Memory Systems
and Memory-Centric Computing Systems

Part 4: Low-Latency Memory

Prof. Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu
7 July 2019
SAMOS Tutorial
Four Key Issues in Future Platforms

- Fundamentally **Secure/Reliable/Safe** Architectures

- Fundamentally **Energy-Efficient** Architectures
  - Memory-centric (Data-centric) Architectures

- Fundamentally **Low-Latency** Architectures

- Architectures for **Genomics, Medicine, Health**
Solving the Hardest Problems

Source: http://spectrum.ieee.org/image/MjYzMzAyMg.jpeg
Maslow’s Hierarchy of Needs, A Third Time


Source: https://www.simplypsychology.org/maslow.html
Challenge and Opportunity for Future

Fundamentally Low-Latency Computing Architectures
Main Memory Latency Lags Behind

- Capacity: 128x
- Bandwidth: 20x
- Latency: 1.3x

Memory latency remains almost constant.
A Closer Look …

Figure 1: DRAM latency trends over time [20, 21, 23, 51].

DRAM Latency Is Critical for Performance

**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]
DRAM Latency Is Critical for Performance

In-memory Databases

Graph/Tree Processing

Long memory latency → performance bottleneck

In-Memory Data Analytics
[Clapp+ (Intel), IISWC’15; Awan+, BDCloud’15]

Datacenter Workloads
[Kanev+ (Google), ISCA’15]
The Memory Latency Problem

- High memory latency is a significant \textit{limiter of system performance and energy-efficiency}.

- It is becoming increasingly so with \textit{higher memory contention} in multi-core and heterogeneous architectures:
  - Exacerbating the bandwidth need
  - Exacerbating the QoS problem

- It increases \textit{processor design complexity} due to the mechanisms incorporated to tolerate memory latency.
Retrospective: Conventional Latency Tolerance Techniques

- Caching [initially by Wilkes, 1965]
  - Widely used, simple, effective, but inefficient, passive
  - Not all applications/phases exhibit temporal or spatial locality

- Prefetching [initially in IBM 360/91, 1967]
  - Works well for regular memory access patterns
  - Prefetching irregular access patterns is difficult, inaccurate, and hardware-intensive

- Multithreading [initially in CDC 6600, 1964]
  - Works well if there are multiple threads
  - Improving single thread performance using multithreading hardware is an ongoing research effort

- Out-of-order execution [initially by Tomasulo, 1967]
  - Tolerates cache misses that cannot be prefetched
  - Requires extensive hardware resources for tolerating long latencies

None of These Fundamentally Reduce Memory Latency
Runahead Execution
Perfect Caches:
Load 1 Miss
Load 2 Hit

Small OoO Instruction Window:
Load 1 Miss
Load 2 Miss

Runahead:
Load 1 Miss
Load 2 Miss
Load 1 Hit
Load 2 Hit

Saved Cycles

Runahead Execution Example
Effect of Runahead in Sun ROCK

- Shailender Chaudhry talk, Aug 2008.
More on Runahead Execution

- Onur Mutlu, Jared Stark, Chris Wilkerson, and Yale N. Patt, "Runahead Execution: An Alternative to Very Large Instruction Windows for Out-of-order Processors"

More on Runahead Execution (Short)

Runahead Readings

- Required

- Recommended
Retrospective: Conventional Latency Tolerance Techniques

- Caching [initially by Wilkes, 1965]
  - Widely used, simple, effective, but inefficient, passive
  - Not all applications/phases exhibit temporal or spatial locality

- Prefetching [initially in IBM 360/91, 1967]
  - Works well for regular memory access patterns
  - Prefetching irregular access patterns is difficult, inaccurate, and hardware-intensive

- Multithreading [initially in CDC 6600, 1964]
  - Works well if there are multiple threads
  - Improving single thread performance using multithreading hardware is an ongoing research effort

Out-of-order execution [initially by Tomasulo, 1967]
  - Tolerates cache misses that cannot be prefetched
  - Requires extensive hardware resources for tolerating long latencies
Two Major Sources of Latency Inefficiency

- Modern DRAM is not designed for low latency
  - Main focus is cost-per-bit (capacity)

- Modern DRAM latency is determined by worst case conditions and worst case devices
  - Much of memory latency is unnecessary

Our Goal: Reduce Memory Latency at the Source of the Problem
Truly Reducing Memory Latency
Two Major Sources of Latency Inefficiency

- Modern DRAM is **not** designed for low latency
  - Main focus is cost-per-bit (capacity)

- Modern DRAM latency is determined by **worst case** conditions and **worst case** devices
  - Much of memory latency is unnecessary

**Our Goal:** Reduce Memory Latency at the Source of the Problem
Why the Long Memory Latency?

- **Reason 1: Design of DRAM Micro-architecture**
  - Goal: Maximize capacity/area, not minimize latency

- **Reason 2: “One size fits all” approach to latency specification**
  - Same latency parameters for all temperatures
  - Same latency parameters for all DRAM chips
  - Same latency parameters for all parts of a DRAM chip
  - Same latency parameters for all supply voltage levels
  - Same latency parameters for all application data
  - ...

SAFARI
Tackling the Fixed Latency Mindset

- Reliable operation latency is actually very heterogeneous
  - Across temperatures, chips, parts of a chip, voltage levels, ...

- Idea: Dynamically find out and use the lowest latency one can reliably access a memory location with
  - Adaptive-Latency DRAM [HPCA 2015]
  - Flexible-Latency DRAM [SIGMETRICS 2016]
  - Design-Induced Variation-Aware DRAM [SIGMETRICS 2017]
  - Voltron [SIGMETRICS 2017]
  - DRAM Latency PUF [HPCA 2018]
  - DRAM Latency True Random Number Generator [HPCA 2019]
  - ...

- We would like to find sources of latency heterogeneity and exploit them to minimize latency (or create other benefits)
Latency Variation in Memory Chips

Heterogeneous manufacturing & operating conditions → latency variation in timing parameters

DRAM A
DRAM B
DRAM C

Low
High

DRAM Latency

Slow cells
Why is Latency High?

- **DRAM latency**: Delay as specified in DRAM standards
  - Doesn’t reflect true DRAM device latency
- Imperfect manufacturing process $\rightarrow$ latency variation
- **High standard latency** chosen to increase yield

![Diagram showing DRAM manufacturing variation and standard latency](image-url)
What Causes the Long Memory Latency?

- **Conservative timing margins!**

- DRAM timing parameters are set to cover the worst case

- **Worst-case temperatures**
  - 85 degrees vs. common-case
  - to enable a wide range of operating conditions

- **Worst-case devices**
  - DRAM cell with smallest charge across any acceptable device
  - to tolerate process variation at acceptable yield

- This leads to large timing margins for the common case
Understanding and Exploiting Variation in DRAM Latency

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

[github.com/CMU-SAFARI/SoftMC](https://github.com/CMU-SAFARI/SoftMC)
SoftMC: Open Source DRAM Infrastructure

- [https://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\textsuperscript{1,2,3} Nandita Vijaykumar\textsuperscript{3} Samira Khan\textsuperscript{4,3} Saugata Ghose\textsuperscript{3} Kevin Chang\textsuperscript{3} Gennady Pekhimenko\textsuperscript{5,3} Donghyuk Lee\textsuperscript{6,3} Oguz Ergin\textsuperscript{2} Onur Mutlu\textsuperscript{1,3}

\textsuperscript{1}ETH Zürich \hspace{1cm} \textsuperscript{2}TOBB University of Economics & Technology \hspace{1cm} \textsuperscript{3}Carnegie Mellon University
\textsuperscript{4}University of Virginia \hspace{1cm} \textsuperscript{5}Microsoft Research \hspace{1cm} \textsuperscript{6}NVIDIA Research
Adaptive-Latency DRAM

- **Key idea**
  - Optimize DRAM timing parameters online

- **Two components**
  - DRAM manufacturer provides multiple sets of reliable DRAM timing parameters at different temperatures for each DIMM
  - System monitors DRAM temperature & uses appropriate DRAM timing parameters

Latency Reduction Summary of 115 DIMMs

- Latency reduction for read & write (55°C)
  - Read Latency: 32.7%
  - Write Latency: 55.1%

- Latency reduction for each timing parameter (55°C)
  - Sensing: 17.3%
  - Restore: 37.3% (read), 54.8% (write)
  - Precharge: 35.2%

AL-DRAM: Real System Evaluation

• **System**
  – **CPU**: AMD 4386 (8 Cores, 3.1GHz, 8MB LLC)

---

**D18F2x200 dct[0]_mp[1:0] DDR3 DRAM Timing 0**

Reset: 0F05_0505h. See 2.9.3 [DCT Configuration Registers].

<table>
<thead>
<tr>
<th>Bits</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>31:30</td>
<td>Reserved.</td>
</tr>
<tr>
<td>29:24</td>
<td><strong>Tras</strong>: row active strobe. Read-write. BIOS: See 2.9.7.5 [SPD ROM-Based Configuration]. Specifies the minimum time in memory clock cycles from an activate command to a precharge command, both to the same chip select bank.</td>
</tr>
<tr>
<td>07h-00h</td>
<td>Reserved</td>
</tr>
<tr>
<td>2Ah-08h</td>
<td>&lt;Tras&gt; clocks</td>
</tr>
<tr>
<td>3Fh-2Bh</td>
<td>Reserved</td>
</tr>
<tr>
<td>23:21</td>
<td>Reserved.</td>
</tr>
<tr>
<td>20:16</td>
<td><strong>Trp</strong>: row precharge time. Read-write. BIOS: See 2.9.7.5 [SPD ROM-Based Configuration]. Specifies the minimum time in memory clock cycles from a precharge command to an activate command or auto refresh refresh command, both to the same bank.</td>
</tr>
</tbody>
</table>
**AL-DRAM: Single-Core Evaluation**

![Bar Chart]

**AL-DRAM improves single-core performance on a real system**

**SAFARI**
AL-DRAM: Multi-Core Evaluation

AL-DRAM provides higher performance on multi-programmed & multi-threaded workloads
Reducing Latency Also Reduces Energy

- AL-DRAM reduces DRAM power consumption by 5.8%
- Major reason: reduction in row activation time
AL-DRAM: Advantages & Disadvantages

- **Advantages**
  - Simple mechanism to reduce latency
  - Significant system performance and energy benefits
    - Benefits higher at low temperature
  - Low cost, low complexity

- **Disadvantages**
  - Need to determine reliable operating latencies for different temperatures and different DIMMs \(\rightarrow\) higher testing cost
    (might not be that difficult for low temperatures)
More on Adaptive-Latency DRAM


[Slides (pptx) (pdf)] [Full data sets]

Adaptive-Latency DRAM: Optimizing DRAM Timing for the Common-Case

Donghyuk Lee   Yoongu Kim   Gennady Pekhimenko
Samira Khan    Vivek Seshadri  Kevin Chang   Onur Mutlu

Carnegie Mellon University
Different Types of Latency Variation

- AL-DRAM exploits latency variation
  - Across time (different temperatures)
  - Across chips

- Is there also latency variation within a chip?
  - Across different parts of a chip
Activation errors are concentrated at certain columns of cells
Heterogeneous Latency within A Chip

• **Observation:** DRAM timing errors (slow DRAM cells) are concentrated on certain regions

• **Flexible-Latency (FLY) DRAM**
  – A software-transparent design that reduces latency

• **Key idea:**
  1) Divide memory into regions of different latencies
  2) **Memory controller:** Use lower latency for regions without slow cells; higher latency for other regions

Chang+, “*Understanding Latency Variation in Modern DRAM Chips: Experimental Characterization, Analysis, and Optimization*”, SIGMETRICS 2016.
Heterogeneous Latency within A Chip

FLY-DRAM: Advantages & Disadvantages

- **Advantages**
  + Reduces latency significantly
    + Exploits significant within-chip latency variation

- **Disadvantages**
  - Need to determine reliable operating latencies for different parts of a chip → higher testing cost
  - Slightly more complicated controller
Analysis of Latency Variation in DRAM Chips

Kevin Chang, Abhijith Kashyap, Hasan Hassan, Samira Khan, Kevin Hsieh, Donghyuk Lee, Saugata Ghose, Gennady Pekhimenko, Tianshi Li, and Onur Mutlu,

"Understanding Latency Variation in Modern DRAM Chips: Experimental Characterization, Analysis, and Optimization"


[Slides (pptx) (pdf)]
[Source Code]
Why Is There Spatial Latency Variation Within a Chip?
What Is Design-Induced Variation?

Systematic variation in cell access times caused by the physical organization of DRAM

across column
distance from wordline driver

Inherently fast

fast
slow

inherently slow

across row
distance from sense amplifier

wordline drivers

sense amplifiers
**DIVA Online Profiling**

*Design-Induced-Variation-Aware*

Profile *only slow regions* to determine min. latency

→ *Dynamic & low cost* latency optimization
DIVA Online Profiling

Design-Induced-Variation-Aware

- Slow cells
- Process variation
- Random error

Error-correcting code

Wordline driver

Sense amplifier

Inherently slow design-induced variation localized error

Online profiling

Combine error-correcting codes & online profiling

→ Reliably reduce DRAM latency

SAFARI
DIVA-DRAM reduces latency more aggressively and uses ECC to correct random slow cells.
DIVA-DRAM: Advantages & Disadvantages

- **Advantages**
  ++ Automatically finds the lowest reliable operating latency at system runtime (lower production-time testing cost)
  + Reduces latency more than prior methods (w/ ECC)
  + Reduces latency at high temperatures as well

- **Disadvantages**
  - Requires knowledge of inherently-slow regions
  - Requires ECC (Error Correcting Codes)
  - Imposes overhead during runtime profiling
Design-Induced Latency Variation in DRAM

- Donghyuk Lee, Samira Khan, Lavanya Subramanian, Saugata Ghose, Rachata Ausavarungnirun, Gennady Pekhimenko, Vivek Seshadri, and Onur Mutlu,

"Design-Induced Latency Variation in Modern DRAM Chips: Characterization, Analysis, and Latency Reduction Mechanisms"

Design-Induced Latency Variation in Modern DRAM Chips: Characterization, Analysis, and Latency Reduction Mechanisms

Donghyuk Lee, NVIDIA and Carnegie Mellon University
Samira Khan, University of Virginia
Lavanya Subramanian, Saugata Ghose, Rachata Ausavarungnirun, Carnegie Mellon University
Gennady Pekhimenko, Vivek Seshadri, Microsoft Research
Onur Mutlu, ETH Zürich and Carnegie Mellon University
Understanding & Exploiting the Voltage-Latency-Reliability Relationship
High DRAM Power Consumption

- **Problem**: High DRAM (memory) power in today’s systems

>40% in POWER7 \(^{\text{Ware+}, \text{HPCA’10}}\) \quad >40% in GPU \(^{\text{Paul+}, \text{ISCA’15}}\)
Executive Summary

• DRAM (memory) power is significant in today’s systems
  – Existing low-voltage DRAM reduces voltage conservatively

• Goal: Understand and exploit the reliability and latency behavior of real DRAM chips under aggressive reduced-voltage operation

• Key experimental observations:
  – Huge voltage margin -- Errors occur beyond some voltage
  – Errors exhibit spatial locality
  – Higher operation latency mitigates voltage-induced errors

• Voltron: A new DRAM energy reduction mechanism
  – Reduce DRAM voltage without introducing errors
  – Use a regression model to select voltage that does not degrade performance beyond a chosen target → 7.3% system energy reduction
Custom Testing Platform

**SoftMC** [Hassan+, HPCA’17]: FPGA testing platform to

1. Adjust supply voltage to DRAM modules
2. Schedule DRAM commands to DRAM modules

Existing systems: DRAM commands not exposed to users

https://github.com/CMU-SAFARI/DRAM-Voltage-Study
Reliability Worsens with Lower Voltage

Errors induced by reduced-voltage operation

Reducing voltage below $V_{\text{min}}$ causes an increasing number of errors
Source of Errors

Detailed circuit simulations (SPICE) of a DRAM cell array to model the behavior of DRAM operations

https://github.com/CMU-SAFARI/DRAM-Voltage-Study

Reliable low-voltage operation requires higher latency
Higher Access Latency → Fewer Errors

Measured minimum latency that does not cause errors in DRAM modules

Distribution of latency in the total population

40% of modules

100% of modules

DRAM requires longer latency to access data without errors at lower voltage
Spatial Locality of Errors

A module under 1.175V (12% voltage reduction)

Errors concentrate in certain regions
Voltron Overview

User specifies the performance loss target

Select the **minimum** DRAM voltage without violating the target

How do we predict performance loss due to increased latency under low DRAM voltage?
Linear Model to Predict Performance

User specifies the performance loss target

Select the minimum DRAM voltage without violating the target

Application’s characteristics

[1.3V, 1.25V, ...] DRAM Voltage

Predicted performance loss

[-1%, -3%, ...]

Min. Voltage

Target

Final Voltage

Linear regression model
Energy Savings with Bounded Performance

MemDVFS  Voltron
[David+, ICAC’11]

More savings for high bandwidth applications

7.3%

3.2%

CPU+DRAM Energy Savings (%)

Low  High

Memory Intensity

Performance Loss (%)

Meets performance target

-1.6%  -1.8%

Performance Target

Low  High

Memory Intensity
Voltron: Advantages & Disadvantages

- **Advantages**
  - Can trade-off between voltage and latency to improve energy or performance
  - Can exploit the high voltage margin present in DRAM

- **Disadvantages**
  - Requires finding the reliable operating voltage for each chip → higher testing cost
Analysis of Latency-Voltage in DRAM Chips

Kevin Chang, A. Giray Yaglikci, Saugata Ghose, Aditya Agrawal, Niladrish Chatterjee, Abhijith Kashyap, Donghyuk Lee, Mike O'Connor, Hasan Hassan, and Onur Mutlu,
"Understanding Reduced-Voltage Operation in Modern DRAM Devices: Experimental Characterization, Analysis, and Mechanisms"
And, What If …

- ... we can sacrifice reliability of some data to access it with even lower latency?
The DRAM Latency PUF:
Quickly Evaluating Physical Unclonable Functions by Exploiting the Latency-Reliability Tradeoff in Modern Commodity DRAM Devices

Jeremie S. Kim  Minesh Patel
Hasan Hassan  Onur Mutlu

SAFARI
Systems@ETH Zürich
ETH Zürich
Carnegie Mellon
Motivation

• A **PUF** is a function that generates a signature unique to a given device

• Used in a **Challenge-Response Protocol**
  - Each device generates a unique **PUF response** depending on the inputs
  - A trusted server **authenticates** a device if it generates the expected PUF response
DRAM Latency Characterization of 223 LPDDR4 DRAM Devices

- Latency failures come from accessing DRAM with reduced timing parameters.

- Key Observations:
  1. A cell’s latency failure probability is determined by random process variation
  2. Latency failure patterns are repeatable and unique to a device
DRAM Latency PUF Key Idea

High % chance to fail with reduced $t_{RCD}$

Low % chance to fail with reduced $t_{RCD}$
DRAM Accesses and Failures

Process variation during manufacturing leads to cells having unique characteristics.

Bitline charge sharing

Ready to Access Voltage Level

V_{dd}, V_{min}, 0.5 \times V_{dd}

Time

t_{RCD}

ACTIVATE, SA Enable, READ
DRAM Accesses and Failures

weaker cells have a higher probability to fail
The DRAM Latency PUF Evaluation

• We generate PUF responses using *latency errors* in a region of DRAM

• The latency error patterns *satisfy PUF requirements*

• The DRAM Latency PUF *generates PUF responses in 88.2ms*
Results

- DL-PUF is **orders of magnitude faster** than prior DRAM PUFs & temperature independent
The DRAM Latency PUF:
Quickly Evaluating Physical Unclonable Functions by Exploiting the Latency-Reliability Tradeoff in Modern Commodity DRAM Devices

Jeremie S. Kim    Minesh Patel
Hasan Hassan      Onur Mutlu

QR Code for the paper
DRAM Latency PUFs

- Jeremie S. Kim, Minesh Patel, Hasan Hassan, and Onur Mutlu, "The DRAM Latency PUF: Quickly Evaluating Physical Unclonable Functions by Exploiting the Latency-Reliability Tradeoff in Modern DRAM Devices"


[Lightning Talk Video]
[Slides (pptx) (pdf)] [Lightning Session Slides (pptx) (pdf)]
**D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput**

Jeremie S. Kim    Minesh Patel
Hasan Hassan      Lois Orosa    Onur Mutlu

HPCA 2019

**SAFARI**

ETH Zürich

Carnegie Mellon
DRAM Latency Characterization of 282 LPDDR4 DRAM Devices

• Latency failures come from accessing DRAM with reduced timing parameters.

• Key Observations:
  1. A cell’s latency failure probability is determined by random process variation
  2. Some cells fail randomly
D-RaNGe Key Idea

High % chance to fail with reduced $t_{RCD}$

Low % chance to fail with reduced $t_{RCD}$

Fails randomly with reduced $t_{RCD}$
D-RaNGe Key Idea

We refer to cells that fail randomly when accessed with a reduced $t_{\text{RCD}}$ as RNG cells.
Our D-RaNGe Evaluation

• We generate random values by repeatedly accessing RNG cells and aggregating the data read.

• The random data satisfies the NIST statistical test suite for randomness.

• The D-RaNGe generates random numbers:
  - Throughput: 717.4 Mb/s
  - Latency: 64 bits in <1us
  - Power: 4.4 nJ/bit
D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput

Jeremie S. Kim    Minesh Patel
Hasan Hassan      Lois Orosa    Onur Mutlu

SAFARI

HPCA 2019

ETH Zürich
Carnegie Mellon
DRAM Latency True Random Number Generator

Jeremie S. Kim, Minesh Patel, Hasan Hassan, Lois Orosa, and Onur Mutlu, "D-RaNGe: Using Commodity DRAM Devices to Generate True Random Numbers with Low Latency and High Throughput"

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

Jeremie S. Kim‡$, Minesh Patel§, Hasan Hassan§, Lois Orosa§, Onur Mutlu$‡
‡Carnegie Mellon University †ETH Zürich
Other Ideas on Reducing DRAM Latency
Solar-DRAM: Exploiting Spatial Variation

Hasan Hassan, Gennady Pekhimenko, Nandita Vijaykumar, Vivek Seshadri, Donghyuk Lee, Oguz Ergin, and Onur Mutlu,
"ChargeCache: Reducing DRAM Latency by Exploiting Row Access Locality"
[Slides (pptx) (pdf)]
[Source Code]
Reducing Refresh Latency

- Anup Das, Hasan Hassan, and Onur Mutlu, "VRL-DRAM: Improving DRAM Performance via Variable Refresh Latency"

Proceedings of the 55th Design Automation Conference (DAC), San Francisco, CA, USA, June 2018.

VRL-DRAM: Improving DRAM Performance via Variable Refresh Latency

Anup Das  
Drexel University  
Philadelphia, PA, USA  
anup.das@drexel.edu

Hasan Hassan  
ETH Zürich  
Zürich, Switzerland  
hhasan@ethz.ch

Onur Mutlu  
ETH Zürich  
Zürich, Switzerland  
onutlu@gmail.com
Why the Long Memory Latency?

- **Reason 1: Design of DRAM Micro-architecture**
  - Goal: Maximize capacity/area, not minimize latency

- **Reason 2: “One size fits all” approach to latency specification**
  - Same latency parameters for all temperatures
  - Same latency parameters for all DRAM chips
  - Same latency parameters for all parts of a DRAM chip
  - Same latency parameters for all supply voltage levels
  - Same latency parameters for all application data
  - ...
Tiered-Latency DRAM

- Donghyuk Lee, Yoongu Kim, Vivek Seshadri, Jamie Liu, Lavanya Subramanian, and Onur Mutlu,

"Tiered-Latency DRAM: A Low Latency and Low Cost DRAM Architecture"

Proceedings of the 19th International Symposium on High-Performance Computer Architecture (HPCA), Shenzhen, China, February 2013. Slides (pptx)
LISA: Low-cost Inter-linked Subarrays

Kevin K. Chang, Prashant J. Nair, Saugata Ghose, Donghyuk Lee, Moinuddin K. Qureshi, and Onur Mutlu,
"Low-Cost Inter-Linked Subarrays (LISA): Enabling Fast Inter-Subarray Data Movement in DRAM"
[Slides (pptx) (pdf)]
[Source Code]

Low-Cost Inter-Linked Subarrays (LISA): Enabling Fast Inter-Subarray Data Movement in DRAM

Kevin K. Chang†, Prashant J. Nair*, Donghyuk Lee†, Saugata Ghose†, Moinuddin K. Qureshi*, and Onur Mutlu†
†Carnegie Mellon University  * Georgia Institute of Technology
Tiered Latency DRAM
What Causes the Long Latency?

DRAM Latency = Subarray Latency + I/O Latency

Dominant
Why is the Subarray So Slow?

- Long bitline
  - Amortizes sense amplifier cost → Small area
  - Large bitline capacitance → High latency & power
Trade-Off: Area (Die Size) vs. Latency

Long Bitline vs. Short Bitline:
- Faster vs. Smaller

Trade-Off: Area vs. Latency
Trade-Off: Area (Die Size) vs. Latency

Normalized DRAM Area

Latency (ns)

Cheaper

Faster

GOAL

32

64

128

256

512 cells/bitline

Fancy DRAM Short Bitline

Commodity DRAM Long Bitline

Cheaper

Faster
Approximating the Best of Both Worlds

- **Long Bitline**
  - Small Area
  - High Latency

- **Our Proposal**
  - Need Isolation
  - Add Isolation Transistors

- **Short Bitline**
  - Large Area
  - Low Latency
  - Fast Bitline

---

96
Approximating the Best of Both Worlds

Long Bitline Tiered-Latency DRAM Short Bitline

Small Area

Small Area

Large Area

High Latency

Low Latency

Low Latency

Small area using long bitline
Commodity DRAM vs. TL-DRAM [HPCA 2013]

- DRAM Latency ($t_{RC}$)
- DRAM Power

![Latency Comparison](chart)

<table>
<thead>
<tr>
<th></th>
<th>Commodity DRAM</th>
<th>Near TL-DRAM</th>
<th>Far TL-DRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latency (ns)</td>
<td>(52.5)</td>
<td>-56%</td>
<td>+23%</td>
</tr>
<tr>
<td>Power</td>
<td>-51%</td>
<td>+49%</td>
<td></td>
</tr>
</tbody>
</table>

- DRAM Area Overhead

~3%: mainly due to the isolation transistors
Trade-Off: Area (Die-Area) vs. Latency

- Cheaper
- Normalized DRAM Area

- Faster

Near Segment: 32 cells/bitline
- 64 cells/bitline
- 128 cells/bitline
- 256 cells/bitline
- 512 cells/bitline

Far Segment

GOAL
Leveraging Tiered-Latency DRAM

- TL-DRAM is a *substrate* that can be leveraged by the hardware and/or software

- Many potential uses
  1. Use near segment as hardware-managed *inclusive* cache to far segment
  2. Use near segment as hardware-managed *exclusive* cache to far segment
  3. Profile-based page mapping by operating system
  4. Simply replace DRAM with TL-DRAM

Near Segment as Hardware-Managed Cache

• **Challenge 1:** How to efficiently migrate a row between segments?

• **Challenge 2:** How to efficiently manage the cache?
Using near segment as a cache improves performance and reduces power consumption

More on TL-DRAM

- Donghyuk Lee, Yoongu Kim, Vivek Seshadri, Jamie Liu, Lavanya Subramanian, and Onur Mutlu,
"Tiered-Latency DRAM: A Low Latency and Low Cost DRAM Architecture"
Proceedings of the 19th International Symposium on High-Performance Computer Architecture (HPCA), Shenzhen, China, February 2013. Slides (pptx)
LISA: Low-Cost Inter-Linked Subarrays [HPCA 2016]
Problem: Inefficient Bulk Data Movement

Bulk data movement is a key operation in many applications

– `memmove` & `memcpy`: 5% cycles in Google's datacenter [Kanev+ ISCA’15]

Long latency and high energy
Goal: Provide a new substrate to enable wide connectivity between subarrays
Key Idea and Applications

• Low-cost Inter-linked subarrays (LISA)
  – Fast bulk data movement between subarrays
  – Wide datapath via isolation transistors: 0.8% DRAM chip area

• LISA is a versatile substrate → new applications
  Fast bulk data copy: Copy latency 1.363ms→0.148ms (9.2x)
  → 66% speedup, -55% DRAM energy
  In-DRAM caching: Hot data access latency 48.7ns→21.5ns (2.2x)
  → 5% speedup
  Fast precharge: Precharge latency 13.1ns→5.0ns (2.6x)
  → 8% speedup
3. Linked Precharge (LIP)

- **Problem:** The precharge time is limited by the strength of one precharge unit
- **Linked Precharge (LIP):** LISA precharges a subarray using multiple precharge units

Reduces precharge latency by 2.6x (43% guardband)
More on LISA

Kevin K. Chang, Prashant J. Nair, Saugata Ghose, Donghyuk Lee, Moinuddin K. Qureshi, and Onur Mutlu,

"Low-Cost Inter-Linked Subarrays (LISA): Enabling Fast Inter-Subarray Data Movement in DRAM"


[Slides (pptx) (pdf)]
[Source Code]
CROW: The Copy Row Substrate
[ISCA 2019]
Challenges of DRAM Scaling

1. access latency
2. refresh overhead
3. exposure to vulnerabilities
Conventional DRAM

[Diagram of DRAM Subarray with row decoder and sense amplifiers]
Copy Row DRAM (CROW)

Row copy

Multiple row activation

DRAM Subarray

regular rows

copy rows

sense amplifier
Use Cases of CROW

- **CROW-cache**
  - reduces *access latency*

- **CROW-ref**
  - reduces DRAM *refresh overhead*

- A mechanism for protecting against *RowHammer*
Key Results

CROW-cache + CROW-ref
• 20% speedup
• 22% less DRAM energy

Hardware Overhead
• 0.5% DRAM chip area
• 1.6% DRAM capacity
• 11.3 KiB memory controller storage
More on CROW

- Hasan Hassan, Minesh Patel, Jeremie S. Kim, A. Giray Yaglikci, Nandita Vijaykumar, Nika Mansourighiasi, Saugata Ghose, and Onur Mutlu,

"CROW: A Low-Cost Substrate for Improving DRAM Performance, Energy Efficiency, and Reliability"
SALP: Reducing DRAM Bank Conflict Impact

Kim, Seshadri, Lee, Liu, Mutlu

A Case for Exploiting Subarray-Level Parallelism (SALP) in DRAM

ISCA 2012.
Problem: Bank conflicts are costly for performance and energy
  - serialized requests, wasted energy (thrashing of row buffer, busy wait)
Goal: Reduce bank conflicts without adding more banks (low cost)
Observation 1: A DRAM bank is divided into subarrays and each subarray has its own local row buffer
Observation 2: Subarrays are mostly independent
- Except when sharing global structures to reduce cost

Key Idea of SALP: Minimally reduce sharing of global structures
Reduce the sharing of ...
Global decoder $\rightarrow$ Enables almost parallel access to subarrays
Global row buffer $\rightarrow$ Utilizes multiple local row buffers
SALP: Reduce Sharing of Global Decoder

Instead of a global latch, have *per-subarray latches*.
SALP: Reduce Sharing of Global Row-Buffer

Selectively connect local row-buffers to global row-buffer using a *Designated* single-bit latch.

Selectively connect local row-buffers to global row-buffer using a *Designated* single-bit latch.
SALP: Baseline Bank Organization

- **Global Decoder**
- **Latch**
- **Local row-buffer**
- **Global bitlines**
- **Local row-buffer**
- **Global row-buffer**
SALP: Proposed Bank Organization

Global Decoder

Latch

Latch

Global bitlines

Local row-buffer

Local row-buffer

Global row-buffer

Overhead of SALP in DRAM chip: 0.15%

1. Global latch → per-subarray local latches
2. Designated bit latches and wire to selectively enable a subarray
SALP: Results

- Wide variety of systems with different #channels, banks, ranks, subarrays
- Server, streaming, random-access, SPEC workloads
- **Dynamic DRAM energy reduction:** 19%
  - DRAM row hit rate improvement: 13%
- **System performance improvement:** 17%
  - Within 3% of ideal (all independent banks)
- **DRAM die area overhead:** 0.15%
  - vs. 36% overhead of independent banks
More on SALP

- Yoongu Kim, Vivek Seshadri, Donghyuk Lee, Jamie Liu, and Onur Mutlu,

"A Case for Exploiting Subarray-Level Parallelism (SALP) in DRAM"
Proceedings of the 39th International Symposium on Computer Architecture (ISCA), Portland, OR, June 2012. Slides (pptx)
More on SALP

DRAM Process Scaling Challenges

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

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
More on SALP

Sub-array Level Parallelism with tWR Relaxation

- tWR relaxation
  - Relaxing tWR results in DRAM yield improvement but can degrade performance requiring new compensating features
  - By increasing tWR 5X (from 15ns to 75ns), fail bit counts are expected to reduce by 1 to 2 orders of magnitudes

- Sub-array level parallelism (SALP)
  - Allows a page in another sub-array in the same bank to be opened in parallel with the currently activated sub-array
  - Results in performance gain by increasing the row access parallelism within a bank
    ⇒ Used to compensate for the performance loss caused by tWR relaxation

Performance Impact of SALP and tWR relaxation

- Performance simulations run for various workloads when tWR is relaxed by 2X and 3X, and when SALP is applied with 2 sub-banks.

- Results show that performance is reduced by ~5% and ~2% in average if tWR is relaxed by 3X and 2X, respectively.

- Results also show that performance is compensated, and even improved to up to ~3% in average when SALP is applied, even with tWR relaxed by 3X.
Summary: Low-Latency Memory
Summary: Tackling Long Memory Latency

Reason 1: Design of DRAM Micro-architecture
- Goal: Maximize capacity/area, not minimize latency

Reason 2: “One size fits all” approach to latency specification
- Same latency parameters for all temperatures
- Same latency parameters for all DRAM chips
- Same latency parameters for all parts of a DRAM chip
- Same latency parameters for all supply voltage levels
- Same latency parameters for all application data
- ...
Challenge and Opportunity for Future

Fundamentally Low-Latency Computing Architectures
One Important Takeaway

Main Memory Needs
Intelligent Controllers
On DRAM Power Consumption
VAMPIRE DRAM Power Model

- Saugata Ghose, A. Giray Yaglikci, Raghav Gupta, Donghyuk Lee, Kais Kudrolli, William X. Liu, Hasan Hassan, Kevin K. Chang, Niladrish Chatterjee, Aditya Agrawal, Mike O'Connor, and Onur Mutlu,

"What Your DRAM Power Models Are Not Telling You: Lessons from a Detailed Experimental Study"


[Abstract]
More Motivation
to Reduce Memory Latency
Workload-DRAM Interaction Analysis

- Saugata Ghose, Tianshi Li, Nastaran Hajinazar, Damla Senol Cali, and Onur Mutlu, "Demystifying Workload–DRAM Interactions: An Experimental Study"
[Preliminary arXiv Version]
[Abstract]
[Slides (pptx) (pdf)]
Memory Systems
and Memory-Centric Computing Systems

Part 4: Low-Latency Memory

Prof. Onur Mutlu
omutlu@gmail.com
https://people.inf.ethz.ch/omutlu

7 July 2019
SAMOS Tutorial