

Memory Systems and Memory-Centric Computing Systems

Part 1: Memory Importance and Trends

Prof. Onur Mutlu

omutlu@gmail.com

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

7 July 2019

SAMOS Tutorial

SAFARI

ETH zürich

Carnegie Mellon

Brief Self Introduction



■ Onur Mutlu

- ❑ Full Professor @ ETH Zurich CS (EE), since September 2015
- ❑ Strecker Professor @ Carnegie Mellon University ECE/CS, 2009-2016, 2016-...
- ❑ PhD from UT-Austin, worked at Google, VMware, Microsoft Research, Intel, AMD
- ❑ <https://people.inf.ethz.ch/omutlu/>
- ❑ omutlu@gmail.com (Best way to reach me)
- ❑ <https://people.inf.ethz.ch/omutlu/projects.htm>

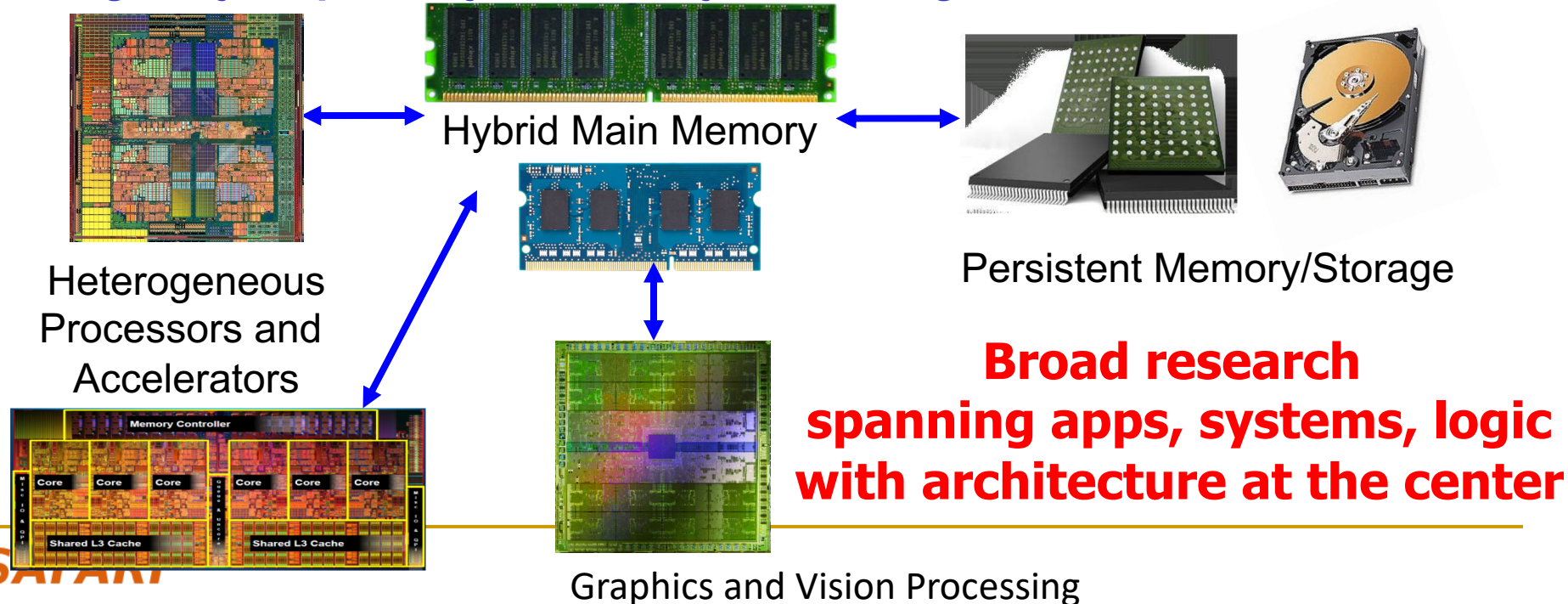
■ Research and Teaching in:

- ❑ Computer architecture, computer systems, hardware security, bioinformatics
- ❑ Memory and storage systems
- ❑ Hardware security, safety, predictability
- ❑ Fault tolerance
- ❑ Hardware/software cooperation
- ❑ Architectures for bioinformatics, health, medicine
- ❑ ...

Current Research Focus Areas

Research Focus: Computer architecture, HW/SW, bioinformatics, security

- Memory and storage (DRAM, flash, emerging) interconnects
- Heterogeneous & parallel systems, GPUs, systems for data analytics
- System/architecture interaction, new execution models, new interfaces
- Hardware security, energy efficiency, fault tolerance, performance
- Genome sequence analysis & assembly algorithms and architectures
- Biologically inspired systems & system design for bio/medicine



Four Key Directions

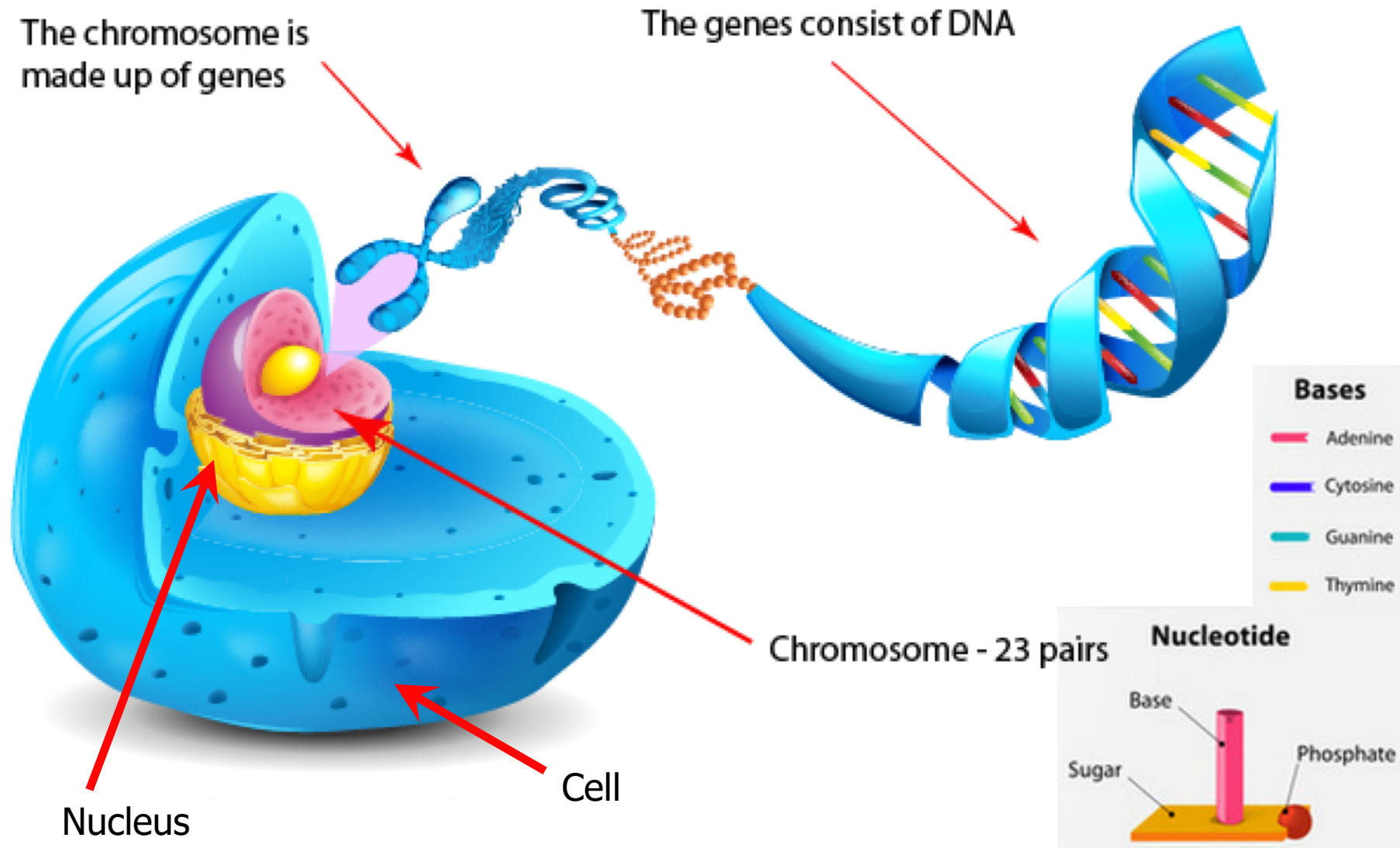
- Fundamentally **Secure/Reliable/Safe** Architectures
- Fundamentally **Energy-Efficient** Architectures
 - **Memory-centric** (Data-centric) Architectures
- Fundamentally **Low-Latency** Architectures
- Architectures for **Genomics, Medicine, Health**

A Motivating Detour: Genome Sequence Analysis

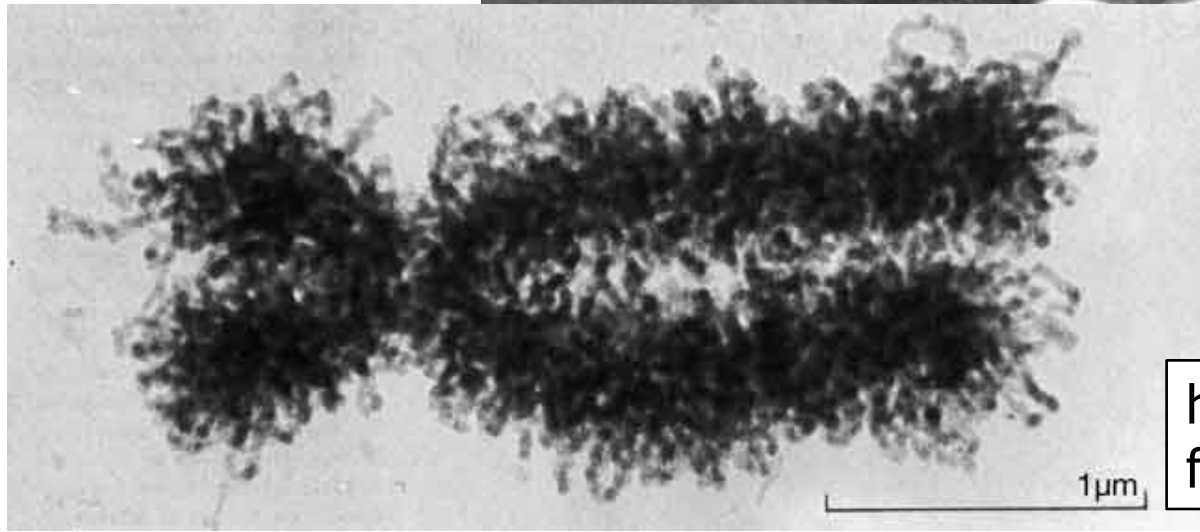
Our Dream (circa 2007)

- An embedded device that can perform comprehensive genome analysis in real time (within a minute)
 - Which of these DNAs does this DNA segment match with?
 - What is the likely genetic disposition of this patient to this drug?
 - . . .

What Is a Genome Made Of?



DNA Under Electron Microscope



human chromosome #12
from HeLa's cell

DNA Sequencing

- Goal:

- Find the complete sequence of A, C, G, T's in DNA.

- Challenge:

- There is no machine that takes long DNA as an input, and gives the complete sequence as output
- All sequencing machines chop DNA into pieces and identify relatively small pieces (but not how they fit together)

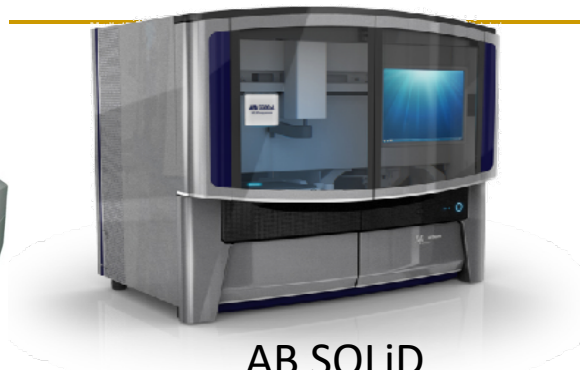
Untangling Yarn Balls & DNA Sequencing



Genome Sequencers



Roche/454



AB SOLiD



Illumina MiSeq



Complete Genomics



Illumina HiSeq2000



Pacific Biosciences RS



Oxford Nanopore MinION



Illumina NovaSeq 6000



SAFARI

Ion Torrent PGM



Ion Torrent Proton

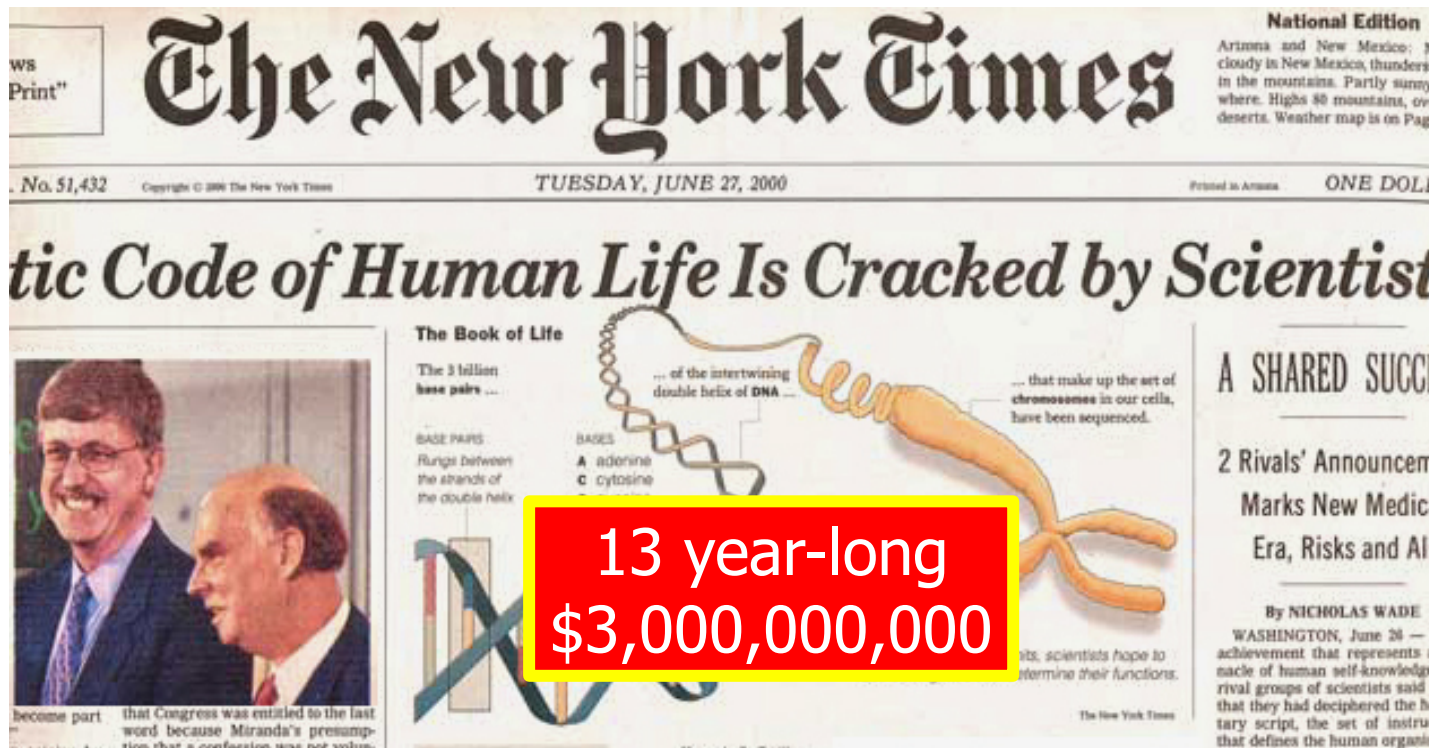


Oxford Nanopore GridION

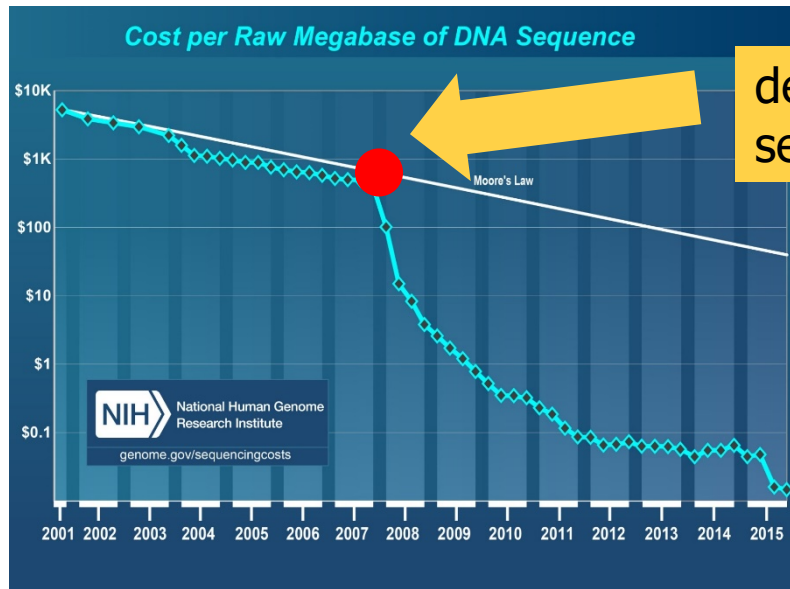
... and more! All produce data with different properties.

The Genomic Era

- 1990-2003: The Human Genome Project (HGP) provides a complete and accurate sequence of all **DNA base pairs** that make up the human genome and finds 20,000 to 25,000 human genes.

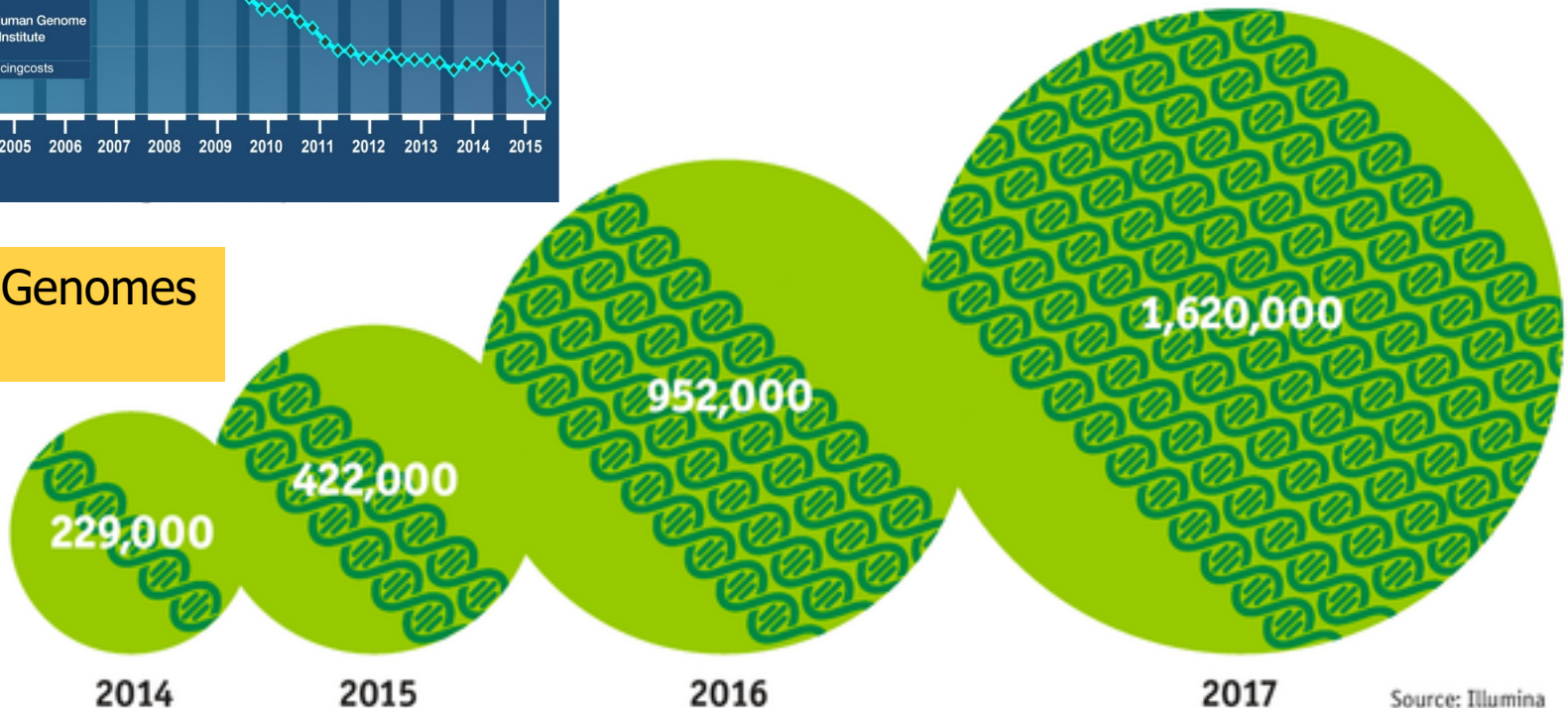


The Genomic Era (continued)



development of high-throughput sequencing (HTS) technologies

Number of Genomes Sequenced

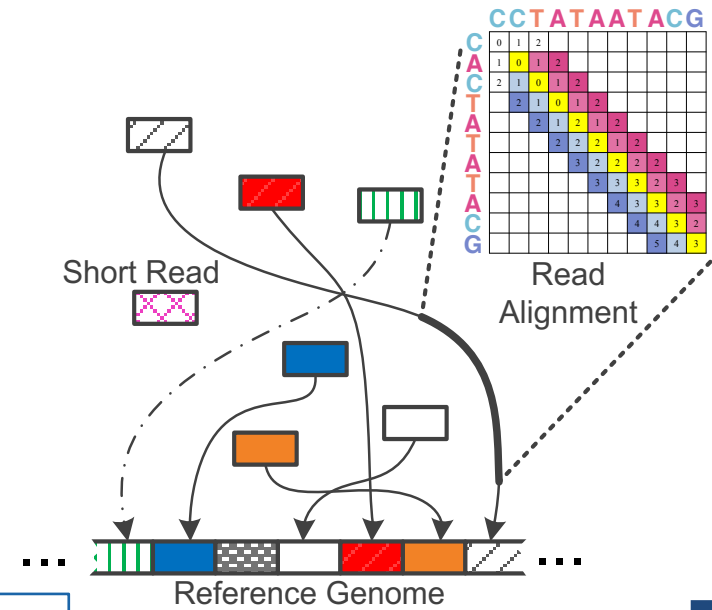
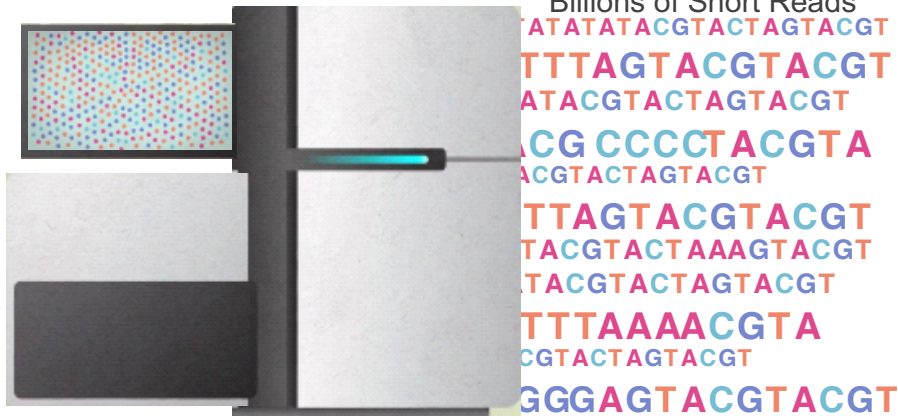


The Economist

SAFARI

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

Source: Illumina



1 Sequencing

Genome Analysis

2 Read Mapping

reference: TTTATCGCTTCCATGACGCAG

read1: ATCGCATCC

read2: TATCGCATC

read3: CATCCATGA

read4: CGCTTCCAT

read5: CCATGACGC

read6: TTCCATGAC



3 Variant Calling

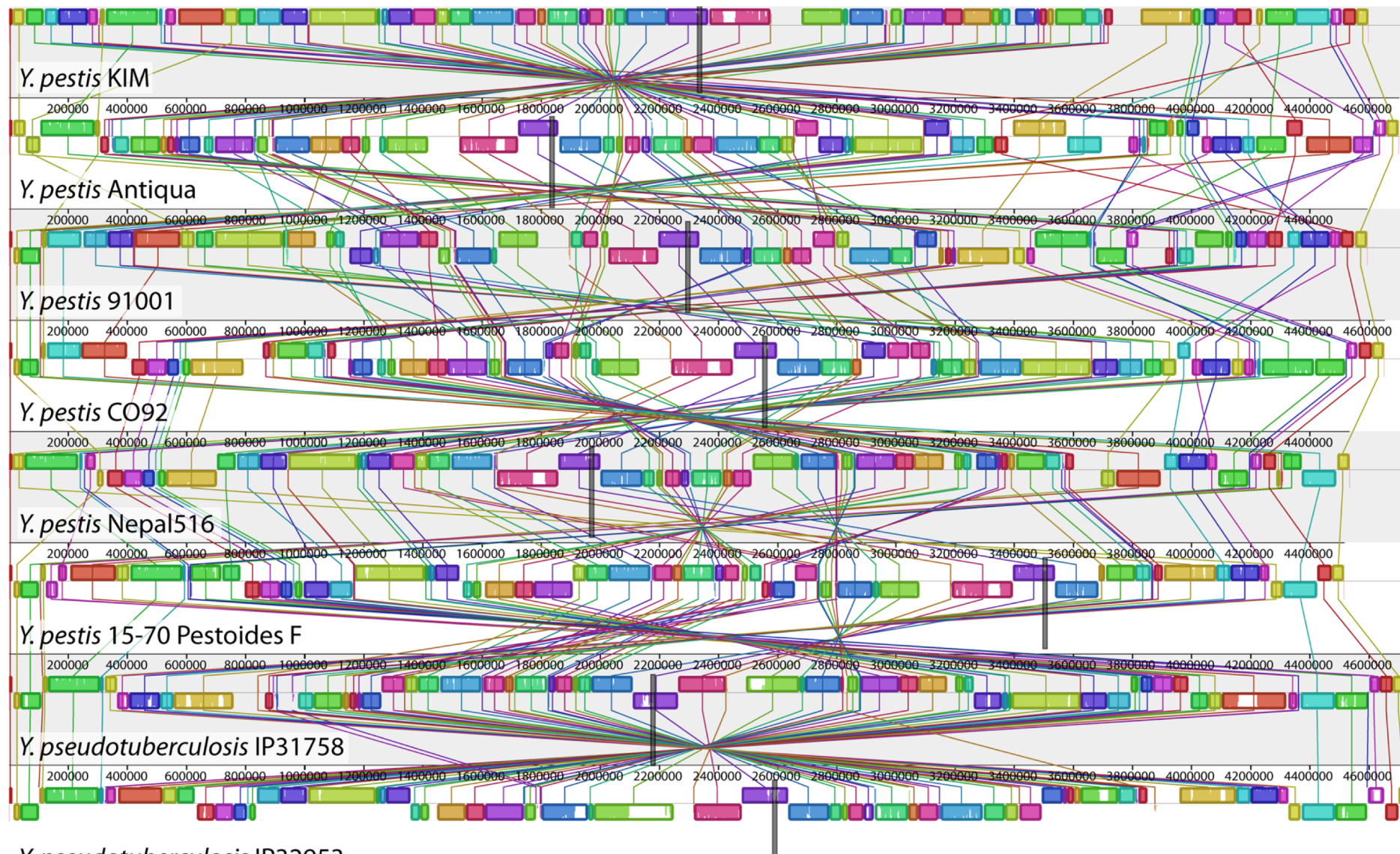
4 Scientific Discovery

Multiple sequence alignment

PHDHtm			-----MMMMMMMMMMMMMMMMMMMM-----	
16082665	<i>T acid</i>	10	----MASDRKSEGFQSGAGLIRYFEEEEIKGPALDPKLVVYMGIAVAIIVEIAKIFWFP---	(55)
13541150	<i>T volc</i>	10	----MASDRKSEGFQSGAGLIRYFEEEEIKGPALDPKLVVYIGIAVAIMVELAKIFWFP---	(55)
RFAC01077	<i>F acid</i>	13	-MTSMAKDNQNFQSGAGLIRYFNEEEIKGPAIDPKLIITYIGIAMGVIVELAKVFWFPV---	(58)
15791336	<i>H NRC1</i>	10	----MSSGQNSGGLMSSSAGLVRYFDSEDSNALQIDPRSVVAVGAFFGLVLVLLAQFFA----	(53)
RAG22196	<i>A fulg</i>	14	MAKAPKKGAKTPPLMSSSAGIMRYFEE-EKTQIKVSPKTI LAAGIVTGVLI IILNAYYGLWP-	(68)
RPO01000	<i>P abys</i>	9	-----MAKEKTTLPPTGAGLMRFFDE-DTRAIKITPKGAVALTLILIIIFEIILHVVGPRIFG	(56)
RPH01741	<i>P hori</i>	9	-----MAKEKTTLPPTGAGLMRFFDE-DTRAIKITPKGAIALVLILIIIFEIILHVVGPRIFG	(56)
AE000914	<i>M ther</i>	10	----MAKKDKKTLPPSGAGLVRYFEE-ETKGEKLTPEQVVVMSIILAVFCLVLRFSG----	(52)
RMJ09857	<i>M jann</i>	9	-----MSKREESTGLATSAGLIRYMDE-TFSKIRVKPEHVIGVTVAFVIIIEAILTYGRFL---	(53)
15920503	<i>S toko</i>	13	-MPSSKKKKSTVPLASMAGLIRYYEE-ENEKIKISPKLLIIISIMVAGVIVASILIPPP--	(58)
AE006662	<i>S solf</i>	11	-MPSSKKKKSTVPMVMAGLIRYYEE-ENEKVKISPKIVIGASLALTIIIVIVITKLF-----	(55)
RPK02491	<i>P aero</i>	12	--MARRRKYEGLNPFVAAGLIKFSSEGELEKIKLTPRAAVVISLAIIGLLIAINLLLPL--	(58)
RAP00437	<i>A pern</i>	13	-MSVRRRRRERRATPVTAAGLLSFYEE-YEGKIKISPTIVVGAILVSAVVAABIFLPAVP-	(59)
5803165	<i>H sapi</i>	49	-----SAGTGGMWRFYTE-DSPGLKVGVPVFLVMSLLFIASVFMLHIWGTKYTRS	(96)
13324684	<i>M musc</i>	49	-----SAGTGGMWRFYTE-DSPGLKVGVPVFLVMSLLFIAAVFMLHIWGTKYTRS	(96)
6002114	<i>D mela</i>	53	-----GAGTGGMWRFYTD-DSPGIRKVGVPVFLVMSLLFIASVFMLHIWGTKYNRS	(100)
14574310	<i>C eleg</i>	32	-----GGNNGGLWRFYTE-DSTGLKIGVPVFLVMSLVFIASVFVLHIWGTKFTRS	(81)
10697176	<i>Y lipo</i>	41	-----GGSSSTMLKLYTD-ESQGLKVDPVVVMVLSLGFIFSVVALHLLAKVSTK	(91)
6320857	<i>S cere</i>	40	-----GGSSSSILKLYTD-EANGFRVDSLVLFLSVGFIFSVIALHLLTKFTHI	(88)
6320932	<i>S cere</i>	33	-----TNSNNSILKIYSD-EATGLRVDPLVLFLAVGFIFSVVALHVISKVAGK	(82)

Example Question: If I give you a bunch of sequences, tell me where they are the same and where they are different.

Genome Sequence Alignment: Example



Source: By Aaron E. Darling, István Miklós, Mark A. Ragan - Figure 1 from Darling AE, Miklós I, Ragan MA (2008).

"Dynamics of Genome Rearrangement in Bacterial Populations". PLOS Genetics. DOI:10.1371/journal.pgen.1000128., CC BY 2.5, <https://commons.wikimedia.org/w/index.php?curid=30550950>

The Genetic Similarity Between Species



Human ~ Human
99.9%



Human ~ Chimpanzee
96%



Human ~ Cat
90%



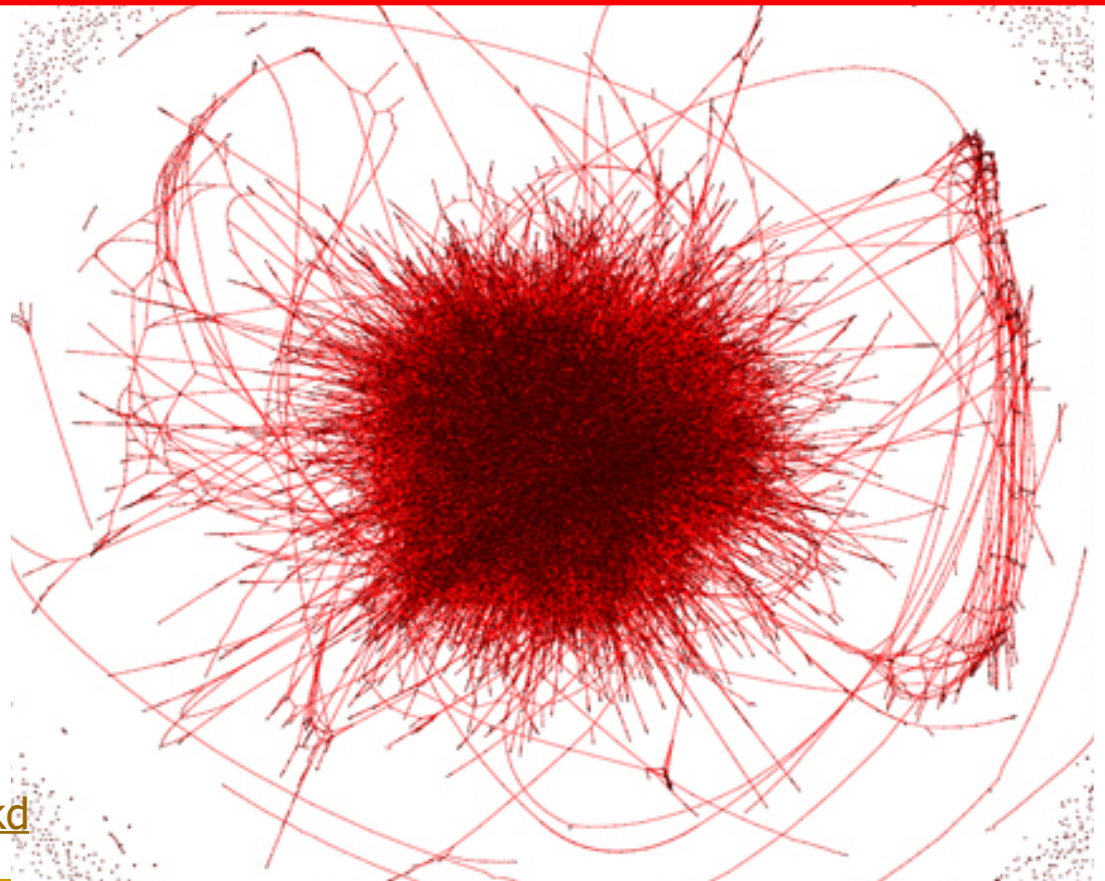
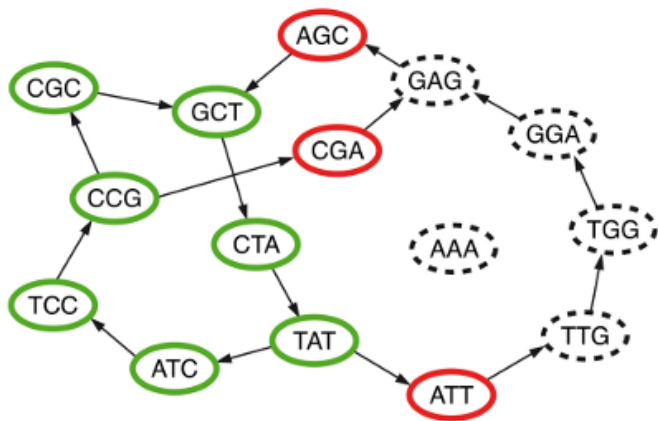
Human ~ Cow
80%



Human ~ Banana
50-60%

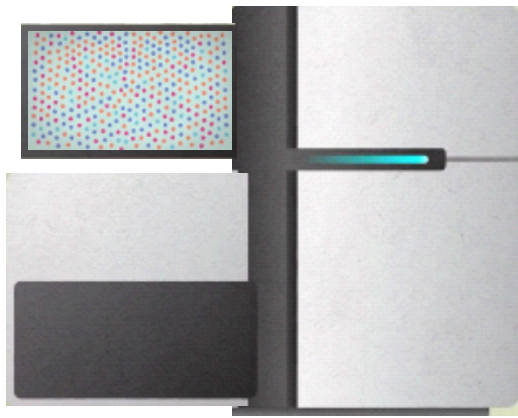
Metagenomics, genome assembly, de novo sequencing

**Question 2: Given a bunch of short sequences,
Can you identify the approximate species cluster
for genomically unknown organisms?**



uncleaned de Bruijn graph

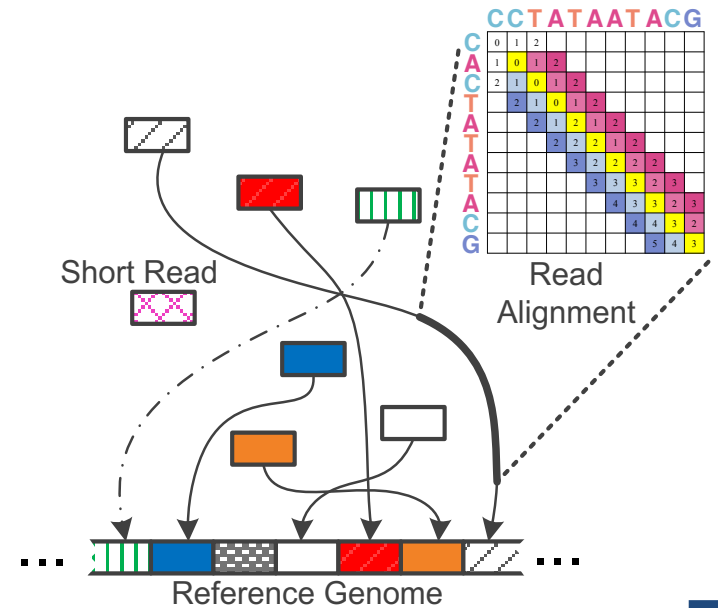
<http://math.oregonstate.edu/~koslickd/>



Billions of Short Reads

ATATATACGTACTAGTACGT
 TTTAGTACGTACGT
 ATACGTACTAGTACGT
 CGCCCCTACGTA
 ACGTACTAGTACGT
 TTAGTACGTACGT
 TACGTACTAAAGTACGT
 TACGTACTAGTACGT
 TTTAAACGTA
 CGTACTAGTACGT
 GGGAGTACGTACGT

1 Sequencing



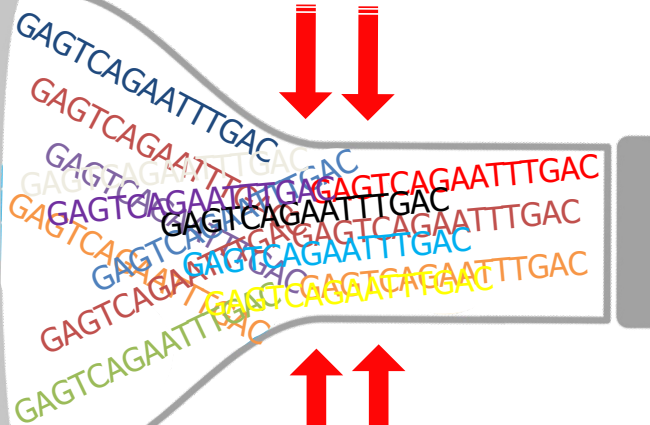
Read Mapping 2

Bottlenecked in Mapping!!

Illumina HiSeq4000

300 M

bases/min



on average

2 M

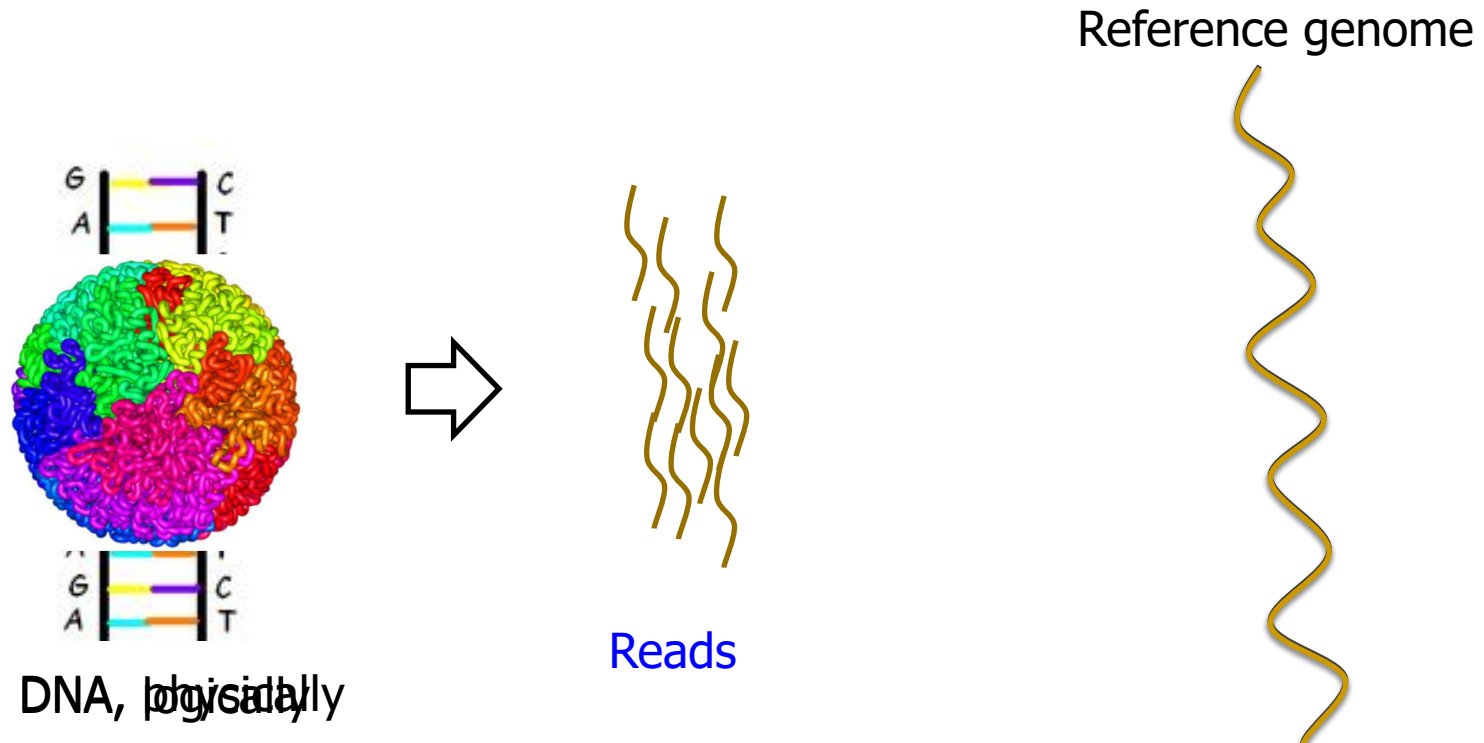
bases/min

(0.6%)

**Need to construct
the entire genome
from many reads**

Read Mapping

- Map many short DNA fragments (**reads**) to a known reference genome with some differences allowed



Mapping short reads to reference genome is challenging (billions of 50-300 base pair reads)

Read Alignment/Verification

- **Edit distance** is defined as the minimum number of edits (i.e. insertions, deletions, or substitutions) needed to make the read exactly match the reference segment.

NETHERLANDS x SWITZERLAND

N	E	-	T	H	E	R	L	A	N	D	S
S	W	I	T	Z	E	R	L	A	N	D	-

match
deletion
insertion
mismatch

Challenges in Read Mapping

- Need to find many mappings of each read
 - How can we find all mappings efficiently?
- Need to tolerate small variances/errors in each read
 - Each individual is different: Subject's DNA may slightly differ from the reference (Mismatched, insertions, deletions)
 - How can we efficiently map each read with up to e errors present?
- Need to map each read very fast (i.e., performance is important)
 - Human DNA is 3.2 billion base pairs long → Millions to billions of reads (State-of-the-art mappers take weeks to map a human's DNA)
 - How can we design a much higher performance read mapper?

Our First Step: Comprehensive Mapping

- + Guaranteed to find *a//* mappings → sensitive
- + Can tolerate up to *e* errors

nature
genetics

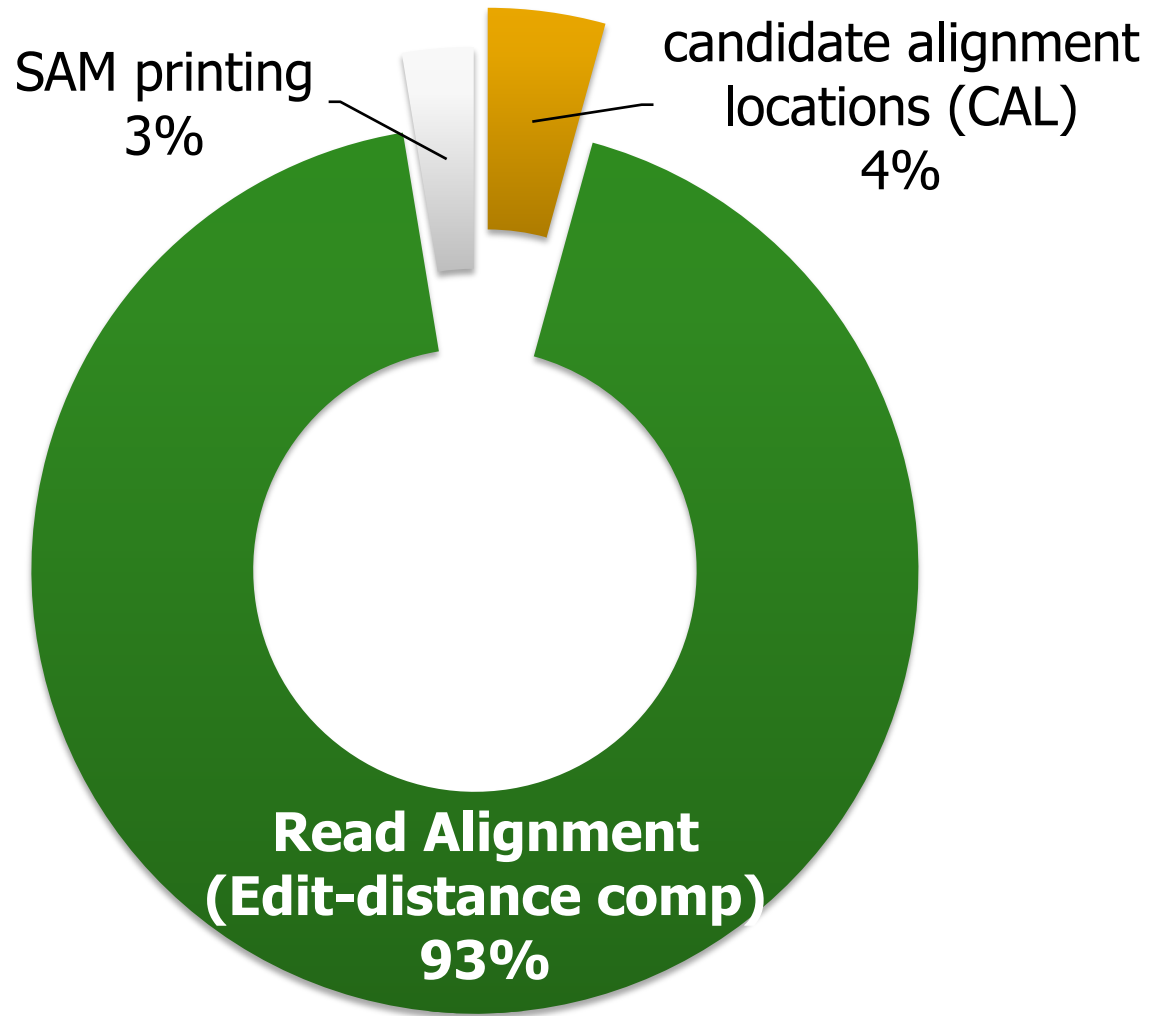
<http://mrfast.sourceforge.net/>

Personalized copy number and segmental duplication maps using next-generation sequencing

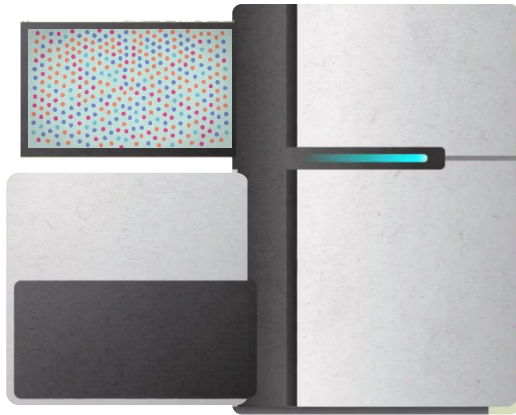
Can Alkan^{1,2}, Jeffrey M Kidd¹, Tomas Marques-Bonet^{1,3}, Gozde Aksay¹, Francesca Antonacci¹, Fereydoun Hormozdiari⁴, Jacob O Kitzman¹, Carl Baker¹, Maika Malig¹, Onur Mutlu⁵, S Cenk Sahinalp⁴, Richard A Gibbs⁶ & Evan E Eichler^{1,2}

Alkan+, "**Personalized copy number and segmental duplication maps using next-generation sequencing**", Nature Genetics 2009.

Read Mapping Execution Time Breakdown



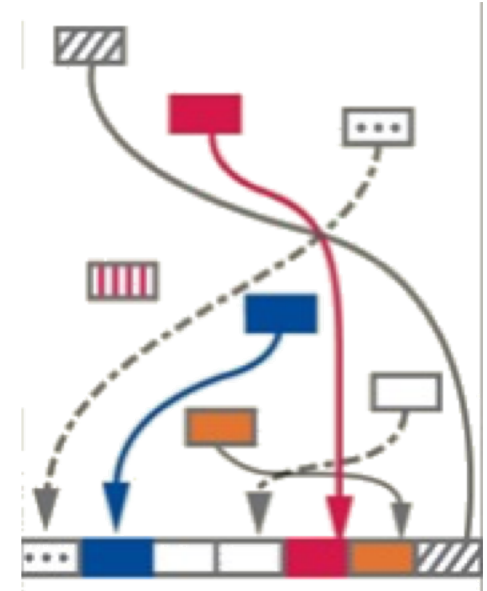
The Read Mapping Bottleneck



Illumina HiSeq4000

ACGTACGTACGTACGT
CCCCCCTATATATACGTACTAGTACGT
CGACTTTAGTACGTACGT
TATATATACGTACTAGTACGT
ACGTACGCCCCGTACGTA
TATATATACGTACTAGTACGT
GACTTTAGTACGTACGT
TATATATACGTACTAAAGTACGT
TATATATACGTACTAGTACGT
CGTTTTTAAACGTA
ATATATACGTACTAGTACGT
GACGGGGAGTACGTACGT
TATATATACGTACTAAAGTACGT

300 Million
bases/minute



2 Million
bases/minute

150X slower

Filter fast before you align

Minimize costly

“approximate string comparisons”

Our First Filter: Pure Software Approach

- Download the source code and try for yourself
 - [Download link to FastHASH](#)

Xin *et al.* *BMC Genomics* 2013, **14**(Suppl 1):S13
<http://www.biomedcentral.com/1471-2164/14/S1/S13>



PROCEEDINGS

Open Access

Accelerating read mapping with FastHASH

Hongyi Xin¹, Donghyuk Lee¹, Farhad Hormozdiari², Samihan Yedkar¹, Onur Mutlu^{1*}, Can Alkan^{3*}

From The Eleventh Asia Pacific Bioinformatics Conference (APBC 2013)
Vancouver, Canada. 21-24 January 2013

Shifted Hamming Distance: SIMD Acceleration

<https://github.com/CMU-SAFARI/Shifted-Hamming-Distance>

Bioinformatics, 31(10), 2015, 1553–1560

doi: 10.1093/bioinformatics/btu856

Advance Access Publication Date: 10 January 2015

Original Paper

OXFORD

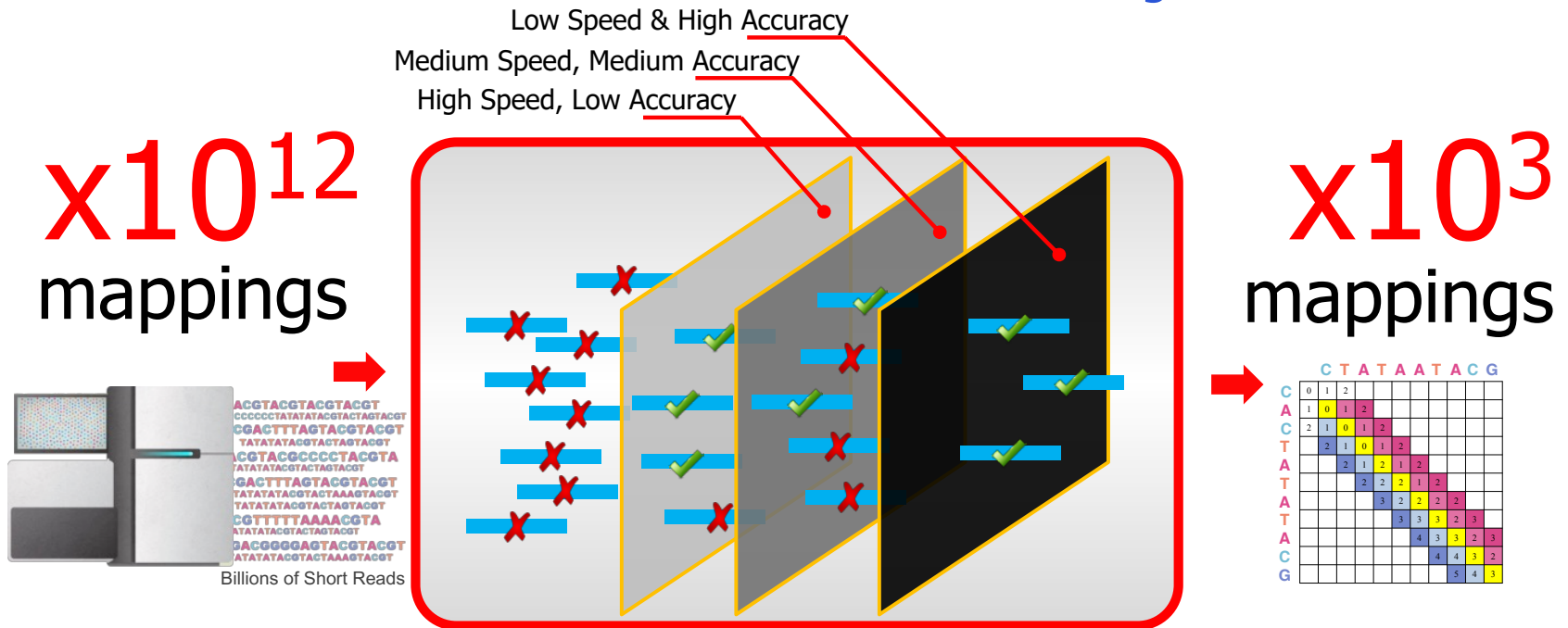
Sequence analysis

Shifted Hamming distance: a fast and accurate SIMD-friendly filter to accelerate alignment verification in read mapping

Hongyi Xin^{1,*}, John Greth², John Emmons², Gennady Pekhimenko¹,
Carl Kingsford³, Can Alkan^{4,*} and Onur Mutlu^{2,*}

Xin+, **"Shifted Hamming Distance: A Fast and Accurate SIMD-friendly Filter to Accelerate Alignment Verification in Read Mapping", *Bioinformatics* 2015.**

GateKeeper: FPGA-Based Alignment Filtering



- 1 High throughput DNA sequencing (HTS) technologies
- 2 Read Pre-Alignment Filtering
Fast & Low False Positive Rate
- 3 Read Alignment
Slow & Zero False Positives

GateKeeper: FPGA-Based Alignment Filtering

- Mohammed Alser, Hasan Hassan, Hongyi Xin, Oguz Ergin, Onur Mutlu, and Can Alkan
"GateKeeper: A New Hardware Architecture for Accelerating Pre-Alignment in DNA Short Read Mapping"
Bioinformatics, [published online, May 31], 2017.
[[Source Code](#)]
[[Online link at Bioinformatics Journal](#)]

GateKeeper: a new hardware architecture for accelerating pre-alignment in DNA short read mapping

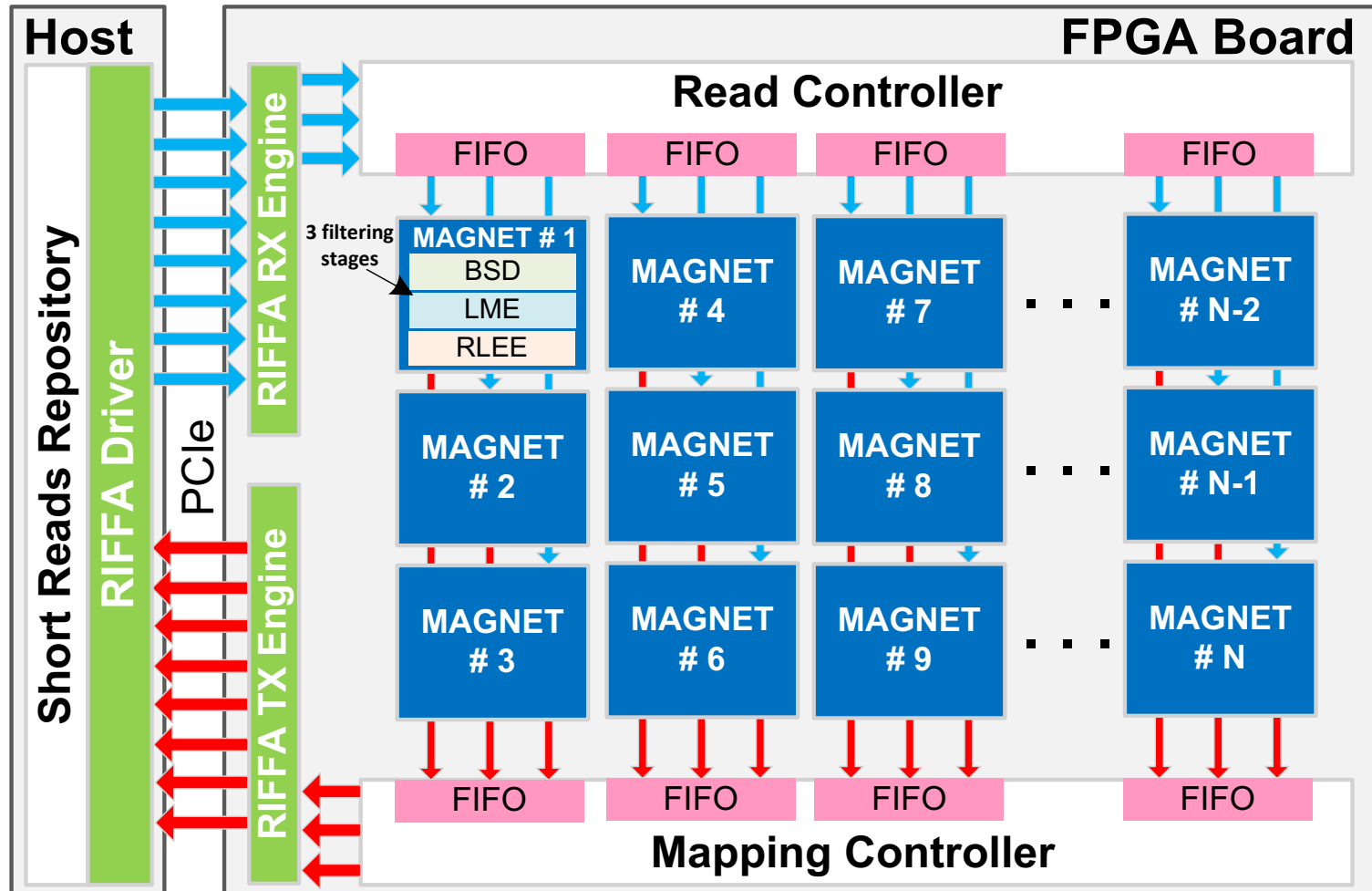
Mohammed Alser ✉, Hasan Hassan, Hongyi Xin, Oğuz Ergin, Onur Mutlu ✉, Can Alkan ✉

Bioinformatics, Volume 33, Issue 21, 1 November 2017, Pages 3355–3363,

<https://doi.org/10.1093/bioinformatics/btx342>

Published: 31 May 2017 **Article history** ▼

MAGNET Accelerator [Alser+, TIR 2017]



Newest Work: Shouji [Alser+, Bioinformatics 2019]

Mohammed Alser, Hasan Hassan, Akash Kumar, Onur Mutlu, and Can Alkan,
"Shouji: A Fast and Efficient Pre-Alignment Filter for Sequence Alignment"
Bioinformatics, [published online, March 28], 2019.

[\[Source Code\]](#)

[\[Online link at Bioinformatics Journal\]](#)

Bioinformatics, 2019, 1–9

doi: 10.1093/bioinformatics/btz234

Advance Access Publication Date: 28 March 2019

Original Paper

OXFORD

Sequence alignment

Shouji: a fast and efficient pre-alignment filter for sequence alignment

Mohammed Alser^{1,2,3,*}, Hasan Hassan¹, Akash Kumar², Onur Mutlu^{1,3,*}
and Can Alkan^{3,*}

¹Computer Science Department, ETH Zürich, Zürich 8092, Switzerland, ²Chair for Processor Design, Center For Advancing Electronics Dresden, Institute of Computer Engineering, Technische Universität Dresden, 01062 Dresden, Germany and ³Computer Engineering Department, Bilkent University, 06800 Ankara, Turkey

*To whom correspondence should be addressed.

Associate Editor: Inanc Birol

Received on September 13, 2018; revised on February 27, 2019; editorial decision on March 7, 2019; accepted on March 27, 2019

DNA Read Mapping & Filtering

- **Problem: Heavily bottlenecked by Data Movement**
- GateKeeper FPGA performance limited by DRAM bandwidth [Alser+, Bioinformatics 2017]
- Ditto for SHD on SIMD [Xin+, Bioinformatics 2015]
- **Solution: Processing-in-memory can alleviate the bottleneck**
- However, we need to design mapping & filtering algorithms to fit processing-in-memory

In-Memory DNA Sequence Analysis

- Jeremie S. Kim, Damla Senol Cali, Hongyi Xin, Donghyuk Lee, Saugata Ghose, Mohammed Alser, Hasan Hassan, Oguz Ergin, Can Alkan, and Onur Mutlu, **"GRIM-Filter: Fast Seed Location Filtering in DNA Read Mapping Using Processing-in-Memory Technologies"** ***BMC Genomics***, 2018.
Proceedings of the 16th Asia Pacific Bioinformatics Conference (APBC), Yokohama, Japan, January 2018.
[arxiv.org Version \(pdf\)](#)

GRIM-Filter: Fast seed location filtering in DNA read mapping using processing-in-memory technologies

Jeremie S. Kim^{1,6*}, Damla Senol Cali¹, Hongyi Xin², Donghyuk Lee³, Saugata Ghose¹, Mohammed Alser⁴, Hasan Hassan⁶, Oguz Ergin⁵, Can Alkan^{4*} and Onur Mutlu^{6,1*}

From The Sixteenth Asia Pacific Bioinformatics Conference 2018
Yokohama, Japan. 15-17 January 2018

Quick Note: Key Principles and Results

- Two key principles:
 - ❑ **Exploit the structure of the genome** to minimize computation
 - ❑ **Morph and exploit the structure of the underlying hardware** to maximize performance and efficiency
- **Algorithm-architecture co-design** for DNA read mapping
 - ❑ **Speeds up** read mapping by **~300X (sometimes more)**
 - ❑ **Improves accuracy** of read mapping in the presence of errors

Xin et al., "Accelerating Read Mapping with FastHASH," BMC Genomics 2013.

Xin et al., "Shifted Hamming Distance: A Fast and Accurate SIMD-friendly Filter to Accelerate Alignment Verification in Read Mapping," Bioinformatics 2015.

Alser et al., "GateKeeper: A New Hardware Architecture for Accelerating Pre-Alignment in DNA Short Read Mapping," Bioinformatics 2017.

Kim et al., "Genome Read In-Memory (GRIM) Filter," BMC Genomics 2018.

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.

[[Preliminary arxiv.org version](#)]

Nanopore Genome Assembly Pipeline

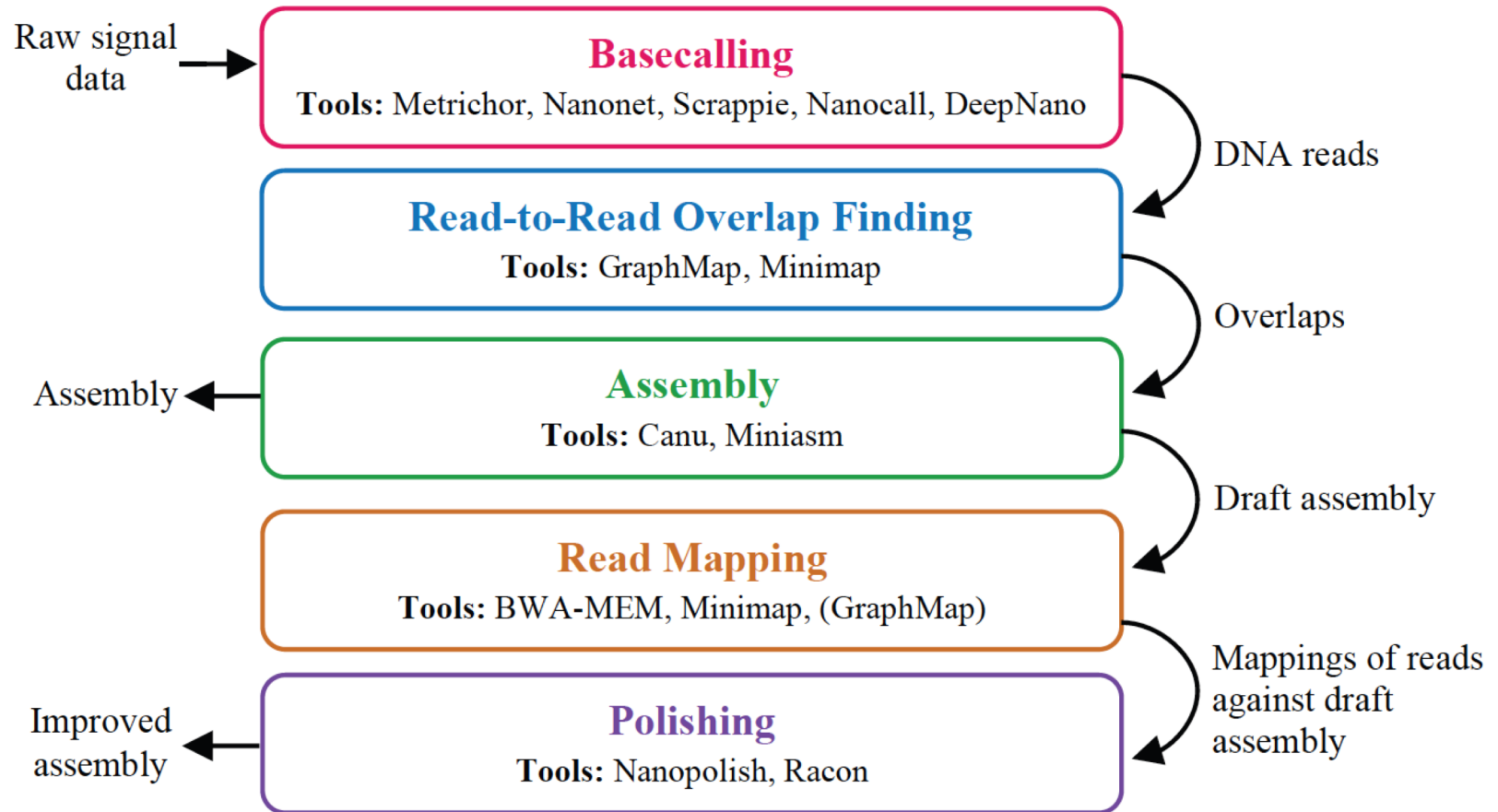


Figure 1. The analyzed genome assembly pipeline using nanopore sequence data, with its five steps and the associated tools for each step.

Recall Our Dream

- An embedded device that can perform comprehensive genome analysis in real time (within a minute)
- Still a long ways to go
 - Energy efficiency
 - Performance (latency)
 - Security
 - **Huge memory bottleneck**

More on Genome Analysis: Another Talk

- Onur Mutlu,
"Accelerating Genome Analysis: A Primer on an Ongoing Journey"
Keynote talk at *2nd Workshop on Accelerator Architecture in Computational Biology and Bioinformatics (AACBB)*, Washington, DC, USA, February 2019.
[[Slides \(pptx\)\(pdf\)](#)]
[[Video](#)]

Accelerating Genome Analysis A Primer on an Ongoing Journey

Onur Mutlu

omutlu@gmail.com

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

16 February 2019

AACBB Keynote Talk

SAFARI

ETH zürich

Carnegie Mellon

Four Key Directions

- Fundamentally Secure/Reliable/Safe Architectures
 - Fundamentally Energy-Efficient Architectures
 - Memory-centric (Data-centric) Architectures
 - Fundamentally Low-Latency Architectures
-
- Architectures for Genomics, Medicine, Health

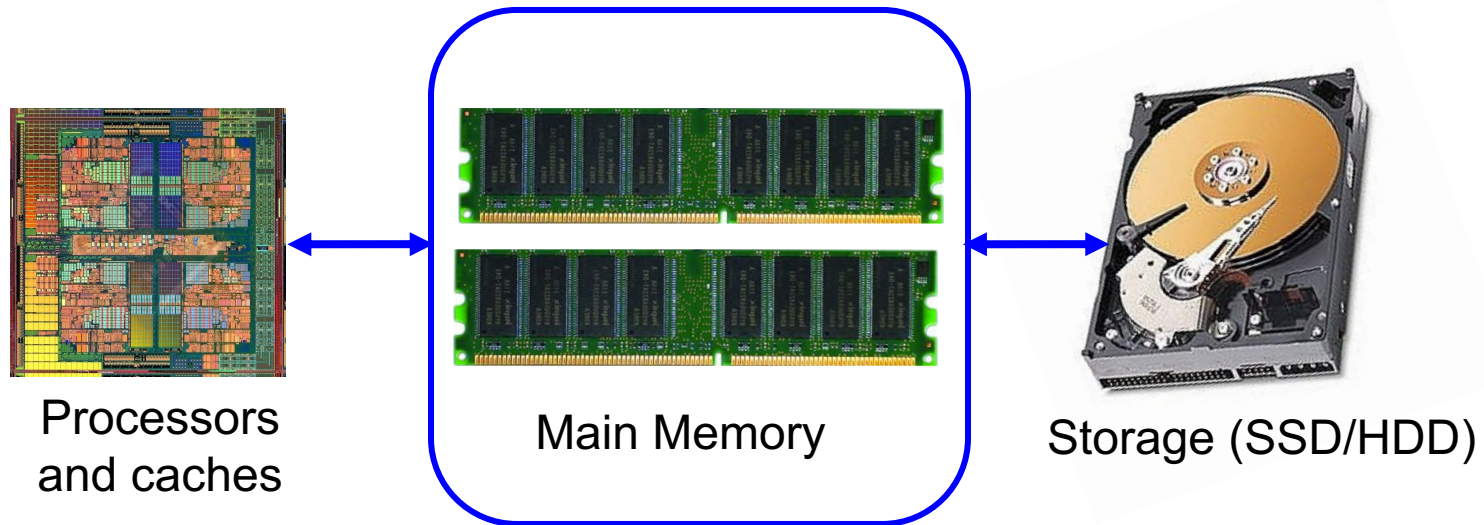
Memory & Storage

Why Is Memory So Important? (Especially Today)

Importance of Main Memory

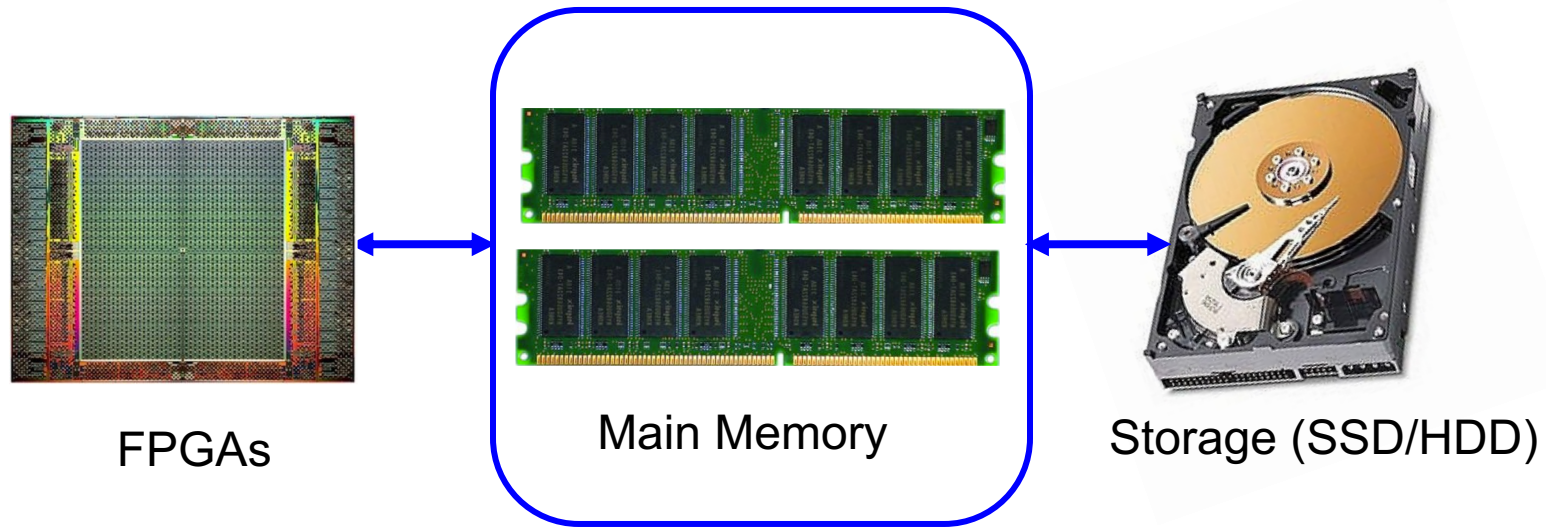
- The Performance Perspective
- The Energy Perspective
- The Scaling/Reliability/Security Perspective
- Trends/Challenges/Opportunities in Main Memory

The Main Memory System



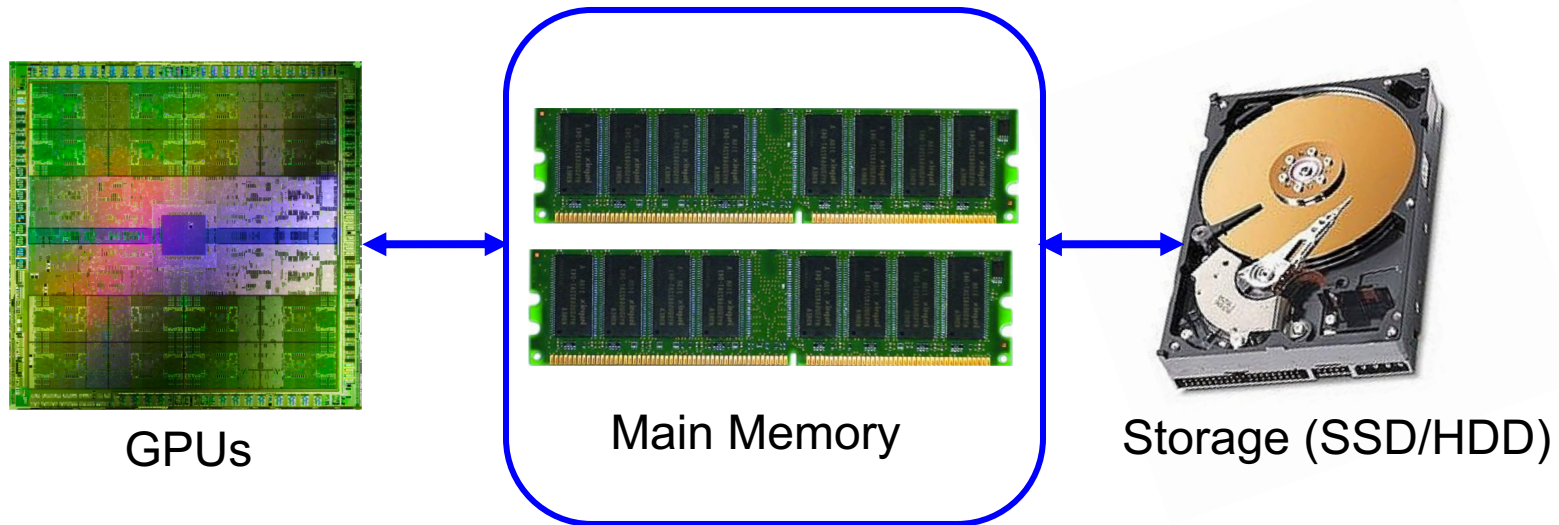
- Main memory is a critical component of all computing systems: server, mobile, embedded, desktop, sensor
- Main memory system must scale (in *size, technology, efficiency, cost, and management algorithms*) to maintain performance growth and technology scaling benefits

The Main Memory System



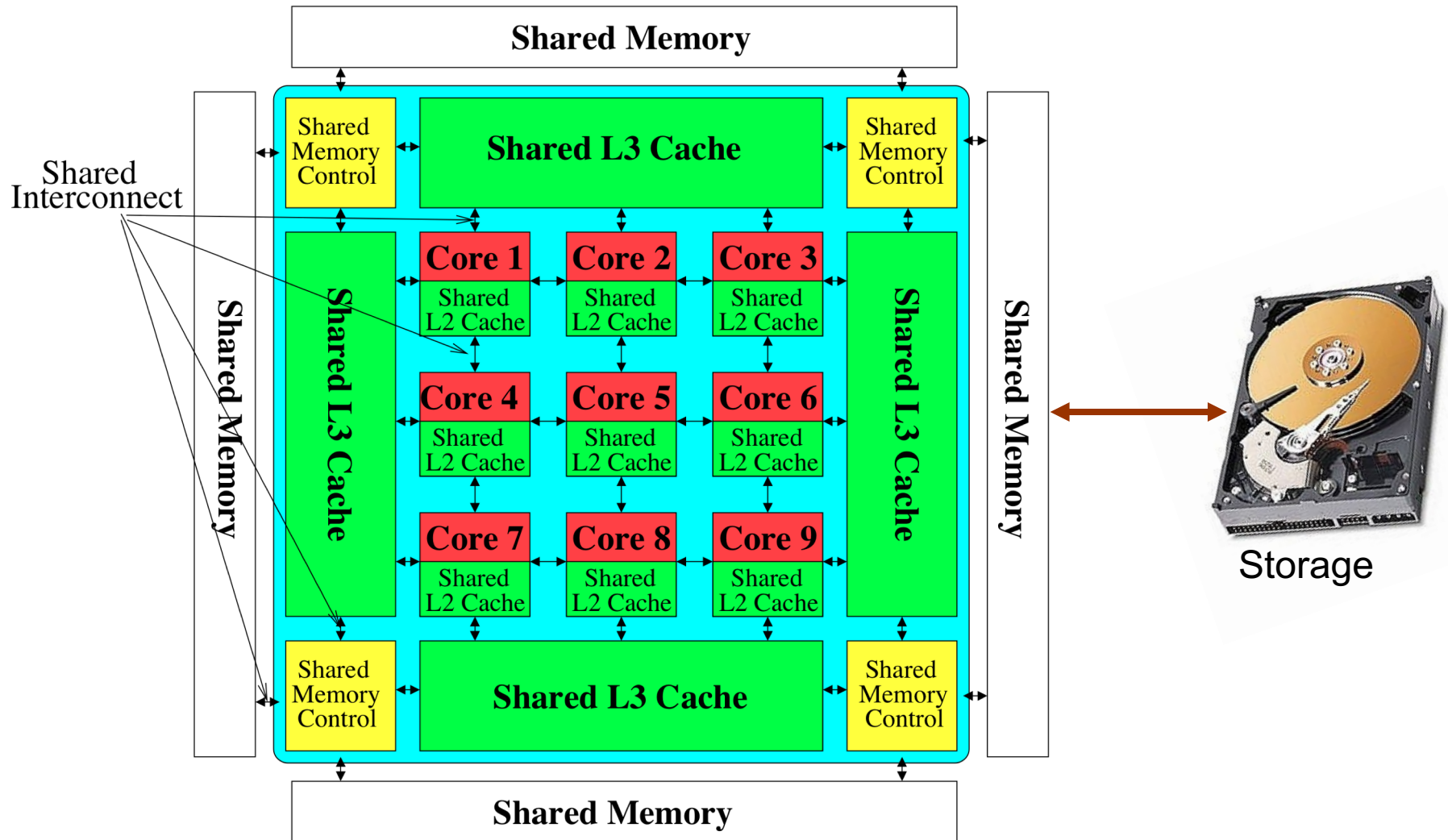
- Main memory is a critical component of all computing systems: server, mobile, embedded, desktop, sensor
- Main memory system must scale (in *size, technology, efficiency, cost, and management algorithms*) to maintain performance growth and technology scaling benefits

The Main Memory System



- Main memory is a critical component of all computing systems: server, mobile, embedded, desktop, sensor
- Main memory system must scale (in *size, technology, efficiency, cost, and management algorithms*) to maintain performance growth and technology scaling benefits

Memory System: A *Shared Resource* View



Most of the system is dedicated to storing and moving data

State of the Main Memory System

- Recent technology, architecture, and application trends
 - lead to new requirements
 - exacerbate old requirements
- DRAM and memory controllers, as we know them today, are (will be) unlikely to satisfy all requirements
- Some emerging non-volatile memory technologies (e.g., PCM) enable new opportunities: memory+storage merging
- We need to rethink the main memory system
 - to fix DRAM issues and enable emerging technologies
 - to satisfy all requirements

Major Trends Affecting Main Memory (I)

- Need for main memory capacity, bandwidth, QoS increasing
- Main memory energy/power is a key system design concern
- DRAM technology scaling is ending

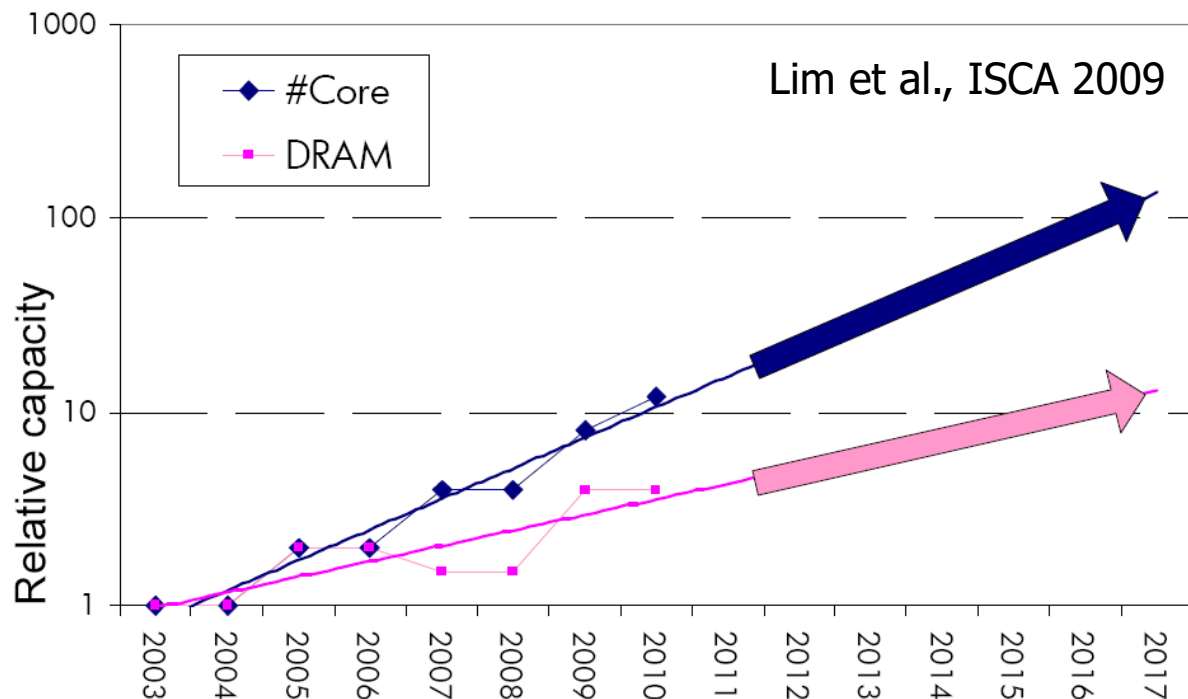
Major Trends Affecting Main Memory (II)

- Need for main memory capacity, bandwidth, QoS increasing
 - **Multi-core**: increasing number of cores/agents
 - **Data-intensive applications**: increasing demand/hunger for data
 - **Consolidation**: cloud computing, GPUs, mobile, heterogeneity
- Main memory energy/power is a key system design concern
- DRAM technology scaling is ending

Consequence: The Memory Capacity Gap

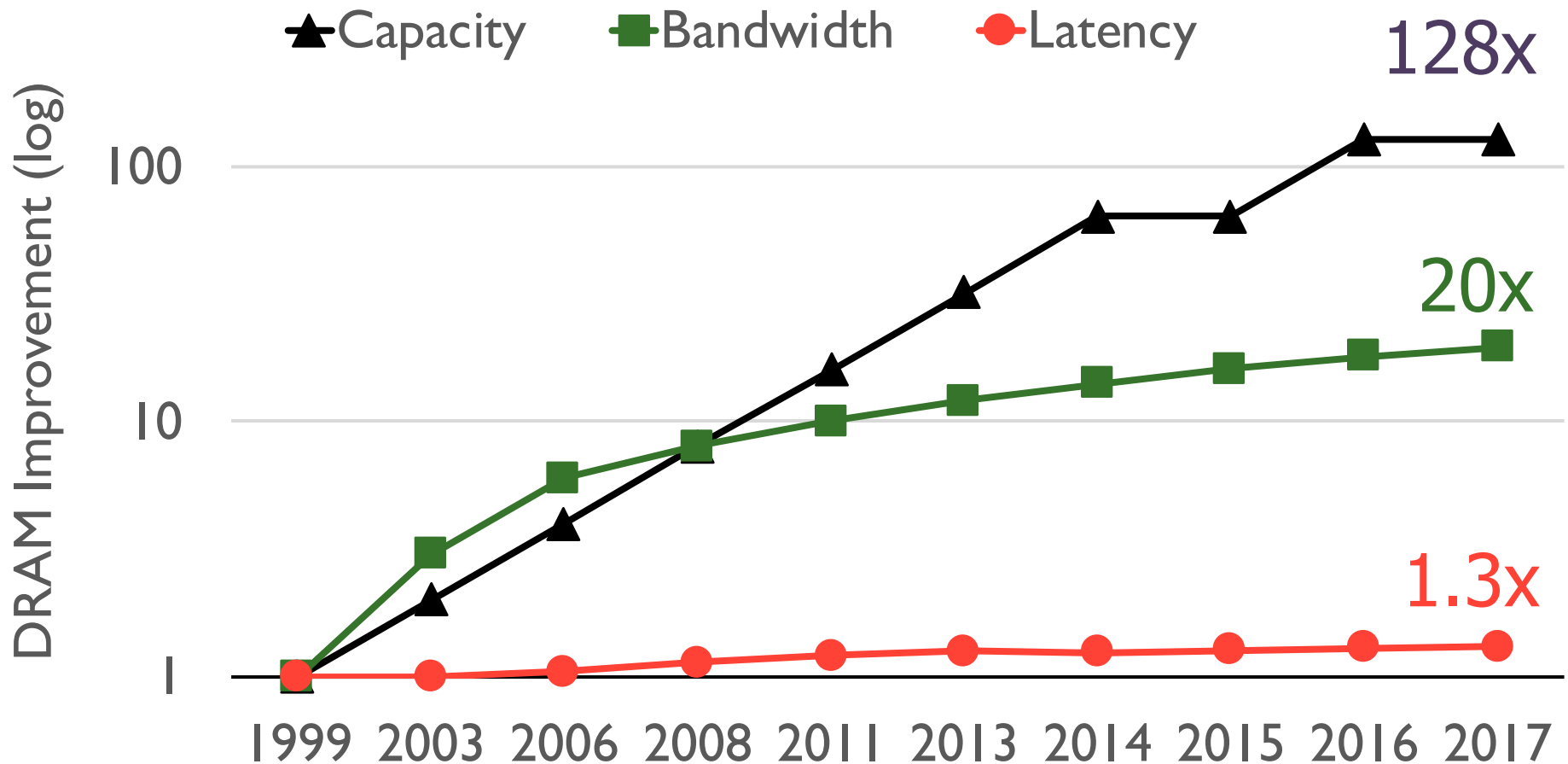
Core count doubling ~ every 2 years

DRAM DIMM capacity doubling ~ every 3 years

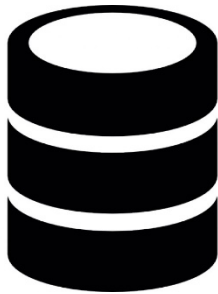


- *Memory capacity per core* expected to drop by 30% every two years
- Trends worse for *memory bandwidth per core*!

DRAM Capacity, Bandwidth & Latency



DRAM Is Critical for Performance



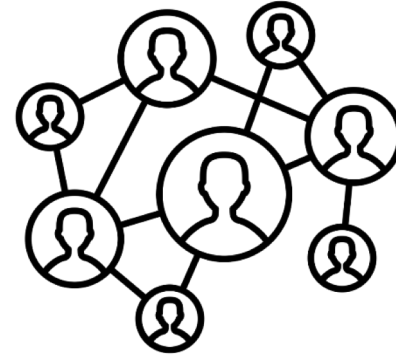
In-memory Databases

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



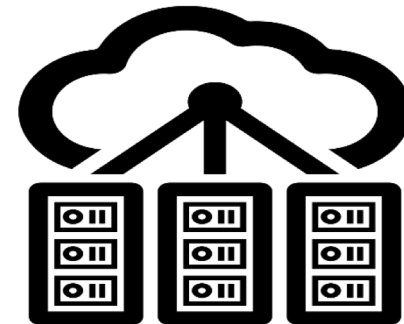
In-Memory Data Analytics

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



Graph/Tree Processing

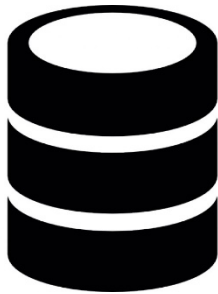
[Xu+, IISWC'12; Umuroglu+, FPL'15]



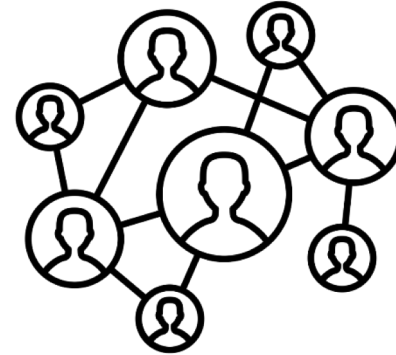
Datacenter Workloads

[Kanev+ (Google), ISCA'15]

DRAM Is Critical for Performance



In-memory Databases



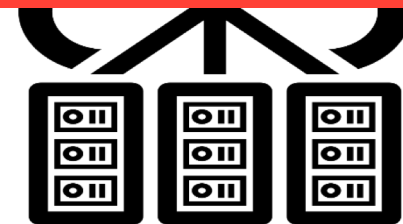
Graph/Tree Processing

Memory → performance bottleneck



In-Memory Data Analytics

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



Datacenter Workloads

[Kanev+ (Google), ISCA'15]

DRAM Is Critical for Performance



Chrome

Google's web browser



TensorFlow Mobile

Google's machine learning
framework

VP9



Video Playback

Google's **video codec**

VP9



Video Capture

Google's **video codec**

DRAM Is Critical for Performance



Chrome



TensorFlow Mobile

Memory → performance bottleneck

VP9



Video Playback

Google's **video codec**

VP9

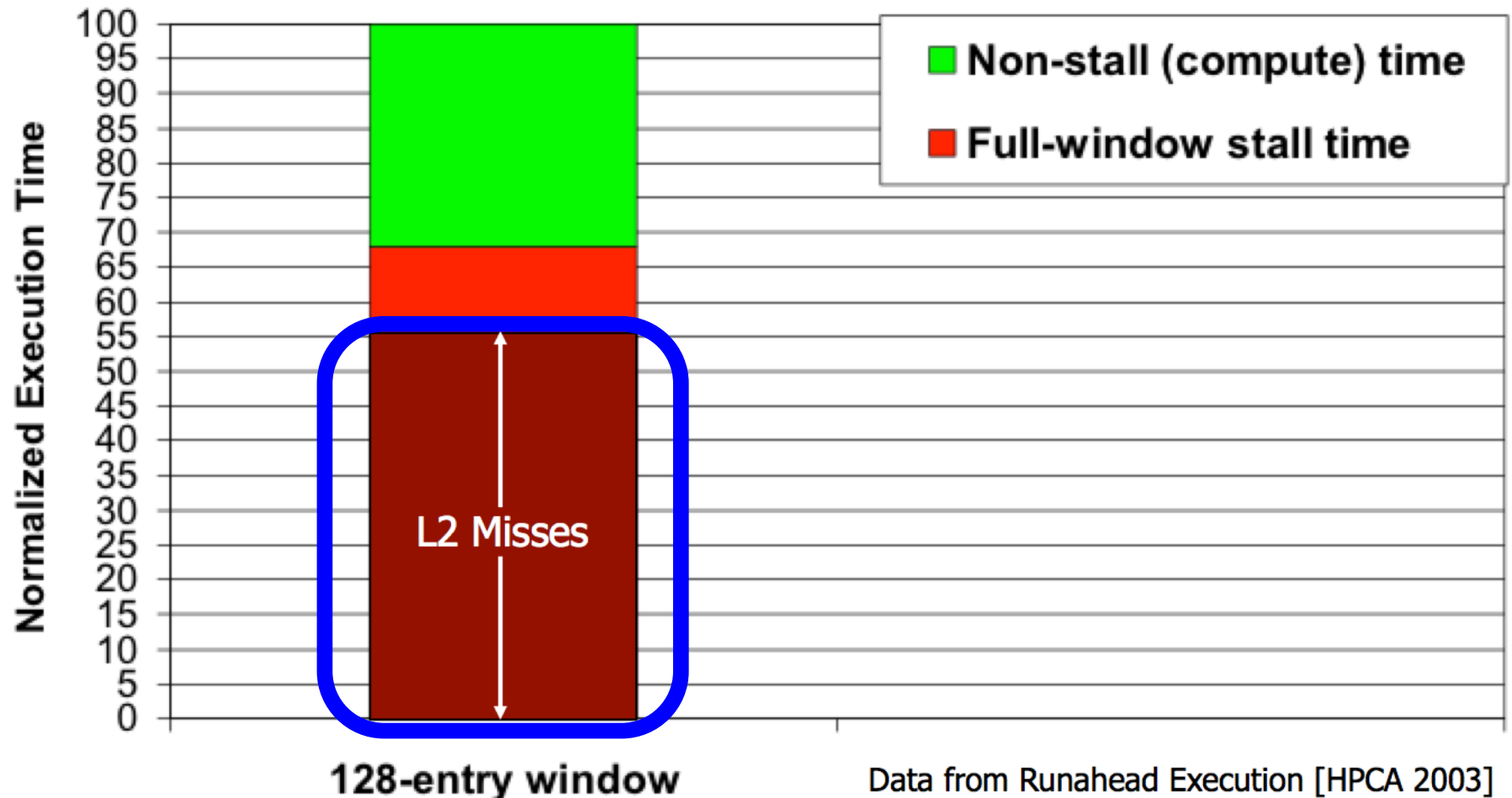


Video Capture

Google's **video codec**

Performance Cost of Data Movement

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



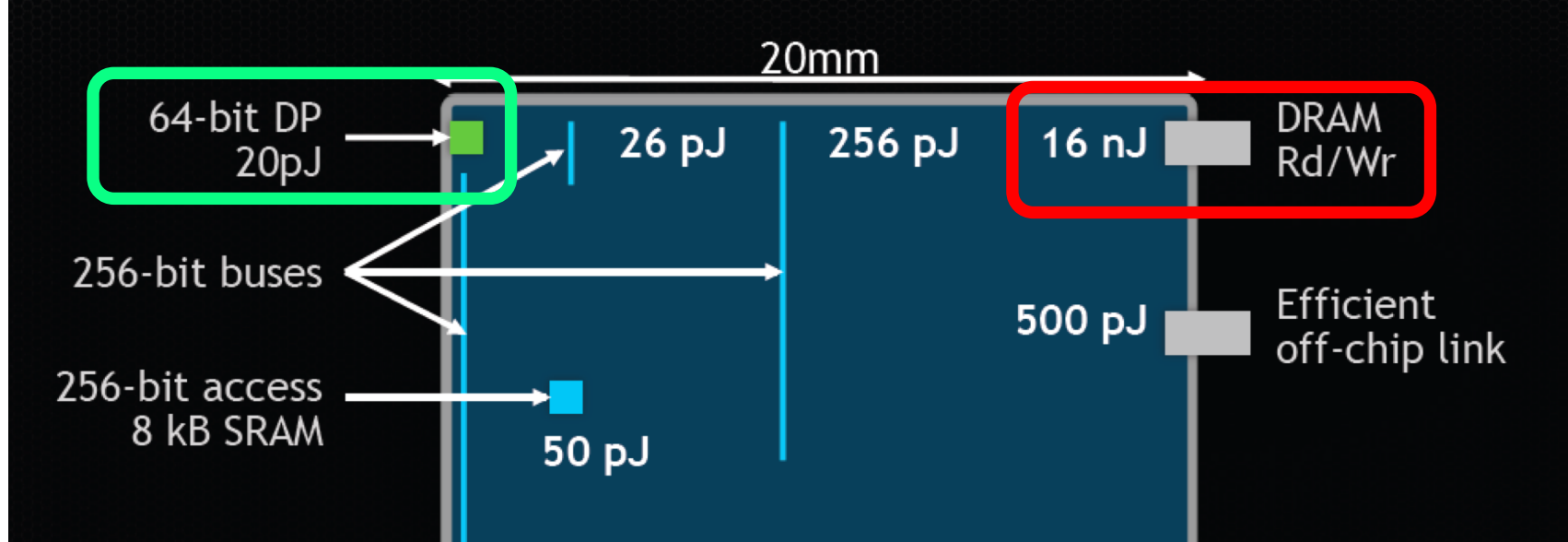
Major Trends Affecting Main Memory (III)

- Need for main memory capacity, bandwidth, QoS increasing
- Main memory energy/power is a key system design concern
 - ~40-50% energy spent in off-chip memory hierarchy [Lefurgy, IEEE Computer'03] >40% power in DRAM [Ware, HPCA'10][Paul, ISCA'15]
 - DRAM consumes power even when not used (periodic refresh)
- DRAM technology scaling is ending

Energy Cost of Data Movement

Communication Dominates Arithmetic

Dally, HiPEAC 2015



A memory access consumes $\sim 1000\times$ the energy of a complex 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 Boroumand¹

Saugata Ghose¹

Youngsok Kim²

Rachata Ausavarungnirun¹

Eric Shiu³

Rahul Thakur³

Daehyun Kim^{4,3}

Aki Kuusela³

Allan Knies³

Parthasarathy Ranganathan³

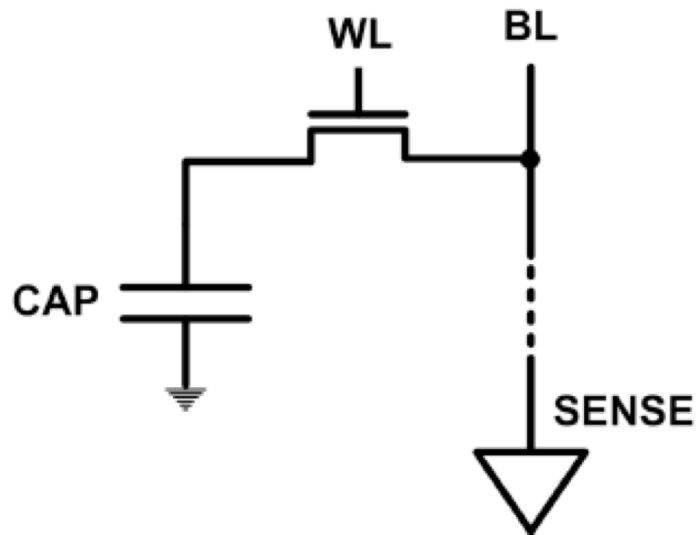
Onur Mutlu^{5,1}

Major Trends Affecting Main Memory (IV)

- Need for main memory capacity, bandwidth, QoS increasing
- Main memory energy/power is a key system design concern
- DRAM technology scaling is ending
 - ITRS projects DRAM will not scale easily below X nm
 - Scaling has provided many benefits:
 - higher capacity (density), lower cost, lower energy

The DRAM Scaling Problem

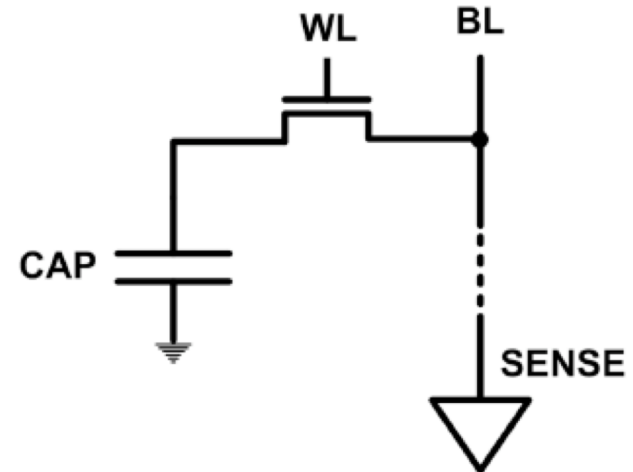
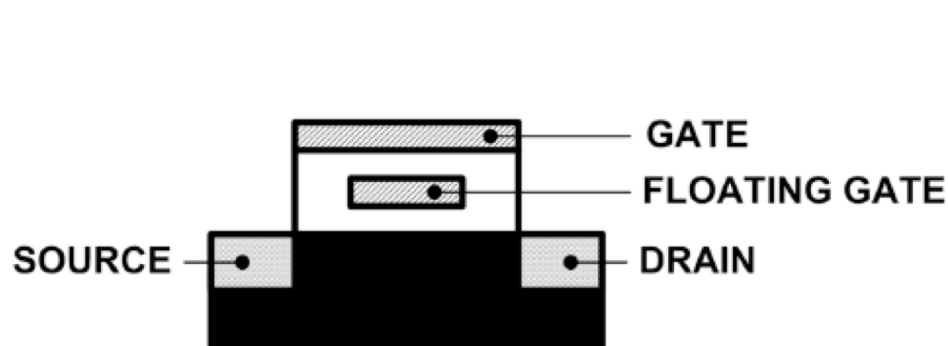
- DRAM stores charge in a capacitor (charge-based memory)
 - Capacitor must be large enough for reliable sensing
 - Access transistor should be large enough for low leakage and high retention time
 - Scaling beyond 40-35nm (2013) is challenging [ITRS, 2009]



- DRAM capacity, cost, and energy/power hard to scale

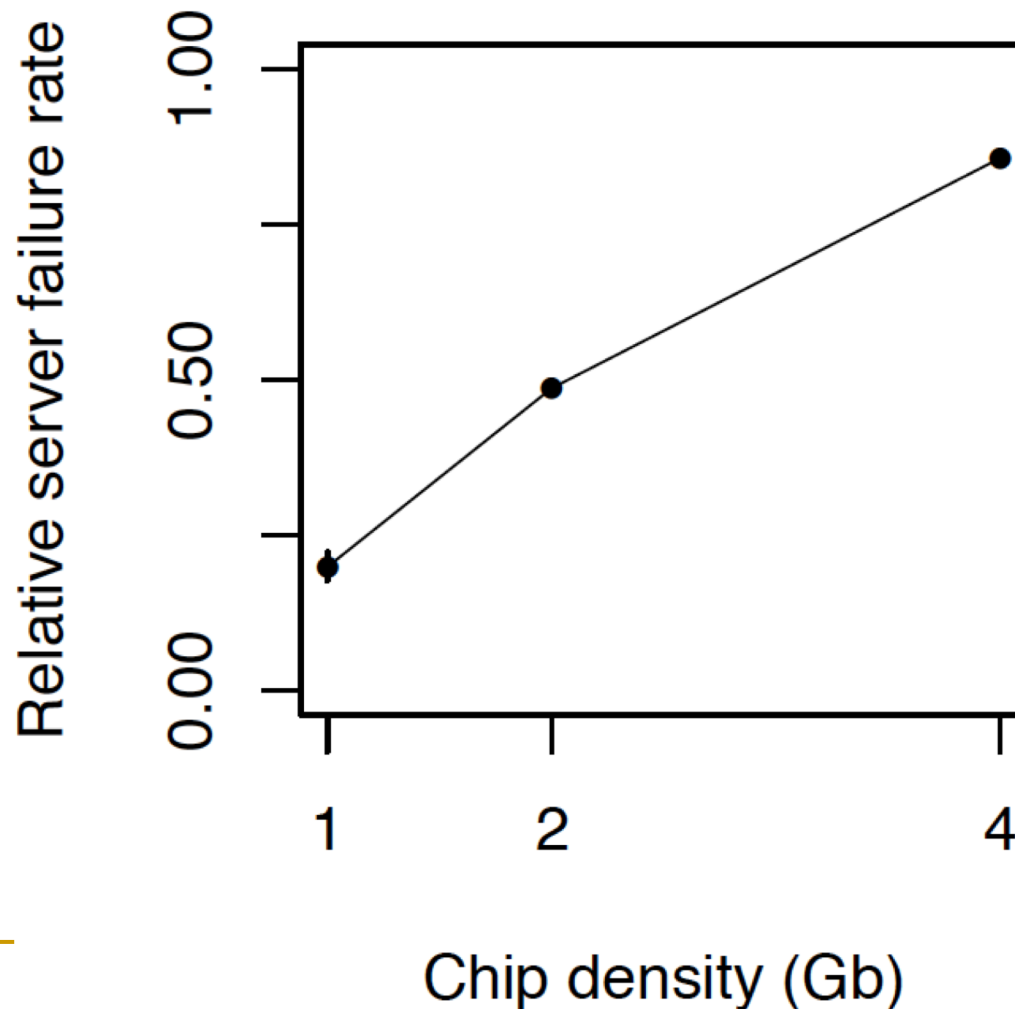
Limits of Charge Memory

- Difficult charge placement and control
 - Flash: floating gate charge
 - DRAM: capacitor charge, transistor leakage
- Reliable sensing becomes difficult as charge storage unit size reduces



As Memory Scales, It Becomes Unreliable

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



*Intuition:
quadratic
increase
in
capacity*

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](#)]

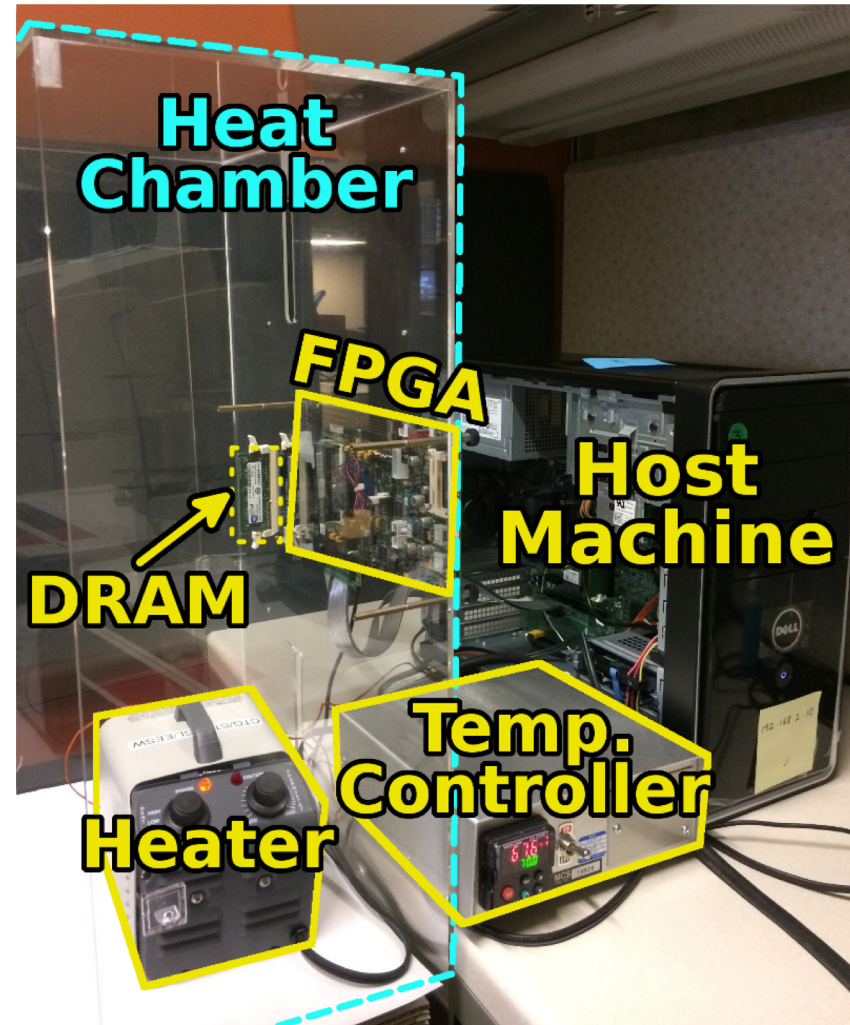
Revisiting Memory Errors in Large-Scale Production Data Centers: Analysis and Modeling of New Trends from the Field

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

The photograph shows the physical implementation of the temperature control system. On the left, a black PC tower is labeled "PC". Above it, a white Temperature Controller unit is labeled "Temperature Controller". In the center, a silver heater unit is labeled "Heater". To the right of the heater are three FPGA development boards, each labeled "FPGAs". The components are interconnected with numerous cables, including blue USB cables, black power and signal wires, and wide flat ribbon drives used for data transfer between the FPGAs and the PC.

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



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

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

Hasan Hassan^{1,2,3} Nandita Vijaykumar³ Samira Khan^{4,3} Saugata Ghose³ Kevin Chang³
Gennady Pekhimenko^{5,3} Donghyuk Lee^{6,3} Oguz Ergin² 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*

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

WIRED

Forget Software—Now Hackers Are Exploiting Physics

BUSINESS	CULTURE	DESIGN	GEAR	SCIENCE
----------	---------	--------	------	---------

ANDY GREENBERG SECURITY 08.31.16 7:00 AM

SHARE



SHARE
18276



TWEET

FORGET SOFTWARE—NOW HACKERS ARE EXPLOITING PHYSICS

The Reliability & Security Perspectives

- Onur Mutlu,
"The RowHammer Problem and Other Issues We May Face as Memory Becomes Denser"

Invited Paper in Proceedings of the Design, Automation, and Test in Europe Conference (DATE), Lausanne, Switzerland, March 2017.

[[Slides \(pptx\)](#) ([pdf](#))]

The RowHammer Problem and Other Issues We May Face as Memory Becomes Denser

Onur Mutlu
ETH Zürich
onur.mutlu@inf.ethz.ch
<https://people.inf.ethz.ch/omutlu>

A RowHammer Retrospective

- 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](#)]

RowHammer: A Retrospective

Onur Mutlu^{§‡} Jeremie S. Kim^{‡§}
§ETH Zürich ‡Carnegie Mellon University

Major Trends Affecting Main Memory (V)

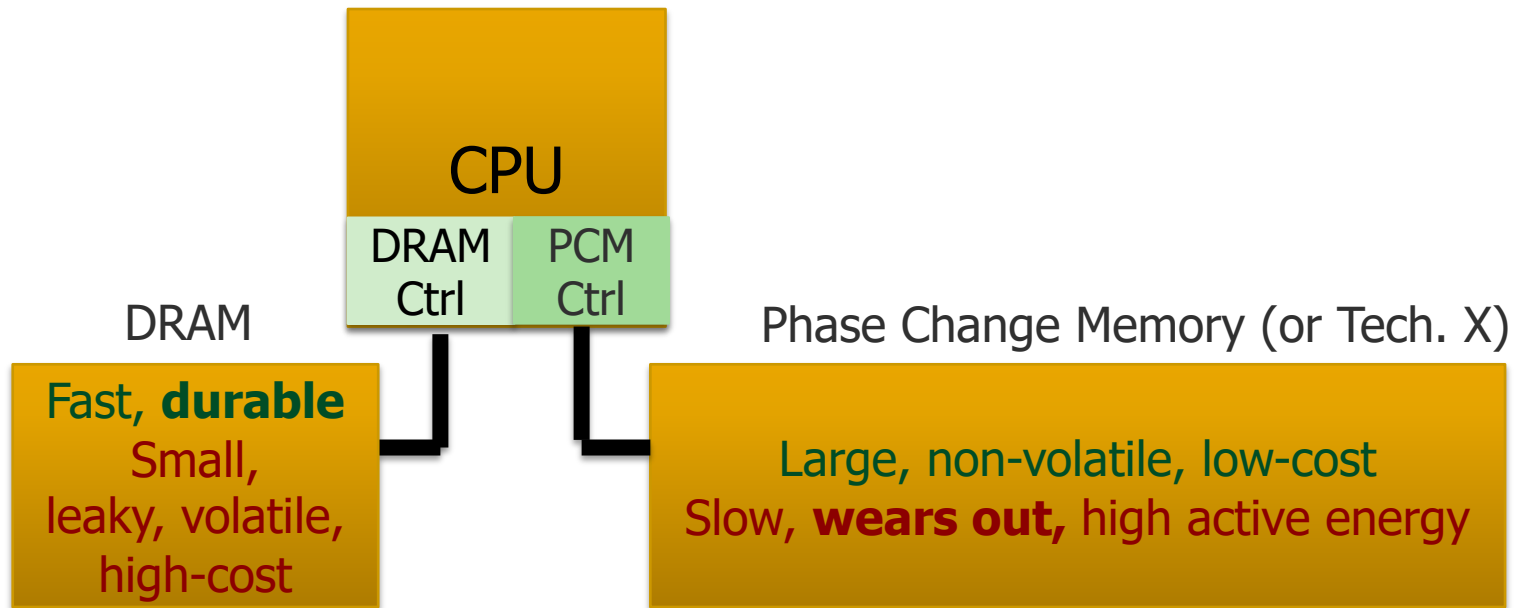
- DRAM scaling has already become increasingly difficult
 - Increasing cell leakage current, reduced cell reliability, increasing manufacturing difficulties [Kim+ ISCA 2014], [Liu+ ISCA 2013], [Mutlu IMW 2013], [Mutlu DATE 2017]
 - **Difficult to significantly improve capacity, energy**
- **Emerging memory technologies** are promising

Major Trends Affecting Main Memory (V)

- DRAM scaling has already become increasingly difficult
 - Increasing cell leakage current, reduced cell reliability, increasing manufacturing difficulties [Kim+ ISCA 2014], [Liu+ ISCA 2013], [Mutlu IMW 2013], [Mutlu DATE 2017]
 - **Difficult to significantly improve capacity, energy**
- **Emerging memory technologies** are promising

3D-Stacked DRAM	higher bandwidth	smaller capacity
Reduced-Latency DRAM (e.g., RL/TL-DRAM, FLY-RAM)	lower latency	higher cost
Low-Power DRAM (e.g., LPDDR3, LPDDR4, Voltron)	lower power	higher latency higher cost
Non-Volatile Memory (NVM) (e.g., PCM, STTRAM, ReRAM, 3D Xpoint)	larger capacity	higher latency higher dynamic power lower endurance

Major Trend: Hybrid Main Memory



Hardware/software manage data allocation and movement
to achieve the best of multiple technologies

Meza+, "[Enabling Efficient and Scalable Hybrid Memories](#)," IEEE Comp. Arch. Letters, 2012.
Yoon+, "[Row Buffer Locality Aware Caching Policies for Hybrid Memories](#)," ICCD 2012 Best Paper Award.

Main Memory Needs Intelligent Controllers

Industry Is Writing Papers About It, Too

DRAM Process Scaling Challenges

❖ Refresh

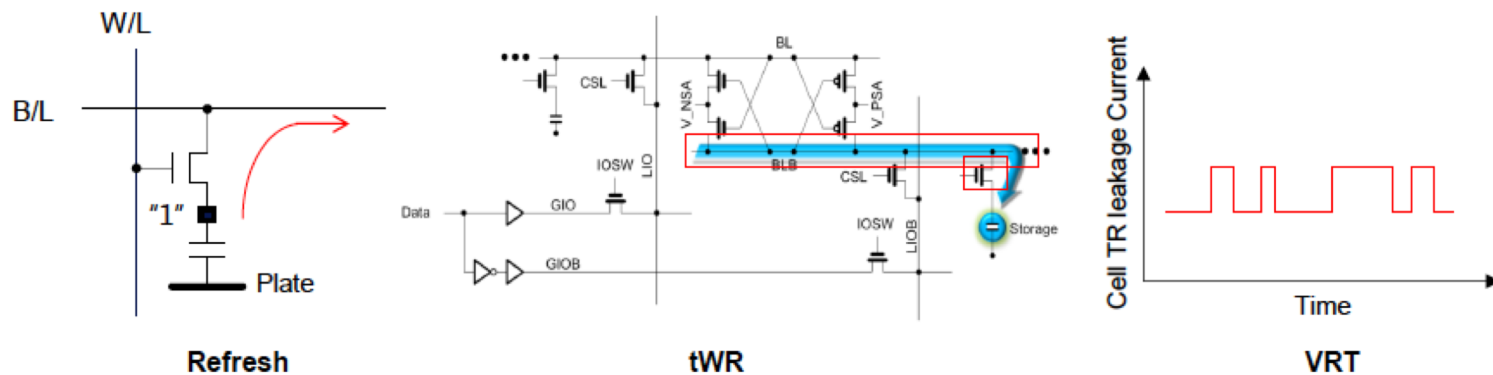
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance
- Leakage current of cell access transistors increasing

❖ tWR

- Contact resistance between the cell capacitor and access transistor increasing
- On-current of the cell access transistor decreasing
- Bit-line resistance increasing

❖ VRT

- Occurring more frequently with cell capacitance decreasing



Call for Intelligent Memory Controllers

DRAM Process Scaling Challenges

❖ Refresh

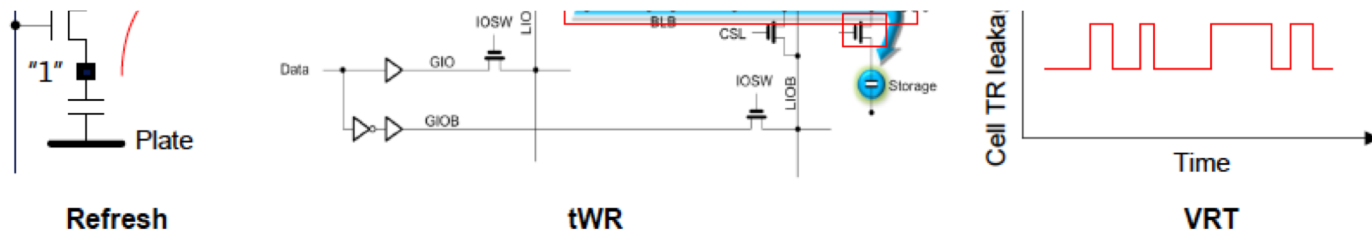
- Difficult to build high-aspect ratio cell capacitors decreasing cell capacitance

THE MEMORY FORUM 2014

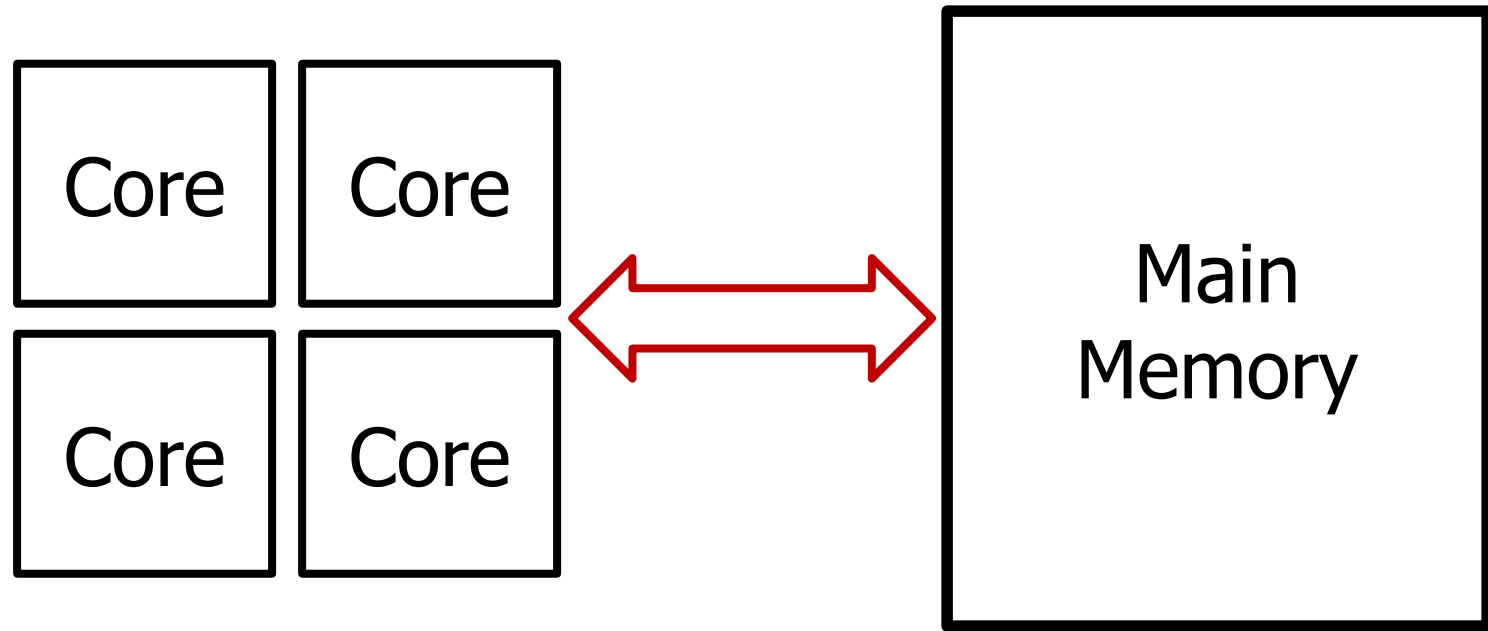
Co-Architecting Controllers and DRAM to Enhance DRAM Process Scaling

Uksong Kang, Hak-soo Yu, Churoo Park, *Hongzhong Zheng,
**John Halbert, **Kuljit Bains, SeongJin Jang, and Joo Sun Choi

*Samsung Electronics, Hwasung, Korea / *Samsung Electronics, San Jose / **Intel*

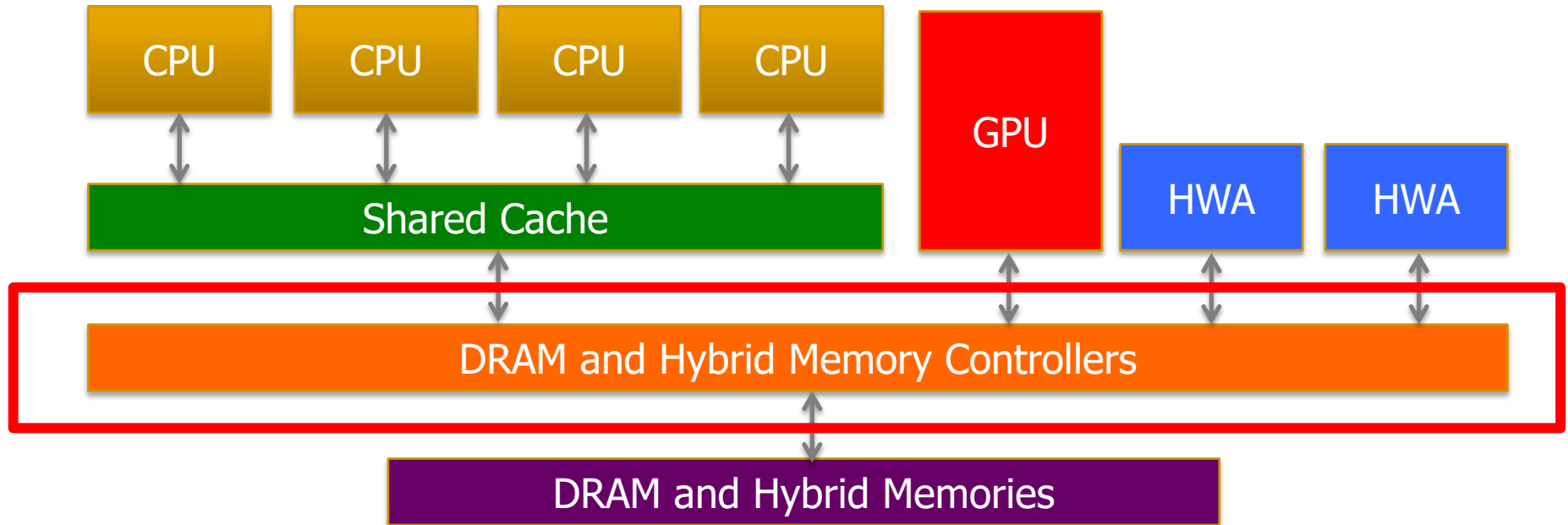


An Orthogonal Issue: Memory Interference



Cores' interfere with each other when accessing shared main memory
Uncontrolled interference leads to many problems (QoS, performance)

Goal: Predictable Performance in Complex Systems



- Heterogeneous agents: CPUs, GPUs, and HWAs
- Main memory interference between CPUs, GPUs, HWAs

How to allocate resources to heterogeneous agents to mitigate interference and provide predictable performance?

Main Memory Needs Intelligent Controllers

Trends, Challenges, and Opportunities in Main Memory

How Do We Solve The Memory Problem?

- **Fix it:** Make memory and controllers more intelligent
 - **New interfaces, functions, architectures:** system-mem codesign
- **Eliminate or minimize it:** Replace or (more likely) augment DRAM with a different technology
 - **New technologies and system-wide rethinking** of memory & storage
- **Embrace it:** Design heterogeneous memories (none of which are perfect) and map data intelligently across them
 - **New models for data management and maybe usage**
- ...

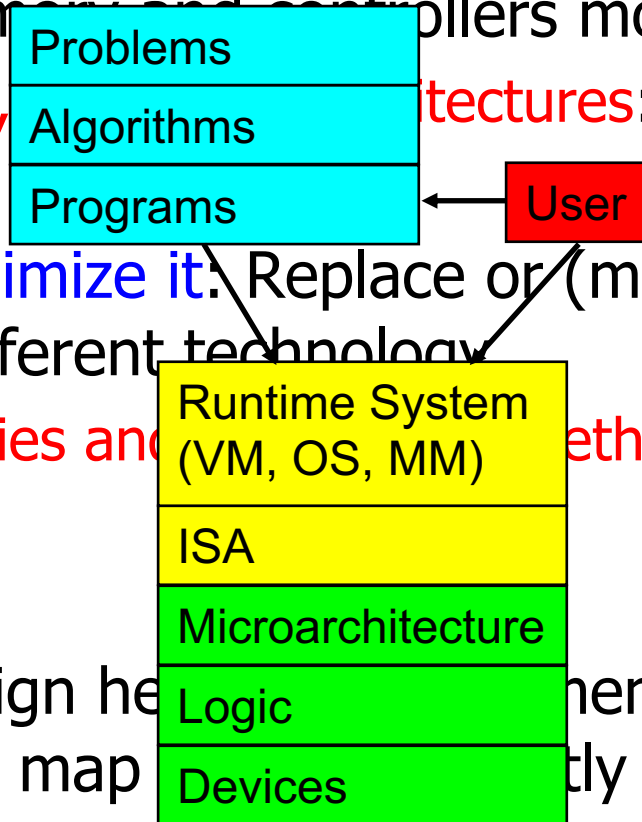
How Do We Solve The Memory Problem?

- **Fix it:** Make memory and controllers more intelligent
 - **New interfaces, functions, architectures:** system-mem codesign
- **Eliminate or minimize it:** Replace or (more likely) augment DRAM with a different technology
 - **New technologies and system-wide rethinking** of memory & storage
- **Embrace it:** Design heterogeneous memories (none of which are perfect) and map data intelligently across them
 - **New models for data management and maybe usage**

Solutions (to memory scaling) require software/hardware/device cooperation

How Do We Solve The Memory Problem?

- **Fix it:** Make memory and controllers more intelligent
 - **New interfaces, architectures:** system-mem codesign
- **Eliminate or minimize it:** Replace or (more likely) augment DRAM with a different technology
 - **New technologies and storage**
- **Embrace it:** Design heterogeneous memories (none of which are perfect) and map applications across them
 - **New models for data management and maybe usage**



Solutions (to memory scaling) require software/hardware/device cooperation

Solution 1: New Memory Architectures

- Overcome memory shortcomings with
 - ❑ Memory-centric system design
 - ❑ Novel memory architectures, interfaces, functions
 - ❑ Better waste management (efficient utilization)
- Key issues to tackle
 - ❑ Enable reliability at low cost → high capacity
 - ❑ Reduce energy
 - ❑ Reduce latency
 - ❑ Improve bandwidth
 - ❑ Reduce waste (capacity, bandwidth, latency)
 - ❑ Enable computation close to data

Solution 1: New Memory Architectures

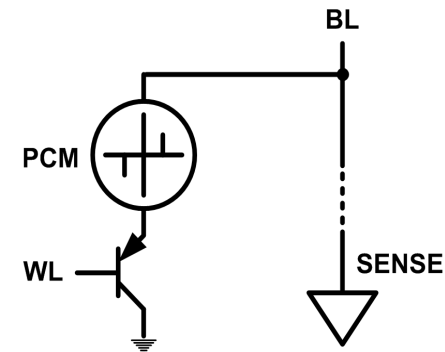
- Liu+, "RAIDR: Retention-Aware Intelligent DRAM Refresh," ISCA 2012.
- Kim+, "A Case for Exploiting Subarray-Level Parallelism in DRAM," ISCA 2012.
- Lee+, "Tiered-Latency DRAM: A Low Latency and Low Cost DRAM Architecture," HPCA 2013.
- Liu+, "An Experimental Study of Data Retention Behavior in Modern DRAM Devices," ISCA 2013.
- Seshadri+, "RowClone: Fast and Efficient In-DRAM Copy and Initialization of Bulk Data," MICRO 2013.
- Pekhimenko+, "Linearly Compressed Pages: A Main Memory Compression Framework," MICRO 2013.
- Chang+, "Improving DRAM Performance by Parallelizing Refreshes with Accesses," HPCA 2014.
- Khan+, "The Efficacy of Error Mitigation Techniques for DRAM Retention Failures: A Comparative Experimental Study," SIGMETRICS 2014.
- Luo+, "Characterizing Application Memory Error Vulnerability to Optimize Data Center Cost," DSN 2014.
- Kim+, "Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors," ISCA 2014.
- Lee+, "Adaptive-Latency DRAM: Optimizing DRAM Timing for the Common-Case," HPCA 2015.
- Qureshi+, "AVATAR: A Variable-Retention-Time (VRT) Aware Refresh for DRAM Systems," DSN 2015.
- Meza+, "Revisiting Memory Errors in Large-Scale Production Data Centers: Analysis and Modeling of New Trends from the Field," DSN 2015.
- Kim+, "Ramulator: A Fast and Extensible DRAM Simulator," IEEE CAL 2015.
- Seshadri+, "Fast Bulk Bitwise AND and OR in DRAM," IEEE CAL 2015.
- Ahn+, "A Scalable Processing-in-Memory Accelerator for Parallel Graph Processing," ISCA 2015.
- Ahn+, "PIM-Enabled Instructions: A Low-Overhead, Locality-Aware Processing-in-Memory Architecture," ISCA 2015.
- Lee+, "Decoupled Direct Memory Access: Isolating CPU and IO Traffic by Leveraging a Dual-Data-Port DRAM," PACT 2015.
- Seshadri+, "Gather-Scatter DRAM: In-DRAM Address Translation to Improve the Spatial Locality of Non-unit Strided Accesses," MICRO 2015.
- Lee+, "Simultaneous Multi-Layer Access: Improving 3D-Stacked Memory Bandwidth at Low Cost," TACO 2016.
- Hassan+, "ChargeCache: Reducing DRAM Latency by Exploiting Row Access Locality," HPCA 2016.
- Chang+, "Low-Cost Inter-Linked Subarrays (LISA): Enabling Fast Inter-Subarray Data Migration in DRAM," HPCA 2016.
- Chang+, "Understanding Latency Variation in Modern DRAM Chips: Experimental Characterization, Analysis, and Optimization," SIGMETRICS 2016.
- Khan+, "PARBOR: An Efficient System-Level Technique to Detect Data Dependent Failures in DRAM," DSN 2016.
- Hsieh+, "Transparent Offloading and Mapping (TOM): Enabling Programmer-Transparent Near-Data Processing in GPU Systems," ISCA 2016.
- Hashemi+, "Accelerating Dependent Cache Misses with an Enhanced Memory Controller," ISCA 2016.
- Boroumand+, "LazyPIM: An Efficient Cache Coherence Mechanism for Processing-in-Memory," IEEE CAL 2016.
- Pattnaik+, "Scheduling Techniques for GPU Architectures with Processing-In-Memory Capabilities," PACT 2016.
- Hsieh+, "Accelerating Pointer Chasing in 3D-Stacked Memory: Challenges, Mechanisms, Evaluation," ICCD 2016.
- Hashemi+, "Continuous Runahead: Transparent Hardware Acceleration for Memory Intensive Workloads," MICRO 2016.
- Khan+, "A Case for Memory Content-Based Detection and Mitigation of Data-Dependent Failures in DRAM," IEEE CAL 2016.
- Hassan+, "SoftMC: A Flexible and Practical Open-Source Infrastructure for Enabling Experimental DRAM Studies," HPCA 2017.
- Mutlu, "The RowHammer Problem and Other Issues We May Face as Memory Becomes Denser," DATE 2017.
- Lee+, "Design-Induced Latency Variation in Modern DRAM Chips: Characterization, Analysis, and Latency Reduction Mechanisms," SIGMETRICS 2017.
- Chang+, "Understanding Reduced-Voltage Operation in Modern DRAM Devices: Experimental Characterization, Analysis, and Mechanisms," SIGMETRICS 2017.
- Patel+, "The Reach Profiler (REAPER): Enabling the Mitigation of DRAM Retention Failures via Profiling at Aggressive Conditions," ISCA 2017.
- Seshadri and Mutlu, "Simple Operations in Memory to Reduce Data Movement," ADCOM 2017.
- Liu+, "Concurrent Data Structures for Near-Memory Computing," SPAA 2017.
- Khan+, "Detecting and Mitigating Data-Dependent DRAM Failures by Exploiting Current Memory Content," MICRO 2017.
- Seshadri+, "Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology," MICRO 2017.
- Kim+, "GRIM-Filter: Fast Seed Location Filtering in DNA Read Mapping Using Processing-in-Memory Technologies," BMC Genomics 2018.
- Kim+, "The DRAM Latency PUF: Quickly Evaluating Physical Undoneable Functions by Exploiting the Latency-Reliability Tradeoff in Modern DRAM Devices," HPCA 2018.
- Boroumand+, "Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks," ASPLOS 2018.
- Das+, "VRL-DRAM: Improving DRAM Performance via Variable Refresh Latency," DAC 2018.
- Ghose+, "What Your DRAM Power Models Are Not Telling You: Lessons from a Detailed Experimental Study," SIGMETRICS 2018.
- Kim+, "Solar-DRAM: Reducing DRAM Access Latency by Exploiting the Variation in Local Bitlines," ICCD 2018.
- Wang+, "Reducing DRAM Latency via Charge-Level-Aware Look-Ahead Partial Restoration," MICRO 2018.
- Kim+, "D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput," HPCA 2019.
- Singh+, "NAPEL: Near-Memory Computing Application Performance Prediction via Ensemble Learning," DAC 2019.
- Ghose+, "Demystifying Workload-DRAM Interactions: An Experimental Study," SIGMETRICS 2019.
- Patel+, "Understanding and Modeling On-Die Error Correction in Modern DRAM: An Experimental Study Using Real Devices," DSN 2019.
- Boroumand+, "CoNDA: Efficient Cache Coherence Support for Near-Data Accelerators," ISCA 2019.
- Hassan+, "CROW: A Low-Cost Substrate for Improving DRAM Performance, Energy Efficiency, and Reliability," ISCA 2019.
- Mutlu and Kim, "RowHammer: A Retrospective," TCAD 2019.
- Mutlu+, "Processing Data Where It Makes Sense: Enabling In-Memory Computation," MICPRO 2019.
- Seshadri and Mutlu, "In-DRAM Bulk Bitwise Execution Engine," ADCOM 2020.
- Avoid DRAM:
 - Seshadri+, "The Evicted-Address Filter: A Unified Mechanism to Address Both Cache Pollution and Thrashing," PACT 2012.
 - Pekhimenko+, "Base-Delta-Immediate Compression: Practical Data Compression for On-Chip Caches," PACT 2012.
 - Seshadri+, "The Dirty-Block Index," ISCA 2014.
 - Pekhimenko+, "Exploiting Compressed Block Size as an Indicator of Future Reuse," HPCA 2015.
 - Vijaykumar+, "A Case for Core-Assisted Bottleneck Acceleration in GPUs: Enabling Flexible Data Compression with Assist Warps," ISCA 2015.
 - Pekhimenko+, "Toggle-Aware Bandwidth Compression for GPUs," HPCA 2016.

Solution 2: Emerging Memory Technologies

- Some emerging **resistive** memory technologies seem more scalable than DRAM (and they are non-volatile)

- Example: Phase Change Memory

- Data stored by changing phase of material
- Data read by detecting material's resistance
- Expected to scale to 9nm (2022 [ITRS 2009])
- Prototyped at 20nm (Raoux+, IBM JRD 2008)
- Expected to be denser than DRAM: can store multiple bits/cell

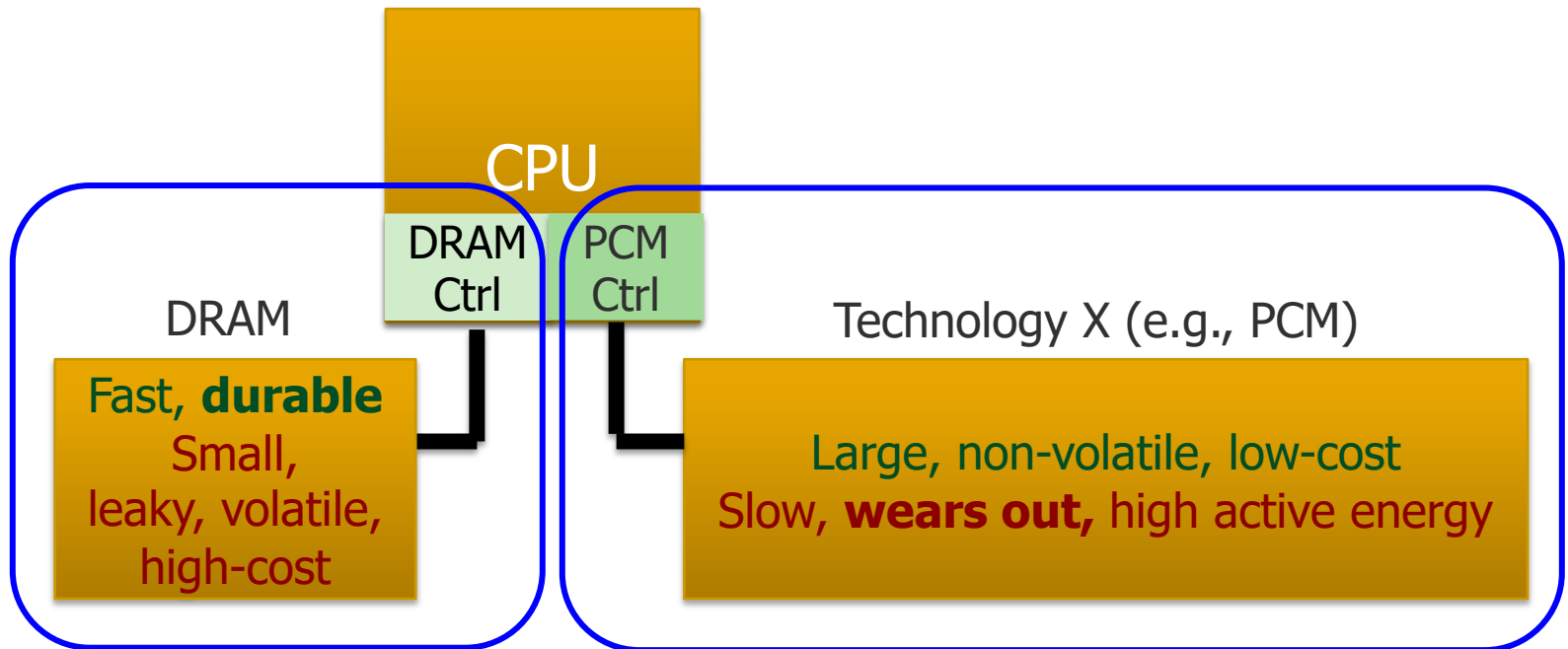


- But, emerging technologies have (many) shortcomings
 - Can they be enabled to replace/augment/surpass DRAM?

Solution 2: Emerging Memory Technologies

- Lee+, “[Architecting Phase Change Memory as a Scalable DRAM Alternative](#),” ISCA’09, CACM’10, IEEE Micro’10.
- Meza+, “[Enabling Efficient and Scalable Hybrid Memories](#),” IEEE Comp. Arch. Letters 2012.
- Yoon, Meza+, “[Row Buffer Locality Aware Caching Policies for Hybrid Memories](#),” ICCD 2012.
- Kultursay+, “[Evaluating STT-RAM as an Energy-Efficient Main Memory Alternative](#),” ISPASS 2013.
- Meza+, “[A Case for Efficient Hardware-Software Cooperative Management of Storage and Memory](#),” WEED 2013.
- Lu+, “[Loose Ordering Consistency for Persistent Memory](#),” ICCD 2014.
- Zhao+, “[FIRM: Fair and High-Performance Memory Control for Persistent Memory Systems](#),” MICRO 2014.
- Yoon, Meza+, “[Efficient Data Mapping and Buffering Techniques for Multi-Level Cell Phase-Change Memories](#),” TACO 2014.
- Ren+, “[ThyNVM: Enabling Software-Transparent Crash Consistency in Persistent Memory Systems](#),” MICRO 2015.
- Chauhan+, “[NVMove: Helping Programmers Move to Byte-Based Persistence](#),” INFLOW 2016.
- Li+, “[Utility-Based Hybrid Memory Management](#),” CLUSTER 2017.
- Yu+, “[Banshee: Bandwidth-Efficient DRAM Caching via Software/Hardware Cooperation](#),” MICRO 2017.
- Tavakkol+, “[MQSim: A Framework for Enabling Realistic Studies of Modern Multi-Queue SSD Devices](#),” FAST 2018.
- Tavakkol+, “[FLIN: Enabling Fairness and Enhancing Performance in Modern NVMe Solid State Drives](#),” ISCA 2018.
- Sadrosadati+. “[LTRF: Enabling High-Capacity Register Files for GPUs via Hardware/Software Cooperative Register Prefetching](#),” ASPLOS 2018.
- Salkhordeh+, “[An Analytical Model for Performance and Lifetime Estimation of Hybrid DRAM-NVM Main Memories](#),” TC 2019.
- Wang+, “[Panthera: Holistic Memory Management for Big Data Processing over Hybrid Memories](#),” PLDI 2019.

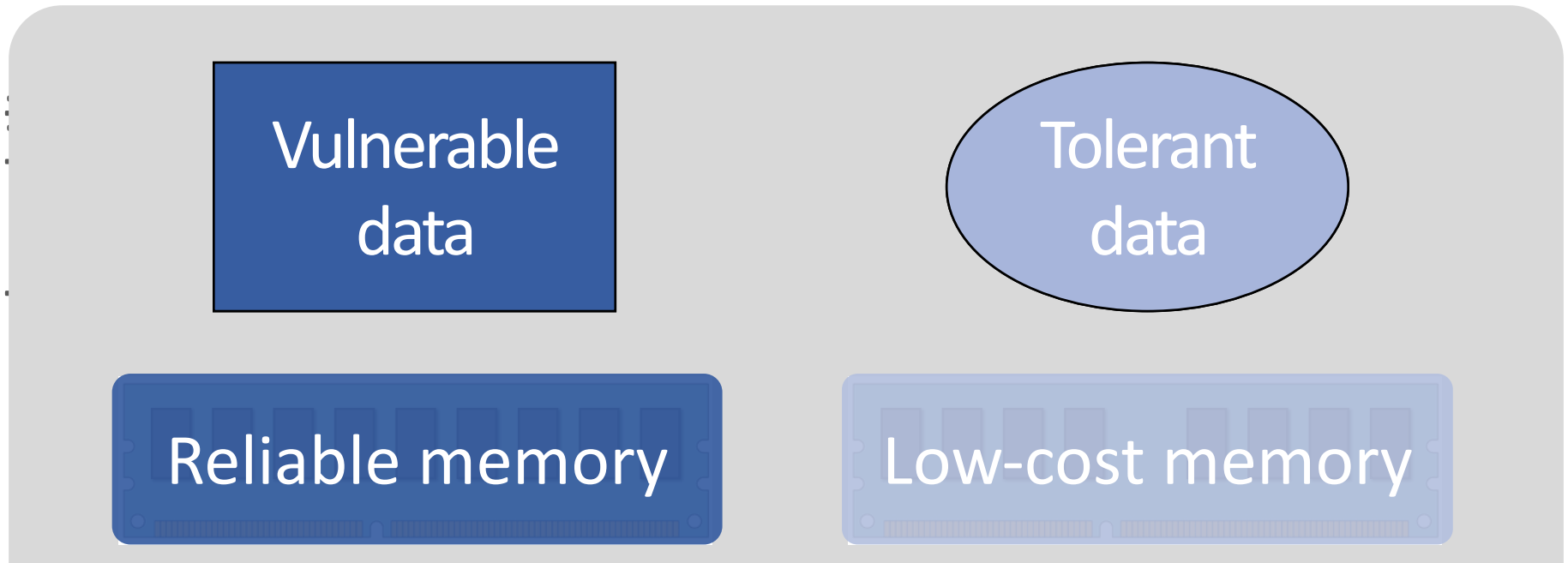
Combination: Hybrid Memory Systems



Hardware/software manage data allocation and movement
to achieve the best of multiple technologies

Meza+, "Enabling Efficient and Scalable Hybrid Memories," IEEE Comp. Arch. Letters, 2012.
Yoon, Meza et al., "Row Buffer Locality Aware Caching Policies for Hybrid Memories," ICCD
2012 Best Paper Award.

Exploiting Memory Error Tolerance with Hybrid Memory Systems



On Microsoft's Web Search workload

Reduces server hardware **cost** by **4.7 %**

Achieves single server **availability** target of **99.90 %**

Heterogeneous-Reliability Memory [DSN 2014]

More on Heterogeneous Reliability Memory

- Yixin Luo, Sriram Govindan, Bikash Sharma, Mark Santaniello, Justin Meza, Aman Kansal, Jie Liu, Badriddine Khessib, Kushagra Vaid, and Onur Mutlu,
"Characterizing Application Memory Error Vulnerability to Optimize Data Center Cost via Heterogeneous-Reliability Memory"
Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Atlanta, GA, June 2014. [[Summary](#)]
[[Slides \(pptx\)](#)] [[pdf](#)] [[Coverage on ZDNet](#)]

Characterizing Application Memory Error Vulnerability to Optimize Datacenter Cost via Heterogeneous-Reliability Memory

Yixin Luo Sriram Govindan* Bikash Sharma* Mark Santaniello* Justin Meza
Aman Kansal* Jie Liu* Badriddine Khessib* Kushagra Vaid* Onur Mutlu

Carnegie Mellon University, yixinluo@cs.cmu.edu, {meza, onur}@cmu.edu

*Microsoft Corporation, {srgovin, bsharma, marksan, kansal, jie.liu, bk Hessib, kvaid}@microsoft.com

An Orthogonal Issue: Memory Interference

- Problem: **Memory interference between cores is uncontrolled**
 - unfairness, starvation, low performance
 - **uncontrollable, unpredictable, vulnerable system**
 - Solution: **QoS-Aware Memory Systems**
 - Hardware designed to provide a configurable fairness substrate
 - Application-aware memory scheduling, partitioning, throttling
 - Software designed to configure the resources to satisfy different QoS goals
 - QoS-aware memory systems can provide predictable performance and higher efficiency
-

Strong Memory Service Guarantees

- Goal: Satisfy performance/SLA requirements in the presence of shared main memory, heterogeneous agents, and hybrid memory/storage
- Approach:
 - Develop techniques/models to accurately estimate the performance loss of an application/agent in the presence of resource sharing
 - Develop mechanisms (hardware and software) to enable the resource partitioning/prioritization needed to achieve the required performance levels for all applications
 - All the while providing high system performance
- Subramanian et al., “MISE: Providing Performance Predictability and Improving Fairness in Shared Main Memory Systems,” HPCA 2013.
- Subramanian et al., “The Application Slowdown Model,” MICRO 2015.

Agenda for Today

- Memory Importance and Trends
- RowHammer: Memory Reliability and Security
- In-Memory Computation
- Low-Latency Memory
- Guiding Principles
- Conclusion

Memory Systems and Memory-Centric Computing Systems

Part 1: Memory Importance and Trends

Prof. Onur Mutlu

omutlu@gmail.com

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

7 July 2019

SAMOS Tutorial

SAFARI

ETH zürich

Carnegie Mellon

Backup Slides

Readings, Videos, Reference Materials

Accelerated Memory Course (~6.5 hours)

■ ACACES 2018

- ❑ Memory Systems and Memory-Centric Computing Systems
- ❑ Taught by Onur Mutlu July 9-13, 2018
- ❑ ~6.5 hours of lectures

■ Website for the Course including Videos, Slides, Papers

- ❑ https://safari.ethz.ch/memory_systems/ACACES2018/
- ❑ <https://www.youtube.com/playlist?list=PL5Q2soXY2Zi-HXxomthrpDpMJm05P6J9x>

■ All Papers are at:

- ❑ <https://people.inf.ethz.ch/omutlu/projects.htm>
- ❑ Final lecture notes and readings (for all topics)

Required: Reference Overview Paper I

Processing Data Where It Makes Sense: Enabling In-Memory Computation

Onur Mutlu^{a,b}, Saugata Ghose^b, Juan Gómez-Luna^a, Rachata Ausavarungnirun^{b,c}

^a*ETH Zürich*

^b*Carnegie Mellon University*

^c*King Mongkut's University of Technology North Bangkok*

Onur Mutlu, Saugata Ghose, Juan Gomez-Luna, and Rachata Ausavarungnirun,
**"Processing Data Where It Makes Sense: Enabling In-Memory
Computation"**

*Invited paper in Microprocessors and Microsystems (**MICPRO**), June 2019.
[arXiv version]*

Required: Reference Overview Paper II

- 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](#)]

RowHammer: A Retrospective

Onur Mutlu^{§‡} Jeremie S. Kim^{‡§}
§ETH Zürich ‡Carnegie Mellon University

Required: Reference Overview Paper III

- Onur Mutlu and Lavanya Subramanian,
"Research Problems and Opportunities in Memory Systems"
Invited Article in Supercomputing Frontiers and Innovations (SUPERFRI), 2014/2015.

Research Problems and Opportunities in Memory Systems

Onur Mutlu¹, Lavanya Subramanian¹

Reference Overview Paper IV

Enabling the Adoption of Processing-in-Memory: Challenges, Mechanisms, Future Research Directions

SAUGATA GHOSE, KEVIN HSIEH, AMIRALI BOROUMAND,
RACHATA AUSAVARUNGNIRUN

Carnegie Mellon University

ONUR MUTLU

ETH Zürich and Carnegie Mellon University

Saugata Ghose, Kevin Hsieh, Amirali Boroumand, Rachata Ausavarungnirun, Onur Mutlu,
**"Enabling the Adoption of Processing-in-Memory: Challenges, Mechanisms,
Future Research Directions"**

Invited Book Chapter, to appear in 2018.

[[Preliminary arxiv.org version](https://arxiv.org/pdf/1802.00320.pdf)]

Reference Overview Paper V

- Onur Mutlu,
"The RowHammer Problem and Other Issues We May Face as Memory Becomes Denser"
*Invited Paper in Proceedings of the Design, Automation, and Test in Europe Conference (**DATE**), Lausanne, Switzerland, March 2017.*
[[Slides \(pptx\)](#) ([pdf](#))]

The RowHammer Problem and Other Issues We May Face as Memory Becomes Denser

Onur Mutlu
ETH Zürich
onur.mutlu@inf.ethz.ch
<https://people.inf.ethz.ch/omutlu>

Reference Overview Paper VI

- Onur Mutlu,
"Memory Scaling: A Systems Architecture Perspective"

*Technical talk at MemCon 2013 (**MEMCON**), Santa Clara, CA, August 2013. [[Slides \(pptx\)](#)] [[pdf](#)]
[[Video](#)] [[Coverage on StorageSearch](#)]*

Memory Scaling: A Systems Architecture Perspective

Onur Mutlu
Carnegie Mellon University
onur@cmu.edu
<http://users.ece.cmu.edu/~omutlu/>



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

Related Videos and Course Materials (I)

- **Undergraduate Computer Architecture Course Lecture Videos (2015, 2014, 2013)**
- **Undergraduate Computer Architecture Course Materials (2015, 2014, 2013)**

- **Graduate Computer Architecture Course Lecture Videos (2017, 2015, 2013)**
- **Graduate Computer Architecture Course Materials (2017, 2015, 2013)**

- **Parallel Computer Architecture Course Materials (Lecture Videos)**

Related Videos and Course Materials (II)

- **Freshman Digital Circuits and Computer Architecture Course Lecture Videos (2018, 2017)**
- **Freshman Digital Circuits and Computer Architecture Course Materials (2018)**
- **Memory Systems Short Course Materials (Lecture Video on Main Memory and DRAM Basics)**

Some Open Source Tools (I)

- Rowhammer – Program to Induce RowHammer Errors
 - <https://github.com/CMU-SAFARI/rowhammer>
- Ramulator – Fast and Extensible DRAM Simulator
 - <https://github.com/CMU-SAFARI/ramulator>
- MemSim – Simple Memory Simulator
 - <https://github.com/CMU-SAFARI/memsim>
- NOCulator – Flexible Network-on-Chip Simulator
 - <https://github.com/CMU-SAFARI/NOCulator>
- SoftMC – FPGA-Based DRAM Testing Infrastructure
 - <https://github.com/CMU-SAFARI/SoftMC>
- Other open-source software from my group
 - <https://github.com/CMU-SAFARI/>
 - <http://www.ece.cmu.edu/~safari/tools.html>

Some Open Source Tools (II)

- MQSim – A Fast Modern SSD Simulator
 - <https://github.com/CMU-SAFARI/MQSim>
- Mosaic – GPU Simulator Supporting Concurrent Applications
 - <https://github.com/CMU-SAFARI/Mosaic>
- IMPICA – Processing in 3D-Stacked Memory Simulator
 - <https://github.com/CMU-SAFARI/IMPICA>
- SMLA – Detailed 3D-Stacked Memory Simulator
 - <https://github.com/CMU-SAFARI/SMLA>
- HWASim – Simulator for Heterogeneous CPU-HWA Systems
 - <https://github.com/CMU-SAFARI/HWASim>
- Other open-source software from my group
 - <https://github.com/CMU-SAFARI/>
 - <http://www.ece.cmu.edu/~safari/tools.html>

More Open Source Tools (III)

- A lot more open-source software from my group
 - ❑ <https://github.com/CMU-SAFARI/>
 - ❑ <http://www.ece.cmu.edu/~safari/tools.html>



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.

📍 ETH Zurich and Carnegi... 🔗 <http://www.ece.cmu.ed...> ✉ omutlu@gmail.com

📁 Repositories 30

👤 People 27

👥 Teams 1

📁 Projects 0

⚙ Settings

Type: All ▾

Language: All ▾

Customize pinned repositories

New

MQSim

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 implementations, steady-state SSD conditions, and the full end-to-end latency of requests in modern SSDs. It is described in detail in the FAST 2018 paper by A...

🔗 14 ⭐ 14 🏢 MIT Updated 8 days ago



Top languages

● C++ ● C ● C# ● AGS Script
● Verilog

Most used topics

Manage

dram reliability

Referenced Papers

- All are available at

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

<http://scholar.google.com/citations?user=7XyGUGkAAAAJ&hl=en>

<https://people.inf.ethz.ch/omutlu/acaces2018.html>

Ramulator: A Fast and Extensible DRAM Simulator

[IEEE Comp Arch Letters'15]

Ramulator Motivation

- DRAM and Memory Controller landscape is changing
- Many new and upcoming standards
- Many new controller designs
- A fast and easy-to-extend simulator is very much needed

<i>Segment</i>	<i>DRAM Standards & Architectures</i>
Commodity	DDR3 (2007) [14]; DDR4 (2012) [18]
Low-Power	LPDDR3 (2012) [17]; LPDDR4 (2014) [20]
Graphics	GDDR5 (2009) [15]
Performance	eDRAM [28], [32]; RLDram3 (2011) [29]
3D-Stacked	WIO (2011) [16]; WIO2 (2014) [21]; MCDRAM (2015) [13]; HBM (2013) [19]; HMC1.0 (2013) [10]; HMC1.1 (2014) [11]
Academic	SBA/SSA (2010) [38]; Staged Reads (2012) [8]; RAIDR (2012) [27]; SALP (2012) [24]; TL-DRAM (2013) [26]; RowClone (2013) [37]; Half-DRAM (2014) [39]; Row-Buffer Decoupling (2014) [33]; SARP (2014) [6]; AL-DRAM (2015) [25]

Table 1. Landscape of DRAM-based memory

Ramulator

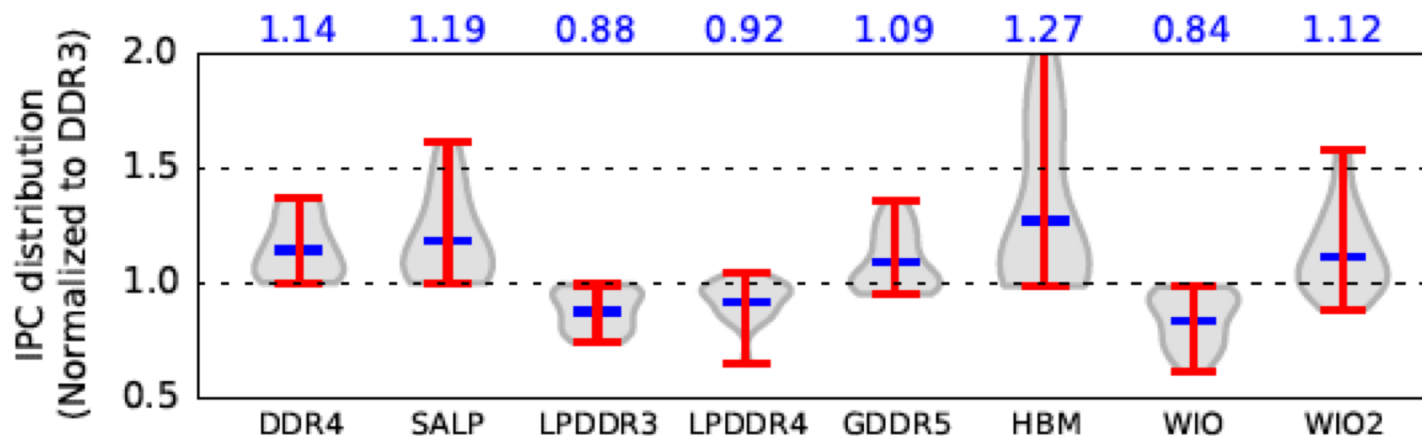
- Provides out-of-the box support for many DRAM standards:
 - DDR3/4, LPDDR3/4, GDDR5, WIO1/2, HBM, plus new proposals (SALP, AL-DRAM, TLDRAM, RowClone, and SARP)
- ~2.5X faster than fastest open-source simulator
- Modular and extensible to different standards

<i>Simulator</i> (clang -O3)	<i>Cycles (10⁶)</i>		<i>Runtime (sec.)</i>		<i>Req/sec (10³)</i>		<i>Memory</i> (MB)
	<i>Random</i>	<i>Stream</i>	<i>Random</i>	<i>Stream</i>	<i>Random</i>	<i>Stream</i>	
Ramulator	652	411	752	249	133	402	2.1
DRAMSim2	645	413	2,030	876	49	114	1.2
USIMM	661	409	1,880	750	53	133	4.5
DrSim	647	406	18,109	12,984	6	8	1.6
NVMain	666	413	6,881	5,023	15	20	4,230.0

Table 3. Comparison of five simulators using two traces

Case Study: Comparison of DRAM Standards

<i>Standard</i>	<i>Rate (MT/s)</i>	<i>Timing (CL-RCD-RP)</i>	<i>Data-Bus (Width×Chan.)</i>	<i>Rank-per-Chan</i>	<i>BW (GB/s)</i>
DDR3	1,600	11-11-11	64-bit × 1	1	11.9
DDR4	2,400	16-16-16	64-bit × 1	1	17.9
SALP [†]	1,600	11-11-11	64-bit × 1	1	11.9
LPDDR3	1,600	12-15-15	64-bit × 1	1	11.9
LPDDR4	2,400	22-22-22	32-bit × 2*	1	17.9
GDDR5 [12]	6,000	18-18-18	64-bit × 1	1	44.7
HBM	1,000	7-7-7	128-bit × 8*	1	119.2
WIO	266	7-7-7	128-bit × 4*	1	15.9
WIO2	1,066	9-10-10	128-bit × 8*	1	127.2



Across 22 workloads, simple CPU model

Figure 2. Performance comparison of DRAM standards

Ramulator Paper and Source Code

- Yoongu Kim, Weikun Yang, and Onur Mutlu,
"Ramulator: A Fast and Extensible DRAM Simulator"
IEEE Computer Architecture Letters (**CAL**), March 2015.
[[Source Code](#)]
- Source code is released under the liberal MIT License
 - <https://github.com/CMU-SAFARI/ramulator>

Ramulator: A Fast and Extensible DRAM Simulator

Yoongu Kim¹ Weikun Yang^{1,2} Onur Mutlu¹
¹Carnegie Mellon University ²Peking University

Optional Assignment

- Review the Ramulator paper
 - Email me your review (omutlu@gmail.com)
- Download and run Ramulator
 - Compare DDR3, DDR4, SALP, HBM for the libquantum benchmark (provided in Ramulator repository)
 - Email me your report (omutlu@gmail.com)
- This **will** help you get into **memory systems research**

Some More Suggested Readings

Some Key Readings on DRAM (I)

■ DRAM Organization and Operation

- ❑ Lee et al., “Tiered-Latency DRAM: A Low Latency and Low Cost DRAM Architecture,” HPCA 2013.
https://people.inf.ethz.ch/omutlu/pub/tldram_hpca13.pdf
- ❑ Kim et al., “A Case for Subarray-Level Parallelism (SALP) in DRAM,” ISCA 2012.
https://people.inf.ethz.ch/omutlu/pub/salp-dram_isca12.pdf
- ❑ Lee et al., “Simultaneous Multi-Layer Access: Improving 3D-Stacked Memory Bandwidth at Low Cost,” ACM TACO 2016.
https://people.inf.ethz.ch/omutlu/pub/smla_high-bandwidth-3d-stacked-memory_taco16.pdf

Some Key Readings on DRAM (II)

■ DRAM Refresh

- ❑ Liu et al., “RAIDR: Retention-Aware Intelligent DRAM Refresh,” ISCA 2012.
https://people.inf.ethz.ch/omutlu/pub/raidr-dram-refresh_isca12.pdf
- ❑ Chang et al., “Improving DRAM Performance by Parallelizing Refreshes with Accesses,” HPCA 2014.
https://people.inf.ethz.ch/omutlu/pub/dram-access-refresh-parallelization_hpca14.pdf
- ❑ Patel et al., “The Reach Profiler (REAPER): Enabling the Mitigation of DRAM Retention Failures via Profiling at Aggressive Conditions,” ISCA 2017.
https://people.inf.ethz.ch/omutlu/pub/reaper-dram-retention-profiling-lpddr4_isca17.pdf

Reading on Simulating Main Memory

- How to evaluate future main memory systems?
- An open-source simulator and its brief description
- Yoongu Kim, Weikun Yang, and Onur Mutlu,
"Ramulator: A Fast and Extensible DRAM Simulator"
IEEE Computer Architecture Letters (**CAL**), March 2015.
[[Source Code](#)]

Some Key Readings on Memory Control 1

- ❑ Mutlu+, "Parallelism-Aware Batch Scheduling: Enhancing both Performance and Fairness of Shared DRAM Systems," ISCA 2008.
https://people.inf.ethz.ch/omutlu/pub/parbs_isca08.pdf
- ❑ Kim et al., "Thread Cluster Memory Scheduling: Exploiting Differences in Memory Access Behavior," MICRO 2010.
https://people.inf.ethz.ch/omutlu/pub/tcm_micro10.pdf
- ❑ Subramanian et al., "BLISS: Balancing Performance, Fairness and Complexity in Memory Access Scheduling," TPDS 2016.
https://people.inf.ethz.ch/omutlu/pub/bliss-memory-scheduler_ieee-tpds16.pdf
- ❑ Usui et al., "DASH: Deadline-Aware High-Performance Memory Scheduler for Heterogeneous Systems with Hardware Accelerators," TACO 2016.
https://people.inf.ethz.ch/omutlu/pub/dash_deadline-aware-heterogeneous-memory-scheduler_taco16.pdf

Some Key Readings on Memory Control 2

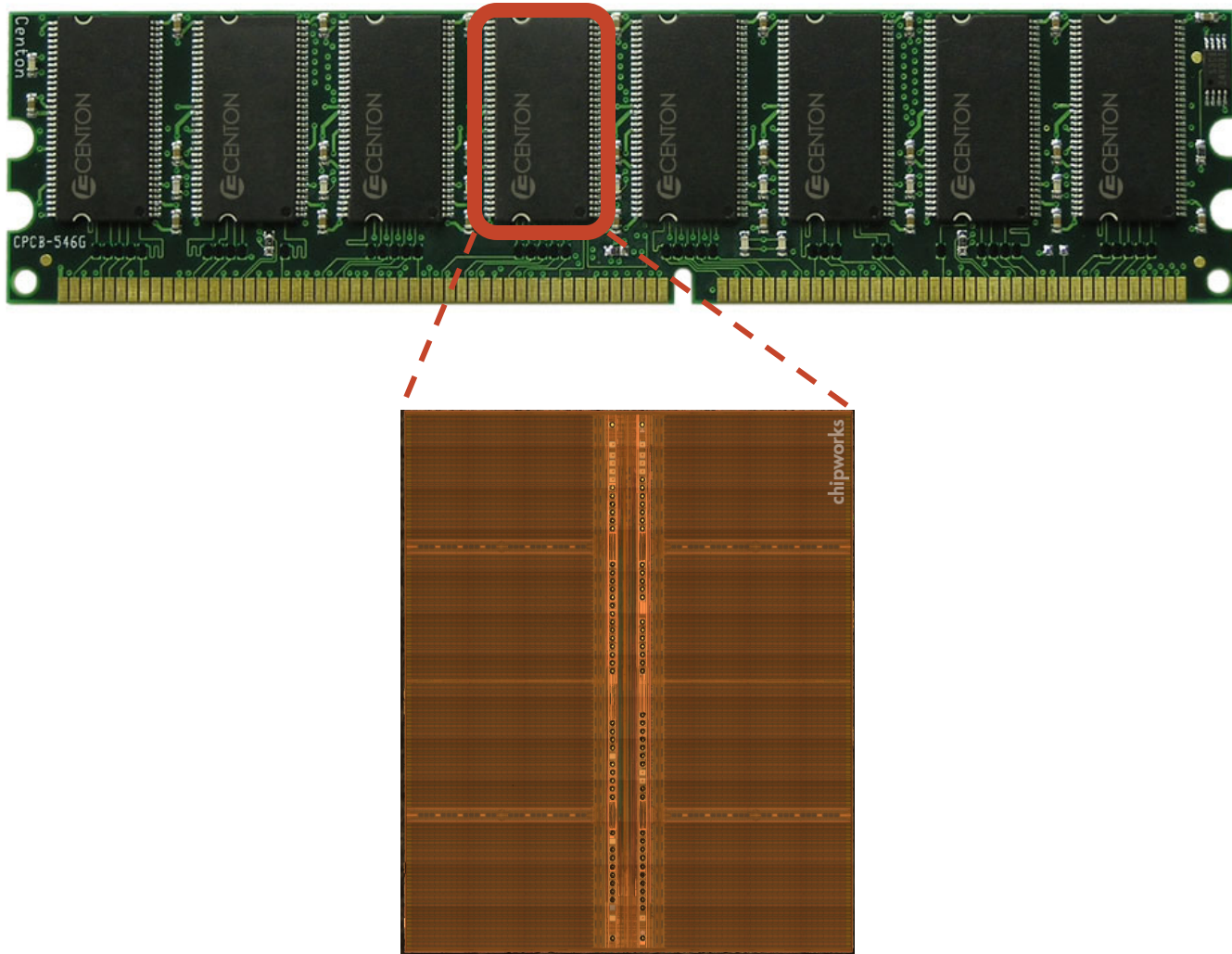
- ❑ Ipek+, "Self Optimizing Memory Controllers: A Reinforcement Learning Approach," ISCA 2008.
https://people.inf.ethz.ch/omutlu/pub/rlmc_isca08.pdf
- ❑ Ebrahimi et al., "Fairness via Source Throttling: A Configurable and High-Performance Fairness Substrate for Multi-Core Memory Systems," ASPLOS 2010.
https://people.inf.ethz.ch/omutlu/pub/fst_asplos10.pdf
- ❑ Subramanian et al., "The Application Slowdown Model: Quantifying and Controlling the Impact of Inter-Application Interference at Shared Caches and Main Memory," MICRO 2015.
https://people.inf.ethz.ch/omutlu/pub/application-slowdown-model_micro15.pdf
- ❑ Lee et al., "Decoupled Direct Memory Access: Isolating CPU and IO Traffic by Leveraging a Dual-Data-Port DRAM," PACT 2015.
https://people.inf.ethz.ch/omutlu/pub/decoupled-dma_pact15.pdf

More Readings

- To come as we cover the future topics
- Search for “DRAM” or “Memory” in:
 - <https://people.inf.ethz.ch/omutlu/projects.htm>

Inside A DRAM Chip

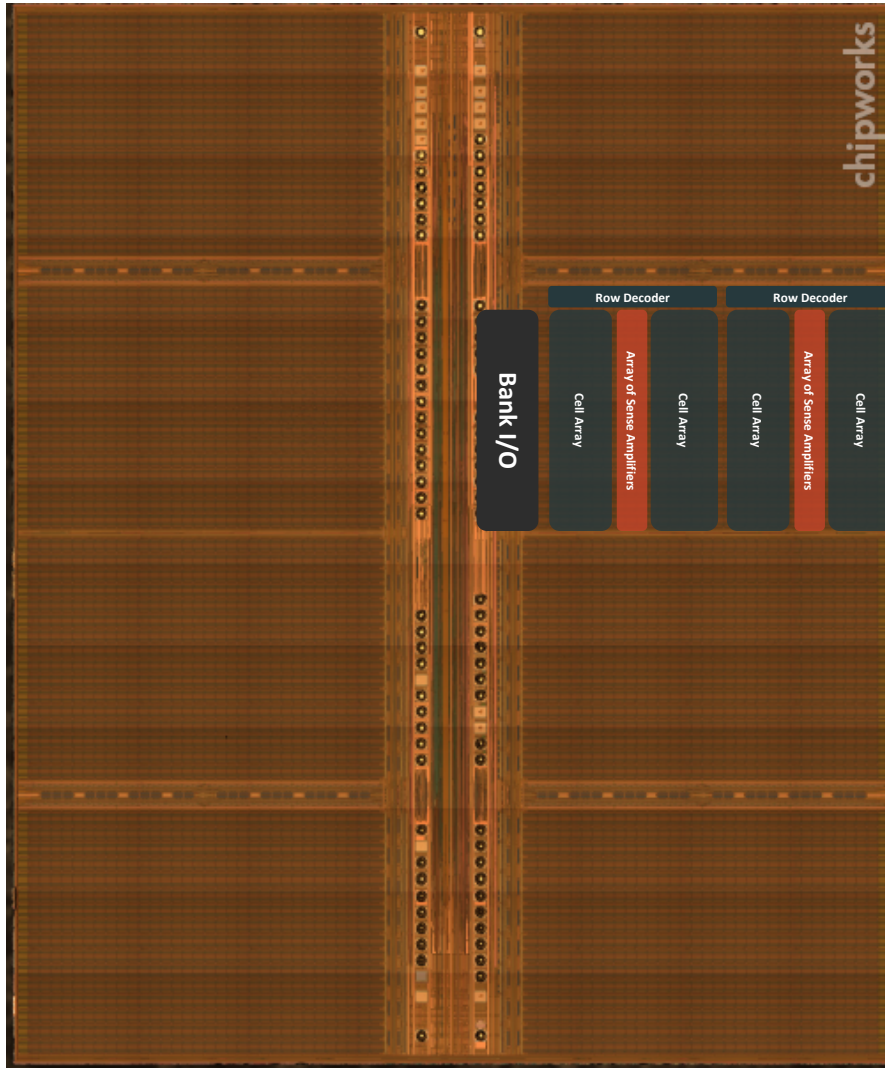
DRAM Module and Chip



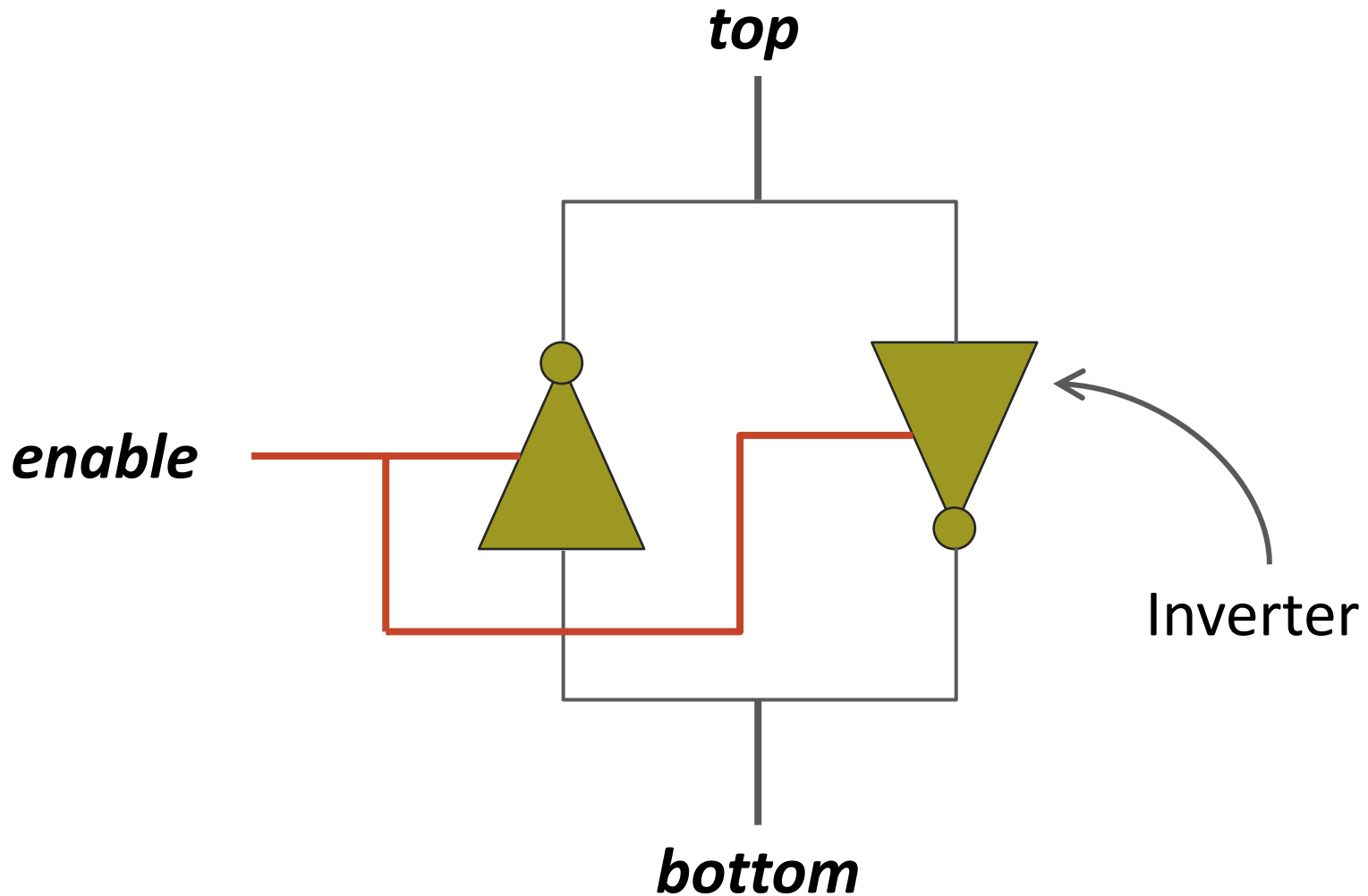
Goals

- Cost
- Latency
- Bandwidth
- Parallelism
- Power
- Energy
- Reliability
- ...

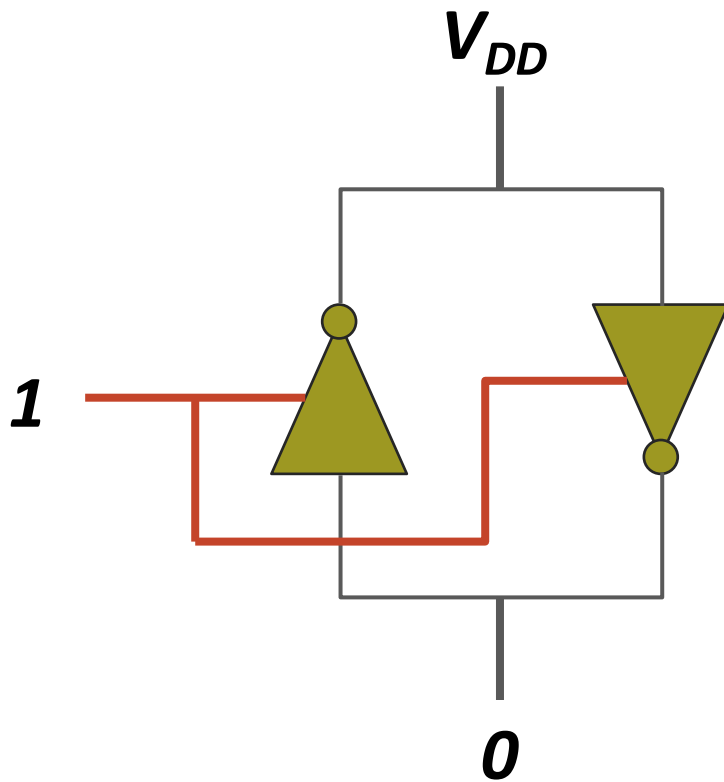
DRAM Chip



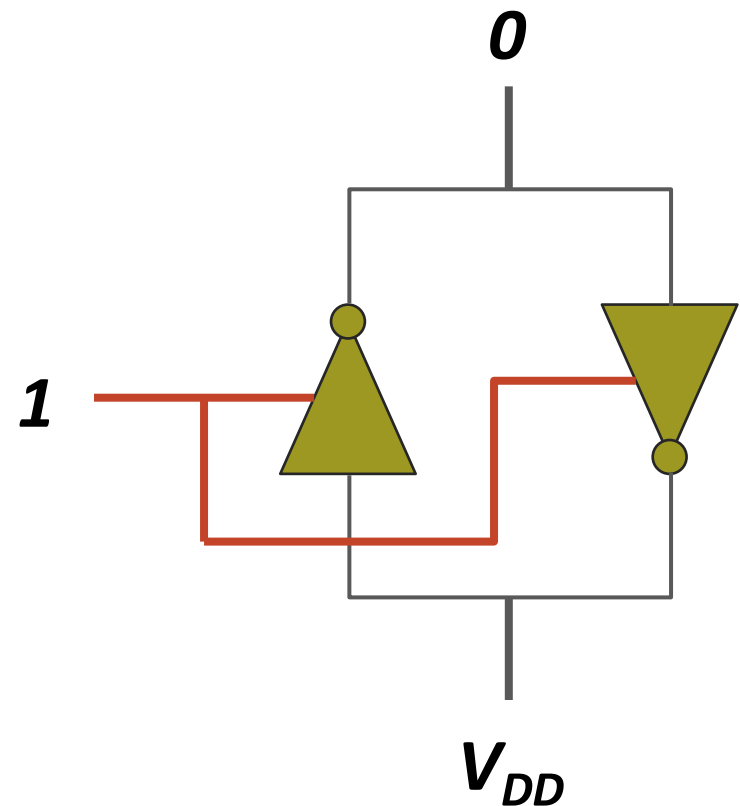
Sense Amplifier



Sense Amplifier – Two Stable States

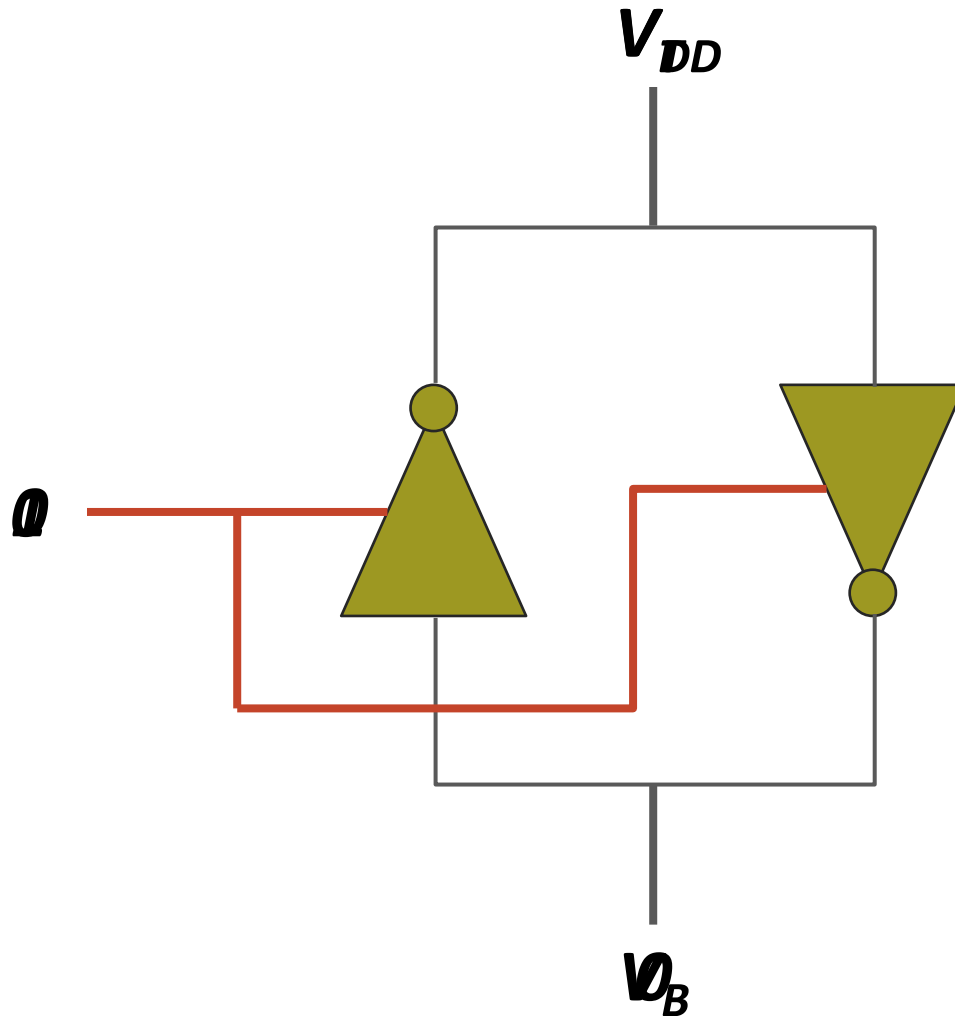


Logical "1"



Logical "0"

Sense Amplifier Operation

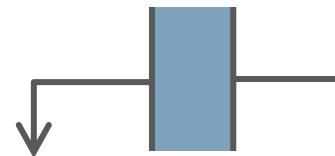


$$V_T > V_B$$

DRAM Cell – Capacitor



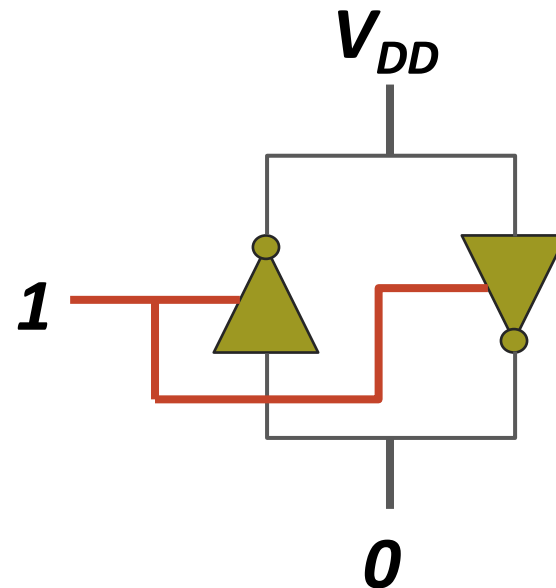
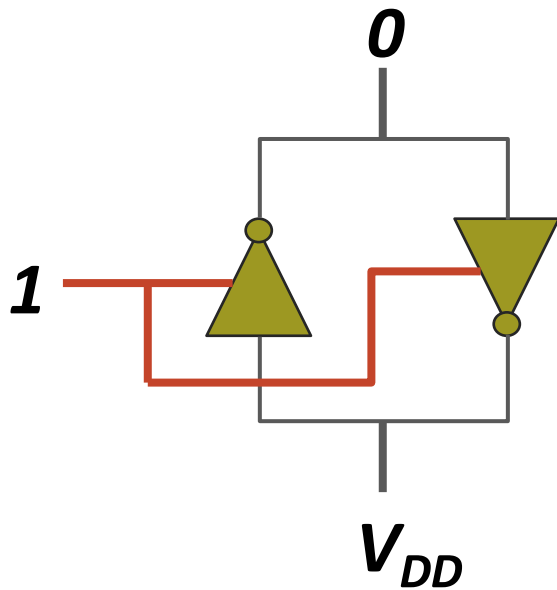
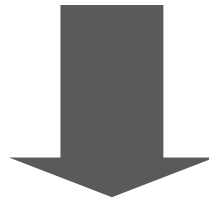
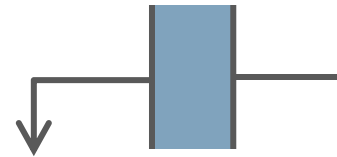
Empty State
Logical “0”



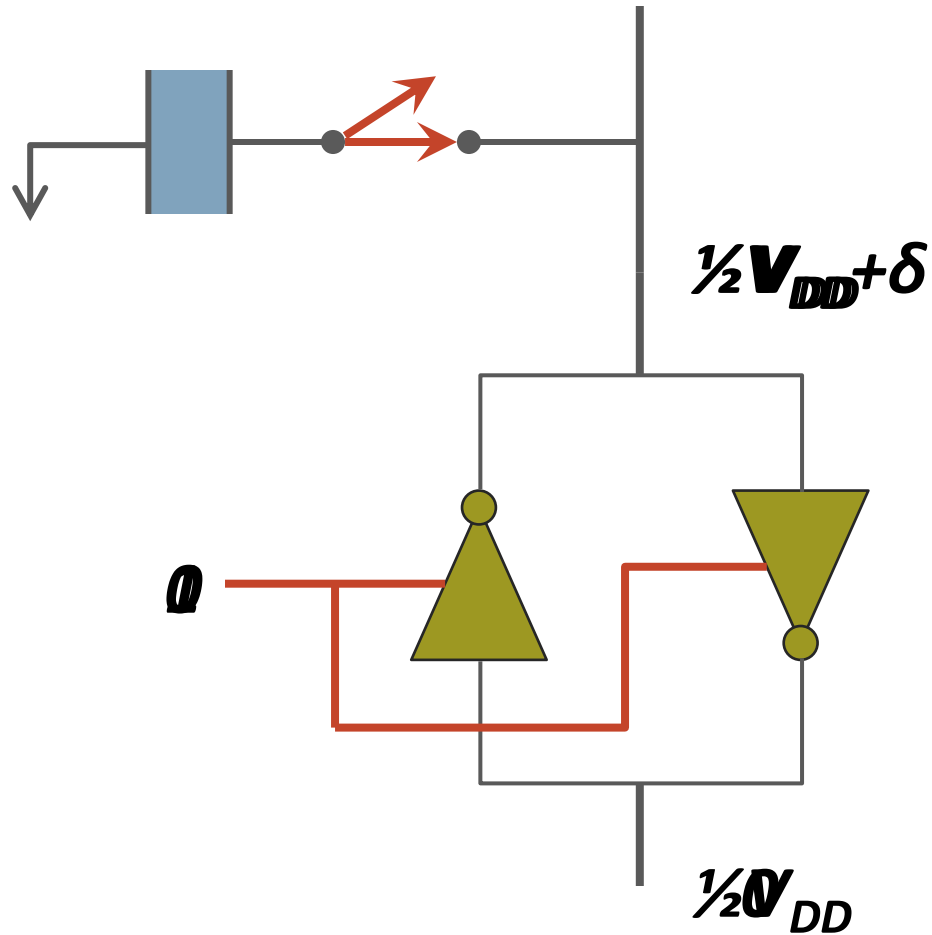
Fully Charged State
Logical “1”

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

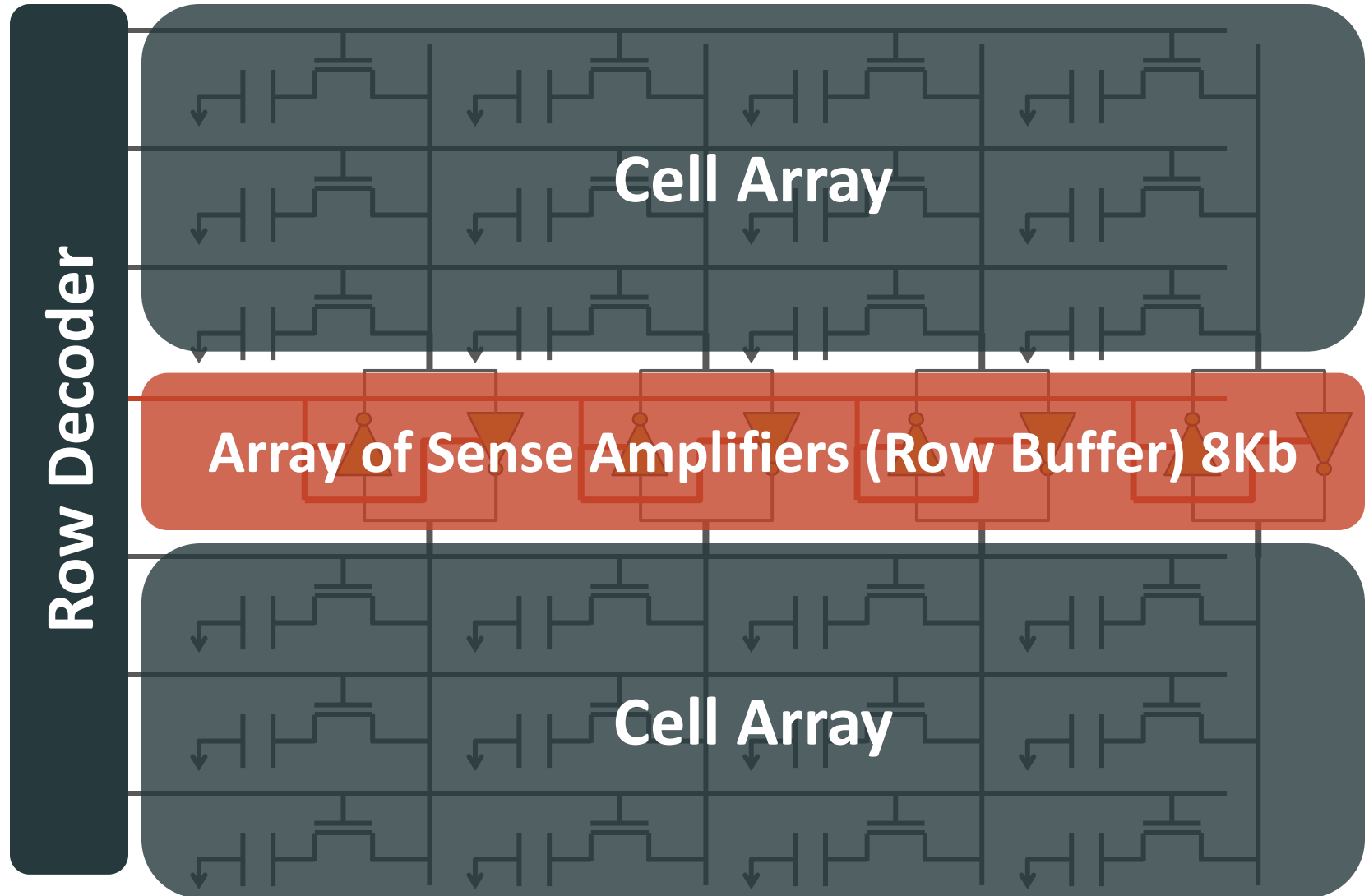
Capacitor to Sense Amplifier



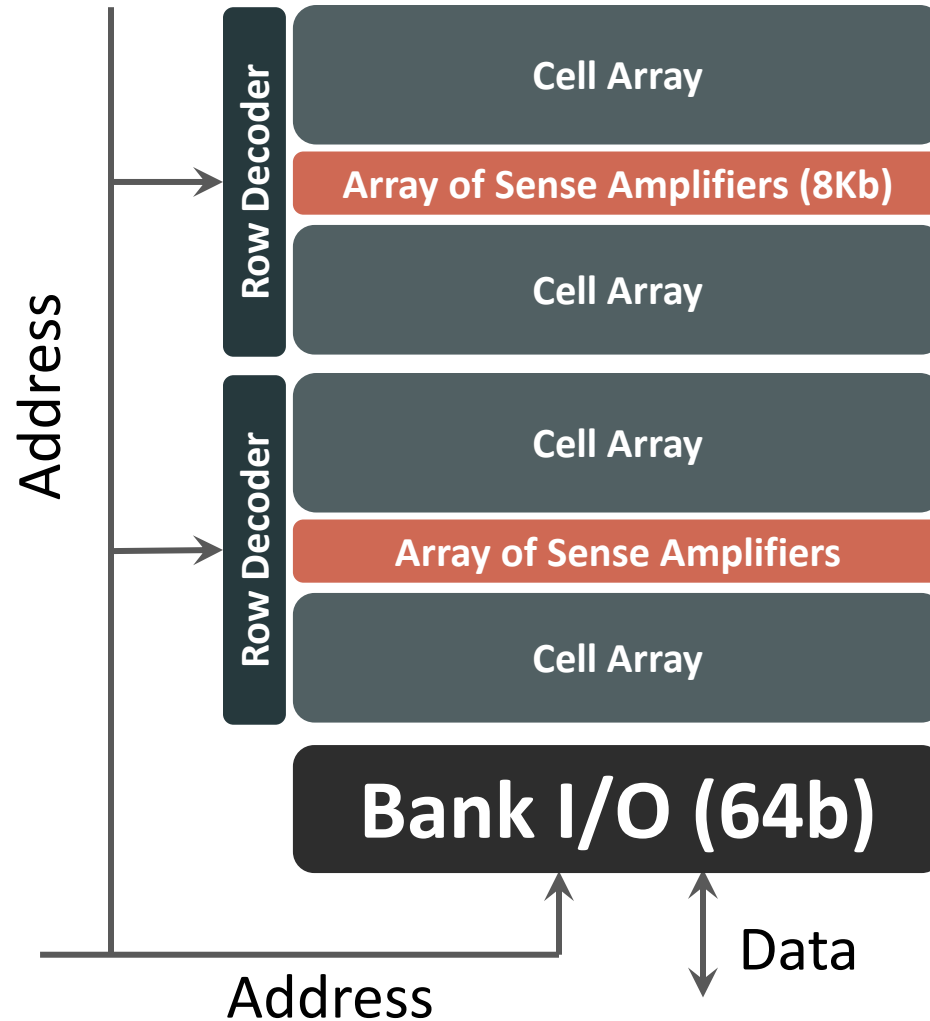
DRAM Cell Operation



DRAM Subarray – Building Block for DRAM Chip

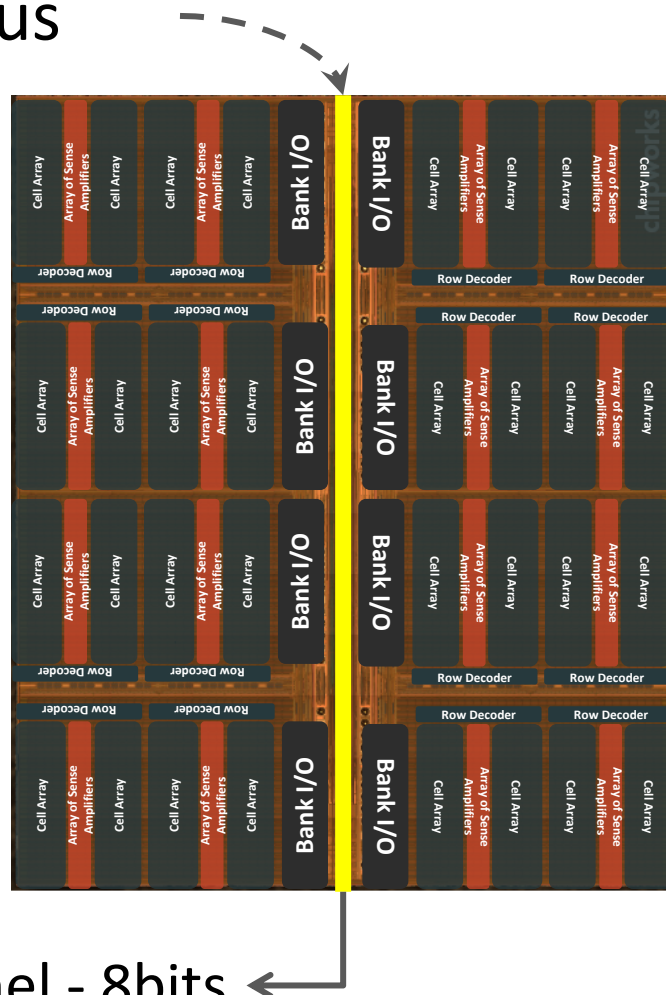


DRAM Bank



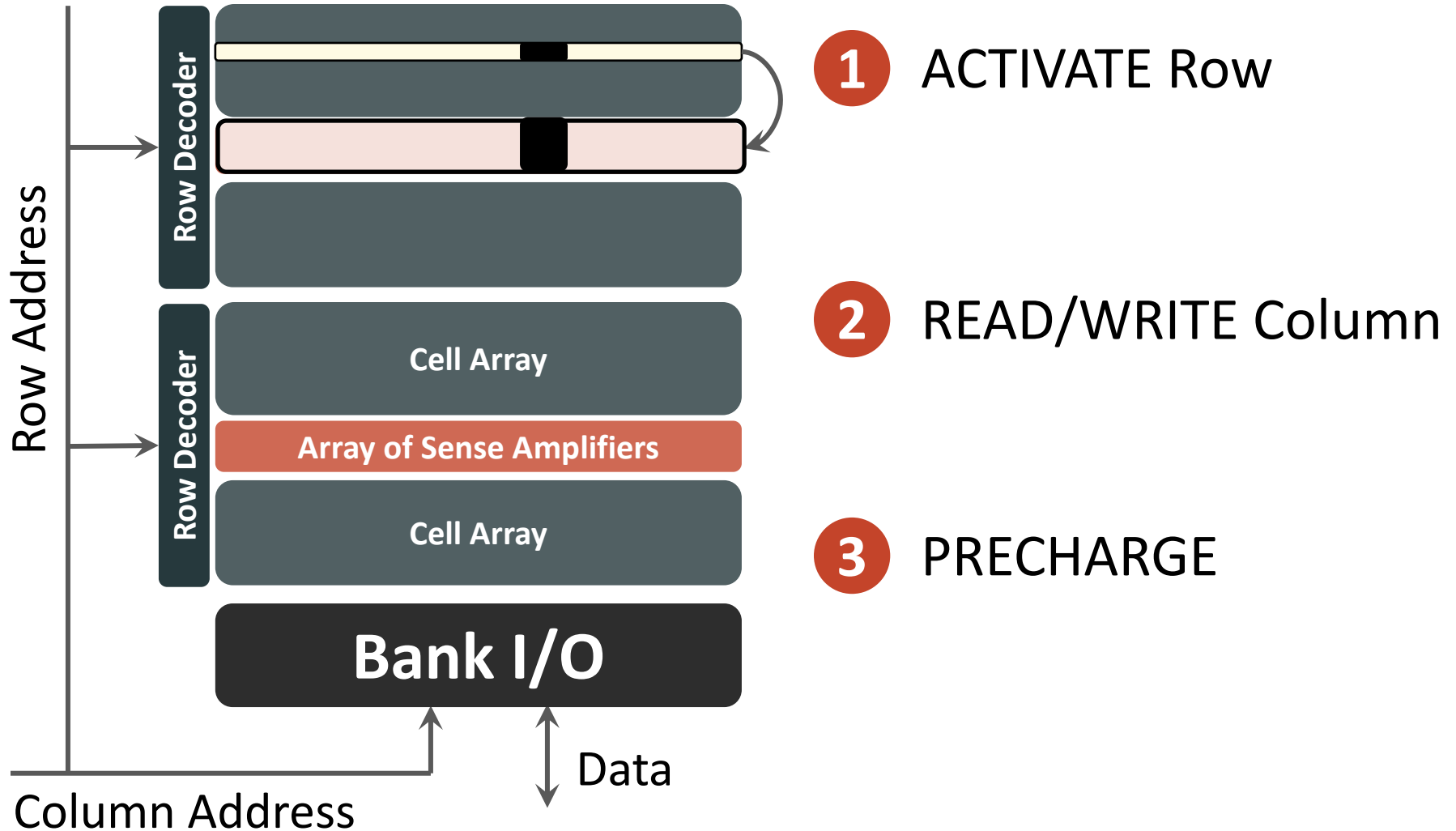
DRAM Chip

Shared internal bus



Memory channel - 8bits

DRAM Operation



End of Backup Slides