# DRAM REFRESH MANAGEMENT

Mahdi Nazm Bojnordi

Assistant Professor

School of Computing

University of Utah

THE UNIVERSITY OF UTAH

# Overview

- Upcoming deadline
  - <span style="color:red">Feb. 27th: homework assignment will be posted</span>

- This lecture
  - DRAM address mapping
  - DRAM refresh basics
  - Smart refresh
  - Elastic refresh
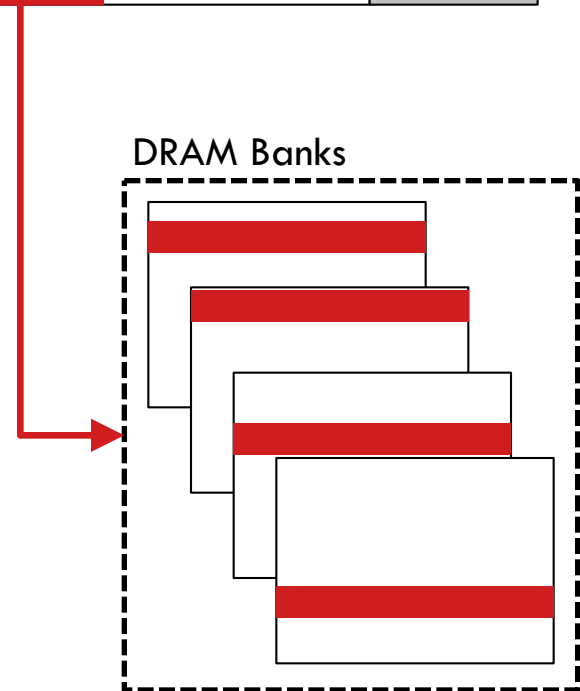  - Avoiding or pausing refreches

# DRAM Address Mapping

□ Where to store cache lines in main memory?

Typical Mapping

| Row | Bank | Column | Block |
|---|---|---|---|

DRAM Banks

**Application A:**

Good distribution of memory requests among DRAM banks.

# DRAM Address Mapping

- Where to store cache lines in main memory?

Typical Mapping

| Row | Bank | Column | Block |
|---|---|---|---|

DRAM Banks

**Application B:**

Unbalanced distribution of memory requests among DRAM banks.
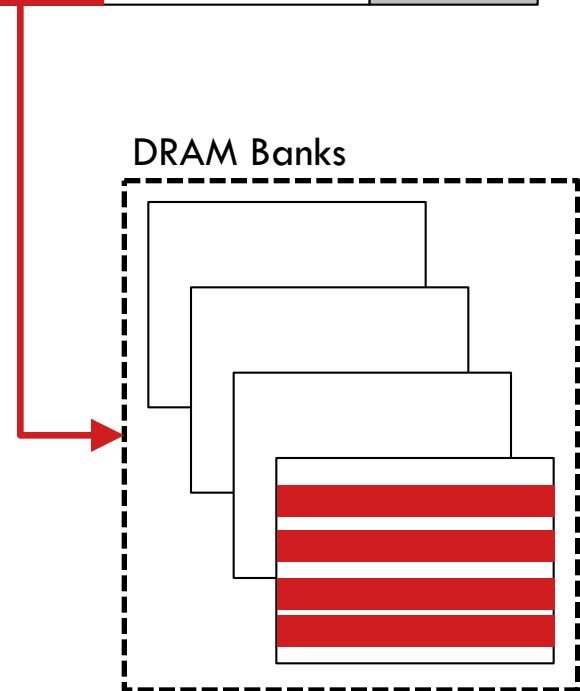
# DRAM Address Mapping

☐ How to compute bank ID?

Custom Mapping
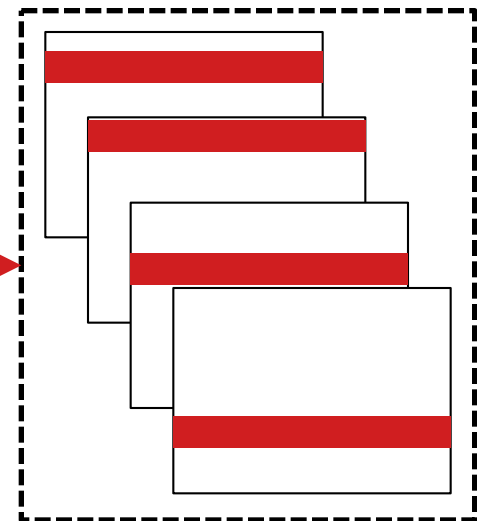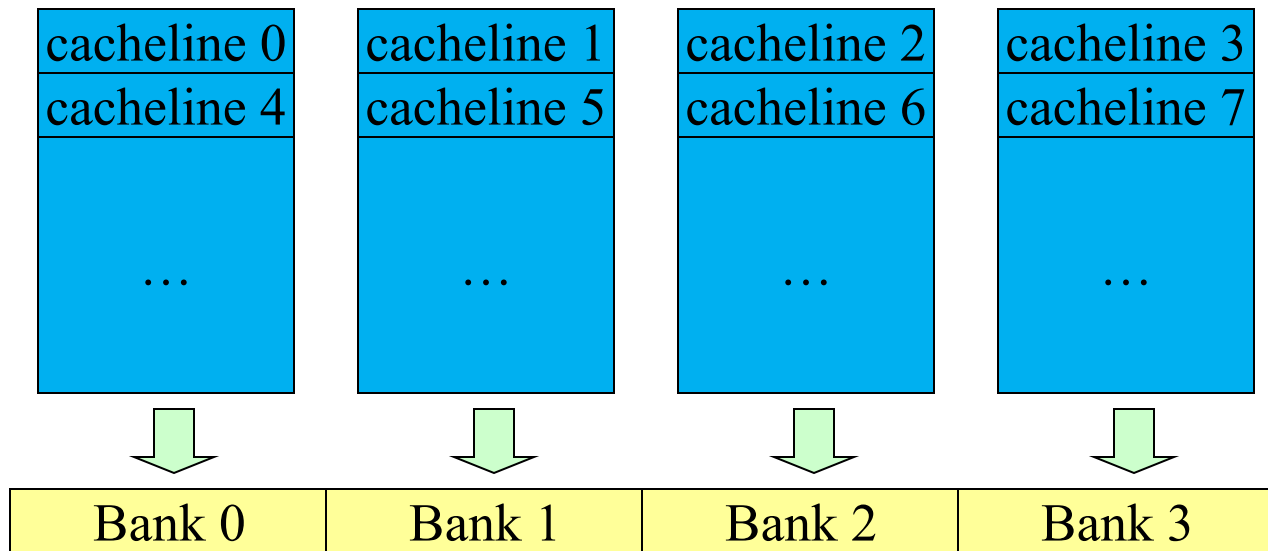
| Row | **Bank** | Row | Column | Block |
|-----|----------|-----|--------|-------|

DRAM Banks

**Application B:**

Good distribution of memory requests among DRAM banks.

# Cache Line Interleaving

| cacheline 0 | cacheline 1 | cacheline 2 | cacheline 3 |
| cacheline 4 | cacheline 5 | cacheline 6 | cacheline 7 |
| … | … | … | … |

| Bank 0 | Bank 1 | Bank 2 | Bank 3 |

Address format

| r | p-b | k | b |
|---|---|---|---|
| page index | page offset | bank | page offset |
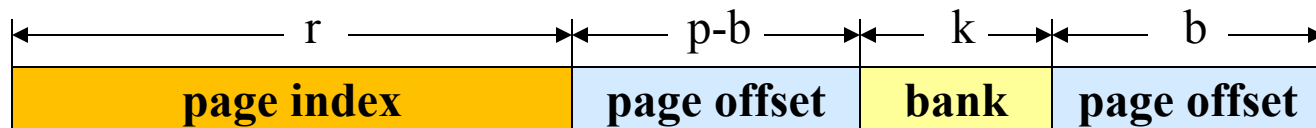
Spatial locality is not well preserved!

# Page Interleaving

| | | | |
|---|---|---|---|
| Page 0 | Page 1 | Page 2 | Page 3 |
| Page 4 | Page 5 | Page 6 | Page 7 |
| … | … | … | … |

| Bank 0 | Bank 1 | Bank 2 | Bank 3 |
|---|---|---|---|

Address format
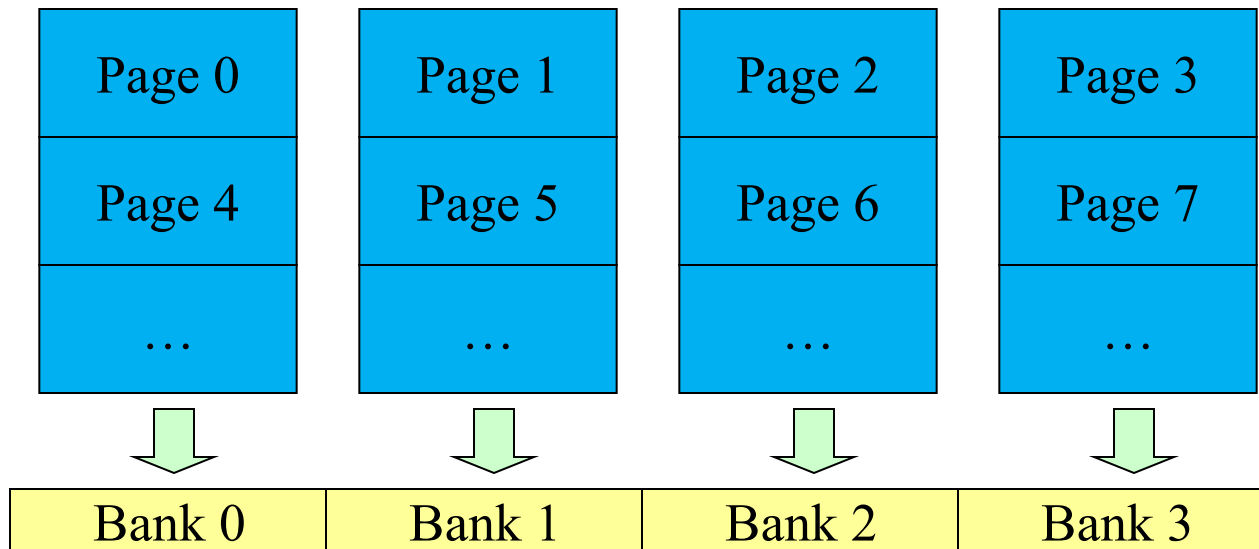
| ← r → | ← k → | ← p → |
|---|---|---|
| **page index** | **bank** | **page offset** |
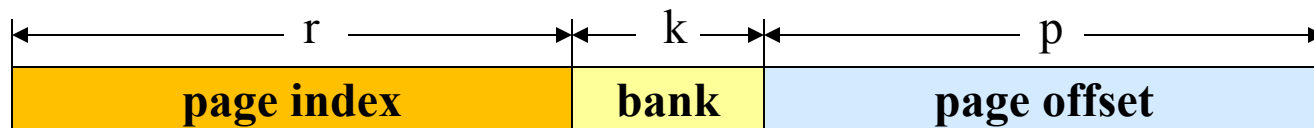
# Cache Line Mapping

☐ Bank index is a subset of set index

**Cache line interleaving**

| ← r → | ← p-b → | ← k → | ← b → |
|---|---|---|---|
| page index | page offset | bank | page offset |

**Page interleaving**

| ← r → | ← k → | ← p → |
|---|---|---|
| page index | bank | page offset |
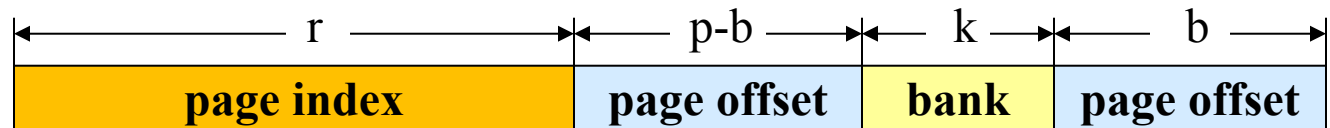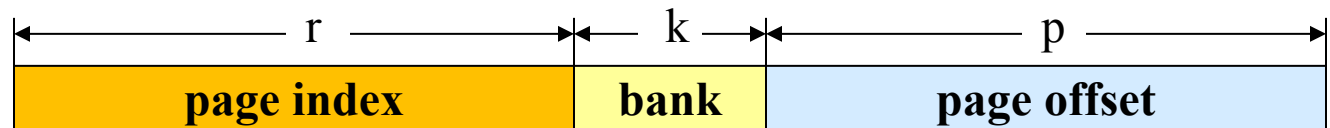
**Cache-related representation**

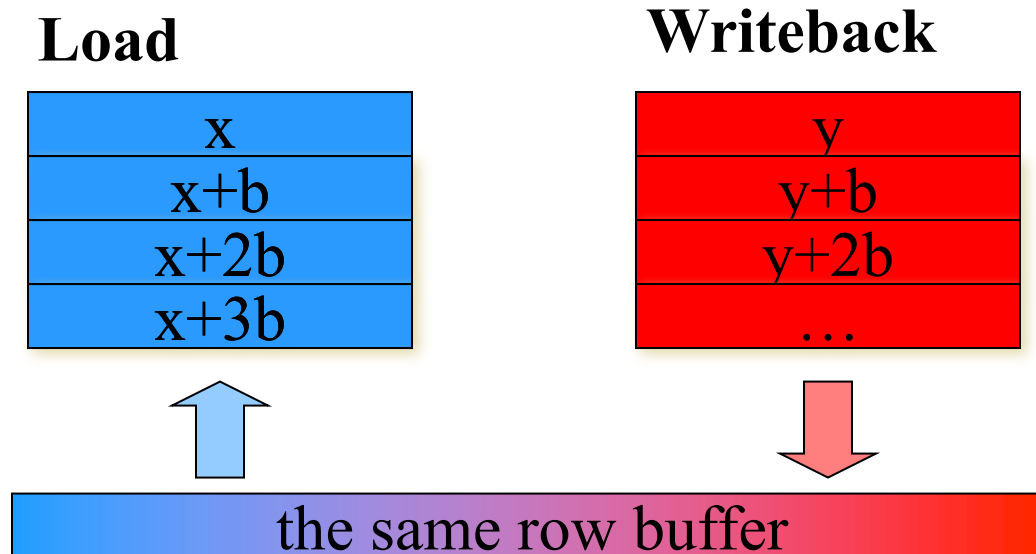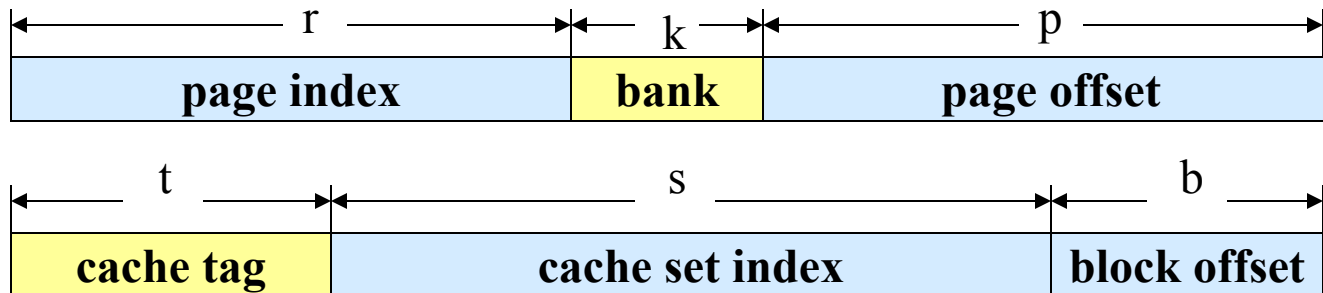| ← t → | ← s → | ← b → |
|---|---|---|
| cache tag | cache set index | block offset |
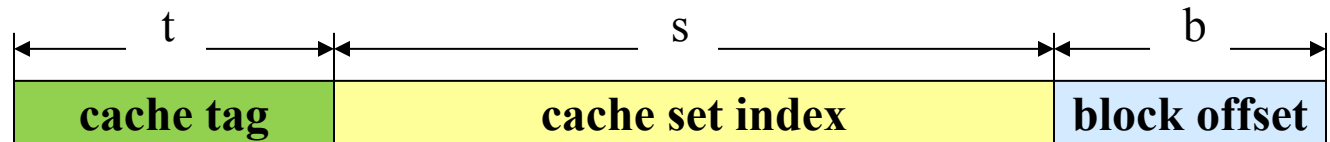
# Row Buffer Conflict

□ Problem: interleaving load and writeback streams with the same access pattern to the banks may result in row buffer misses
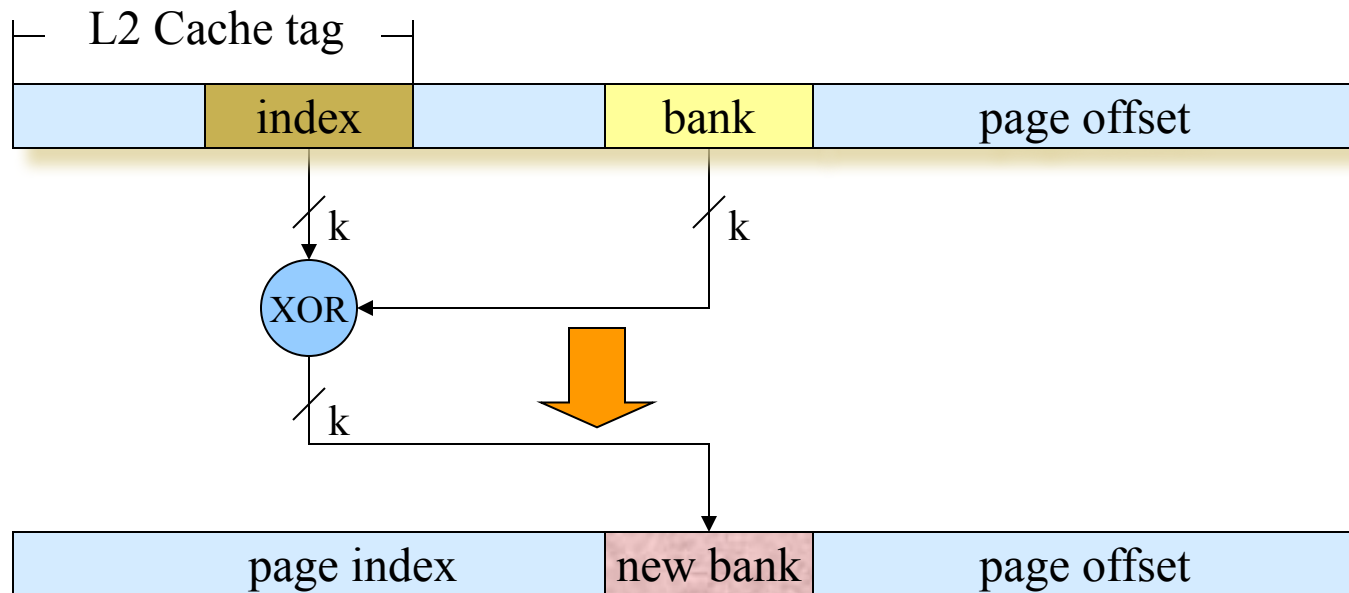
# Key Issues

☐ To exploit spatial locality, use maximal interleaving granularity (or row-buffer size)

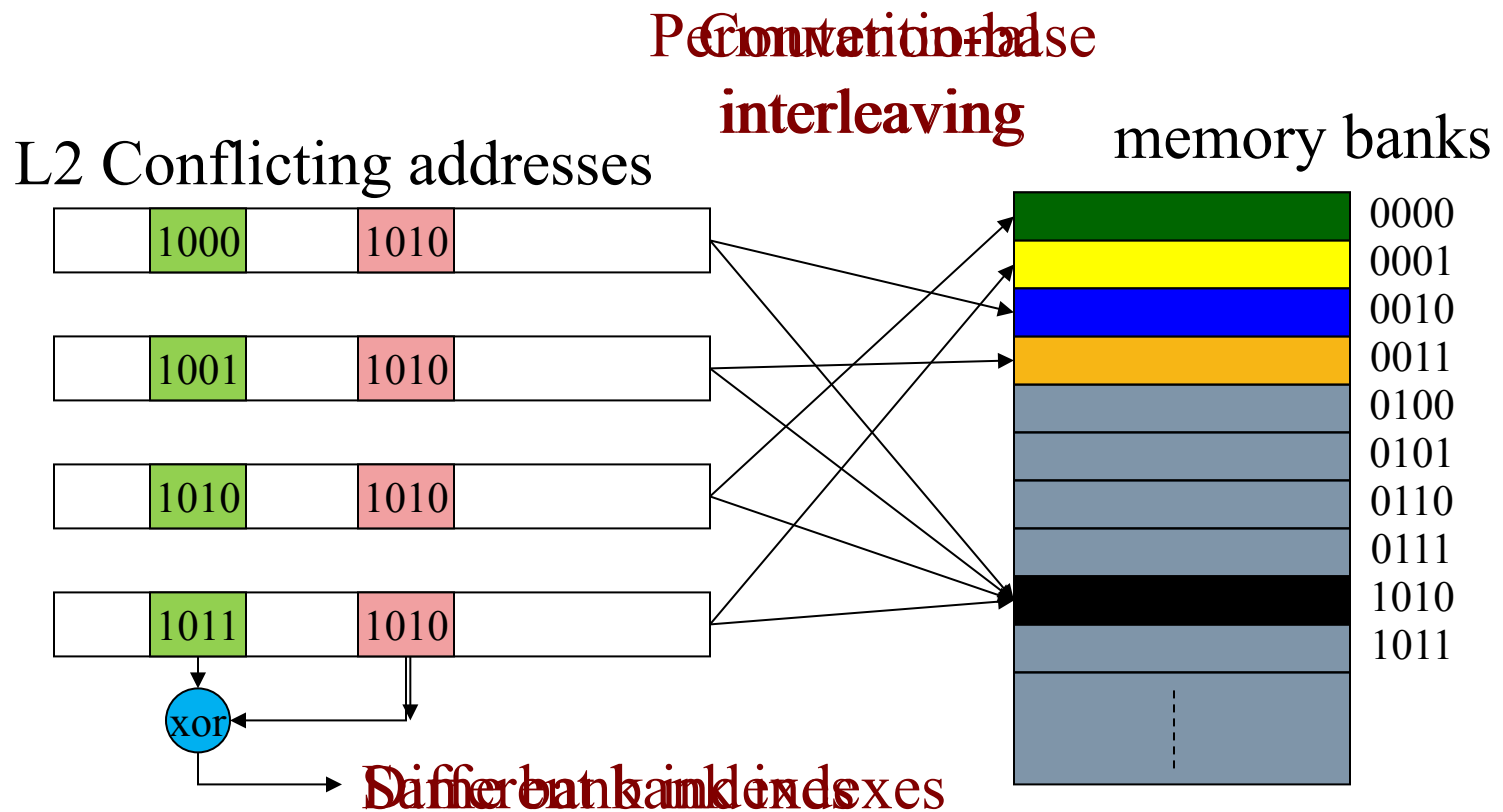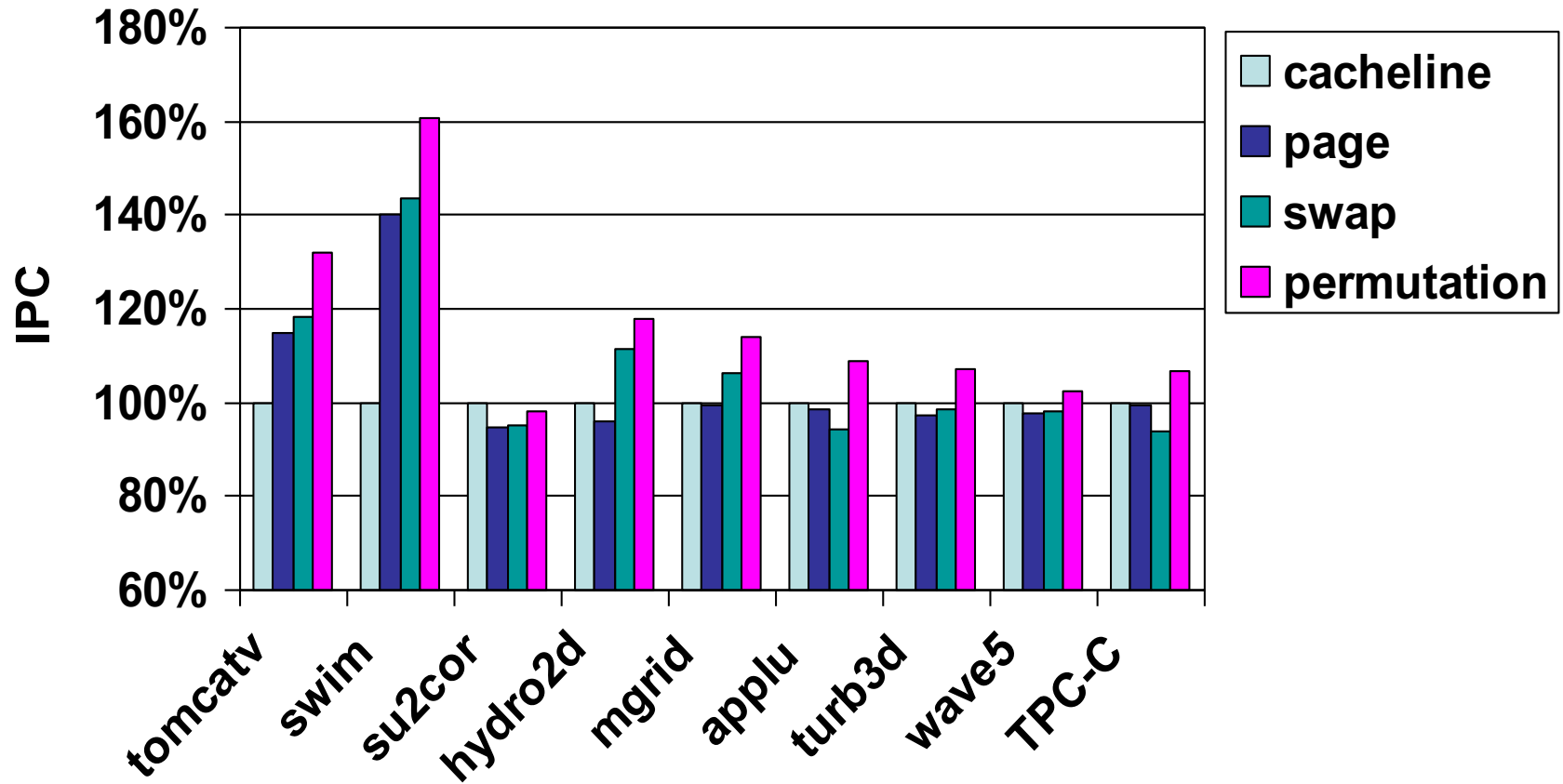☐ To reduce row buffer conflicts, use only those bits in cache set index for "bank bits"

| ← r → | ← k → | ← p → |
|-------|-------|-------|
| page index | bank | page offset |

| ← t → | ← s → | ← b → |
|-------|-------|-------|
| cache tag | cache set index | block offset |

# Permutation-based Interleaving



*[Zhang'00]*

# Permutation-based Interleaving

□ New bank index



[Zhang'00]

# Permutation-based Interleaving



*[Zhang'00]*

# DRAM Refresh

- DRAM cells lose charge over time

- Periodic refresh operations are required to avoid data loss

- Two main strategies for refreshing DRAM cells
  - Burst refresh: refresh all of the cells each time
    - Simple control mechanism (e.g., LPDDRx)
  - Distributed refresh: a group of cells are refreshed
    - Avoid blocking memory for a long time
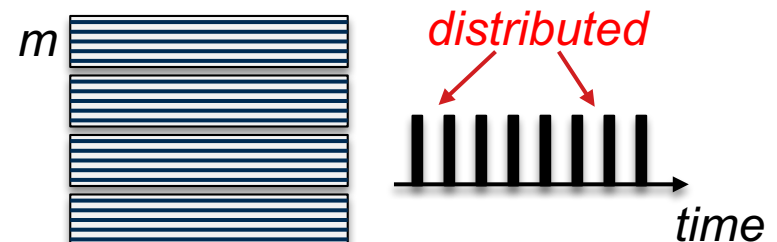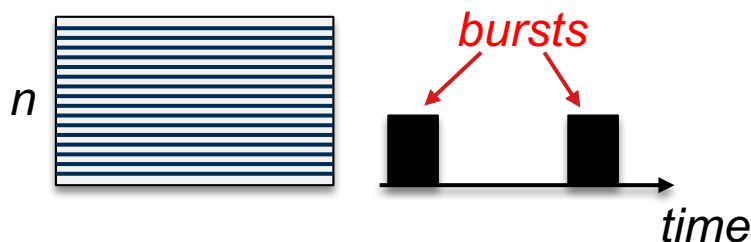
# Refresh Basics

- tRET: the retention time of DRAM leaky cells (64ms)
  - All cells must be refreshed within tRET to avoid data loss

- tREFI: refresh interval, which is the gap between two refresh commands issues by the memory controller
  - MC sends 8192 auto-refresh commands to refresh one bin at a time
    - tREFI = tRET/8192 = 7.8us

- tRFC: the time to finish refreshing a bin (refresh completion)

- What is the bin size?

# Refresh Basics

☐ tRFC increases with chip capacity

**Impact of chip density on refresh completion time**



*[Stuecheli'10]*
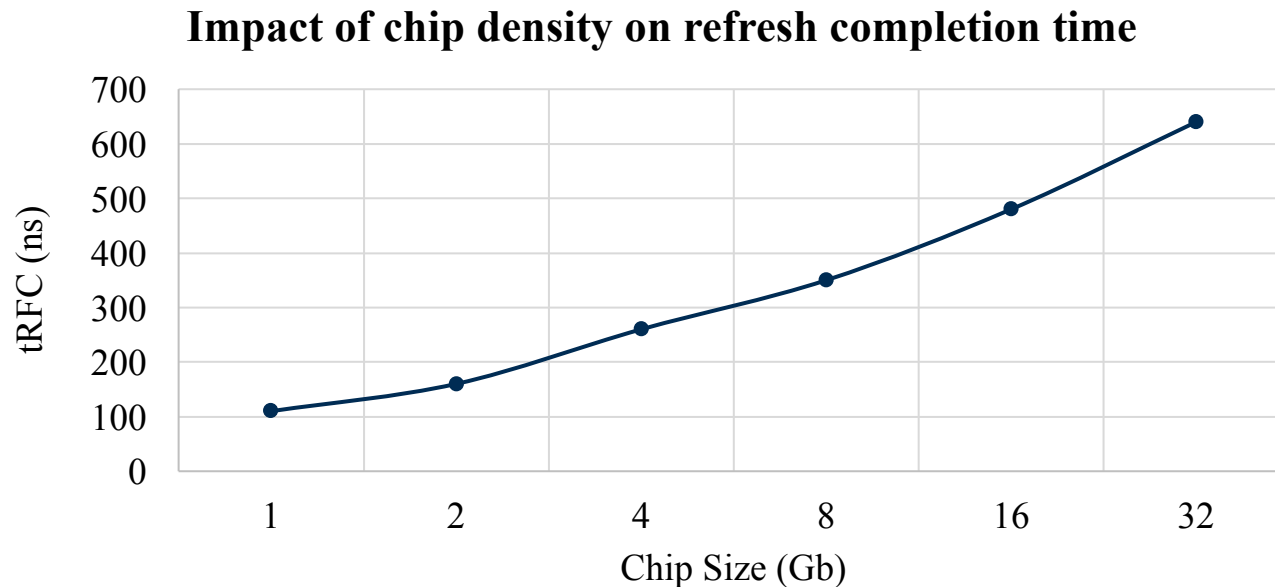
# Controlling Refresh Operations

- CAS before RAS (CBR)
  - DRAM memory keeps track of the addresses using an internal counter

- RAS only refresh (ROR)
  - Row address is specified by the controller; similar to a pair of activate and precharge

- Auto-refresh vs. self refresh
  - Every 7.8us a REF command is sent to DRAM (tRAS+tRP)
  - LPDDR turns off IO for saving power while refreshing multiple rows

# Refresh Granularity
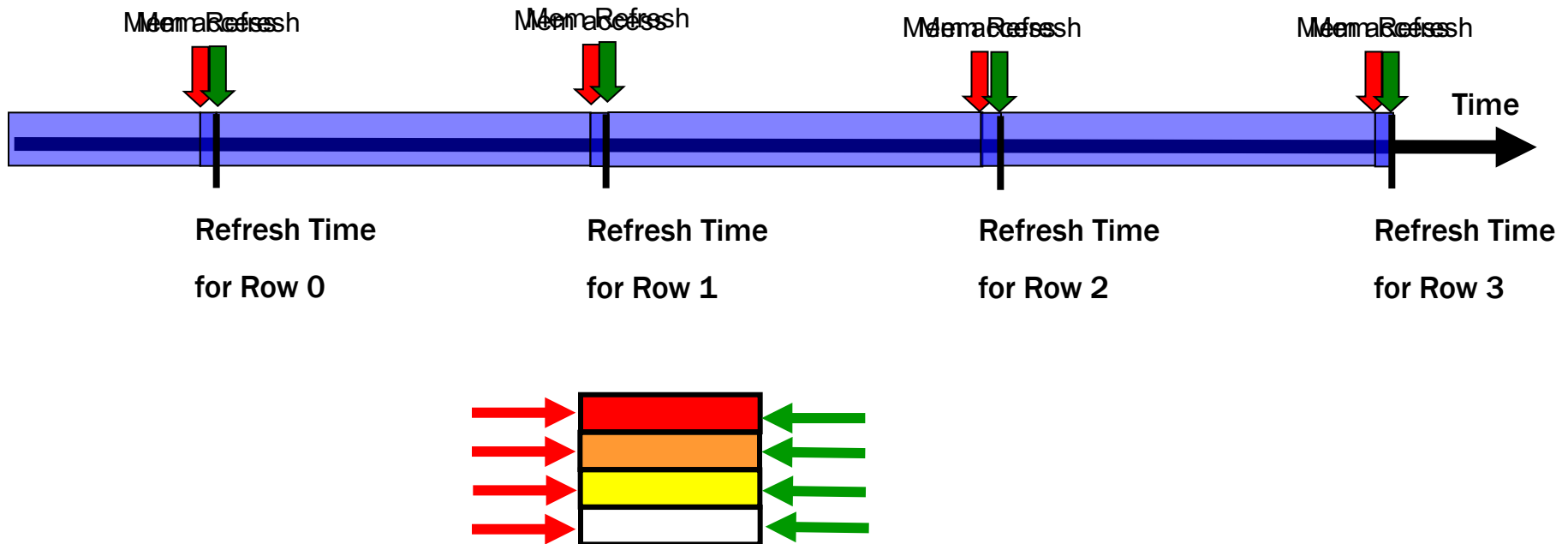
☐ All bank vs. per bank refresh
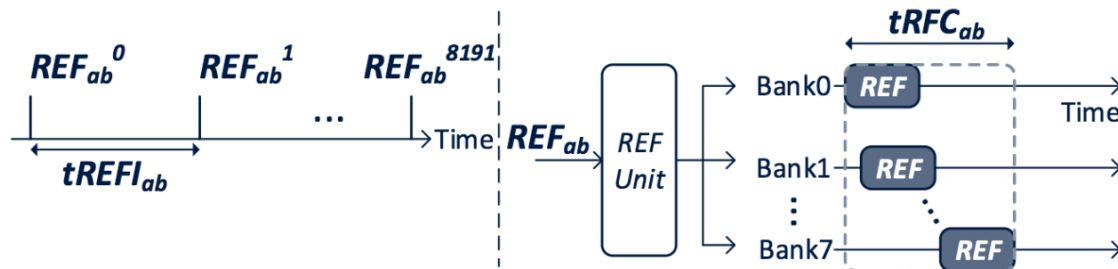


(a) All-bank refresh ($REF_{ab}$) frequency and granularity.
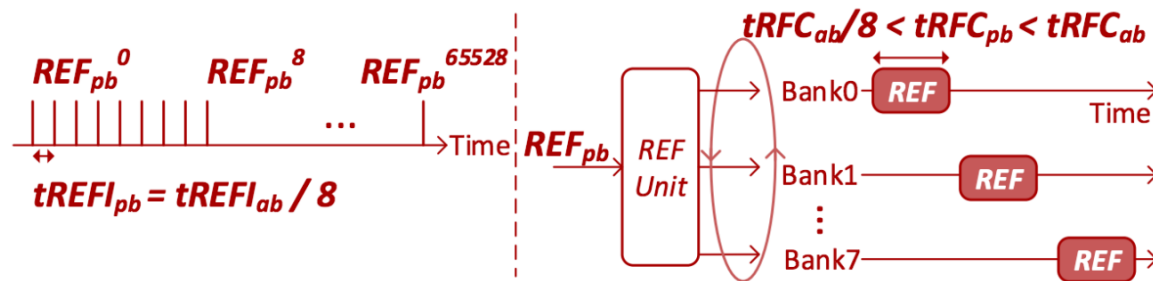
(b) Per-bank refresh ($REF_{pb}$) frequency and granularity.

# Optimizing DRAM Refresh

☐ Observation: each row may be accessed as soon as it is to be refreshed

# Smart Refresh

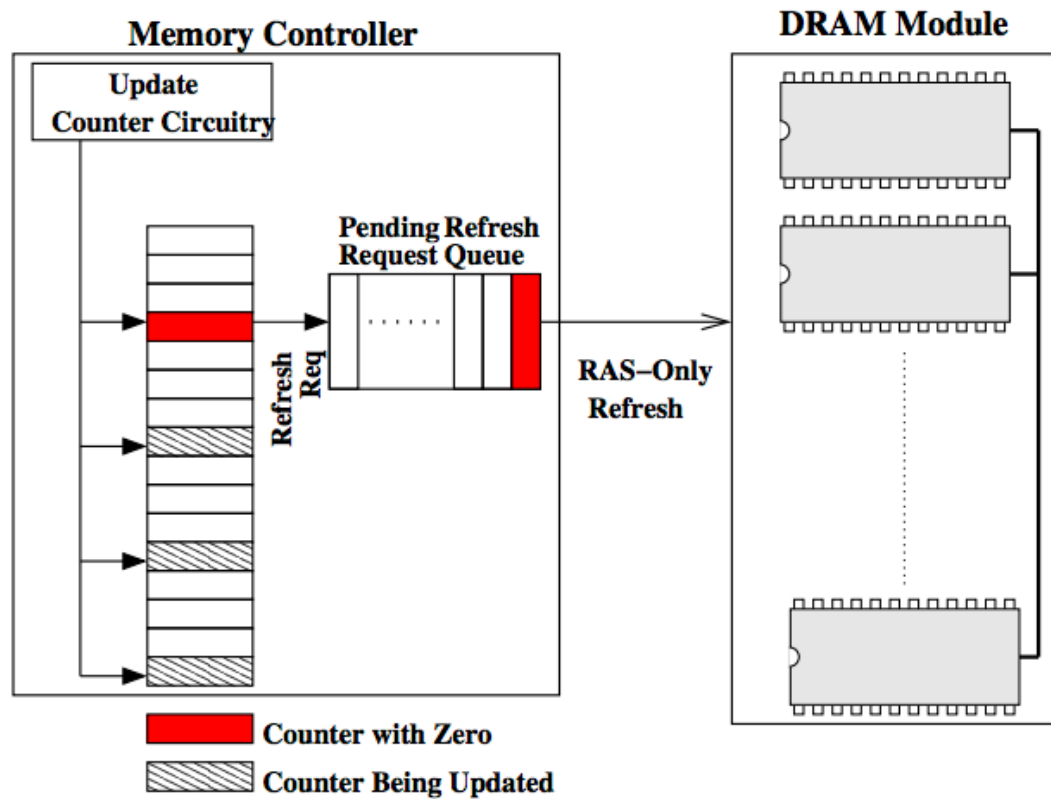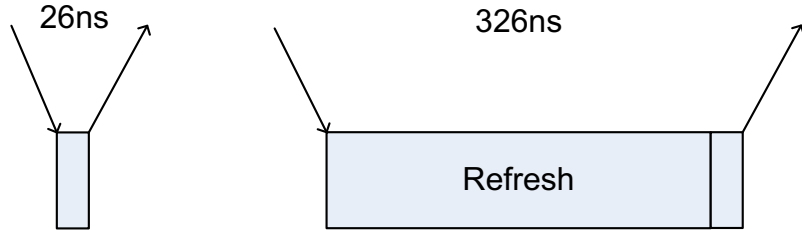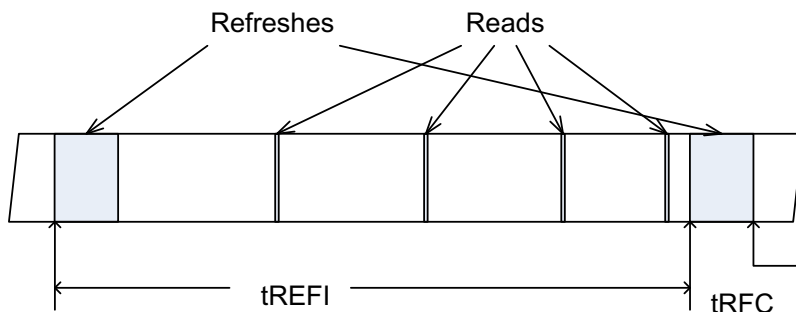☐ Idea: avoid refreshing recently accessed rows



**Figure 5: Smart Refresh Control Schematic**

*[Ghosh'07]*

# Diverse Impacts of Refresh

26ns

326ns

Refresh

Worst Case Refresh Hit DRAM Read

| DRAM capacity | tRFC | bandwidth overhead (95$^o$C per Rank) | latency overhead (95$^o$C) |
|---|---|---|---|
| 512Mb | 90ns | 2.7% | 1.4ns |
| 1Gb | 110ns | 3.3% | 2.1ns |
| 2Gb | 160ns | 5.0% | 4.9ns |
| 4Gb | 300ns | 7.7% | 11.5ns |
| 8Gb | 350ns | 9.0% | 15.7ns |

Refreshes    Reads

tREFI    tRFC

*[Stuecheli'10]*

# Elastic Refresh

- Send refreshes during periods of inactivity
- Non-uniform request distribution
- Refresh overhead just has to fit in free cycles
- Initially not aggressive, converges with delay until empty (DUE) as refresh backlog grows
- Latency sensitive workloads are often lower bandwidth
- Decrease the probability of reads conflicting with refreshes
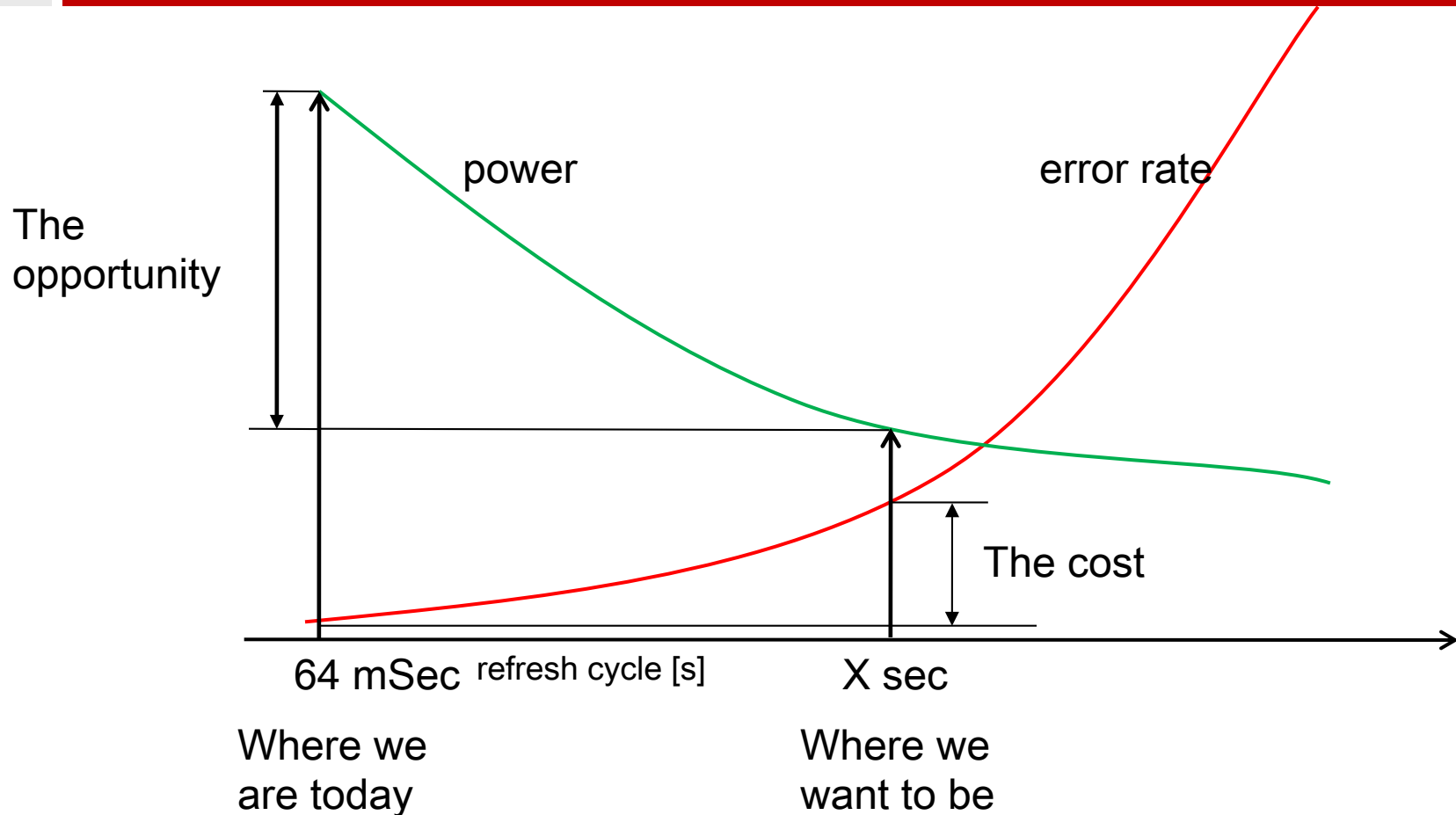
*[Stuecheli'10]*

# Elastic Refresh

☐ Introduce refresh backlog dependent idle threshold

☐ With a log backlog, there is no reason to send refresh command

☐ With a bursty request stream, the probability of a future request decreases with time

☐ As backlog grows, decrease this delay threshold
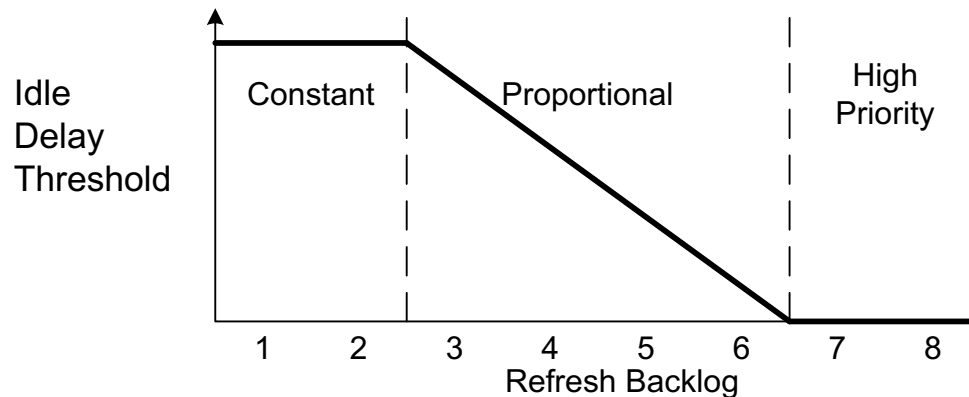


**Key:  to reduce REF and READ conflicts**
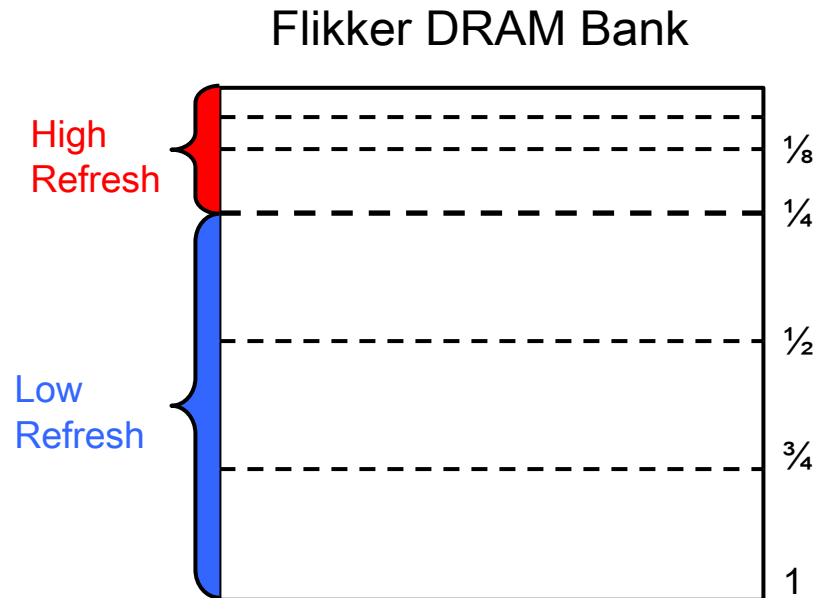
*[Stuecheli'10]*

# DRAM Refresh vs. ERROR Rate



power

error rate

The opportunity

The cost

64 mSec  refresh cycle [s]

X sec

Where we are today

Where we want to be

If software is able to tolerate errors, we can lower DRAM refresh rates to achieve considerable power savings
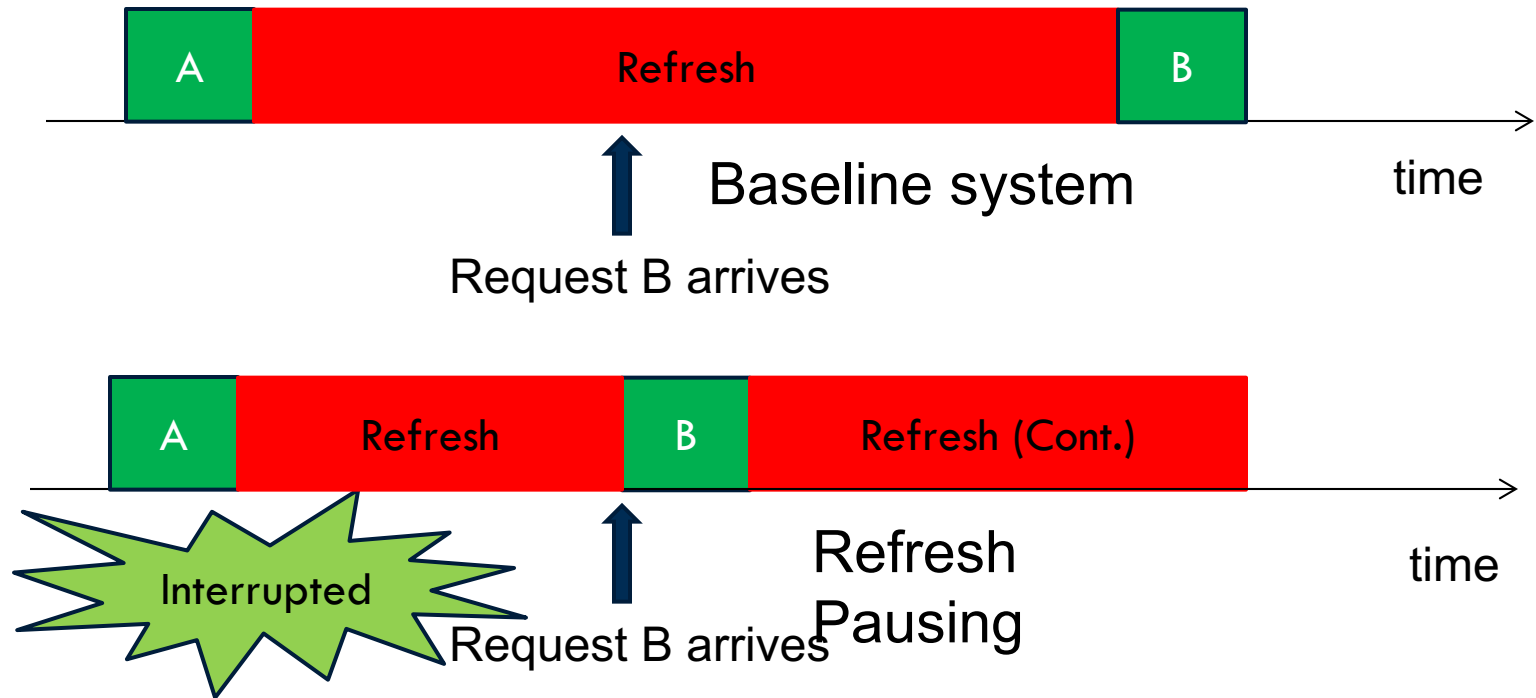
# Flikker

- Divide memory bank into high refresh part and low refresh parts

- Size of high-refresh portion can be configured at runtime

- Small modification of the Partial Array Self-Refresh (PASR) mode

Flikker DRAM Bank

High Refresh

Low Refresh

⅛
¼
½
¾
1

*[Song'14]*

# Refresh Pausing



Baseline system

Request B arrives

Refresh Pausing

Request B arrives

Interrupted

Pausing at arbitrary point can cause data loss

Pausing Refresh reduces wait time for Reads

# Performance Results