CLR-DRAM:
A Low-Cost DRAM Architecture Enabling Dynamic Capacity-Latency Trade-off

Haocong Luo Taha Shahroodi Hasan Hassan Minesh Patel
A. Giray Yaglıkçı Lois Orosa Jisung Park Onur Mutlu

ETH Zürich
SAFARI Research Group
safari.ethz.ch
上海科技大学
ShanghaiTech University

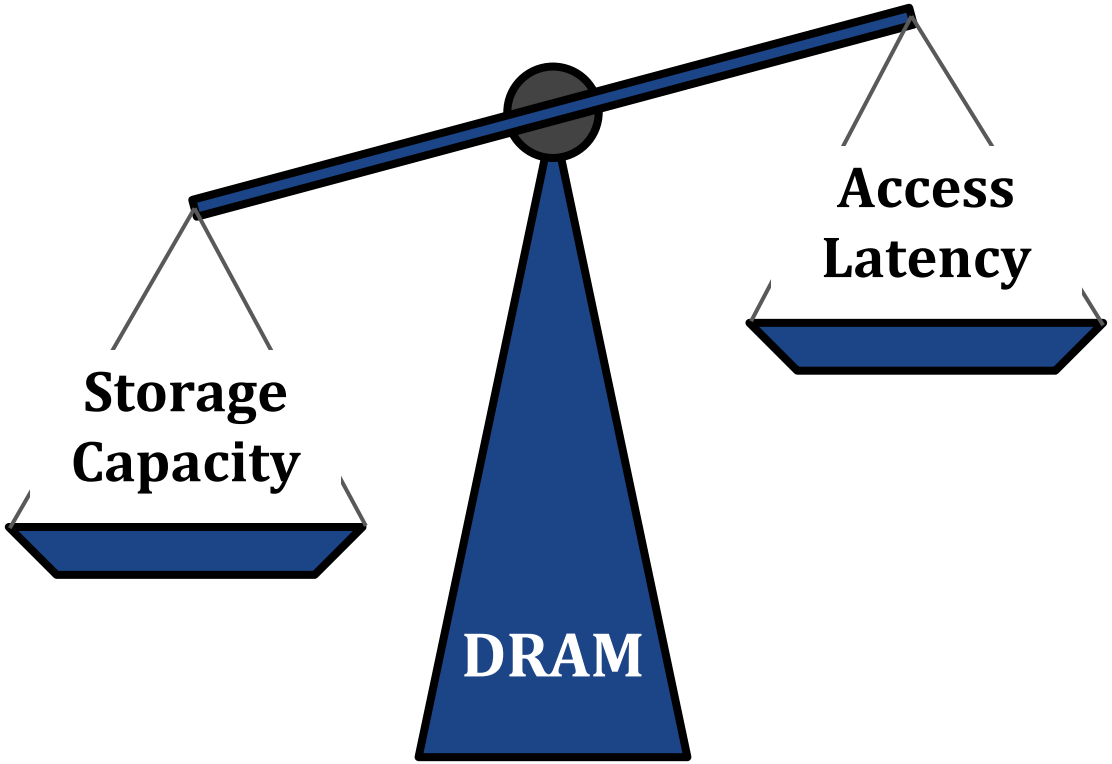
Executive Summary

- **Motivation:** Workloads and systems have varying memory capacity and latency demands.
- **Problem**: Commodity DRAM makes a *static* capacity-latency trade-off at design-time.
  - Existing DRAM cannot adapt to varying capacity and latency demands.
- **Goal**: Design a low-cost DRAM architecture that can be dynamically configured to have high capacity or low latency at a fine granularity (i.e., at the granularity of a row).
- **CLR-DRAM** (**C**apacity-**L**atency-**R**econfigurable **DRAM**):
  A single DRAM row can ***dynamically*** switch between either:
  - **Max-capacity mode** with *high* storage density.
  - **High-performance mode** with *low* access latency and *low* refresh overhead.
- **Key Mechanism**:
  - Couple two adjacent cells and sense amplifiers to operate as a high-performance logical cell.
  - Dynamically turn on or off this coupling at row granularity to switch between two modes.
- **Results:**
  - Reduces key DRAM timing parameters by **35.2%** to **64.2%**.
  - Improves average system performance by **18.6%** and saves DRAM energy by **29.7%**.

Talk Outline

- **Motivation & Goal**
- **DRAM Background**
- **CLR-DRAM (Capacity-Latency-Reconfigurable DRAM)**
- **High-Performance Mode Benefits**
  - Reducing DRAM Access Latency
  - Mitigating DRAM Refresh Overhead
- **Evaluation**
  - SPICE Simulation
  - System-level Evaluation
- **Conclusion**

Fundamental Capacity-Latency Tradeoff in DRAM



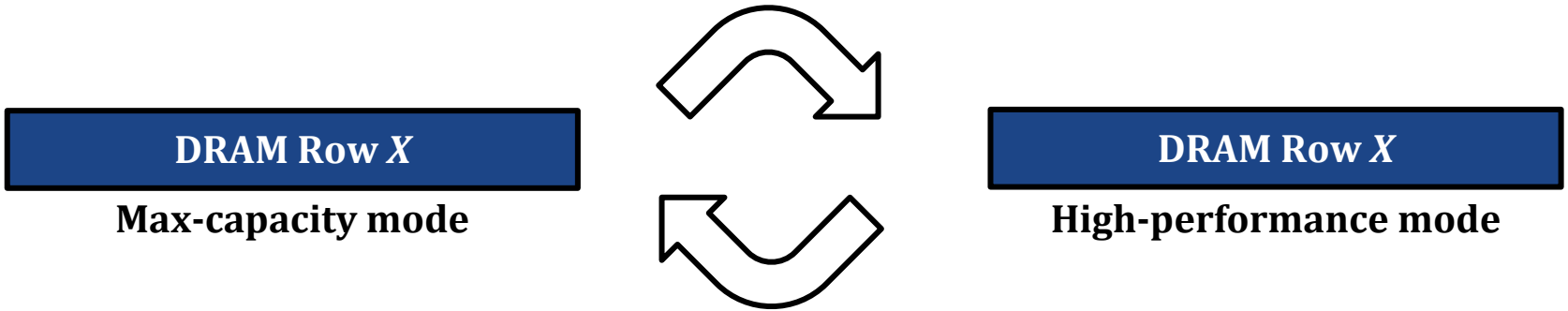
Motivation

- **Motivation:** Existing systems miss opportunities to improve performance by adapting to changes in main memory capacity and latency demands.
  - The memory capacity of a system is usually *overprovisioned*.
  - Many workloads *underutilize* the system's memory capacity.
    - e.g., HPC [Panwar+, MICRO'19], Cloud [Chen+, ICPADS'18], and Enterprise [Di+, CLUSTER'12].

- **Problem**: Commodity DRAM makes a *static* capacity-latency trade-off at design-time.
  - Existing DRAM cannot adapt to varying capacity and latency demands.
  - Some state-of-the-art heterogeneous DRAM architectures [Lee+, HPCA'13, Son+, ISCA'13] employ only a *fixed-size* and *small* low-latency region.
    - Does *not* always provide the best possible operating point within the DRAM capacity-latency trade-off spectrum for all workloads.

Goal

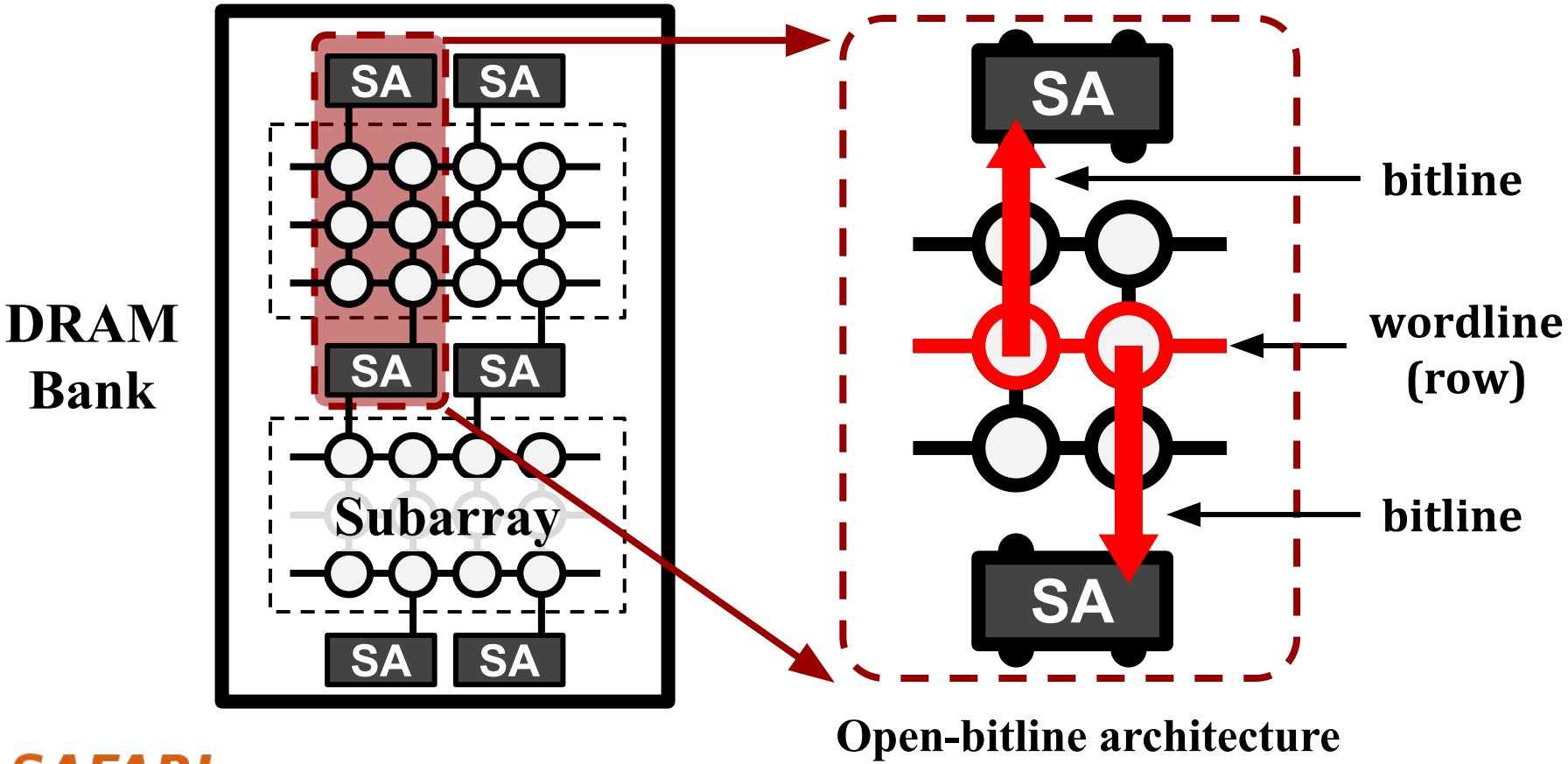
- **Goal**: Design a low-cost DRAM architecture that can be dynamically configured to have high capacity or low latency at a fine granularity (i.e., at the granularity of a row).



Talk Outline

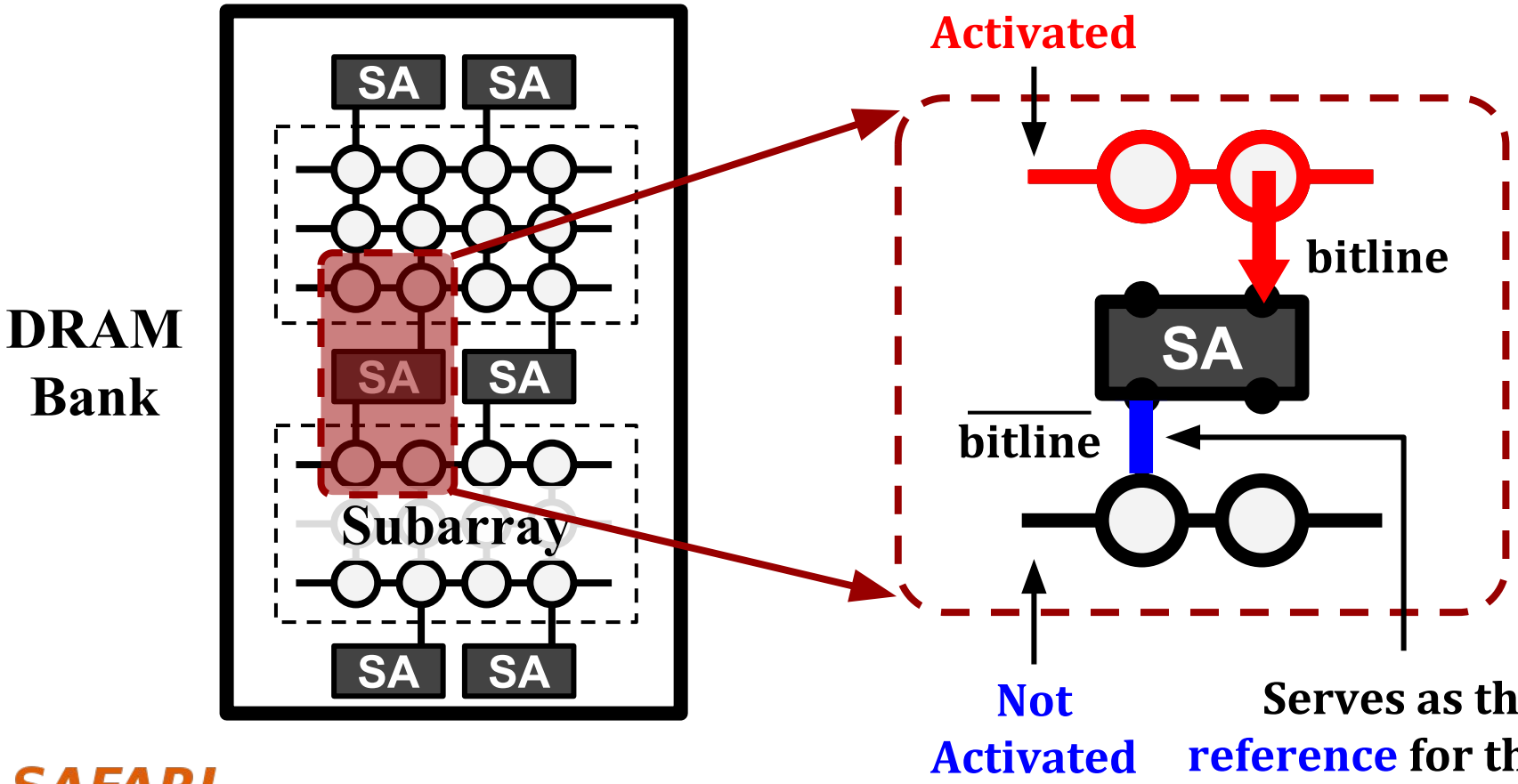
- Motivation & Goal
- **DRAM Background**
- CLR-DRAM (Capacity-Latency-Reconfigurable DRAM)
- High-Performance Mode Benefits
  - Reducing DRAM Access Latency
  - Mitigating DRAM Refresh Overhead
- Evaluation
  - SPICE Simulation
  - System-level Evaluation
- Conclusion

DRAM Background - Array Architecture

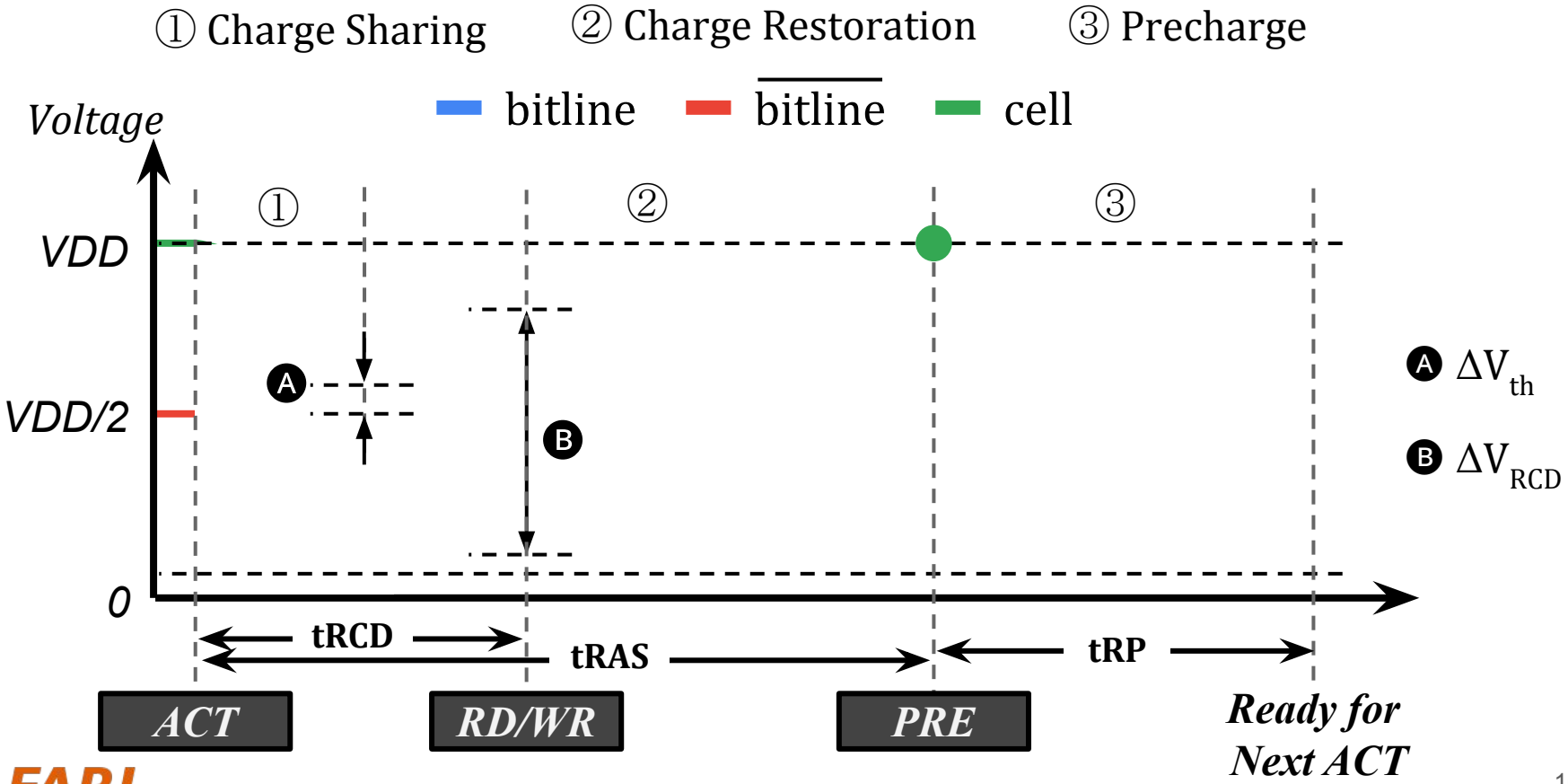


Open-bitline architecture

DRAM Background - Sense Amplifier



DRAM Background - Accessing a Cell



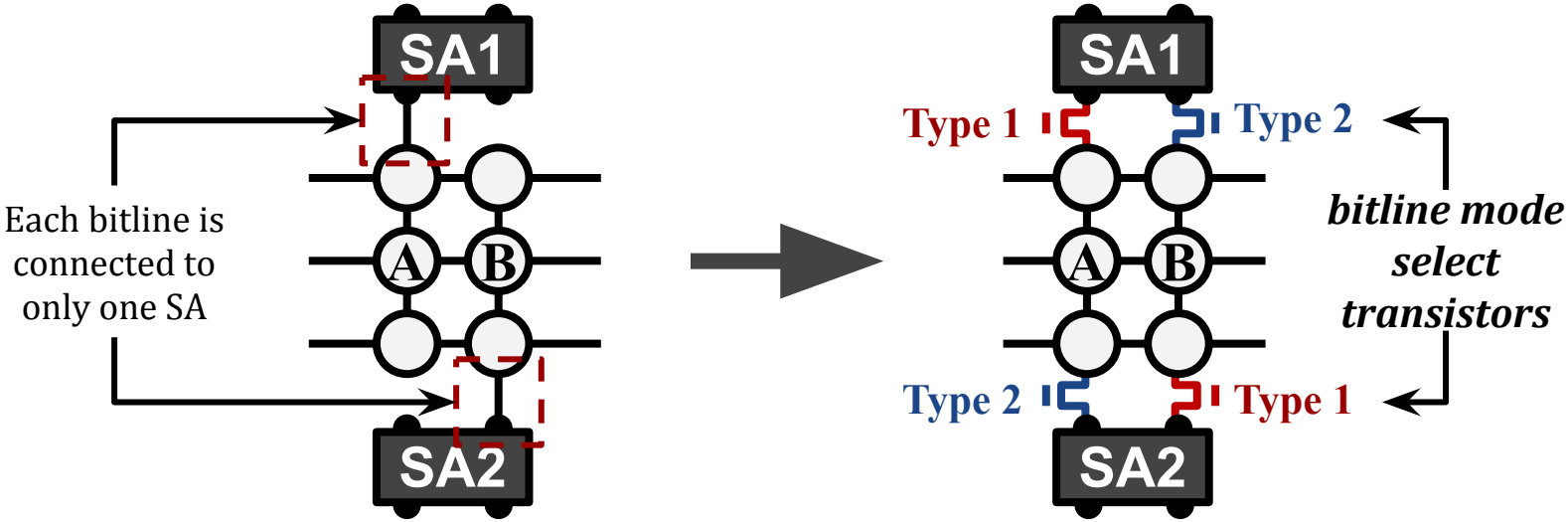
CLR-DRAM Outline

- Motivation & Goal
- DRAM Background
- **CLR-DRAM (Capacity-Latency-Reconfigurable DRAM)**
- High-Performance Mode Benefits
  - Reducing DRAM Access Latency
  - Mitigating DRAM Refresh Overhead
- Evaluation
  - SPICE Simulation
  - System-level Evaluation
- Conclusion

CLR-DRAM (Capacity-Latency-Reconfigurable DRAM)

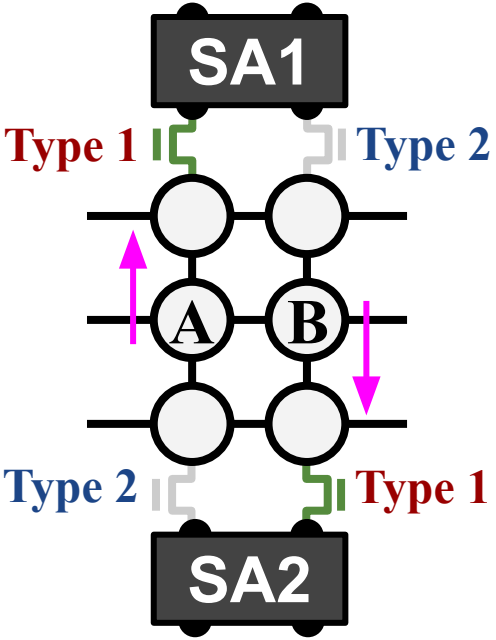
- **CLR-DRAM**: Enables a single DRAM row to *dynamically* switch between **max-capacity mode** or **high-performance mode** with low cost.

- **Key Idea**:
  Dynamically configure the connections between DRAM cells and sense amplifiers in the density-optimized open-bitline architecture.



Max-Capacity Mode

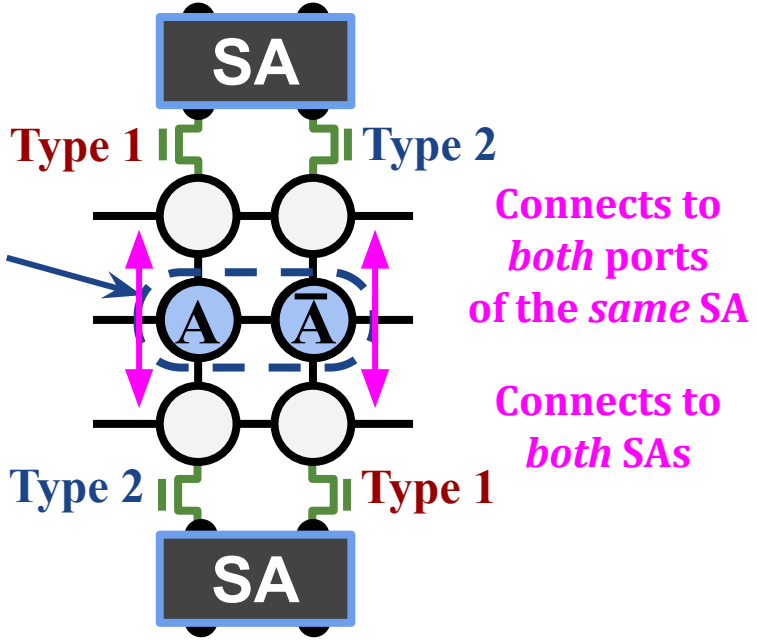
- Max-capacity mode **mimics** the cell-to-SA connections as in the open-bitline architecture.
  - Enable **Type 1** transistors
  - Disable **Type 2** transistors
- Every single cell and its SA operate individually.



Max-capacity mode achieves the same storage capacity as the conventional open-bitline architecture

High-Performance Mode

- High-performance mode **couples every two adjacent DRAM cells** in a row and their **SAs**.
  - Enable **Type 1** transistors
  - Enable **Type 2** transistors
- Two adjacent DRAM cells in a row coupled as a **single logical cell**.
- Two SAs of the two coupled cells coupled as a **single logical SA**.



High-performance mode reduces access latency and refresh overhead via coupled cell/SA operations

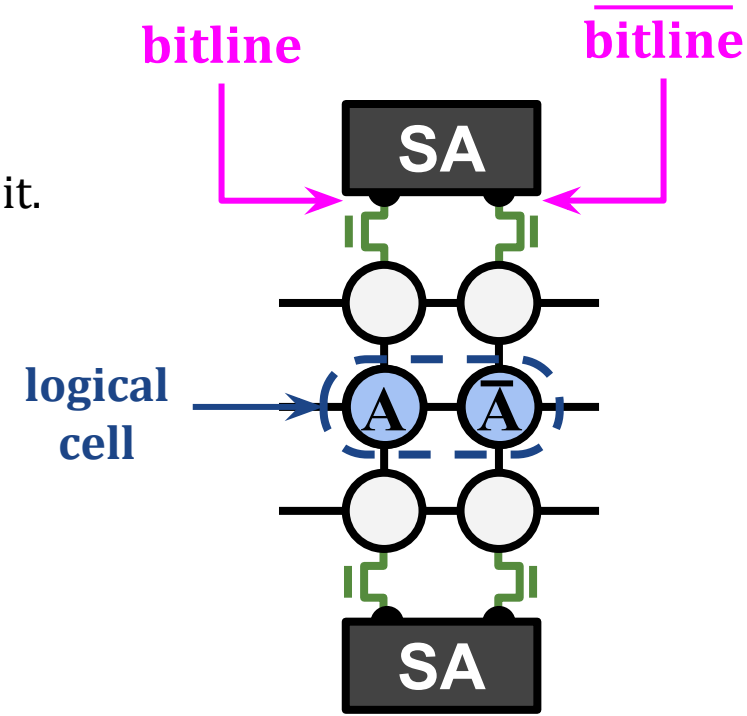
CLR-DRAM Outline

- Motivation & Goal
- DRAM Background
- CLR-DRAM (Capacity-Latency-Reconfigurable DRAM)
- **High-Performance Mode Benefits**
  - Reducing DRAM Access Latency
  - Mitigating DRAM Refresh Overhead
- Evaluation
  - SPICE Simulation
  - System-level Evaluation
- Conclusion

High-Performance Mode Benefits: Coupled Cells

- A logical cell (two coupled cells) always stores *opposite* charge levels representing the same bit.

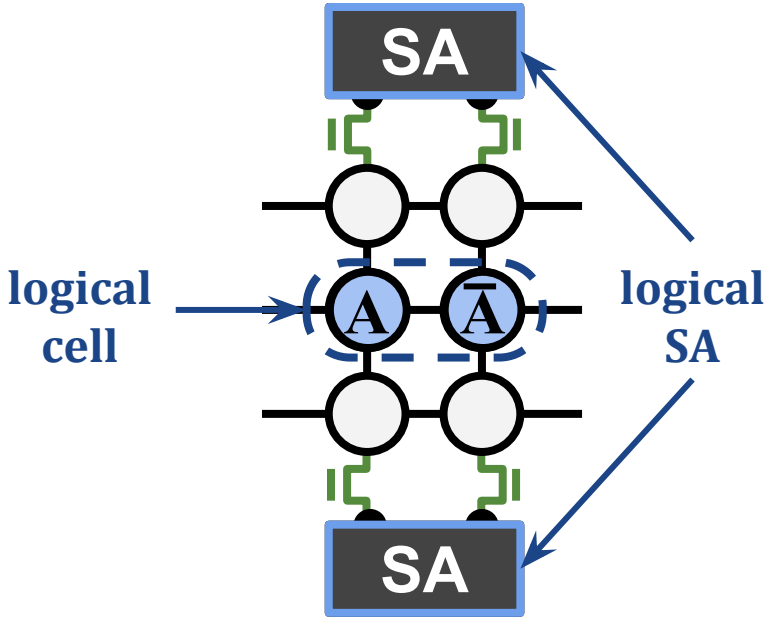
- This enables three benefits:
  - Reducing latency of charge sharing.
  - Early-termination of charge restoration.
  - Retaining data for longer time.



High-Performance Mode Benefits: Coupled SAs

- A logical SA operates faster by having two SAs driving the same logical cell.

- This enables three benefits:
  - Reducing latency of charge restoration.
  - Reducing latency of precharge.
  - Completing refresh in shorter time.



CLR-DRAM Outline

- Motivation & Goal
- DRAM Background
- CLR-DRAM
- **High-Performance Mode Benefits**
  - Reducing DRAM Access Latency
  - Mitigating DRAM Refresh Overhead
- Evaluation
  - SPICE Simulation
  - System-level Evaluation
- Conclusion

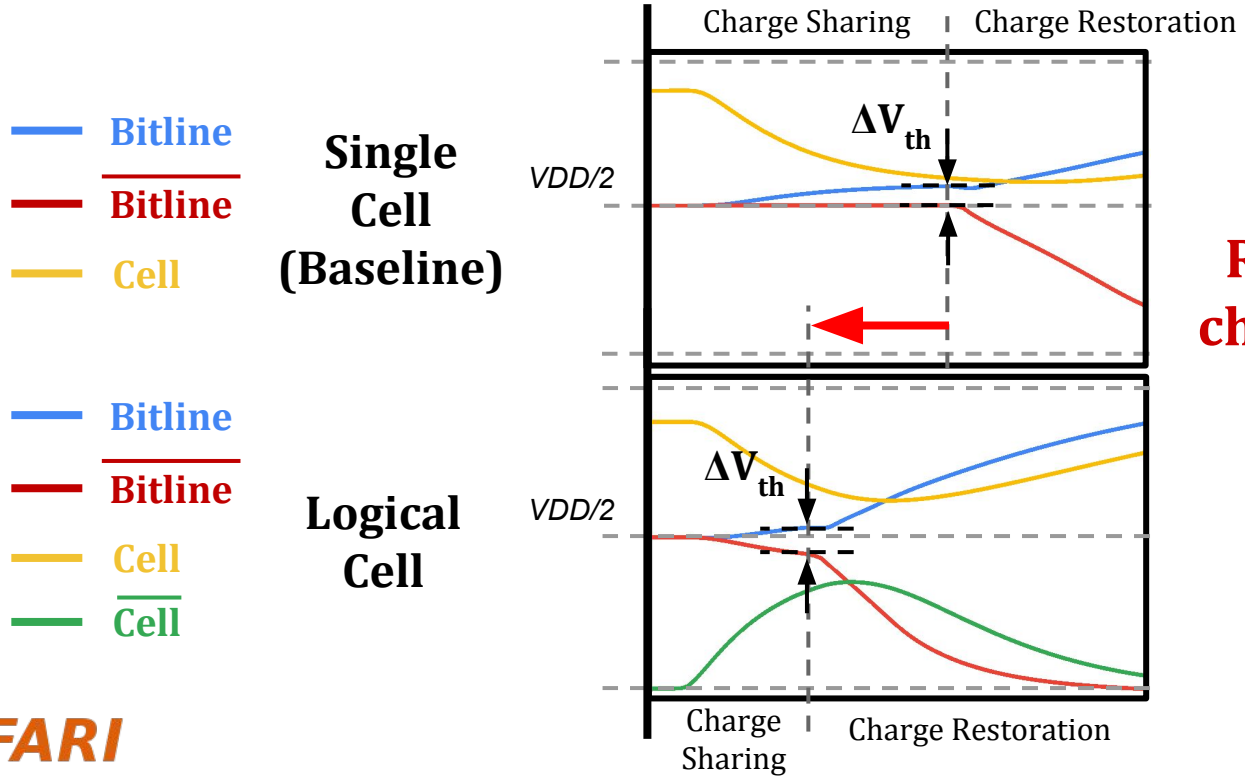
Reducing DRAM Latency: Three Ways

- Reducing latency of charge sharing.
- Early-termination of charge restoration.
- Reducing latency of charge restoration and precharge.

High-performance mode reduces activation (tRCD), restoration (tRAS) and precharge (tRP) latencies

1. Reducing Charge Sharing Latency

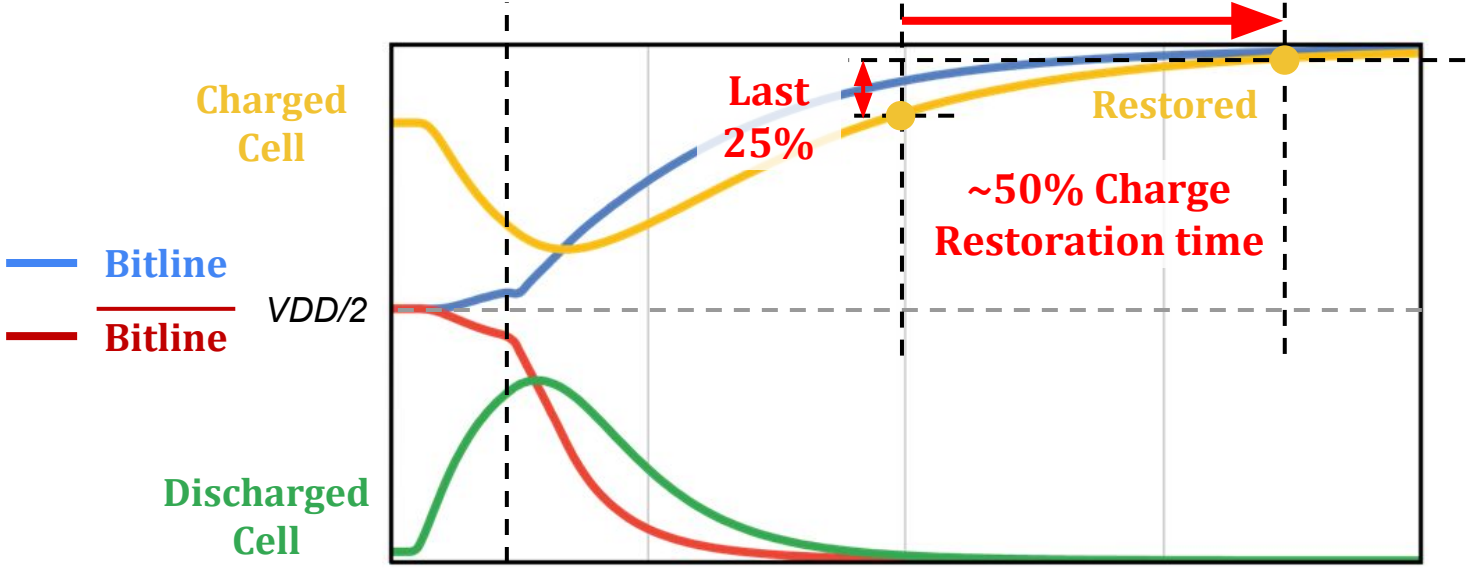
- Coupled cells always store *opposite* charge levels representing the *same* bit.
  - Drive both bitlines of a SA into *opposite* directions during charge sharing.



Reduction in charge sharing latency

2. Early Termination of Charge Restoration

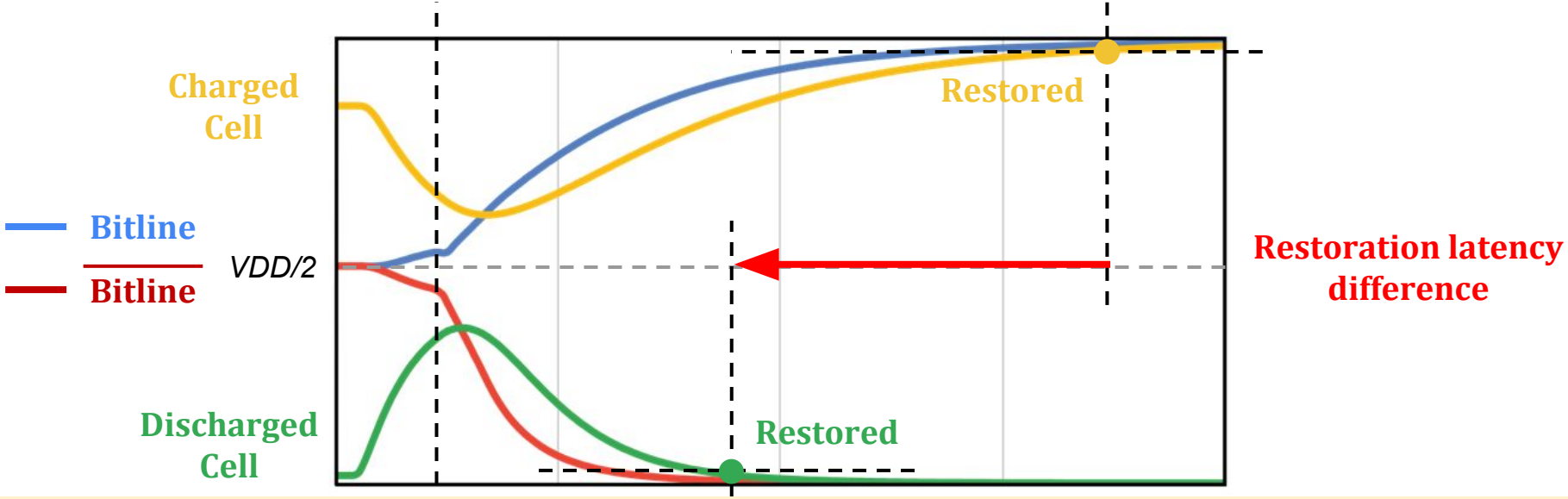
- **Observation 1:** Charge restoration has a long "tail latency".



Terminating charge restoration early does *not* significantly degrade the charge level in the cell

2. Early Termination of Charge Restoration

- **Observation 2:** A discharged cell restores *faster* than a charged one.



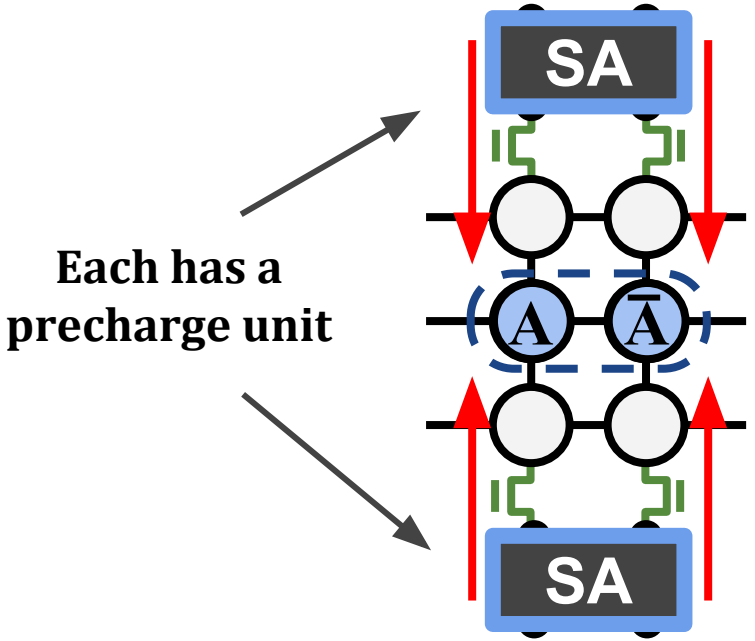
Terminating charge restoration early can still *fully restore* the discharged cell.

3. Reducing Charge Restoration & Precharge Latency

- Logical SA contains two physical SAs.
  - Drive the *same* logical cell from *both* ends of the bitlines.

Faster Charge Restoration

Faster Precharge



Reducing DRAM Latency: Three Ways

- Reducing latency of charge sharing.
- Early-termination of charge restoration.
- Reducing latency of charge restoration and precharge.

High-performance mode reduces activation (tRCD), restoration (tRAS) and precharge (tRP) latencies

CLR-DRAM Outline

- Motivation & Goal
- DRAM Background
- CLR-DRAM
- **High-Performance Mode Benefits**
  - Reducing DRAM Access Latency
  - Mitigating DRAM Refresh Overhead
- Evaluation
  - SPICE Simulation
  - System-level Evaluation
- Conclusion

Mitigating Refresh Overhead

CLR-DRAM reduces refresh overhead of high-performance rows in two different ways:

1. **Reducing Refresh Latency**
   - Refresh is essentially activation + precharge.
   - All latency reductions (activation, restoration, precharge) apply to reduce each refresh operation's latency.

2. **Reducing Refresh Rate**
   - A logical cell has larger capacitance.
   - Tolerates more leakage.
   - Can be refreshed *less* frequently.

High-performance mode reduces refresh latency (tRFC) and refresh rate (increases tREFW)

CLR-DRAM Outline

- Motivation & Goal
- DRAM Background
- CLR-DRAM
- High-Performance Mode Benefits
  - Reducing DRAM Access Latency
  - Mitigating DRAM Refresh Overhead
- **Evaluation**
  - SPICE Simulation
  - System-level Evaluation
- Conclusion

CLR-DRAM Outline

- Motivation & Goal
- DRAM Background
- CLR-DRAM
- High-Performance Mode Benefits
  - Reducing DRAM Access Latency
  - Mitigating DRAM Refresh Overhead
- **Evaluation**
  - SPICE Simulation
  - System-level Evaluation
- Conclusion

SPICE Simulation

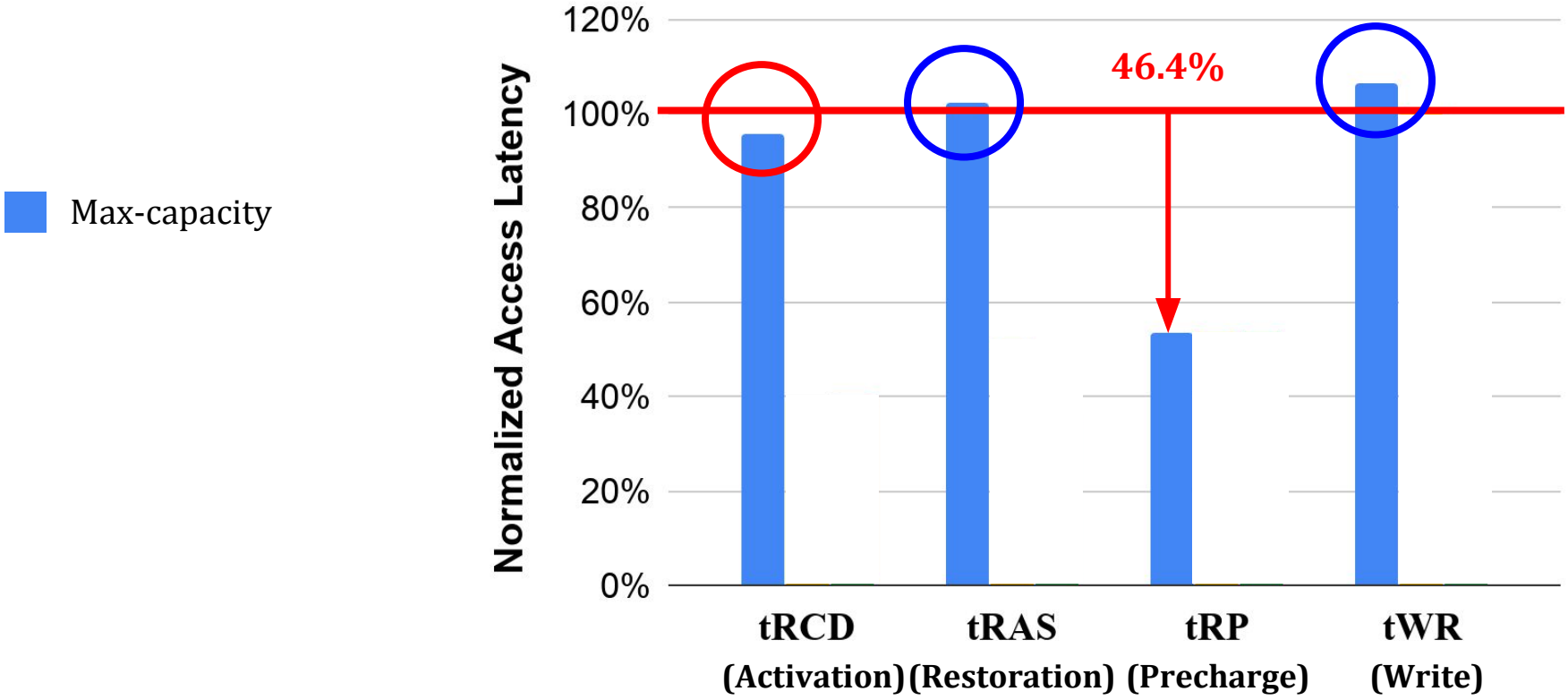
Methodology

- Model a DRAM subarray based on Rambus DRAM technology parameters [1].
- Scaled to 22 nm according to the ITRS roadmap [2].
- 22nm PTM-HP transistor model [3].

SPICE model will be available in July:
github.com/CMU-SAFARI/clrdram

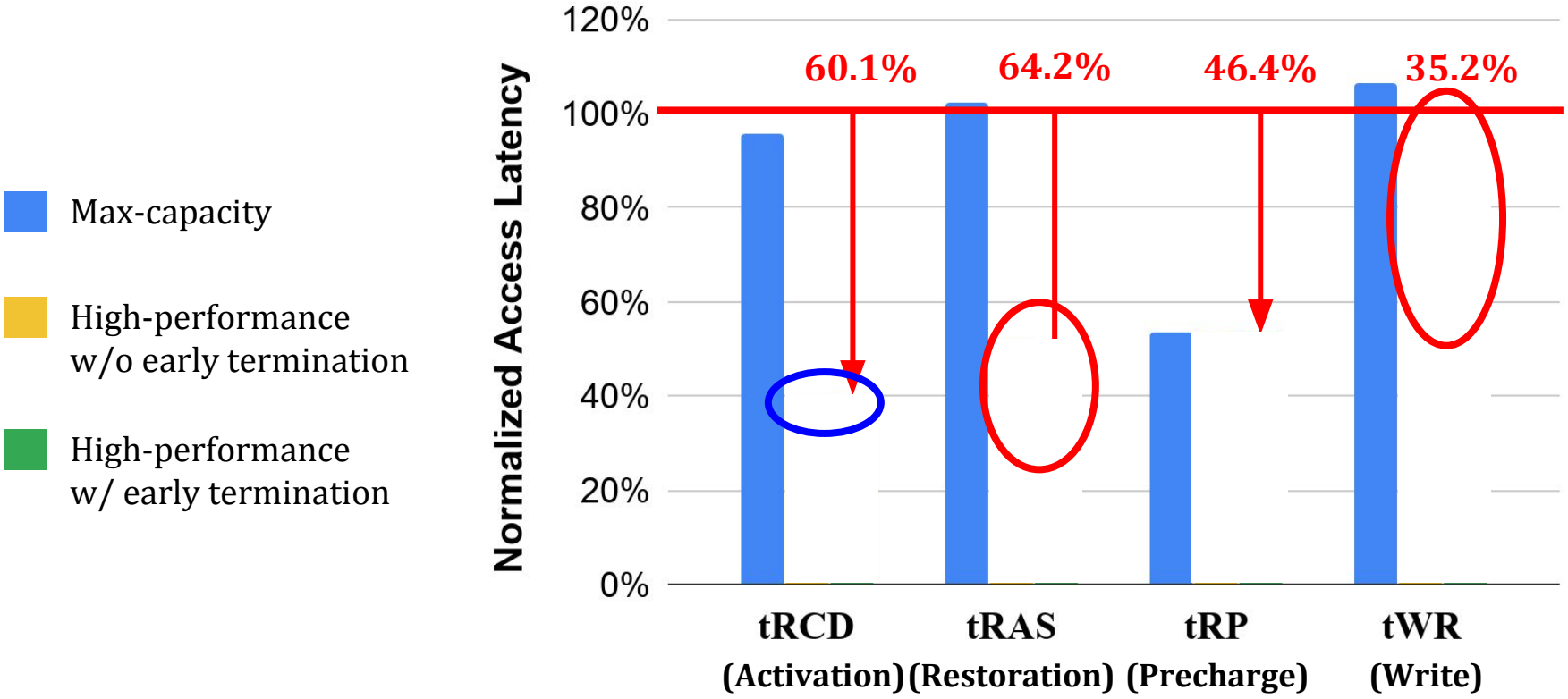
[1] Rambus, "DRAM Power Model (2010)," http://www.rambus.com/energy
[2] ITRS Roadmap, http://www.itrs2.net/itrs-reports.html
[3] http://ptm.asu.edu/

SPICE Simulation: Max-Capacity Mode Latencies



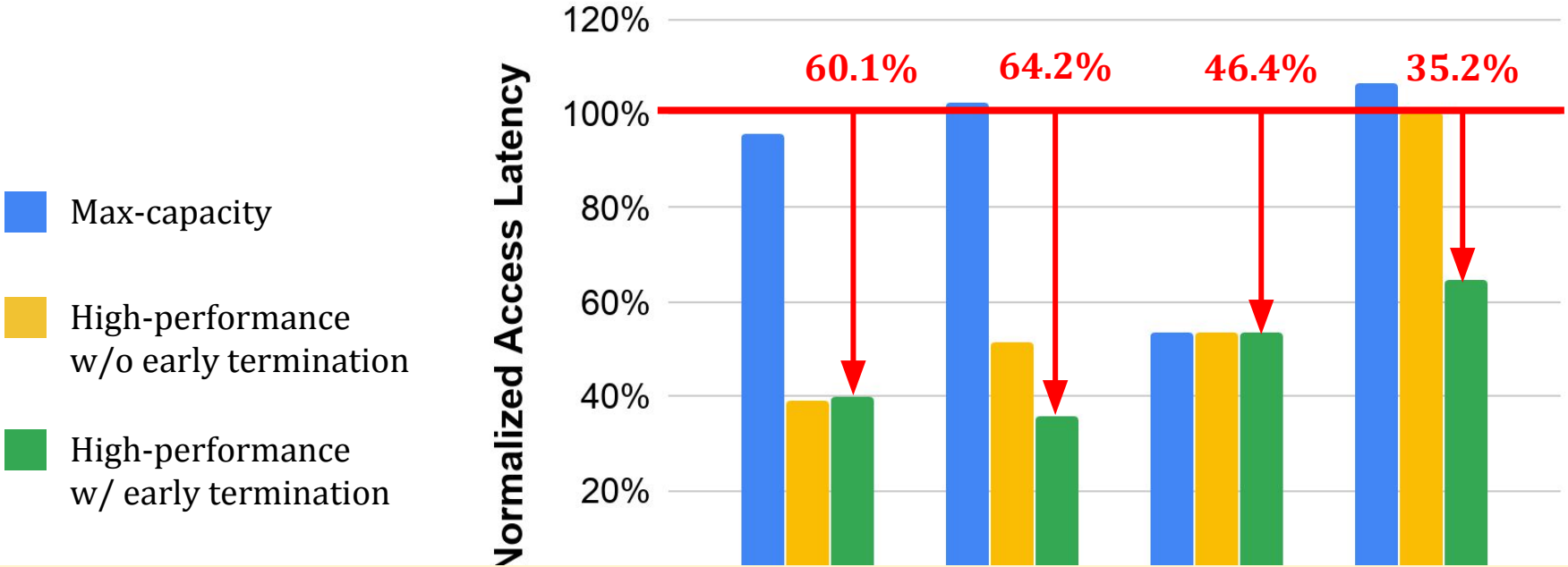
*The **tRP** reduction of coupling precharge units also applies to max-capacity mode.

SPICE Simulation: High-Performance Mode Latencies



*The **tRP** reduction of coupling precharge units also applies to max-capacity mode.

SPICE Simulation: High-Performance Mode Latencies



CLR-DRAM *reduces* DRAM latency by 35.2% to 64.2% in high-performance mode

*The **tRP** reduction of coupling precharge units also applies to max-capacity mode.

CLR-DRAM Outline

- Motivation & Goal
- DRAM Background
- CLR-DRAM
- High-Performance Mode Benefits
  - Reducing DRAM Access Latency
  - Mitigating DRAM Refresh Overhead
- **Evaluation**
  - SPICE Simulation
  - **System-level Evaluation**
- Conclusion

System-Level Evaluation - Methodology

Simulator:

Cycle-level DRAM simulator: Ramulator [Kim+, CAL'15]

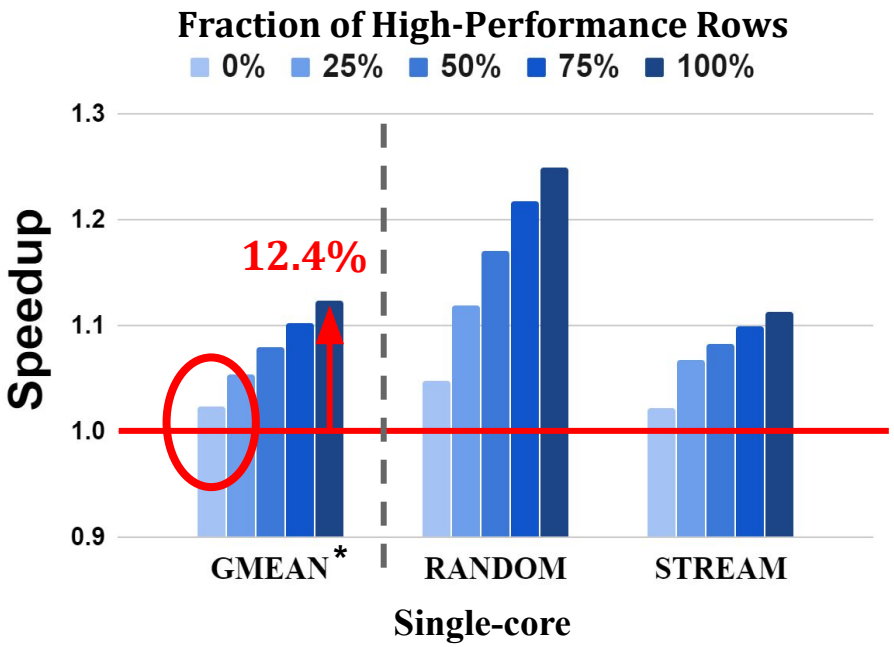
Workloads:

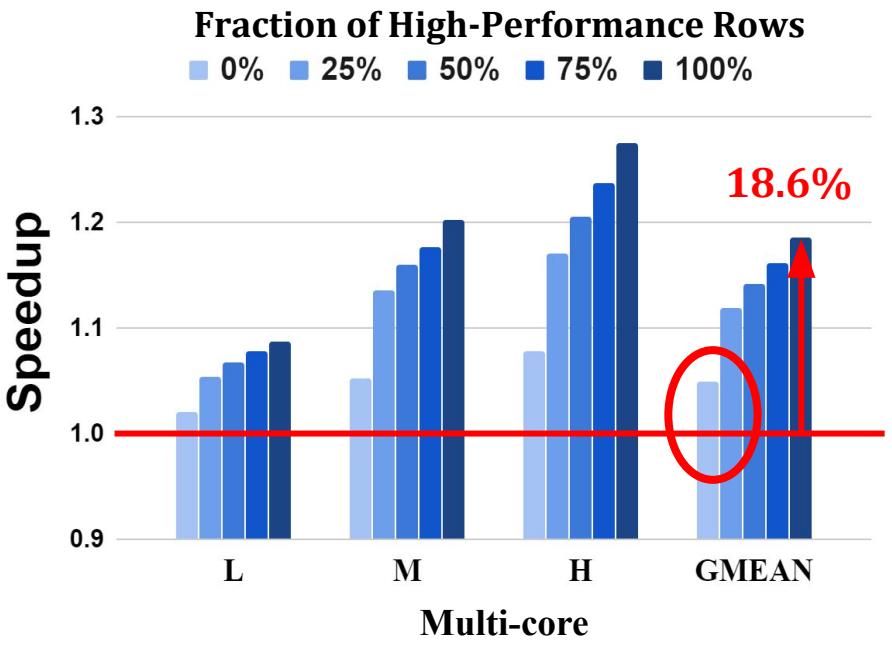
- 41 single-core workloads from SPEC CPU2006, TPC, MediaBench
- 30 in-house synthetic random and stream access workloads
- 90 multi-programmed four-core workloads
  - *By randomly choosing from our real single-core workloads*

System Parameters:

- 1/4 core system with 8MB LLC
- 5 configurations: **X**% of the DRAM rows configured to high-performance mode.
  - **X** = **25**, **50**, **75**, **100**. Plus a **X**=**0** case where all rows are max-capacity mode.
  - Map **X**% of the most accessed pages of workloads to high-performance mode rows.

CLR-DRAM Performance





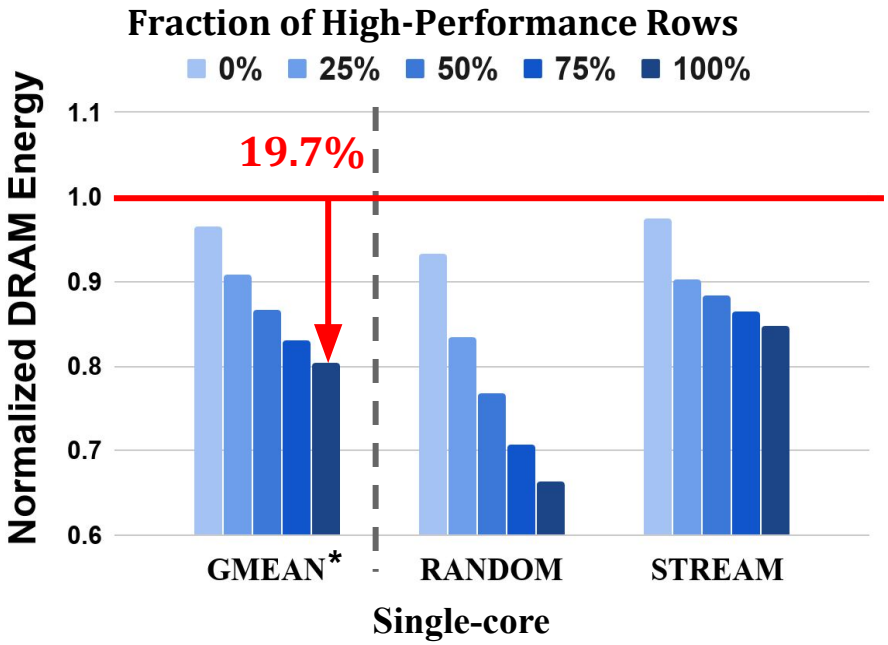
CLR-DRAM *improves* system performance for both single-core and multi-core workloads

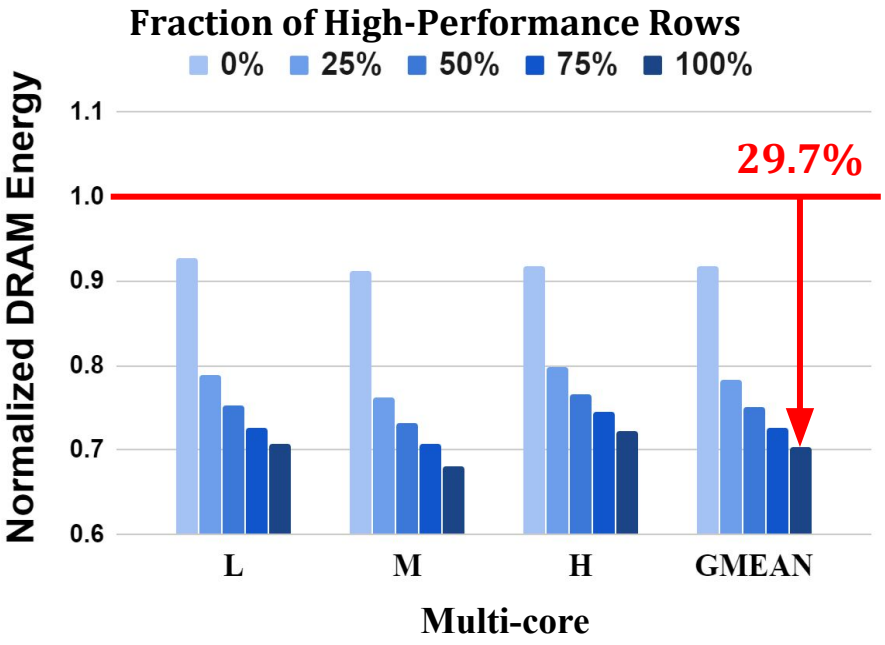
SAFARI

***GMEAN** is the geometric mean of the speed up of the 41 real single-core workloads.

L, M, H stand for different multi-core workload groups with different memory-intensity.

CLR-DRAM Energy Savings



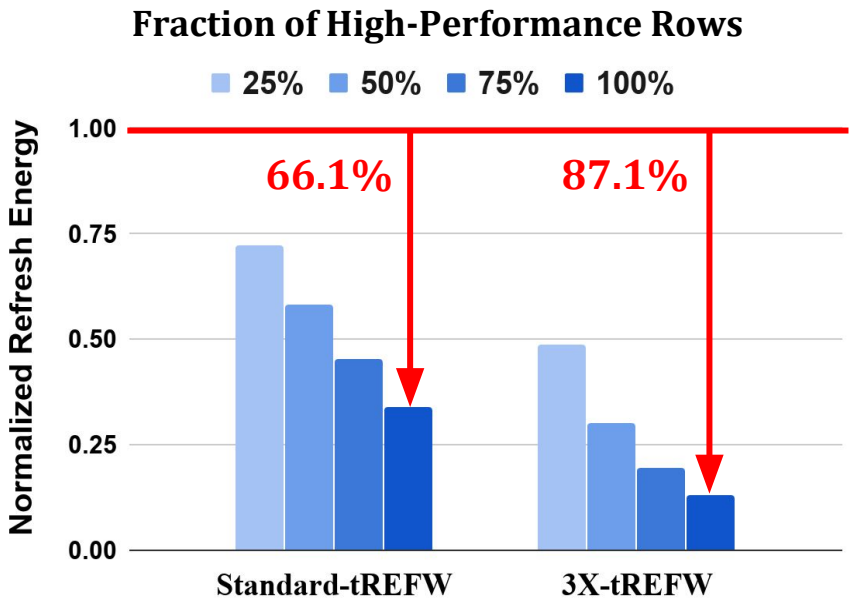


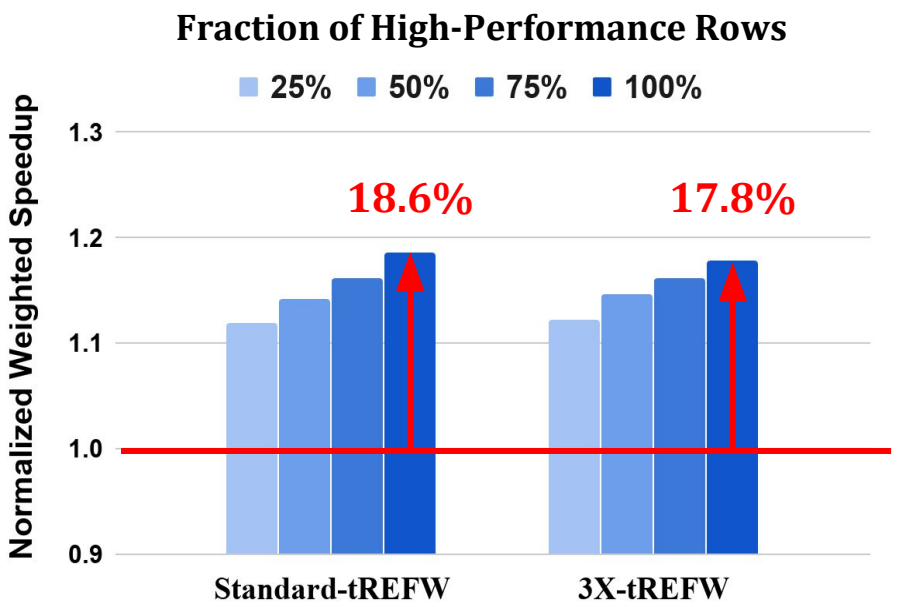
CLR-DRAM *saves* DRAM energy for both single-core and multi-core workloads

***GMEAN** is the geometric mean of the speed up of the 41 real single-core workloads.

L, M, H stand for different multi-core workload groups with different memory-intensity.

Mitigating Refresh Overhead





CLR-DRAM significantly reduces DRAM refresh energy

Overhead of CLR-DRAM

DRAM Chip Area Overhead:

- 3.2% based on our conservative estimates (real overhead is likely lower).

Memory Capacity Overhead:

- **X**% of the rows in high-performance mode incurs **X/2**% capacity overhead.

[More details in the paper]

CLR-DRAM is a low-cost architecture

Other Results, Analyses and Design Details in the Paper

Sensitivity Study of Reducing Refresh Rate (increasing tREFW)

- The trade-off between less refresh operations (increase tREFW) and increased access latency (tRCD and tRAS).
- The system-level performance and DRAM refresh energy impact of the trade-off.

Efficient Control of the Bitline Mode Select Transistors

- Only two control signals required per-bank for *all* its subarrays.
  - Ensures correct SA operation in max-capacity mode.
  - Maximizing latency-reduction in high-performance mode.

Modifications to Subarray Column Access Circuitry

- Column (read/write) access to a high-performance row maintain full bandwidth.

CLR-DRAM Outline

- Motivation & Goal
- DRAM Background
- CLR-DRAM
- High-Performance Mode Benefits
  - Reducing DRAM Access Latency
  - Mitigating DRAM Refresh Overhead
- Evaluation
  - SPICE Simulation
  - System-level Evaluation
- Conclusion

Conclusion

- We introduce **CLR-DRAM** (Capacity-Latency-Reconfigurable **DRAM**)
  A new DRAM architecture enabling dynamic fine-grained reconfigurability between high-capacity and low-latency operation.

- **CLR-DRAM** can dynamically reconfigure every single DRAM row to operate in either
  - **Max-capacity mode**: almost the same storage density as the baseline density-optimized architecture by letting each DRAM cell operate separately.
  - **High-performance mode:** low access latency and low refresh overhead by coupling every two adjacent DRAM cells in the row and their sense amplifiers.

- **Key Results**
  - Reduces four major DRAM timing parameters by **35.2-64.2%.**
  - Improves average system performance by **18.6%** and saves DRAM energy by **29.7%**.

- We hope that CLR-DRAM can be exploited to develop more flexible systems that can adapt to the diverse and changing DRAM capacity and latency demands of workloads.

CLR-DRAM:
A Low-Cost DRAM Architecture Enabling Dynamic Capacity-Latency Trade-off

Haocong Luo Taha Shahroodi Hasan Hassan Minesh Patel
A. Giray Yaglıkçı Lois Orosa Jisung Park Onur Mutlu

ETH Zürich
SAFARI Research Group
safari.ethz.ch
上海科技大学
ShanghaiTech University