Future of Computer Architecture and Hardware Security

Onur Mutlu

omutlu@gmail.com

https://people.inf.ethz.ch/omutlu

25 October 2024

Hardwear.io MemSec Keynote Talk

SAFARI ETH zürich Carnegie Mellon

Agenda

- Computer Architecture Today
  - What is it and where it is going

- Three Major Hardware Issues That Affect Security
  - Technology scaling problems
  - Growing system complexity; old methods not keeping up
  - New architectures and technologies

Why Do We Do Computing?

Answer

To Solve Problems

Answer Reworded

To Gain Insight

Hamming, "Numerical Methods for Scientists and Engineers," 1962.

Answer Extended

To Enable

a Better Life & Future

How Does a Computer Solve Problems?

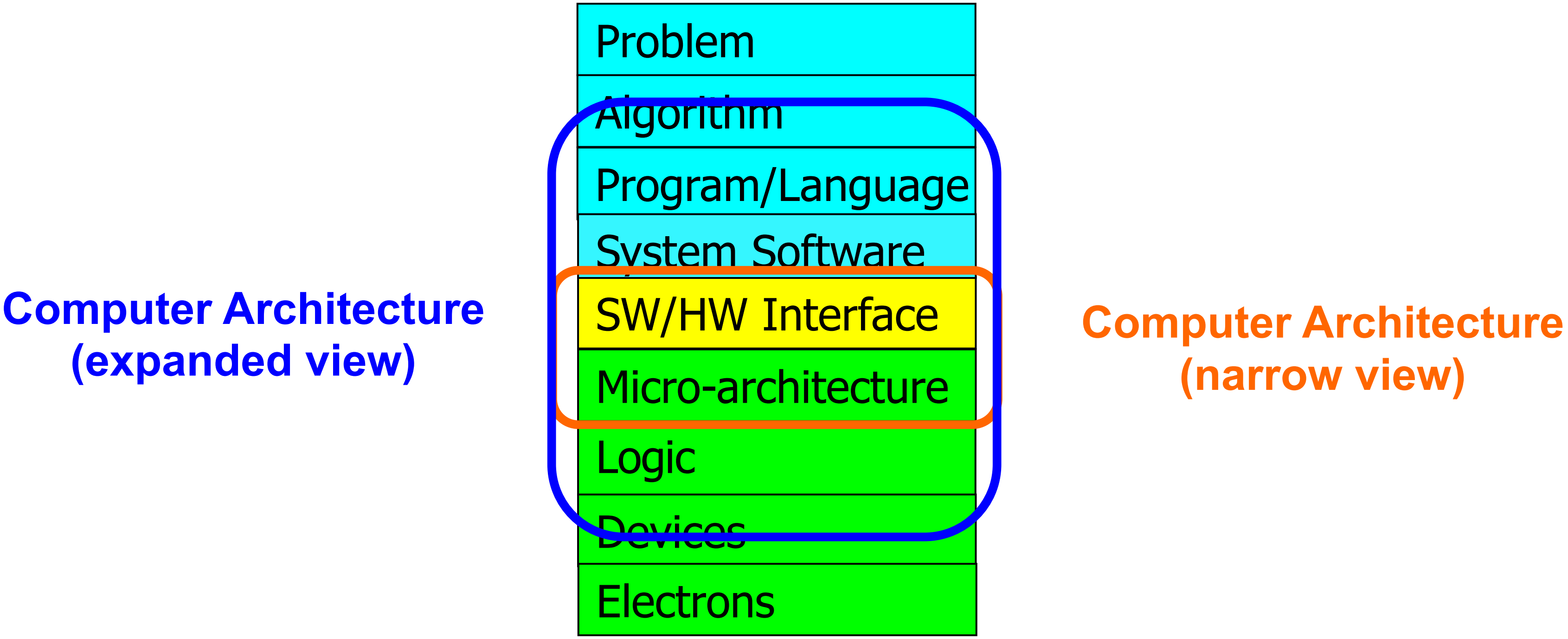
Answer

Orchestrating Electrons

In today's dominant technologies

How Do Problems Get Solved by Electrons?

The Transformation Hierarchy



Computer Architecture

- is the science and art of designing computing platforms (hardware, interface, system SW, and programming model)

- to achieve a set of design goals
  - E.g., highest performance on earth on workloads X, Y, Z
  - E.g., longest battery life at a form factor that fits in your pocket with cost < $$$ CHF
  - E.g., best average performance across all known workloads at the best performance/cost ratio
  - …

  - Designing a supercomputer is different from designing a smartphone → But, many fundamental principles are similar

Different Platforms, Different Goals

Source: http://www.sia-online.org (semiconductor industry association)

Different Platforms, Different Goals

Source: https://iq.intel.com/5-awesome-uses-for-drone-technology/

Different Platforms, Different Goals

Source: https://taxistartup.com/wp-content/uploads/2015/03/UK-Self-Driving-Cars.jpg

Different Platforms, Different Goals

Source: http://sm.pcmag.com/pcmag_uk/photo/g/google-self-driving-car-the-guts/google-self-driving-car-the-guts_dwx8.jpg

Different Platforms, Different Goals

Mohammed Alser, Zülal Bingöl, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, Onur Mutlu
"Accelerating Genome Analysis: A Primer on an Ongoing Journey" IEEE Micro, August 2020.

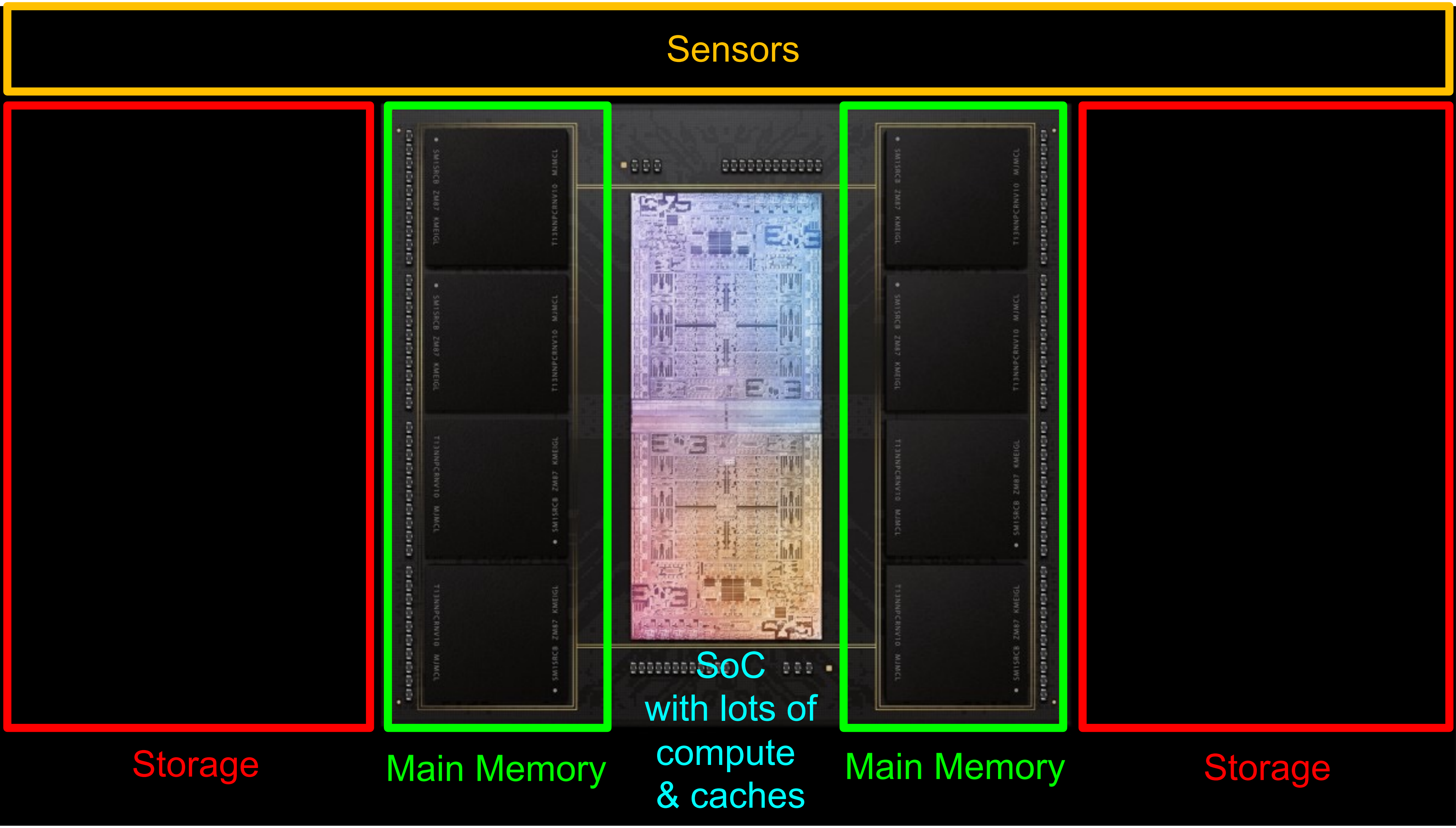
MinION from ONT

Accelerating Genome Analysis: A Primer on an Ongoing Journey
Sept.-Oct. 2020, pp. 65-75, vol. 40
DOI Bookmark: 10.1109/MM.2020.3013728

FPGA-Based Near-Memory Acceleration of Modern Data-Intensive Applications
July-Aug. 2021, pp. 39-48, vol. 41
DOI Bookmark: 10.1109/MM.2021.3088396

SmidgION from ONT

An Example System in Your Pocket



Sensors

Storage

Main Memory

SoC with lots of compute & caches

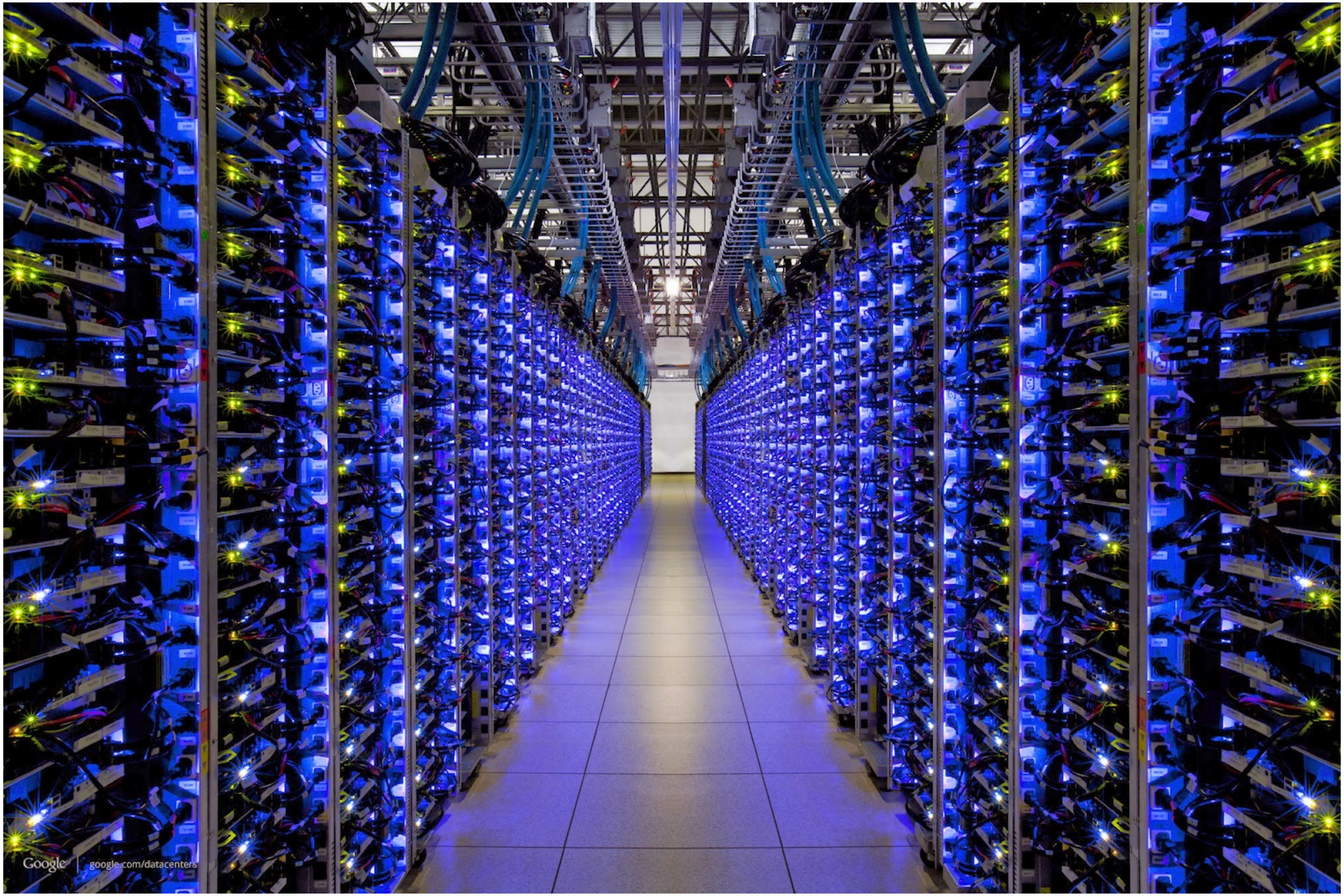
Main Memory

Storage

Apple M1 Ultra System (2022)

https://www.gsmarena.com/apple_announces_m1_ultra_with_20core_cpu_and_64core_gpu-news-53481.php

Different Platforms, Different Goals



Source: http://datacentervoice.com/wp-content/uploads/2015/10/data-center.jpg

Different Platforms, Different Goals

Source: https://fossbytes.com/wp-content/uploads/2015/06/Supercomputer-TIANHE2-china.jpg

Different Platforms, Different Goals

Source: https://www.itmagazine.ch/artikel/72401/Fugaku_Der_schnellste_Supercomputer_der_Welt.html

Different Platforms, Different Goals

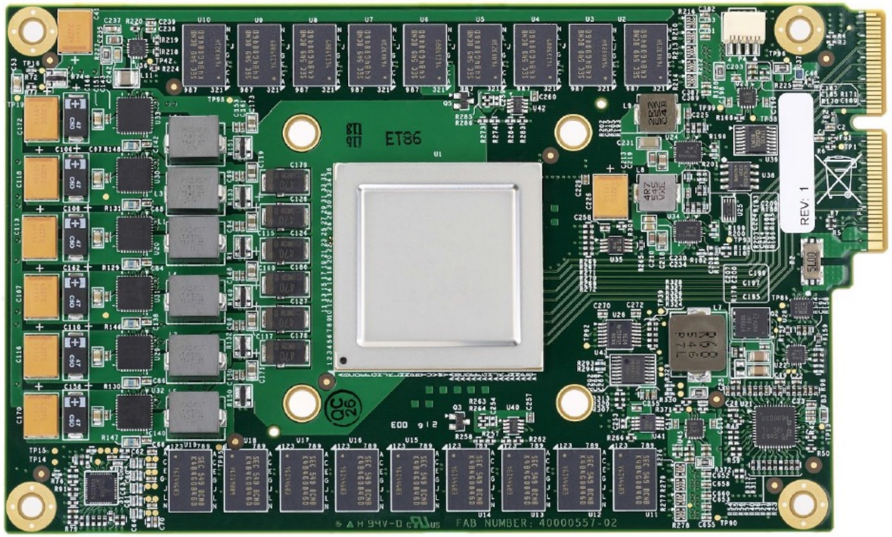


Figure 3. TPU Printed Circuit Board. It can be inserted in the slot for an SATA disk in a server, but the card uses PCIe Gen3 x16.

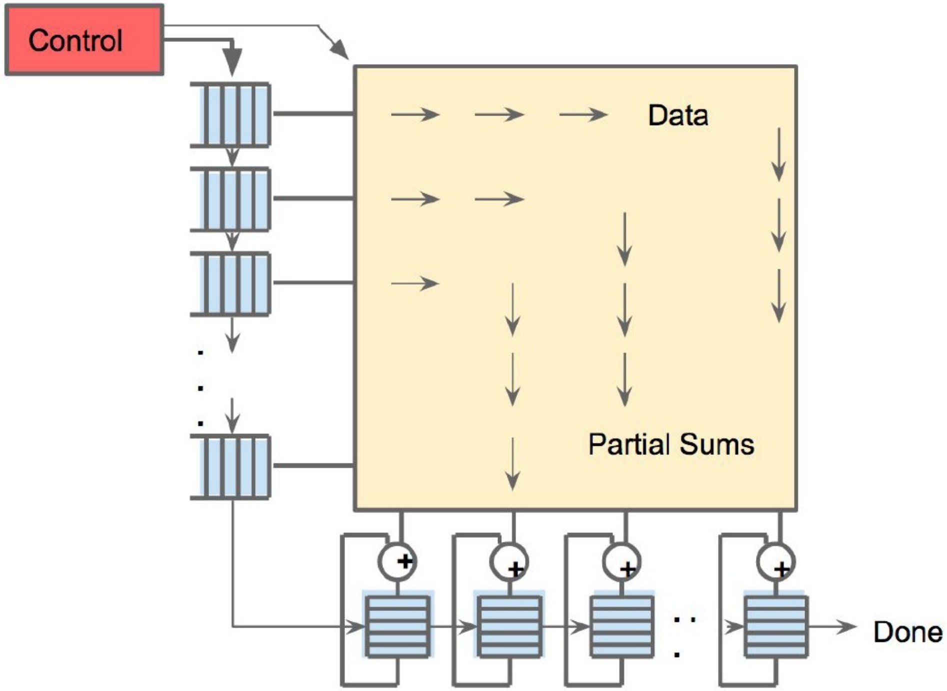
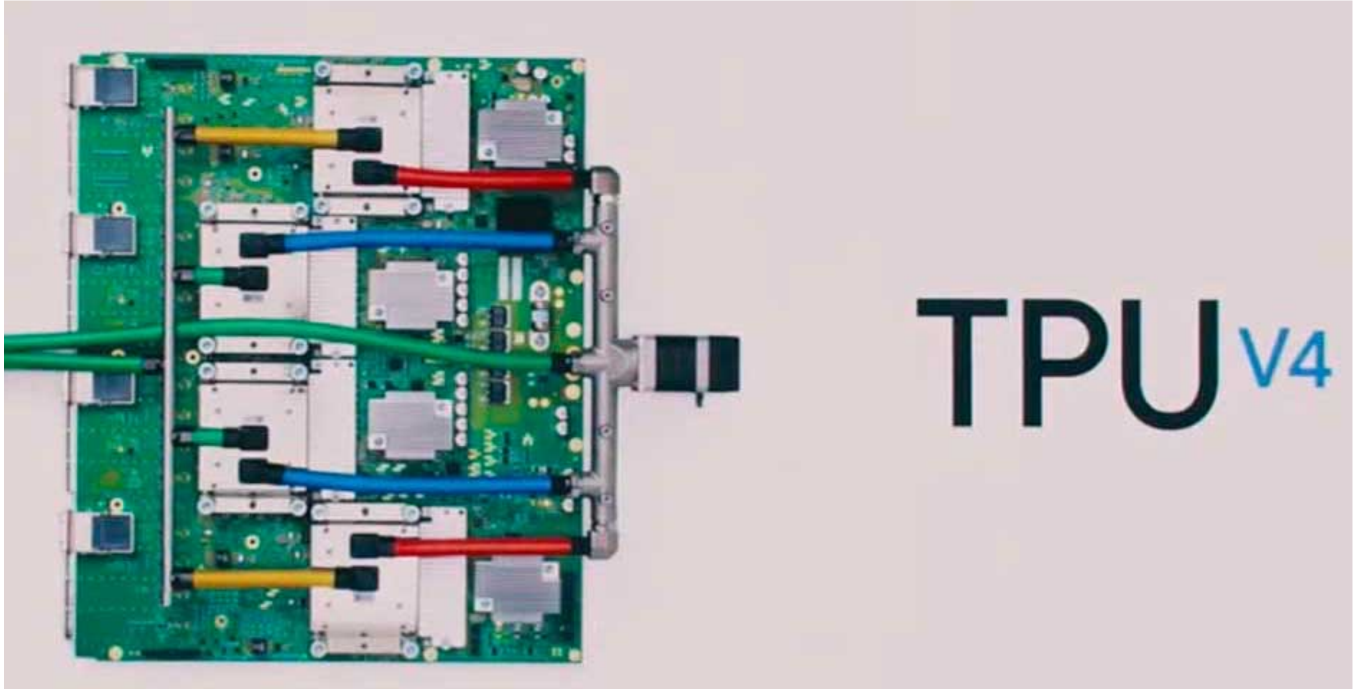


Figure 4. Systolic data flow of the Matrix Multiply Unit. Software has the illusion that each 256B input is read at once, and they instantly update one location of each of 256 accumulator RAMs.

Jouppi et al., "In-Datacenter Performance Analysis of a Tensor Processing Unit", ISCA 2017.

Different Platforms, Different Goals



New ML applications (vs. TPU3):

- Computer vision
- Natural Language Processing (NLP)
- Recommender system
- Reinforcement learning that plays Go

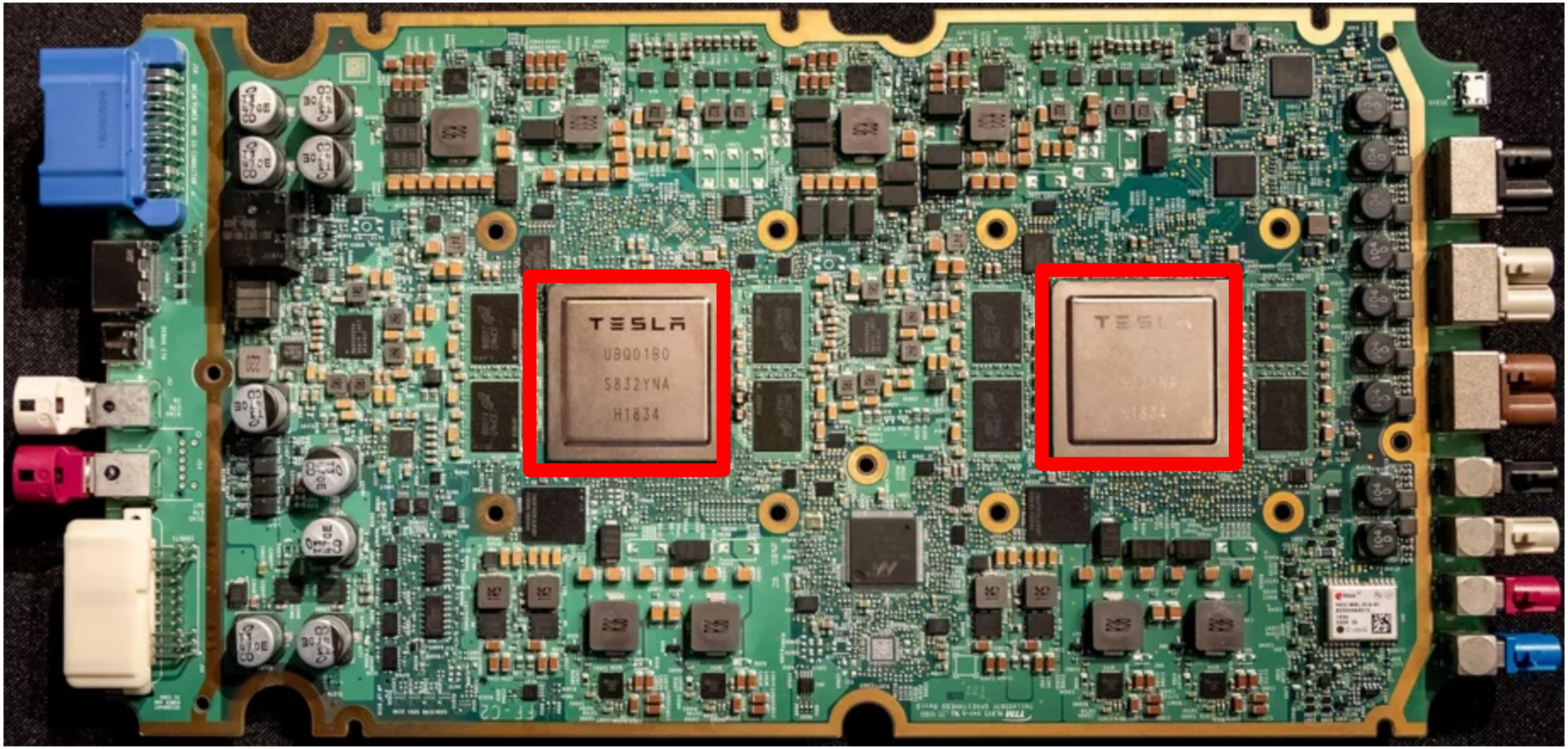
250 TFLOPS per chip in 2021
vs 90 TFLOPS in TPU3

1 ExaFLOPS per board

https://spectrum.ieee.org/tech-talk/computing/hardware/heres-how-googles-tpu-v4-ai-chip-stacked-up-in-training-tests

Different Platforms, Different Goals

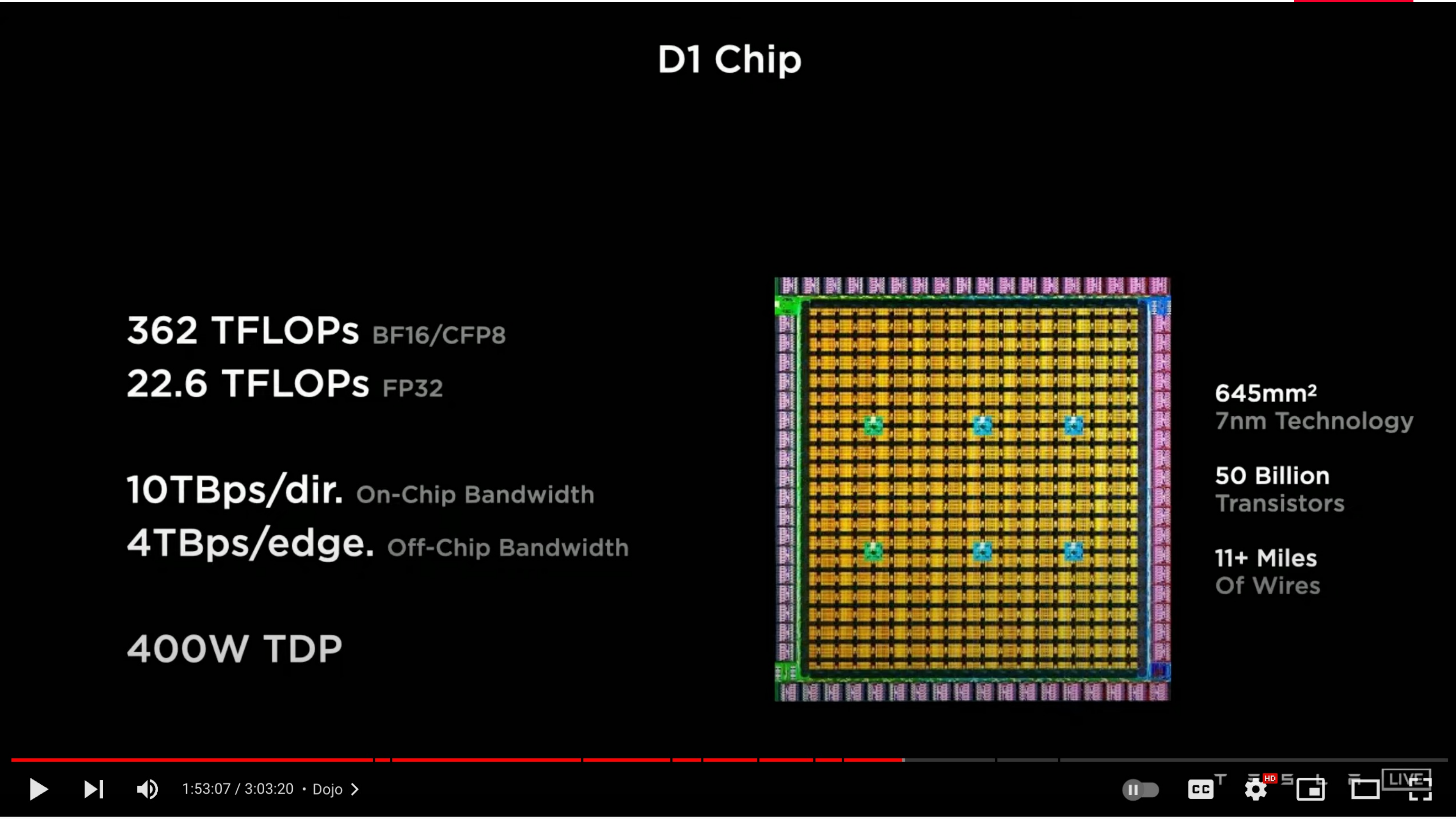
- ML accelerator: 260 mm^2, 6 billion transistors, 600 GFLOPS GPU, 12 ARM 2.2 GHz CPUs.
- Two redundant chips for better safety.



https://youtu.be/Ucp0TTmvqOE?t=4236

Different Platforms, Different Goals

- Tesla Dojo Chip & System



D1 Chip

362 TFLOPs BF16/CFP8
22.6 TFLOPs FP32

10TBps/dir. On-Chip Bandwidth
4TBps/edge. Off-Chip Bandwidth

400W TDP

645mm² 7nm Technology

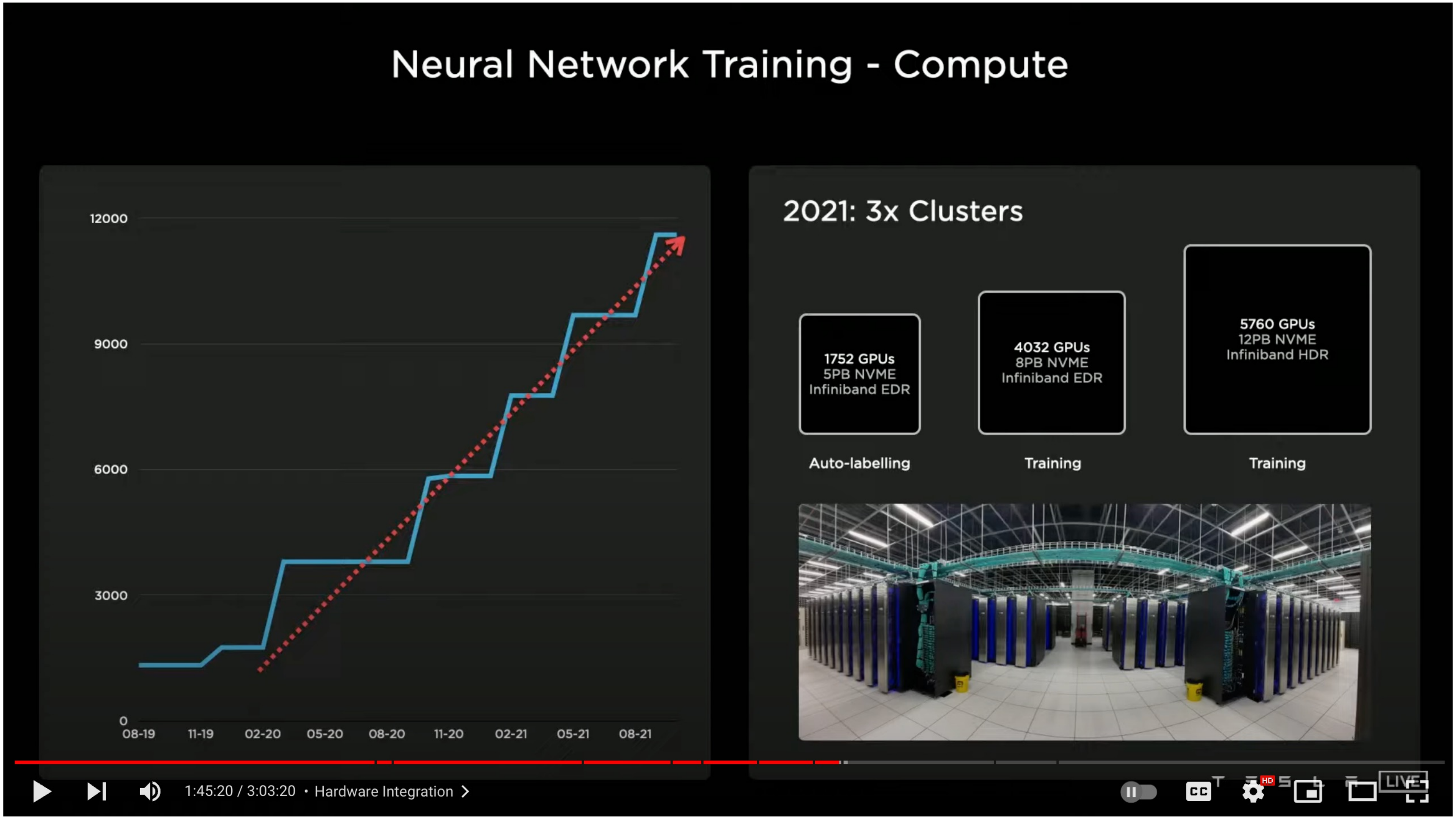
50 Billion Transistors

11+ Miles Of Wires

https://www.youtube.com/watch?v=j0z4FweCy4M&t=6340s

Different Platforms, Different Goals

- Tesla Dojo Chip & System



Neural Network Training - Compute

2021: 3x Clusters

1752 GPUs
5PB NVME
Infiniband EDR

Auto-labelling

4032 GPUs
8PB NVME
Infiniband EDR

Training

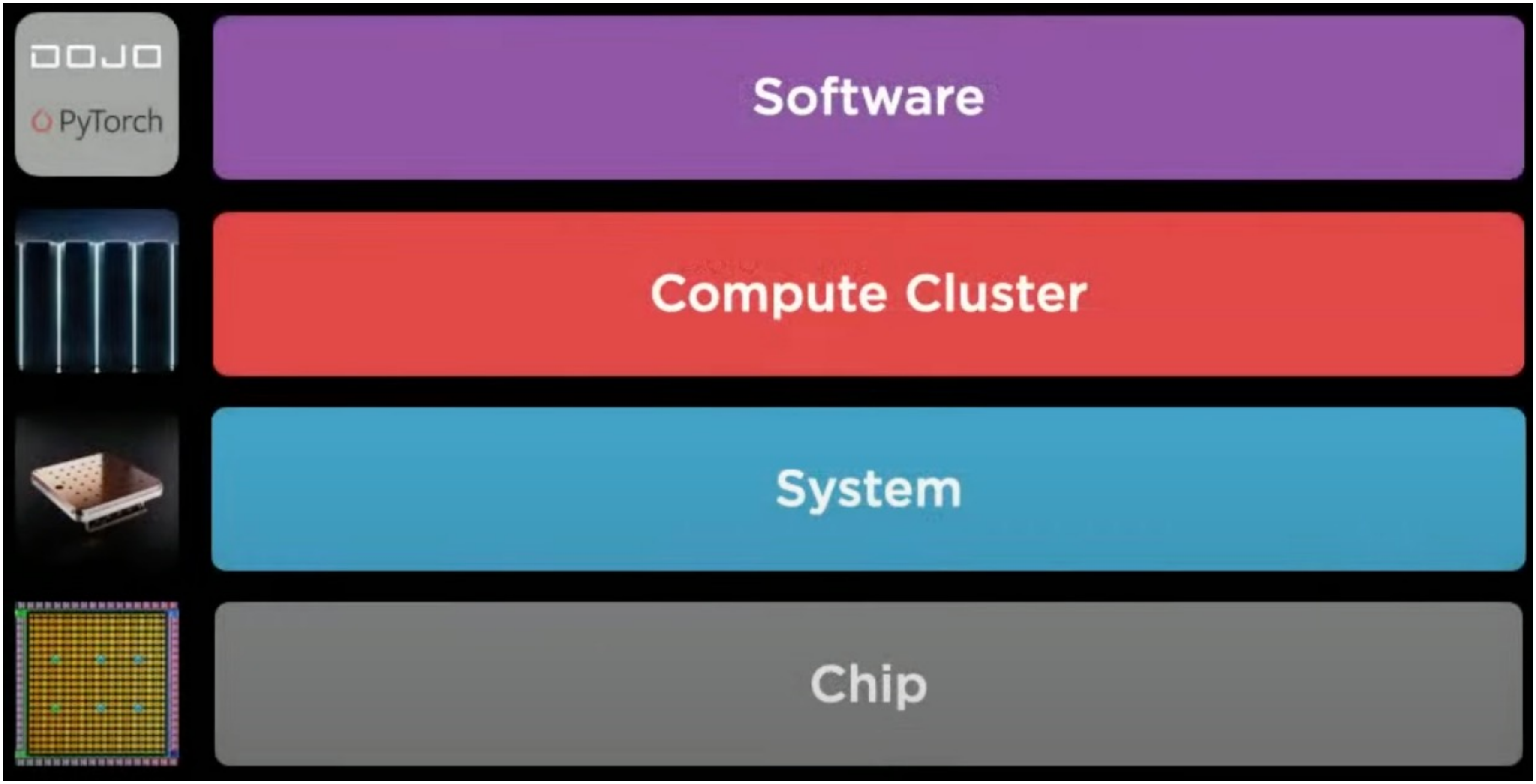
5760 GPUs
12PB NVME
Infiniband HDR

Training

https://www.youtube.com/watch?v=j0z4FweCy4M&t=6340s

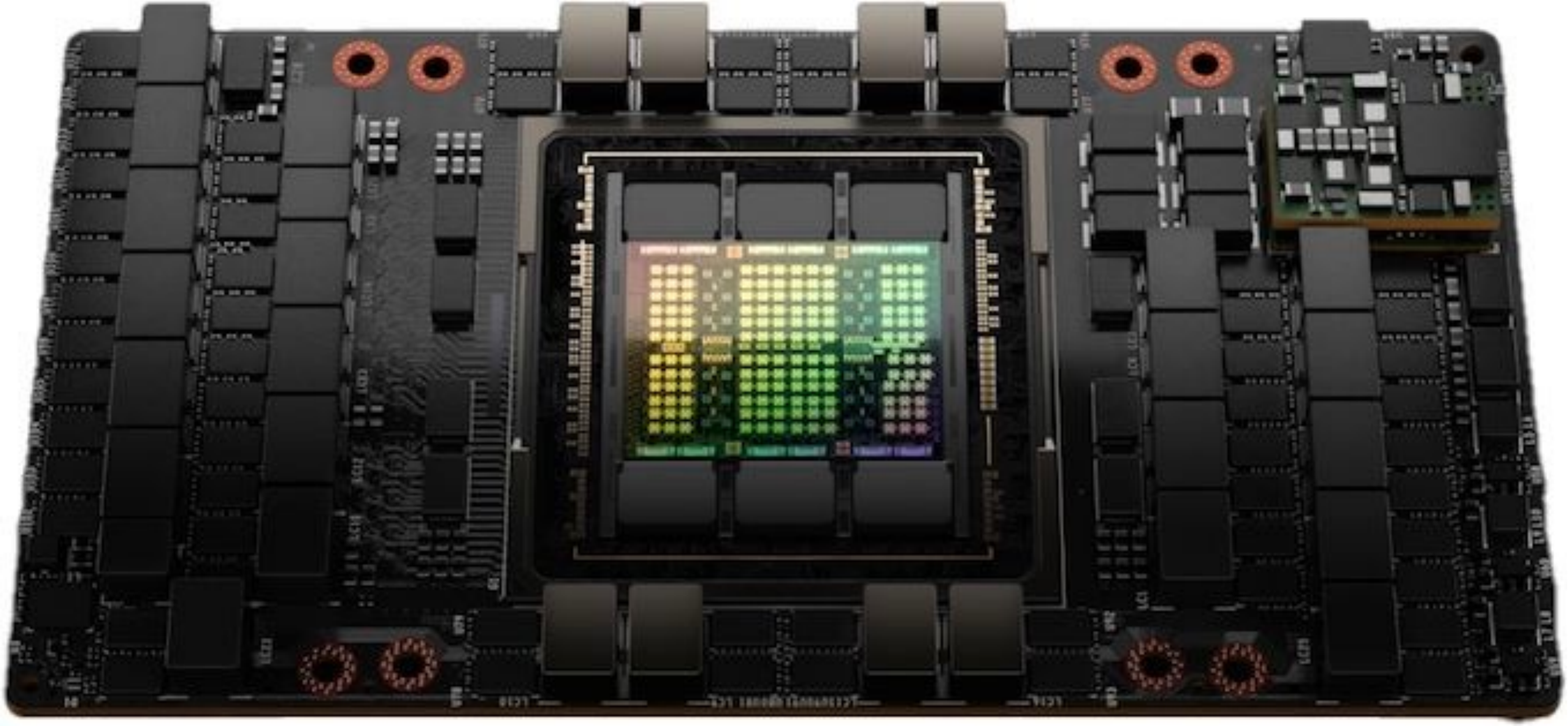
Different Platforms, Different Goals

- Tesla Dojo Chip & System

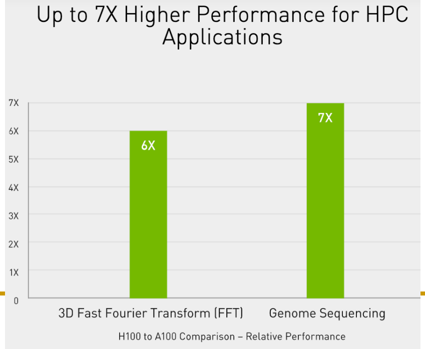


https://www.youtube.com/watch?v=j0z4FweCy4M&t=6340s

Different Platforms, Different Goals

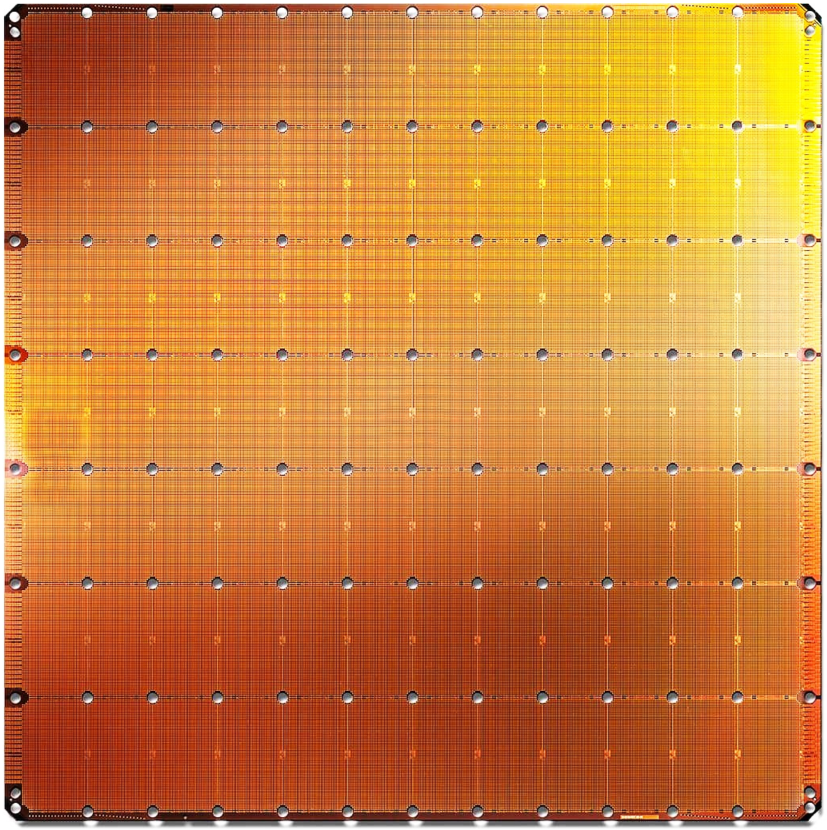


NVIDIA is claiming a **7x improvement** in dynamic programming algorithm (**DPX instructions**) performance on a single H100 versus naïve execution on an A100.



https://www.nvidia.com/en-us/data-center/h100/

Cerebras's Wafer Scale Engine (2019)



- The largest ML accelerator chip
- 400,000 cores

Cerebras WSE
1.2 Trillion transistors
46,225 mm^2

Largest GPU
21.1 Billion transistors
815 mm^2

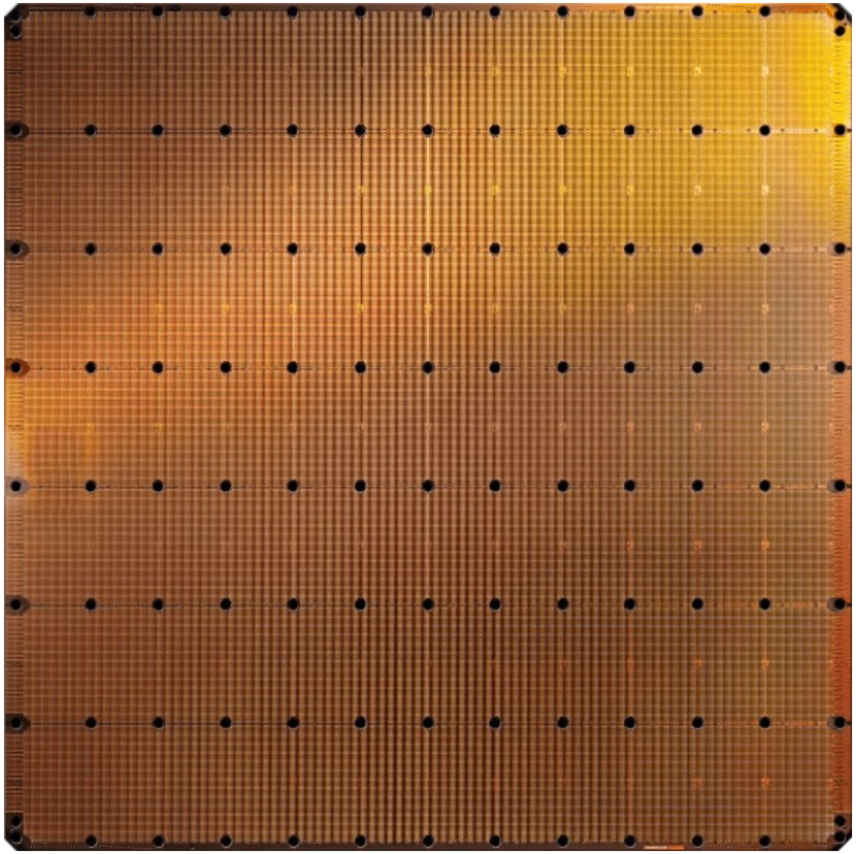
NVIDIA TITAN V

https://www.anandtech.com/show/14758/hot-chips-31-live-blogs-cerebras-wafer-scale-deep-learning

https://www.cerebras.net/cerebras-wafer-scale-engine-why-we-need-big-chips-for-deep-learning/

Cerebras's Wafer Scale Engine-2 (2021)

- The largest ML accelerator chip (2021)
- 850,000 cores



Cerebras WSE-2
2.6 Trillion transistors
46,225 mm²

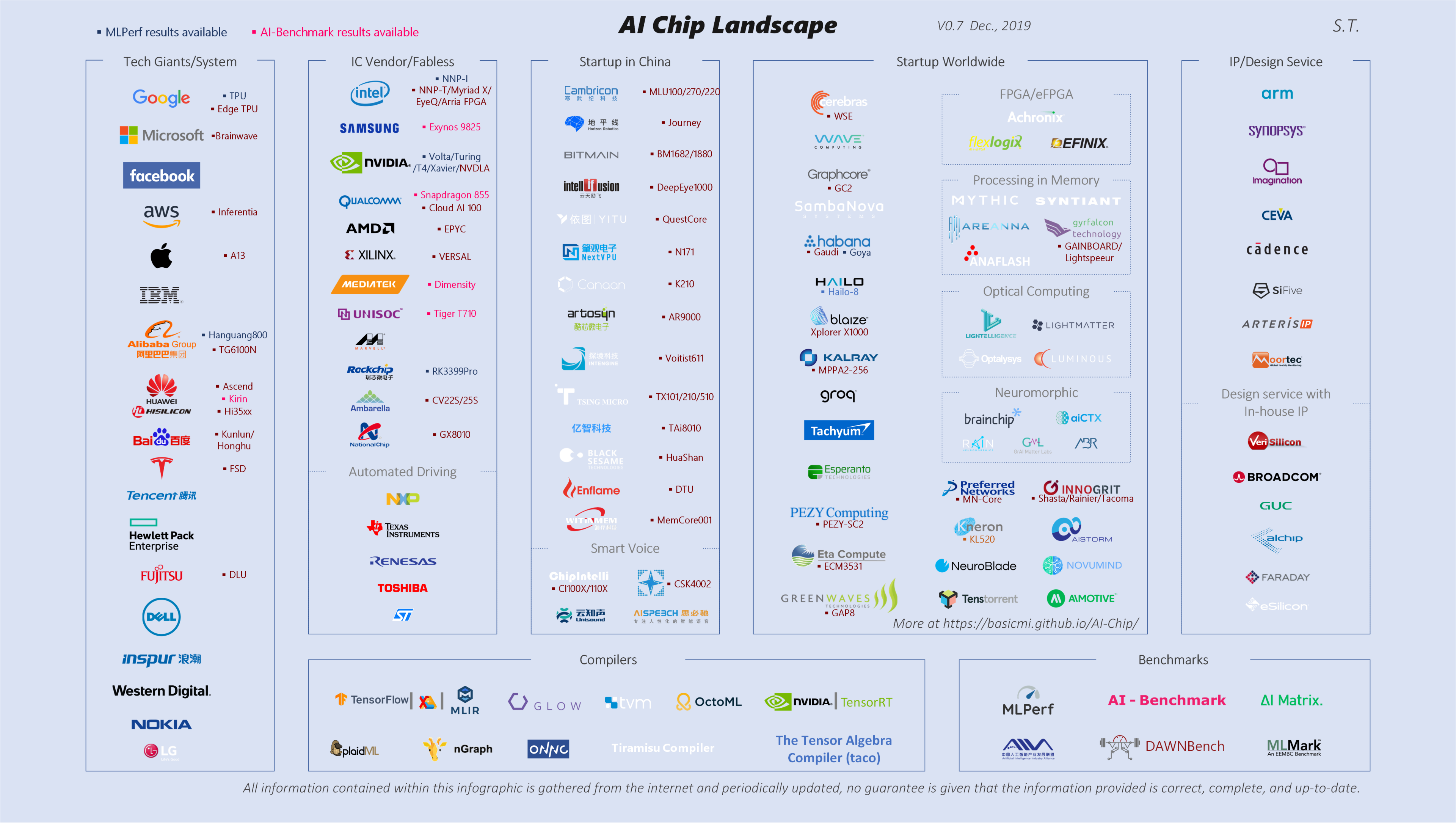
Largest GPU
54.2 Billion transistors
826 mm²

NVIDIA Ampere GA100

https://www.anandtech.com/show/14758/hot-chips-31-live-blogs-cerebras-wafer-scale-deep-learning

https://www.cerebras.net/cerebras-wafer-scale-engine-why-we-need-big-chips-for-deep-learning/

Many (Other) AI/ML Chips (2021)



▪ MLPerf results available ▪ AI-Benchmark results available

AI Chip Landscape

V0.7 Dec., 2019

S.T.

Tech Giants/System

- Google ▪ TPU ▪ Edge TPU
- Microsoft ▪Brainwave
- facebook
- aws ▪ Inferentia
- ▪ A13
- IBM
- Alibaba Group 阿里巴巴集团 ▪ Hanguang800 ▪ TG6100N
- HUAWEI HISILICON ▪ Ascend ▪ Kirin ▪ Hi35xx
- Bai du 百度 ▪ Kunlun/Honghu
- ▪ FSD
- Tencent 腾讯
- Hewlett Packard Enterprise
- FUJITSU ▪ DLU
- DELL
- inspur 浪潮
- Western Digital
- NOKIA
- LG

IC Vendor/Fabless

- intel ▪ NNP-I ▪ NNP-T/Myriad X/EyeQ/Arria FPGA
- SAMSUNG ▪ Exynos 9825
- NVIDIA ▪ Volta/Turing /T4/Xavier/NVDLA
- QUALCOMM ▪ Snapdragon 855 ▪ Cloud AI 100
- AMD ▪ EPYC
- XILINX ▪ VERSAL
- MEDIATEK ▪ Dimensity
- UNISOC ▪ Tiger T710
- MARVELL
- Rockchip 瑞芯微电子 ▪ RK3399Pro
- Ambarella ▪ CV22S/25S
- NationalChip ▪ GX8010

Automated Driving

- NXP
- Texas Instruments
- RENESAS
- TOSHIBA
- ST

Startup in China

- Cambricon 寒武纪科技 ▪ MLU100/270/220
- 地平线 Horizon Robotics ▪ Journey
- BITMAIN ▪ BM1682/1880
- intellifusion 云天励飞 ▪ DeepEye1000
- 依图 YITU ▪ QuestCore
- 肇观电子 NextVPU ▪ N171
- Canaan ▪ K210
- artosyn 酷芯微电子 ▪ AR9000
- 探境科技 INTENGINE ▪ Voitist611
- TSING MICRO ▪ TX101/210/510
- 亿智科技 ▪ TAi8010
- BLACK SESAME TECHNOLOGIES ▪ HuaShan
- Enflame ▪ DTU
- WITINMEM 知存科技 ▪ MemCore001

Smart Voice

- Chipintelli ▪ CI100X/110X
- ▪ CSK4002
- 云知声 Unisound
- AISPEECH 思必驰

Startup Worldwide

- cerebras ▪ WSE
- WAVE COMPUTING
- Graphcore ▪ GC2
- SambaNova SYSTEMS
- habana ▪ Gaudi ▪ Goya
- HAILO ▪ Hailo-8
- blaize Xplorer X1000
- KALRAY ▪ MPPA2-256
- groq
- Tachyum
- Esperanto TECHNOLOGIES
- PEZY Computing ▪ PEZY-SC2
- Eta Compute ▪ ECM3531
- GREENWAVES TECHNOLOGIES ▪ GAP8

FPGA/eFPGA

- Achronix
- flexlogix
- EFINIX

Processing in Memory

- MYTHIC
- SYNTIANT
- AREANNA
- gyrfalcon technology ▪ GAINBOARD/Lightspeeur
- ANAFLASH

Optical Computing

- LIGHTELLIGENCE
- LIGHTMATTER
- Optalysys
- LUMINOUS

Neuromorphic

- brainchip
- aiCTX
- RAIN NEUROMORPHICS
- GML GrAI Matter Labs
- ABR

- Preferred Networks ▪ MN-Core
- INNOGRIT ▪ Shasta/Rainier/Tacoma
- Kneron ▪ KL520
- AISTORM
- NeuroBlade
- NOVUMIND
- Tenstorrent
- AIMOTIVE

More at https://basicmi.github.io/AI-Chip/

IP/Design Sevice

- arm
- SYNOPSYS
- imagination
- CEVA
- cadence
- SiFive
- ARTERIS IP
- moortec

Design service with In-house IP

- VeriSilicon
- BROADCOM
- GUC
- alchip
- FARADAY
- eSilicon

Compilers

- TensorFlow
- MLIR
- GLOW
- tvm
- OctoML
- NVIDIA TensorRT
- plaidML
- nGraph
- ONNC
- Tiramisu Compiler
- The Tensor Algebra Compiler (taco)

Benchmarks

- MLPerf
- AI - Benchmark
- AI Matrix.
- 中国人工智能产业发展联盟
- DAWNBench
- MLMark An EEMBC Benchmark

All information contained within this infographic is gathered from the internet and periodically updated, no guarantee is given that the information provided is correct, complete, and up-to-date.

SAFARI

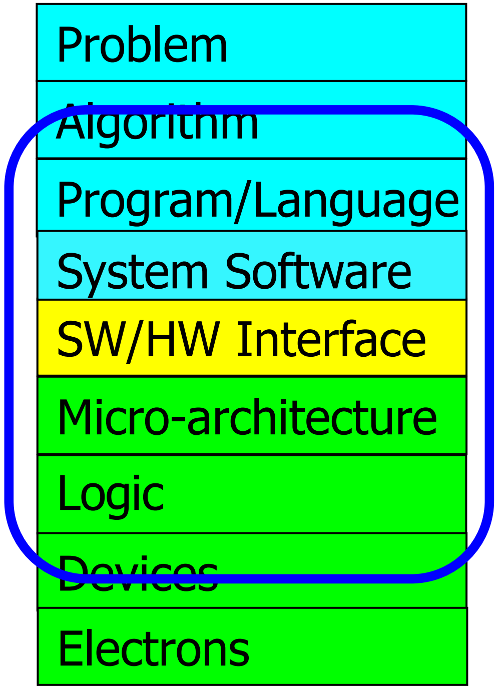
https://basicmi.github.io/AI-Chip/

Axiom

To achieve the highest efficiency, performance, robustness:

we must take the expanded view

of computer architecture



Co-design across the hierarchy:
Algorithms to devices

Specialize as much as possible
within the design goals

What Limits Us in Computing Today?

Increasingly Demanding Applications

Dream…

and, they will come

As applications push boundaries, computing platforms become increasingly strained

Many Metrics to Optimize for

- Performance
- Energy/Power
- Correctness
- Robustness (Safety, Security, Reliability, Availability)
- Cost
- Programming Ease
- Usability (Ease of Use)
- Scalability
- Simplicity (Complexity)
- …

Challenging especially with complex systems & hardware

Three Major Limiters to Computing

- Technology scaling is not going well
- System complexity is increasing; old methods not keeping up
- Processor-centric designs are not keeping up
- These affect all metrics we care about
- These have fundamental impact on security and how we build secure systems

Technology Scaling

Technology Scaling Problems

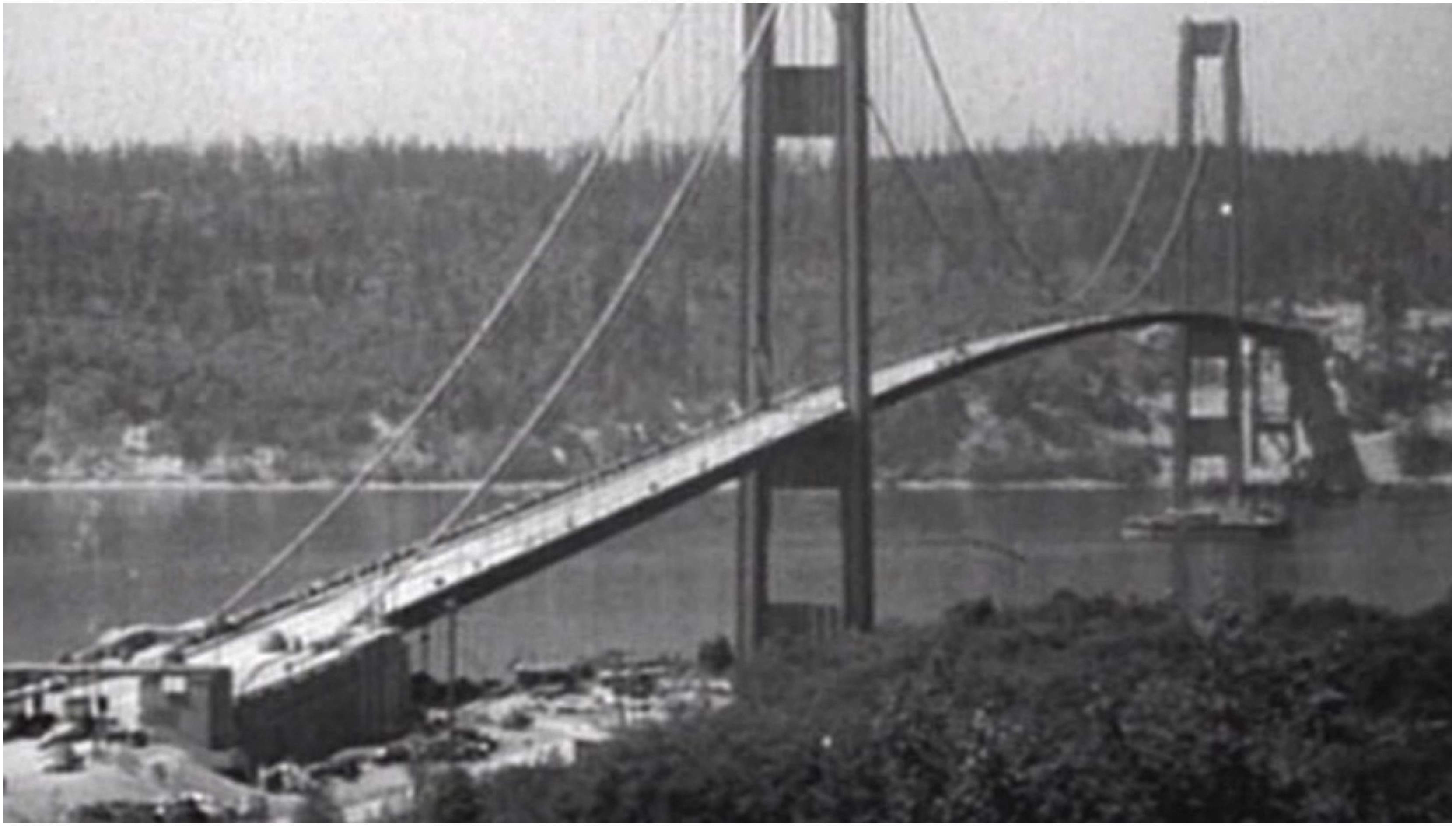
- Circuit size and energy reduction has enabled continuous innovation at all levels of the computing stack

- As circuits become smaller, they become less reliable

- More flaky circuits are a problem for robust (reliable, safe, secure) operation

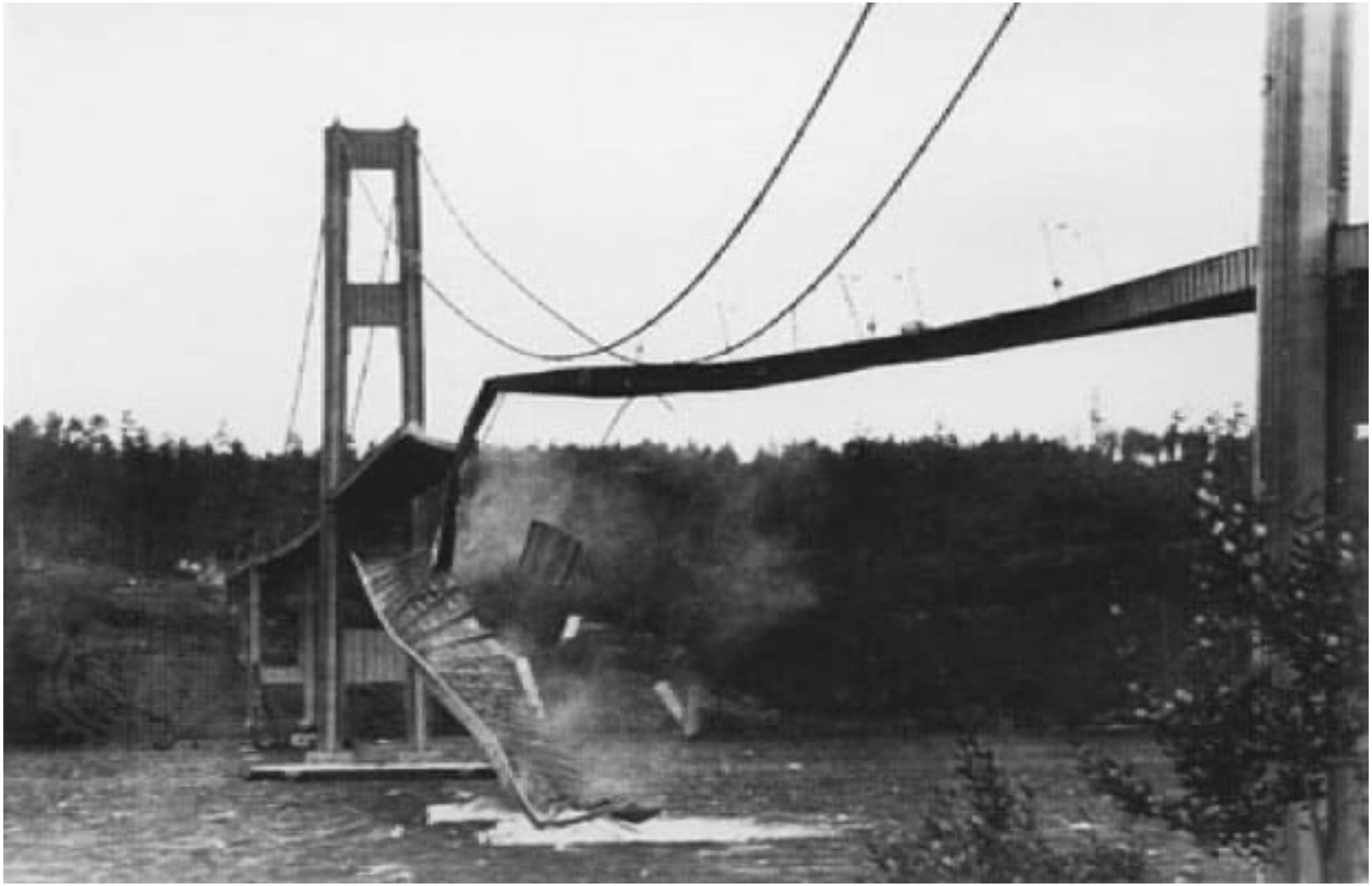
- If circuits produce wrong results, security can be affected (along with safety, reliability, availability)

How Reliable/Secure/Safe is This Bridge?



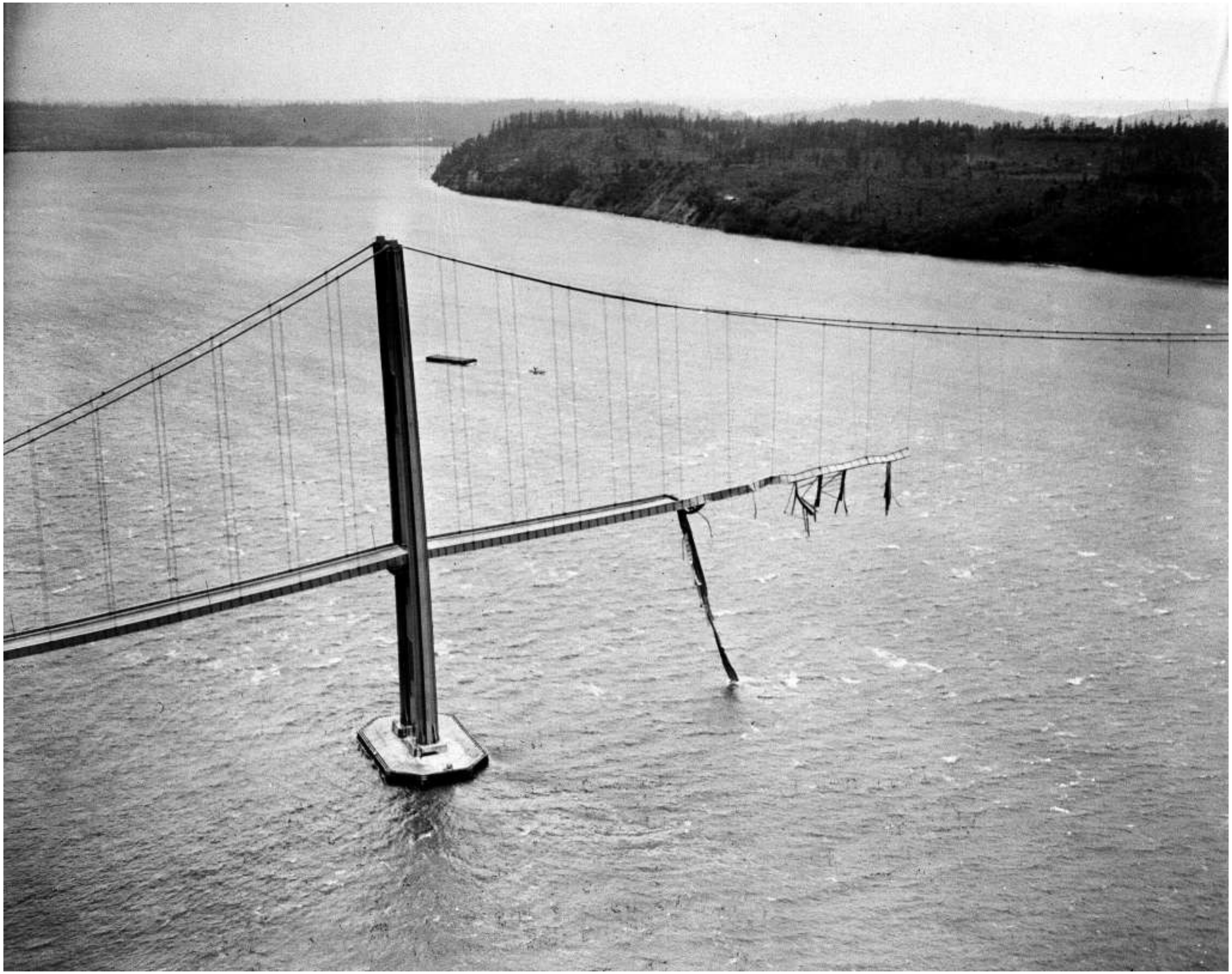
Source: http://www.technologystudent.com/struct1/tacom1.png

Collapse of the "Galloping Gertie"



Source: AP
http://www.wsdot.wa.gov/tnbhistory/connections/connections3.htm

Another View



Source: AP   Source: http://www.seattlepi.com/science/article/A-Tacoma-Narrows-Galloping-Gertie-bridge-6617030.php

How Secure Are These People?

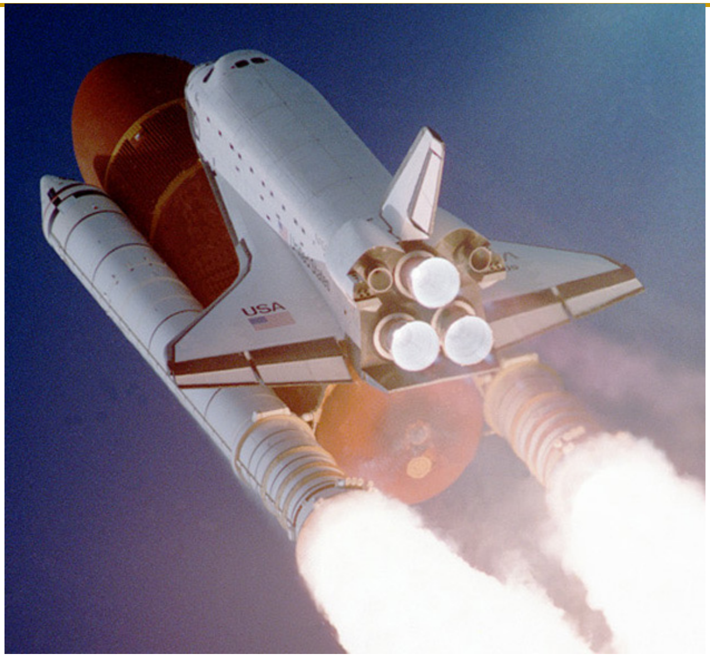
Security is about preventing unforeseen consequences

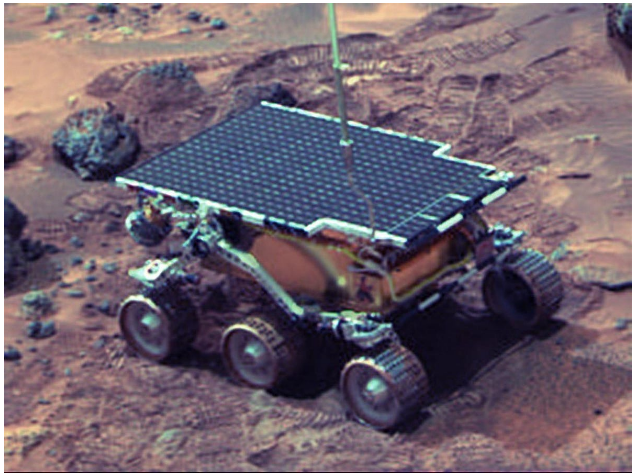
Source: https://s-media-cache-ak0.pinimg.com/originals/48/09/54/4809543a9c7700246a0cf8acdae27abf.jpg

How Safe & Secure Is **This** Platform?

Source: https://taxistartup.com/wp-content/uploads/2015/03/UK-Self-Driving-Cars.jpg

How Robust Are **These** Platforms?





https://www.kennedyspacecenter.com/explore-attractions/nasa-now
https://www.cnet.com/pictures/nasas-wildest-rides-extreme-vehicles-for-earth-and-beyond/7/

Challenge and Opportunity for Future

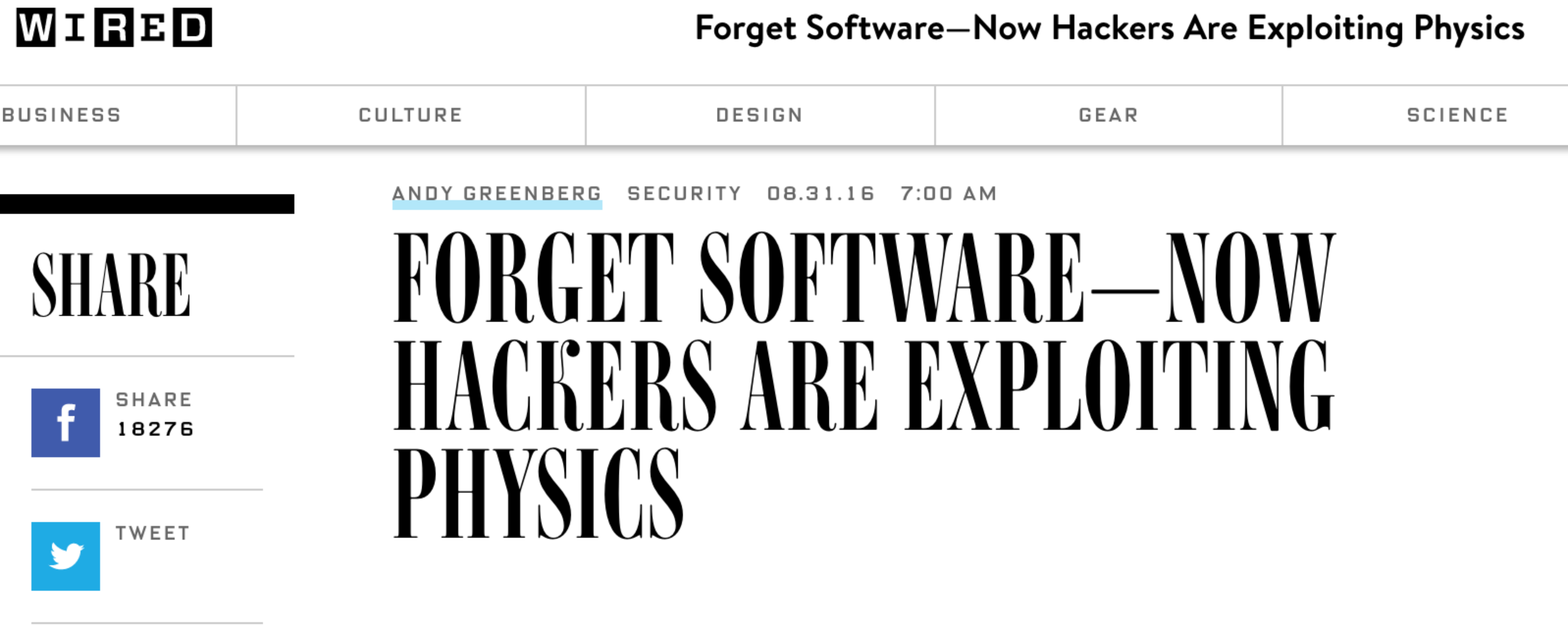
Robust

(Reliable, Secure, Safe)

An Example: The RowHammer Problem

- One can predictably induce bit flips in commodity DRAM chips
  - All recent DRAM chips are fundamentally vulnerable

- First example of how a simple hardware failure mechanism can create a widespread system security vulnerability



WIRED

Forget Software—Now Hackers Are Exploiting Physics

ANDY GREENBERG SECURITY 08.31.16 7:00 AM

FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

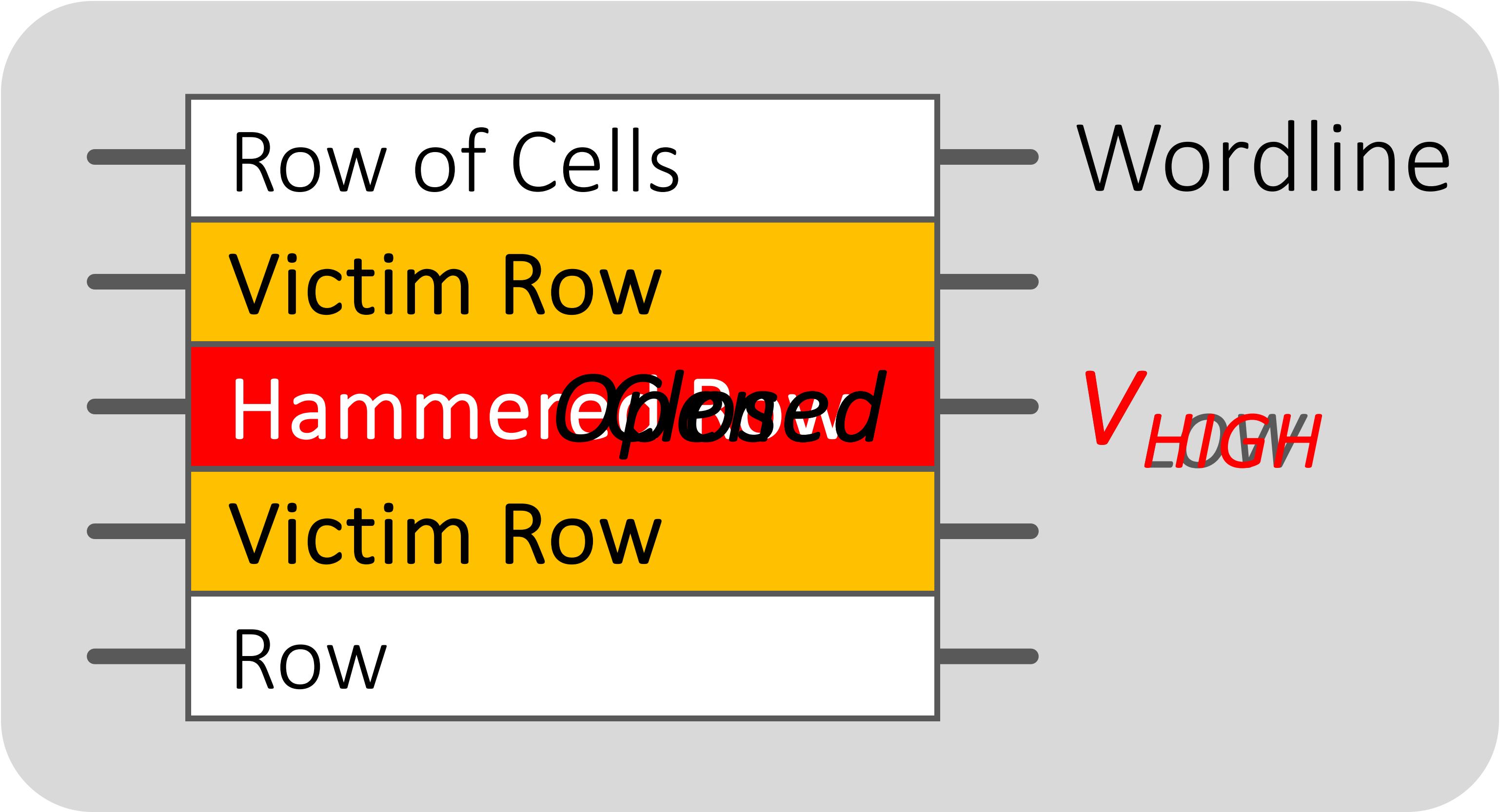
A Curious Phenomenon [Kim et al., ISCA 2014]

One can
predictably induce errors
in DRAM memory chips

Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.

Rowhammer

Modern Memory is Prone to Disturbance Errors



Row of Cells

Wordline

Victim Row

Hammered Row

Opened

Closed

V_{LOW}

V_{HIGH}

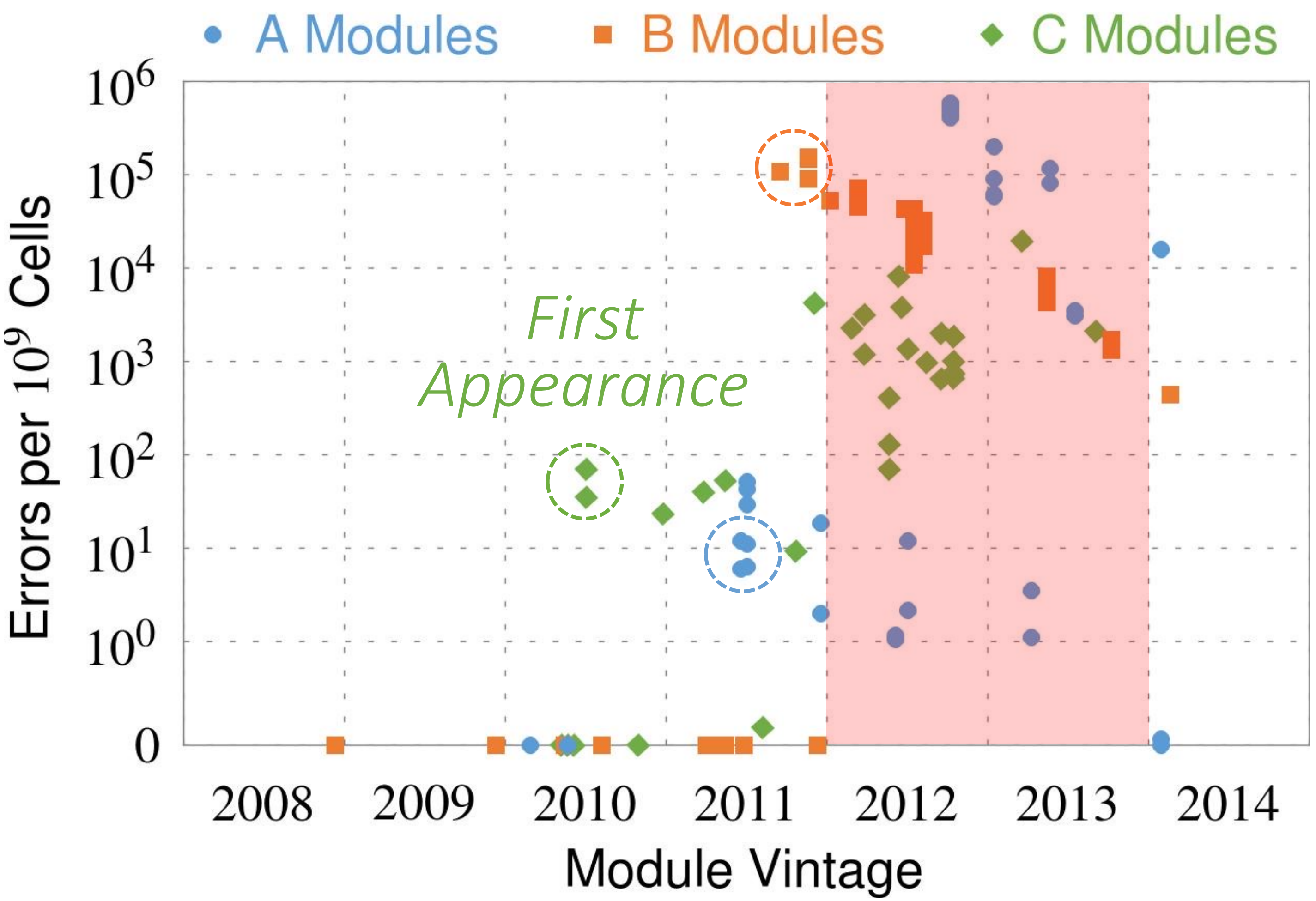
Victim Row

Row

Repeatedly reading a row enough times (before memory gets refreshed) induces disturbance errors in **adjacent rows** in most real DRAM chips you can buy today

Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors, (Kim et al., ISCA 2014)

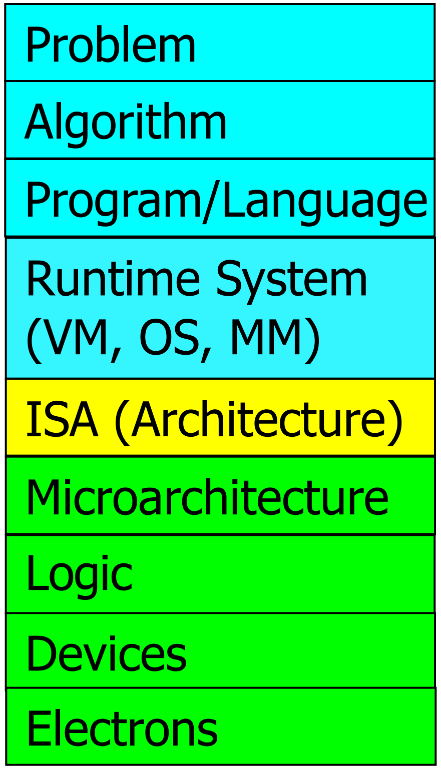
Recent DRAM Is More Vulnerable



All modules from 2012–2013 are vulnerable

Higher-Level Implications

- This simple circuit level failure mechanism has enormous implications on upper layers of the transformation hierarchy



Problem

Algorithm

Program/Language

Runtime System (VM, OS, MM)

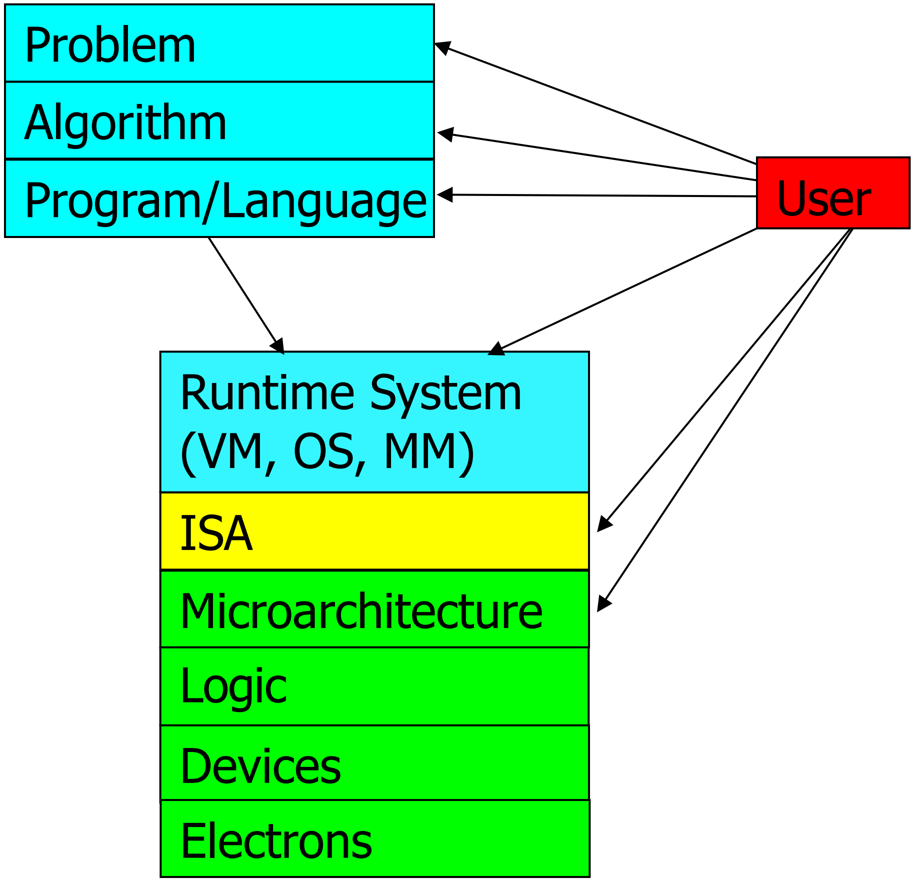
ISA (Architecture)

Microarchitecture

Logic

Devices

Electrons



Problem

Algorithm

Program/Language

User

Runtime System (VM, OS, MM)

ISA

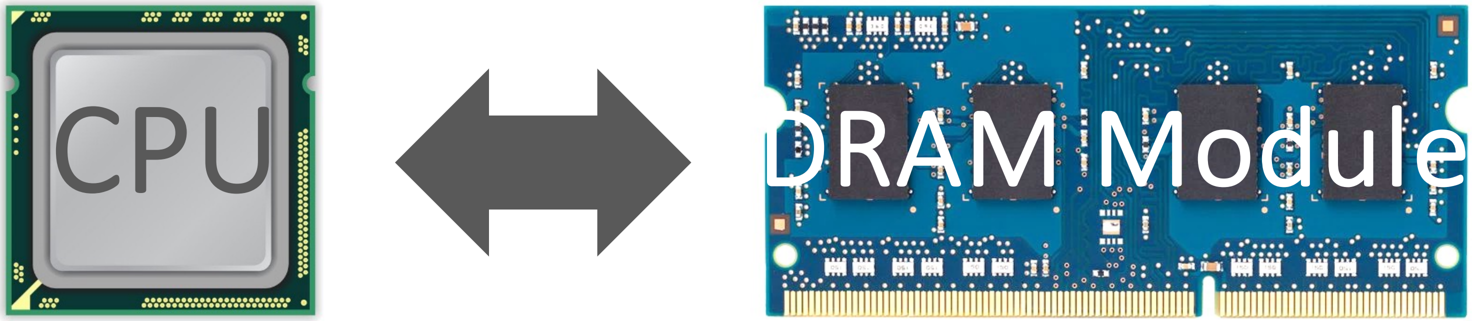
Microarchitecture

Logic

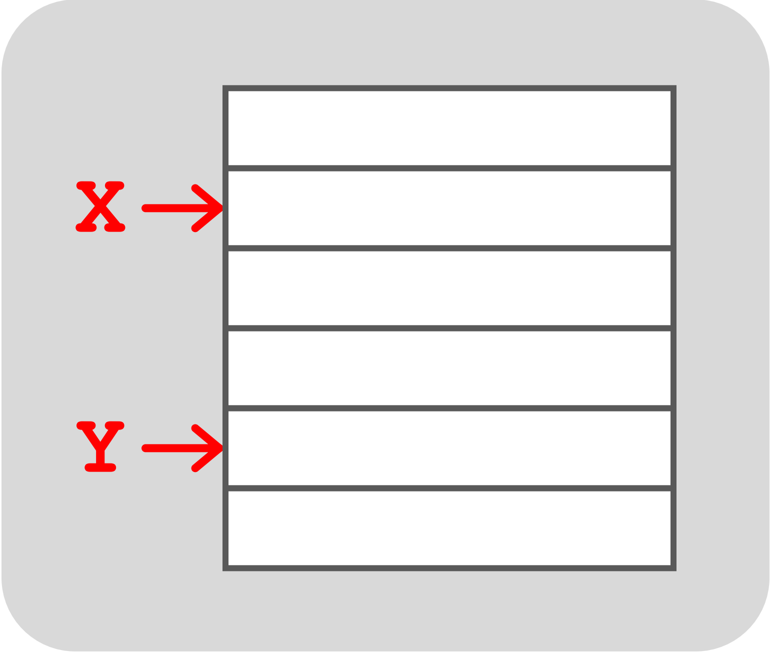
Devices

Electrons

A Simple Program Can Induce Many Errors

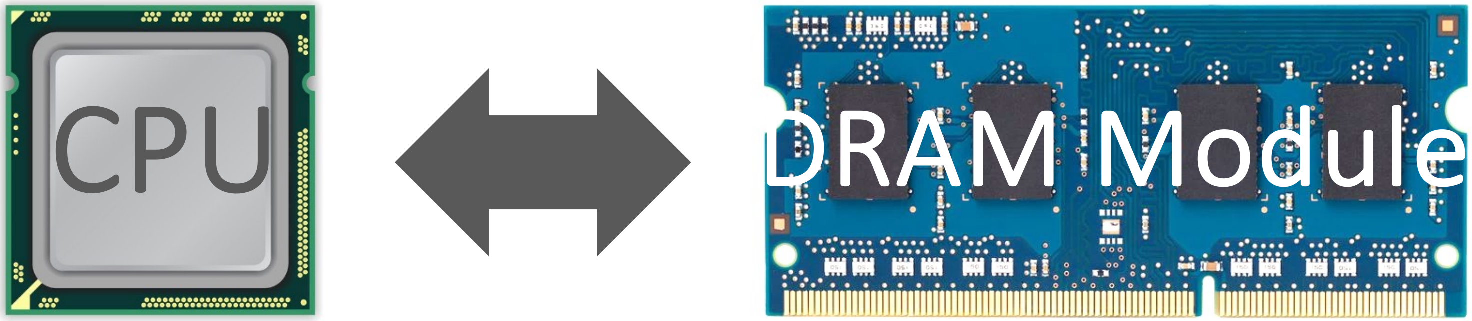


```
loop:
   mov  (X), %eax
   mov  (Y), %ebx
   clflush (X)
   clflush (Y)
   mfence
   jmp loop
```



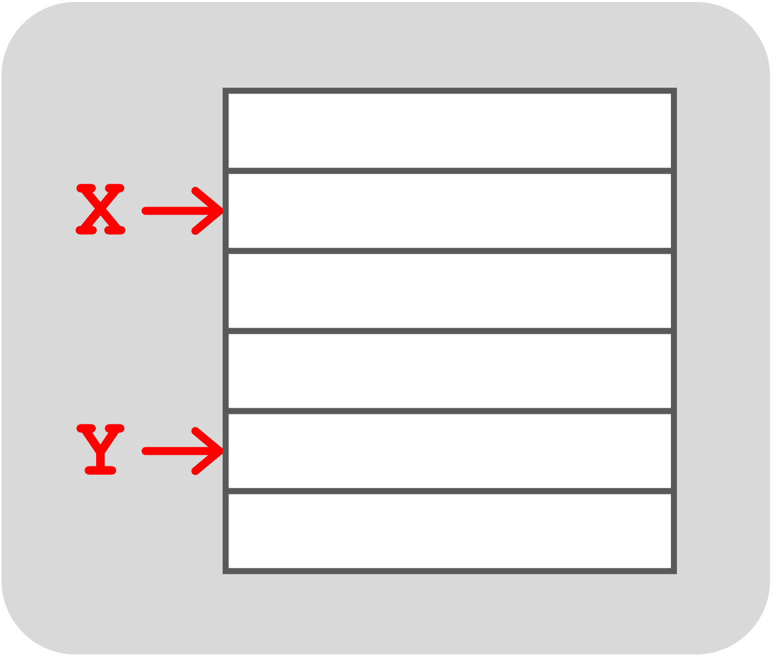
Download from: https://github.com/CMU-SAFARI/rowhammer

A Simple Program Can Induce Many Errors



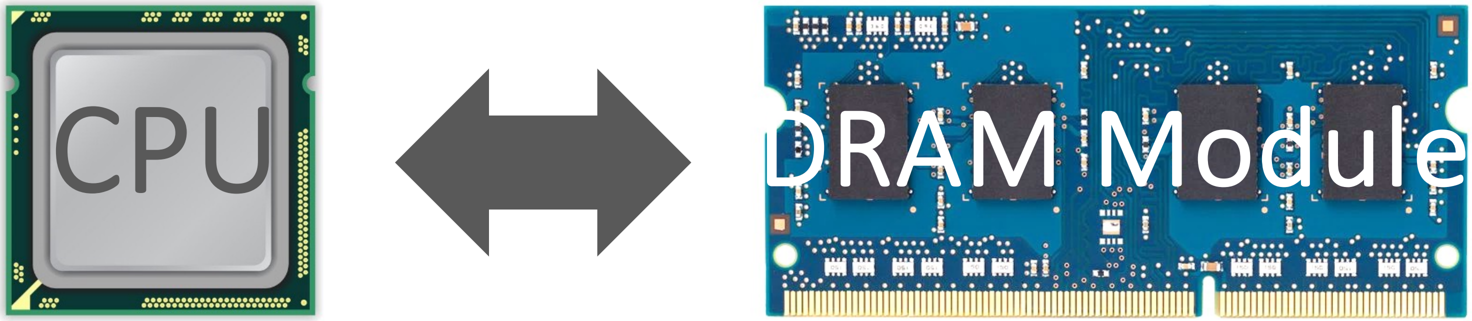
1. Avoid ***cache hits***
   - Flush **X** from cache

2. Avoid ***row hits*** to **X**
   - Read **Y** in another row

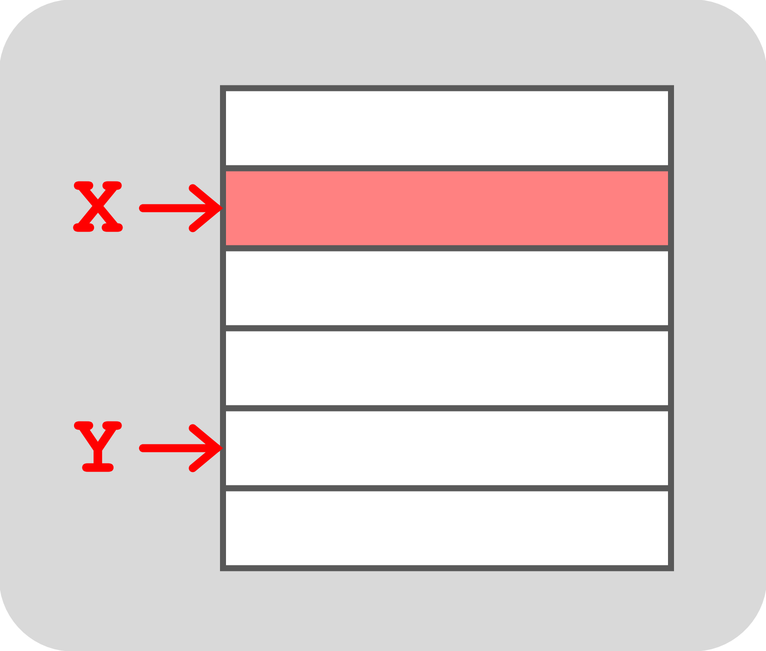


Download from: https://github.com/CMU-SAFARI/rowhammer

A Simple Program Can Induce Many Errors

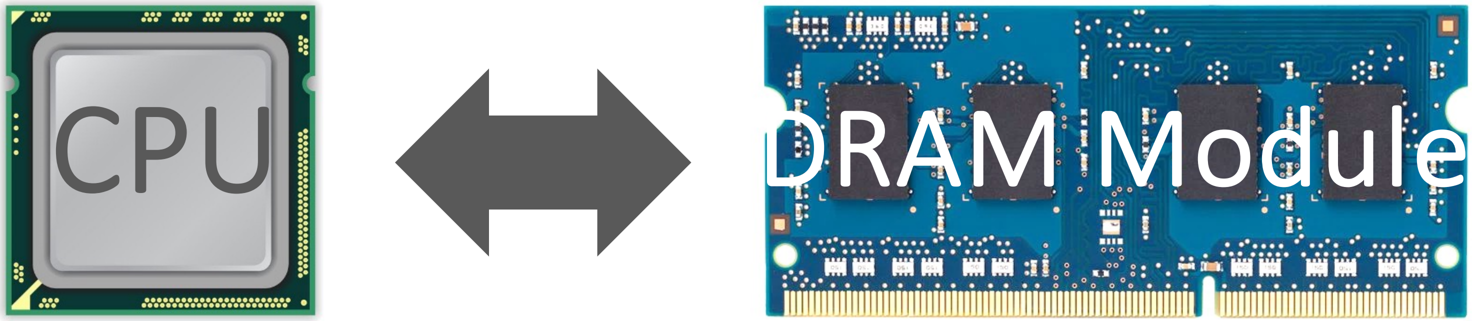


```
loop:
   mov  (X), %eax
   mov  (Y), %ebx
   clflush (X)
   clflush (Y)
   mfence
   jmp loop
```

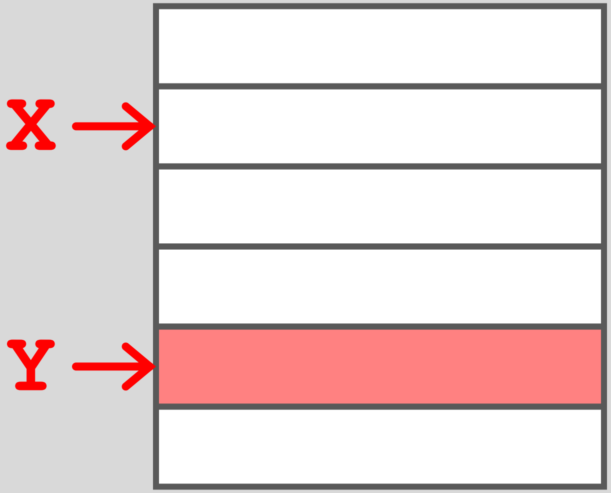


Download from: https://github.com/CMU-SAFARI/rowhammer

A Simple Program Can Induce Many Errors

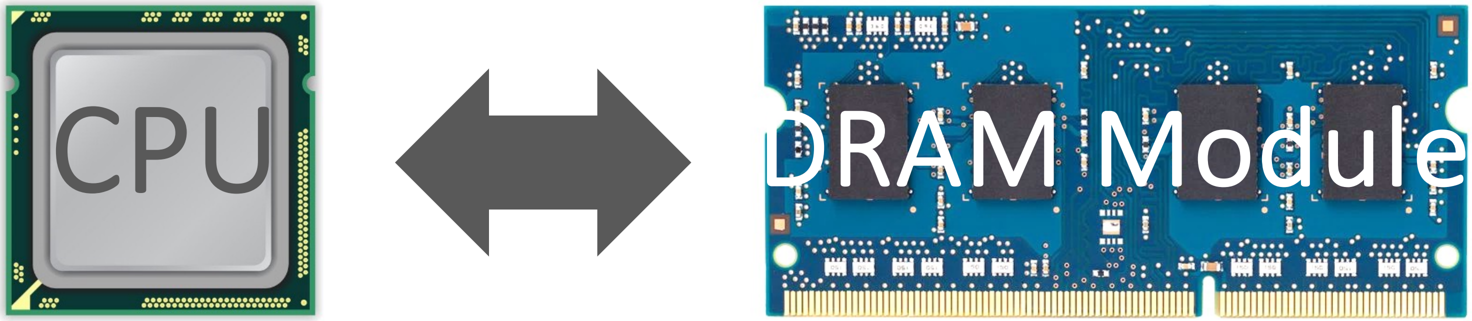


```
loop:
   mov (X), %eax
   mov (Y), %ebx
   clflush (X)
   clflush (Y)
   mfence
   jmp loop
```

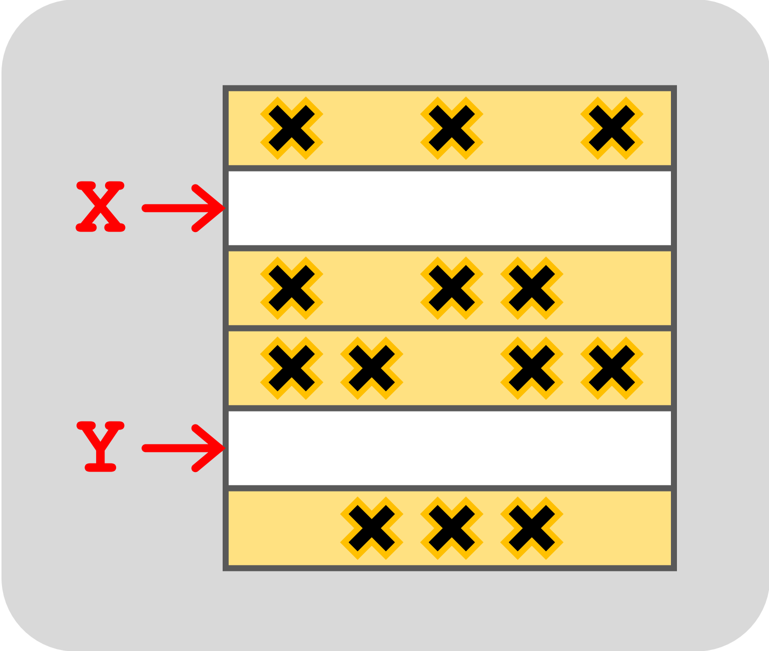


Download from: https://github.com/CMU-SAFARI/rowhammer

A Simple Program Can Induce Many Errors



```
loop:
  mov (X), %eax
  mov (Y), %ebx
  clflush (X)
  clflush (Y)
  mfence
  jmp loop
```



Download from: https://github.com/CMU-SAFARI/rowhammer

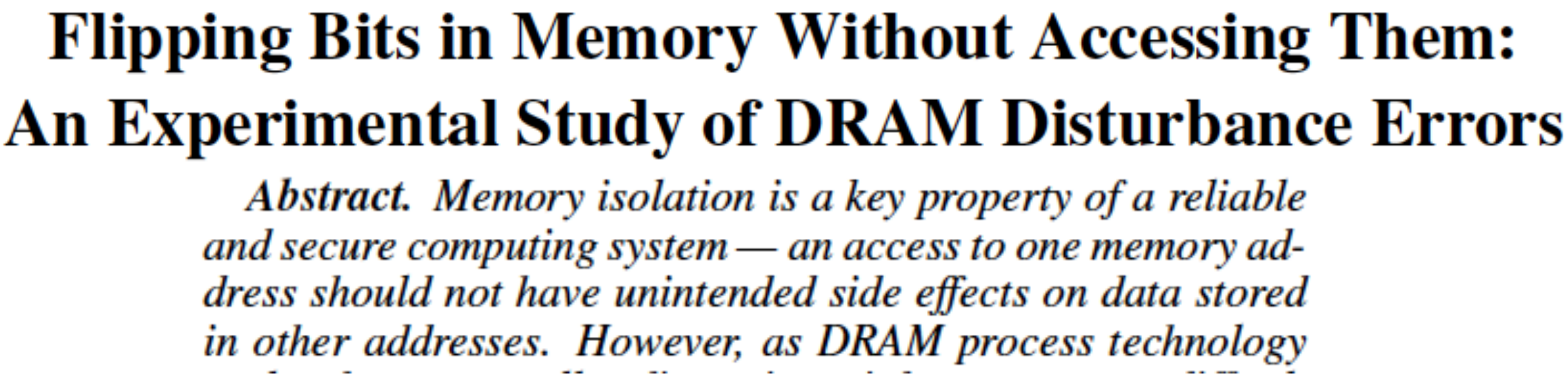
Observed Errors in Real Systems

CPU Architecture	Errors	Access-Rate
Intel Haswell (2013)	22.9K	12.3M/sec
Intel Ivy Bridge (2012)	20.7K	11.7M/sec
Intel Sandy Bridge (2011)	16.1K	11.6M/sec
AMD Piledriver (2012)	59	6.1M/sec

A real reliability, security, safety issue

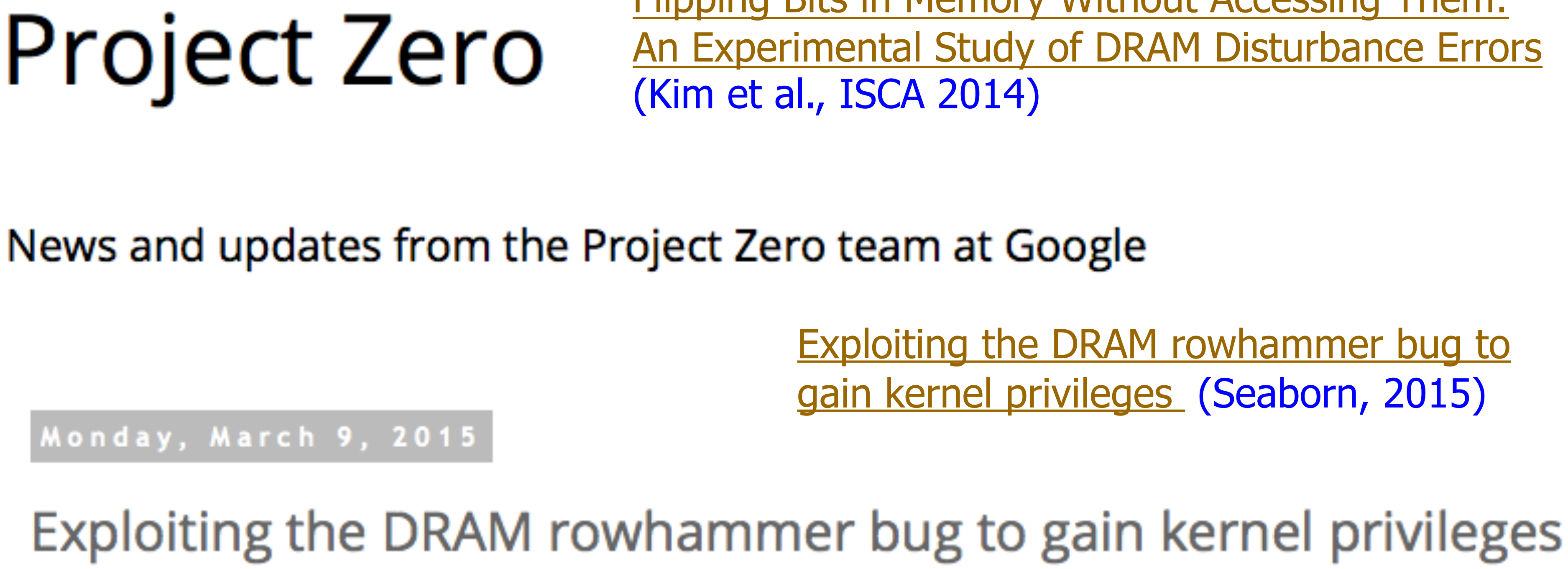
Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.

One Can Take Over an Otherwise-Secure System



Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors

Abstract. Memory isolation is a key property of a reliable and secure computing system — an access to one memory address should not have unintended side effects on data stored in other addresses. However, as DRAM process technology



Project Zero

News and updates from the Project Zero team at Google

Monday, March 9, 2015

Exploiting the DRAM rowhammer bug to gain kernel privileges

Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors (Kim et al., ISCA 2014)

Exploiting the DRAM rowhammer bug to gain kernel privileges (Seaborn, 2015)

Many RowHammer Security Exploits

- One can exploit RowHammer to
- Take over a system
- Read data they do not have access to
- Break out of virtual machine sandboxes
- Corrupt important data → render ML inference useless
- Steal secret data (e.g., crypto keys & ML model parameters)

Security Implications

Security Implications

It’s like breaking into an apartment by repeatedly slamming a neighbor’s door until the vibrations open the door you were after

Infrastructures to Understand Such Issues

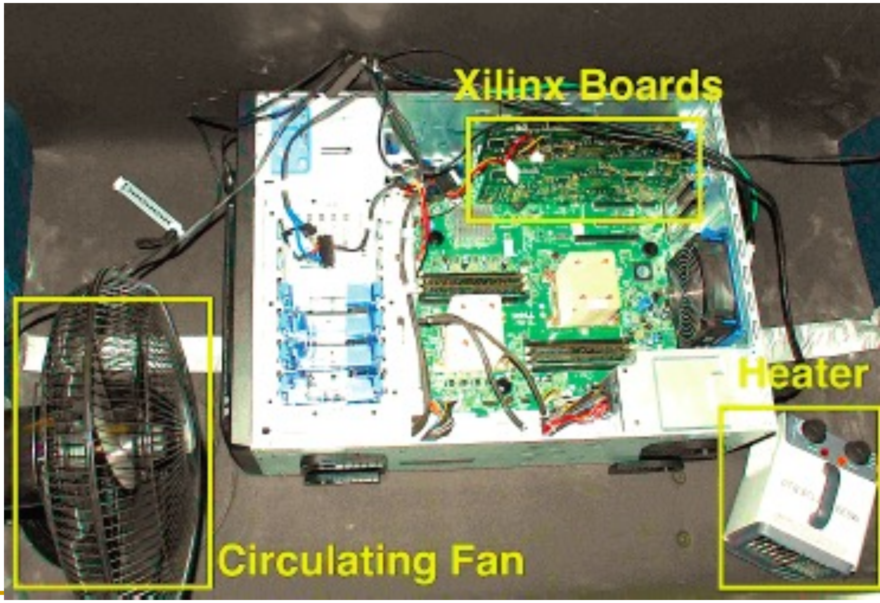
Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors (Kim et al., ISCA 2014)

Adaptive-Latency DRAM: Optimizing DRAM Timing for the Common-Case (Lee et al., HPCA 2015)

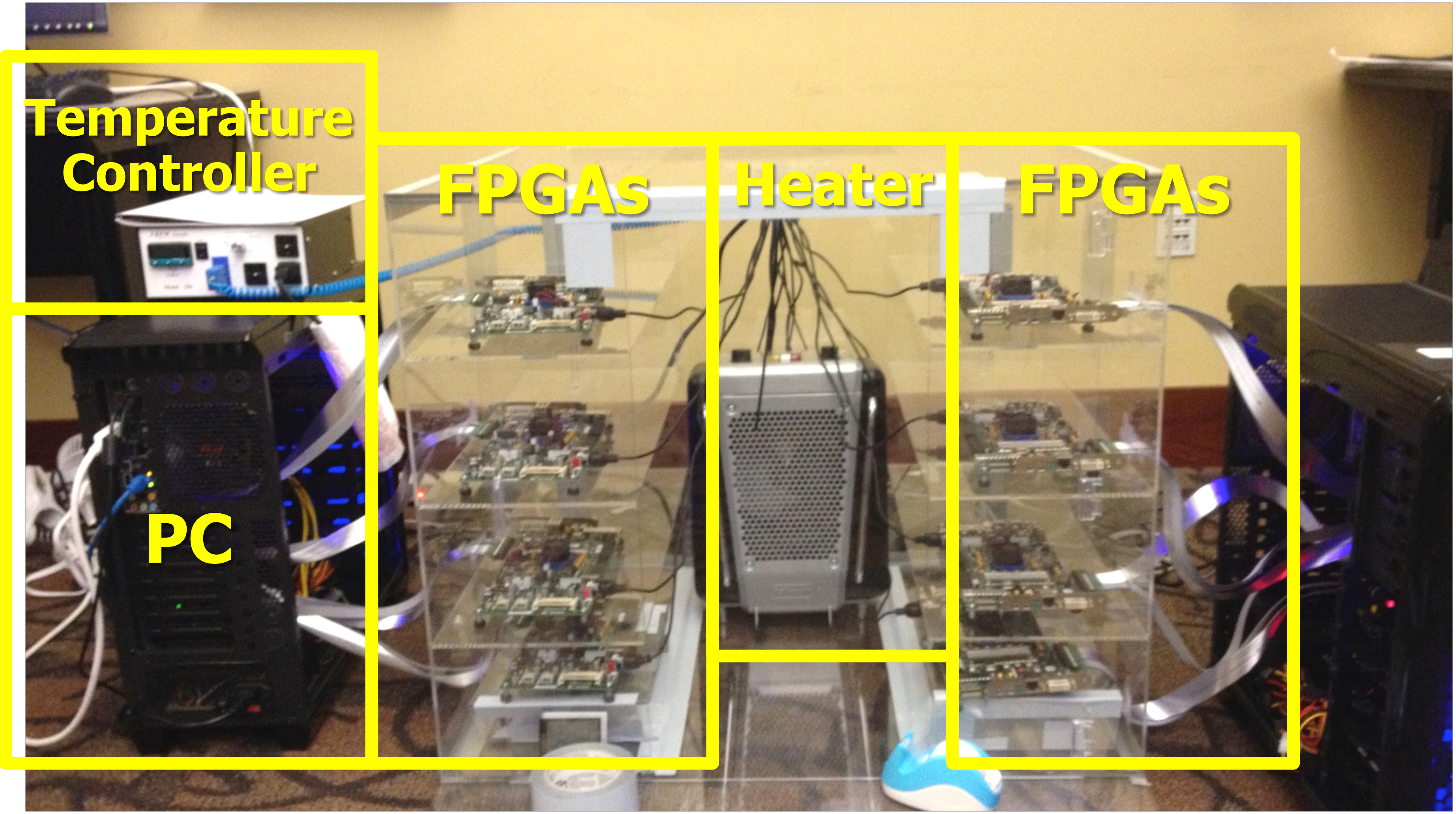
AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems (Qureshi et al., DSN 2015)

An Experimental Study of Data Retention Behavior in Modern DRAM Devices: Implications for Retention Time Profiling Mechanisms (Liu et al., ISCA 2013)

The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study (Khan et al., SIGMETRICS 2014)



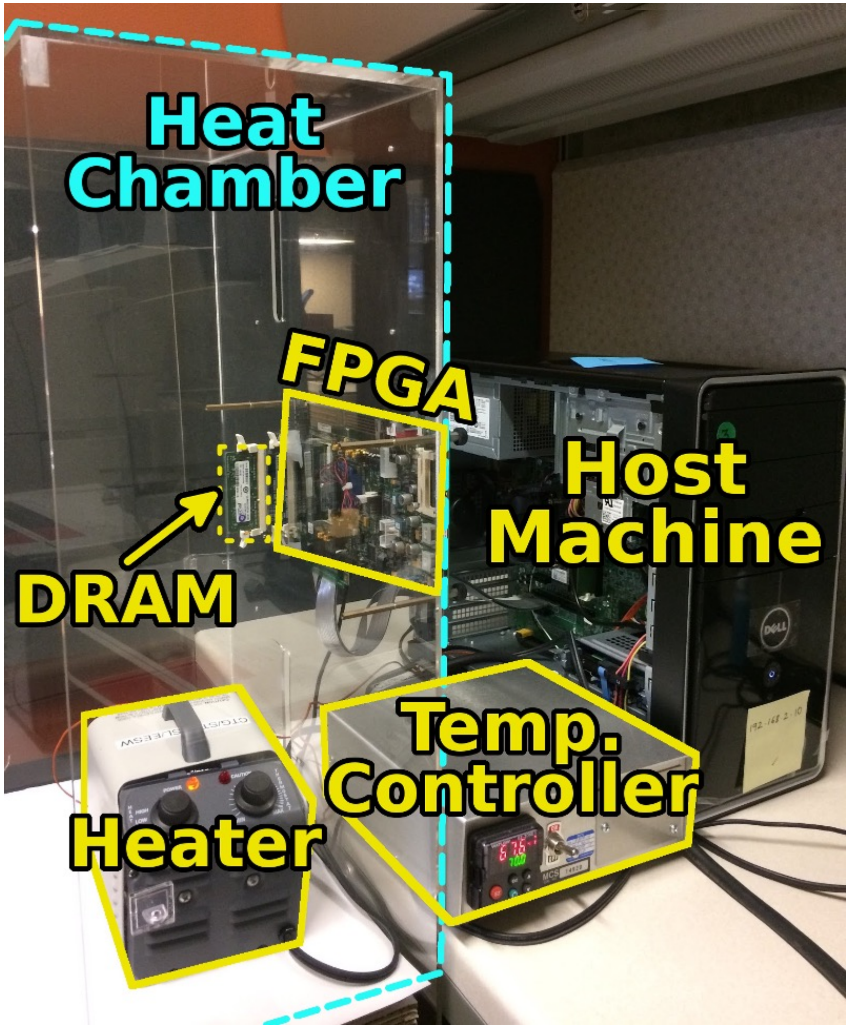
Infrastructures to Understand Such Issues



Kim+, “Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors,” ISCA 2014.

SoftMC: Open Source DRAM Infrastructure

- Hasan Hassan et al., "**SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies**," HPCA 2017.

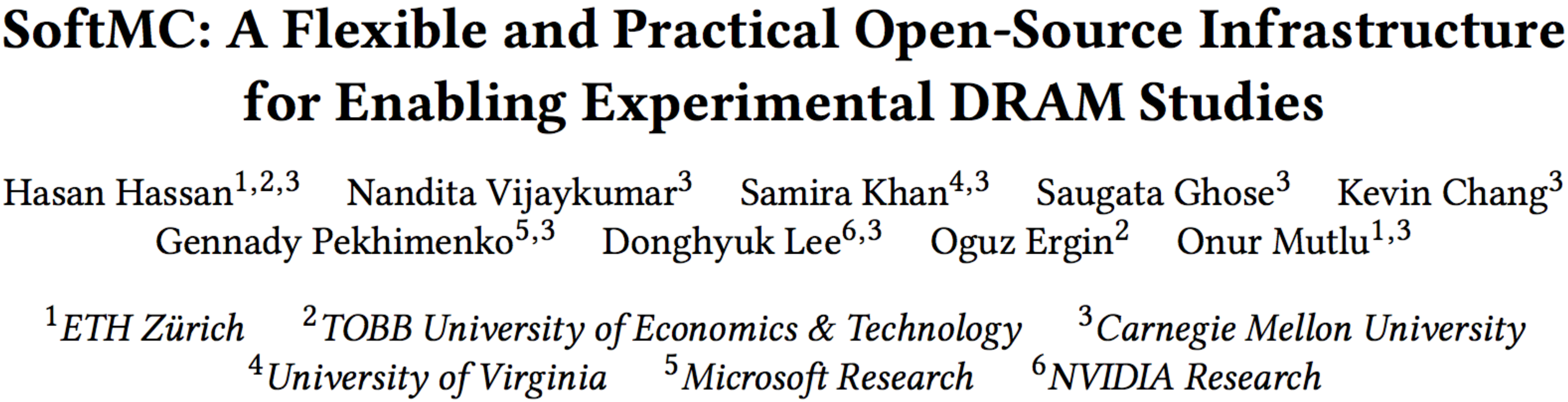


- **Flexible**
- **Easy to Use (C++ API)**
- **Open-source**

  github.com/CMU-SAFARI/SoftMC

SoftMC: Open Source DRAM Infrastructure

- Hasan Hassan, Nandita Vijaykumar, Samira Khan, Saugata Ghose, Kevin Chang, Gennady Pekhimenko, Donghyuk Lee, Oguz Ergin, and Onur Mutlu,
  "SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies"
  Proceedings of the 23rd International Symposium on High-Performance Computer Architecture (***HPCA***), Austin, TX, USA, February 2017.
  [Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)]
  [Full Talk Lecture (39 minutes)]
  [Source Code]



SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies

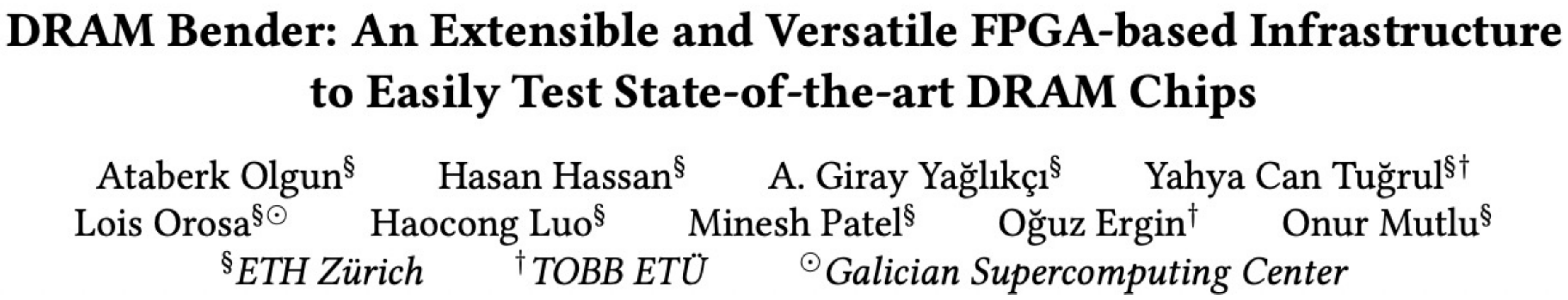
Hasan Hassan[1,2,3] Nandita Vijaykumar[3] Samira Khan[4,3] Saugata Ghose[3] Kevin Chang[3]
Gennady Pekhimenko[5,3] Donghyuk Lee[6,3] Oguz Ergin[2] Onur Mutlu[1,3]

[1]*ETH Zürich* [2]*TOBB University of Economics & Technology* [3]*Carnegie Mellon University*
[4]*University of Virginia* [5]*Microsoft Research* [6]*NVIDIA Research*

https://github.com/CMU-SAFARI/SoftMC

DRAM Bender: New DRAM Infrastructure

- Ataberk Olgun, Hasan Hassan, A Giray Yağlıkçı, Yahya Can Tuğrul, Lois Orosa, Haocong Luo, Minesh Patel, Oğuz Ergin, and Onur Mutlu,
  "DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips"
  IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (***TCAD***), 2023.
  [Extended arXiv version]
  [DRAM Bender Source Code]
  [DRAM Bender Tutorial Video (43 minutes)]



DRAM Bender: An Extensible and Versatile FPGA-based Infrastructure to Easily Test State-of-the-art DRAM Chips

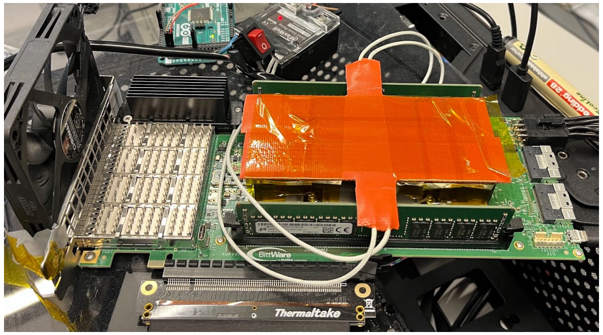
Ataberk Olgun[§] Hasan Hassan[§] A. Giray Yağlıkçı[§] Yahya Can Tuğrul[§†]
Lois Orosa[§⊙] Haocong Luo[§] Minesh Patel[§] Oğuz Ergin[†] Onur Mutlu[§]
[§] *ETH Zürich* [†] *TOBB ETÜ* [⊙] *Galician Supercomputing Center*

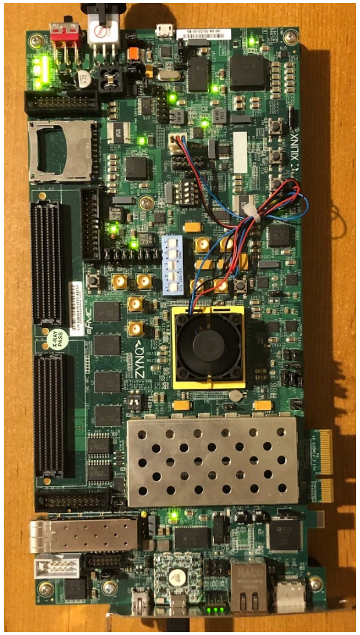
https://github.com/CMU-SAFARI/DRAM-Bender

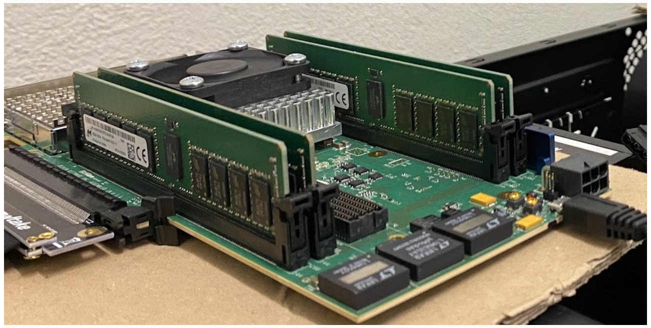
DRAM Bender: FPGA Prototypes

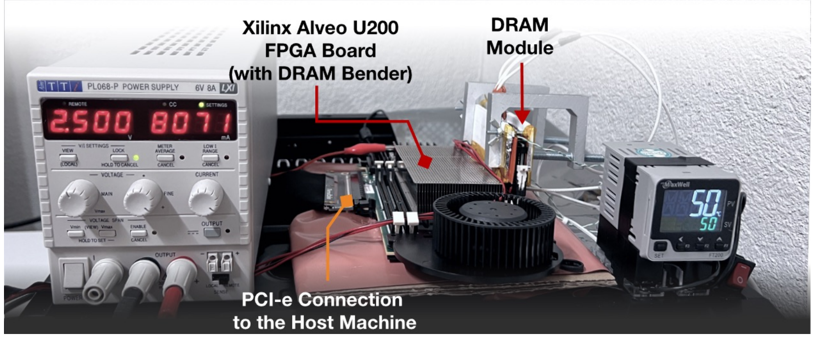
Testing Infrastructure	**Protocol Support**	**FPGA Support**
SoftMC [134]	DDR3	One Prototype
LiteX RowHammer Tester (LRT) [17]	DDR3/4, LPDDR4	Two Prototypes
DRAM Bender (this work)	**DDR3/DDR4**	**Five Prototypes**

Five out of the box FPGA-based prototypes





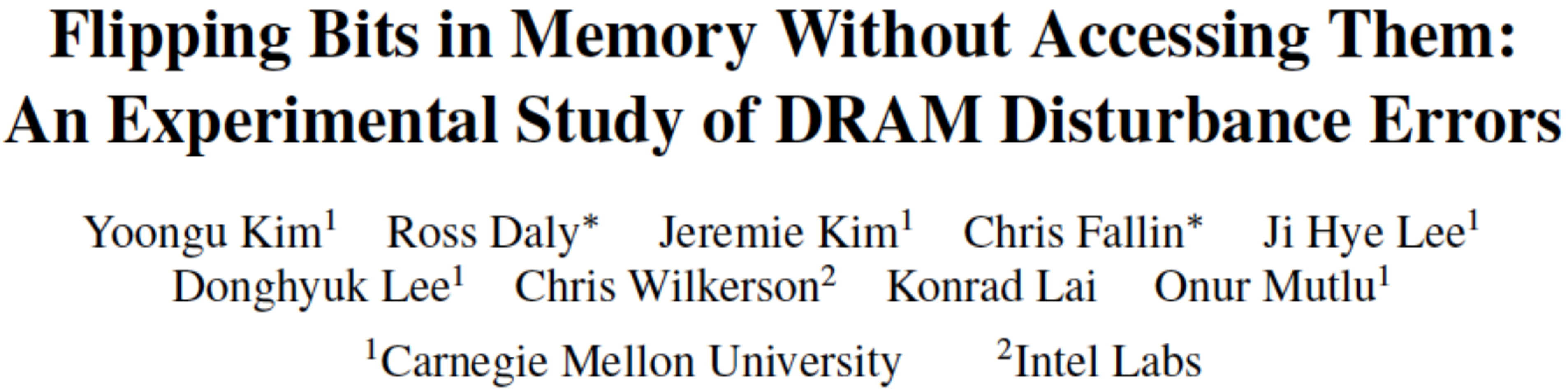




https://github.com/CMU-SAFARI/DRAM-Bender

RowHammer **[ISCA 2014]**

- Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu,
  "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors"
  Proceedings of the 41st International Symposium on Computer Architecture (***ISCA***), Minneapolis, MN, June 2014.
  [Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)] [Source Code and Data] [Lecture Video (1 hr 49 mins), 25 September 2020]
  One of the 7 papers of 2012-2017 selected as Top Picks in Hardware and Embedded Security for IEEE TCAD (link). Selected to the ISCA-50 25-Year Retrospective Issue covering 1996-2020 in 2023 (Retrospective (pdf) Full Issue). Winner of the 2024 IFIP Jean-Claude Laprie Award in dependable computing (link).



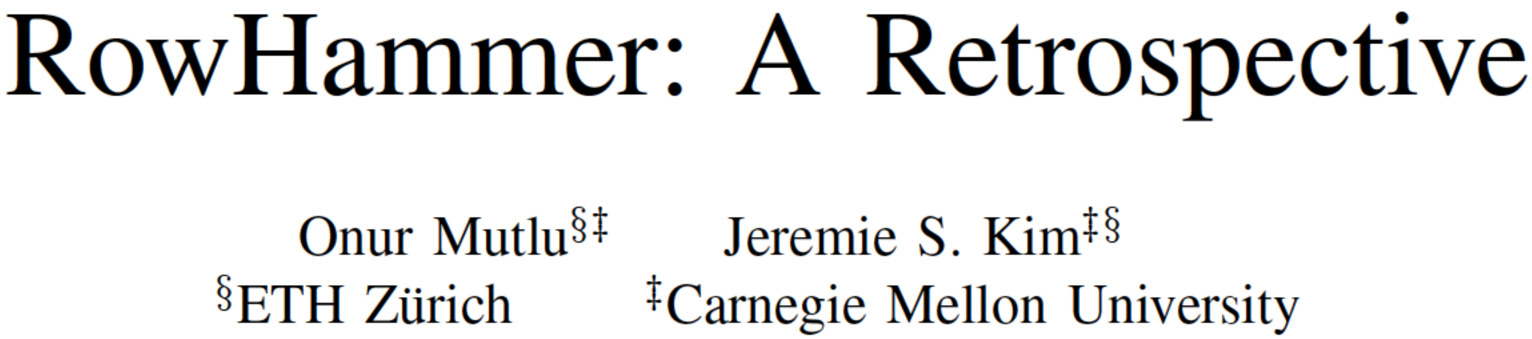
Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors

Yoongu Kim[1] Ross Daly* Jeremie Kim[1] Chris Fallin* Ji Hye Lee[1]
Donghyuk Lee[1] Chris Wilkerson[2] Konrad Lai Onur Mutlu[1]

[1]Carnegie Mellon University [2]Intel Labs

Memory Scaling Issues **Are** Real

- Onur Mutlu and Jeremie Kim,
  "RowHammer: A Retrospective"
  IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (***TCAD***) *Special Issue on Top Picks in Hardware and Embedded Security*, 2019.
  [Preliminary arXiv version]
  [Slides from COSADE 2019 (pptx)]
  [Slides from VLSI-SOC 2020 (pptx) (pdf)]
  [Talk Video (1 hr 15 minutes, with Q&A)]



RowHammer: A Retrospective

Onur Mutlu[§‡] Jeremie S. Kim[‡§]
[§]ETH Zürich [‡]Carnegie Mellon University

Memory Scaling Issues **Are** Real

- Onur Mutlu, Ataberk Olgun, and A. Giray Yaglikci,
  "Fundamentally Understanding and Solving RowHammer"
  *Invited Special Session Paper at the 28th Asia and South Pacific Design Automation Conference (**ASP-DAC**)*, Tokyo, Japan, January 2023.
  [arXiv version]
  [Slides (pptx) (pdf)]
  [Talk Video (26 minutes)]

Fundamentally Understanding and Solving RowHammer

Onur Mutlu
onur.mutlu@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

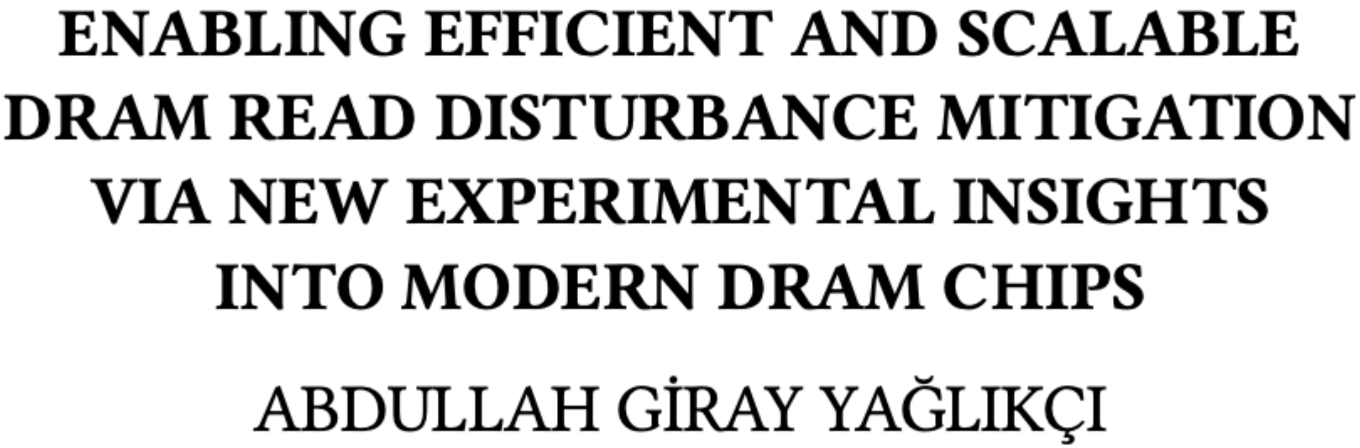
Ataberk Olgun
ataberk.olgun@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

A. Giray Yağlıkcı
giray.yaglikci@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

https://arxiv.org/pdf/2211.07613.pdf

A Very Recent PhD Thesis

- A. Giray Yaglikci, "**Enabling Efficient and Scalable DRAM Read Disturbance Mitigation via New Experimental Insights into Modern DRAM Chips**," PhD Thesis, ETH Zürich, 2024.
  [Slides (pdf) (pptx)]
  [Thesis arXiv (abs) (pdf)]
  [SAFARI News]



ENABLING EFFICIENT AND SCALABLE DRAM READ DISTURBANCE MITIGATION VIA NEW EXPERIMENTAL INSIGHTS INTO MODERN DRAM CHIPS

ABDULLAH GİRAY YAĞLIKÇI

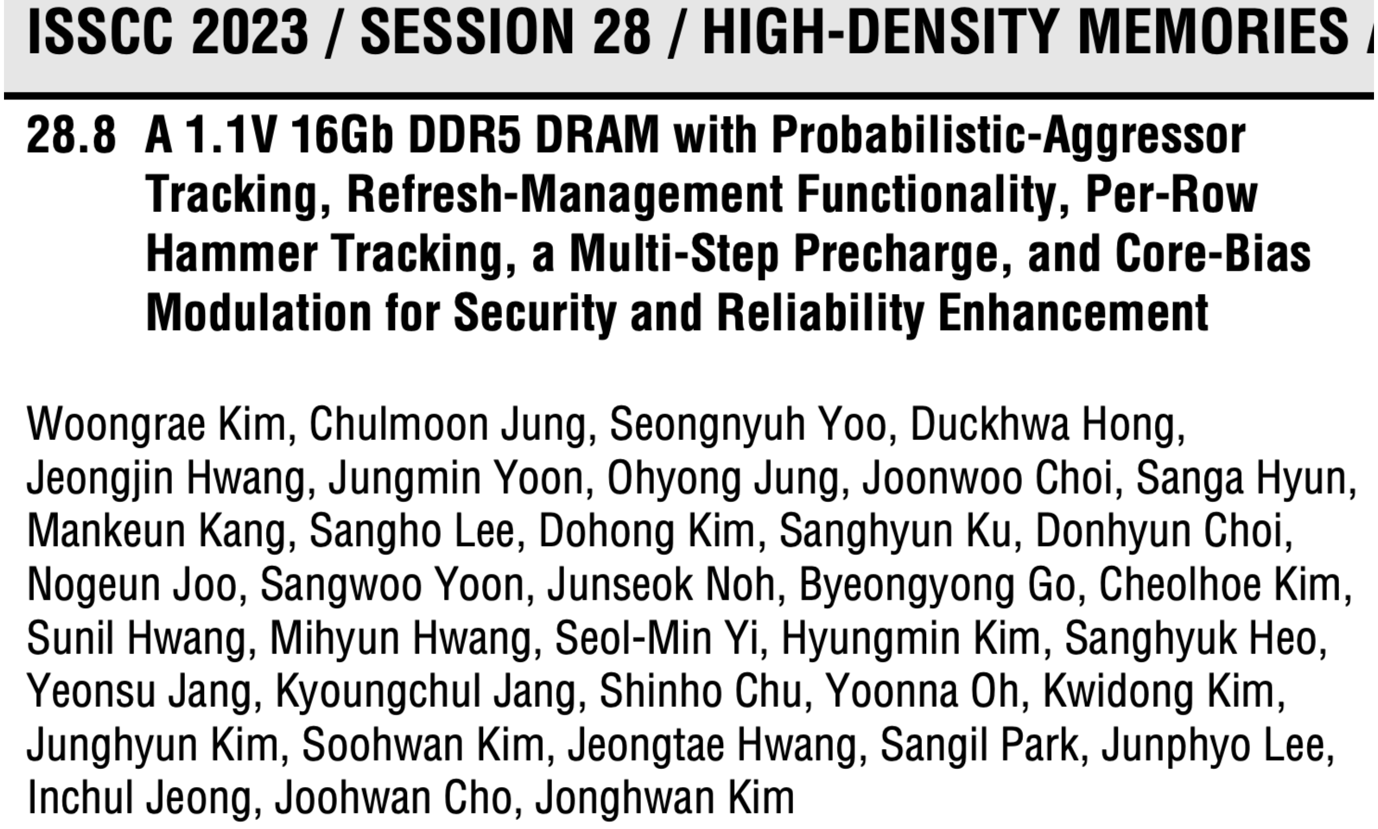
https://arxiv.org/pdf/2408.15044.pdf

The Push from Circuits and Devices

Main Memory Needs

Intelligent Controllers

Industry's Intelligent DRAM Controllers (I)



ISSCC 2023 / SESSION 28 / HIGH-DENSITY MEMORIES /

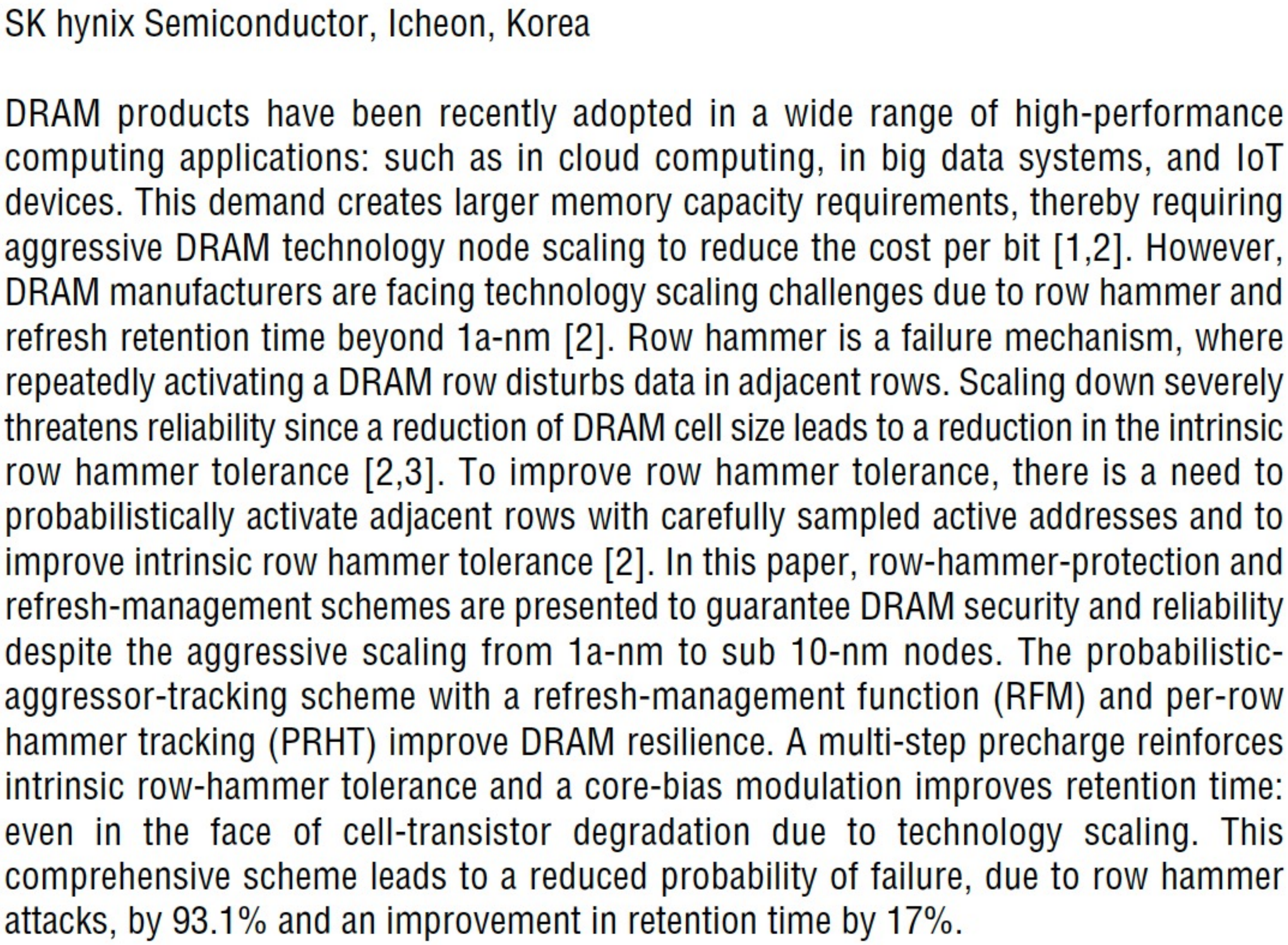
28.8 A 1.1V 16Gb DDR5 DRAM with Probabilistic-Aggressor Tracking, Refresh-Management Functionality, Per-Row Hammer Tracking, a Multi-Step Precharge, and Core-Bias Modulation for Security and Reliability Enhancement

Woongrae Kim, Chulmoon Jung, Seongnyuh Yoo, Duckhwa Hong, Jeongjin Hwang, Jungmin Yoon, Ohyong Jung, Joonwoo Choi, Sanga Hyun, Mankeun Kang, Sangho Lee, Dohong Kim, Sanghyun Ku, Donhyun Choi, Nogeun Joo, Sangwoo Yoon, Junseok Noh, Byeongyong Go, Cheolhoe Kim, Sunil Hwang, Mihyun Hwang, Seol-Min Yi, Hyungmin Kim, Sanghyuk Heo, Yeonsu Jang, Kyoungchul Jang, Shinho Chu, Yoonna Oh, Kwidong Kim, Junghyun Kim, Soohwan Kim, Jeongtae Hwang, Sangil Park, Junphyo Lee, Inchul Jeong, Joohwan Cho, Jonghwan Kim

SK hynix Semiconductor, Icheon, Korea

ISSCC 2023
2023 International Solid-State Circuits Conference
February 19-23, 2023 San Francisco, CA

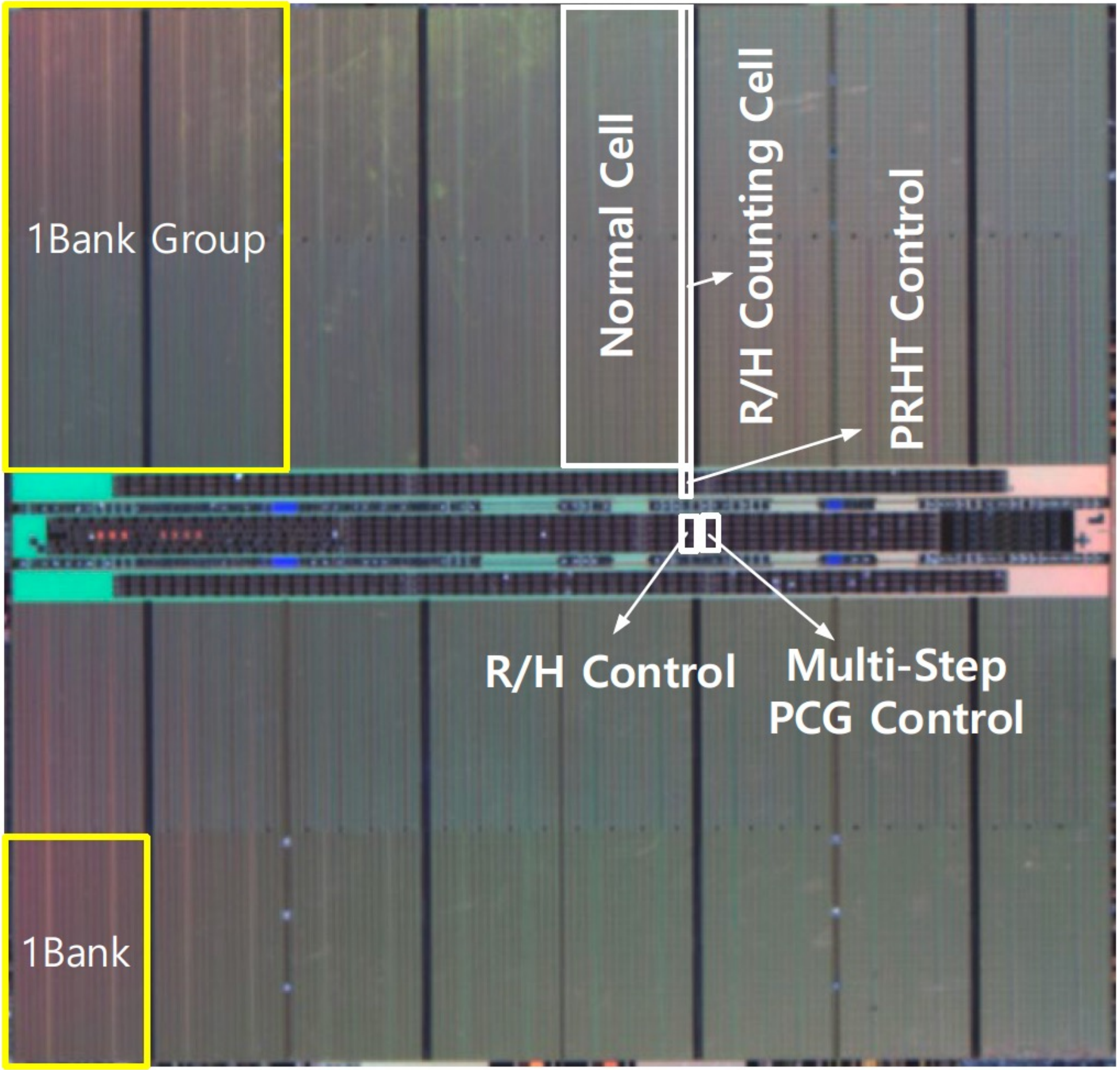
Industry's Intelligent DRAM Controllers (II)

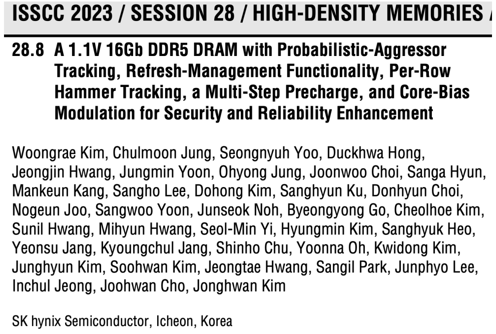


SK hynix Semiconductor, Icheon, Korea

DRAM products have been recently adopted in a wide range of high-performance computing applications: such as in cloud computing, in big data systems, and IoT devices. This demand creates larger memory capacity requirements, thereby requiring aggressive DRAM technology node scaling to reduce the cost per bit [1,2]. However, DRAM manufacturers are facing technology scaling challenges due to row hammer and refresh retention time beyond 1a-nm [2]. Row hammer is a failure mechanism, where repeatedly activating a DRAM row disturbs data in adjacent rows. Scaling down severely threatens reliability since a reduction of DRAM cell size leads to a reduction in the intrinsic row hammer tolerance [2,3]. To improve row hammer tolerance, there is a need to probabilistically activate adjacent rows with carefully sampled active addresses and to improve intrinsic row hammer tolerance [2]. In this paper, row-hammer-protection and refresh-management schemes are presented to guarantee DRAM security and reliability despite the aggressive scaling from 1a-nm to sub 10-nm nodes. The probabilistic-aggressor-tracking scheme with a refresh-management function (RFM) and per-row hammer tracking (PRHT) improve DRAM resilience. A multi-step precharge reinforces intrinsic row-hammer tolerance and a core-bias modulation improves retention time: even in the face of cell-transistor degradation due to technology scaling. This comprehensive scheme leads to a reduced probability of failure, due to row hammer attacks, by 93.1% and an improvement in retention time by 17%.

Industry's Intelligent DRAM Controllers (III)





ISSCC 2023 / SESSION 28 / HIGH-DENSITY MEMORIES /

28.8 A 1.1V 16Gb DDR5 DRAM with Probabilistic-Aggressor Tracking, Refresh-Management Functionality, Per-Row Hammer Tracking, a Multi-Step Precharge, and Core-Bias Modulation for Security and Reliability Enhancement

Woongrae Kim, Chulmoon Jung, Seongnyuh Yoo, Duckhwa Hong, Jeongjin Hwang, Jungmin Yoon, Ohyong Jung, Joonwoo Choi, Sanga Hyun, Mankeun Kang, Sangho Lee, Dohong Kim, Sanghyun Ku, Donhyun Choi, Nogeun Joo, Sangwoo Yoon, Junseok Noh, Byeongyong Go, Cheolhoe Kim, Sunil Hwang, Mihyun Hwang, Seol-Min Yi, Hyungmin Kim, Sanghyuk Heo, Yeonsu Jang, Kyoungchul Jang, Shinho Chu, Yoonna Oh, Kwidong Kim, Junghyun Kim, Soohwan Kim, Jeongtae Hwang, Sangil Park, Junphyo Lee, Inchul Jeong, Joohwan Cho, Jonghwan Kim

SK hynix Semiconductor, Icheon, Korea

Industry's Intelligent DRAM Controllers (IV)

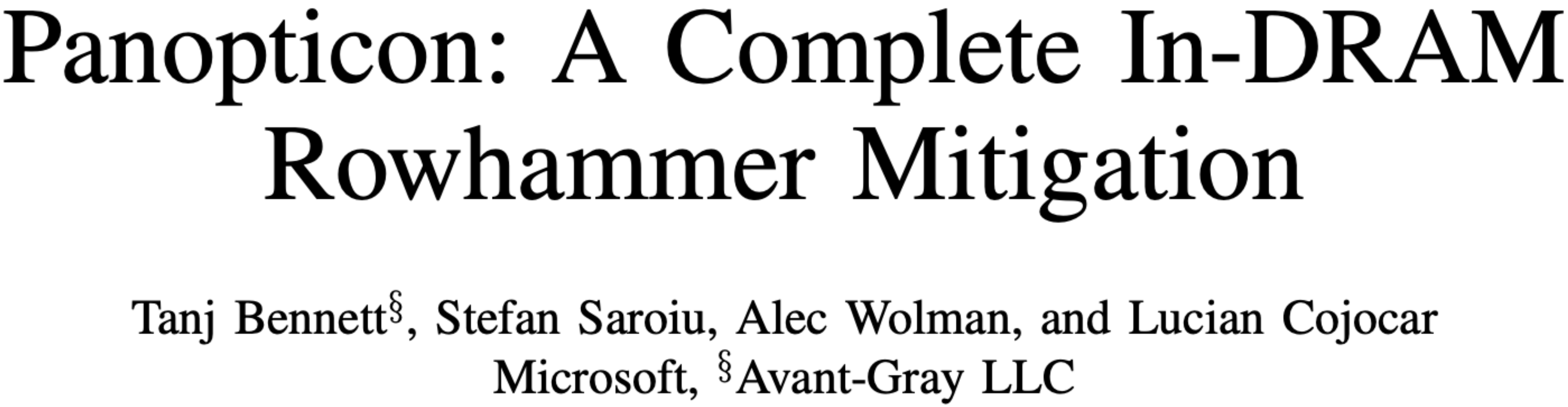
DSAC: Low-Cost Rowhammer Mitigation Using In-DRAM Stochastic and Approximate Counting Algorithm

Seungki Hong Dongha Kim Jaehyung Lee Reum Oh
Changsik Yoo Sangjoon Hwang Jooyoung Lee

DRAM Design Team, Memory Division, Samsung Electronics

https://arxiv.org/pdf/2302.03591v1.pdf

A Solution from Microsoft

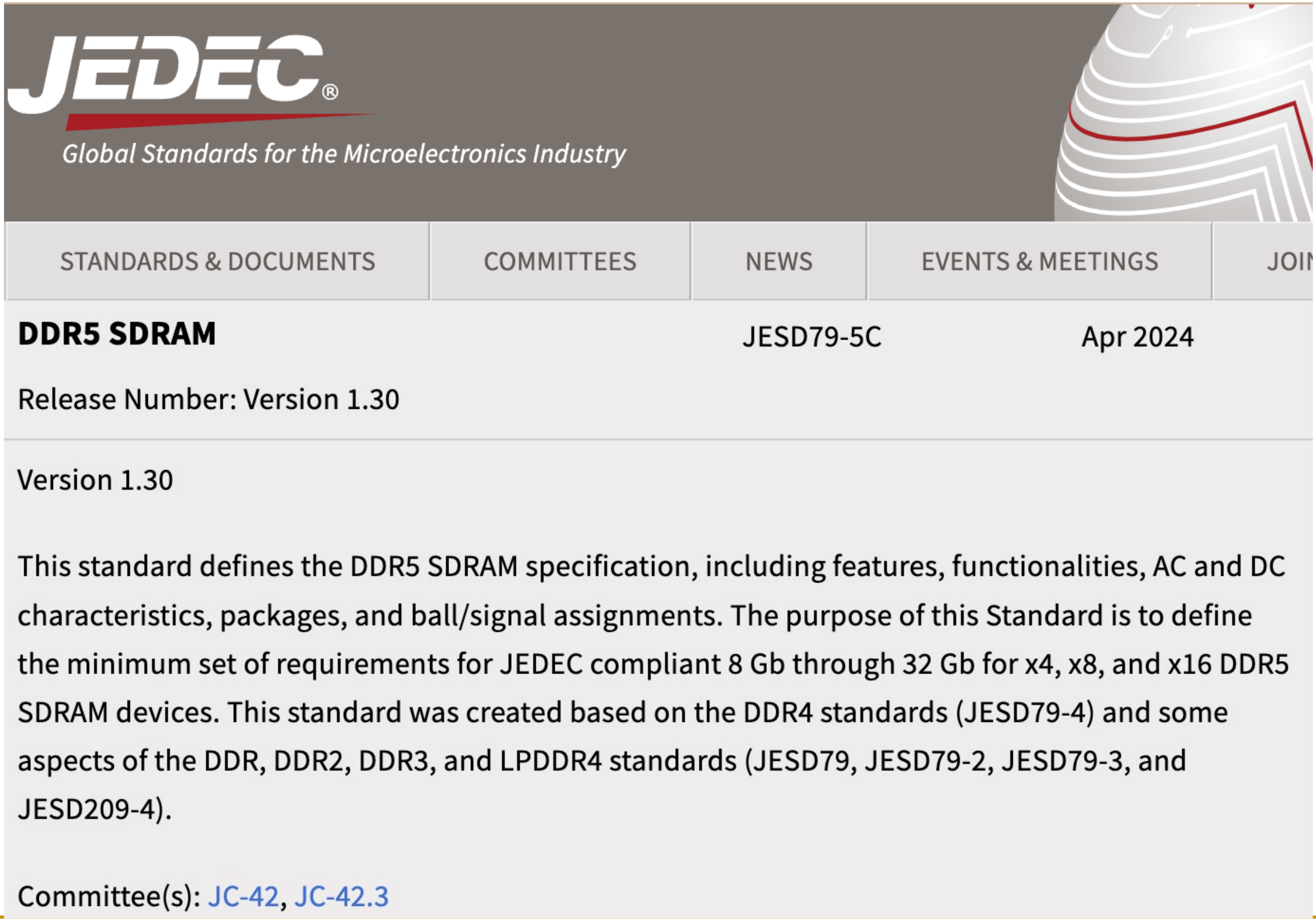


Panopticon: A Complete In-DRAM Rowhammer Mitigation

Tanj Bennett[§], Stefan Saroiu, Alec Wolman, and Lucian Cojocar
Microsoft, [§]Avant-Gray LLC

https://stefan.t8k2.com/publications/dramsec/2021/panopticon.pdf

Recent Improvements in JEDEC (2024)



DDR5 SDRAM JESD79-5C Apr 2024

Release Number: Version 1.30

Version 1.30

This standard defines the DDR5 SDRAM specification, including features, functionalities, AC and DC characteristics, packages, and ball/signal assignments. The purpose of this Standard is to define the minimum set of requirements for JEDEC compliant 8 Gb through 32 Gb for x4, x8, and x16 DDR5 SDRAM devices. This standard was created based on the DDR4 standards (JESD79-4) and some aspects of the DDR, DDR2, DDR3, and LPDDR4 standards (JESD79, JESD79-2, JESD79-3, and JESD209-4).

Committee(s): JC-42, JC-42.3

Evaluation of Industry’s Recent Solutions

- **Appears at DRAMSec 2024**

Understanding the Security Benefits and Overheads of Emerging Industry Solutions to DRAM Read Disturbance

Oğuzhan Canpolat[§][†] A. Giray Yağlıkçı[§] Geraldo F. Oliveira[§] Ataberk Olgun[§]
Oğuz Ergin[†] Onur Mutlu[§]
[§]*ETH Zürich* [†]*TOBB University of Economics and Technology*

https://arxiv.org/pdf/2406.19094

https://github.com/CMU-SAFARI/ramulator2

Are Solutions Good?

https://www.reddit.com/r/CrappyDesign/comments/arw0q8/now_this_this_is_poor_fencing/

Question

Are we now

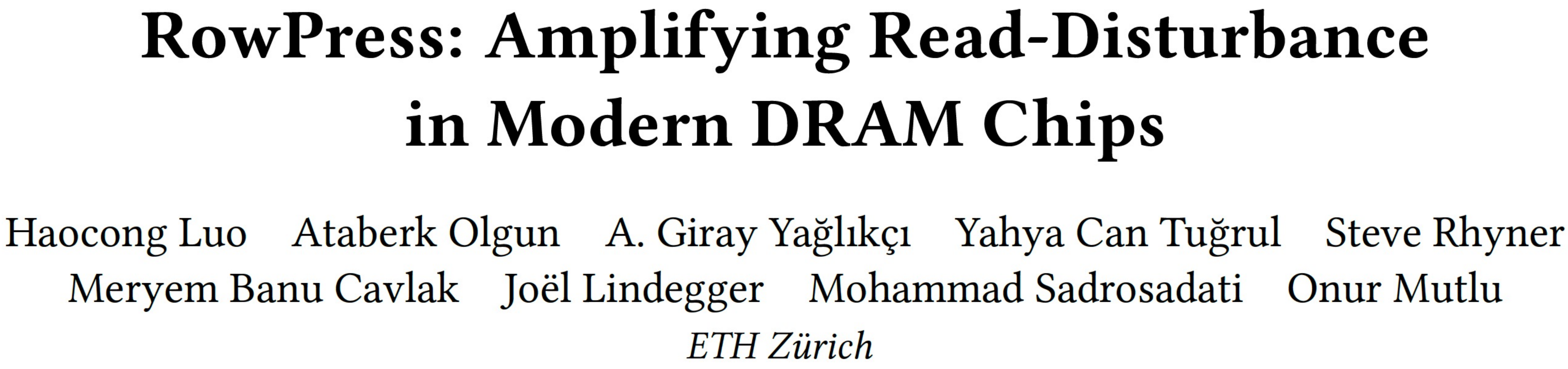
BitFlip-free

in 2024 and Beyond?

Are We Now BitFlip Free?

- **Appears at ISCA 2023**

What if there is another phenomenon that
does NOT require high row activation count?



RowPress: Amplifying Read-Disturbance in Modern DRAM Chips

Haocong Luo Ataberk Olgun A. Giray Yağlıkçı Yahya Can Tuğrul Steve Rhyner
Meryem Banu Cavlak Joël Lindegger Mohammad Sadrosadati Onur Mutlu
ETH Zürich

RowPress [ISCA 2023]

- Haocong Luo, Ataberk Olgun, Giray Yaglikci, Yahya Can Tugrul, Steve Rhyner, M. Banu Cavlak, Joel Lindegger, Mohammad Sadrosadati, and Onur Mutlu,
"RowPress: Amplifying Read Disturbance in Modern DRAM Chips"
*Proceedings of the 50th International Symposium on Computer Architecture (**ISCA**)*, Orlando, FL, USA, June 2023.
[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Lightning Talk Video (3 minutes)]
[RowPress Source Code and Datasets (Officially Artifact Evaluated with All Badges)]
Officially artifact evaluated as available, reusable and reproducible.
Best artifact award at ISCA 2023. IEEE Micro Top Pick in 2024.

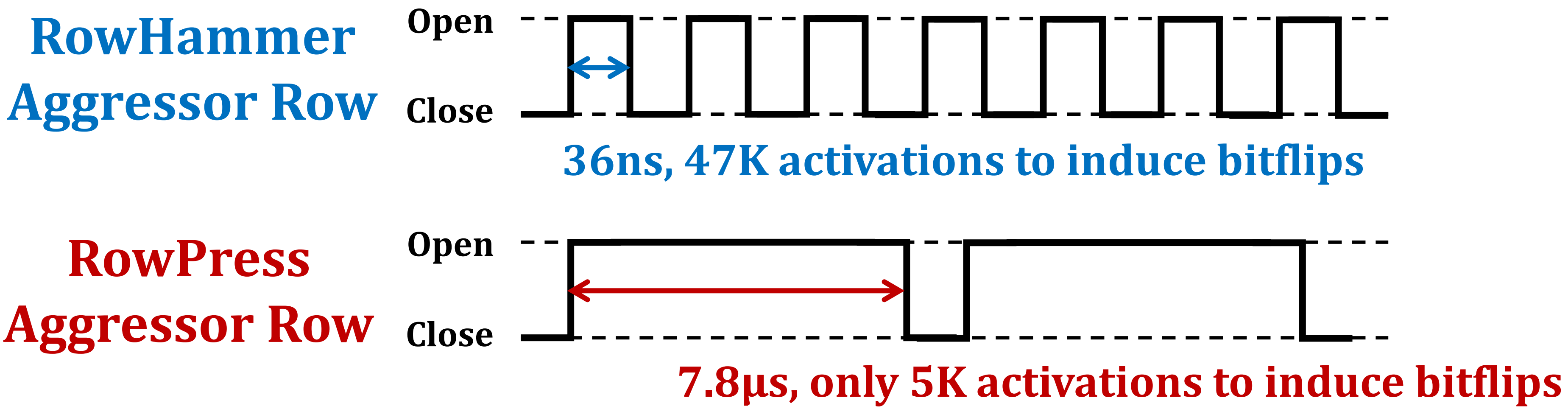
RowPress: Amplifying Read-Disturbance in Modern DRAM Chips

Haocong Luo   Ataberk Olgun   A. Giray Yağlıkçı   Yahya Can Tuğrul   Steve Rhyner
Meryem Banu Cavlak   Joël Lindegger   Mohammad Sadrosadati   Onur Mutlu
ETH Zürich

RowPress vs. RowHammer

Instead of using a high activation count,

☞ increase the time that the aggressor row stays open

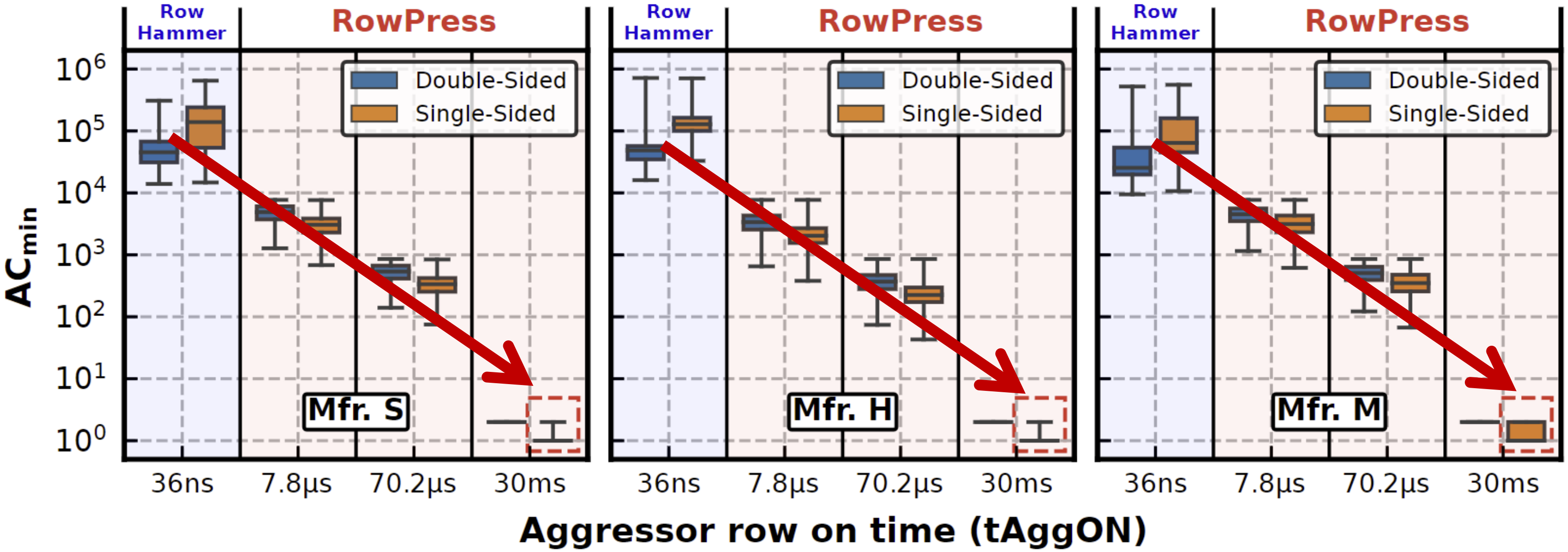

RowHammer Aggressor Row
Open
Close
36ns, 47K activations to induce bitflips
RowPress Aggressor Row
Open
Close
7.8µs, only 5K activations to induce bitflips

We observe bitflips even with **ONLY ONE activation** in extreme cases where the row stays open for 30ms

Key Characteristics of RowPress (I)

RowPress Amplifies Read Disturbance in DRAM

- Reduces the minimum number of row activations needed to induce a bitflip (**ACmin**) by **1-2 orders of magnitude**
- In extreme cases, activating a row **only once** induces bitflips



SAFARI

Real-System Demonstration (I)

Intel Core i5-10400
(Comet Lake)

Samsung DDR4 Module
M378A2K43CB1-CTD
(Date Code: 20-10)
w/ TRR RowHammer Mitigation

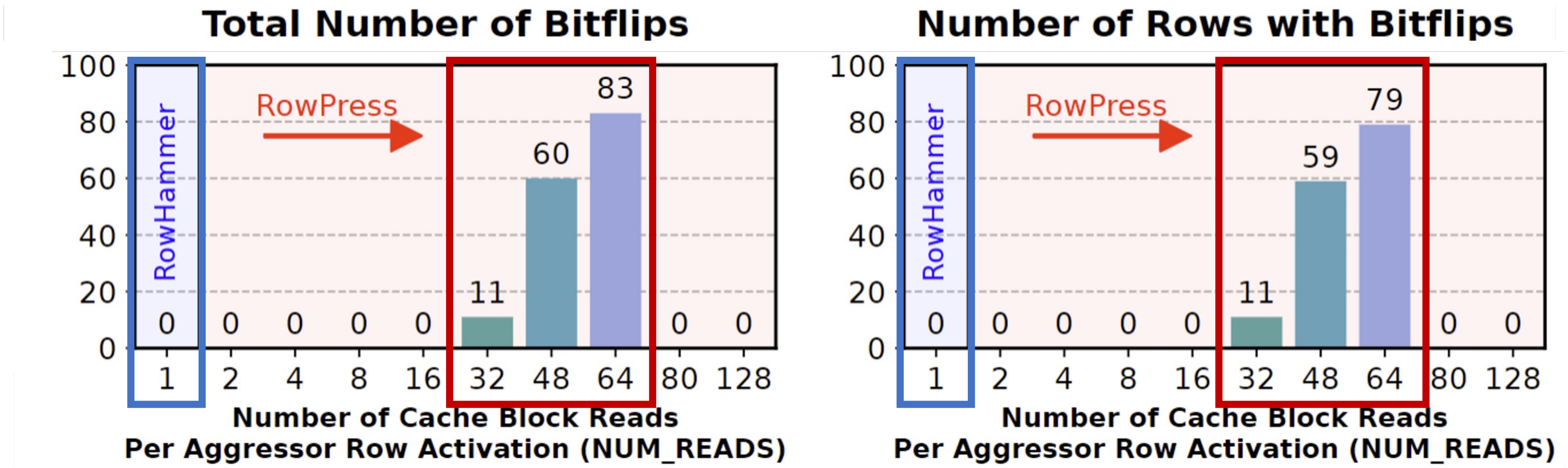
Key Idea: A proof-of-concept RowPress program keeps a DRAM row open for a longer period by **keeping on accessing different cache blocks in the row**

```
// Sync with Refresh and Loop Below
  for (k = 0; k < NUM_AGGR_ACTS; k++)
    for (j = 0; j < NUM_READS; j++) *AGGRESSOR1[j];
    for (j = 0; j < NUM_READS; j++) *AGGRESSOR2[j];
    for (j = 0; j < NUM_READS; j++)
      clflushopt(AGGRESSOR1[j]);
      clflushopt(AGGRESSOR2[j]);
    mfence();
  activate_dummy_rows();
```

Number of Cache Blocks Accessed Per Aggressor Row ACT (NUM_READS=1 is Rowhammer)

Real-System Demonstration (II)

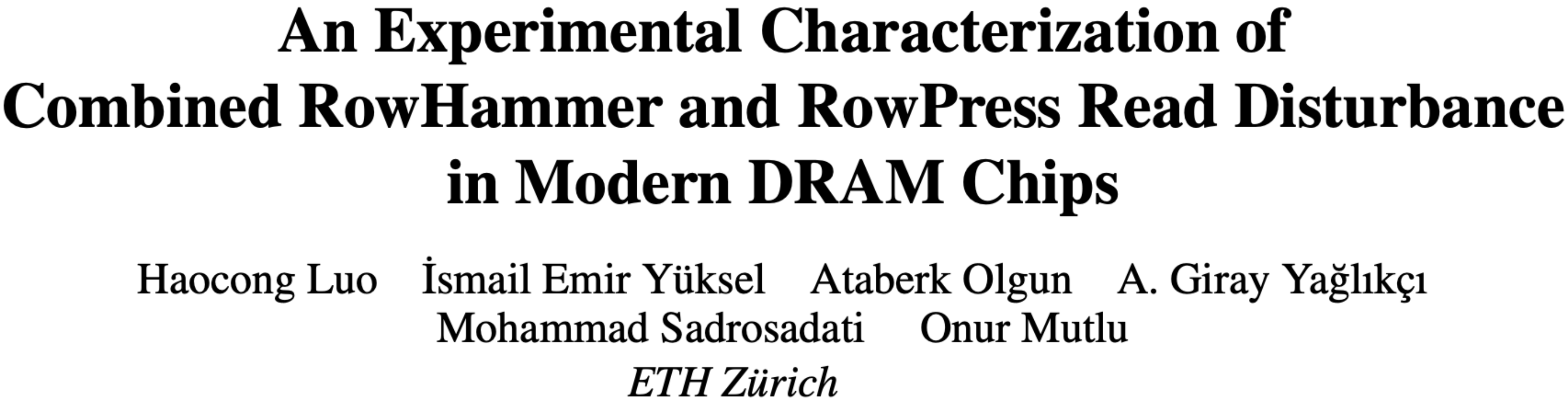
On 1500 victim rows



Leveraging RowPress, our user-level program induces bitflips when RowHammer cannot

Combining RowHammer and RowPress

- **Appears at DSN Disrupt 2024**



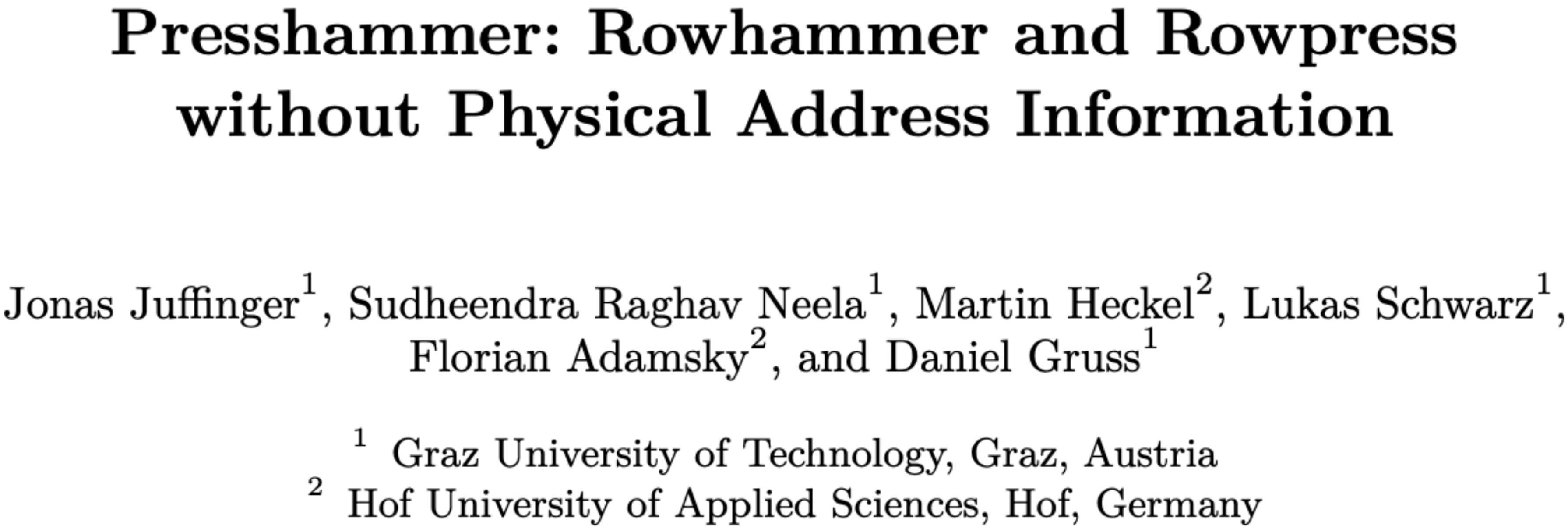
An Experimental Characterization of Combined RowHammer and RowPress Read Disturbance in Modern DRAM Chips

Haocong Luo İsmail Emir Yüksel Ataberk Olgun A. Giray Yağlıkçı
Mohammad Sadrosadati Onur Mutlu
ETH Zürich

https://arxiv.org/pdf/2406.13080v2

Combining RowHammer and RowPress

- **Appears at DIMVA 2024**



Presshammer: Rowhammer and Rowpress without Physical Address Information

Jonas Juffinger[1], Sudheendra Raghav Neela[1], Martin Heckel[2], Lukas Schwarz[1], Florian Adamsky[2], and Daniel Gruss[1]

[1] Graz University of Technology, Graz, Austria
[2] Hof University of Applied Sciences, Hof, Germany

https://www.jonasjuffinger.com/papers/presshammer.pdf

Understanding RowPress

- **Appears in IEEE TED, 2024**

IEEE TRANSACTIONS ON ELECTRON DEVICES

Unveiling RowPress in Sub-20 nm DRAM Through Comparative Analysis With Row Hammer: From Leakage Mechanisms to Key Features

Longda Zhou, Sheng Ye, Runsheng Wang, *Member, IEEE*, and Zhigang Ji

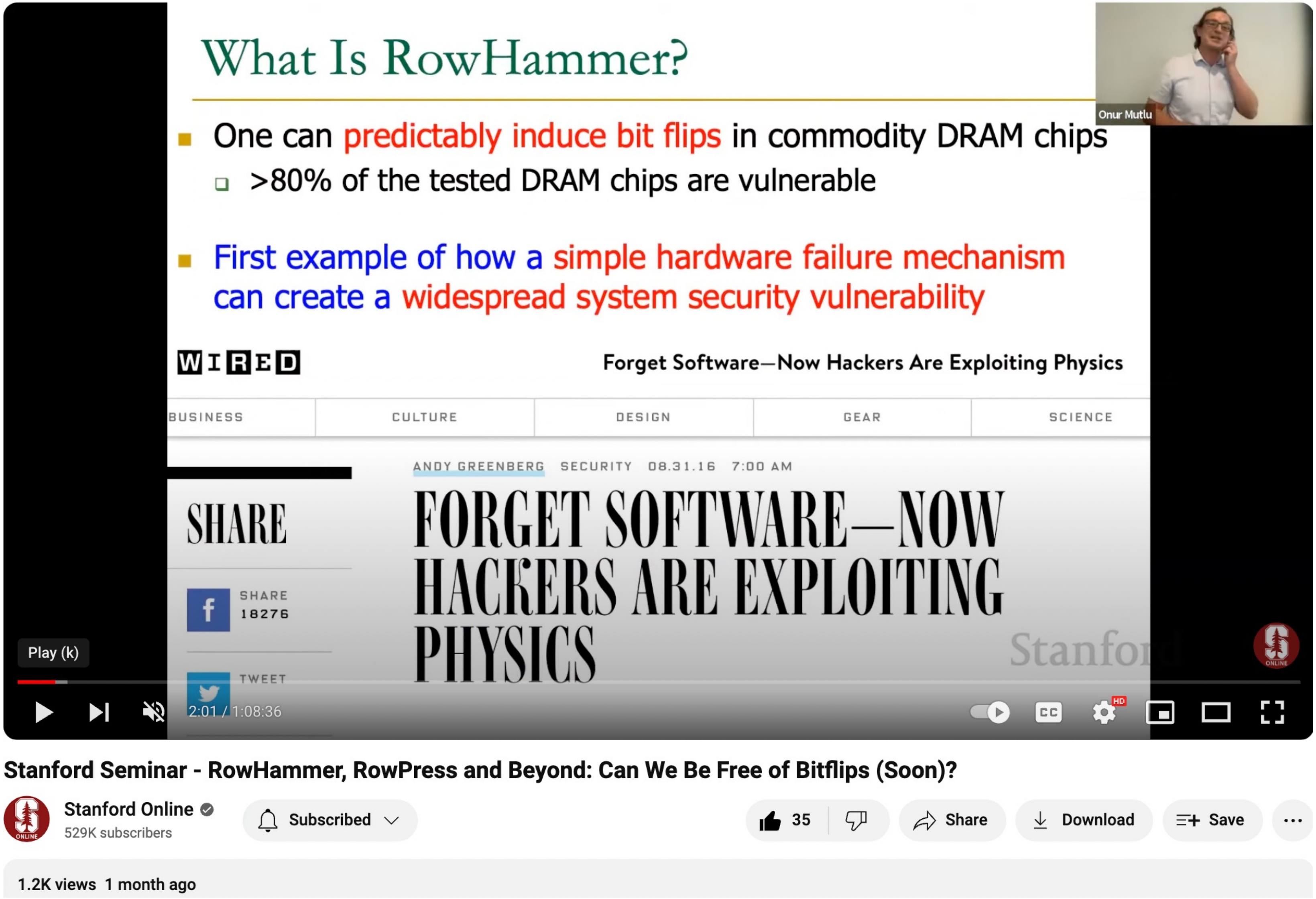
Key Takeaways

Read disturbance is a technology scaling problem

Finding a good solution to read disturbance is difficult (and will become more so)

More to Come…

A Recent RowHammer Lecture



What Is RowHammer?

- One can predictably induce bit flips in commodity DRAM chips
  - >80% of the tested DRAM chips are vulnerable
- First example of how a simple hardware failure mechanism can create a widespread system security vulnerability

WIRED

Forget Software—Now Hackers Are Exploiting Physics

ANDY GREENBERG SECURITY 08.31.16 7:00 AM

FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

Stanford Seminar - RowHammer, RowPress and Beyond: Can We Be Free of Bitflips (Soon)?

Stanford Online

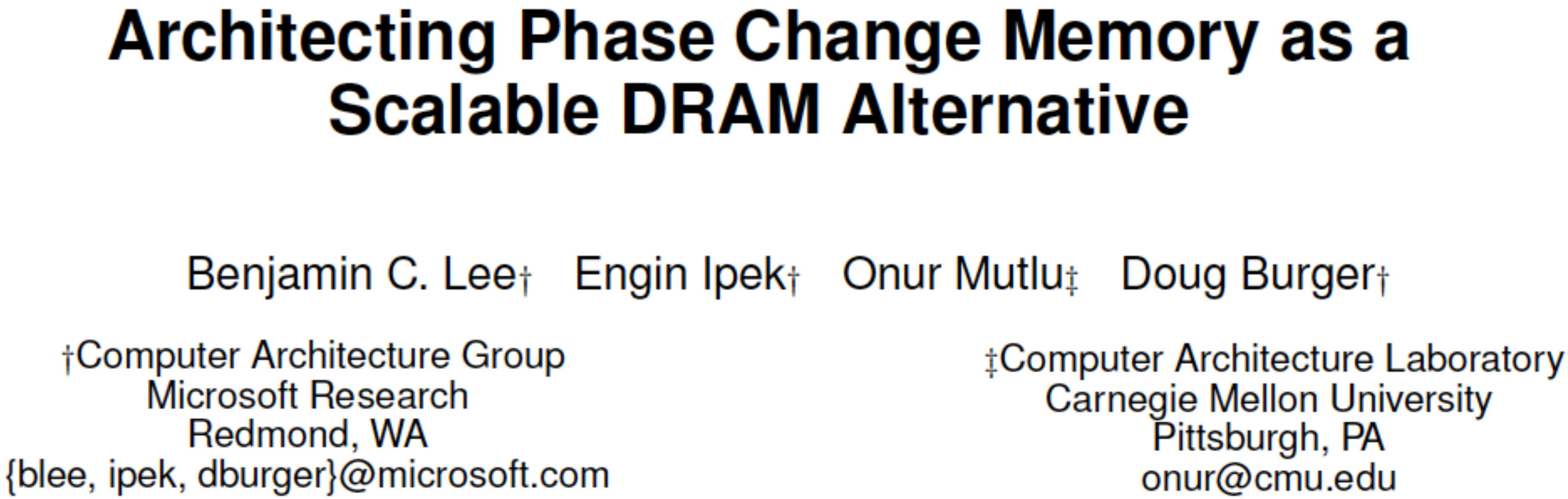
529K subscribers

1.2K views 1 month ago

https://www.youtube.com/watch?v=0W7YRRhnunw

Emerging Memories Also Need Intelligent Controllers

- Benjamin C. Lee, Engin Ipek, Onur Mutlu, and Doug Burger, **"Architecting Phase Change Memory as a Scalable DRAM Alternative"** *Proceedings of the 36th International Symposium on Computer Architecture* (***ISCA***), pages 2-13, Austin, TX, June 2009. Slides (pdf)
One of the 13 computer architecture papers of 2009 selected as Top Picks by IEEE Micro. Selected as a CACM Research Highlight.
2022 Persistent Impact Prize.



Architecting Phase Change Memory as a Scalable DRAM Alternative

Benjamin C. Lee† Engin Ipek† Onur Mutlu‡ Doug Burger†

†Computer Architecture Group
Microsoft Research
Redmond, WA
{blee, ipek, dburger}@microsoft.com

‡Computer Architecture Laboratory
Carnegie Mellon University
Pittsburgh, PA
onur@cmu.edu

The Takeaway

Intelligent

Memory Controllers

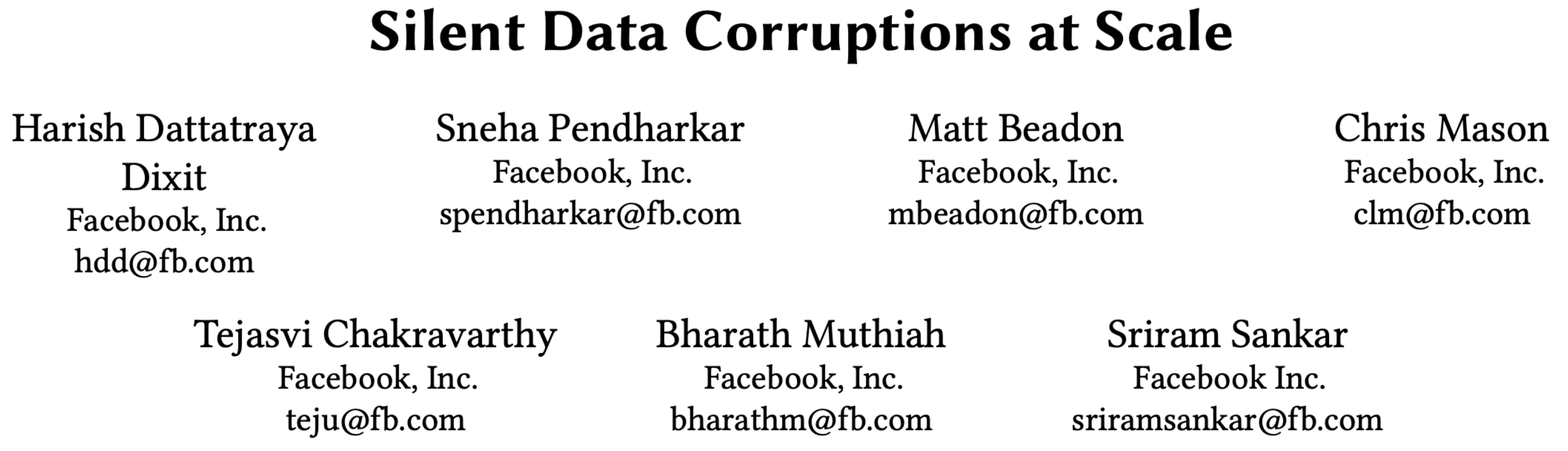
Can Enhance Security

& Enable Better Scaling

Data Corruption is in CPU Logic, Too

- Intermittent defects can cause silent data corruption
- They may be hard to detect or replicate
- They may be exploitable

Silent Data Corruption in Logic (2021)



Silent Data Corruptions at Scale

Harish Dattatraya Dixit
Facebook, Inc.
hdd@fb.com

Sneha Pendharkar
Facebook, Inc.
spendharkar@fb.com

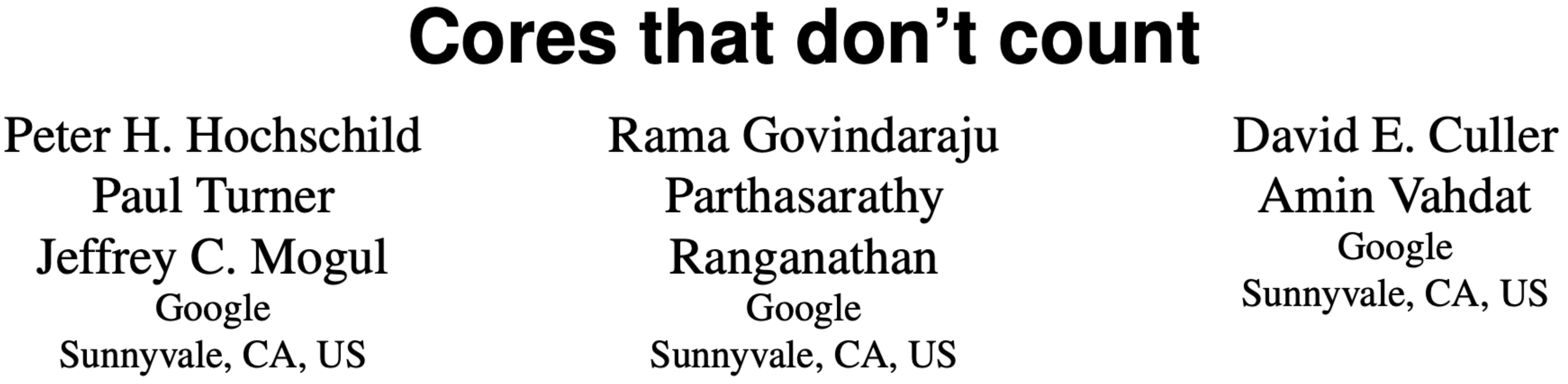
Matt Beadon
Facebook, Inc.
mbeadon@fb.com

Chris Mason
Facebook, Inc.
clm@fb.com

Tejasvi Chakravarthy
Facebook, Inc.
teju@fb.com

Bharath Muthiah
Facebook, Inc.
bharathm@fb.com

Sriram Sankar
Facebook Inc.
sriramsankar@fb.com



Cores that don't count

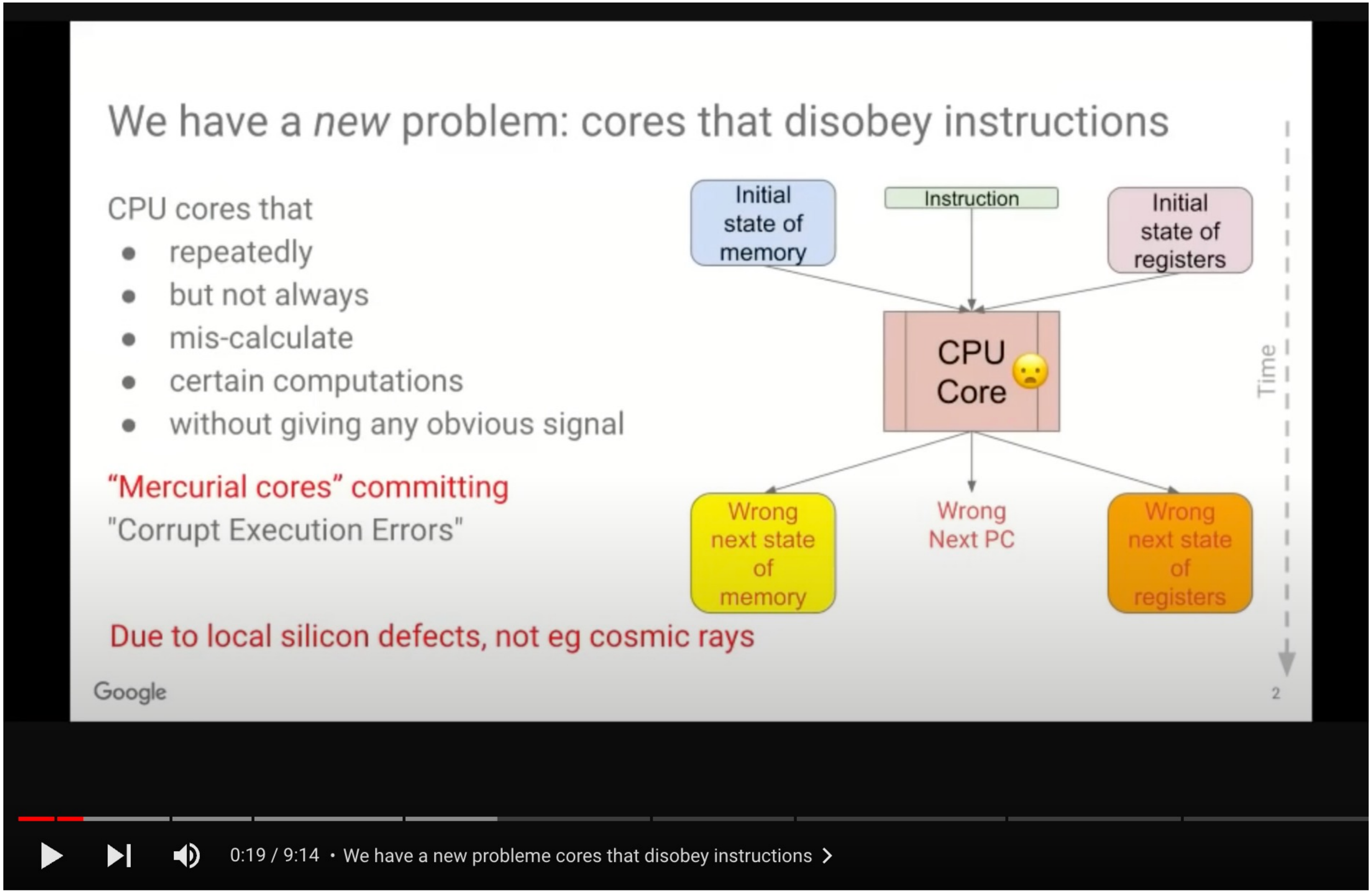
Peter H. Hochschild
Paul Turner
Jeffrey C. Mogul
Google
Sunnyvale, CA, US

Rama Govindaraju
Parthasarathy
Ranganathan
Google
Sunnyvale, CA, US

David E. Culler
Amin Vahdat
Google
Sunnyvale, CA, US

https://www.youtube.com/watch?v=QMF3rqhjYuM

Silent Data Corruption In-the-Field (2021)



We have a *new* problem: cores that disobey instructions

CPU cores that

- repeatedly
- but not always
- mis-calculate
- certain computations
- without giving any obvious signal

“Mercurial cores” committing

"Corrupt Execution Errors"

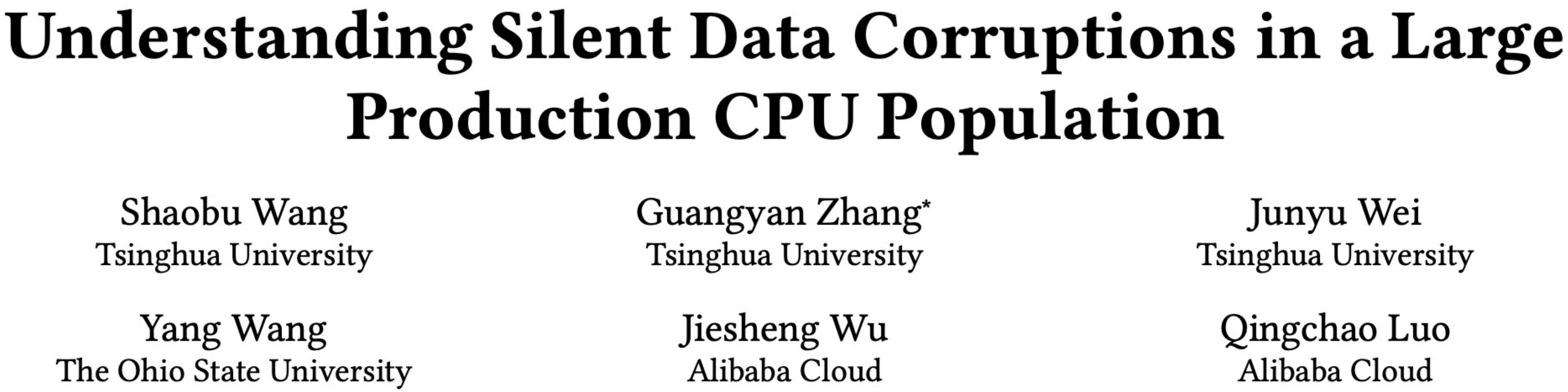
Due to local silicon defects, not eg cosmic rays

Initial state of memory
Instruction
Initial state of registers
CPU Core
Wrong next state of memory
Wrong Next PC
Wrong next state of registers
Time
Google
2
0:19 / 9:14 • We have a new probleme cores that disobey instructions

HotOS 2021: Cores That Don’t Count (Fun Hardware)

https://www.youtube.com/watch?v=QMF3rqhjYuM

Silent Data Corruption in Logic (2023)



Understanding Silent Data Corruptions in a Large Production CPU Population

Shaobu Wang
Tsinghua University

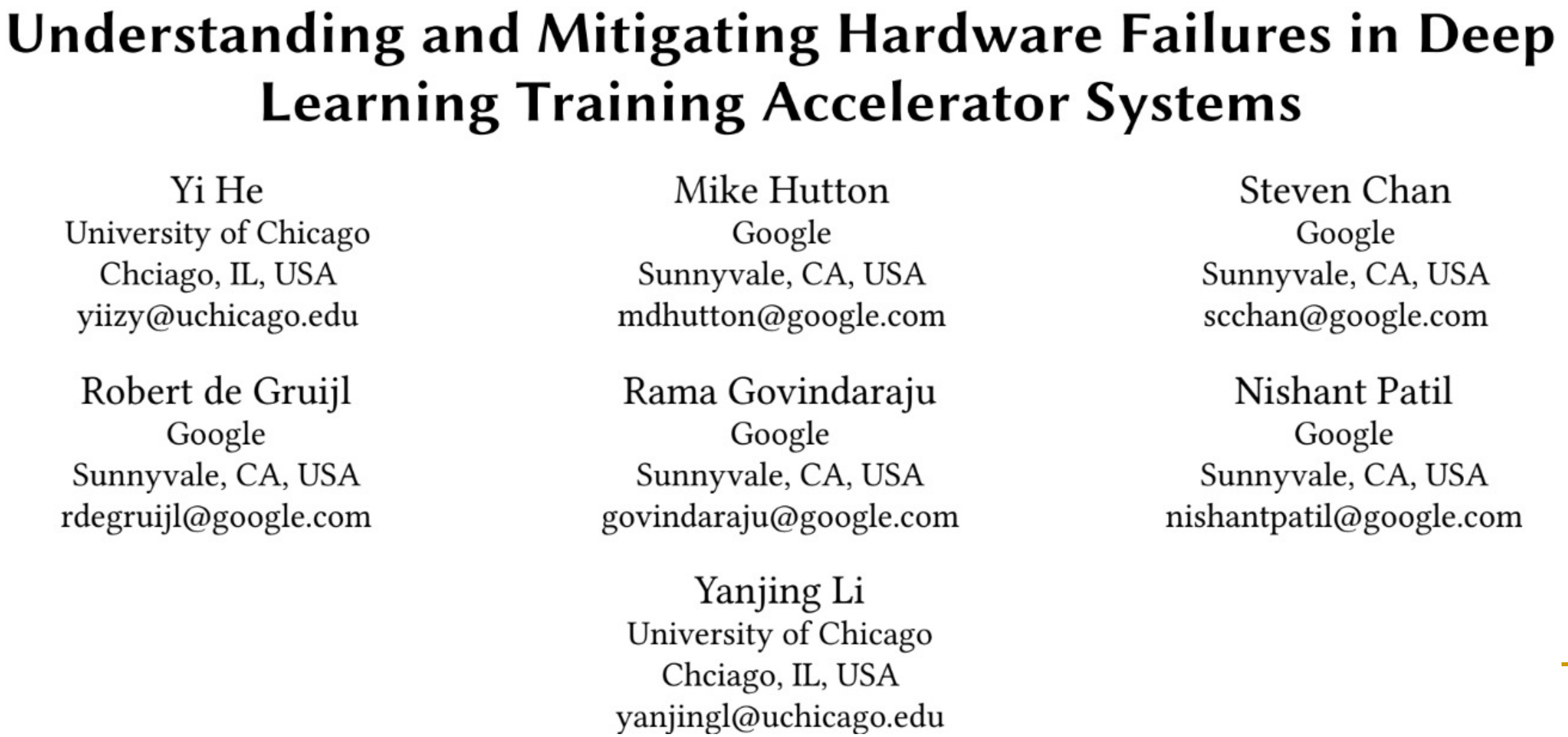
Guangyan Zhang*
Tsinghua University

Junyu Wei
Tsinghua University

Yang Wang
The Ohio State University

Jiesheng Wu
Alibaba Cloud

Qingchao Luo
Alibaba Cloud



Understanding and Mitigating Hardware Failures in Deep Learning Training Accelerator Systems

Yi He
University of Chicago
Chciago, IL, USA
yiizy@uchicago.edu

Mike Hutton
Google
Sunnyvale, CA, USA
mdhutton@google.com

Steven Chan
Google
Sunnyvale, CA, USA
scchan@google.com

Robert de Gruijl
Google
Sunnyvale, CA, USA
rdegruijl@google.com

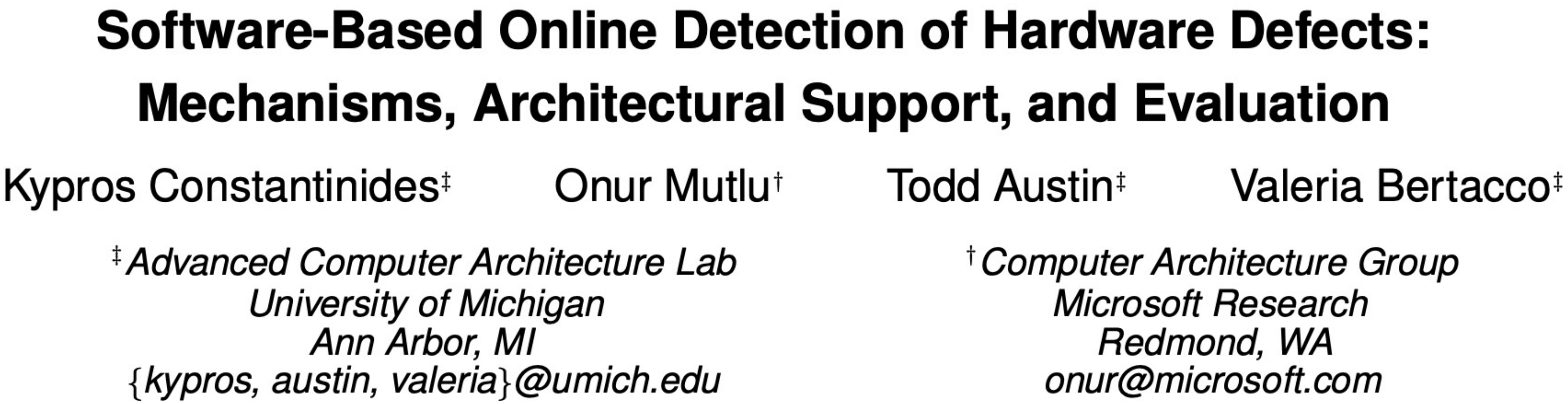
Rama Govindaraju
Google
Sunnyvale, CA, USA
govindaraju@google.com

Nishant Patil
Google
Sunnyvale, CA, USA
nishantpatil@google.com

Yanjing Li
University of Chicago
Chciago, IL, USA
yanjingl@uchicago.edu

How to Detect Hardware Errors? (I)

- Kypros Constantinides, Onur Mutlu, Todd Austin, and Valeria Bertacco, **"Software-Based Online Detection of Hardware Defects: Mechanisms, Architectural Support, and Evaluation"** *Proceedings of the 40th International Symposium on Microarchitecture* (***MICRO***), pages 97-108, Chicago, IL, December 2007. Slides (ppt)



Software-Based Online Detection of Hardware Defects: Mechanisms, Architectural Support, and Evaluation

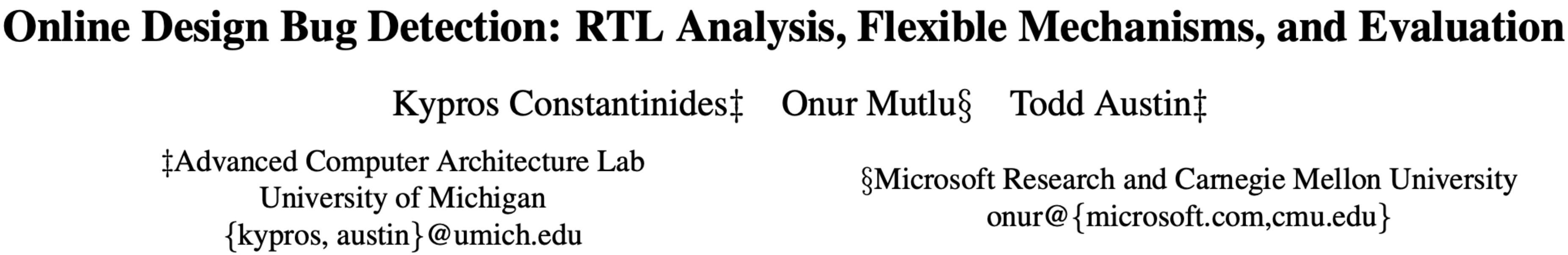
Kypros Constantinides[‡] Onur Mutlu[†] Todd Austin[‡] Valeria Bertacco[‡]

[‡] *Advanced Computer Architecture Lab*
University of Michigan
Ann Arbor, MI
{kypros, austin, valeria}@umich.edu

[†] *Computer Architecture Group*
Microsoft Research
Redmond, WA
onur@microsoft.com

How to Detect Hardware Errors? (II)

- Kypros Constantinides, Onur Mutlu, and Todd Austin, **"Online Design Bug Detection: RTL Analysis, Flexible Mechanisms, and Evaluation"** *Proceedings of the 41st International Symposium on Microarchitecture* (***MICRO***), pages 282-293, Lake Como, Italy, November 2008. Slides (ppt)



Online Design Bug Detection: RTL Analysis, Flexible Mechanisms, and Evaluation

Kypros Constantinides‡ Onur Mutlu§ Todd Austin‡

‡Advanced Computer Architecture Lab
University of Michigan
{kypros, austin}@umich.edu

§Microsoft Research and Carnegie Mellon University
onur@{microsoft.com,cmu.edu}

How to Detect Hardware Errors? (III)

- Yanjing Li, Onur Mutlu, and Subhasish Mitra,
  "Operating System Scheduling for Efficient Online Self-Test in Robust Systems"
  Proceedings of the International Conference on Computer-Aided Design (***ICCAD***), pages 201-208, San Jose, CA, November 2009. Slides (ppt) (pdf)

Operating System Scheduling for Efficient Online Self-Test in Robust Systems

Yanjing Li
Stanford University

Onur Mutlu
Carnegie Mellon University

Subhasish Mitra
Stanford University

How to Detect Hardware Errors? (IV)

- Yanjing Li, Onur Mutlu, Donald S. Gardner, and Subhasish Mitra, **"Concurrent Autonomous Self-Test for Uncore Components in System-on-Chips"**
  Proceedings of the 28th IEEE VLSI Test Symposium (***VTS***), pages 232-237, Santa Cruz, CA, April 2010. Slides (ppt)
  Best paper award at VTS 2010.

Concurrent Autonomous Self-Test for Uncore Components in System-on-Chips

Yanjing Li
Stanford University

Onur Mutlu
Carnegie Mellon University

Donald S. Gardner
Intel Corporation

Subhasish Mitra
Stanford University

How to Detect Hardware Errors? (V)

- Kypros Constantinides, Onur Mutlu, Todd Austin, and Valeria Bertacco, **"A Flexible Software-Based Framework for Online Detection of Hardware Defects"**
  IEEE Transactions on Computers (***TC***), Vol. 58, No. 8, pages 1063-1079, August 2009.

A Flexible Software-Based Framework for Online Detection of Hardware Defects

Kypros Constantinides, *Student Member*, *IEEE*, Onur Mutlu, *Member*, *IEEE*, Todd Austin, *Member*, *IEEE*, and Valeria Bertacco, *Member*, *IEEE*

Takeaways

- Both memory and logic errors will become worse with technology scaling

- Hardware errors will create worse robustness problems

- **We cannot afford to ignore data corruption**

System Complexity

Complex Systems Cause Many Issues

- Many hardware components, complex components

- Harder to design & verify

- Harder to reason about operational behavior
  - Correctness, performance, energy, security, privacy, …

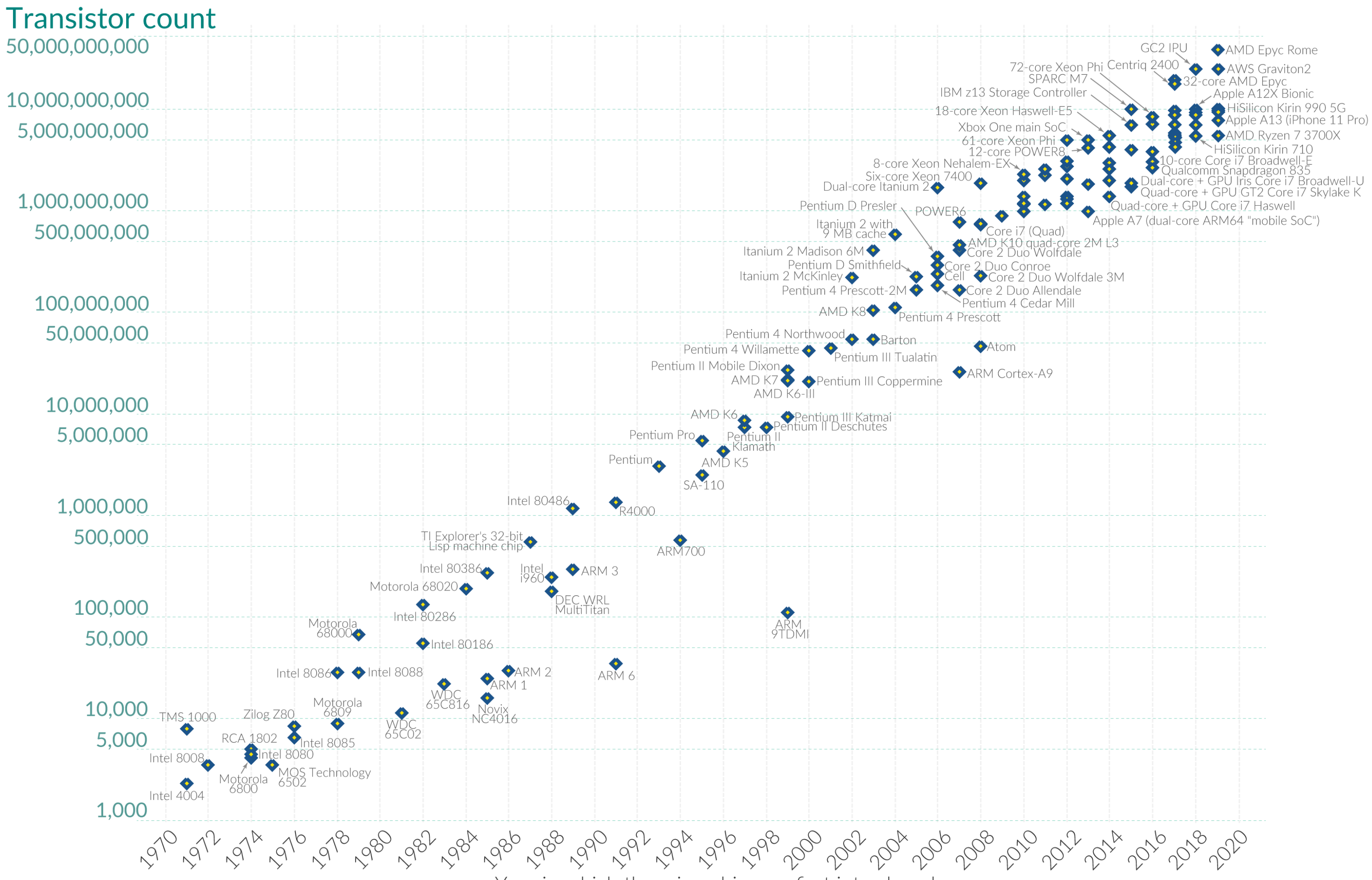
- Harder to control interactions between components and avoid information leakage

- Old methods do not keep up with new trends and complexity
  - Virtual memory a prime example, also coherence & verification

Processor Complexity Is Growing

Moore’s Law: The number of transistors on microchips doubles every two years

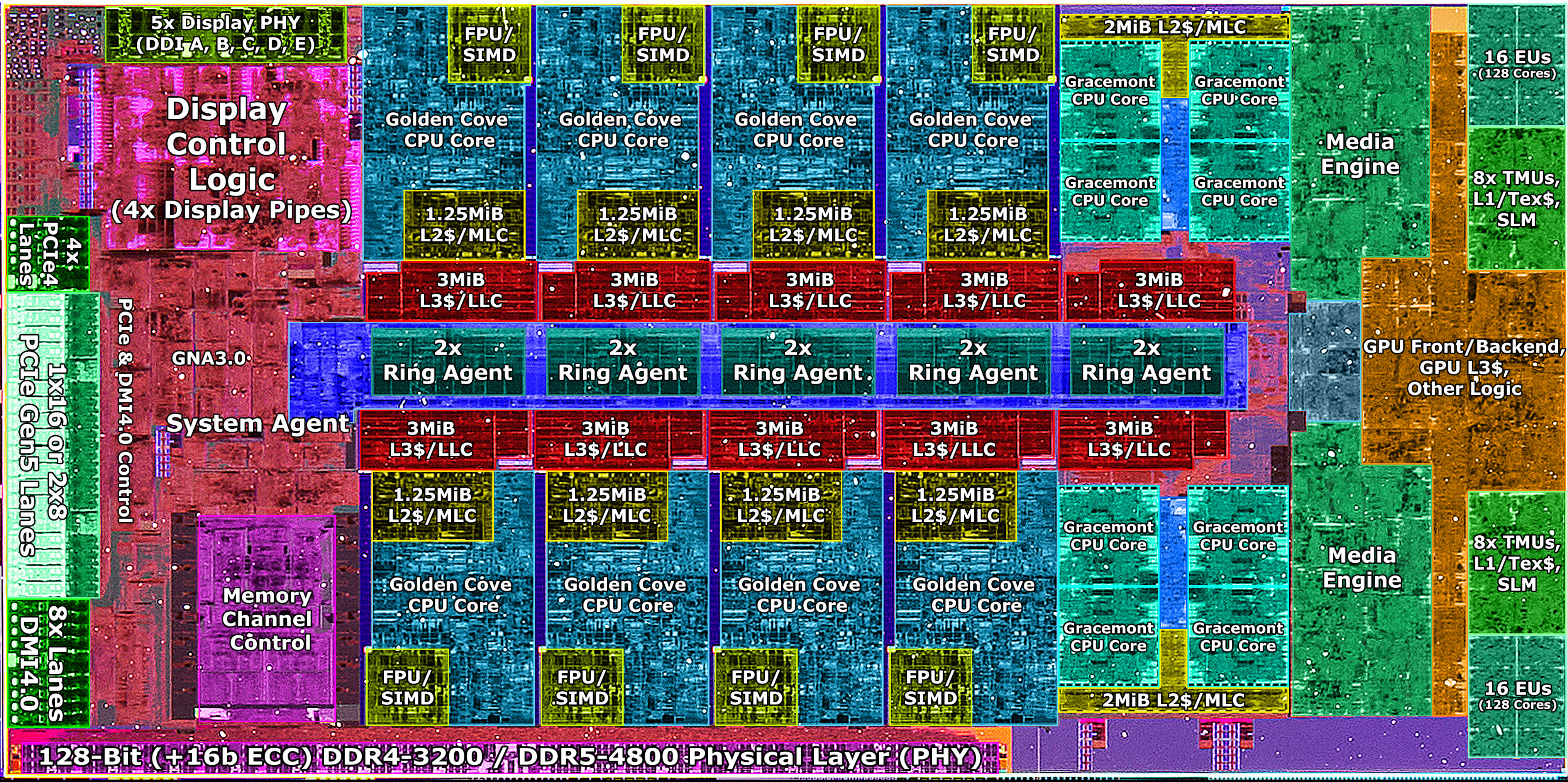
Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.



Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

OurWorldinData.org – Research and data to make progress against the world’s largest problems.

Complex CPUs and Memory Hierarchies

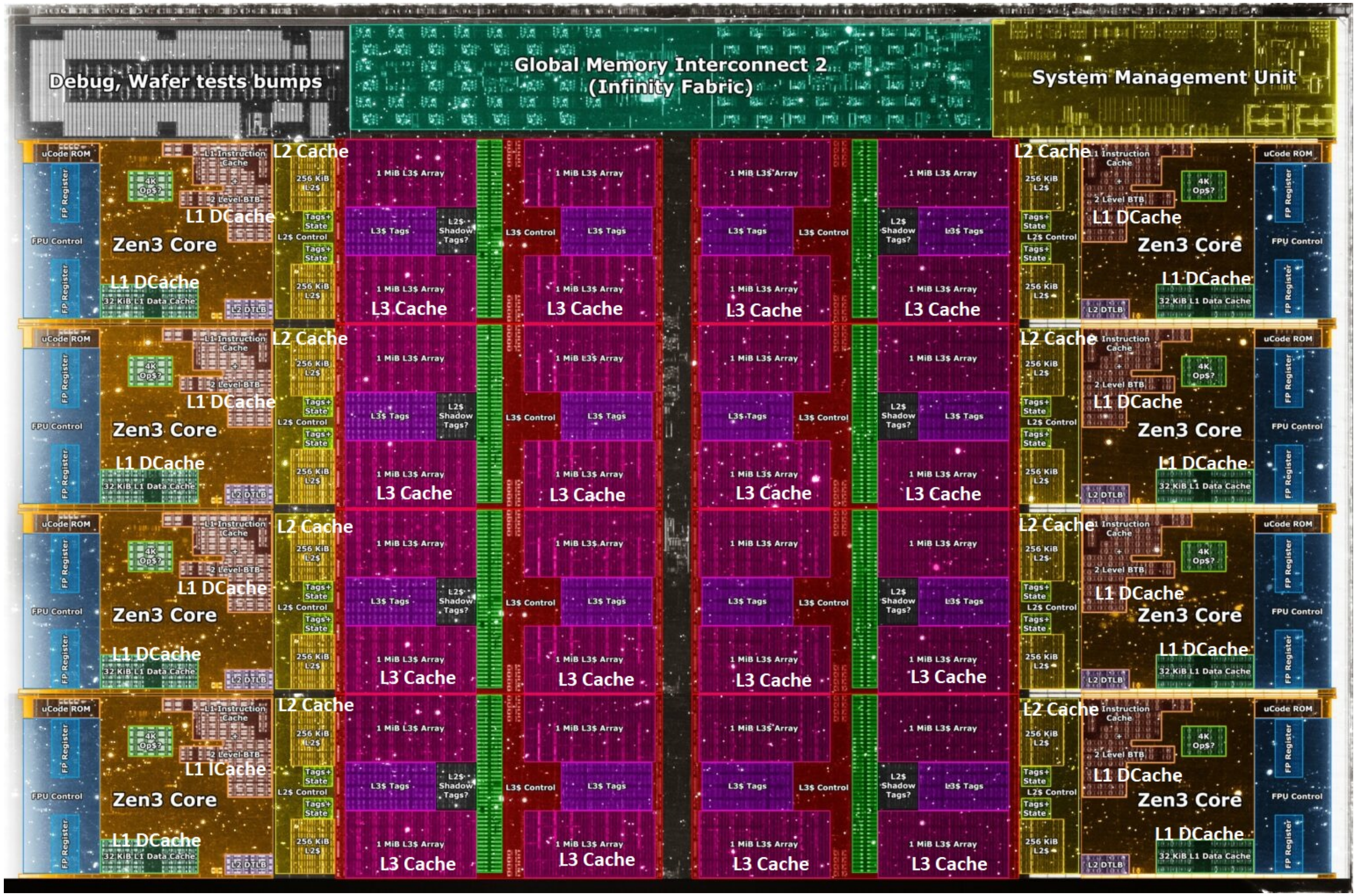


10nm ESF=Intel 7 Alder Lake die shot (~209mm²) from Intel: https://www.intel.com/content/www/us/en/newsroom/news/12th-gen-core-processors.html
Die shot interpretation by Locuza, October 2021

Intel Alder Lake, 2021

Source: https://twitter.com/Locuza_/status/1454152714930331652

Complex CPUs and Memory Hierarchies



Core Count:

8 cores/16 threads

L1 Caches:

32 KB per core

L2 Caches:

512 KB per core

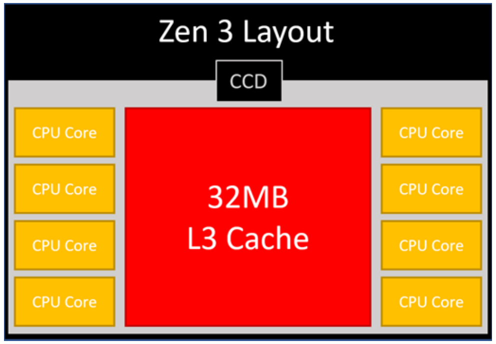
L3 Cache:

32 MB shared

AMD Ryzen 5000, 2020

https://wccftech.com/amd-ryzen-5000-zen-3-vermeer-undressed-high-res-die-shots-close-ups-pictured-detailed/

Complexity Growing with 3D (2021)

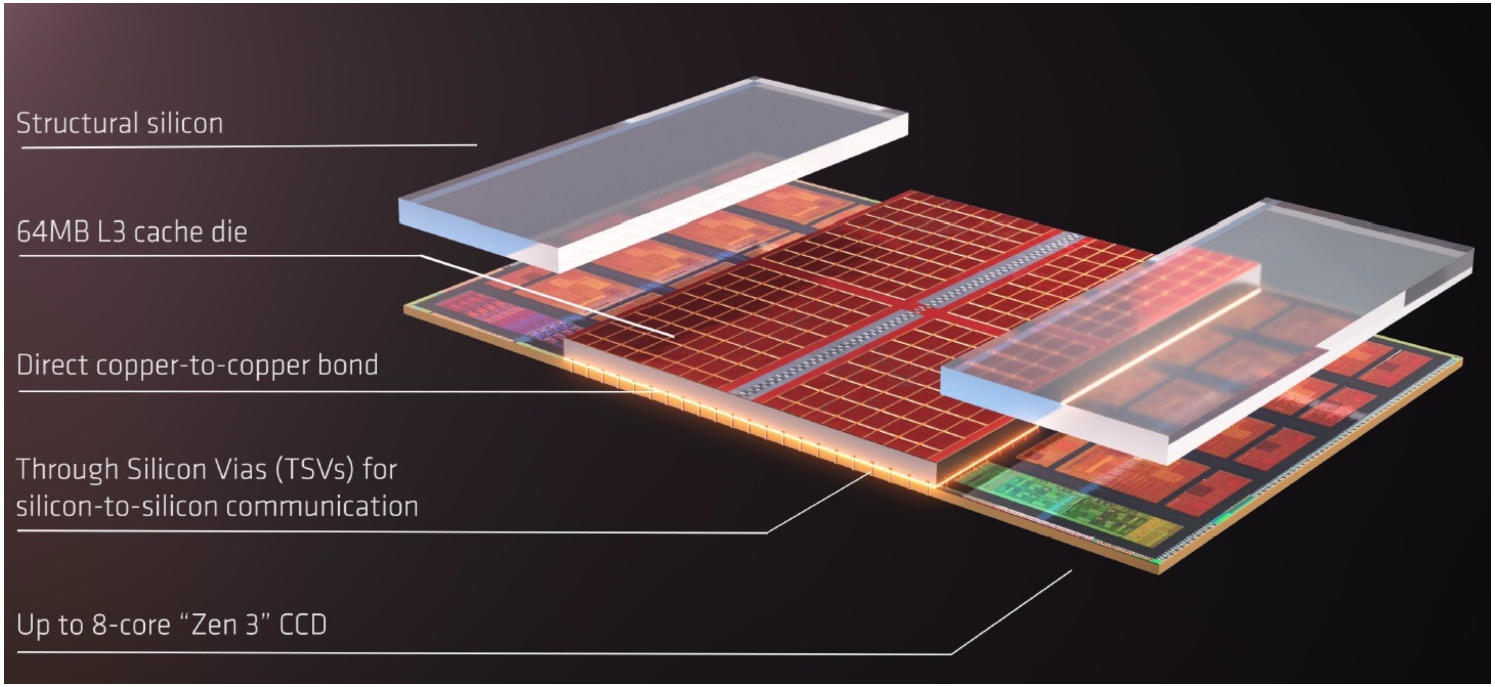


https://community.microcenter.com/discussion/5134/comparing-zen-3-to-zen-2

AMD increases the L3 size of their 8-core Zen 3 processors from 32 MB to 96 MB

Additional 64 MB L3 cache die stacked on top of the processor die

- Connected using Through Silicon Vias (TSVs)
- Total of 96 MB L3 cache


Structural silicon
64MB L3 cache die
Direct copper-to-copper bond
Through Silicon Vias (TSVs) for silicon-to-silicon communication
Up to 8-core "Zen 3" CCD

https://youtu.be/gqAYMx34euU
https://www.tech-critter.com/amd-keynote-computex-2021/

Processor Complexity and Features

- Leads to many (endless) side and covert channels
  - Spectre and Meltdown are prime recent examples
  - These will not go away

- Leads to many bugs and unintended behavior
  - Especially with new features or complex interactions
  - Some can be exploitable

- **How to tame processor complexity & resulting issues?**

Access Control & Protection Mechanisms

- Are based on virtual memory (VM), invented in 1950s

- VM has not changed much even after decades of technology scaling and memory system improvements

- VM causes large performance problems and is responsible for large complexity, power, energy
- VM is poor for fine-grained security and access control
- VM hinders innovation in heterogeneous (e.g., accelerator) systems and new architectures (e.g., processing near data)

- **It is time to rethink virtual memory**

Virtual Memory: Parting Thoughts

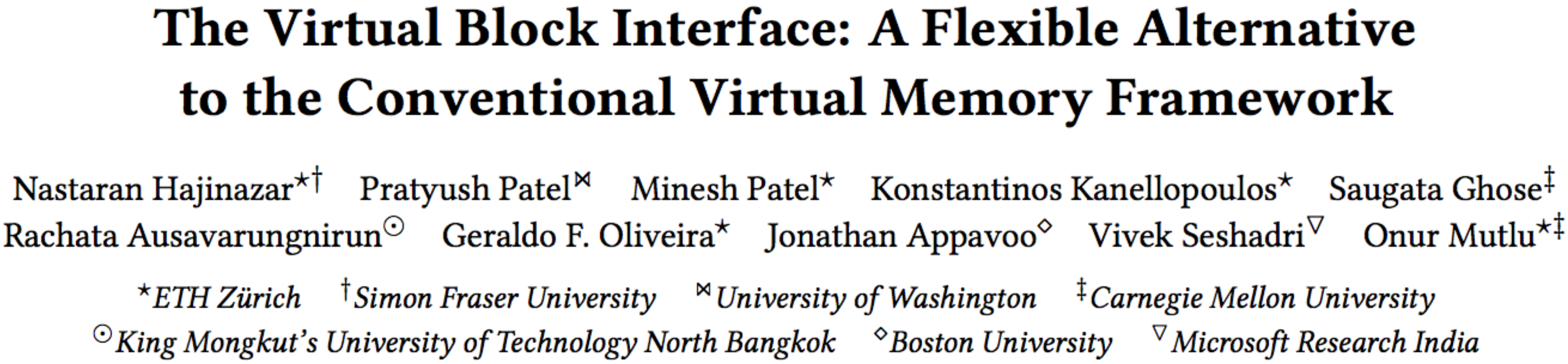
- Virtual Memory is one of the most successful examples of
  - architectural support for programmers
  - how to partition work between hardware and software
  - hardware/software cooperation
  - programmer/architect tradeoff

- Going forward: How does virtual memory fare and scale into the future? Five key trends:
  - Increasing, huge physical memory sizes (local & remote)
  - Hybrid physical memory systems (DRAM + NVM + SSD)
  - Many accelerators in the system accessing physical memory
  - Virtualized systems (hypervisors, software virtualization, local and remote memories)
  - Processing in memory systems – near-data accelerators

From my Digital Design & Comp Arch freshman class: https://www.youtube.com/watch?v=UaS0oqgWiIk

Rethinking Virtual Memory

Nastaran Hajinazar, Pratyush Patel, Minesh Patel, Konstantinos Kanellopoulos, Saugata Ghose, Rachata Ausavarungnirun, Geraldo Francisco de Oliveira Jr., Jonathan Appavoo, Vivek Seshadri, and Onur Mutlu,
"The Virtual Block Interface: A Flexible Alternative to the Conventional Virtual Memory Framework"
*Proceedings of the 47th International Symposium on Computer Architecture (**ISCA**)*, Virtual, June 2020.
[Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[ARM Research Summit Poster (pptx) (pdf)]
[Talk Video (26 minutes)]
[Lightning Talk Video (3 minutes)]
[Lecture Video (43 minutes)]



The Virtual Block Interface: A Flexible Alternative to the Conventional Virtual Memory Framework

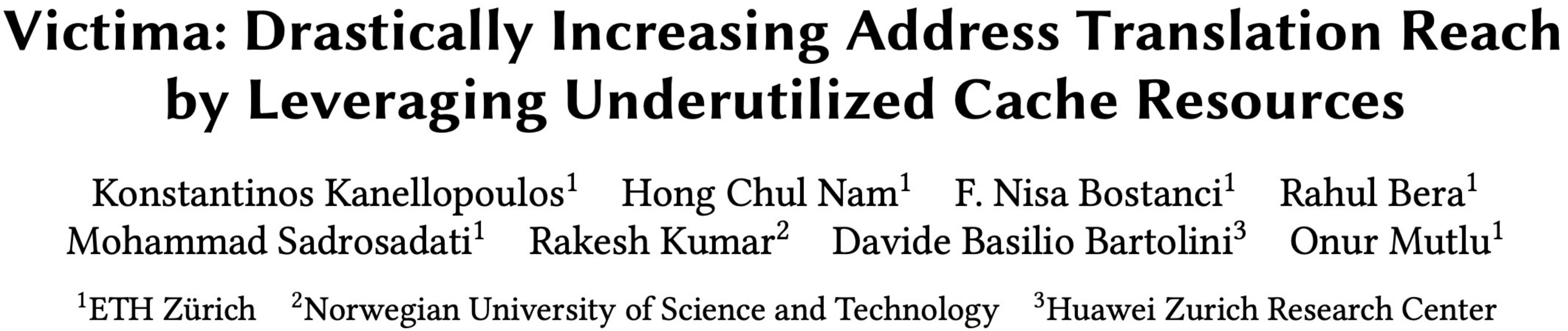
Nastaran Hajinazar[⋆†] Pratyush Patel[⋈] Minesh Patel[⋆] Konstantinos Kanellopoulos[⋆] Saugata Ghose[‡]
Rachata Ausavarungnirun[⊙] Geraldo F. Oliveira[⋆] Jonathan Appavoo[◇] Vivek Seshadri[▽] Onur Mutlu[⋆‡]

[⋆]*ETH Zürich* [†]*Simon Fraser University* [⋈]*University of Washington* [‡]*Carnegie Mellon University*
[⊙]*King Mongkut's University of Technology North Bangkok* [◇]*Boston University* [▽]*Microsoft Research India*

https://people.inf.ethz.ch/omutlu/pub/VBI-virtual-block-interface_isca20.pdf

Better Virtual Memory (I)

Konstantinos Kanellopoulos, Hong Chul Nam, F. Nisa Bostanci, Rahul Bera, Mohammad Sadrosadati, Rakesh Kumar, Davide Basilio Bartolini, and Onur Mutlu,
"Victima: Drastically Increasing Address Translation Reach by Leveraging Underutilized Cache Resources"
*Proceedings of the 56th International Symposium on Microarchitecture (**MICRO**)*, Toronto, ON, Canada, November 2023.
[Slides (pptx) (pdf)]
[arXiv version]
[Victima Source Code (Officially Artifact Evaluated with All Badges)]
Officially artifact evaluated as available, functional, reusable and reproducible.
Distinguished artifact award at MICRO 2023.



Victima: Drastically Increasing Address Translation Reach by Leveraging Underutilized Cache Resources

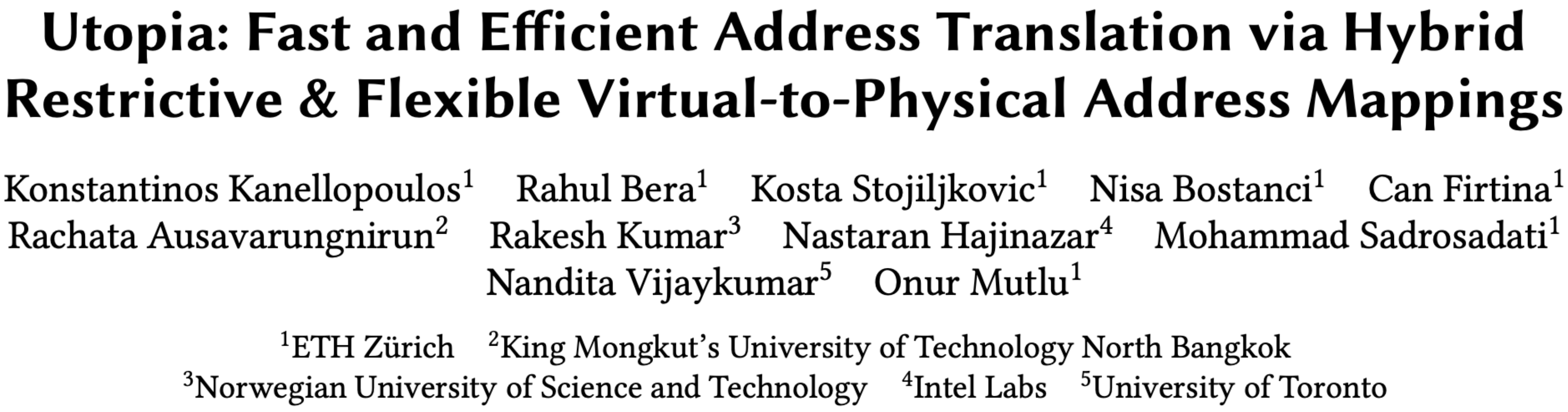
Konstantinos Kanellopoulos[1] Hong Chul Nam[1] F. Nisa Bostanci[1] Rahul Bera[1]
Mohammad Sadrosadati[1] Rakesh Kumar[2] Davide Basilio Bartolini[3] Onur Mutlu[1]

[1]ETH Zürich [2]Norwegian University of Science and Technology [3]Huawei Zurich Research Center

https://arxiv.org/pdf/2310.04158

Better Virtual Memory (II)

Konstantinos Kanellopoulos, Rahul Bera, Kosta Stojiljkovic, Nisa Bostanci, Can Firtina, Rachata Ausavarungnirun, Rakesh Kumar, Nastaran Hajinazar, Mohammad Sadrosadati, Nandita Vijaykumar, and Onur Mutlu,
"Utopia: Fast and Efficient Address Translation via Hybrid Restrictive & Flexible Virtual-to-Physical Address Mappings"
Proceedings of the 56th International Symposium on Microarchitecture (***MICRO***), Toronto, ON, Canada, November 2023.
[Slides (pptx) (pdf)]
[arXiv version]
[Utopia Source Code]



Utopia: Fast and Efficient Address Translation via Hybrid Restrictive & Flexible Virtual-to-Physical Address Mappings

Konstantinos Kanellopoulos[1] Rahul Bera[1] Kosta Stojiljkovic[1] Nisa Bostanci[1] Can Firtina[1]
Rachata Ausavarungnirun[2] Rakesh Kumar[3] Nastaran Hajinazar[4] Mohammad Sadrosadati[1]
Nandita Vijaykumar[5] Onur Mutlu[1]

[1]ETH Zürich [2]King Mongkut's University of Technology North Bangkok
[3]Norwegian University of Science and Technology [4]Intel Labs [5]University of Toronto

https://arxiv.org/abs/2211.12205

New Architectures & Technologies

New Architectures & Technologies

- Can have large impact on security and robustness
  - Positive or negative

- They need to be designed with system security in mind
  - Ideally as a first-class design goal

- Multiple potentially paradigm-changing new technologies and architectures
  - Processing in memory
  - Accelerator-based computing
  - Quantum computing

Processing in Memory

Problem

Computing

is Bottlenecked by Data

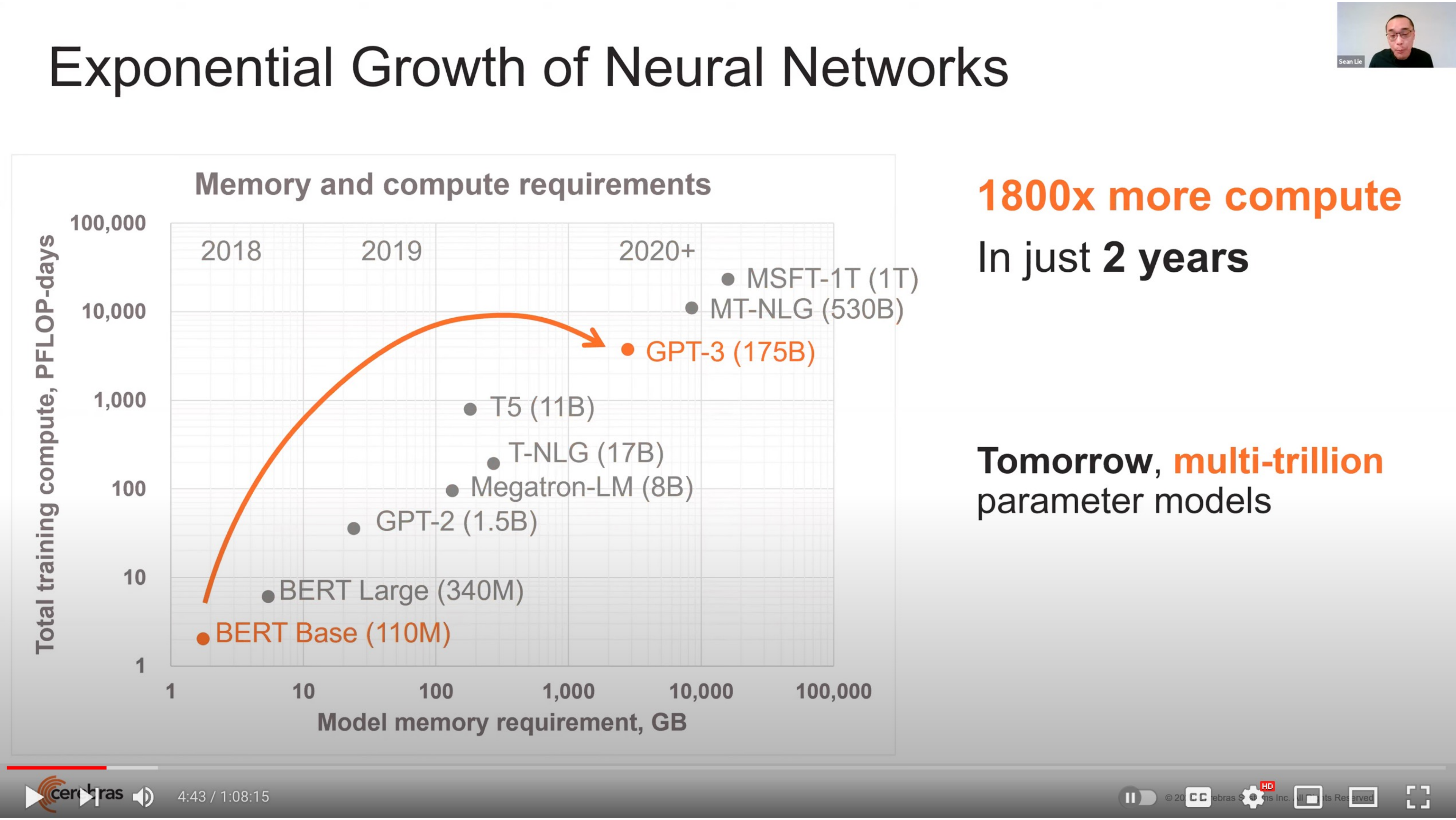
Data is Key for AI, ML, Genomics, …

- Important workloads are all data intensive

- They require rapid and efficient processing of large amounts of data

- Data is increasing
  - We can generate more than we can process
  - We need to perform more sophisticated analyses on more data

Huge Demand for Performance & Efficiency



Exponential Growth of Neural Networks

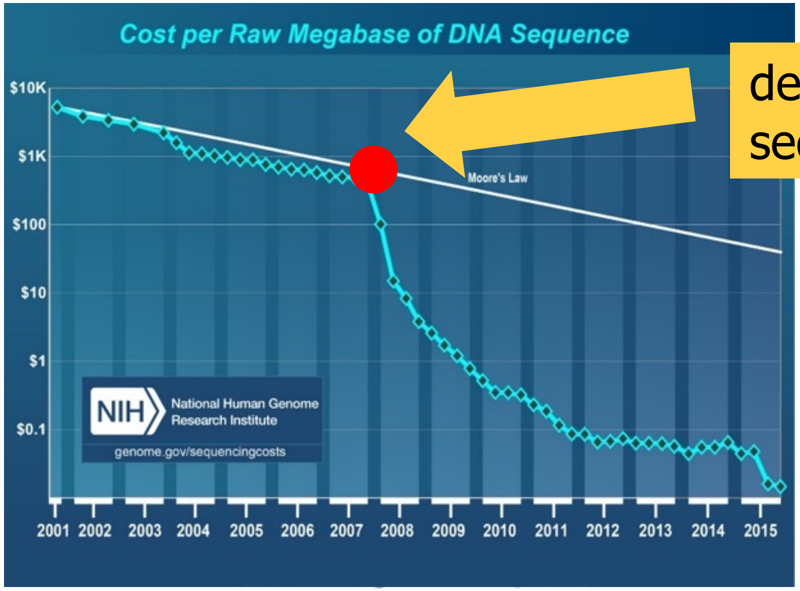
1800x more compute

In just **2 years**

Tomorrow, **multi-trillion** parameter models

Source: https://youtu.be/Bh13Idwcb0Q?t=283

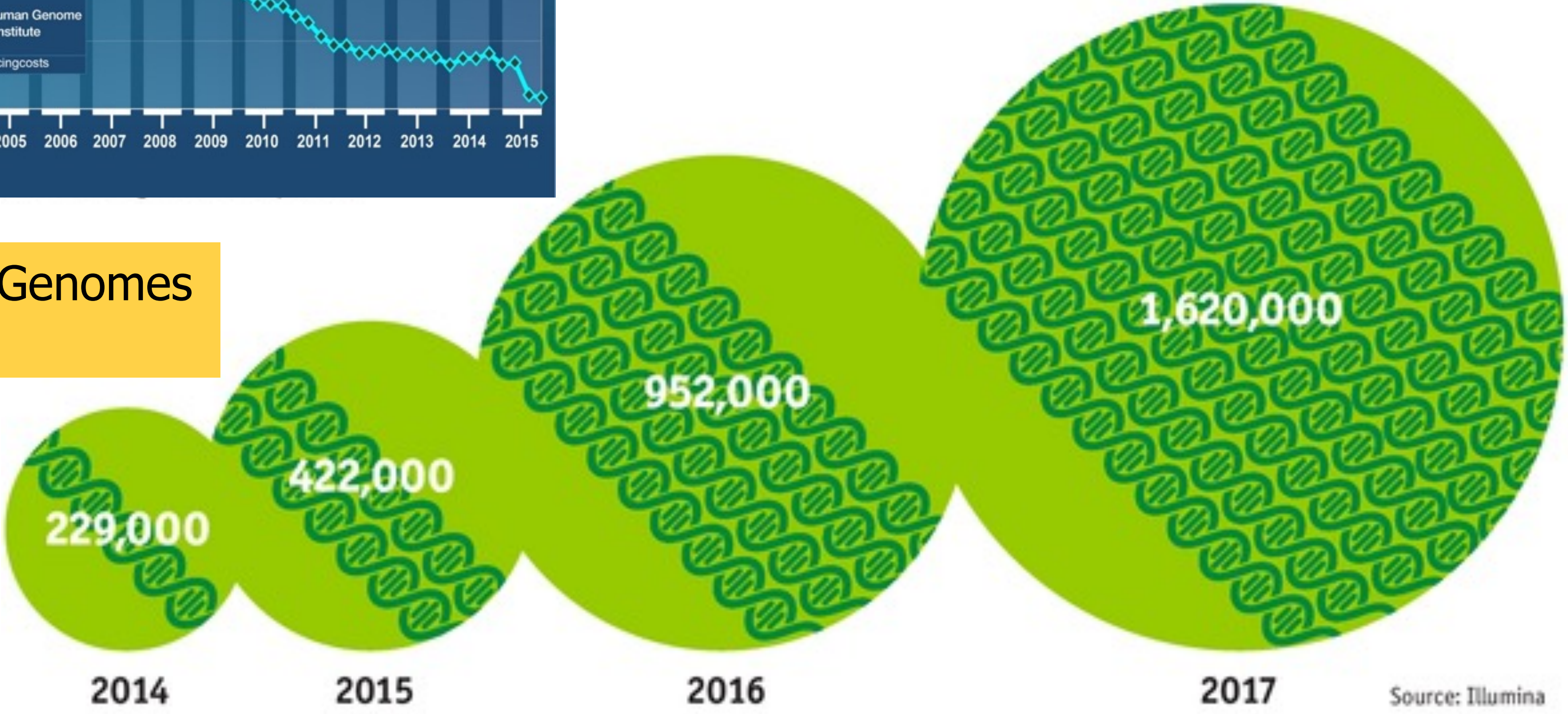
Huge Demand for Performance & Efficiency



development of new sequencing technologies

Oxford Nanopore MinION

Number of Genomes Sequenced



Source: Illumina

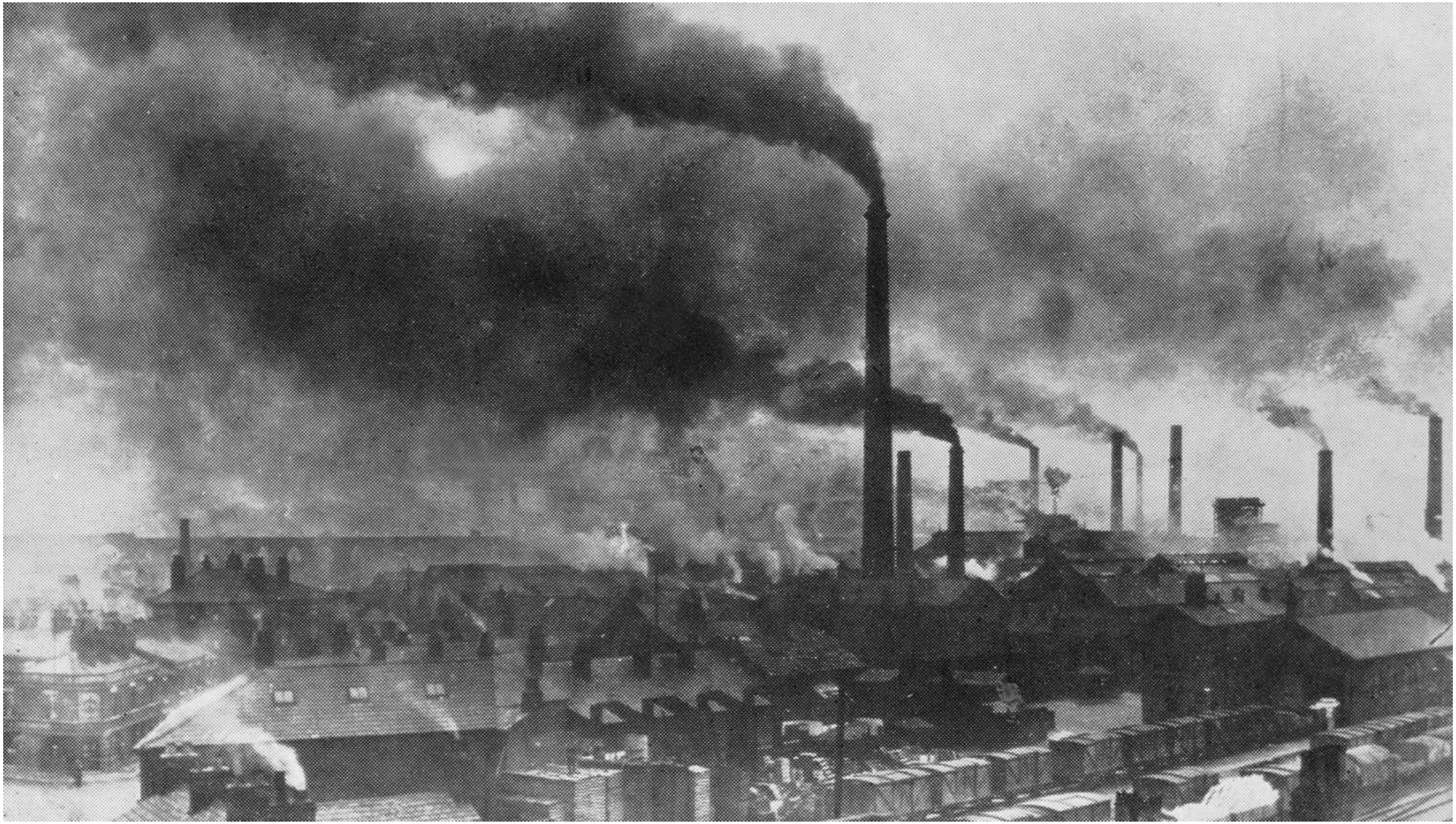
The Economist

http://www.economist.com/news/21631808-so-much-genetic-data-so-many-uses-genes-unzipped

Do We Want This?

Source: V. Milutinovic

Or This?



Source: V. Milutinovic

Challenge and Opportunity for Future

High Performance,

Energy Efficient,

Sustainable

(All at the Same Time)

The Problem

Data access is the major performance and energy bottleneck

Our current

design principles

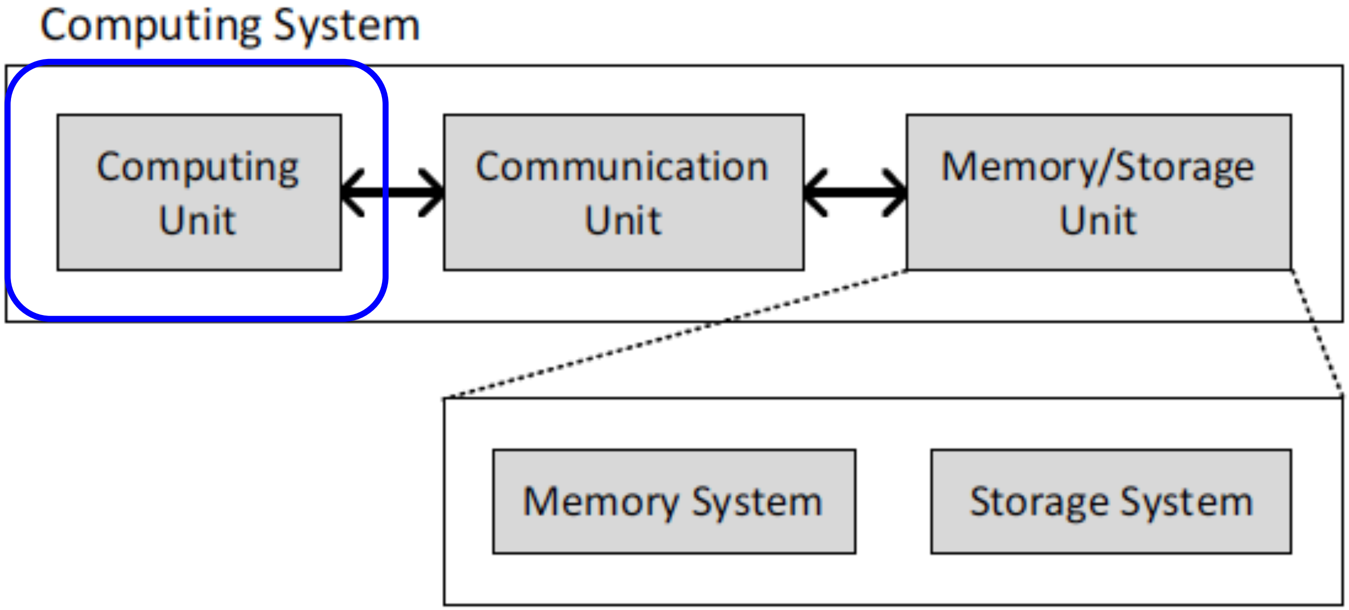
cause great energy waste

(and great performance loss)

Today’s Computing Systems

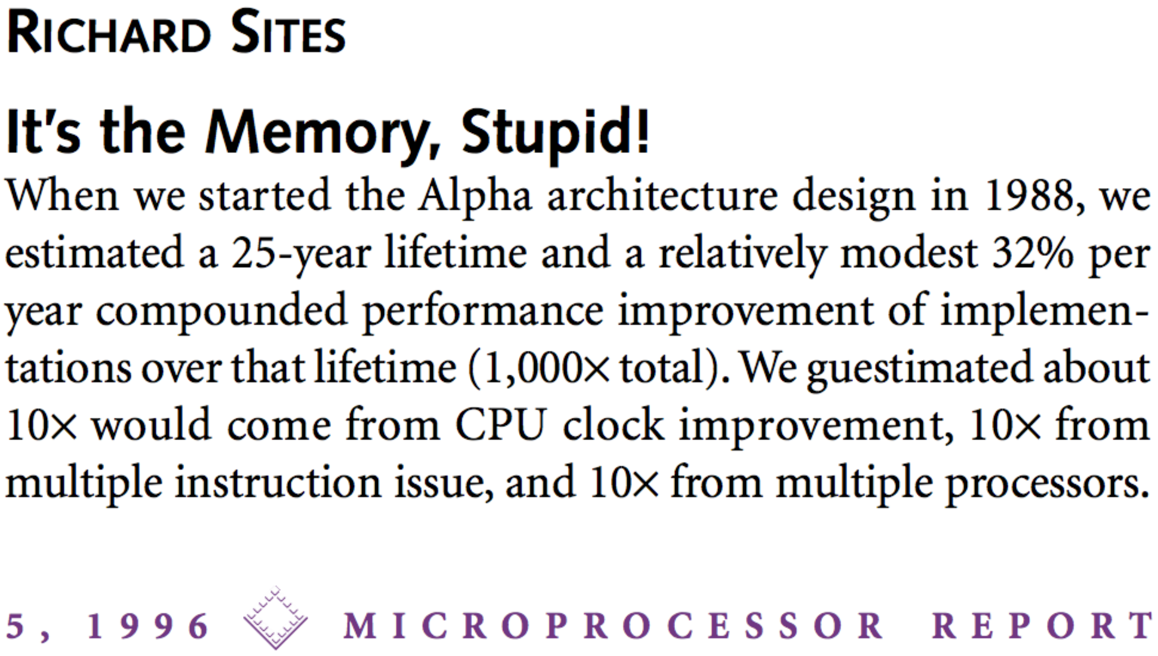
- Processor centric

- All data processed in the processor → at great system cost



It’s the Memory, Stupid!

- “**It’s the Memory, Stupid!**” (Richard Sites, MPR, 1996)

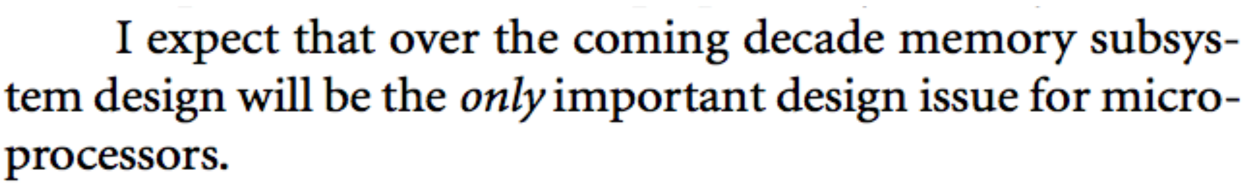


RICHARD SITES

It's the Memory, Stupid!

When we started the Alpha architecture design in 1988, we estimated a 25-year lifetime and a relatively modest 32% per year compounded performance improvement of implementations over that lifetime (1,000× total). We guestimated about 10× would come from CPU clock improvement, 10× from multiple instruction issue, and 10× from multiple processors.

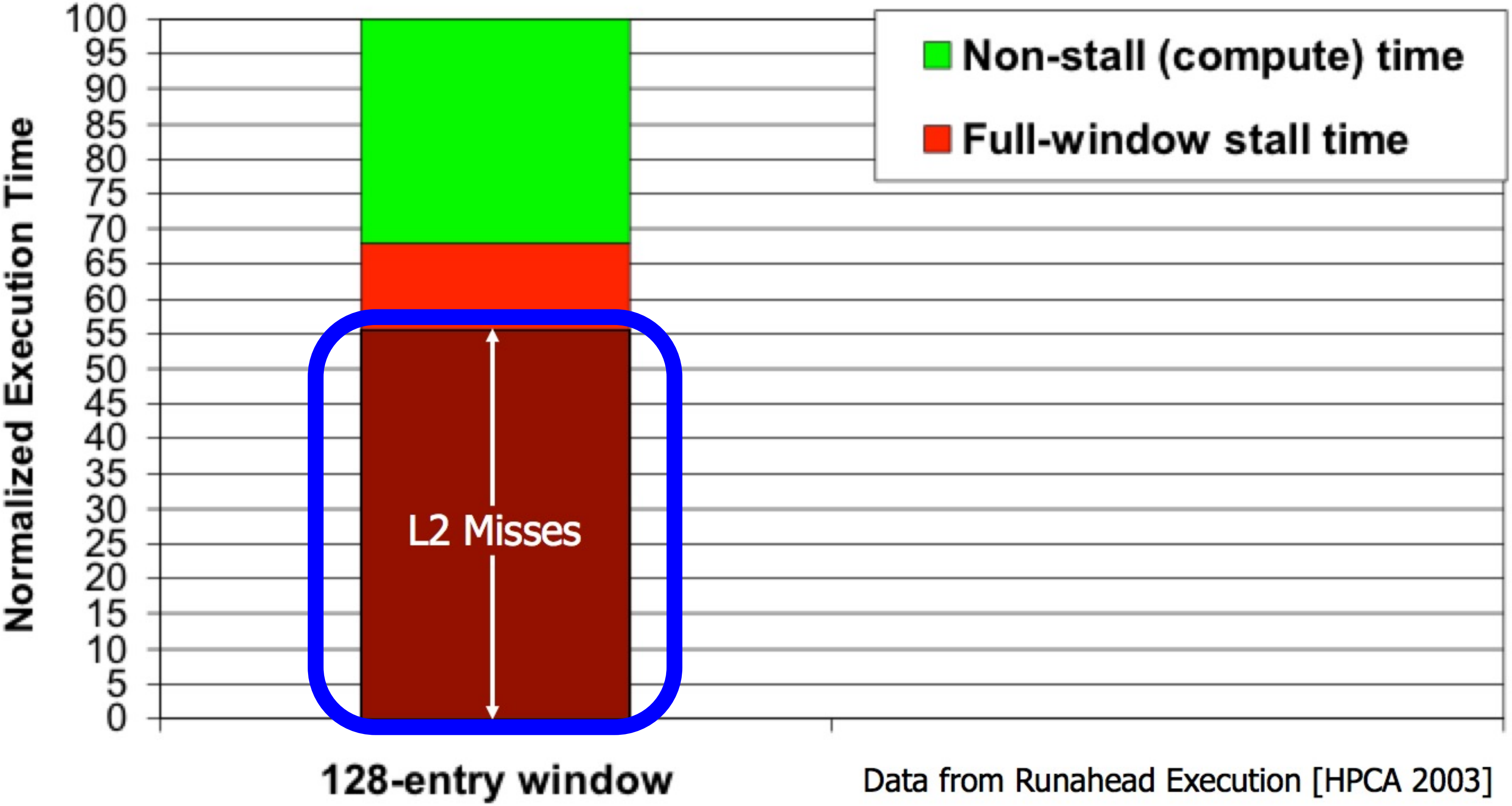
5, 1996 MICROPROCESSOR REPORT



I expect that over the coming decade memory subsystem design will be the *only* important design issue for microprocessors.

http://cva.stanford.edu/classes/cs99s/papers/architects_look_to_future.pdf

Processor-Centric System Performance

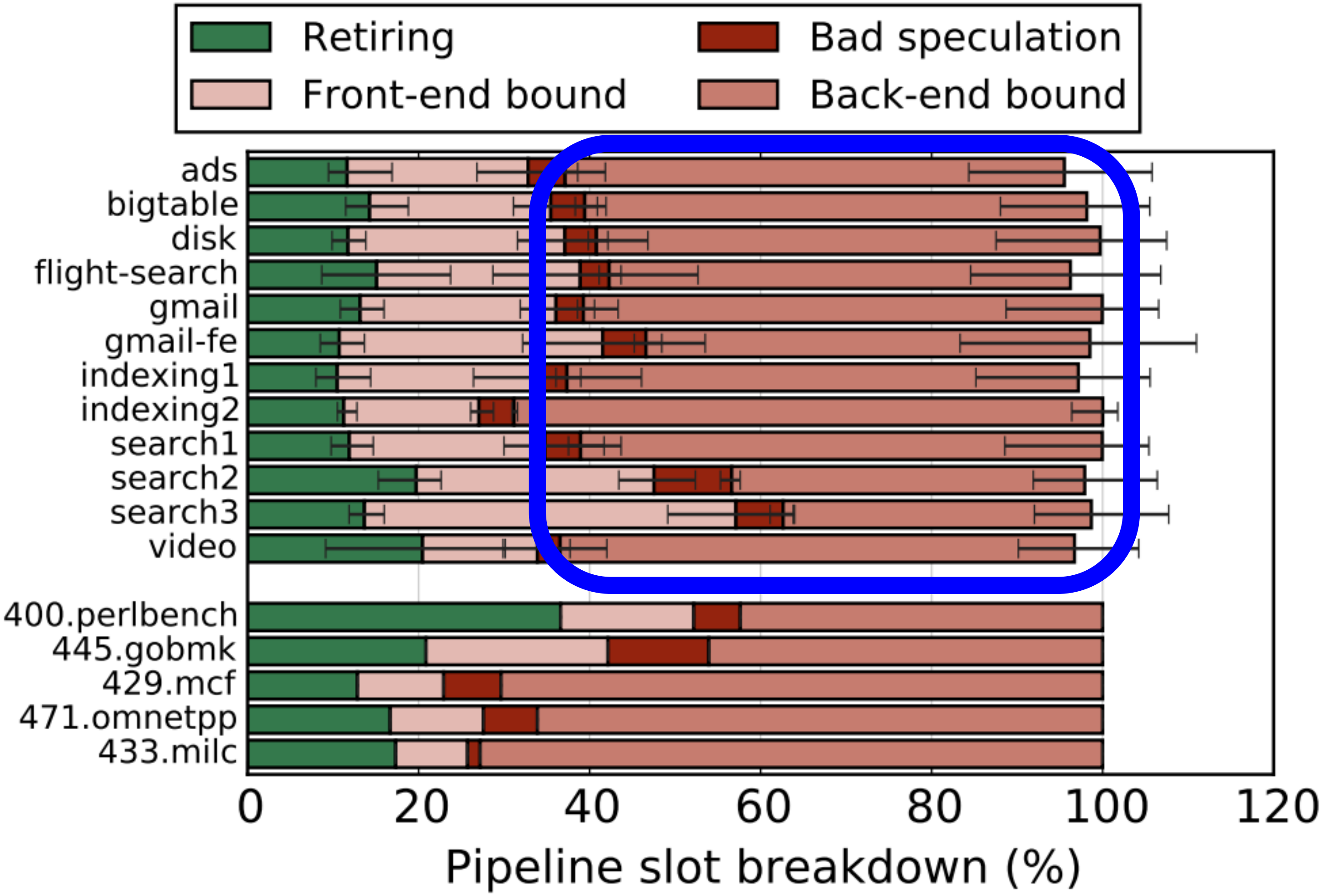


Data from Runahead Execution [HPCA 2003]

Mutlu+, "Runahead Execution: An Alternative to Very Large Instruction Windows for Out-of-Order Processors," HPCA 2003.

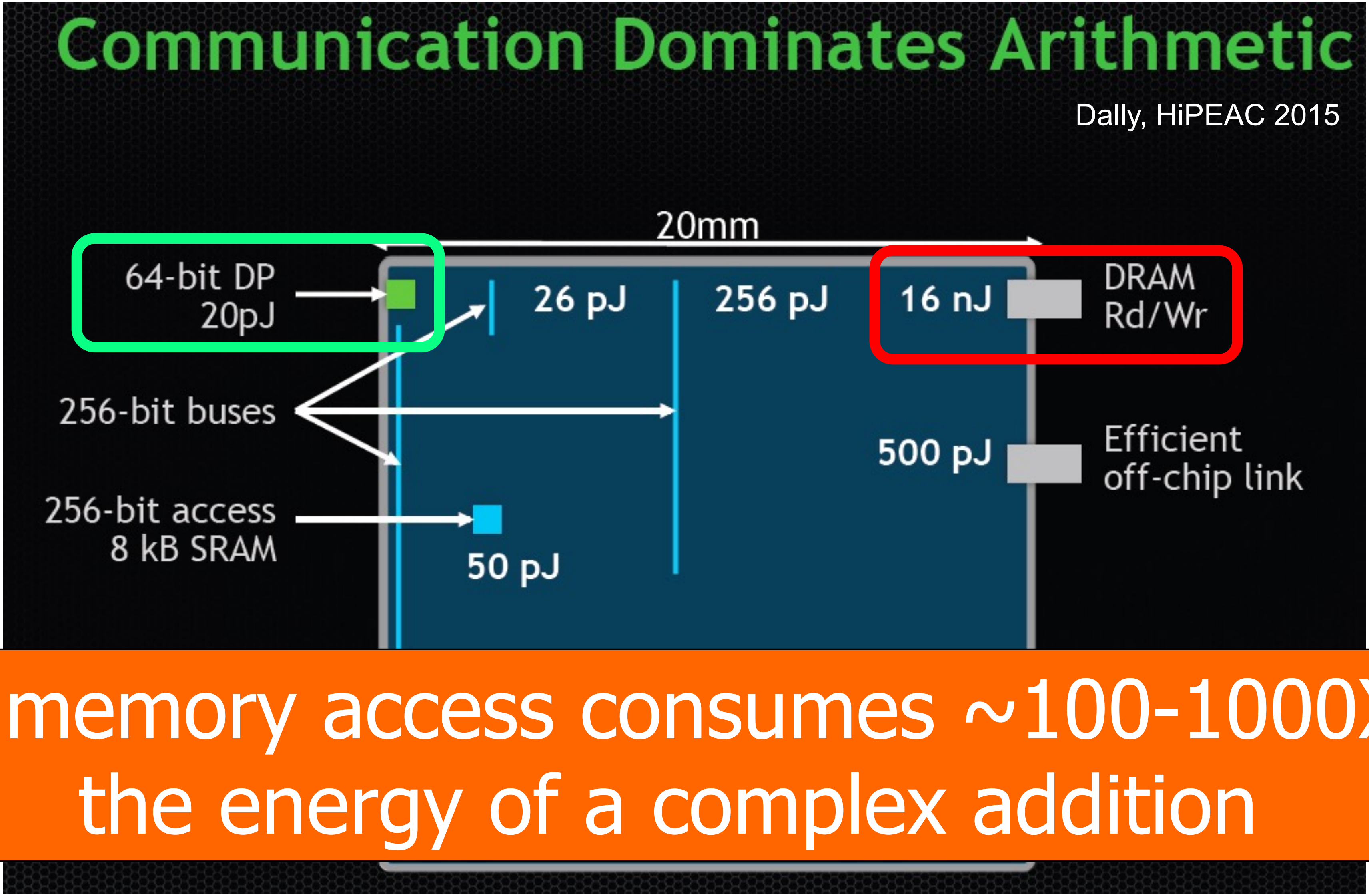
Processor-Centric System Performance

- All of Google's Data Center Workloads (2015):



Kanev+, "Profiling a Warehouse-Scale Computer," ISCA 2015.

Data Movement vs. Computation Energy

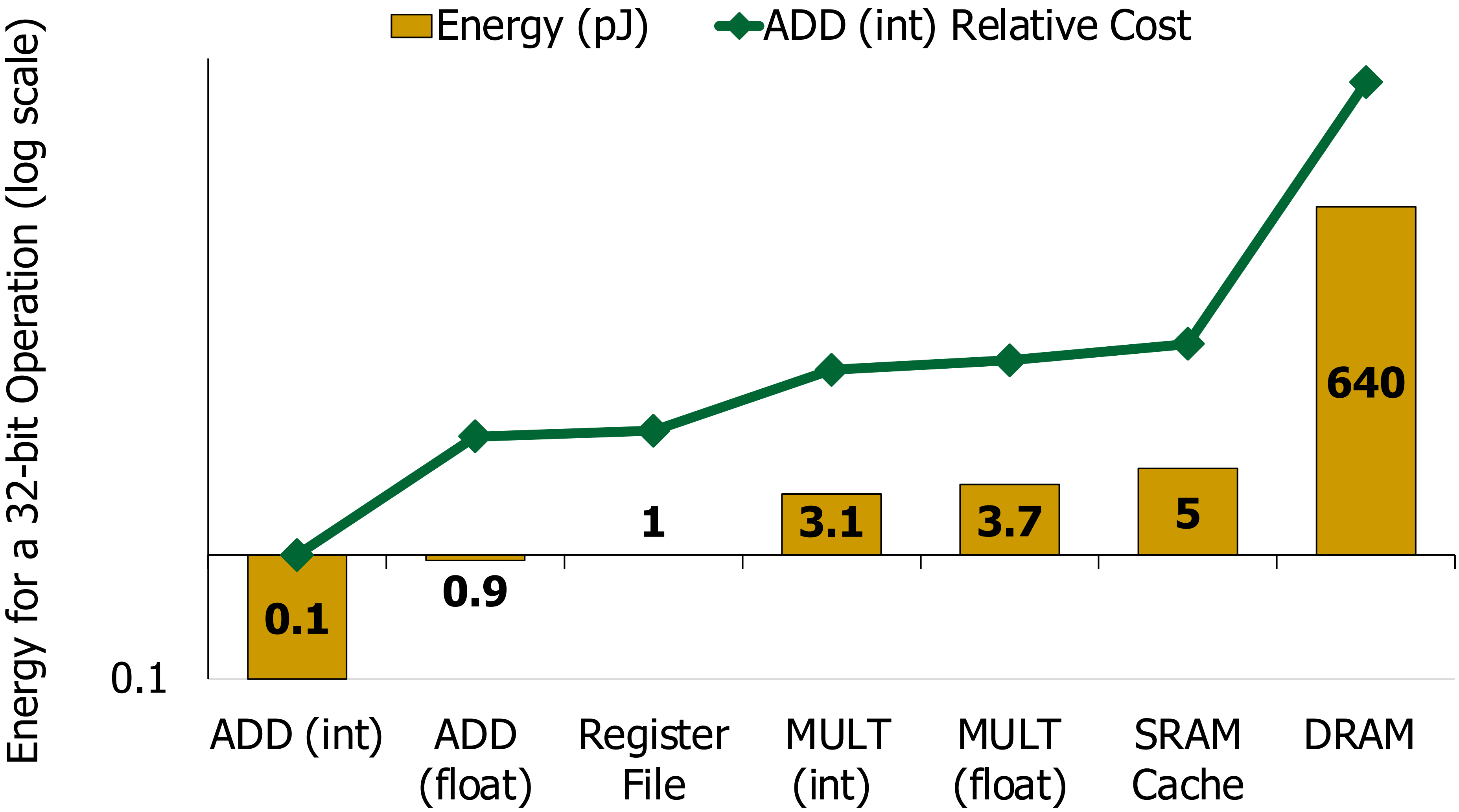


Communication Dominates Arithmetic

Dally, HiPEAC 2015

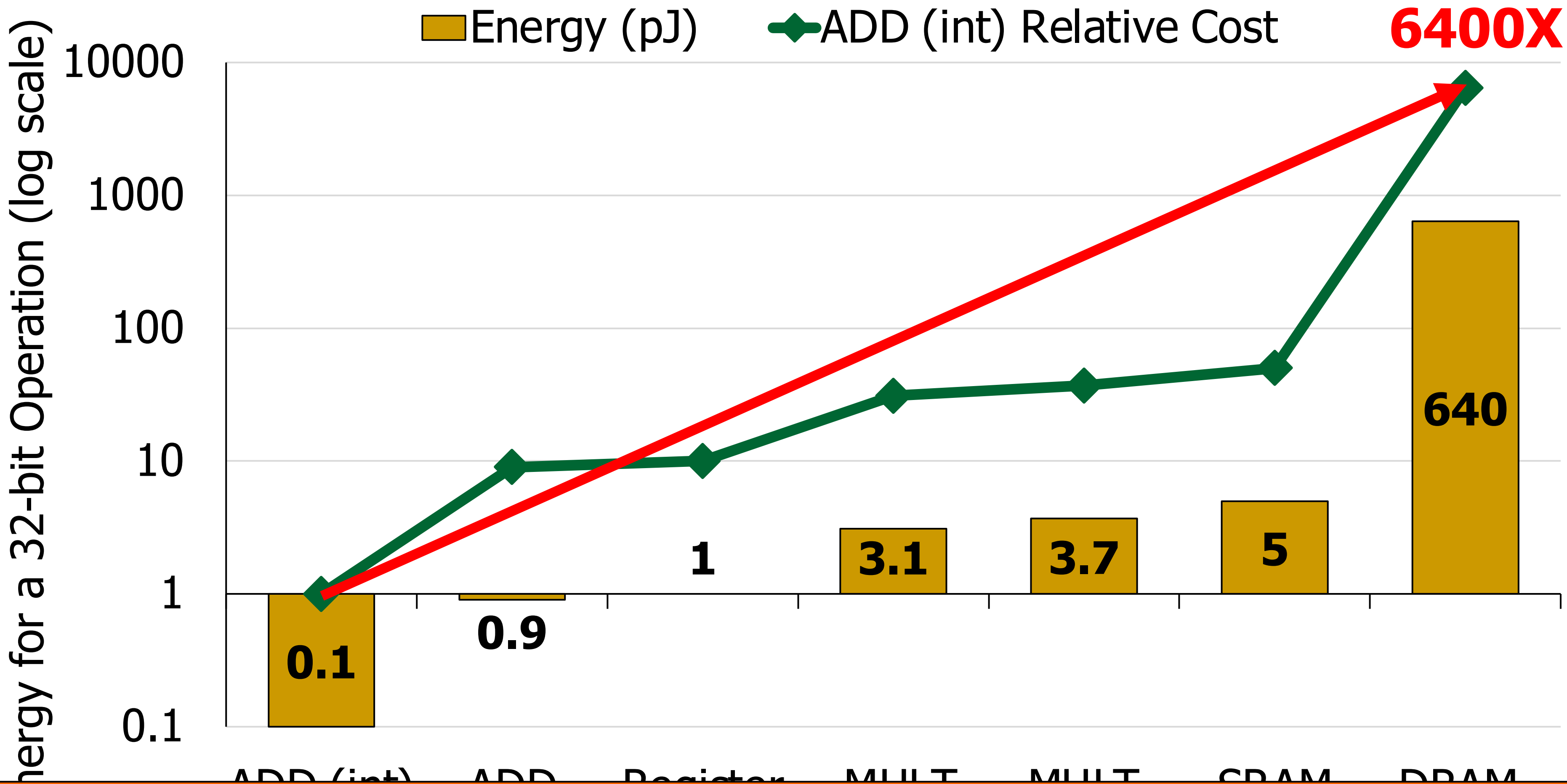
A memory access consumes ~100-1000X the energy of a complex addition

Data Movement vs. Computation Energy



Han+, "EIE: Efficient Inference Engine on Compressed Deep Neural Network," ISCA 2016.

Data Movement vs. Computation Energy



A memory access consumes 6400X the energy of a simple integer addition

Energy Waste in Mobile Devices

- Amirali Boroumand, Saugata Ghose, Youngsok Kim, Rachata Ausavarungnirun, Eric Shiu, Rahul Thakur, Daehyun Kim, Aki Kuusela, Allan Knies, Parthasarathy Ranganathan, and Onur Mutlu, **"Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks"** *Proceedings of the 23rd International Conference on Architectural Support for Programming Languages and Operating Systems* (***ASPLOS***), Williamsburg, VA, USA, March 2018.

62.7% of the total system energy is spent on data movement

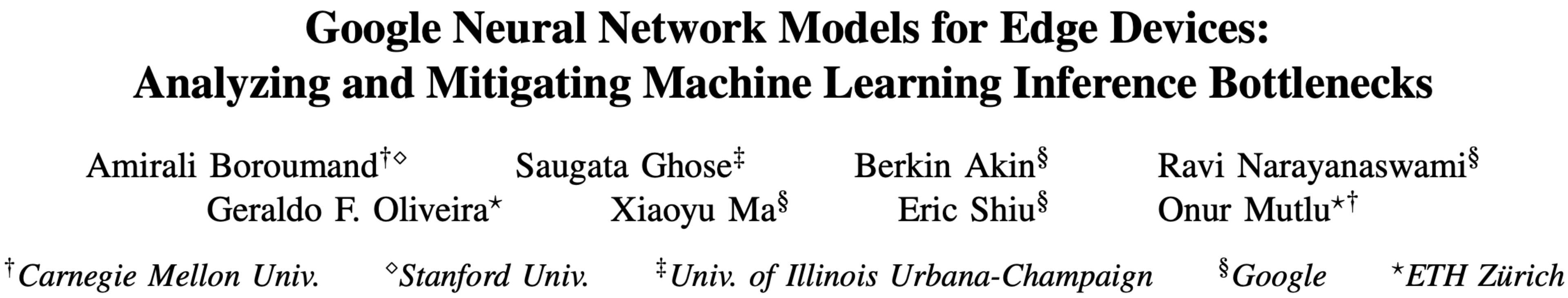
Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks

Amirali Boroumand[1] Saugata Ghose[1] Youngsok Kim[2]
Rachata Ausavarungnirun[1] Eric Shiu[3] Rahul Thakur[3] Daehyun Kim[4,3]
Aki Kuusela[3] Allan Knies[3] Parthasarathy Ranganathan[3] Onur Mutlu[5,1]

Energy Waste in Accelerators

- Amirali Boroumand, Saugata Ghose, Berkin Akin, Ravi Narayanaswami, Geraldo F. Oliveira, Xiaoyu Ma, Eric Shiu, and Onur Mutlu,
  "Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks"
  Proceedings of the 30th International Conference on Parallel Architectures and Compilation Techniques (***PACT***), Virtual, September 2021.
  [Slides (pptx) (pdf)]
  [Talk Video (14 minutes)]

> 90% of the total system energy is spent on memory in large ML models

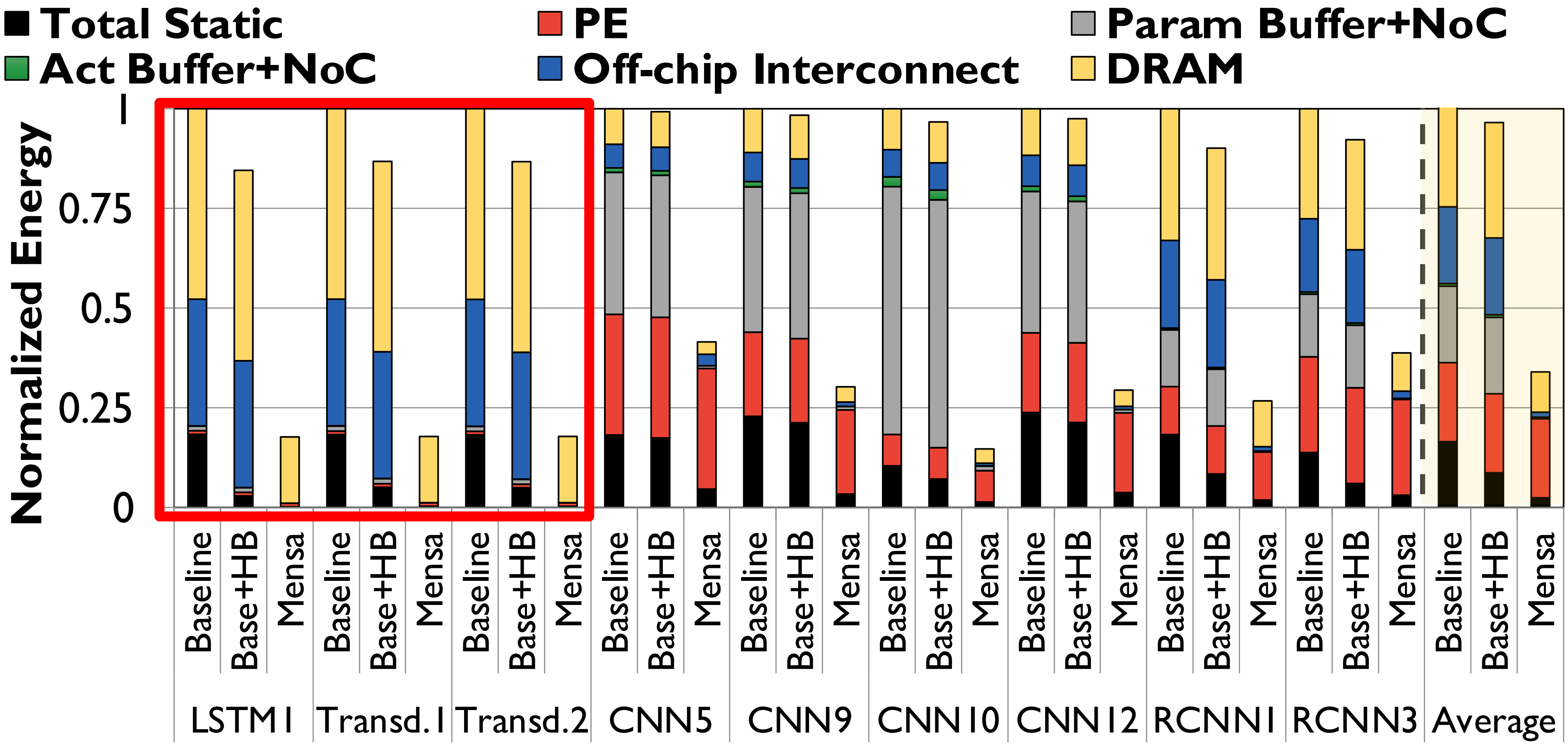


Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks

Amirali Boroumand[†◇] Saugata Ghose[‡] Berkin Akin[§] Ravi Narayanaswami[§]
Geraldo F. Oliveira[⋆] Xiaoyu Ma[§] Eric Shiu[§] Onur Mutlu[⋆†]

[†]*Carnegie Mellon Univ.* [◇]*Stanford Univ.* [‡]*Univ. of Illinois Urbana-Champaign* [§]*Google* [⋆]*ETH Zürich*

Example Energy Breakdowns



In LSTMs and Transducers used by Google, >90% energy spent on off-chip interconnect and DRAM

https://arxiv.org/pdf/2109.14320

SAFARI

Fundamental Problem

Processing of data
is performed
far away from the data

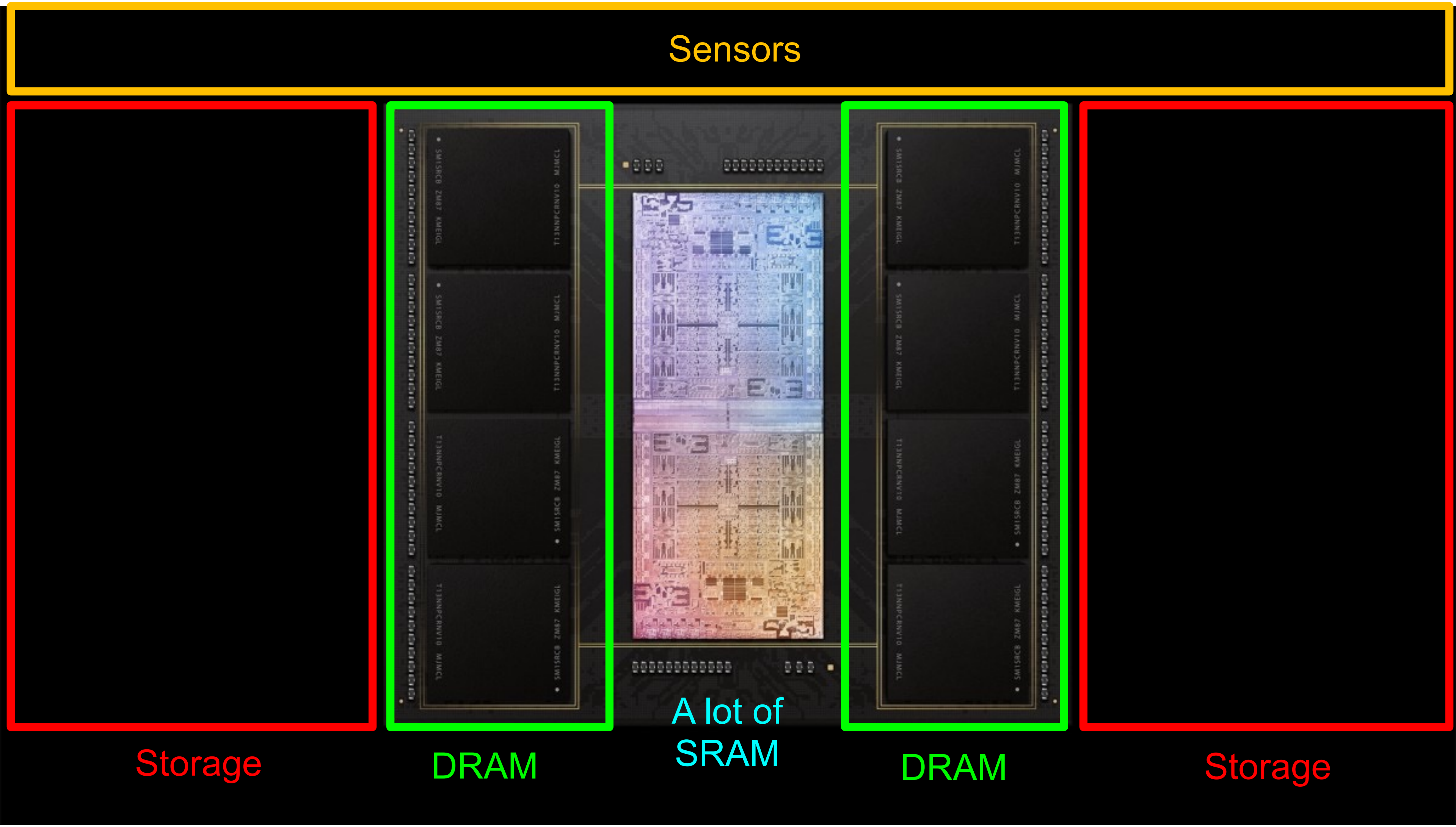
We Need A **Paradigm Shift** To …

- Enable computation with minimal data movement

- Compute where it makes sense (where data resides)

- Make computing architectures more data-centric

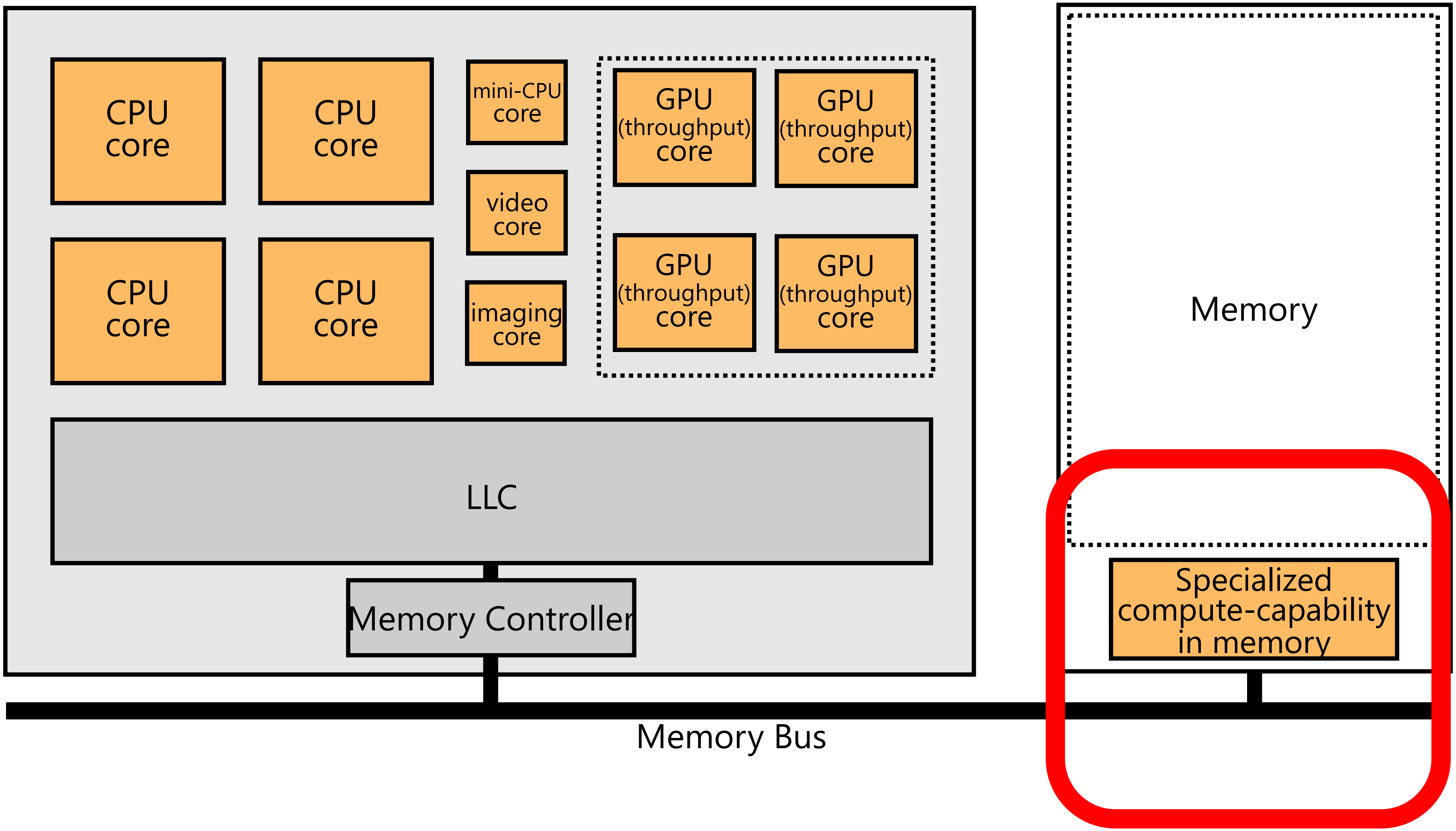
Process Data Where It Makes Sense



Apple M1 Ultra System (2022)

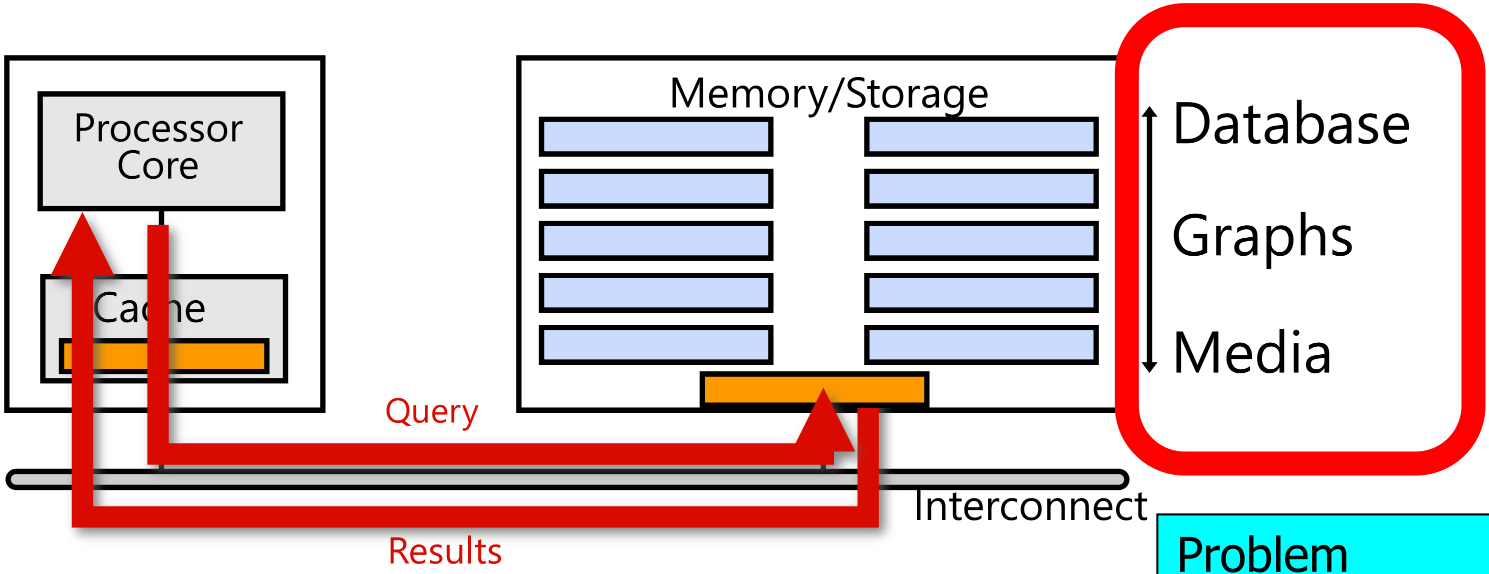
https://www.gsmarena.com/apple_announces_m1_ultra_with_20core_cpu_and_64core_gpu-news-53481.php

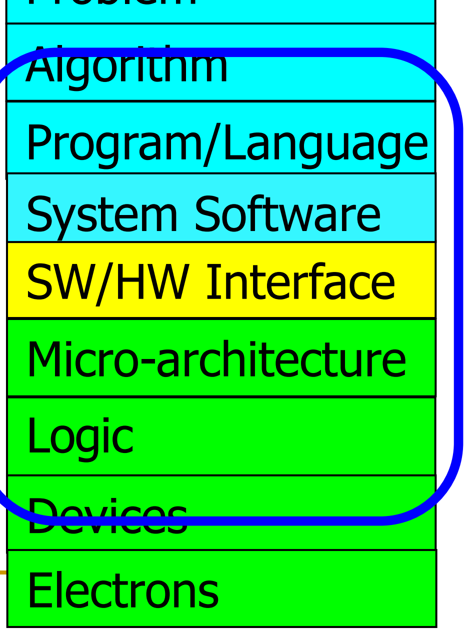
Memory as an Accelerator



Memory similar to a "conventional" accelerator

Goal: Processing Inside Memory/Storage



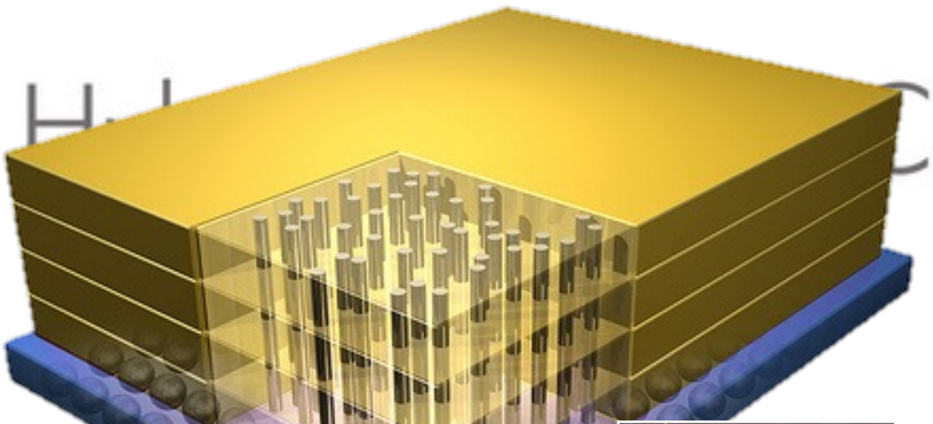


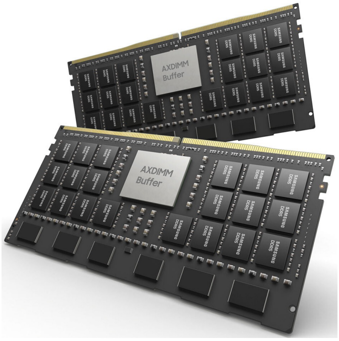
- Many questions … How do we design the:
  - compute-capable memory & controllers?
  - processors & communication units?
  - software & hardware interfaces?
  - system software, compilers, languages?
  - algorithms & theoretical foundations?

Processing in Memory: Two Types

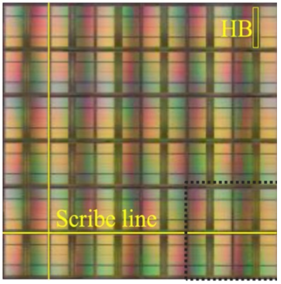
1. Processing **near** Memory
2. Processing **using** Memory

Processing-in-Memory Landscape Today

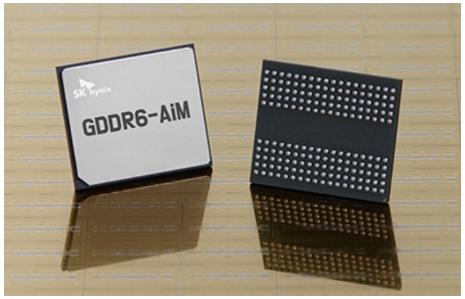




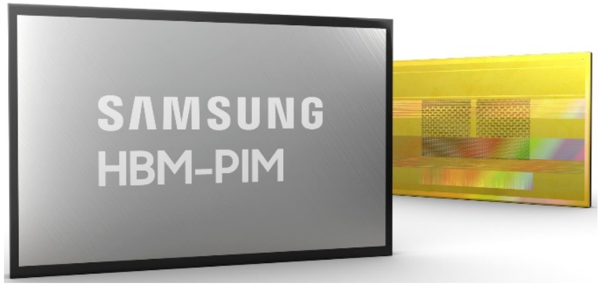
[Samsung 2021]



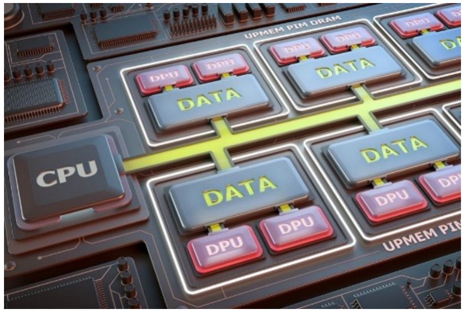
[Alibaba 2022]



[SK Hynix 2022]



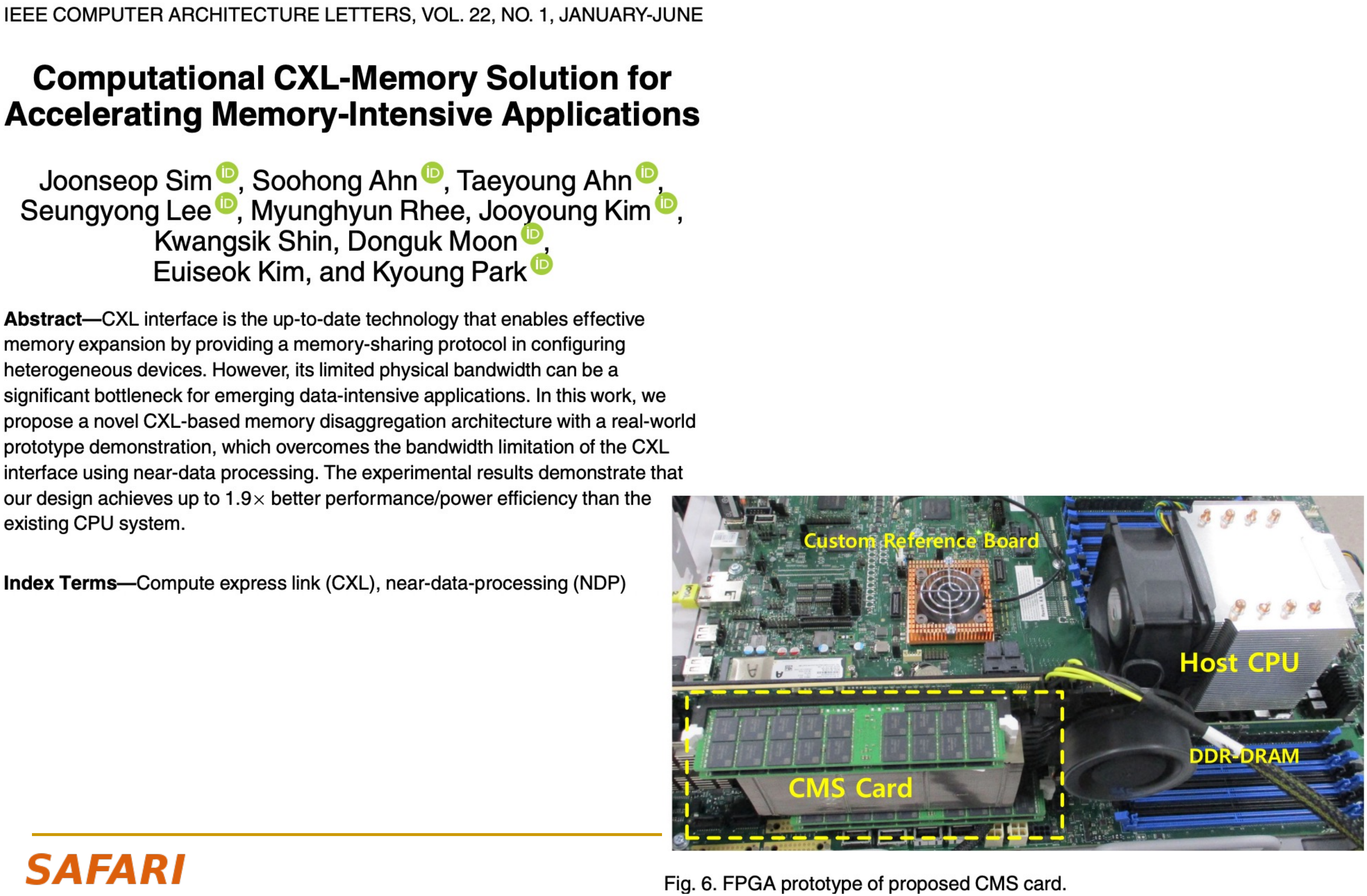
[Samsung 2021]



[UPMEM 2019]

And, many other experimental chips and startups

Processing-in-Memory Landscape Today



IEEE COMPUTER ARCHITECTURE LETTERS, VOL. 22, NO. 1, JANUARY-JUNE

Computational CXL-Memory Solution for Accelerating Memory-Intensive Applications

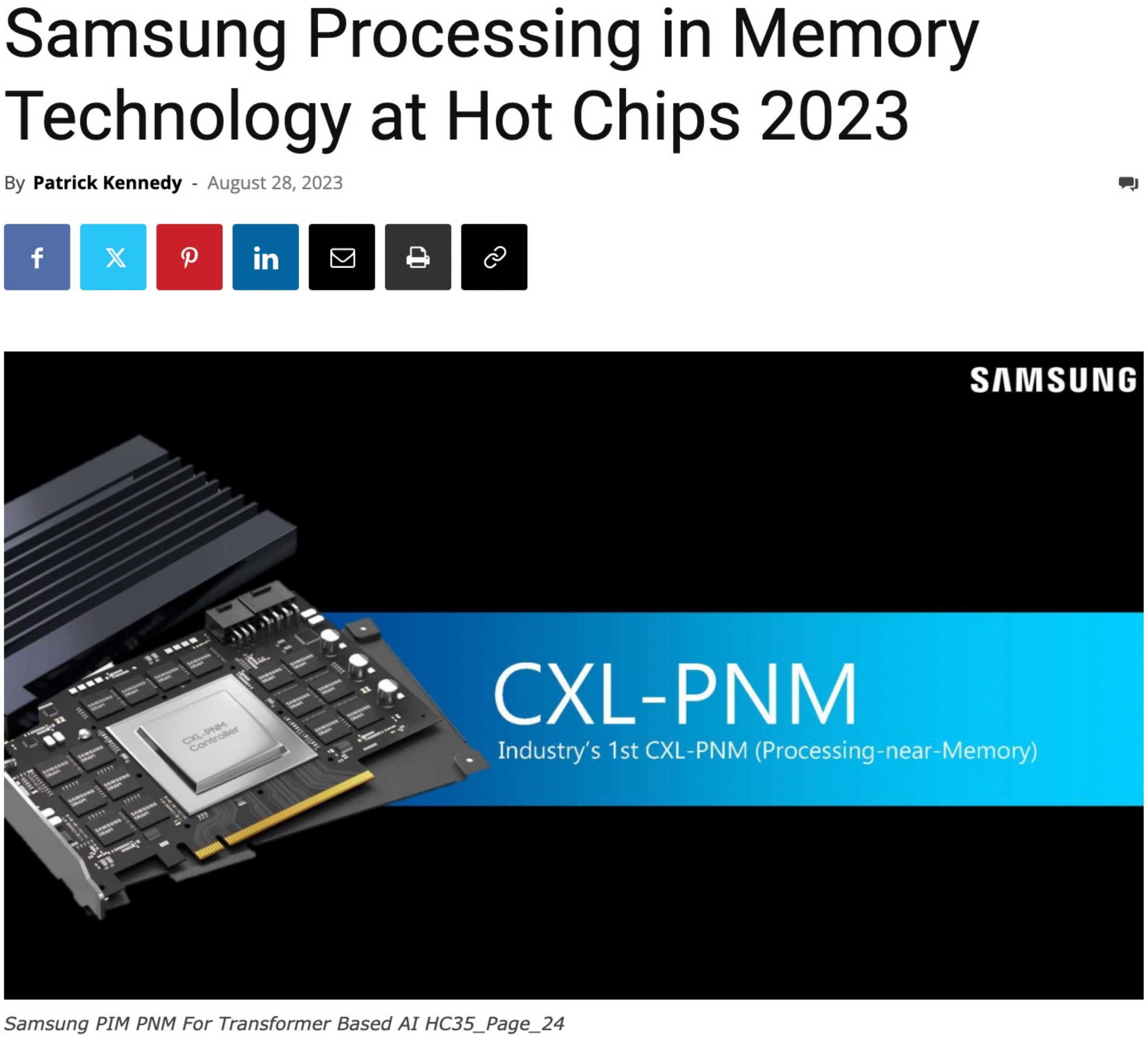
Joonseop Sim, Soohong Ahn, Taeyoung Ahn,
Seungyong Lee, Myunghyun Rhee, Jooyoung Kim,
Kwangsik Shin, Donguk Moon,
Euiseok Kim, and Kyoung Park

Abstract—CXL interface is the up-to-date technology that enables effective memory expansion by providing a memory-sharing protocol in configuring heterogeneous devices. However, its limited physical bandwidth can be a significant bottleneck for emerging data-intensive applications. In this work, we propose a novel CXL-based memory disaggregation architecture with a real-world prototype demonstration, which overcomes the bandwidth limitation of the CXL interface using near-data processing. The experimental results demonstrate that our design achieves up to 1.9× better performance/power efficiency than the existing CPU system.

Index Terms—Compute express link (CXL), near-data-processing (NDP)

Fig. 6. FPGA prototype of proposed CMS card.

Processing-in-Memory Landscape Today



Samsung Processing in Memory Technology at Hot Chips 2023

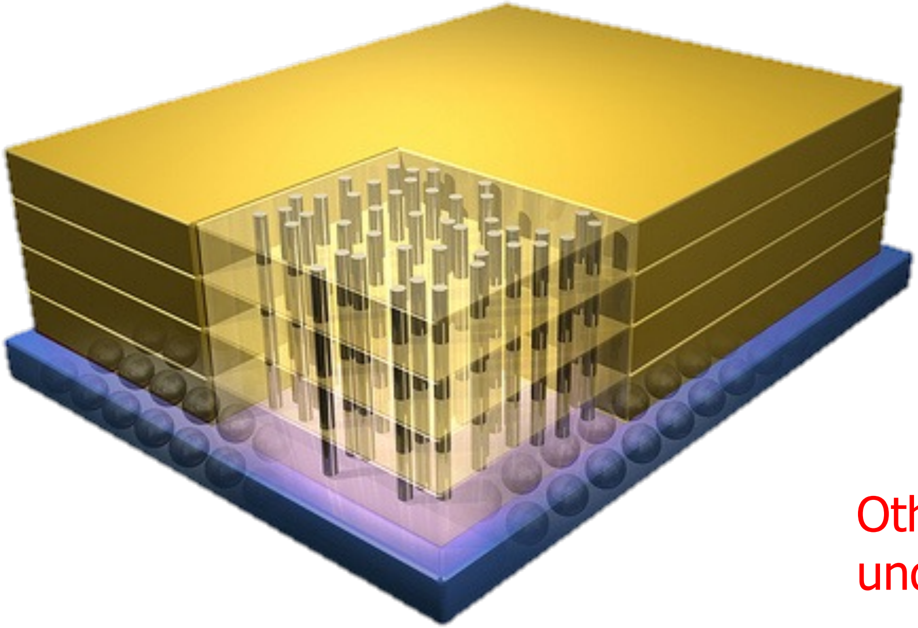
By **Patrick Kennedy** - August 28, 2023

Samsung PIM PNM For Transformer Based AI HC35_Page_24

https://www.servethehome.com/samsung-processing-in-memory-technology-at-hot-chips-2023/

Opportunity: 3D-Stacked Logic+Memory

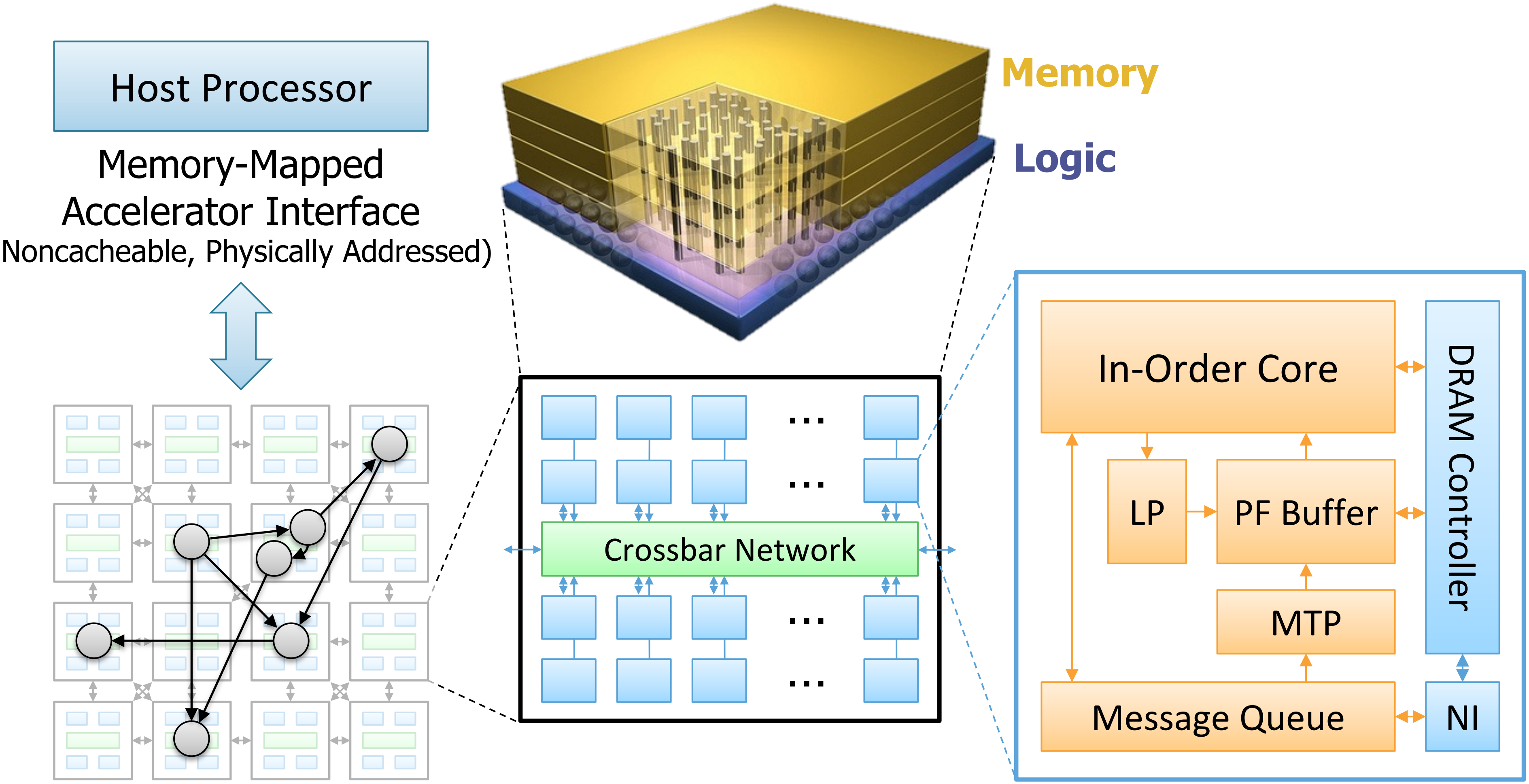
Hybrid Memory Cube
CONSORTIUM



Other “True 3D” technologies under development

Tesseract System for Graph Processing

Interconnected set of 3D-stacked memory+logic chips with simple cores



Host Processor

Memory-Mapped Accelerator Interface
Noncacheable, Physically Addressed)

Memory

Logic

Crossbar Network

In-Order Core

LP

PF Buffer

MTP

Message Queue

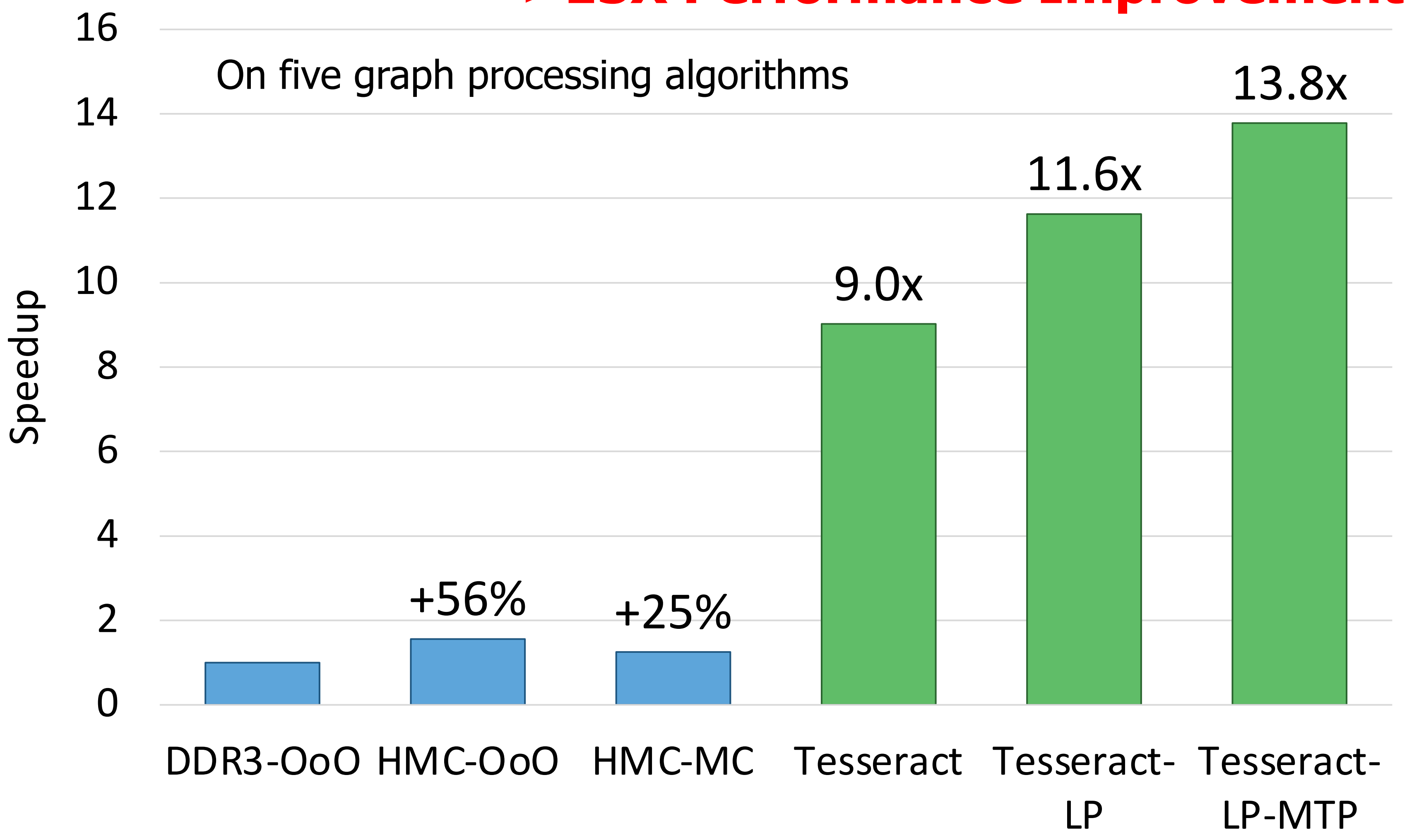
DRAM Controller

NI

Ahn+, "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing" ISCA 2015.

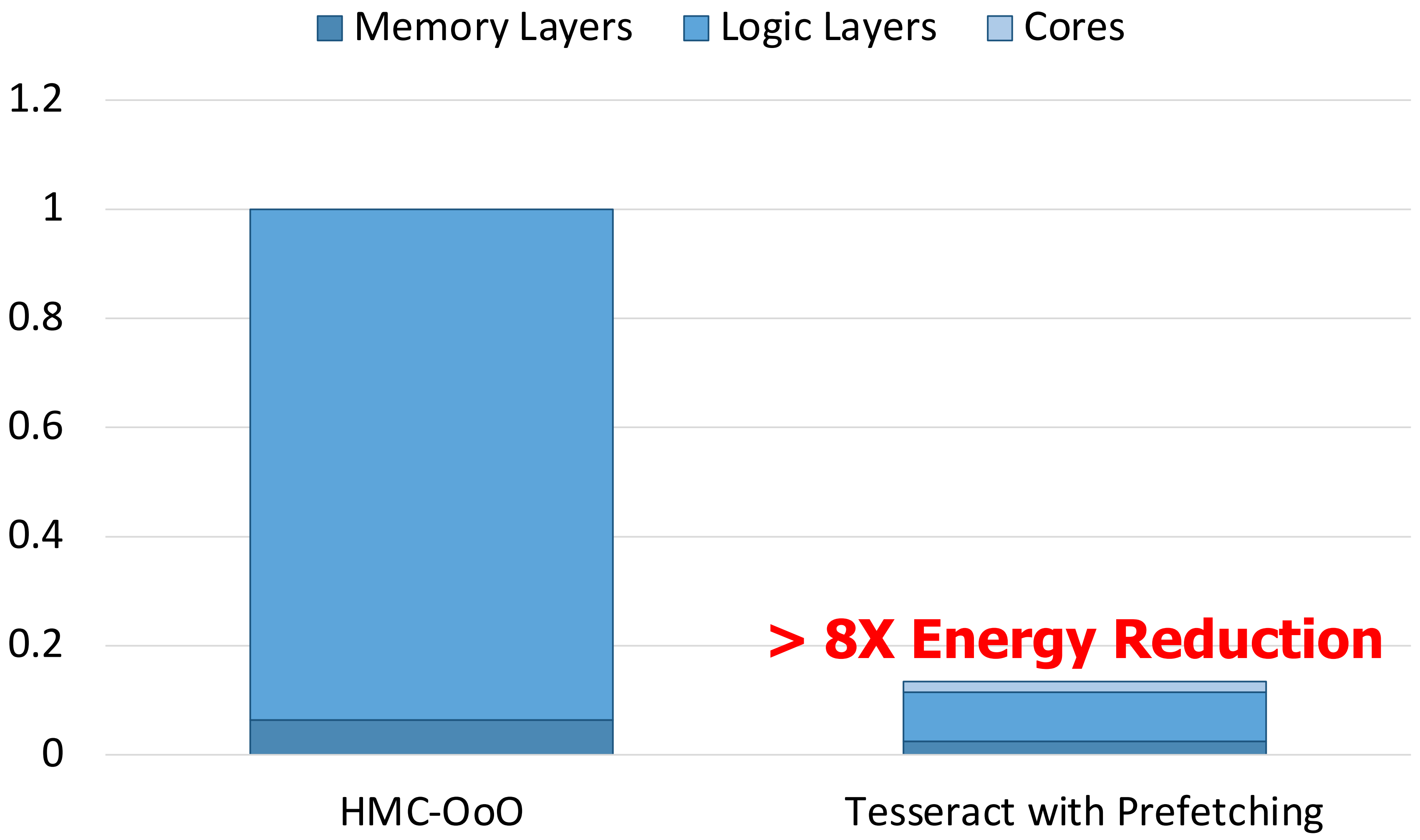
Tesseract Graph Processing Performance

>13X Performance Improvement



SAFARI Ahn+, "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing" ISCA 2015.

Tesseract Graph Processing System Energy



> 8X Energy Reduction

SAFARI Ahn+, "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing" ISCA 2015.

Processing using DRAM

- We can support
  - Bulk bitwise AND, OR, NOT, MAJ
  - Bulk bitwise COPY and INIT/ZERO
  - True Random Number Generation; Physical Unclonable Functions
  - More complex computation using Lookup Tables
- At low cost
- Using analog computation capability of DRAM
  - Idea: activating (multiple) rows performs computation
    - Even in commodity off-the-shelf DRAM chips!

- 30X-257X performance and energy improvements

Seshadri+"RowClone: Fast and Efficient In-DRAM Copy and Initialization of Bulk Data," MICRO 2013.

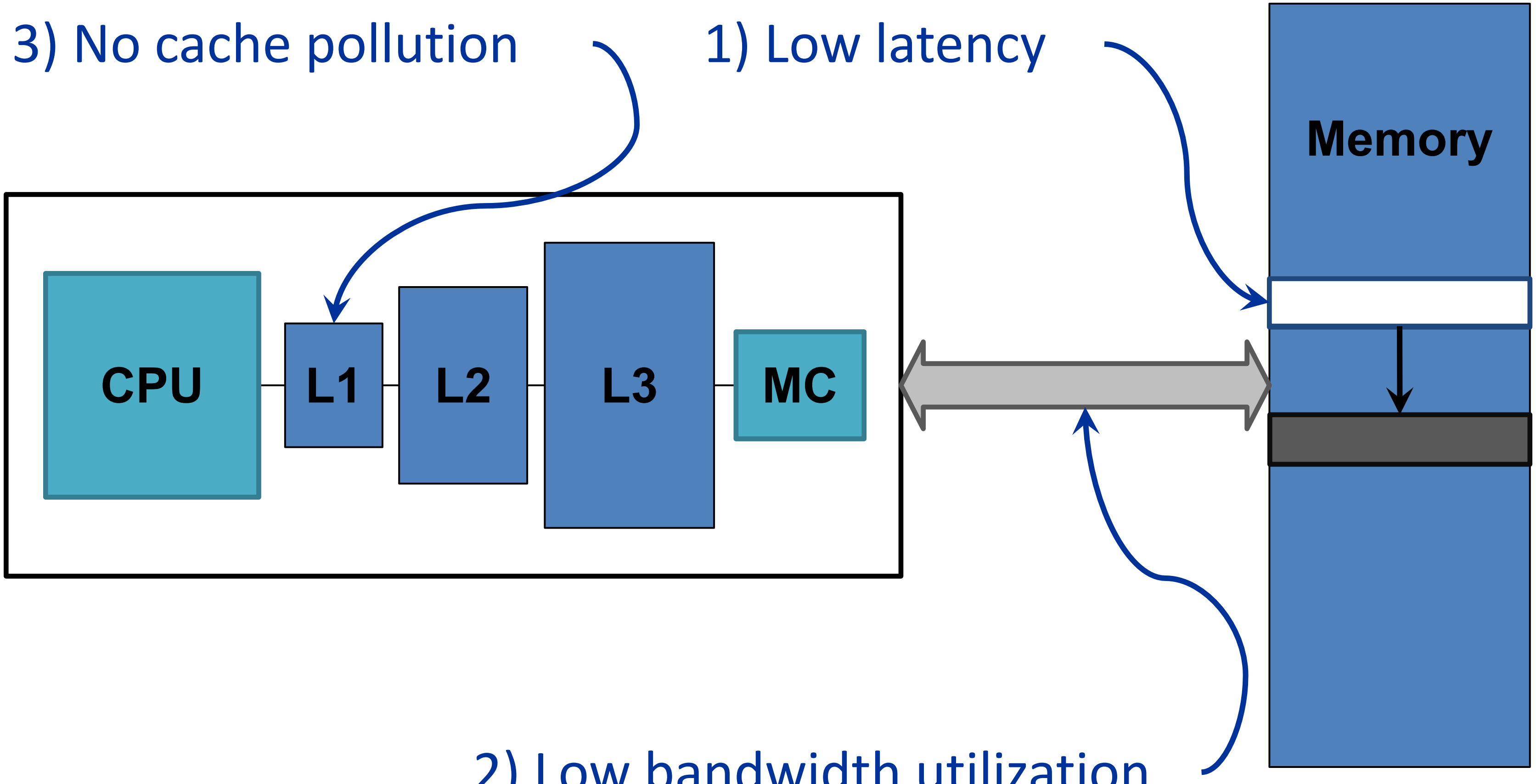
Seshadri+, "Fast Bulk Bitwise AND and OR in DRAM", IEEE CAL 2015.

Seshadri+, "Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology," MICRO 2017.

Hajinazar+, "SIMDRAM: A Framework for Bit-Serial SIMD Processing using DRAM," ASPLOS 2021.

Oliveira+, "MIMDRAM: An End-to-End Processing-Using-DRAM System for High-Throughput, Energy-Efficient and Programmer-Transparent Multiple-Instruction Multiple-Data Processing," HPCA 2024.

Future Systems: In-Memory Copy



3) No cache pollution

1) Low latency

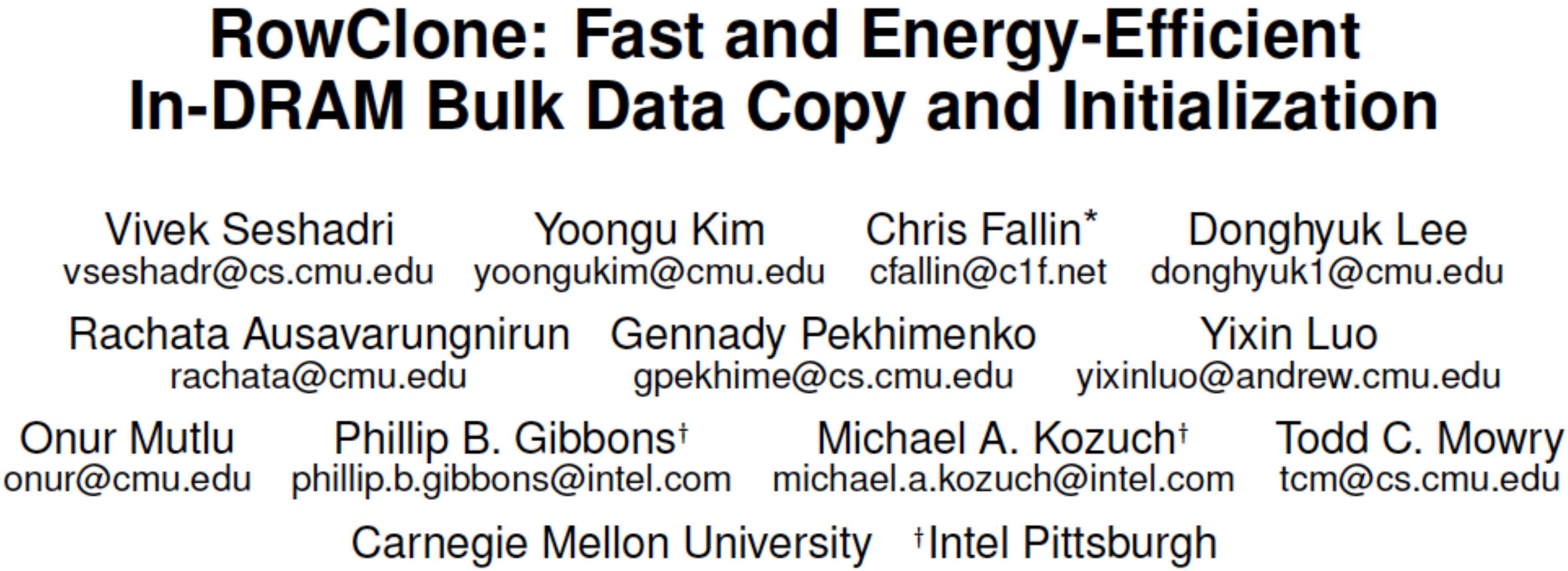
2) Low bandwidth utilization

4) No unwanted data movement

1046ns, 3.6uJ → 90ns, 0.04uJ

More on RowClone

- Vivek Seshadri, Yoongu Kim, Chris Fallin, Donghyuk Lee, Rachata Ausavarungnirun, Gennady Pekhimenko, Yixin Luo, Onur Mutlu, Michael A. Kozuch, Phillip B. Gibbons, and Todd C. Mowry, **"RowClone: Fast and Energy-Efficient In-DRAM Bulk Data Copy and Initialization"** *Proceedings of the 46th International Symposium on Microarchitecture (**MICRO**)*, Davis, CA, December 2013. [Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)] [Poster (pptx) (pdf)]



RowClone: Fast and Energy-Efficient In-DRAM Bulk Data Copy and Initialization

Vivek Seshadri vseshadr@cs.cmu.edu
Yoongu Kim yoongukim@cmu.edu
Chris Fallin* cfallin@c1f.net
Donghyuk Lee donghyuk1@cmu.edu
Rachata Ausavarungnirun rachata@cmu.edu
Gennady Pekhimenko gpekhime@cs.cmu.edu
Yixin Luo yixinluo@andrew.cmu.edu
Onur Mutlu onur@cmu.edu
Phillip B. Gibbons† phillip.b.gibbons@intel.com
Michael A. Kozuch† michael.a.kozuch@intel.com
Todd C. Mowry tcm@cs.cmu.edu

Carnegie Mellon University †Intel Pittsburgh

Capabilities of Off-The-Shelf Memory

Existing DRAM Chips

Are Already Quite Capable

Real Processing Using Memory Prototype

- End-to-end RowClone & TRNG using off-the-shelf DRAM chips
- Idea: Violate DRAM timing parameters to mimic RowClone

PiDRAM: A Holistic End-to-end FPGA-based Framework for Processing-in-DRAM

Ataberk Olgun[§][†] Juan Gómez Luna[§] Konstantinos Kanellopoulos[§] Behzad Salami[§][*]
Hasan Hassan[§] Oğuz Ergin[†] Onur Mutlu[§]

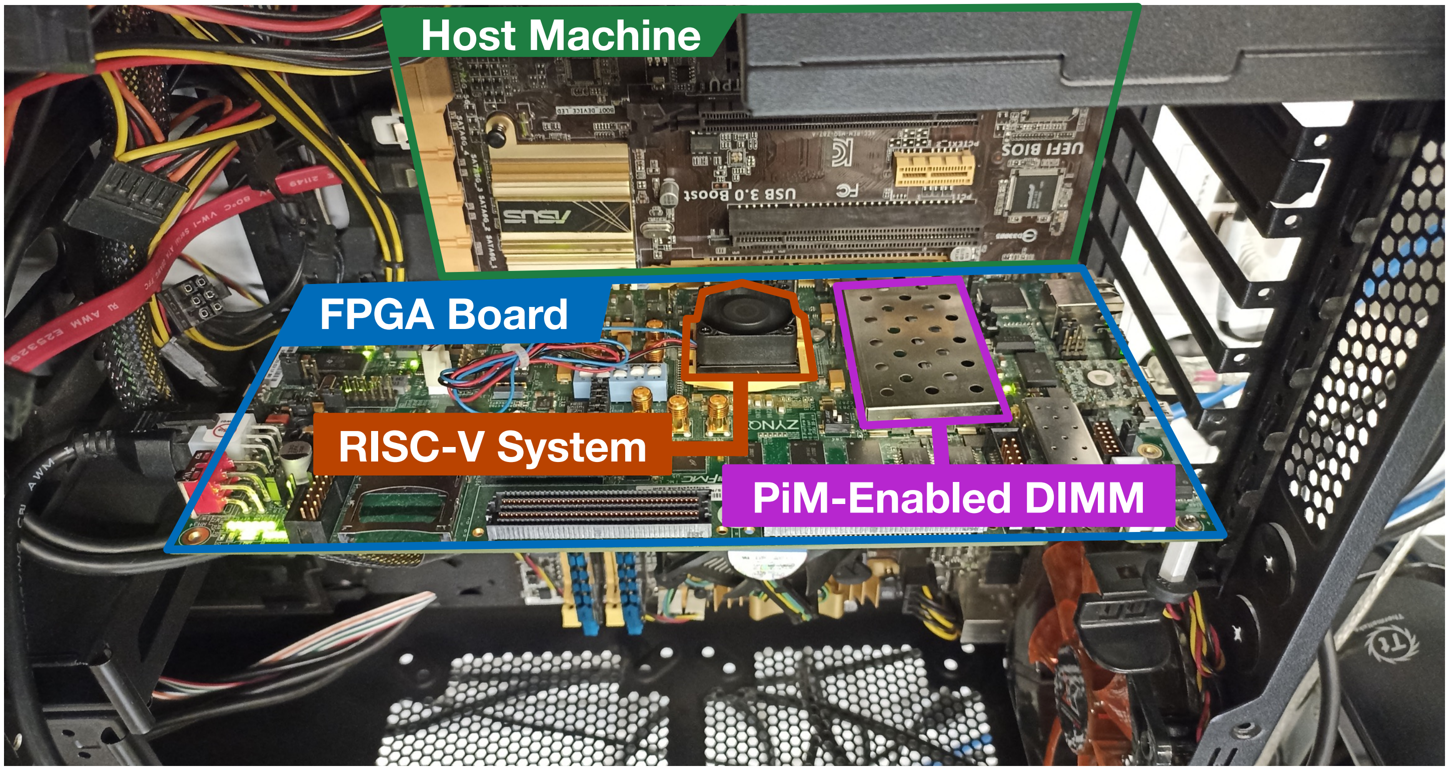
[§]ETH Zürich [†]TOBB ETÜ [*]BSC

https://arxiv.org/pdf/2111.00082.pdf

https://github.com/cmu-safari/pidram

https://www.youtube.com/watch?v=qeukNs5XI3g&t=4192s

Real Processing-using-Memory Prototype

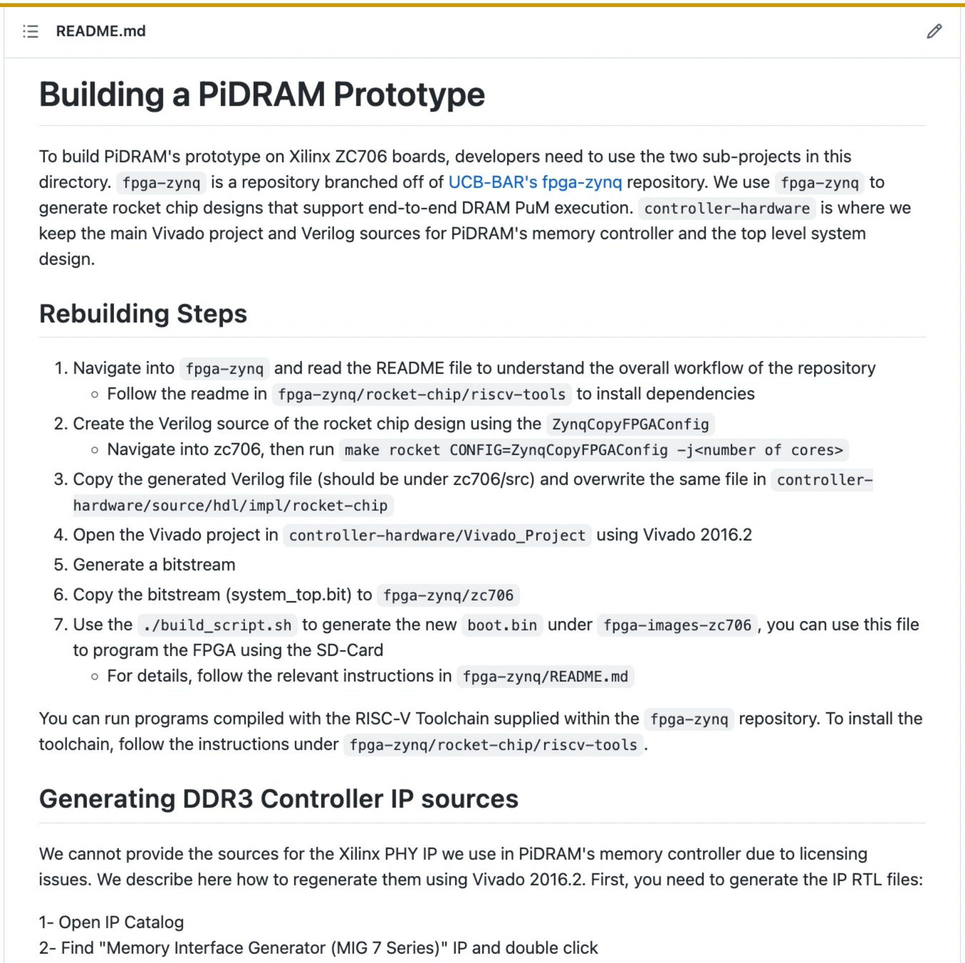


https://arxiv.org/pdf/2111.00082.pdf

https://github.com/cmu-safari/pidram

https://www.youtube.com/watch?v=qeukNs5XI3g&t=4192s

Real Processing-using-Memory Prototype



README.md

Building a PiDRAM Prototype

To build PiDRAM's prototype on Xilinx ZC706 boards, developers need to use the two sub-projects in this directory. `fpga-zynq` is a repository branched off of UCB-BAR's fpga-zynq repository. We use `fpga-zynq` to generate rocket chip designs that support end-to-end DRAM PuM execution. `controller-hardware` is where we keep the main Vivado project and Verilog sources for PiDRAM's memory controller and the top level system design.

Rebuilding Steps

1. Navigate into `fpga-zynq` and read the README file to understand the overall workflow of the repository
   - Follow the readme in `fpga-zynq/rocket-chip/riscv-tools` to install dependencies
2. Create the Verilog source of the rocket chip design using the `ZynqCopyFPGAConfig`
   - Navigate into zc706, then run `make rocket CONFIG=ZynqCopyFPGAConfig -j<number of cores>`
3. Copy the generated Verilog file (should be under zc706/src) and overwrite the same file in `controller-hardware/source/hdl/impl/rocket-chip`
4. Open the Vivado project in `controller-hardware/Vivado_Project` using Vivado 2016.2
5. Generate a bitstream
6. Copy the bitstream (system_top.bit) to `fpga-zynq/zc706`
7. Use the `./build_script.sh` to generate the new `boot.bin` under `fpga-images-zc706`, you can use this file to program the FPGA using the SD-Card
   - For details, follow the relevant instructions in `fpga-zynq/README.md`

You can run programs compiled with the RISC-V Toolchain supplied within the `fpga-zynq` repository. To install the toolchain, follow the instructions under `fpga-zynq/rocket-chip/riscv-tools`.

Generating DDR3 Controller IP sources

We cannot provide the sources for the Xilinx PHY IP we use in PiDRAM's memory controller due to licensing issues. We describe here how to regenerate them using Vivado 2016.2. First, you need to generate the IP RTL files:

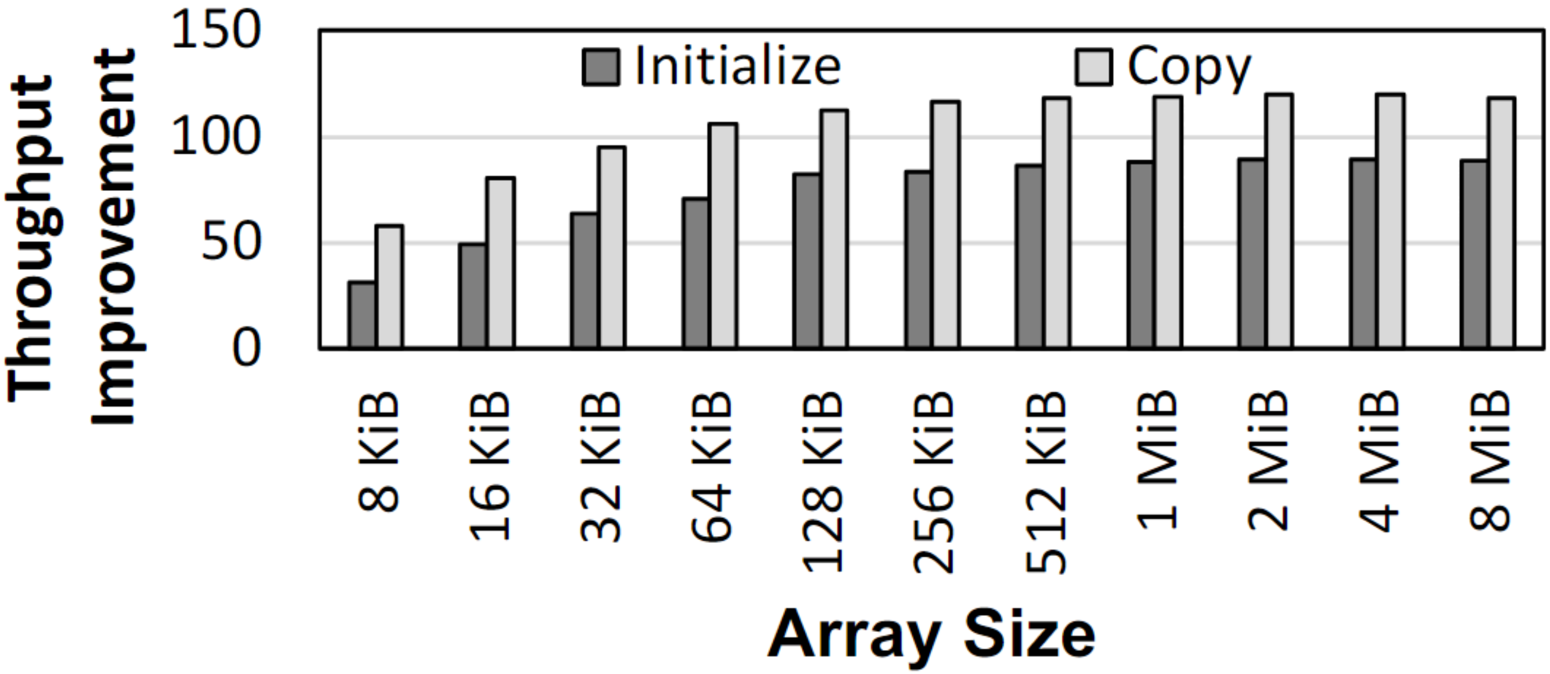
1- Open IP Catalog
2- Find "Memory Interface Generator (MIG 7 Series)" IP and double click

https://arxiv.org/pdf/2111.00082.pdf

https://github.com/cmu-safari/pidram

https://www.youtube.com/watch?v=qeukNs5XI3g&t=4192s

Microbenchmark Copy/Initialization Throughput



In-DRAM Copy and Initialization improve throughput by 119x and 89x

More on PiDRAM

- Ataberk Olgun, Juan Gomez Luna, Konstantinos Kanellopoulos, Behzad Salami, Hasan Hassan, Oguz Ergin, and Onur Mutlu,
  "PiDRAM: A Holistic End-to-end FPGA-based Framework for Processing-in-DRAM"
  ACM Transactions on Architecture and Code Optimization (***TACO***), March 2023.
  [arXiv version]
  Presented at the 18th HiPEAC Conference, Toulouse, France, January 2023.
  [Slides (pptx) (pdf)]
  [Longer Lecture Slides (pptx) (pdf)]
  [Lecture Video (40 minutes)]
  [PiDRAM Source Code]

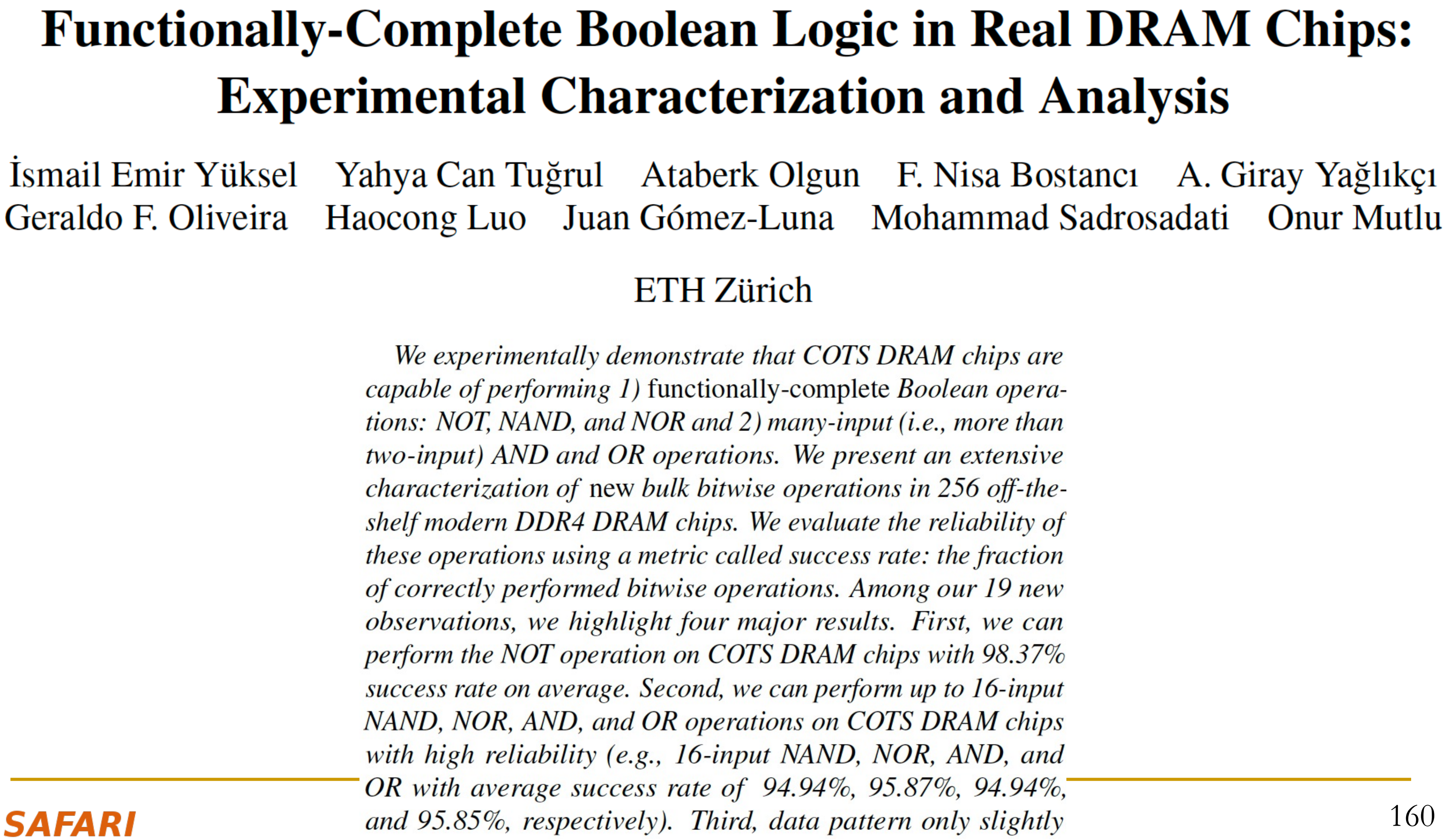
PiDRAM: A Holistic End-to-end FPGA-based Framework for Processing-in-DRAM

Ataberk Olgun[§] Juan Gómez Luna[§] Konstantinos Kanellopoulos[§] Behzad Salami[§]
Hasan Hassan[§] Oğuz Ergin[†] Onur Mutlu[§]

[§]*ETH Zürich* [†]*TOBB University of Economics and Technology*

DRAM Chips Are Already (Quite) Capable!

- **Appears at HPCA 2024** https://arxiv.org/pdf/2402.18736.pdf



Functionally-Complete Boolean Logic in Real DRAM Chips: Experimental Characterization and Analysis

İsmail Emir Yüksel Yahya Can Tuğrul Ataberk Olgun F. Nisa Bostancı A. Giray Yağlıkçı
Geraldo F. Oliveira Haocong Luo Juan Gómez-Luna Mohammad Sadrosadati Onur Mutlu

ETH Zürich

We experimentally demonstrate that COTS DRAM chips are capable of performing 1) functionally-complete *Boolean operations: NOT, NAND, and NOR and 2) many-input (i.e., more than two-input) AND and OR operations. We present an extensive characterization of* new *bulk bitwise operations in 256 off-the-shelf modern DDR4 DRAM chips. We evaluate the reliability of these operations using a metric called success rate: the fraction of correctly performed bitwise operations. Among our 19 new observations, we highlight four major results. First, we can perform the NOT operation on COTS DRAM chips with 98.37% success rate on average. Second, we can perform up to 16-input NAND, NOR, AND, and OR operations on COTS DRAM chips with high reliability (e.g., 16-input NAND, NOR, AND, and OR with average success rate of 94.94%, 95.87%, 94.94%, and 95.85%, respectively). Third, data pattern only slightly*

The Capability of COTS DRAM Chips

We **demonstrate** that **COTS DRAM chips:**

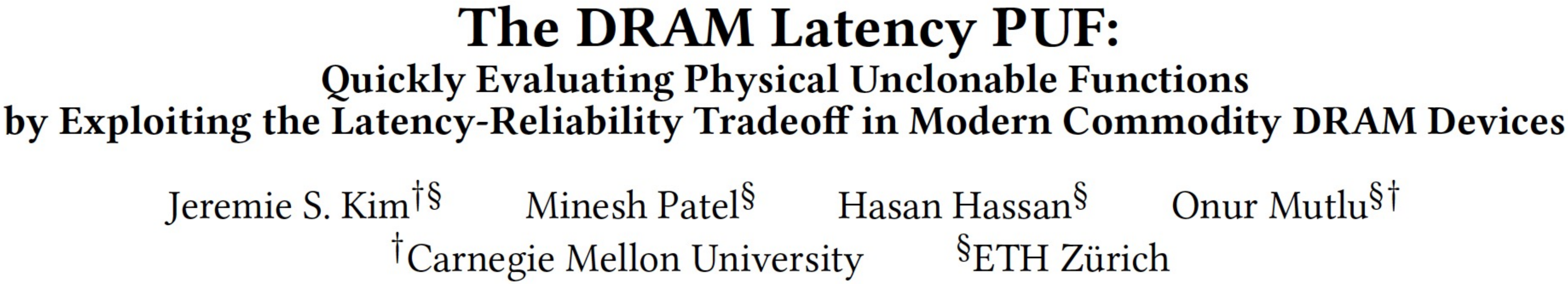
1 Can **copy one row into up to 31 other rows with >99.98% success rate**

2 Can perform **NOT operation** with up to **32 output operands**

3 Can perform up to **16-input AND, NAND, OR, and NOR** operations

In-DRAM Physical Unclonable Functions

- Jeremie S. Kim, Minesh Patel, Hasan Hassan, and Onur Mutlu,
  "The DRAM Latency PUF: Quickly Evaluating Physical Unclonable Functions by Exploiting the Latency-Reliability Tradeoff in Modern DRAM Devices"
  *Proceedings of the 24th International Symposium on High-Performance Computer Architecture (**HPCA**)*, Vienna, Austria, February 2018.
  [Lightning Talk Video]
  [Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)]
  [Full Talk Lecture Video (28 minutes)]

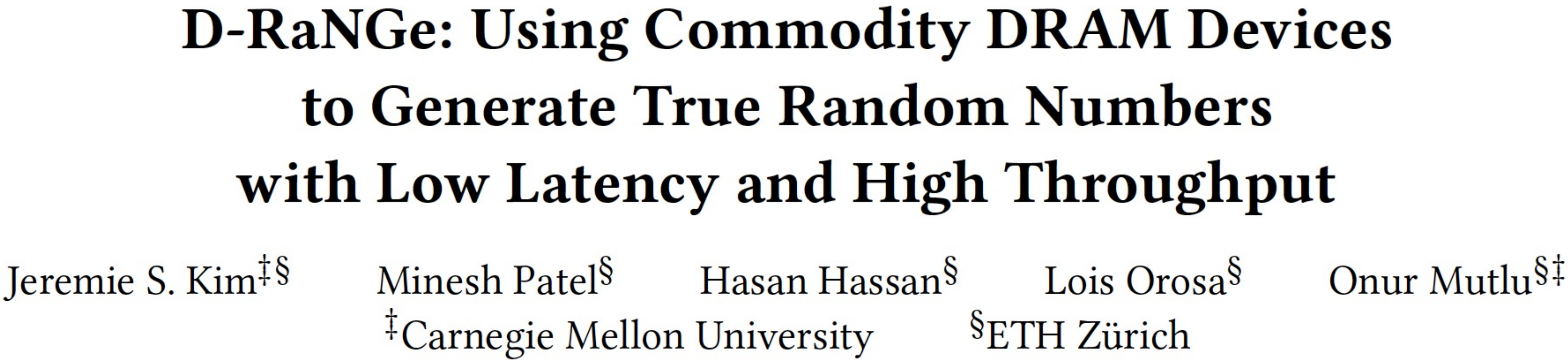


The DRAM Latency PUF:
Quickly Evaluating Physical Unclonable Functions by Exploiting the Latency-Reliability Tradeoff in Modern Commodity DRAM Devices

Jeremie S. Kim[†§] Minesh Patel[§] Hasan Hassan[§] Onur Mutlu[§†]
[†]Carnegie Mellon University [§]ETH Zürich

In-DRAM True Random Number Generation

- Jeremie S. Kim, Minesh Patel, Hasan Hassan, Lois Orosa, and Onur Mutlu, **"D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput"**
  *Proceedings of the 25th International Symposium on High-Performance Computer Architecture (**HPCA**)*, Washington, DC, USA, February 2019.
  [Slides (pptx) (pdf)]
  [Full Talk Video (21 minutes)]
  [Full Talk Lecture Video (27 minutes)]
  Top Picks Honorable Mention by IEEE Micro.

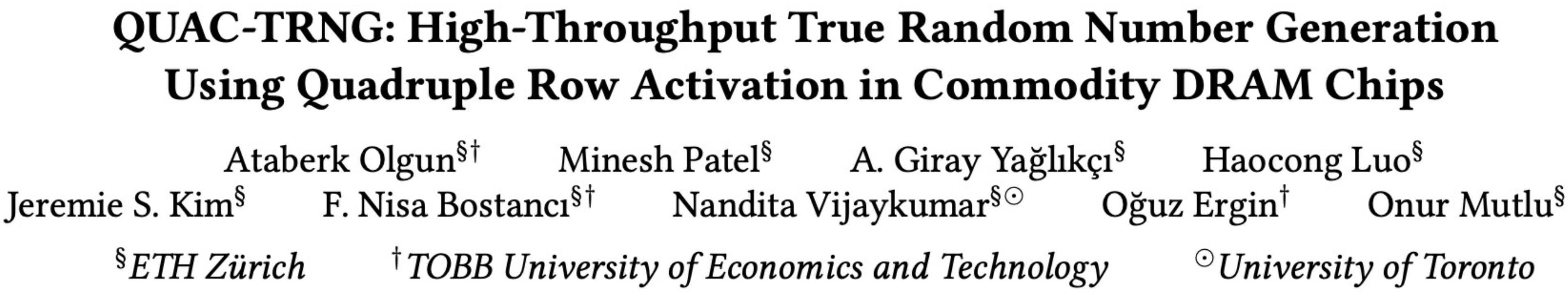


D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput

Jeremie S. Kim[‡§] Minesh Patel[§] Hasan Hassan[§] Lois Orosa[§] Onur Mutlu[§‡]
[‡]Carnegie Mellon University [§]ETH Zürich

In-DRAM True Random Number Generation

- Ataberk Olgun, Minesh Patel, A. Giray Yaglikci, Haocong Luo, Jeremie S. Kim, F. Nisa Bostanci, Nandita Vijaykumar, Oguz Ergin, and Onur Mutlu,
  "QUAC-TRNG: High-Throughput True Random Number Generation Using Quadruple Row Activation in Commodity DRAM Chips"
  Proceedings of the 48th International Symposium on Computer Architecture (***ISCA***), Virtual, June 2021.
  [Slides (pptx) (pdf)]
  [Short Talk Slides (pptx) (pdf)]
  [Talk Video (25 minutes)]
  [SAFARI Live Seminar Video (1 hr 26 mins)]



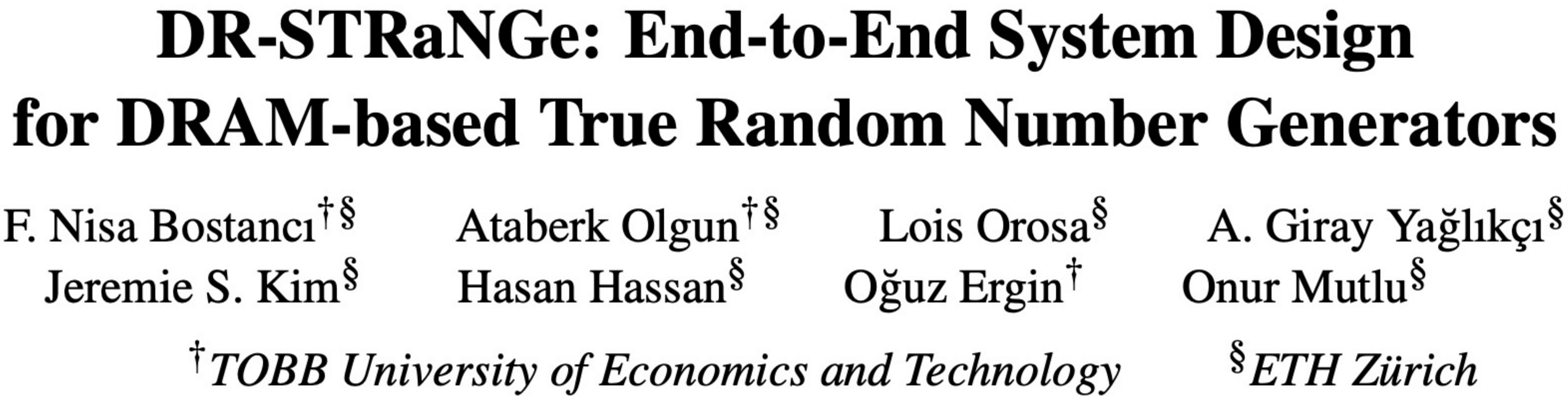
QUAC-TRNG: High-Throughput True Random Number Generation Using Quadruple Row Activation in Commodity DRAM Chips

Ataberk Olgun[§†] Minesh Patel[§] A. Giray Yağlıkçı[§] Haocong Luo[§]
Jeremie S. Kim[§] F. Nisa Bostancı[§†] Nandita Vijaykumar[§⊙] Oğuz Ergin[†] Onur Mutlu[§]

[§] *ETH Zürich* [†] *TOBB University of Economics and Technology* [⊙] *University of Toronto*

In-DRAM True Random Number Generation

- F. Nisa Bostanci, Ataberk Olgun, Lois Orosa, A. Giray Yaglikci, Jeremie S. Kim, Hasan Hassan, Oguz Ergin, and Onur Mutlu,
  "DR-STRaNGe: End-to-End System Design for DRAM-based True Random Number Generators"
  Proceedings of the 28th International Symposium on High-Performance Computer Architecture (***HPCA***), Virtual, April 2022.
  [Slides (pptx) (pdf)]
  [Short Talk Slides (pptx) (pdf)]



DR-STRaNGe: End-to-End System Design for DRAM-based True Random Number Generators

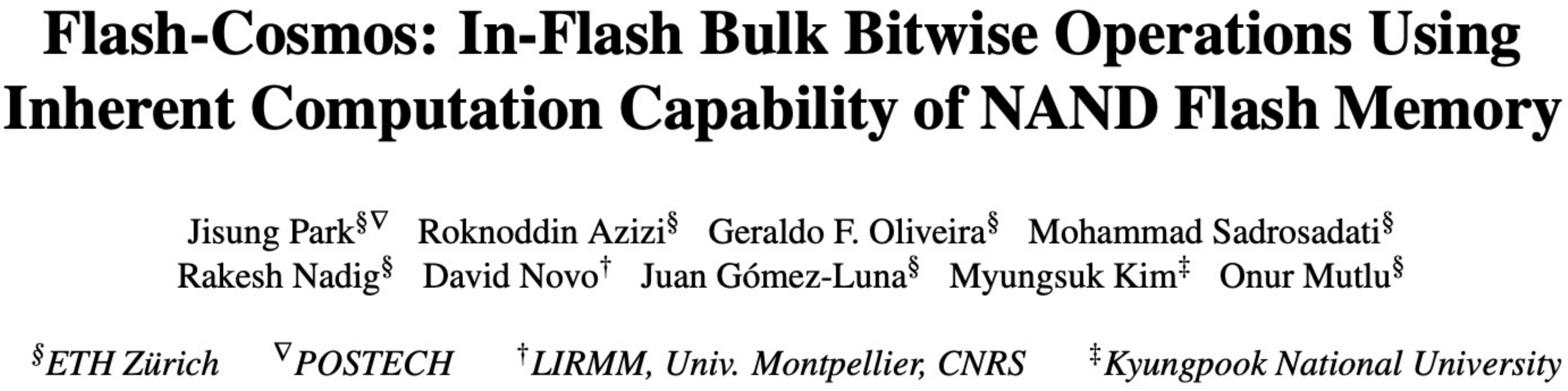
F. Nisa Bostancı[†][§] Ataberk Olgun[†][§] Lois Orosa[§] A. Giray Yağlıkçı[§]
Jeremie S. Kim[§] Hasan Hassan[§] Oğuz Ergin[†] Onur Mutlu[§]

[†] *TOBB University of Economics and Technology* [§] *ETH Zürich*

https://arxiv.org/pdf/2201.01385.pdf

In-Flash Bulk Bitwise Execution

- Jisung Park, Roknoddin Azizi, Geraldo F. Oliveira, Mohammad Sadrosadati, Rakesh Nadig, David Novo, Juan Gómez-Luna, Myungsuk Kim, and Onur Mutlu, **"Flash-Cosmos: In-Flash Bulk Bitwise Operations Using Inherent Computation Capability of NAND Flash Memory"**
  Proceedings of the 55th International Symposium on Microarchitecture (***MICRO***), Chicago, IL, USA, October 2022.
  [Slides (pptx) (pdf)]
  [Longer Lecture Slides (pptx) (pdf)]
  [Lecture Video (44 minutes)]
  [arXiv version]



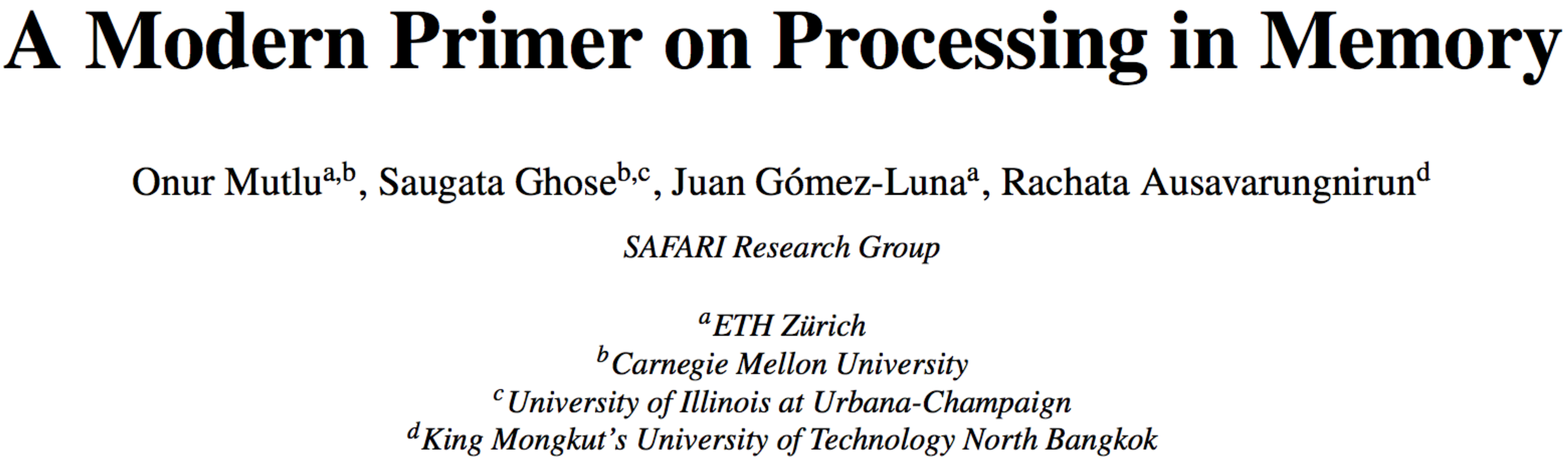
Flash-Cosmos: In-Flash Bulk Bitwise Operations Using Inherent Computation Capability of NAND Flash Memory

Jisung Park[§∇] Roknoddin Azizi[§] Geraldo F. Oliveira[§] Mohammad Sadrosadati[§]
Rakesh Nadig[§] David Novo[†] Juan Gómez-Luna[§] Myungsuk Kim[‡] Onur Mutlu[§]

[§]*ETH Zürich* [∇]*POSTECH* [†]*LIRMM, Univ. Montpellier, CNRS* [‡]*Kyungpook National University*

SAFARI **https://arxiv.org/pdf/2209.05566.pdf**

PIM Review and Open Problems



A Modern Primer on Processing in Memory

Onur Mutlu[a,b], Saugata Ghose[b,c], Juan Gómez-Luna[a], Rachata Ausavarungnirun[d]

SAFARI Research Group

[a] *ETH Zürich*
[b] *Carnegie Mellon University*
[c] *University of Illinois at Urbana-Champaign*
[d] *King Mongkut's University of Technology North Bangkok*

Onur Mutlu, Saugata Ghose, Juan Gomez-Luna, and Rachata Ausavarungnirun,
"A Modern Primer on Processing in Memory"
Invited Book Chapter in ***Emerging Computing: From Devices to Systems - Looking Beyond Moore and Von Neumann***, Springer, to be published in 2021.

https://arxiv.org/pdf/2012.03112.pdf

Eliminating the Adoption Barriers

How to Enable Adoption of Processing in Memory

Potential Barriers to Adoption of PIM

1. **Applications** & **software** for PIM

2. Ease of **programming** (interfaces and compiler/HW support)

3. **System** and **security** support: coherence, synchronization, virtual memory, isolation, communication interfaces, …

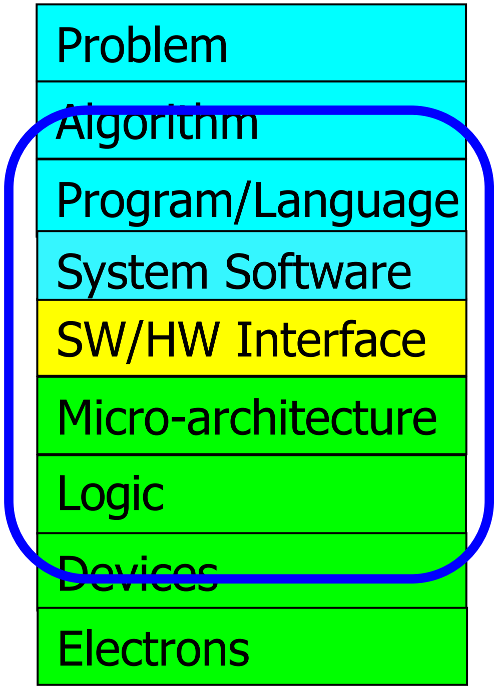
4. **Runtime** and **compilation** systems for adaptive scheduling, data mapping, access/sharing control, …

5. **Infrastructures** to assess benefits and feasibility

All can be solved with change of mindset

We Need to Revisit the Entire Stack

- With a **memory-centric mindset**



Problem

Algorithm

Program/Language

System Software

SW/HW Interface

Micro-architecture

Logic

Devices

Electrons

We can get there step by step

Security Issues in Processing in Memory

- Does PIM make security better or easier?

- Does PIM make security worse?

- Many interesting questions here

- Some recent papers:
  - Evaluating Homomorphic Operations on a Real-World Processing-In-Memory System **[IISWC 2023]**
  - Amplifying Main Memory-Based Timing Covert and Side Channels using Processing-in-Memory Operations **[arxiv 2024]**

Potential Security Issues & Benefits (I)

- **Can PIM worsen security?**
  - Worsened or easier-to-induce physical issues (e.g., RowHammer)?
  - Worsened or new side channels?
  - Hardware bugs?
  - New threat models?
  - …

- **Can PIM enhance security?**
  - Less exposure of data (& keys?)
  - In-memory (homomorphic) encryption & cryptographic hashing
  - Execution of security functions; trusted execution in memory
  - Support for security primitives (TRNGs, PUFs, encryption, …)
  - More or better isolation, virtualization, containerization?
  - …

Potential Security Issues & Benefits (II)

- Security analysis of PIM Systems
  - Different types of PIM: PnM vs. PuM
  - Different locations: cache, MC, DRAM, NVM, storage, remote, …
  - General-purpose vs. special-purpose PIM?
  - Multi tenancy vs. single workload?
  - Concurrent host and PIM access?
  - Memory bus protection; memory wire(s) protection?
  - Robustness issues like RowHammer, RowPress, …
  - …

- Can PIM support (more) secure execution of workloads?
  - What is needed to do so?
  - Secure PIM enclaves?
  - …

PIM Helps Security: Many Examples

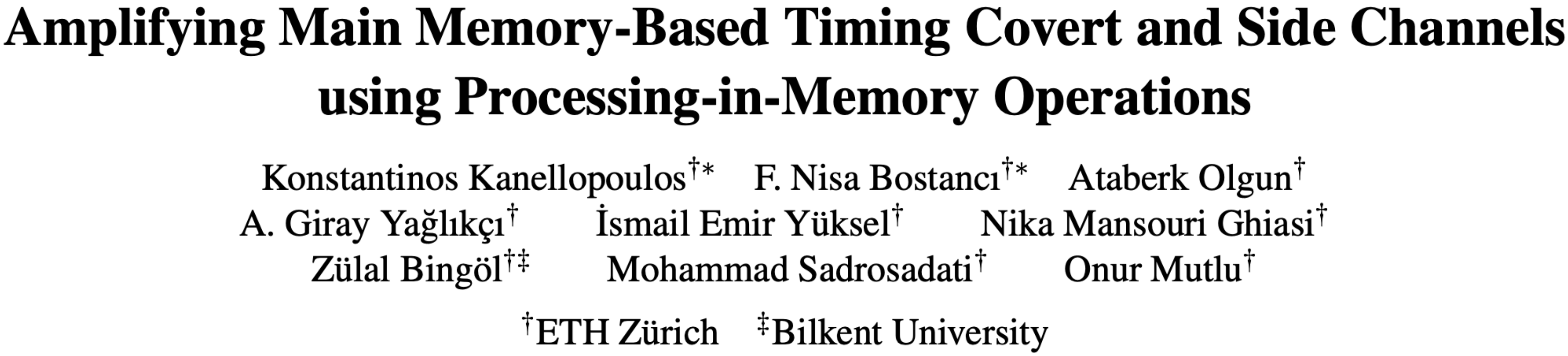
Harshita Gupta, Mayank Kabra, Juan Gómez-Luna, Konstantinos Kanellopoulos, and Onur Mutlu,
"Evaluating Homomorphic Operations on a Real-World Processing-In-Memory System"
*Proceedings of the 2023 IEEE International Symposium on Workload Characterization Poster Session (**IISWC**)*, Ghent, Belgium, October 2023.
[arXiv version]
[Lightning Talk Slides (pptx) (pdf)]
[Poster (pptx) (pdf)]

Evaluating Homomorphic Operations on a Real-World Processing-In-Memory System

Harshita Gupta* Mayank Kabra* Juan Gómez-Luna Konstantinos Kanellopoulos Onur Mutlu
ETH Zürich

https://arxiv.org/pdf/2309.06545

PIM Worsens Side Channels: An Example



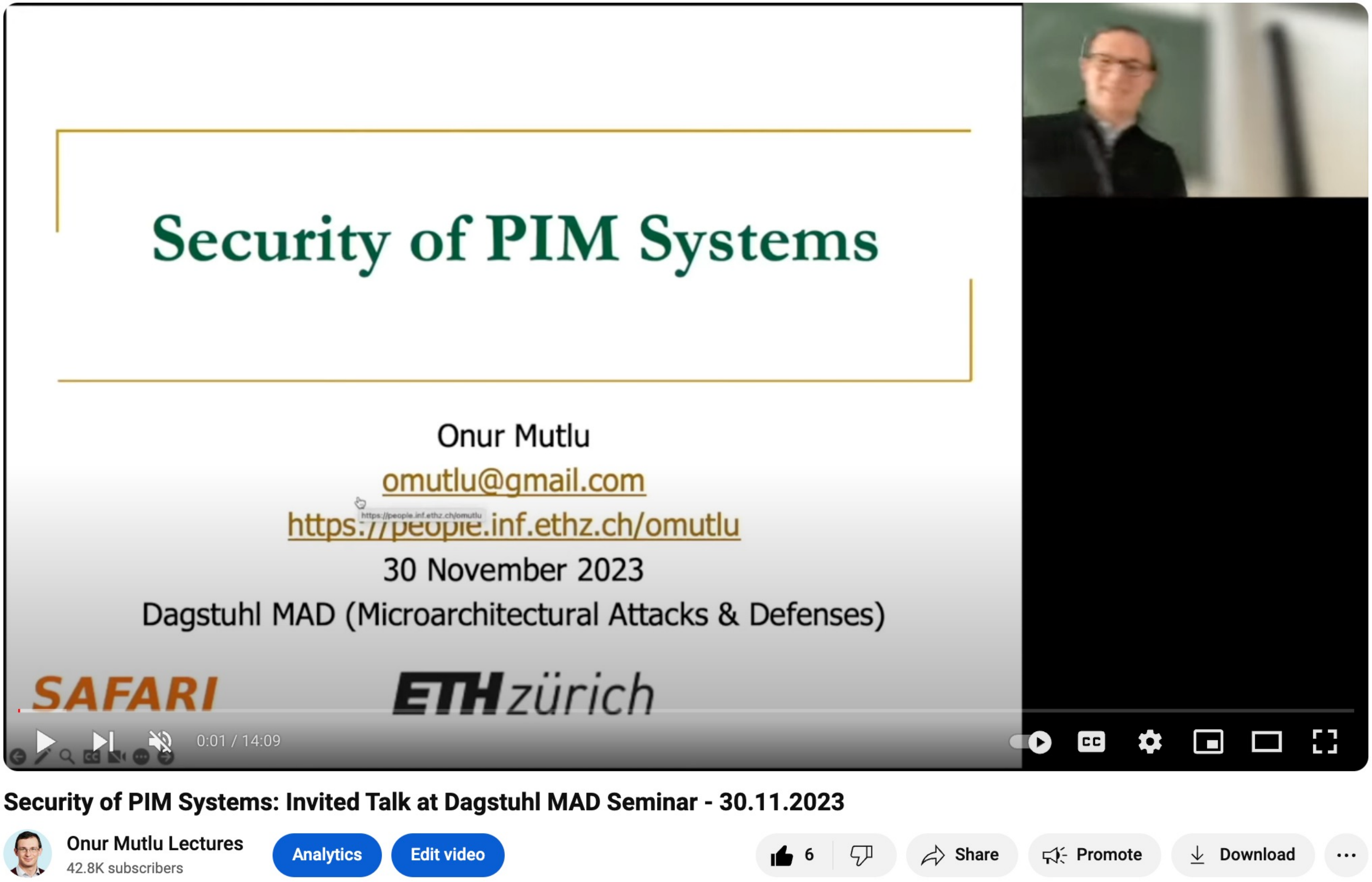
Amplifying Main Memory-Based Timing Covert and Side Channels using Processing-in-Memory Operations

Konstantinos Kanellopoulos[†*] F. Nisa Bostancı[†*] Ataberk Olgun[†]
A. Giray Yağlıkçı[†] İsmail Emir Yüksel[†] Nika Mansouri Ghiasi[†]
Zülal Bingöl[†‡] Mohammad Sadrosadati[†] Onur Mutlu[†]

[†]ETH Zürich [‡]Bilkent University

https://arxiv.org/pdf/2404.11284

A Short Talk on Security of PIM Systems



Security of PIM Systems

Onur Mutlu

omutlu@gmail.com

https://people.inf.ethz.ch/omutlu

30 November 2023

Dagstuhl MAD (Microarchitectural Attacks & Defenses)

SAFARI ETHzürich

Security of PIM Systems: Invited Talk at Dagstuhl MAD Seminar - 30.11.2023

Onur Mutlu Lectures

42.8K subscribers

https://www.youtube.com/watch?v=UjE9hygFXEM

Concluding Remarks

Summary: Three Major Limiters

- Technology scaling is not going well
- System complexity is increasing; old methods not keeping up
- Processor-centric designs are not keeping up
- These affect all metrics we care about
- These have fundamental impact on security and how we build secure systems
- **We need to revisit how we build architectures and how we secure them**

Funding Acknowledgments

- Alibaba, AMD, ASML, Google, Facebook, Hi-Silicon, HP Labs, Huawei, IBM, Intel, Microsoft, Nvidia, Oracle, Qualcomm, Rambus, Samsung, Seagate, VMware, Xilinx
- NSF
- NIH
- GSRC
- SRC
- CyLab
- EFCL
- SNSF
- ACCESS

Thank you!

Acknowledgments

SAFARI

SAFARI Research Group

safari.ethz.ch

Think BIG, Aim HIGH!

https://safari.ethz.ch

SAFARI Newsletter June 2023 Edition

- https://safari.ethz.ch/safari-newsletter-june-2023/

SAFARI
SAFARI Research Group

Think Big, Aim High

ETH zürich

View in your browser
June 2023

SAFARI Newsletter July 2024 Edition

- https://safari.ethz.ch/safari-newsletter-july-2024/

Referenced Papers, Talks, Artifacts

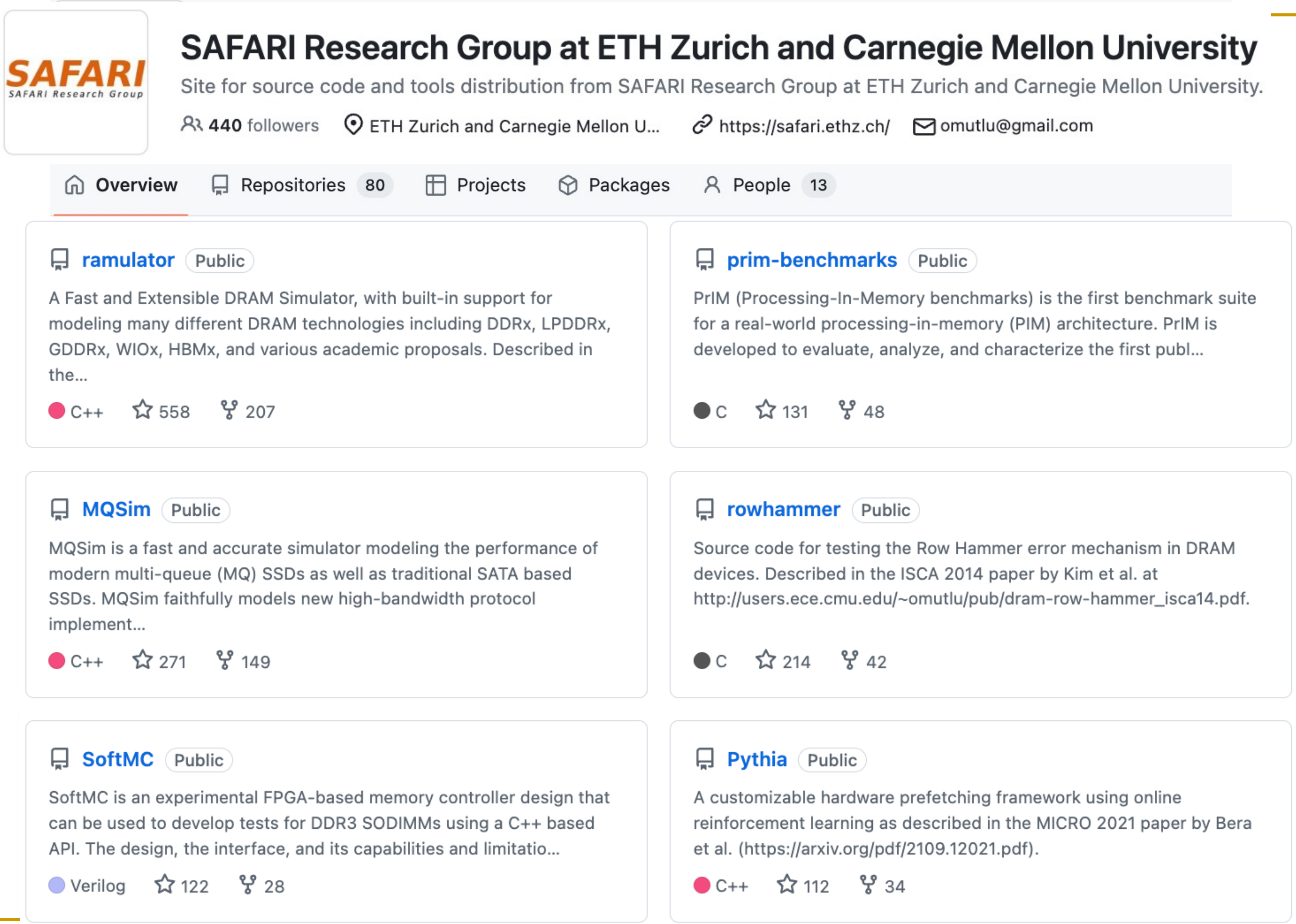
- All are available at

  https://people.inf.ethz.ch/omutlu/projects.htm

  https://www.youtube.com/onurmutlulectures

  https://github.com/CMU-SAFARI/

Open Source Tools: SAFARI GitHub



SAFARI Research Group at ETH Zurich and Carnegie Mellon University

Site for source code and tools distribution from SAFARI Research Group at ETH Zurich and Carnegie Mellon University.

440 followers · ETH Zurich and Carnegie Mellon U... · https://safari.ethz.ch/ · omutlu@gmail.com

ramulator Public

A Fast and Extensible DRAM Simulator, with built-in support for modeling many different DRAM technologies including DDRx, LPDDRx, GDDRx, WIOx, HBMx, and various academic proposals. Described in the...

● C++ ☆ 558 207

prim-benchmarks Public

PrIM (Processing-In-Memory benchmarks) is the first benchmark suite for a real-world processing-in-memory (PIM) architecture. PrIM is developed to evaluate, analyze, and characterize the first publ...

● C ☆ 131 48

MQSim Public

MQSim is a fast and accurate simulator modeling the performance of modern multi-queue (MQ) SSDs as well as traditional SATA based SSDs. MQSim faithfully models new high-bandwidth protocol implement...

● C++ ☆ 271 149

rowhammer Public

Source code for testing the Row Hammer error mechanism in DRAM devices. Described in the ISCA 2014 paper by Kim et al. at http://users.ece.cmu.edu/~omutlu/pub/dram-row-hammer_isca14.pdf.

● C ☆ 214 42

SoftMC Public

SoftMC is an experimental FPGA-based memory controller design that can be used to develop tests for DDR3 SODIMMs using a C++ based API. The design, the interface, and its capabilities and limitatio...

● Verilog ☆ 122 28

Pythia Public

A customizable hardware prefetching framework using online reinforcement learning as described in the MICRO 2021 paper by Bera et al. (https://arxiv.org/pdf/2109.12021.pdf).

● C++ ☆ 112 34

https://github.com/CMU-SAFARI/

Future of Computer Architecture and Hardware Security

Onur Mutlu

omutlu@gmail.com

https://people.inf.ethz.ch/omutlu

25 October 2024

Hardwear.io MemSec Keynote Talk

SAFARI ETH zürich Carnegie Mellon

Backup Slides – Longer Version

How Do We Make Sure Solution is Good?

- Many challenges (some below)
- Security by obscurity (as done in JEDEC DDR5 spec) unhelpful

- How do we guarantee we use **correct thresholds**?
  - Determining RH threshold is not easy
  - Many factors & conditions: RowPress, temperature, voltage, spatial variation, aging, voltage, and the unknowns

- How do we guarantee the **correct mitigating actions**?
  - In the presence of blast radius, address remapping, …

- How do we guarantee we perform **accurate bookkeeping**?
  - Updating counters properly? Row open time? When to reset? Worst-case access patterns?, …

Good at What Cost?

- Even if we **magically assume** the solution prevents bitflips…

- Does the problem turn into a memory performance attack?

- How do we avoid large performance and energy losses?

- Is there a smarter way of handling things more holistically?
  - Better partitioning of responsibilities between CPU and memory
  - Our memory interface is terrible
    - DRAM has "no real freedom" to do things internally
    - DRAM should be more self-managing – we need more open minds

What is RowPress?

Keeping a DRAM row **open for a long time** causes bitflips in adjacent rows

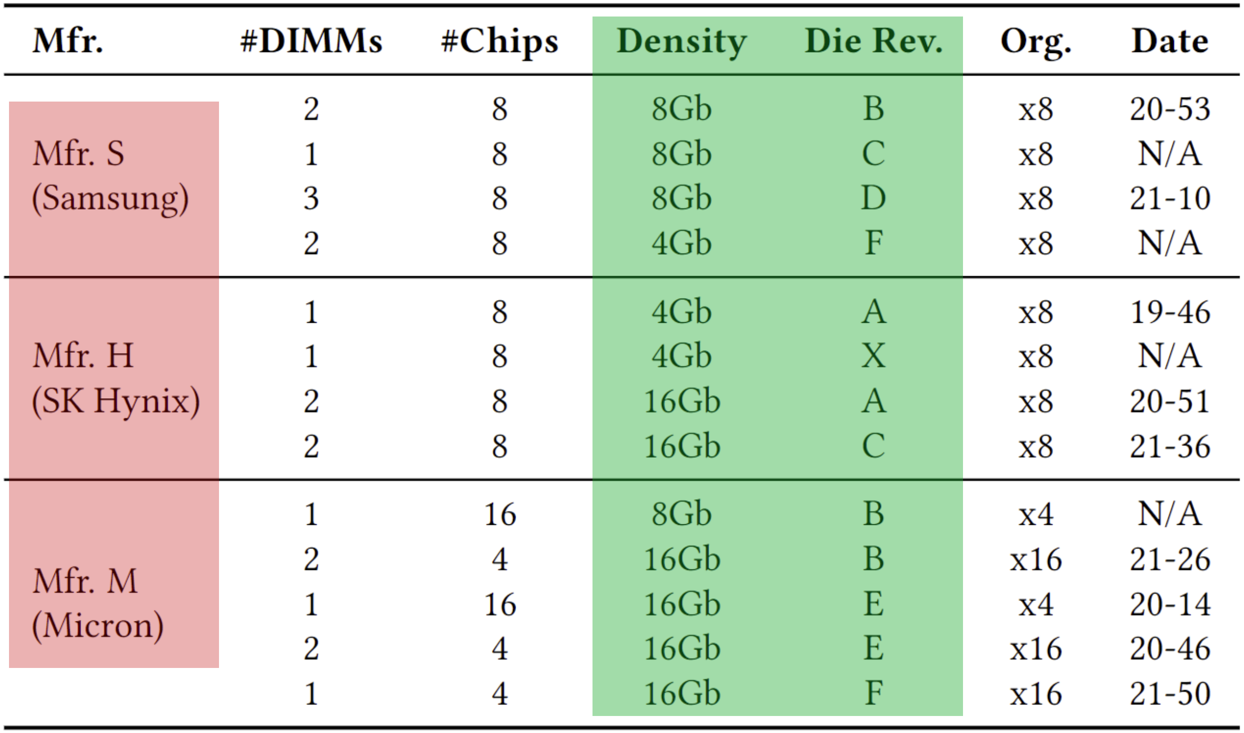
These bitflips do **NOT** require many row activations

Only one activation is enough in some cases!

Real DRAM Chip Characterization (II)

DRAM chips tested

- 164 DDR4 chips from all 3 major DRAM manufacturers
- Covers different die densities and revisions



Mfr.	#DIMMs	#Chips	Density	Die Rev.	Org.	Date
Mfr. S (Samsung)	2	8	8Gb	B	x8	20-53
	1	8	8Gb	C	x8	N/A
	3	8	8Gb	D	x8	21-10
	2	8	4Gb	F	x8	N/A
Mfr. H (SK Hynix)	1	8	4Gb	A	x8	19-46
	1	8	4Gb	X	x8	N/A
	2	8	16Gb	A	x8	20-51
	2	8	16Gb	C	x8	21-36
Mfr. M (Micron)	1	16	8Gb	B	x4	N/A
	2	4	16Gb	B	x16	21-26
	1	16	16Gb	E	x4	20-14
	2	4	16Gb	E	x16	20-46
	1	4	16Gb	F	x16	21-50

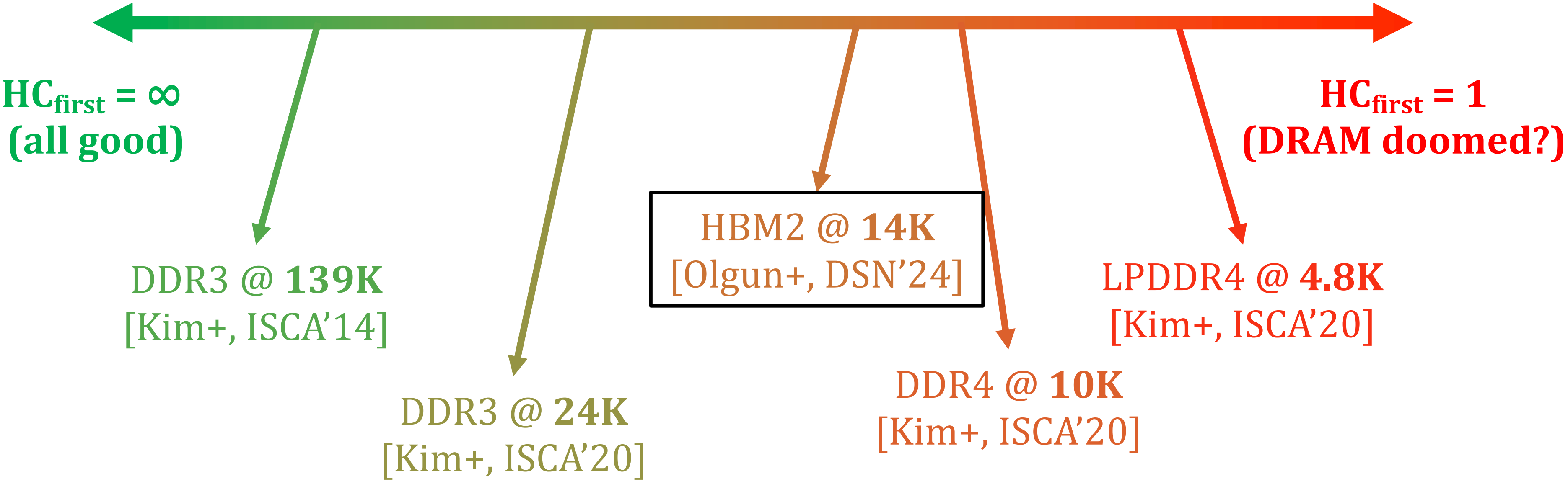
Major Takeaways from Real DRAM Chips

RowPress significantly **amplifies** DRAM's vulnerability to read disturbance

RowPress has a **different** underlying error **mechanism** from RowHammer

Reported HC$_{\text{first}}$ Values (2012 – Now)

HC$_{\text{first}}$: Number of hammers for the first bitflip



HC$_{\text{first}}$ = ∞
(all good)

DDR3 @ **139K**
[Kim+, ISCA’14]

DDR3 @ **24K**
[Kim+, ISCA’20]

HBM2 @ **14K**
[Olgun+, DSN’24]

DDR4 @ **10K**
[Kim+, ISCA’20]

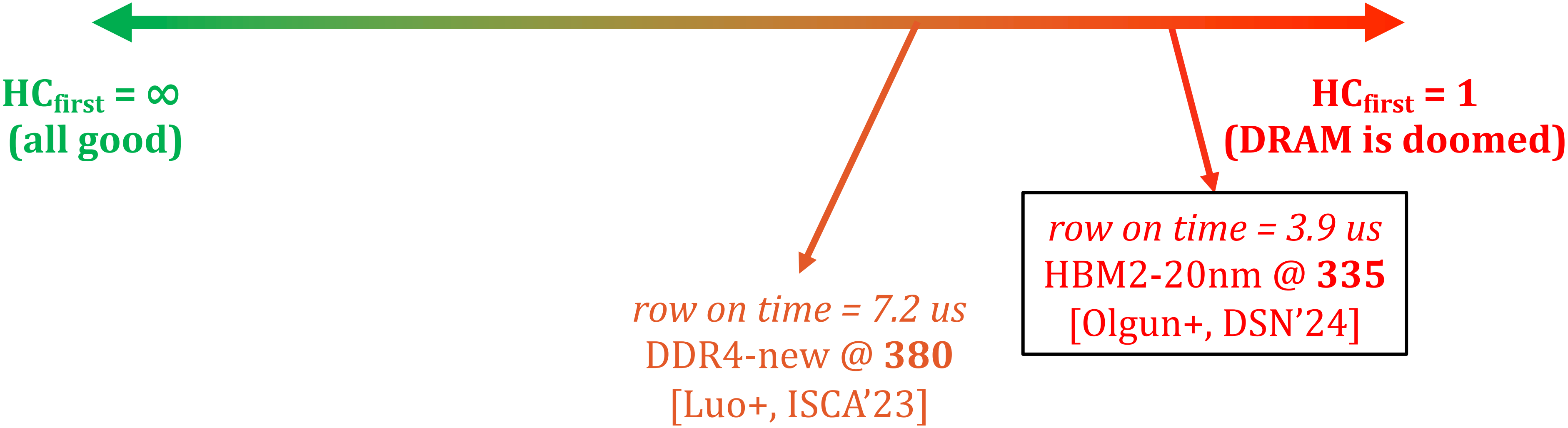
LPDDR4 @ **4.8K**
[Kim+, ISCA’20]

HC$_{\text{first}}$ = 1
(DRAM doomed?)

*Not shown: Significant variance in HC$_{\text{first}}$ across vendors and die variations

RowPress at t_{AggON} = Refresh Interval

HC_{first} : Number of hammers for the first bitflip*



HC_{first} = ∞
(all good)

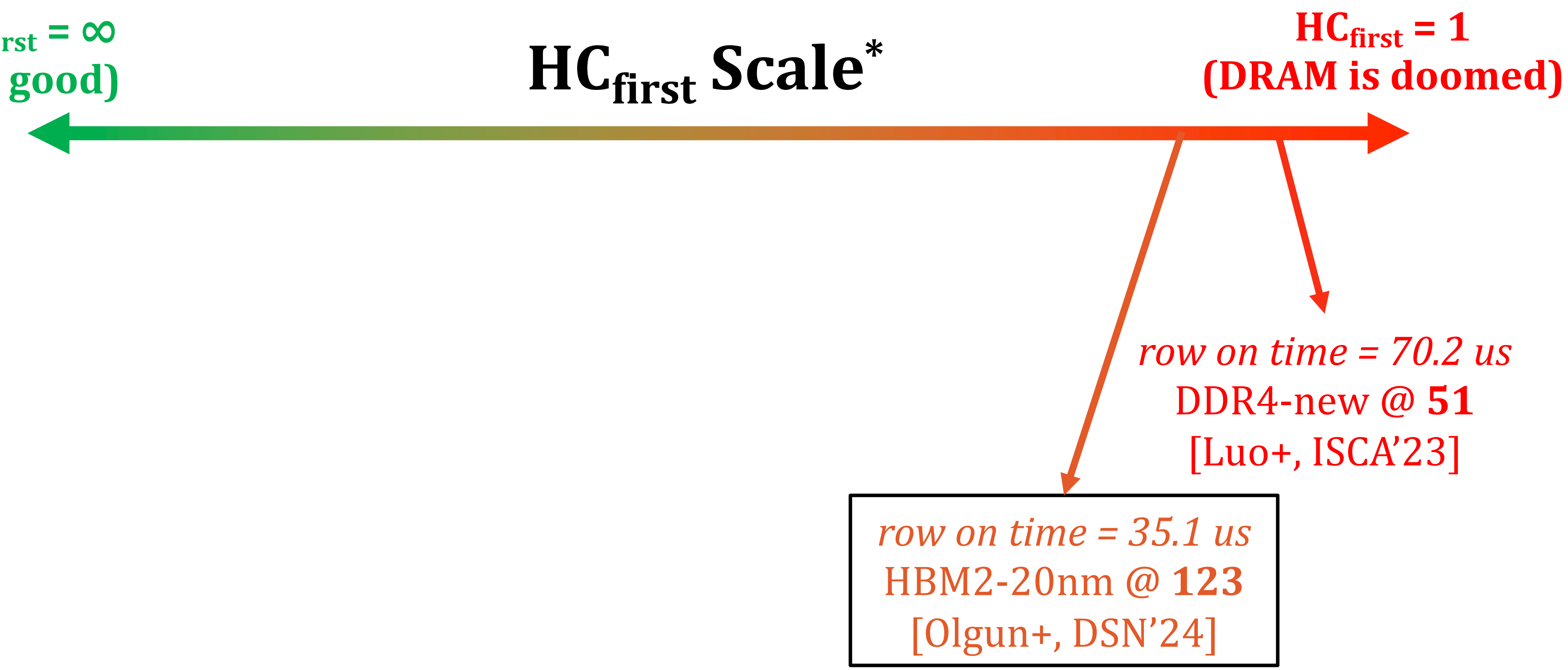
HC_{first} = 1
(DRAM is doomed)

row on time = 7.2 us
DDR4-new @ **380**
[Luo+, ISCA'23]

row on time = 3.9 us
HBM2-20nm @ **335**
[Olgun+, DSN'24]

*Not shown: Significant variance in HC_{first} across vendors and die variations

RowPress at t_{AggON} = 9 * Refresh Interval



HC_{first} = ∞
(all good)

HC_{first} Scale*

HC_{first} = 1
(DRAM is doomed)

row on time = 70.2 us
DDR4-new @ **51**
[Luo+, ISCA'23]

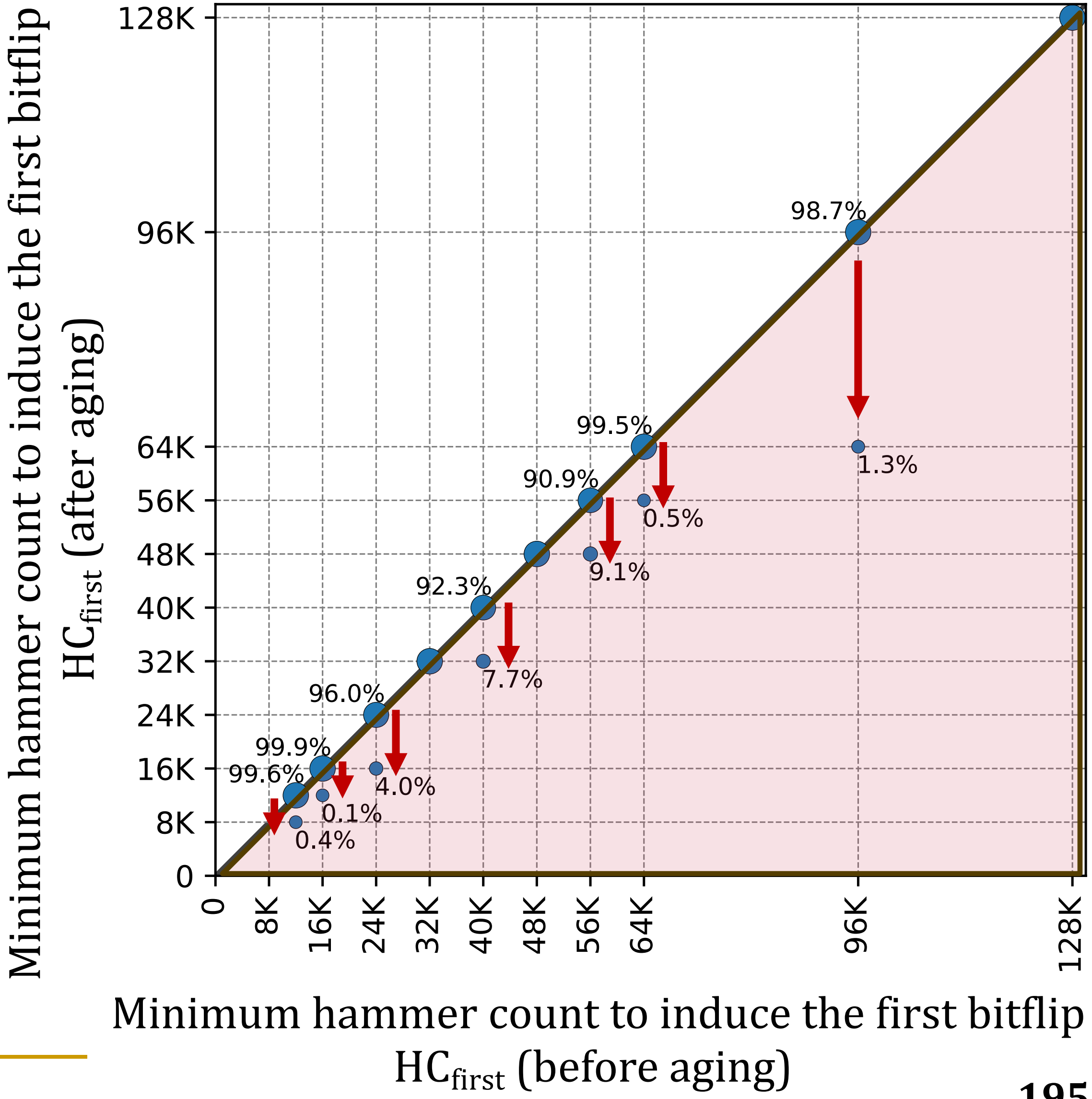
row on time = 35.1 us
HBM2-20nm @ **123**
[Olgun+, DSN'24]

*Not shown: Significant variance in HC_{first} across vendors and die variations

RowHammer Becomes Worse with Aging

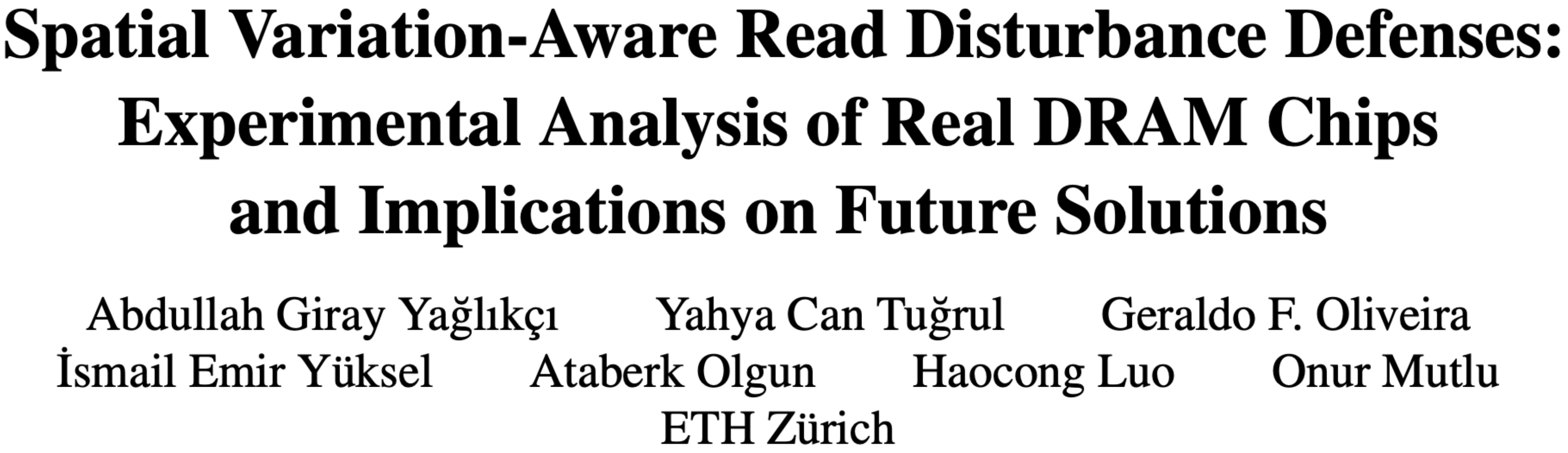
Preliminary data on aging via 68-day of continuous hammering

Aging can lead to read disturbance bitflips at **smaller** hammer counts



RowHammer (Spatial Variation) Analysis (2024)

- **Appears at HPCA 2024**



Spatial Variation-Aware Read Disturbance Defenses: Experimental Analysis of Real DRAM Chips and Implications on Future Solutions

Abdullah Giray Yağlıkçı Yahya Can Tuğrul Geraldo F. Oliveira
İsmail Emir Yüksel Ataberk Olgun Haocong Luo Onur Mutlu
ETH Zürich

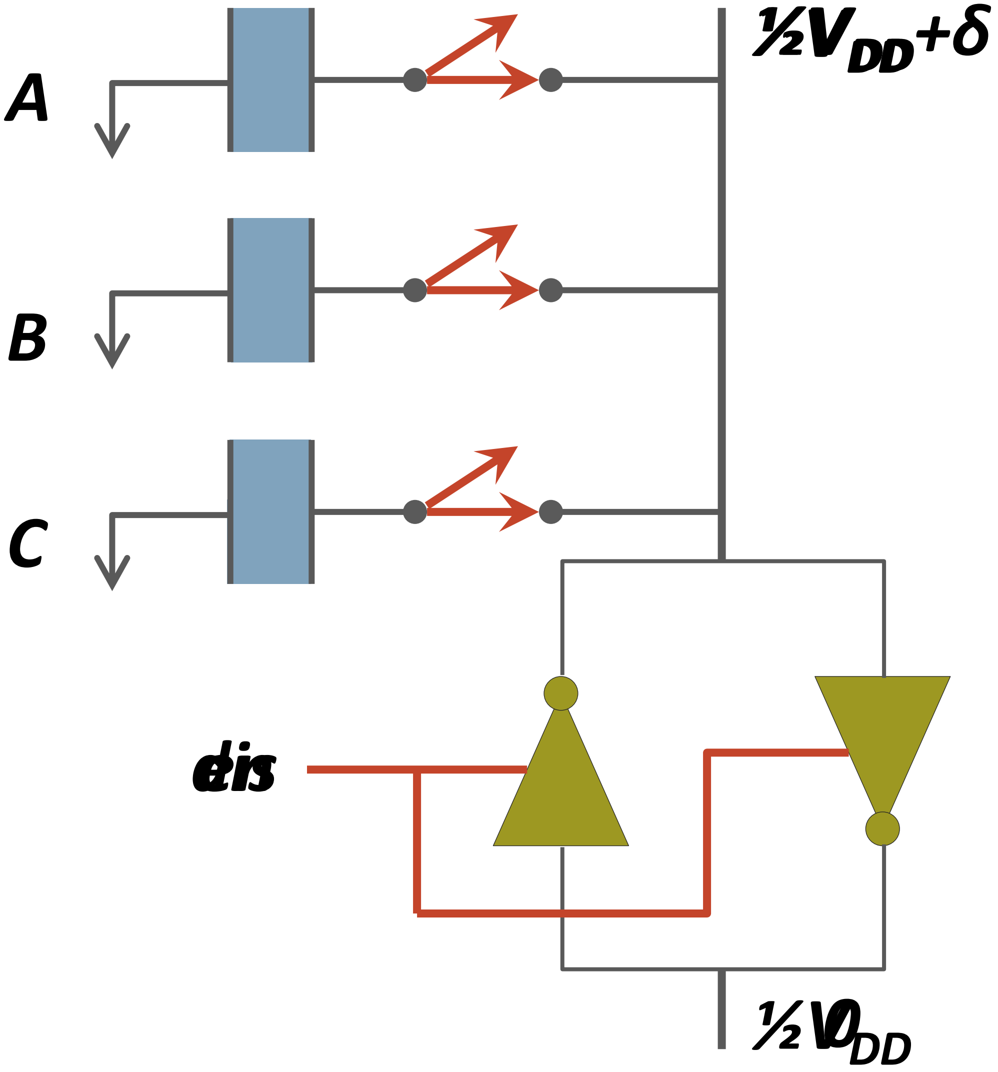
https://arxiv.org/pdf/2402.18652

Two Major Directions

- **Understanding Bitflips (Hardware errors in general)**
  - Many effects on bitflips still need to be rigorously examined
    - Aging of DRAM Chips
    - Environmental Conditions (e.g., Process, Voltage, Temperature)
    - Memory Access Patterns
    - Memory Controller & System Design Decisions
    - …

- **Solving Bitflips (Hardware errors in general)**
  - Flexible and efficient solutions are necessary
    - In-field patchable / reconfigurable / programmable solutions
  - Co-architecting across the system stack/components is important
    - To avoid performance and denial-of-service problems

In-DRAM AND/OR: Triple Row Activation



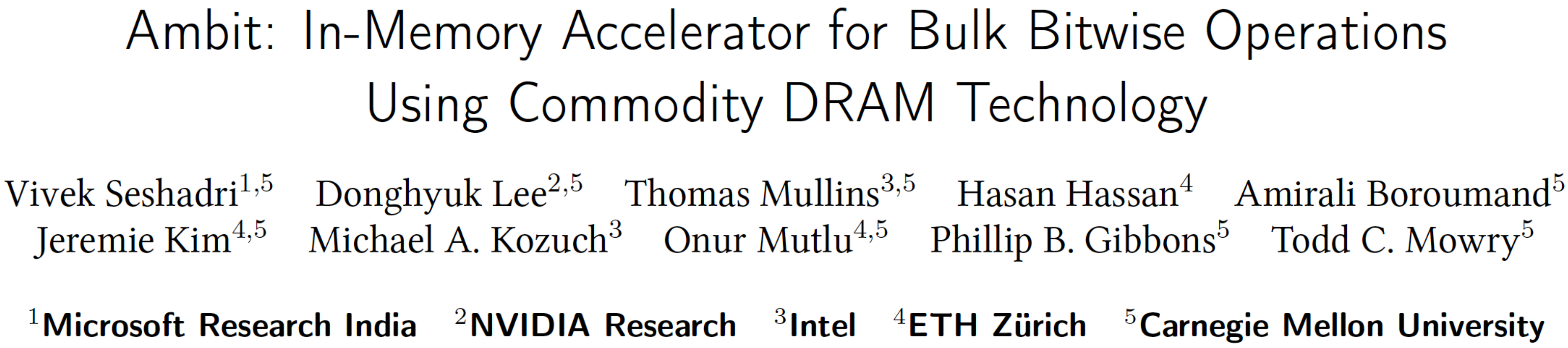
Final State
AB + BC + AC

C(A + B) +
~C(AB)

Seshadri+, "Fast Bulk Bitwise AND and OR in DRAM", IEEE CAL 2015.

More on Ambit

- Vivek Seshadri, Donghyuk Lee, Thomas Mullins, Hasan Hassan, Amirali Boroumand, Jeremie Kim, Michael A. Kozuch, Onur Mutlu, Phillip B. Gibbons, and Todd C. Mowry,
  "Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology"
  *Proceedings of the 50th International Symposium on Microarchitecture (**MICRO**)*, Boston, MA, USA, October 2017.
  [Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)] [Poster (pptx) (pdf)]



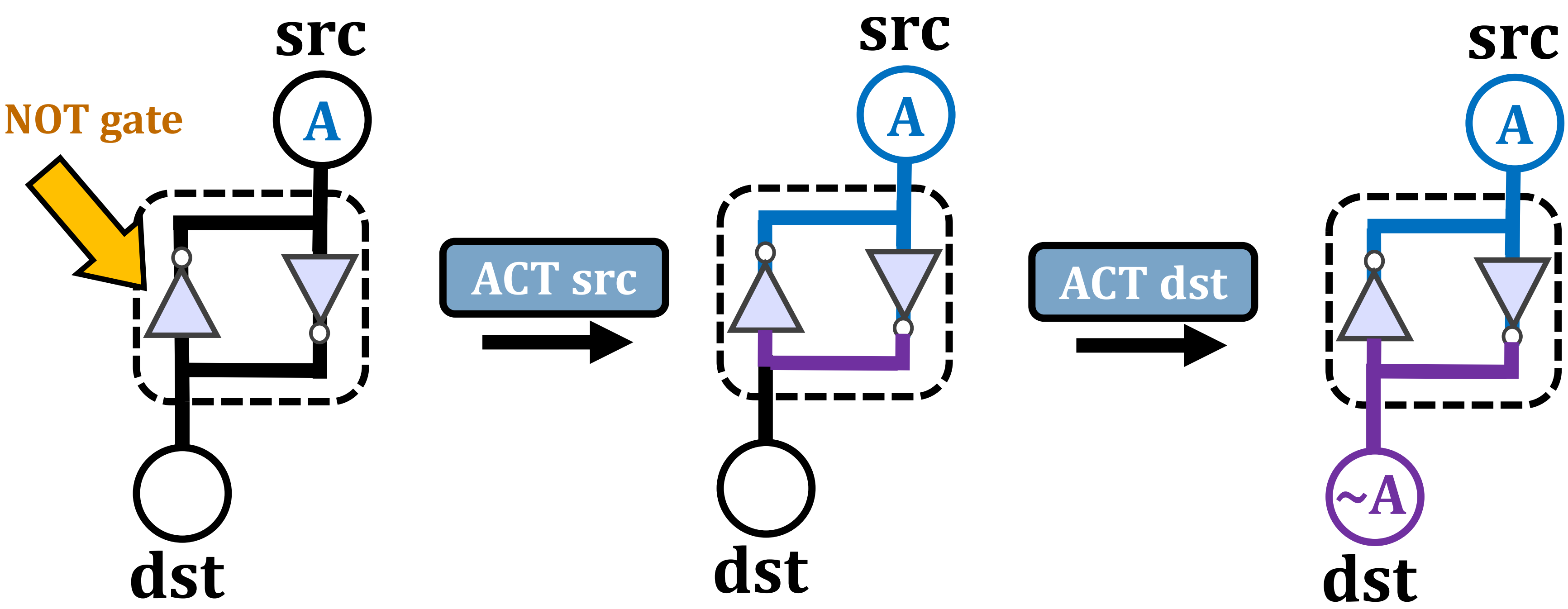
Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology

Vivek Seshadri[1,5] Donghyuk Lee[2,5] Thomas Mullins[3,5] Hasan Hassan[4] Amirali Boroumand[5]
Jeremie Kim[4,5] Michael A. Kozuch[3] Onur Mutlu[4,5] Phillip B. Gibbons[5] Todd C. Mowry[5]

[1]Microsoft Research India [2]NVIDIA Research [3]Intel [4]ETH Zürich [5]Carnegie Mellon University

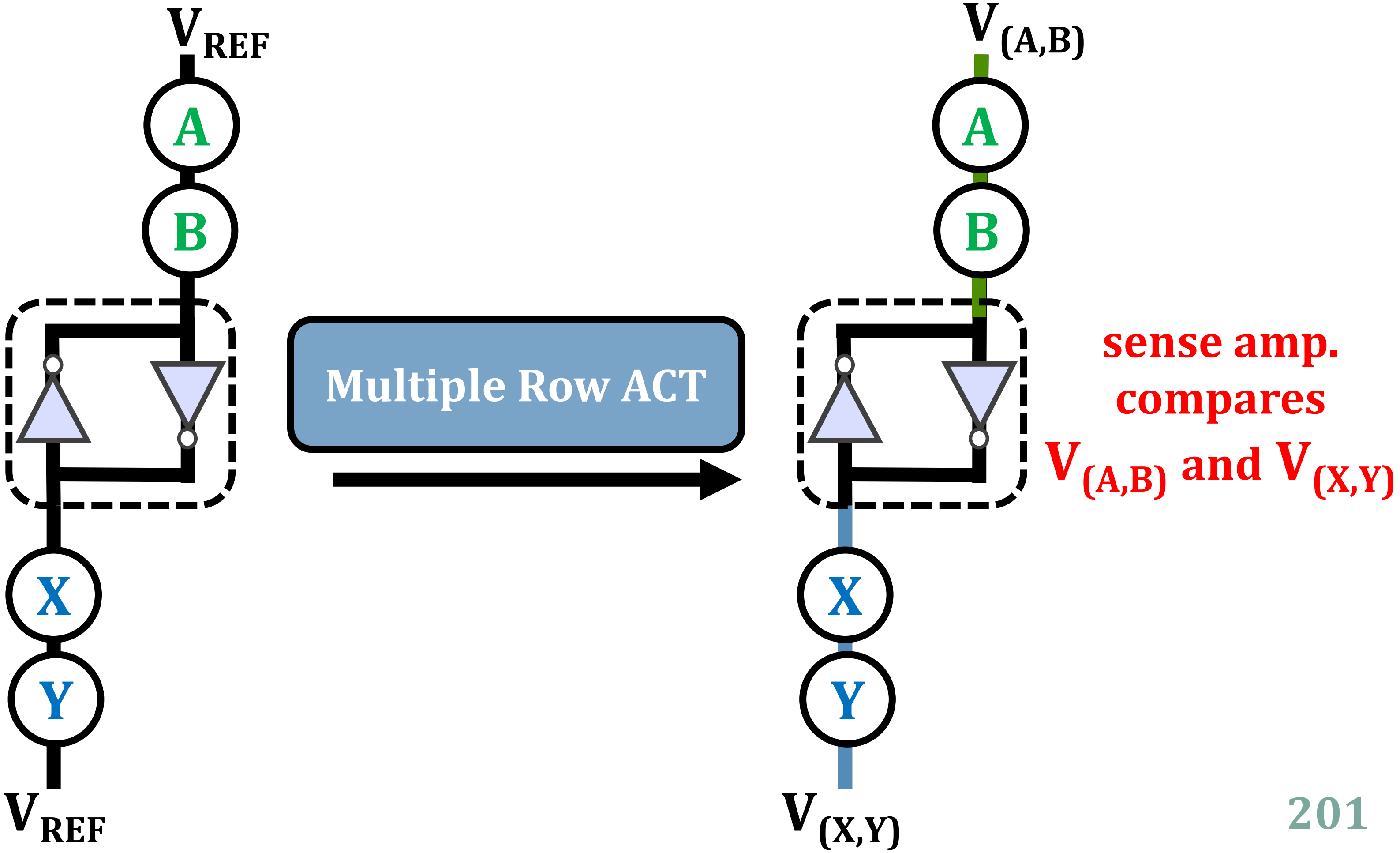
Key Idea: NOT Operation

Connect rows in neighboring subarrays
through **a NOT gate** by simultaneously activating rows



Key Idea: NAND, NOR, AND, OR

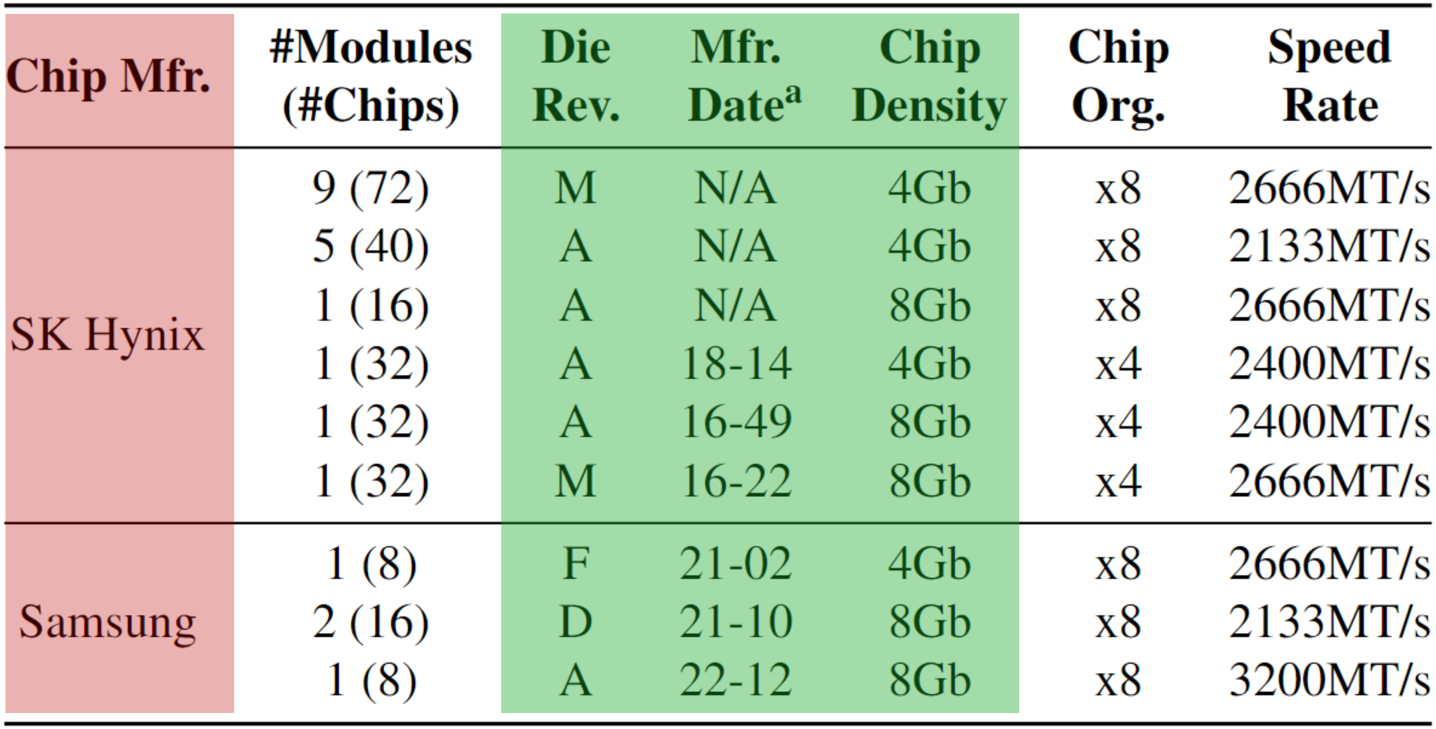
Manipulate the bitline voltage to express **a wide variety of functions** using multiple-row activation in neighboring subarrays



sense amp. compares $V_{(A,B)}$ and $V_{(X,Y)}$

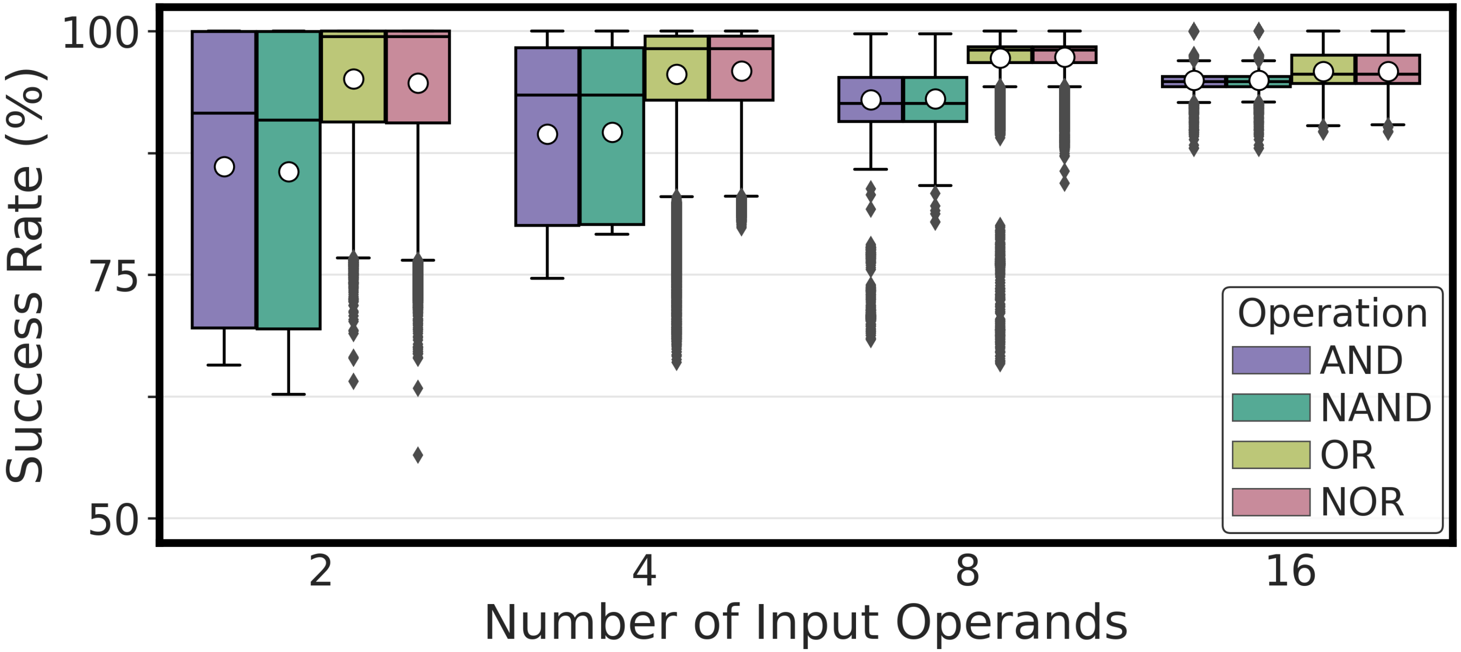
DRAM Chips Tested

- 256 DDR4 chips from two major DRAM manufacturers
- Covers different die revisions and chip densities



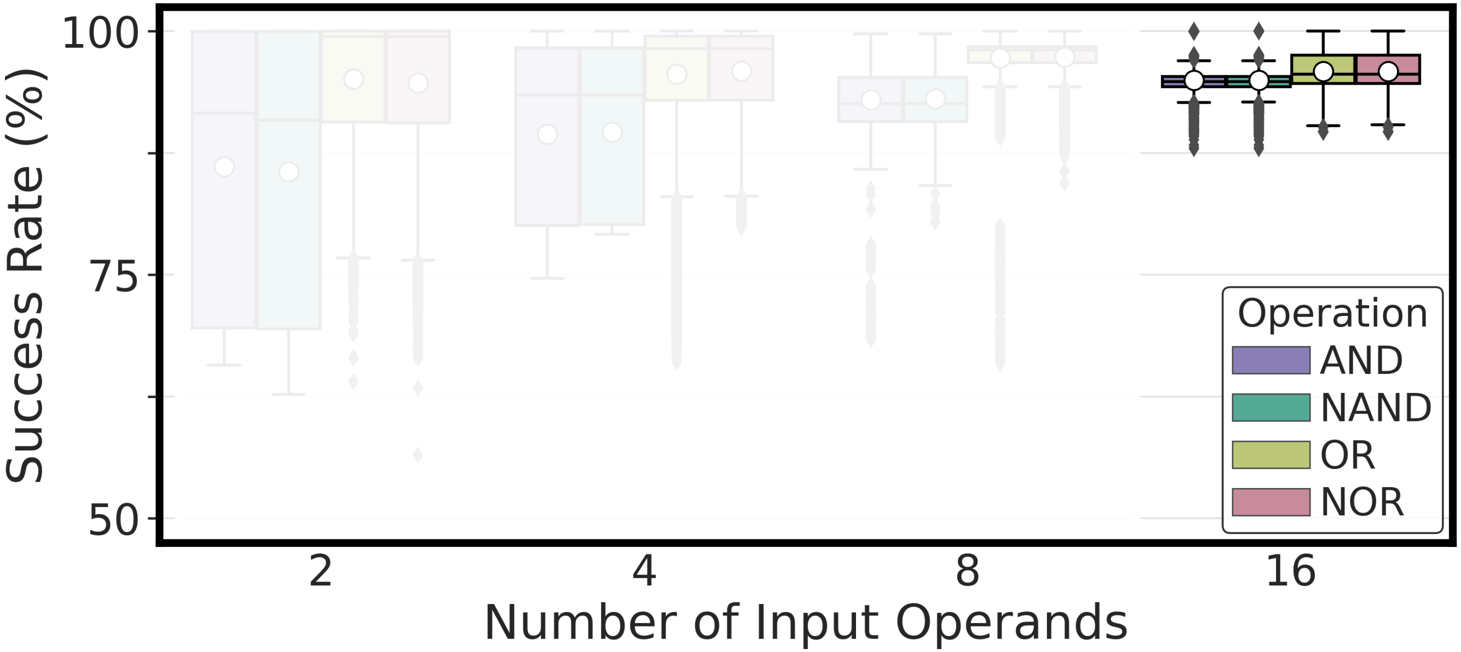
Chip Mfr.	#Modules (#Chips)	Die Rev.	Mfr. Date[a]	Chip Density	Chip Org.	Speed Rate
SK Hynix	9 (72)	M	N/A	4Gb	x8	2666MT/s
	5 (40)	A	N/A	4Gb	x8	2133MT/s
	1 (16)	A	N/A	8Gb	x8	2666MT/s
	1 (32)	A	18-14	4Gb	x4	2400MT/s
	1 (32)	A	16-49	8Gb	x4	2400MT/s
	1 (32)	M	16-22	8Gb	x4	2666MT/s
Samsung	1 (8)	F	21-02	4Gb	x8	2666MT/s
	2 (16)	D	21-10	8Gb	x8	2133MT/s
	1 (8)	A	22-12	8Gb	x8	3200MT/s

Performing AND, NAND, OR, and NOR



COTS DRAM chips can perform {2, 4, 8, 16}-input AND, NAND, OR, and NOR operations

Performing AND, NAND, OR, and NOR



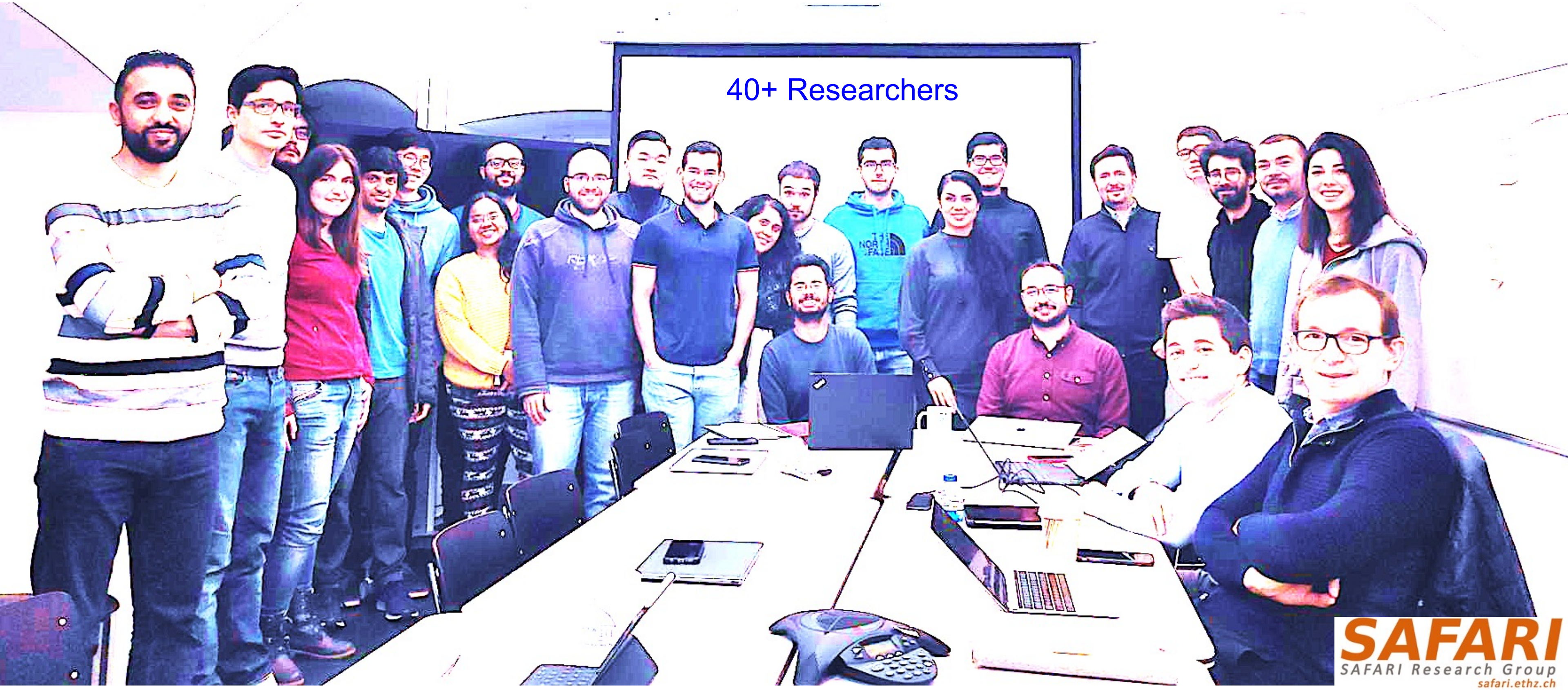
COTS DRAM chips can perform 16-input AND, NAND, OR, and NOR operations with very high success rate (>94%)

Other Backup Slides

Onur Mutlu's SAFARI Research Group

Computer architecture, HW/SW, systems, bioinformatics, security, memory

https://safari.ethz.ch/safari-newsletter-january-2021/



Think BIG, Aim HIGH!

SAFARI

https://safari.ethz.ch

SAFARI Newsletter December 2021 Edition

- https://safari.ethz.ch/safari-newsletter-december-2021/

SAFARI

SAFARI Research Group

Think Big, Aim High

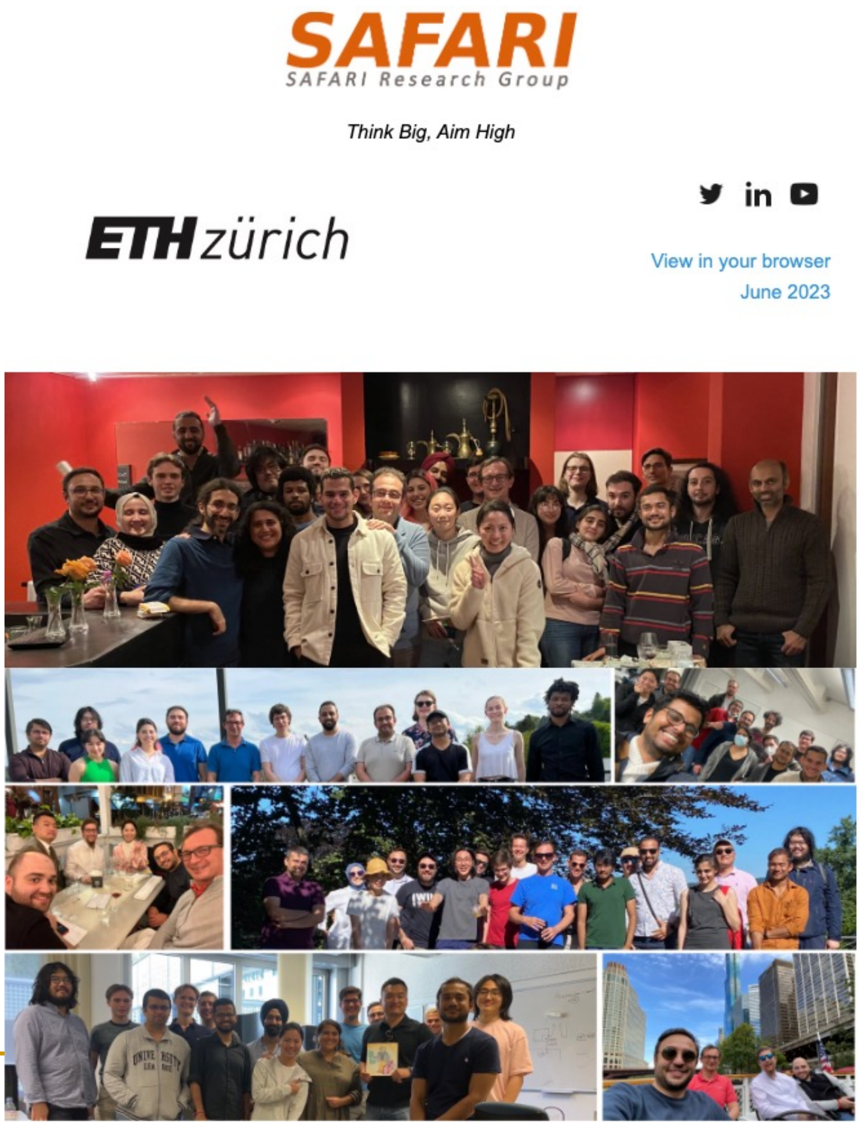
ETH zürich

View in your browser

December 2021

SAFARI Newsletter June 2023 Edition

- https://safari.ethz.ch/safari-newsletter-june-2023/



SAFARI

SAFARI Research Group

Think Big, Aim High

ETH zürich

View in your browser

June 2023

SAFARI Introduction & Research

Computer architecture, HW/SW, systems, bioinformatics, security, memory

SAFARI Research Group

Introduction & Research

Onur Mutlu

omutlu@gmail.com

https://people.inf.ethz.ch/omutlu

23 March 2023

Computer Architecture Seminar

Seminar in Computer Architecture - Lecture 5: Potpourri of Research Topics (Spring 2023)

Onur Mutlu Lectures

32.6K subscribers

719 views Streamed 1 month ago Livestream - Seminar in Computer Architecture - ETH Zürich (Spring 2023)

SAFARI

SAFARI Research Group

safari.ethz.ch

Think BIG, Aim HIGH!

https://www.youtube.com/watch?v=mV2OuB2djEs

SAFARI PhD and Post-Doc Alumni

- https://safari.ethz.ch/safari-alumni/

- Hasan Hassan (Rivos), EDAA Outstanding Dissertation Award 2023; S&P 2020 Best Paper Award, 2020 Pwnie Award, IEEE Micro TP HM 2020
- Christina Giannoula (Univ. of Toronto), NTUA Best Dissertation Award 2023
- Minesh Patel (Rutgers, Asst. Prof.), DSN Carter Award Best Thesis 2022; ETH Medal 2023; MICRO'20 & DSN'20 Best Paper Awards; ISCA HoF 2021
- Damla Senol Cali (Bionano Genomics), SRC TECHCON 2019 Best Student Presentation Award; RECOMB-Seq 2018 Best Poster Award
- Nastaran Hajinazar (Intel)
- Gagandeep Singh (AMD/Xilinx), FPL 2020 Best Paper Award Finalist
- Amirali Boroumand (Stanford Univ → Google), SRC TECHCON 2018 Best Presentation Award
- Jeremie Kim (Apple), EDAA Outstanding Dissertation Award 2020; IEEE Micro Top Picks 2019; ISCA/MICRO HoF 2021
- Nandita Vijaykumar (Univ. of Toronto, Assistant Professor), ISCA Hall of Fame 2021
- Kevin Hsieh (Microsoft Research, Senior Researcher)
- Justin Meza (Facebook), HiPEAC 2015 Best Student Presentation Award; ICCD 2012 Best Paper Award
- Mohammed Alser (ETH Zurich), IEEE Turkey Best PhD Thesis Award 2018
- Yixin Luo (Google), HPCA 2015 Best Paper Session
- Kevin Chang (Facebook), SRC TECHCON 2016 Best Student Presentation Award
- Rachata Ausavarungnirun (KMUNTB, Assistant Professor), NOCS 2015 and NOCS 2012 Best Paper Award Finalist
- Gennady Pekhimenko (Univ. of Toronto, Assistant Professor), ISCA Hall of Fame 2021; ASPLOS 2015 SRC Winner
- Vivek Seshadri (Microsoft Research)
- Donghyuk Lee (NVIDIA Research, Senior Researcher), HPCA Hall of Fame 2018
- Yoongu Kim (Software Robotics → Google), TCAD'19 Top Pick Award; IEEE Micro Top Picks'10; HPCA'10 Best Paper Session
- Lavanya Subramanian (Intel Labs → Facebook)

- Samira Khan (Univ. of Virginia, Assistant Professor), HPCA 2014 Best Paper Session
- Saugata Ghose (Univ. of Illinois, Assistant Professor), DFRWS-EU 2017 Best Paper Award
- Jawad Haj-Yahya (Huawei Research Zurich, Principal Researcher)
- Lois Orosa (Galicia Supercomputing Center, Director)
- Jisung Park (POSTECH, Assistant Professor)
- Gagandeep Singh (AMD/Xilinx, Researcher)
- Juan Gomez-Luna (NVIDIA, Researcher), ISPASS 2023 Best Paper Session

Processing in Memory: Evaluation Methods

Simulators (Open Source)

- Ramulator 2.0 & Ramulator-PIM
- DAMOVSim
- UPMEMSim (UPMEM)
- AiMSim (SK Hynix)
- …

Ramulator + Gem5

- Haocong Luo, Yahya Can Tugrul, F. Nisa Bostanci, Ataberk Olgun, A. Giray Yaglikci, and Onur Mutlu,
"Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator"
Preprint on ***arxiv***, August 2023.
[arXiv version]
[Ramulator 2.0 Source Code]

Ramulator 2.0: A Modern, Modular, and Extensible DRAM Simulator

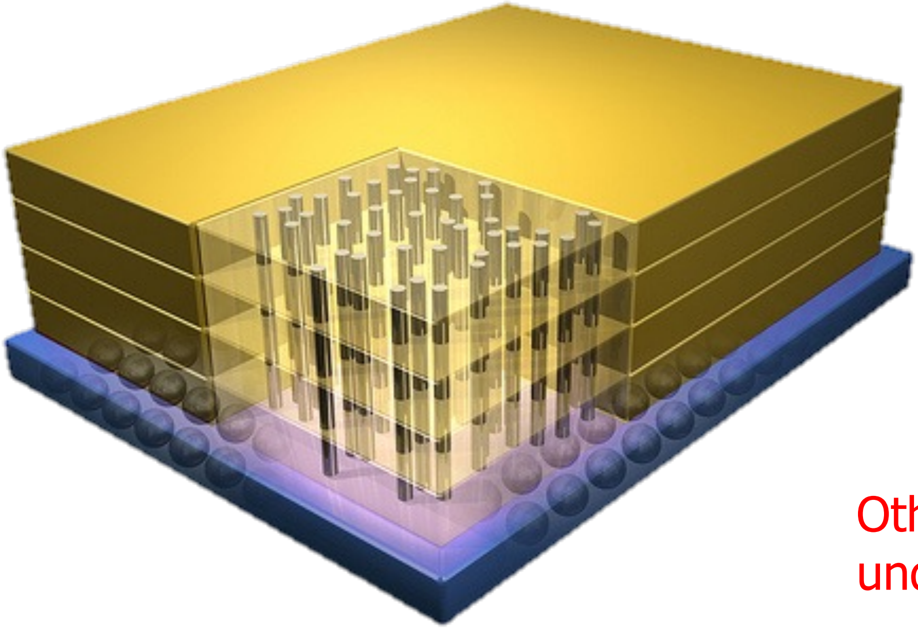
Haocong Luo, Yahya Can Tuğrul, F. Nisa Bostancı, Ataberk Olgun, A. Giray Yağlıkçı, and Onur Mutlu

https://arxiv.org/pdf/2308.11030.pdf

https://github.com/CMU-SAFARI/ramulator2

Opportunity: 3D-Stacked Logic+Memory

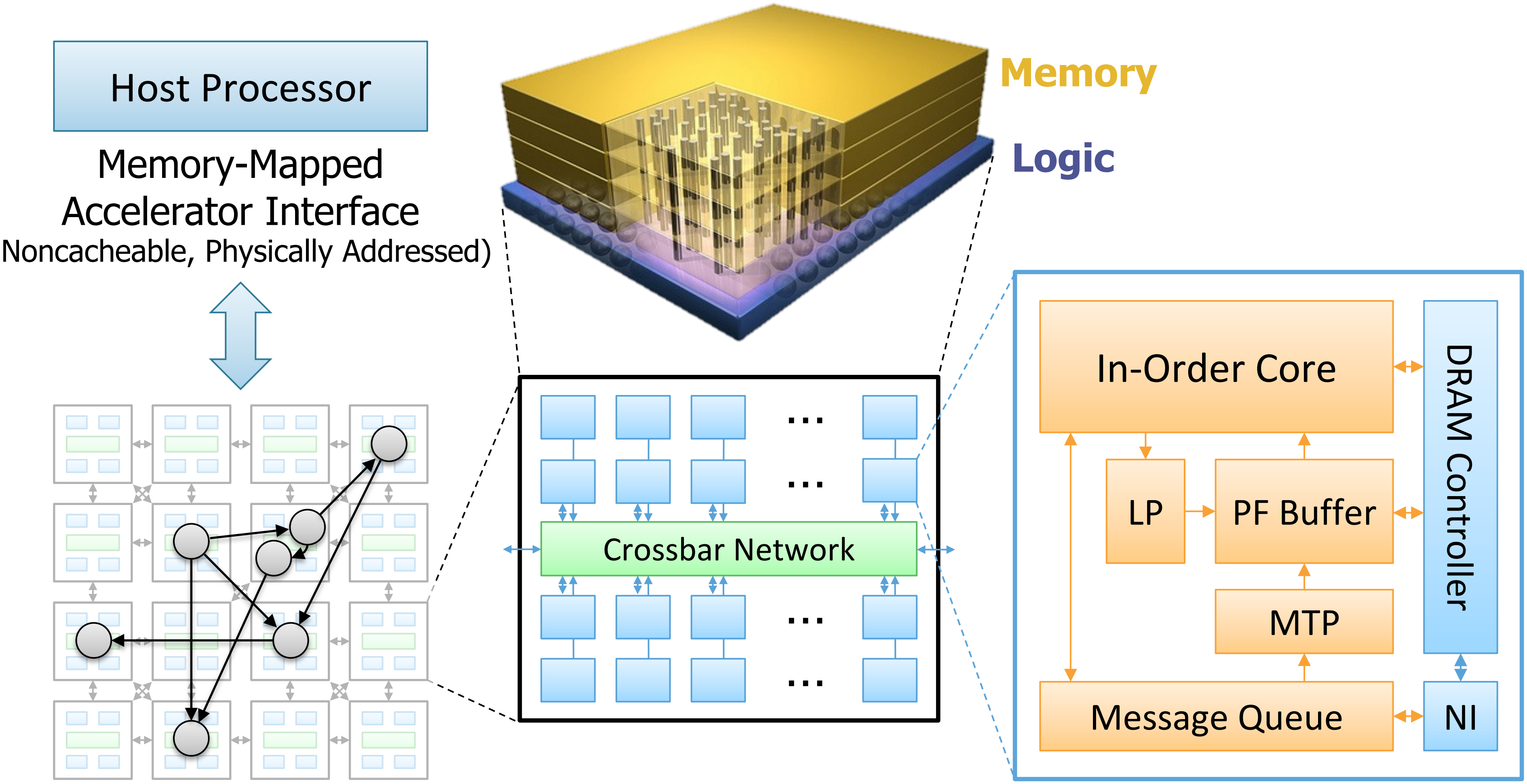
Hybrid Memory Cube
CONSORTIUM



Other "True 3D" technologies under development

Tesseract System for Graph Processing

Interconnected set of 3D-stacked memory+logic chips with simple cores



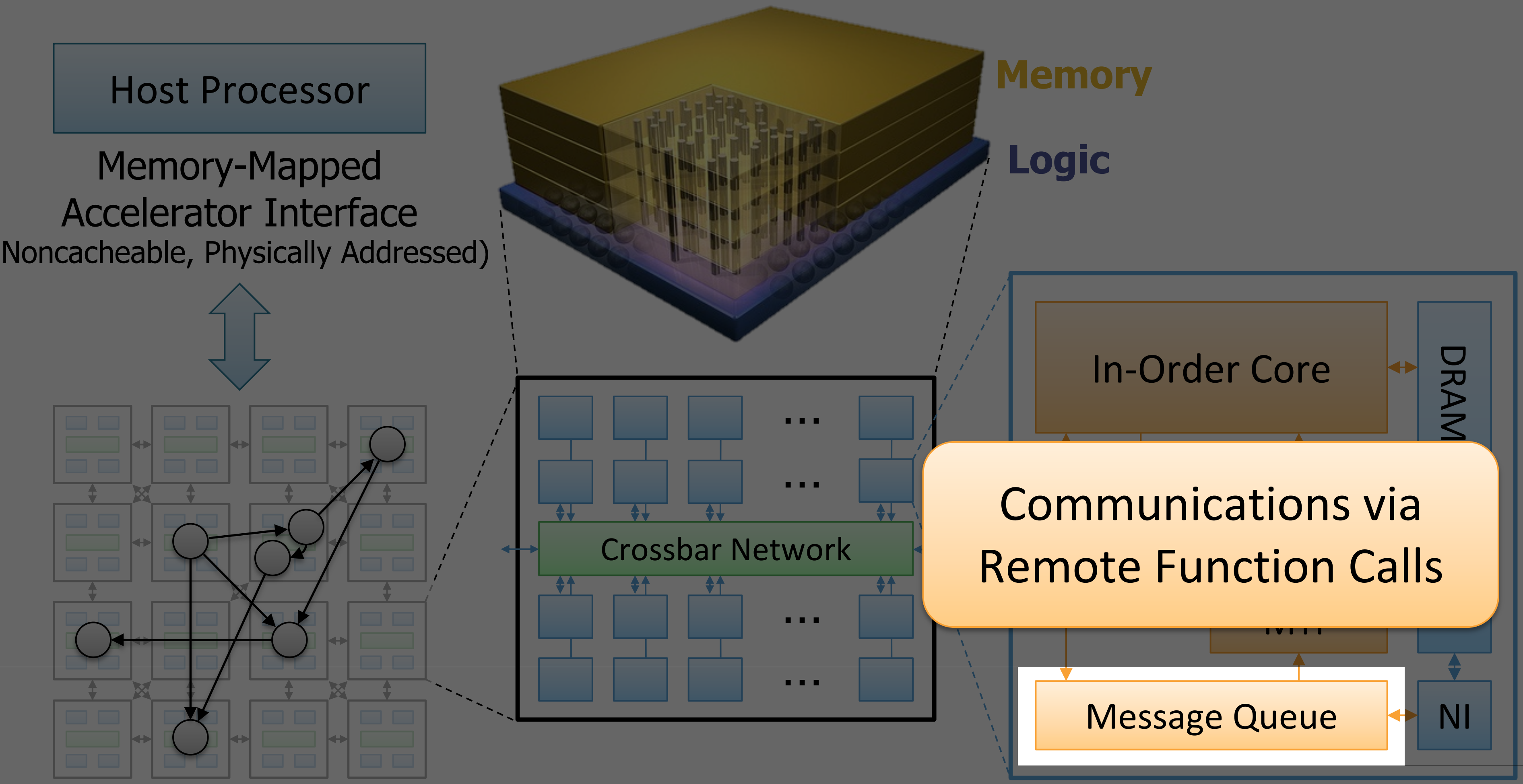
Host Processor

Memory-Mapped Accelerator Interface

Noncacheable, Physically Addressed)

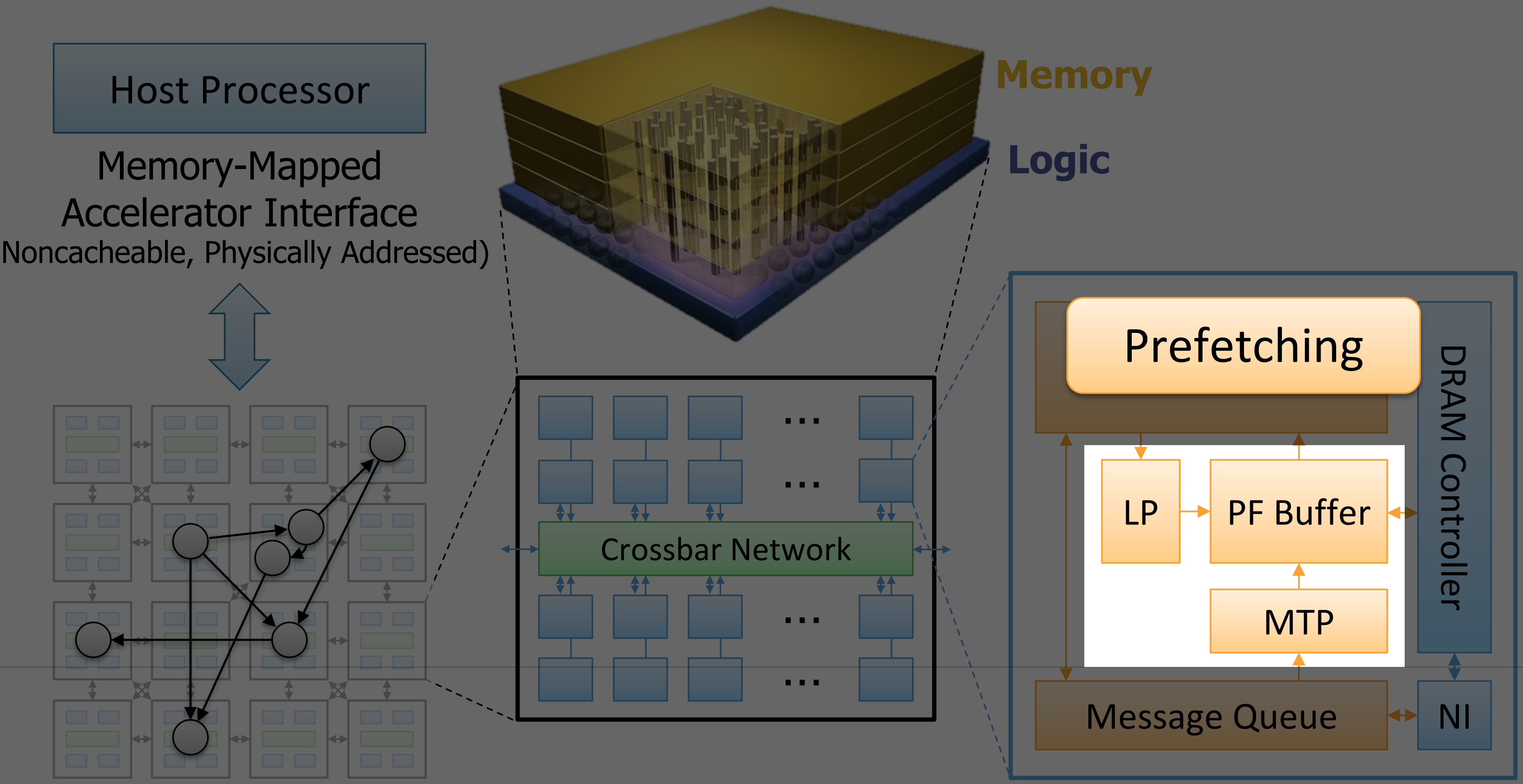
SAFARI Ahn+, "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing" ISCA 2015.

Tesseract System for Graph Processing

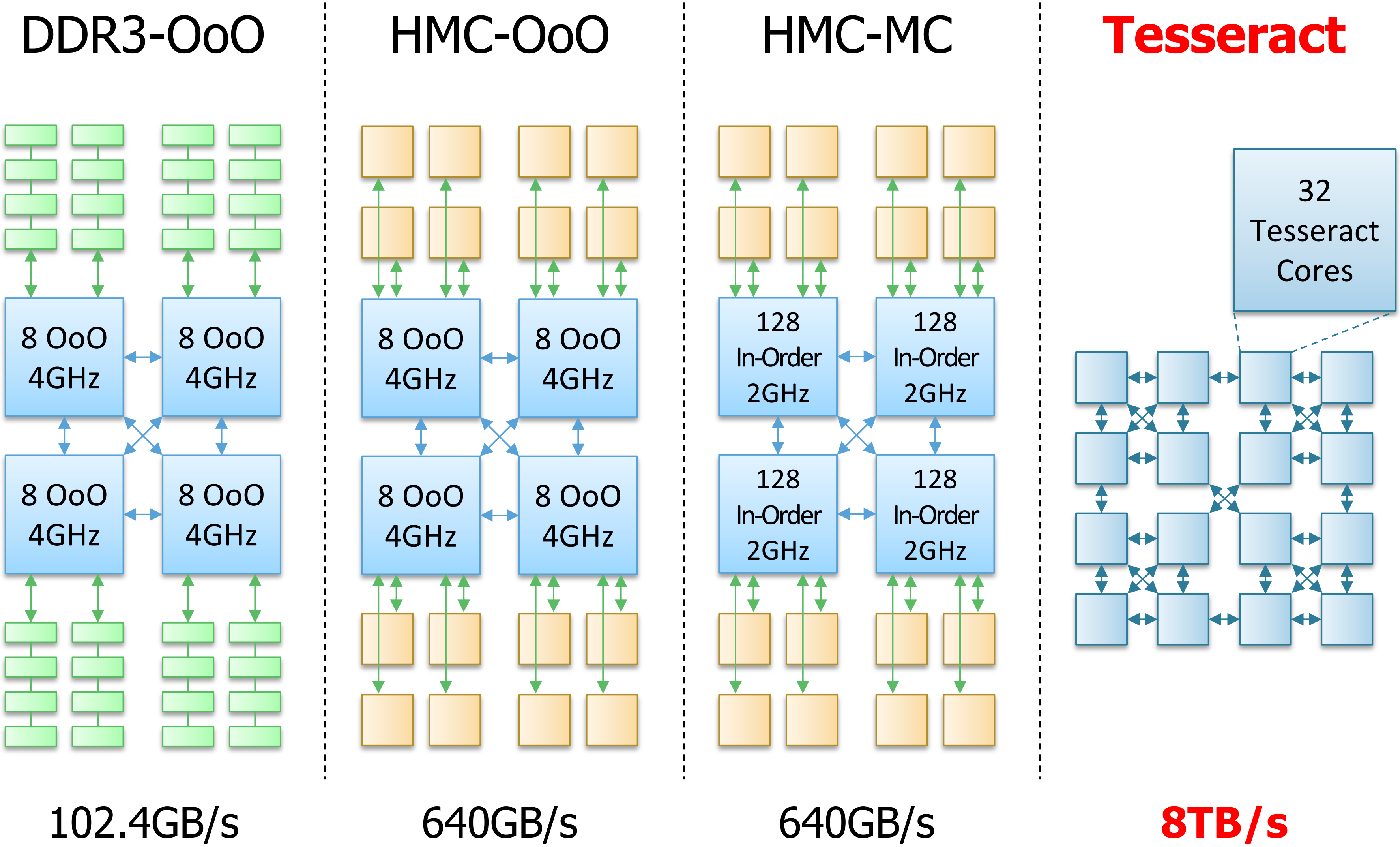


Communications via Remote Function Calls

Tesseract System for Graph Processing



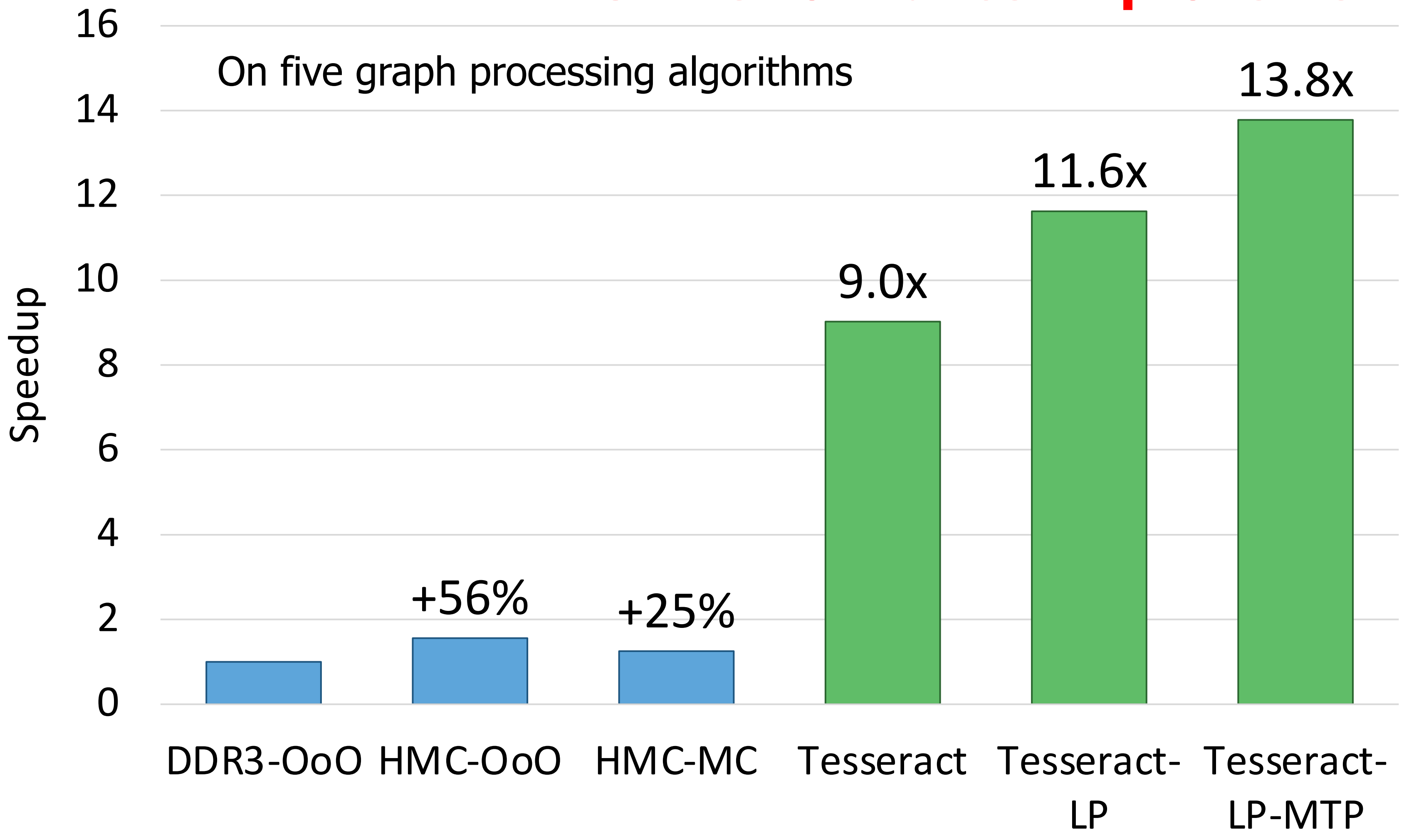
Simulated Systems



Ahn+, "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing" ISCA 2015.

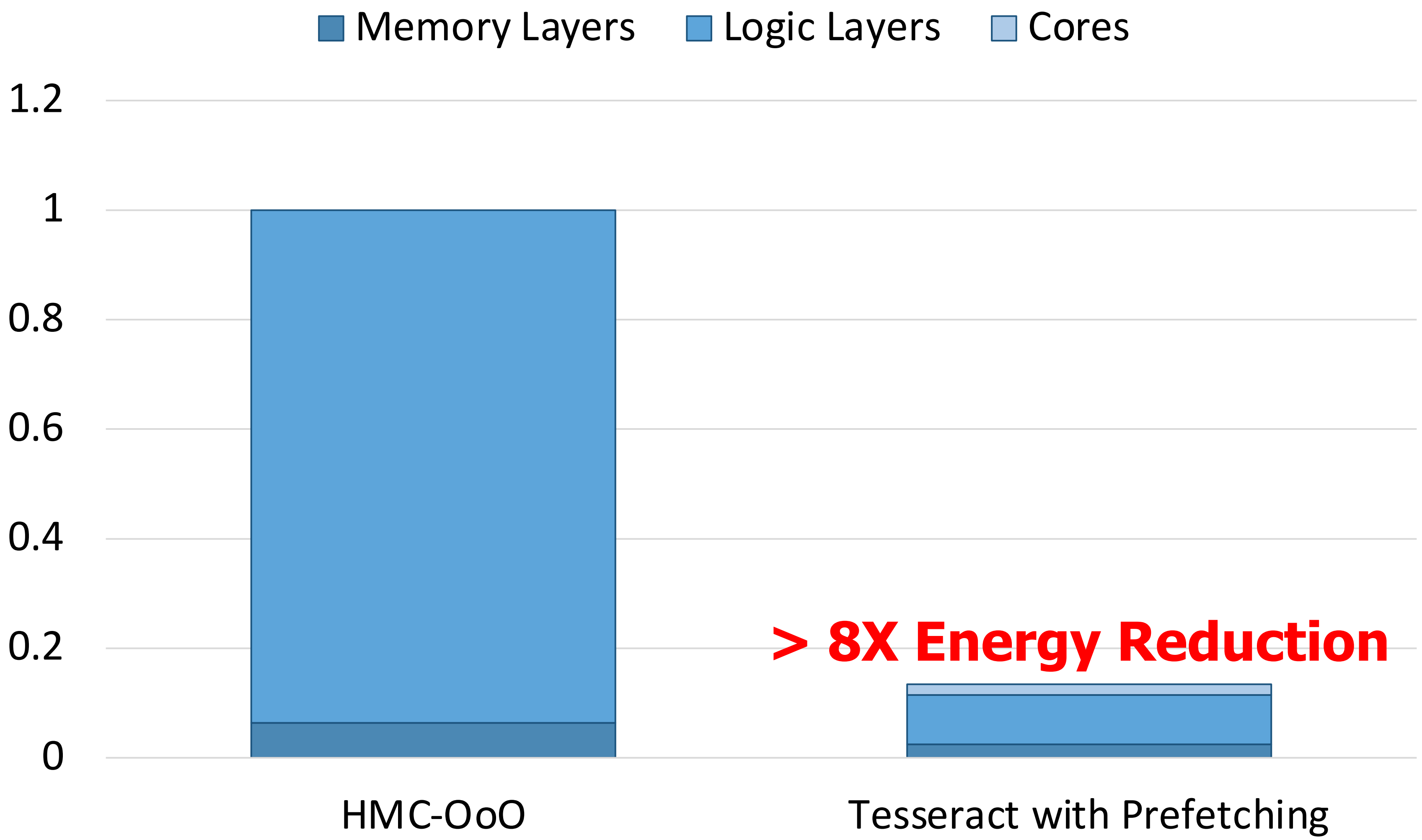
Tesseract Graph Processing Performance

>13X Performance Improvement



Ahn+, "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing" ISCA 2015.

Tesseract Graph Processing System Energy

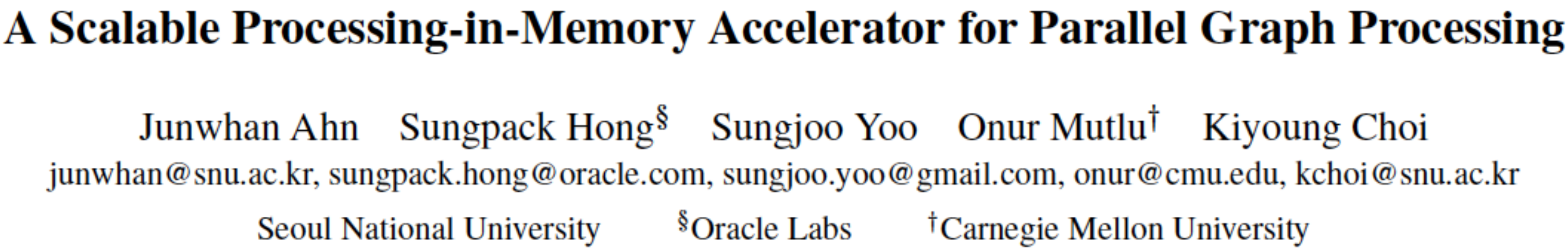


> 8X Energy Reduction

Ahn+, "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing" ISCA 2015.

More on Tesseract

- Junwhan Ahn, Sungpack Hong, Sungjoo Yoo, Onur Mutlu, and Kiyoung Choi,
  "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing"
  *Proceedings of the 42nd International Symposium on Computer Architecture (**ISCA**)*, Portland, OR, June 2015.
  [Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)]
  Top Picks Honorable Mention by IEEE Micro.
  Selected to the ISCA-50 25-Year Retrospective Issue covering 1996-2020 in 2023 (Retrospective (pdf) Full Issue).



A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing

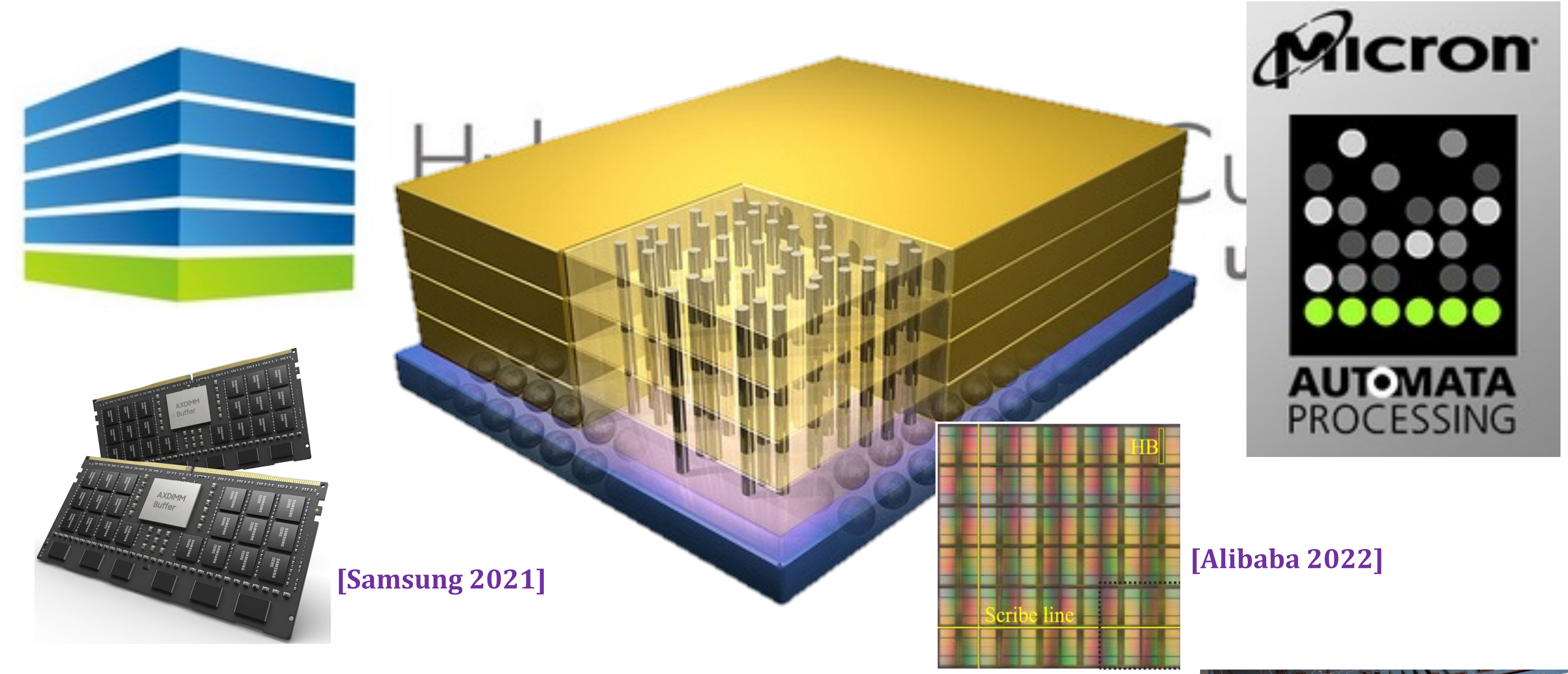
Junwhan Ahn Sungpack Hong[§] Sungjoo Yoo Onur Mutlu[†] Kiyoung Choi
junwhan@snu.ac.kr, sungpack.hong@oracle.com, sungjoo.yoo@gmail.com, onur@cmu.edu, kchoi@snu.ac.kr
Seoul National University [§]Oracle Labs [†]Carnegie Mellon University

PIM Prototypes

Processing-in-Memory

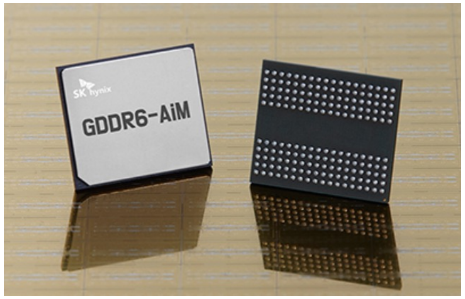
in the Real World

Processing-in-Memory Landscape Today

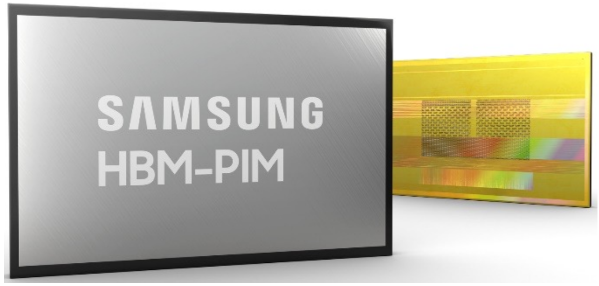


[Samsung 2021]

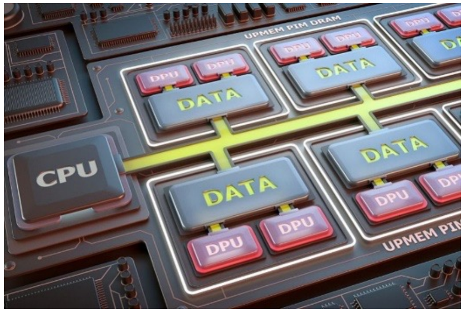
[Alibaba 2022]



[SK Hynix 2022]



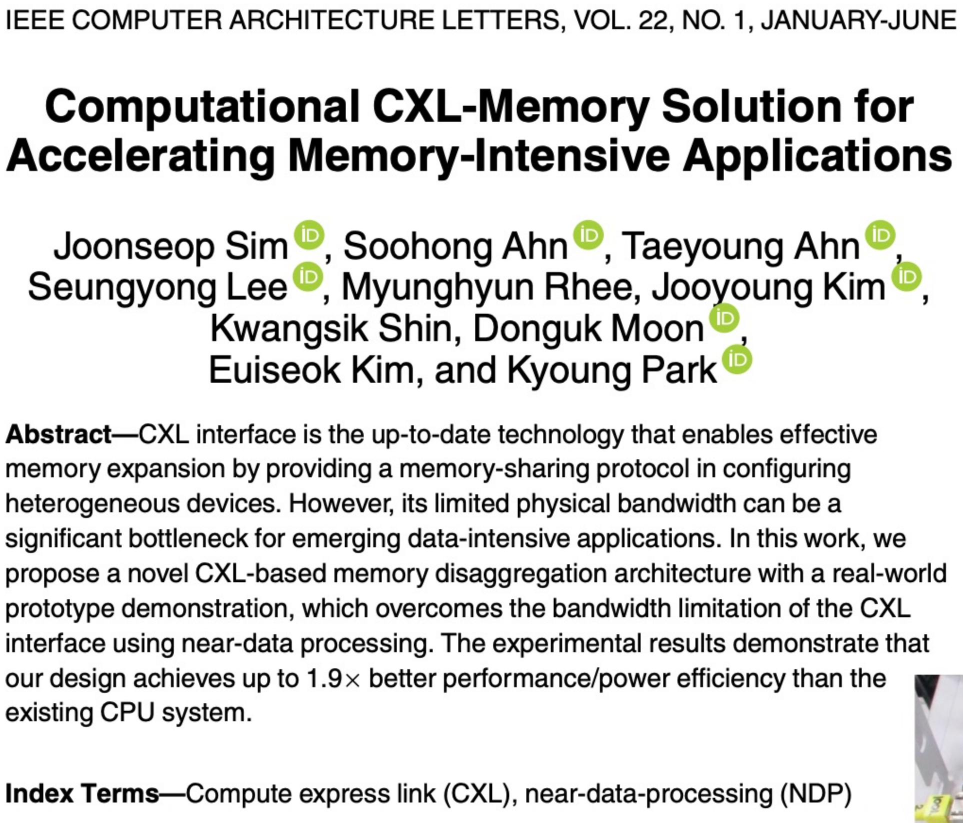
[Samsung 2021]



[UPMEM 2019]

And, many other experimental chips and startups

Processing-in-Memory Landscape Today



IEEE COMPUTER ARCHITECTURE LETTERS, VOL. 22, NO. 1, JANUARY-JUNE

Computational CXL-Memory Solution for Accelerating Memory-Intensive Applications

Joonseop Sim, Soohong Ahn, Taeyoung Ahn, Seungyong Lee, Myunghyun Rhee, Jooyoung Kim, Kwangsik Shin, Donguk Moon, Euiseok Kim, and Kyoung Park

Abstract—CXL interface is the up-to-date technology that enables effective memory expansion by providing a memory-sharing protocol in configuring heterogeneous devices. However, its limited physical bandwidth can be a significant bottleneck for emerging data-intensive applications. In this work, we propose a novel CXL-based memory disaggregation architecture with a real-world prototype demonstration, which overcomes the bandwidth limitation of the CXL interface using near-data processing. The experimental results demonstrate that our design achieves up to 1.9× better performance/power efficiency than the existing CPU system.

Index Terms—Compute express link (CXL), near-data-processing (NDP)

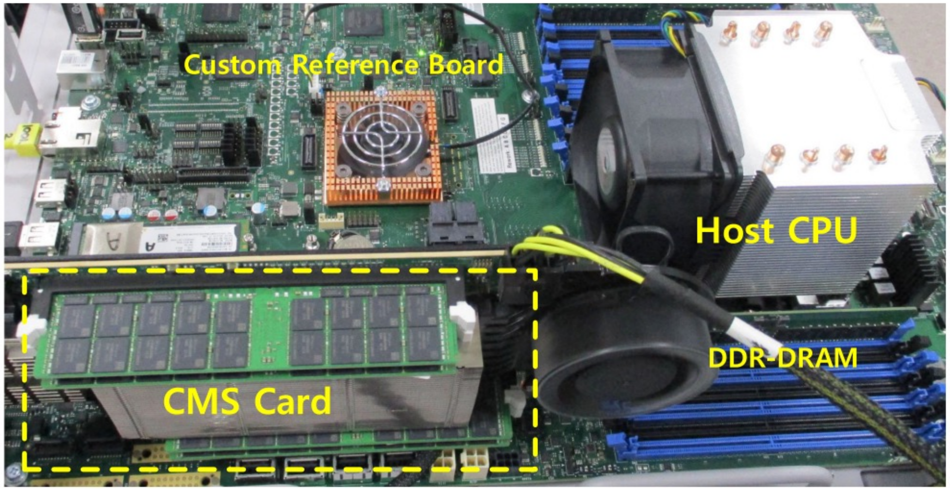
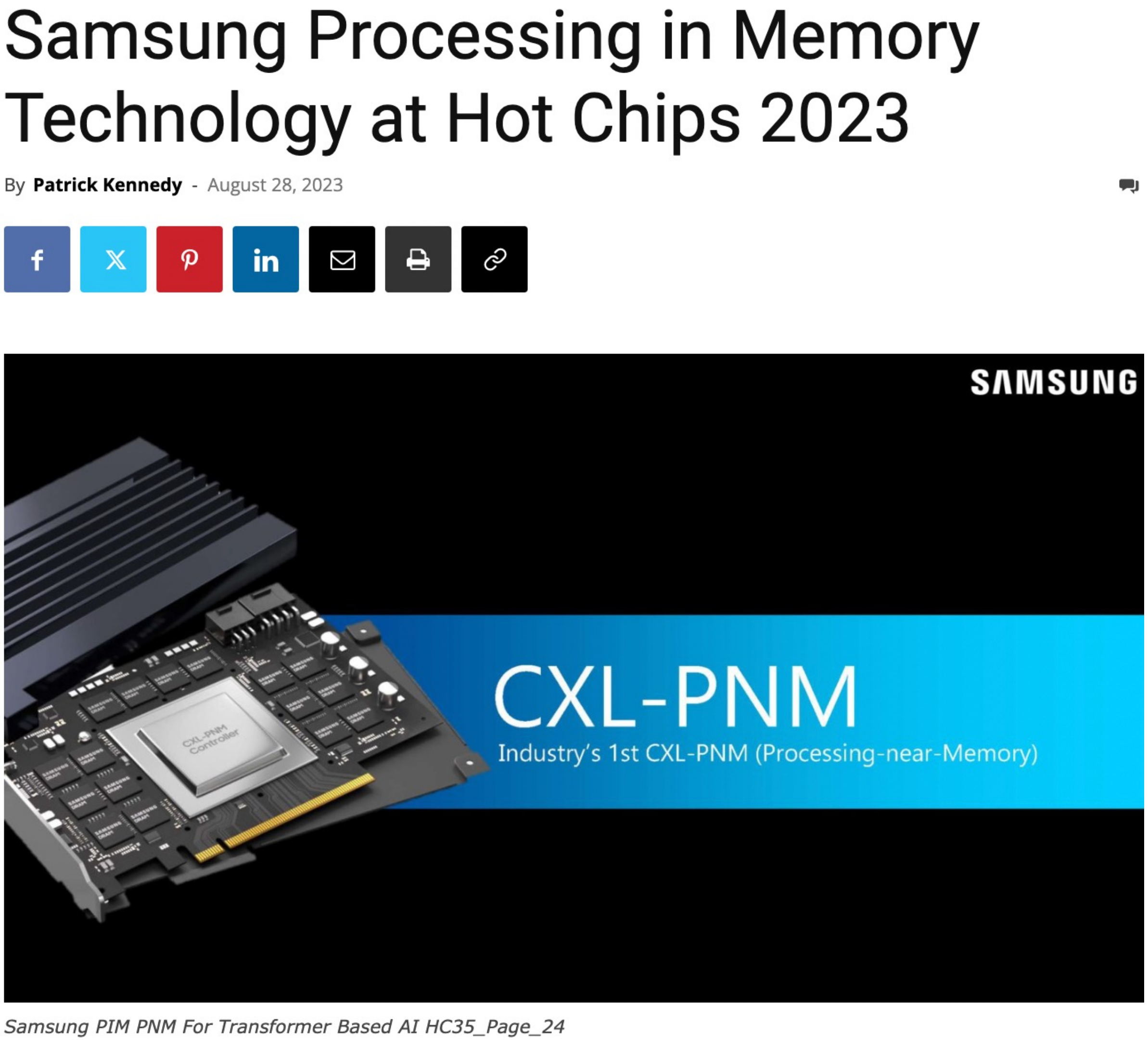


Fig. 6. FPGA prototype of proposed CMS card.

Processing-in-Memory Landscape Today



Samsung Processing in Memory Technology at Hot Chips 2023

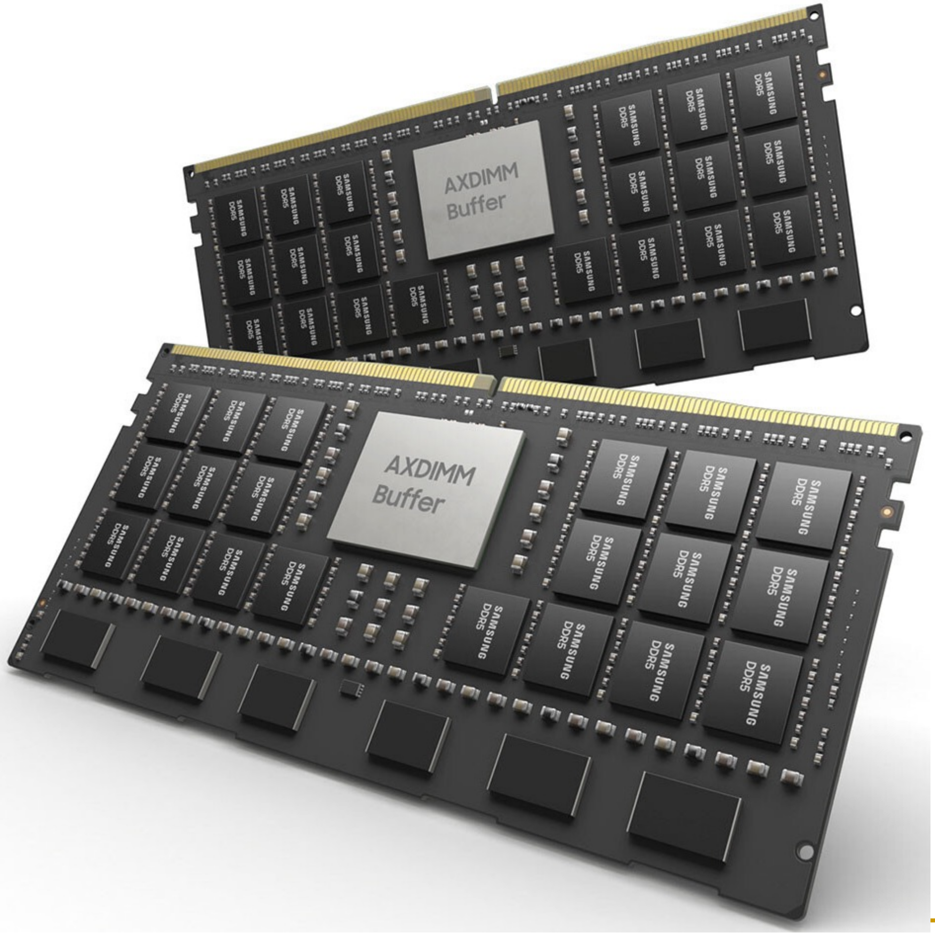
By **Patrick Kennedy** - August 28, 2023

Samsung PIM PNM For Transformer Based AI HC35_Page_24

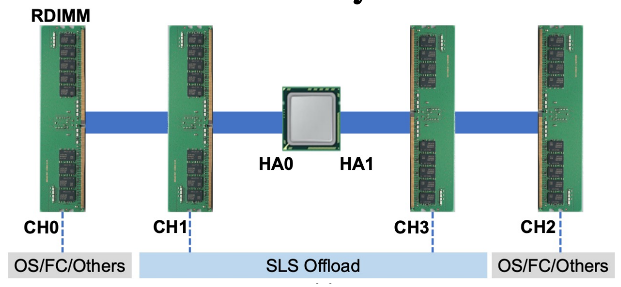
https://www.servethehome.com/samsung-processing-in-memory-technology-at-hot-chips-2023/

Samsung AxDIMM (2021)

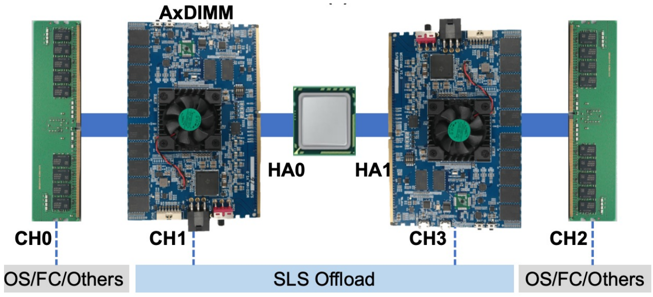
- DDRx-PIM
  - DLRM recommendation system

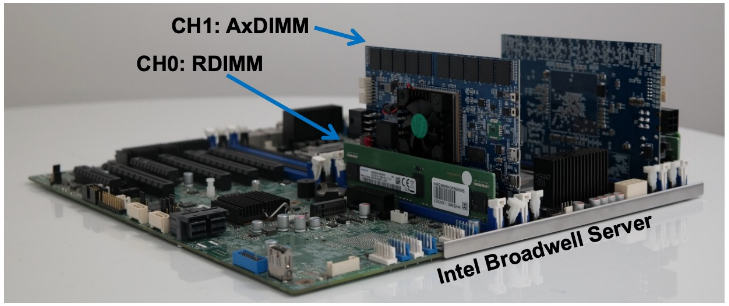


Baseline System



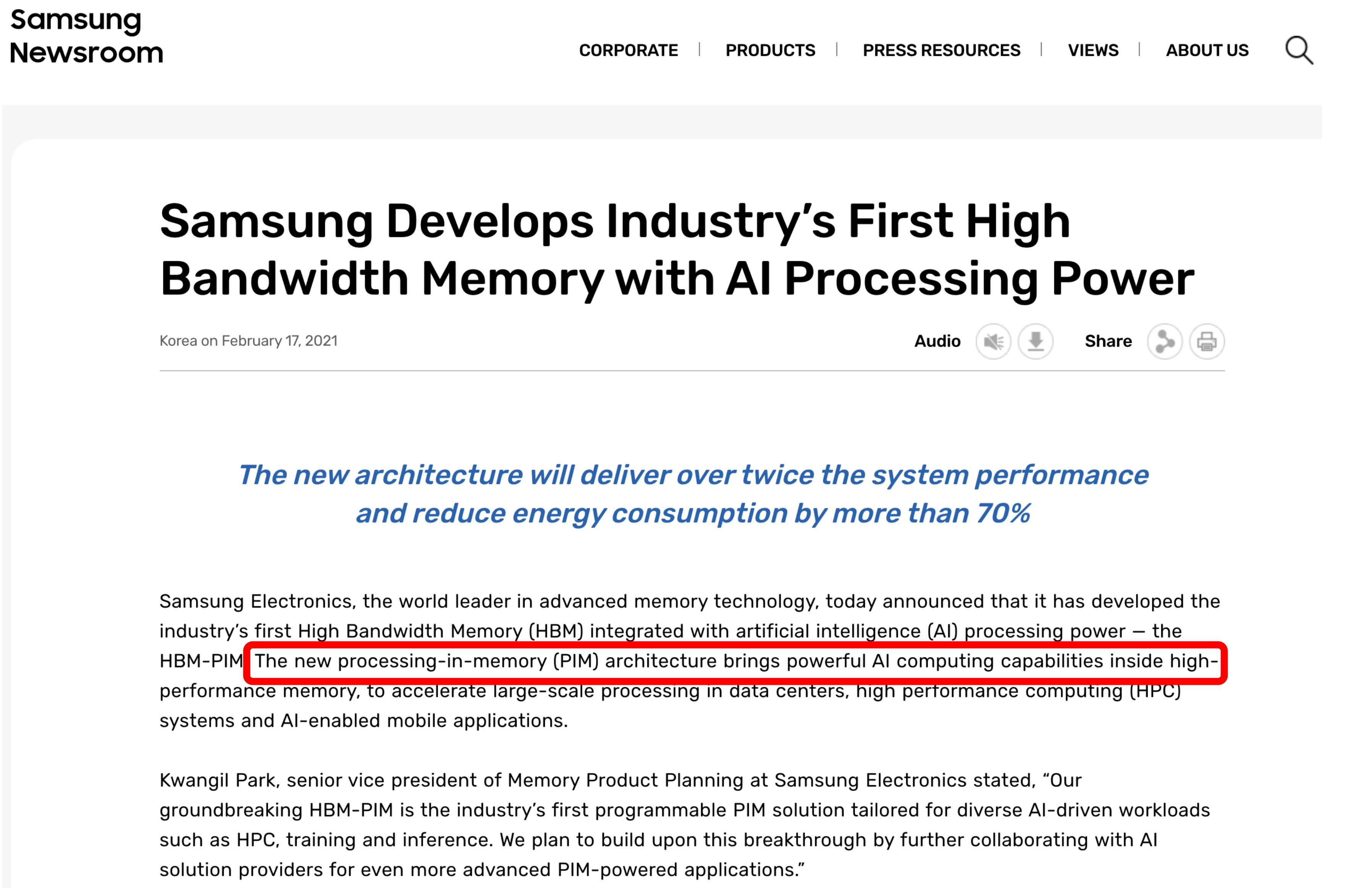
AxDIMM System





Ke et al. "Near-Memory Processing in Action: Accelerating Personalized Recommendation with AxDIMM", IEEE Micro (2021)

Samsung Function-in-Memory DRAM (2021)



Samsung Develops Industry's First High Bandwidth Memory with AI Processing Power

Korea on February 17, 2021

The new architecture will deliver over twice the system performance and reduce energy consumption by more than 70%

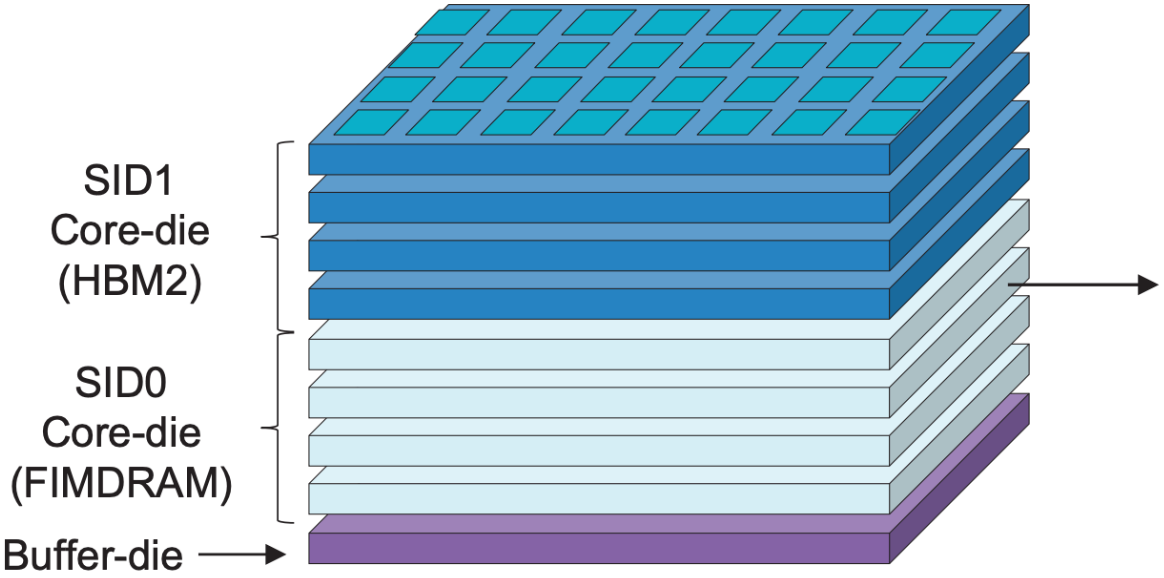
Samsung Electronics, the world leader in advanced memory technology, today announced that it has developed the industry's first High Bandwidth Memory (HBM) integrated with artificial intelligence (AI) processing power — the HBM-PIM. The new processing-in-memory (PIM) architecture brings powerful AI computing capabilities inside high-performance memory, to accelerate large-scale processing in data centers, high performance computing (HPC) systems and AI-enabled mobile applications.

Kwangil Park, senior vice president of Memory Product Planning at Samsung Electronics stated, "Our groundbreaking HBM-PIM is the industry's first programmable PIM solution tailored for diverse AI-driven workloads such as HPC, training and inference. We plan to build upon this breakthrough by further collaborating with AI solution providers for even more advanced PIM-powered applications."

https://news.samsung.com/global/samsung-develops-industrys-first-high-bandwidth-memory-with-ai-processing-power

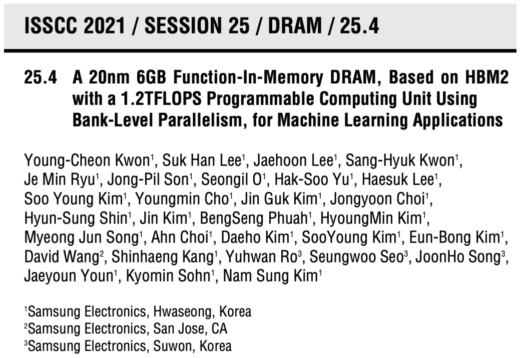
Samsung Function-in-Memory DRAM (2021)

- FIMDRAM based on HBM2



[3D Chip Structure of HBM with FIMDRAM]

Chip Specification
128DQ / 8CH / 16 banks / BL4
32 PCU blocks (1 FIM block/2 banks)
1.2 TFLOPS (4H)
FP16 ADD / Multiply (MUL) / Multiply-Accumulate (MAC) / Multiply-and- Add (MAD)



ISSCC 2021 / SESSION 25 / DRAM / 25.4

25.4 A 20nm 6GB Function-In-Memory DRAM, Based on HBM2 with a 1.2TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications

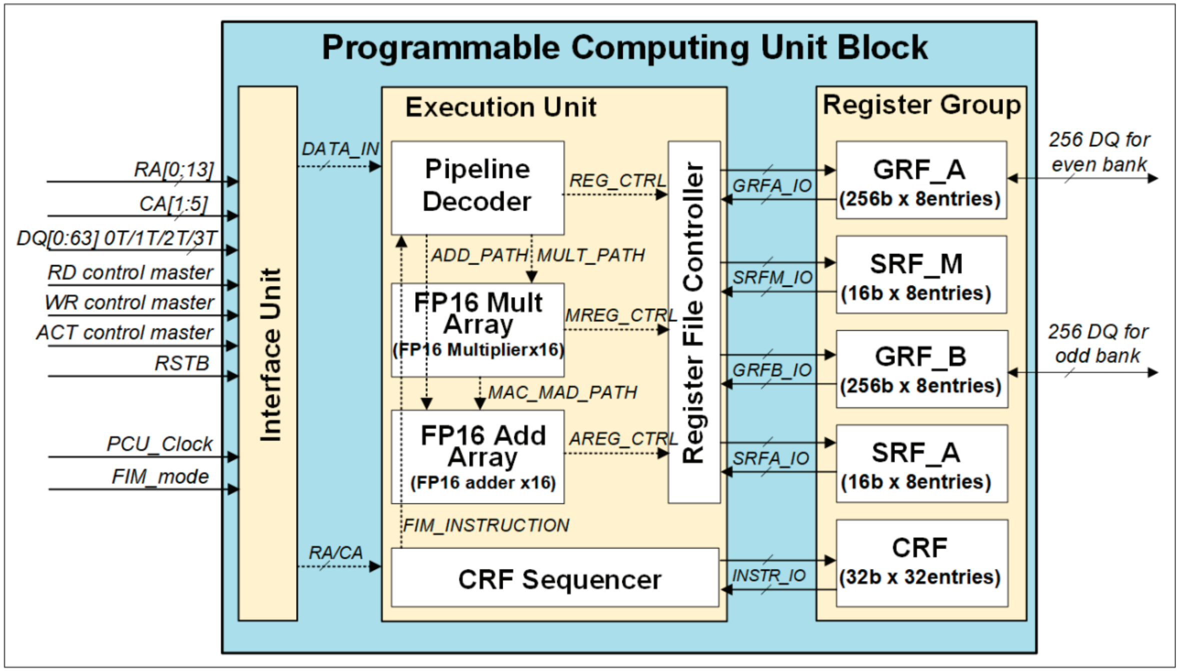
Young-Cheon Kwon[1], Suk Han Lee[1], Jaehoon Lee[1], Sang-Hyuk Kwon[1],
Je Min Ryu[1], Jong-Pil Son[1], Seongil O[1], Hak-Soo Yu[1], Haesuk Lee[1],
Soo Young Kim[1], Youngmin Cho[1], Jin Guk Kim[1], Jongyoon Choi[1],
Hyun-Sung Shin[1], Jin Kim[1], BengSeng Phuah[1], HyoungMin Kim[1],
Myeong Jun Song[1], Ahn Choi[1], Daeho Kim[1], SooYoung Kim[1], Eun-Bong Kim[1],
David Wang[2], Shinhaeng Kang[1], Yuhwan Ro[3], Seungwoo Seo[3], JoonHo Song[3],
Jaeyoun Youn[1], Kyomin Sohn[1], Nam Sung Kim[1]

[1]Samsung Electronics, Hwaseong, Korea
[2]Samsung Electronics, San Jose, CA
[3]Samsung Electronics, Suwon, Korea

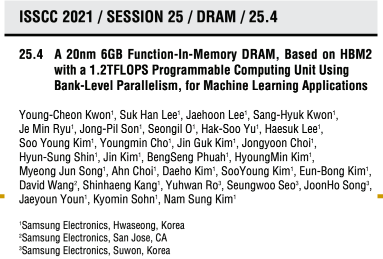
Samsung Function-in-Memory DRAM (2021)

Programmable Computing Unit

- Configuration of PCU block
  - Interface unit to control data flow
  - Execution unit to perform operations
  - Register group
    - 32 entries of CRF for instruction memory
    - 16 GRF for weight and accumulation
    - 16 SRF to store constants for MAC operations



[Block diagram of PCU in FIMDRAM]



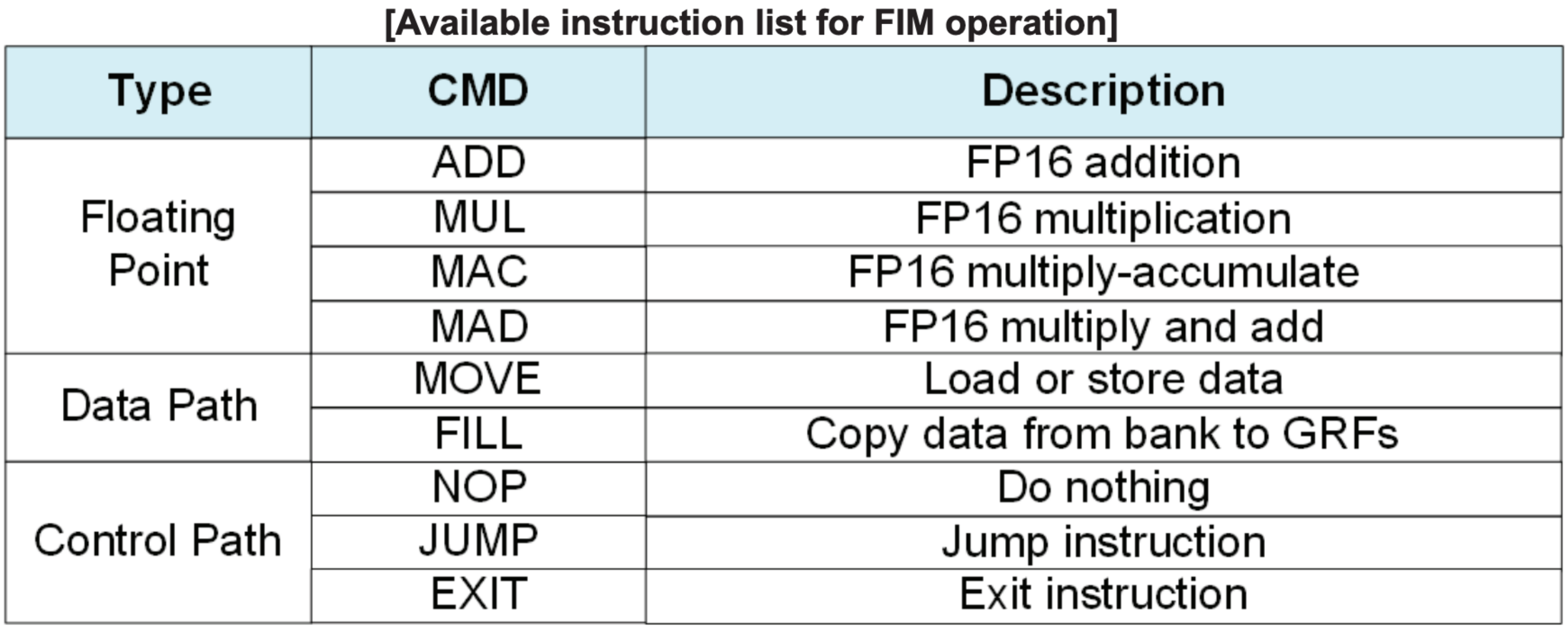
ISSCC 2021 / SESSION 25 / DRAM / 25.4

25.4 A 20nm 6GB Function-In-Memory DRAM, Based on HBM2 with a 1.2TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications

Young-Cheon Kwon[1], Suk Han Lee[1], Jaehoon Lee[1], Sang-Hyuk Kwon[1], Je Min Ryu[1], Jong-Pil Son[1], Seongil O[1], Hak-Soo Yu[1], Haesuk Lee[1], Soo Young Kim[1], Youngmin Cho[1], Jin Guk Kim[1], Jongyoon Choi[1], Hyun-Sung Shin[1], Jin Kim[1], BengSeng Phuah[1], HyoungMin Kim[1], Myeong Jun Song[1], Ahn Choi[1], Daeho Kim[1], SooYoung Kim[1], Eun-Bong Kim[1], David Wang[2], Shinhaeng Kang[1], Yuhwan Ro[3], Seungwoo Seo[3], JoonHo Song[3], Jaeyoun Youn[1], Kyomin Sohn[1], Nam Sung Kim[1]

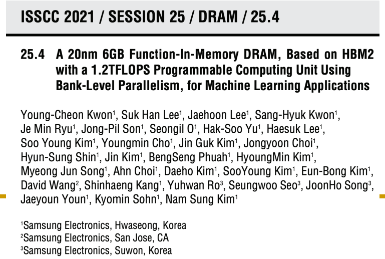
[1]Samsung Electronics, Hwaseong, Korea
[2]Samsung Electronics, San Jose, CA
[3]Samsung Electronics, Suwon, Korea

Samsung Function-in-Memory DRAM (2021)



[Available instruction list for FIM operation]

Type	CMD	Description
Floating Point	ADD	FP16 addition
	MUL	FP16 multiplication
	MAC	FP16 multiply-accumulate
	MAD	FP16 multiply and add
Data Path	MOVE	Load or store data
	FILL	Copy data from bank to GRFs
Control Path	NOP	Do nothing
	JUMP	Jump instruction
	EXIT	Exit instruction



ISSCC 2021 / SESSION 25 / DRAM / 25.4

25.4 A 20nm 6GB Function-In-Memory DRAM, Based on HBM2 with a 1.2TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications

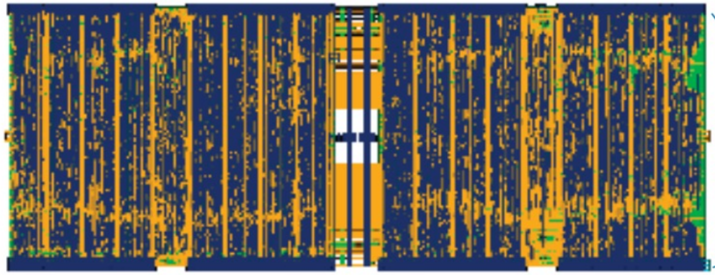
Young-Cheon Kwon[1], Suk Han Lee[1], Jaehoon Lee[1], Sang-Hyuk Kwon[1],
Je Min Ryu[1], Jong-Pil Son[1], Seongil O[1], Hak-Soo Yu[1], Haesuk Lee[1],
Soo Young Kim[1], Youngmin Cho[1], Jin Guk Kim[1], Jongyoon Choi[1],
Hyun-Sung Shin[1], Jin Kim[1], BengSeng Phuah[1], HyoungMin Kim[1],
Myeong Jun Song[1], Ahn Choi[1], Daeho Kim[1], SooYoung Kim[1], Eun-Bong Kim[1],
David Wang[2], Shinhaeng Kang[1], Yuhwan Ro[3], Seungwoo Seo[3], JoonHo Song[3],
Jaeyoun Youn[1], Kyomin Sohn[1], Nam Sung Kim[1]

[1]Samsung Electronics, Hwaseong, Korea
[2]Samsung Electronics, San Jose, CA
[3]Samsung Electronics, Suwon, Korea

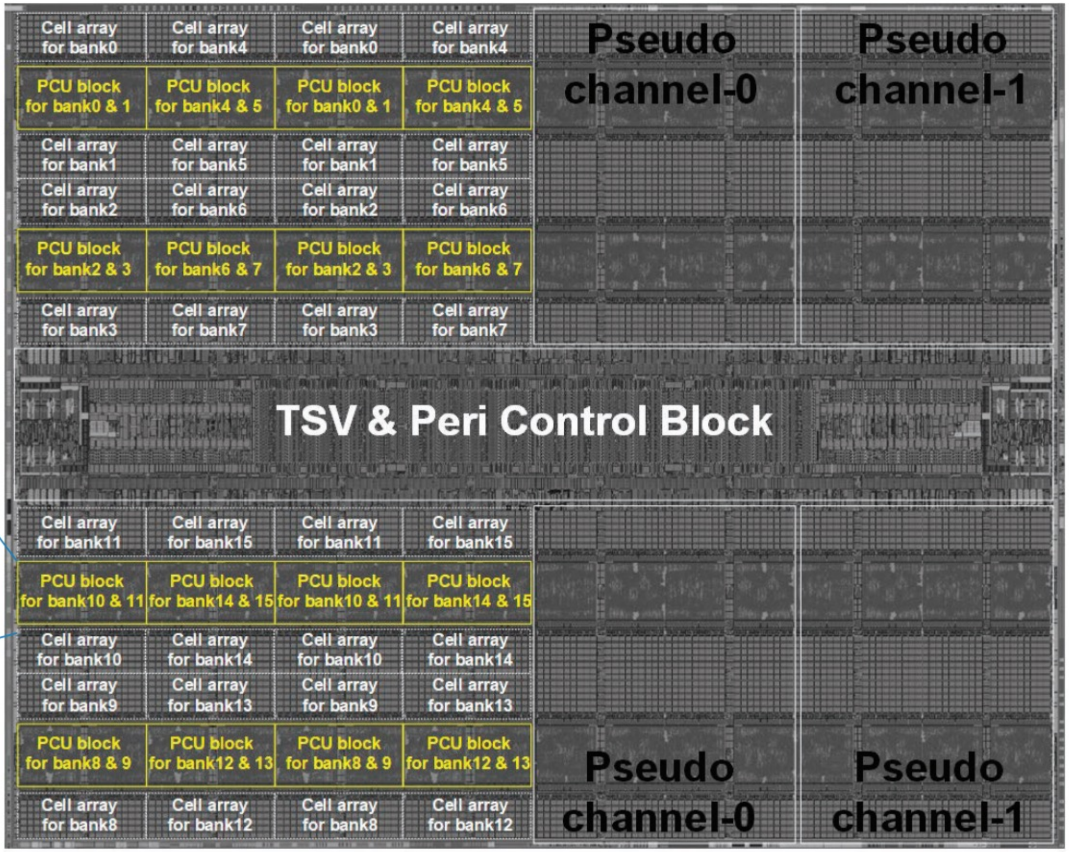
Samsung Function-in-Memory DRAM (2021)

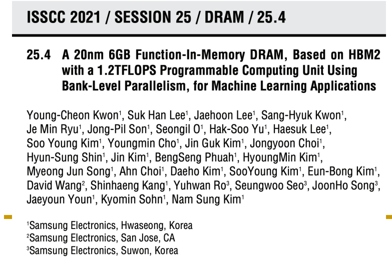
Chip Implementation

- Mixed design methodology to implement FIMDRAM
  - Full-custom + Digital RTL



[Digital RTL design for PCU block]





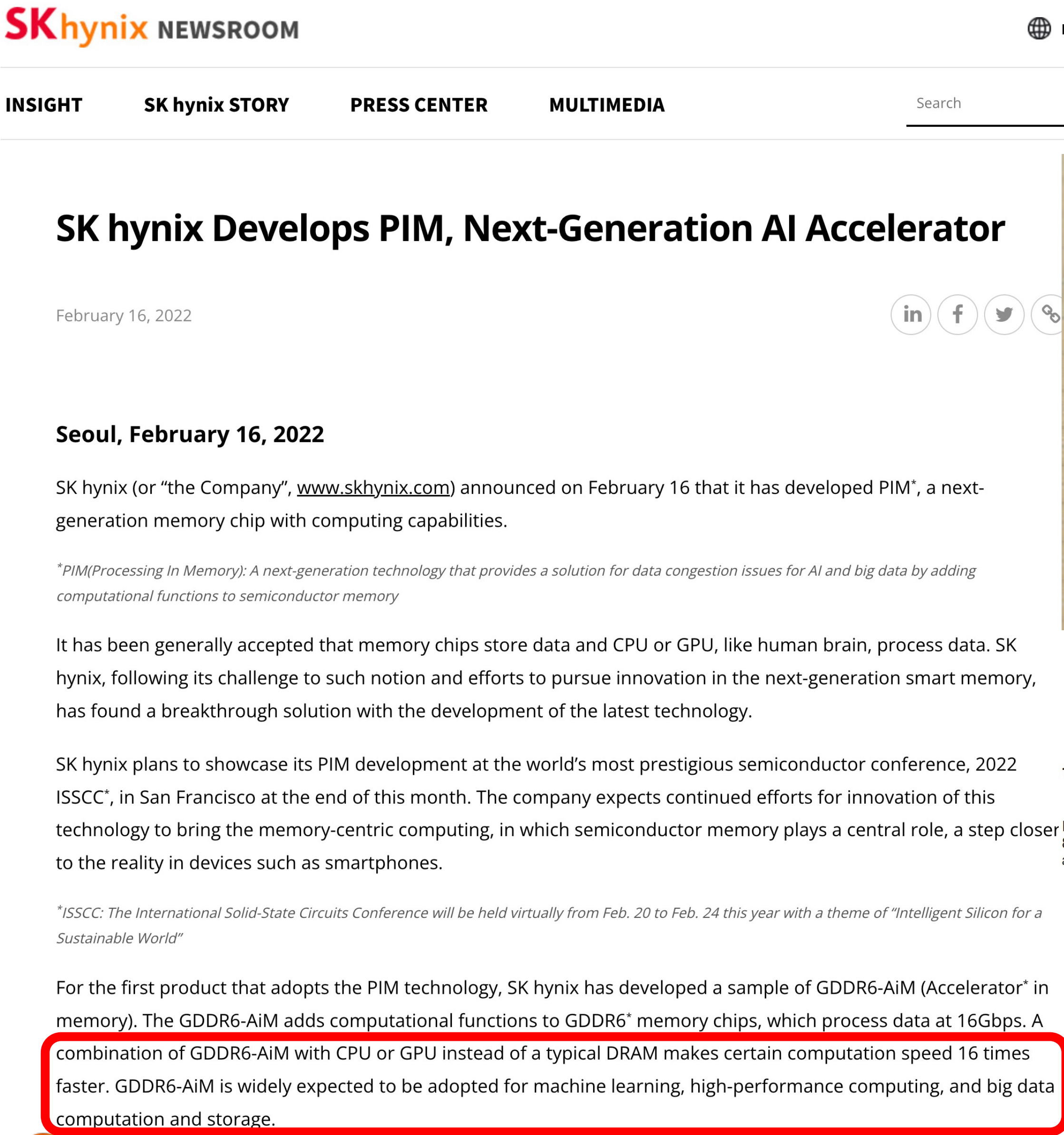
ISSCC 2021 / SESSION 25 / DRAM / 25.4

25.4 A 20nm 6GB Function-In-Memory DRAM, Based on HBM2 with a 1.2TFLOPS Programmable Computing Unit Using Bank-Level Parallelism, for Machine Learning Applications

Young-Cheon Kwon[1], Suk Han Lee[1], Jaehoon Lee[1], Sang-Hyuk Kwon[1], Je Min Ryu[1], Jong-Pil Son[1], Seongil O[1], Hak-Soo Yu[1], Haesuk Lee[1], Soo Young Kim[1], Youngmin Cho[1], Jin Guk Kim[1], Jongyoon Choi[1], Hyun-Sung Shin[1], Jin Kim[1], BengSeng Phuah[1], HyoungMin Kim[1], Myeong Jun Song[1], Ahn Choi[1], Daeho Kim[1], SooYoung Kim[1], Eun-Bong Kim[1], David Wang[2], Shinhaeng Kang[1], Yuhwan Ro[3], Seungwoo Seo[3], JoonHo Song[3], Jaeyoun Youn[1], Kyomin Sohn[1], Nam Sung Kim[1]

[1]Samsung Electronics, Hwaseong, Korea
[2]Samsung Electronics, San Jose, CA
[3]Samsung Electronics, Suwon, Korea

SK Hynix Accelerator-in-Memory (2022)



SK hynix Develops PIM, Next-Generation AI Accelerator

February 16, 2022

Seoul, February 16, 2022

SK hynix (or "the Company", www.skhynix.com) announced on February 16 that it has developed PIM*, a next-generation memory chip with computing capabilities.

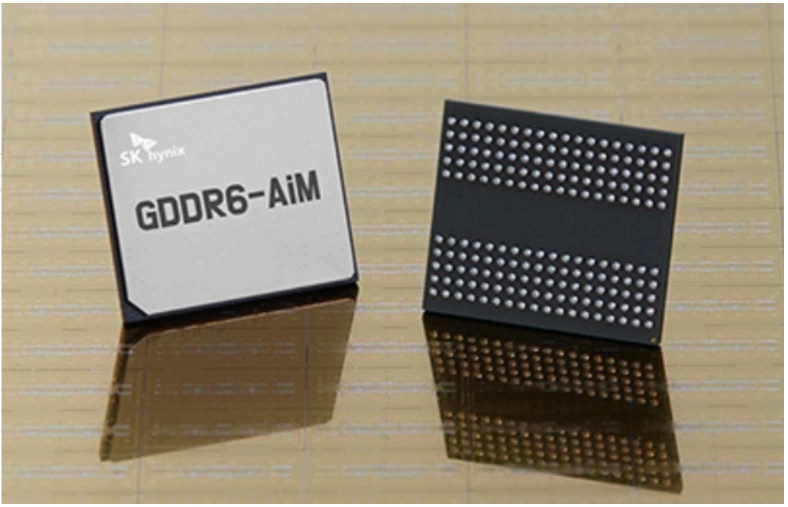
**PIM(Processing In Memory): A next-generation technology that provides a solution for data congestion issues for AI and big data by adding computational functions to semiconductor memory*

It has been generally accepted that memory chips store data and CPU or GPU, like human brain, process data. SK hynix, following its challenge to such notion and efforts to pursue innovation in the next-generation smart memory, has found a breakthrough solution with the development of the latest technology.

SK hynix plans to showcase its PIM development at the world's most prestigious semiconductor conference, 2022 ISSCC*, in San Francisco at the end of this month. The company expects continued efforts for innovation of this technology to bring the memory-centric computing, in which semiconductor memory plays a central role, a step closer to the reality in devices such as smartphones.

**ISSCC: The International Solid-State Circuits Conference will be held virtually from Feb. 20 to Feb. 24 this year with a theme of "Intelligent Silicon for a Sustainable World"*

For the first product that adopts the PIM technology, SK hynix has developed a sample of GDDR6-AiM (Accelerator* in memory). The GDDR6-AiM adds computational functions to GDDR6* memory chips, which process data at 16Gbps. A combination of GDDR6-AiM with CPU or GPU instead of a typical DRAM makes certain computation speed 16 times faster. GDDR6-AiM is widely expected to be adopted for machine learning, high-performance computing, and big data computation and storage.



11.1 A 1ynm 1.25V 8Gb, 16Gb/s/pin GDDR6-based Accelerator-in-Memory supporting 1TFLOPS MAC Operation and Various Activation Functions for Deep-Learning Applications

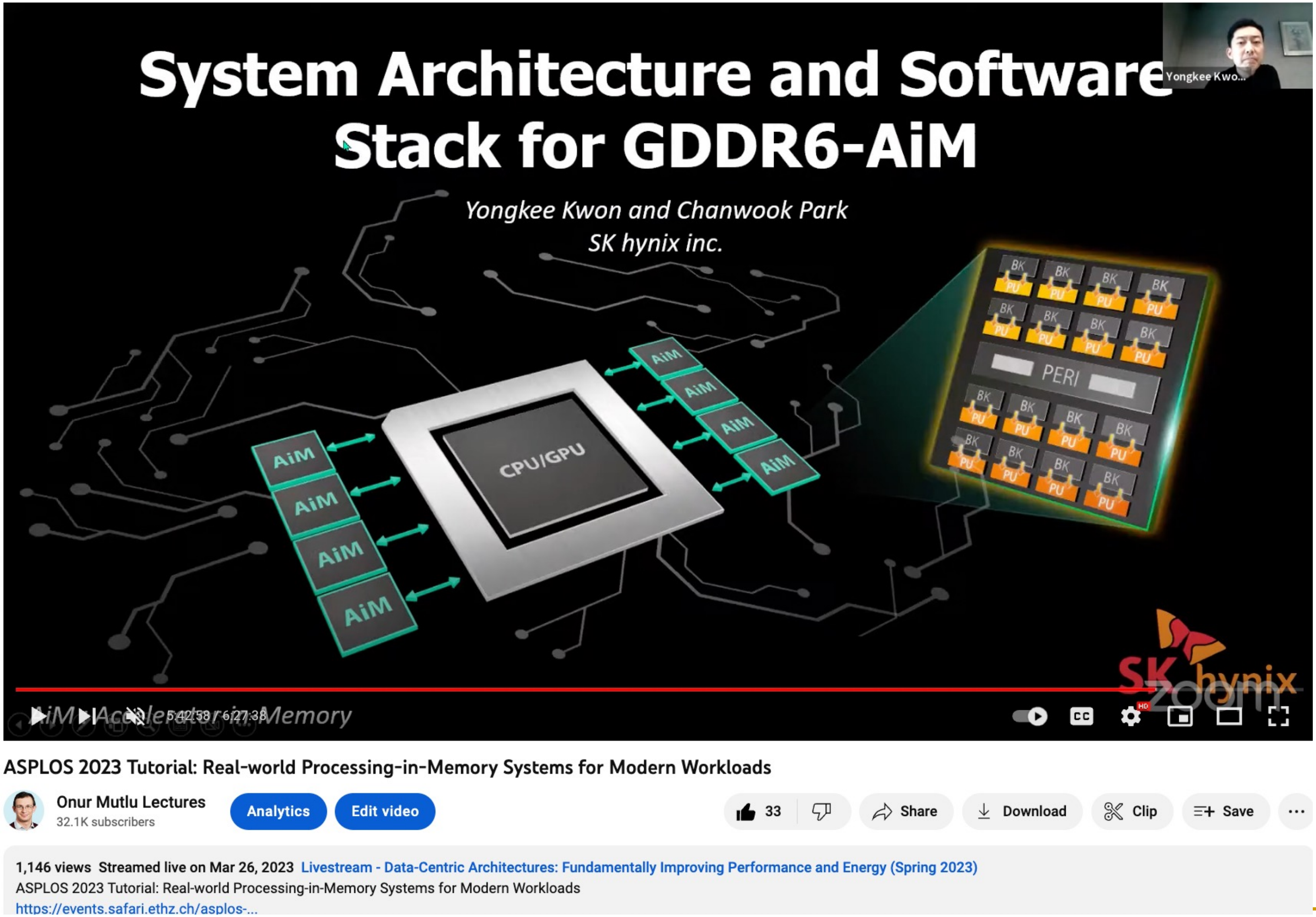
Seongju Lee, SK hynix, Icheon, Korea

In Paper 11.1, SK Hynix describes an 1ynm, GDDR6-based accelerator-in-memory with a command set for deep-learning operation. The 8Gb design achieves a peak throughput of 1TFLOPS with 1GHz MAC operations and supports major activation functions to improve accuracy.

SAFARI

https://news.skhynix.com/sk-hynix-develops-pim-next-generation-ai-accelerator/

SK Hynix Accelerator-in-Memory (2022)



System Architecture and Software Stack for GDDR6-AiM

Yongkee Kwon and Chanwook Park
SK hynix inc.

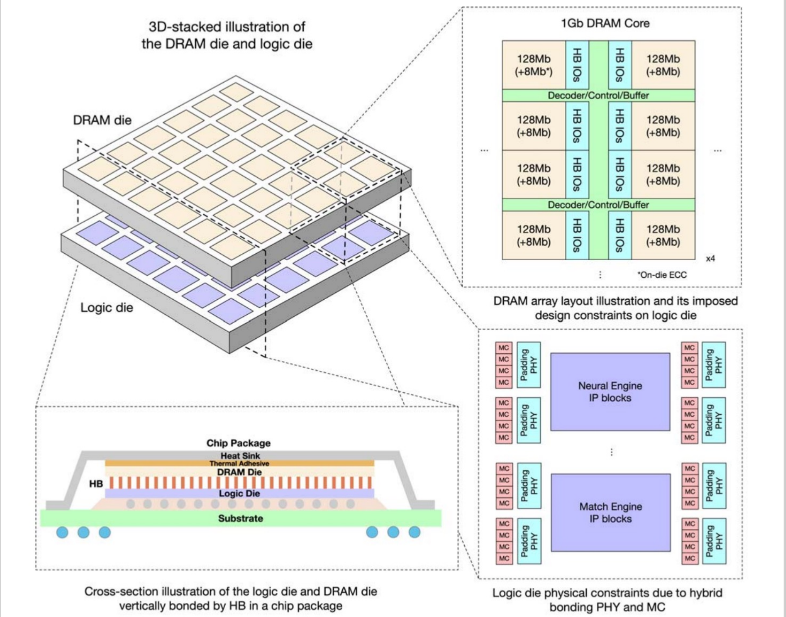
ASPLOS 2023 Tutorial: Real-world Processing-in-Memory Systems for Modern Workloads

1,146 views Streamed live on Mar 26, 2023 Livestream - Data-Centric Architectures: Fundamentally Improving Performance and Energy (Spring 2023)
ASPLOS 2023 Tutorial: Real-world Processing-in-Memory Systems for Modern Workloads
https://events.safari.ethz.ch/asplos-...

https://www.youtube.com/watch?v=oYCaLcT0Kmo

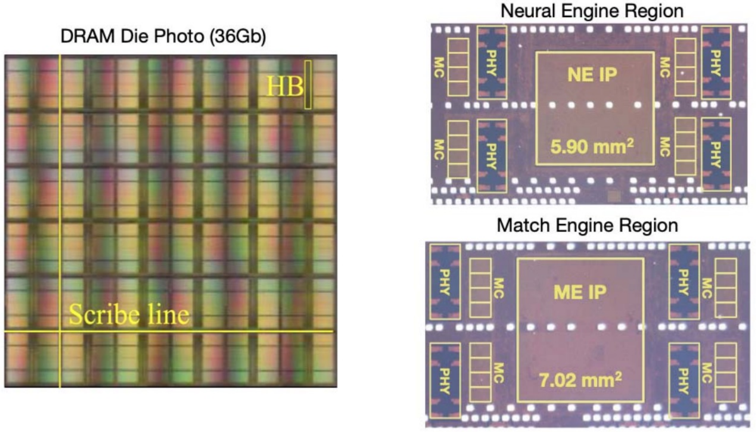
AliBaba PIM Recommendation System (2022)

ISSCC 2022 / February 24, 2022 / 8:30 AM



3D-stacked illustration of the DRAM die and logic die
DRAM die
Logic die
1Gb DRAM Core
128Mb (+8Mb*)
HB IOs
128Mb (+8Mb)
Decoder/Control/Buffer
x4
*On-die ECC
DRAM array layout illustration and its imposed design constraints on logic die
MC
Padding PHY
Neural Engine IP blocks
Match Engine IP blocks
Chip Package
Heat Sink
Thermal Adhesive
DRAM Die
HB
Logic Die
Substrate
Cross-section illustration of the logic die and DRAM die vertically bonded by HB in a chip package
Logic die physical constraints due to hybrid bonding PHY and MC

Figure 29.1.2: Illustration of 3D-stacked chip, cross-illustration of package, DRAM array layout and design blocks on logic die.



DRAM Die		
Technology	25nm	
Area	Total*	602.22 mm²
	Neural Engine	32 mm²
	Match Engine	32 mm²
Voltage	1.1 V	
Frequency (max)	150 MHz	
Power	300 mW per 1Gb	
Bandwidth**	153.60 GB/s / 1.38 TB/s	

* Total die size of test chip
** Memory bandwidth of NE and ME / Total bandwidth of DRAM die

Logic Die		
Technology	55nm	
Area	Total*	602.22 mm²
	Neural Engine	5.90 mm²
	Match Engine	7.02 mm²
# of MC	16 per IP	
Voltage	1.2 V	
Frequency	300 MHz	
Power	977.70 mW	
Precision	INT8	

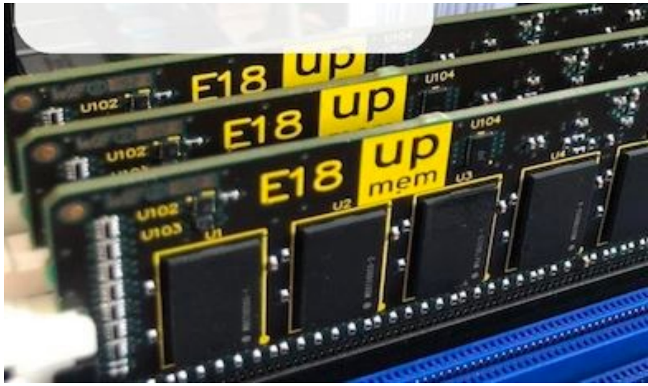
Figure 29.1.7: Die micrographs of DRAM die, NE and ME. Detailed specifications of DRAM die and logic die.

29.1 184QPS/W 64Mb/mm² 3D Logic-to-DRAM Hybrid Bonding with Process-Near-Memory Engine for Recommendation System

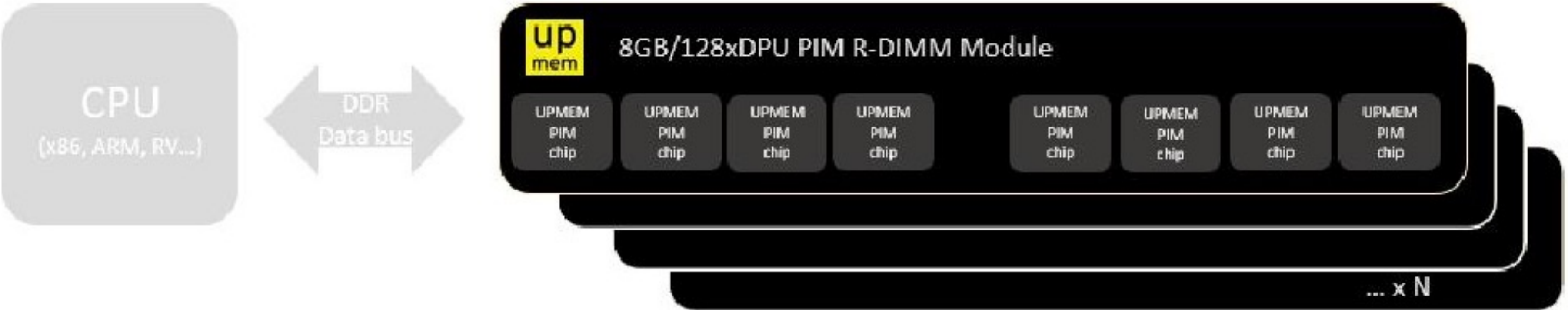
Dimin Niu[1], Shuangchen Li[1], Yuhao Wang[1], Wei Han[1], Zhe Zhang[2], Yijin Guan[2], Tianchan Guan[3], Fei Sun[1], Fei Xue[1], Lide Duan[1], Yuanwei Fang[1], Hongzhong Zheng[1], Xiping Jiang[4], Song Wang[4], Fengguo Zuo[4], Yubing Wang[4], Bing Yu[4], Qiwei Ren[4], Yuan Xie[1]

UPMEM Processing-in-DRAM Engine (2019)

- Processing in DRAM Engine
- Includes **standard DIMM modules**, with a **large number of DPU processors** combined with DRAM chips.



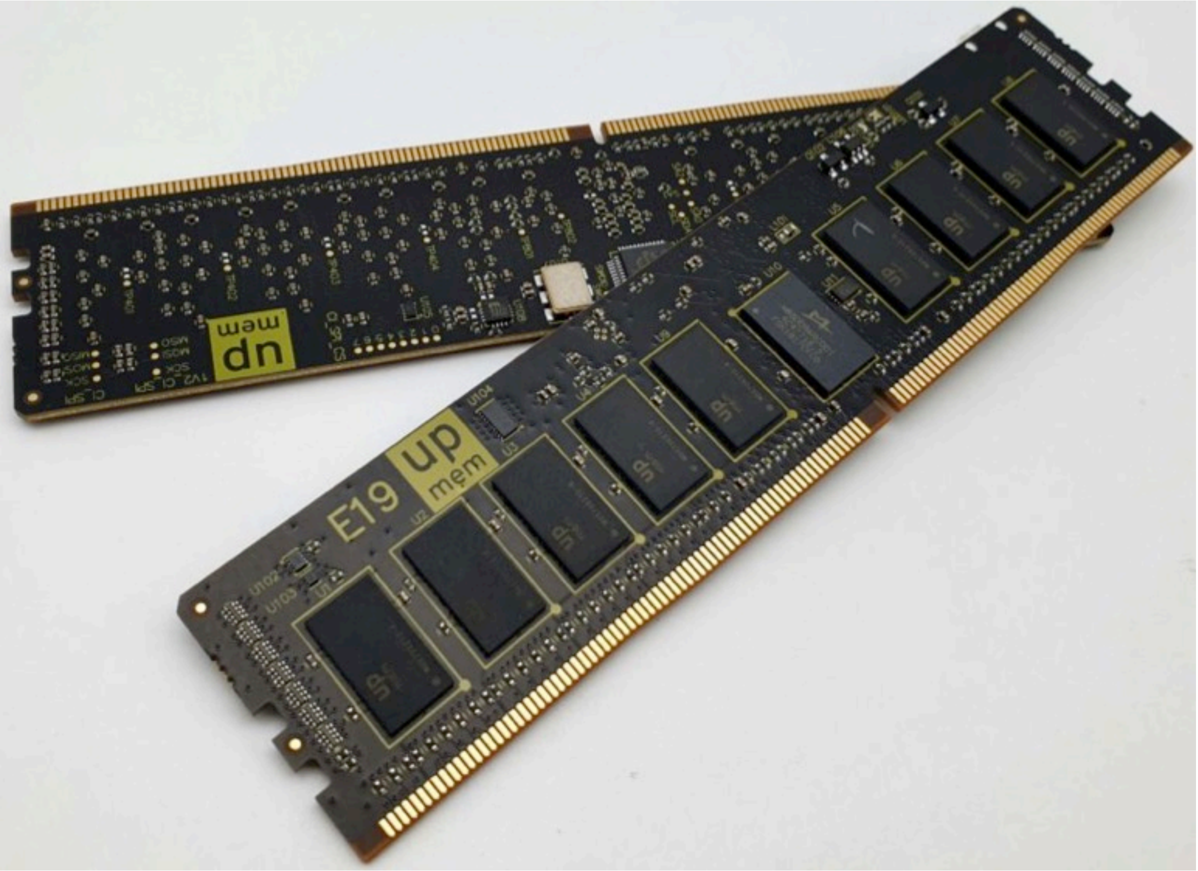
- Replaces **standard** DIMMs
  - DDR4 R-DIMM modules
    - 8GB+128 DPUs (16 PIM chips)
    - Standard 2x-nm DRAM process
  - **Large amounts of** compute & memory bandwidth



https://www.anandtech.com/show/14750/hot-chips-31-analysis-inmemory-processing-by-upmem
https://www.upmem.com/video-upmem-presenting-its-true-processing-in-memory-solution-hot-chips-2019/

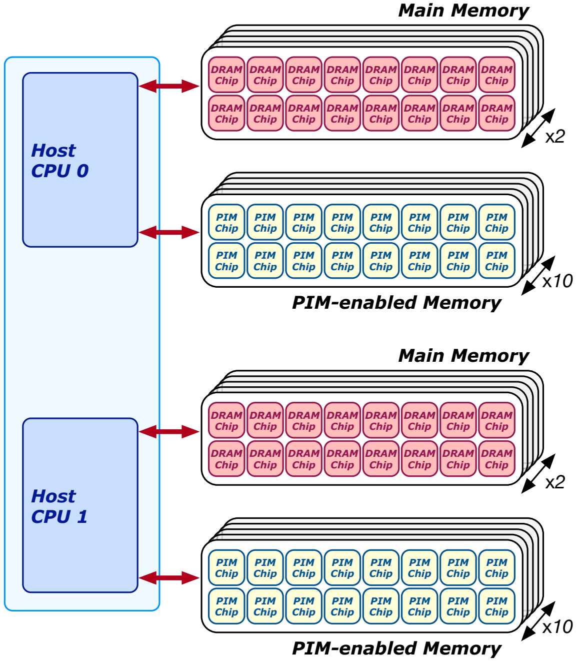
UPMEM Memory Modules

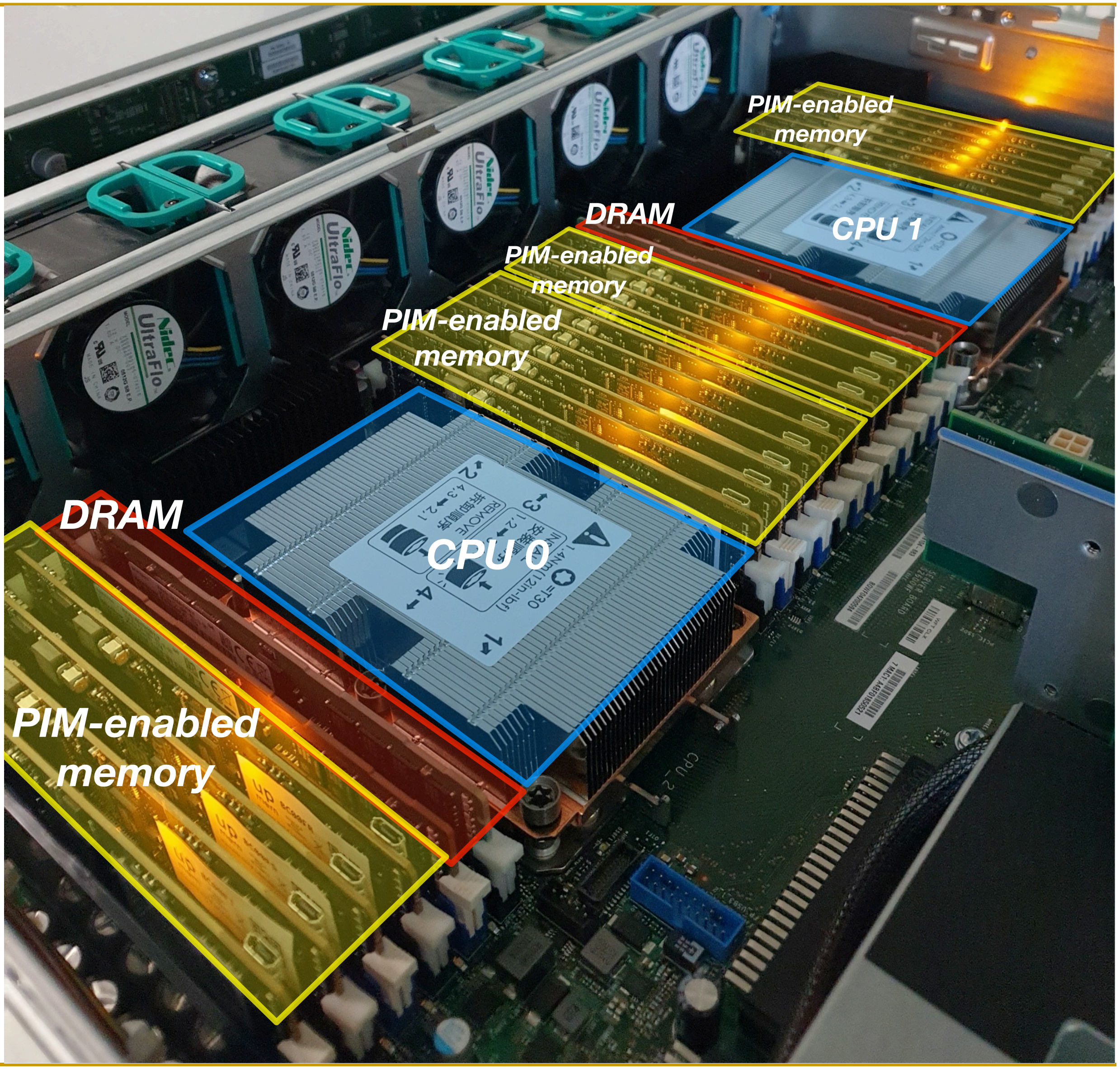
- E19: 8 chips DIMM (1 rank). DPUs @ 267 MHz
- P21: 16 chips DIMM (2 ranks). DPUs @ 350 MHz



www.upmem.com

2,560-DPU Processing-in-Memory System





Benchmarking a New Paradigm: An Experimental Analysis of a Real Processing-in-Memory Architecture

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland
IZZAT EL HAJJ, American University of Beirut, Lebanon
IVAN FERNANDEZ, ETH Zürich, Switzerland and University of Malaga, Spain
CHRISTINA GIANNOULA, ETH Zürich, Switzerland and NTUA, Greece
GERALDO F. OLIVEIRA, ETH Zürich, Switzerland
ONUR MUTLU, ETH Zürich, Switzerland

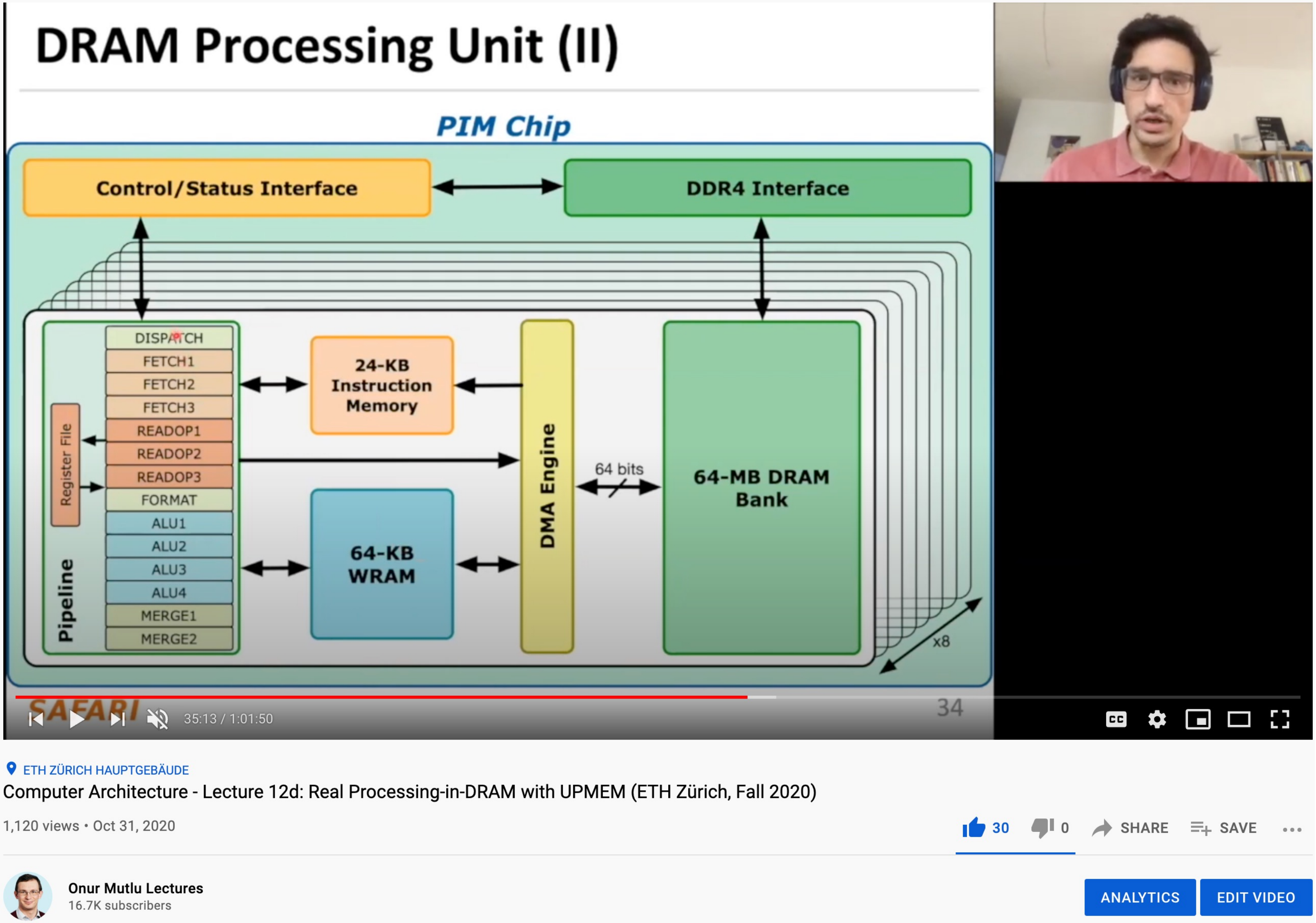
Many modern workloads, such as neural networks, databases, and graph processing, are fundamentally memory-bound. For such workloads, the data movement between main memory and CPU cores imposes a significant overhead in terms of both latency and energy. A major reason is that this communication happens through a narrow bus with high latency and limited bandwidth, and the low data reuse in memory-bound workloads is insufficient to amortize the cost of main memory access. Fundamentally addressing this *data movement bottleneck* requires a paradigm where the memory system assumes an active role in computing by integrating processing capabilities. This paradigm is known as *processing-in-memory* (*PIM*).

Recent research explores different forms of PIM architectures, motivated by the emergence of new 3D-stacked memory technologies that integrate memory with a logic layer where processing elements can be easily placed. Past works evaluate these architectures in simulation or, at best, with simplified hardware prototypes. In contrast, the UPMEM company has designed and manufactured the first publicly-available real-world PIM architecture. The UPMEM PIM architecture combines traditional DRAM memory arrays with general-purpose in-order cores, called *DRAM Processing Units* (*DPUs*), integrated in the same chip.

This paper provides the first comprehensive analysis of the first publicly-available real-world PIM architecture. We make two key contributions. First, we conduct an experimental characterization of the UPMEM-based PIM system using microbenchmarks to assess various architecture limits such as compute throughput and memory bandwidth, yielding new insights. Second, we present *PrIM* (*Processing-In-Memory benchmarks*), a benchmark suite of 16 workloads from different application domains (e.g., dense/sparse linear algebra, databases, data analytics, graph processing, neural networks, bioinformatics, image processing), which we identify as memory-bound. We evaluate the performance and scaling characteristics of PrIM benchmarks on the UPMEM PIM architecture, and compare their performance and energy consumption to their state-of-the-art CPU and GPU counterparts. Our extensive evaluation conducted on two real UPMEM-based PIM systems with 640 and 2,556 DPUs provides new insights about suitability of different workloads to the PIM system, programming recommendations for software designers, and suggestions and hints for hardware and architecture designers of future PIM systems.

https://arxiv.org/pdf/2105.03814.pdf

More on the UPMEM PIM System



DRAM Processing Unit (II)

ETH ZÜRICH HAUPTGEBÄUDE

Computer Architecture - Lecture 12d: Real Processing-in-DRAM with UPMEM (ETH Zürich, Fall 2020)

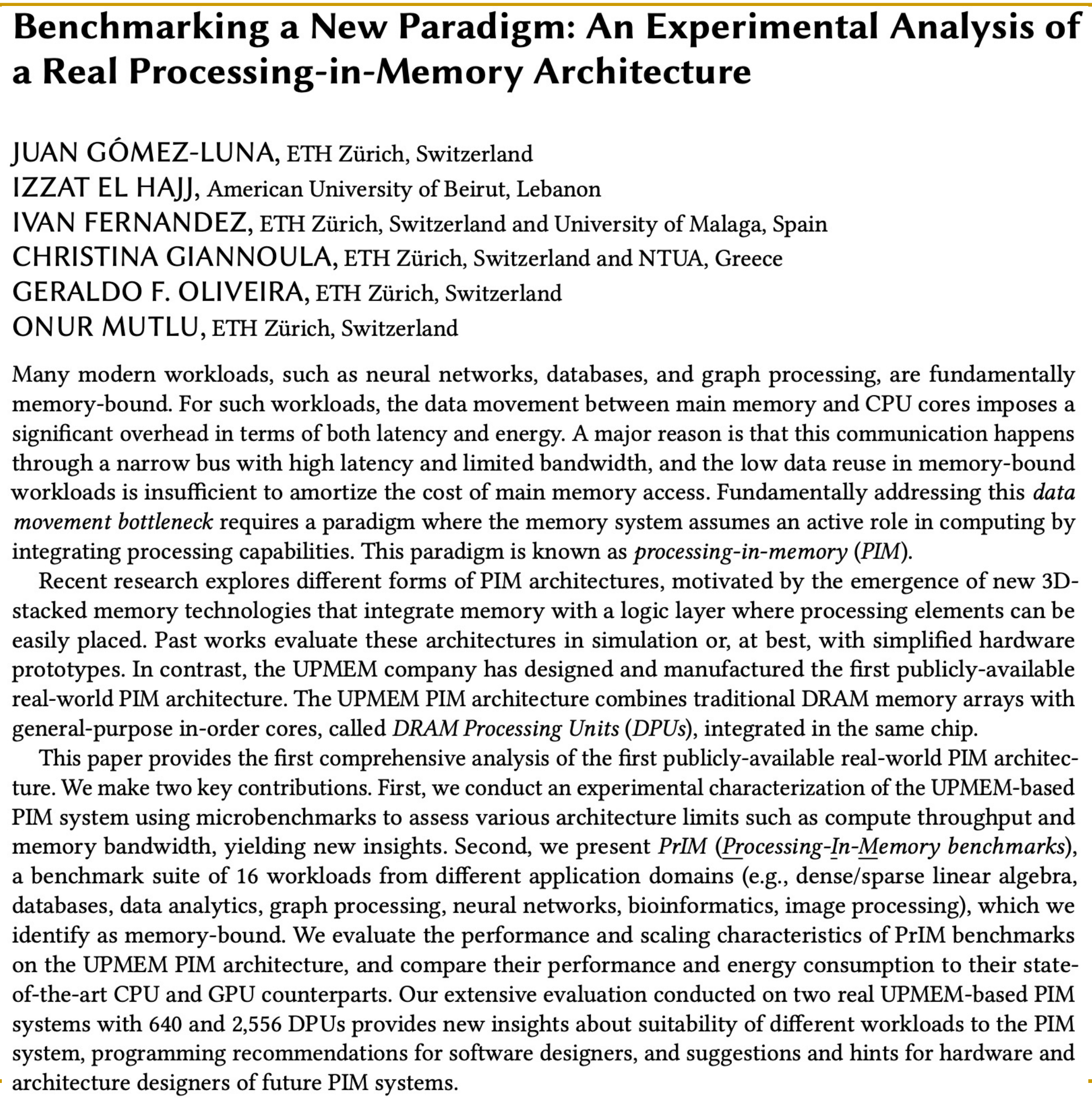
1,120 views • Oct 31, 2020

Onur Mutlu Lectures

16.7K subscribers

https://www.youtube.com/watch?v=Sscy1Wrr22A&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=26

Experimental Analysis of the UPMEM PIM Engine



Benchmarking a New Paradigm: An Experimental Analysis of a Real Processing-in-Memory Architecture

JUAN GÓMEZ-LUNA, ETH Zürich, Switzerland
IZZAT EL HAJJ, American University of Beirut, Lebanon
IVAN FERNANDEZ, ETH Zürich, Switzerland and University of Malaga, Spain
CHRISTINA GIANNOULA, ETH Zürich, Switzerland and NTUA, Greece
GERALDO F. OLIVEIRA, ETH Zürich, Switzerland
ONUR MUTLU, ETH Zürich, Switzerland

Many modern workloads, such as neural networks, databases, and graph processing, are fundamentally memory-bound. For such workloads, the data movement between main memory and CPU cores imposes a significant overhead in terms of both latency and energy. A major reason is that this communication happens through a narrow bus with high latency and limited bandwidth, and the low data reuse in memory-bound workloads is insufficient to amortize the cost of main memory access. Fundamentally addressing this *data movement bottleneck* requires a paradigm where the memory system assumes an active role in computing by integrating processing capabilities. This paradigm is known as *processing-in-memory* (*PIM*).

Recent research explores different forms of PIM architectures, motivated by the emergence of new 3D-stacked memory technologies that integrate memory with a logic layer where processing elements can be easily placed. Past works evaluate these architectures in simulation or, at best, with simplified hardware prototypes. In contrast, the UPMEM company has designed and manufactured the first publicly-available real-world PIM architecture. The UPMEM PIM architecture combines traditional DRAM memory arrays with general-purpose in-order cores, called *DRAM Processing Units* (*DPUs*), integrated in the same chip.

This paper provides the first comprehensive analysis of the first publicly-available real-world PIM architecture. We make two key contributions. First, we conduct an experimental characterization of the UPMEM-based PIM system using microbenchmarks to assess various architecture limits such as compute throughput and memory bandwidth, yielding new insights. Second, we present *PrIM* (*Processing-In-Memory benchmarks*), a benchmark suite of 16 workloads from different application domains (e.g., dense/sparse linear algebra, databases, data analytics, graph processing, neural networks, bioinformatics, image processing), which we identify as memory-bound. We evaluate the performance and scaling characteristics of PrIM benchmarks on the UPMEM PIM architecture, and compare their performance and energy consumption to their state-of-the-art CPU and GPU counterparts. Our extensive evaluation conducted on two real UPMEM-based PIM systems with 640 and 2,556 DPUs provides new insights about suitability of different workloads to the PIM system, programming recommendations for software designers, and suggestions and hints for hardware and architecture designers of future PIM systems.

https://arxiv.org/pdf/2105.03814.pdf

UPMEM PIM System Summary & Analysis

- Juan Gomez-Luna, Izzat El Hajj, Ivan Fernandez, Christina Giannoula, Geraldo F. Oliveira, and Onur Mutlu,
  "Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-in-Memory Hardware"
  *Invited Paper at Workshop on Computing with Unconventional Technologies (**CUT**)*, Virtual, October 2021.
  [arXiv version]
  [PrIM Benchmarks Source Code]
  [Slides (pptx) (pdf)]
  [Talk Video (37 minutes)]
  [Lightning Talk Video (3 minutes)]

Benchmarking Memory-Centric Computing Systems: Analysis of Real Processing-in-Memory Hardware

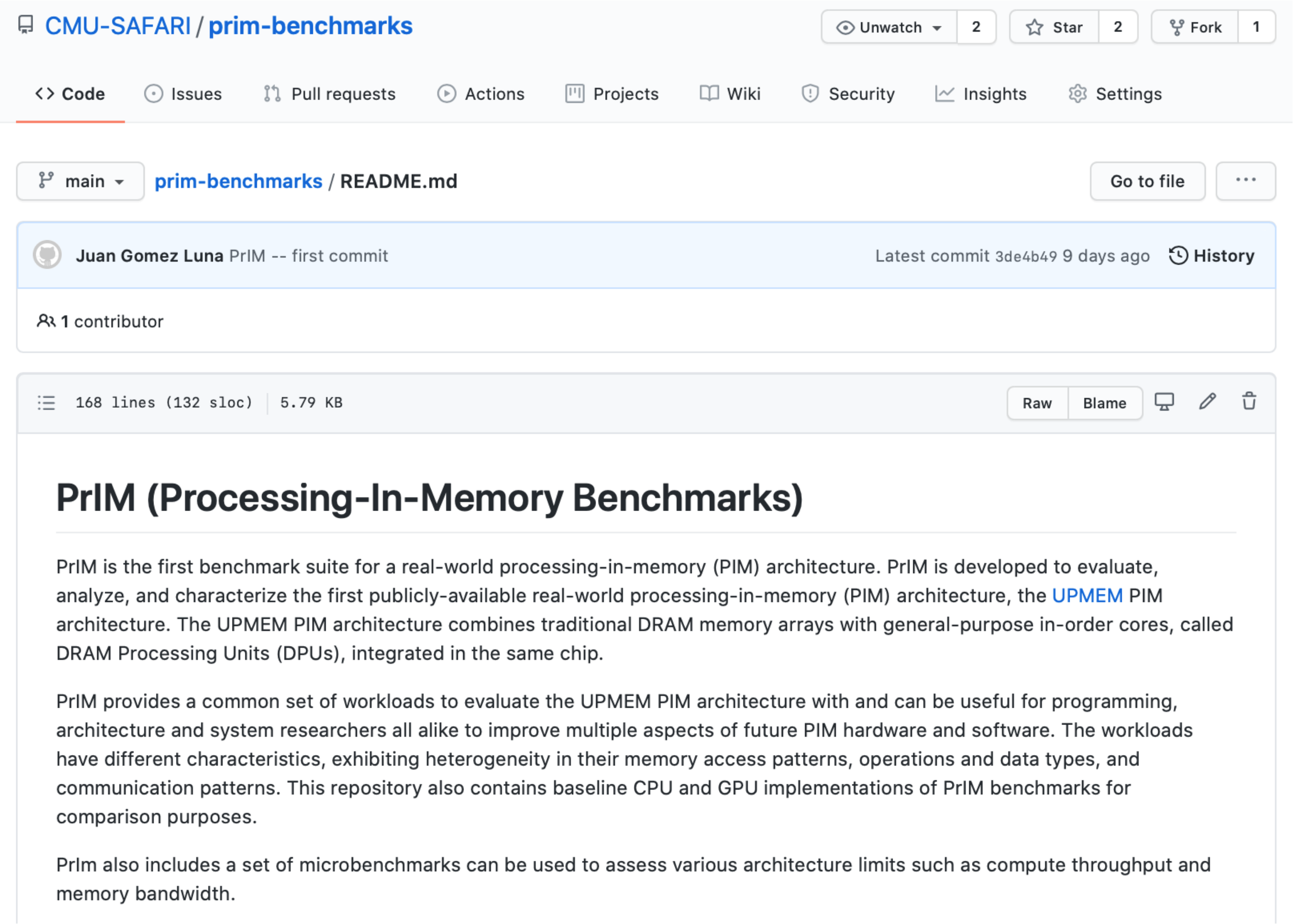
Juan Gómez-Luna, *ETH Zürich*
Izzat El Hajj, *American University of Beirut*
Ivan Fernandez, *University of Malaga*
Christina Giannoula, *National Technical University of Athens*
Geraldo F. Oliveira, *ETH Zürich*
Onur Mutlu, *ETH Zürich*

PrIM Benchmarks: Application Domains

Domain	Benchmark	Short name
Dense linear algebra	Vector Addition	VA
	Matrix-Vector Multiply	GEMV
Sparse linear algebra	Sparse Matrix-Vector Multiply	SpMV
Databases	Select	SEL
	Unique	UNI
Data analytics	Binary Search	BS
	Time Series Analysis	TS
Graph processing	Breadth-First Search	BFS
Neural networks	Multilayer Perceptron	MLP
Bioinformatics	Needleman-Wunsch	NW
Image processing	Image histogram (short)	HST-S
	Image histogram (large)	HST-L
Parallel primitives	Reduction	RED
	Prefix sum (scan-scan-add)	SCAN-SSA
	Prefix sum (reduce-scan-scan)	SCAN-RSS
	Matrix transposition	TRNS

PrIM Benchmarks are Open Source

- All microbenchmarks, benchmarks, and scripts
- https://github.com/CMU-SAFARI/prim-benchmarks


CMU-SAFARI / prim-benchmarks
Unwatch 2
Star 2
Fork 1
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
main
prim-benchmarks / README.md
Go to file
Juan Gomez Luna PrIM -- first commit
Latest commit 3de4b49 9 days ago
History
1 contributor
168 lines (132 sloc) | 5.79 KB
Raw
Blame

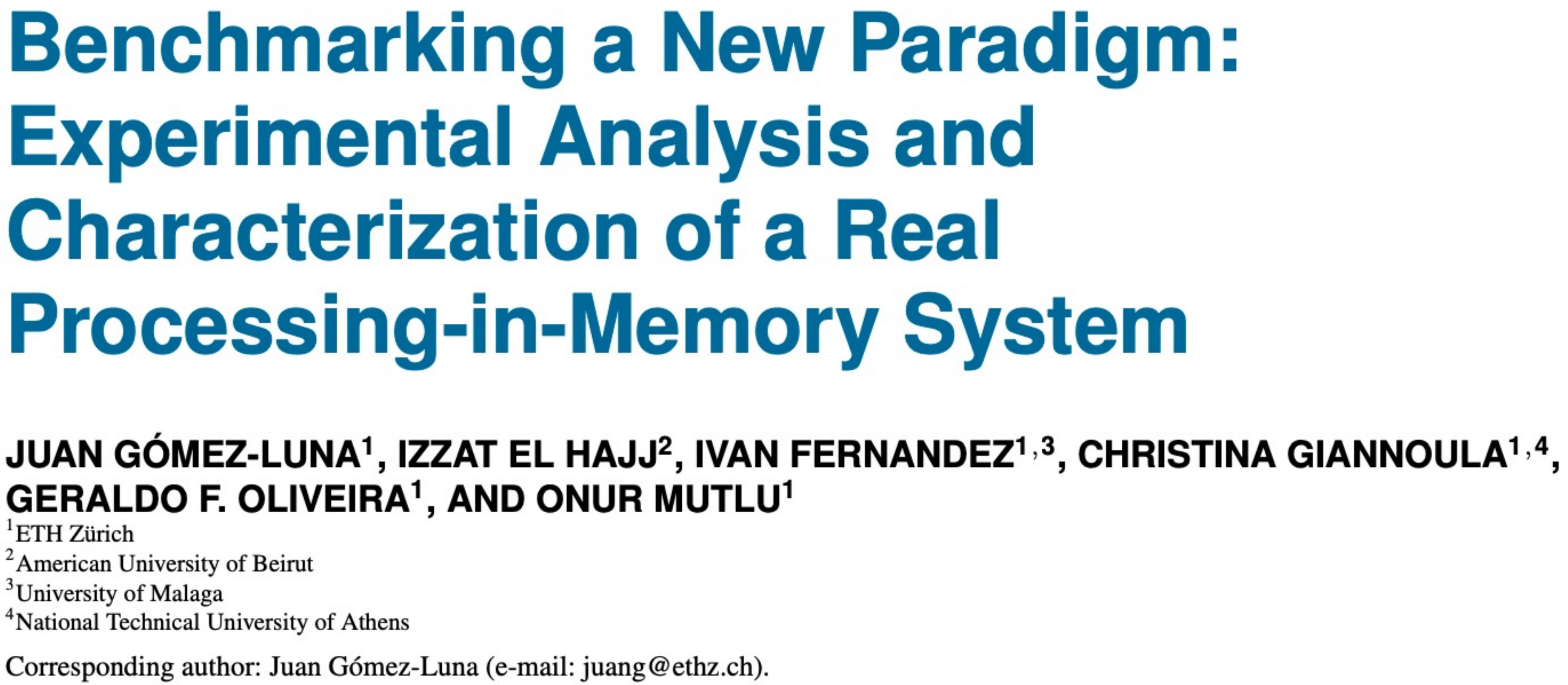
PrIM (Processing-In-Memory Benchmarks)

PrIM is the first benchmark suite for a real-world processing-in-memory (PIM) architecture. PrIM is developed to evaluate, analyze, and characterize the first publicly-available real-world processing-in-memory (PIM) architecture, the UPMEM PIM architecture. The UPMEM PIM architecture combines traditional DRAM memory arrays with general-purpose in-order cores, called DRAM Processing Units (DPUs), integrated in the same chip.

PrIM provides a common set of workloads to evaluate the UPMEM PIM architecture with and can be useful for programming, architecture and system researchers all alike to improve multiple aspects of future PIM hardware and software. The workloads have different characteristics, exhibiting heterogeneity in their memory access patterns, operations and data types, and communication patterns. This repository also contains baseline CPU and GPU implementations of PrIM benchmarks for comparison purposes.

PrIm also includes a set of microbenchmarks can be used to assess various architecture limits such as compute throughput and memory bandwidth.

Understanding a Modern PIM Architecture



Benchmarking a New Paradigm: Experimental Analysis and Characterization of a Real Processing-in-Memory System

JUAN GÓMEZ-LUNA[1], IZZAT EL HAJJ[2], IVAN FERNANDEZ[1,3], CHRISTINA GIANNOULA[1,4], GERALDO F. OLIVEIRA[1], AND ONUR MUTLU[1]

[1]ETH Zürich
[2]American University of Beirut
[3]University of Malaga
[4]National Technical University of Athens

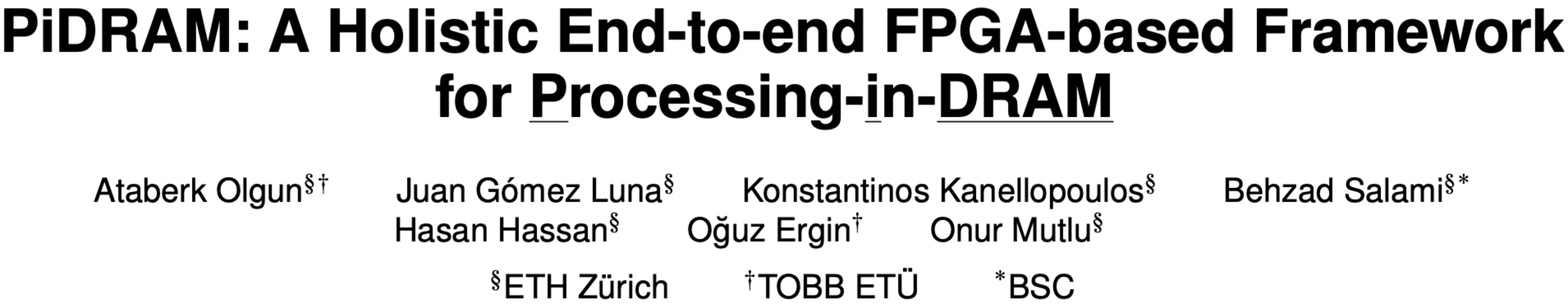
Corresponding author: Juan Gómez-Luna (e-mail: juang@ethz.ch).

https://arxiv.org/pdf/2105.03814.pdf

https://github.com/CMU-SAFARI/prim-benchmarks

Real Processing Using Memory Prototype

- End-to-end RowClone & TRNG using off-the-shelf DRAM chips
- Idea: Violate DRAM timing parameters to mimic RowClone



PiDRAM: A Holistic End-to-end FPGA-based Framework for Processing-in-DRAM

Ataberk Olgun[§, †] Juan Gómez Luna[§] Konstantinos Kanellopoulos[§] Behzad Salami[§, *]
Hasan Hassan[§] Oğuz Ergin[†] Onur Mutlu[§]

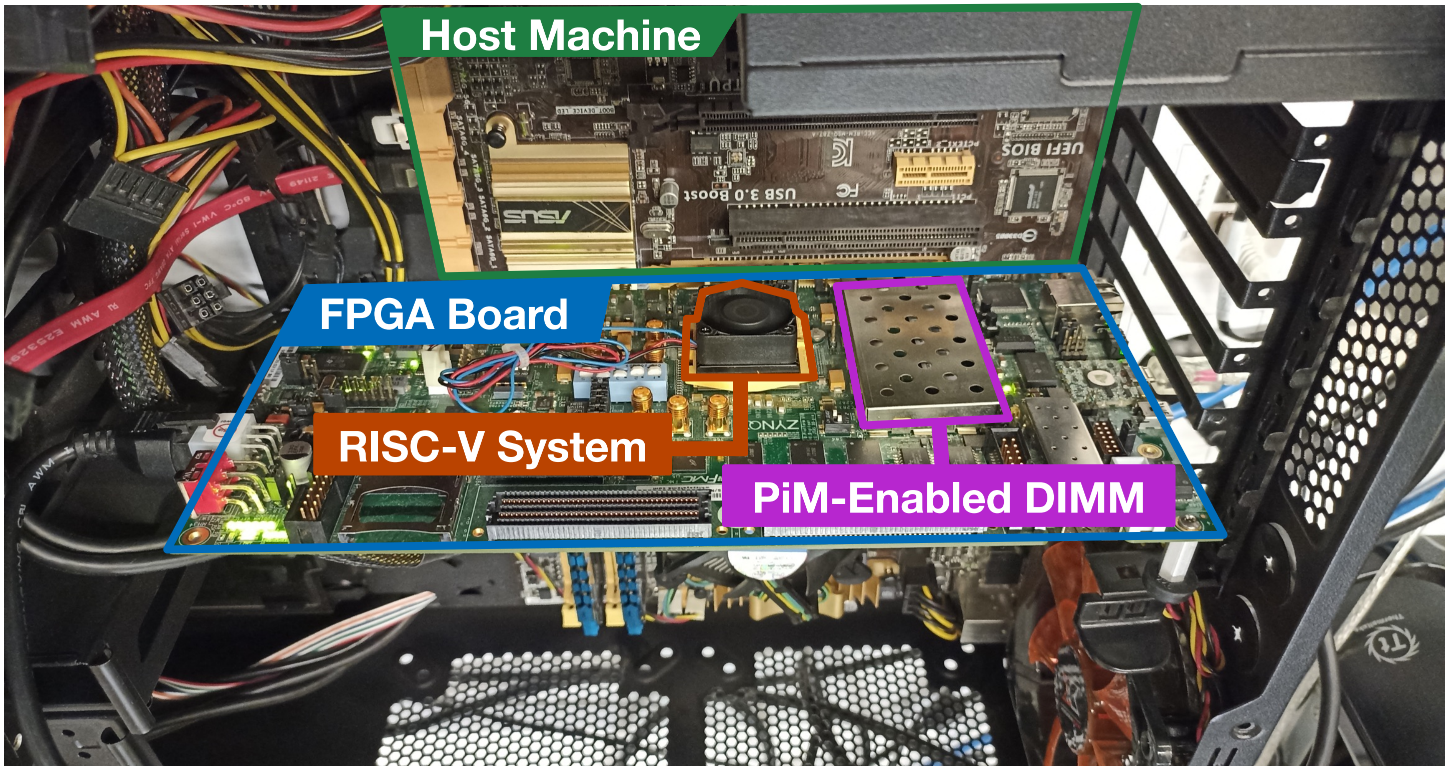
[§]ETH Zürich [†]TOBB ETÜ [*]BSC

https://arxiv.org/pdf/2111.00082.pdf

https://github.com/cmu-safari/pidram

https://www.youtube.com/watch?v=qeukNs5XI3g&t=4192s

Real Processing Using Memory Prototype

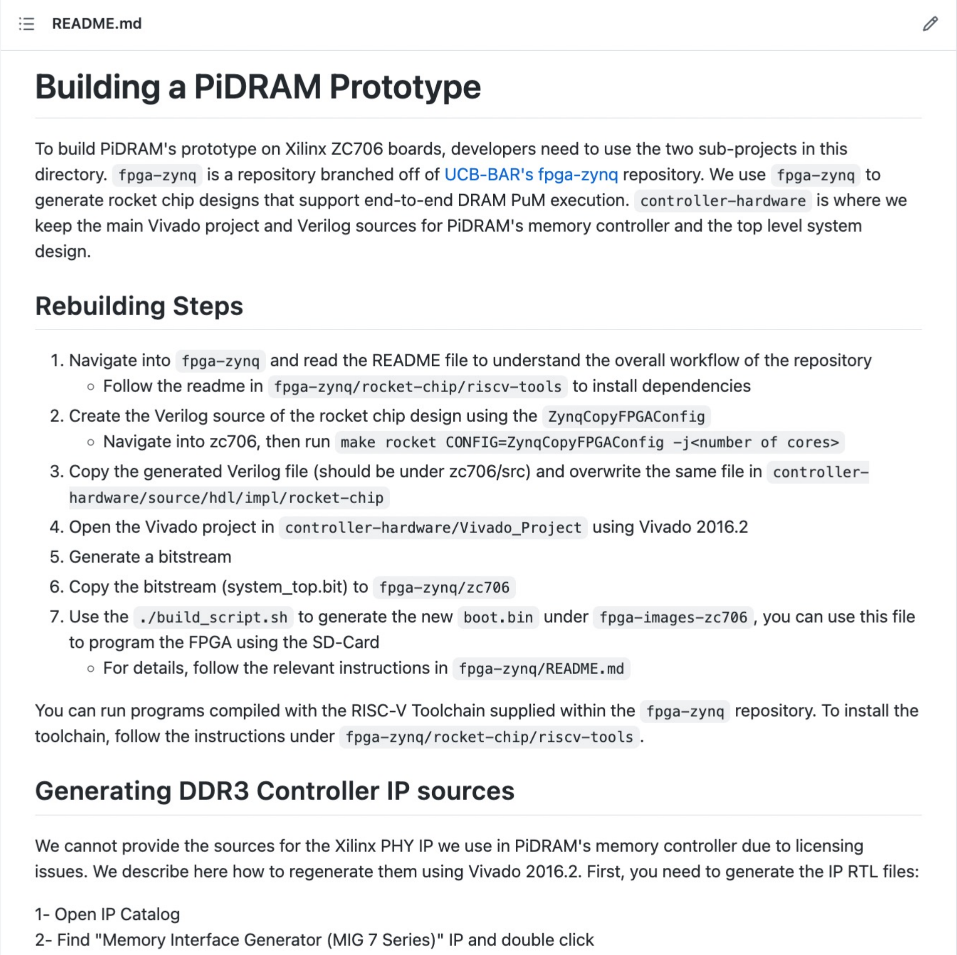


https://arxiv.org/pdf/2111.00082.pdf

https://github.com/cmu-safari/pidram

https://www.youtube.com/watch?v=qeukNs5XI3g&t=4192s

Real Processing Using Memory Prototype



README.md

Building a PiDRAM Prototype

To build PiDRAM's prototype on Xilinx ZC706 boards, developers need to use the two sub-projects in this directory. `fpga-zynq` is a repository branched off of UCB-BAR's fpga-zynq repository. We use `fpga-zynq` to generate rocket chip designs that support end-to-end DRAM PuM execution. `controller-hardware` is where we keep the main Vivado project and Verilog sources for PiDRAM's memory controller and the top level system design.

Rebuilding Steps

1. Navigate into `fpga-zynq` and read the README file to understand the overall workflow of the repository
   - Follow the readme in `fpga-zynq/rocket-chip/riscv-tools` to install dependencies
2. Create the Verilog source of the rocket chip design using the `ZynqCopyFPGAConfig`
   - Navigate into zc706, then run `make rocket CONFIG=ZynqCopyFPGAConfig -j<number of cores>`
3. Copy the generated Verilog file (should be under zc706/src) and overwrite the same file in `controller-hardware/source/hdl/impl/rocket-chip`
4. Open the Vivado project in `controller-hardware/Vivado_Project` using Vivado 2016.2
5. Generate a bitstream
6. Copy the bitstream (system_top.bit) to `fpga-zynq/zc706`
7. Use the `./build_script.sh` to generate the new `boot.bin` under `fpga-images-zc706`, you can use this file to program the FPGA using the SD-Card
   - For details, follow the relevant instructions in `fpga-zynq/README.md`

You can run programs compiled with the RISC-V Toolchain supplied within the `fpga-zynq` repository. To install the toolchain, follow the instructions under `fpga-zynq/rocket-chip/riscv-tools`.

Generating DDR3 Controller IP sources

We cannot provide the sources for the Xilinx PHY IP we use in PiDRAM's memory controller due to licensing issues. We describe here how to regenerate them using Vivado 2016.2. First, you need to generate the IP RTL files:

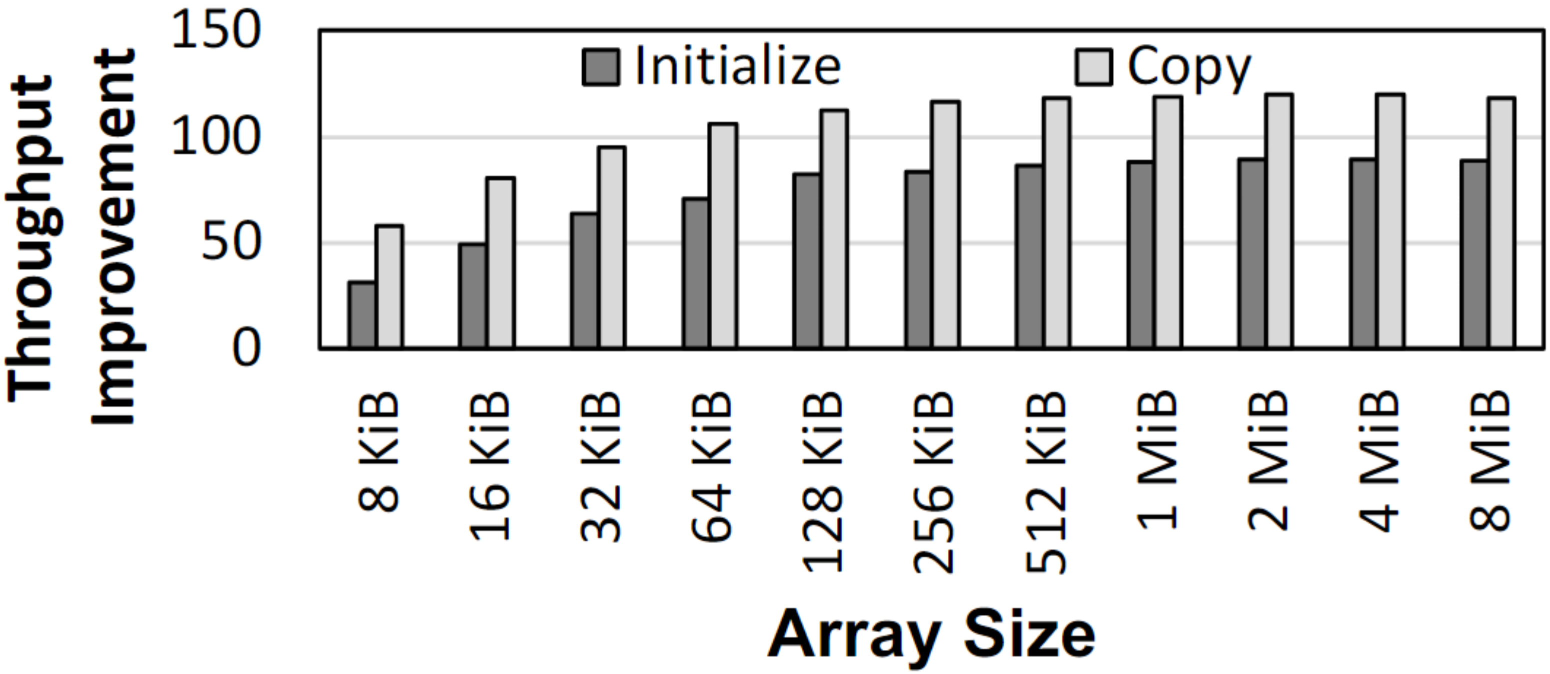
1- Open IP Catalog
2- Find "Memory Interface Generator (MIG 7 Series)" IP and double click

https://arxiv.org/pdf/2111.00082.pdf

https://github.com/cmu-safari/pidram

https://www.youtube.com/watch?v=qeukNs5XI3g&t=4192s

Microbenchmark Copy/Initialization Throughput



In-DRAM Copy and Initialization improve throughput by 119x and 89x

More on PiDRAM

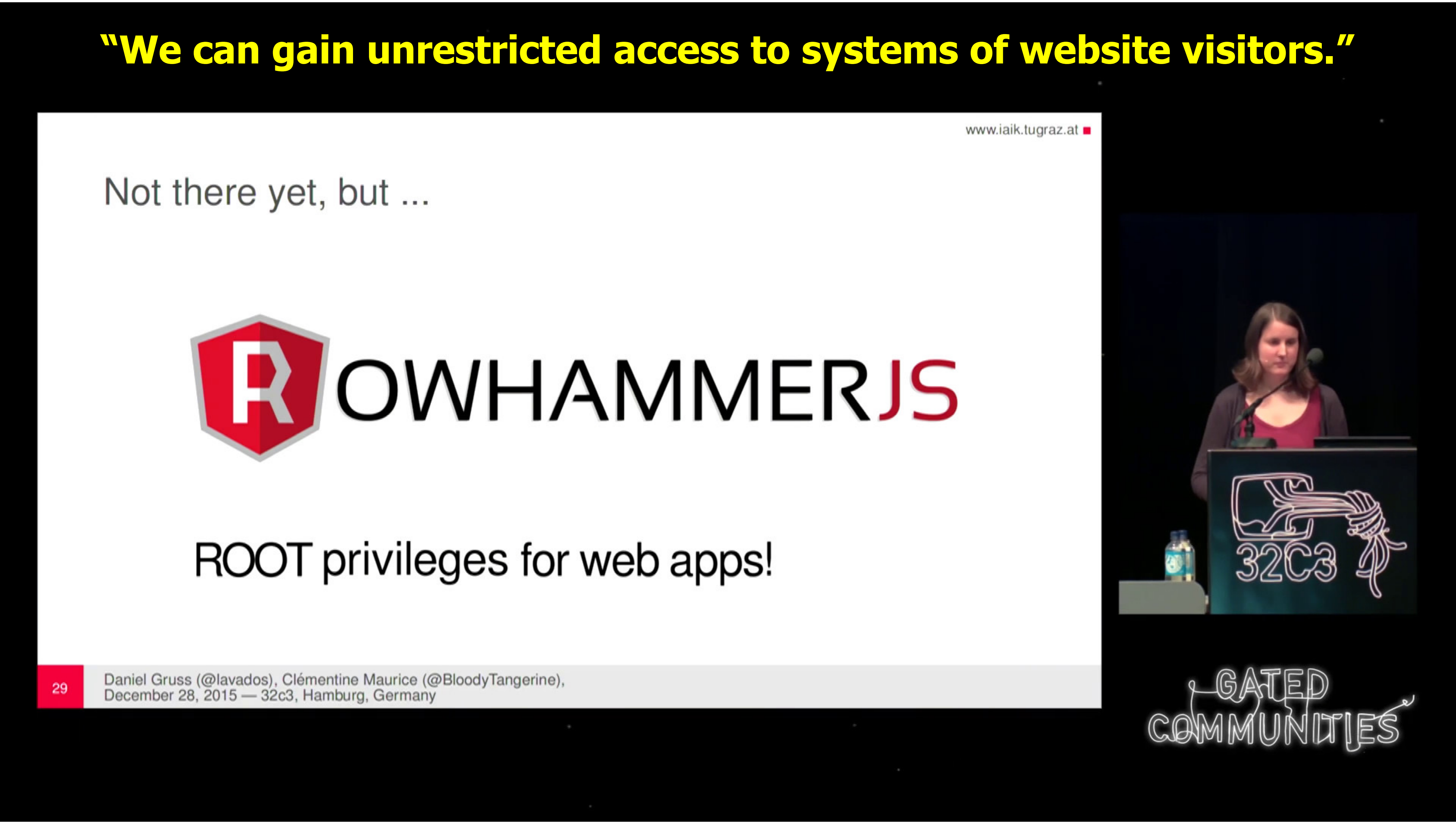
- Ataberk Olgun, Juan Gomez Luna, Konstantinos Kanellopoulos, Behzad Salami, Hasan Hassan, Oguz Ergin, and Onur Mutlu,
"PiDRAM: A Holistic End-to-end FPGA-based Framework for Processing-in-DRAM"
ACM Transactions on Architecture and Code Optimization (***TACO***), March 2023.
[arXiv version]
Presented at the 18th HiPEAC Conference, Toulouse, France, January 2023.
[Slides (pptx) (pdf)]
[Longer Lecture Slides (pptx) (pdf)]
[Lecture Video (40 minutes)]
[PiDRAM Source Code]

PiDRAM: A Holistic End-to-end FPGA-based Framework for Processing-in-DRAM

Ataberk Olgun[§] Juan Gómez Luna[§] Konstantinos Kanellopoulos[§] Behzad Salami[§]
Hasan Hassan[§] Oğuz Ergin[†] Onur Mutlu[§]

[§]*ETH Zürich* [†]*TOBB University of Economics and Technology*

More Security Implications (I)



"We can gain unrestricted access to systems of website visitors."

www.iaik.tugraz.at

Not there yet, but ...

ROWHAMMERJS

ROOT privileges for web apps!

29 Daniel Gruss (@lavados), Clémentine Maurice (@BloodyTangerine), December 28, 2015 — 32c3, Hamburg, Germany

Rowhammer.js: A Remote Software-Induced Fault Attack in JavaScript (DIMVA'16)

Source: https://lab.dsst.io/32c3-slides/7197.html

More Security Implications (II)

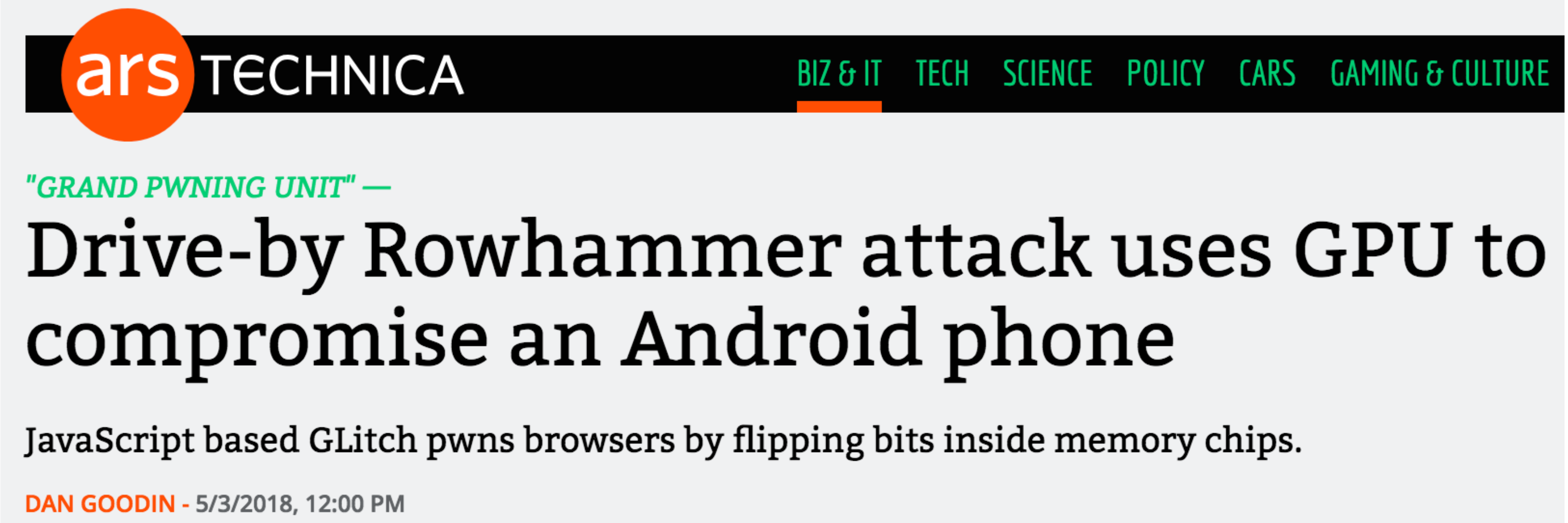
"Can gain control of a smart phone deterministically"

Drammer: Deterministic Rowhammer Attacks on Mobile Platforms, CCS'16

Source: https://fossbytes.com/drammer-rowhammer-attack-android-root-devices/

More Security Implications (III)

- Using an integrated GPU in a mobile system to remotely escalate privilege via the WebGL interface. IEEE S&P 2018

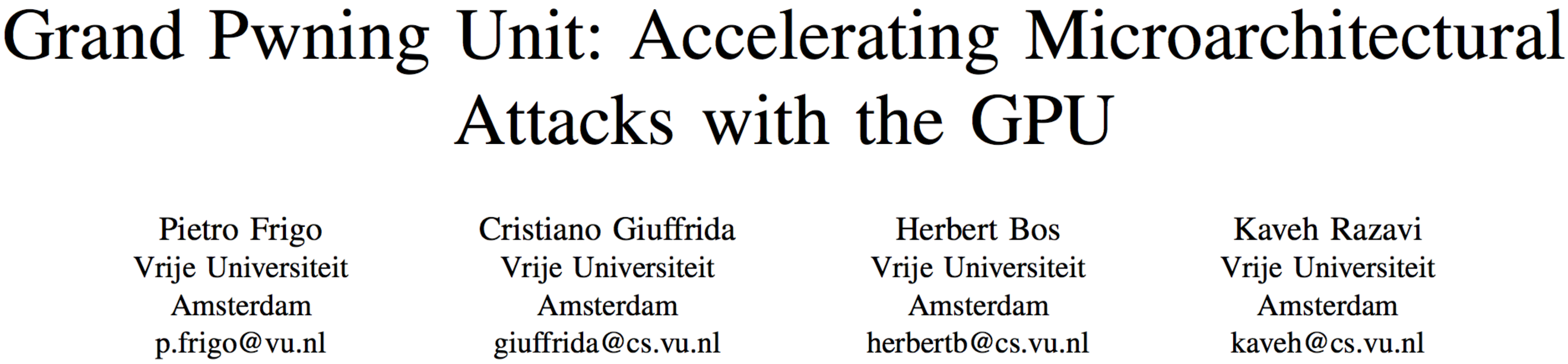


"GRAND PWNING UNIT" —

Drive-by Rowhammer attack uses GPU to compromise an Android phone

JavaScript based GLitch pwns browsers by flipping bits inside memory chips.

DAN GOODIN - 5/3/2018, 12:00 PM



Grand Pwning Unit: Accelerating Microarchitectural Attacks with the GPU

Pietro Frigo
Vrije Universiteit
Amsterdam
p.frigo@vu.nl

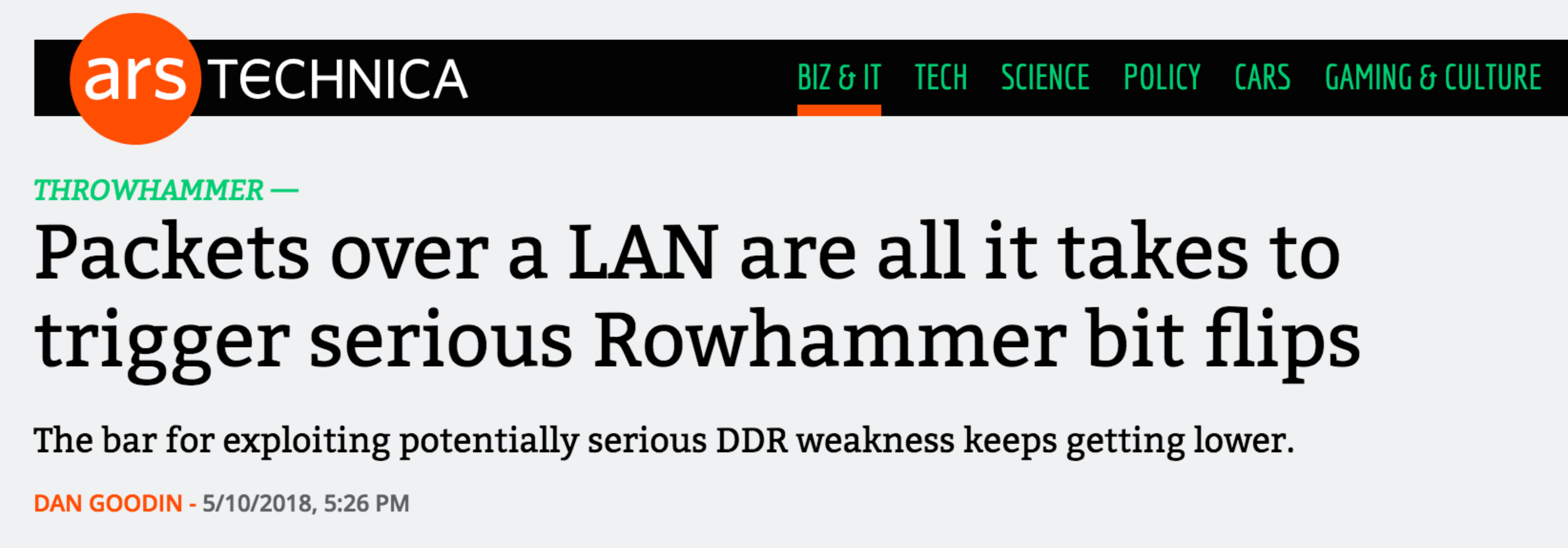
Cristiano Giuffrida
Vrije Universiteit
Amsterdam
giuffrida@cs.vu.nl

Herbert Bos
Vrije Universiteit
Amsterdam
herbertb@cs.vu.nl

Kaveh Razavi
Vrije Universiteit
Amsterdam
kaveh@cs.vu.nl

More Security Implications (IV)

- Rowhammer over RDMA (I) USENIX ATC 2018

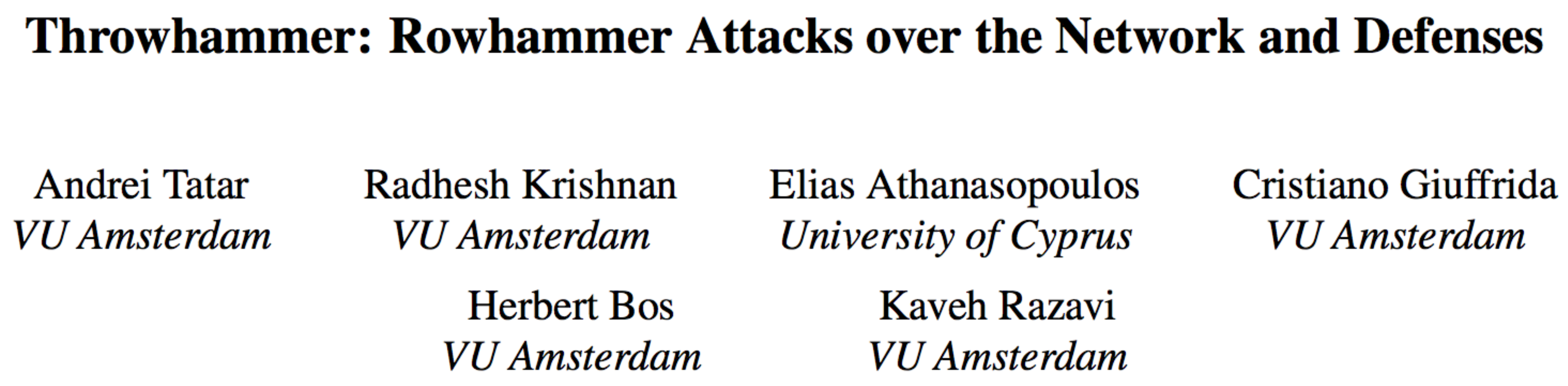


THROWHAMMER —

Packets over a LAN are all it takes to trigger serious Rowhammer bit flips

The bar for exploiting potentially serious DDR weakness keeps getting lower.

DAN GOODIN - 5/10/2018, 5:26 PM



Throwhammer: Rowhammer Attacks over the Network and Defenses

Andrei Tatar
VU Amsterdam

Radhesh Krishnan
VU Amsterdam

Elias Athanasopoulos
University of Cyprus

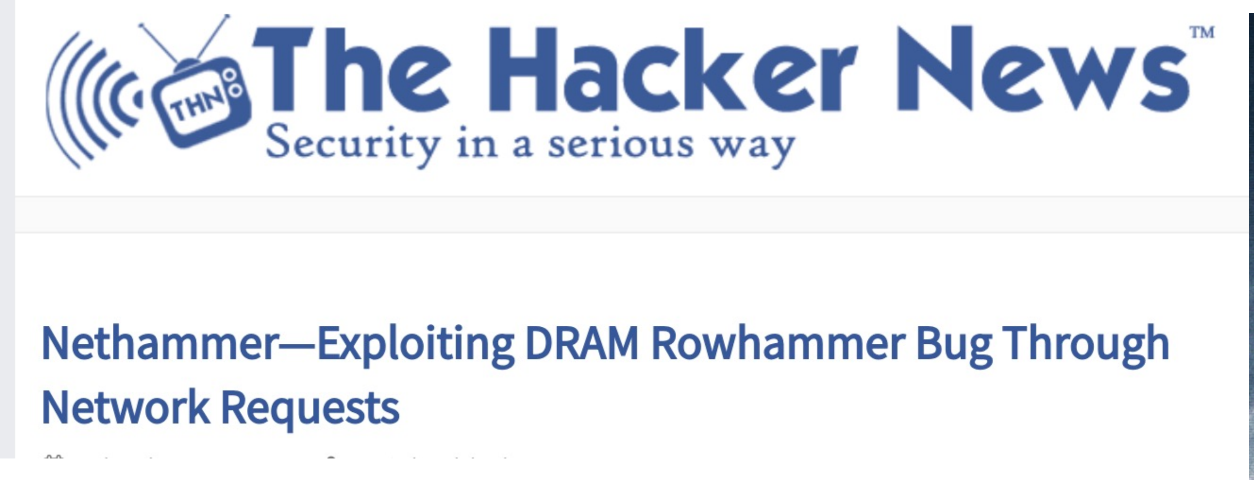
Cristiano Giuffrida
VU Amsterdam

Herbert Bos
VU Amsterdam

Kaveh Razavi
VU Amsterdam

More Security Implications (V)

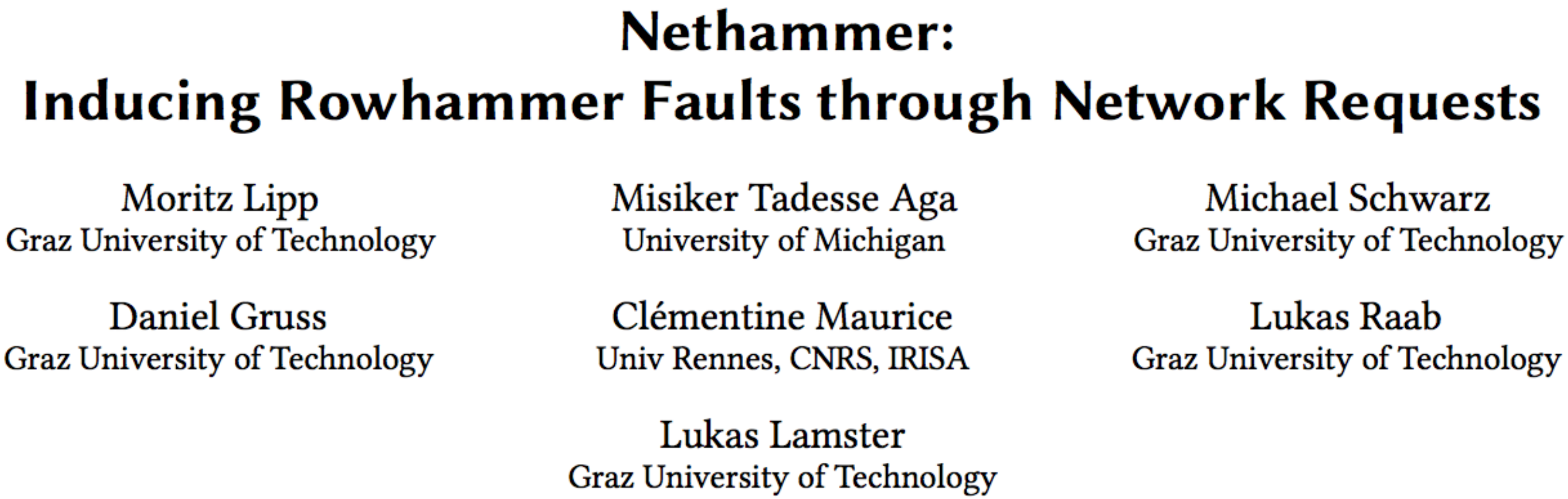
- Rowhammer over RDMA (II)



The Hacker News™

Security in a serious way

Nethammer—Exploiting DRAM Rowhammer Bug Through Network Requests



Nethammer:

Inducing Rowhammer Faults through Network Requests

Moritz Lipp
Graz University of Technology

Misiker Tadesse Aga
University of Michigan

Michael Schwarz
Graz University of Technology

Daniel Gruss
Graz University of Technology

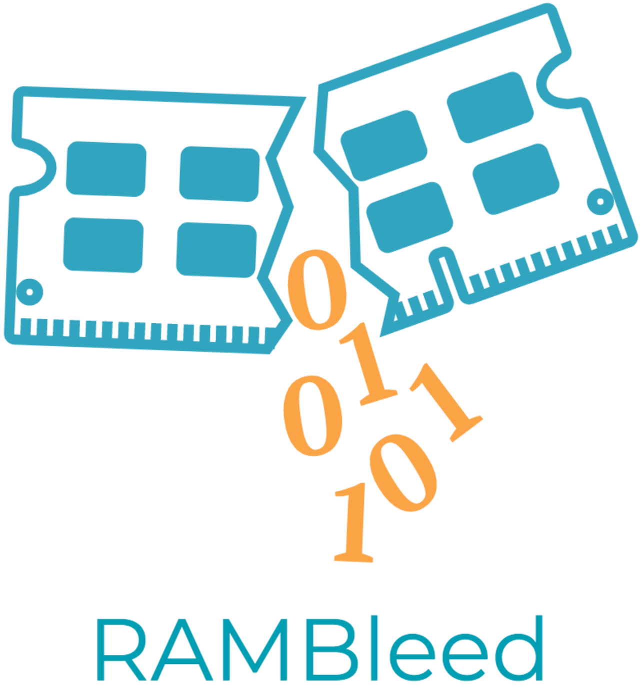
Clémentine Maurice
Univ Rennes, CNRS, IRISA

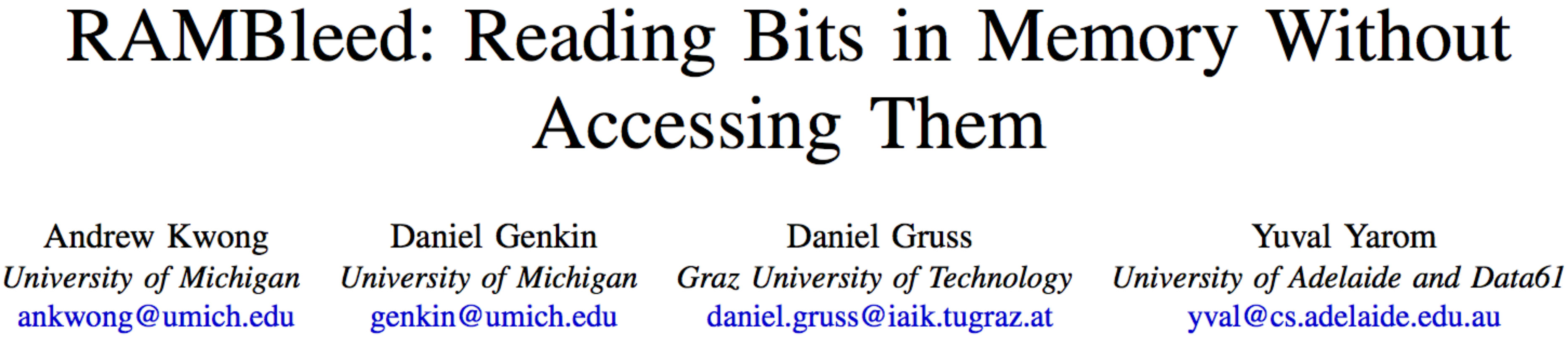
Lukas Raab
Graz University of Technology

Lukas Lamster
Graz University of Technology

More Security Implications (VI)

- IEEE S&P 2020





RAMBleed: Reading Bits in Memory Without Accessing Them

Andrew Kwong
University of Michigan
ankwong@umich.edu

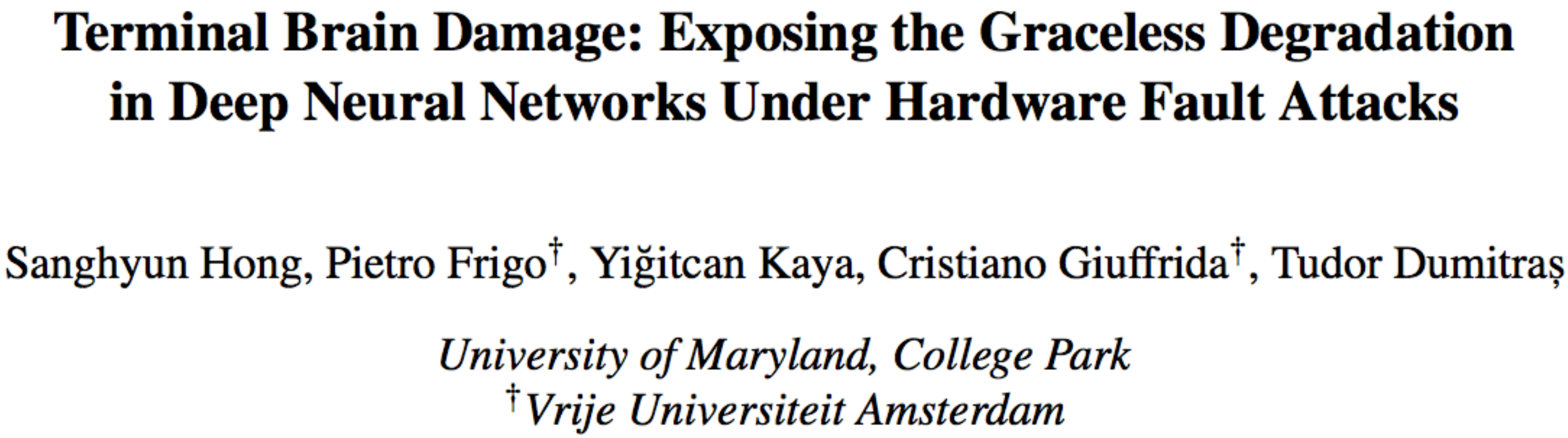
Daniel Genkin
University of Michigan
genkin@umich.edu

Daniel Gruss
Graz University of Technology
daniel.gruss@iaik.tugraz.at

Yuval Yarom
University of Adelaide and Data61
yval@cs.adelaide.edu.au

More Security Implications (VII)

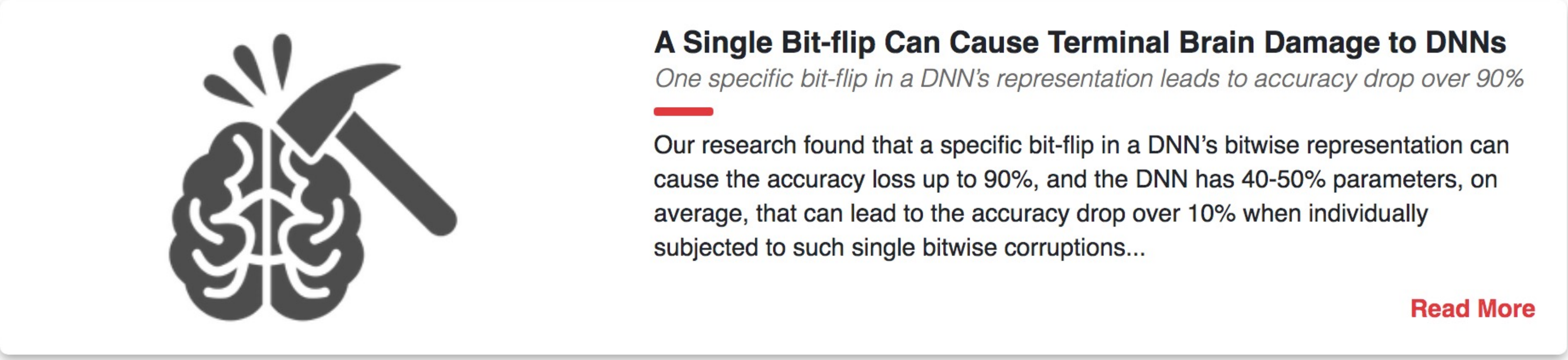
- USENIX Security 2019



Terminal Brain Damage: Exposing the Graceless Degradation in Deep Neural Networks Under Hardware Fault Attacks

Sanghyun Hong, Pietro Frigo[†], Yiğitcan Kaya, Cristiano Giuffrida[†], Tudor Dumitraș

University of Maryland, College Park
[†]Vrije Universiteit Amsterdam



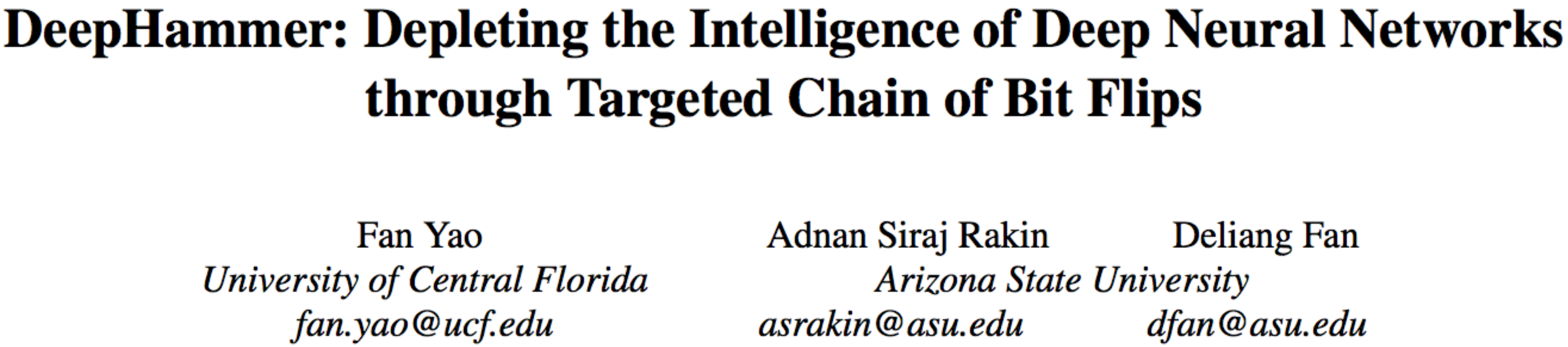
A Single Bit-flip Can Cause Terminal Brain Damage to DNNs

One specific bit-flip in a DNN's representation leads to accuracy drop over 90%

Our research found that a specific bit-flip in a DNN's bitwise representation can cause the accuracy loss up to 90%, and the DNN has 40-50% parameters, on average, that can lead to the accuracy drop over 10% when individually subjected to such single bitwise corruptions...

More Security Implications (VIII)

- USENIX Security 2020



DeepHammer: Depleting the Intelligence of Deep Neural Networks through Targeted Chain of Bit Flips

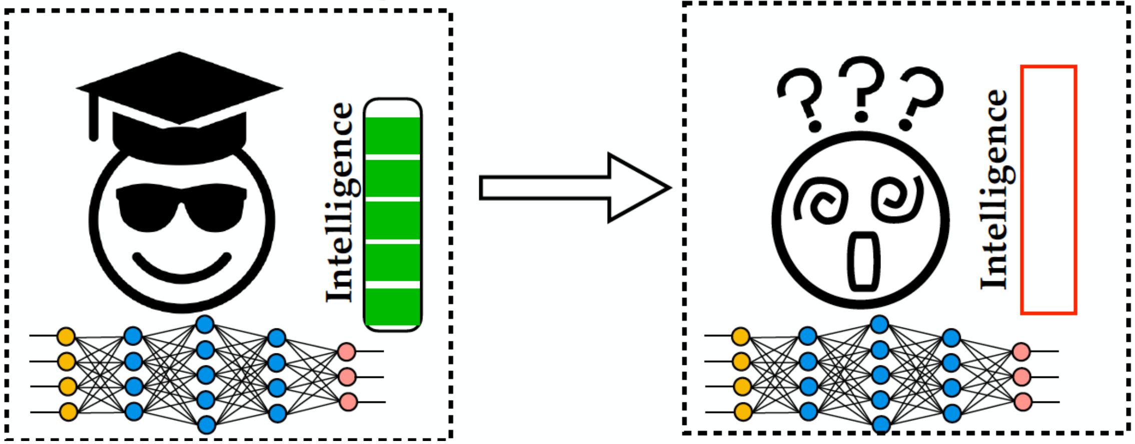
Fan Yao
University of Central Florida
fan.yao@ucf.edu

Adnan Siraj Rakin
Deliang Fan
Arizona State University
asrakin@asu.edu
dfan@asu.edu

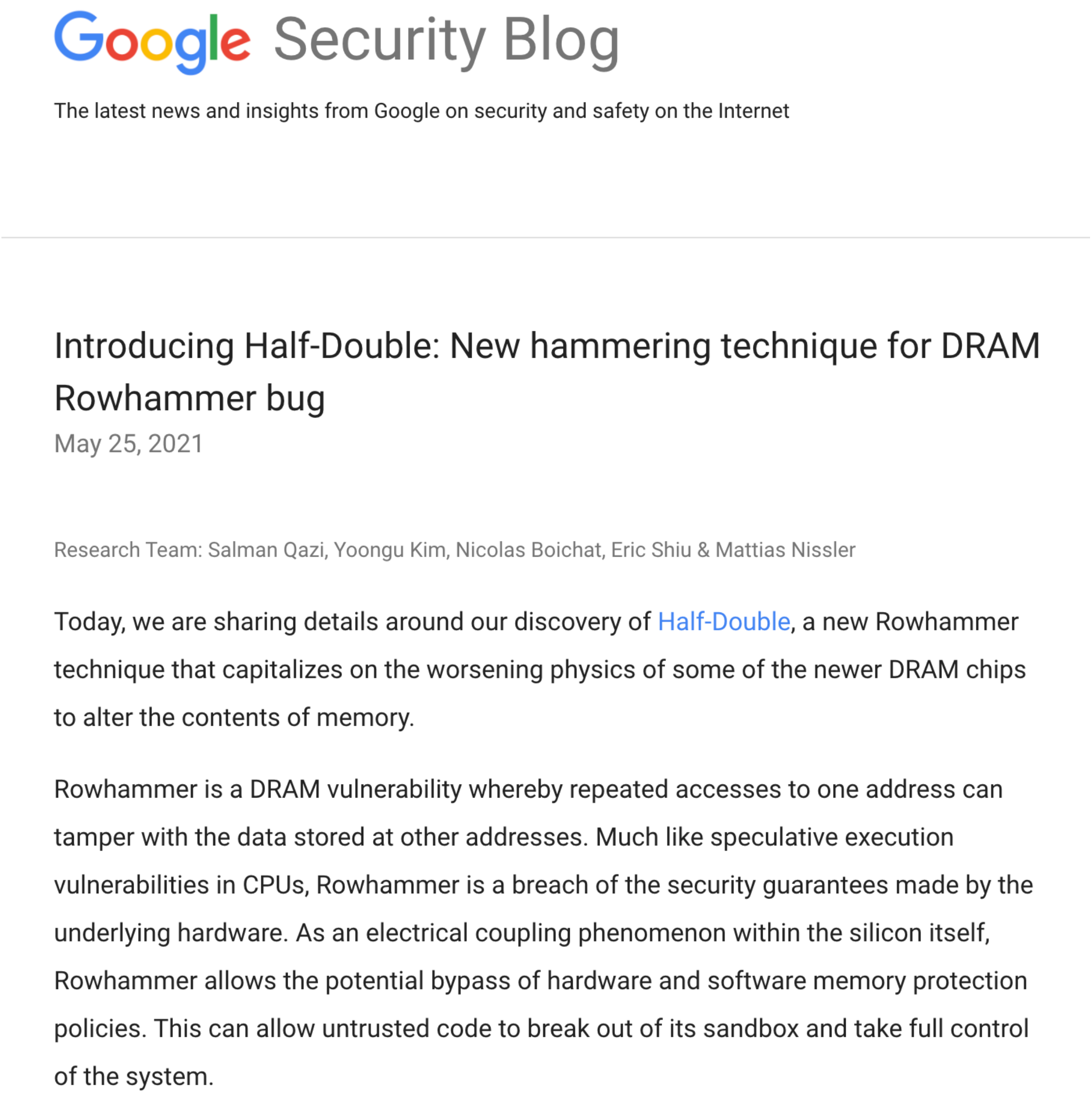
Degrade the **inference accuracy** to the level of **Random Guess**

Example: ResNet-20 for CIFAR-10, **10** output classes

Before attack, **Accuracy: 90.2%** After attack, **Accuracy: ~10% (1/10)**



Google's Half-Double RowHammer Attack (May 2021)



Google Security Blog

The latest news and insights from Google on security and safety on the Internet

Introducing Half-Double: New hammering technique for DRAM Rowhammer bug

May 25, 2021

Research Team: Salman Qazi, Yoongu Kim, Nicolas Boichat, Eric Shiu & Mattias Nissler

Today, we are sharing details around our discovery of Half-Double, a new Rowhammer technique that capitalizes on the worsening physics of some of the newer DRAM chips to alter the contents of memory.

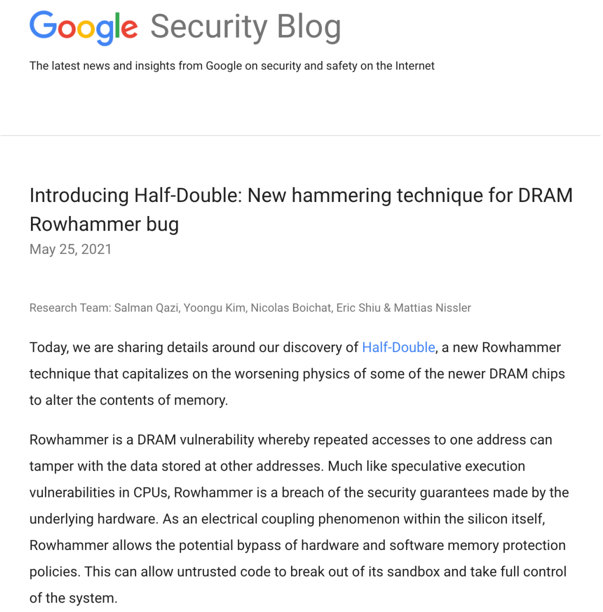
Rowhammer is a DRAM vulnerability whereby repeated accesses to one address can tamper with the data stored at other addresses. Much like speculative execution vulnerabilities in CPUs, Rowhammer is a breach of the security guarantees made by the underlying hardware. As an electrical coupling phenomenon within the silicon itself, Rowhammer allows the potential bypass of hardware and software memory protection policies. This can allow untrusted code to break out of its sandbox and take full control of the system.

https://security.googleblog.com/2021/05/introducing-half-double-new-hammering.html

More Security Implications (VIII)

- **USENIX Security 2022**

- Google's Half-Double RowHammer Attack



Google Security Blog

The latest news and insights from Google on security and safety on the Internet

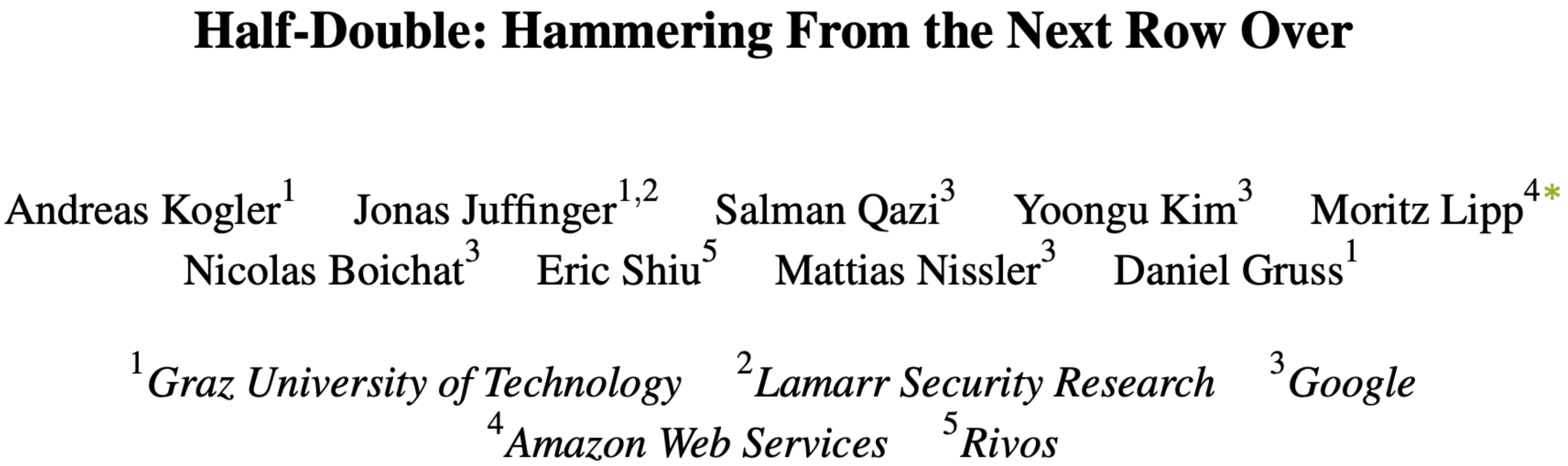
Introducing Half-Double: New hammering technique for DRAM Rowhammer bug

May 25, 2021

Research Team: Salman Qazi, Yoongu Kim, Nicolas Boichat, Eric Shiu & Mattias Nissler

Today, we are sharing details around our discovery of Half-Double, a new Rowhammer technique that capitalizes on the worsening physics of some of the newer DRAM chips to alter the contents of memory.

Rowhammer is a DRAM vulnerability whereby repeated accesses to one address can tamper with the data stored at other addresses. Much like speculative execution vulnerabilities in CPUs, Rowhammer is a breach of the security guarantees made by the underlying hardware. As an electrical coupling phenomenon within the silicon itself, Rowhammer allows the potential bypass of hardware and software memory protection policies. This can allow untrusted code to break out of its sandbox and take full control of the system.



Half-Double: Hammering From the Next Row Over

Andreas Kogler[1] Jonas Juffinger[1,2] Salman Qazi[3] Yoongu Kim[3] Moritz Lipp[4]*
Nicolas Boichat[3] Eric Shiu[5] Mattias Nissler[3] Daniel Gruss[1]

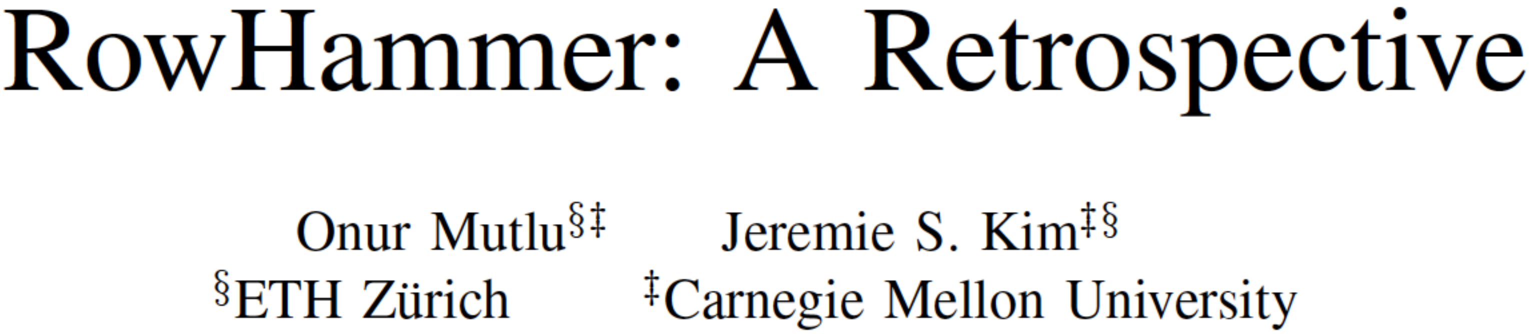
[1] *Graz University of Technology* [2] *Lamarr Security Research* [3] *Google*
[4] *Amazon Web Services* [5] *Rivos*

https://www.usenix.org/system/files/sec22-kogler-half-double.pdf

More Security Implications?

A RowHammer Survey Across the Stack

- Onur Mutlu and Jeremie Kim,
  "RowHammer: A Retrospective"
  *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (**TCAD**) Special Issue on Top Picks in Hardware and Embedded Security*, 2019.
  [Preliminary arXiv version]
  [Slides from COSADE 2019 (pptx)]
  [Slides from VLSI-SOC 2020 (pptx) (pdf)]
  [Talk Video (1 hr 15 minutes, with Q&A)]

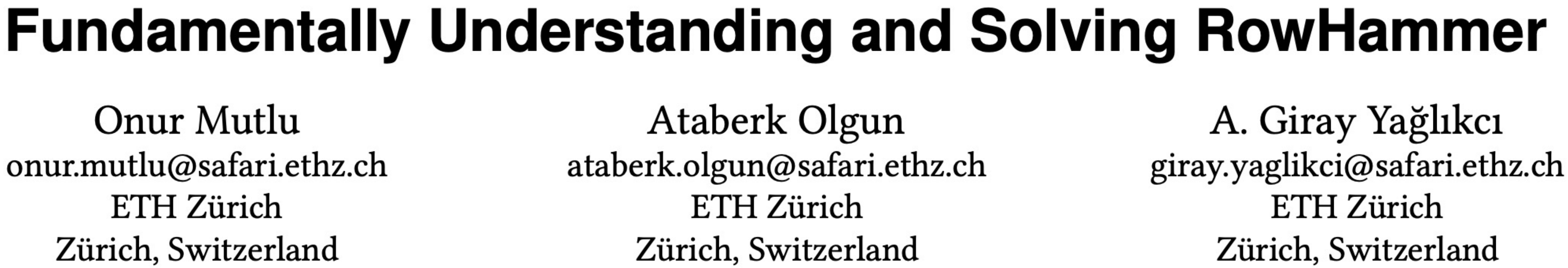


RowHammer: A Retrospective

Onur Mutlu[§‡] Jeremie S. Kim[‡§]
[§]ETH Zürich [‡]Carnegie Mellon University

A RowHammer Survey: Recent Update

- Onur Mutlu, Ataberk Olgun, and A. Giray Yaglikci, **"Fundamentally Understanding and Solving RowHammer"** *Invited Special Session Paper at the 28th Asia and South Pacific Design Automation Conference (**ASP-DAC**)*, Tokyo, Japan, January 2023.
  [arXiv version]
  [Slides (pptx) (pdf)]
  [Talk Video (26 minutes)]



Fundamentally Understanding and Solving RowHammer

Onur Mutlu
onur.mutlu@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

Ataberk Olgun
ataberk.olgun@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

A. Giray Yağlıkcı
giray.yaglikci@safari.ethz.ch
ETH Zürich
Zürich, Switzerland

https://arxiv.org/pdf/2211.07613.pdf

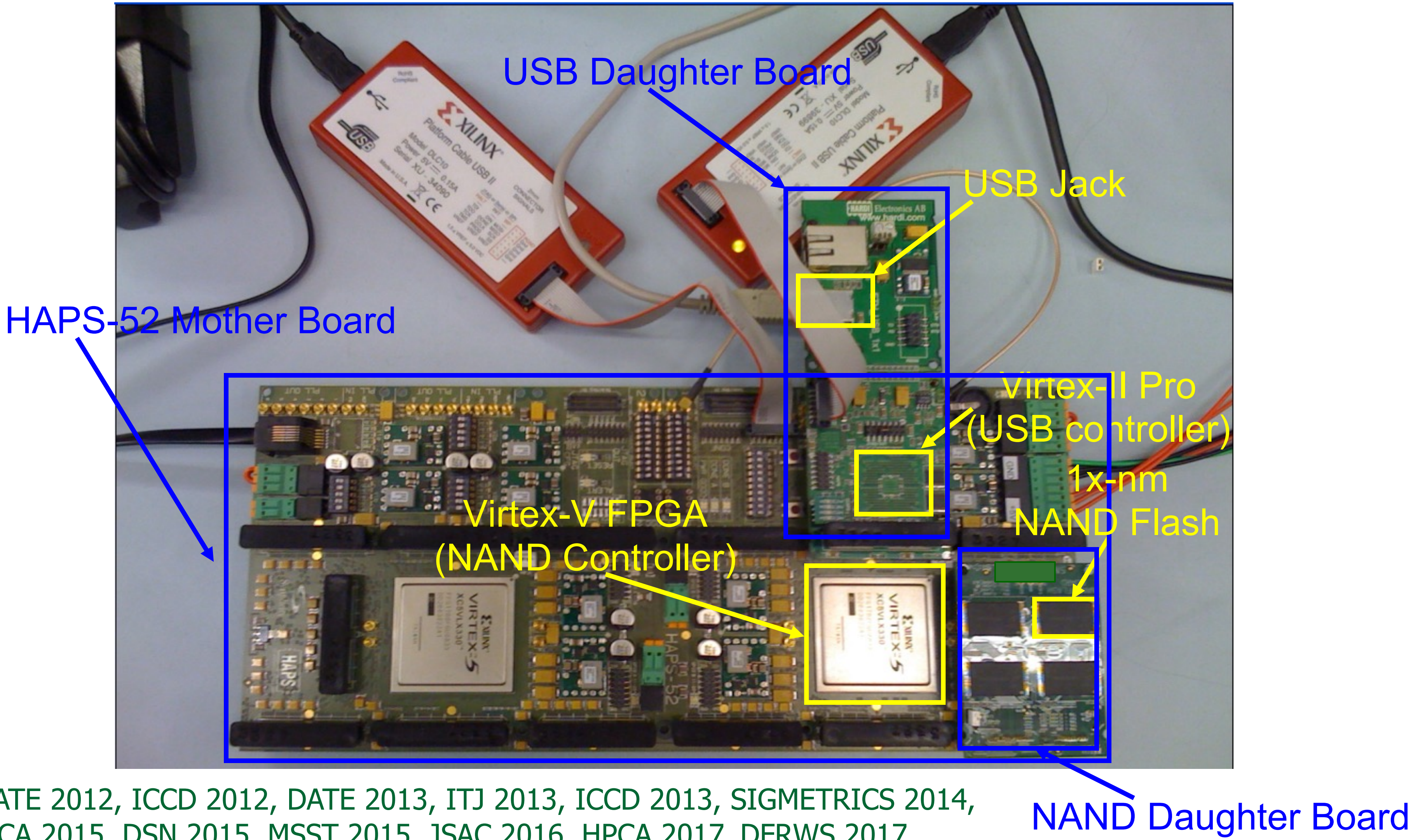
A Takeaway

Main Memory Needs

Intelligent Controllers

for Security, Safety, Reliability, Scaling

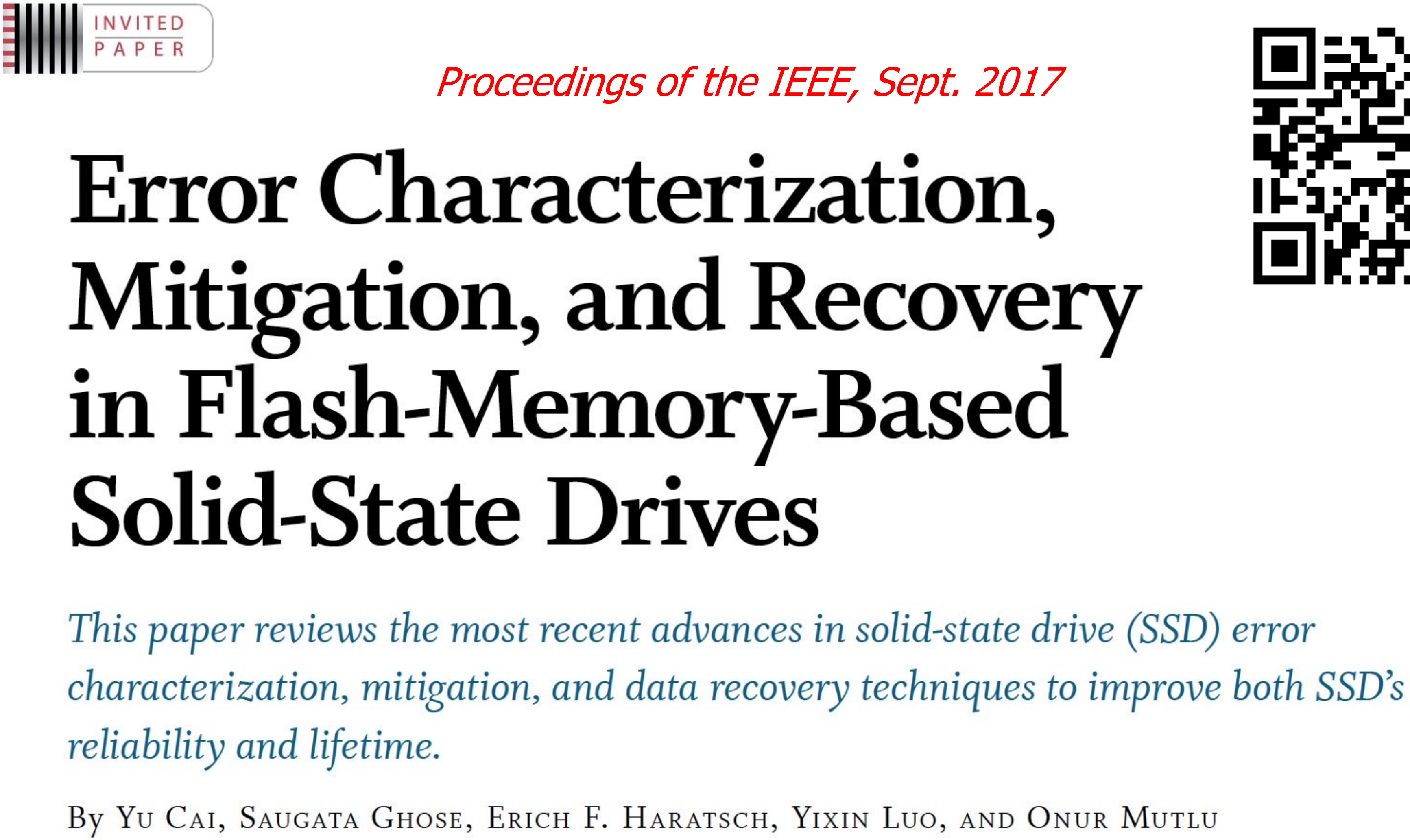
Aside: Intelligent Controller for NAND Flash



[DATE 2012, ICCD 2012, DATE 2013, ITJ 2013, ICCD 2013, SIGMETRICS 2014, HPCA 2015, DSN 2015, MSST 2015, JSAC 2016, HPCA 2017, DFRWS 2017, PIEEE 2017, HPCA 2018, SIGMETRICS 2018]

Cai+, "Error Characterization, Mitigation, and Recovery in Flash Memory Based Solid State Drives," Proc. IEEE 2017.

Intelligent Flash Controllers **[PIEEE'17]**



INVITED PAPER

Proceedings of the IEEE, Sept. 2017

Error Characterization, Mitigation, and Recovery in Flash-Memory-Based Solid-State Drives

This paper reviews the most recent advances in solid-state drive (SSD) error characterization, mitigation, and data recovery techniques to improve both SSD's reliability and lifetime.

By Yu Cai, Saugata Ghose, Erich F. Haratsch, Yixin Luo, and Onur Mutlu

https://arxiv.org/pdf/1706.08642

Two Major RowHammer Directions

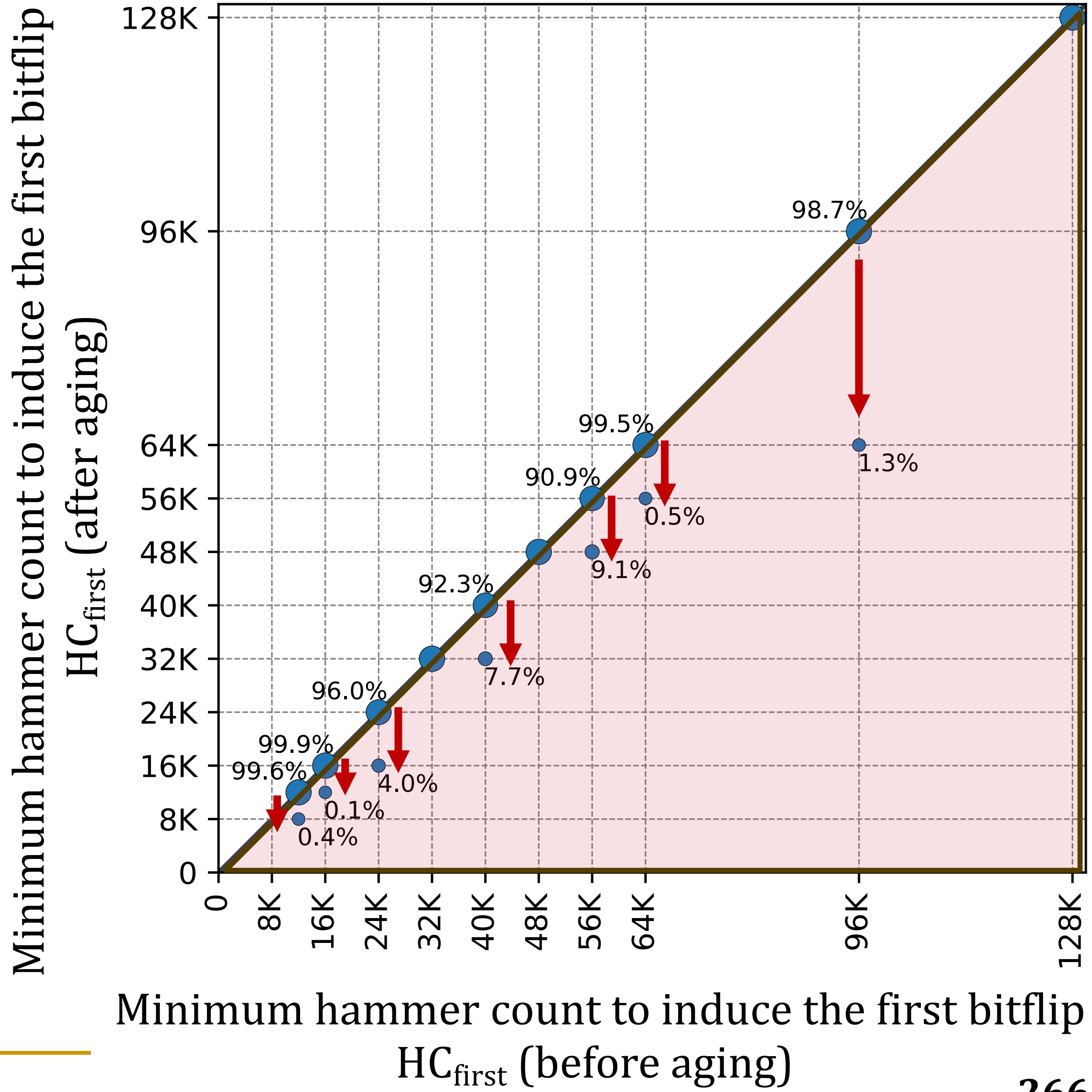
- **Understanding RowHammer**
  - Many effects still need to be rigorously examined
    - Aging of DRAM Chips
    - Environmental Conditions (e.g., Process, Voltage, Temperature)
    - Memory Access Patterns
    - Memory Controller & System Design Decisions
    - …

- **Solving RowHammer**
  - Flexible and efficient solutions are necessary
    - In-field patchable / reconfigurable / programmable solutions
  - Co-architecting System and Memory is important
    - To avoid performance and denial-of-service problems

RowHammer Becomes Worse with Aging

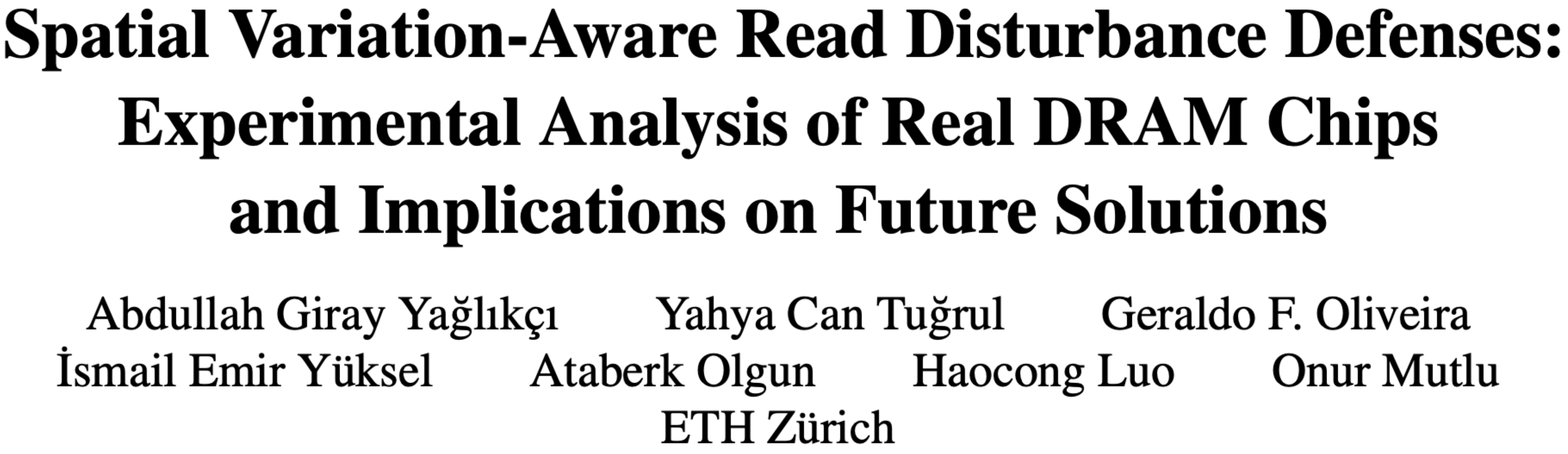
Preliminary data on aging via 68-day of continuous hammering

Aging can lead to read disturbance bitflips at **smaller** hammer counts



RowHammer Spatial Variation Analysis (2024)

- **Appears at HPCA 2024**



Spatial Variation-Aware Read Disturbance Defenses: Experimental Analysis of Real DRAM Chips and Implications on Future Solutions

Abdullah Giray Yağlıkçı Yahya Can Tuğrul Geraldo F. Oliveira
İsmail Emir Yüksel Ataberk Olgun Haocong Luo Onur Mutlu
ETH Zürich

https://arxiv.org/pdf/2402.18652

Two Major Directions

- **Understanding Bitflips (Hardware errors in general)**
  - Many effects on bitflips still need to be rigorously examined
    - Aging of DRAM Chips
    - Environmental Conditions (e.g., Process, Voltage, Temperature)
    - Memory Access Patterns
    - Memory Controller & System Design Decisions
    - …

- **Solving Bitflips (Hardware errors in general)**
  - Flexible and efficient solutions are necessary
    - In-field patchable / reconfigurable / programmable solutions
  - Co-architecting across the system stack/components is important
    - To avoid performance and denial-of-service problems