ETH zürich

SAFARI

Enabling Efficient and Scalable DRAM Read Disturbance Mitigation via New Experimental Insights

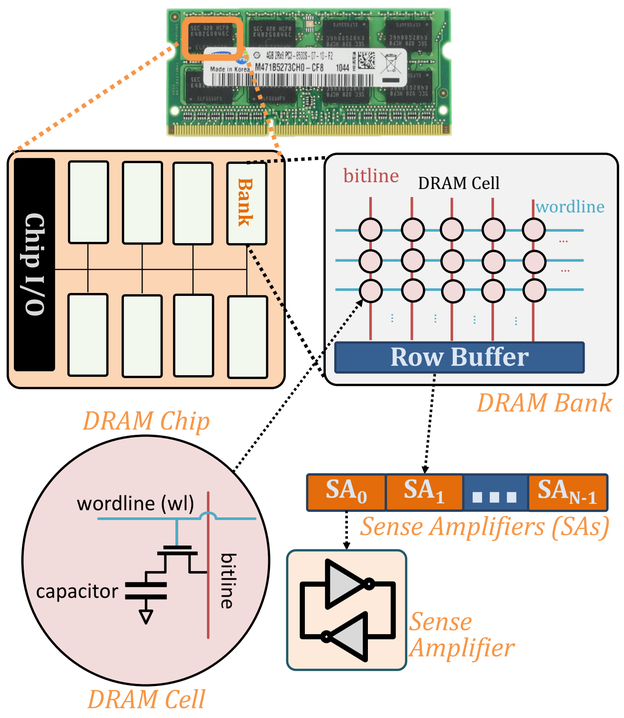
Abdullah Giray Yaglikci Onur Mutlu

Contact Info

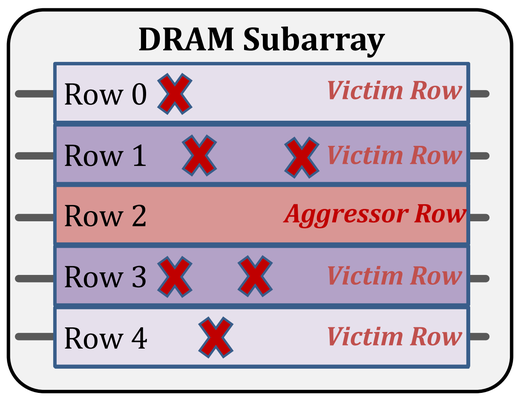
Lightning Talk

1. DRAM Background

DRAM Organization



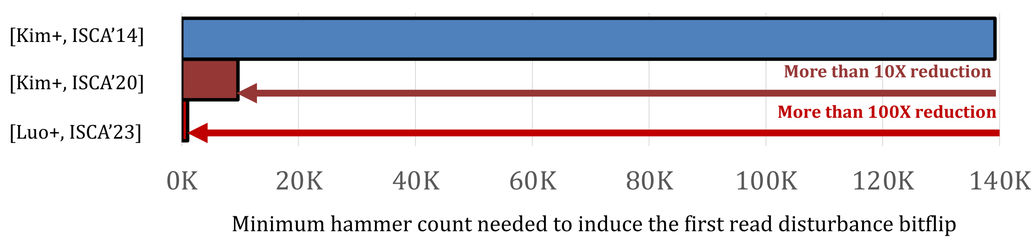
DRAM Read Disturbance



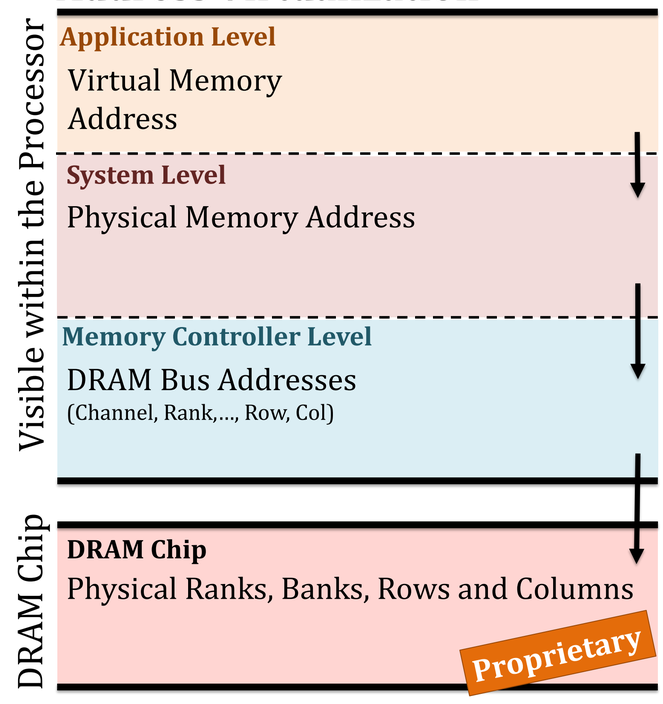
RowHammer [Kim+ ISCA'14]

RowPress [Luo+ ISCA'23]

Repeatedly **opening** and **closing** **OR** **keeping open** a DRAM row causes **bitflips** in nearby cells



Address Virtualization



2. Problem Definition

DRAM read disturbance mitigation mechanisms

- **poorly scale** with shrinking technology node
- rely on **proprietary information**

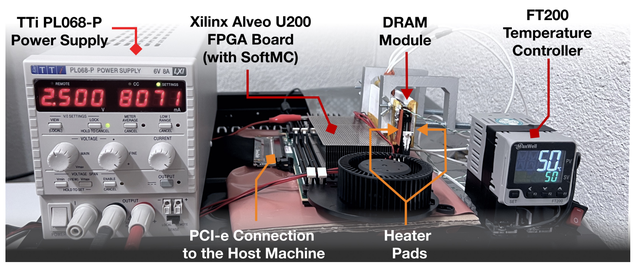
3. Thesis Statement

We can **mitigate DRAM read disturbance efficiently** and **scalably** by

1. building a **detailed understanding** of DRAM read disturbance
2. leveraging **insights into modern DRAM chips** and **memory controllers**
3. devising **novel solutions** that do not require **proprietary knowledge** of DRAM chip internals

3: Building a Detailed Understanding of DRAM Read Disturbance

Methodology



Full Paper
DRAM Bender
[Olgun+ TCAD'23]

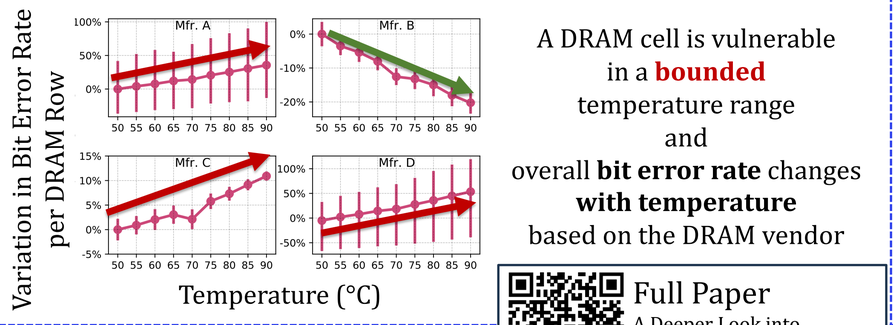
GitHub Repository

Fine-grained control over

- **DRAM commands**
  e.g., ACT, PRE, RD, WR, and REF
- **Timing parameters (±1.5ns)**,
- **Temperature (±0.5°C),**
- **Wordline Voltage (±1mV)**

We test 250+ DRAM Chips

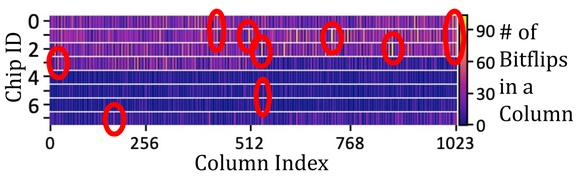
DRAM Chip Temperature

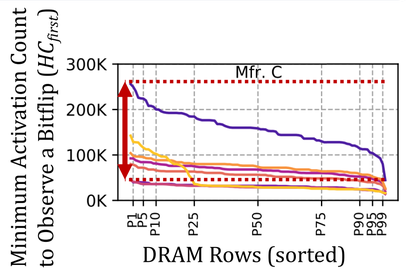


A DRAM cell is vulnerable in a **bounded** temperature range and overall **bit error rate** changes **with temperature** based on the DRAM vendor

Full Paper
A Deeper Look into RowHammer's Sensitivities
[Orosa, Yaglikci+ MICRO'21]

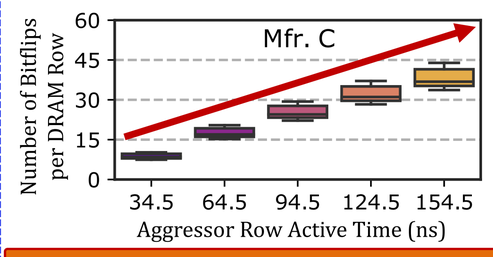
Physical Location





Both **manufacturing process** and **design decisions** affect **vulnerability to read disturbance**

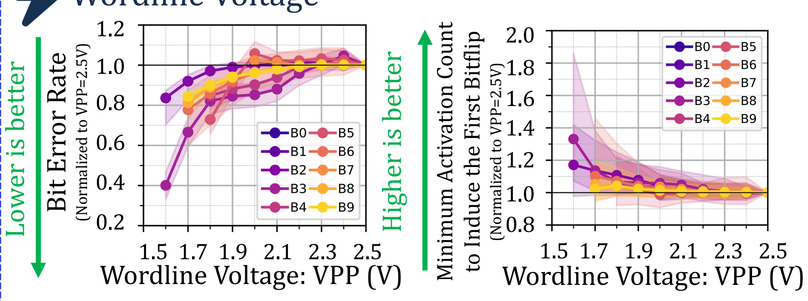
Memory Access Patterns



As the **aggressor row stays active longer**, **more DRAM cells** experience bitflips and bitflips occur at **lower activation counts**

Paved the way to the discovery of RowPress [Luo+ ISCA'23]

Wordline Voltage



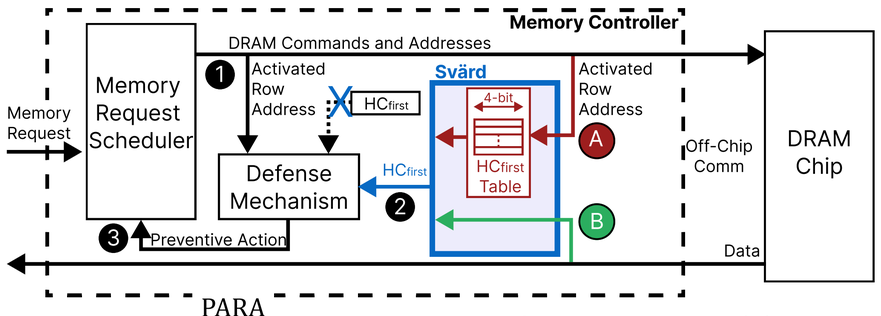
Full Paper
RowHammer under Reduced Wordline Voltage
[Yaglikci+ DSN'22]

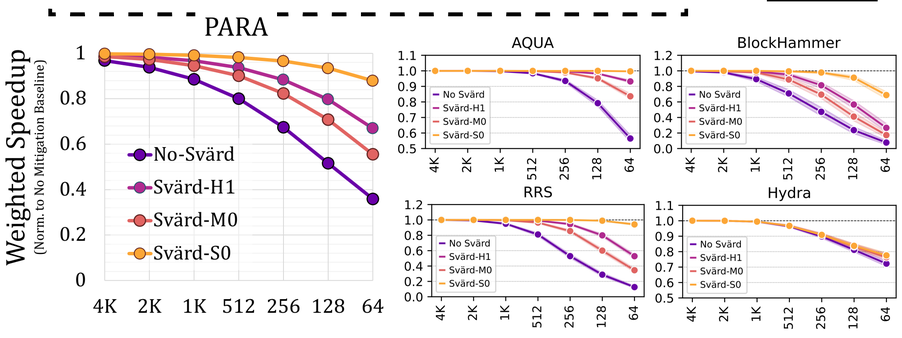
Fewer DRAM cells experience bitflips and bitflips occur at **higher activation counts** under **reduced wordline voltage**

4: Leveraging Insights into Modern DRAM Chips and Memory Controllers

Svärd: Spatial Variation-Aware Read Disturbance Defenses

Dynamically adapts the aggressiveness of a defense **to the victim row's** vulnerability level

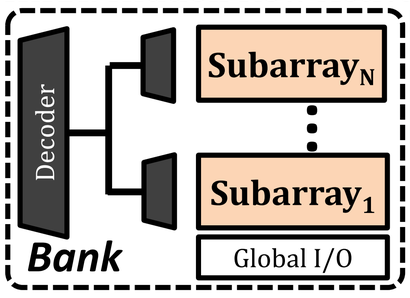




Svärd **significantly increases** system performance by 1.63x, 4.88x, 1.07x, 1.95x, and 4.80x, over **AQUA, BlockHammer**, **Hydra**, **PARA**, and **RRS,** respectively, for HC_{first} of **64**

Full Paper
Spatial Variation-Aware RowHammer Defenses
[Yaglikci+ HPCA'24]

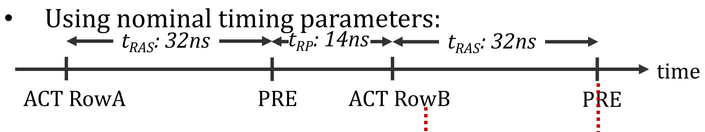
HiRA: Hidden Row Activation



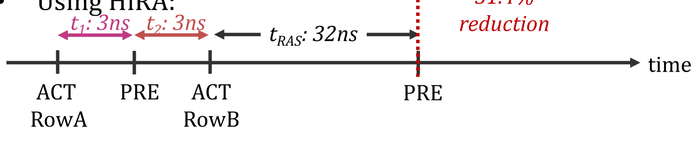
Full Paper
HiRA: Hidden Row Activation
[Yaglikci+ MICRO'22]

Refreshing two rows in two different subarrays:

- Using nominal timing parameters:



- Using HiRA:

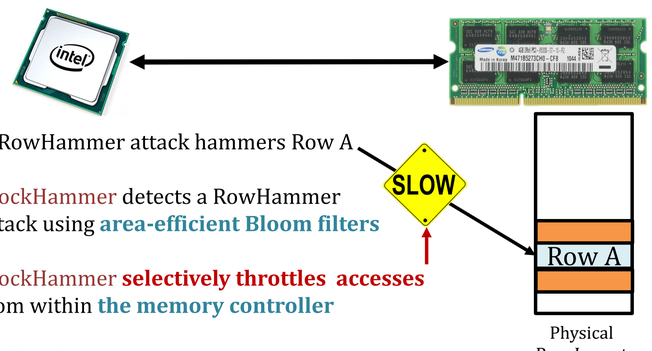


- HiRA works in 56 off-the-shelf DRAM chips from SK Hynix

12.6% speedup by hiding *periodic refresh* latency

3.7x speedup by hiding *preventive refresh* latency

BlockHammer: Selectively Throttling Accesses



- A RowHammer attack hammers Row A
- BlockHammer detects a RowHammer attack using **area-efficient Bloom filters**
- BlockHammer **selectively throttles accesses** from within **the memory controller**
- Bitflips **do not** occur
- BlockHammer can **inform the system software** about the attack

Full Paper
BlockHammer
[Yaglikci+ HPCA'21]

GitHub
Simulator
Source Code

FINALIST 2022

Scalability with Worsening DRAM Read Disturbance:

Competitive with state-of-the-art mechanisms (<0.6% overhead) **when there is no attack**

Superior performance (71% speedup) and DRAM energy (32% reduction) **when a RowHammer attack is present**

Compatibility with Commodity DRAM Chips:

BlockHammer requires

- **no proprietary information** of in-DRAM row mapping
- **no changes** to DRAM chips

More info: agyaglikci@gmail.com & https://agyaglikci.github.io

5: No Proprietary Knowledge