Digital Design & Computer Arch.

Lecture 22: Memory Overview, Organization & Technology

Prof. Onur Mutlu

ETH Zürich
Spring 2022
19 May 2022

Extra Assignment 3: Amdahl's Law

- **Paper review**
  - G. M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," AFIPS 1967.

- **Optional Assignment – for 1% extra credit**
  - **Write a 1-page review**
  - Upload PDF file to Moodle – Deadline: June 15

- Strongly recommended that you follow my guidelines for (paper) review

Readings for This Lecture and Next

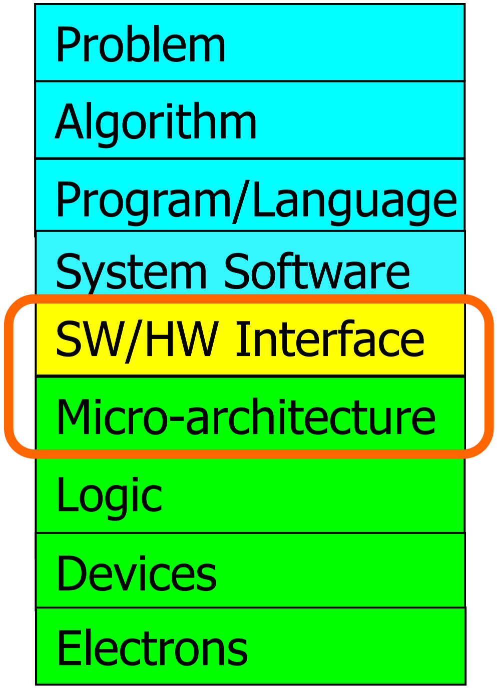
- Memory Hierarchy and Caches

- Required
  - H&H Chapters 8.1-8.3
  - Refresh: P&P Chapter 3.5
  - Kim & Mutlu, "Memory Systems," Computing Handbook, 2014.
    - https://people.inf.ethz.ch/omutlu/pub/memory-systems-introduction_computing-handbook14.pdf

- Recommended
  - An early cache paper by Maurice Wilkes
    - Wilkes, "Slave Memories and Dynamic Storage Allocation," IEEE Trans. On Electronic Computers, 1965.

We Are **Done** With This…

- Dataflow (at the ISA level)
- Superscalar Execution
- VLIW
- Systolic Arrays
- Decoupled Access Execute
- SIMD Processing (Vector and Array processors)
- Graphics Processing Units (GPUs)



Problem
Algorithm
Program/Language
System Software
SW/HW Interface
Micro-architecture
Logic
Devices
Electrons

Approaches to (Instruction-Level) Concurrency

- Pipelining
- Fine-Grained Multithreading
- Out-of-order Execution
- Dataflow (at the ISA level)
- Superscalar Execution
- VLIW
- Systolic Arrays
- Decoupled Access Execute
- SIMD Processing (Vector and Array processors, GPUs)

Now you are very familiar with many processing paradigms

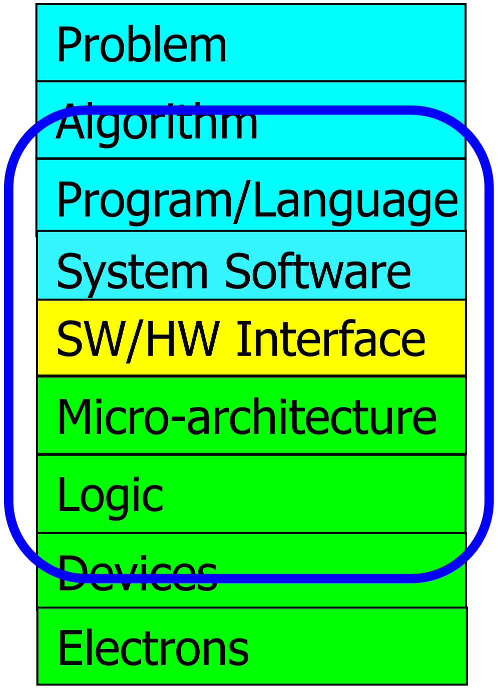
Approaches to (Instruction-Level) Concurrency

- Pipelining
- Fine-Grained Multithreading
- Out-of-order Execution
- Dataflow (at the ISA level)
- Superscalar Execution
- VLIW
- Systolic Arrays
- Decoupled Access Execute
- SIMD Processing (Vector and Array processors, GPUs)

Food for thought:

tradeoffs of these different processing paradigms

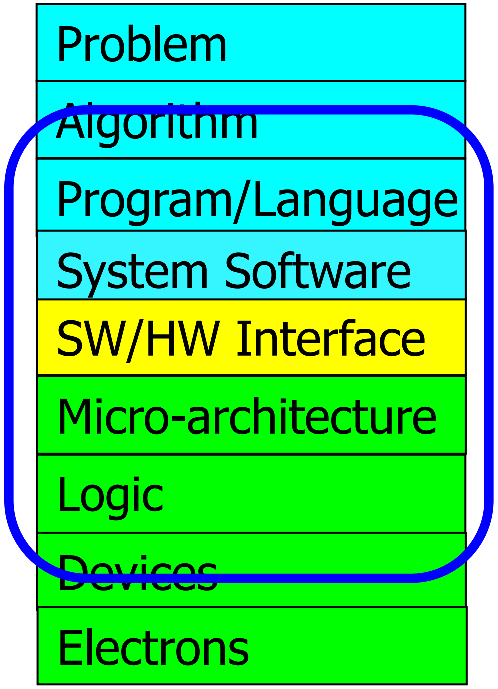
Tradeoffs of Processing Paradigms



Food for thought:

tradeoffs of these different processing paradigms

Tradeoffs of Processing Paradigms



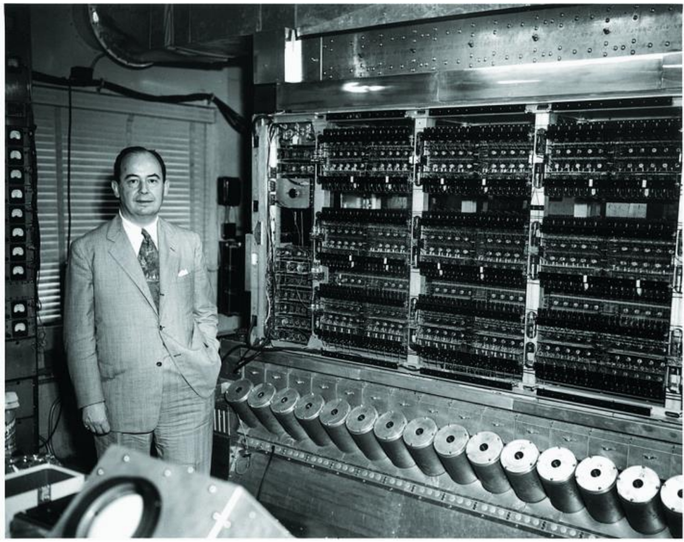
Food for thought:

what determines the widespread success of a paradigm

Let Us Now Take A Step Back

A Computing System

- Three key components
- Computation
- Communication
- Storage/memory



Burks, Goldstein, von Neumann, "Preliminary discussion of the logical design of an electronic computing instrument," 1946.

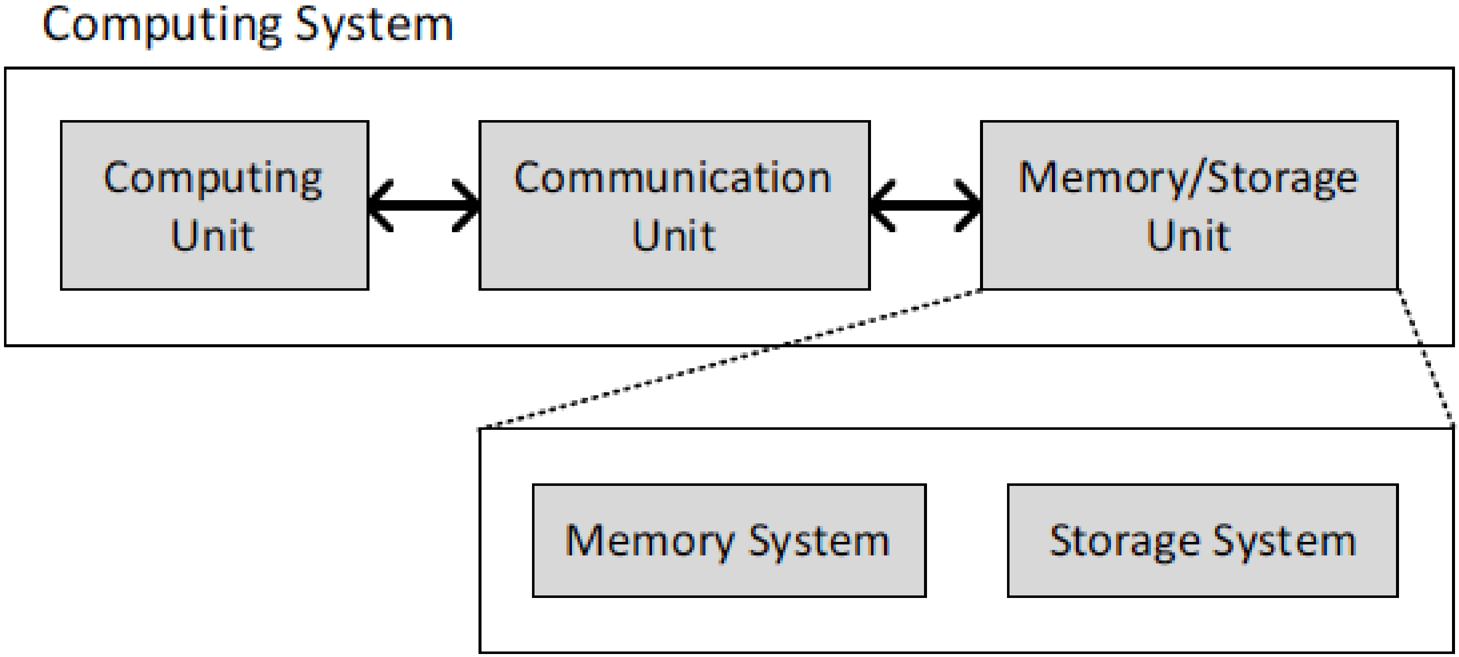
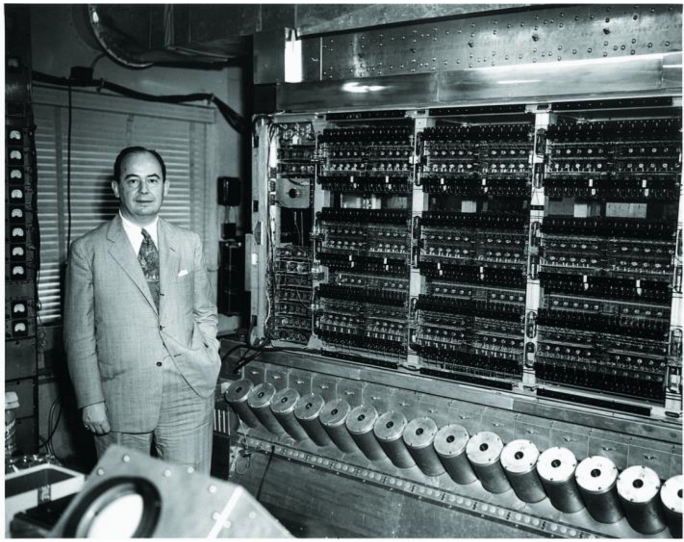


Image source: https://lbsitbytes2010.wordpress.com/2013/03/29/john-von-neumann-roll-no-15/

A Computing System

- Three key components
- Computation
- Communication
- Storage/memory



Burks, Goldstein, von Neumann, "Preliminary discussion of the logical design of an electronic computing instrument," 1946.

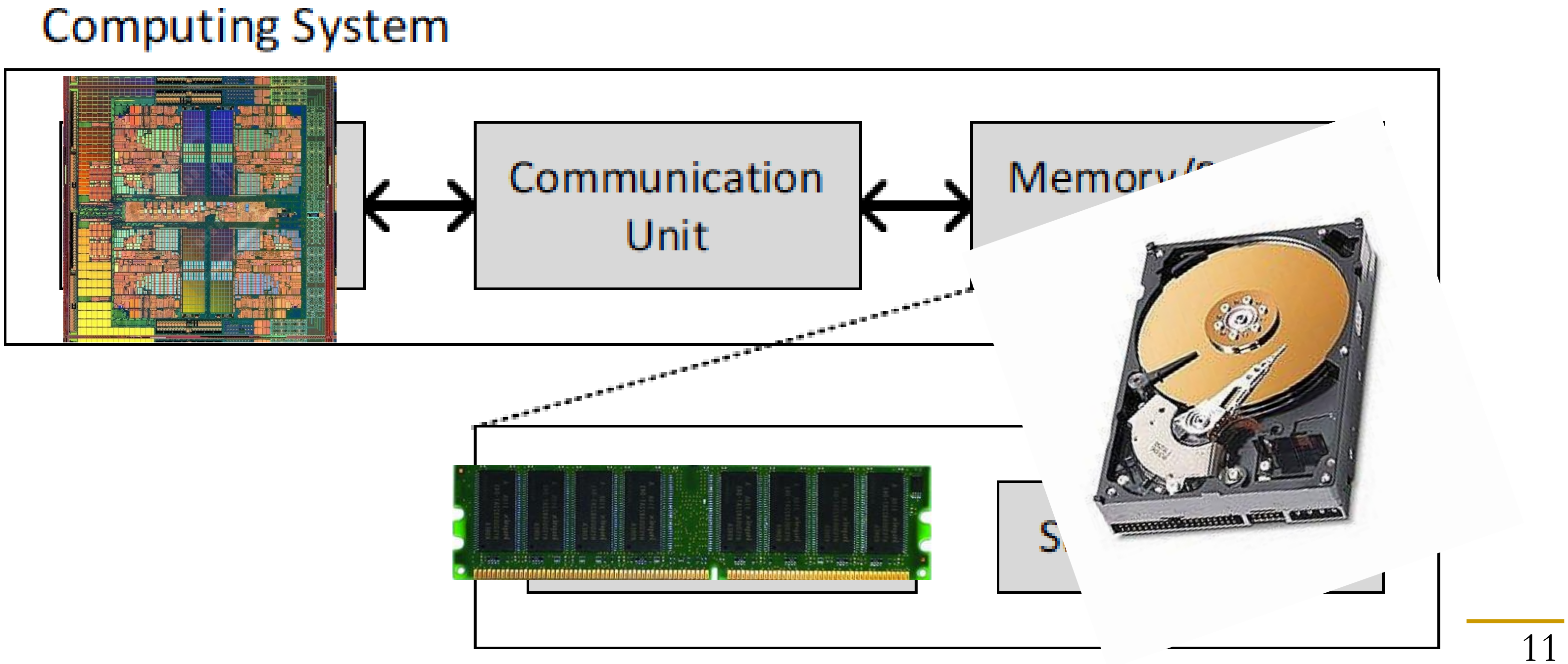
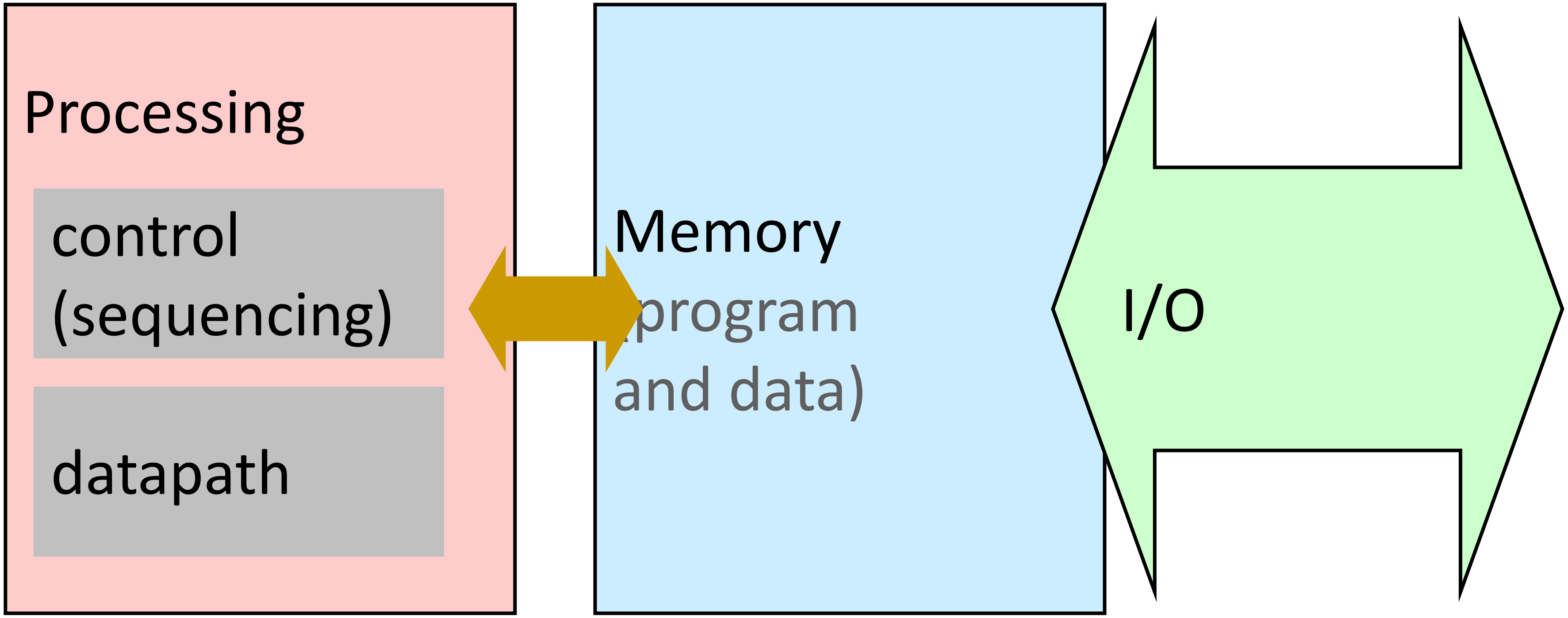


Image source: https://lbsitbytes2010.wordpress.com/2013/03/29/john-von-neumann-roll-no-15/

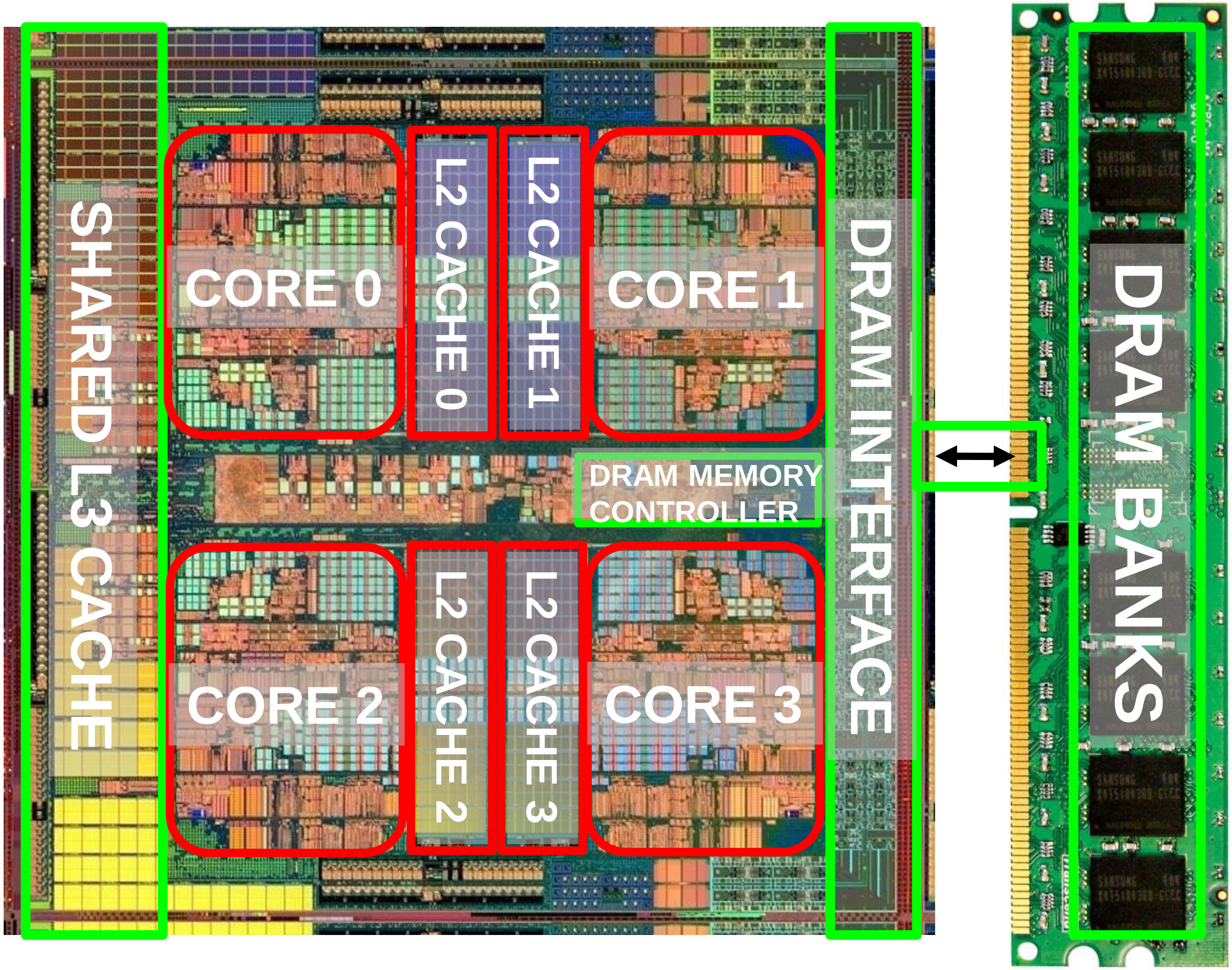
Recall: What is A Computer?

- We will cover all three components



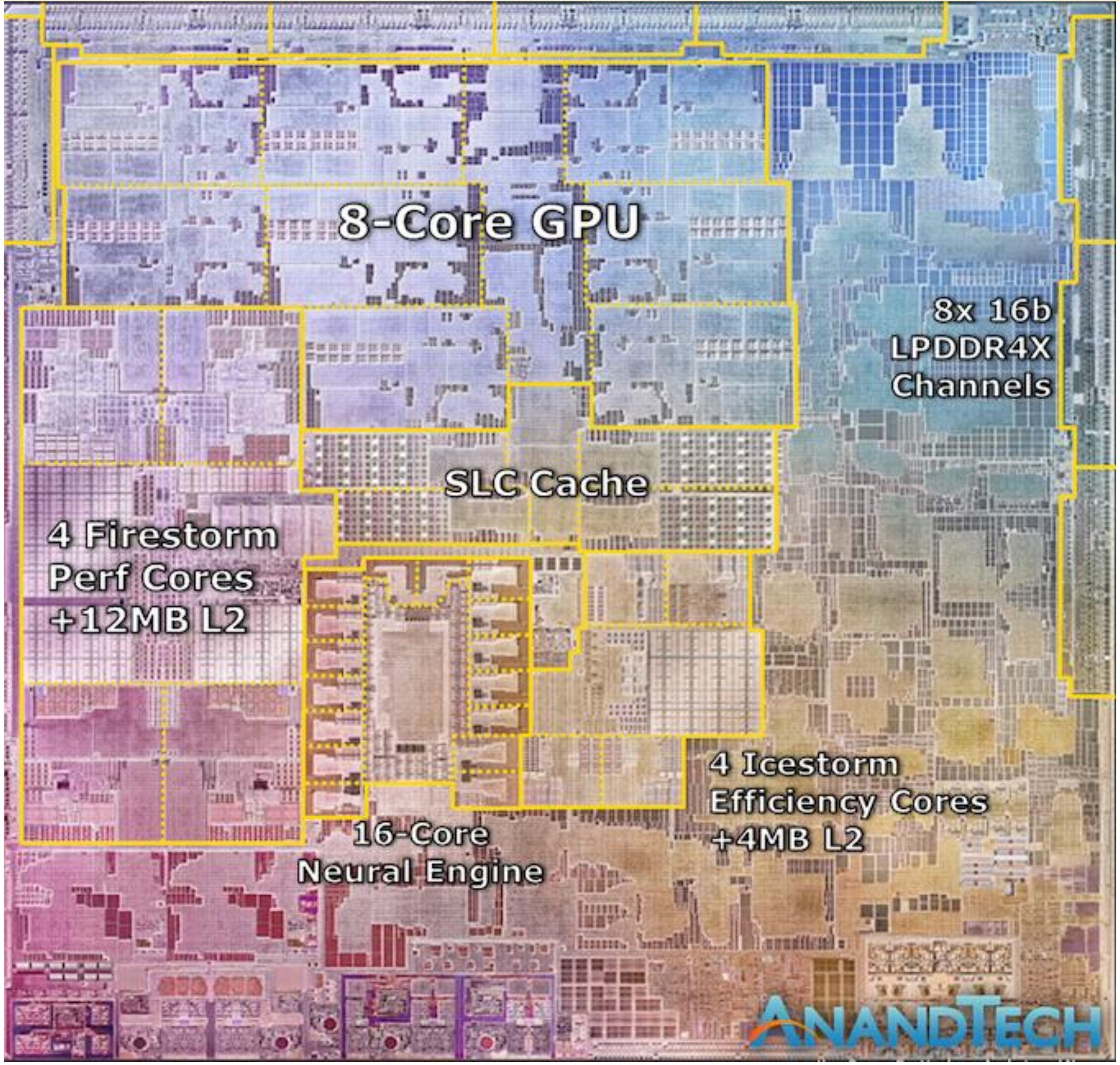
Memory Is Critically Important

Memory in a Modern System



AMD Barcelona, 2006

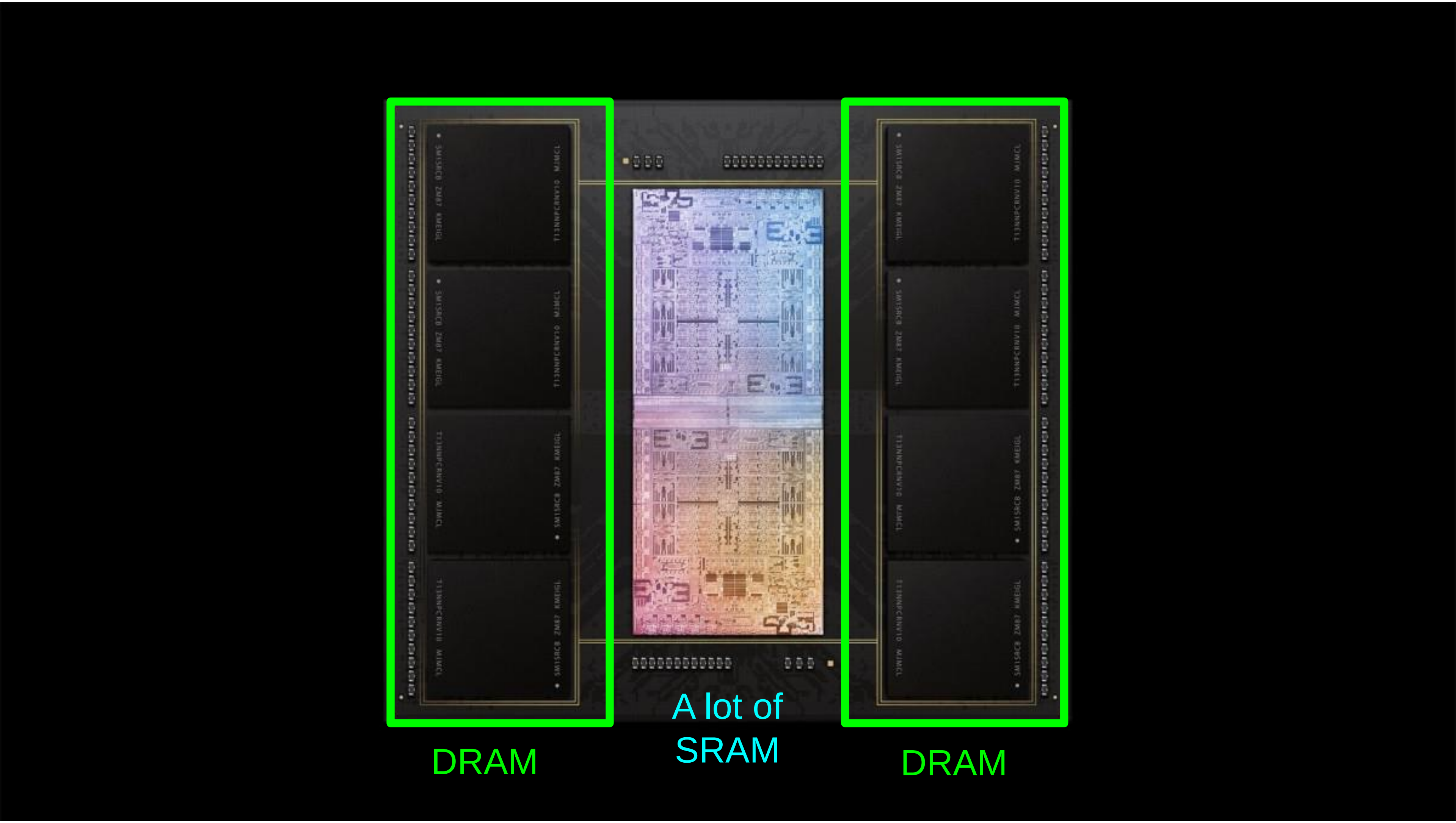
A Large Fraction of Modern Chips is Memory



Apple M1, 2021

Source: https://www.anandtech.com/show/16252/mac-mini-apple-m1-tested

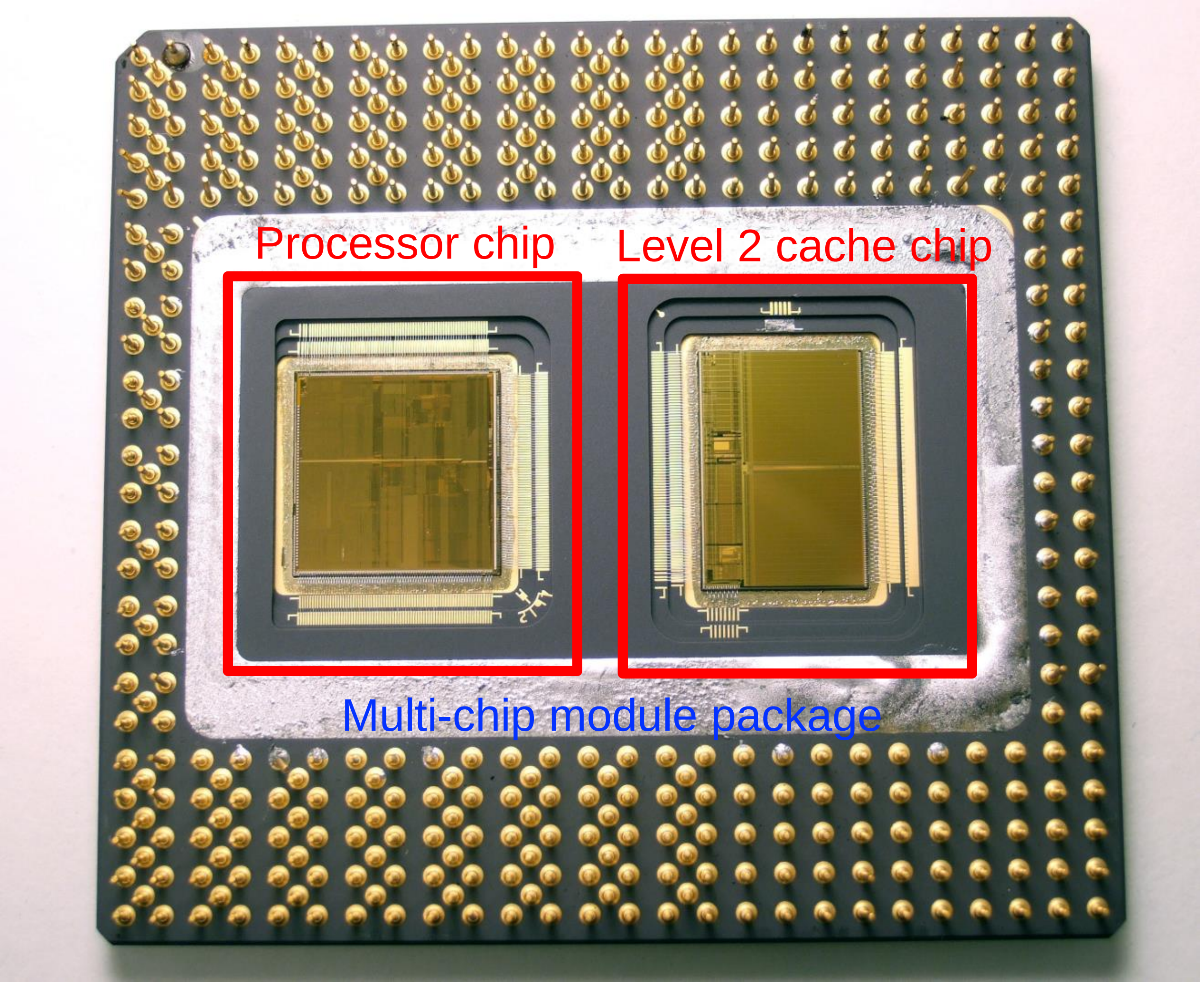
A Large Fraction of Modern Systems is Memory



Apple M1 Ultra System (2022)

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

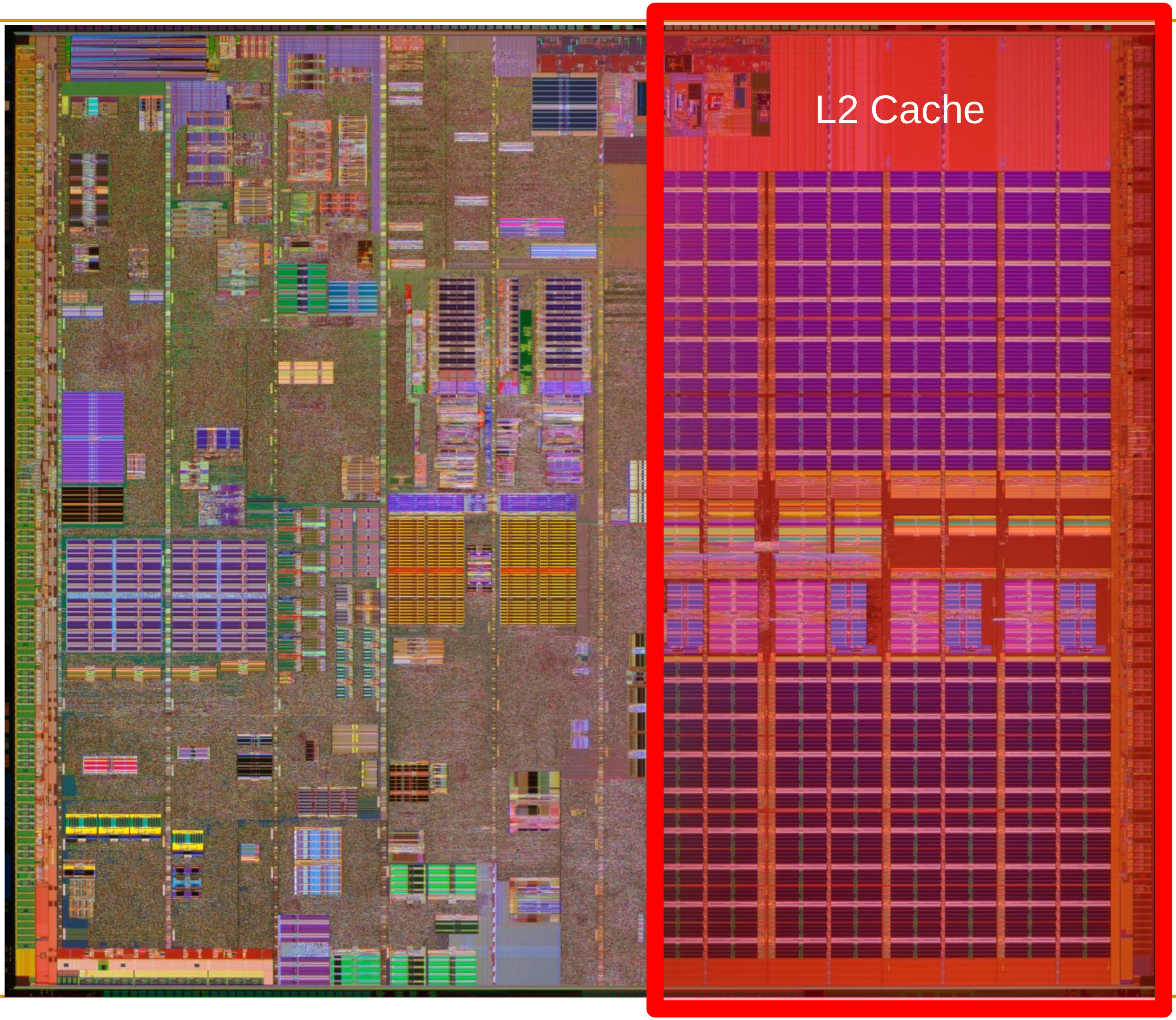
A Large Fraction of Modern Systems is Memory



Intel Pentium Pro, 1995

By Moshen - http://en.wikipedia.org/wiki/Image:Pentiumpro_moshen.jpg, CC BY-SA 2.5, https://commons.wikimedia.org/w/index.php?curid=2262471

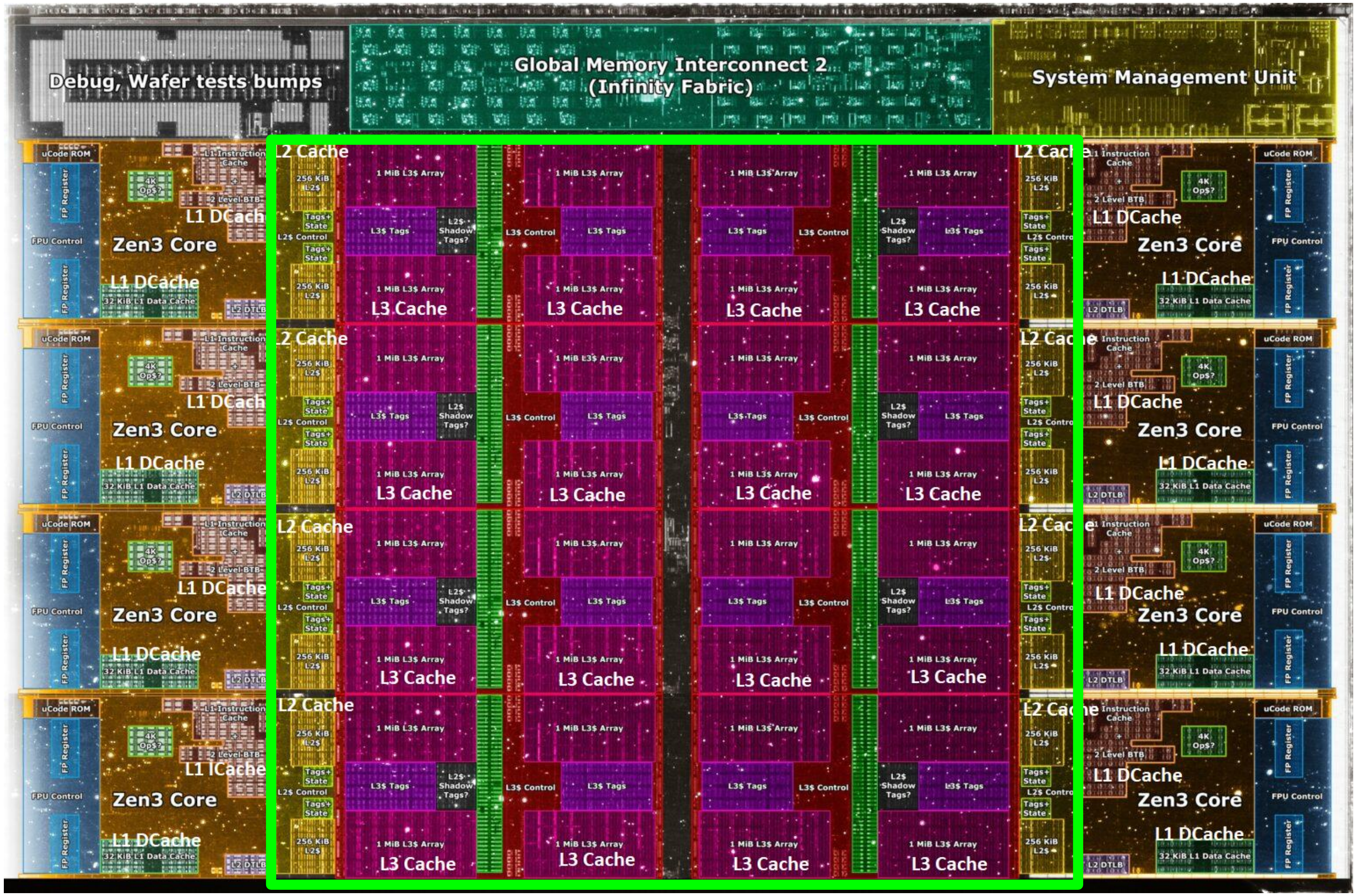
A Large Fraction of Modern Systems is Memory



https://download.intel.com/newsroom/kits/40thanniversary/gallery/images/Pentium_4_6xx-die.jpg

Intel Pentium 4, 2000

A Large Fraction of Modern Systems is Memory



AMD Ryzen 5000, 2020

Core Count:
8 cores/16 threads

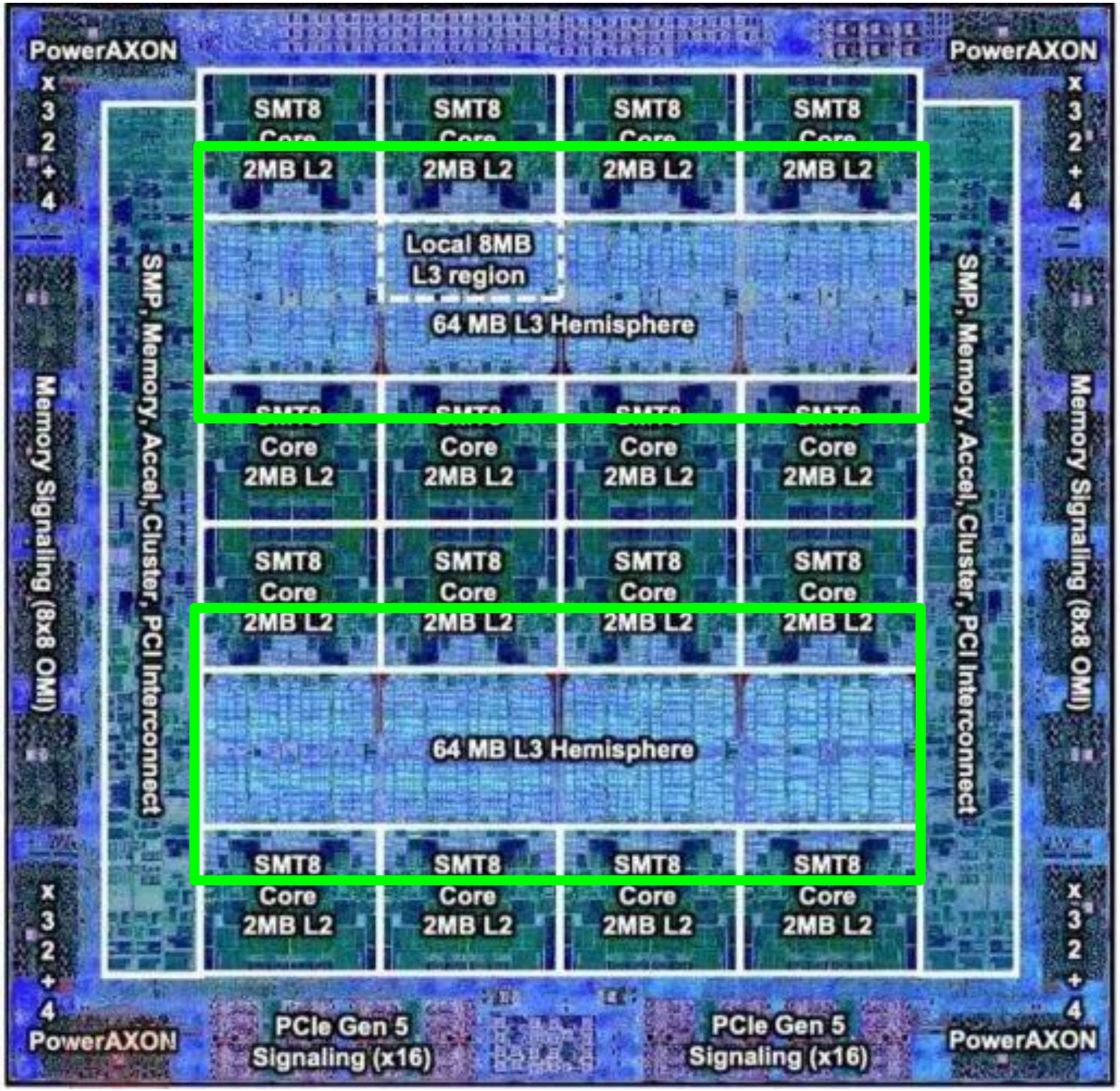
L1 Caches:
32 KB per core

L2 Caches:
512 KB per core

L3 Cache:
32 MB shared

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

A Large Fraction of Modern Systems is Memory



IBM POWER10, 2020

Cores:

15-16 cores,
8 threads/core

L2 Caches:

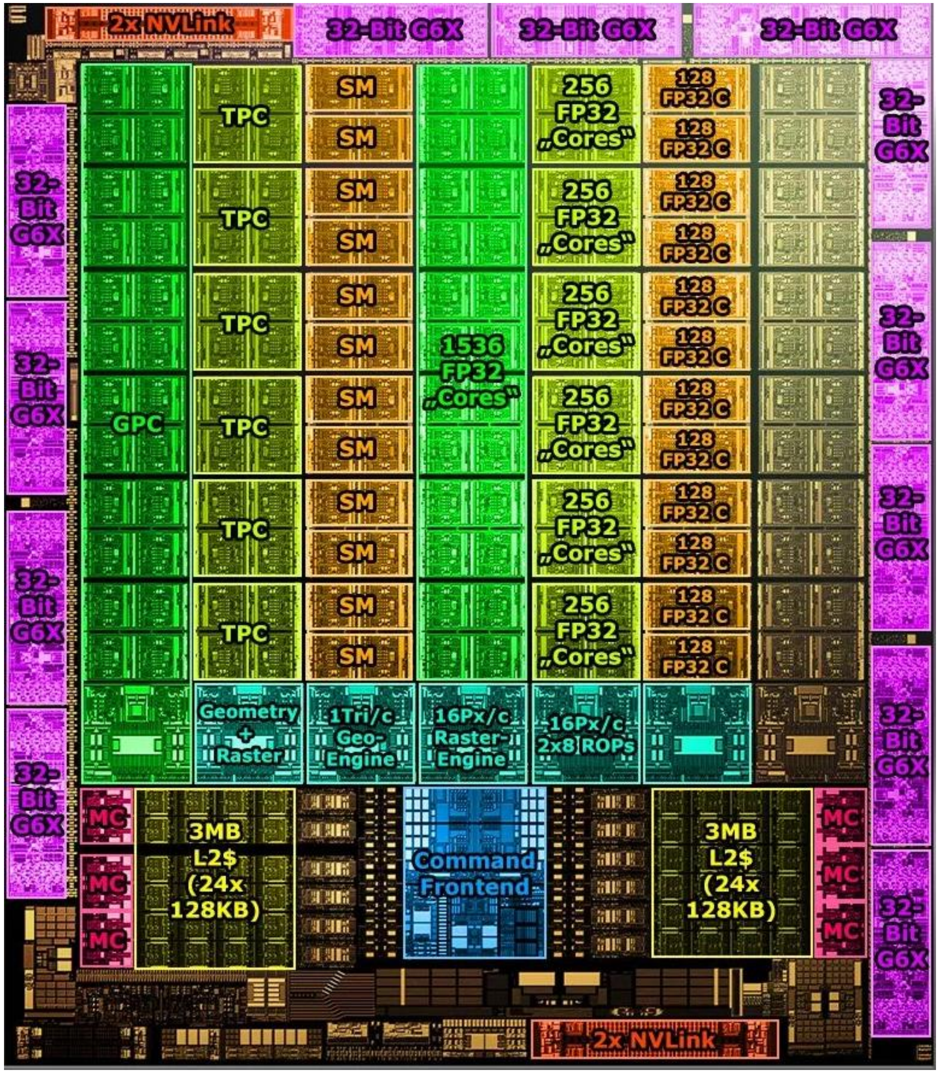
2 MB per core

L3 Cache:

120 MB shared

https://www.it-techblog.de/ibm-power10-prozessor-mehr-speicher-mehr-tempo-mehr-sicherheit/09/2020/

A Large Fraction of Modern Systems is Memory



Nvidia Ampere, 2020

Cores:

128 Streaming Multiprocessors

L1 Cache or Scratchpad:

192KB per SM

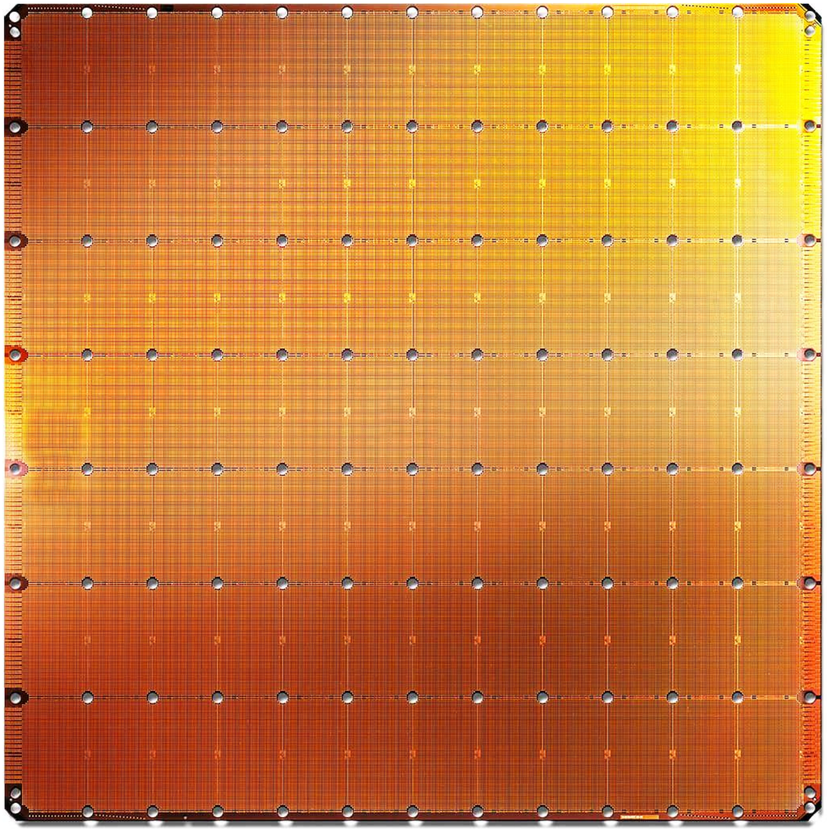
Can be used as L1 Cache and/or Scratchpad

L2 Cache:

40 MB shared

https://www.tomshardware.com/news/infrared-photographer-photos-nvidia-ga102-ampere-silicon

Cerebras's Wafer Scale Engine (2019)



- The largest ML accelerator chip
- 400,000 cores
- **18 GB of on-chip memory**
- **9 PB/s memory bandwidth**

Cerebras WSE
1.2 Trillion transistors
46,225 mm^2

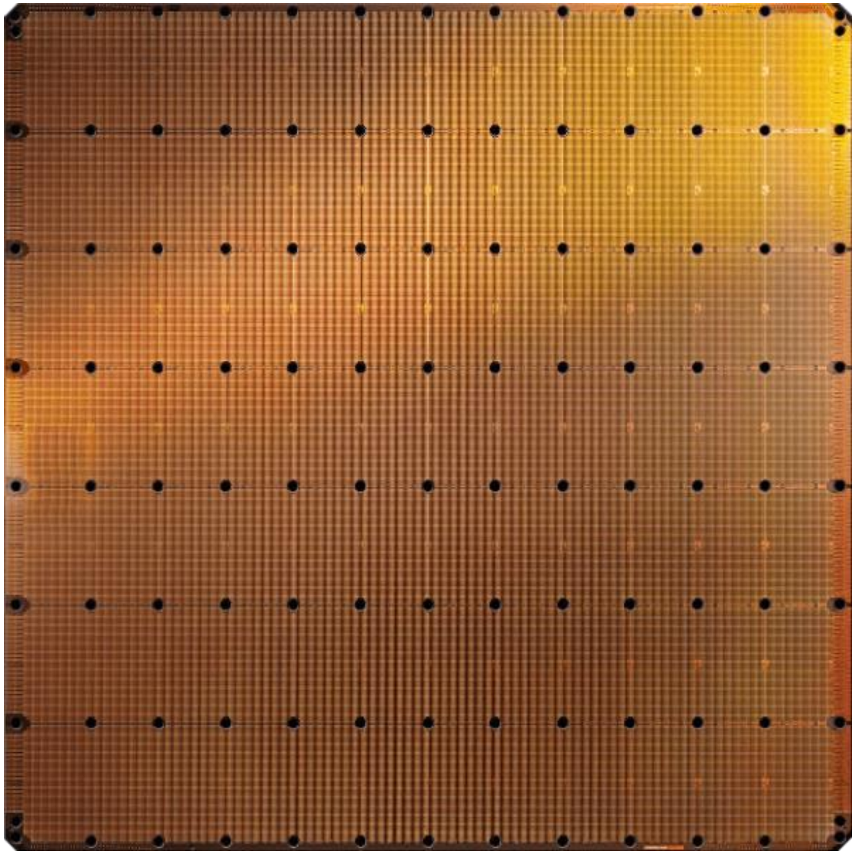
Largest GPU
21.1 Billion transistors
815 mm^2

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)



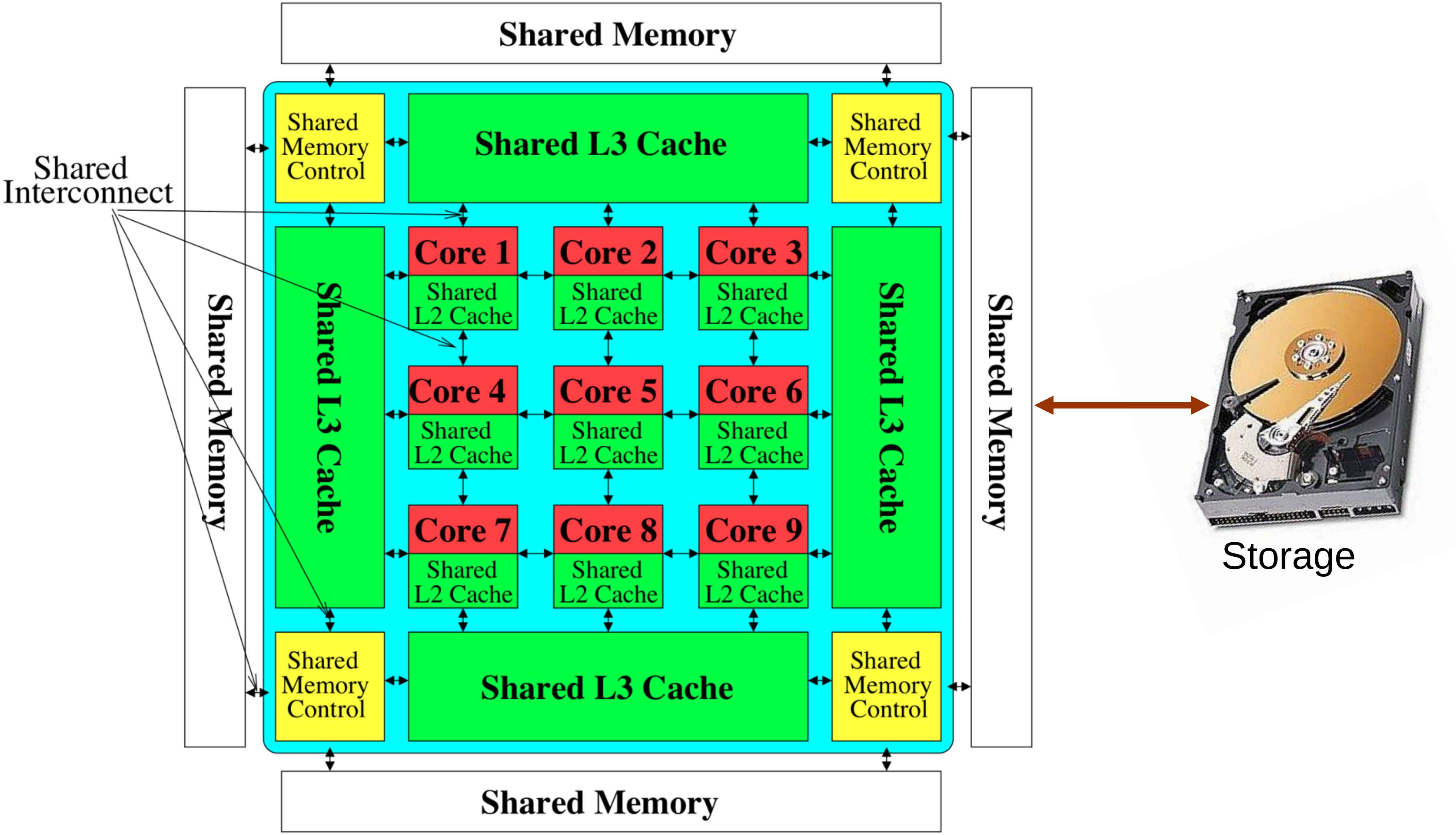
- The largest ML accelerator chip
- 850,000 cores
- **40 GB of on-chip memory**
- **20 PB/s memory bandwidth**

Cerebras WSE-2
2.6 Trillion transistors
46,225 mm^2

https://cerebras.net/product/#overview

Largest GPU
54.2 Billion transistors
826 mm^2
NVIDIA Ampere GA100

Memory System: **Most of the Platform**



Shared Memory

Shared Interconnect

Shared Memory Control | Shared L3 Cache | Shared Memory Control

Shared Memory | Shared L3 Cache | Core 1 Shared L2 Cache | Core 2 Shared L2 Cache | Core 3 Shared L2 Cache | Shared L3 Cache | Shared Memory

Core 4 Shared L2 Cache | Core 5 Shared L2 Cache | Core 6 Shared L2 Cache

Core 7 Shared L2 Cache | Core 8 Shared L2 Cache | Core 9 Shared L2 Cache

Shared Memory Control | Shared L3 Cache | Shared Memory Control

Shared Memory

Storage

Most of the system is dedicated to storing and moving data

Yet, system is still bottlenecked by memory

Memory is Critical for Performance

- We have seen it many times in this course

- Load-related stalls in pipelining
  - Even with magic “1-cycle” memory assumption
- Load/store handling in OoO execution processors
- OoO execution and memory latency tolerance
- VLIW stalls due to long-latency memory operations
- VLIW memory bank disambiguation
- Many memory banks needed in SIMD processors
  - SIMD vector processing performance example
- GPU register files and memory systems
- Fine-grained multithreading to tolerate memory latency
- ...

The Reason

Computing

is Bottlenecked by Data

Computation is Bottlenecked by Memory

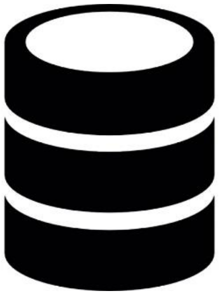
- Important workloads are all data intensive
  - ML/AI, Genomics, Data Analytics, Databases, Graph Analytics, …

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

- Data is increasing
  - We can generate much more than we can process

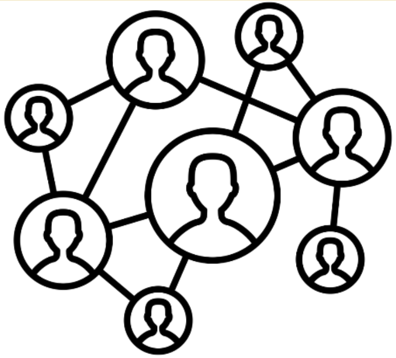
Application Perspective

Memory Is Critical for Performance (I)



In-memory Databases

[Mao+, EuroSys'12;
Clapp+ (**Intel**), IISWC'15]

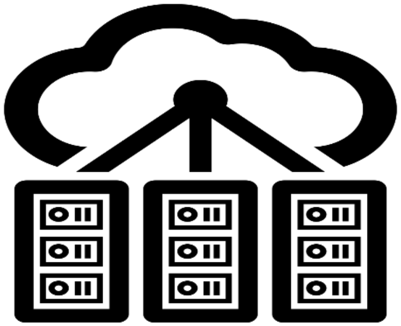


Graph/Tree Processing

[Xu+, IISWC'12; Umuroglu+, FPL'15]

In-Memory Data Analytics

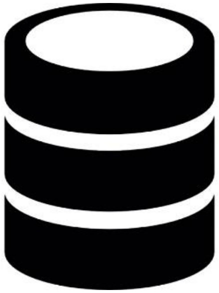
[Clapp+ (**Intel**), IISWC'15;
Awan+, BDCloud'15]



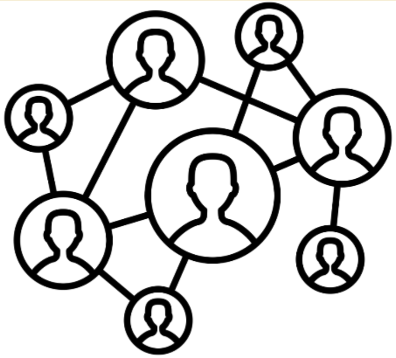
Datacenter Workloads

[Kanev+ (**Google**), ISCA'15]

Memory Is Critical for Performance (I)



In-memory Databases

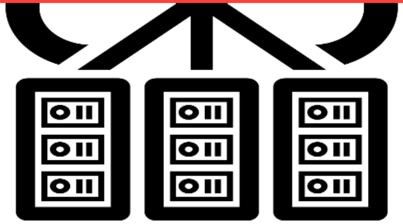


Graph/Tree Processing

Memory → bottleneck

In-Memory Data Analytics

[Clapp+ (**Intel**), IISWC'15;
Awan+, BDCloud'15]



Datacenter Workloads

[Kanev+ (**Google**), ISCA'15]

Memory Is Critical for Performance (II)

Chrome

Google's web browser

TensorFlow Mobile

Google's machine learning framework

Video Playback

Google's video codec

Video Capture

Google's video codec

Memory Is Critical for Performance (II)

Chrome

TensorFlow Mobile

Memory → bottleneck

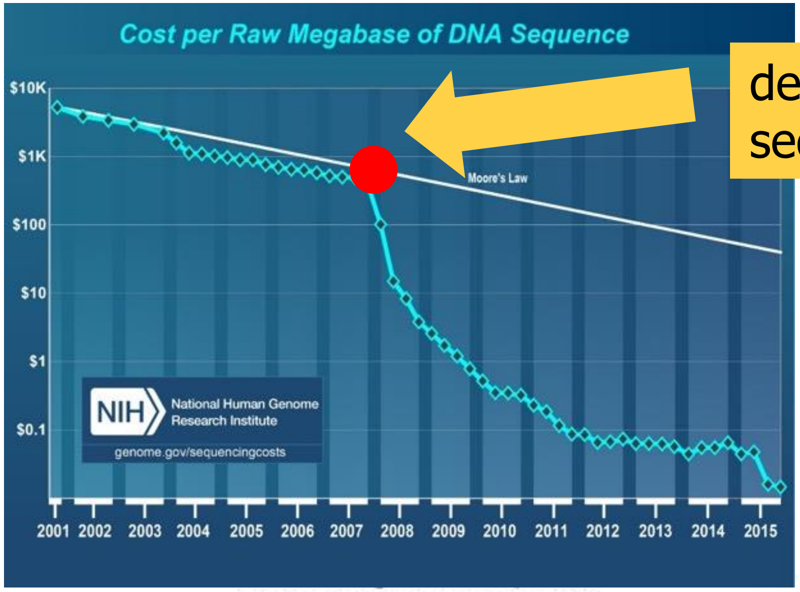
Video Playback

Google's **video codec**

Video Capture

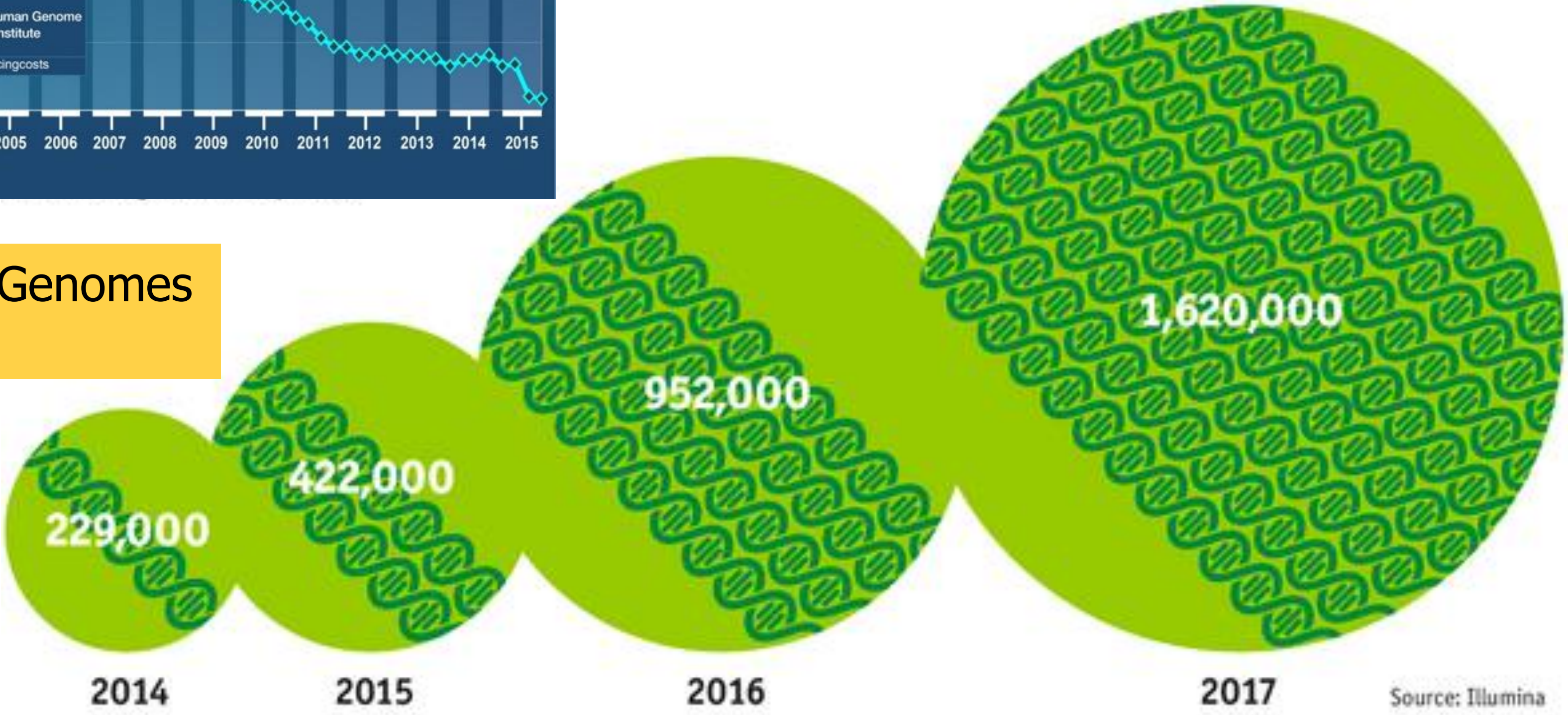
Google's **video codec**

Data is Key for Modern & Future Workloads



development of high-throughput sequencing (HTS) technologies

Number of Genomes Sequenced



Source: Illumina

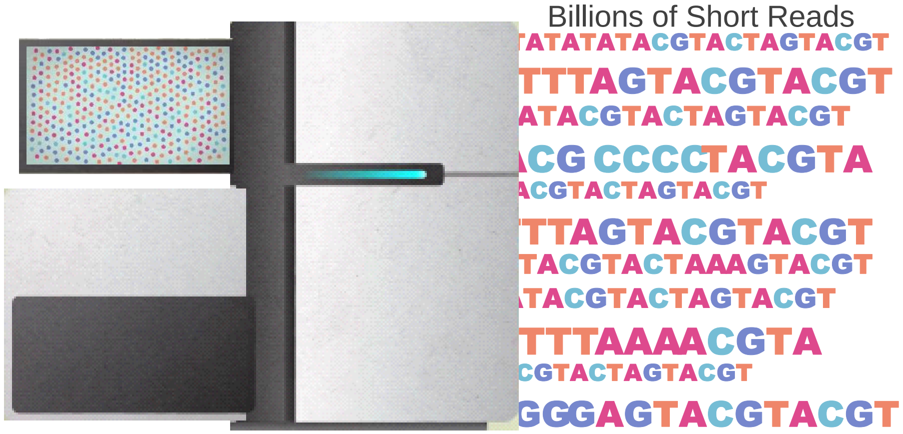
The Economist

SAFARI

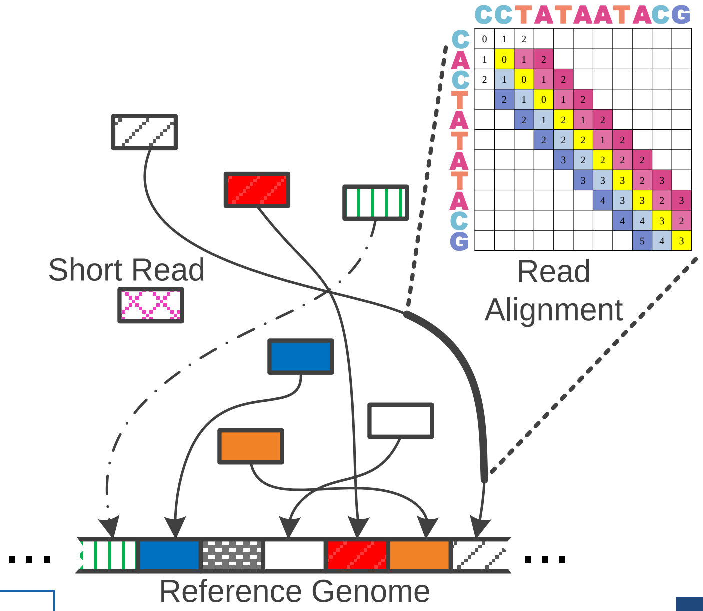
http://www.economist.com/news/21631808-so-much-genetic-data-so-many-uses-genes-unzipped

Genome Analysis

1 Sequencing



2 Read Mapping

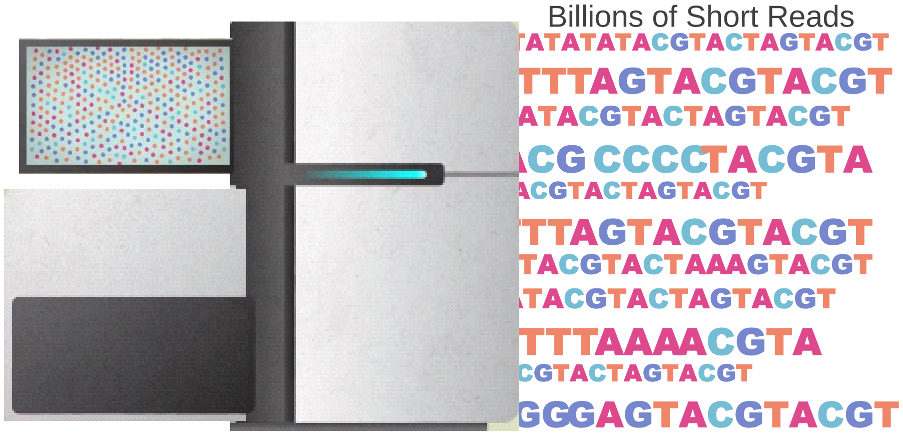


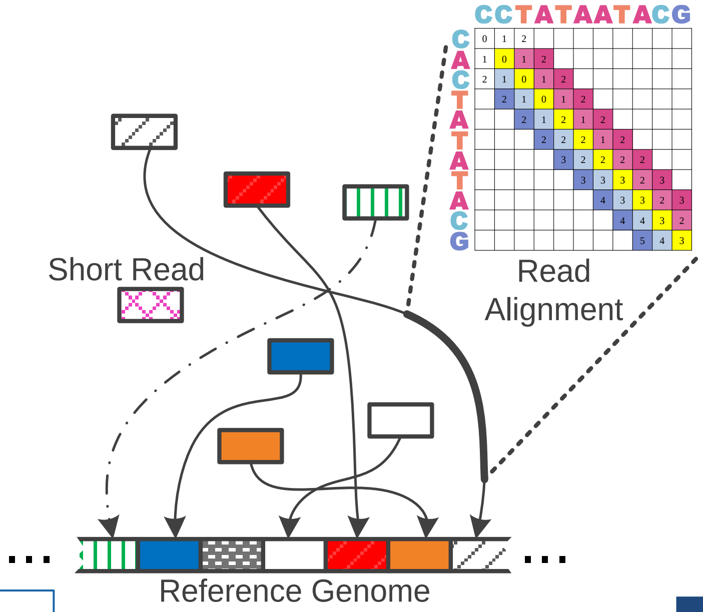
	C	C	T	A	T	A	A	T	A	C	G
C	0	1	2								
A	1	0	1	2							
C	2	1	0	1	2						
T		2	1	0	1	2					
A			2	1	2	1	2				
T				2	2	2	1	2			
A					3	2	2	2	2		
T						3	3	3	2	3	
A							4	3	3	2	3
C								4	4	3	2
G									5	4	3

3 Variant Calling

```
reference: TTTATCGCTTCCATGACGCAG
read1:          ATCGCATCC
read2:         TATCGCATC
read3:              CATCCATGA
read4:            CGCTTCCAT
read5:                 CCATGACGC
read6:               TTCCATGAC
```

4 Scientific Discovery





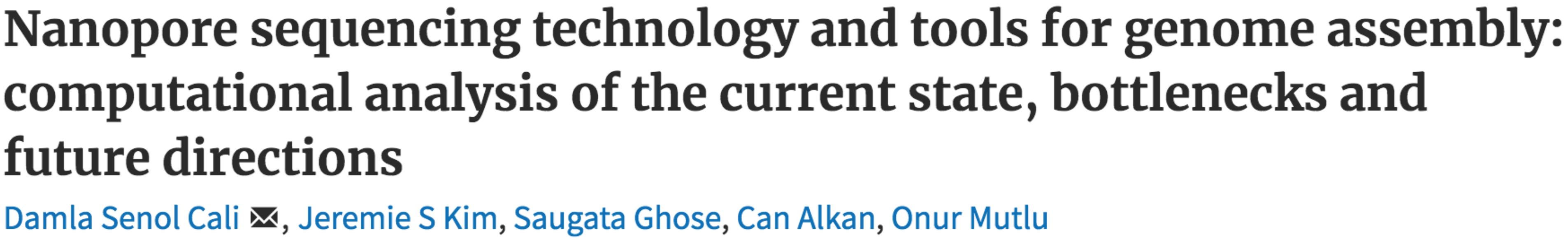
Memory → bottleneck

```
reference: TTTATCGCTTCCATGACGCAG
read1:        ATCGCATCC
read2:       TATCGCATC
read3:            CATCCATGA
read4:          CGCTTCCAT
read5:               CCATGACGC
read6:             TTCCATGAC
```

3 Variant Calling

Scientific Discovery 4

New Genome Sequencing Technologies



Nanopore sequencing technology and tools for genome assembly: computational analysis of the current state, bottlenecks and future directions

Damla Senol Cali ✉, Jeremie S Kim, Saugata Ghose, Can Alkan, Onur Mutlu

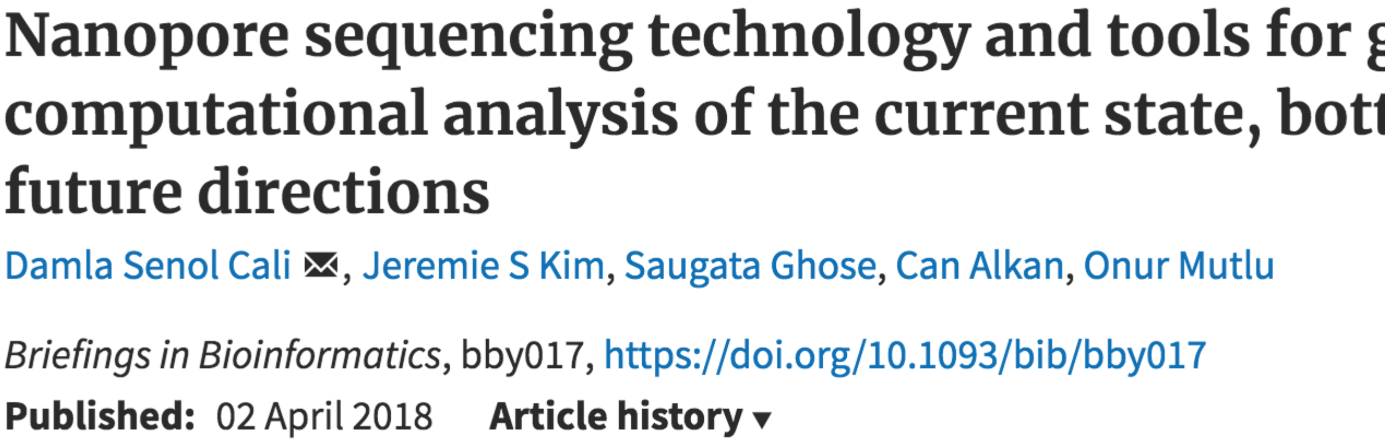
Briefings in Bioinformatics, bby017, https://doi.org/10.1093/bib/bby017

Published: 02 April 2018 **Article history ▾**

Oxford Nanopore MinION

Senol Cali+, "**Nanopore Sequencing Technology and Tools for Genome Assembly: Computational Analysis of the Current State, Bottlenecks and Future Directions**," Briefings in Bioinformatics, 2018.
[Open arxiv.org version]

New Genome Sequencing Technologies



Nanopore sequencing technology and tools for genome assembly: computational analysis of the current state, bottlenecks and future directions

Damla Senol Cali ✉, Jeremie S Kim, Saugata Ghose, Can Alkan, Onur Mutlu

Briefings in Bioinformatics, bby017, https://doi.org/10.1093/bib/bby017

Published: 02 April 2018 **Article history ▾**

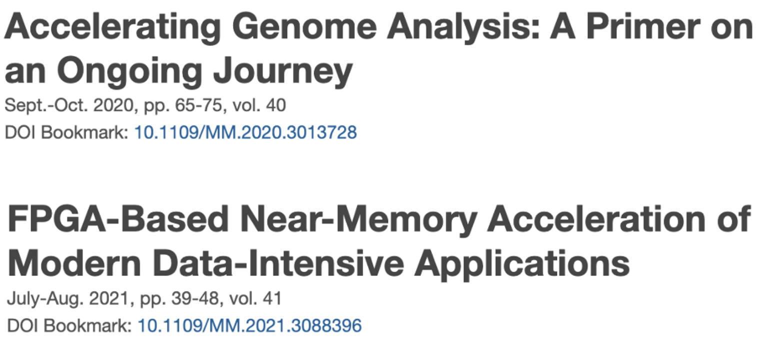
Oxford Nanopore MinION

Memory → bottleneck

Future of Genome Sequencing & Analysis

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



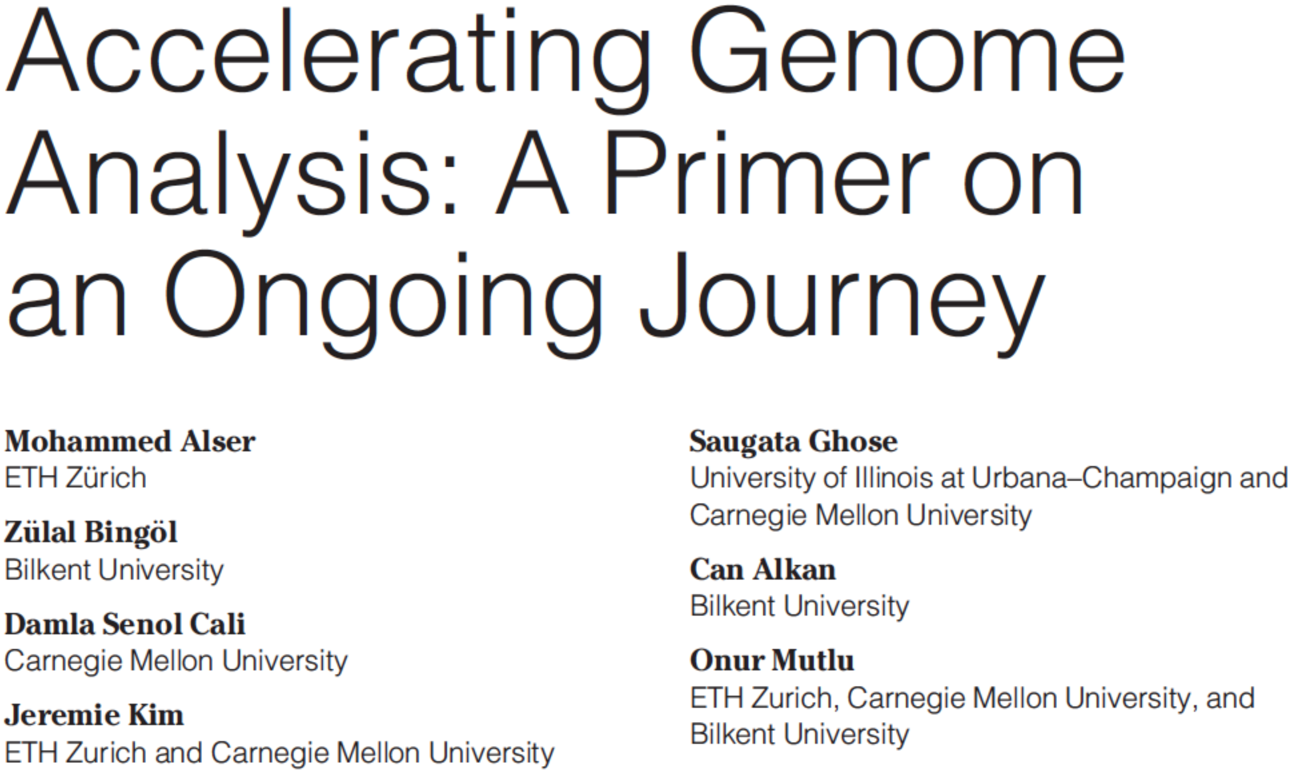
Accelerating Genome Analysis: A Primer on an Ongoing Journey
Sept.-Oct. 2020, pp. 65-75, vol. 40
DOI Bookmark: 10.1109/MM.2020.3013728

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

Accelerating Genome Analysis

- Mohammed Alser, Zulal Bingol, Damla Senol Cali, Jeremie Kim, Saugata Ghose, Can Alkan, and Onur Mutlu,
  "Accelerating Genome Analysis: A Primer on an Ongoing Journey"
  IEEE Micro (***IEEE MICRO***), Vol. 40, No. 5, pages 65-75, September/October 2020.
  [Slides (pptx)(pdf)]
  [Talk Video (1 hour 2 minutes)]



Accelerating Genome Analysis: A Primer on an Ongoing Journey

Mohammed Alser
ETH Zürich

Zülal Bingöl
Bilkent University

Damla Senol Cali
Carnegie Mellon University

Jeremie Kim
ETH Zurich and Carnegie Mellon University

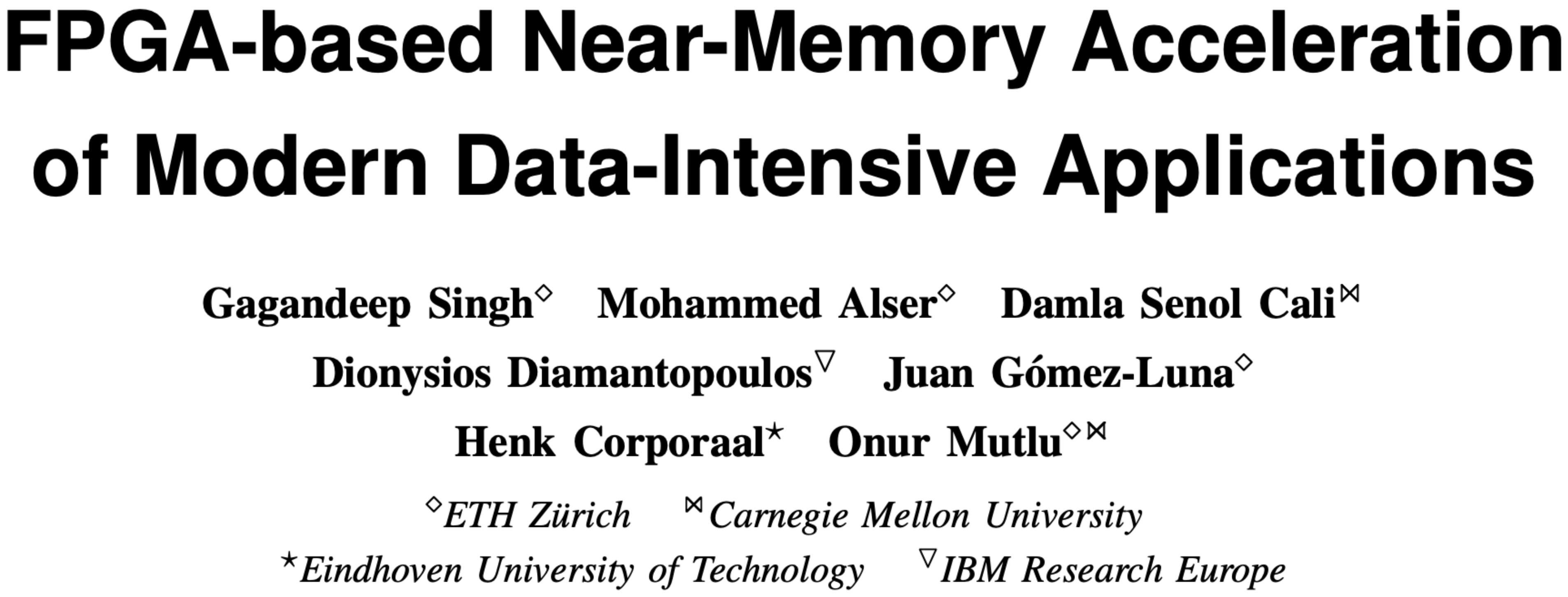
Saugata Ghose
University of Illinois at Urbana–Champaign and Carnegie Mellon University

Can Alkan
Bilkent University

Onur Mutlu
ETH Zurich, Carnegie Mellon University, and Bilkent University

FPGA-based Near-Memory Analytics

- Gagandeep Singh, Mohammed Alser, Damla Senol Cali, Dionysios Diamantopoulos, Juan Gómez-Luna, Henk Corporaal, and Onur Mutlu, **"FPGA-based Near-Memory Acceleration of Modern Data-Intensive Applications"**
  IEEE Micro ***(IEEE MICRO)***, 2021.



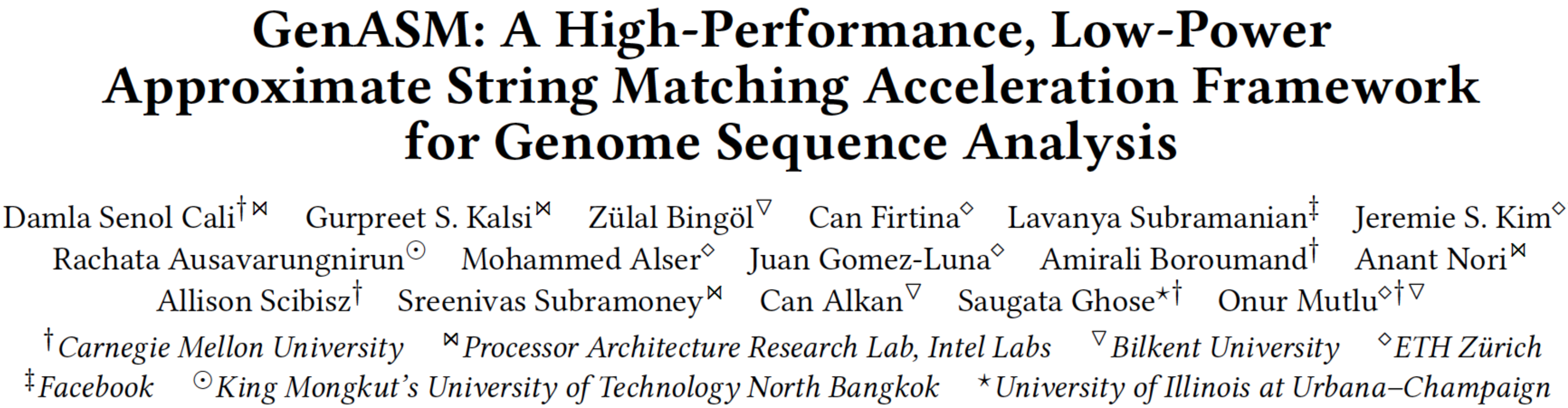
FPGA-based Near-Memory Acceleration of Modern Data-Intensive Applications

Gagandeep Singh[◇] Mohammed Alser[◇] Damla Senol Cali[⋈]
Dionysios Diamantopoulos[▽] Juan Gómez-Luna[◇]
Henk Corporaal[⋆] Onur Mutlu[◇⋈]

[◇]*ETH Zürich* [⋈]*Carnegie Mellon University*
[⋆]*Eindhoven University of Technology* [▽]*IBM Research Europe*

GenASM Acceleration Framework [MICRO 2020]

- Damla Senol Cali, Gurpreet S. Kalsi, Zulal Bingol, Can Firtina, Lavanya Subramanian, Jeremie S. Kim, Rachata Ausavarungnirun, Mohammed Alser, Juan Gomez-Luna, Amirali Boroumand, Anant Nori, Allison Scibisz, Sreenivas Subramoney, Can Alkan, Saugata Ghose, and Onur Mutlu, **"GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis"** *Proceedings of the 53rd International Symposium on Microarchitecture* (***MICRO***), Virtual, October 2020.
  [Lighting Talk Video (1.5 minutes)]
  [Lightning Talk Slides (pptx) (pdf)]
  [Talk Video (18 minutes)]
  [Slides (pptx) (pdf)]



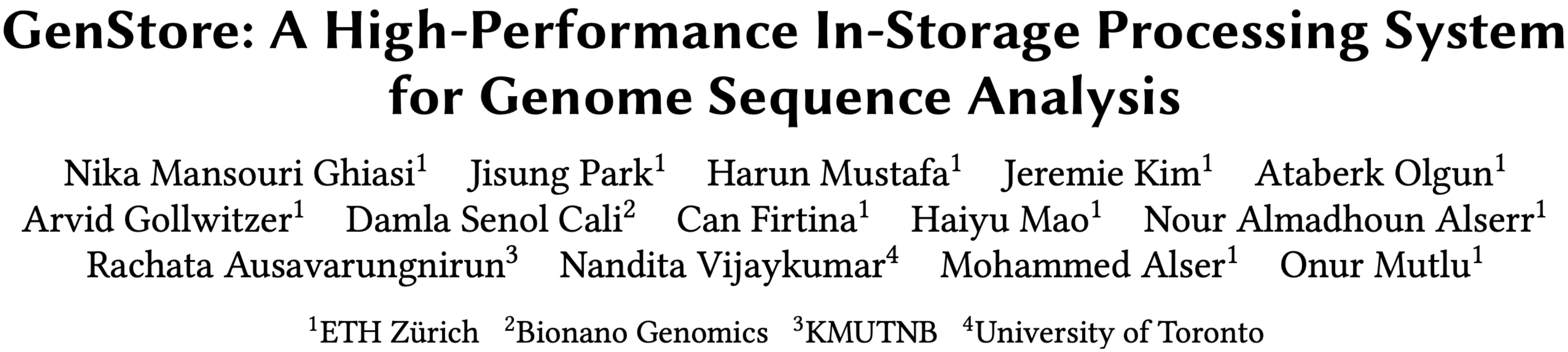
GenASM: A High-Performance, Low-Power Approximate String Matching Acceleration Framework for Genome Sequence Analysis

Damla Senol Cali[†⋈] Gurpreet S. Kalsi[⋈] Zülal Bingöl[▽] Can Firtina[◇] Lavanya Subramanian[‡] Jeremie S. Kim[◇†]
Rachata Ausavarungnirun[⊙] Mohammed Alser[◇] Juan Gomez-Luna[◇] Amirali Boroumand[†] Anant Nori[⋈]
Allison Scibisz[†] Sreenivas Subramoney[⋈] Can Alkan[▽] Saugata Ghose[⋆†] Onur Mutlu[◇†▽]

[†]Carnegie Mellon University [⋈]Processor Architecture Research Lab, Intel Labs [▽]Bilkent University [◇]ETH Zürich
[‡]Facebook [⊙]King Mongkut's University of Technology North Bangkok [⋆]University of Illinois at Urbana–Champaign

In-Storage Genome Filtering **[ASPLOS 2022]**

- Nika Mansouri Ghiasi, Jisung Park, Harun Mustafa, Jeremie Kim, Ataberk Olgun, Arvid Gollwitzer, Damla Senol Cali, Can Firtina, Haiyu Mao, Nour Almadhoun Alserr, Rachata Ausavarungnirun, Nandita Vijaykumar, Mohammed Alser, and Onur Mutlu,
"GenStore: A High-Performance and Energy-Efficient In-Storage Computing System for Genome Sequence Analysis"
*Proceedings of the 27th International Conference on Architectural Support for Programming Languages and Operating Systems (**ASPLOS**)*, Virtual, February-March 2022.
[Talk Slides (pptx) (pdf)]
[Lightning Talk Slides (pptx) (pdf)]
[Lightning Talk Video (90 seconds)]
[Talk Video (17 minutes)]

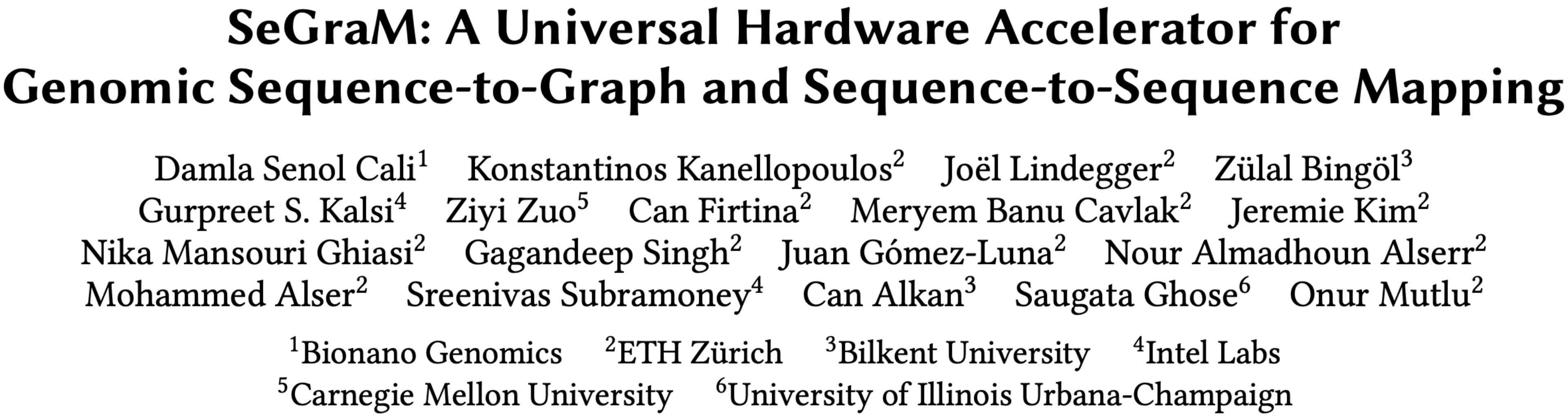


GenStore: A High-Performance In-Storage Processing System for Genome Sequence Analysis

Nika Mansouri Ghiasi[1] Jisung Park[1] Harun Mustafa[1] Jeremie Kim[1] Ataberk Olgun[1]
Arvid Gollwitzer[1] Damla Senol Cali[2] Can Firtina[1] Haiyu Mao[1] Nour Almadhoun Alserr[1]
Rachata Ausavarungnirun[3] Nandita Vijaykumar[4] Mohammed Alser[1] Onur Mutlu[1]

[1]ETH Zürich [2]Bionano Genomics [3]KMUTNB [4]University of Toronto

Sequence-to-Graph Mapping Acceleration [ISCA 2022]



SeGraM: A Universal Hardware Accelerator for Genomic Sequence-to-Graph and Sequence-to-Sequence Mapping

Damla Senol Cali[1] Konstantinos Kanellopoulos[2] Joël Lindegger[2] Zülal Bingöl[3]
Gurpreet S. Kalsi[4] Ziyi Zuo[5] Can Firtina[2] Meryem Banu Cavlak[2] Jeremie Kim[2]
Nika Mansouri Ghiasi[2] Gagandeep Singh[2] Juan Gómez-Luna[2] Nour Almadhoun Alserr[2]
Mohammed Alser[2] Sreenivas Subramoney[4] Can Alkan[3] Saugata Ghose[6] Onur Mutlu[2]

[1]Bionano Genomics [2]ETH Zürich [3]Bilkent University [4]Intel Labs
[5]Carnegie Mellon University [6]University of Illinois Urbana-Champaign

https://arxiv.org/pdf/2205.05883.pdf

More on Fast & Efficient Genome Analysis …

- Onur Mutlu,
  "Accelerating Genome Analysis: A Primer on an Ongoing Journey"
  Invited Lecture at Technion, Virtual, 26 January 2021.
  [Slides (pptx) (pdf)]
  [Talk Video (1 hour 37 minutes, including Q&A)]
  [Related Invited Paper (at IEEE Micro, 2020)]

Population-Scale Microbiome Profiling

SAFARI https://blog.wego.com/7-crowded-places-and-events-that-you-will-love/ 30

Onur Mutlu - Invited Lecture @Technion: Accelerating Genome Analysis: A Primer on an Ongoing Journey

740 views • Premiered Feb 6, 2021

Onur Mutlu Lectures
15.9K subscribers

Detailed Lectures on Genome Analysis

- Computer Architecture, Fall 2020, Lecture 3a
  - **Introduction to Genome Sequence Analysis** (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=CrRb32v7SJc&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=5

- Computer Architecture, Fall 2020, Lecture 8
  - **Intelligent Genome Analysis** (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=ygmQpdDTL7o&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=14

- Computer Architecture, Fall 2020, Lecture 9a
  - **GenASM: Approx. String Matching Accelerator** (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=XoLpzmN-Pas&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=15

- Accelerating Genomics Project Course, Fall 2020, Lecture 1
  - **Accelerating Genomics** (ETH Zürich, Fall 2020)
  - https://www.youtube.com/watch?v=rgjl8ZyLsAg&list=PL5Q2soXY2Zi9E2bBVAgCqLgwiDRQDTyId

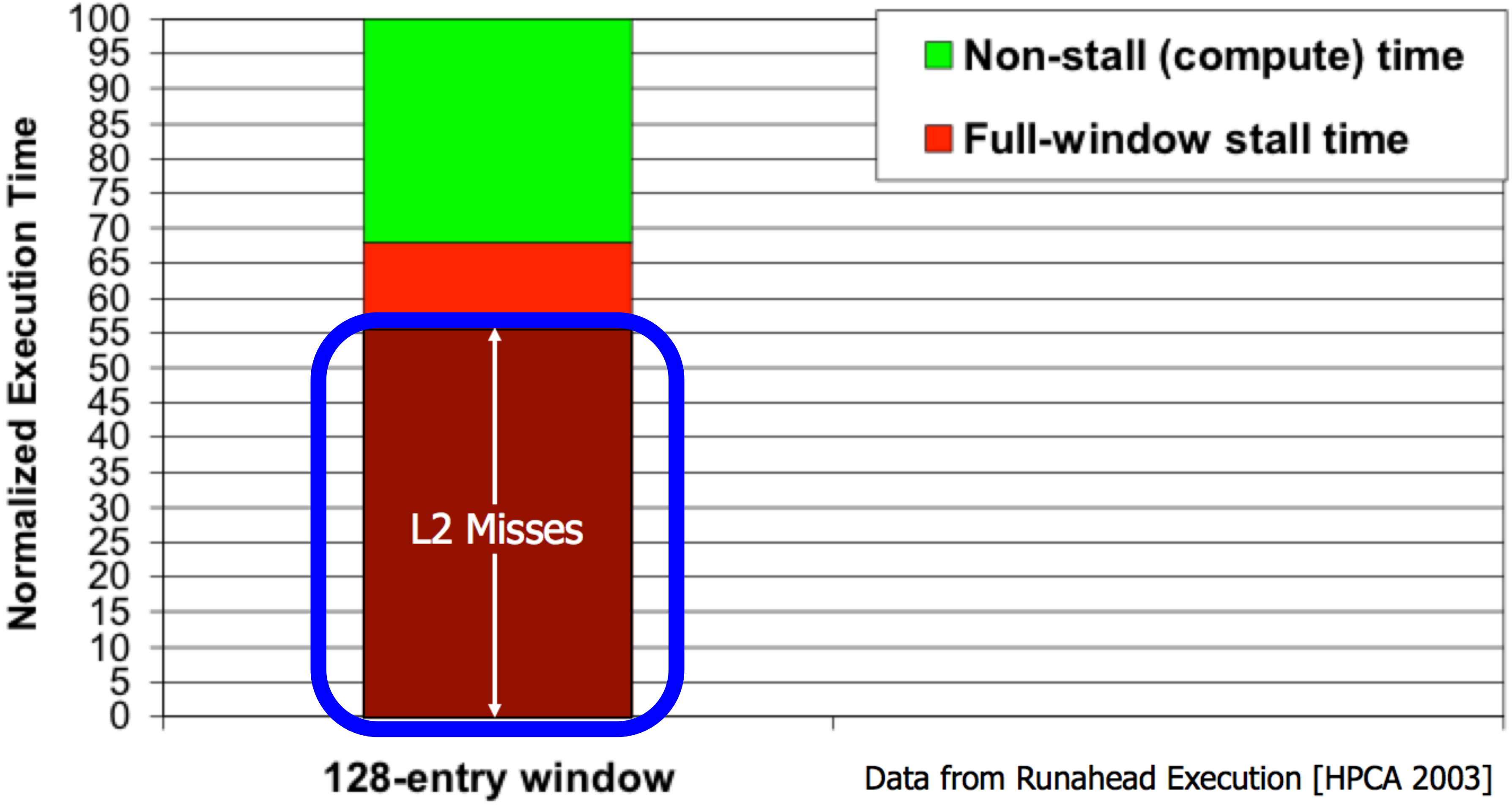
https://www.youtube.com/onurmutlulectures

Performance Perspective

Memory Bottleneck

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

- "**It's the Memory, Stupid!**" (Richard Sites, MPR, 1996)

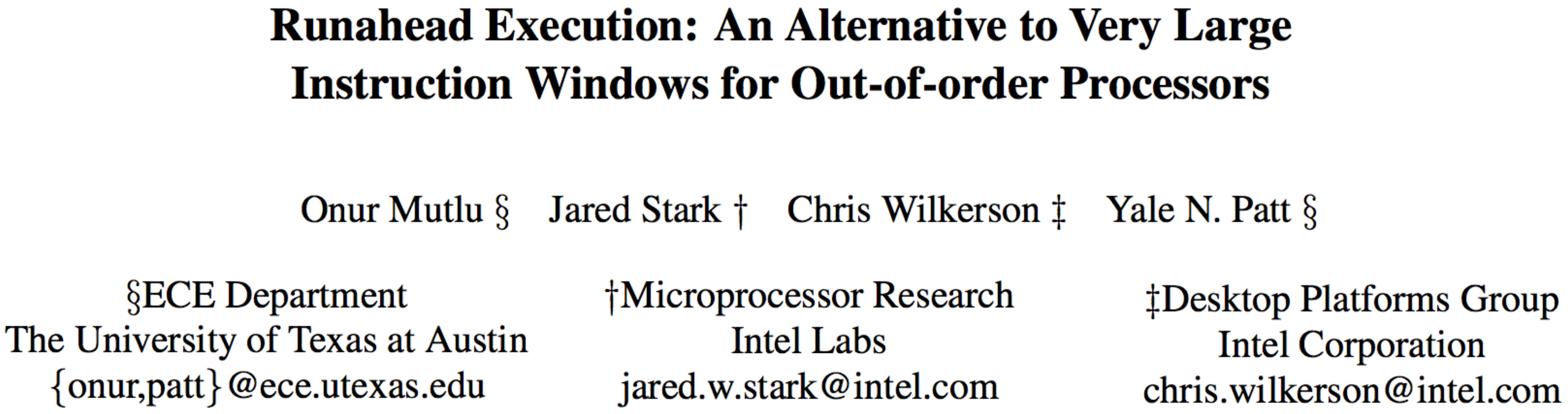


Data from Runahead Execution [HPCA 2003]

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

The Performance Perspective

- Onur Mutlu, Jared Stark, Chris Wilkerson, and Yale N. Patt, **"Runahead Execution: An Alternative to Very Large Instruction Windows for Out-of-order Processors"** *Proceedings of the 9th International Symposium on High-Performance Computer Architecture (**HPCA**)*, pages 129-140, Anaheim, CA, February 2003. Slides (pdf)
  One of the 15 computer arch. papers of 2003 selected as Top Picks by IEEE Micro. HPCA Test of Time Award (awarded in 2021).
  [Lecture Slides (pptx) (pdf)]
  [Lecture Video (1 hr 54 mins)]
  [Retrospective HPCA Test of Time Award Talk Slides (pptx) (pdf)]
  [Retrospective HPCA Test of Time Award Talk Video (14 minutes)]



Runahead Execution: An Alternative to Very Large Instruction Windows for Out-of-order Processors

Onur Mutlu § Jared Stark † Chris Wilkerson ‡ Yale N. Patt §

§ECE Department
The University of Texas at Austin
{onur,patt}@ece.utexas.edu

†Microprocessor Research
Intel Labs
jared.w.stark@intel.com

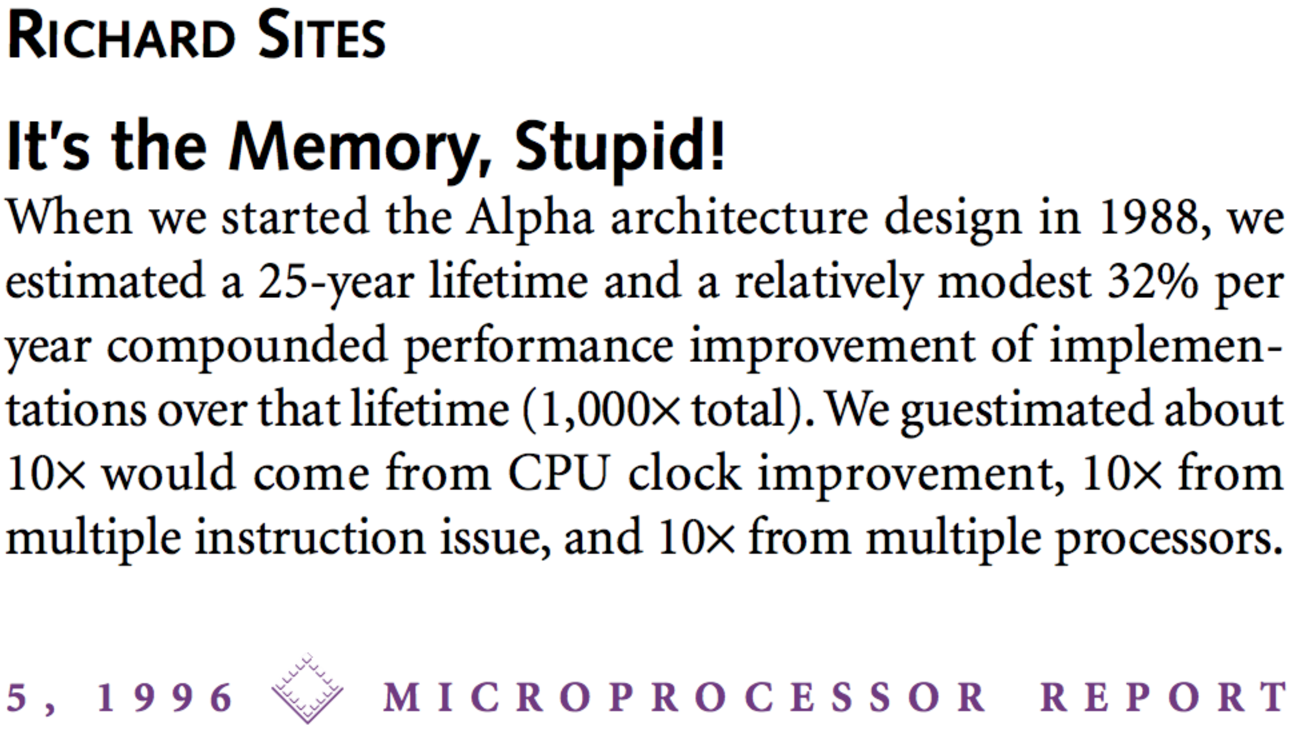
‡Desktop Platforms Group
Intel Corporation
chris.wilkerson@intel.com

The Memory Bottleneck

- Onur Mutlu, Jared Stark, Chris Wilkerson, and Yale N. Patt, **"Runahead Execution: An Effective Alternative to Large Instruction Windows"** *IEEE Micro, Special Issue: Micro's Top Picks from Microarchitecture Conferences* (***MICRO TOP PICKS***), Vol. 23, No. 6, pages 20-25, November/December 2003.

RUNAHEAD EXECUTION:
AN EFFECTIVE ALTERNATIVE TO
LARGE INSTRUCTION WINDOWS

The Memory Bottleneck



RICHARD SITES

It's the Memory, Stupid!

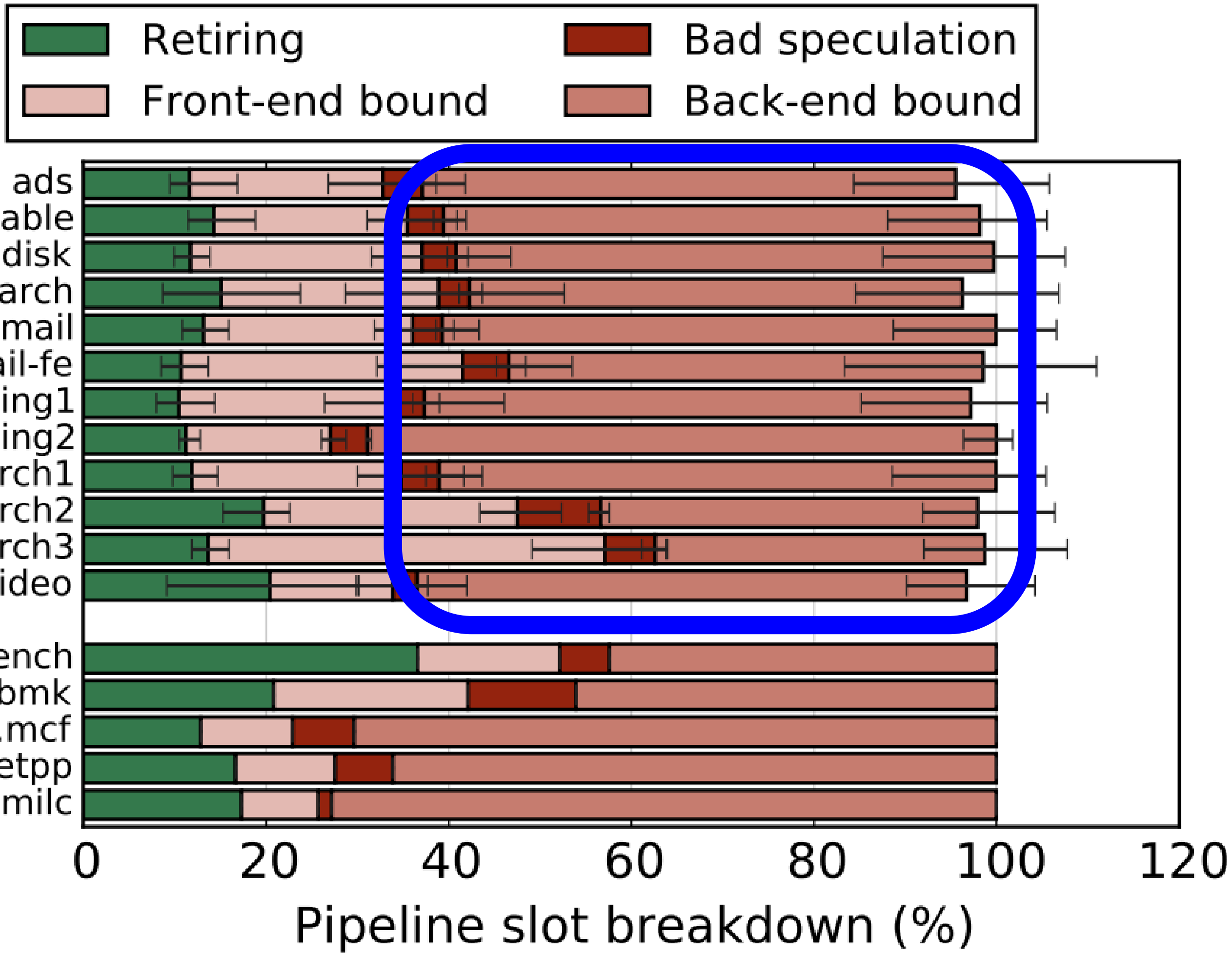
When we started the Alpha architecture design in 1988, we estimated a 25-year lifetime and a relatively modest 32% per year compounded performance improvement of implementations over that lifetime (1,000× total). We guestimated about 10× would come from CPU clock improvement, 10× from multiple instruction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

5, 1996 ◇ MICROPROCESSOR REPORT

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

The Memory Bottleneck

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



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

The Memory Bottleneck

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

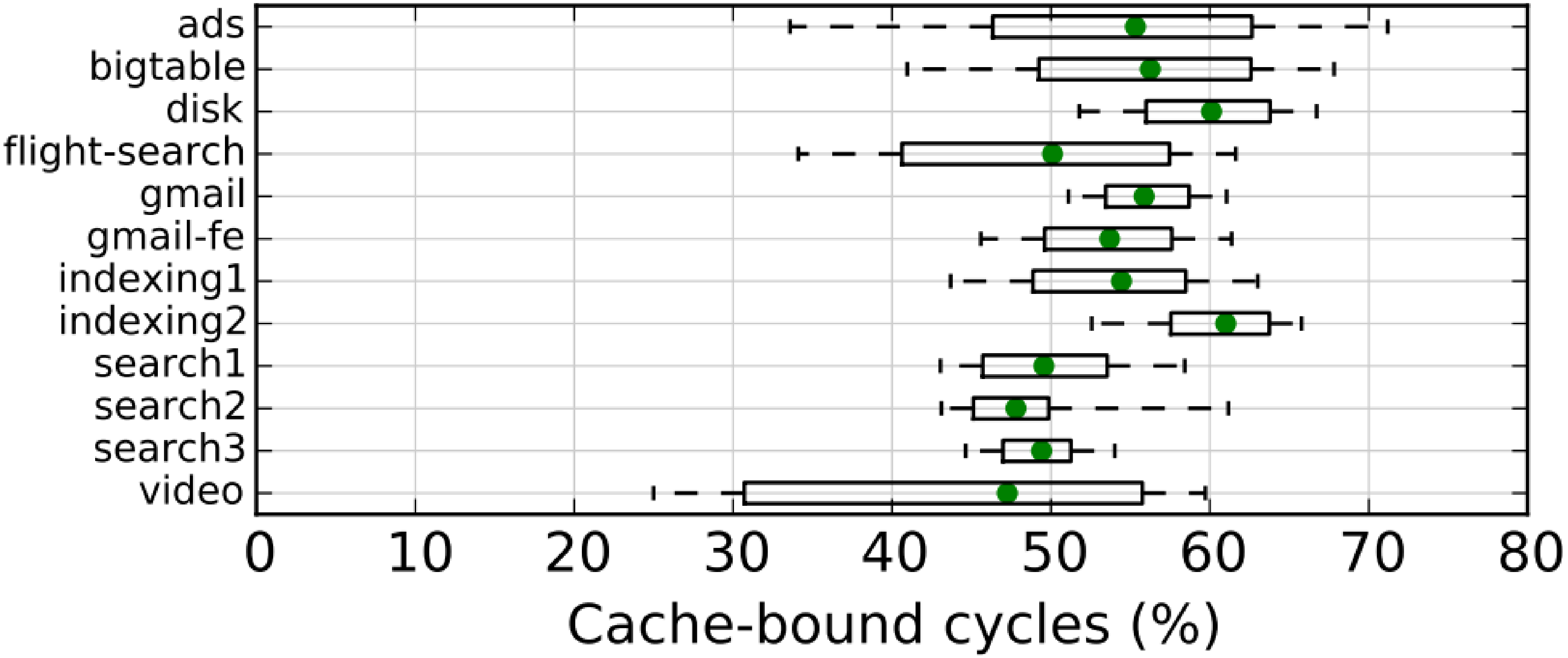
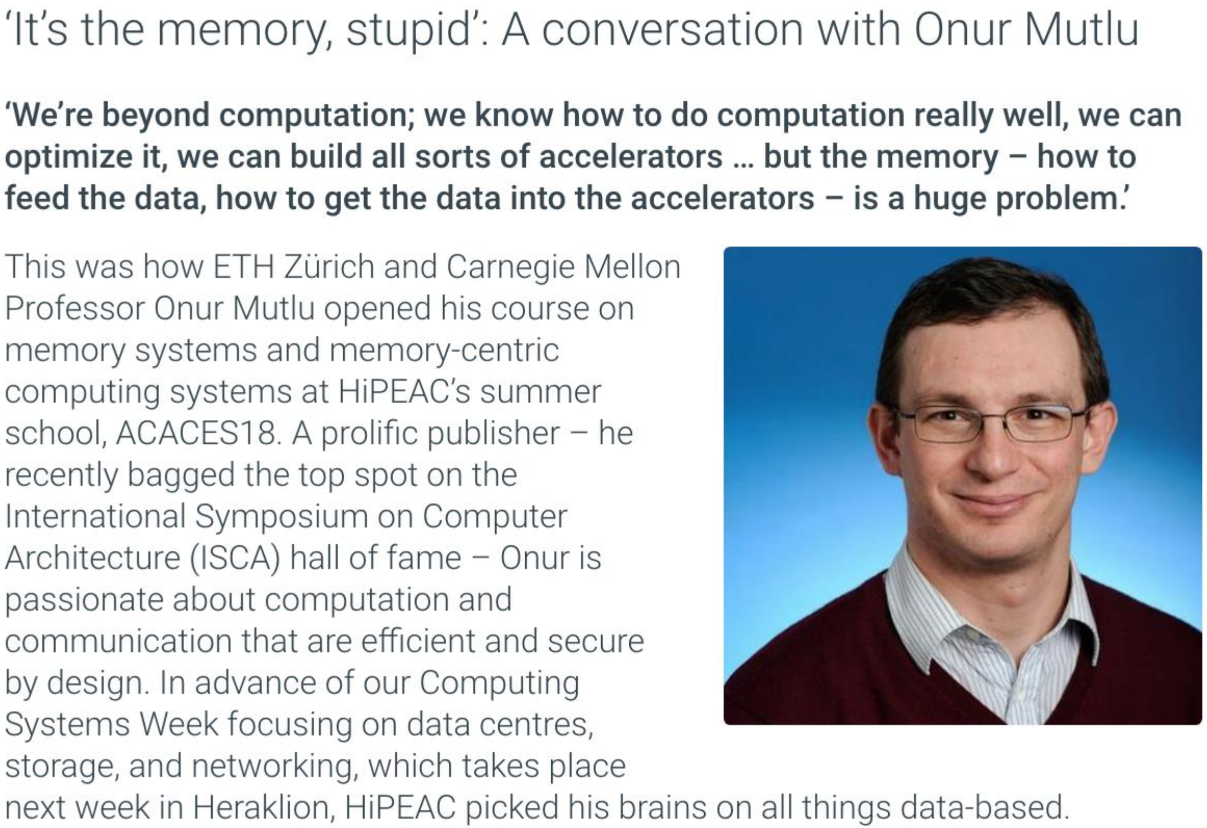


Figure 11: Half of cycles are spent stalled on caches.

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

An Informal Interview on Memory

- Madeleine Gray and Onur Mutlu,
  "'It's the memory, stupid': A conversation with Onur Mutlu"
  HiPEAC info 55, *HiPEAC Newsletter*, October 2018.
  [Shorter Version in Newsletter]
  [Longer Online Version with References]



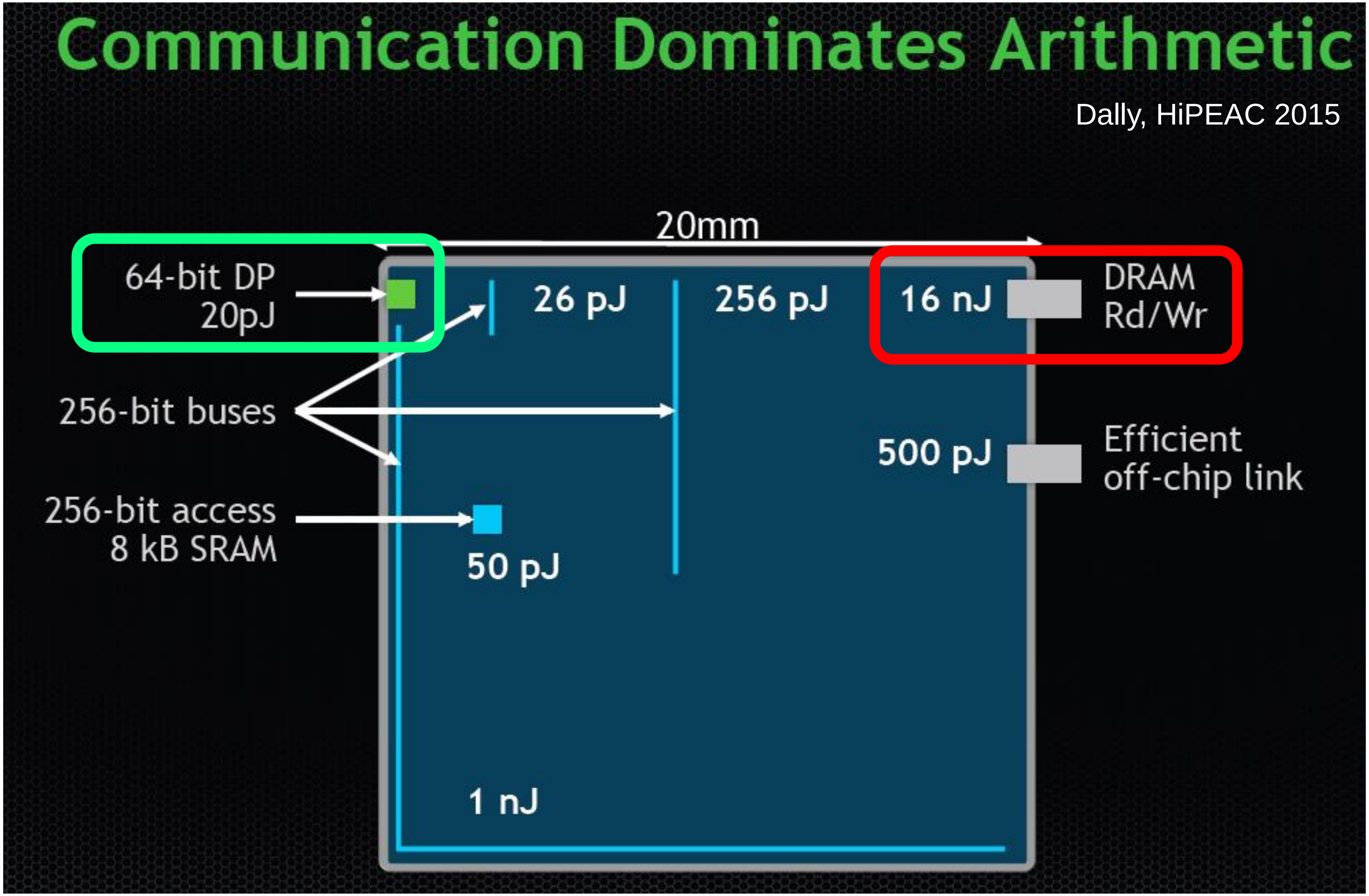
'It's the memory, stupid': A conversation with Onur Mutlu

'We're beyond computation; we know how to do computation really well, we can optimize it, we can build all sorts of accelerators ... but the memory – how to feed the data, how to get the data into the accelerators – is a huge problem.'

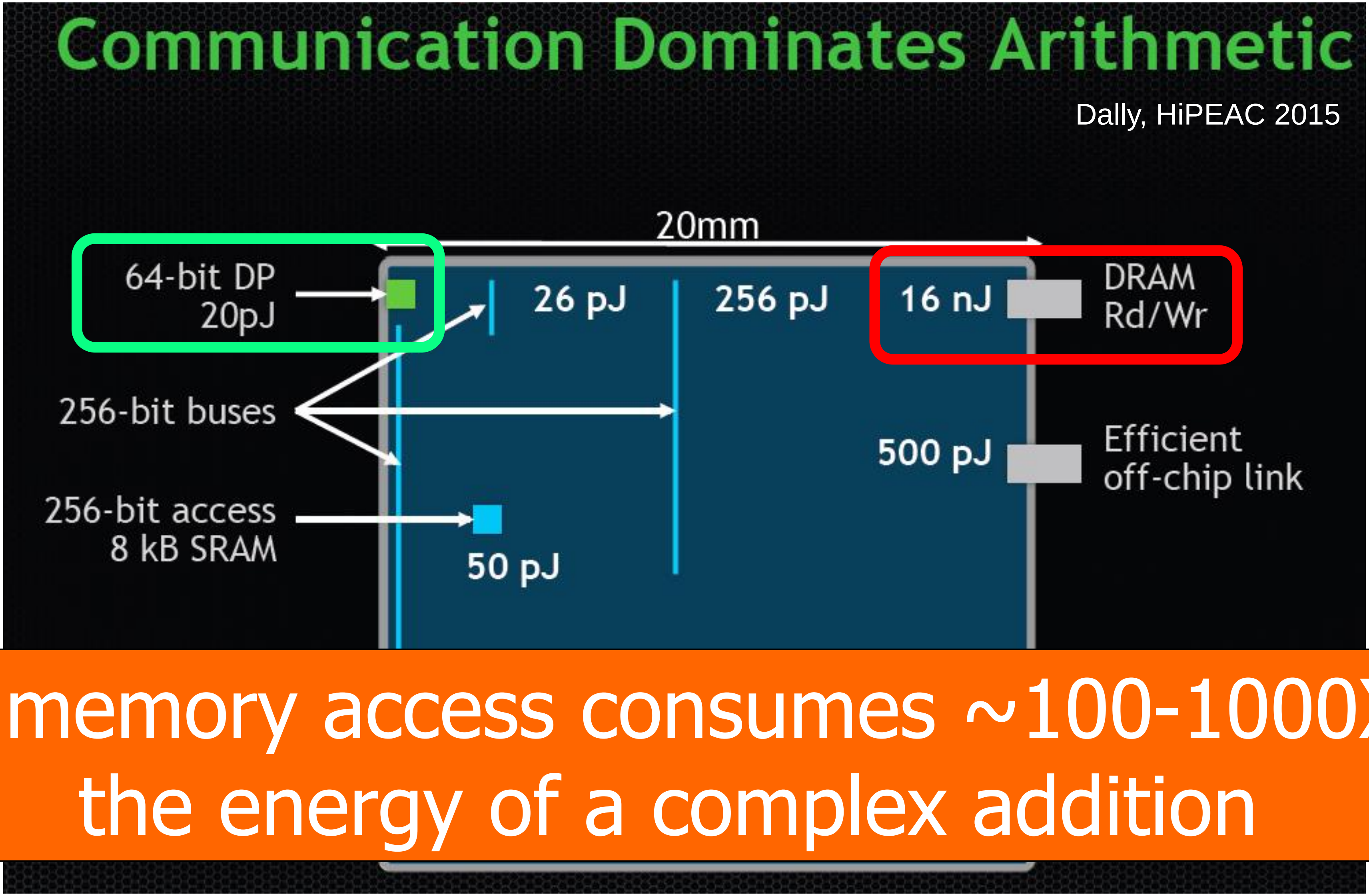
This was how ETH Zürich and Carnegie Mellon Professor Onur Mutlu opened his course on memory systems and memory-centric computing systems at HiPEAC's summer school, ACACES18. A prolific publisher – he recently bagged the top spot on the International Symposium on Computer Architecture (ISCA) hall of fame – Onur is passionate about computation and communication that are efficient and secure by design. In advance of our Computing Systems Week focusing on data centres, storage, and networking, which takes place next week in Heraklion, HiPEAC picked his brains on all things data-based.

Energy Perspective

Data Movement vs. Computation Energy



Data Movement vs. Computation Energy

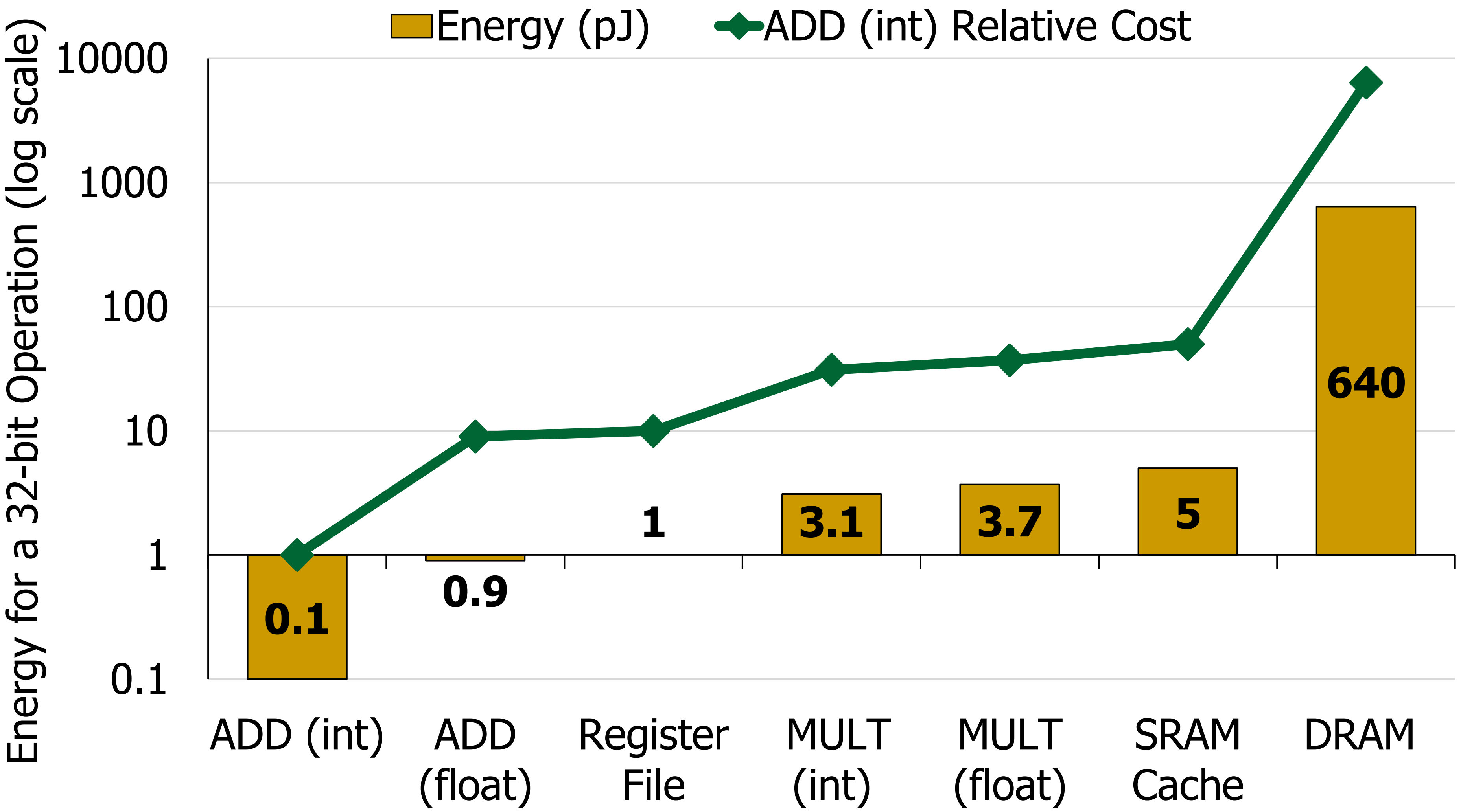


Dally, HiPEAC 2015

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

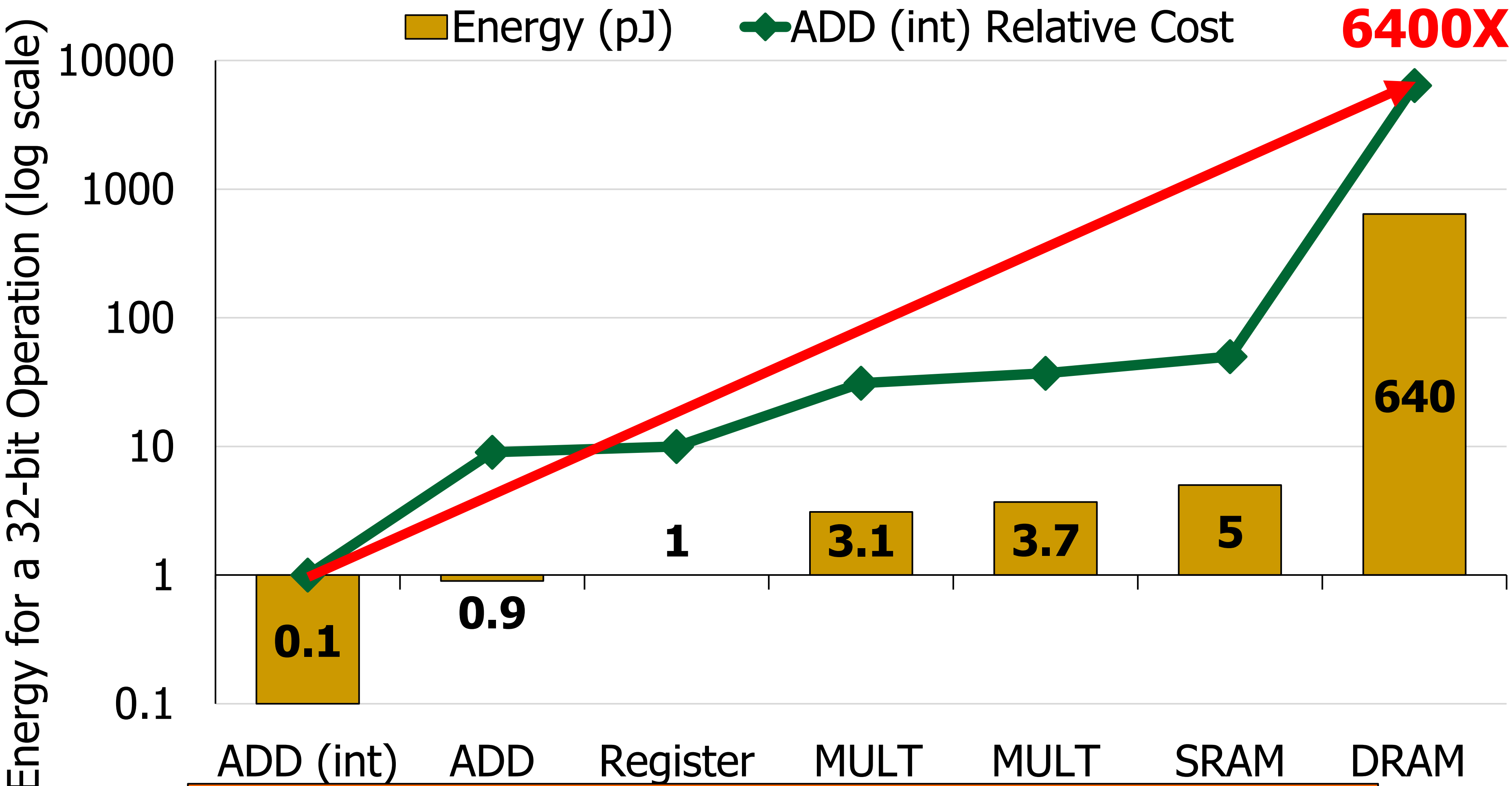
SAFARI

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 an integer addition

Data Movement vs. Computation Energy

32-bit Operation	Energy (pJ)	ADD (int) Relative Cost
ADD (int)	0.1	1
ADD (float)	0.9	9
Register File	1	10
MULT (int)	3.1	31
MULT (float)	3.7	37
SRAM Cache	5	50
DRAM	**640**	**6400**

A memory access consumes ~6400X the energy of an integer addition

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

Memory is Critical for Energy

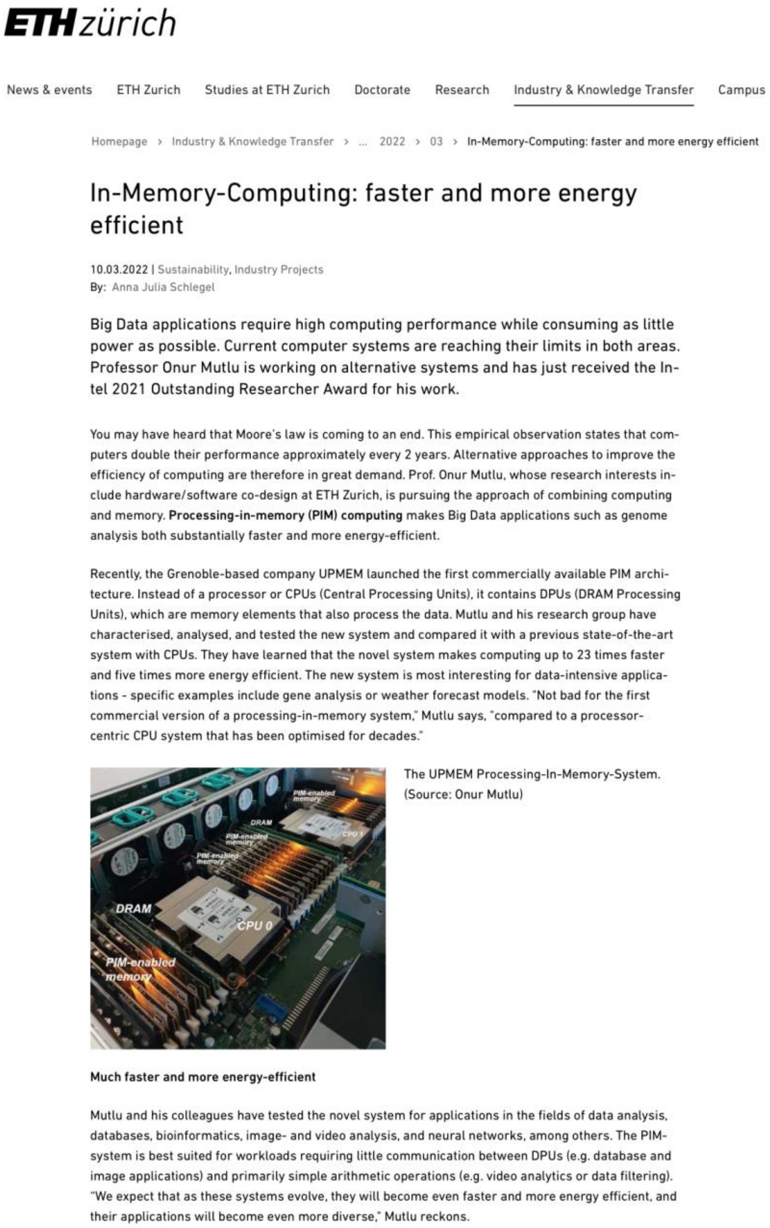
- Amirali Boroumand, Saugata Ghose, Youngsok Kim, Rachata Ausavarungnirun, Eric Shiu, Rahul Thakur, Daehyun Kim, Aki Kuusela, Allan Knies, Parthasarathy Ranganathan, and Onur Mutlu, **"Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks"** *Proceedings of the 23rd International 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]

Processing in Memory: Faster & Low Energy



ETHzürich

News & events ETH Zurich Studies at ETH Zurich Doctorate Research Industry & Knowledge Transfer Campus

Homepage › Industry & Knowledge Transfer › ... 2022 › 03 › In-Memory-Computing: faster and more energy efficient

In-Memory-Computing: faster and more energy efficient

10.03.2022 | Sustainability, Industry Projects
By: Anna Julia Schlegel

Big Data applications require high computing performance while consuming as little power as possible. Current computer systems are reaching their limits in both areas. Professor Onur Mutlu is working on alternative systems and has just received the Intel 2021 Outstanding Researcher Award for his work.

You may have heard that Moore's law is coming to an end. This empirical observation states that computers double their performance approximately every 2 years. Alternative approaches to improve the efficiency of computing are therefore in great demand. Prof. Onur Mutlu, whose research interests include hardware/software co-design at ETH Zurich, is pursuing the approach of combining computing and memory. **Processing-in-memory (PIM) computing** makes Big Data applications such as genome analysis both substantially faster and more energy-efficient.

Recently, the Grenoble-based company UPMEM launched the first commercially available PIM architecture. Instead of a processor or CPUs (Central Processing Units), it contains DPUs (DRAM Processing Units), which are memory elements that also process the data. Mutlu and his research group have characterised, analysed, and tested the new system and compared it with a previous state-of-the-art system with CPUs. They have learned that the novel system makes computing up to 23 times faster and five times more energy efficient. The new system is most interesting for data-intensive applications - specific examples include gene analysis or weather forecast models. "Not bad for the first commercial version of a processing-in-memory system," Mutlu says, "compared to a processor-centric CPU system that has been optimised for decades."

The UPMEM Processing-In-Memory-System. (Source: Onur Mutlu)

Much faster and more energy-efficient

Mutlu and his colleagues have tested the novel system for applications in the fields of data analysis, databases, bioinformatics, image- and video analysis, and neural networks, among others. The PIM-system is best suited for workloads requiring little communication between DPUs (e.g. database and image applications) and primarily simple arithmetic operations (e.g. video analytics or data filtering). "We expect that as these systems evolve, they will become even faster and more energy efficient, and their applications will become even more diverse," Mutlu reckons.

SUSTAINABILITY • INDUSTRY PROJECTS

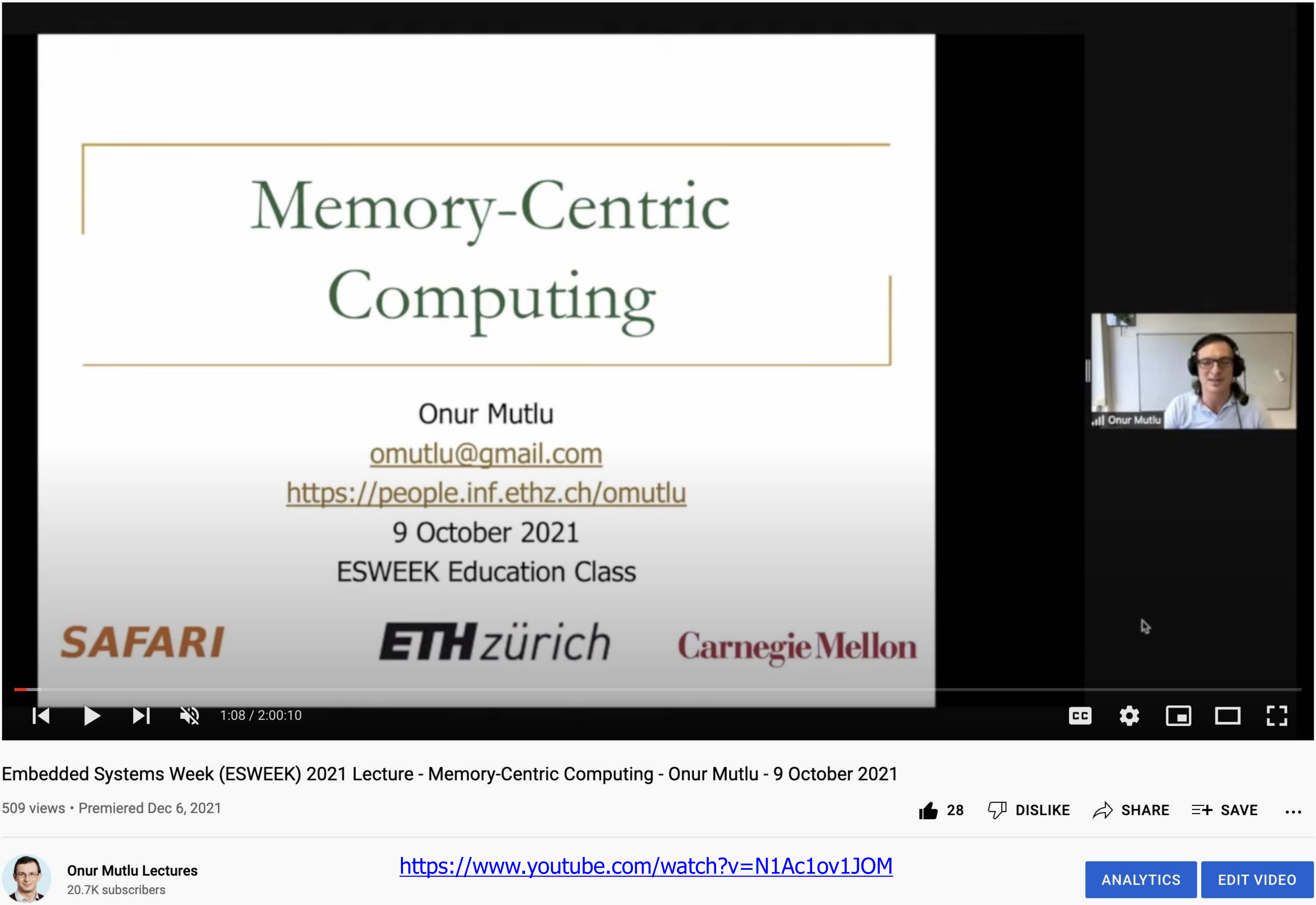
In-Memory-Computing: faster and more energy efficient

https://ethz.ch/en/industry/industry/news/data/2022/03/mehr-daten-schneller-und-energiesparender-verarbeiten.html

Tutorial on Processing in Memory

- Onur Mutlu,
  "Memory-Centric Computing"
  *Education Class at Embedded Systems Week (**ESWEEK**)*,
  Virtual, 9 October 2021.
  [Slides (pptx) (pdf)]
  [Abstract (pdf)]
  [Talk Video (2 hours, including Q&A)]
  [Invited Paper at DATE 2021]
  ["A Modern Primer on Processing in Memory" paper]

https://www.youtube.com/watch?v=N1Ac1ov1JOM



Memory-Centric Computing

Onur Mutlu

omutlu@gmail.com

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

9 October 2021

ESWEEK Education Class

Embedded Systems Week (ESWEEK) 2021 Lecture - Memory-Centric Computing - Onur Mutlu - 9 October 2021

509 views • Premiered Dec 6, 2021

Onur Mutlu Lectures

20.7K subscribers

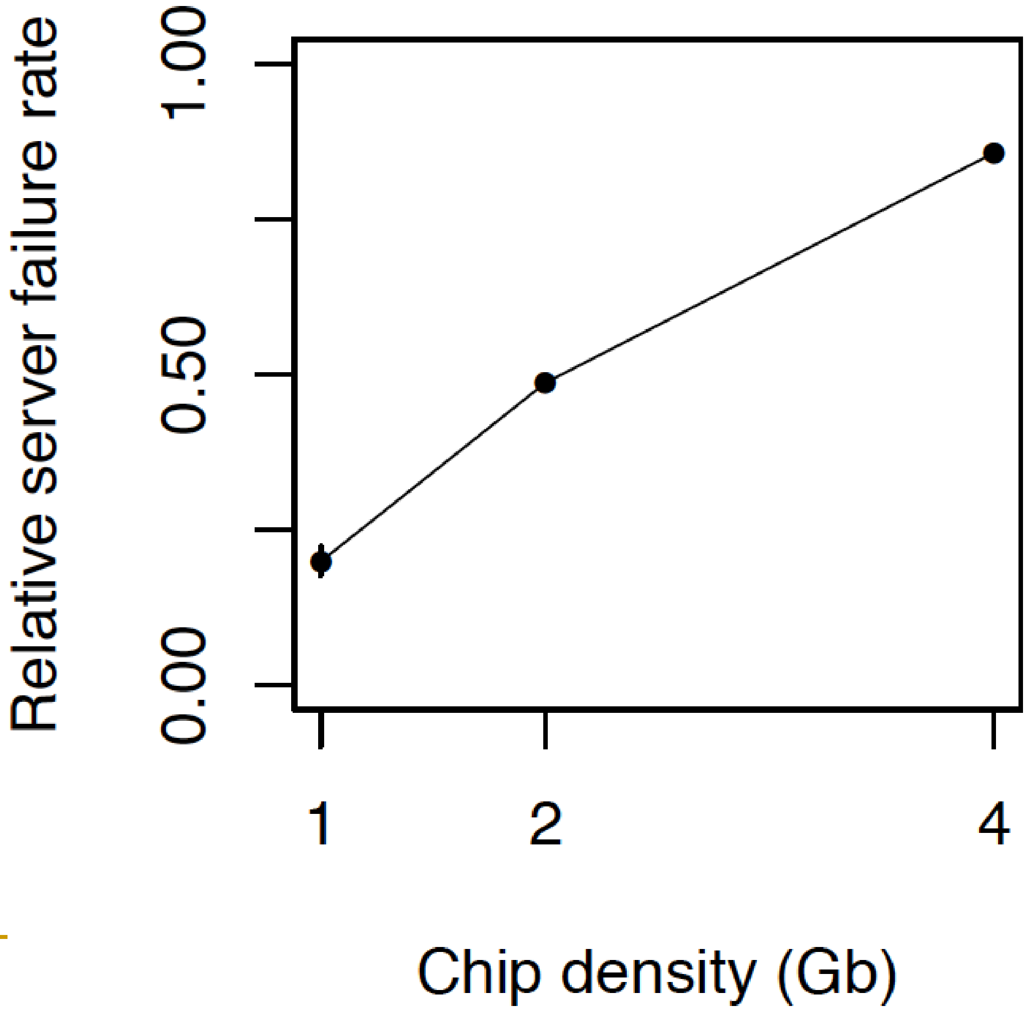
https://www.youtube.com/watch?v=N1Ac1ov1JOM

https://www.youtube.com/onurmutlulectures

Reliability & Security Perspectives

Memory is Critical for Reliability

- Data from all of Facebook's servers worldwide
- Meza+, "Revisiting Memory Errors in Large-Scale Production Data Centers," DSN'15.

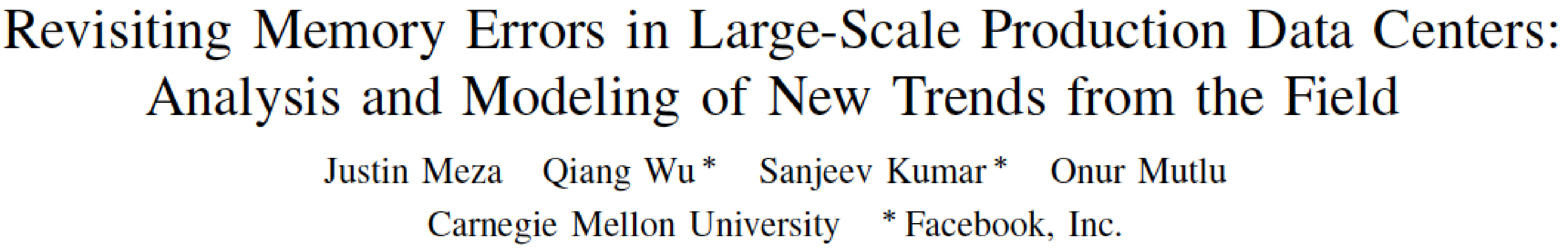


As memory capacity increases, system reliability reduces

Large-Scale Failure Analysis of DRAM Chips

- Analysis and modeling of memory errors found in all of Facebook’s server fleet

- Justin Meza, Qiang Wu, Sanjeev Kumar, and Onur Mutlu, **"Revisiting Memory Errors in Large-Scale Production Data Centers: Analysis and Modeling of New Trends from the Field"** *Proceedings of the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (**DSN**)*, Rio de Janeiro, Brazil, June 2015.
  [Slides (pptx) (pdf)] [DRAM Error Model]



Revisiting Memory Errors in Large-Scale Production Data Centers: Analysis and Modeling of New Trends from the Field

Justin Meza   Qiang Wu *   Sanjeev Kumar *   Onur Mutlu
Carnegie Mellon University   * Facebook, Inc.

A Curious Discovery **[Kim et al., ISCA 2014]**

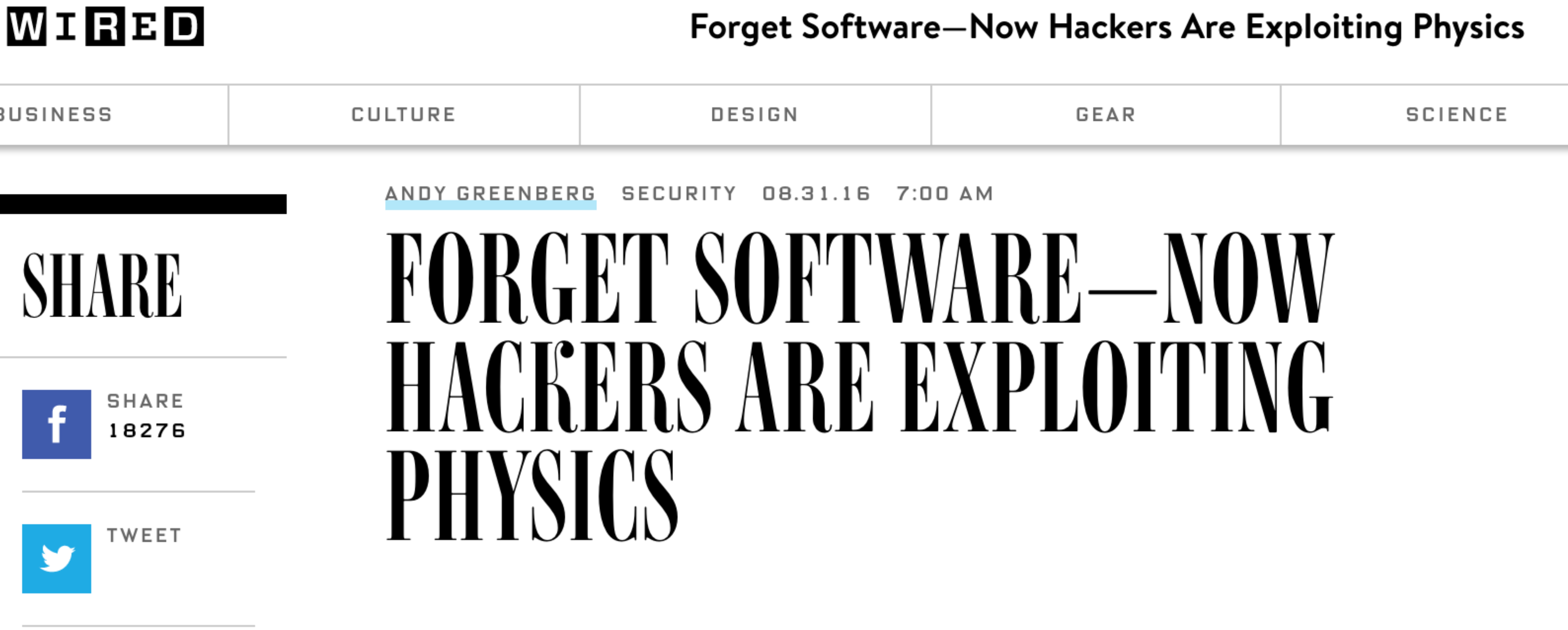
One can

predictably induce errors

in most DRAM memory chips

DRAM RowHammer

A simple hardware failure mechanism
can create a widespread
system security vulnerability

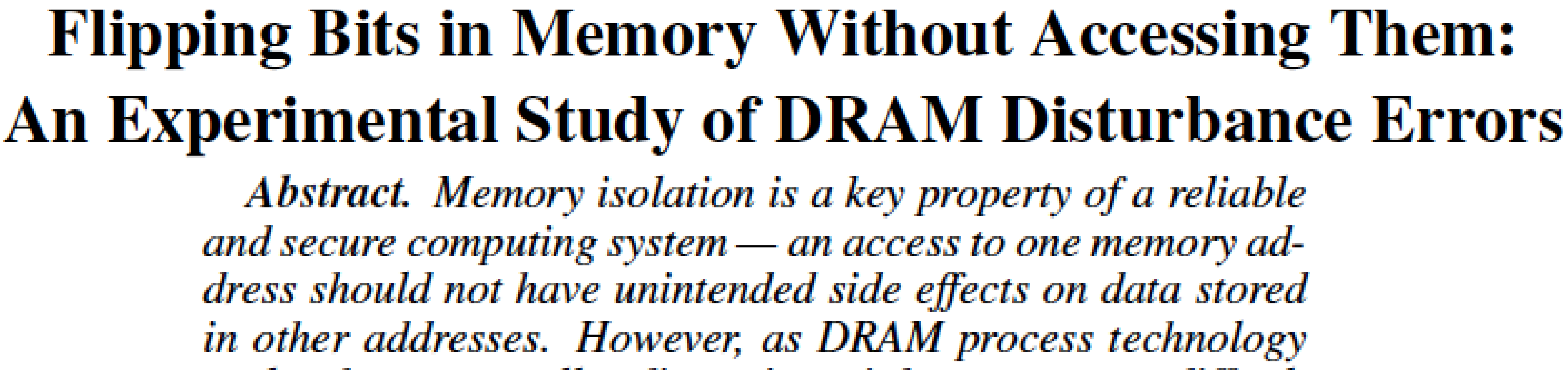


Forget Software—Now Hackers Are Exploiting Physics

ANDY GREENBERG SECURITY 08.31.16 7:00 AM

FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

One Can Take Over an Otherwise-Secure System



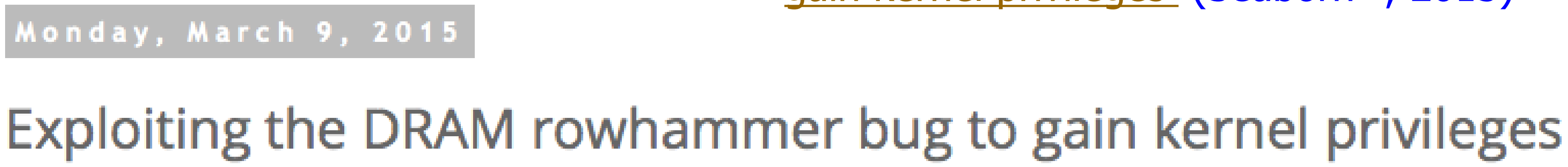
Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors

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

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



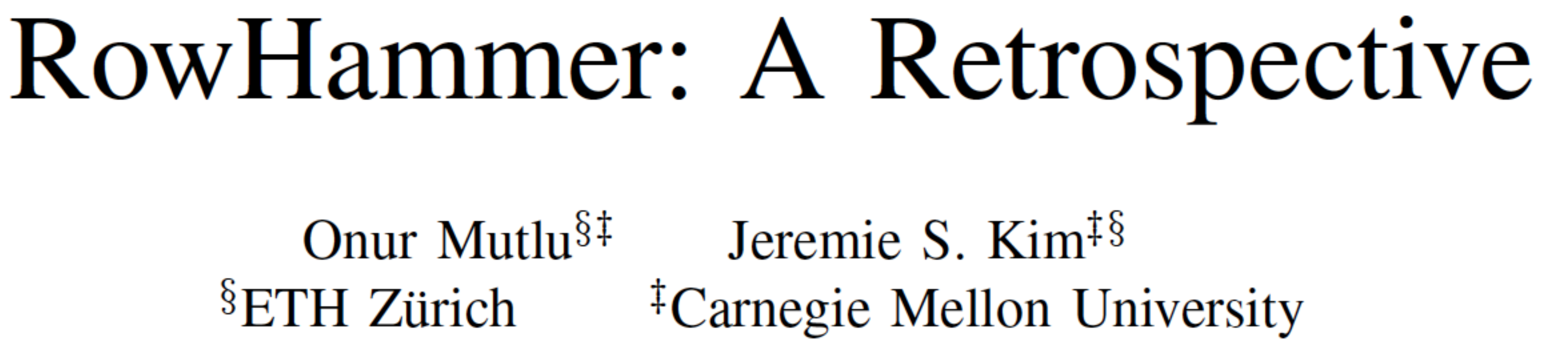
Monday, March 9, 2015

Exploiting the DRAM rowhammer bug to gain kernel privileges

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

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

Memory is Critical for Security

Detailed Lectures on RowHammer

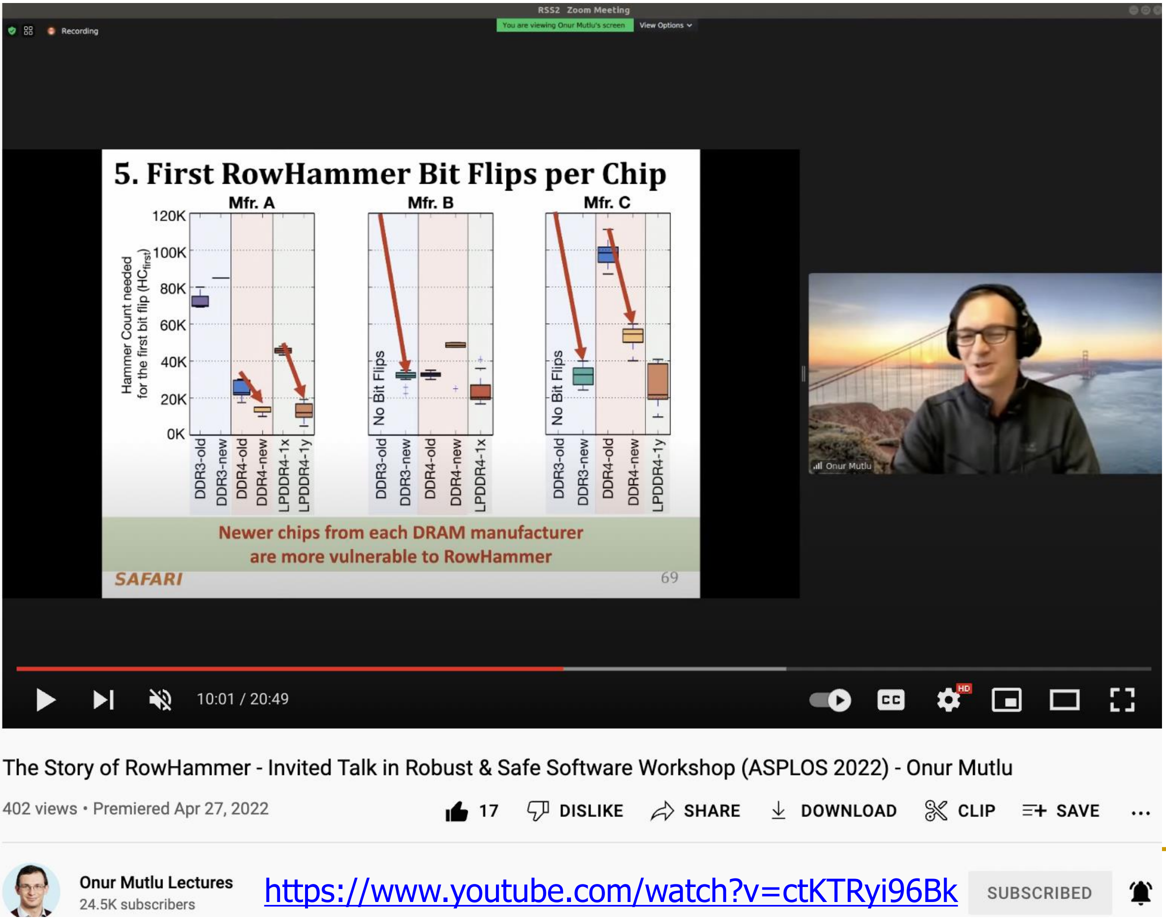
- Computer Architecture, Fall 2021, Lecture 5
  - RowHammer (ETH Zürich, Fall 2021)
  - https://www.youtube.com/watch?v=7wVKnPj3NVw&list=PL5Q2soXY2Zi-Mnk1PxjEIG32HAGILkTOF&index=5

- Computer Architecture, Fall 2021, Lecture 6
  - RowHammer and Secure & Reliable Memory (ETH Zürich, Fall 2021)
  - https://www.youtube.com/watch?v=HNd4skQrt6I&list=PL5Q2soXY2Zi-Mnk1PxjEIG32HAGILkTOF&index=6

https://www.youtube.com/onurmutlulectures

10 Years of RowHammer in 20 Minutes

- Onur Mutlu,
  "The Story of RowHammer"
  *Invited Talk at the Workshop on Robust and Safe Software 2.0 (**RSS2**)*, held with *the 27th International Conference on Architectural Support for Programming Languages and Operating Systems (**ASPLOS**)*, Virtual, 28 February 2022.
  [Slides (pptx) (pdf)]


RSS2 Zoom Meeting
Recording
You are viewing Onur Mutlu's screen
View Options
5. First RowHammer Bit Flips per Chip
Mfr. A
Mfr. B
Mfr. C
Hammer Count needed for the first bit flip (HC_{first})
120K
100K
80K
60K
40K
20K
0K
No Bit Flips
No Bit Flips
DDR3-old
DDR3-new
DDR4-old
DDR4-new
LPDDR4-1x
LPDDR4-1y
Newer chips from each DRAM manufacturer are more vulnerable to RowHammer
SAFARI
69
Onur Mutlu
10:01 / 20:49
HD

The Story of RowHammer - Invited Talk in Robust & Safe Software Workshop (ASPLOS 2022) - Onur Mutlu

402 views • Premiered Apr 27, 2022

https://www.youtube.com/watch?v=ctKTRyi96Bk

Memory Is Critical for Computing

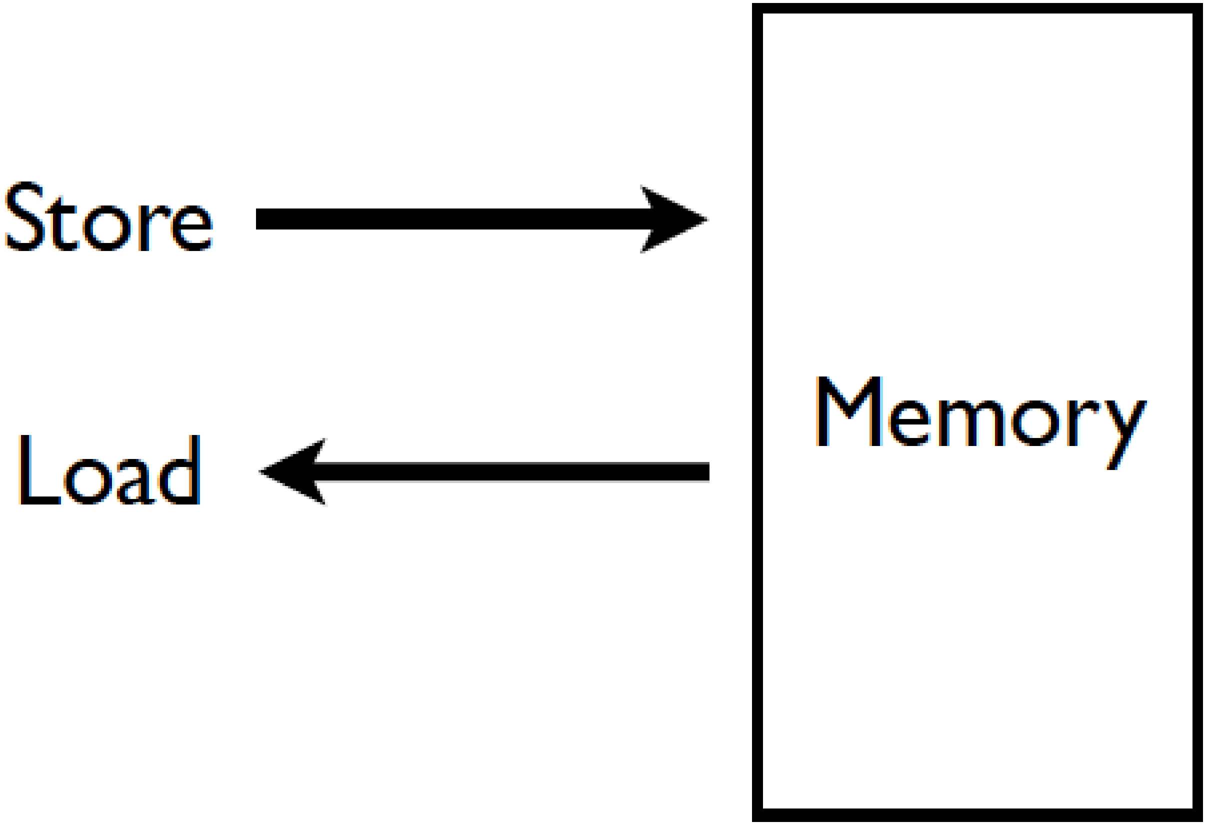
Memory Is Critical for Computing

- Performance
- Energy
- Reliability
- Security & Safety
- Cost
- Form Factor
- Quality of Service & Predictability
- …

Memory Fundamentals

Memory Organization & Technology

Memory (Programmer's View)



Abstraction: Virtual vs. Physical Memory

- Programmer sees virtual memory
  - Can assume the memory is "infinite"
- Reality: Physical memory size is much smaller than what the programmer assumes
- The system (system software + hardware, cooperatively) maps virtual memory addresses to physical memory
  - The system automatically manages the physical memory space transparently to the programmer

+ Programmer does not need to know the physical size of memory nor manage it → A small physical memory can appear as a huge one to the programmer → Life is easier for the programmer

-- More complex system software and architecture

A classic example of the programmer/(micro)architect tradeoff

(Physical) Memory System

- You need a larger level of storage to manage a small amount of physical memory automatically

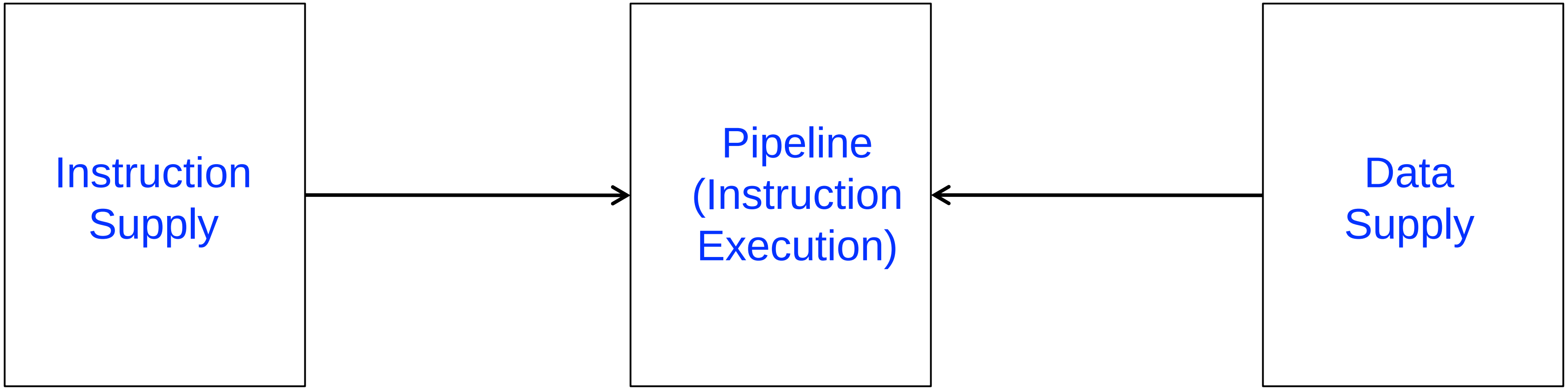
  → Physical memory has a backing store: disk

- We will first start with the physical memory system

- For now, ignore the virtual→physical indirection

- We will get back to it later, if time permits…

Idealism



- Zero latency access
- Infinite capacity
- Zero cost
- Perfect control flow

- No pipeline stalls
- Perfect data flow (no reg/memory dependences)
- Zero-cycle interconnect (operand communication)
- Enough functional units
- Zero latency compute

- Zero latency access
- Infinite capacity
- Infinite bandwidth
- Zero cost

Quick Overview of Memory Arrays

How Can We Store Data?

- Flip-Flops (or Latches)
  - Very fast, parallel access
  - Very expensive (one bit costs tens of transistors)

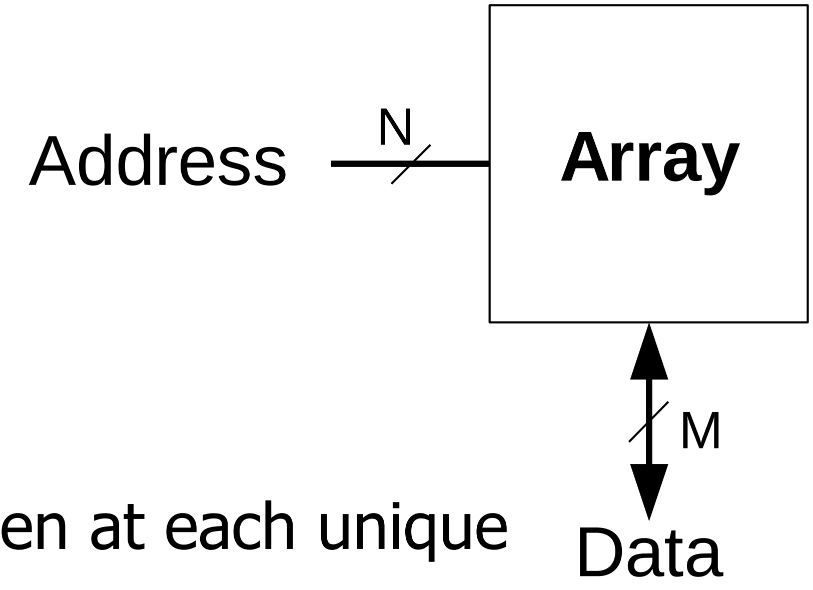
- Static RAM (we will describe them in a moment)
  - Relatively fast, only one data word at a time
  - Expensive (one bit costs 6+ transistors)

- Dynamic RAM (we will describe them in a moment)
  - Slower, one data word at a time, reading destroys content (refresh), needs special process for manufacturing
  - Cheap (one bit costs only one transistor plus one capacitor)

- Other storage technology (flash memory, hard disk, tape)
  - Much slower, access takes a long time, non-volatile
  - Very cheap (one transistor stores many bits or no transistors involved)

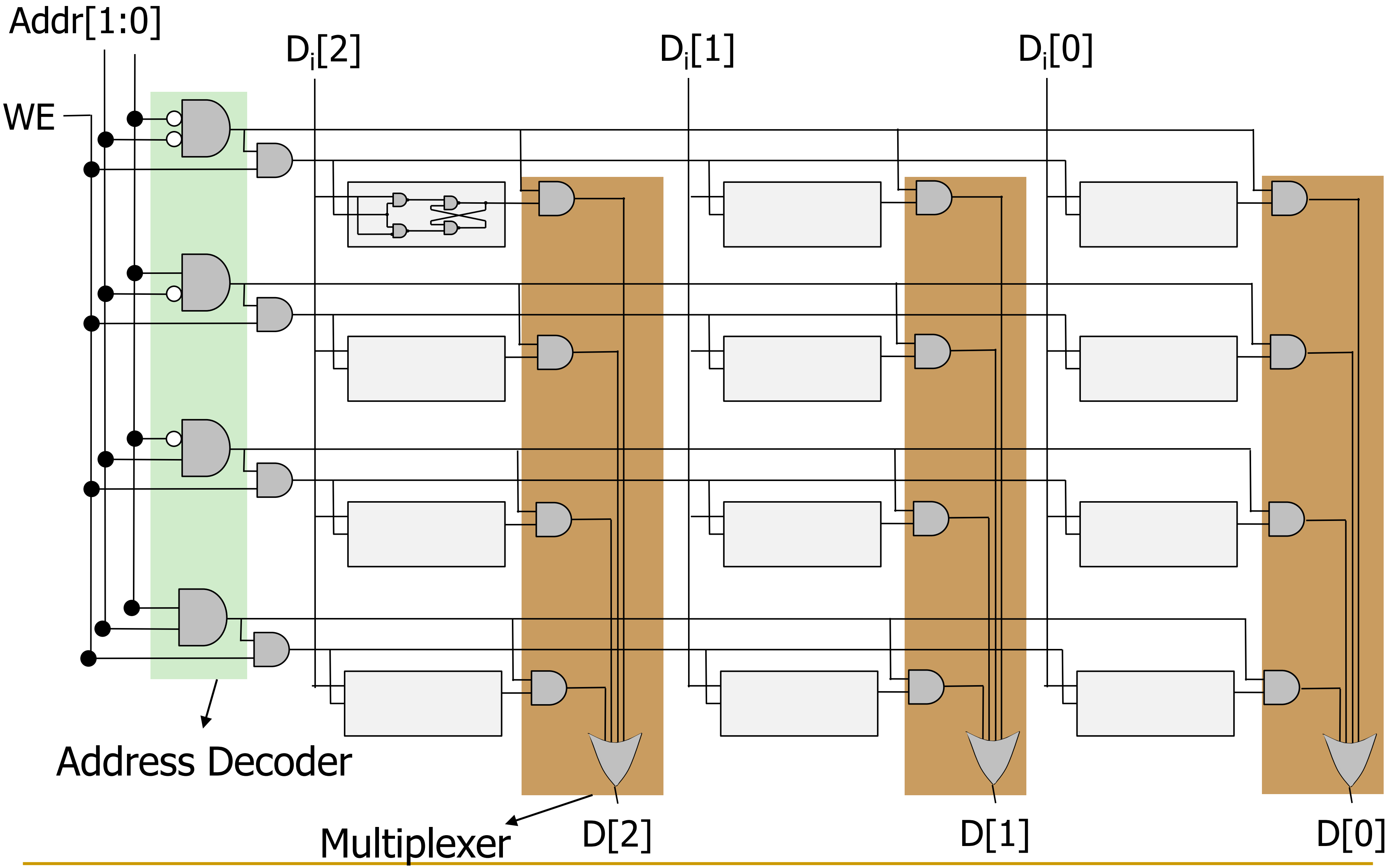
Array Organization of Memories

- Goal: Efficiently store large amounts of data
  - A memory array (stores data)
  - Address selection logic (selects one row of the array)
  - Readout circuitry (reads data out)



- An M-bit value can be read or written at each unique N-bit address
  - All values can be accessed, but only M-bits at a time
  - Access restriction allows more compact organization

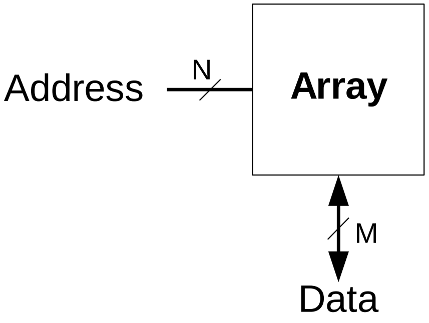
Recall: A Bigger Memory Array (4 locations X 3 bits)

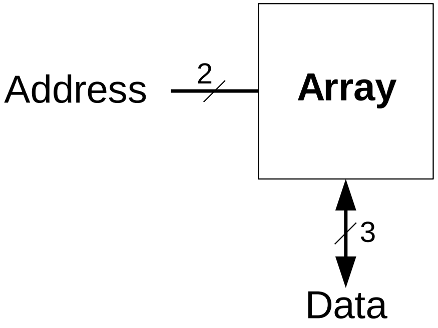


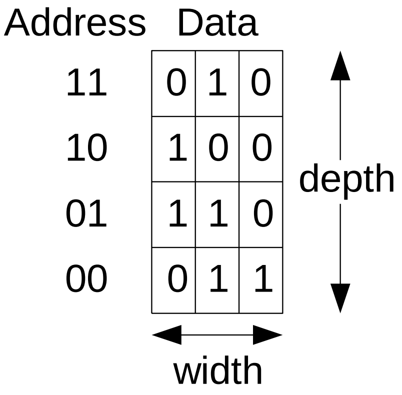
https://www.youtube.com/watch?v=QcFP4kNdKt0&list=PL5Q2soXY2Zi97Ya5DEUpMpO2bbAoaG7c6&index=6

Memory Arrays

- Two-dimensional array of bit cells
  - Each bit cell stores one bit
- An array with N address bits and M data bits:
  - 2^N rows and M columns
  - Depth: number of rows (can be number of “words”)
  - Width: number of columns (can be the “word” size)
  - Array size: depth × width = $2^N \times M$



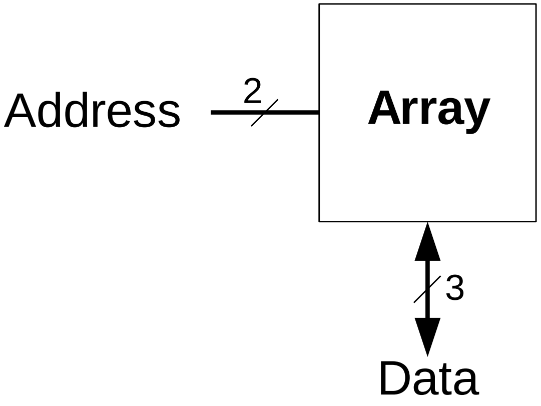


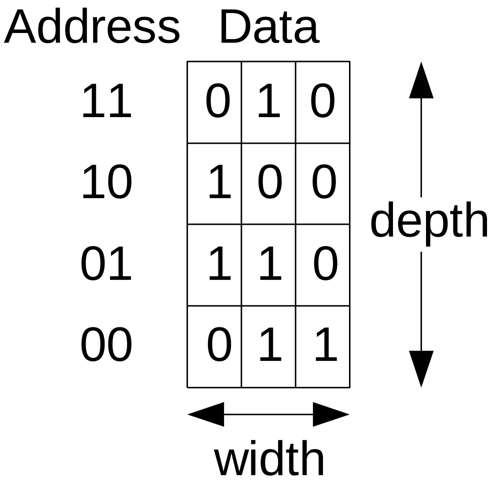


Address	Data		
11	0	1	0
10	1	0	0
01	1	1	0
00	0	1	1

Memory Array Example

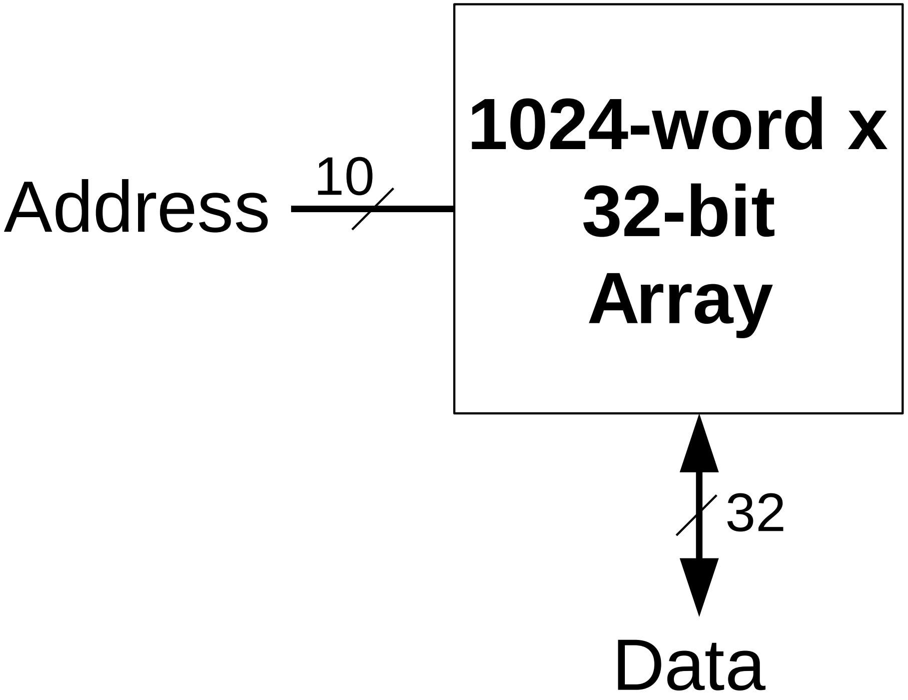
- 2^2 × 3-bit array
- Number of rows: 4
- Row size: 3 bits
- For example, the 3-bit data stored at row 10 is 100





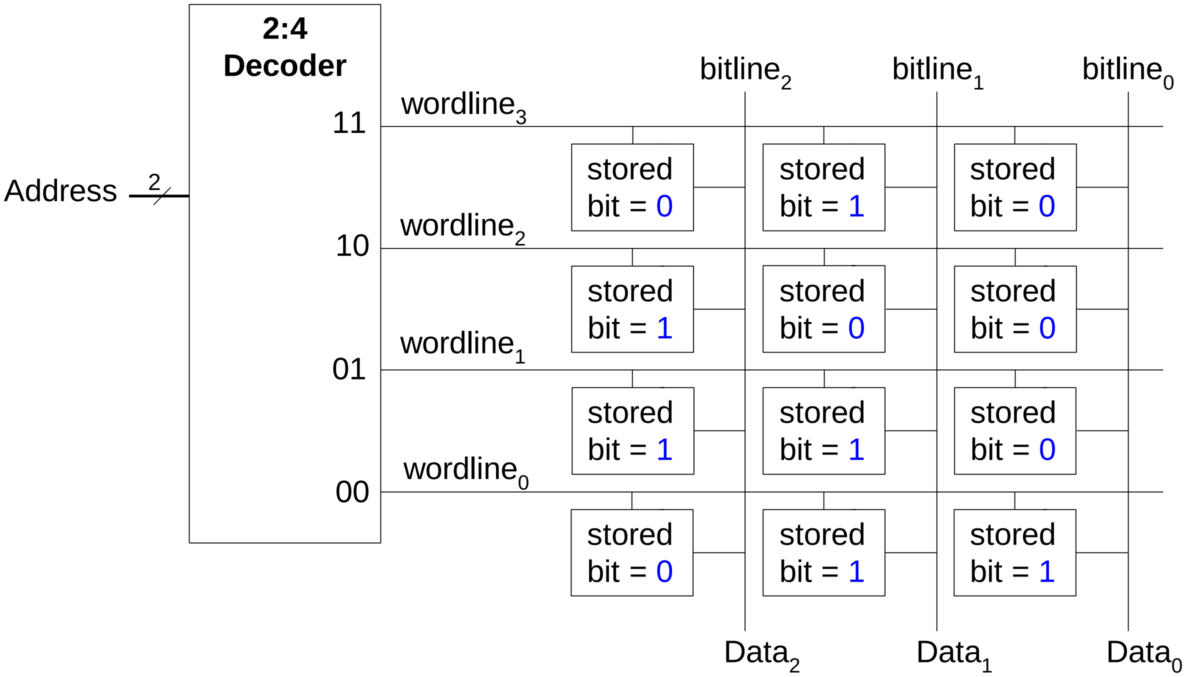
Address	Data		
11	0	1	0
10	1	0	0
01	1	1	0
00	0	1	1

Larger and Wider Memory Array Example



Memory Array Organization (I)

- Storage nodes in one column connected to one bitline
- Address decoder activates only ONE wordline
- Content of one line of storage available at output



2:4 Decoder

Address 2

11 wordline$_3$

10 wordline$_2$

01 wordline$_1$

00 wordline$_0$

bitline$_2$ bitline$_1$ bitline$_0$

stored bit = 0 | stored bit = 1 | stored bit = 0

stored bit = 1 | stored bit = 0 | stored bit = 0

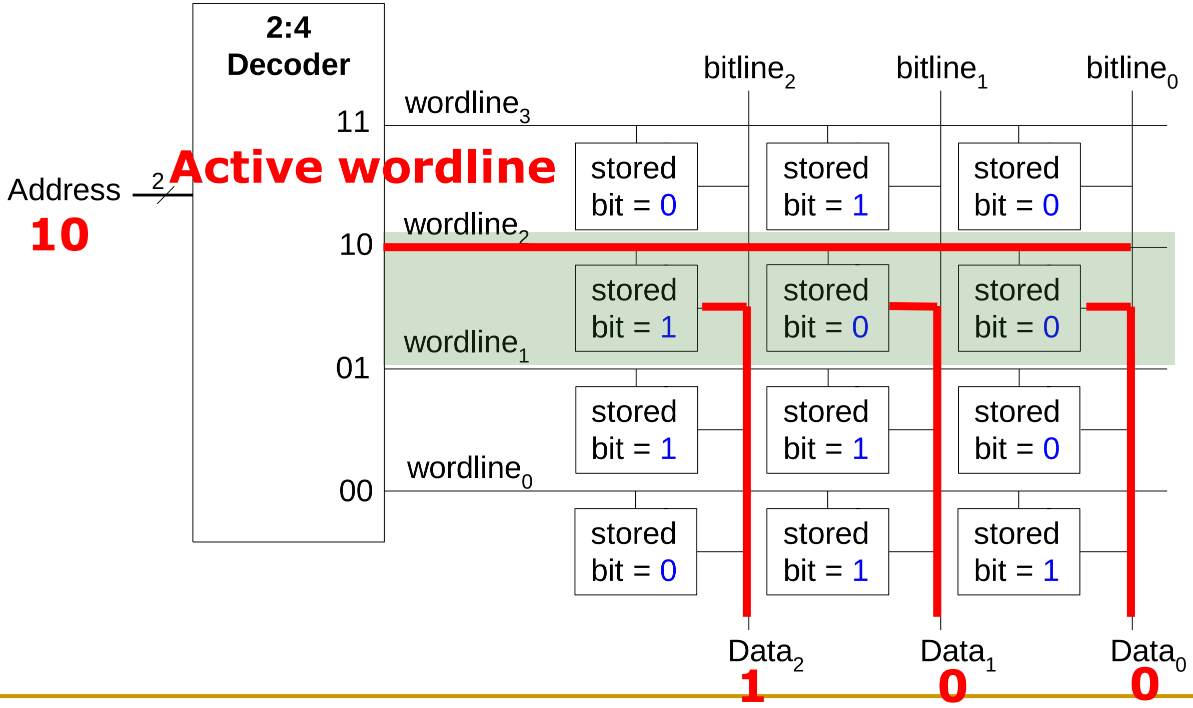
stored bit = 1 | stored bit = 1 | stored bit = 0

stored bit = 0 | stored bit = 1 | stored bit = 1

Data$_2$ Data$_1$ Data$_0$

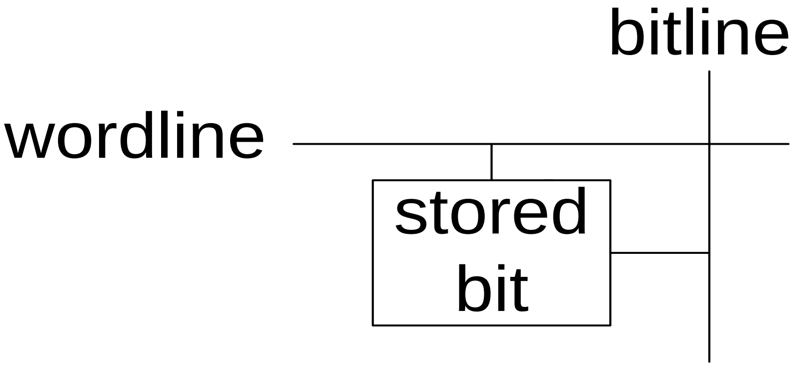
Memory Array Organization (II)

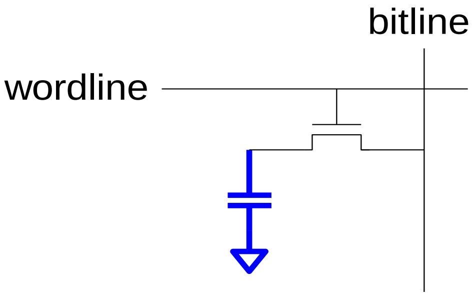
- Storage nodes in one column connected to one bitline
- Address decoder activates only ONE wordline
- Content of one line of storage available at output



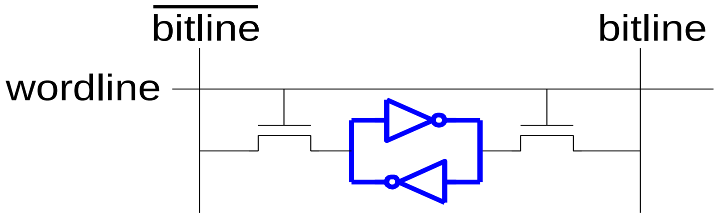
How is Access Controlled?

- Access transistors (that are configured as switches) connect the bit storage to the bitline
- Access controlled by the wordline





DRAM



SRAM

Building Larger Memories

- Requires larger memory arrays

- Large → slow

- How do we make the memory large without making it too slow?

- Idea: Divide the memory into smaller arrays and interconnect the arrays to input/output buses
  - Large memories are hierarchical array structures
  - DRAM: Channel → Rank → Bank → Subarrays → Mats

General Principle: Interleaving (Banking)

- Interleaving (banking)
  - Problem: a single monolithic large memory array takes long to access and does not enable multiple accesses in parallel

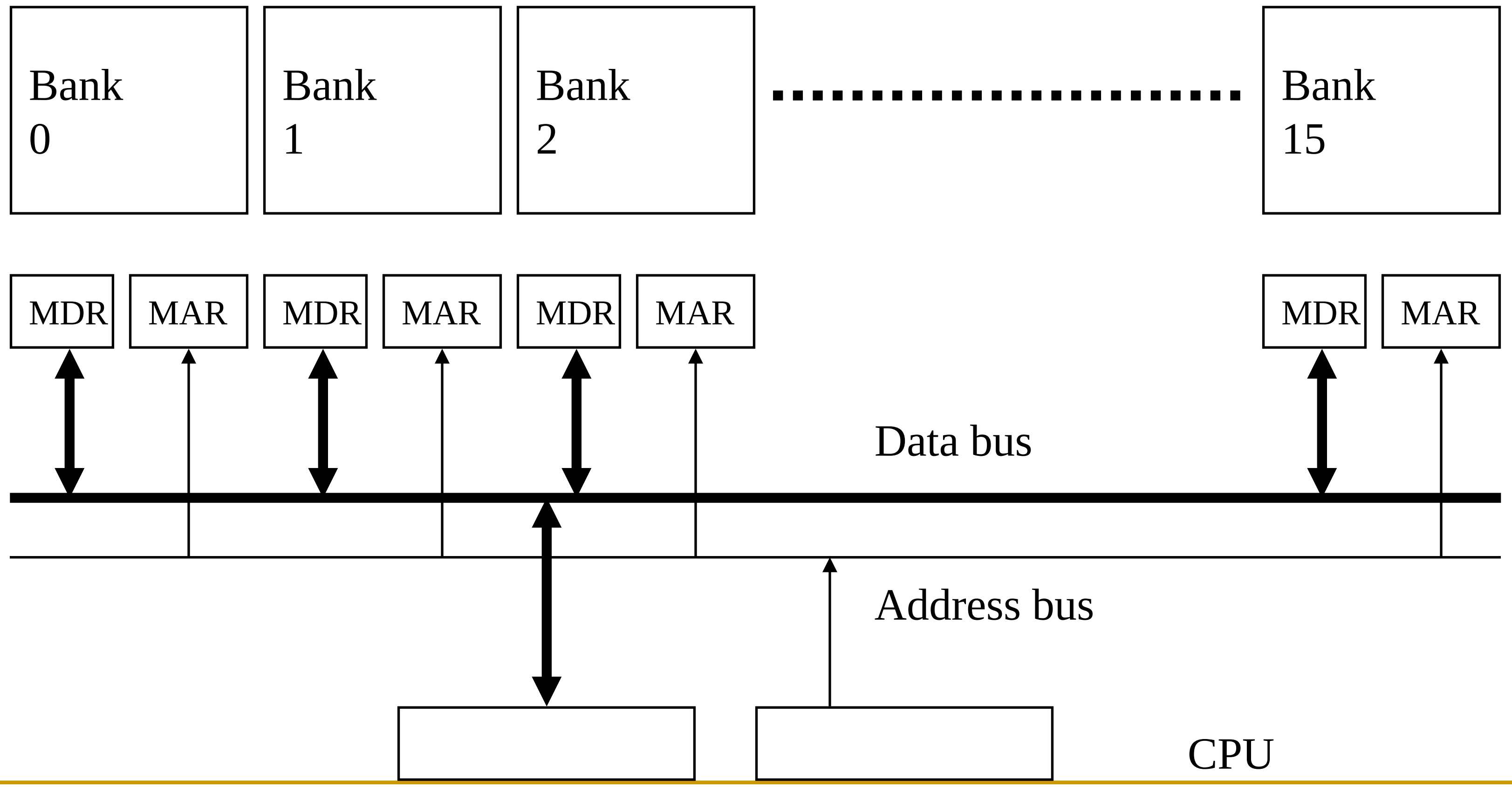
  - Goal: Reduce the latency of memory array access and enable multiple accesses in parallel

  - Idea: Divide a large array into multiple banks that can be accessed independently (in the same cycle or in consecutive cycles)
    - Each bank is smaller than the entire memory storage
    - Accesses to different banks can be overlapped

  - A Key Issue: How do you map data to different banks? (i.e., how do you interleave data across banks?)

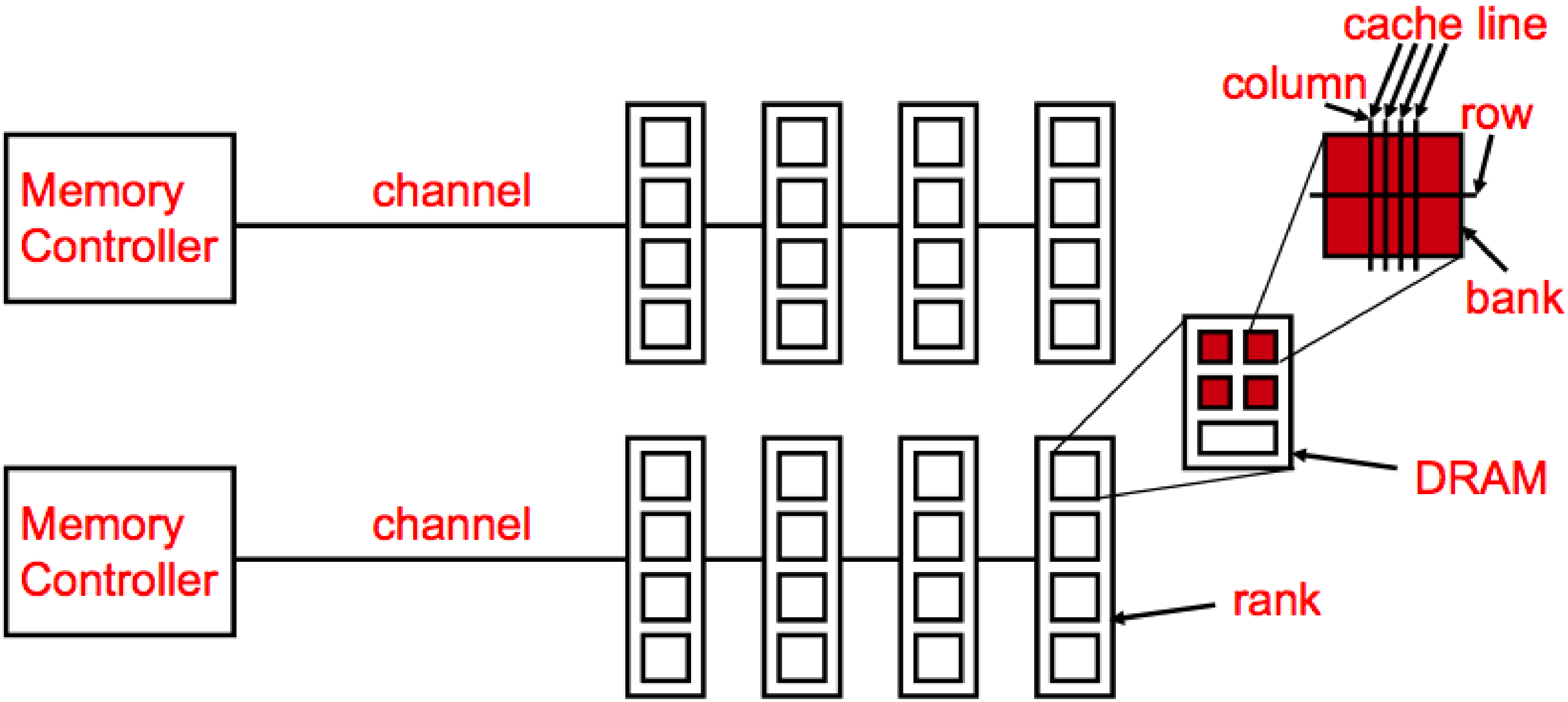
Recall: Memory Banking

- Memory is divided into banks that can be accessed independently; banks share address and data buses (to minimize pin cost)
- Can start and complete one bank access per cycle
- Can sustain N concurrent accesses if all N go to different banks

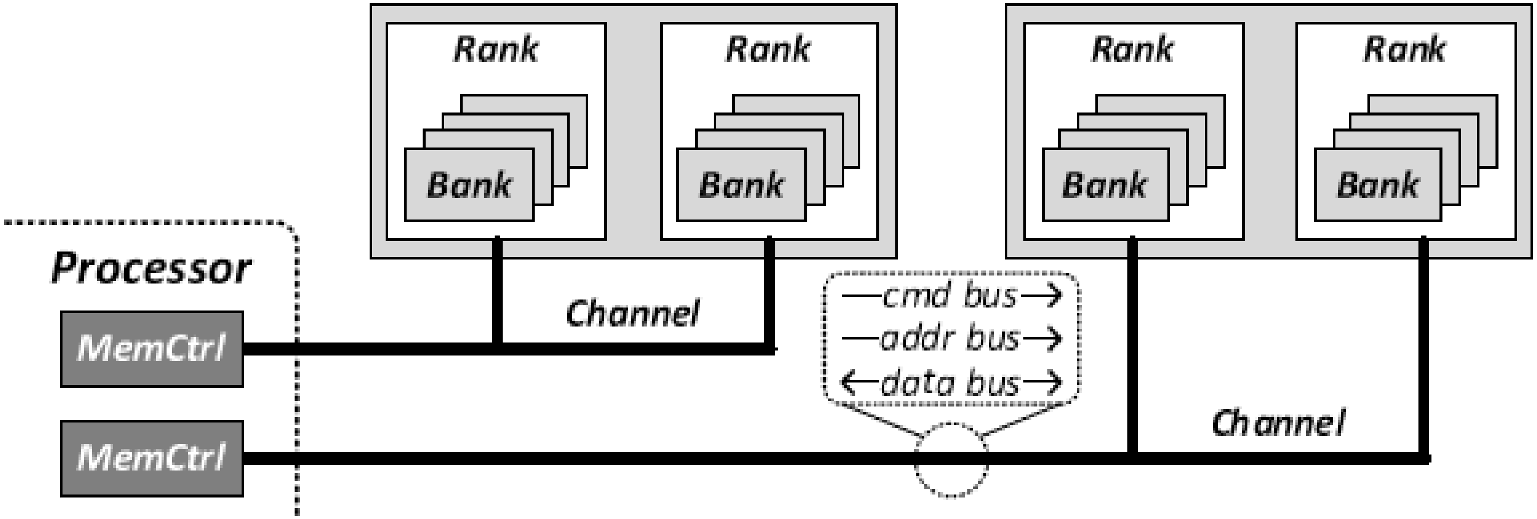


Picture credit: Derek Chiou

Generalized Memory Structure



Generalized Memory Structure

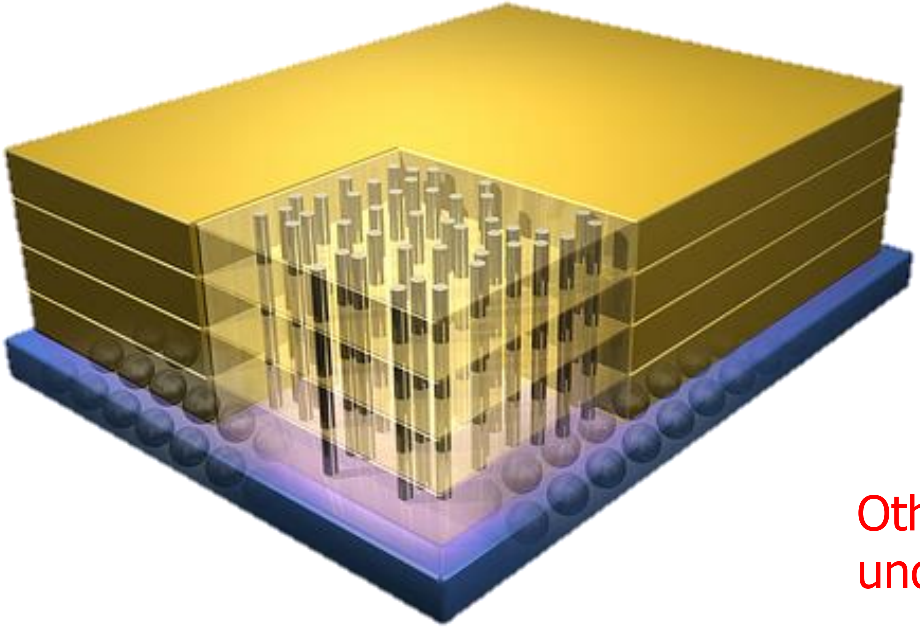


Kim+, "A Case for Exploiting Subarray-Level Parallelism in DRAM," ISCA 2012.

Lee+, "Decoupled Direct Memory Access," PACT 2015.

Cutting Edge: 3D-Stacking of Memory & Logic

Hybrid Memory Cube CONSORTIUM

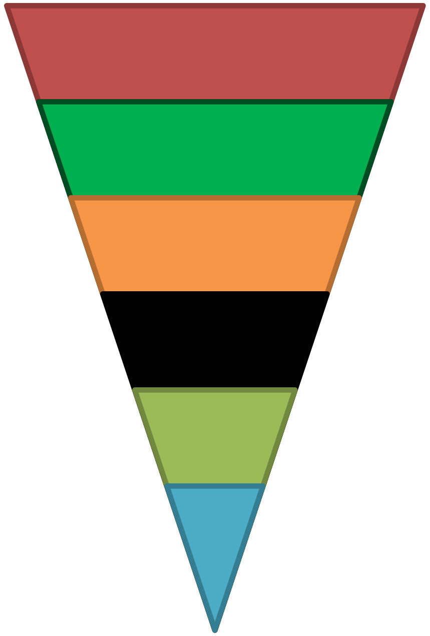


Other "True 3D" technologies under development

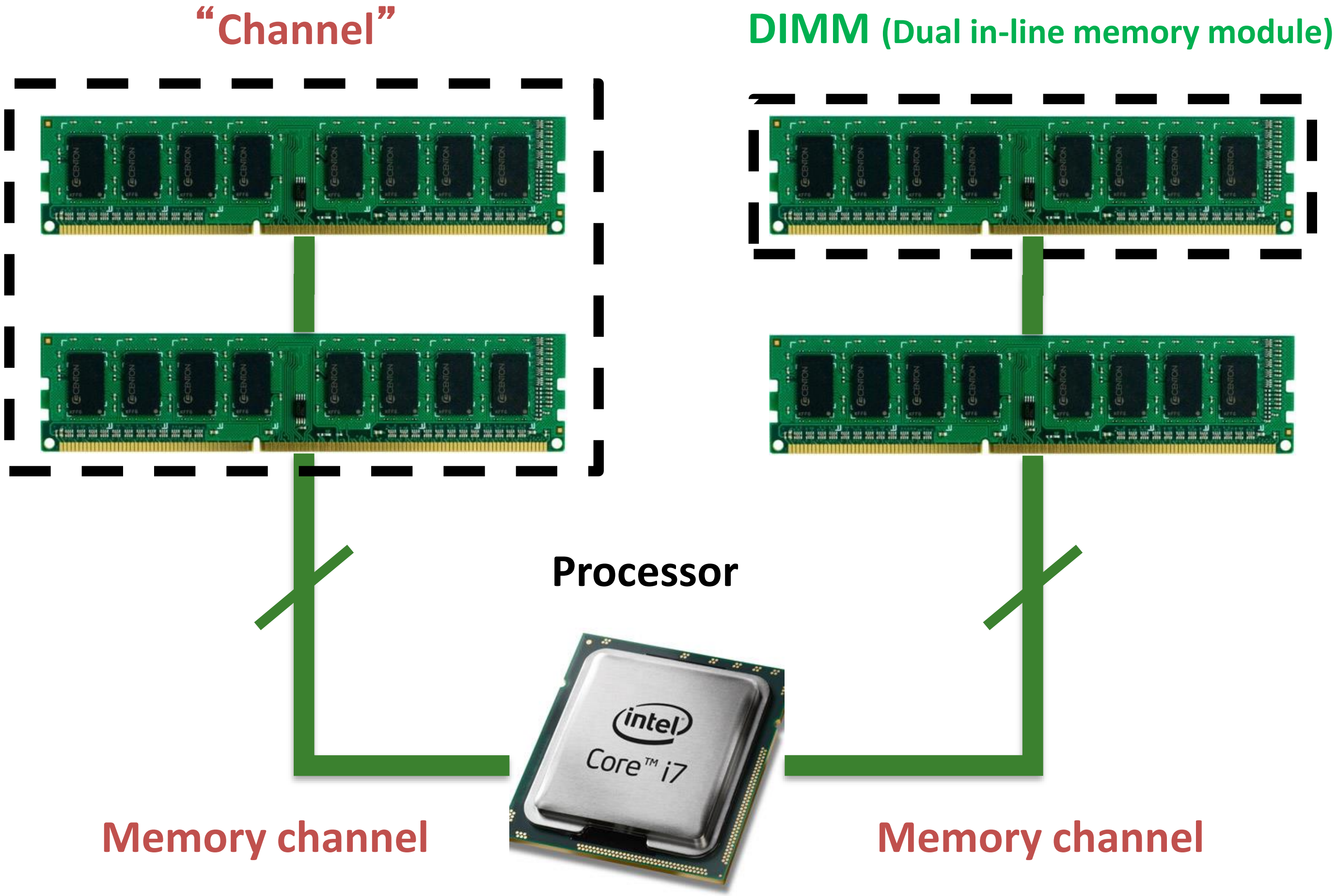
The DRAM Subsystem
A Top-Down View

DRAM Subsystem Organization

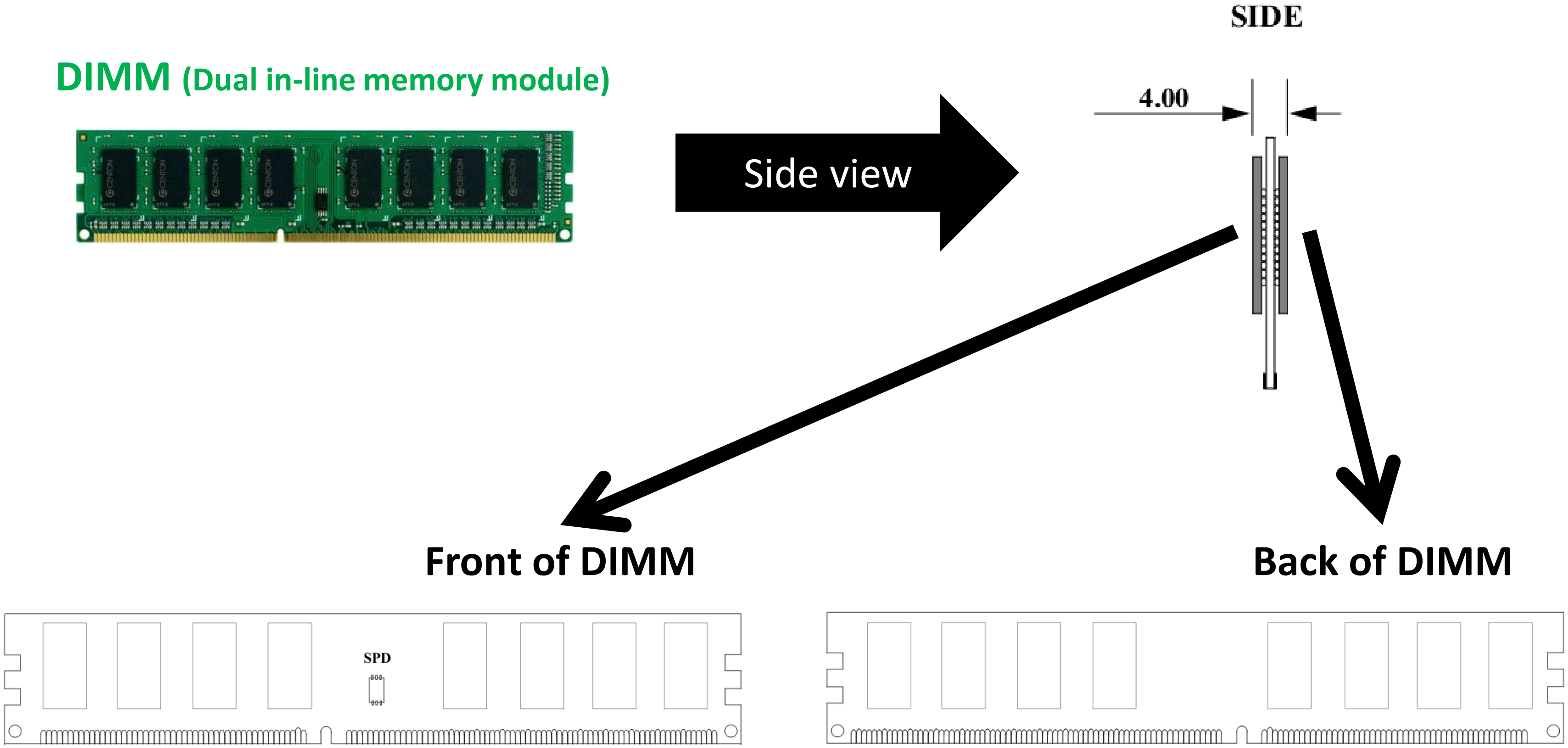
- Channel
- DIMM
- Rank
- Chip
- Bank
- Row/Column



The DRAM Subsystem



Breaking down a DIMM (module)



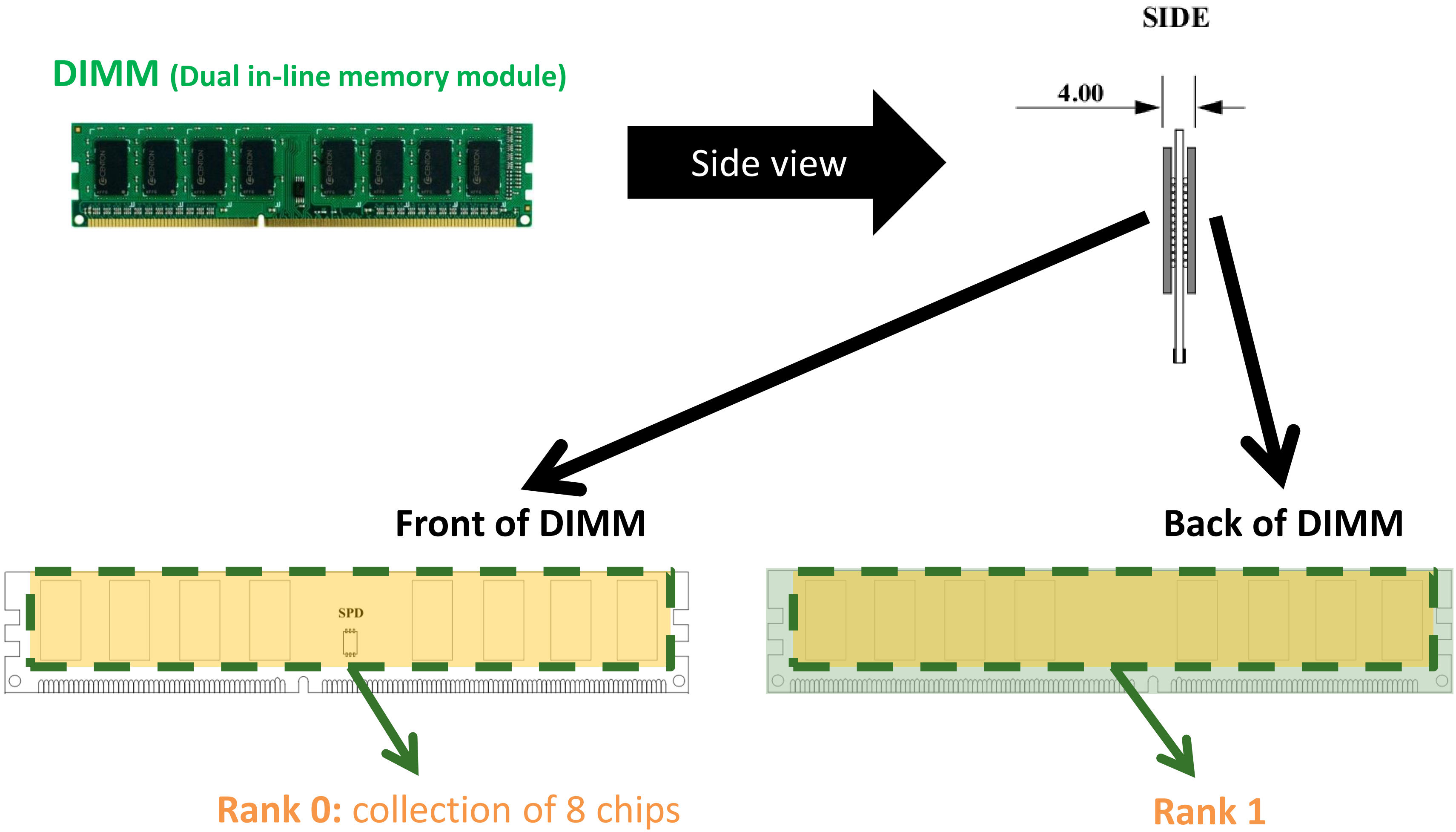
DIMM (Dual in-line memory module)

Side view

Front of DIMM

Back of DIMM

Breaking down a DIMM (module)



DIMM (Dual in-line memory module)

Side view

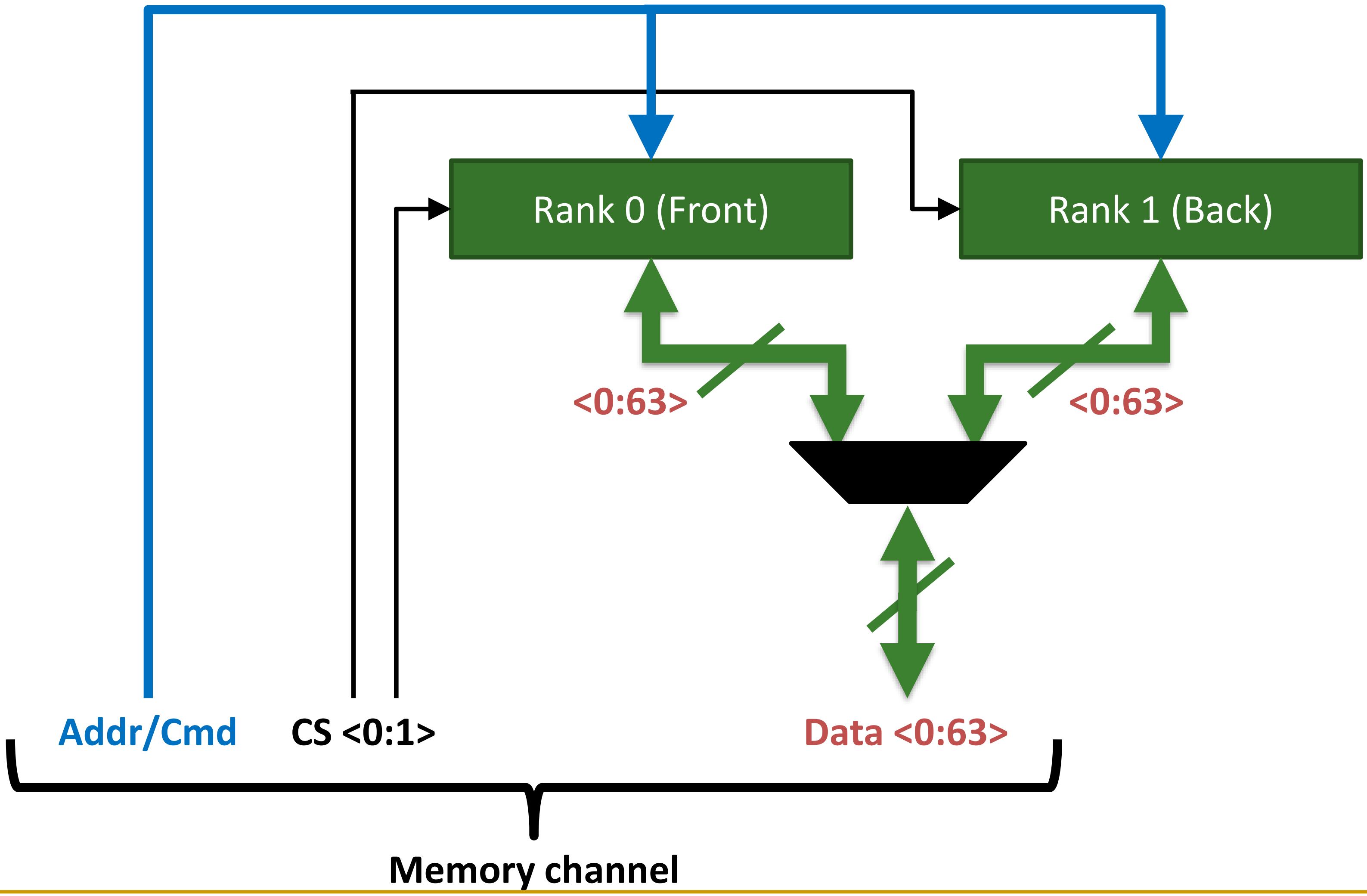
Front of DIMM

Back of DIMM

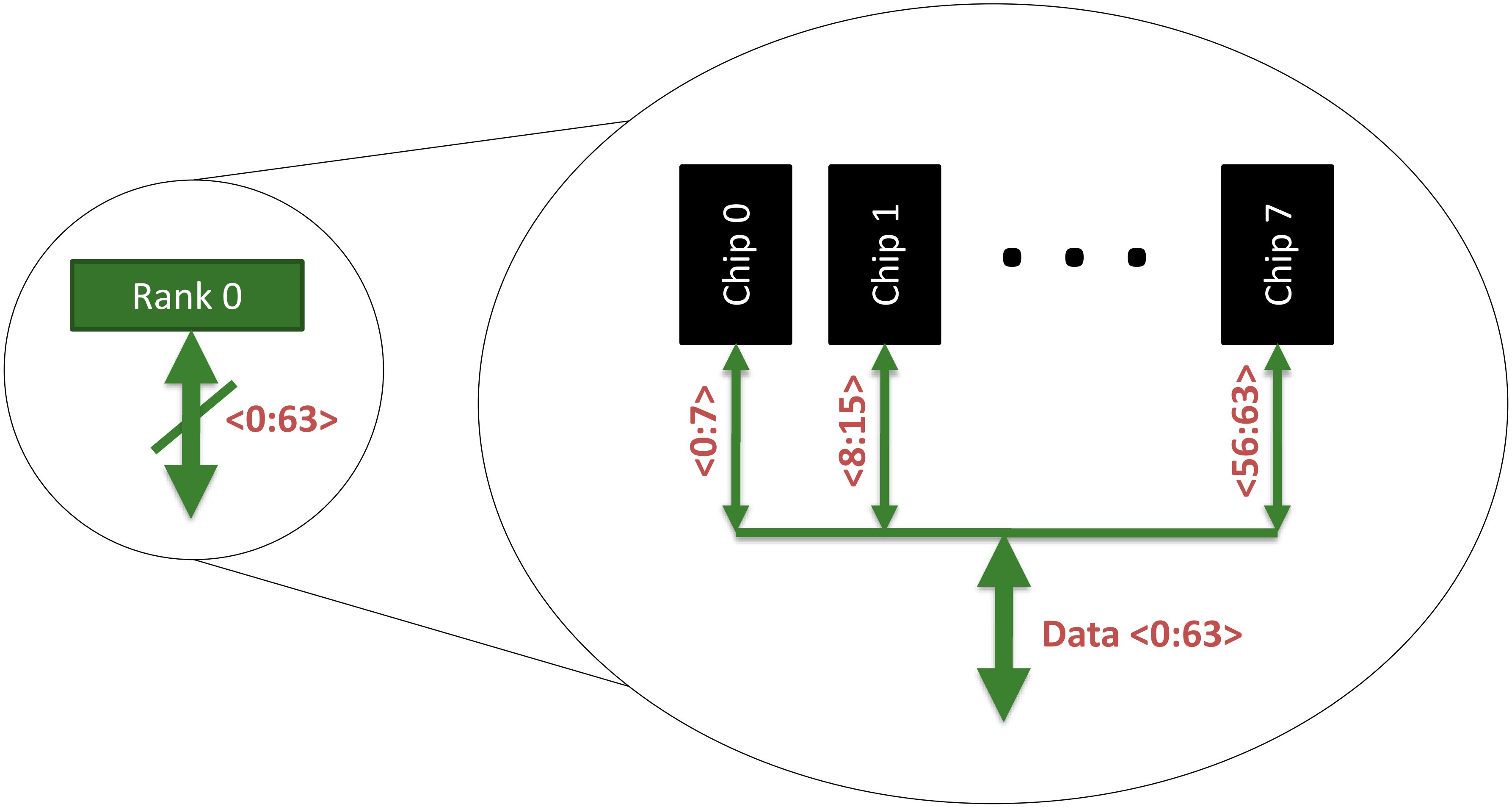
Rank 0: collection of 8 chips

Rank 1

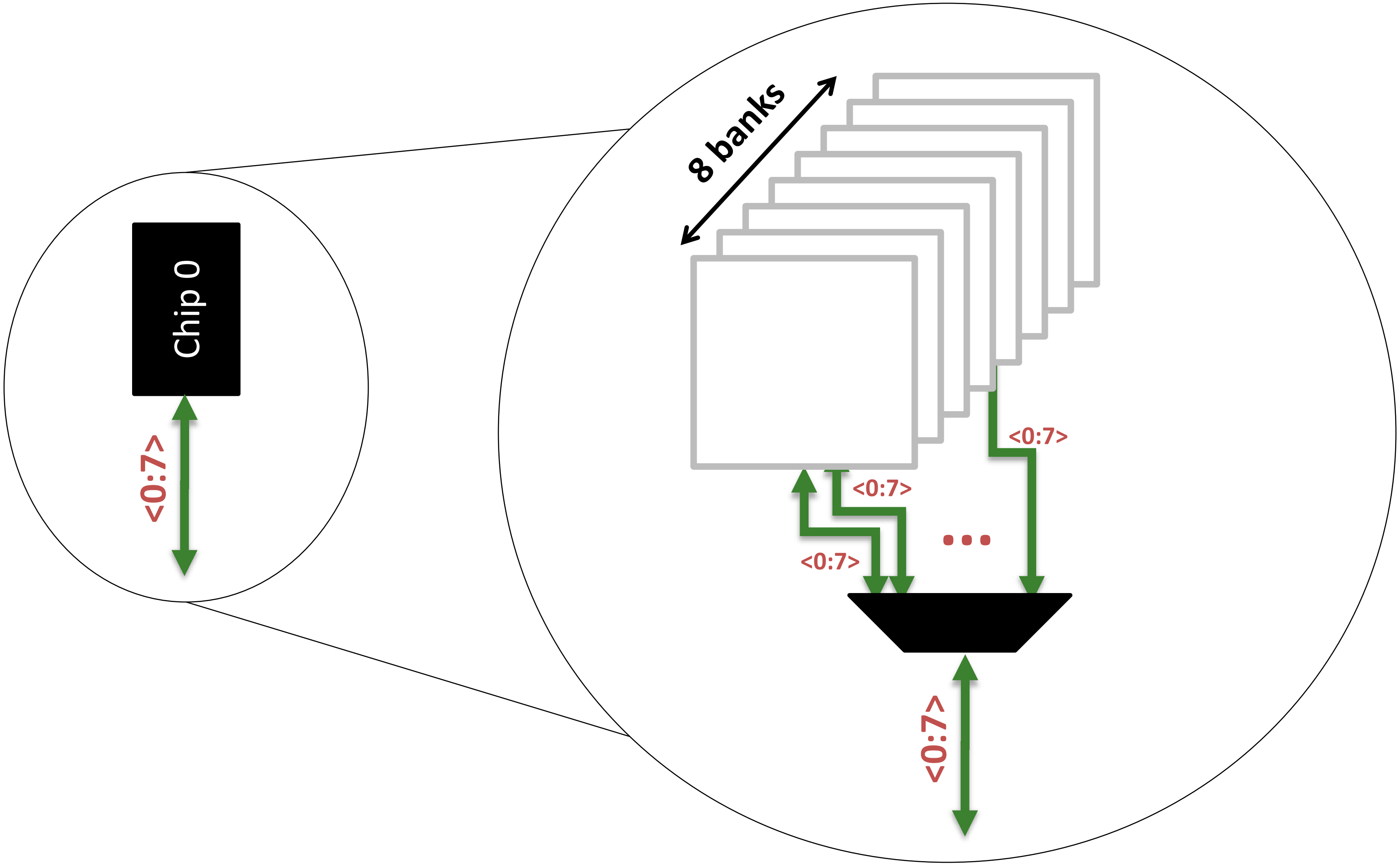
Rank


Rank 0 (Front)
Rank 1 (Back)
<0:63>
<0:63>
Addr/Cmd
CS <0:1>
Data <0:63>
Memory channel

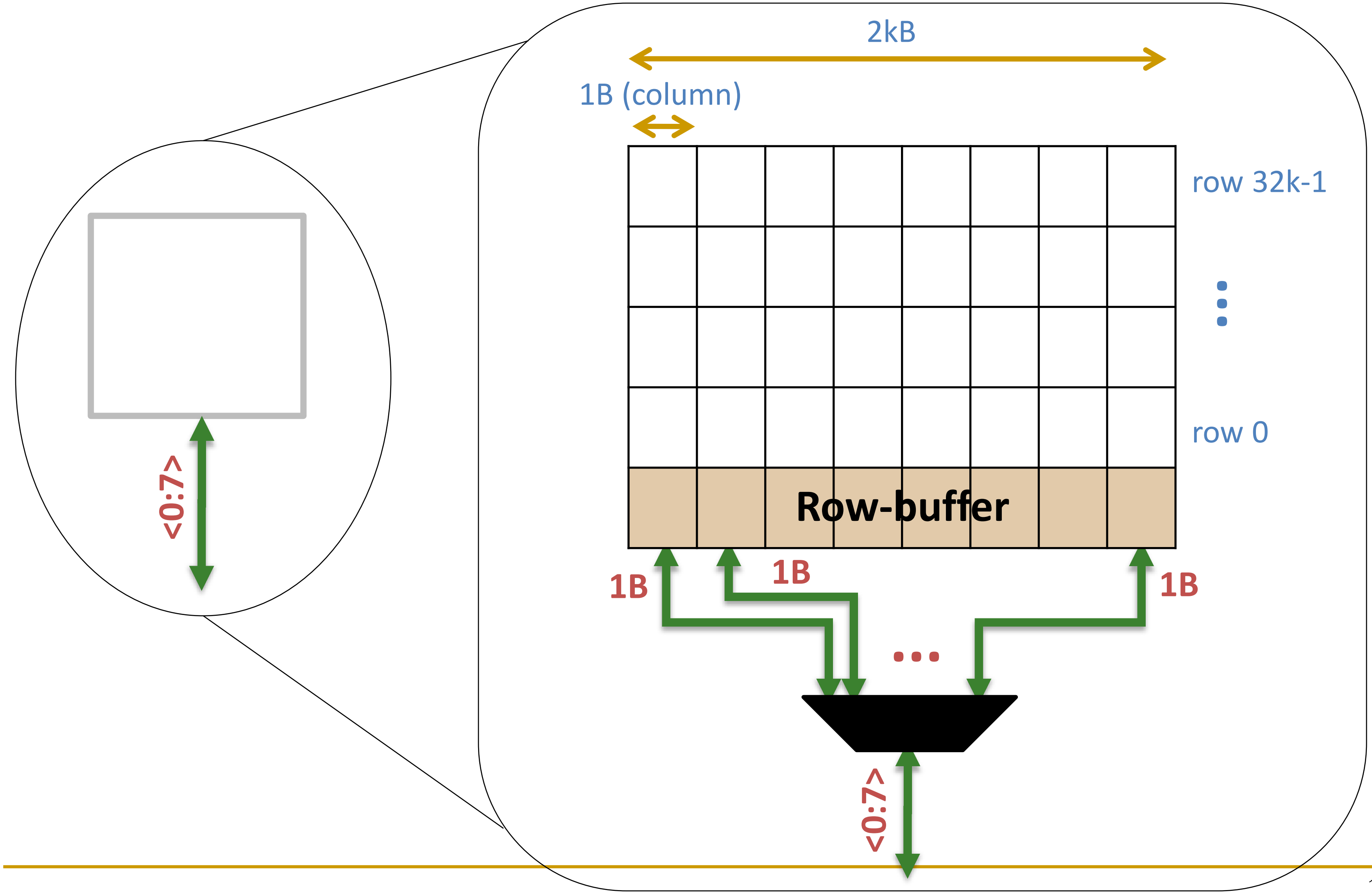
Breaking down a Rank



Breaking down a Chip

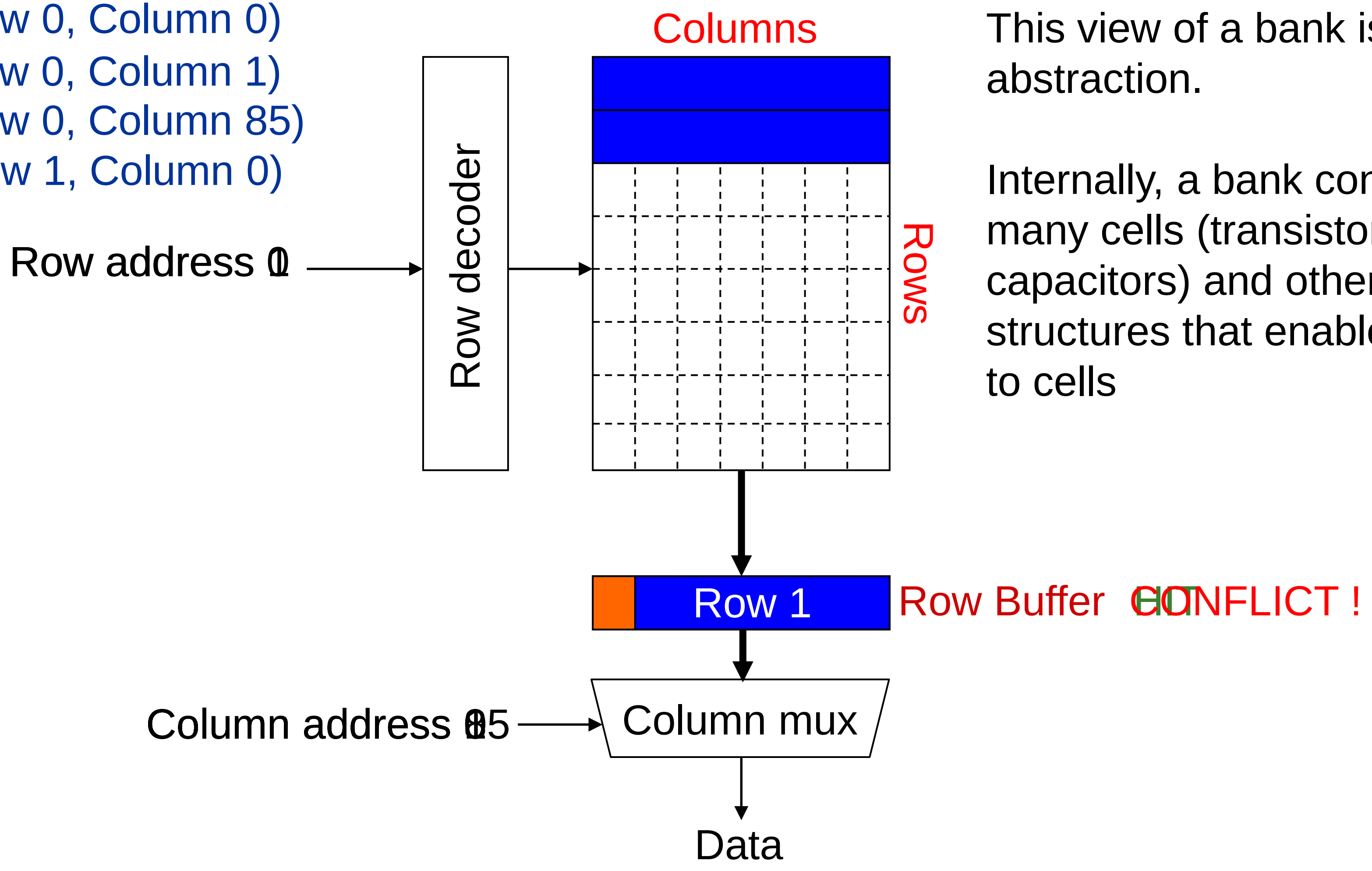


Breaking down a Bank



Digging Deeper: DRAM Bank Operation

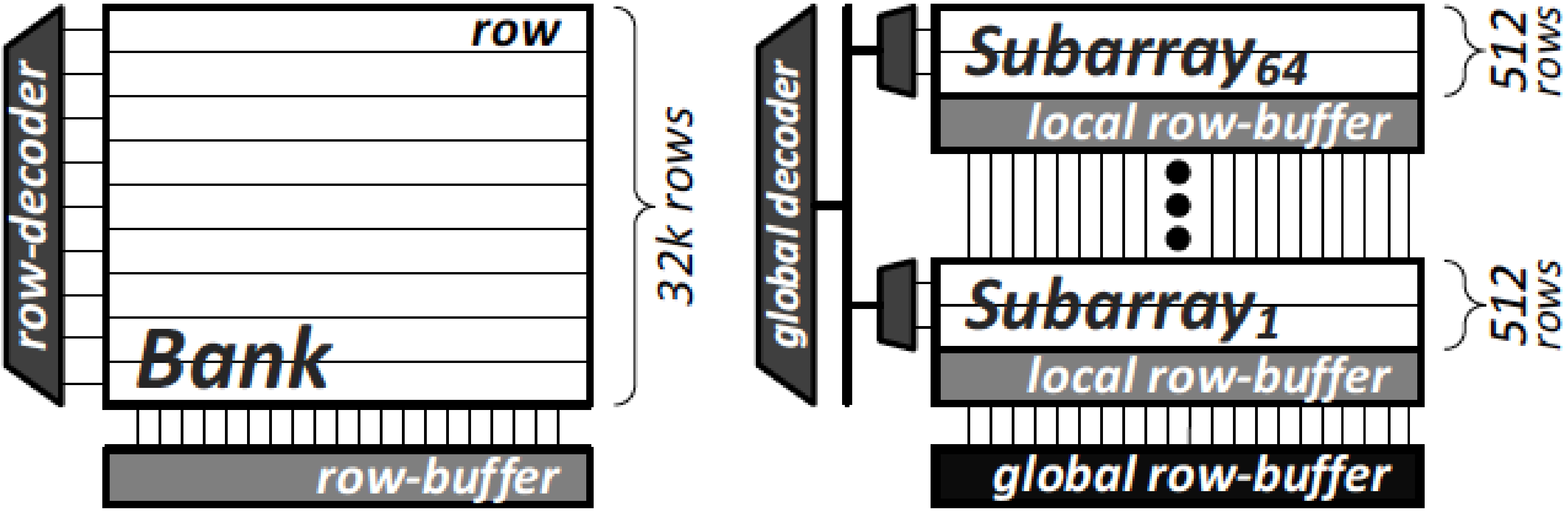
Access Address:
(Row 0, Column 0)
(Row 0, Column 1)
(Row 0, Column 85)
(Row 1, Column 0)



This view of a bank is an abstraction.

Internally, a bank consists of many cells (transistors & capacitors) and other structures that enable access to cells

A DRAM Bank Internally Has Sub-Banks

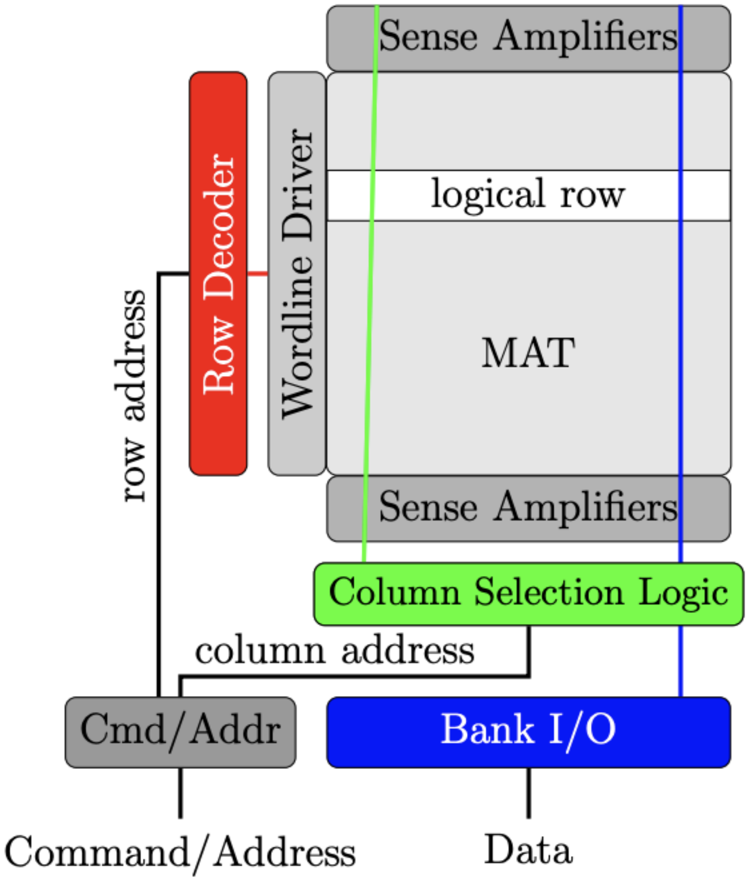


(a) Logical abstraction (b) Physical implementation

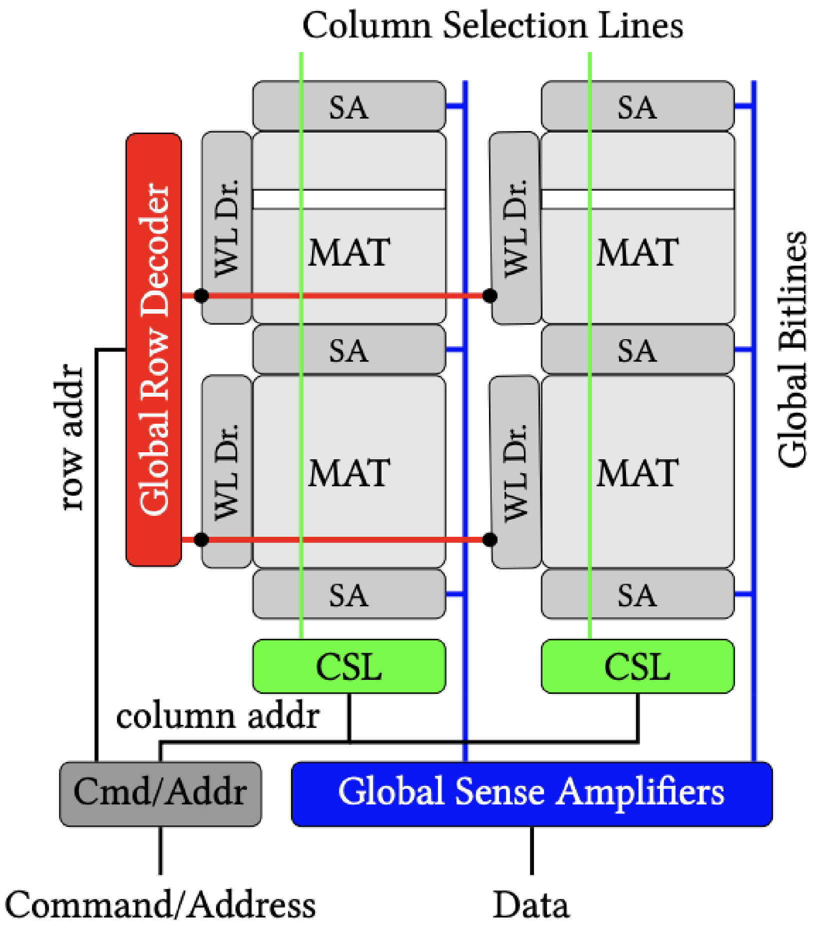
Figure 1. DRAM bank organization

Kim et al., "A Case for Exploiting Subarray-Level Parallelism in DRAM," ISCA 2012.

Another View of a DRAM Bank



Logical Abstraction



Physical View

Seshadri+, "In-DRAM Bulk Bitwise Execution Engine," ADCOM 2020.

More on DRAM Basics & Organization

- Vivek Seshadri and Onur Mutlu, **"In-DRAM Bulk Bitwise Execution Engine"** *Invited Book Chapter in Advances in Computers*, 2020. [Preliminary arXiv version]

See Section 2 for comprehensive DRAM Background

In-DRAM Bulk Bitwise Execution Engine

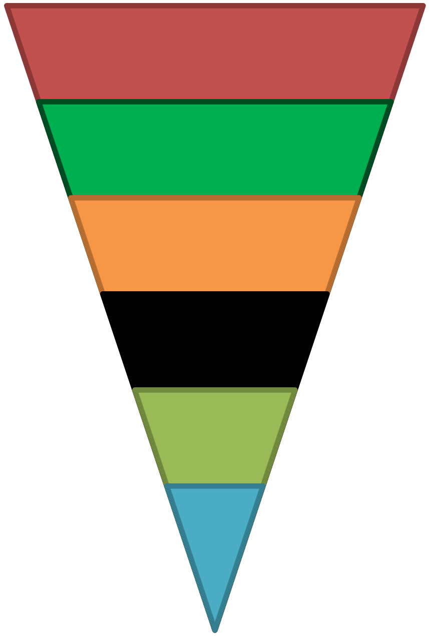
Vivek Seshadri
Microsoft Research India
visesha@microsoft.com

Onur Mutlu
ETH Zürich
onur.mutlu@inf.ethz.ch

https://arxiv.org/pdf/1905.09822.pdf

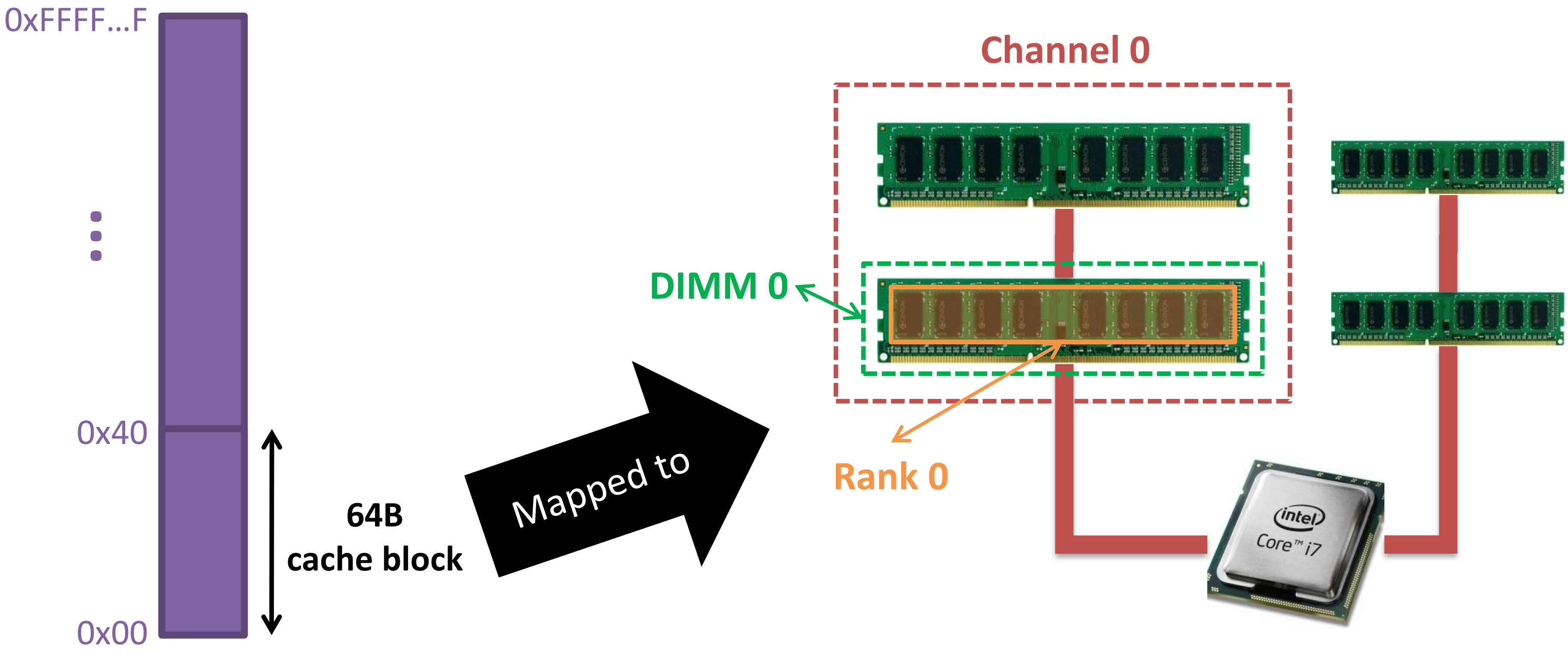
DRAM Subsystem Organization

- Channel
- DIMM
- Rank
- Chip
- Bank
- Row/Column

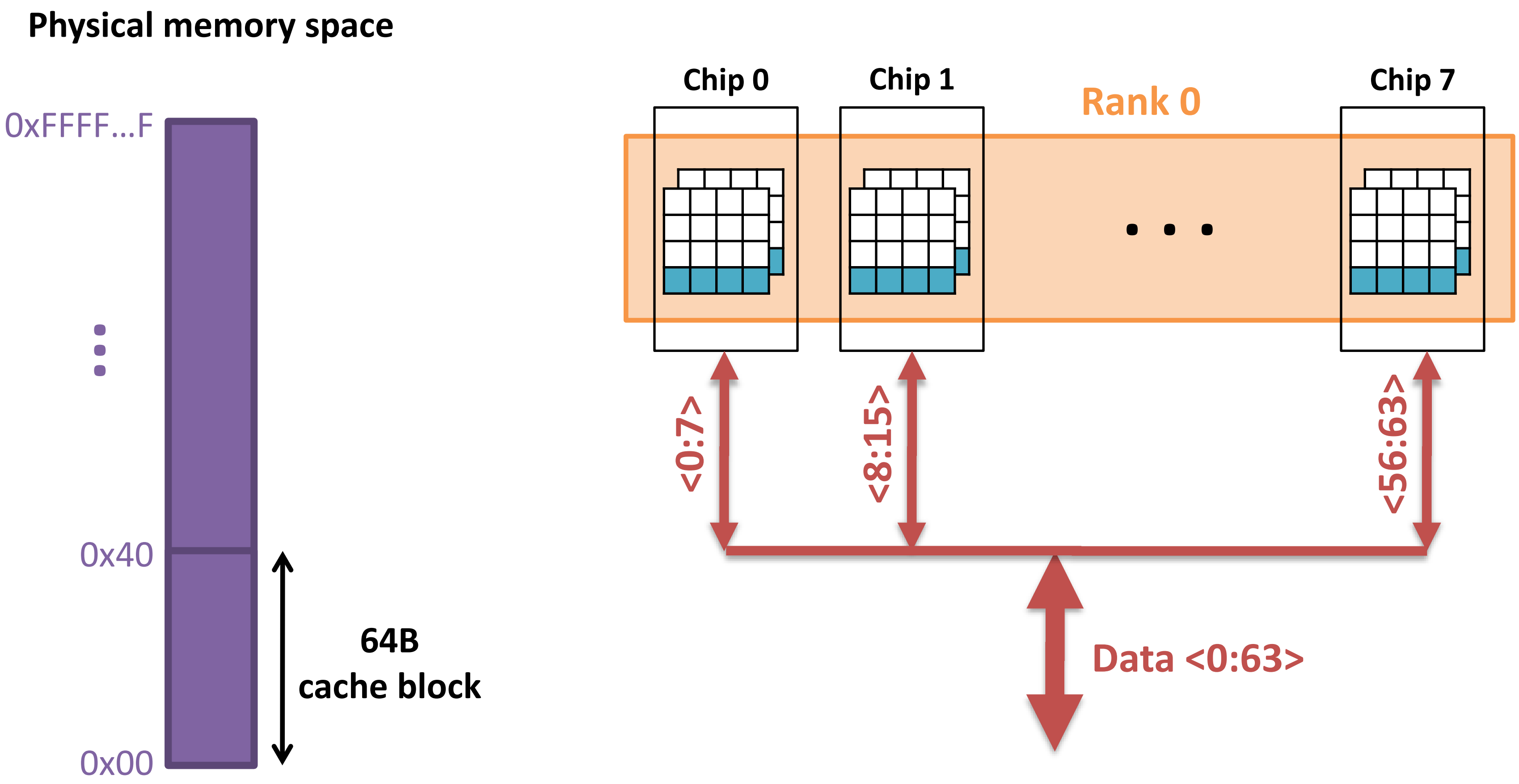


Example: Transferring a cache block

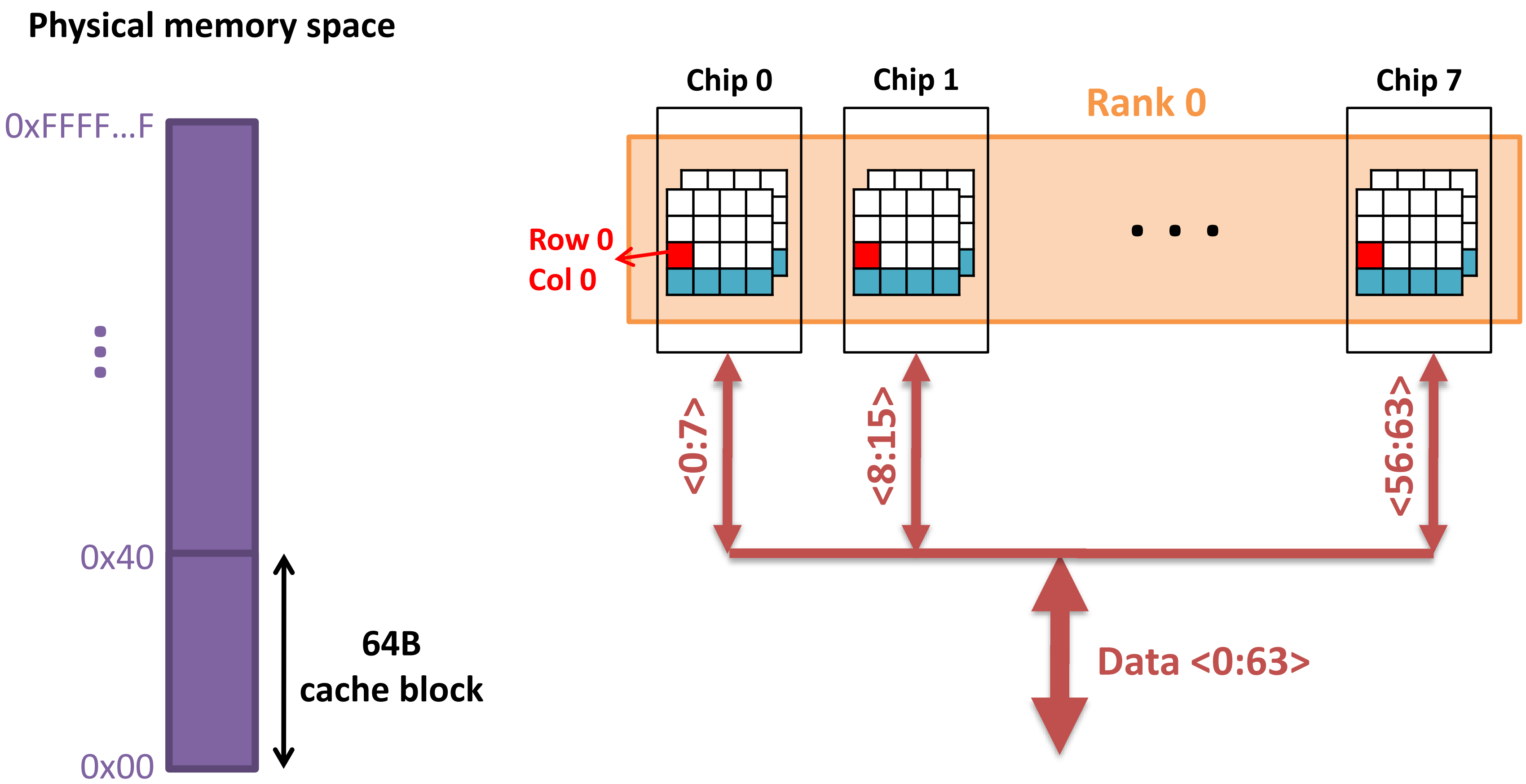
Physical memory space



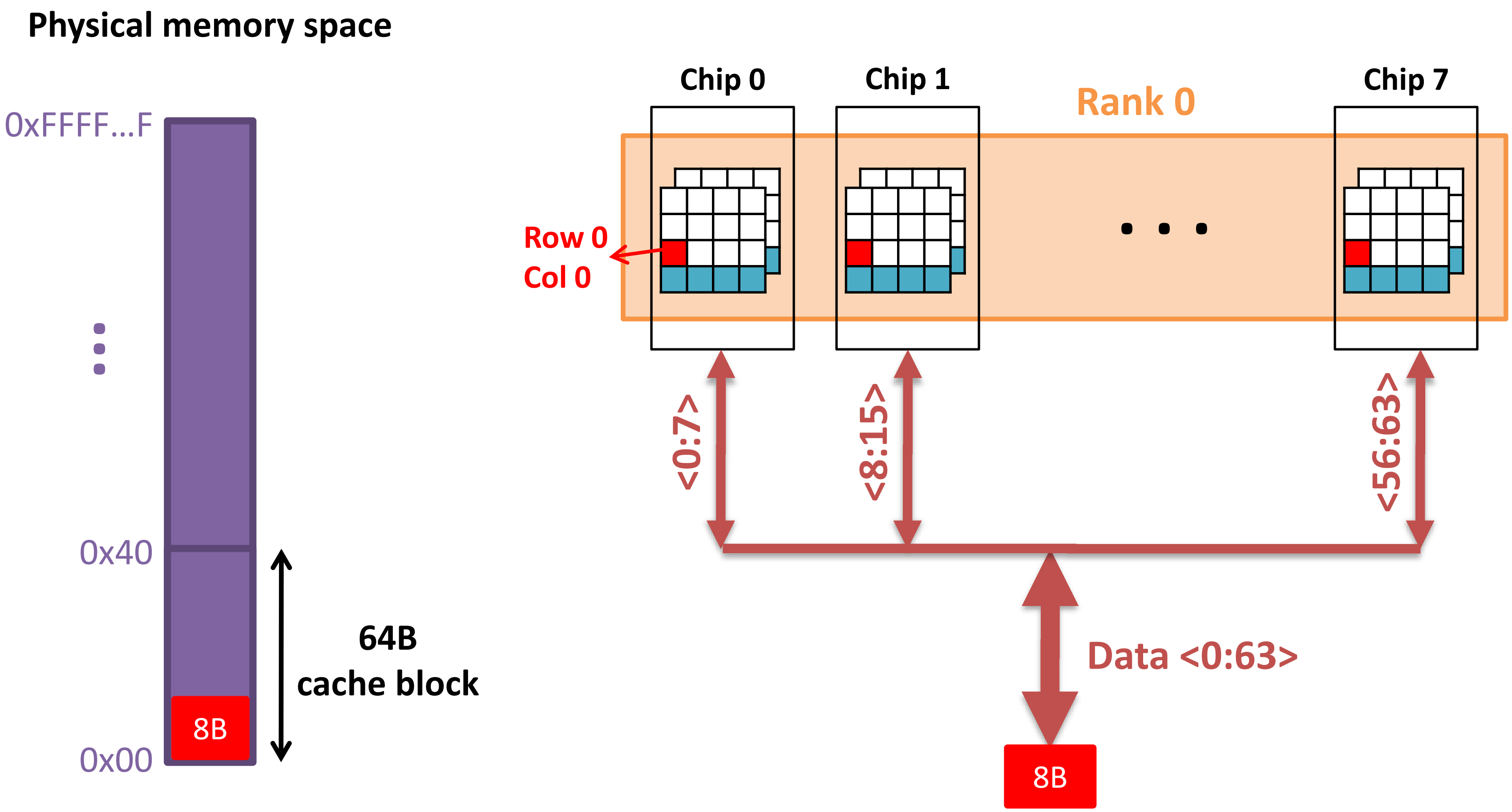
Example: Transferring a cache block



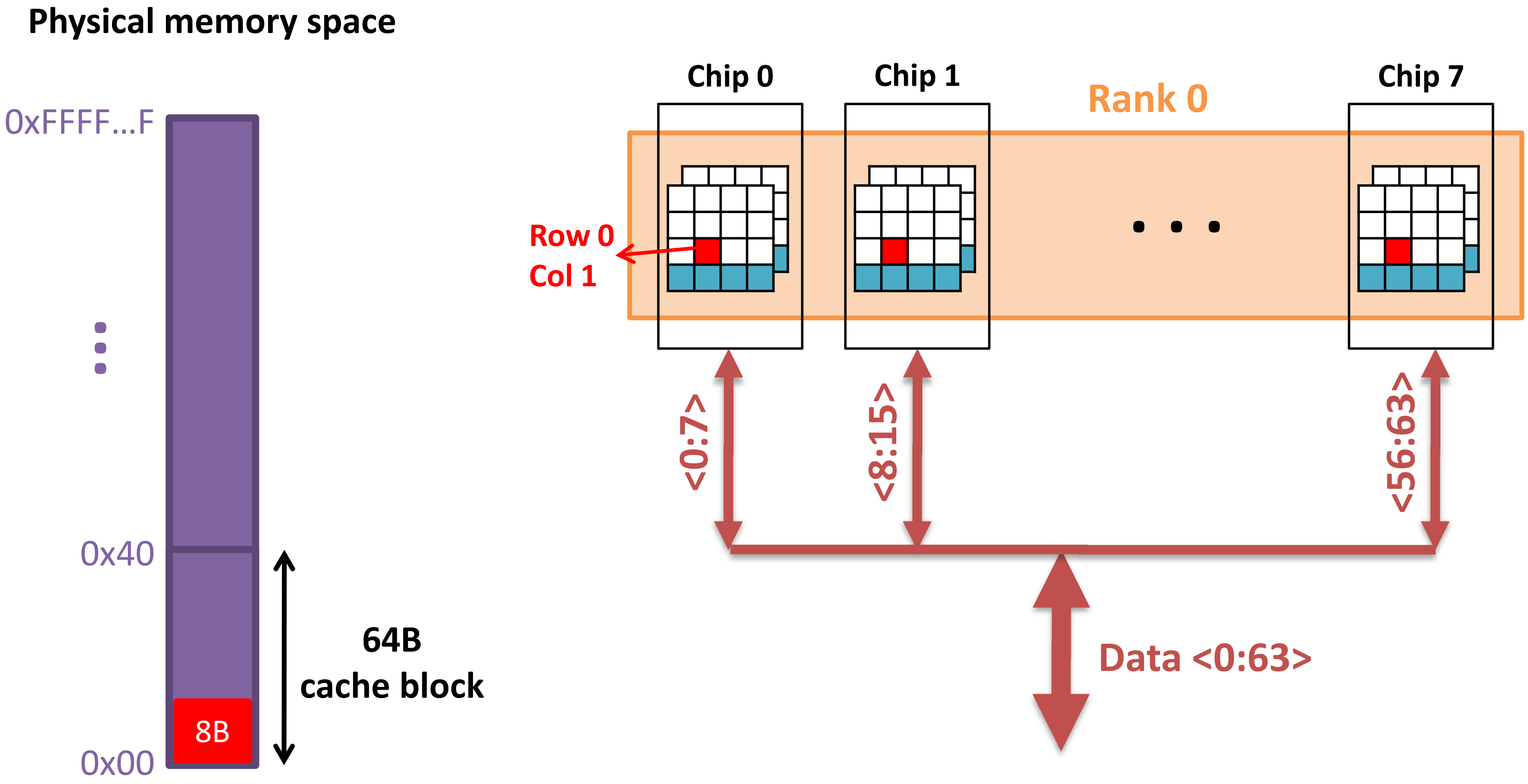
Example: Transferring a cache block



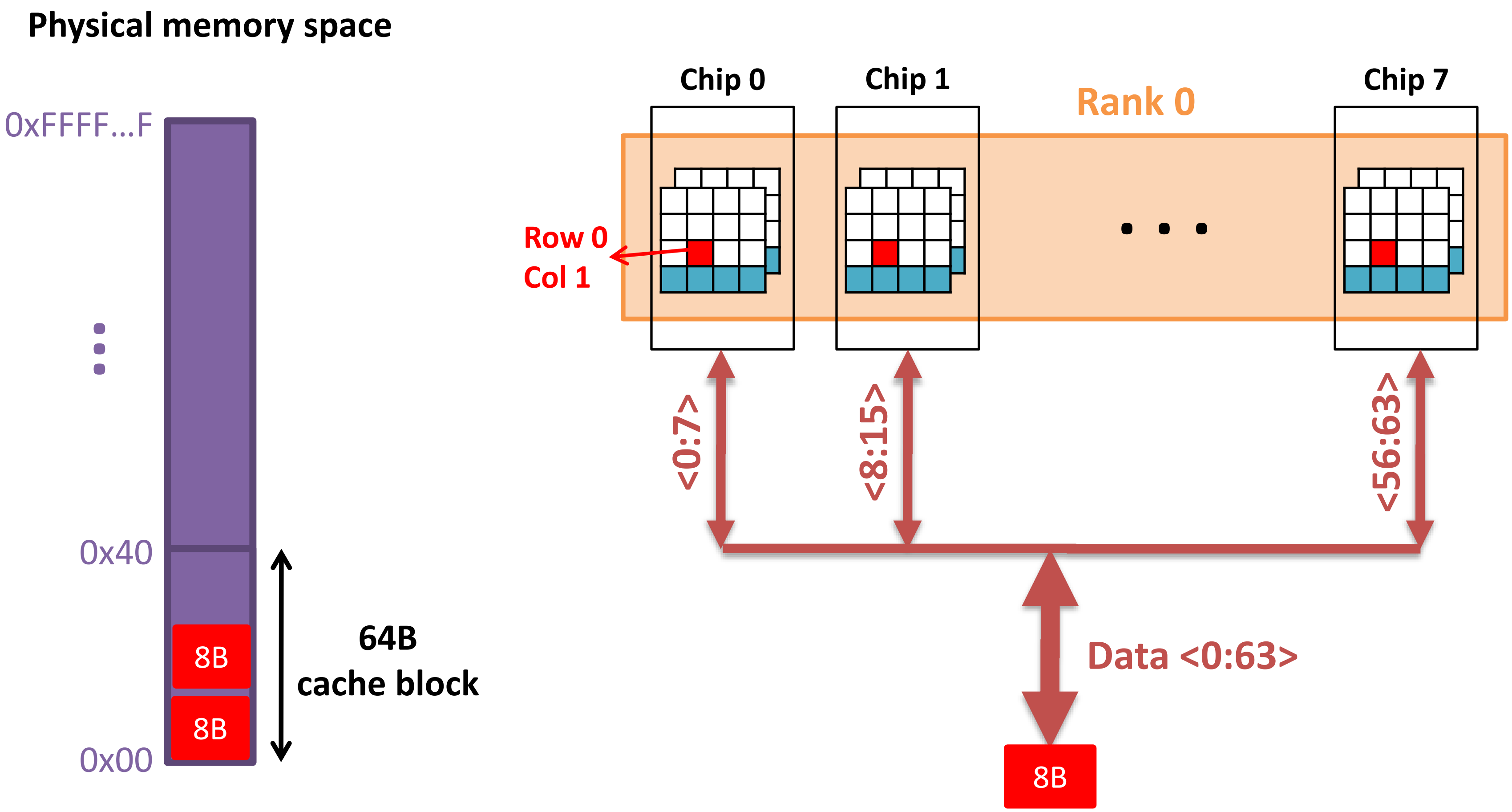
Example: Transferring a cache block



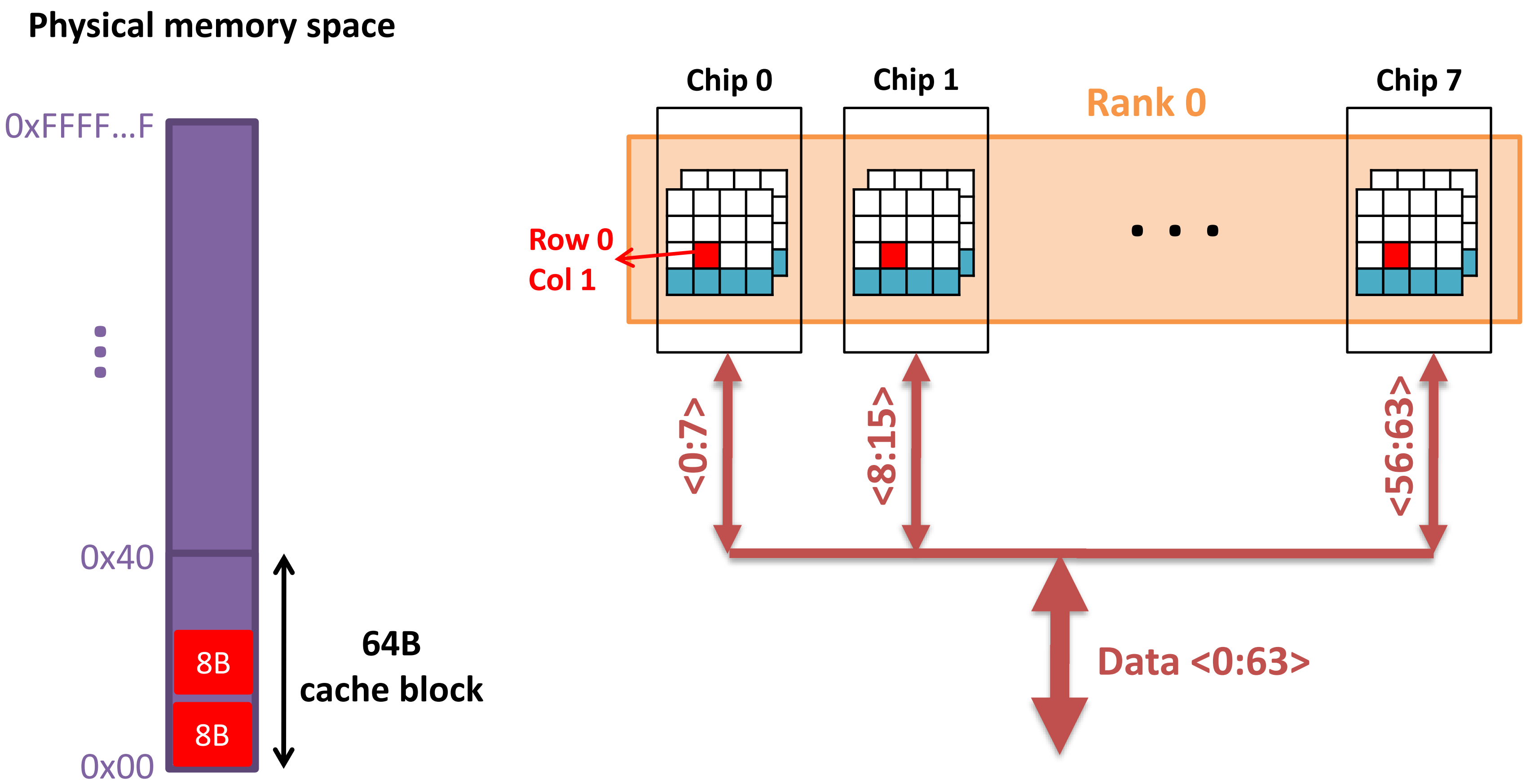
Example: Transferring a cache block



Example: Transferring a cache block



Example: Transferring a cache block



A 64B cache block takes 8 I/O cycles to transfer.

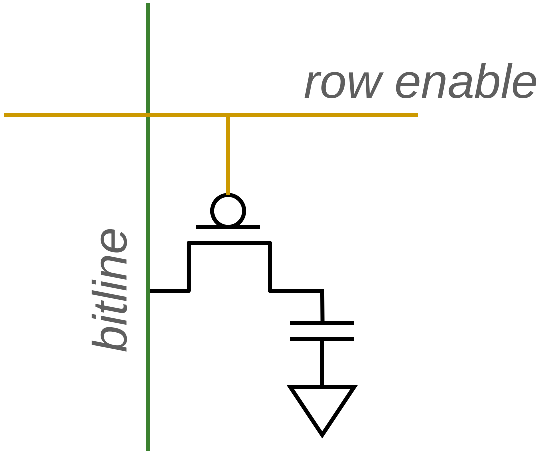
During the process, 8 columns are read sequentially.

Memory Technology: DRAM and SRAM

Memory Technology: DRAM

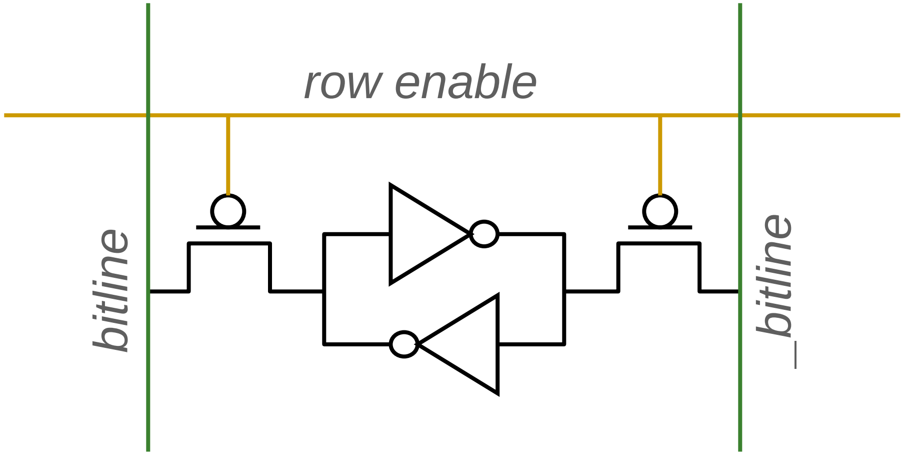
- Dynamic random access memory
- Capacitor charge state indicates stored value
  - Whether the capacitor is charged or discharged indicates storage of 1 or 0
  - 1 capacitor
  - 1 access transistor

- Capacitor leaks through the RC path
  - DRAM cell loses charge over time
  - DRAM cell needs to be refreshed

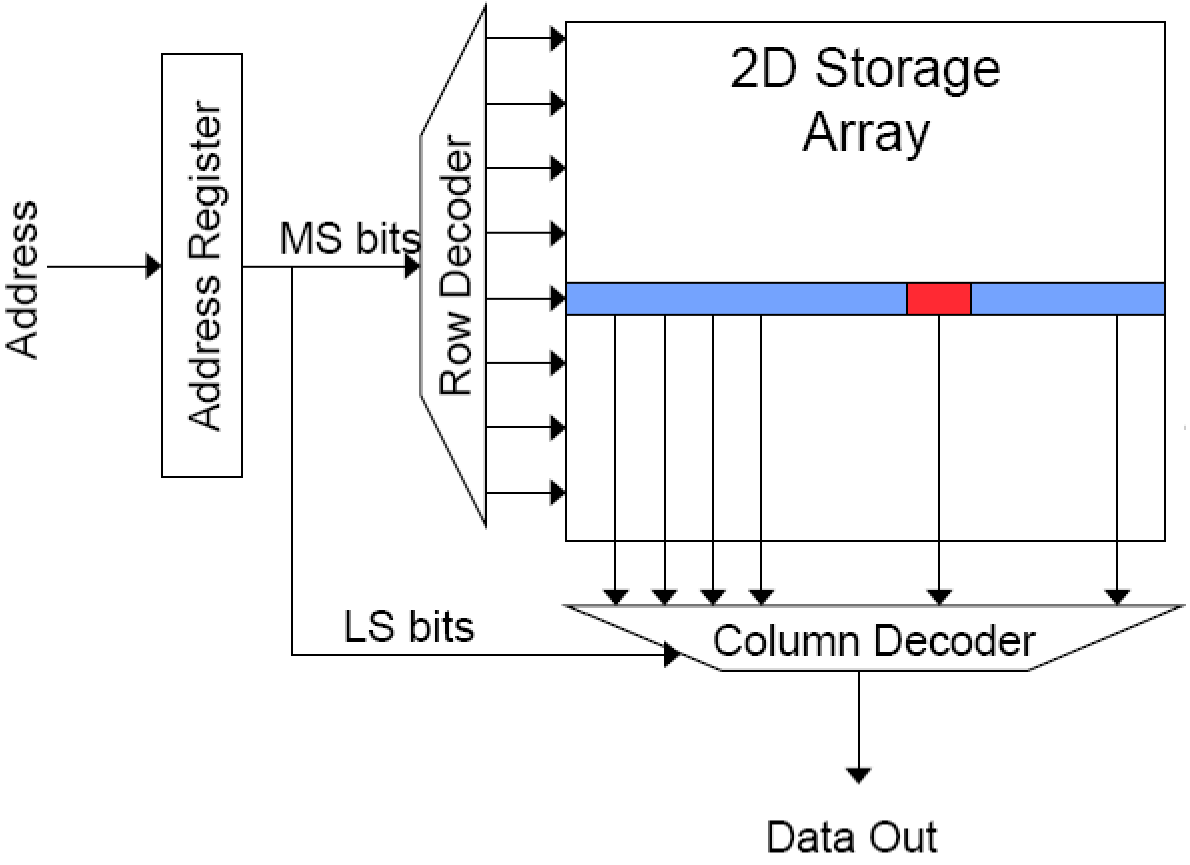


Memory Technology: SRAM

- Static random access memory
- Two cross coupled inverters store a single bit
  - Feedback path enables the stored value to persist in the "cell"
  - 4 transistors for storage
  - 2 transistors for access



Memory Bank Organization and Operation



- Read access sequence:

1. Decode row address & drive word-lines

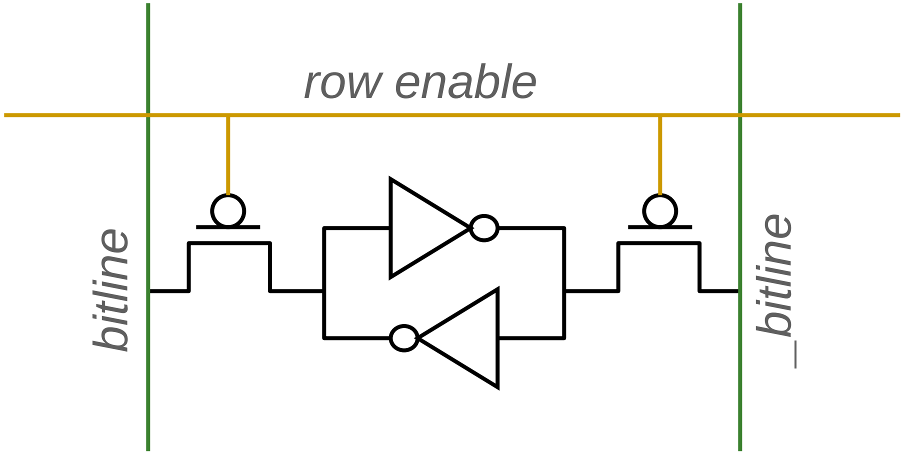
2. Selected bits drive bit-lines
   - Entire row read

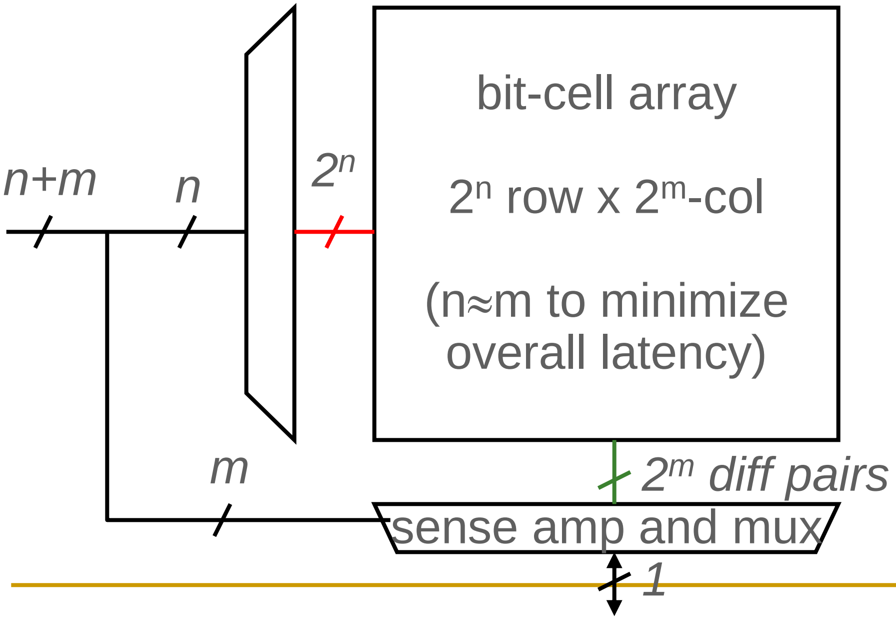
3. Amplify row data

4. Decode column address & select subset of row
   - Send to output

5. Precharge bit-lines
   - For next access

SRAM (Static Random Access Memory)





Read Sequence

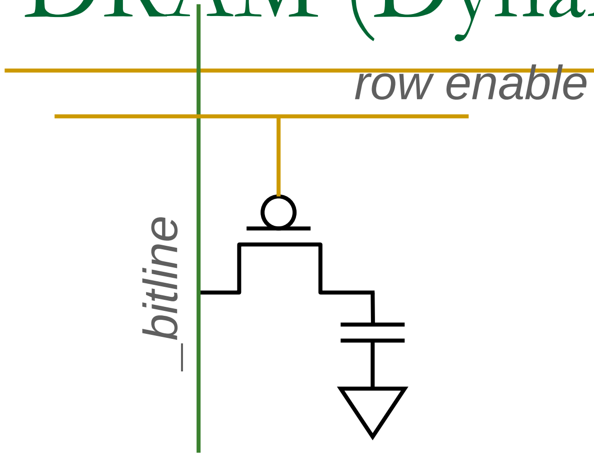
1. address decode
2. drive row select
3. selected bit-cells drive bitlines
   (entire row is read together)
4. differential sensing and column select
   (data is ready)
5. precharge all bitlines
   (for next read or write)

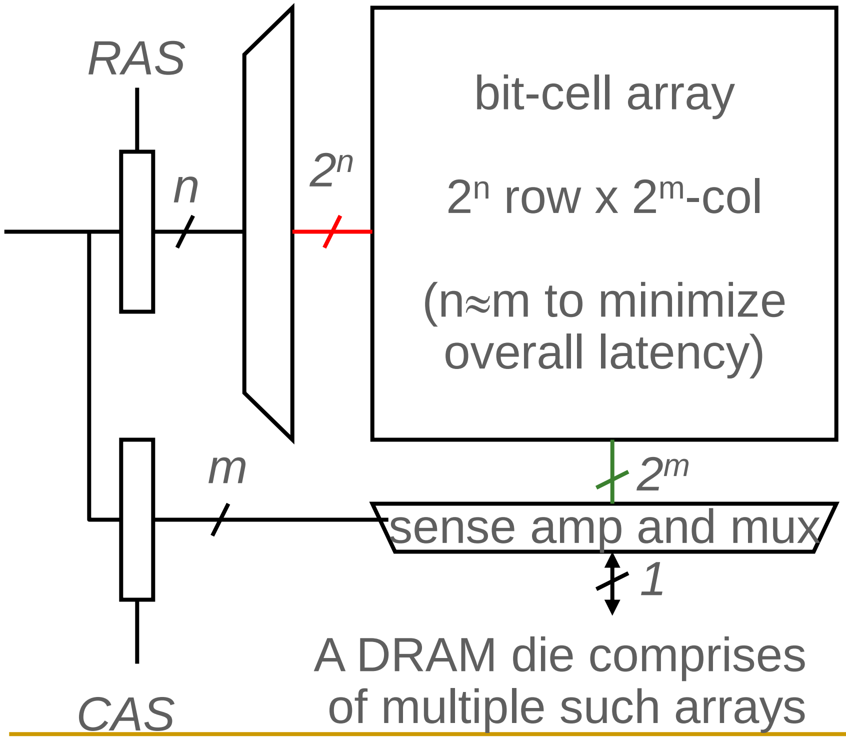
Access latency dominated by steps 2 and 3

Cycling time dominated by steps 2, 3 and 5

- step 2 proportional to 2^m
- step 3 and 5 proportional to 2^n

DRAM (Dynamic Random Access Memory)





A DRAM die comprises of multiple such arrays

Bit stored as charge on node capacitor (non-restorative)

- bit cell loses charge when read
- bit cell loses charge over time

Read Sequence

1~3 same as SRAM

4. a “flip-flopping” sense amp amplifies and regenerates the bitline, data bit is mux’ed out

5. precharge all bitlines

Destructive reads

Charge loss over time

Refresh: A DRAM controller must periodically read each row within the allowed refresh time (10s of ms) such that charge is restored

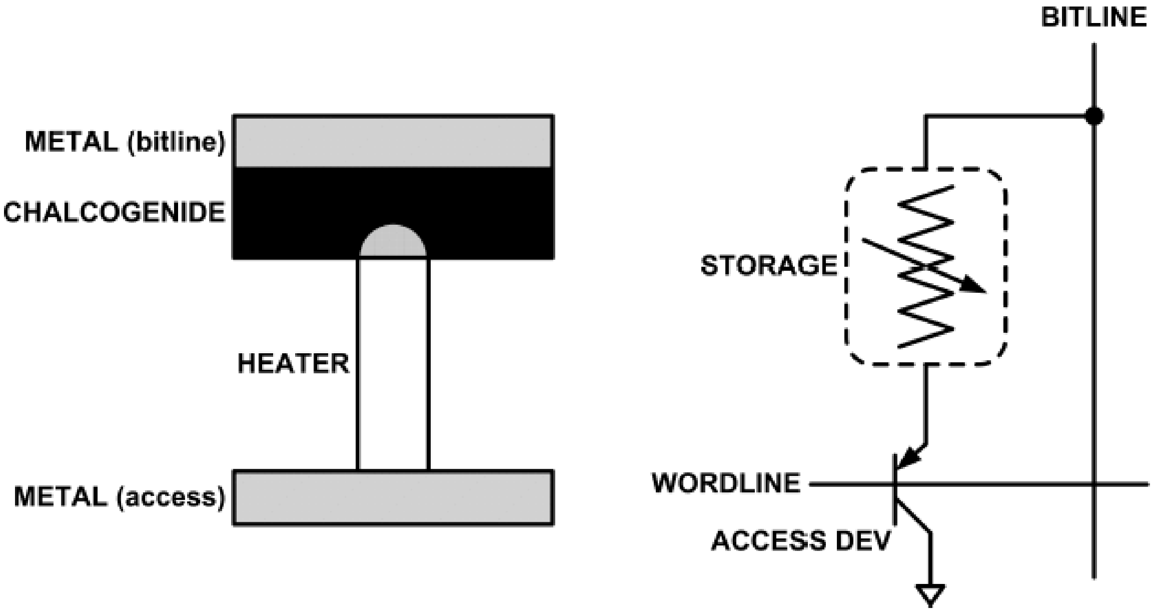
DRAM vs. SRAM

- DRAM
  - Slower access (capacitor)
  - Higher density (1T 1C cell)
  - Lower cost
  - Requires refresh (power, performance, circuitry)
  - Manufacturing requires putting capacitor and logic together

- SRAM
  - Faster access (no capacitor)
  - Lower density (6T cell)
  - Higher cost
  - No need for refresh
  - Manufacturing compatible with logic process (no capacitor)

An Aside: Phase Change Memory

- Phase change material (chalcogenide glass) exists in two states:
  - Amorphous: Low optical reflexivity and high electrical resistivity
  - Crystalline: High optical reflexivity and low electrical resistivity

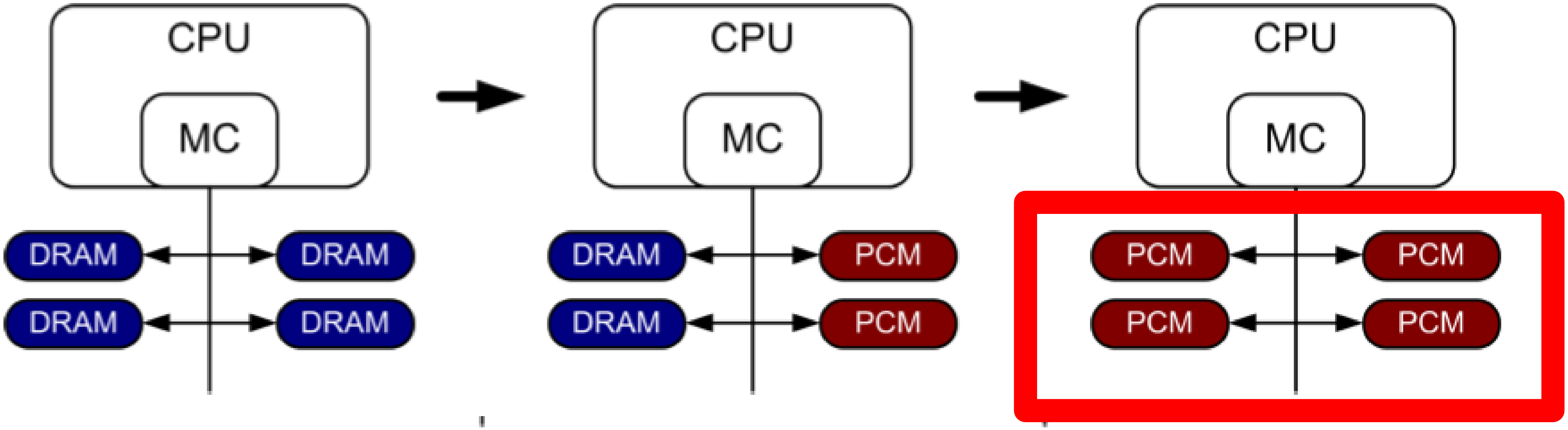


PCM is resistive memory: High resistance (0), Low resistance (1)

Lee, Ipek, Mutlu, Burger, "Architecting Phase Change Memory as a Scalable DRAM Alternative," ISCA 2009.

PCM-based Main Memory

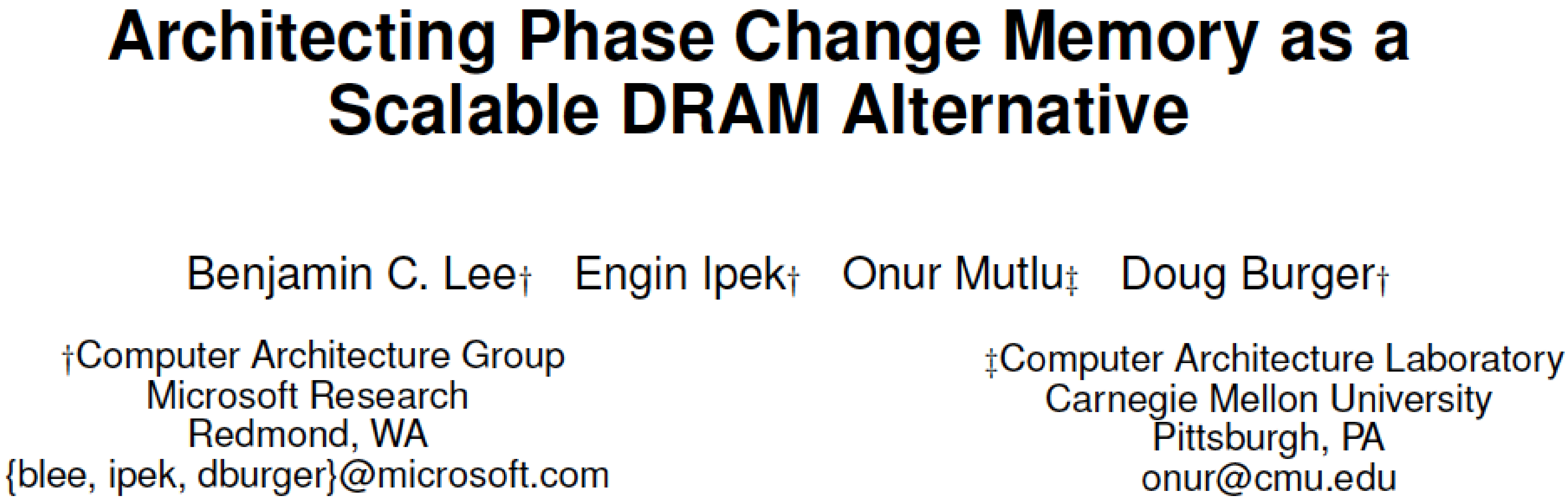
- How should PCM-based (main) memory be organized?



- Pure PCM main memory [Lee et al., ISCA'09, Top Picks'10]
  - How to redesign the system to tolerate PCM shortcomings
- Hybrid PCM+DRAM [Qureshi+ ISCA'09, Dhiman+ DAC'09]
  - How to partition/migrate data between PCM and DRAM

Reading: PCM as Main Memory: Idea in 2009

- 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

Reading: More on PCM As Main Memory

- Benjamin C. Lee, Ping Zhou, Jun Yang, Youtao Zhang, Bo Zhao, Engin Ipek, Onur Mutlu, and Doug Burger, **"Phase Change Technology and the Future of Main Memory"** *IEEE Micro, Special Issue: Micro's Top Picks from 2009 Computer Architecture Conferences (**MICRO TOP PICKS**)*, Vol. 30, No. 1, pages 60-70, January/February 2010.

PHASE-CHANGE TECHNOLOGY AND THE FUTURE OF MAIN MEMORY

Intel Optane Persistent Memory (2019)

- Non-volatile main memory
- Based on 3D-XPoint Technology

https://www.storagereview.com/intel_optane_dc_persistent_memory_module_pmm

DRAM vs. PCM

- DRAM
  - Faster access (capacitor)
  - Lower density (capacitor less scalable) → higher cost
  - Requires refresh (power, performance, circuitry)
  - Manufacturing requires putting capacitor and logic together
  - Volatile (loses data at loss of power)
  - No endurance problems
  - Lower access energy
- PCM
  - Slower access (heating and cooling based "phase change" operation)
  - Higher density (phase change material more scalable) → lower cost
  - No need for refresh
  - Manufacturing requires less conventional processes – less mature
  - Non-volatile (does **not** lose data at loss of power)
  - Endurance problems (a cell cannot be used after N writes to it)
  - Higher access energy

Charge vs. Resistive Memories

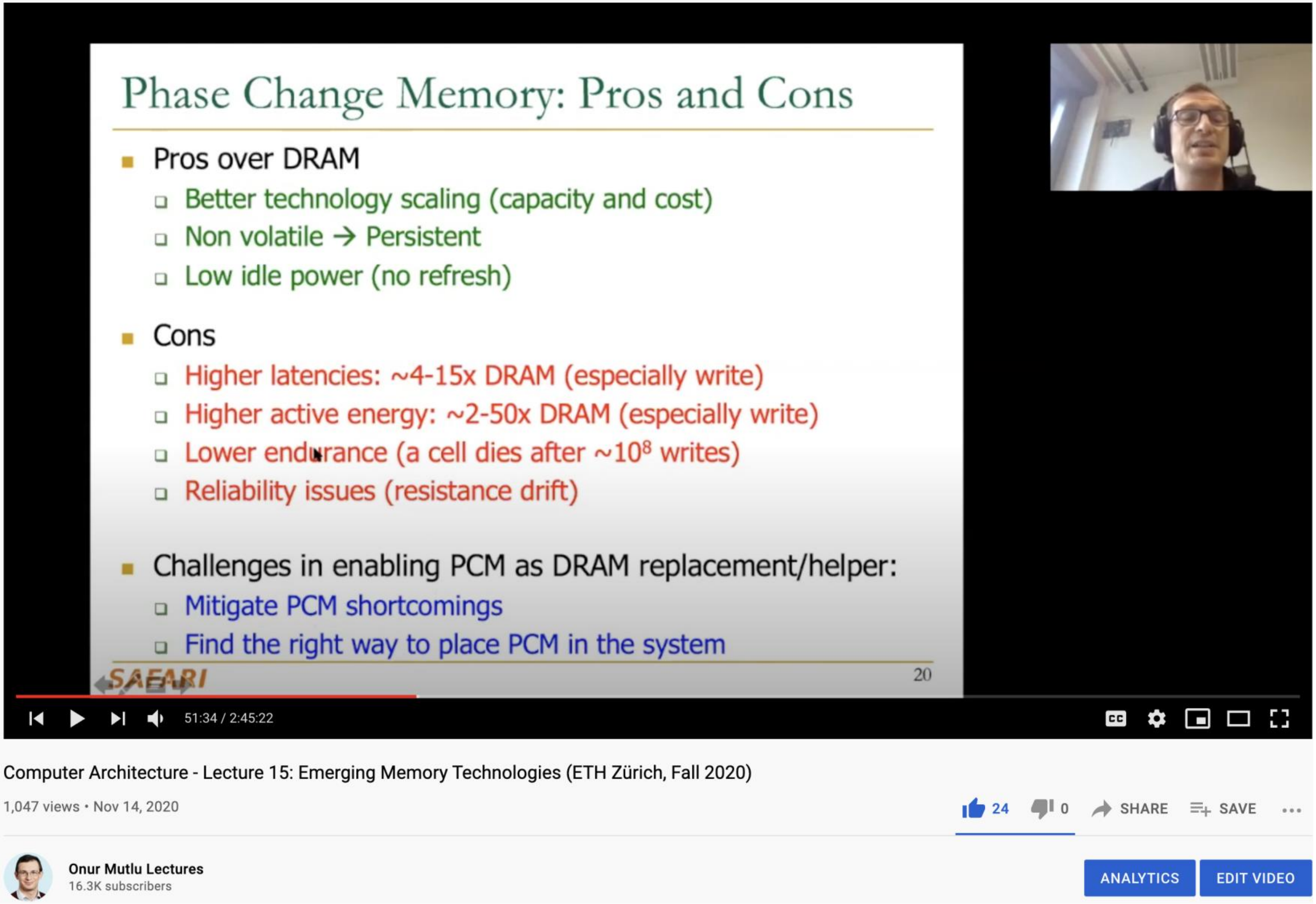
- Charge Memory (e.g., DRAM, Flash)
  - Write data by capturing charge Q
  - Read data by detecting voltage V

- Resistive Memory (e.g., PCM, STT-MRAM, memristors)
  - Write data by pulsing current dQ/dt
  - Read data by detecting resistance R

Promising Resistive Memory Technologies

- PCM
  - Inject current to change material phase
  - Resistance determined by phase
- STT-MRAM
  - Inject current to change magnet polarity
  - Resistance determined by polarity
- Memristors/RRAM/ReRAM
  - Inject current to change atomic structure
  - Resistance determined by atom distance

More on Emerging Memory Technologies



Phase Change Memory: Pros and Cons

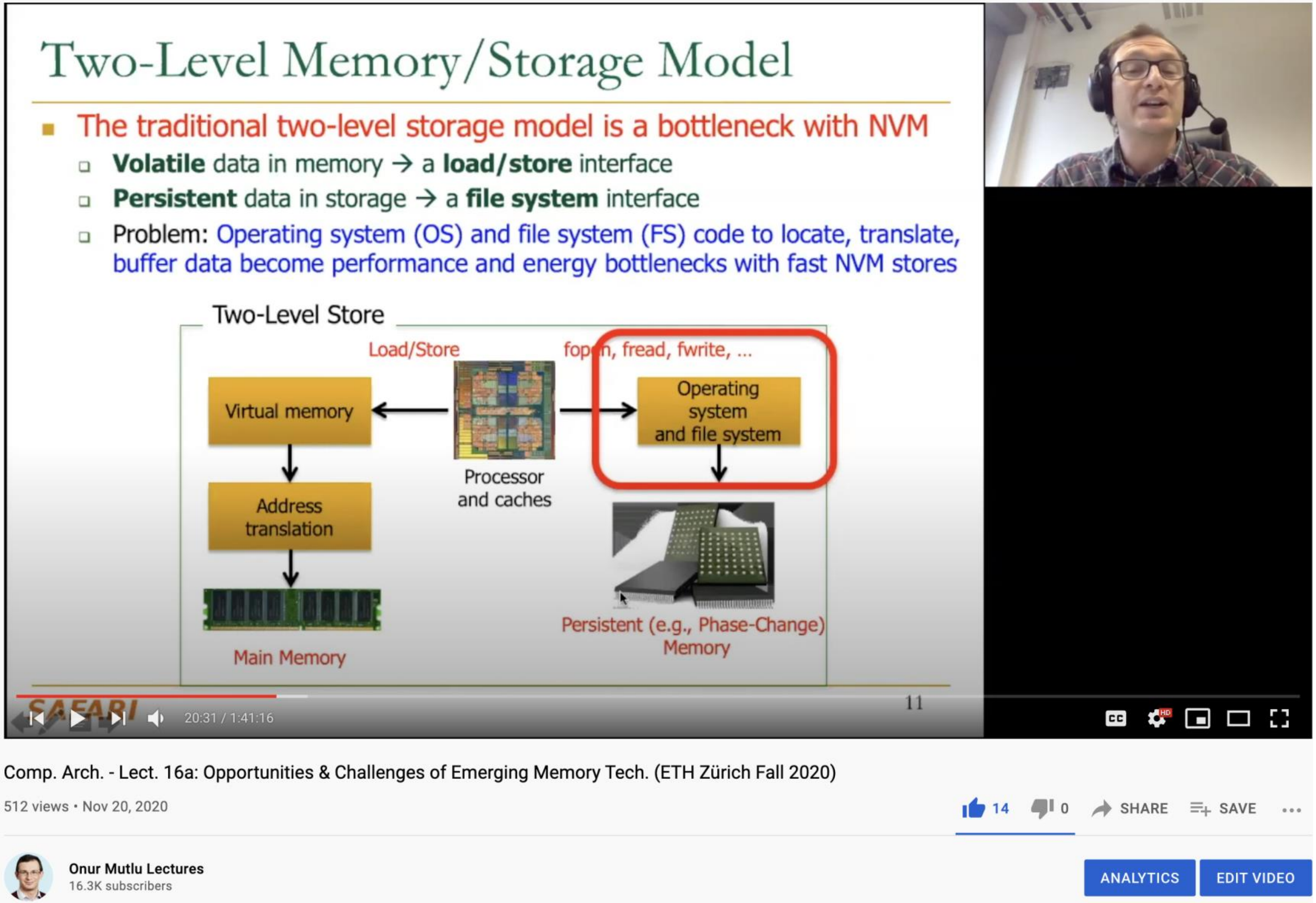
- Pros over DRAM
  - Better technology scaling (capacity and cost)
  - Non volatile → Persistent
  - Low idle power (no refresh)
- Cons
  - Higher latencies: ~4-15x DRAM (especially write)
  - Higher active energy: ~2-50x DRAM (especially write)
  - Lower endurance (a cell dies after ~10^8 writes)
  - Reliability issues (resistance drift)
- Challenges in enabling PCM as DRAM replacement/helper:
  - Mitigate PCM shortcomings
  - Find the right way to place PCM in the system

Computer Architecture - Lecture 15: Emerging Memory Technologies (ETH Zürich, Fall 2020)

1,047 views • Nov 14, 2020

https://www.youtube.com/watch?v=AlE1rD9G_YU&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=28

More on Emerging Memory Technologies



Two-Level Memory/Storage Model

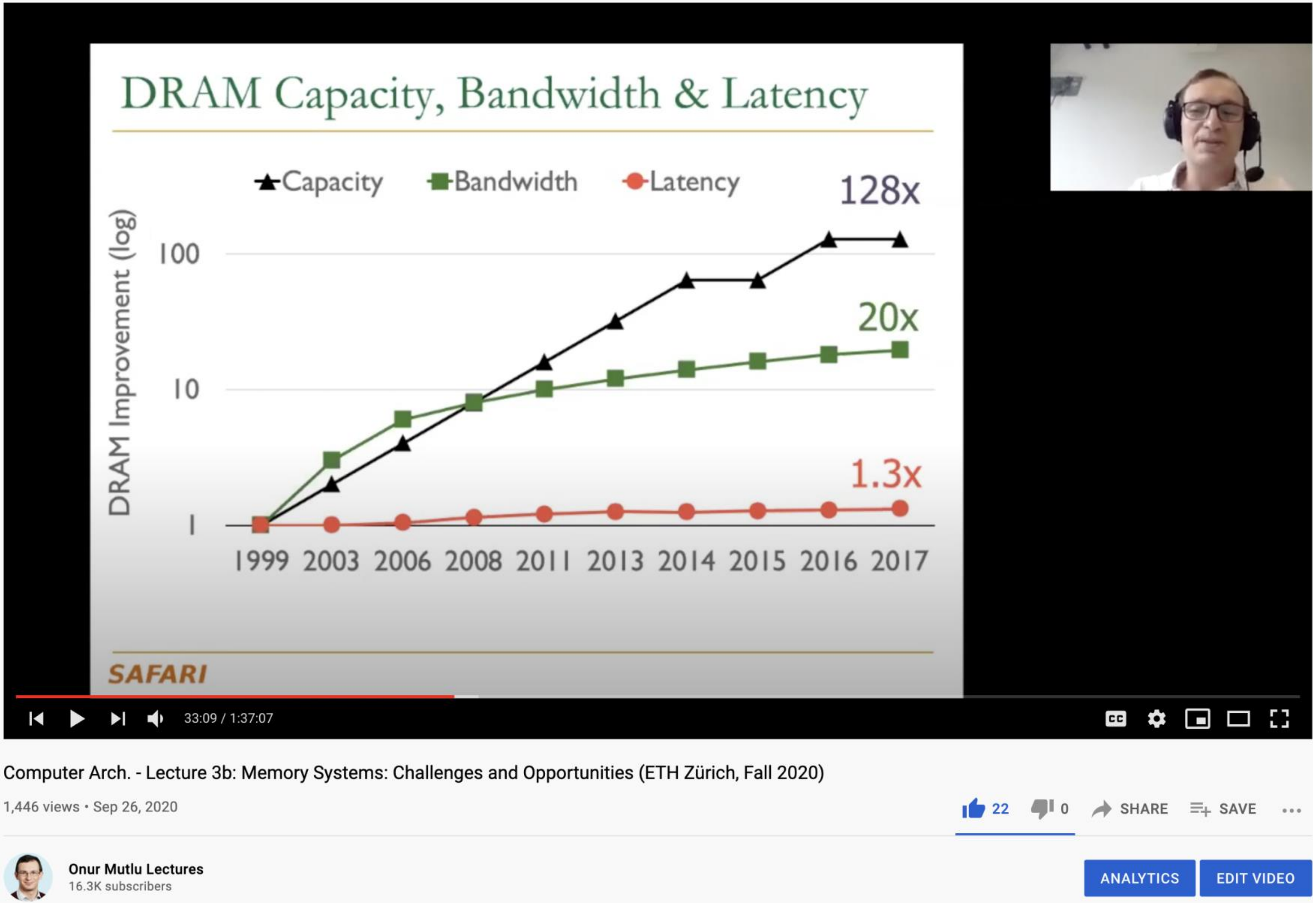
- The traditional two-level storage model is a bottleneck with NVM
  - **Volatile** data in memory → a **load/store** interface
  - **Persistent** data in storage → a **file system** interface
  - Problem: Operating system (OS) and file system (FS) code to locate, translate, buffer data become performance and energy bottlenecks with fast NVM stores

Comp. Arch. - Lect. 16a: Opportunities & Challenges of Emerging Memory Tech. (ETH Zürich Fall 2020)

512 views • Nov 20, 2020

https://www.youtube.com/watch?v=pmLszWGmMGQ&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=29

More on Memory Technologies



DRAM Capacity, Bandwidth & Latency

Computer Arch. - Lecture 3b: Memory Systems: Challenges and Opportunities (ETH Zürich, Fall 2020)

1,446 views • Sep 26, 2020

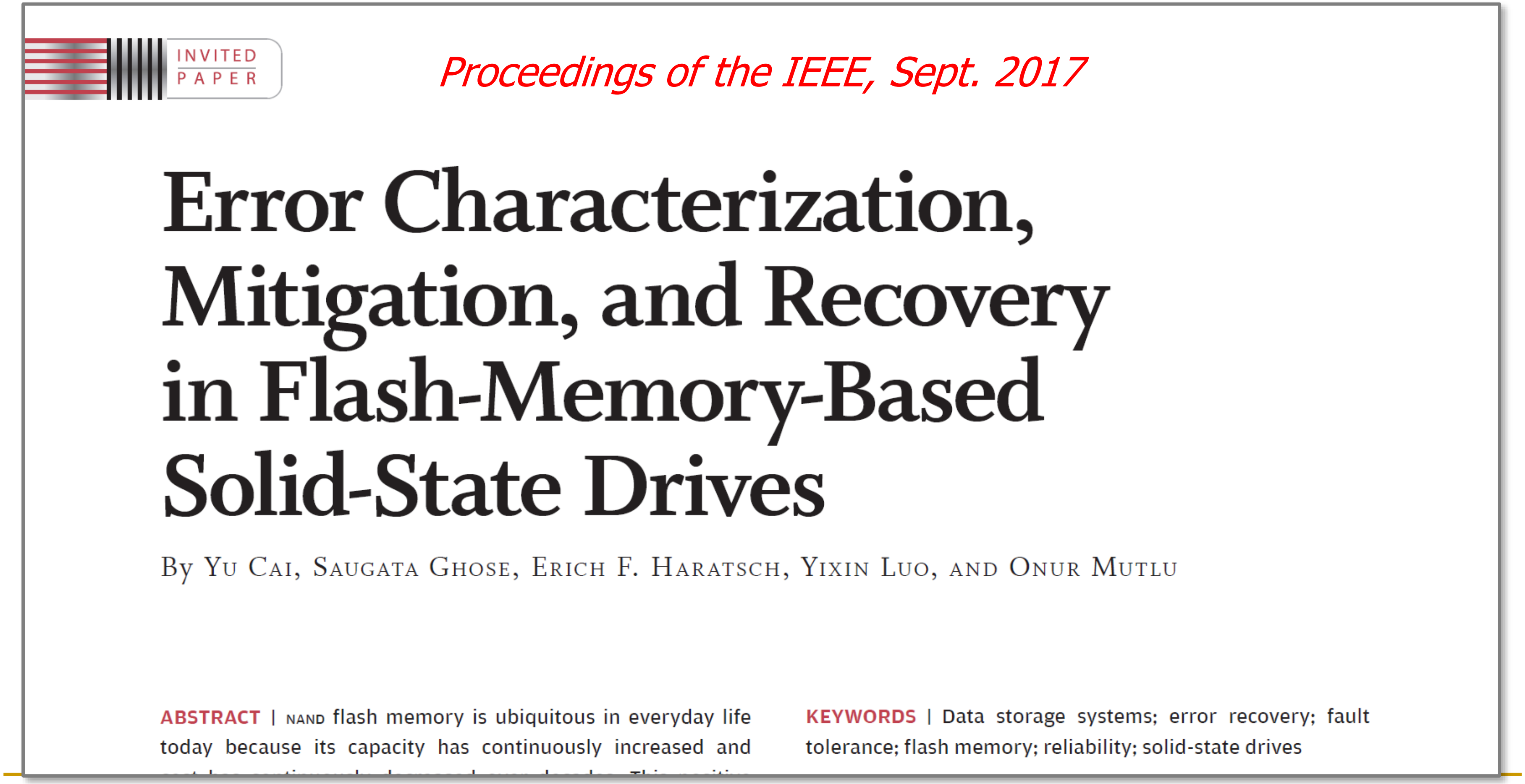
Onur Mutlu Lectures

16.3K subscribers

https://www.youtube.com/watch?v=pmLszWGmMGQ&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=29

A Bit on Flash Memory & SSDs

- Flash memory was a very "doubtful" emerging technology
  - for at least two decades



INVITED PAPER

Proceedings of the IEEE, Sept. 2017

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

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

ABSTRACT | NAND flash memory is ubiquitous in everyday life today because its capacity has continuously increased and cost has continuously decreased over decades. This positive

KEYWORDS | Data storage systems; error recovery; fault tolerance; flash memory; reliability; solid-state drives

https://arxiv.org/pdf/1711.11427.pdf

A Flash Memory SSD Controller

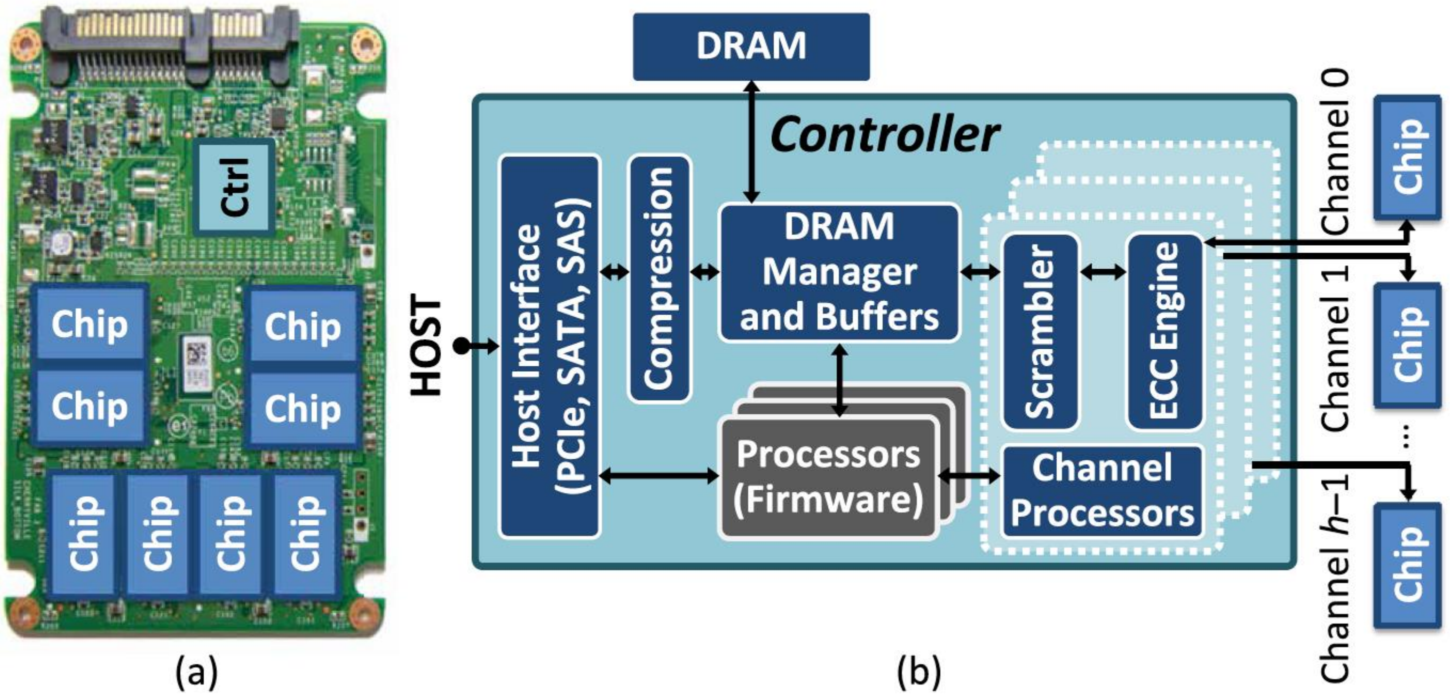
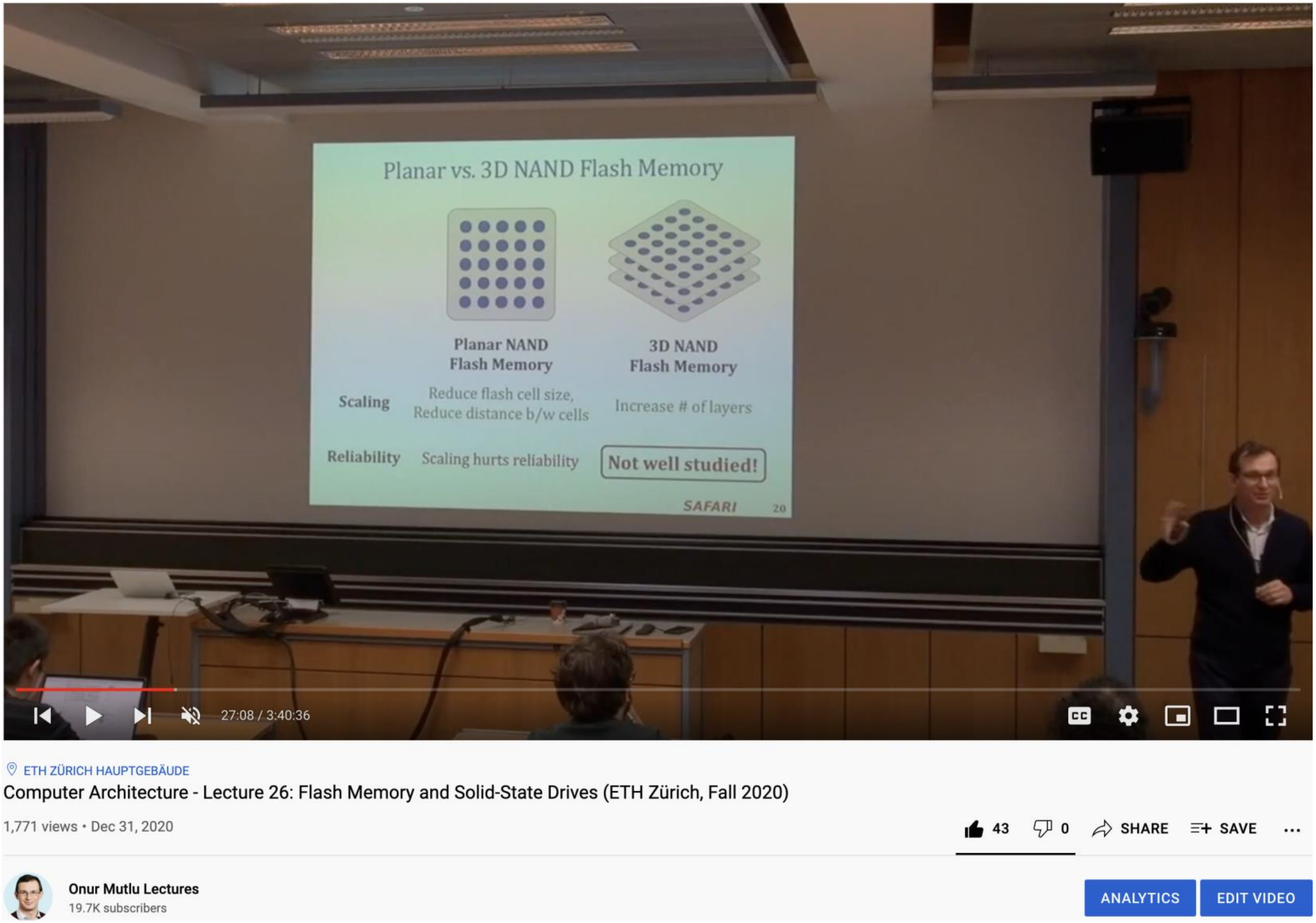


Fig. 1. ***(a) SSD system architecture, showing controller (Ctrl) and chips. (b) Detailed view of connections between controller components and chips.***

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

https://arxiv.org/pdf/1711.11427.pdf

Lecture on Flash Memory & SSDs



Planar vs. 3D NAND Flash Memory

	Planar NAND Flash Memory	3D NAND Flash Memory
Scaling	Reduce flash cell size, Reduce distance b/w cells	Increase # of layers
Reliability	Scaling hurts reliability	Not well studied!

ETH ZÜRICH HAUPTGEBÄUDE

Computer Architecture - Lecture 26: Flash Memory and Solid-State Drives (ETH Zürich, Fall 2020)

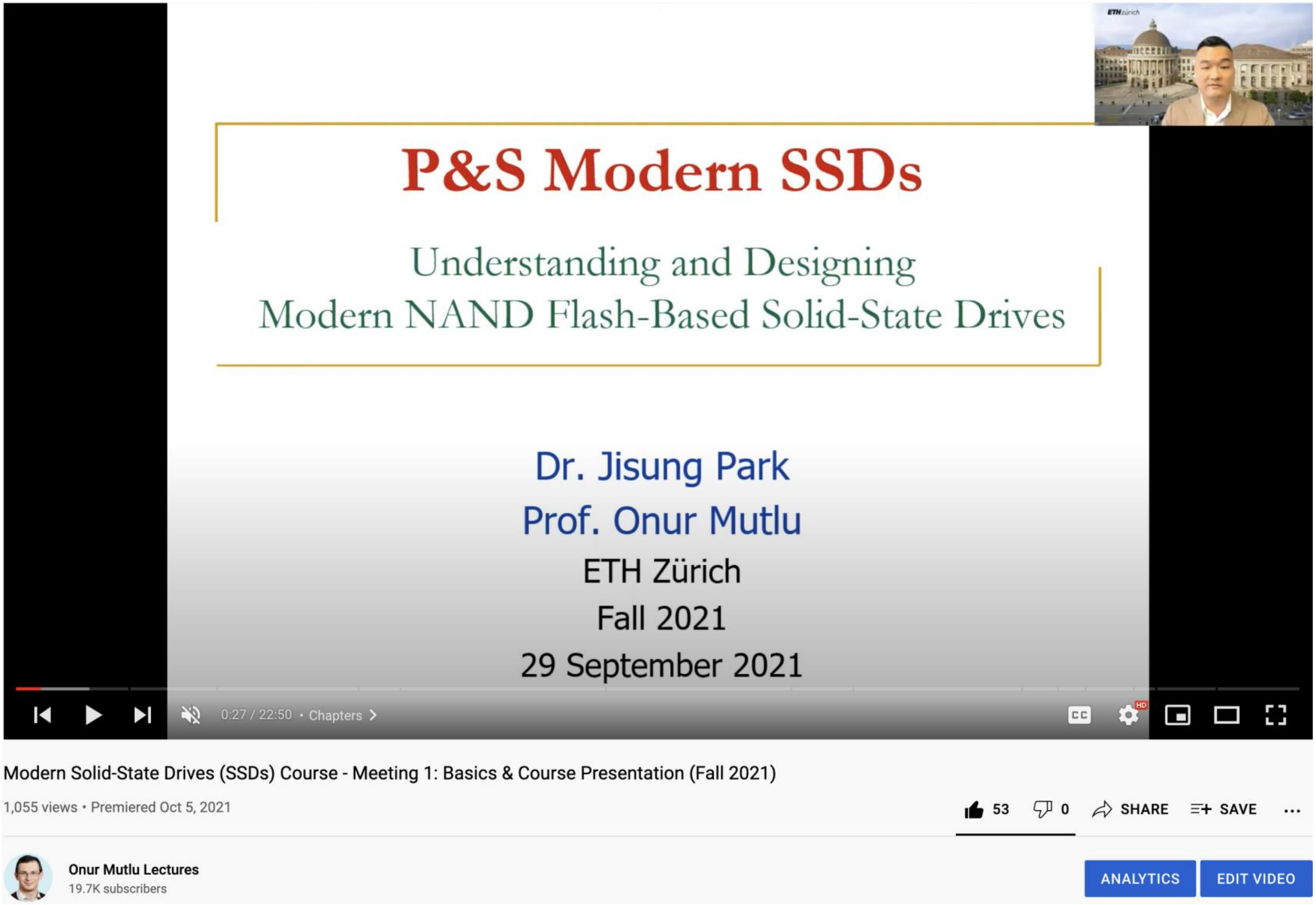
1,771 views • Dec 31, 2020

Onur Mutlu Lectures

19.7K subscribers

SAFARI https://www.youtube.com/watch?v=rninK6KWBeM&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=47

Special Course on Flash Memory & SSDs



P&S Modern SSDs

Understanding and Designing
Modern NAND Flash-Based Solid-State Drives

Dr. Jisung Park
Prof. Onur Mutlu
ETH Zürich
Fall 2021
29 September 2021

Modern Solid-State Drives (SSDs) Course - Meeting 1: Basics & Course Presentation (Fall 2021)

1,055 views • Premiered Oct 5, 2021

Onur Mutlu Lectures
19.7K subscribers

https://www.youtube.com/watch?v=dSsZA6JGcLE&list=PL5Q2soXY2Zi_4tPsgX9m_D7AI_c1T86cO

Lectures on Memory Technologies

- Computer Architecture, Fall 2020, Lecture 15
  - Emerging Memory Technologies (ETH, Fall 2020)
  - https://www.youtube.com/watch?v=AlE1rD9G_YU&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=28

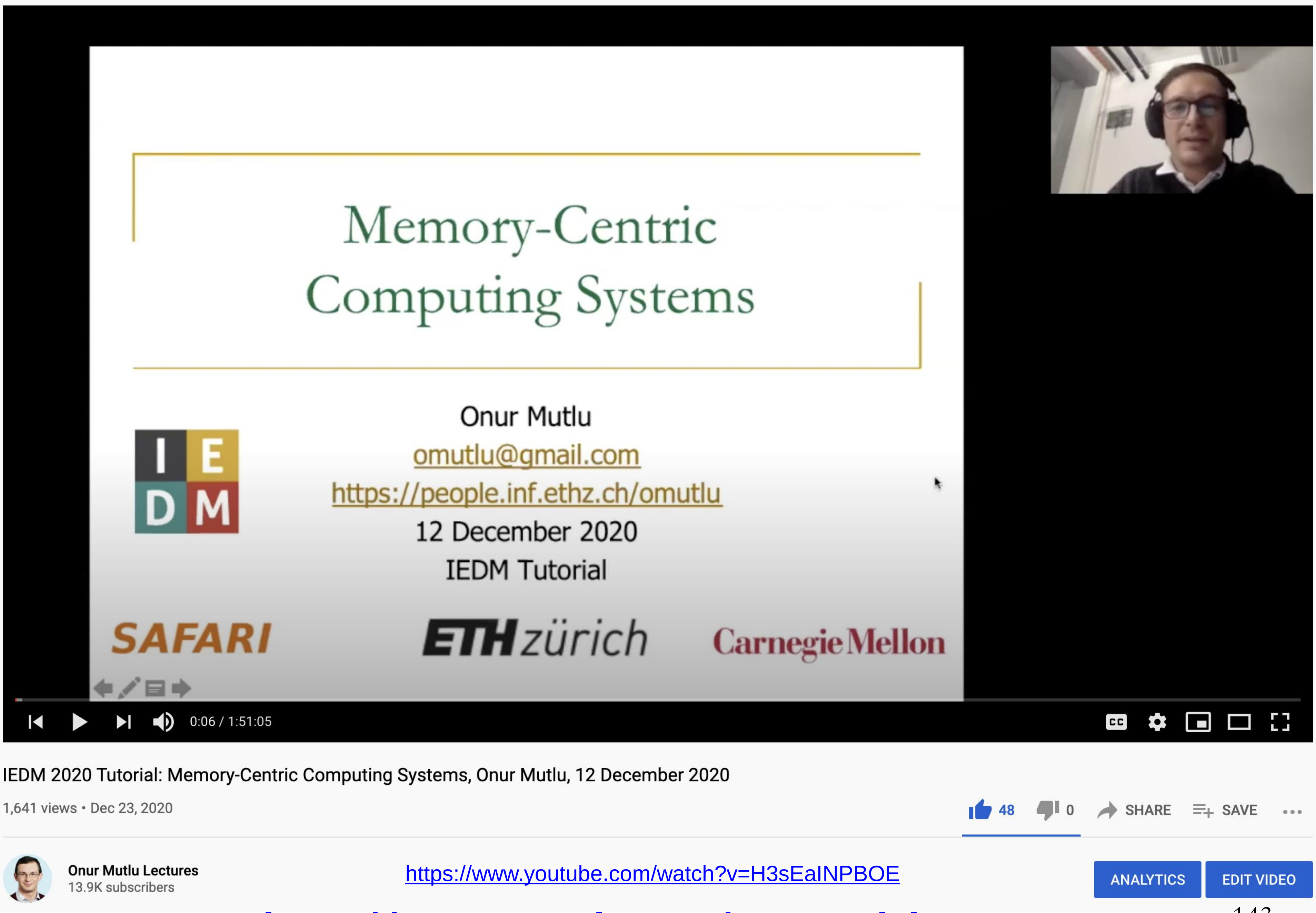
- Computer Architecture, Fall 2020, Lecture 16a
  - Opportunities & Challenges of Emerging Memory Tech (ETH, Fall 2020)
  - https://www.youtube.com/watch?v=pmLszWGmMGQ&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=29

- Computer Architecture, Fall 2020, Lecture 3b
  - Memory Systems: Challenges & Opportunities (ETH, Fall 2020)
  - https://www.youtube.com/watch?v=Q2FbUxD7GHs&list=PL5Q2soXY2Zi9xidyIgBxUz7xRPS-wisBN&index=6

A Tutorial on Memory-Centric Systems

- Onur Mutlu,
  "Memory-Centric Computing Systems"
  Invited Tutorial at *66th International Electron Devices Meeting (**IEDM**)*, Virtual, 12 December 2020.
  [Slides (pptx) (pdf)]
  [Executive Summary Slides (pptx) (pdf)]
  [Tutorial Video (1 hour 51 minutes)]
  [Executive Summary Video (2 minutes)]
  [Abstract and Bio]
  [Related Keynote Paper from VLSI-DAT 2020]
  [Related Review Paper on Processing in Memory]

https://www.youtube.com/watch?v=H3sEaINPBOE



Memory-Centric Computing Systems

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
12 December 2020
IEDM Tutorial

IEDM 2020 Tutorial: Memory-Centric Computing Systems, Onur Mutlu, 12 December 2020

1,641 views • Dec 23, 2020

Onur Mutlu Lectures
13.9K subscribers

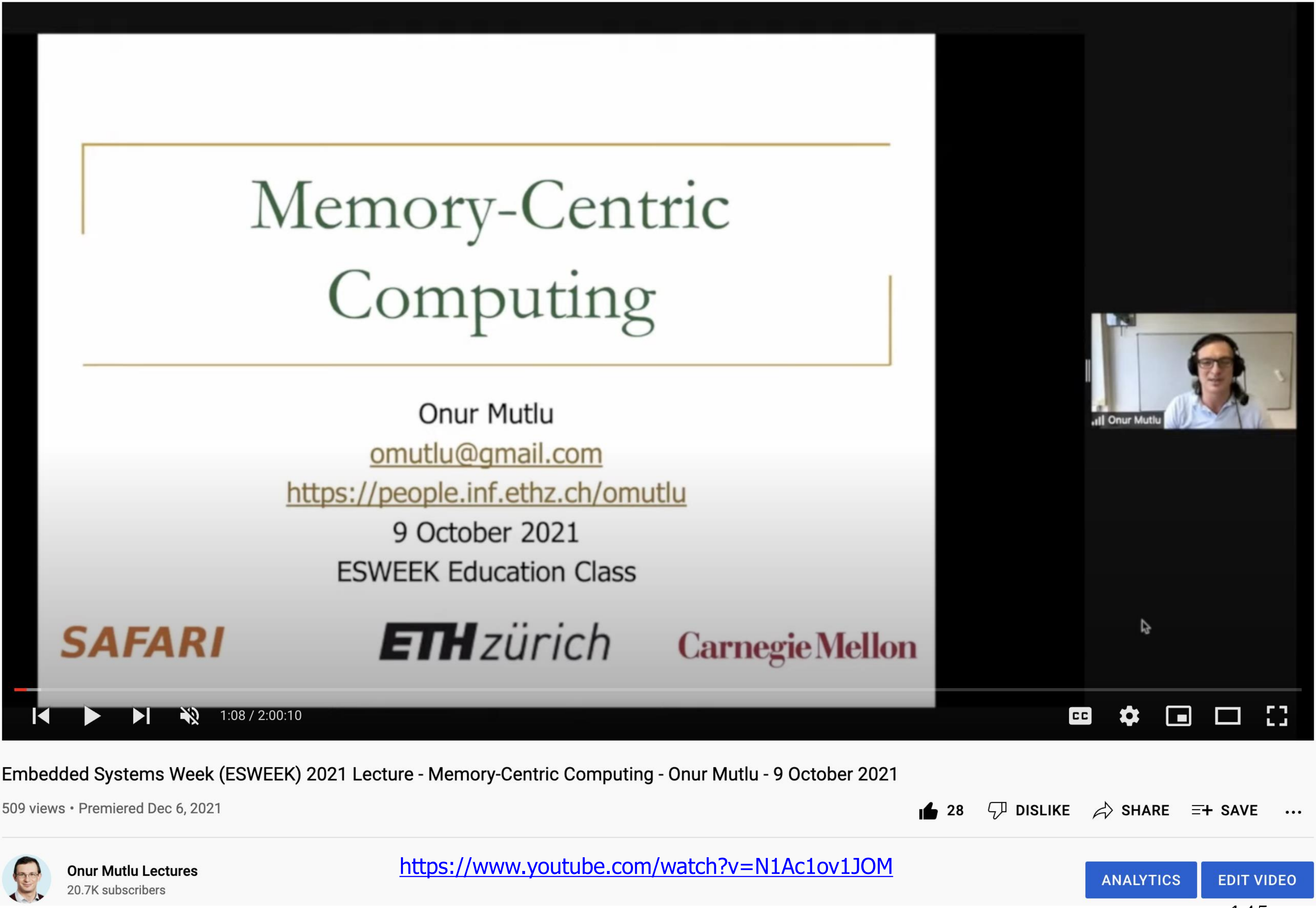
https://www.youtube.com/watch?v=H3sEaINPBOE

https://www.youtube.com/onurmutlulectures

Tutorial on Processing in Memory

- Onur Mutlu,
  "Memory-Centric Computing"
  *Education Class at Embedded Systems Week (**ESWEEK**)*,
  Virtual, 9 October 2021.
  [Slides (pptx) (pdf)]
  [Abstract (pdf)]
  [Talk Video (2 hours, including Q&A)]
  [Invited Paper at DATE 2021]
  ["A Modern Primer on Processing in Memory" paper]

https://www.youtube.com/watch?v=N1Ac1ov1JOM



Memory-Centric Computing

Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
9 October 2021
ESWEEK Education Class

Embedded Systems Week (ESWEEK) 2021 Lecture - Memory-Centric Computing - Onur Mutlu - 9 October 2021

509 views • Premiered Dec 6, 2021

Onur Mutlu Lectures
20.7K subscribers

https://www.youtube.com/watch?v=N1Ac1ov1JOM

https://www.youtube.com/onurmutlulectures

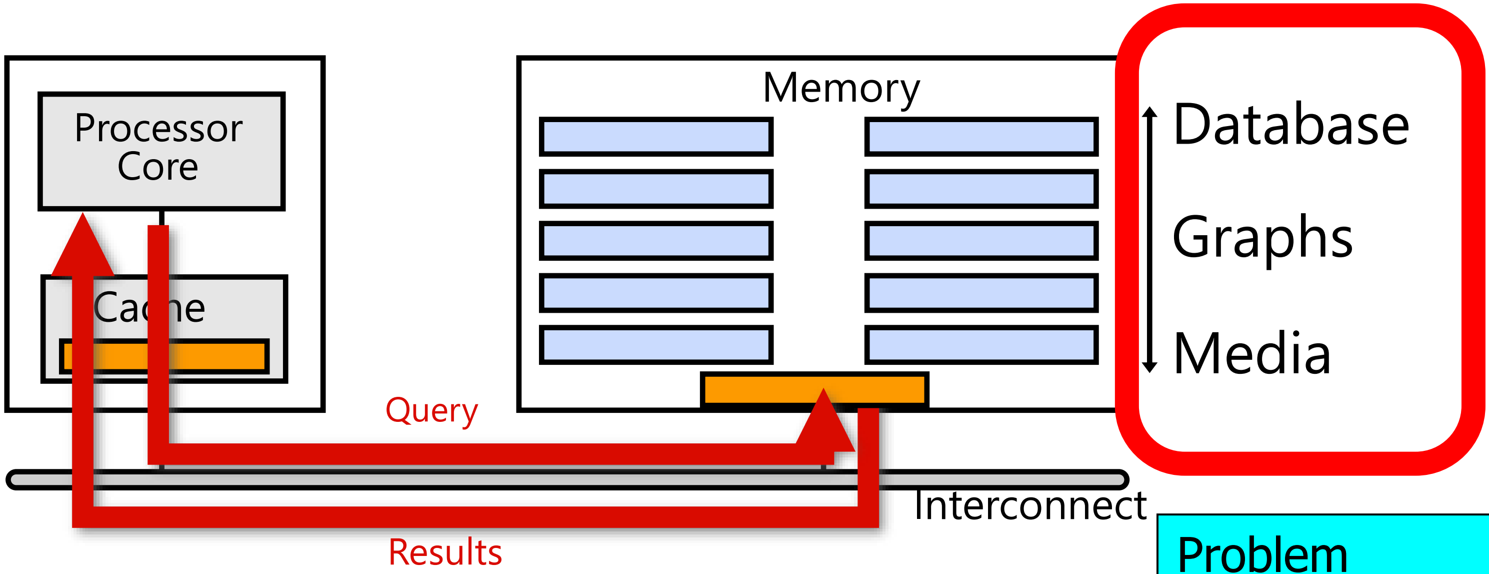
Digital Design & Computer Arch.

Lecture 22: Memory Overview, Organization & Technology

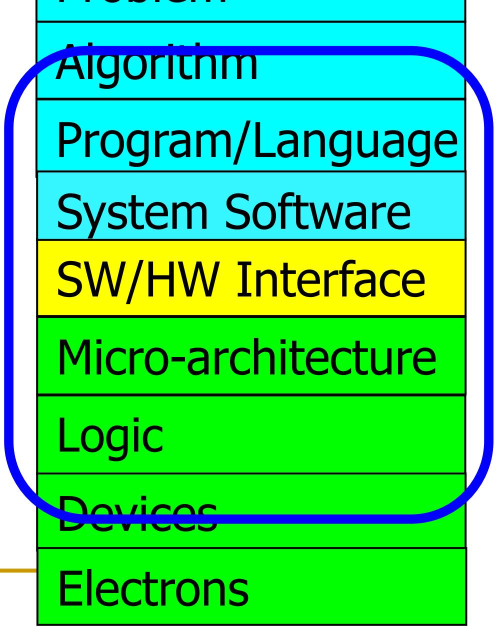
Prof. Onur Mutlu

ETH Zürich
Spring 2022
19 May 2022

Goal: Processing Inside Memory

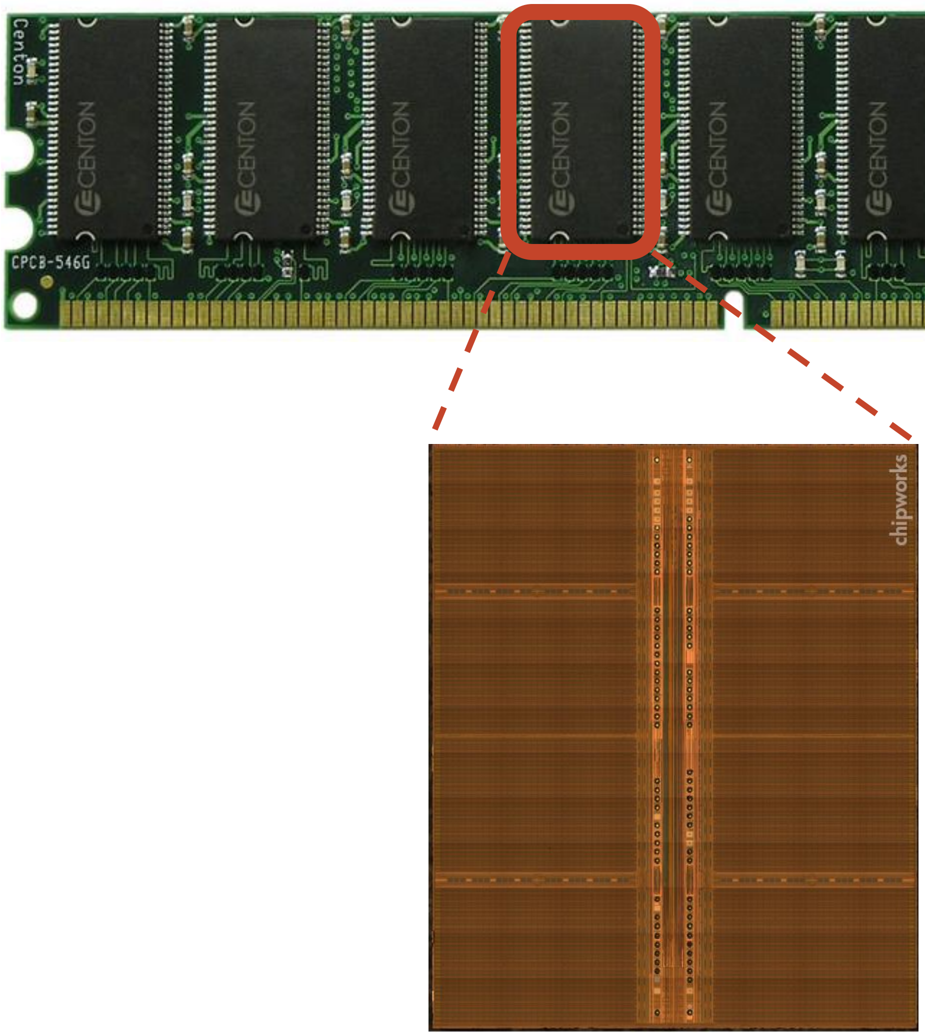


- Many questions … How do we design the:
  - compute-capable memory & controllers?
  - processors & communication units?
  - software & hardware interfaces?
  - system software, compilers, languages?
  - algorithms & theoretical foundations?



Backup Slides: Inside A DRAM Chip

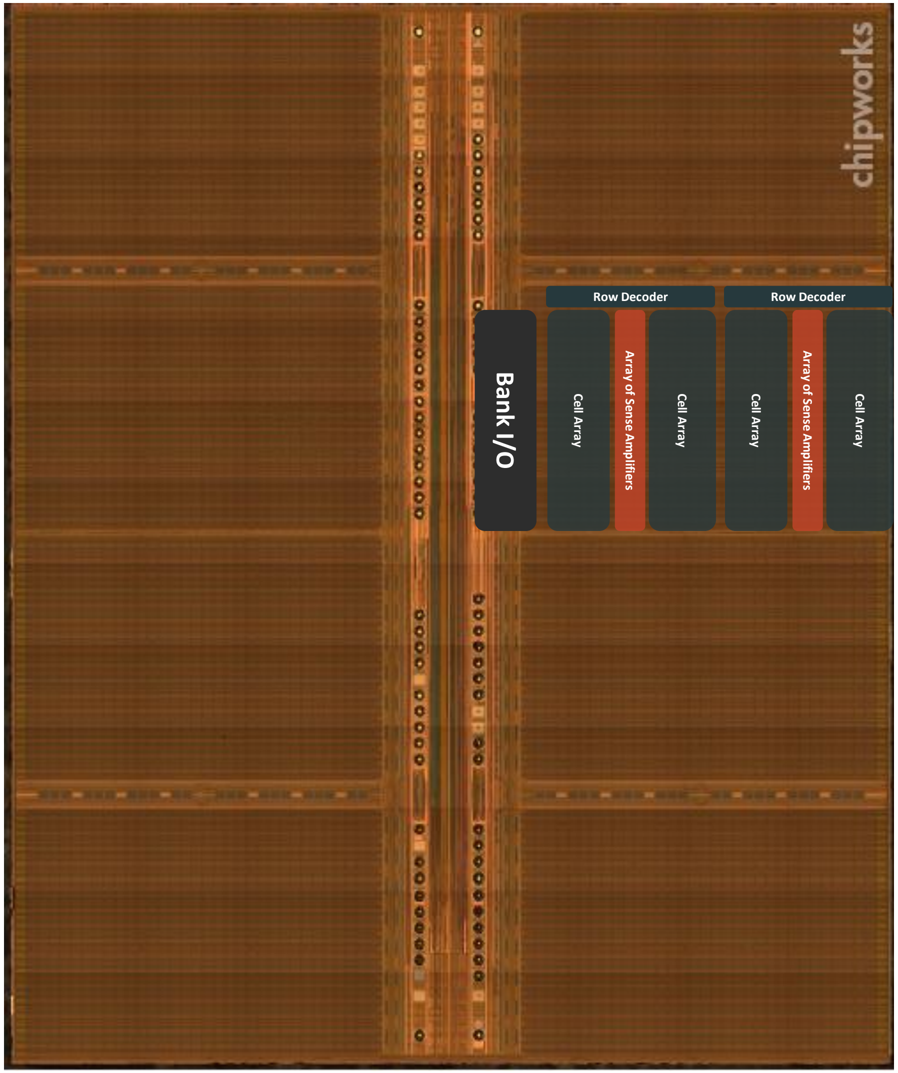
DRAM Module and Chip



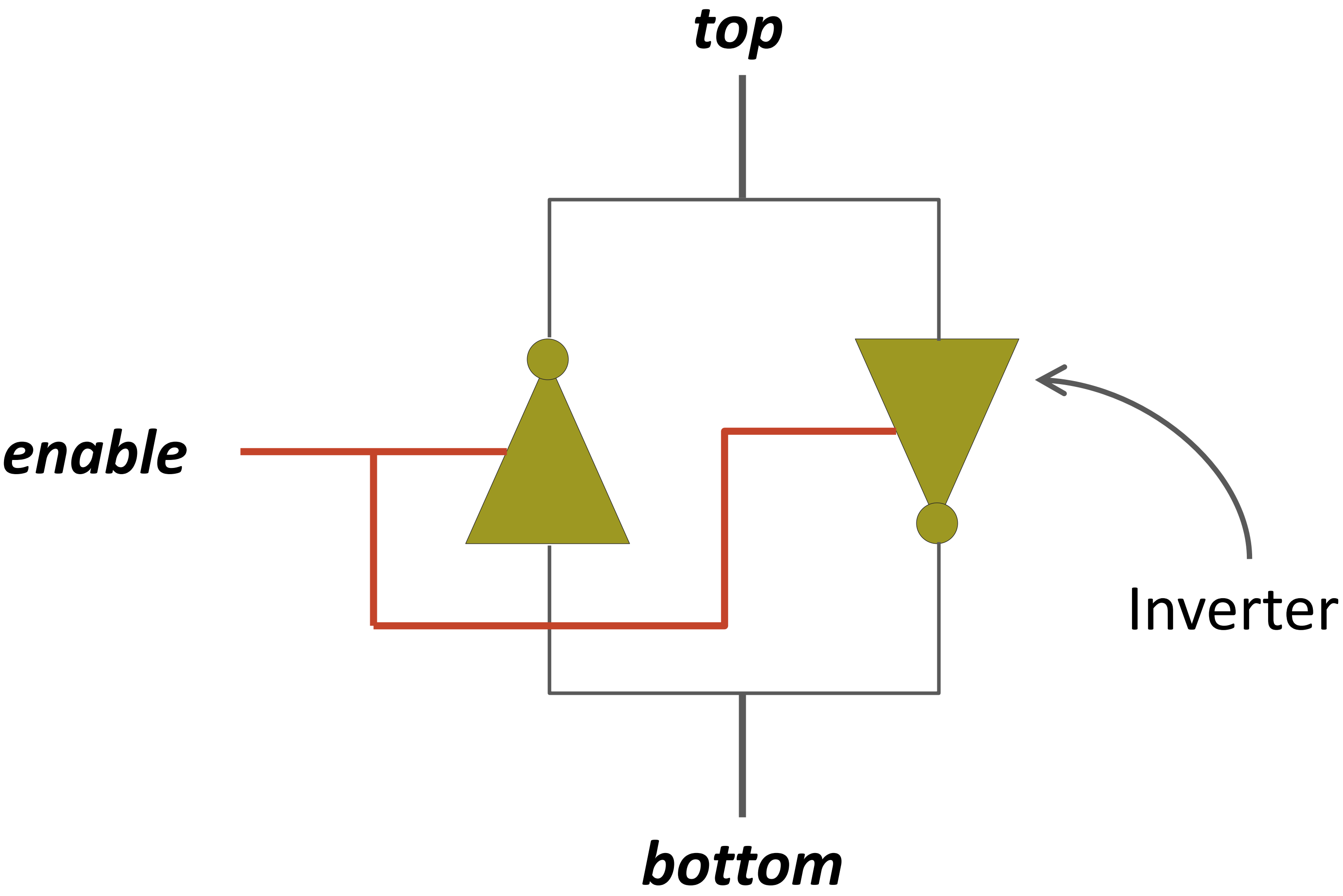
Goals in DRAM Design

- Cost
- Latency
- Bandwidth
- Parallelism
- Power
- Energy
- Reliability
- Security
- …

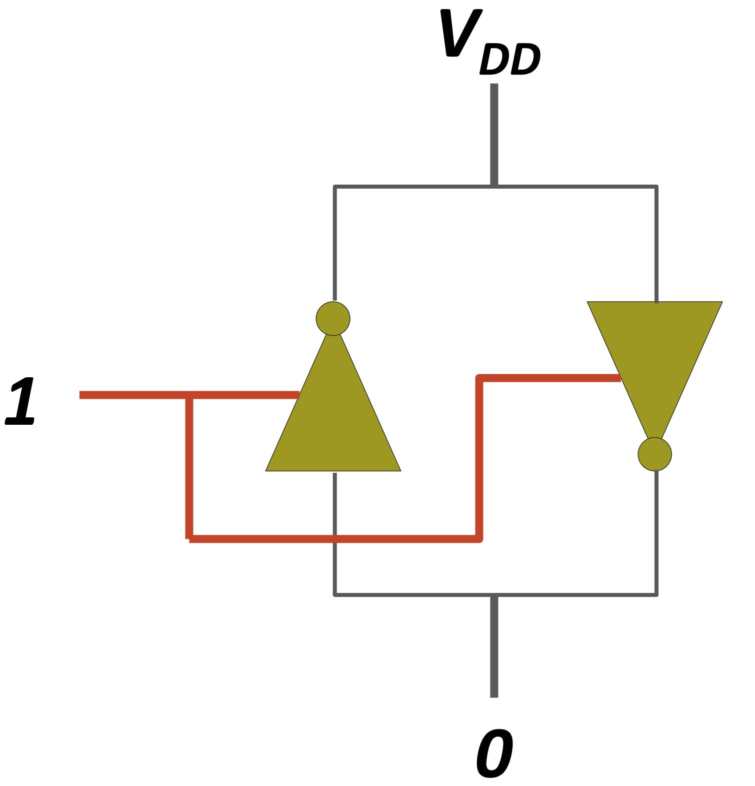
DRAM Chip


chipworks
Row Decoder
Row Decoder
Bank I/O
Cell Array
Array of Sense Amplifiers
Cell Array
Cell Array
Array of Sense Amplifiers
Cell Array

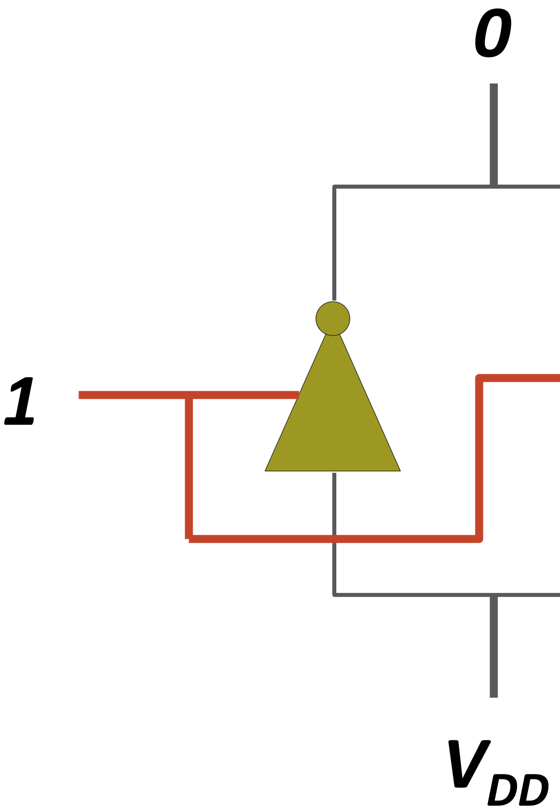
Sense Amplifier



Sense Amplifier – Two Stable States

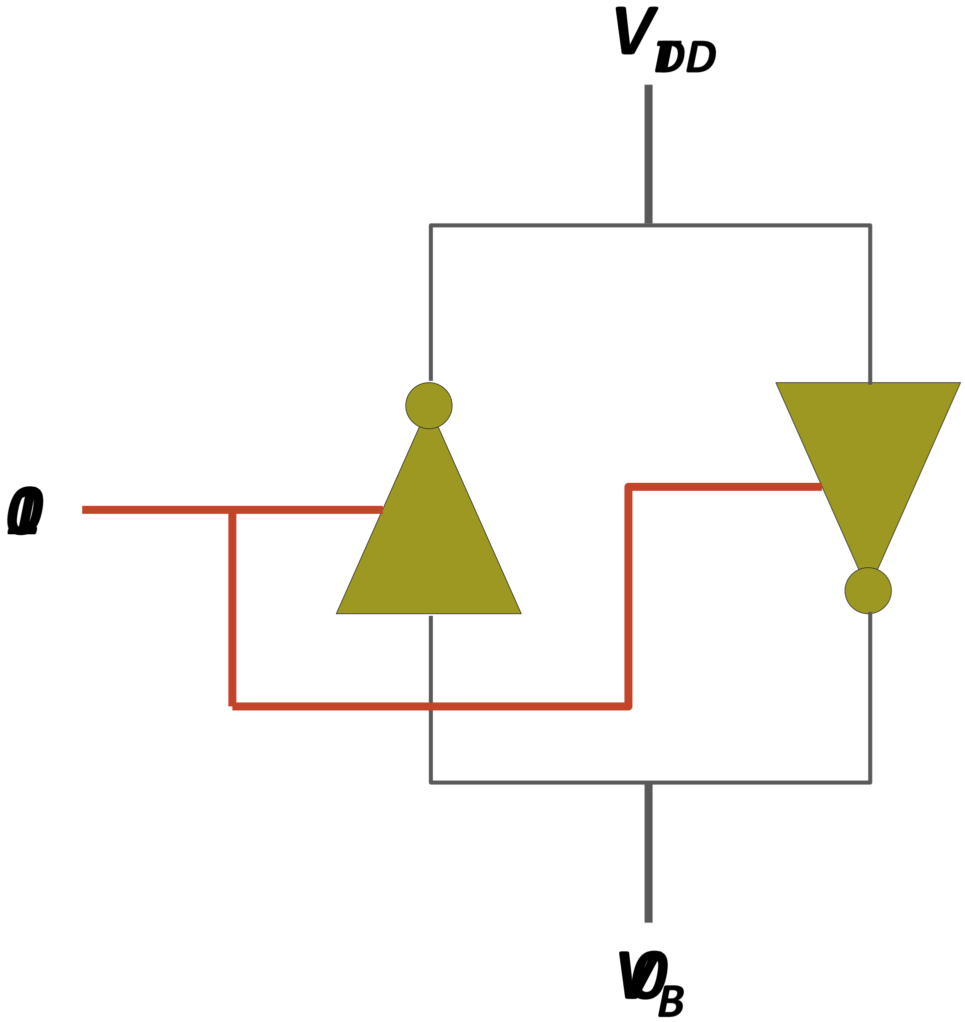


Logical “1”



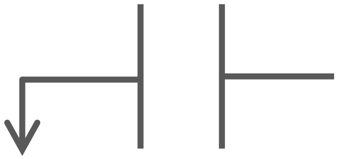
Logical “0”

Sense Amplifier Operation



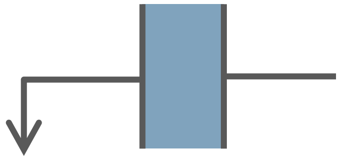
$V_T > V_B$

DRAM Cell – Capacitor



Empty State

Logical "0"

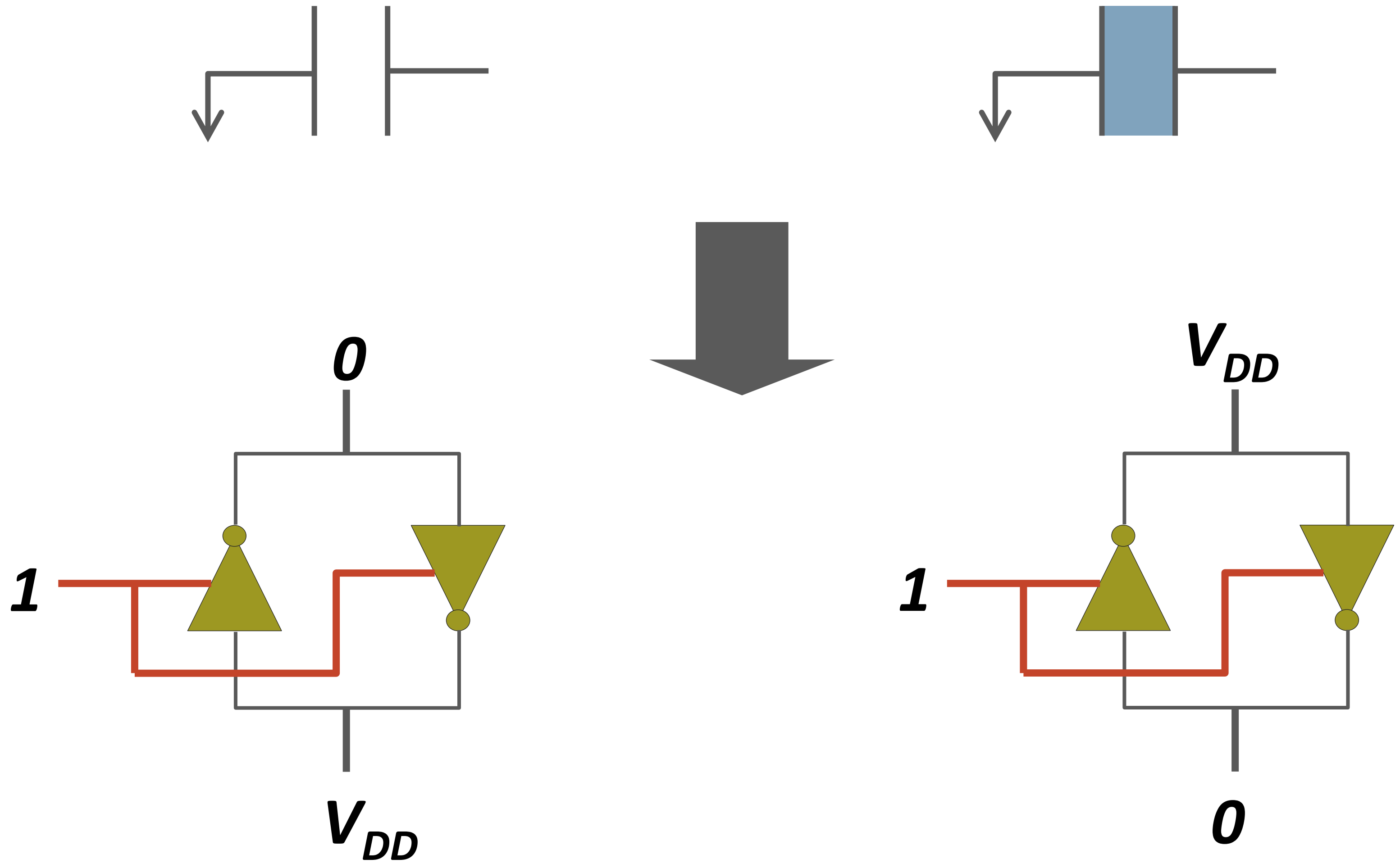


Fully Charged State

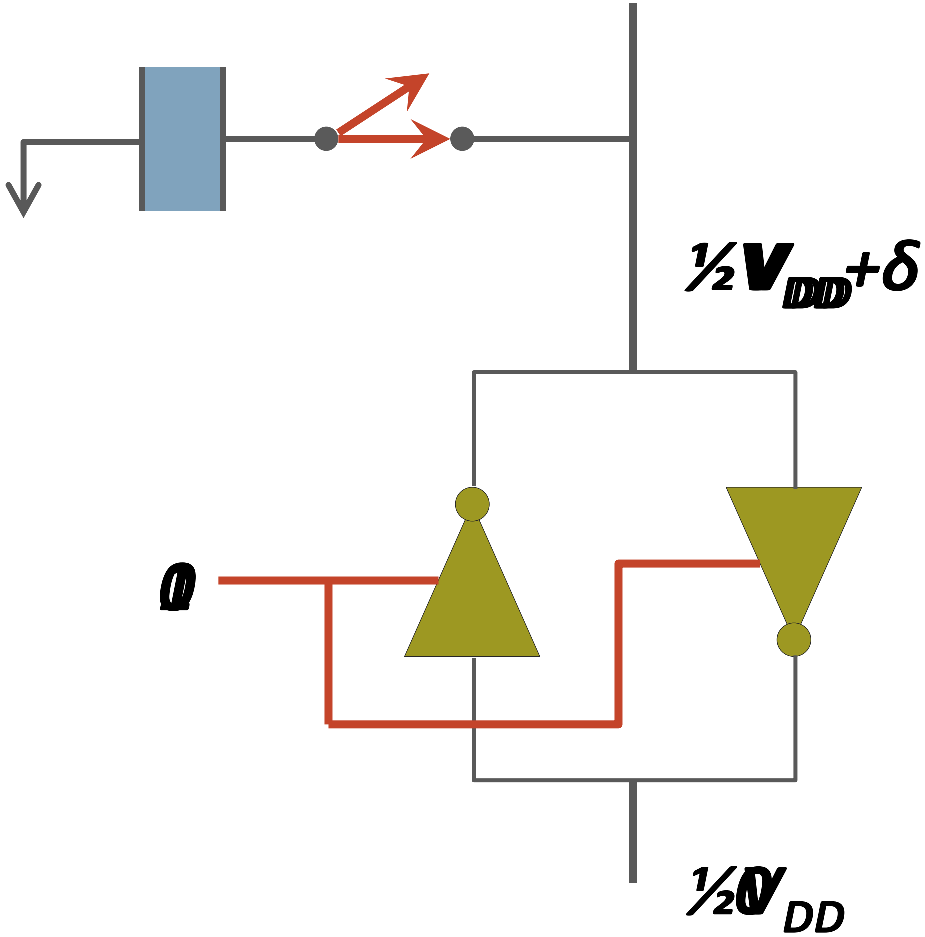
Logical "1"

1. Small – Cannot drive circuits
2. Reading destroys the state

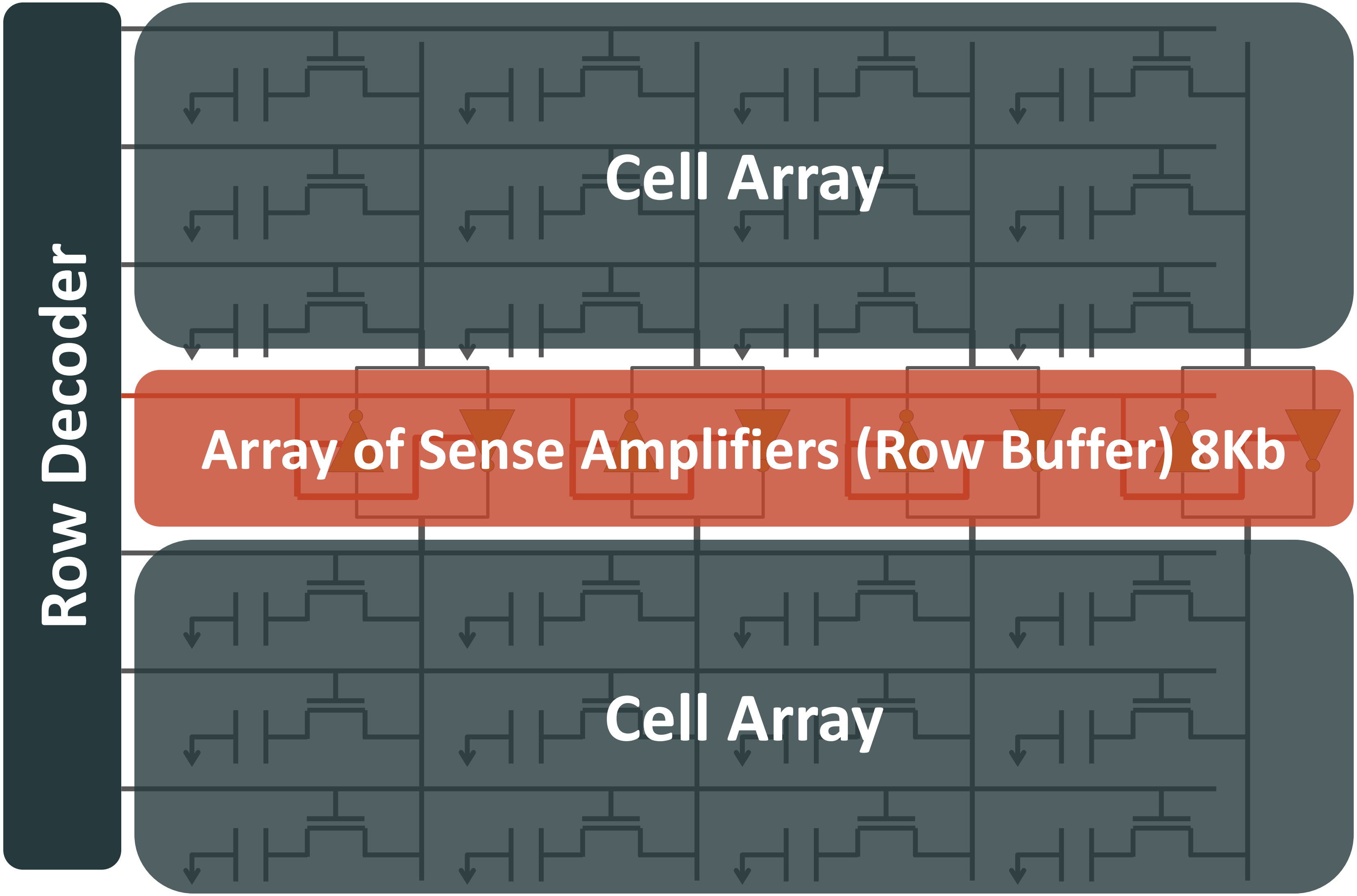
Capacitor to Sense Amplifier



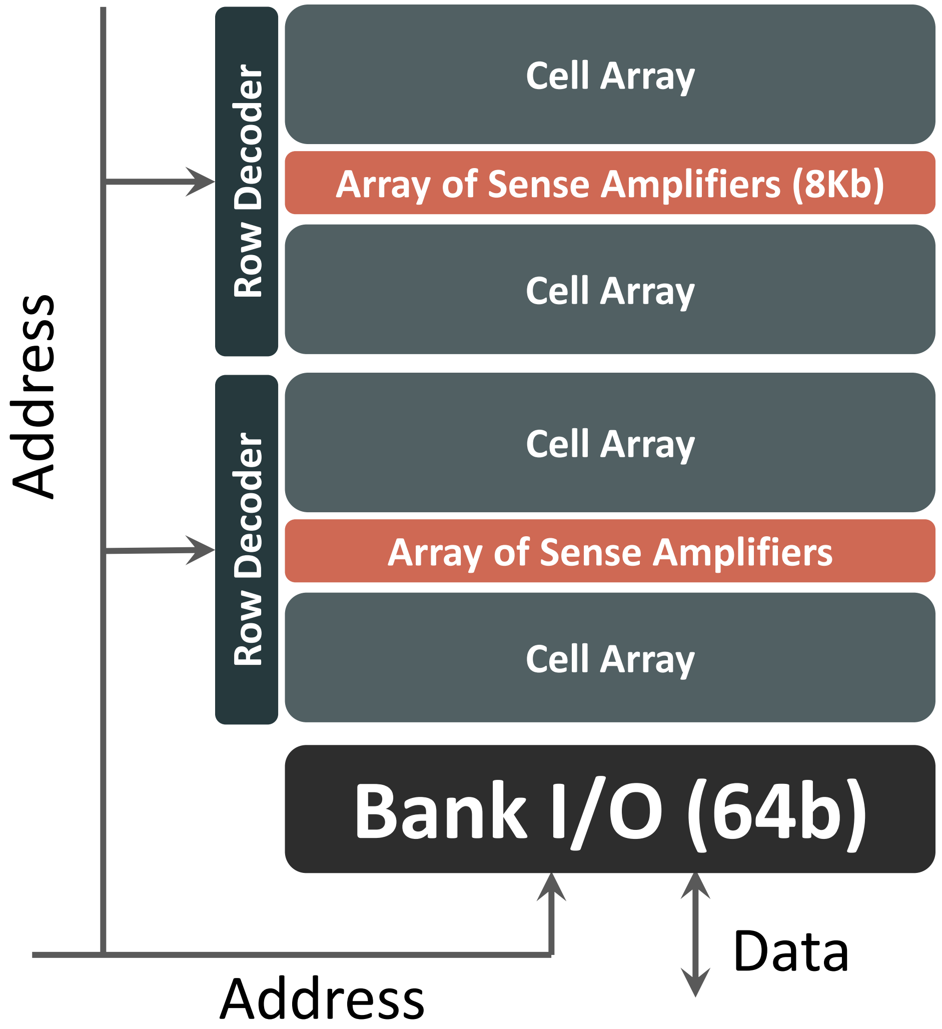
DRAM Cell Operation



DRAM Subarray – Building Block for DRAM Chip

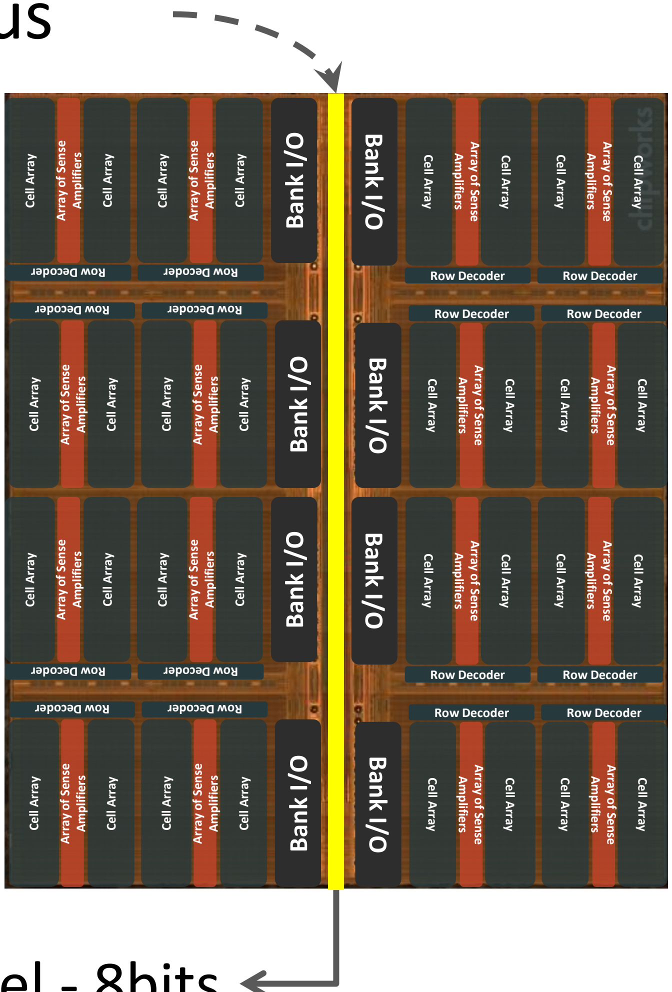


DRAM Bank


Address
Row Decoder
Cell Array
Array of Sense Amplifiers (8Kb)
Cell Array
Row Decoder
Cell Array
Array of Sense Amplifiers
Cell Array
Bank I/O (64b)
Address
Data

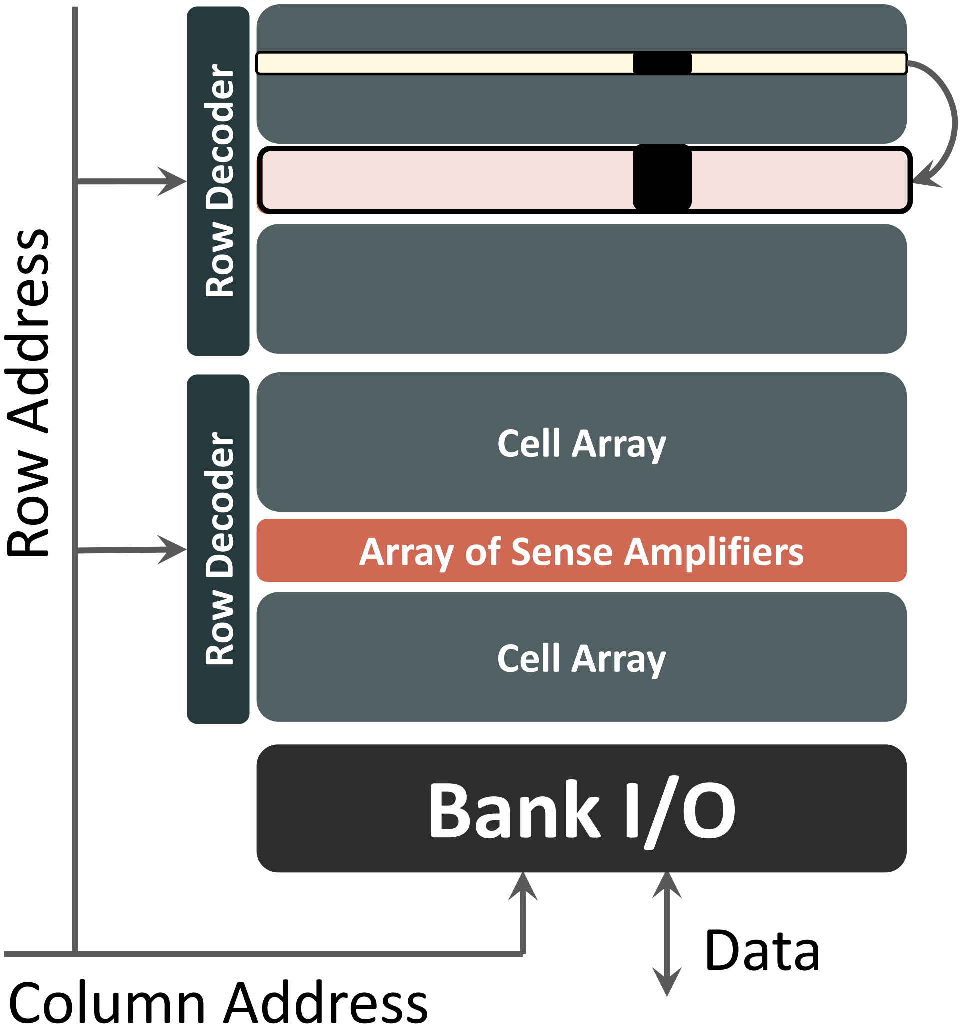
DRAM Chip

Shared internal bus



Memory channel - 8bits

DRAM Operation



1 ACTIVATE Row

2 READ/WRITE Column

3 PRECHARGE

More on DRAM Operation: Section 2

- Vivek Seshadri and Onur Mutlu,
  "In-DRAM Bulk Bitwise Execution Engine"
  Invited Book Chapter in Advances in Computers, to appear in 2020.
  [Preliminary arXiv version]

See Section 2 for comprehensive DRAM Background

In-DRAM Bulk Bitwise Execution Engine

Vivek Seshadri
Microsoft Research India
visesha@microsoft.com

Onur Mutlu
ETH Zürich
onur.mutlu@inf.ethz.ch

https://arxiv.org/pdf/1905.09822.pdf