Reducing DRAM Latency at Low Cost by Exploiting Heterogeneity

Donghyuk Lee

Carnegie Mellon University
Problem: High DRAM Latency

Major bottleneck for system performance
Historical DRAM Trends

**Goal**: Reduce DRAM latency at low cost

**DRAM latency continues to be a critical bottleneck**
Why is DRAM slow?
DRAM Cell Array: Mat

Wordline driver

Sense amplifier

Cell

Mat

Bitline

Wordline

Mat

SAFARI
Cell Array (Mat): High Latency
DRAM Cell Array: High Latency

Inside mat
- Narrow poly wire
  - Large resistance
  - Large capacitance
  ➔ Slow

Outside mat
- Thick metal wire
  - Small resistance
  - Small capacitance
  ➔ Fast

- Small cell
  - Difficult to detect data in small cell
  ➔ Slow

SAFARI
DRAM cell array (mat) is the dominant latency bottleneck due to three reasons
1. Long Narrow Wires

Long narrow wires:
enables small area,
increases latency
2. Operating Conditions

Operating conditions:
- differing latencies,
- uses the same standard value optimized for the worst case

Example:
- small cell vs. normal cell
- hot vs. cool
3. Distance from Peripheral Logic

Distance from peripheral logic: differing latencies uses the same standard value optimized for the farthest cell

e.g., near cell vs. far cell
Three Sources of High Latency

1. Long narrow wires
   - TL-DRAM

2. Operating conditions
   - AL-DRAM

3. Distance from peripheral logic
   - AVA-DRAM

Goal: Reduce DRAM latency at low cost with three approaches
Thesis Statement

**DRAM latency can be reduced by enabling and exploiting latency heterogeneity in DRAM**
Outline

1. TL-DRAM  Reducing DRAM Latency by Modifying Bitline Architecture
2. AL-DRAM  Optimizing DRAM Latency for the Common-Case
3. AVA-DRAM  Lowering DRAM Latency by Exploiting Architectural Variation

Prior Work

Future Research Direction
Long Bitline $\Rightarrow$ High Latency

Long Bitline: Amortize sense amplifier $\Rightarrow$ Small area

Long Bitline: Large bitline cap. $\Rightarrow$ High latency
Trade-Off: Area vs. Latency

Long Bitline

Short Bitline

Faster

Smaller

Trade-Off: Area vs. Latency
Trade-Off: Area vs. Latency

Normalized DRAM Area

Latency (ns)

- Fancy DRAM (Short Bitline)
- Commodity DRAM (Long Bitline)

512 cells/bitline

GOAL

Faster

Cheaper
Approximating Best of Both Worlds

- Long Bitline
  - Small Area
  - High Latency

- Short Bitline
  - Large Area
  - Low Latency

Our Proposal

- Need Isolation
- Add Isolation Transistors
- Short Bitline → Fast
Approximating Best of Both Worlds

Long Bitline Tiered-Latency DRAM Short Bitline

Small Area
- Small Area
- Large Area

High Latency
- Low Latency

Small area using long bitline

Low Latency
Tiered-Latency DRAM

- Divide a bitline into two segments with an isolation transistor

![Diagram showing Tiered-Latency DRAM with Near Segment, Far Segment, Isolation Transistor, and Sense Amplifier]
Near Segment Access

- Turn **off** the isolation transistor

- Reduced bitline length
- Reduced bitline capacitance
  - Low latency & low power

- Isolation Transistor (**off**)
- Near Segment
- Sense Amplifier
Far Segment Access

- Turn *on* the isolation transistor

Long bitline length
Large bitline capacitance
Additional resistance of isolation transistor

→ High latency & high power

---

Isolation Transistor *(on)*
Near Segment
Sense Amplifier
Commodity DRAM vs. TL-DRAM

- **DRAM Latency (tRC)**
  - Commodity DRAM: (52.5ns)
  - Near TL-DRAM: −56%
  - Far TL-DRAM: +23%

- **DRAM Power**
  - Commodity DRAM: +49%
  - Near TL-DRAM: −51%
  - Far TL-DRAM: +49%

- **DRAM Area Overhead**
  ~3%: Mainly due to the isolation transistors
Latency vs. Near Segment Length

Longer near segment length leads to higher near segment latency

Near Segment Length (Cells)

Latency (ns)

1  2  4  8  16  32  64  128  256  512

Near Segment

Ref.
Latency vs. Near Segment Length

Far segment latency is higher than commodity DRAM latency
Trade-Off: Area vs. Latency

Normalized DRAM Area vs. Latency (ns)

- Cheaper: Lower on the y-axis
- Faster: Higher on the x-axis

Near Segment: 32, 64, 128, 256 cells/bitline
Far Segment: 512 cells/bitline

Goal: Lower area and faster latency
Leveraging Tiered-Latency DRAM

• TL-DRAM is a *substrate* that can be leveraged by the hardware and/or software
  – Use near segment as hardware-managed cache to far segment
Using near segment as a cache improves performance and reduces energy consumption.
Summary: TL-DRAM

• Observation
  – Long bitlines are the dominant source of DRAM latency

• Idea
  – Divide a long bitline into two shorter segments
  → Fast and slow segments

• Tiered-latency DRAM: Enables latency heterogeneity
  – Can leverage this in many ways to improve performance and reduce power consumption

• Performance & Power Evaluation
  – When the fast segment is used as a cache to the slow segment
  → Significant performance improvement (>12%) and power reduction (>23%) at low area cost (3%)
Outline

1. TL-DRAM  Reducing DRAM Latency by Modifying Bitline Architecture
2. AL-DRAM  Optimizing DRAM Latency for the Common Case
3. AVA-DRAM  Lowering DRAM Latency by Exploiting Architectural Variation

Prior Work

Future Research Direction
DRAM Stores Data as Charge

Three steps of charge movement

1. Sensing
2. Restore
3. Precharge
Why does DRAM need the extra timing margin?
Two Reasons for Timing Margin

1. Process Variation
   - DRAM cells are not equal
   - Leads to extra timing margin for cells that can store large amount of charge

2. Temperature Dependence
DRAM Cells are Not Equal

Ideal

Same size → Same charge → Same latency

Real

Smallest cell

Different size → Different charge → Different latency

Large variation in cell size → Large variation in charge → Large variation in access latency
Two Reasons for Timing Margin

1. Process Variation
   - DRAM cells are not equal
   - Leads to *extra timing margin* for cells that can store large amount of charge

2. Temperature Dependence
   - DRAM leaks more charge at higher temperature
   - Leads to extra timing margin when operating at low temperature
Cells store small charge at high temperature and large charge at low temperature
→ Large variation in access latency
DRAM Timing Parameters

• DRAM timing parameters are dictated by *the worst case*
  – The smallest cell with the smallest charge in all DRAM products
  – Operating at *the highest temperature*

• Large timing margin for the common case
  ➔ Can lower latency for the common case
DRAM Testing Infrastructure

- Temperature Controller
- FPGAs
- Heater
- FPGAs
Obs 1. Faster Sensing

Typical DIMM at Low Temperature

More charge
Strong charge flow
Faster sensing

Timing ($t_{RCD}$)
17% ↓
No Errors

115 DIMM characterization

$\rightarrow$ More charge $\rightarrow$ Faster sensing
Obs 2. Reducing Restore Time

Typical DIMM at Low Temperature

- Larger cell &
  - Less leakage $\rightarrow$
- Extra charge
- No need to fully restore charge

115 DIMM characterization

- Read ($t_{RAS}$): $37\% \downarrow$
- Write ($t_{WR}$): $54\% \downarrow$
- No Errors

Typical DIMM at lower temperature

$\rightarrow$ More charge $\rightarrow$ Restore time reduction
Obs 3. Reducing Precharge Time

Precharge? – Setting bitline to half-full charge
Obs 3. Reducing Precharge Time

Access empty cell

Not fully precharged

Empty (0V)

Half

bitline

Access full cell

More charge

→ strong sensing

Full (Vdd)

Timing (tRP)

35% ↓

No Errors

Typical DIMM at Lower Temperature

→ More charge

→ Precharge time reduction
Adaptive-Latency DRAM

• Key idea
  – Optimize DRAM timing parameters online

• Two components
  – DRAM manufacturer profiles multiple sets of reliable DRAM timing parameters at different temperatures for each DIMM
  – System monitors DRAM temperature & uses appropriate DRAM timing parameters
Real System Evaluation

AL-DRAM provides high performance improvement, greater for multi-core workloads.
Summary: AL-DRAM

• Observation
  – DRAM timing parameters are dictated by the worst-case cell (smallest cell at highest temperature)

• Our Approach: Adaptive-Latency DRAM (AL-DRAM)
  – Optimizes DRAM timing parameters for the common case (typical DIMM operating at low temperatures)

• Analysis: Characterization of 115 DIMMs
  – Great potential to lower DRAM timing parameters (17 – 54%) without any errors

• Real System Performance Evaluation
  – Significant performance improvement (14% for memory-intensive workloads) without errors (33 days)
Outline

1. TL-DRAM
   Reducing DRAM Latency by Modifying Bitline Architecture

2. AL-DRAM
   Optimizing DRAM Latency for the Common Case

3. AVA-DRAM
   Lowering DRAM Latency by Exploiting Architectural Variation

Prior Work

Future Research Direction
Architectural Variation

Variability in cell access times is caused by the *physical organization* of DRAM.
Our Approach

- **Experimental study of architectural variation**
  - **Goal:** Identify & characterize inherently slower regions
  - **Methodology:** Profile 96 real DRAM modules by using FPGA-based DRAM test infrastructure

- **Exploiting architectural variation**
  - **AVA Online Profiling:** Dynamic & low cost latency optimization mechanism
  - **AVA Data Shuffling:** Improving reliability by avoiding ECC-uncorrectable errors
Challenge: External ≠ Internal

External address → IO interface → Address mapping → Internal address

External address ≠ Internal address
Expected Characteristics

• Variation
  – Some regions are slower than others
  – Some regions are more vulnerable than others when accessed with reduced latency

• Repeatability
  – Latency (error) characteristics repeat periodically, if the same component (e.g., mat) is duplicated

• Similarity
  – Across different organizations (e.g., chip/DIMM) if they share same design
1. Variation & Repeatability in Rows

Latency characteristics vary across 512 rows. Latency characteristics repeat every 512 rows.
1.1. Variation in Rows

**tRP**

<table>
<thead>
<tr>
<th>Error Type</th>
<th>tRP (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Errors</td>
<td>10.0 ns</td>
</tr>
<tr>
<td>Periodic Errors</td>
<td>7.5 ns</td>
</tr>
<tr>
<td>Mostly Errors</td>
<td>5.0 ns</td>
</tr>
</tbody>
</table>

![Graph showing erro...](image-url)
1.2. Repeatability in Rows

Error (latency) characteristics periodically repeat every 512 rows
2. Variation in Columns

Different columns \(\rightarrow\) data from **different locations**
\(\rightarrow\) **different characteristics**
2. Variation in Columns

Error (latency) characteristics in columns have specific patterns (e.g., 16 or 32 row groups)
3. Variation in Data Bits

Data in a request → transferred as multiple data bursts
3. Variation in Data Bits

Processor → Read request → DIMM

64-bit data bus in memory channel
8-bit data bus per chip

64-bit data from different locations in the same row in the same chip

Data bits in a request → different characteristics
3. Variation in Data Bits

Specific bits in a request $\rightarrow$ induce more errors
Our Approach

• Experimental study of architectural variation
  – **Goal:** Identify & characterize inherently slower regions
  – **Methodology:** Profile 96 real DRAM modules by using FPGA-based DRAM test infrastructure

• Exploiting architectural variation
  – **AVA Online Profiling:** Dynamic & low cost latency optimization mechanism
  – **AVA Data Shuffling:** Improving reliability by avoiding ECC-uncorrectable errors
1. Challenges of Lowering Latency

- **Static DRAM latency**
  - DRAM vendors need to provide standard timings, increasing testing costs
  - Doesn’t take into account latency changes over time (e.g., aging and wear out)

- **Conventional online profiling**
  - Takes long time (high cost) to profile all DRAM cells

**Goal:** Dynamic & low cost online latency optimization
1. **AVA Online Profiling**

Architectural-Variation-Aware

Profile *only slow regions* to determine latency

→ *Dynamic & low cost* latency optimization
1. **AVA** Online Profiling

Architectural-Variation-Aware

- slow cells
- process variation
- random error

Combining **error-correcting code** & **online profiling**

→ **Reliably** reduce DRAM latency
2. Challenge of Conventional ECC

Processor  
Read request  
64-bit data bus in memory channel  
8-bit data bus per chip  
DIMM  
Error-Correcting Code (ECC)
2. Challenge of Conventional ECC

Conventional ECC leads to more uncorrectable errors due to architectural variation.
2. **AVA** Data Shuffling

**Architectural-Variation-Aware**

Processor -> error

DIMM

shuffle data burst & shuffle rows

→ *Reduce uncorrectable errors*
2. **AVA Data Shuffling**

AVA Shuffling *reduces uncorrectable errors* significantly.
Latency Reduction

AVA-DRAM reduces latency significantly
AVA-DRAM improves performance significantly
Summary: AVA-DRAM

• Observation: Architectural Variation
  – DRAM has inherently slow regions due to its cell array organization, which leads to high DRAM latency

• Our Approach
  – **AVA Profiling**: Profile only inherently slow regions to determine latency → dynamic & low cost latency optimization
  – **AVA Shuffling**: Distribute data from slow regions to different ECC code words → avoid uncorrectable errors

• Analysis: Characterization of 96 DIMMs
  – Great potential to lower DRAM timing parameters

• System Performance Evaluation
  – Significant performance improvement (15% for memory-intensive workloads)
Outline

1. TL-DRAM  Reducing DRAM Latency by Modifying Bitline Architecture
2. AL-DRAM  Optimizing DRAM Latency for the Common Case
3. AVA-DRAM  Lowering DRAM Latency by Exploiting Architectural Variation

Prior Work

Future Research Direction
Prior Work

- Low latency DRAM
  - Having short bitline
  - Heterogeneous bitline
- Cached DRAM
- DRAM with higher parallelism
  - Subarray level parallelism
  - Parallelizing refreshes with accesses
- Memory scheduling
  - Memory scheduling for more parallelism
  - Application-Aware Memory Scheduling
- Caching, Paging, and Prefetching
Prior Work: Low Latency DRAM

• Having shorter bitlines: FCRAM, RL-DRAM
  – Lower latency compared to conventional DRAM
  – Large area for more sense amplifiers (~55% additional area)

• Having shorter bitline regions: [Son et al., ISCA 13]
  – Lower latency for data in shorter bitline regions
  – Less efficiency due to statically-partitioned lower latency regions
  – Not easy to migrate between fast and slow regions
Prior Work: Cached DRAM

• Implementing low-latency SRAM cache in DRAM
  – Lower latency for accessing recently-accessed requests
  – Large area for SRAM cache (~145% for integrating 6% capacity as SRAM cell)
  – Complex control for SRAM cache
Prior Work: More Parallelism

- Subarray-Level Parallelism: [Kim+, ISCA 2012]
  - Enables independent accesses to different subarrays (a row of mats)
  - Does not reduce latency of a single access

- Parallelizing refreshes with accesses: [Chang+, HPCA 14]
  - Mitigates latency penalty of DRAM refresh operations
  - Does not reduce latency of a single access
Outline

1. TL-DRAM Reducing DRAM Latency by Modifying Bitline Architecture
2. AL-DRAM Optimizing DRAM Latency for the Common Case
3. AVA-DRAM Lowering DRAM Latency by Exploiting Architectural Variation

Prior Work

Future Research Direction
Future Research Direction

• Reducing Latency in 3D-stacked DRAM
  – Power delivered from the bottom layer up to the top layer
    → new source of variation in latency
  – Evaluate & exploit power network related variation

• Exploiting Variation in Retention Time
  – Cells have different retention time based on their contents
    (i.e., 0 vs. 1), but use the same refresh interval
  – Evaluate the relationship between the content in a cell and
    retention time & exploit the variation in retention time
Future Research Direction

• System Design for Heterogeneous-Latency DRAM
  – Design a system that *allocates frequently-used or more critical data to fast regions*
  – Design a system that *optimizes DRAM operating conditions for better performance* (e.g., reducing DRAM temperature by spreading accesses out to different regions)
Conclusion

• Observation
  – DRAM cell array is the dominant source of high latency

• DRAM latency can be reduced by enabling and exploiting latency heterogeneity

• Our Three Approaches
  – TL-DRAM: Enabling latency heterogeneity by changing DRAM architecture
  – AL-DRAM: Exploiting latency heterogeneity from process variation and temperature dependency
  – AVA-DRAM: Exploiting latency heterogeneity from architectural variation

• Evaluation & Result
  – Our mechanisms enable significant latency reduction at low cost and thus improve system performance
Contributions

• Identified three major sources of high DRAM latency
  – Long narrow wires
  – Uniform latencies despite different operating conditions
  – Uniform latencies despite architectural variation

• Evaluated the impact of varying DRAM latencies
  – Simulation with detailed DRAM model
  – Profiled real DRAM (96 – 115 DIMMs) with FPGA-based DRAM test infrastructure

• Developed mechanisms to lower DRAM latency, leading to significant performance improvement
Reducing DRAM Latency at Low Cost by Exploiting Heterogeneity

Donghyuk Lee
Carnegie Mellon University