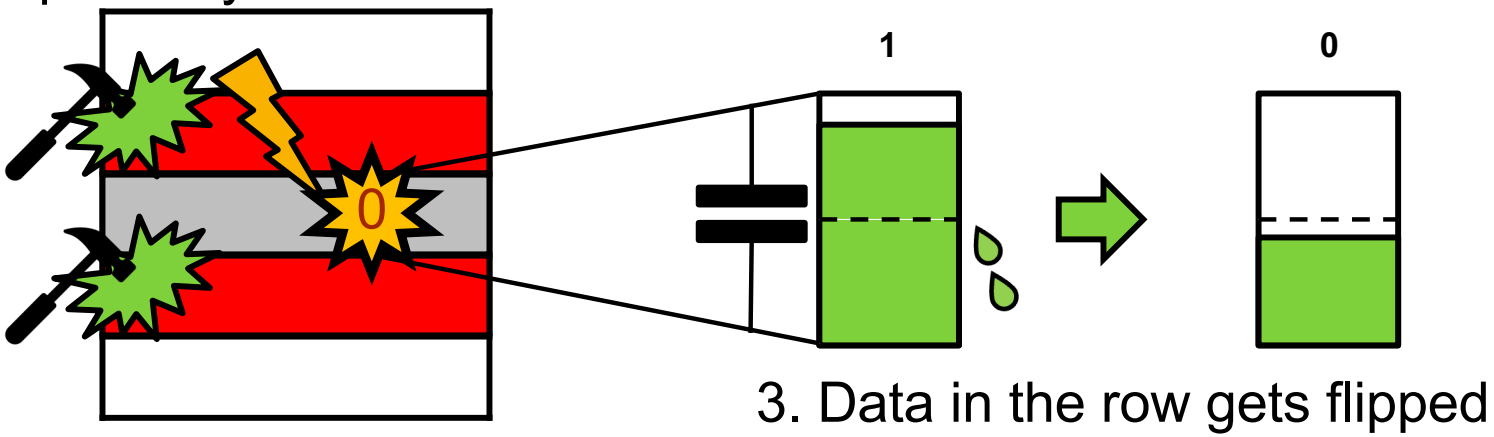
Sledgehammer: Amplifying Rowhammer via Bank-level Parallelism

Ingab Kang, Walter Wang, Jason Kim, Stephan van Schaik, Youssef Tobah, Andrew Kwong, Daniel Genkin, Yuval Yarom

Georgia Institute of Technology

Rowhammer

1. Repeatedly activate or ”hammer” rows

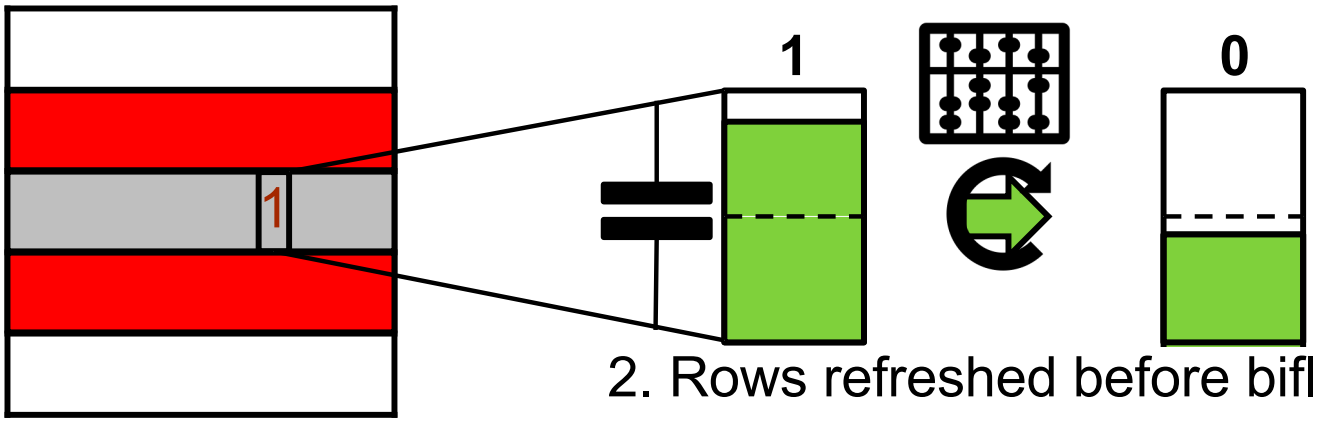


3. Data in the row gets flipped

2. Neighboring row leaks charge

Mitigations in DDR4

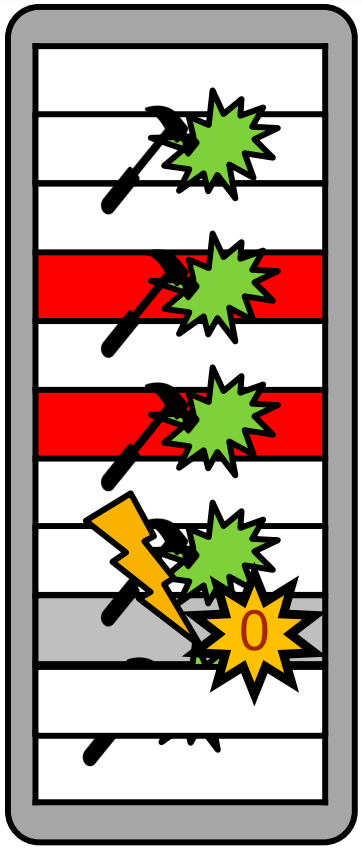
1. Tracks frequently activated rows



2. Rows refreshed before biflip

3. Data integrity preserved

DDR4 Rowhammer



1. Hammering two rows is always tracked
2. Hammer N-rows at the same time
3. Some rows slip by and flips bits! 😈

TRRespass: Exploiting the Many Sides of Target Row Refresh

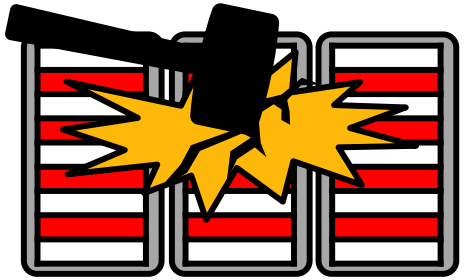
BLACKSMITH: Scalable Rowhammering in the Frequency Domain

SMASH: Synchronized Many-sided Rowhammer Attacks from JavaScript

What's Next?

Preview

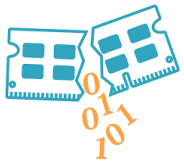
Introduce ”Multibank Hammering”



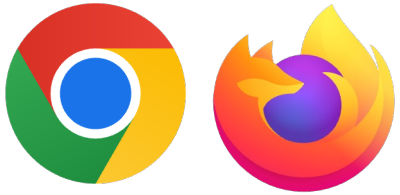
Hammer DDR4 on Intel 12th gen

Optimized attacks in Native

sudo 

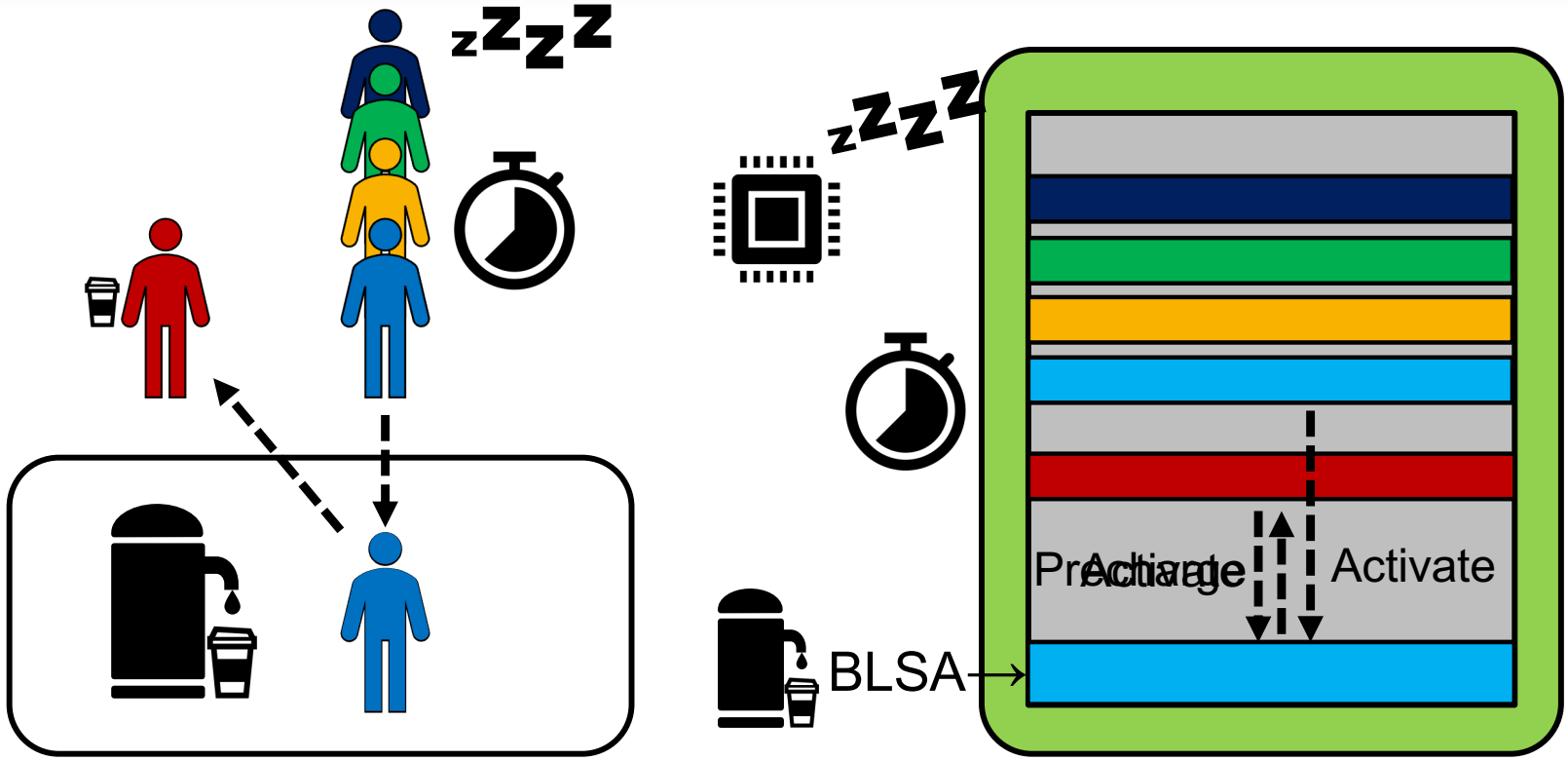
Rambleed 

Rowhammer in Default Browser Settings



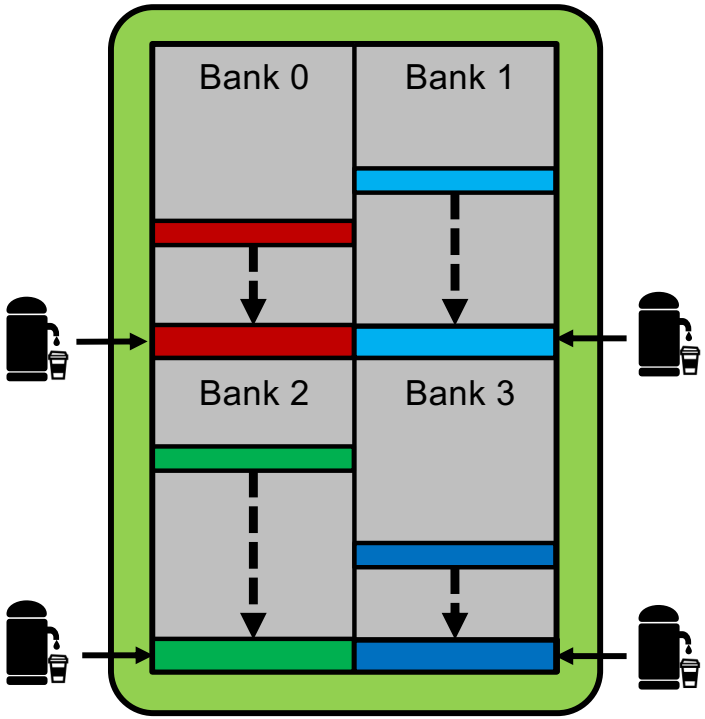
DRAM Operation Background

Coffee Dispenser and Bit Line Sense Amplifiers



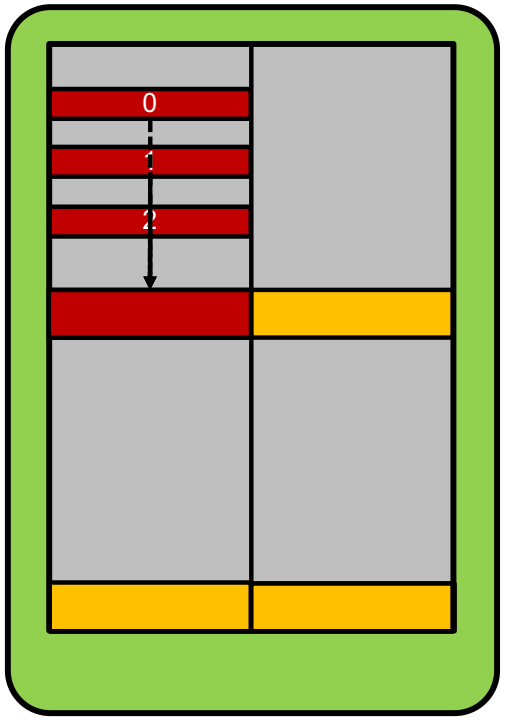
How do we fix queueing delay?

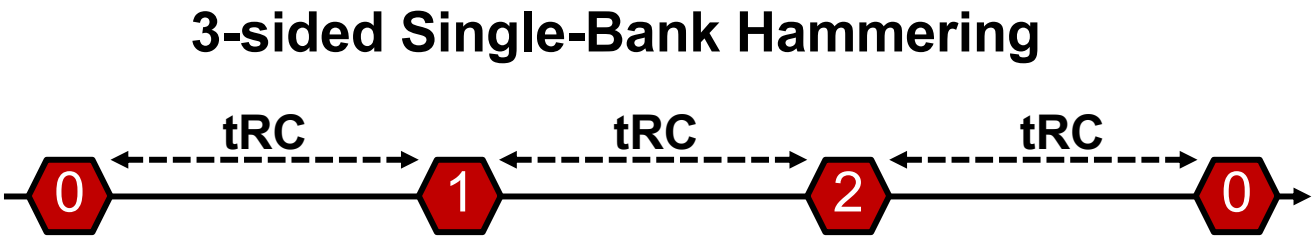
DRAM Bank



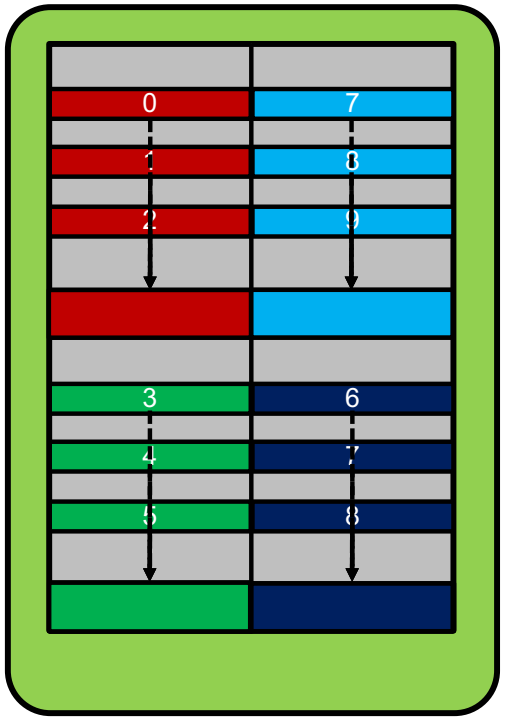
1. Parallelize access by having multiple BLSAs
2. Rows that don't share a BLSA operate independent of each other.
3. Bank = Rows + BLSA

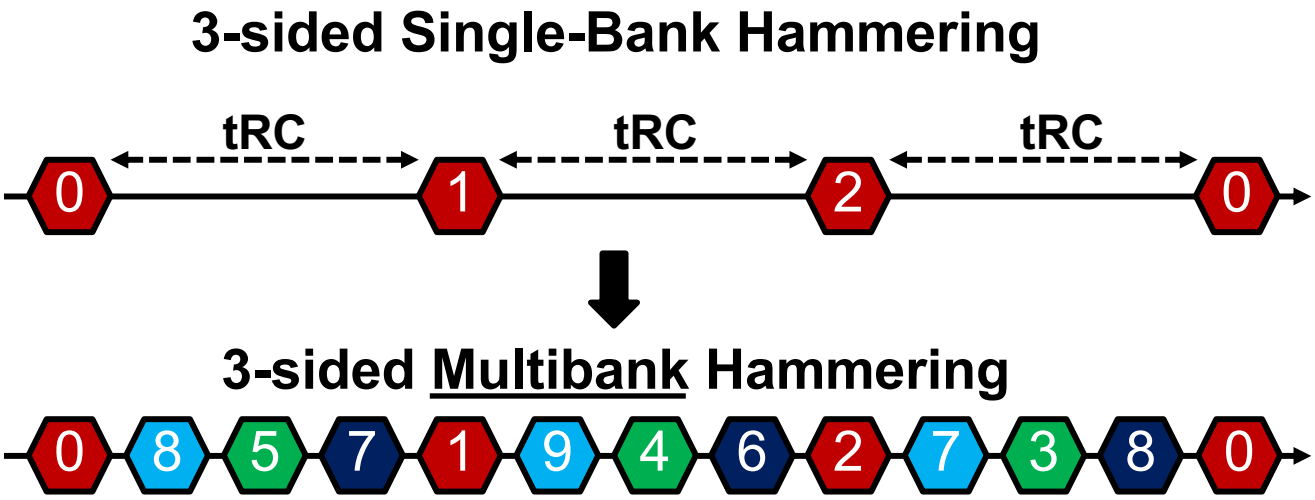
N-sided Hammering



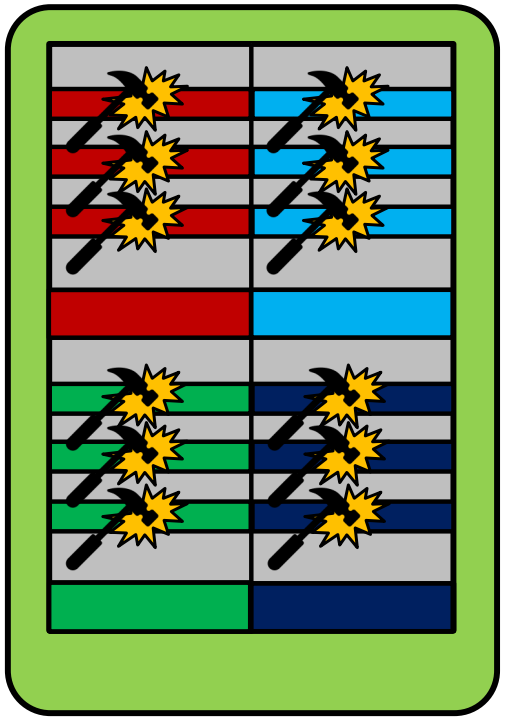


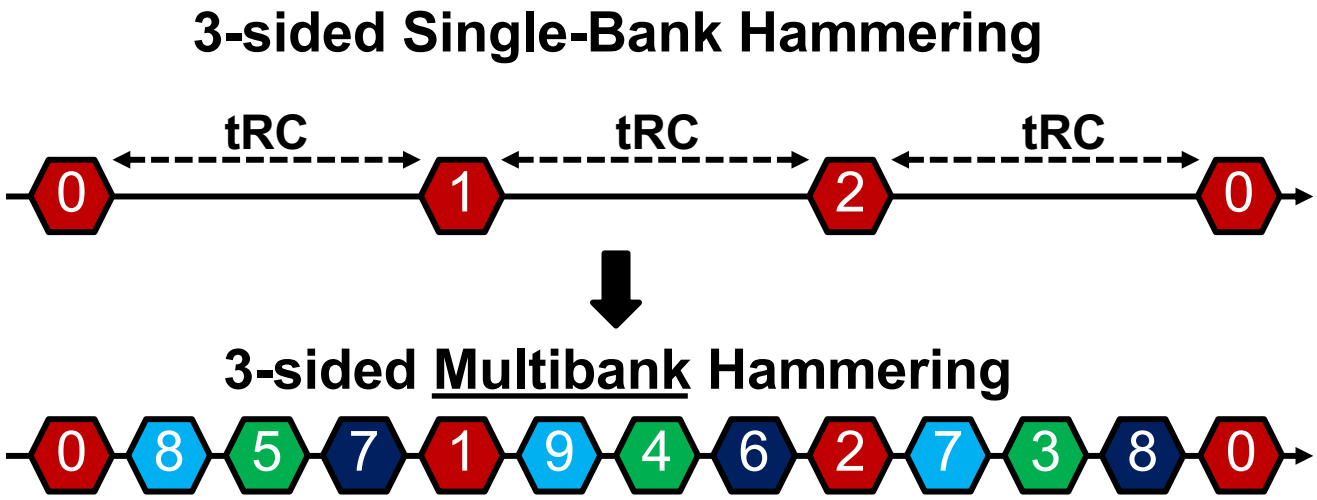
Multibank Hammering



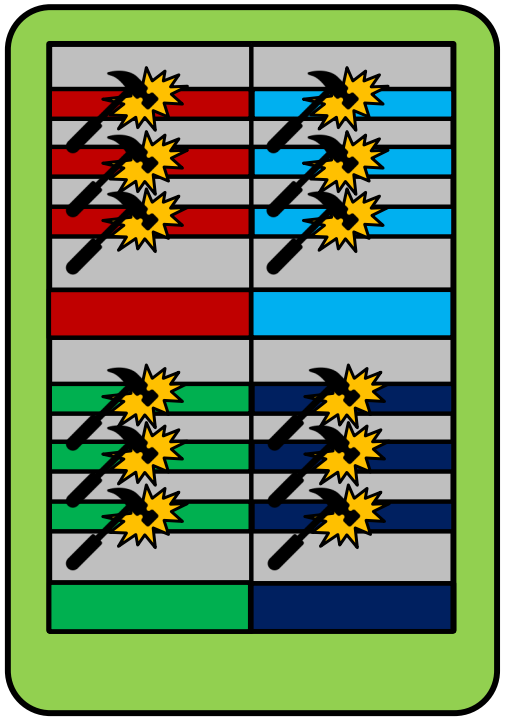


Multibank Hammering





Multibank Hammering



3-sided <u>Multibank</u> Hammering

8x more flips per iteration with 4-bank, i7-7700

19 flips/1000 iters, 2 bank, 12th gen

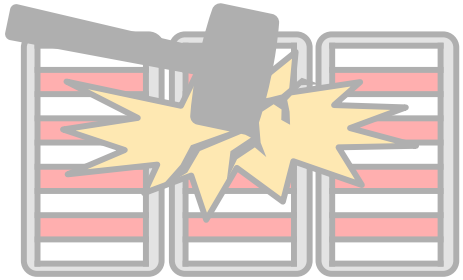
sudo 

390 s average, 879× speedup

1.56 bits/s, first on DDR4 memory

Index

Introduce ”Multibank Hammering”



Hammer DDR4 on Intel 12th gen

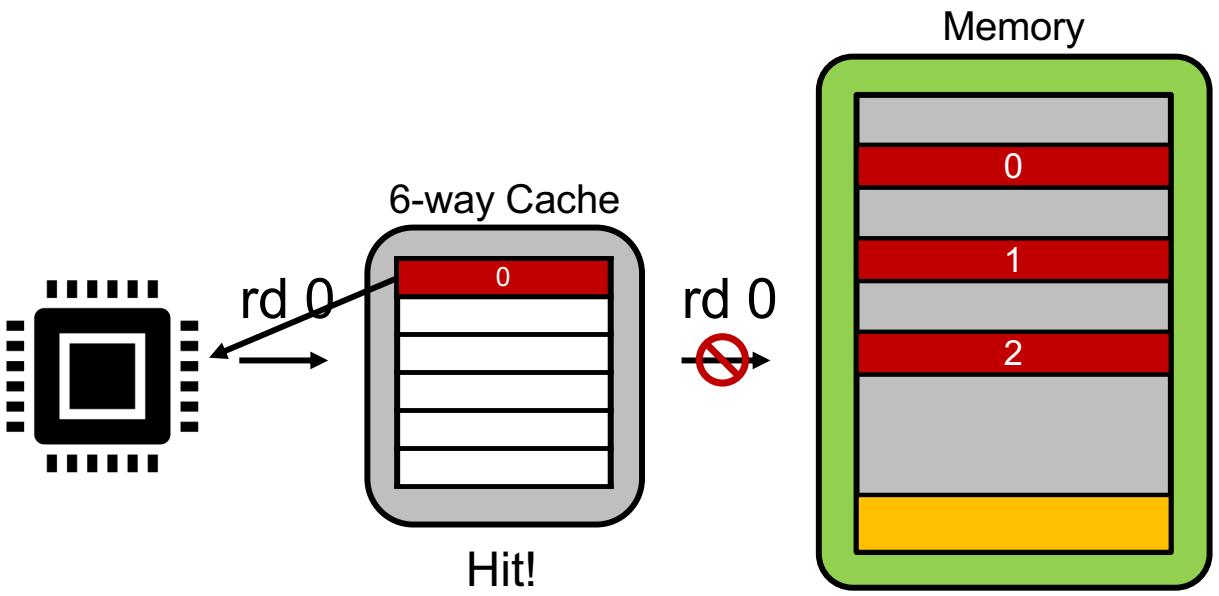
Optimized attacks in Native

sudo 

Rambleed

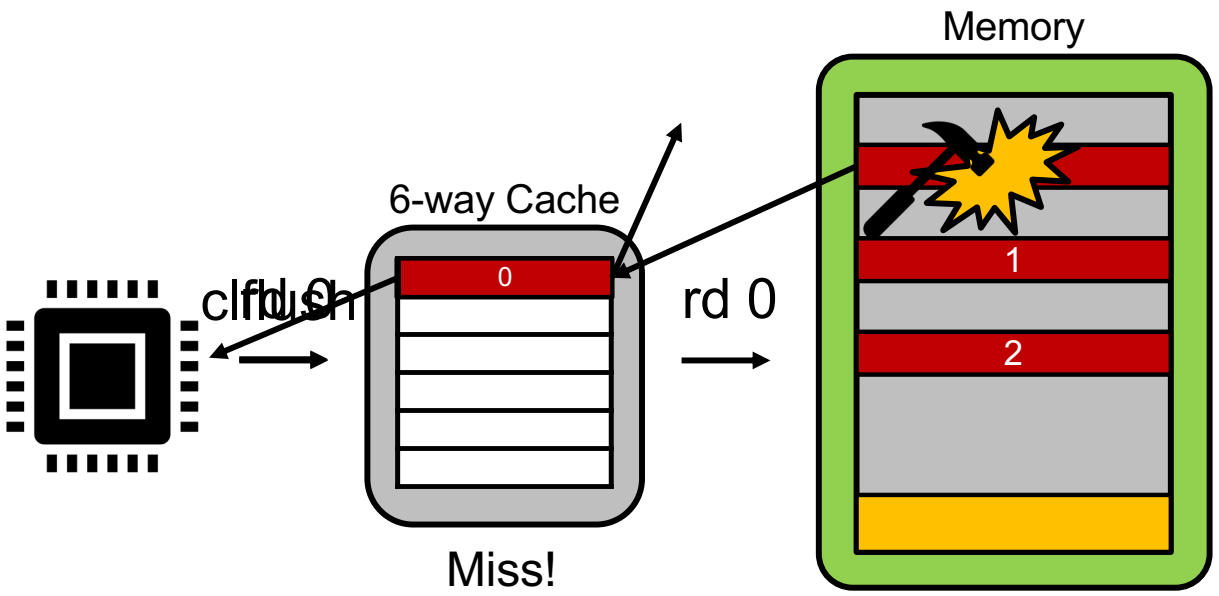
Rowhammer in Browser

Flushing the cache



Need to evict rows from the cache!

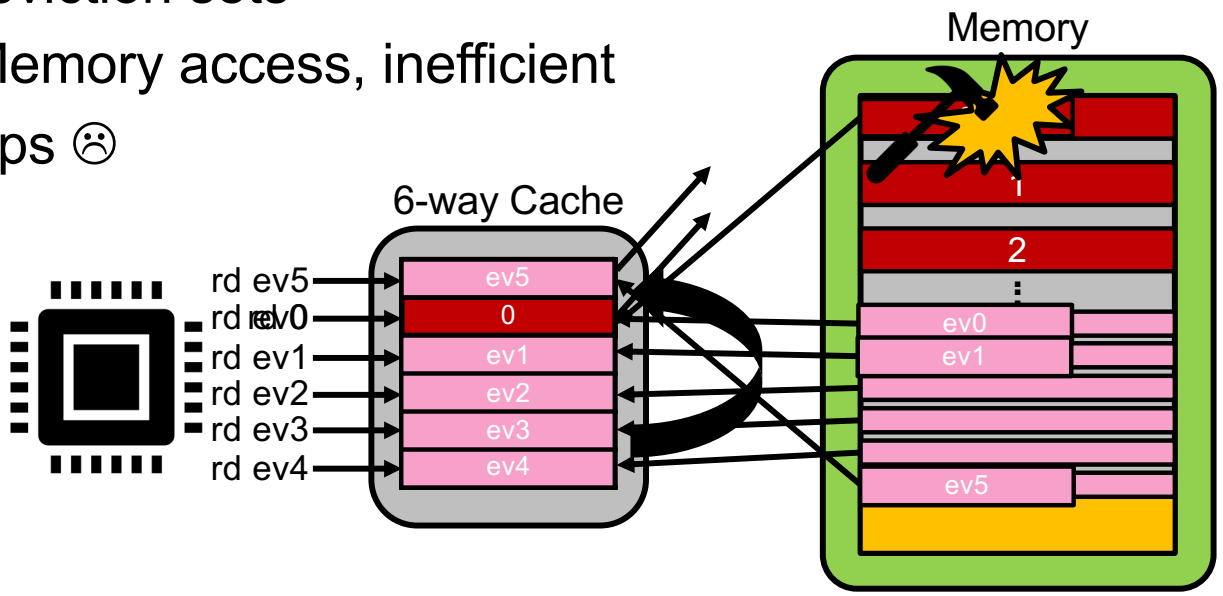
Flushing the cache



No flushing instructions in browsers!

Naïve Approach

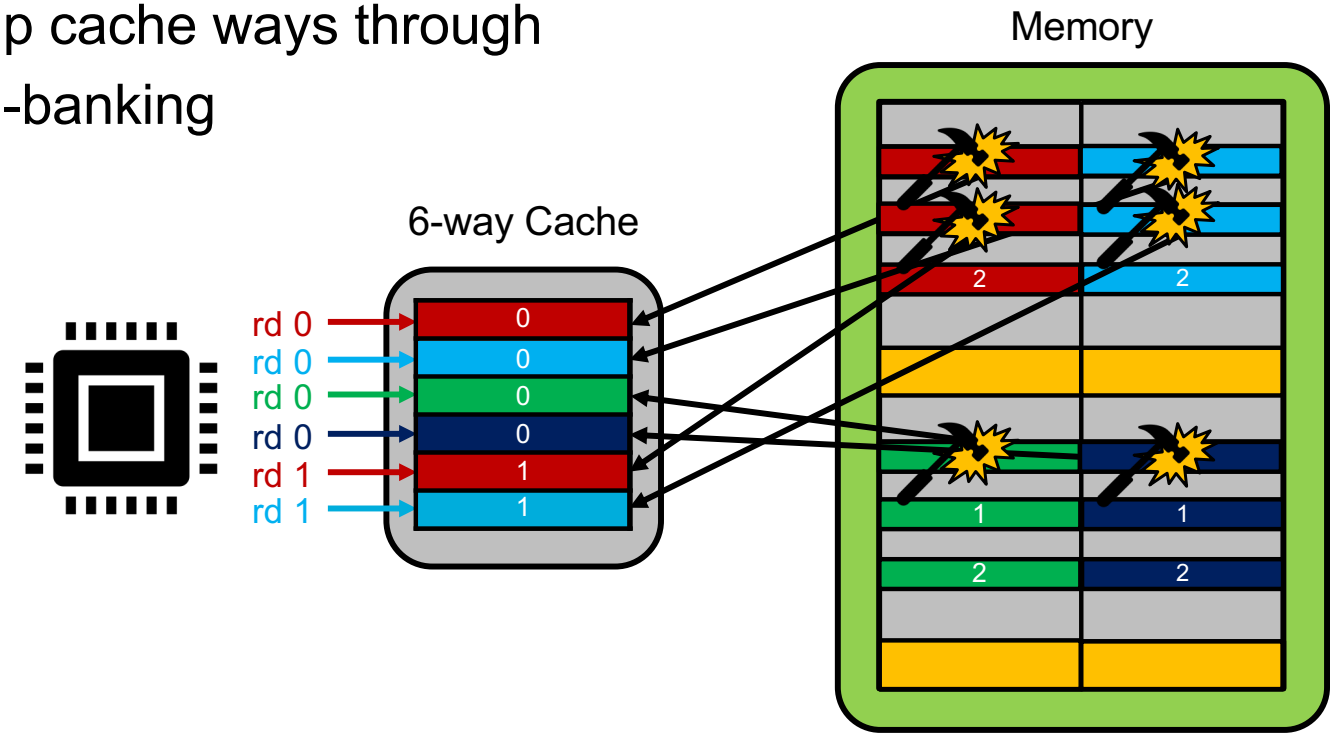
- Use eviction sets
- 1/7 Memory access, inefficient
- No flips ☹



- SMASH: extra cache accesses & careful synchronization
- < 3 bit flip / hr

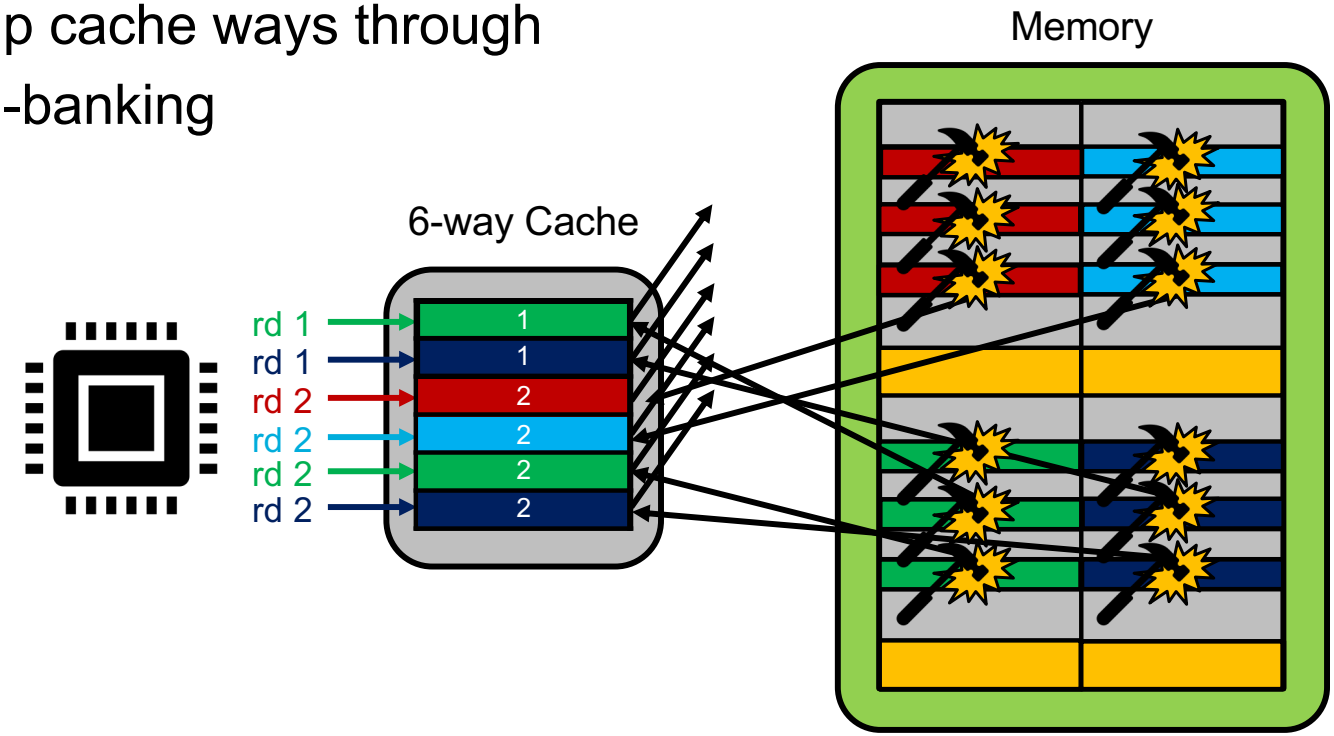
Leveraging Multibank

- Fill up cache ways through multi-banking



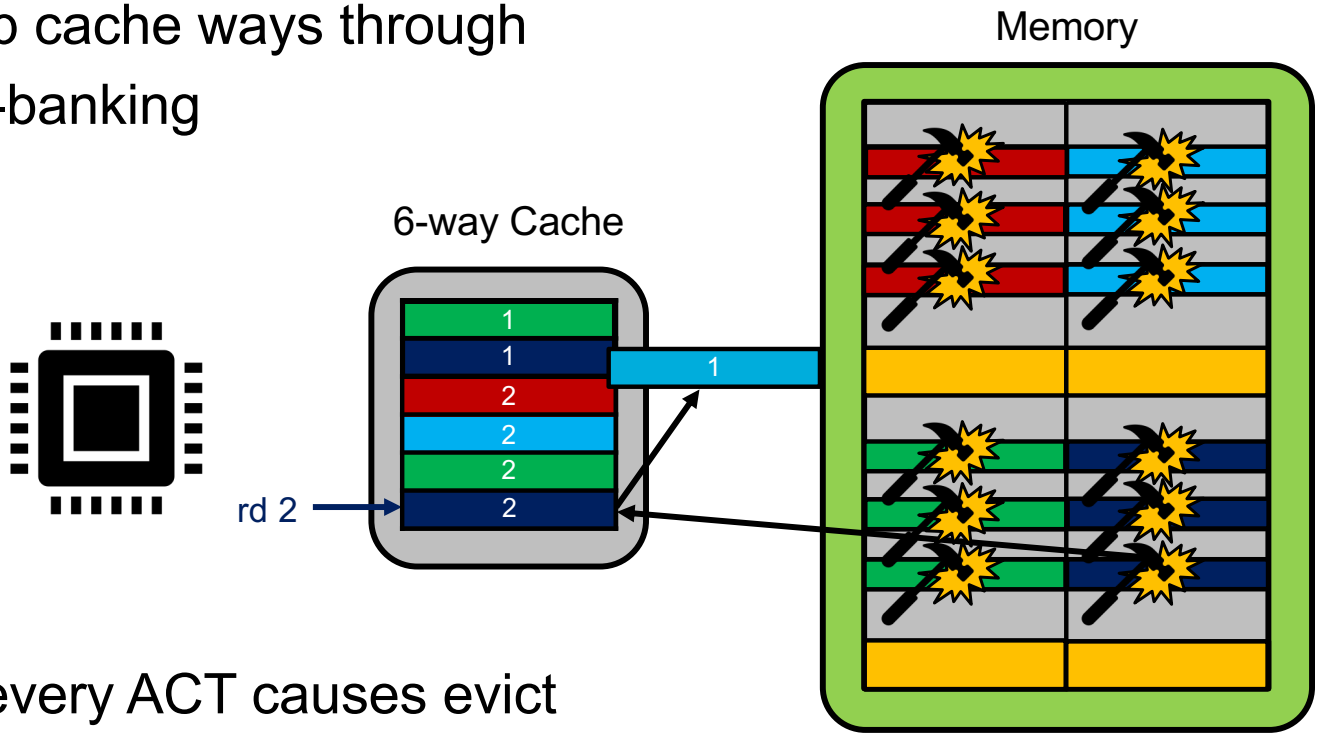
Leveraging Multibank

- Fill up cache ways through multi-banking



Leveraging Multibank

- Fill up cache ways through multi-banking



- 1/1, every ACT causes evict
- Efficiently hammer all rows in all banks

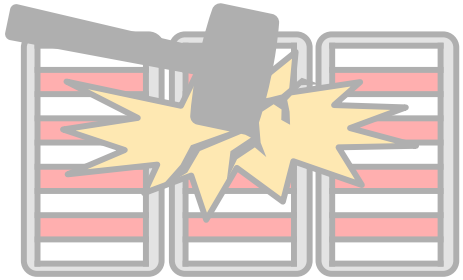
Results

- Test on Chrome and Firefox
- Chrome: 169 flips/hr 
- Firefox: 107 flips/hr 
- End-to-End write primitive on Firefox
  - 2 MB contiguous memory detection sidechannel
  - Removes need for Transparent Huge Pages

First RH attack in browser with default configuration!

Recap

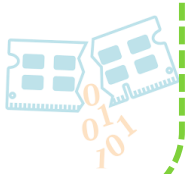
Introduce ”Multibank Hammering”



Hammer DDR4 on Intel 12th gen

Optimized attacks in Native

sudo 

Rambleed 

Rowhammer in Browser

Thank you for listening!

Ingab Kang
igkang@umich.edu
https://architecture.fail