ECE 5730
Memory Systems

Spring 2009

Refresh Management
Memory Power Management

Cornell University

Announcements

- Quiz 9
  - Average = 8.9
- Quiz 10 on Tuesday
- Pick up your project proposal after class
- Project status report
  - Emailed to me by 5pm tomorrow
  - 1-2 paragraphs
  - 2 points off final project grade if late

DRAM Refresh

- *Refresh* involves restoring the charge on the capacitors of a given row through row activation

row activate, then precharge

- Each row must be refreshed at a specified rate
  - Typically, every 64ms in SDRAMs
  - Typically, a refresh op must be performed every 7.8us on average to refresh the entire SDRAM in 64ms

the refresh is always for 8192 rows, so the 64ms timing requirement gives us

$$\frac{64ms}{8192} \simeq 7.8us$$

Asynchronous DRAM Refresh

- **Asynchronous DRAMs were refreshed by the MC performing a RAS operation (*RAS-only refresh*)**
  - **Row address of the row to be refreshed + RAS**

buffer → precharge

- **MC maintained a refresh counter and a refresh row address register**

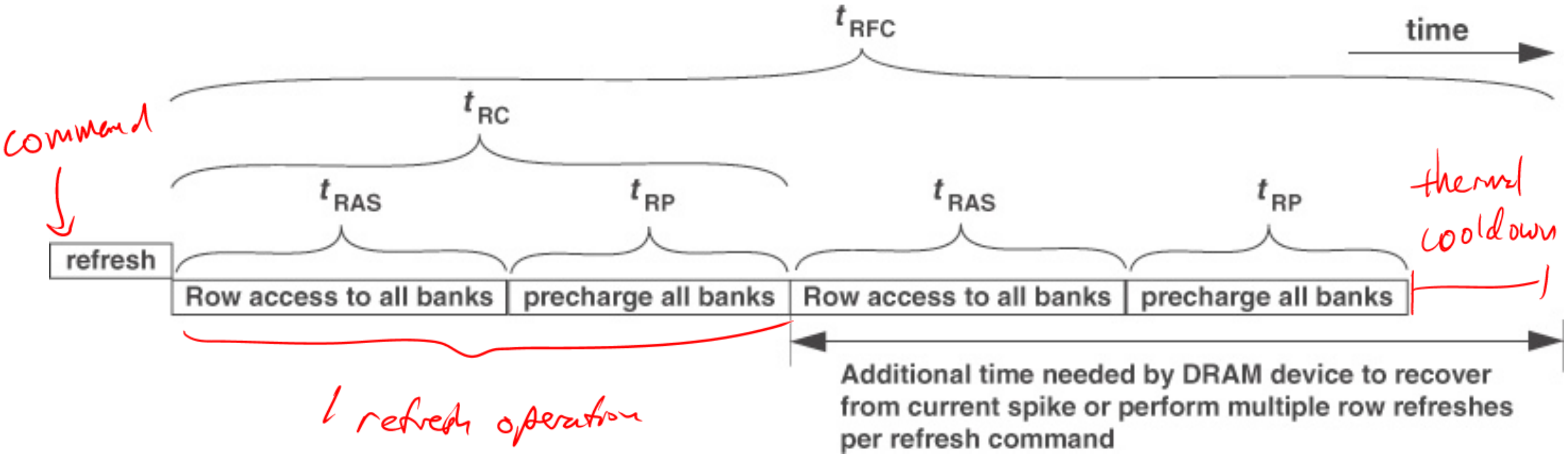
→ To control where we are in the refresh cycle

SDRAM Refresh Operations

- *Auto refresh*
  - The DRAM contains a refresh row address register

    → where are we in the refresh?
  - Auto refresh command causes row(s) to be refreshed and the address register to be incremented

    → advance the pointer to the next row we need to refresh.
  - All rows must be precharged beforehand

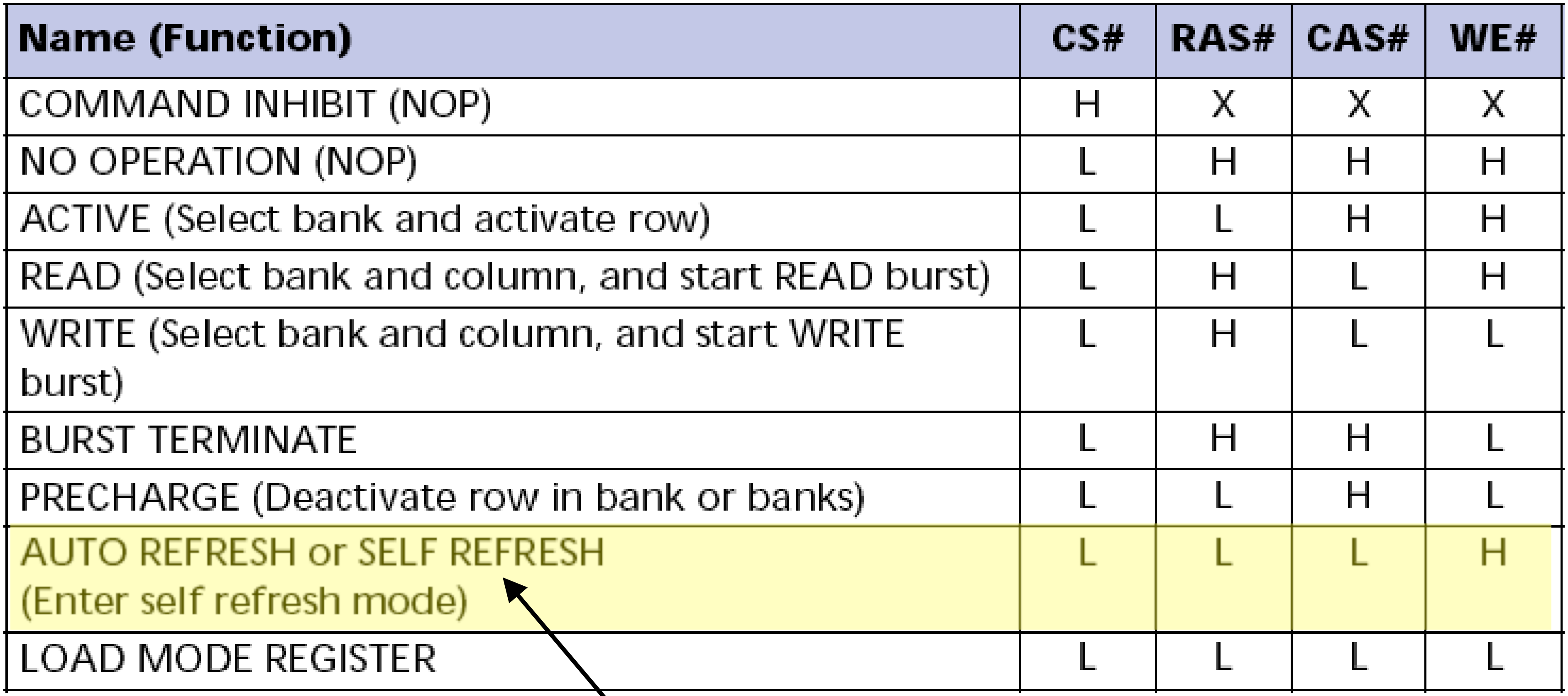


Additional time needed by DRAM device to recover from current spike or perform multiple row refreshes per refresh command

SDRAM Refresh Operations

- *Self refresh*
  - DRAM is put into a low power state → idle state
  - DRAM internally performs periodic refreshes to maintain data integrity → just to keep data alive

SDRAM Refresh Commands



Name (Function)	CS#	RAS#	CAS#	WE#
COMMAND INHIBIT (NOP)	H	X	X	X
NO OPERATION (NOP)	L	H	H	H
ACTIVE (Select bank and activate row)	L	L	H	H
READ (Select bank and column, and start READ burst)	L	H	L	H
WRITE (Select bank and column, and start WRITE burst)	L	H	L	L
BURST TERMINATE	L	H	H	L
PRECHARGE (Deactivate row in bank or banks)	L	L	H	L
AUTO REFRESH or SELF REFRESH (Enter self refresh mode)	L	L	L	H
LOAD MODE REGISTER	L	L	L	L

CKE is deasserted → stop the clock for self-refresh

clock enable

self refresh uses an internal timer

MC Refresh Optimizations

- Deferred refresh
  - Refresh command is delayed until later if a read or high priority write cmd is pending

- Burst refresh
  - Series of refreshes is performed back-to-back

- Free refreshes (leverage read/writes as "free" refreshes)
  - Track read/write addresses to eliminate redundant refresh operations

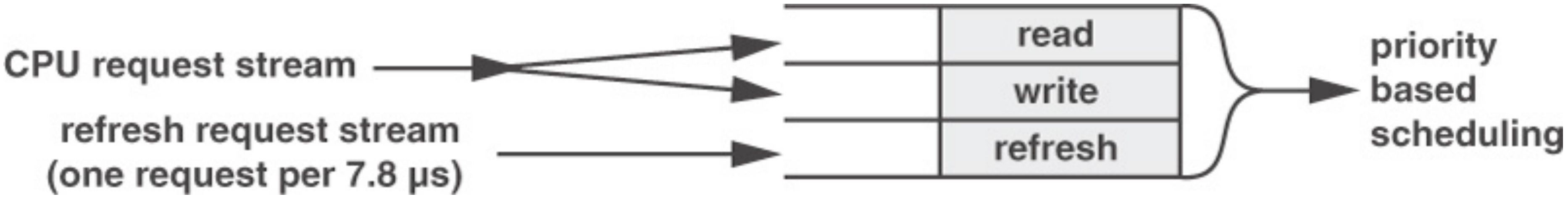
Deferred Refresh

- A refresh op must be performed every 7.8us *on average*

- SDRAM spec allows refresh to be deferred for a short period without data loss
  - For DDR3, can do a maximum of 8 refreshes in a row → for thermal / power management
  - Refresh can be deferred for up to 9 refresh periods (9 × 7.8us) → we can actually wait for 64ms + 9x7.8us, because we can burst 8 at a time

Deferred Refresh

- MC may defer refresh if reads or high priority writes are pending



- Writes are high priority if the write buffer occupancy exceeds a threshold

- After the refresh has been deferred for some time period, it becomes the highest priority cmd
  - May have to perform back-to-back refreshes

[13.8]

Burst Refresh

- Each auto refresh must be preceded by a precharge ALL (across all the banks)

  ⇒ the data in the rows before need to be written back (close all those rows)

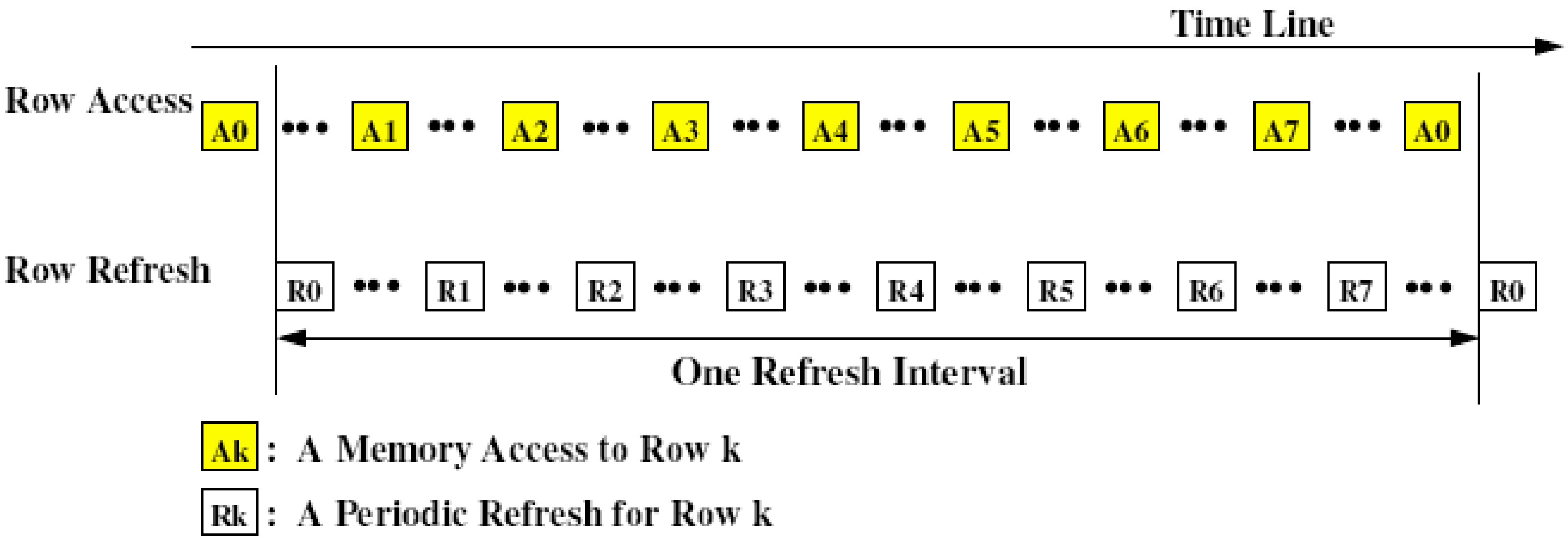
- SDRAM permits burst refreshes (up to eight) without an intervening precharge (don't need to setup the sense amp)
  - Still need t_{RFC} time between refreshes
  - Amortizes MC costs, e.g., cmd arbitration, over multiple refreshes
  - If too many refreshes then potential long read/write wait times
  - Dynamic burst length with information from the core?

    ⇒ change the refresh burst length depending what the core is doing

Free Refreshes

- With RAS-only refresh, MC keeps the next address to be refreshed (has the register)

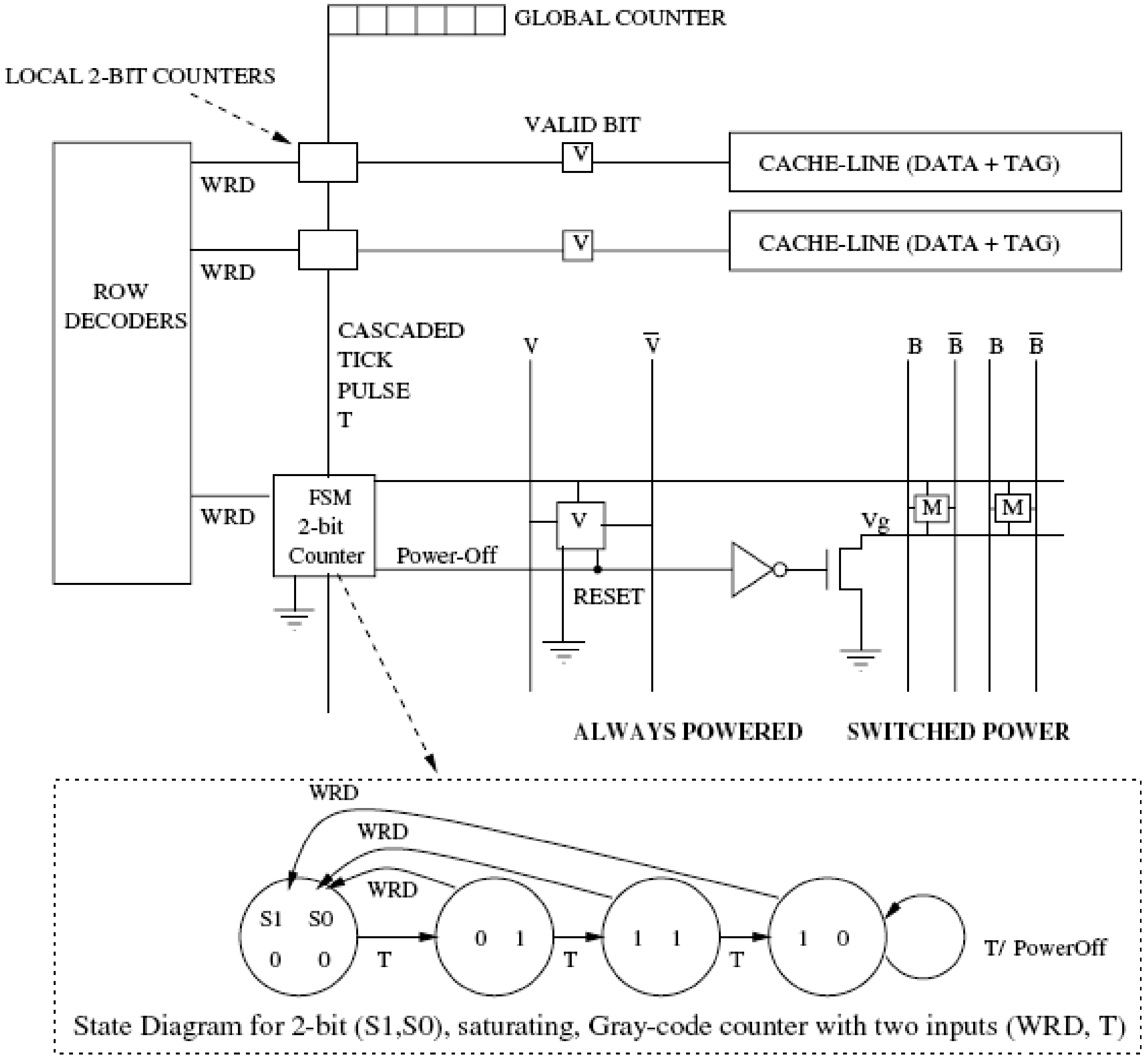
- If a read or write is scheduled to the same address as the next refresh, refresh is “free”



[Ghosh07]

Free Refreshes Using Decay

- **Recall cache decay**



State Diagram for 2-bit (S1,S0), saturating, Gray-code counter with two inputs (WRD, T)

[Kaxiras01]

Free Refreshes Using Decay

- Decay counter for every row
  - Need 768KB for 32GB memory

let the counter
decide when to refresh

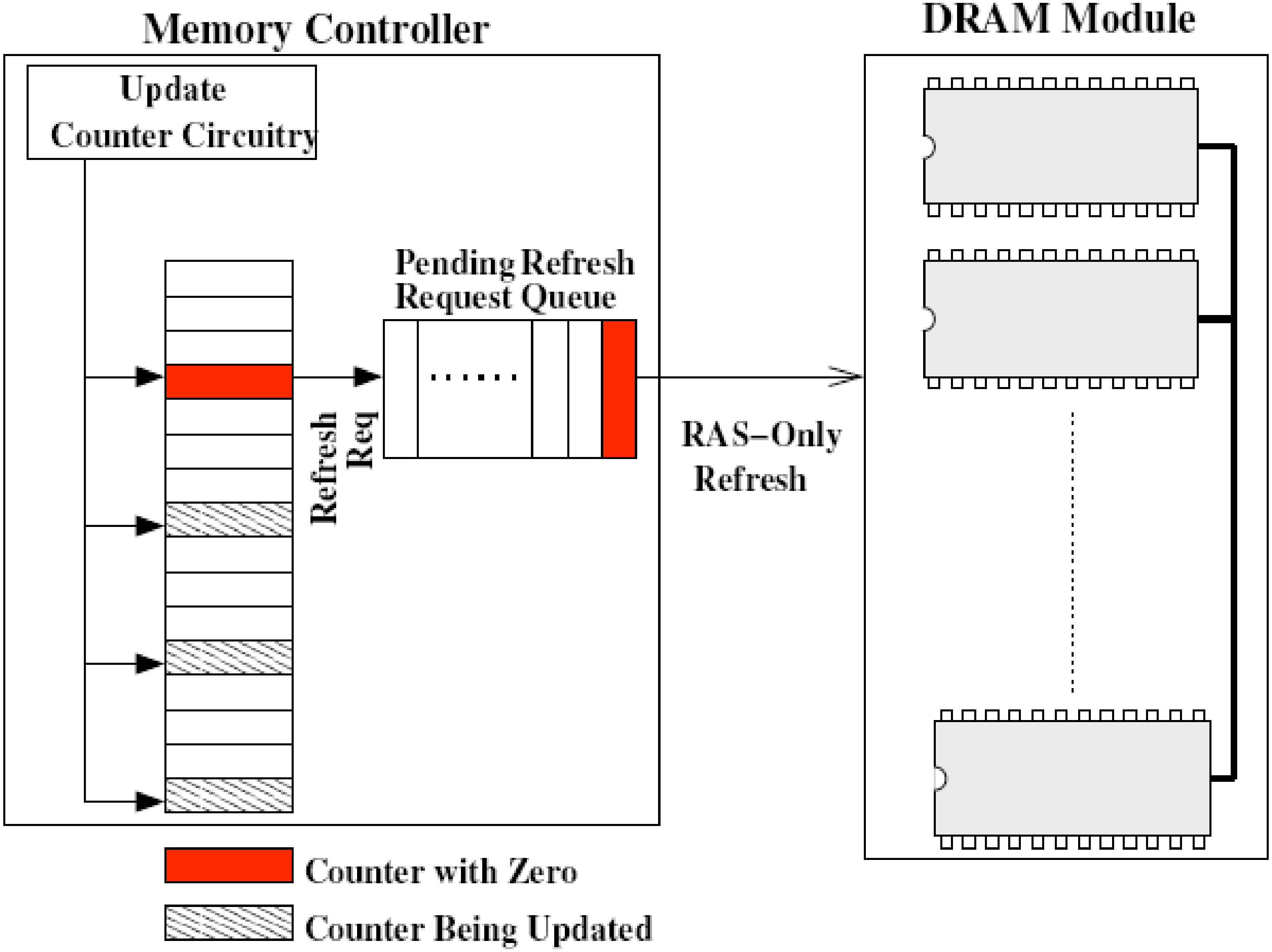
- Counter is set to max value if row is accessed

- Row is refreshed if counter = 0

- Counter decrements are staggered to avoid too many simultaneous refreshes

- Overhead too high if low activity (gate off counter circuitry and do normal refresh)

Free Refreshes Using Decay



[Ghosh07]

SDRAM Power Modes

- In self refresh (SR) mode, SDRAM current is lowered dramatically
  - 200mA read, 75mA idle → 10mA self refresh
- Power-down (PD) mode
  - Does not perform refresh
  - Can be exited more quickly

let data die

(keep the DLL running)

- Also might have

delay lock loop

  - Option to keep DLL running or not → do we kill the clock too?
  - Temperature-compensated self refresh (TCSR)

→ temp changes leakage rate

  - Partial-array self refresh (PASR)

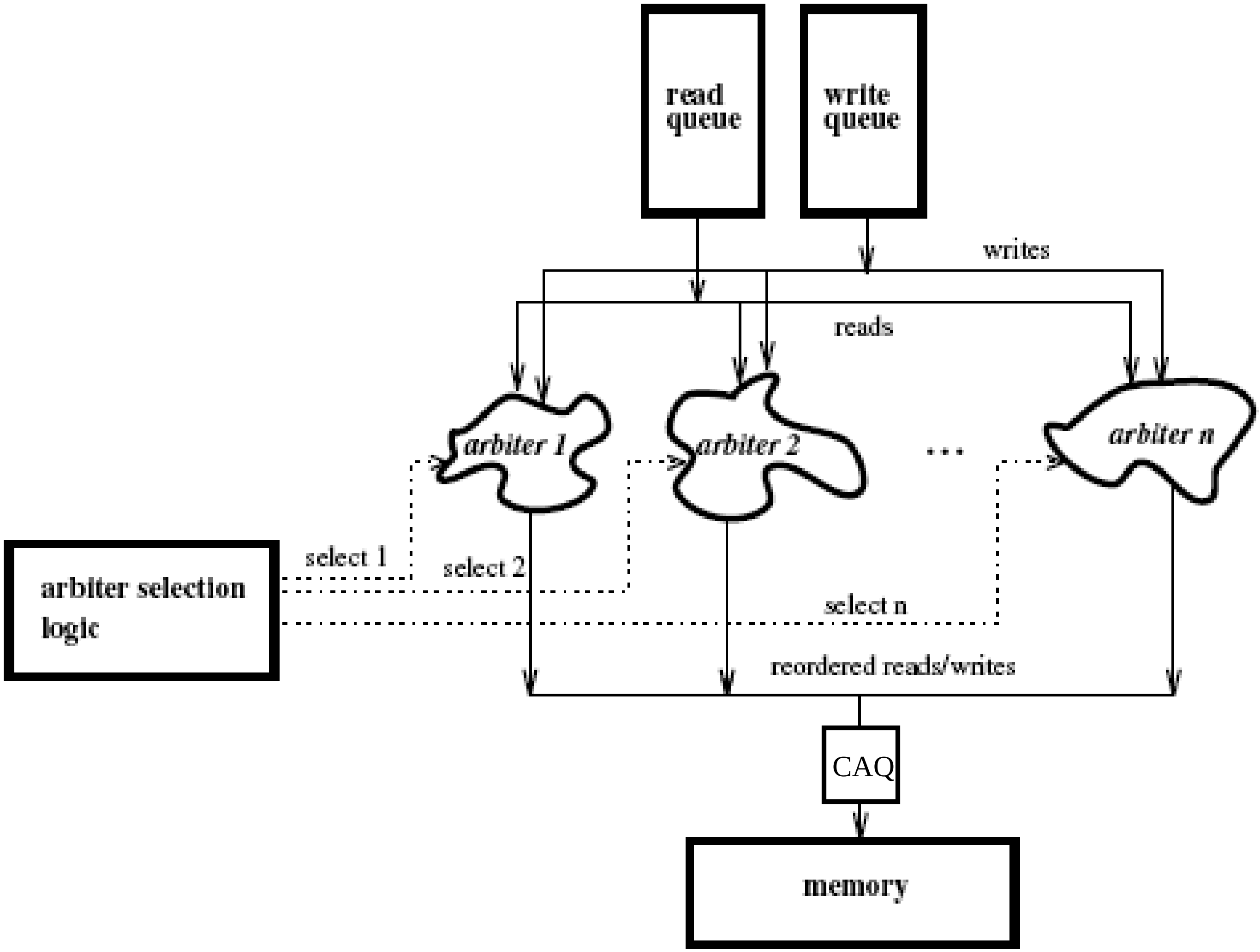
→ only do it for part of the DRAM

Potential Performance Costs

- Commands to enter/exit the mode (2 cycles)
- Precharge ALL before SR (10 cycles)
- Minimum time in SR or PD mode (5 cycles)
- Exiting SR or PD to perform a read or write
  - SR = 512 cycles (relock DLL) we turned off CKE and killed the clock
  - PD = 20 cycles
- Memory power manager must carefully balance saving power with potential performance losses

Making the MC Power-Aware

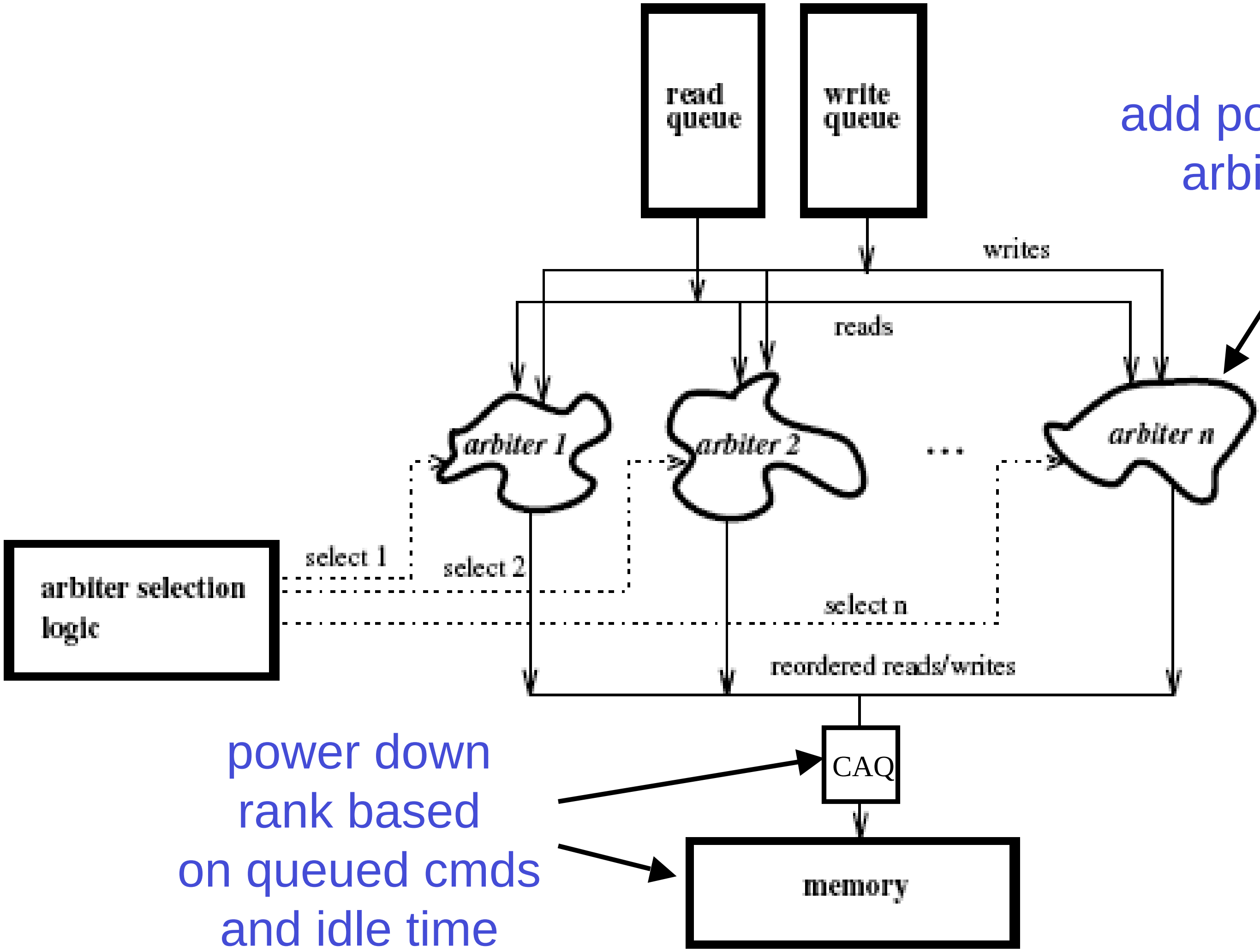
- Recall adaptive history-based scheduling



[Hur04]

Making the MC Power-Aware

- Power-aware adaptive history-based scheduling



add power-aware arbiter FSM

power down rank based on queued cmds and idle time

[Hur04]

Rank Power-Down/Up Algorithm

- Counter for each rank, initialized and decremented each cycle

- Read/write to a rank sets counter to the max of the current value and the cmd latency

max(current val, cmd latency)

↑ don't power down before cmd is done

adds 1 cycle of buffer for another command

- If rank counter = 0, rank is powered down if
  - No cmds in CAQ to this rank
  - No cmd in the CAQ can be issued this cycle

- Rank is powered up and counter reinitialized when cmd for this rank enters the CAQ

↙ centralized arbiter queue

Power-Aware Arbiter FSM

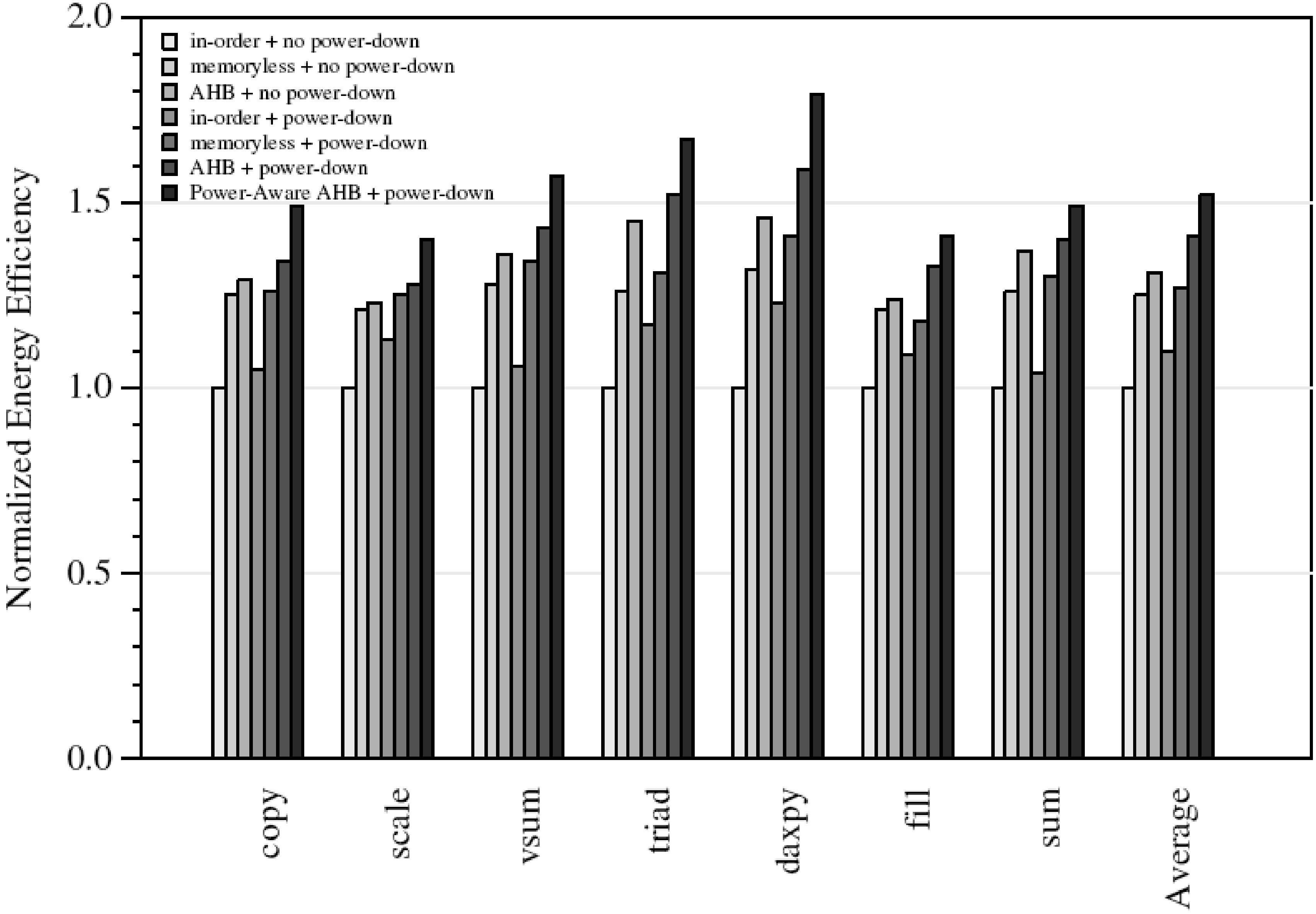
- Attempts to cluster (in time) all operations on a rank to increase opportunities for power down

- Highest priority is cmd to same rank as the last cmd, then same rank as the 2nd to last, etc.

- Power-aware FSM given same weight as command pattern and expected latency arbiters

(random)

Energy Efficiency Improvements



[Hur08]

Power-Aware Page Allocation

- The OS performs on-demand allocation of physical pages in memory, typically randomly

- *Power-aware page allocation*: OS allocates and migrates pages in a way that increases usage of power-down modes by the MC
  - Cluster page allocations within the same rank

→ turn the other ones off

→ lose parallelism, more contention

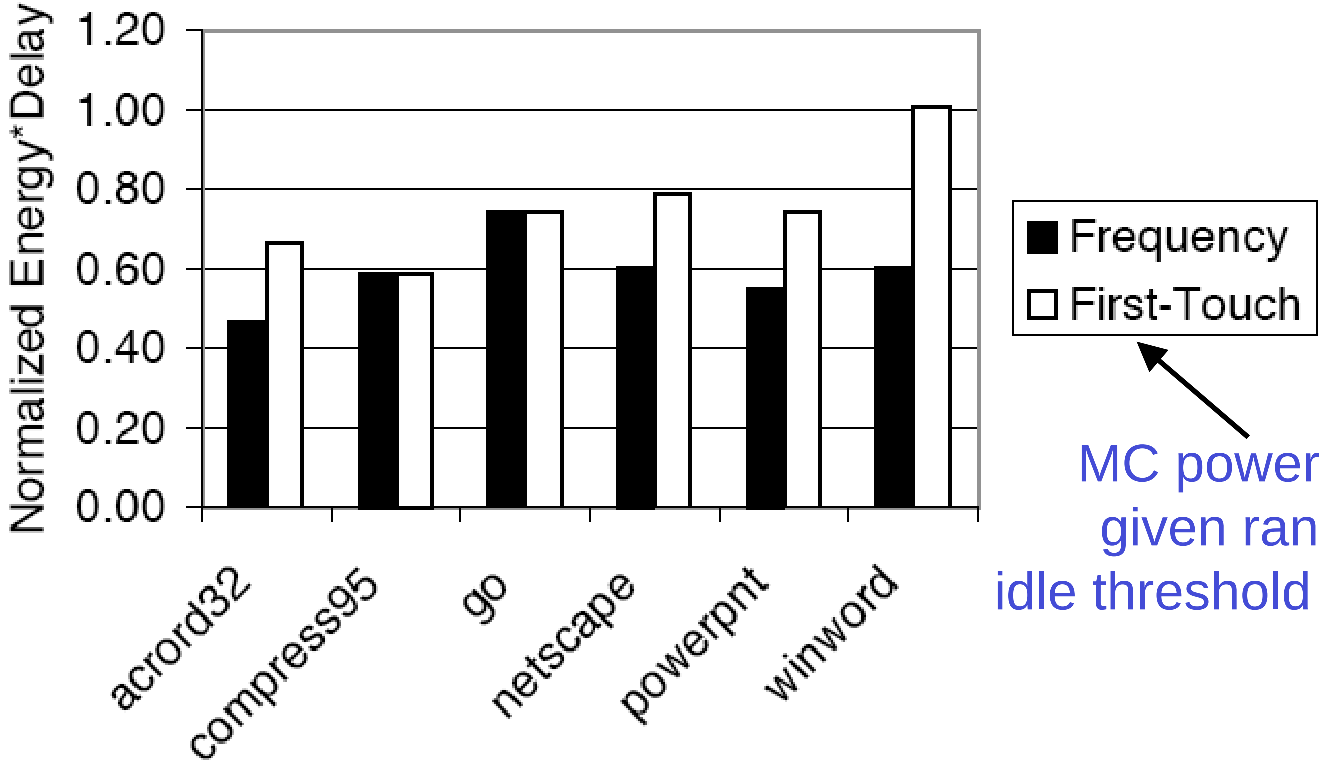
Power-Aware Page Allocation

- Sequential first-touch page allocation
  - Physical pages are sequentially allocated in the order they are accessed
  - Tends to group temporally accessed pages in same rank

- Frequency migration policy
  - Per-page hardware counters that count frequency of access to each page
  - A limited number of the most frequently accessed pages are clustered to the same rank
  - Added migration cost for some of the pages

Energy × Delay Results

- **Baseline**
  - **MC powers down all ranks only when no accesses**
  - **Sequential first-touch page allocation**



MC powers down
given rank after
idle threshold exceeded

[Lebeck00]

Next Time

Case Studies