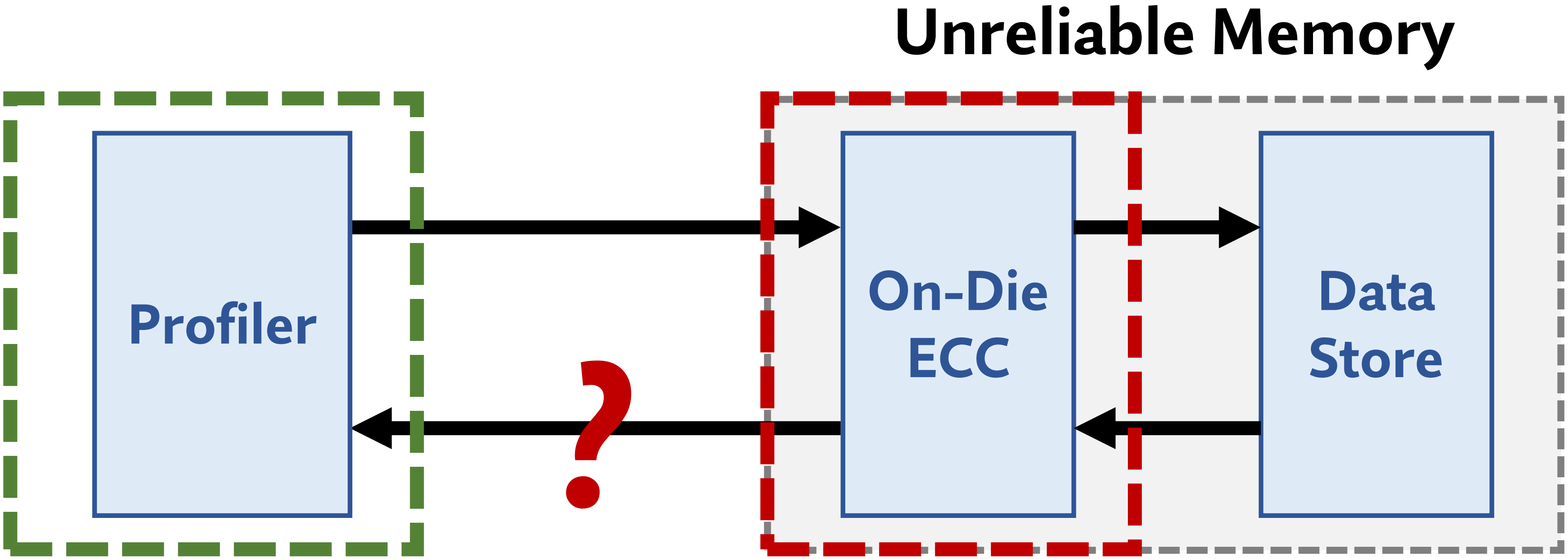
1 **REAPER (ISCA’17)**
Understand the basic properties of DRAM data-retention errors

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

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

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

Profiling a Memory Chip with On-Die ECC



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

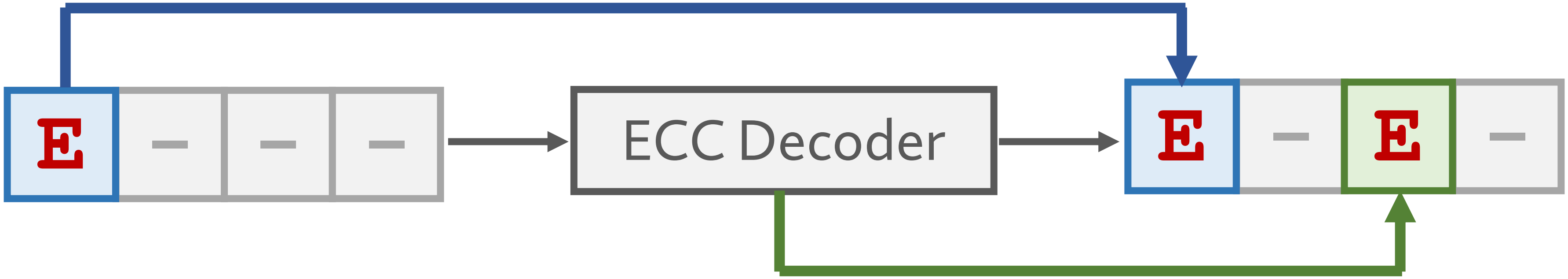
2 Makes it **harder to identify** individual at-risk bits

3 **Interferes** with commonly-used data patterns for memory testing

Key Observation: Two Sources of Errors

1 **Direct error**

Due to errors
in the **memory chip**



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 **memory chip**

Key Idea: **decouple** profiling for **direct** and **indirect** errors

2 **Indirect error**

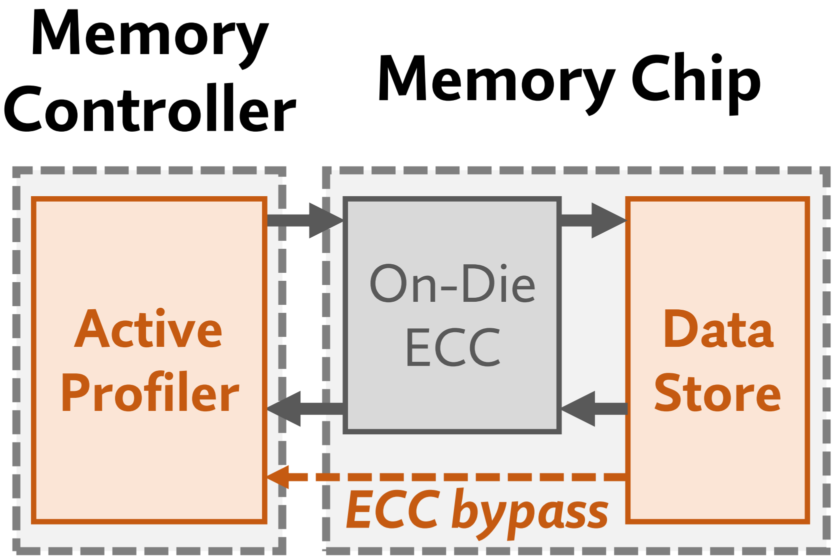
Artifact of the on-die ECC algorithm

Upper-bounded by the ECC algorithm

Hybrid Active-Reactive Profiling (HARP)

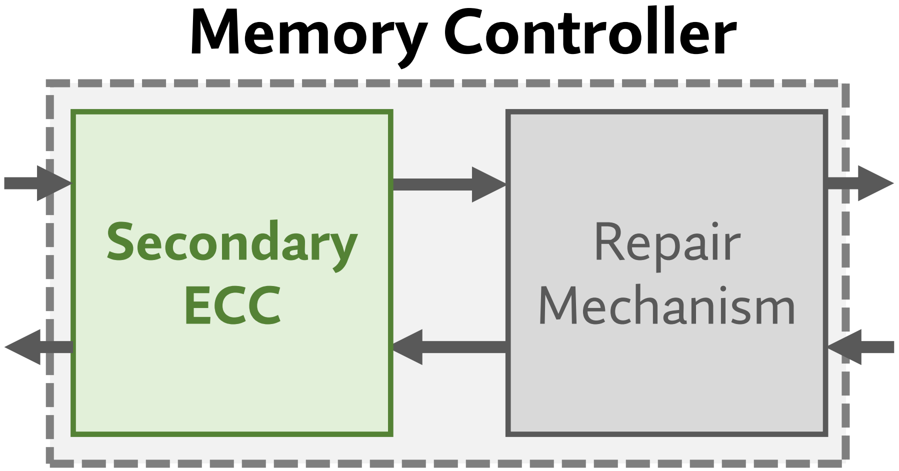
1 Active Profiling

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



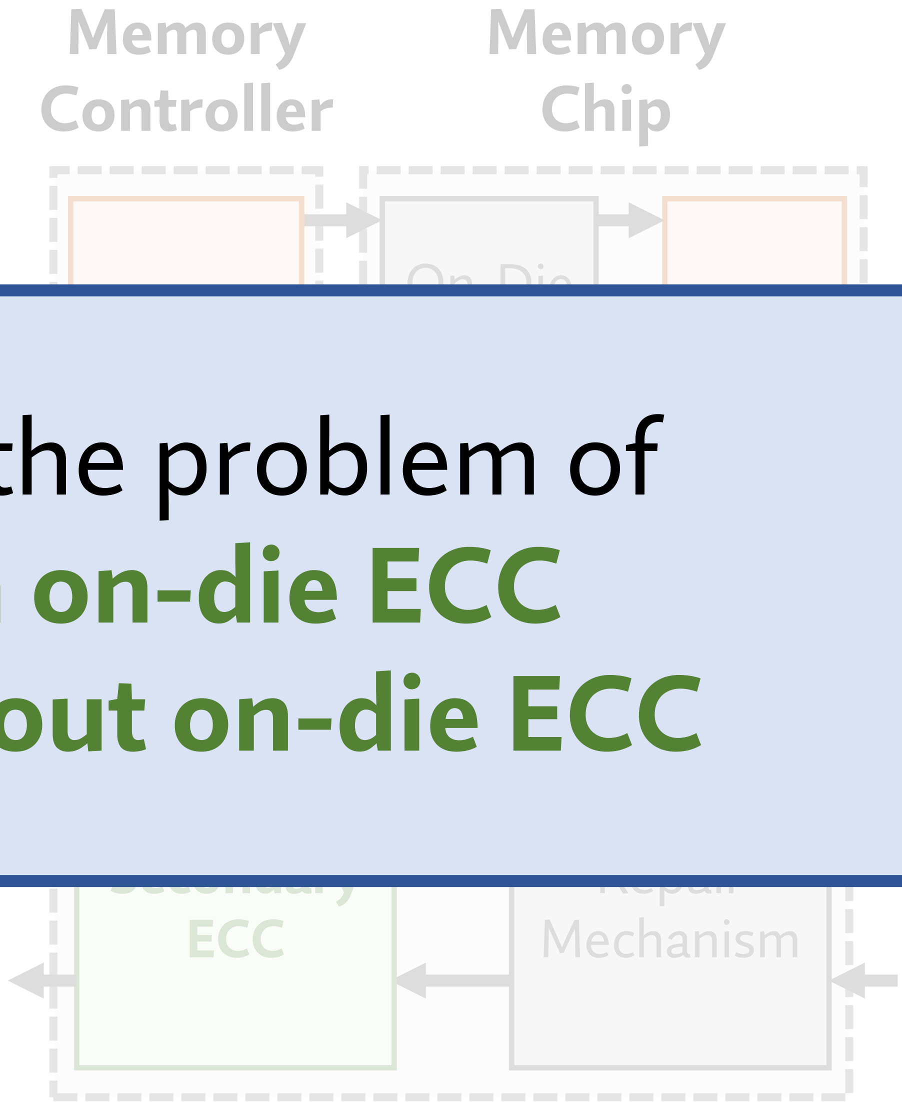
2 Reactive Profiling

Safely identifies **indirect errors** using **secondary ECC** at least as strong as on-die ECC



Hybrid Active-Reactive Profiling (HARP)

1 Active Profiling



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

Safely identifies **indirect errors** using **secondary ECC** at least as strong as on-die ECC

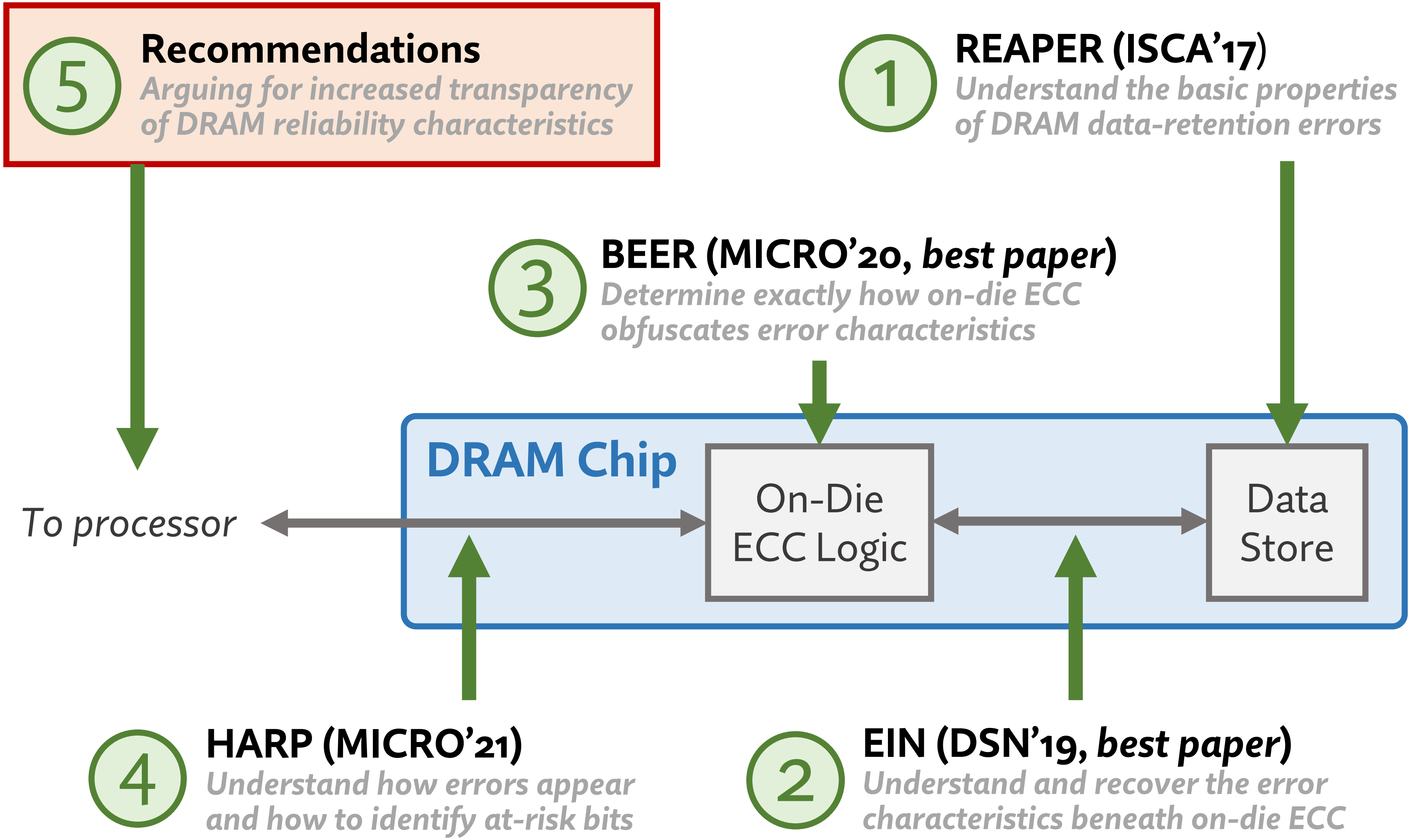
Evaluations

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 99th-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

Core Contributions



5 **Recommendations**
Arguing for increased transparency of DRAM reliability characteristics

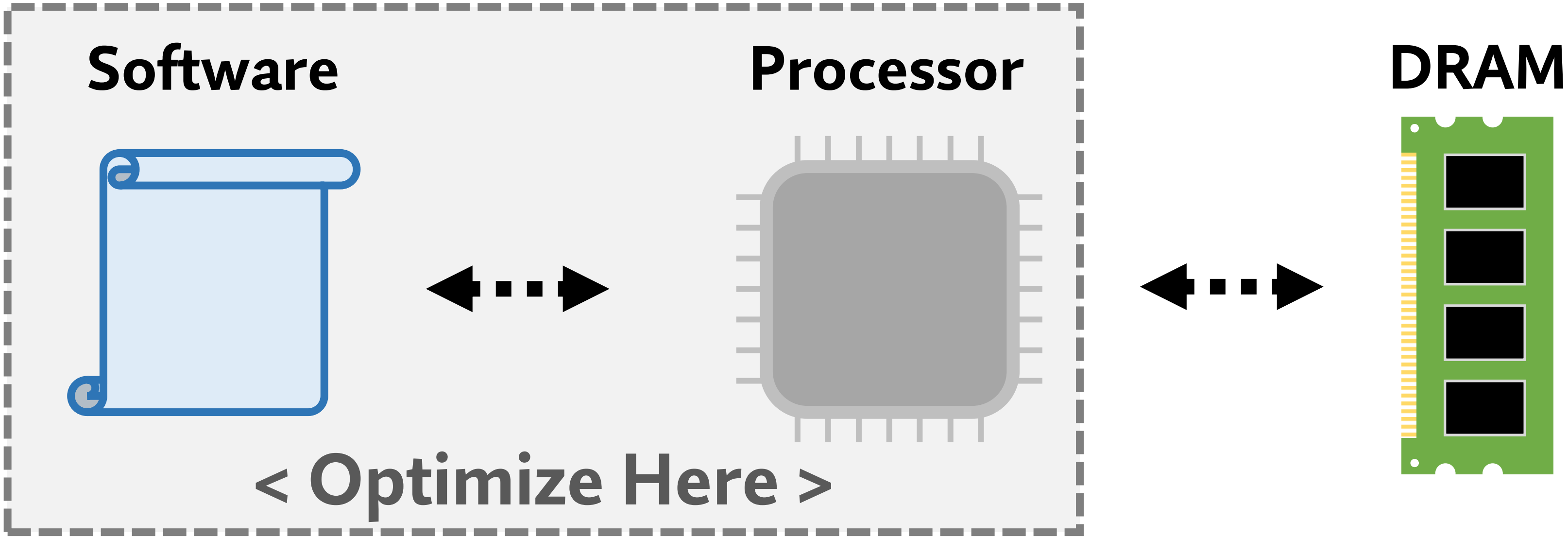
1 **REAPER (ISCA'17)**
Understand the basic properties of DRAM data-retention errors

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

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

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

Many Ways to Exploit Commodity DRAM



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

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)

e.g., ECC, redundancy, replication

Use security enhancements

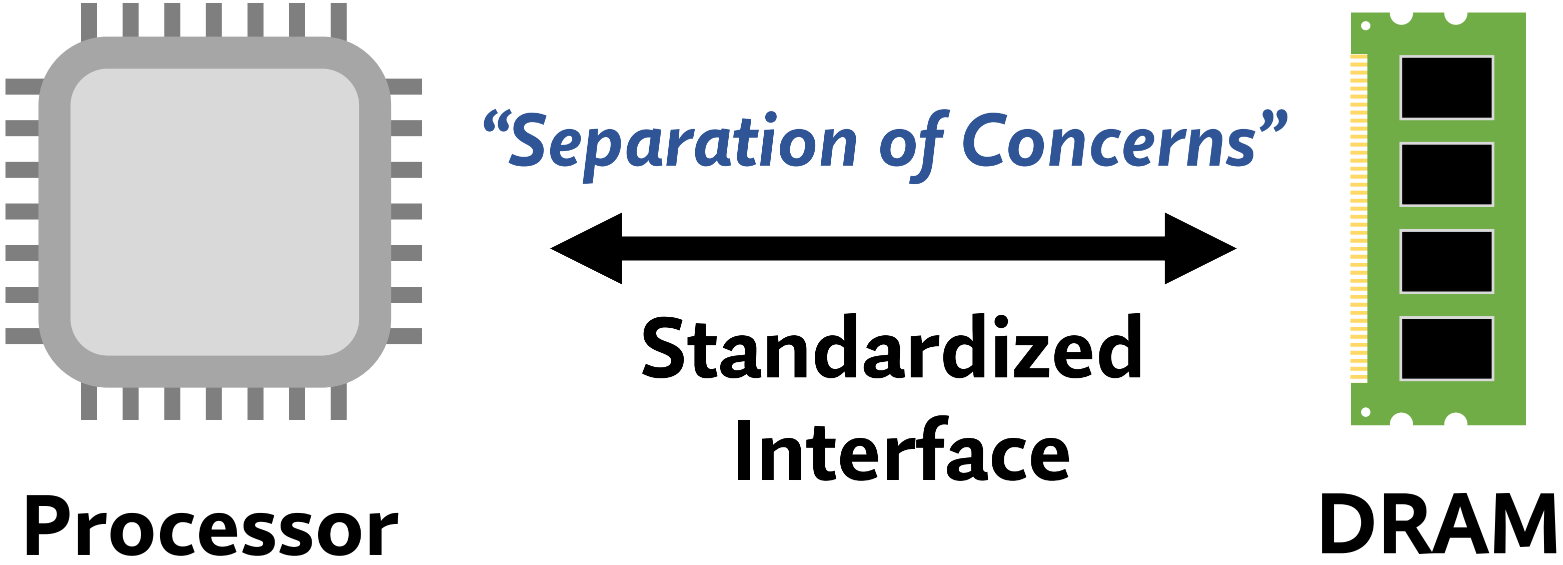
e.g., RowHammer and Cold-Boot defenses

Performance

Energy/Power

Source of the Problem

- Commodity DRAM specifications do not provide this information **by design**



"Separation of Concerns"

Standardized Interface

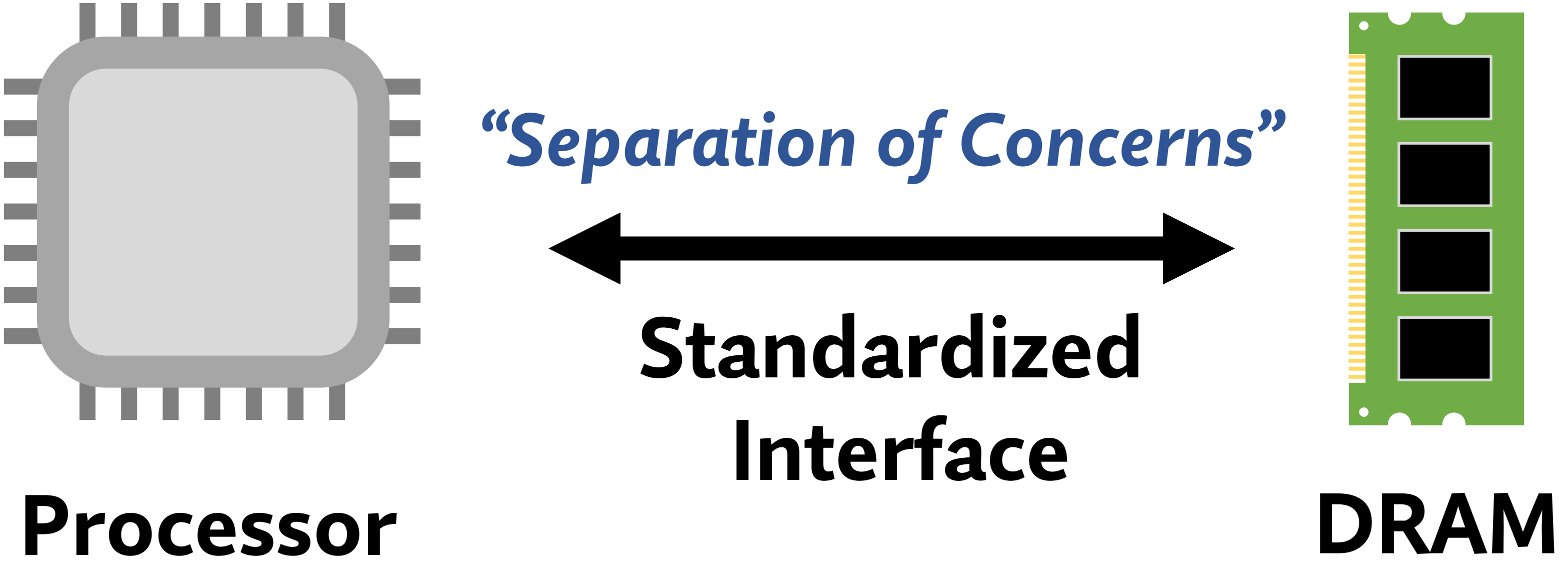
Processor

DRAM

- Unfortunately, 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**



“Separation of Concerns”

Standardized Interface

Processor

DRAM

Proposal: revisit **DRAM specifications** to improve **information transparency**

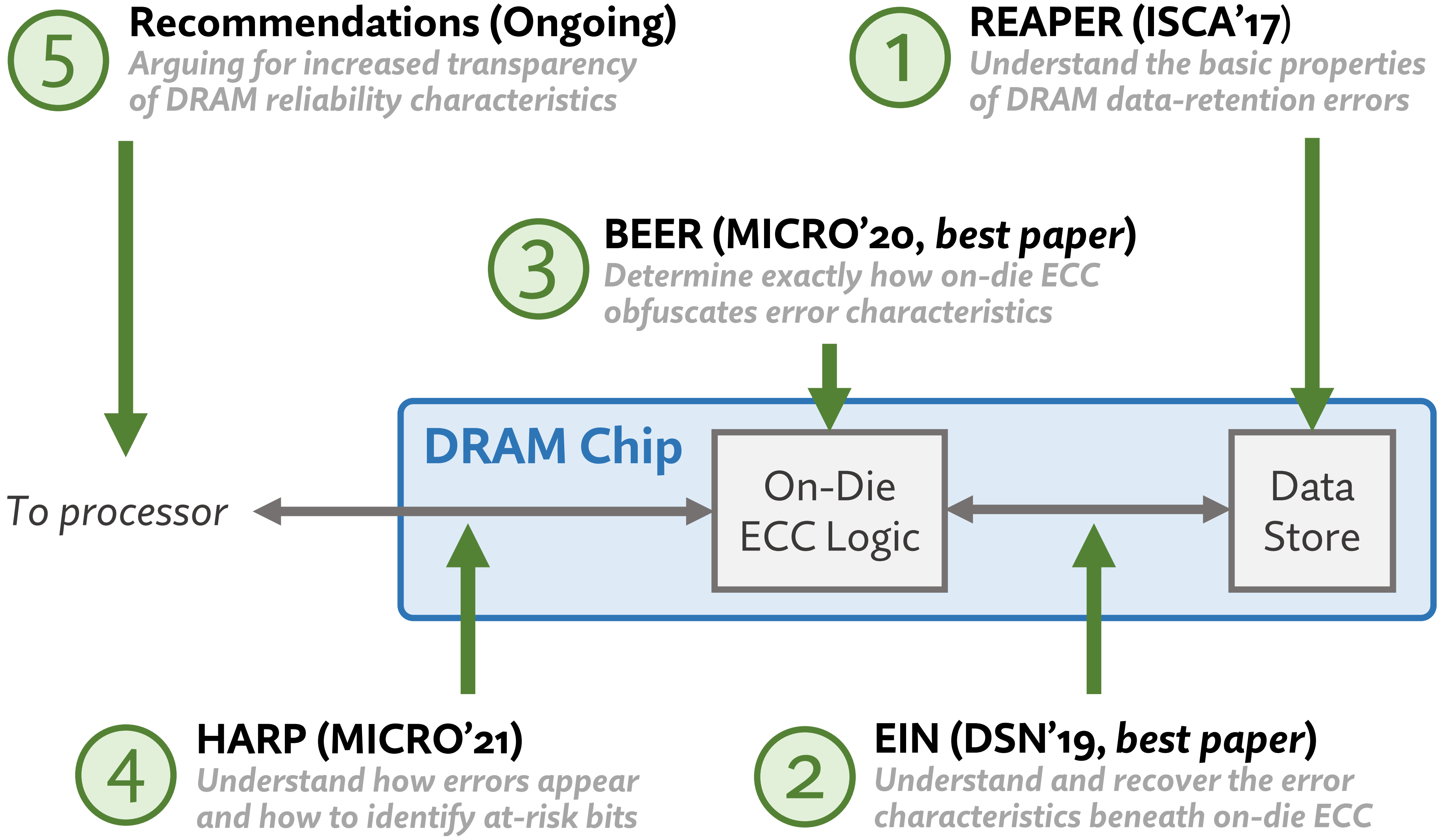
Two-Step Plan for Transparency

- **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
   - 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

Core Contributions



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

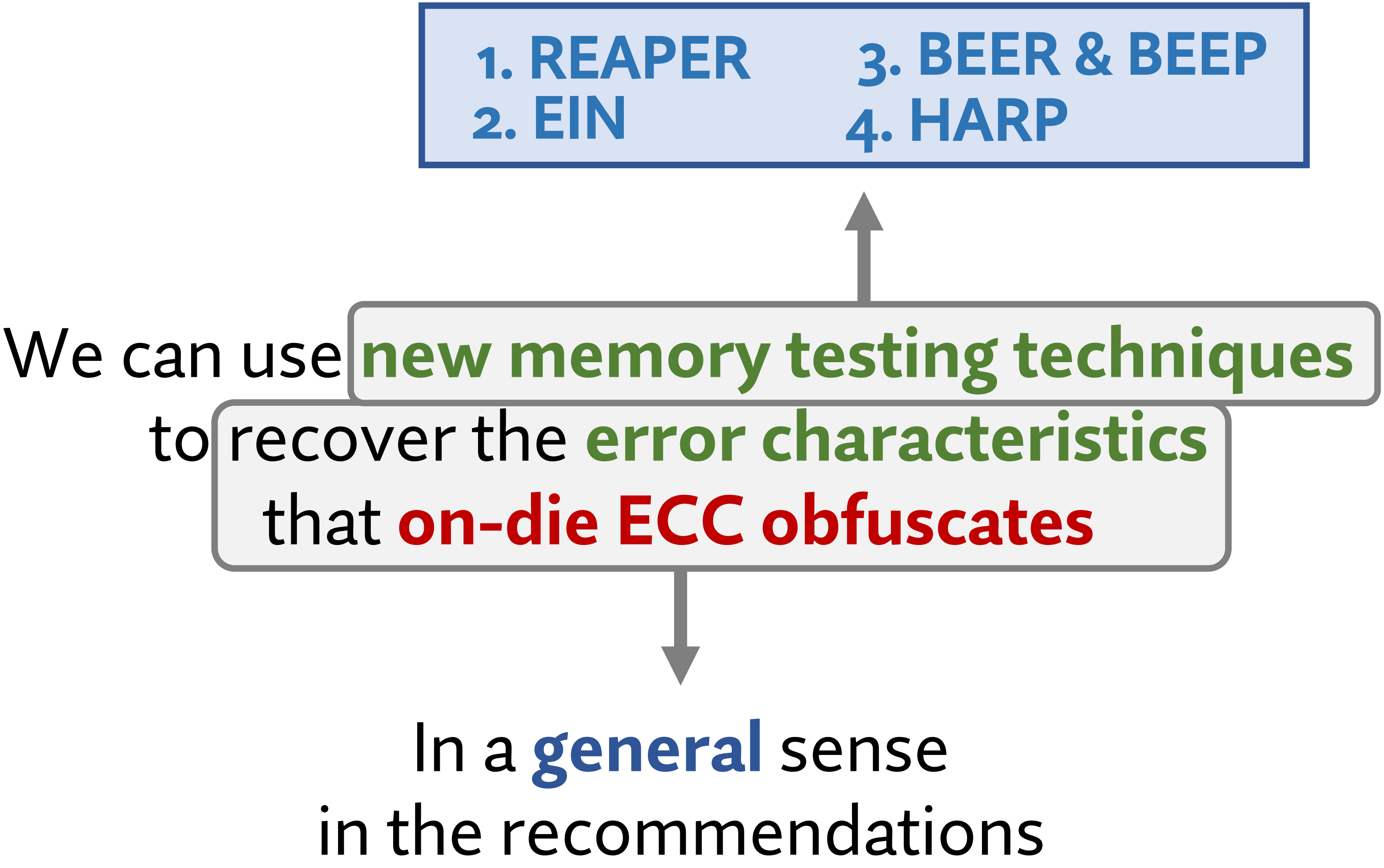
1 **REAPER (ISCA'17)**
Understand the basic properties of DRAM data-retention errors

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

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

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

Thesis Statement



1. REAPER
2. EIN
3. BEER & BEEP
4. HARP

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

In a **general** sense in the recommendations

Future Research Directions

- **Extending the techniques** that we propose to other:
  - ECC types and error mechanisms
  - ECC architectures and memory technologies

- **Using the information** that our techniques reveal
  - Improved system-level error mitigations
  - Error-tolerant computing

- **Devising alternatives** to on-die ECC
  - Different on-die ECC architectures
  - System-level error mitigation mechanisms

- **Improving transparency** of DRAM reliability

My Involvements During the Ph.D.

Thesis Publications	Patel+ ISCA’17 Patel+ DSN’19 Patel+ MICRO’20 Patel+ MICRO’21
Near-Data Processing	Boroumand+ CAL’16 Boroumand+ ISCA’19 Hajinazar+ ASPLOS’21
Reducing Memory Access Latency	Kim+ ICCD’18 Wang+ MICRO’18 Hassan+ ISCA’19 Wang+ MICRO’20 Luo+ ISCA’20
Security Topics	Kim+ HPCA’18 Kim+ HPCA’19 Orosa+ ISCA’21 Olgun+ ISCA’21
Virtual Memory	Hajinazar+ ISCA’20
Understanding RowHammer	Cojocar+ S&P'20 Kim+ ISCA'20 Orosa+ MICRO’21
Mitigating RowHammer	Yağlıkçı+ HPCA'21

2016 *2017* *2018* *2019* *2020* *2021*

Acknowledgments

- Onur Mutlu
- Mattan Erez, Moinuddin Qureshi, Vilas Sridharan, and Christian Weis
- SAFARI group members (from both CMU and ETH)
- Internship mentors and industry sponsors
- Friends and colleagues all over the world
- Family – mum, dad, and sister

Enabling Effective Error Mitigation in Modern Memory Chips that Use On-Die ECC

Minesh Patel

Doctoral Examination

1 October 2021

Advisor:

Onur Mutlu (ETH Zürich)

Co-Examiners:

Mattan Erez (UT Austin)
Moinuddin Qureshi (Georgia Tech)
Vilas Sridharan (AMD)
Christian Weis (TU Kaiserslautern)

ETH zürich

SAFARI