CSC 2224: Parallel Computer Architecture and Programming Main Memory Fundamentals

Prof. Gennady Pekhimenko
University of Toronto
Fall 2019

The content of this lecture is adapted from the slides of Vivek Seshadri, Donghyuk Lee, Yoongu Kim, and lectures of Onur Mutlu @ ETH and CMU

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

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

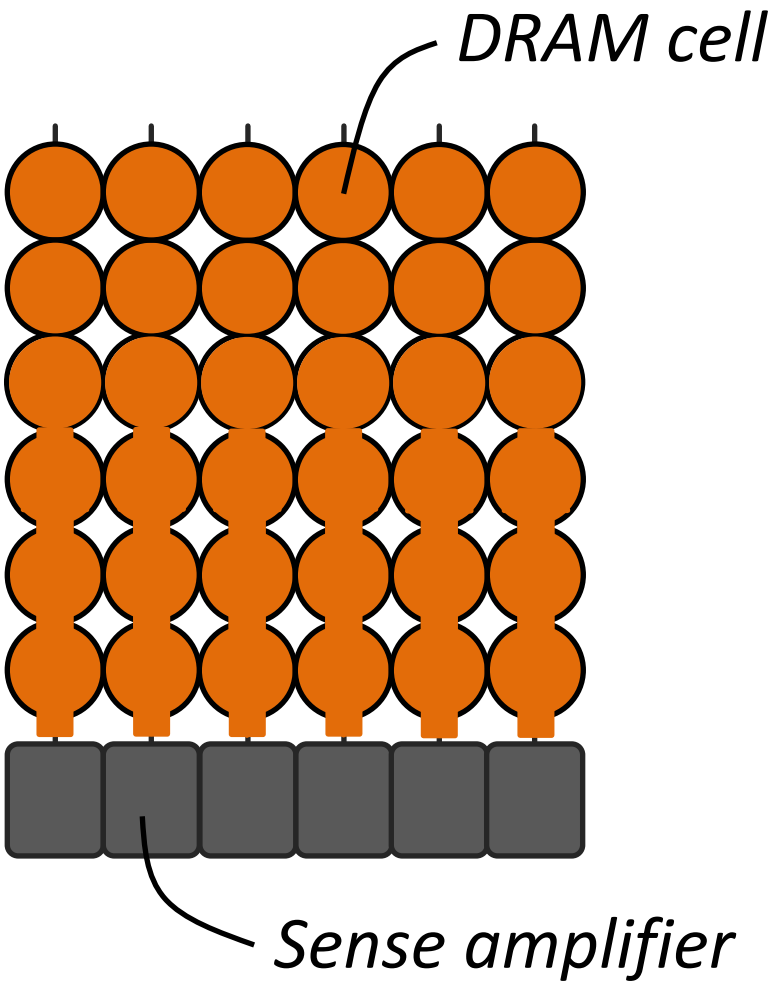
Published in the proceedings of 19th IEEE International Symposium on

High Performance Computer Architecture 2013

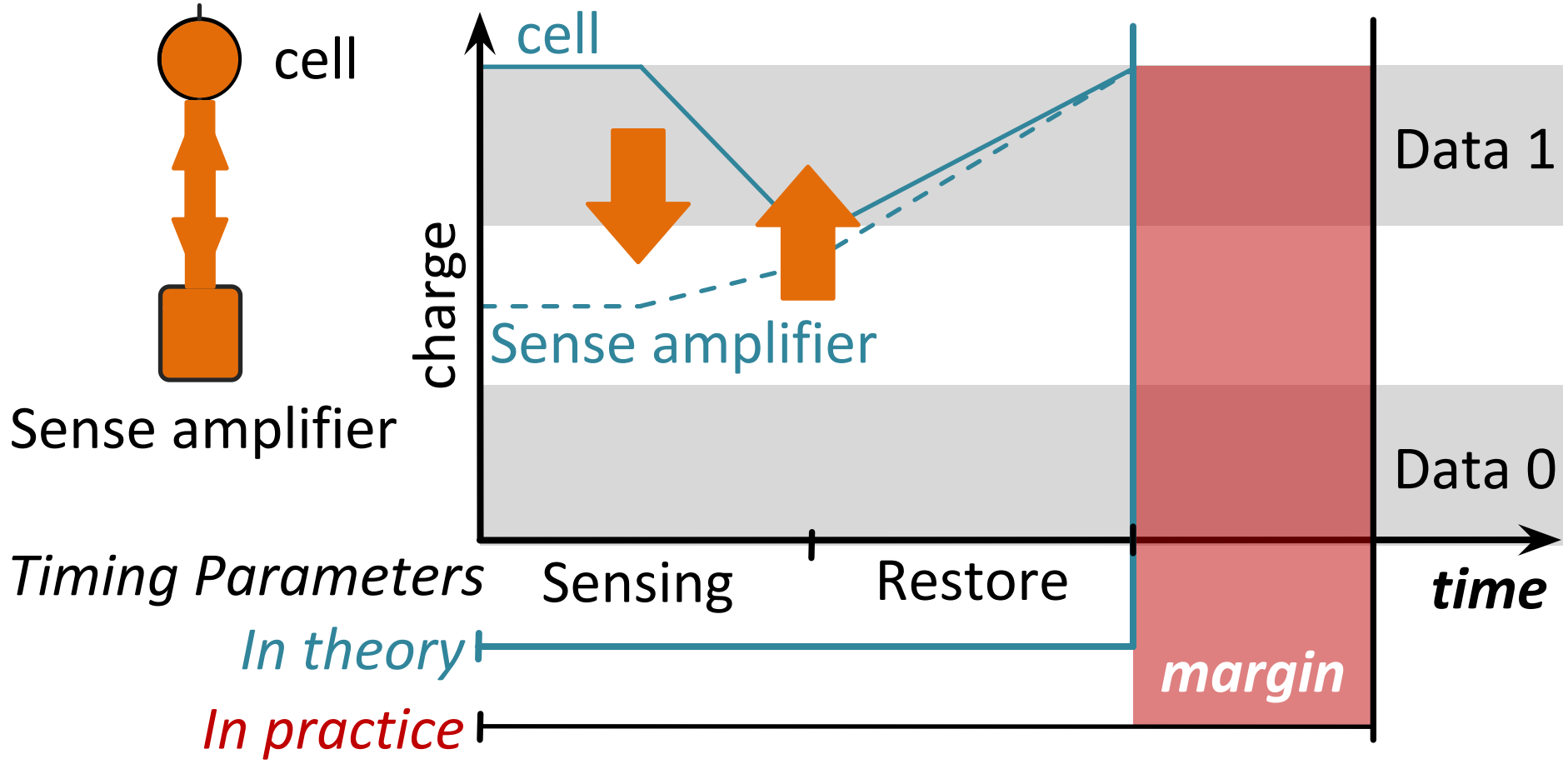
DRAM Stores Data as Charge

Three steps of charge movement

1. Sensing
2. Restore
3. Precharge



DRAM Charge over Time



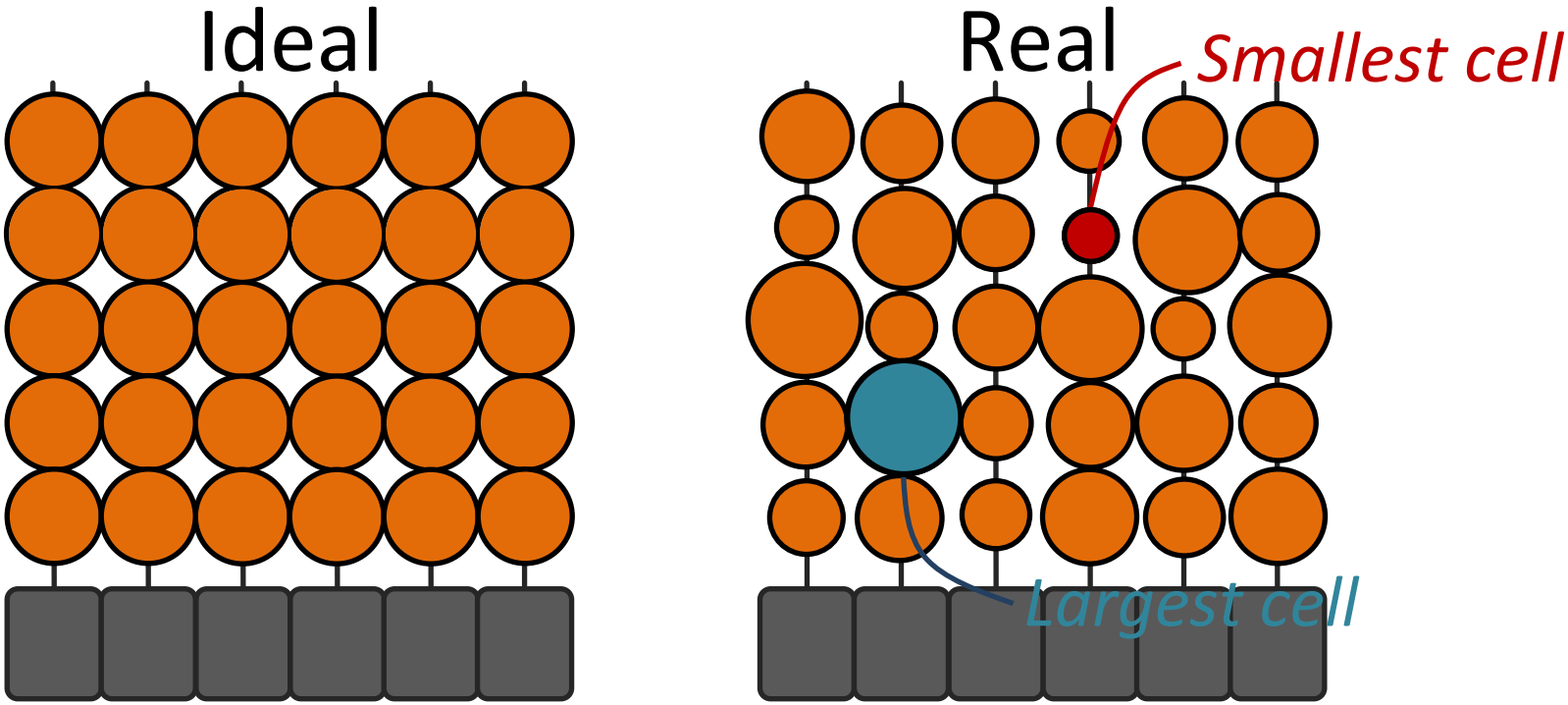
Why does DRAM need the extra timing margin?

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



Same size ?
Same charge ?
Same latency

Large variation in cell size ?
Large variation in charge ?
Large variation in access latency

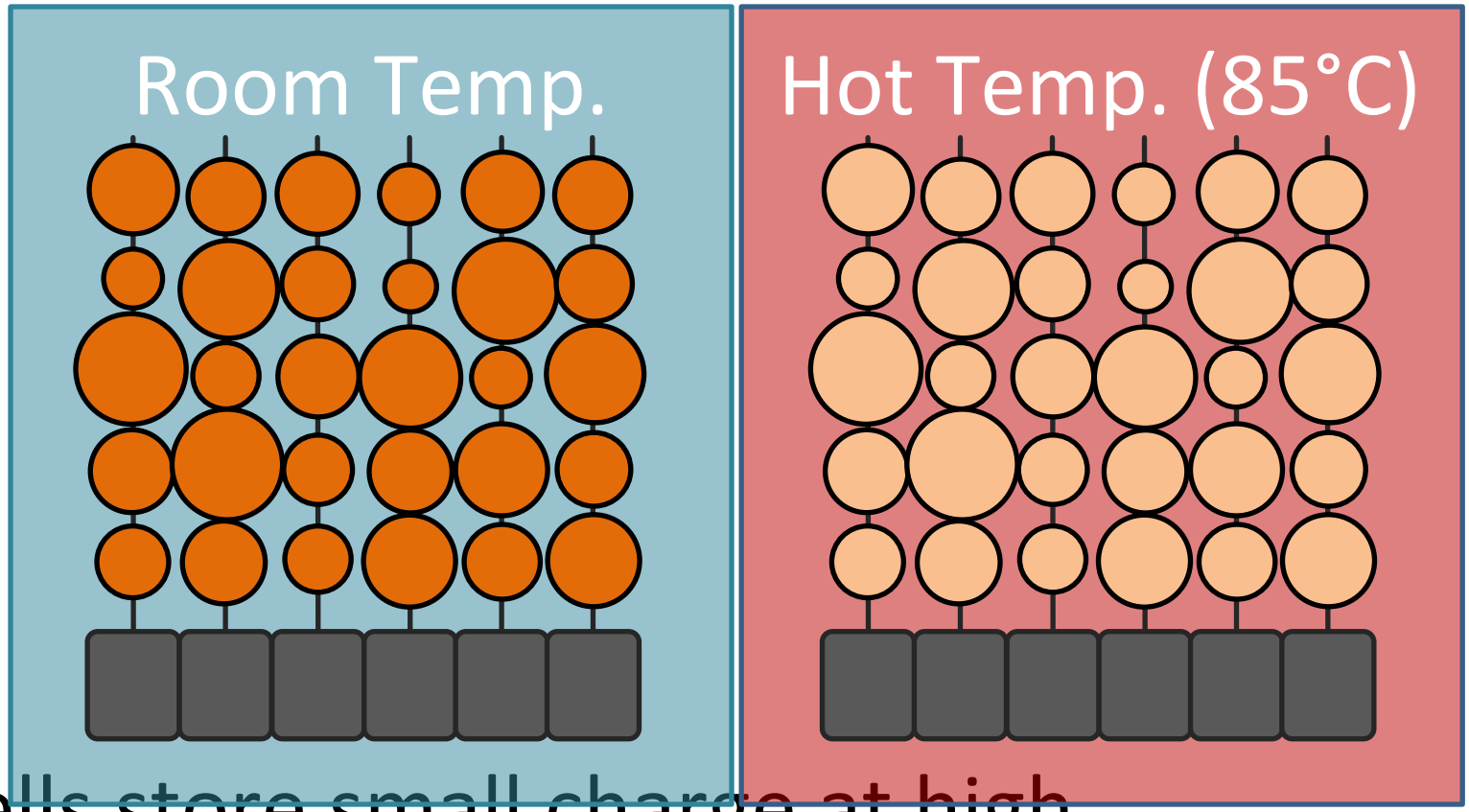
Different size ?
Different charge ?
Different latency

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

Charge Leakage ∝ Temperature



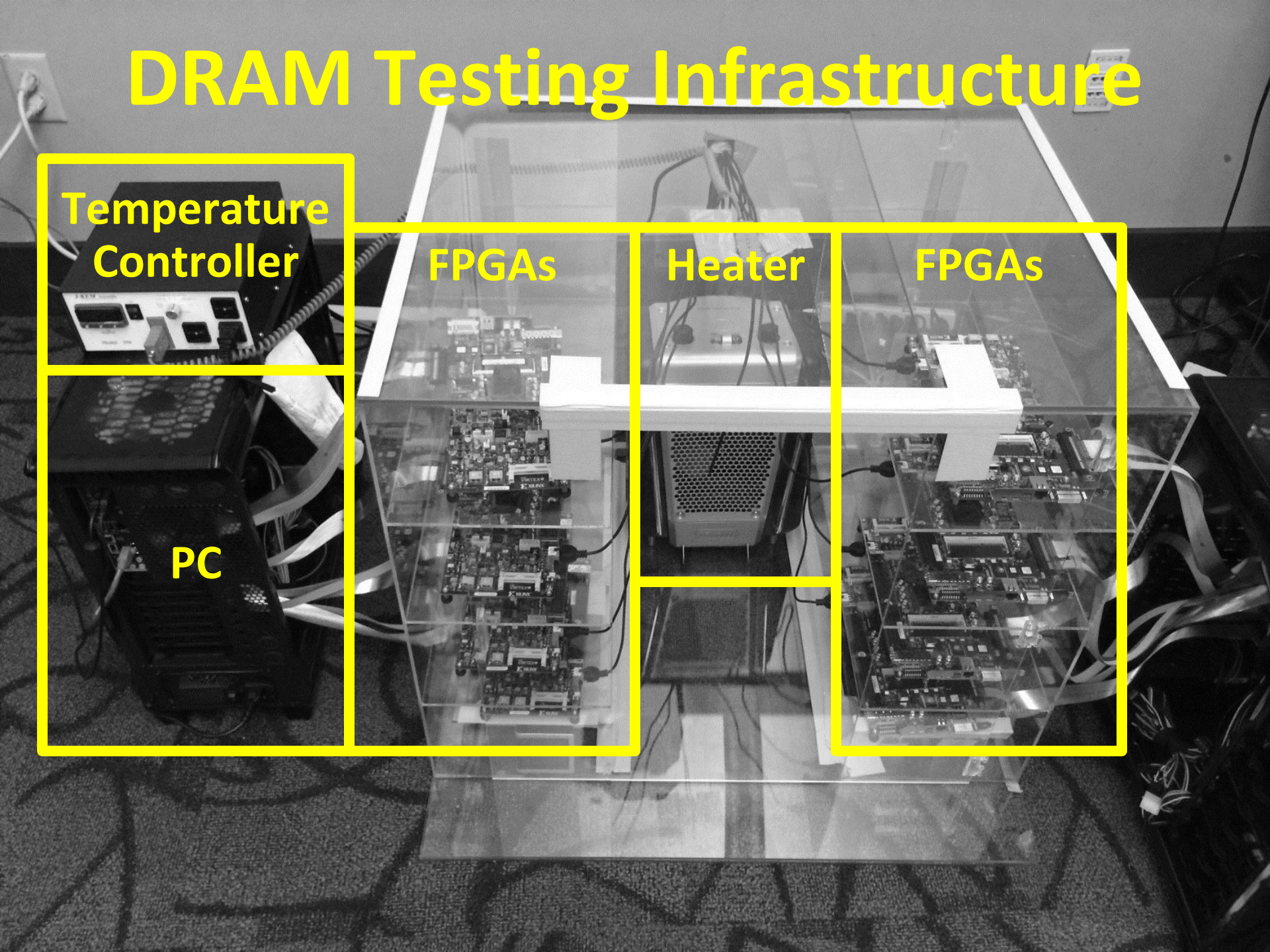
Cells store small charge at high temperature
and large charge at low temperature
 Large variation in access latency

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**

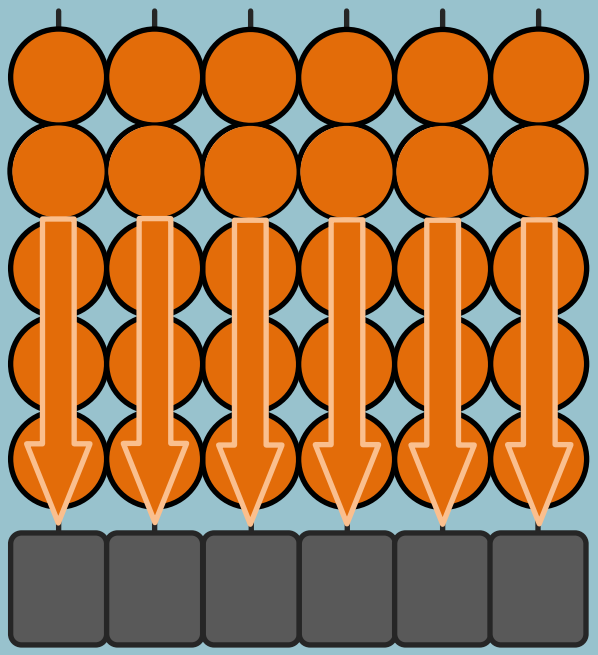
- Large timing margin for the common case

Can lower latency for the common case


DRAM Testing Infrastructure
Temperature Controller
PC
FPGAs
Heater
FPGAs

Obs 1. Faster Sensing

Typical DIMM at Low Temperature



More charge

Strong charge flow

Faster sensing

115 DIMM characterization

Timing (tRCD)

17% ↓

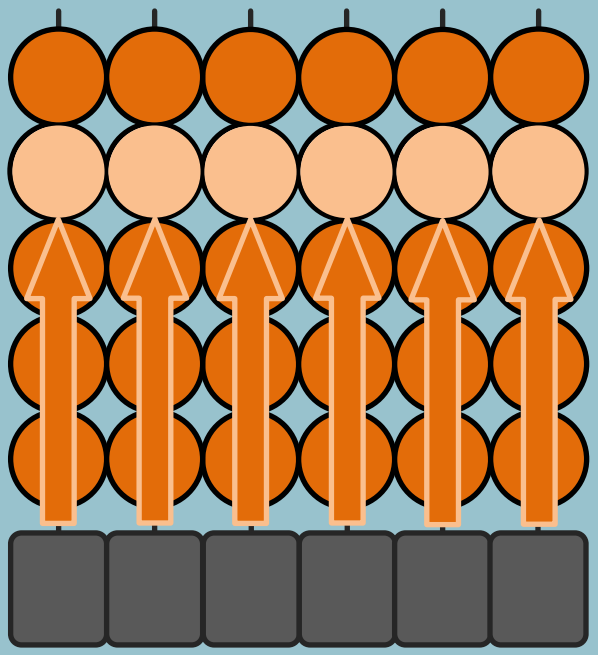
No Errors

Typical DIMM at Low Temperature

⍰ *More charge* ⍰ *Faster sensing*

Obs 2. Reducing Restore Time

Typical DIMM at Low Temperature



Larger cell &
Less leakage □
Extra charge

No need to fully restore charge

115 DIMM characterization

Read (tRAS)

37% ↓

Write (tWR)

54% ↓

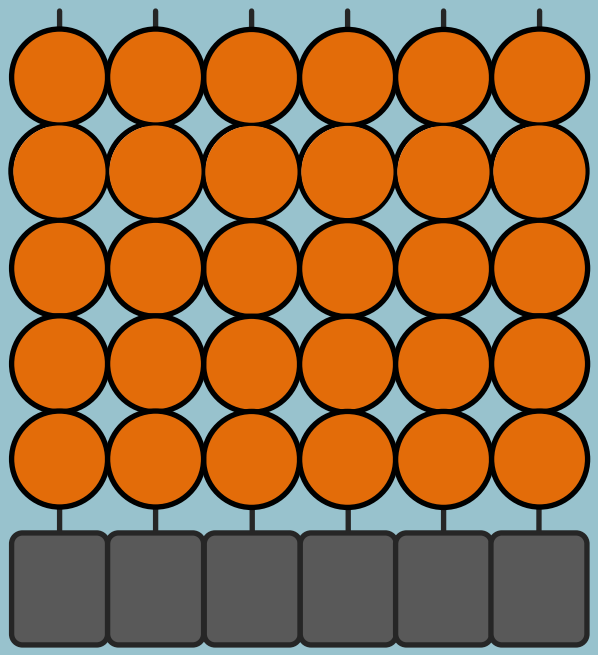
No Errors

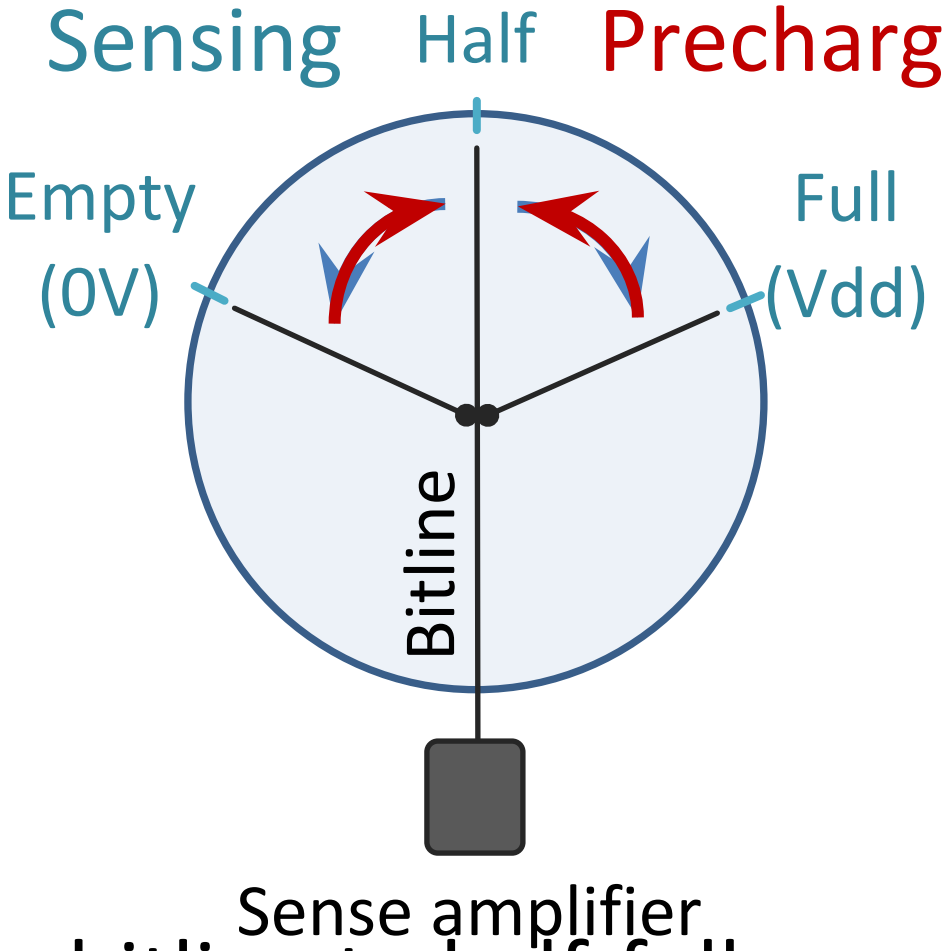
Typical DIMM at lower temperature

□ More charge □ Restore time reduction

Obs 3. Reducing Precharge Time

Typical DIMM at Low Temperature



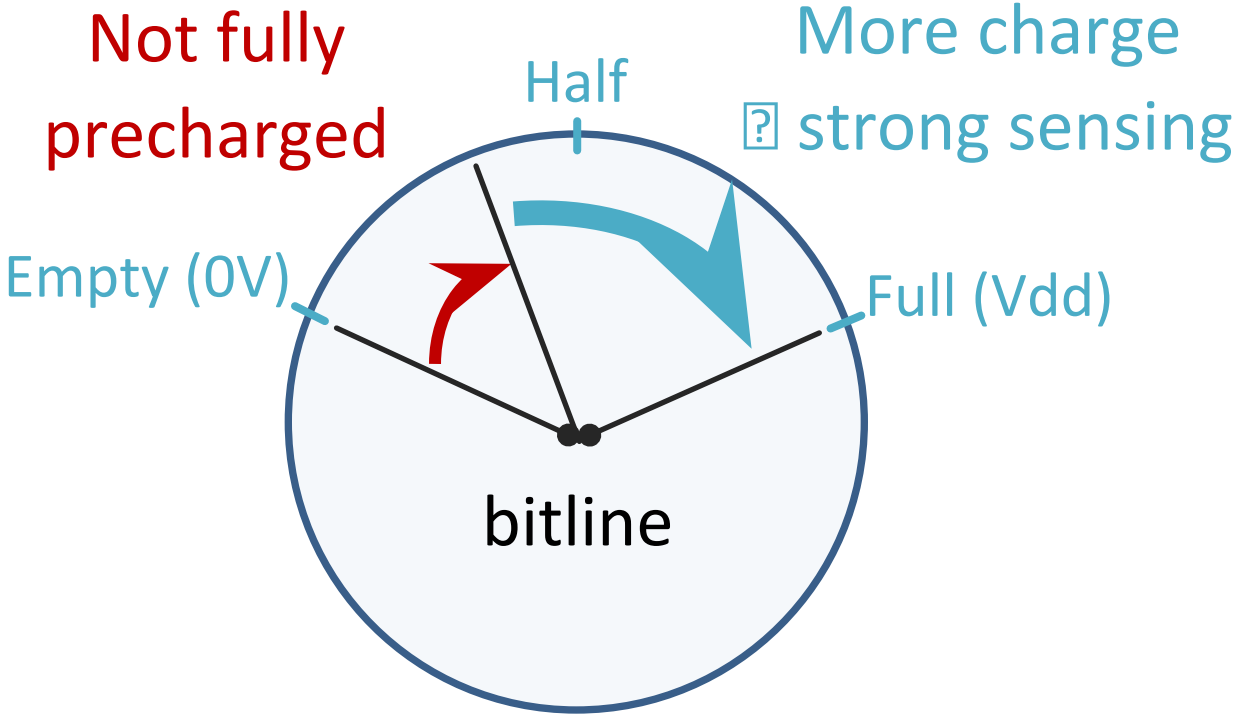


Precharge ? – Setting bitline to half-full charge

Obs 3. Reducing Precharge Time

Access empty cell

Access full cell



115 DIMM characterization

Timing (tRP)

35% ↓

No Errors

Typical DIMM at Lower Temperature

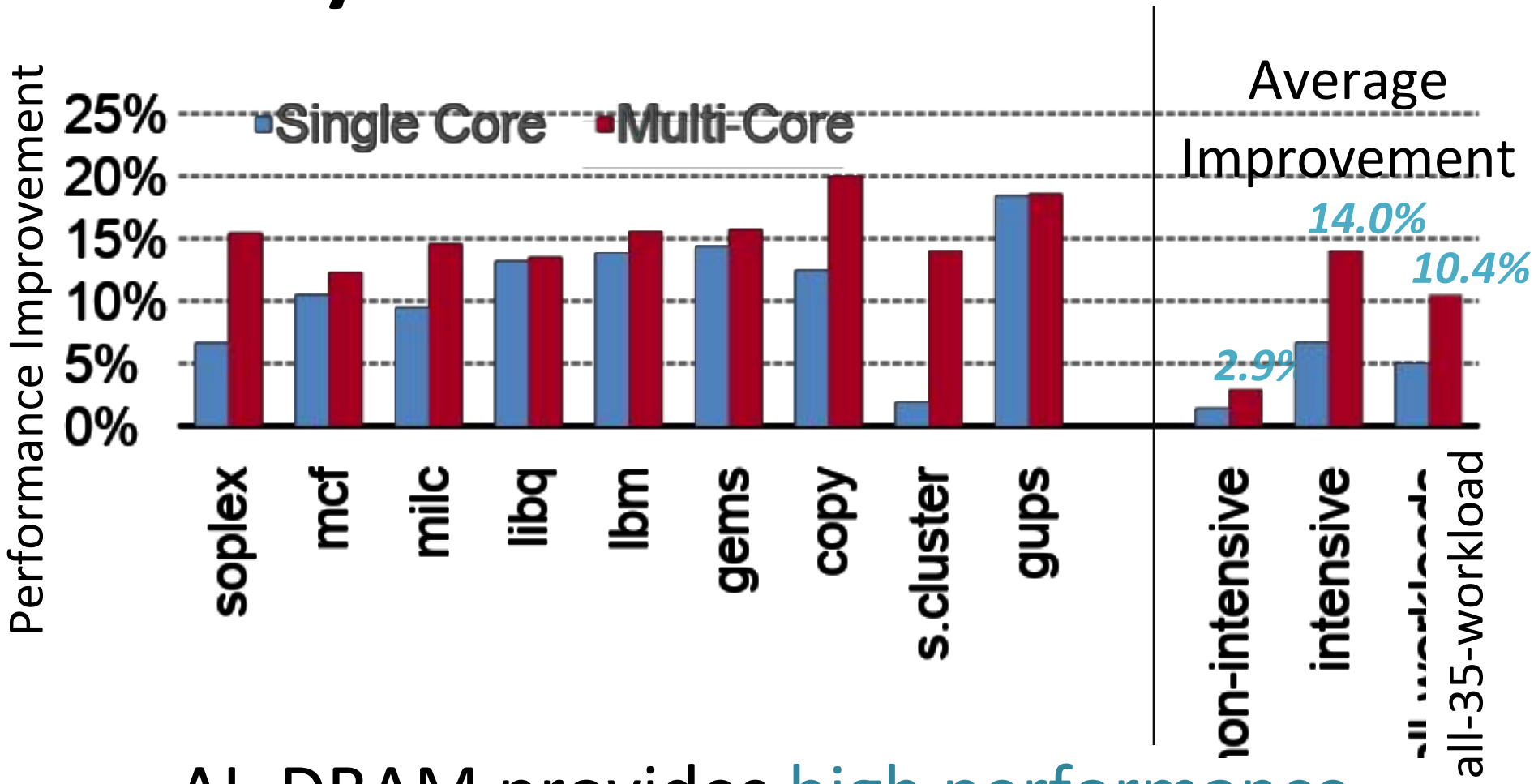
 More charge Precharge time reduction

Adaptive-Latency DRAM

- Key idea
  - Optimize DRAM timing parameters online

- Two components
  - DRAM manufacturer profiles multiple sets of reliable DRAM timing parameters 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)

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

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

Published in the proceedings of 21st
International Symposium on High Performance Computer Architecture 2015

Outline

1. What is DRAM?
2. DRAM Internal Organization
3. Problems and Solutions
   - Latency (Tiered-Latency DRAM, HPCA 2013; Adaptive-Latency DRAM, HPCA 2015)
   - Parallelism (Subarray-level Parallelism, ISCA 2012)

Parallelism: Demand vs. Supply

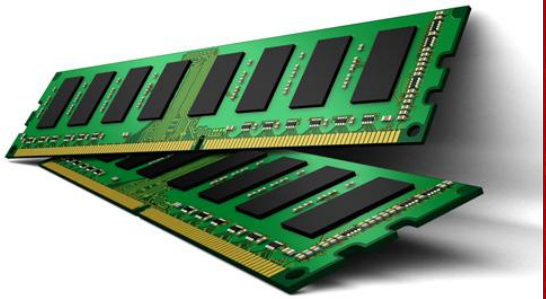
Demand

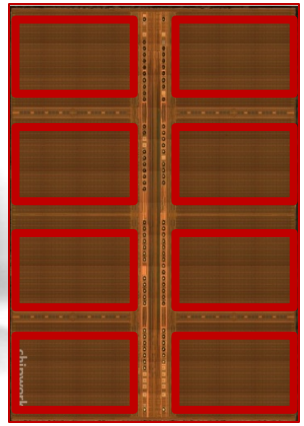
Supply

Out-of-order Execution

Multi-cores

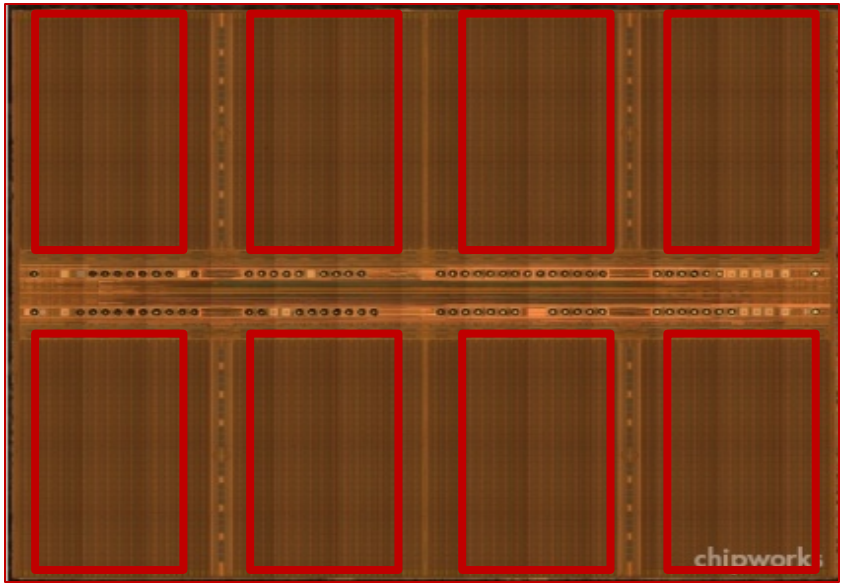
Prefetchers





Multiple Banks

Increasing Number of Banks?

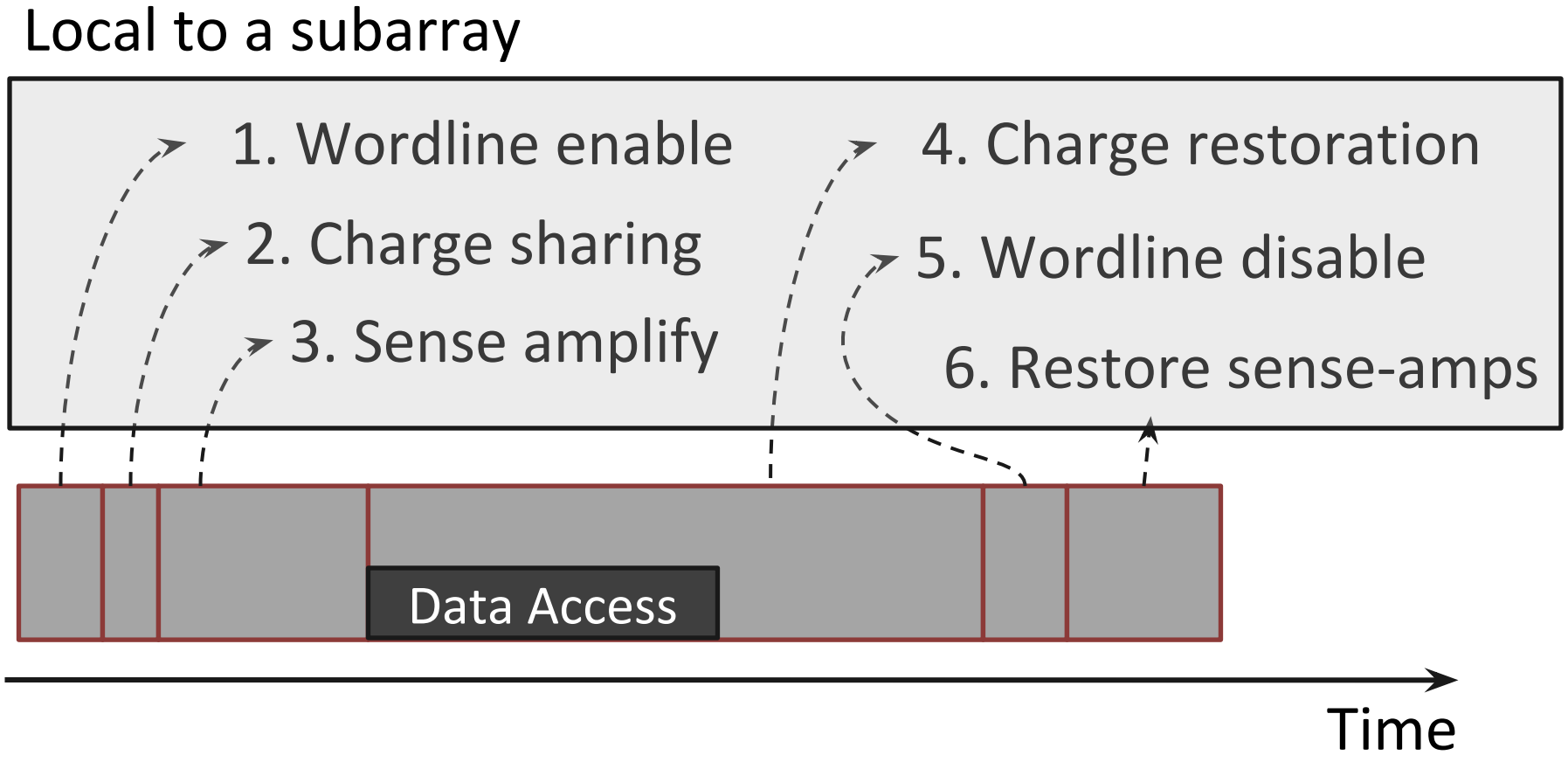


Adding more banks → Replication of shared structures

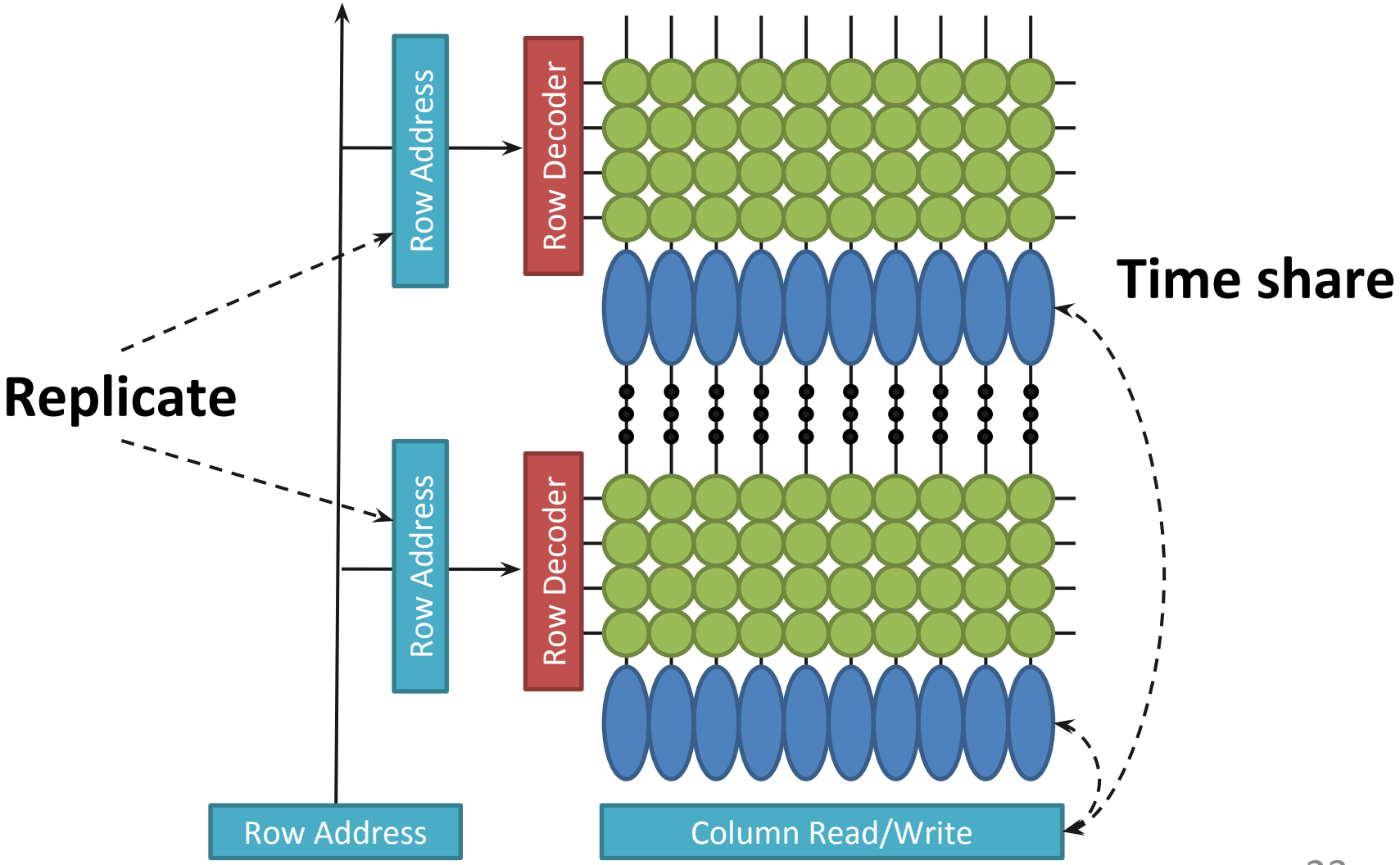
Replication → Cost

How to improve available parallelism within DRAM?

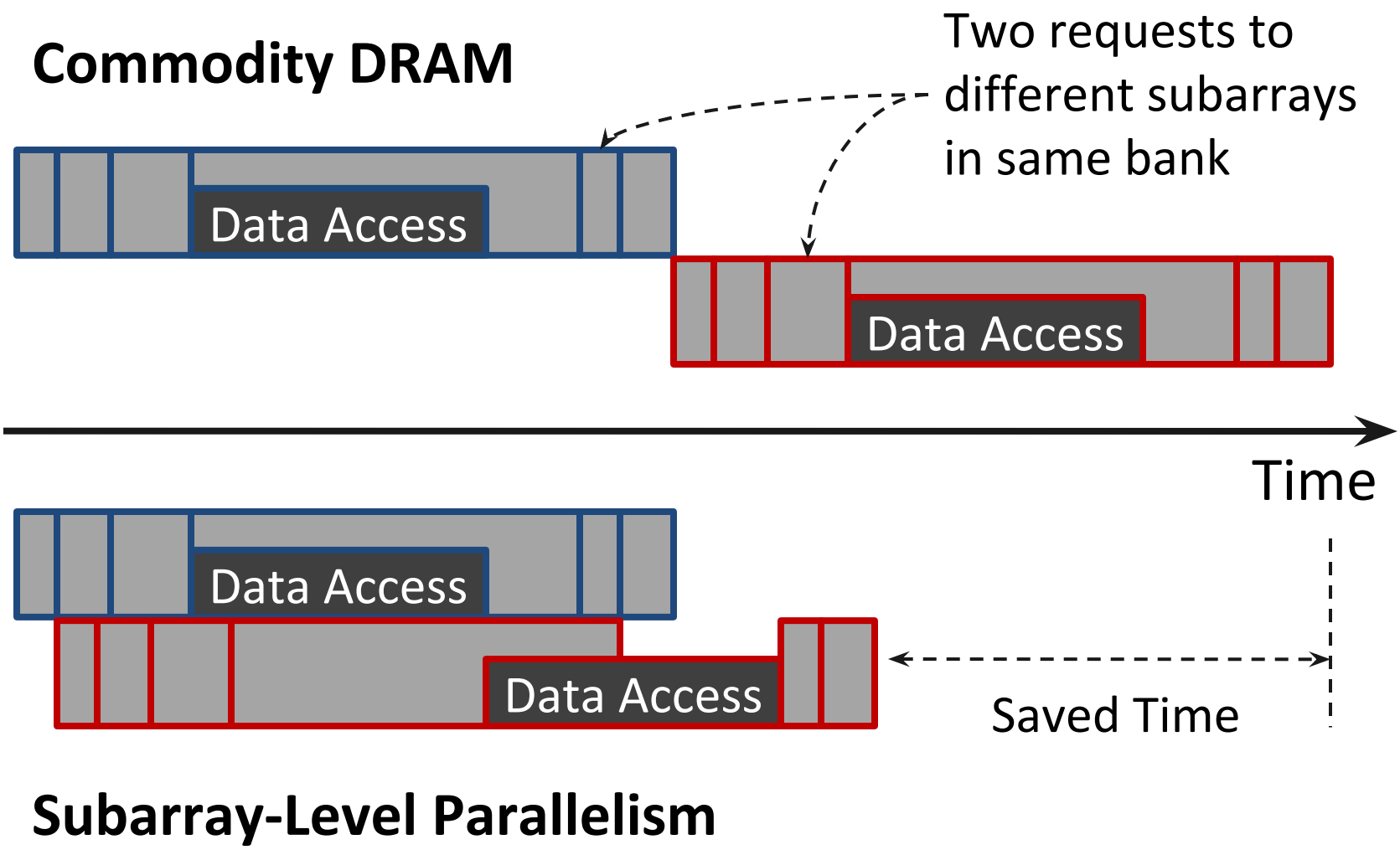
Our Observation



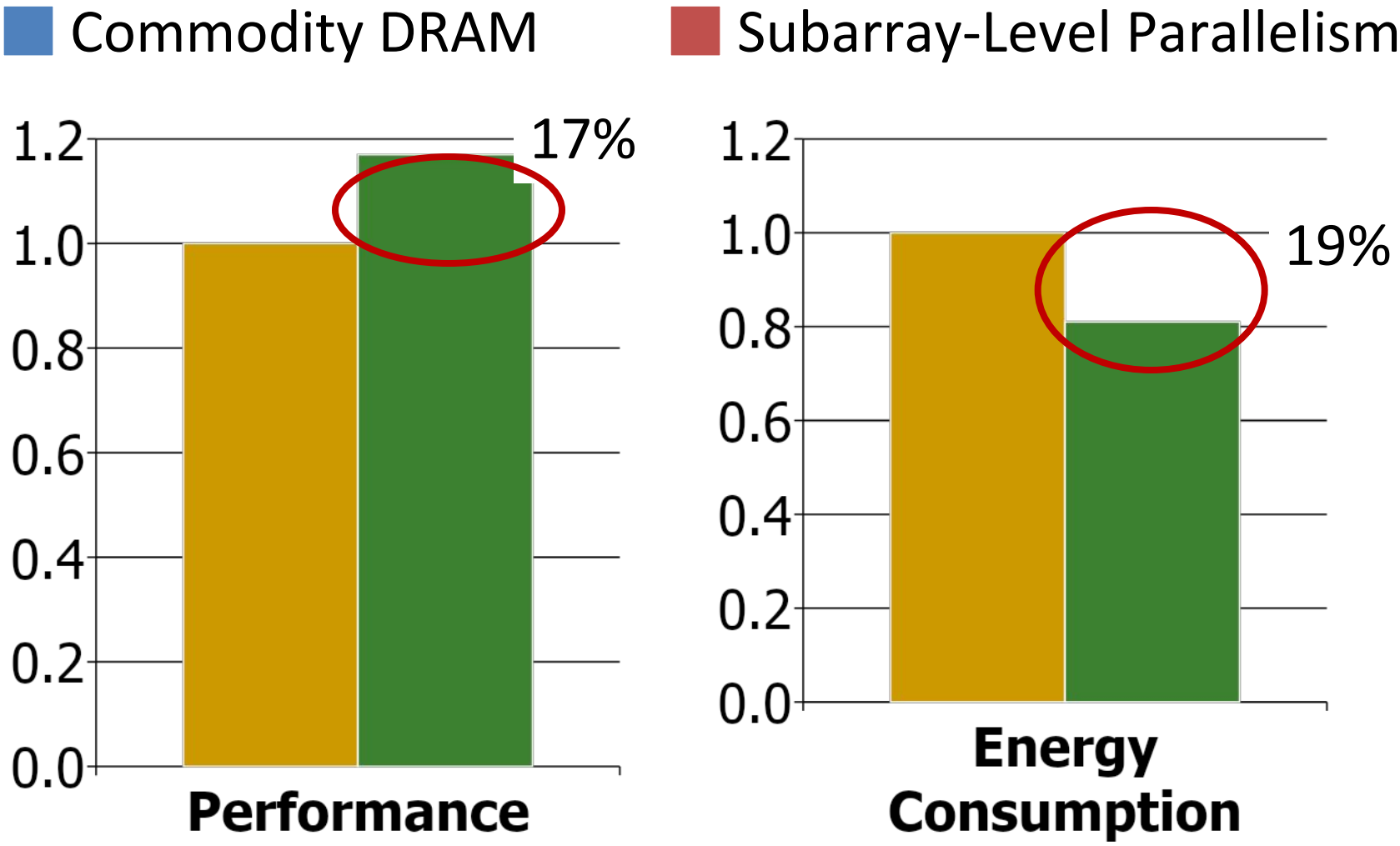
Subarray-Level Parallelism



Subarray-Level Parallelism: Benefits



Results Summary



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

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

Published in the proceedings of 39th

International Symposium on Computer Architecture 2012

CSC 2224: Parallel Computer Architecture and Programming Main Memory Fundamentals

Prof. Gennady Pekhimenko
University of Toronto
Fall 2019

The content of this lecture is adapted from the slides of Vivek Seshadri, Donghyuk Lee, Yoongu Kim, and lectures of Onur Mutlu @ ETH and CMU