CAn’t Touch This

Software-only Mitigation against Rowhammer Attacks targeting Kernel Memory

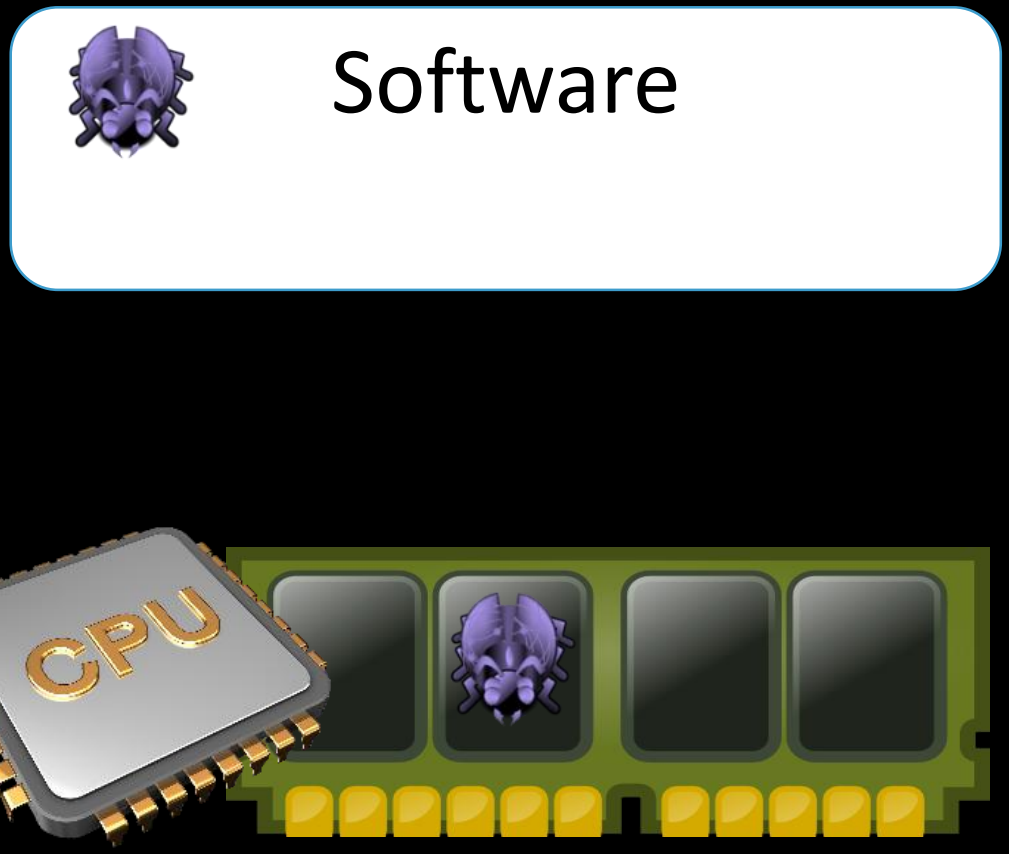
Ferdinand Brasser
David Gens
Christopher Liebchen
Ahmad-Reza Sadeghi

Intel Collaborative Research Institute for Secure Computing
Technische Universität Darmstadt

Lucas Davi

University of Duisburg-Essen

Big Picture: Rowhammer Attacks



Big Picture: Rowhammer Attacks

Software

Big Picture: Rowhammer Attacks

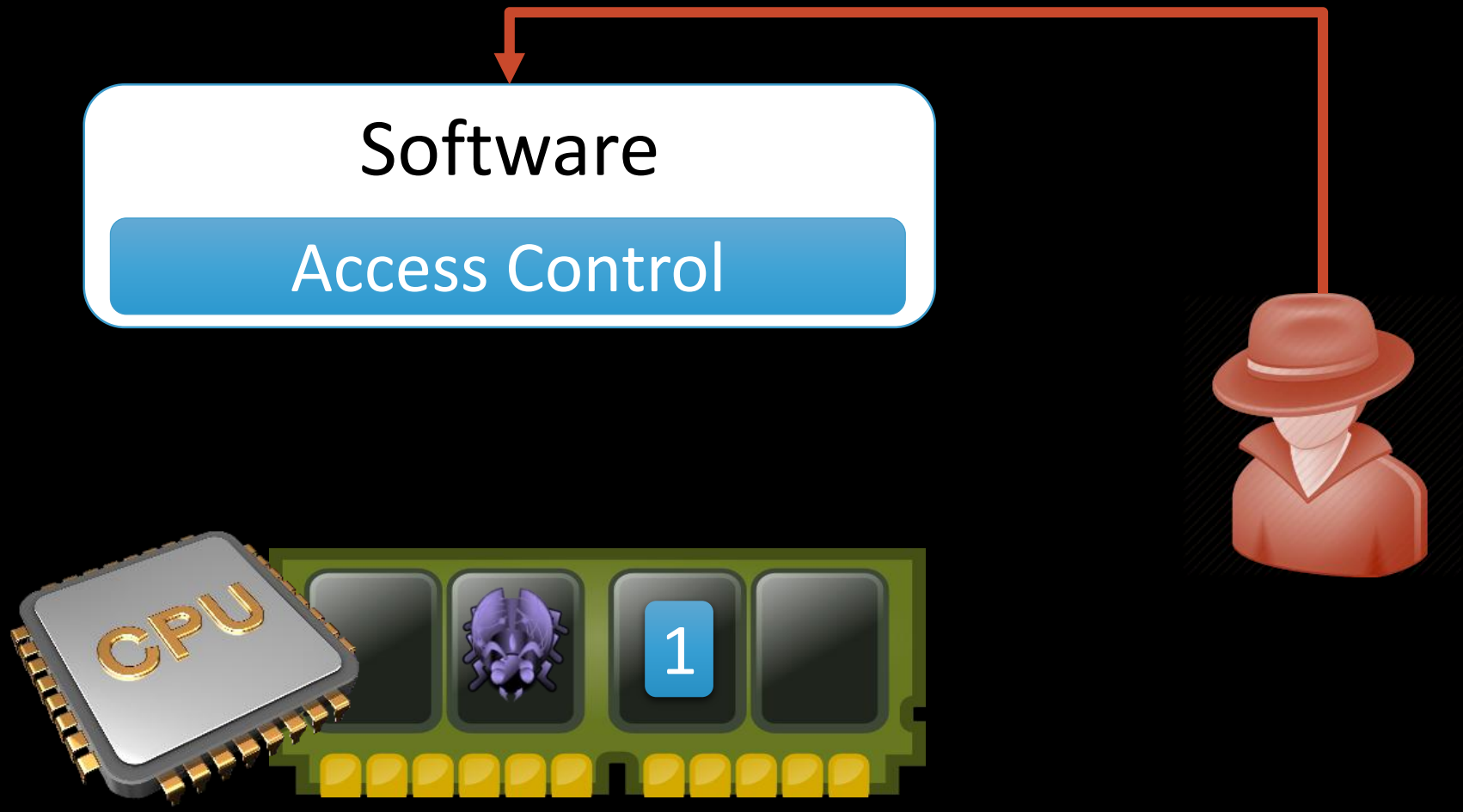
Software

Big Picture: Rowhammer Attacks

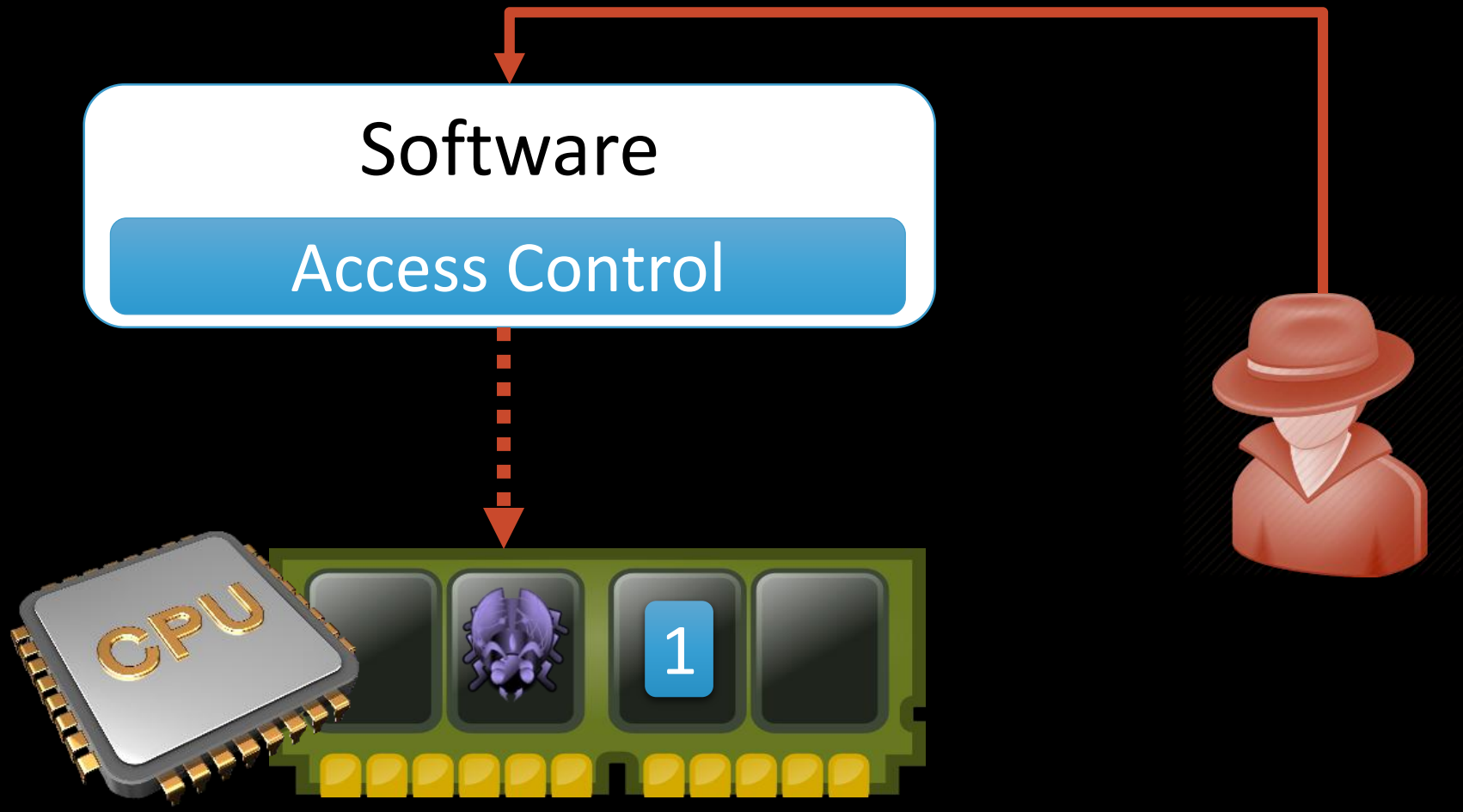
Software

Access Control

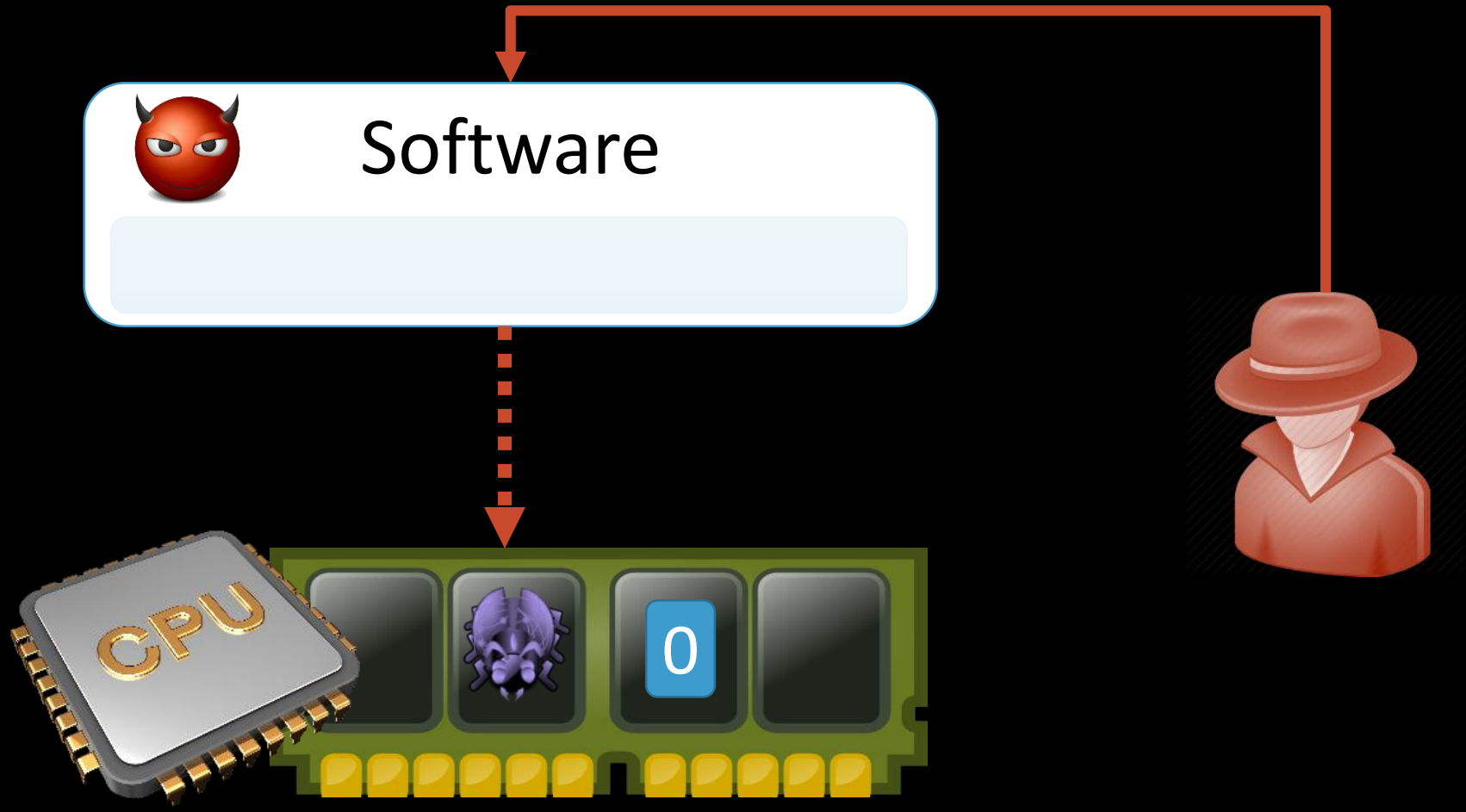
Big Picture: Rowhammer Attacks



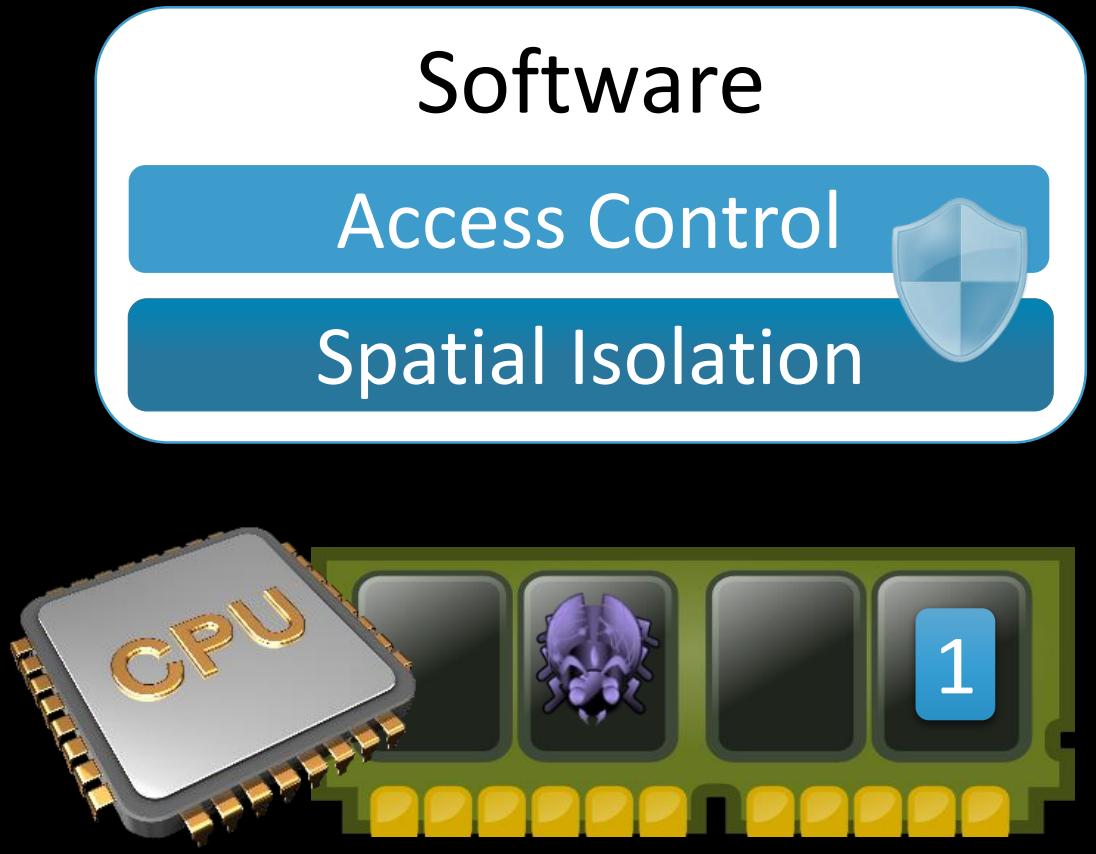
Big Picture: Rowhammer Attacks



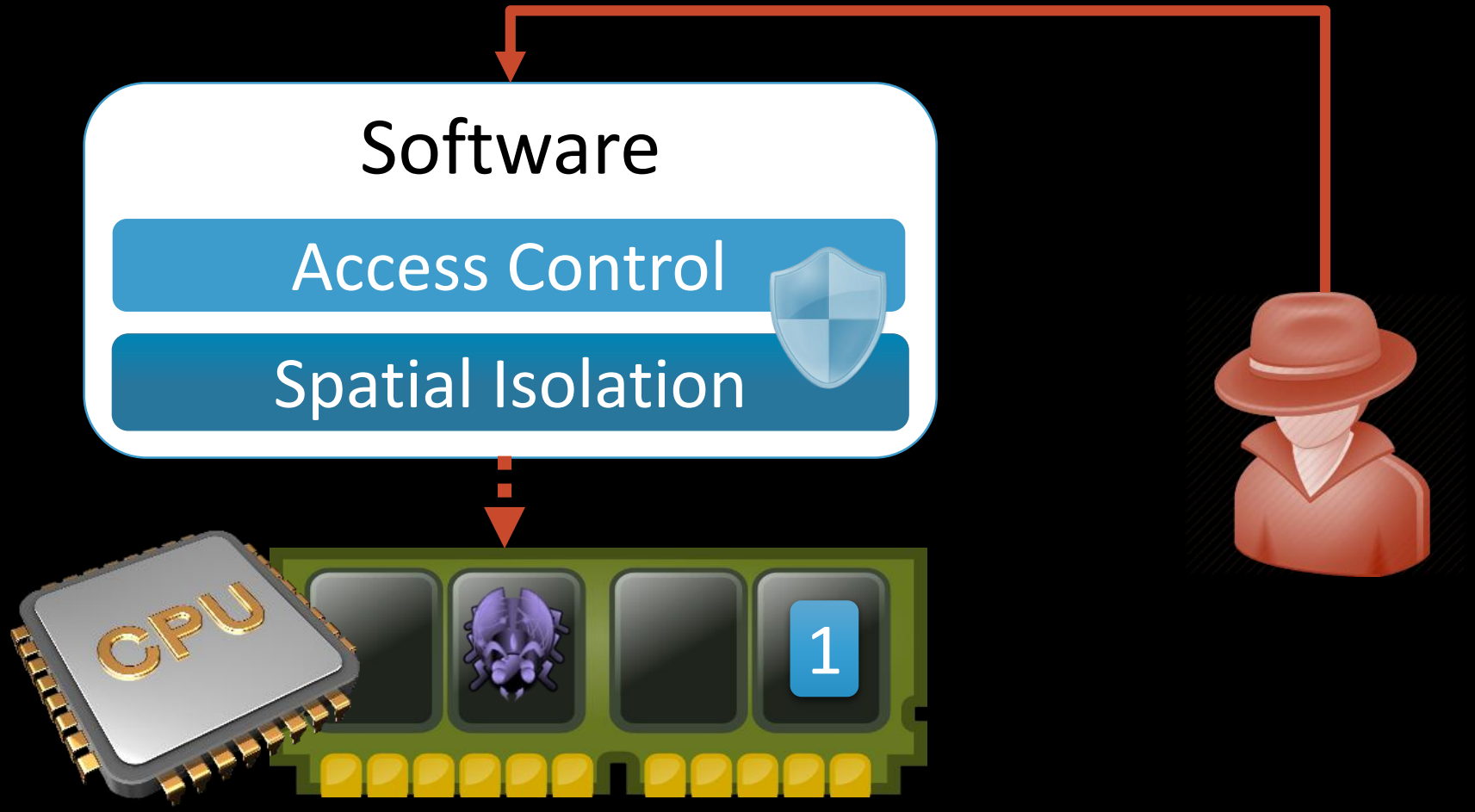
Big Picture: Rowhammer Attacks



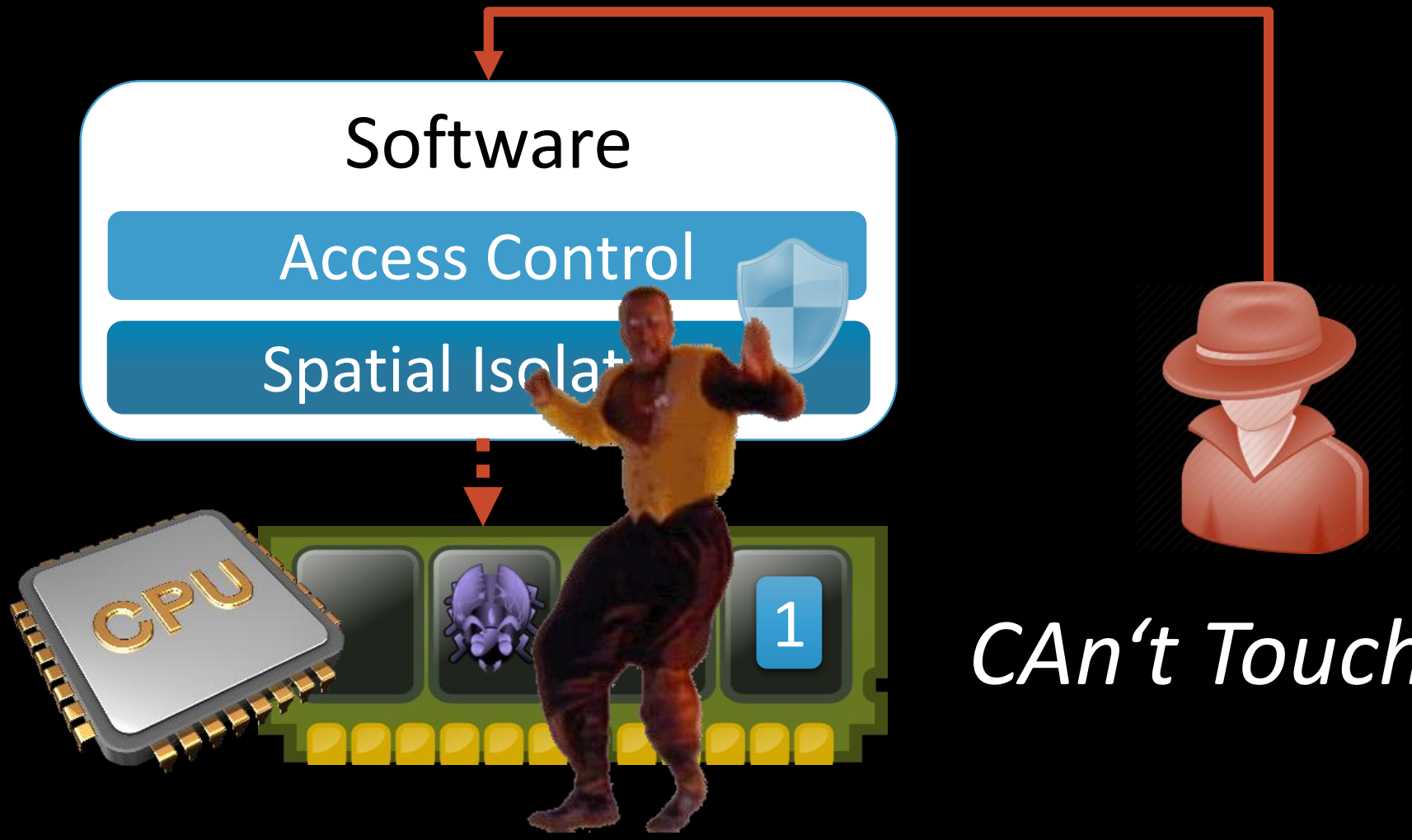
Big Picture: Our Approach



Big Picture: Our Approach

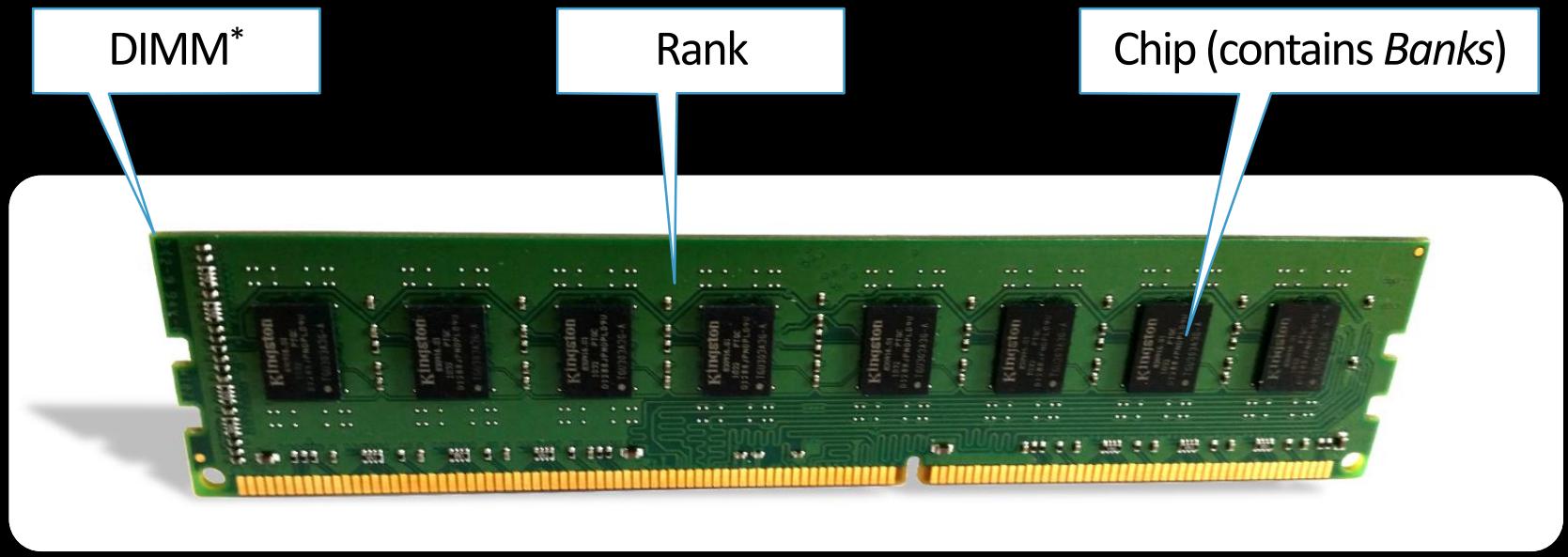


Big Picture: Our Approach



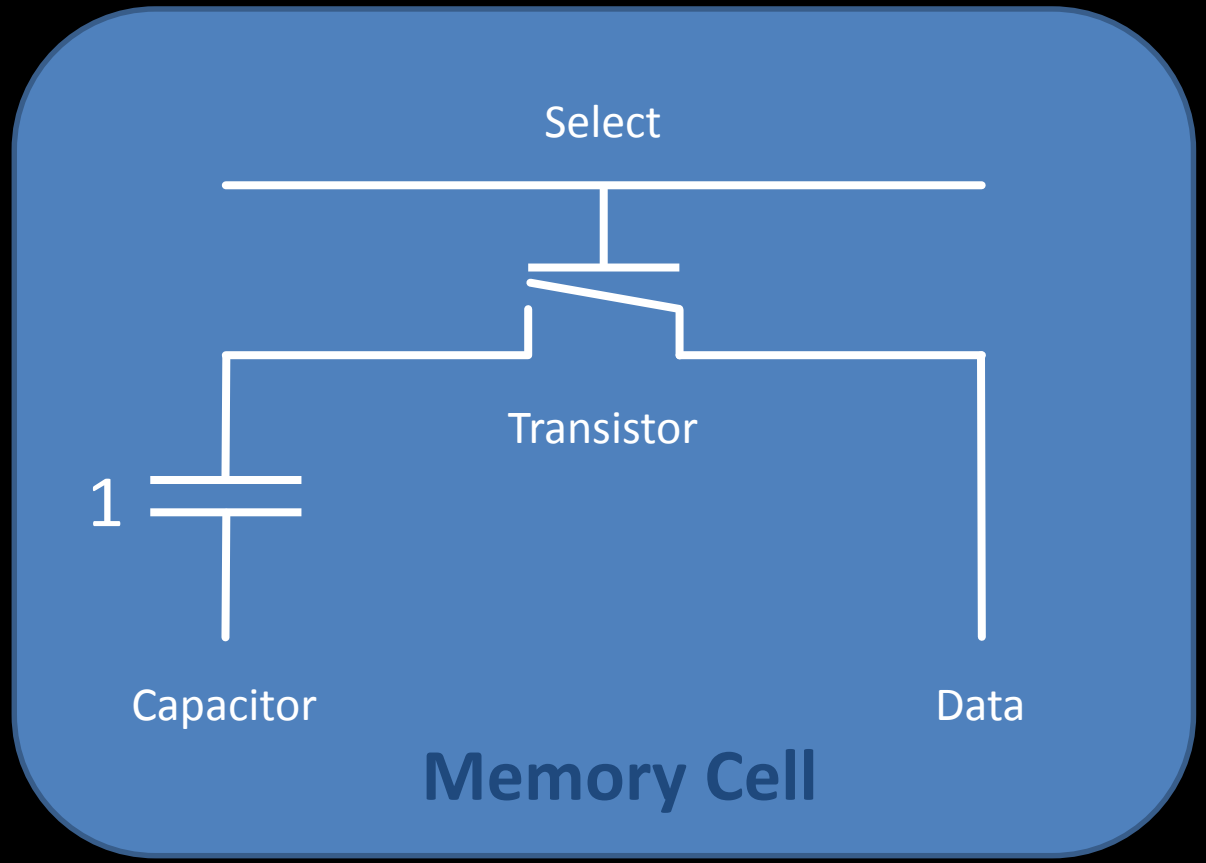
CAn‘t Touch This

Dynamic Random Access Memory (DRAM)



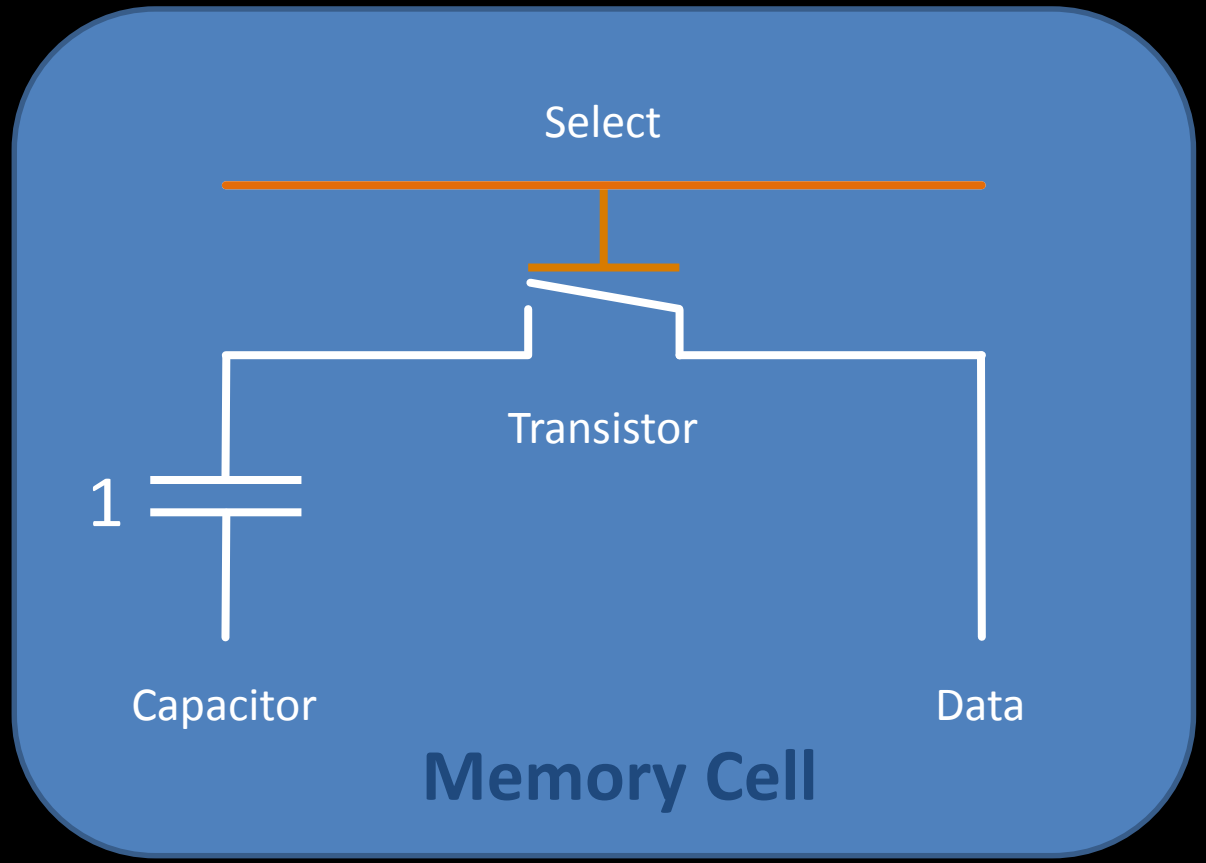
*) Dual Inline Memory Module

DRAM: Storing a Single Bit



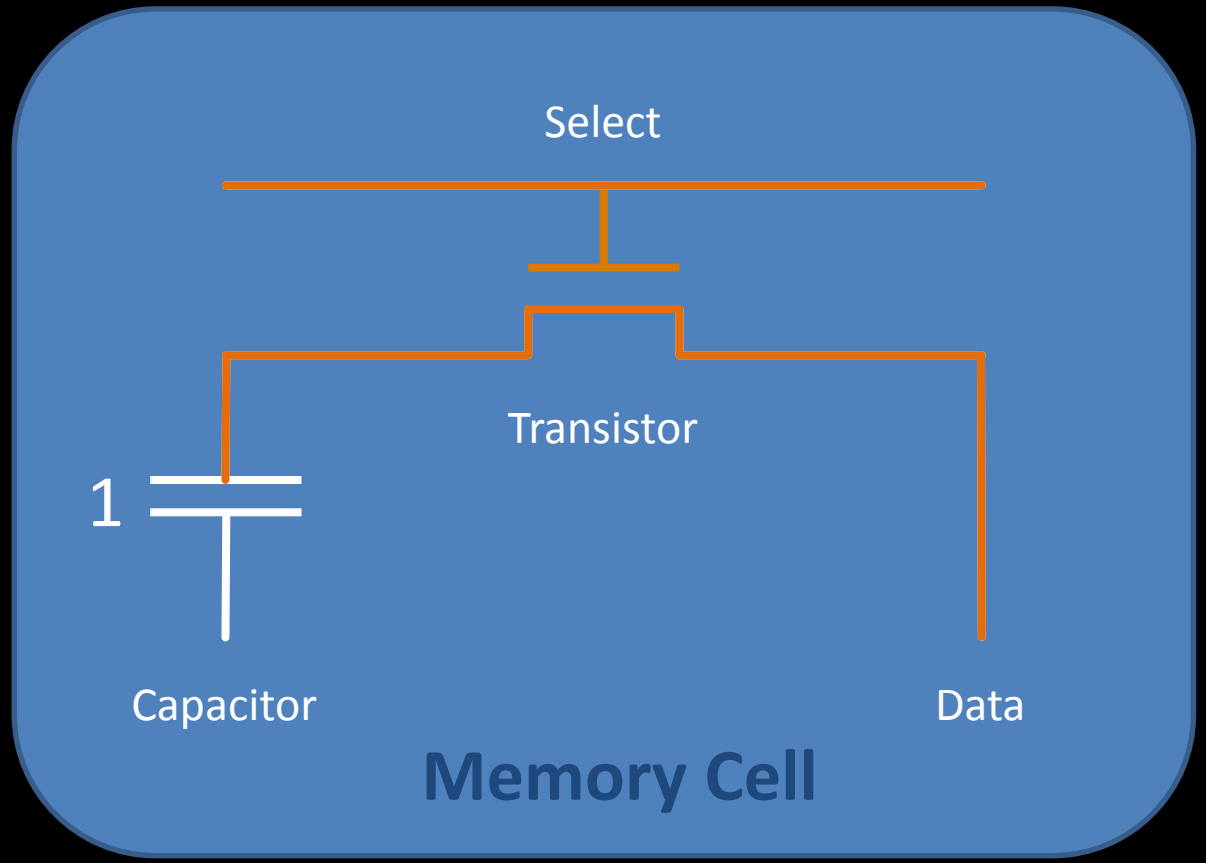
Read Bit

DRAM: Storing a Single Bit



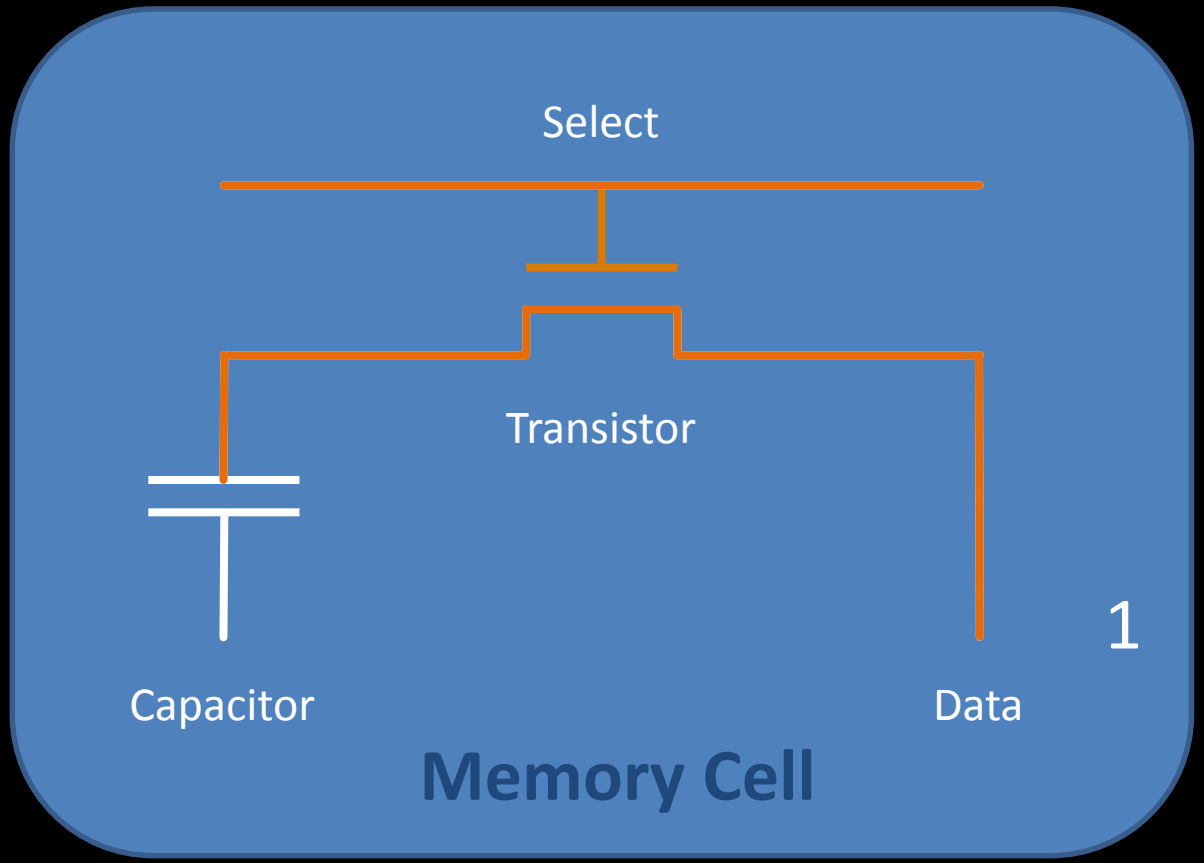
Read Bit

DRAM: Storing a Single Bit



Read Bit

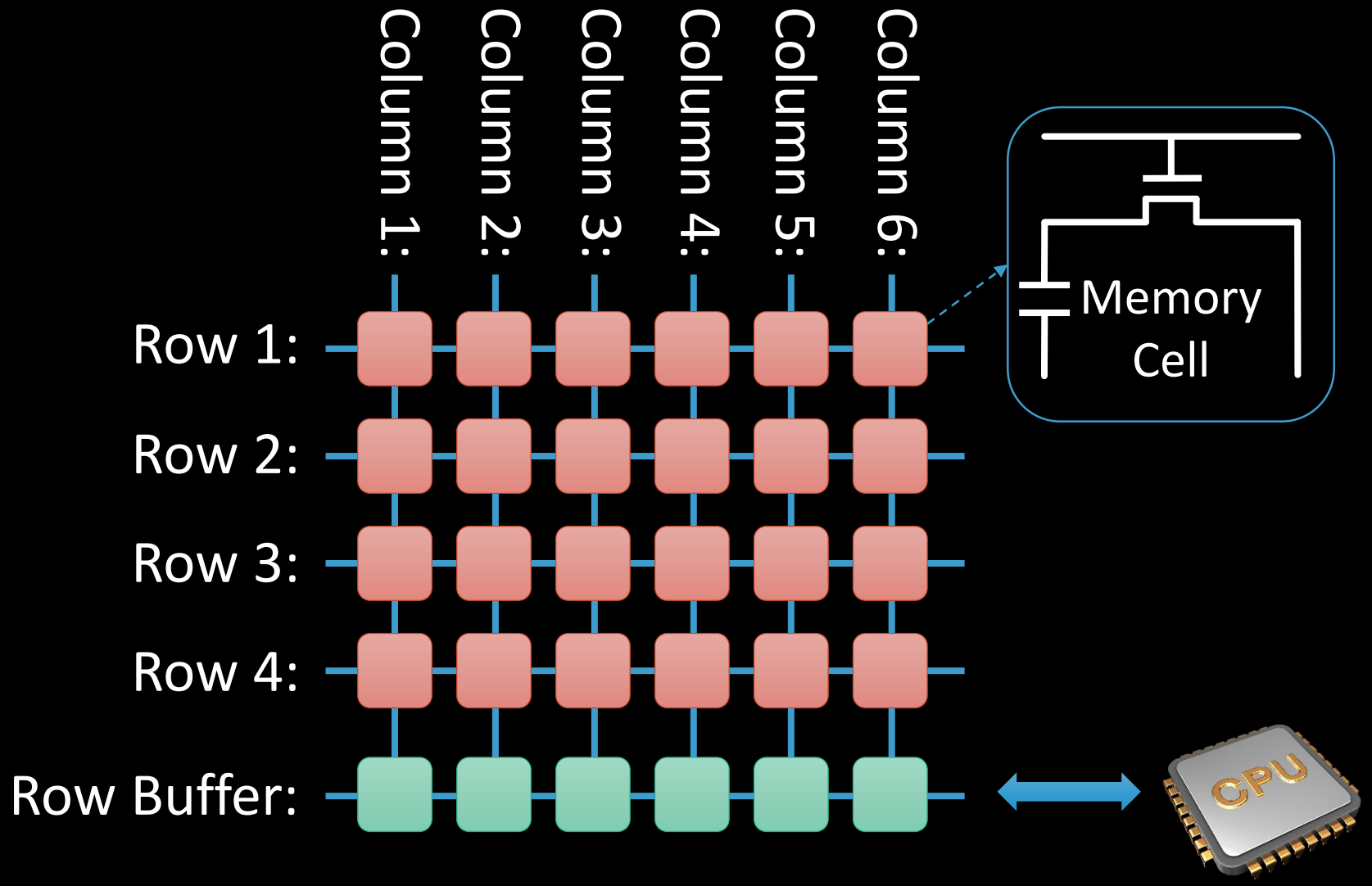
DRAM: Storing a Single Bit



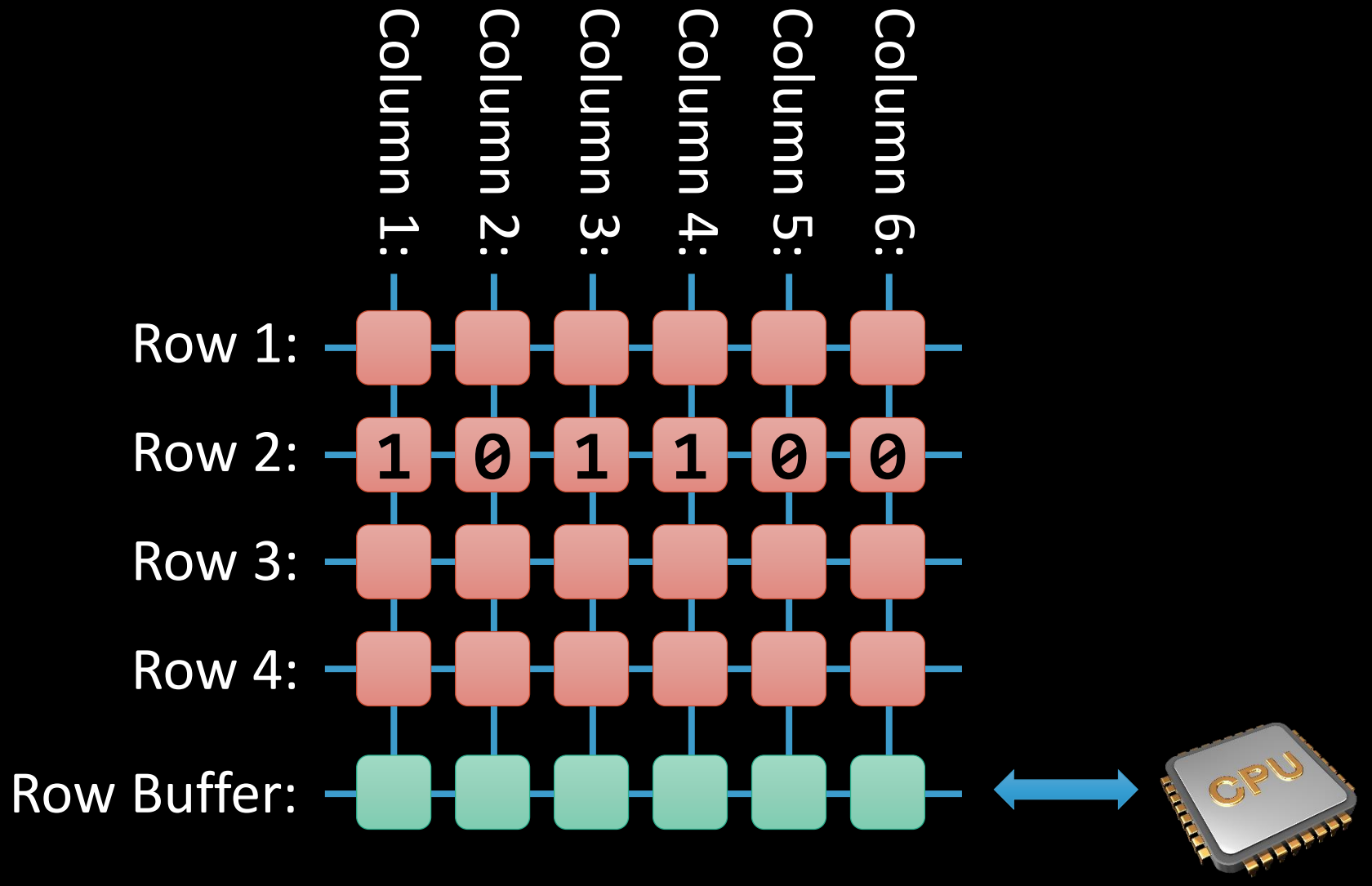
1

Read Bit

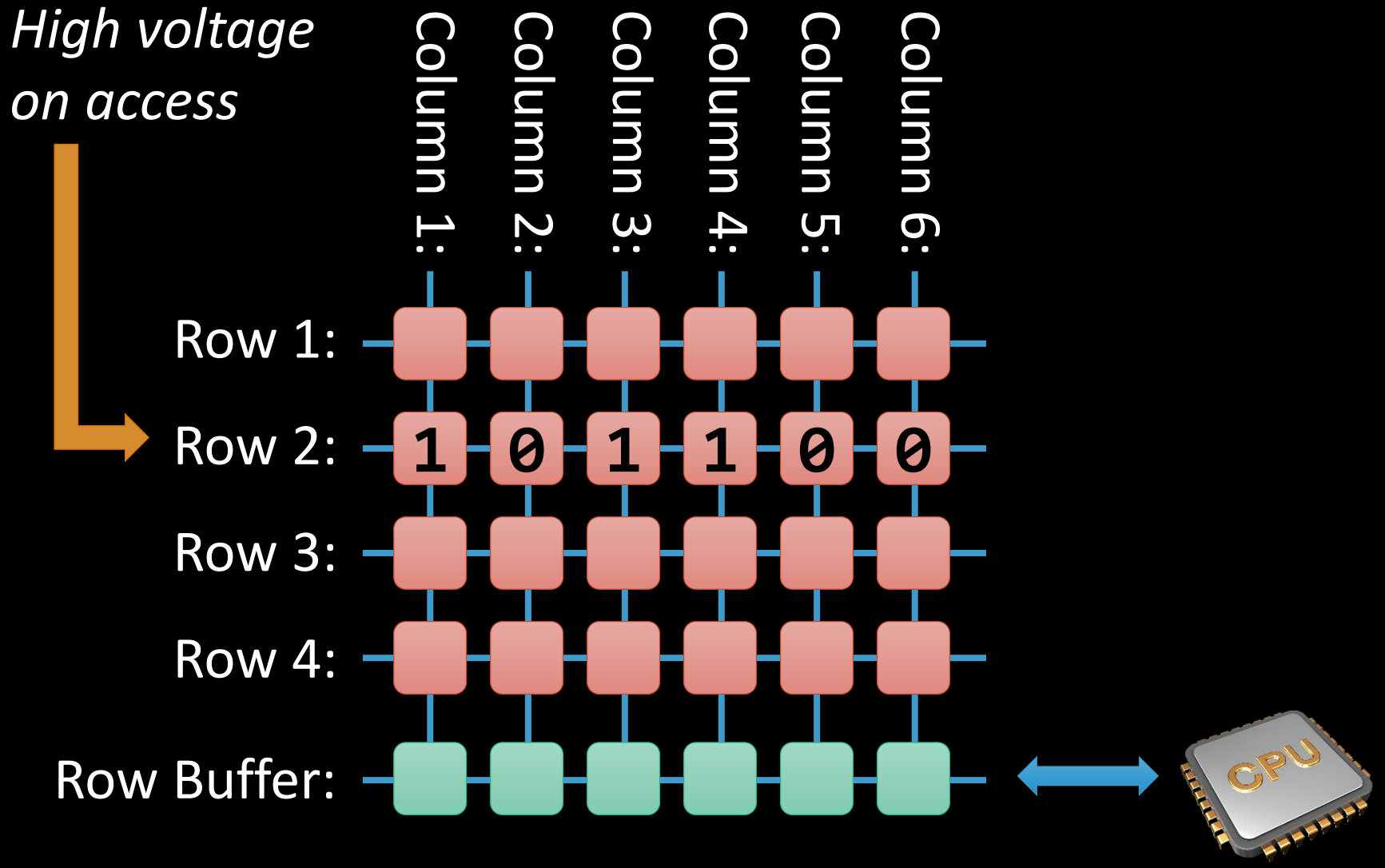
DRAM: Organization inside a Bank



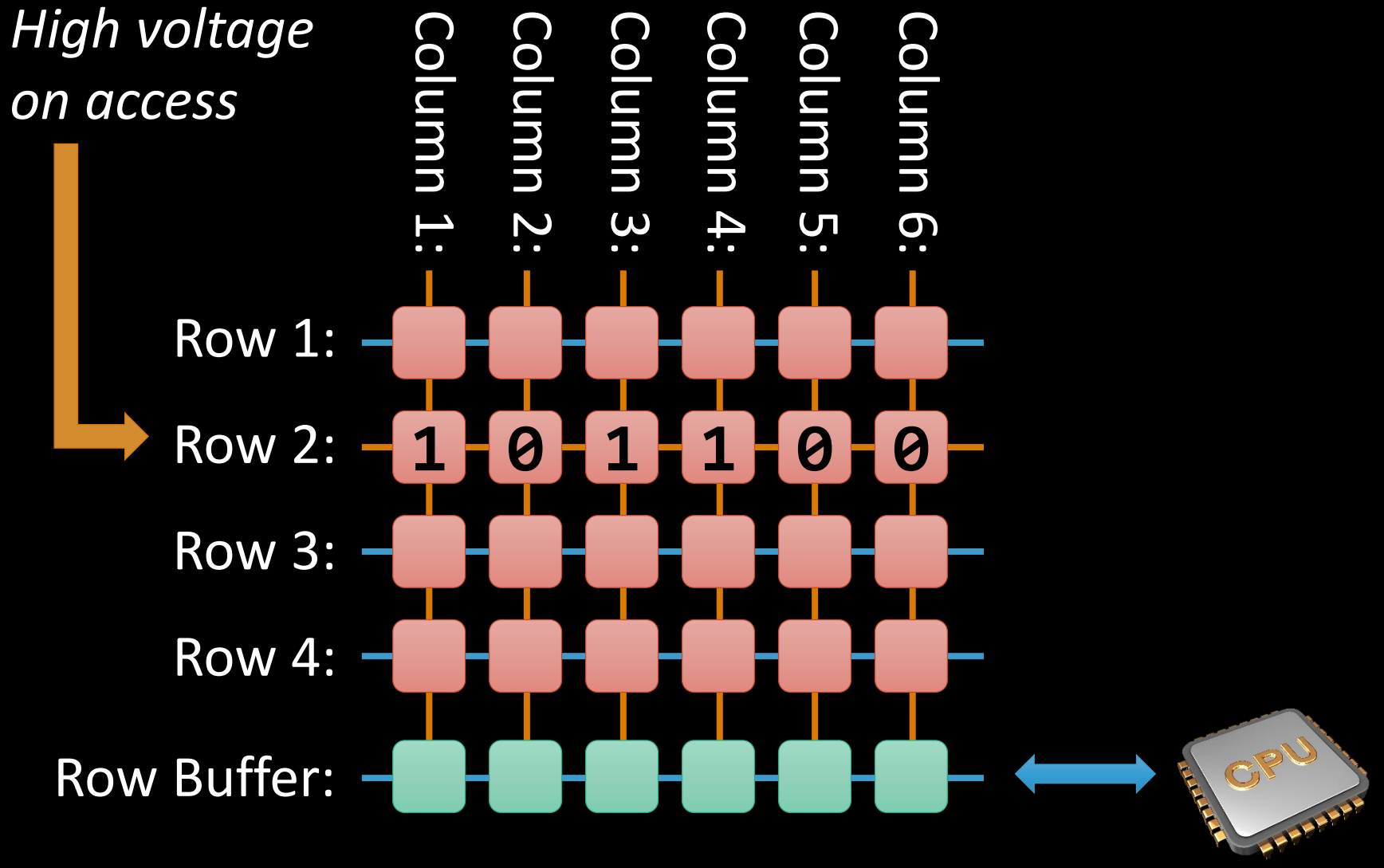
DRAM: Read Access



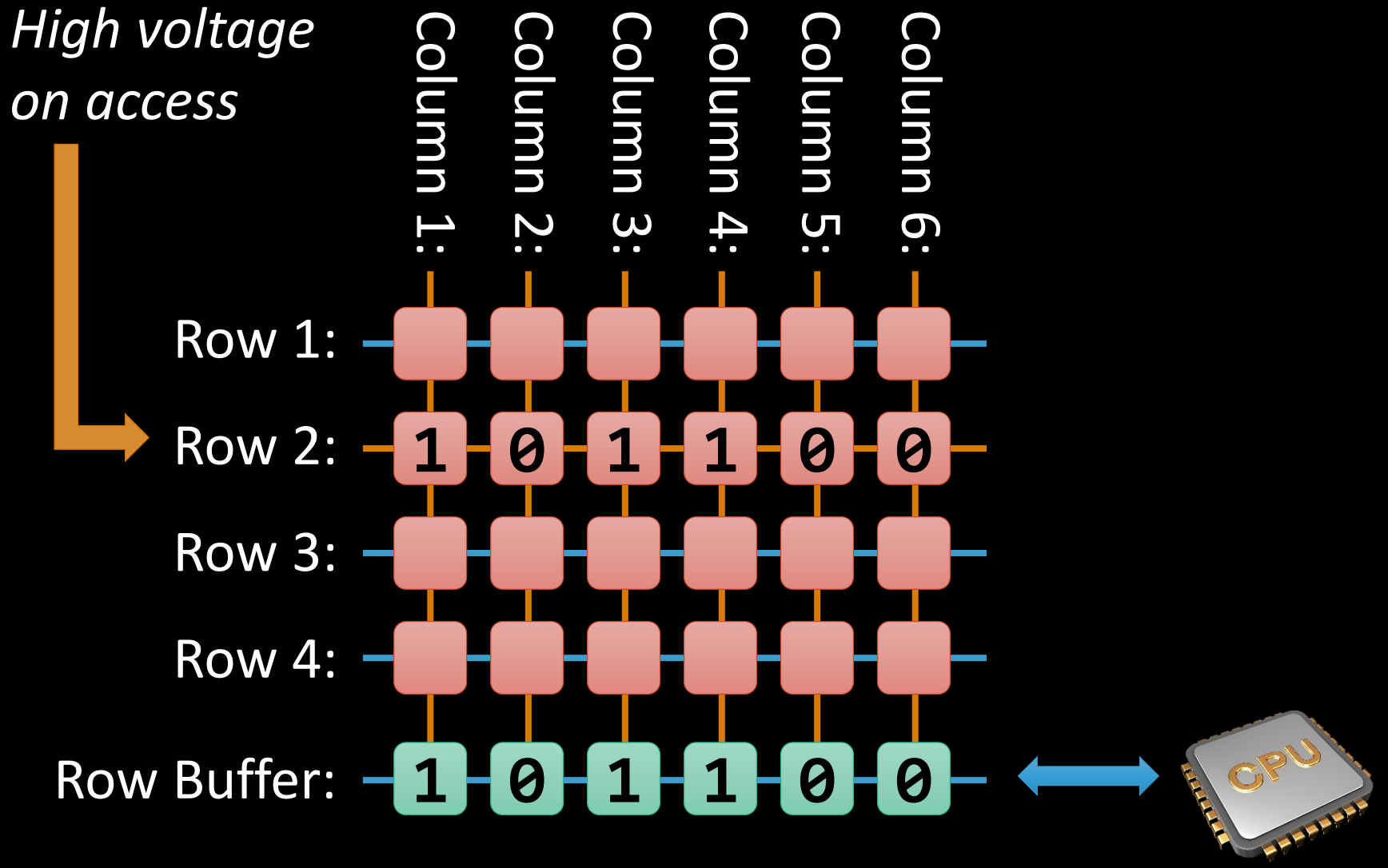
DRAM: Read Access



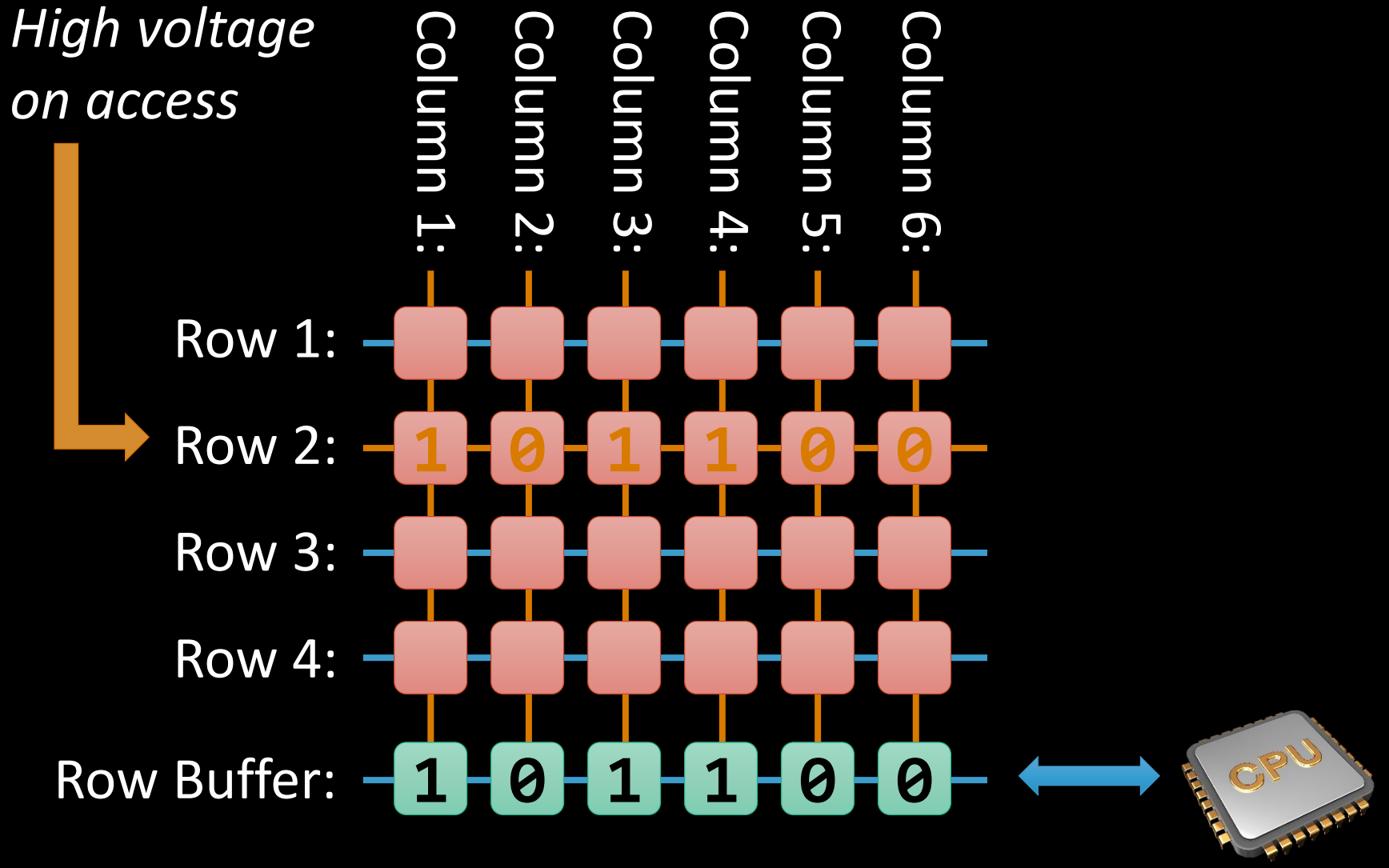
DRAM: Read Access



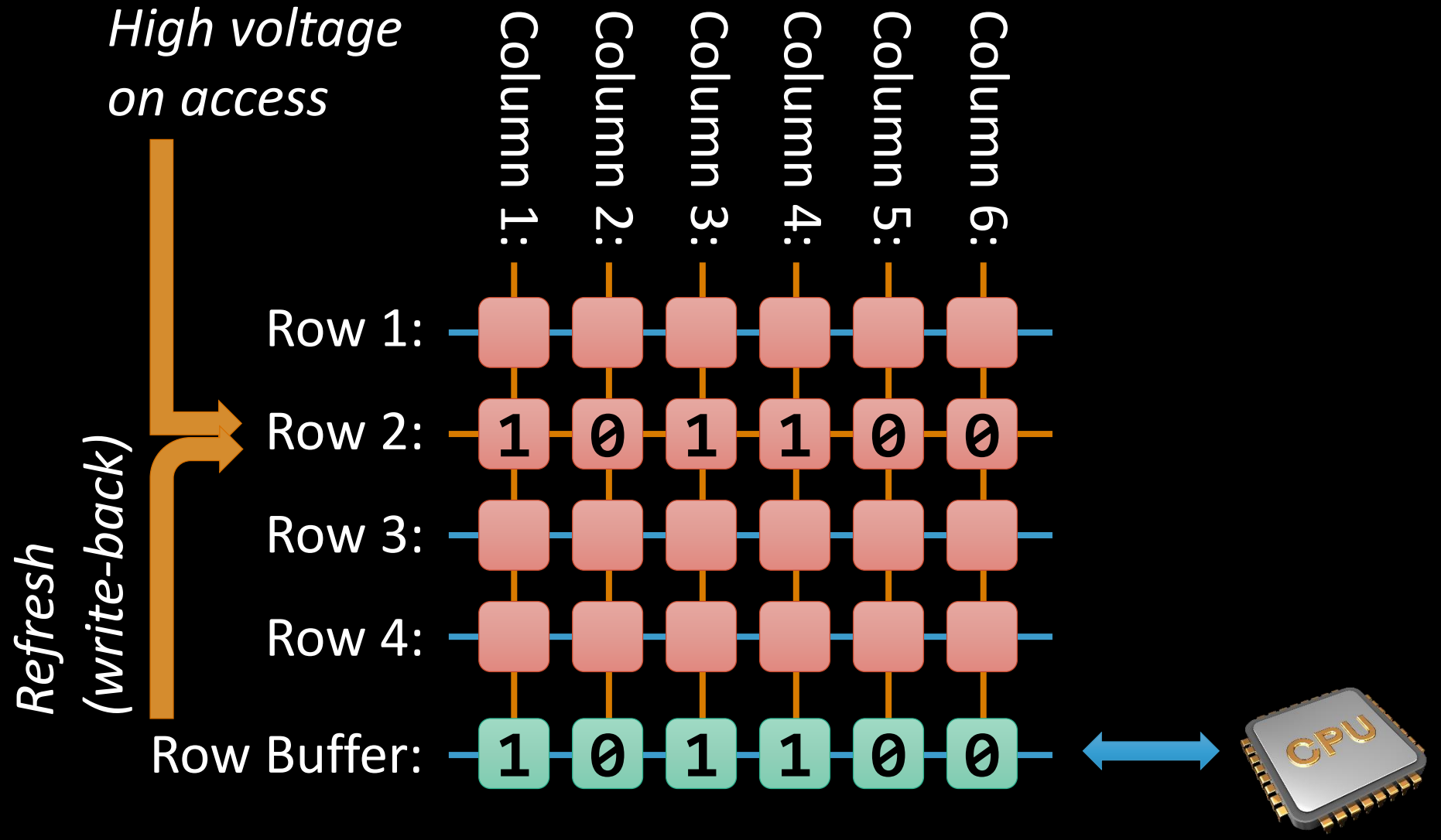
DRAM: Read Access



DRAM: Read Access

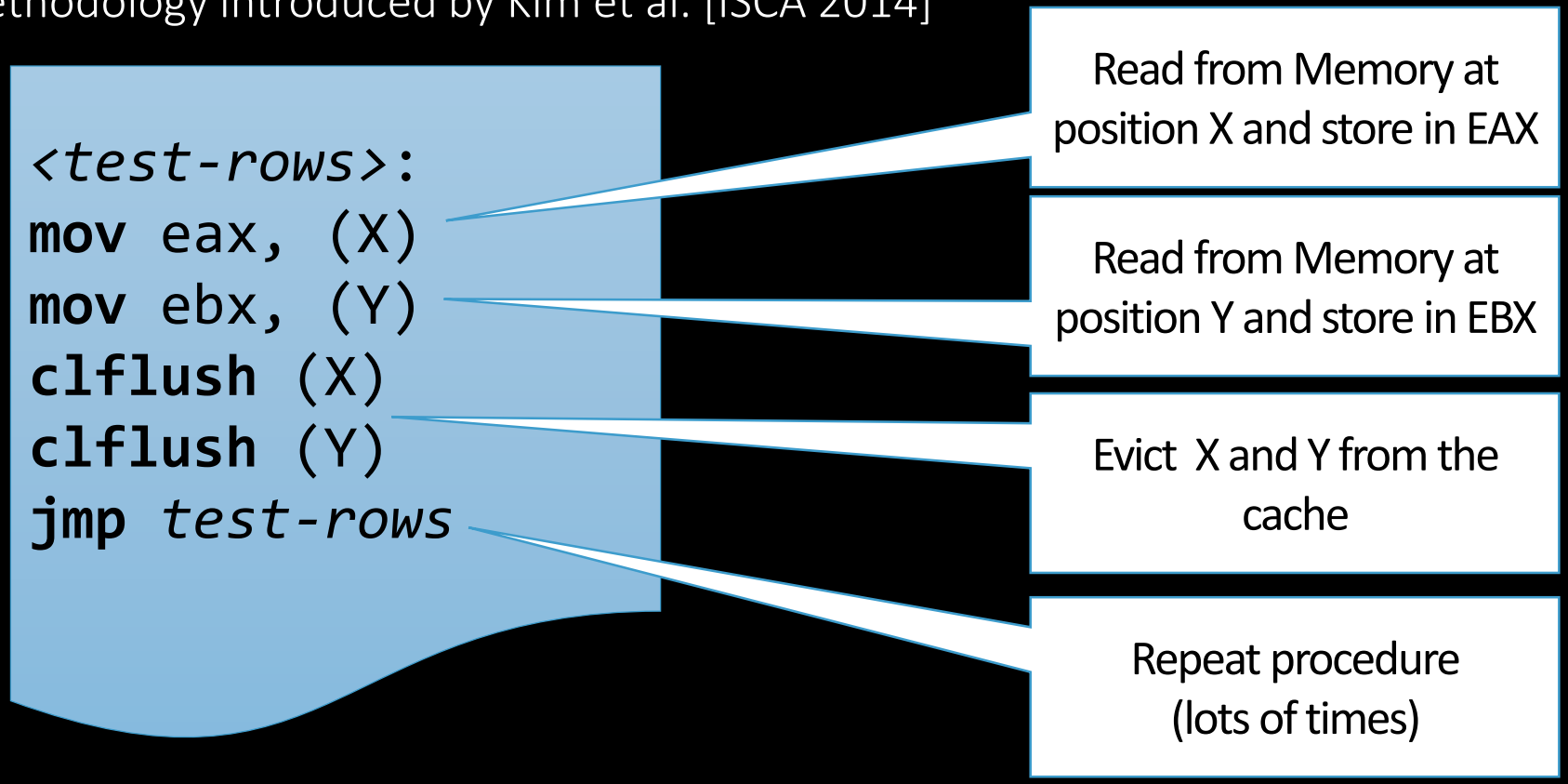


DRAM: Read Access



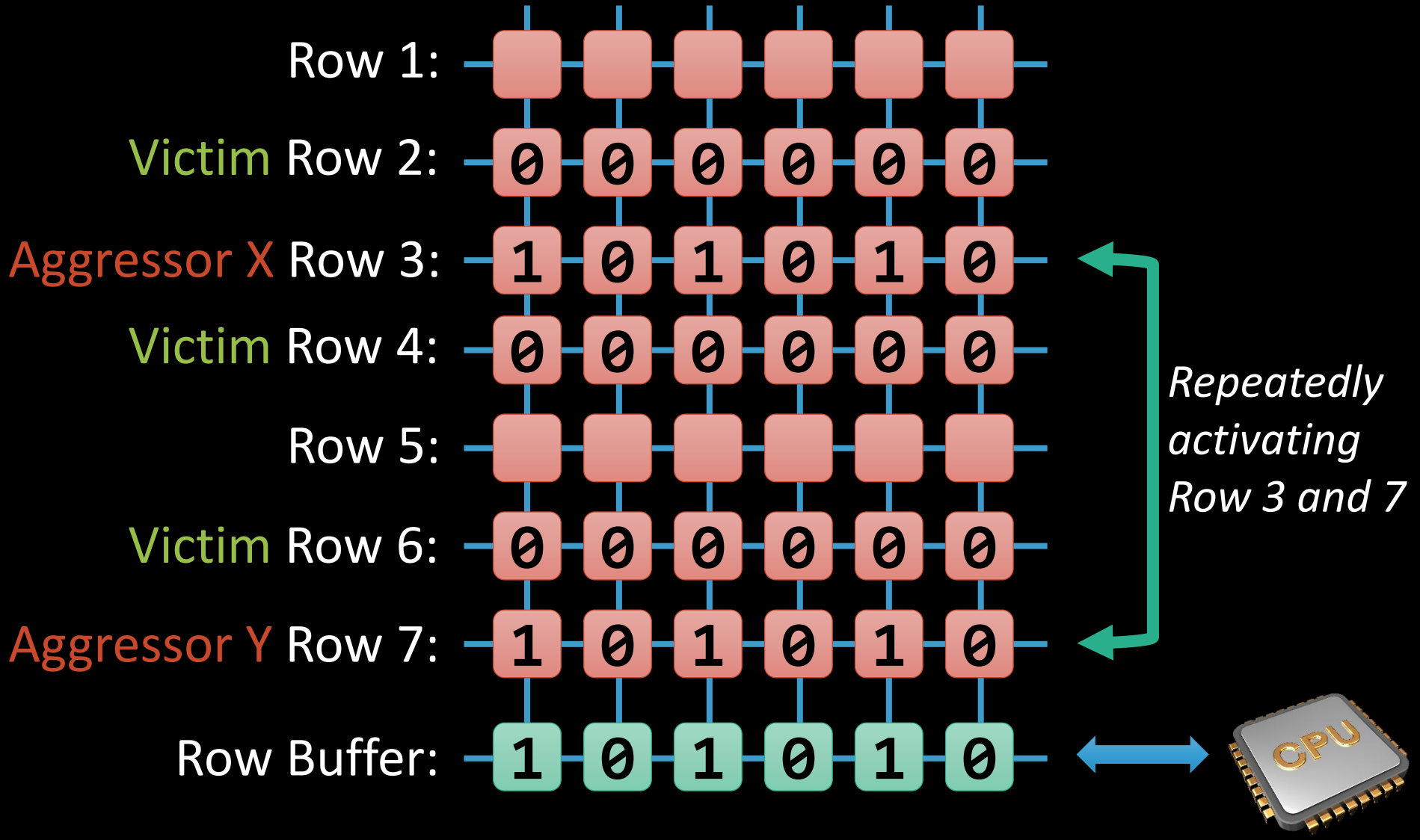
How Reliable is DRAM hardware?

- Testing methodology introduced by Kim et al. [ISCA 2014]

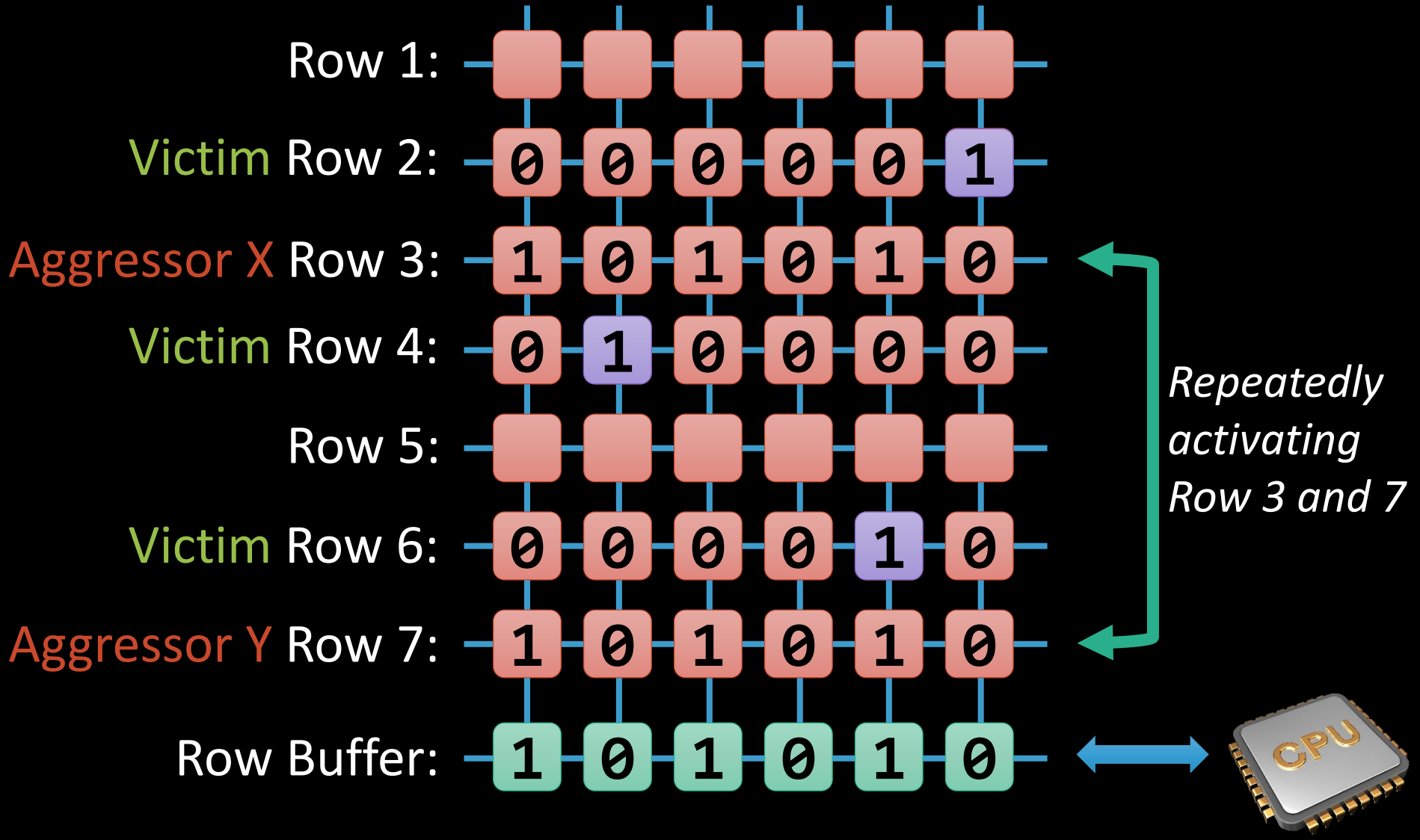


- X and Y need to be on the same bank but in different rows; general pattern: Y = X + 8MB

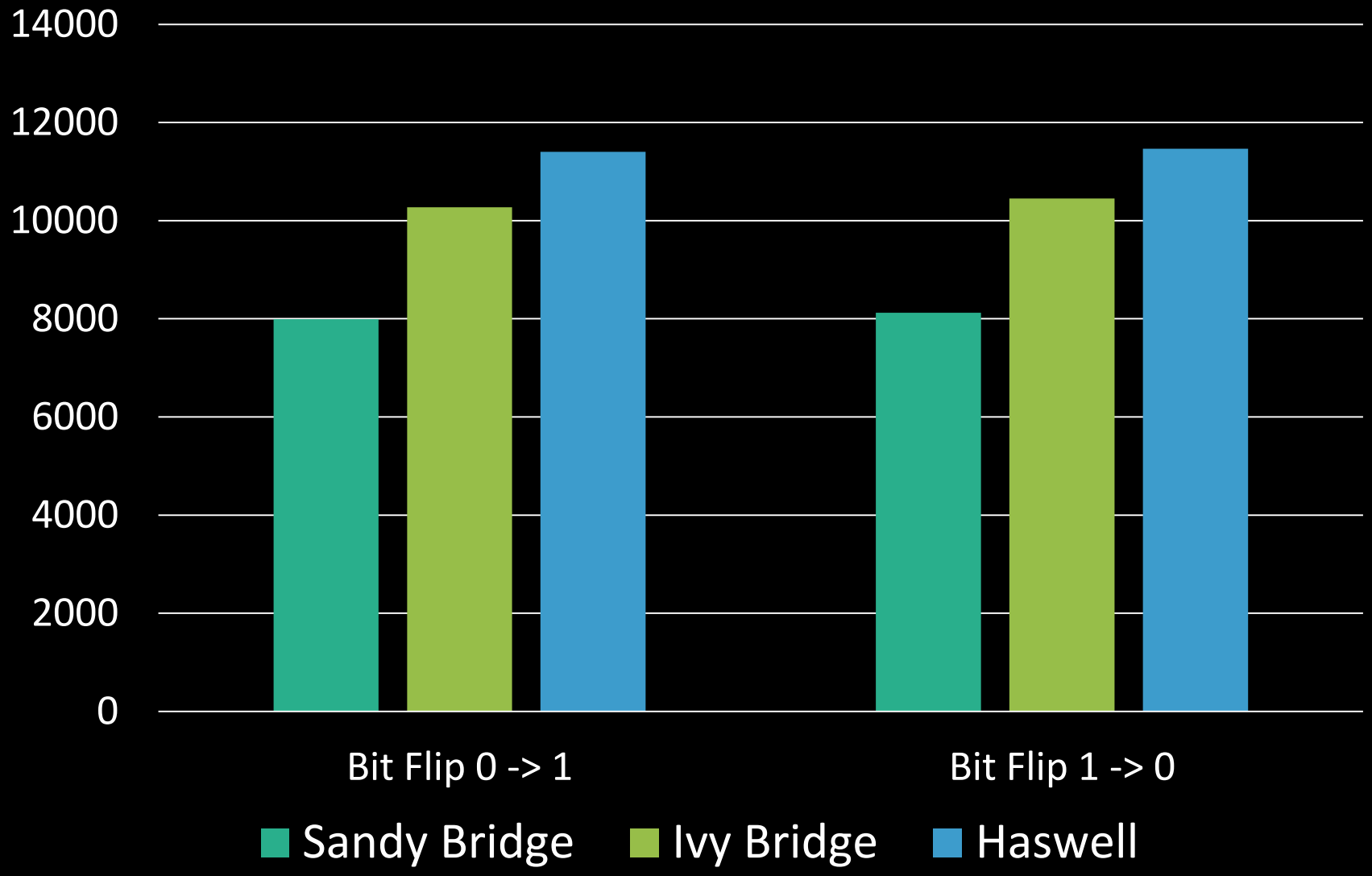
Single-Sided Rowhammer



Single-Sided Rowhammer



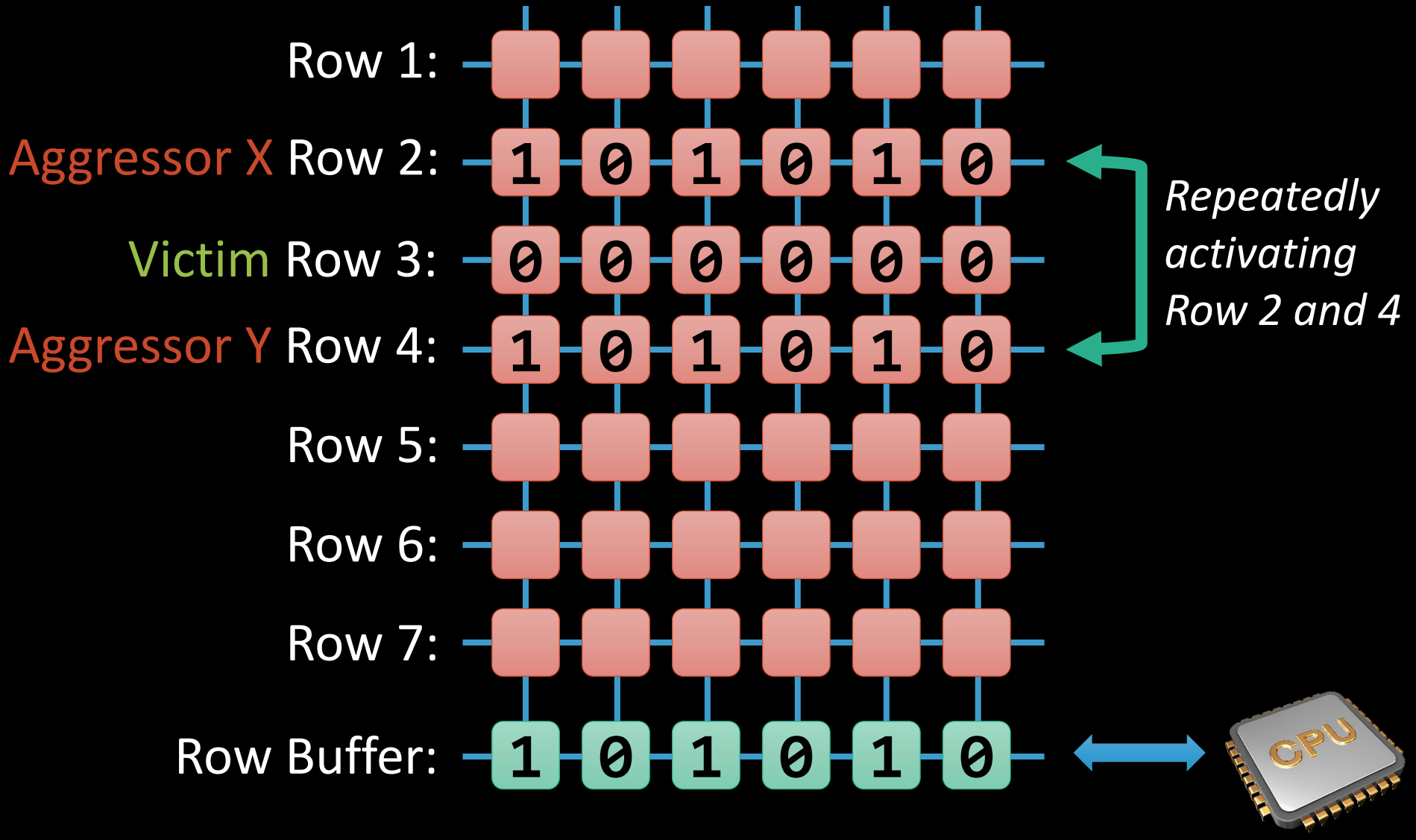
Many Bit Flips Observed



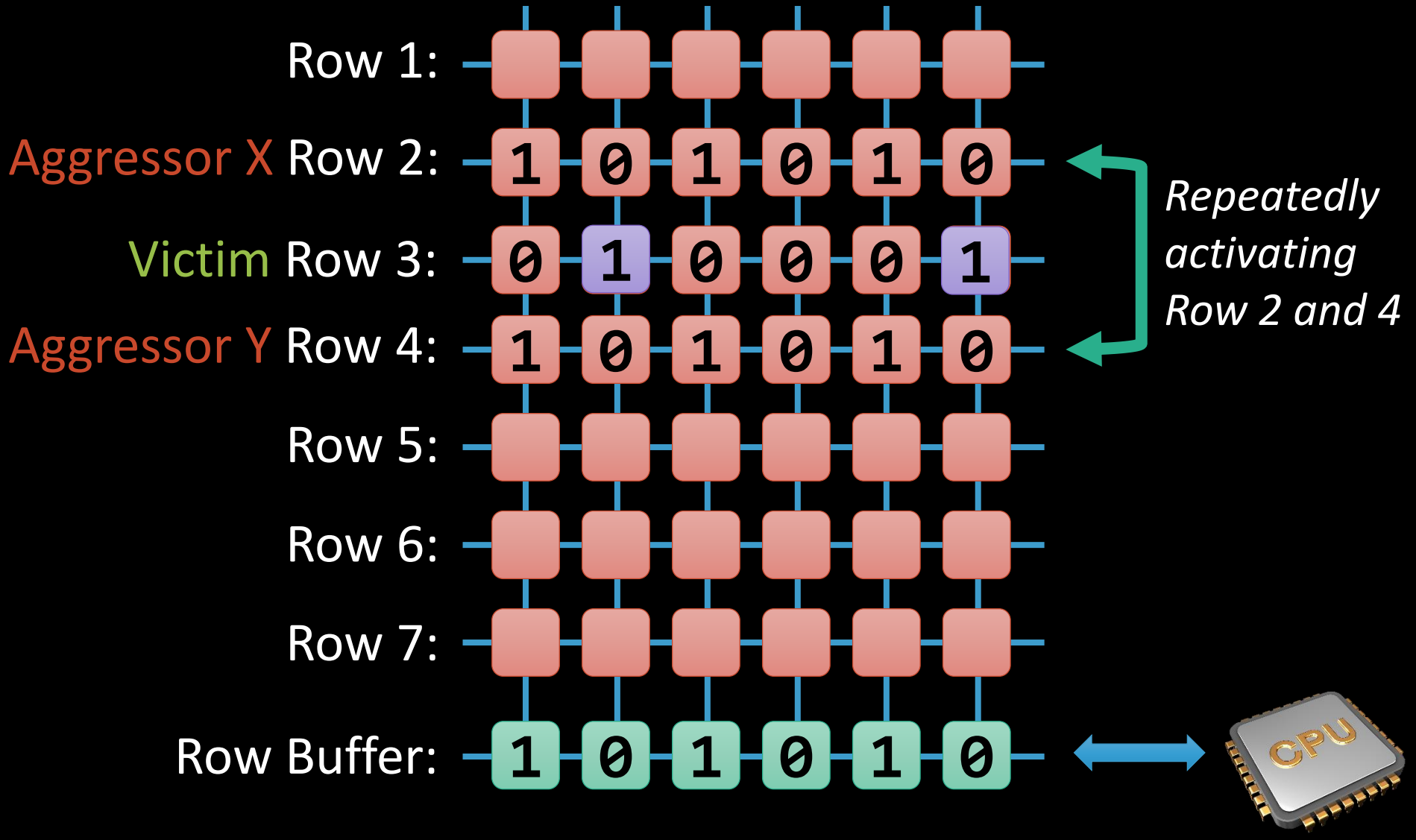
Source: Kim et al., ISCA 2014

Once it’s bad, it gets worse.

Double-Sided Rowhammer



Double-Sided Rowhammer



How Dangerous are Bit Flips?

Rowhammer Timeline and Attacks

SELECTED

2009

DRAM Errors in the Wild
Schroeder et al. (SIGMETRIC)

2012

Row hammer refresh command
Intel Corporation (US Patent)

2014

Flipping Bits in Memory
Kim et al. (ISCA)

2015

Exploiting the DRAM rowhammer bug
Seaborn and Dullien (Black Hat US)

2016

Drammer
van der Veen et al. (CCS)

Improved Rowhammer Attacks
Qiao and Seaborn (HOST)

Flip Feng Shui
Razavi et al. (USENIX Sec.)

Rowhammer.js
Gruss et al. (DIMVA)

One Bit Flips, One Cloud Flops
Xiao et al. (USENIX Sec.)

Related Work: First Defenses

Prohibit CLFLUSH

! Ineffective [Qiao and Seaborn, HOST 2016]

ANVIL [Aweke et al. ASPLOS 2016]

! Heuristic approach (overhead & false positives)

Software

Access Control

! Ineffective [Aweke et al. ASPLOS 2016]

Increase Refresh Interval

! Modifies Hardware (costly & legacy problem)

Probabilistic Adjacent Row Activation [Kim et al. ISCA 2014]

Reviewing Attacker Assumptions

Software

Access Control

1. Vulnerable Cells

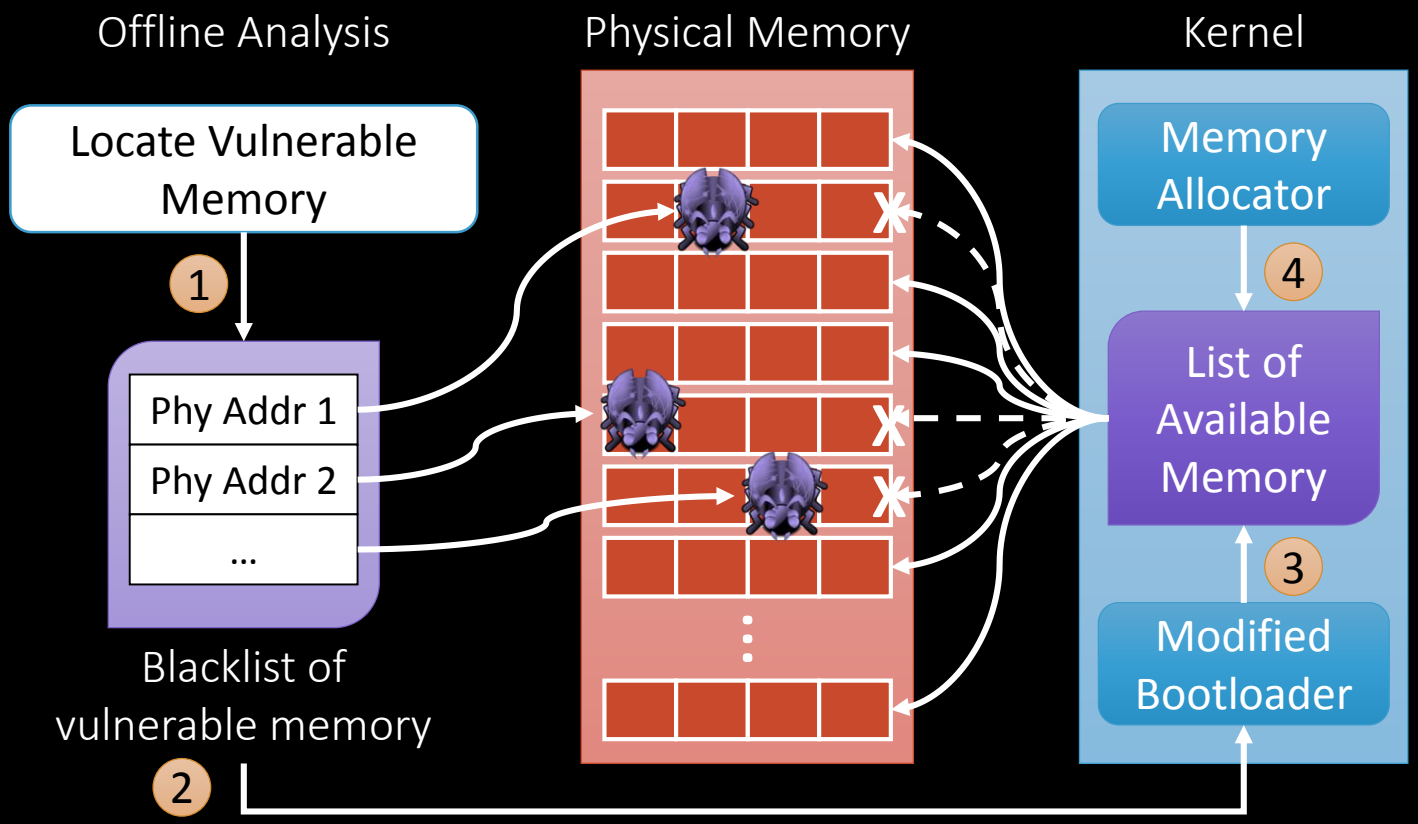
2. Co-location

Our Initial Approach:

Blacklisting

Deactivate Vulnerable Physical Memory

Initial Tests with Blacklisting



For more details check our technical report at
https://arxiv.org/abs/1611.08396

Problems of Blacklisting

- Coverage
- Progression of vulnerable cells over time
- Memory overhead for other systems than our test systems unclear

https://arxiv.org/abs/1611.08396

Our Generic Approach:

CATT

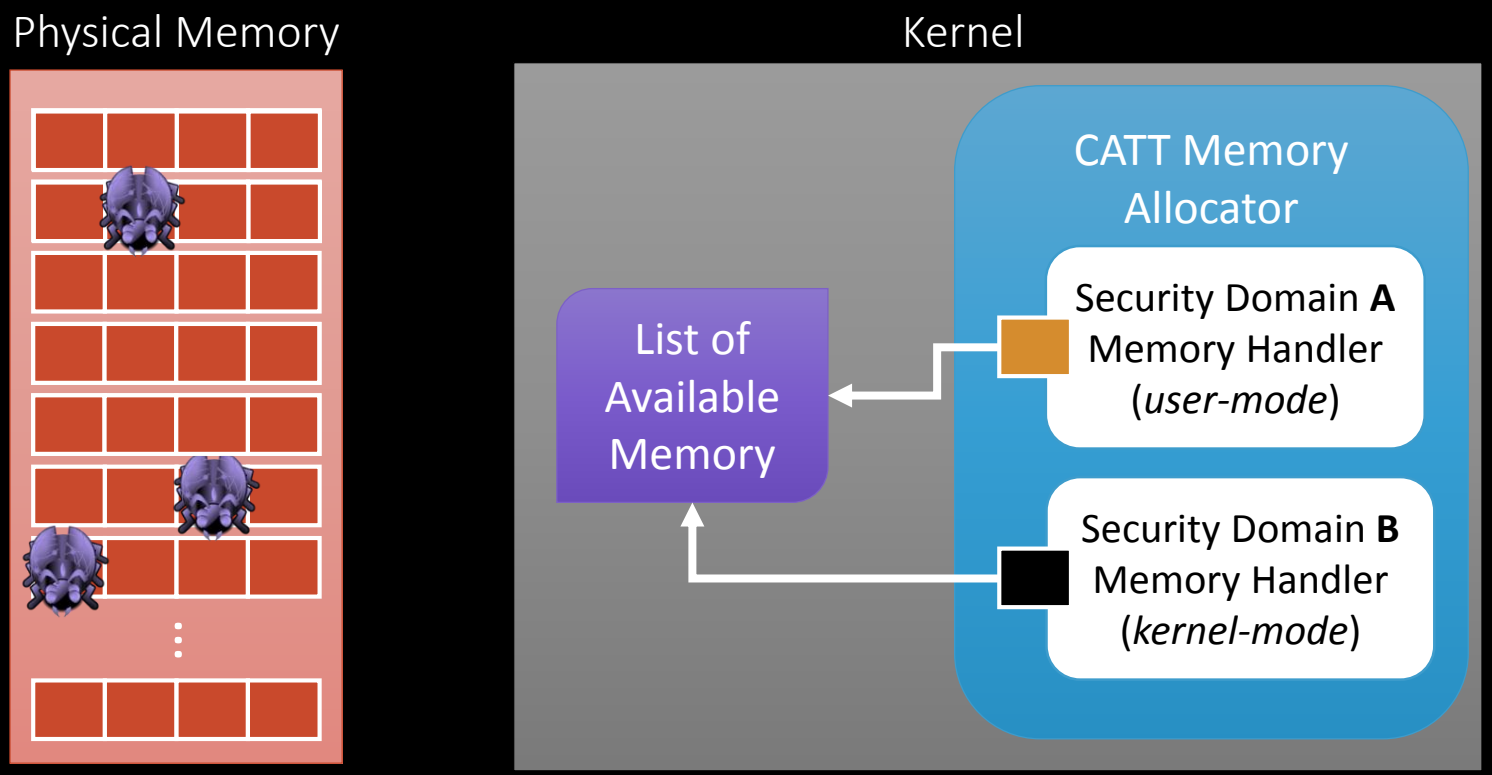
Spatially Isolate Physical Memory
in Software

CATT: Contributions and Challenges

- First defense that enables spatial memory isolation
- Defines and manages different security domains
- Prototype Implementation
  - CATT for the Linux kernel
  - Tested using Real-World Setup
  - Extensive Performance and Security Evaluation

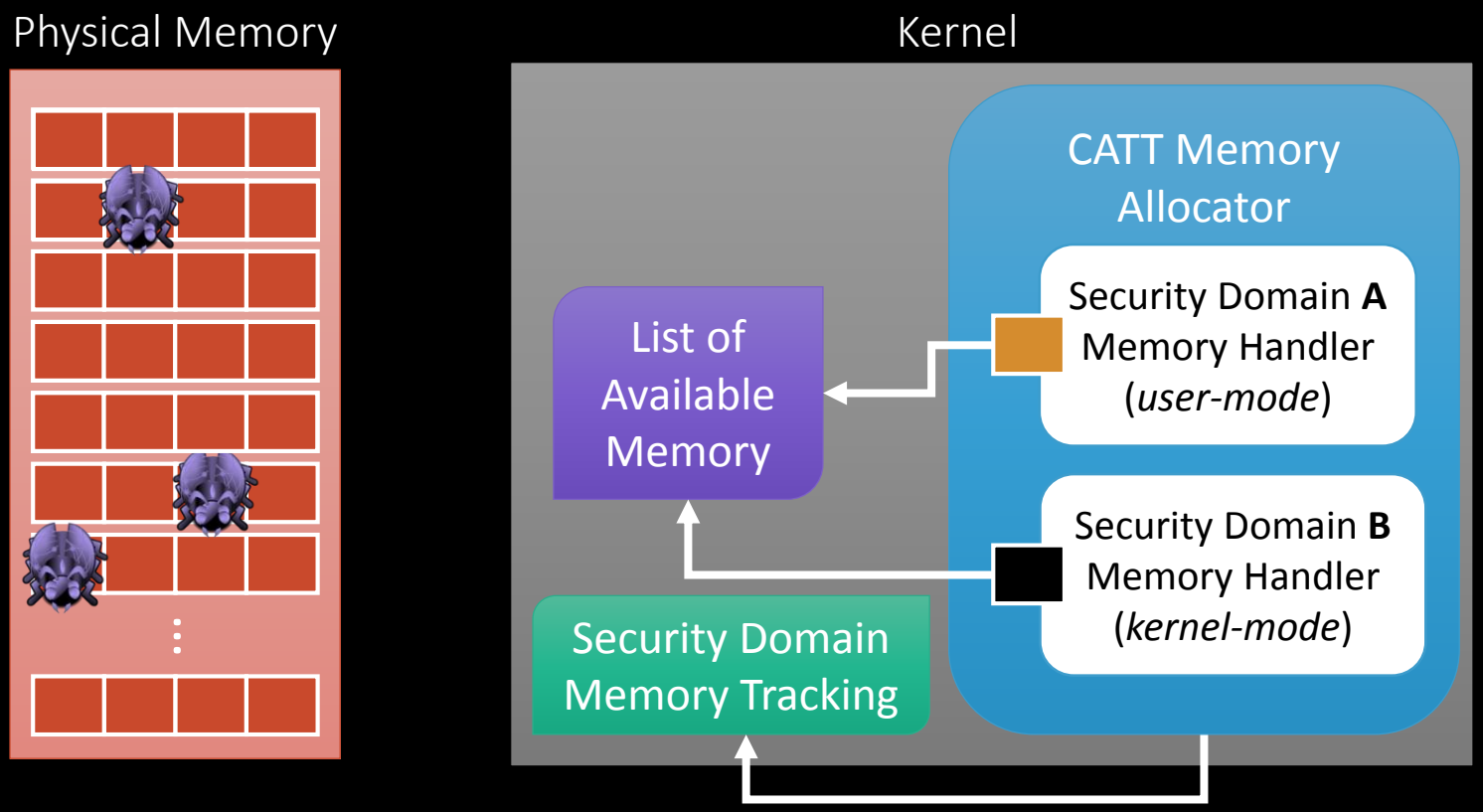
CATT: Design Idea

- Separate security domains *physically*



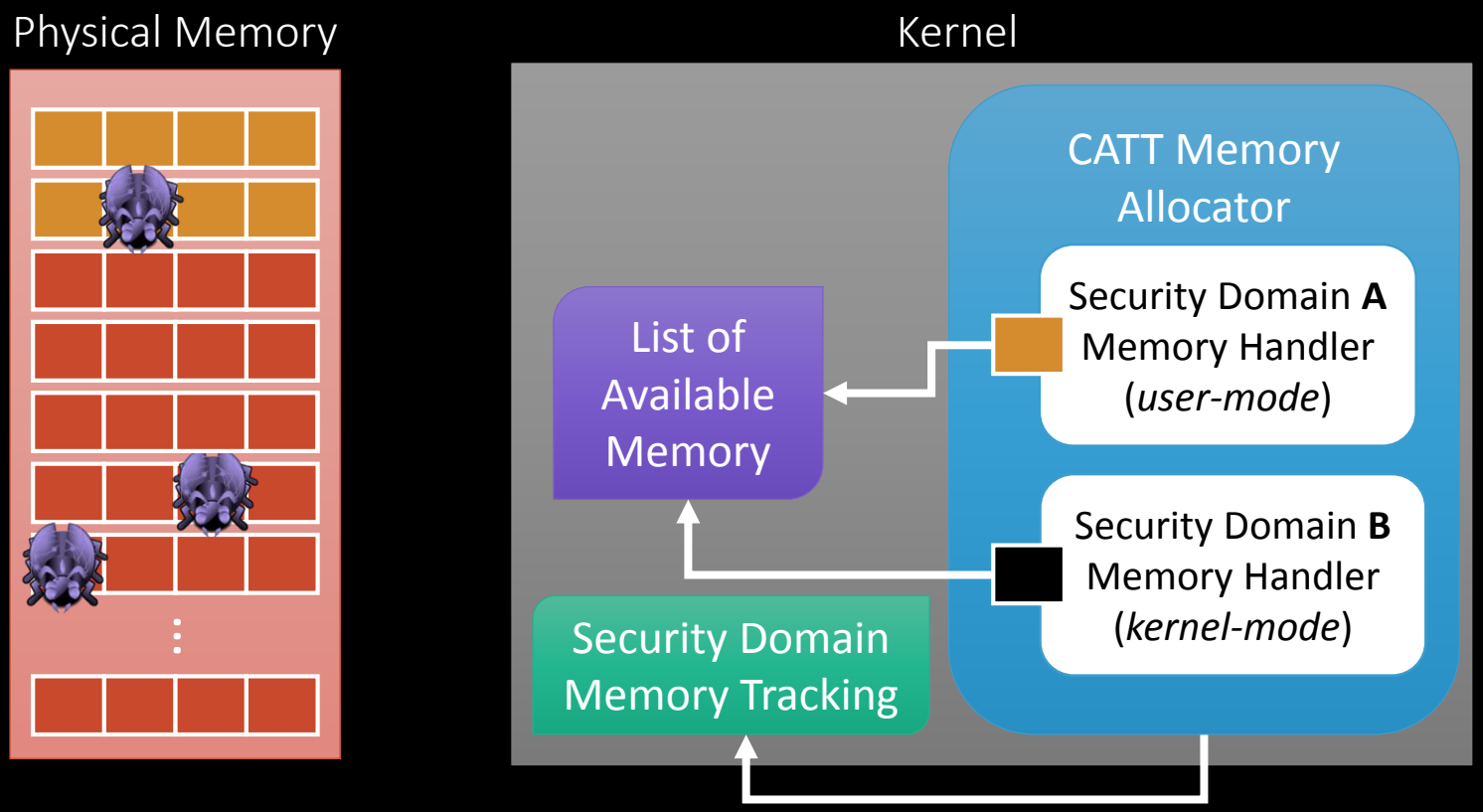
CATT: Design Idea

- Separate security domains *physically*



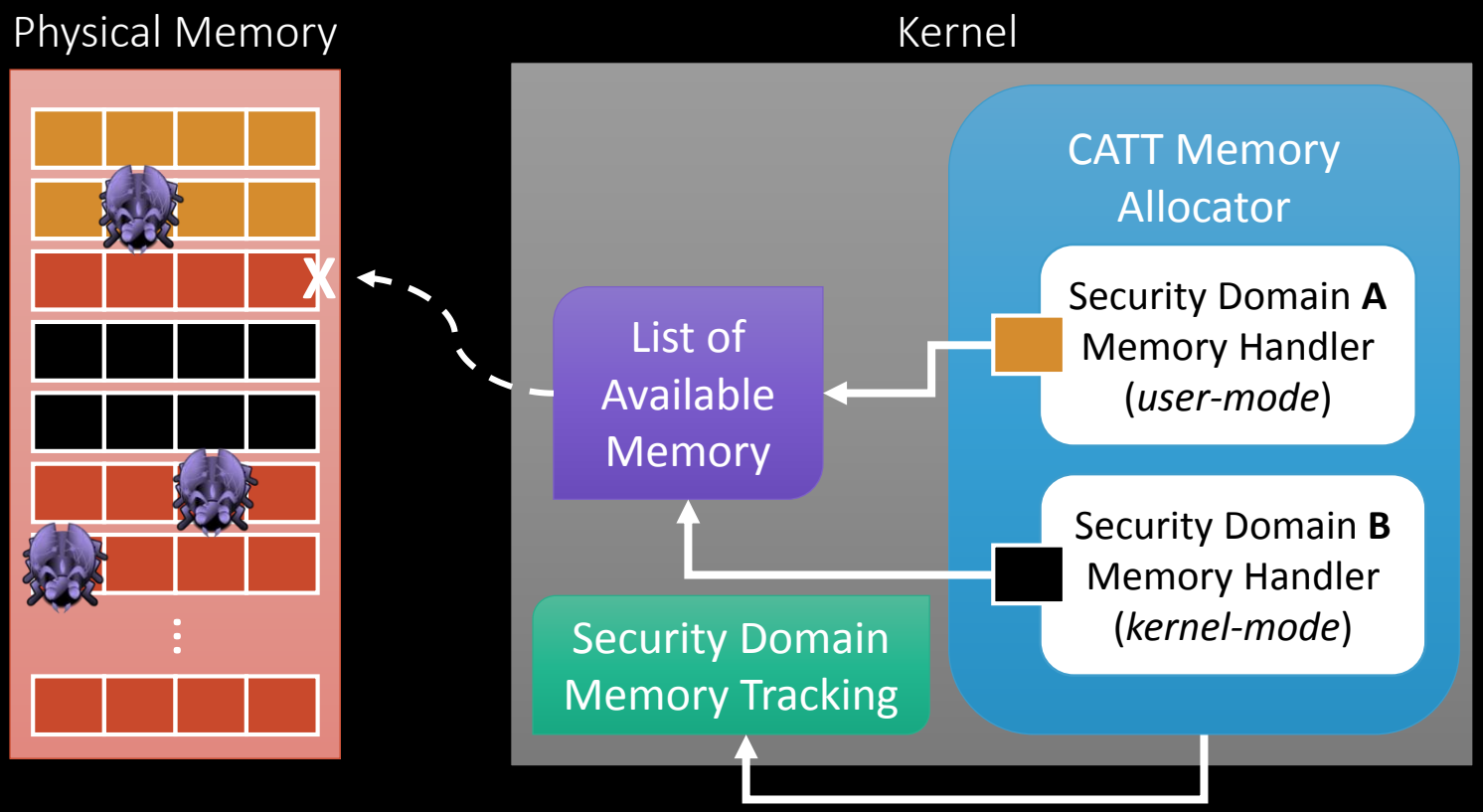
CATT: Design Idea

- Separate security domains *physically*



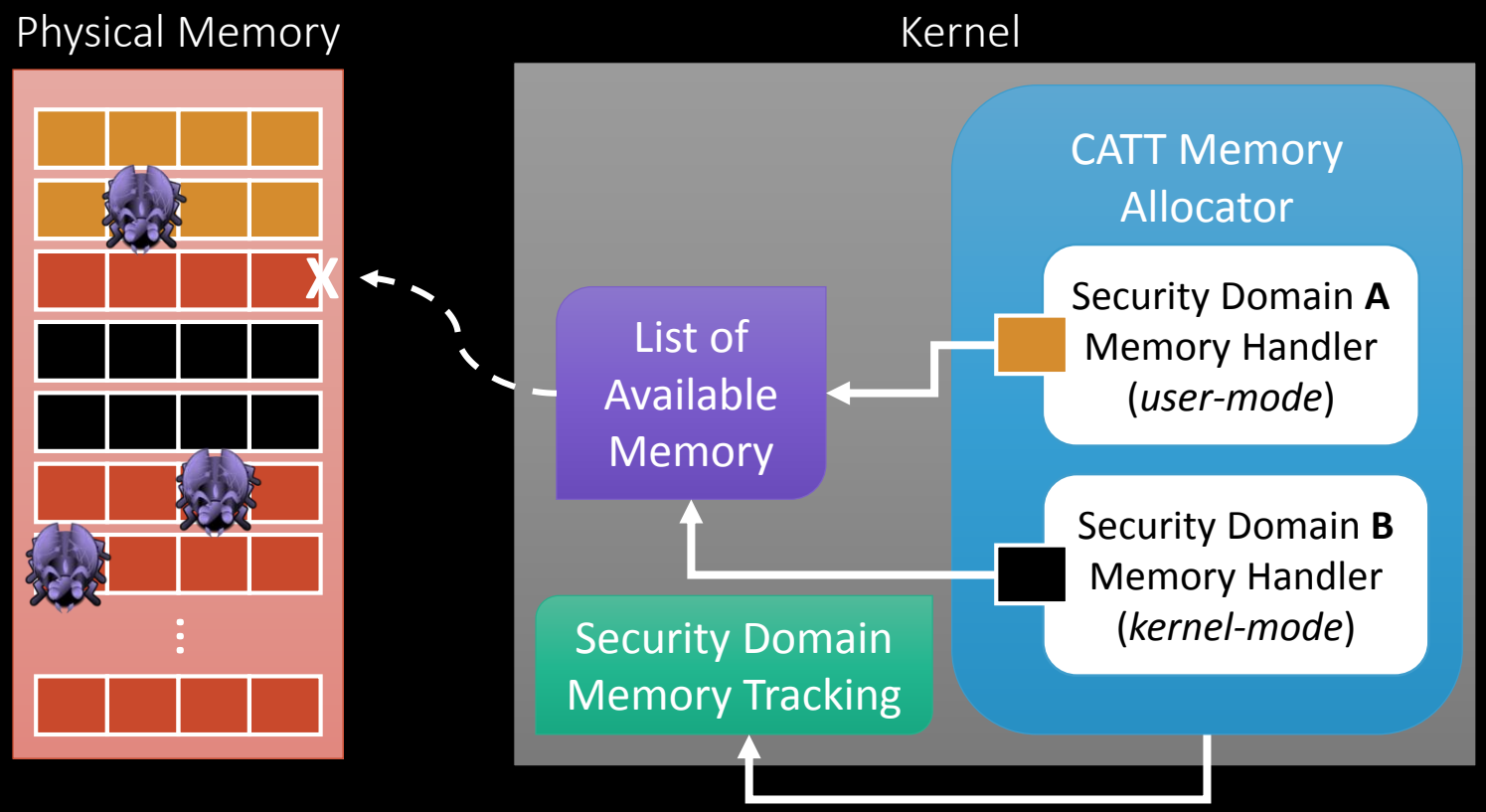
CATT: Design Idea

- Separate security domains *physically*



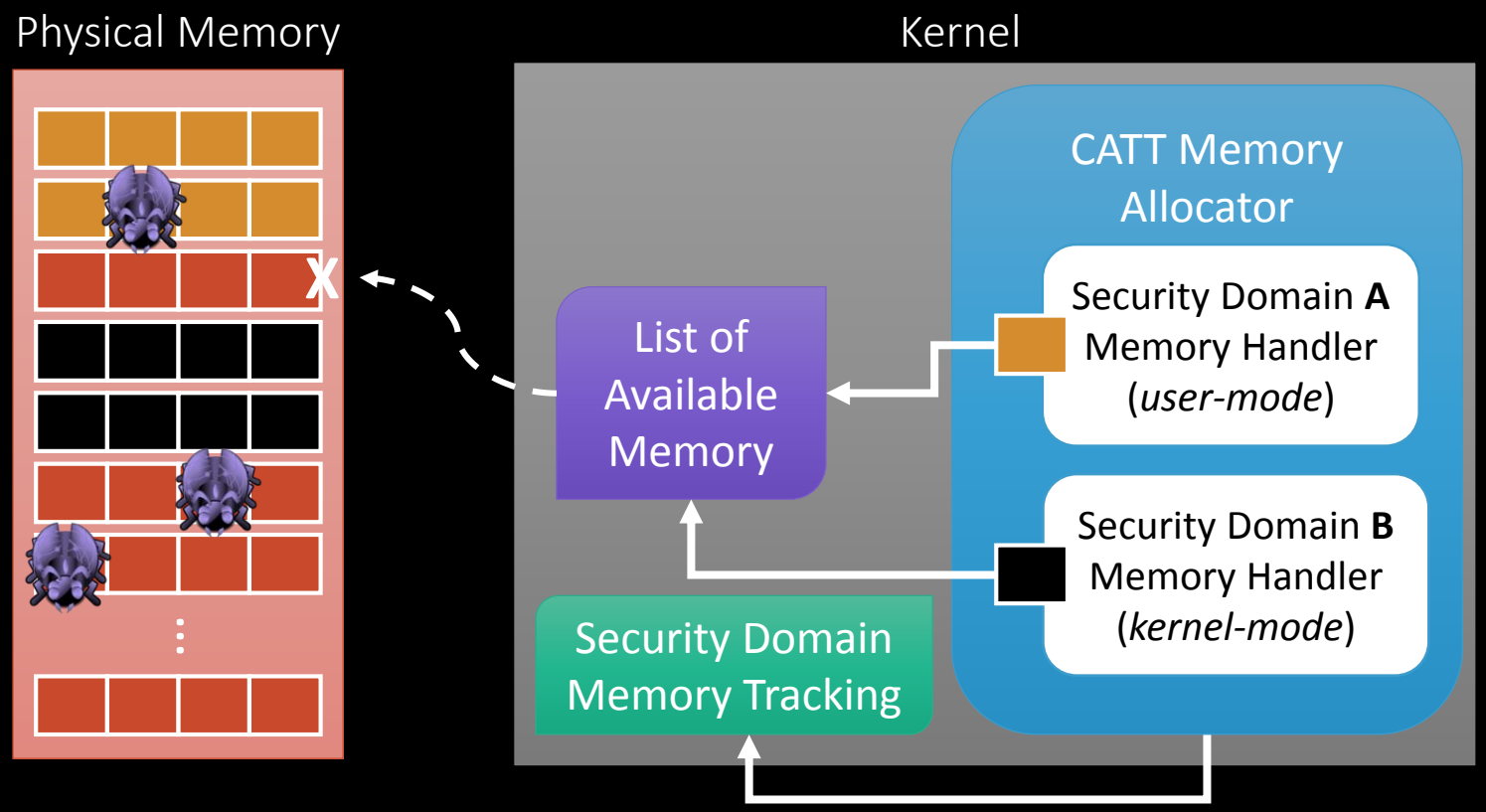
CATT: Design Idea

- Separate security domains *physically*
  - Attacker can still flip bits



CATT: Design Idea

- Separate security domains *physically*
  - Attacker can still flip bits
  - But only within her security domain

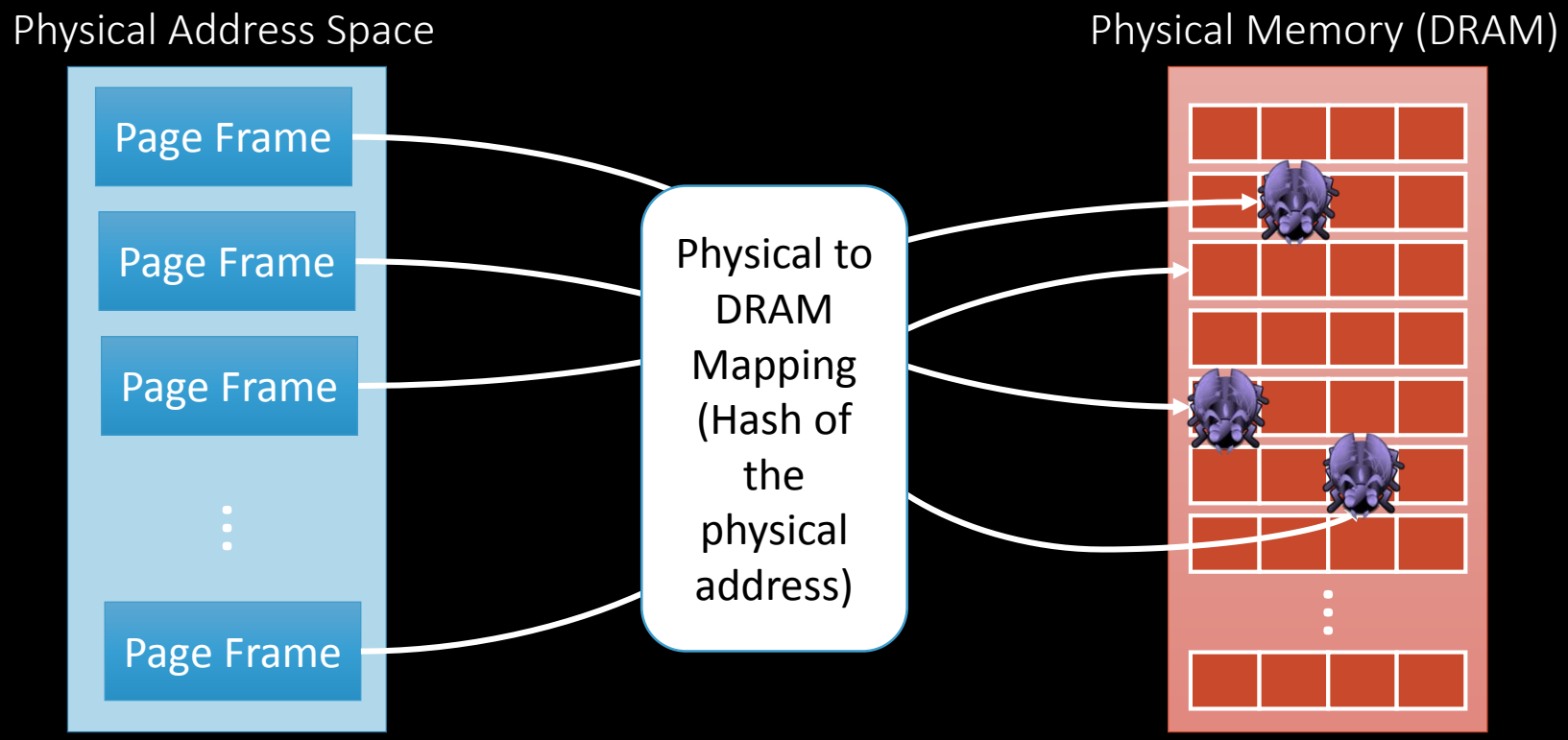


CATT: DRAM-aware Memory Allocation

- Rowhammer exploits physical co-location

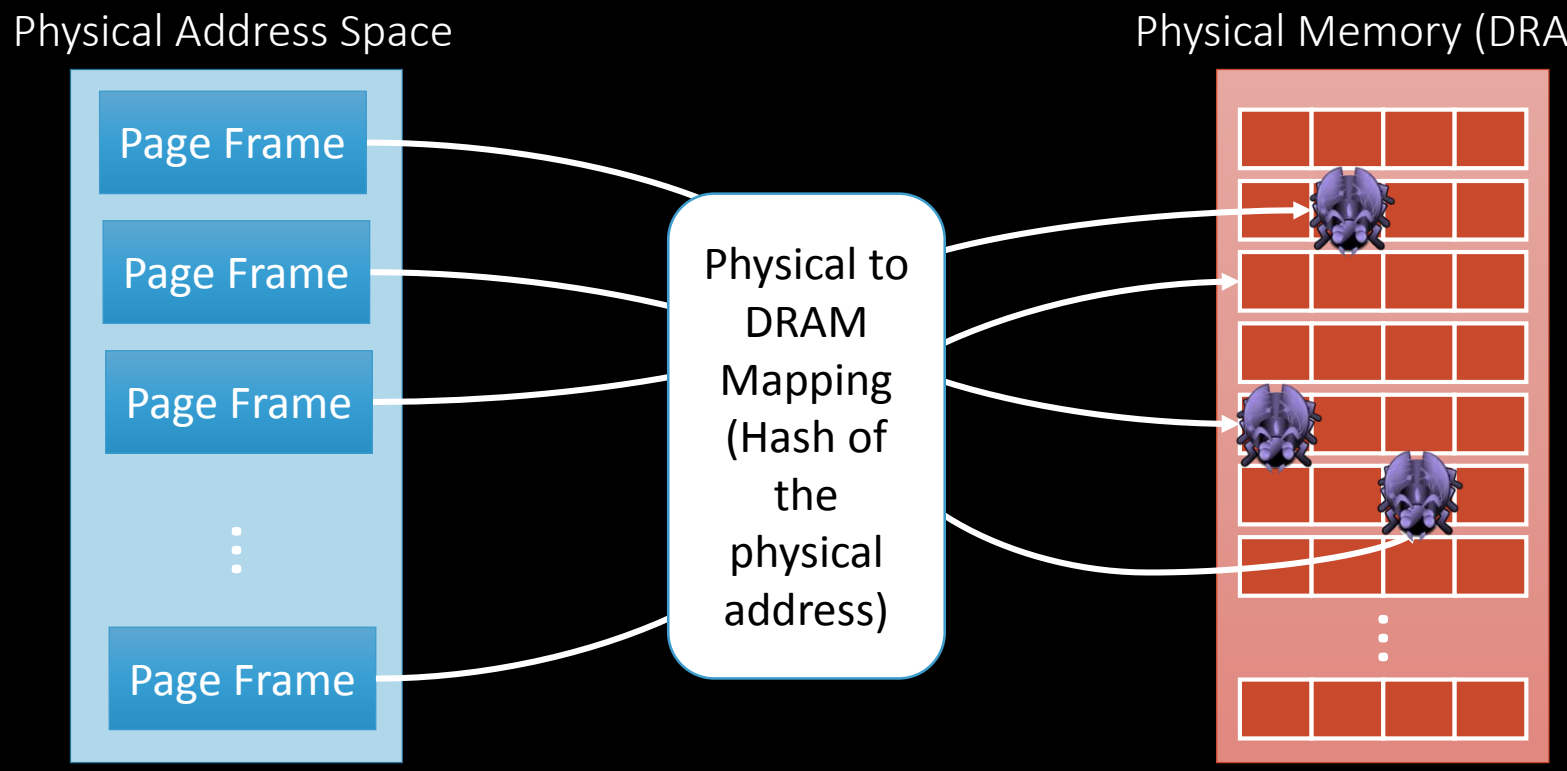
CATT: DRAM-aware Memory Allocation

- Rowhammer exploits physical co-location



CATT: DRAM-aware Memory Allocation

- Rowhammer exploits physical co-location



- If we know the mapping, we know where a Page Frame will be located in DRAM!

CATT: Implementation

- Prototype for the Linux kernel
  - Version 4.6
  - Completely transparent to applications

- Modifies physical page allocator
  - Associates page frames with security domain
  - Adds „kernel“ zone to buddy allocator

Evaluation

System Setup

S1

i7 – Ivy Bridge
8GB DDR3

S2

i5 – Sandy Bridge
8GB DDR3

S3

i5 – Sandy Bridge (Mobile)
8GB DDR3

Security

- Tested blacklisting against previously compiled list of target rows
  - Vulnerable rows are successfully blocked by the bootloader
- Tested CATT against existing Rowhammer kernel exploits [BH15 Seaborn and Dullien]
  - Without our patch: success within minutes
  - With our patch: ran 48+ hours without success

Performance

- SPEC CPU 2006: avg. -0.5% (max 0.29%)
- Phoronix: avg. 0.27% (max. 2.49%)
- LMBench: avg. 0.11% (max. 1.66%)
- Linux Test Project: same results as vanilla kernel (contains stress tests for scheduling, memory, and file accesses)

Conclusion

- Software vulnerabilities are still the predominant attack vector
  - Continuous arms race between attacks and defenses
- Hardware reliability issues lead to severe security consequences
  - Rowhammer corrupts memory without requiring software vulnerabilities
- Good news: Promising research results and insights
  - First software-only defenses against Rowhammer have been proposed to protect legacy systems

Questions?