RESISTIVE MEMORY TECHNOLOGY

Mahdi Nazm Bojnordi

Assistant Professor

School of Computing

University of Utah

THE UNIVERSITY OF UTAH

Overview

- Upcoming deadlines
  - April 7th: Sign up for your student paper presentation

- This lecture
  - Resistive memory technology
  - Write optimization techniques
  - Wear leveling
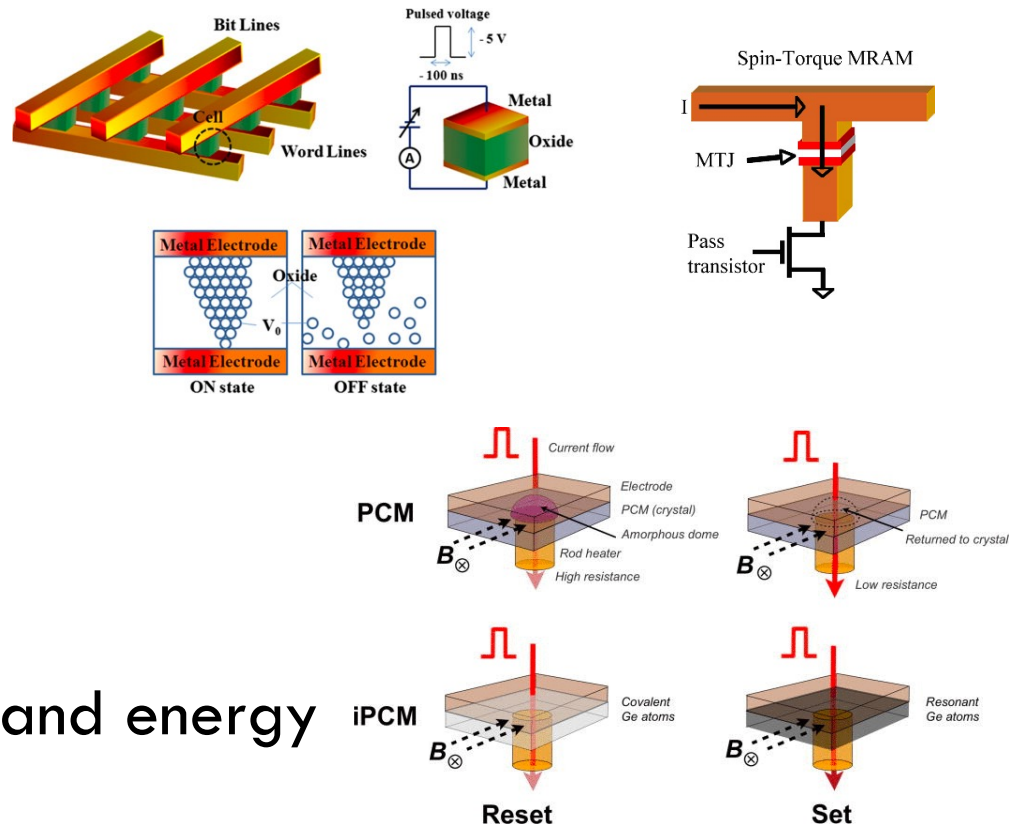 - MLC technologies

Resistive Memory Technology

- ☐ Main benefits
  - ◘ Non-volatile memory
  - ◘ Multi-level storage
  - ◘ Denser cells
  - ◘ Better scalability

- ☐ Shortcomings
  - ◘ Limited endurance
  - ◘ High switching delay and energy
  - **What can we do?**

Comparison of Technologies

☐ Compared to NAND Flash, PCM is byte-addressable, has orders of magnitude lower latency and higher endurance.

	DRAM	PCM	NAND Flash
Page size	64B	64B	4KB
Page read latency	20-50ns	~ 50ns	~ 25 µs
Page write latency	20-50ns	~ 1 µs	~ 500 µs
Write bandwidth	~GB/s per die	50-100 MB/s per die	5-40 MB/s per die
Erase latency	N/A	N/A	~ 2 ms
Endurance	∞	$10^6 - 10^8$	$10^4 - 10^5$
Read energy	0.8 J/GB	1 J/GB	1.5 J/GB [28]
Write energy	1.2 J/GB	6 J/GB	17.5 J/GB [28]
Idle power	~100 mW/GB	~1 mW/GB	1–10 mW/GB
Density	1×	2 – 4×	4×

Sources: [Doller '09] [Lee et al. '09] [Qureshi et al. '09]

Comparison of Technologies

- Compared to DRAM, PCM has better density and scalability and similar read but longer write latencies

	DRAM	PCM	NAND Flash
Page size	64B	64B	4KB
Page read latency	20-50ns	~ 50ns	~ 25 µs
Page write latency	20-50ns	~ 1 µs	~ 500 µs
Write bandwidth	~GB/s per die	50-100 MB/s per die	5-40 MB/s per die
Erase latency	N/A	N/A	~ 2 ms
Endurance	∞	$10^6 - 10^8$	$10^4 - 10^5$
Read energy	0.8 J/GB	1 J/GB	1.5 J/GB [28]
Write energy	1.2 J/GB	6 J/GB	17.5 J/GB [28]
Idle power	~100 mW/GB	~1 mW/GB	1–10 mW/GB
Density	1×	2 – 4×	4×

Sources: [Doller '09] [Lee et al. '09] [Qureshi et al. '09]
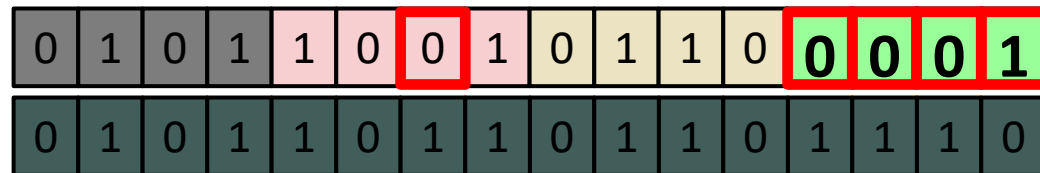
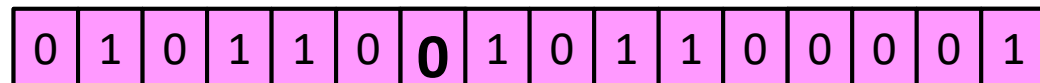# Latency Comparison
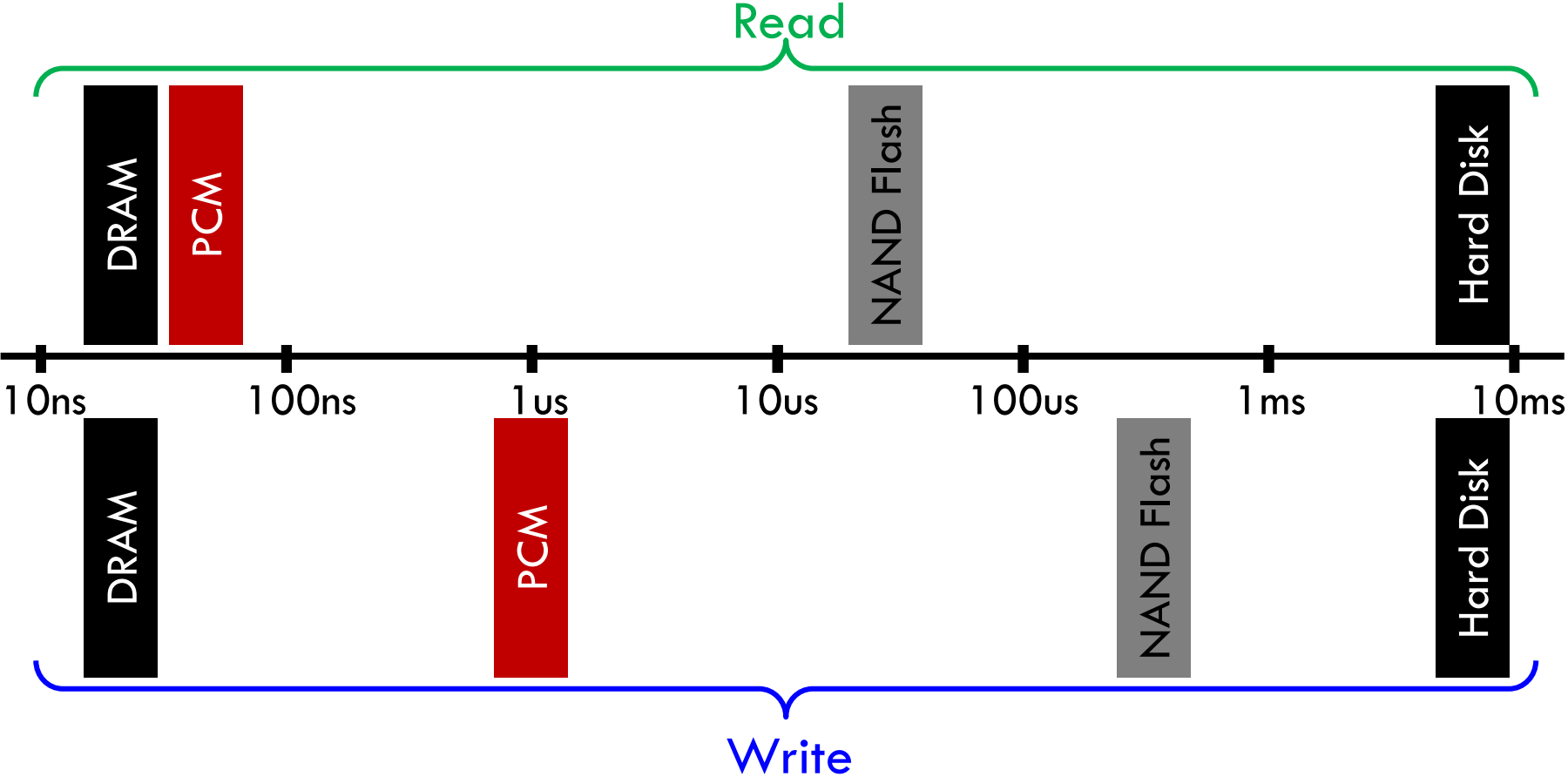
[Qureshi'09]

Read Compare Write

- A cache line is written in several cycles

- Read-compare-write (differential write)
  - Write only modified bits rather than entire cache line

- Skipping parts with no modified bits

Cache line

| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | **0** | **0** | **0** | **1** |

| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |

PCM

| 0 | 1 | 0 | 1 | 1 | 0 | **0** | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |

Reducing Bit Flips

☐ Encode write data into either its regular or inverted form and then pick the encoding that yields in less flips in comparison against old data.

Flip-N-Write [MICRO'09]

Old	0	0	1	0	1	1

New(Regular)	0	1	0	1	0	1

New (Inverted)	1	0	1	0	1	0

Saves 4 bit flips

☐ Encode write data into a set of data vectors and then pick the vector that yields in less flips in comparison against old data.

Flip-Min [HPCA'13]

Old	0	0	1	0	1	1

New_1	0	1	0	1	0	1

New_2	1	0	1	0	1	0

New_3	1	0	1	0	1	1

Saves 5 bit flips

Limited Lifetime
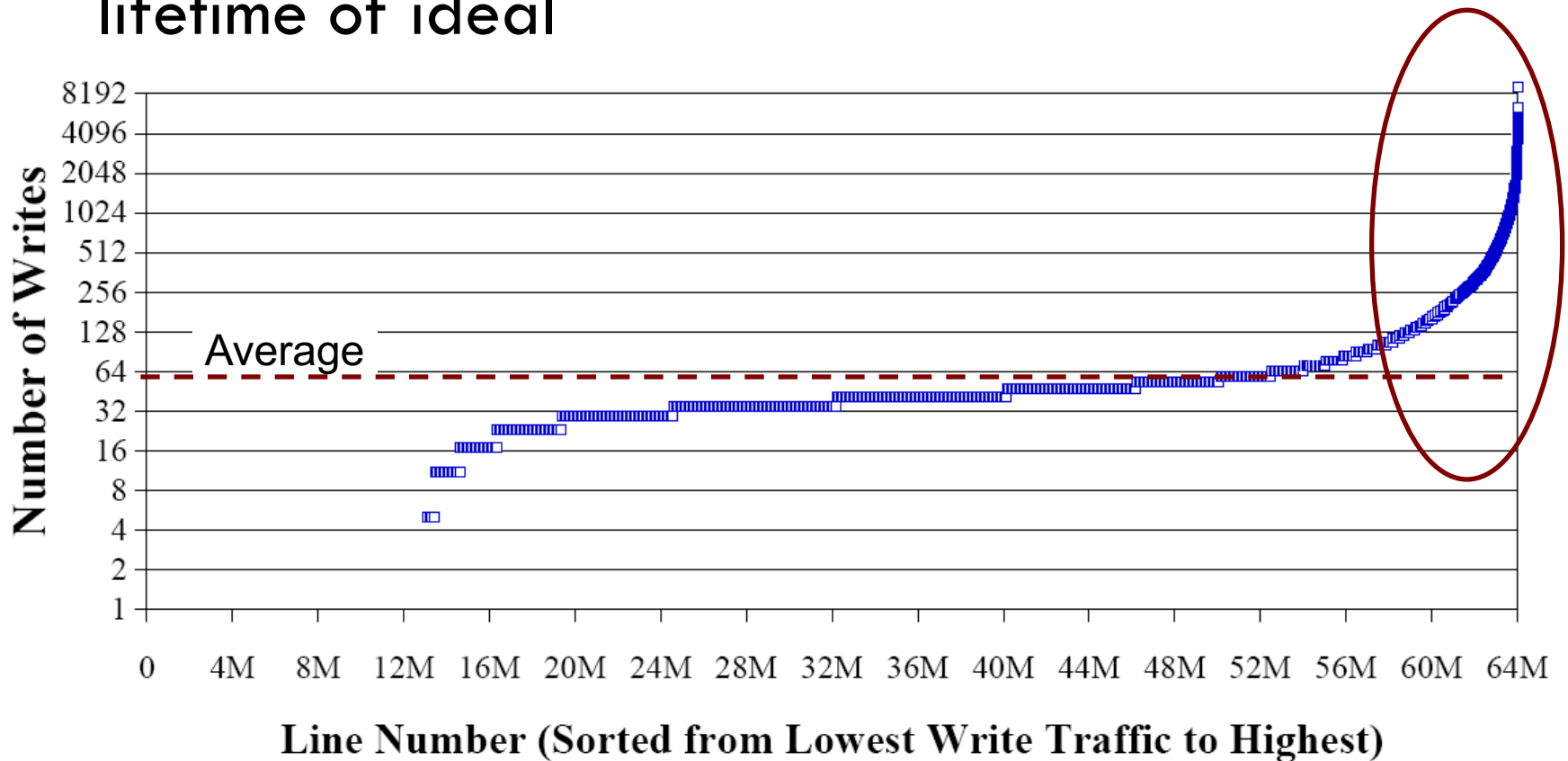
Challenge : Each cell can endure 10-100 Million writes
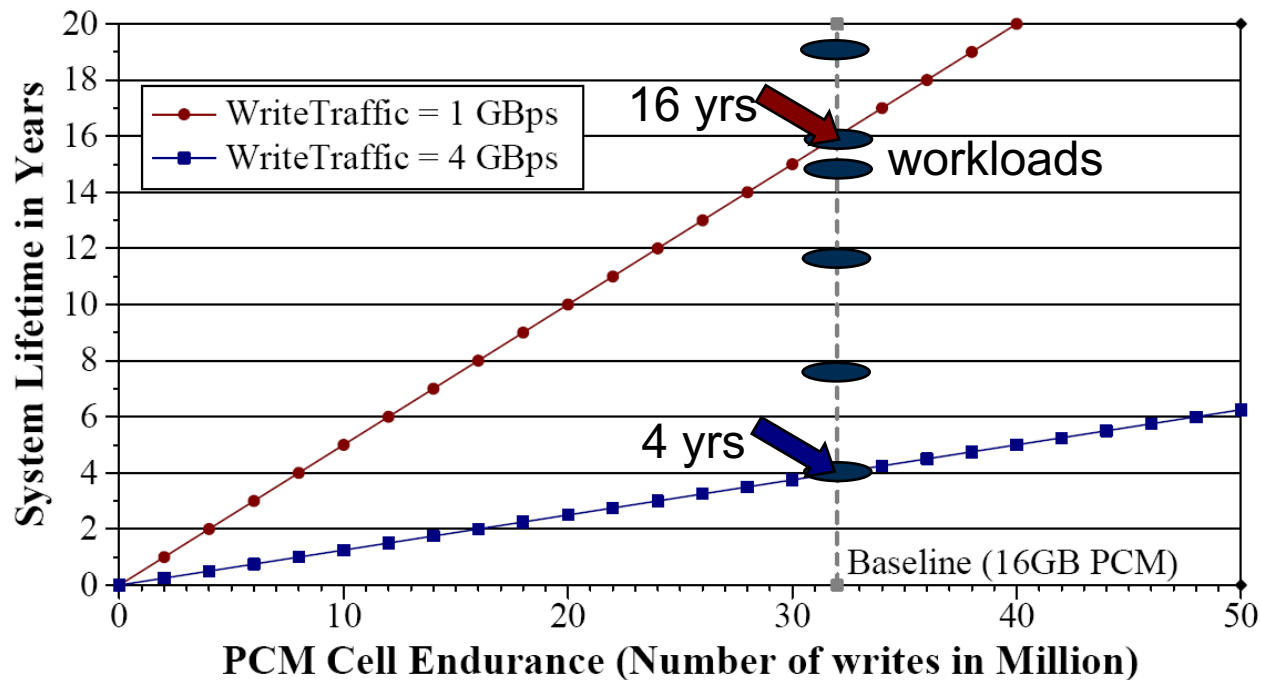
With uniform write traffic, system lifetime ranges from 4-20 years

[Qureshi'09]

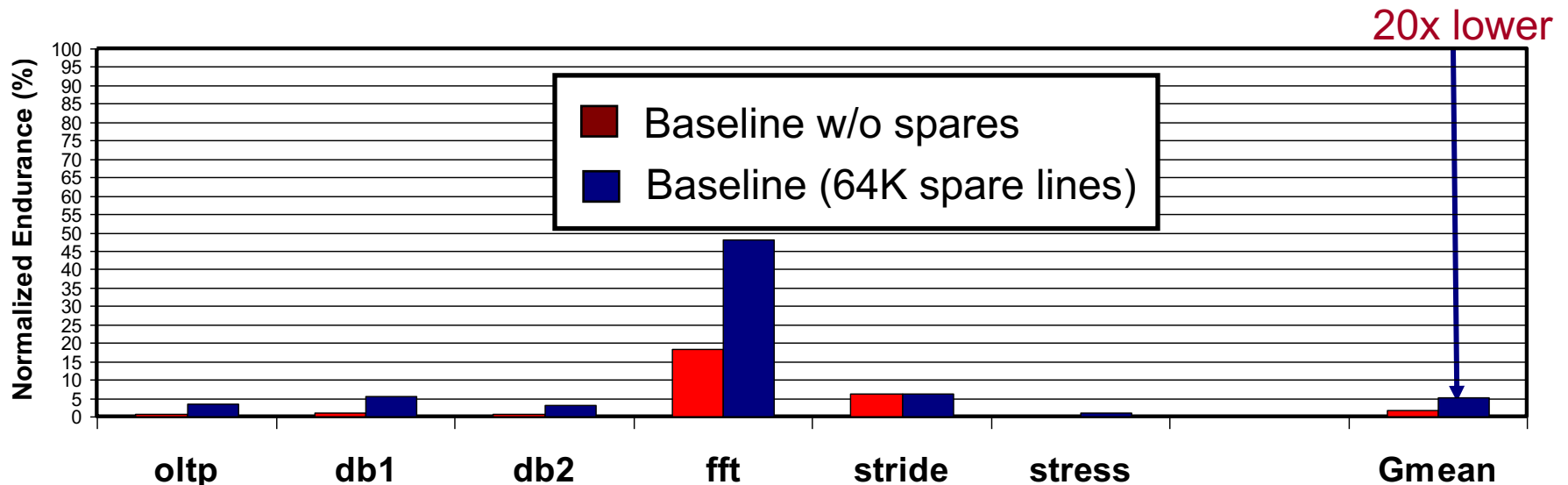# Non-Uniform Writes

- Even with 64K spare lines, baseline gets 5% lifetime of ideal

[Qureshi'09]

Impact of Non-Uniformity
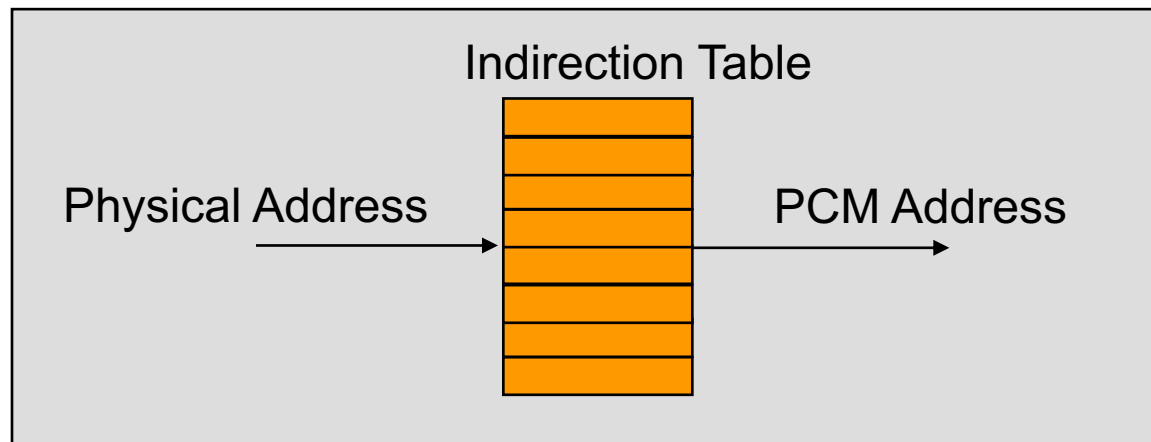
☐ Even with 64K spare lines, baseline gets 5% lifetime of ideal

$$\text{Norm. Endurance} = \frac{\text{Num. writes before system failure}}{\text{Num. writes before failure with uniform writes}} \times 100\%$$

20x lower

Baseline w/o spares
Baseline (64K spare lines)

Normalized Endurance (%)

oltp   db1   db2   fft   stride   stress   Gmean

[Qureshi'09]

Making Writes Uniform

☐ Wear Leveling: make writes uniform by remapping frequently written lines

Line Addr.	Lifetime Count	Period Count
A	99K (Low)	1K (Low)
B	100K (Med)	3K (High)
C	101K (High)	2K (Med)

➔

Line	Remap Addr
A	C
B	A
C	B

Indirection Table

Physical Address ⟶ PCM Address

[Qureshi'09]

How to Remap

□ Tables

  ◘ Area of several (tens of) megabytes

  ◘ Indirection latency (table in EDRAM/DRAM)

□ Area overhead can be reduced with more lines per region

  ◘ Reduced effectiveness (e.g. Line0 always written)

  ◘ Support for swapping large memory regions (complex)

[Qureshi'09]

Start-Gap Wear Leveling

☐ Two registers (Start & Gap) + 1 line (GapLine) to support movement

☐ Move GapLine every 100 writes to memory.
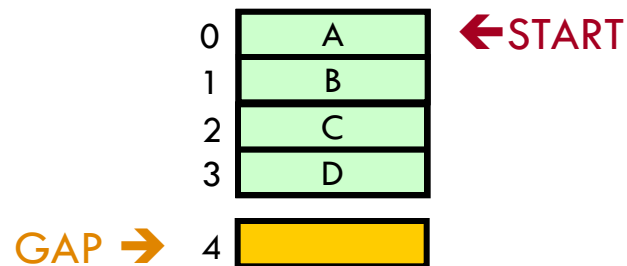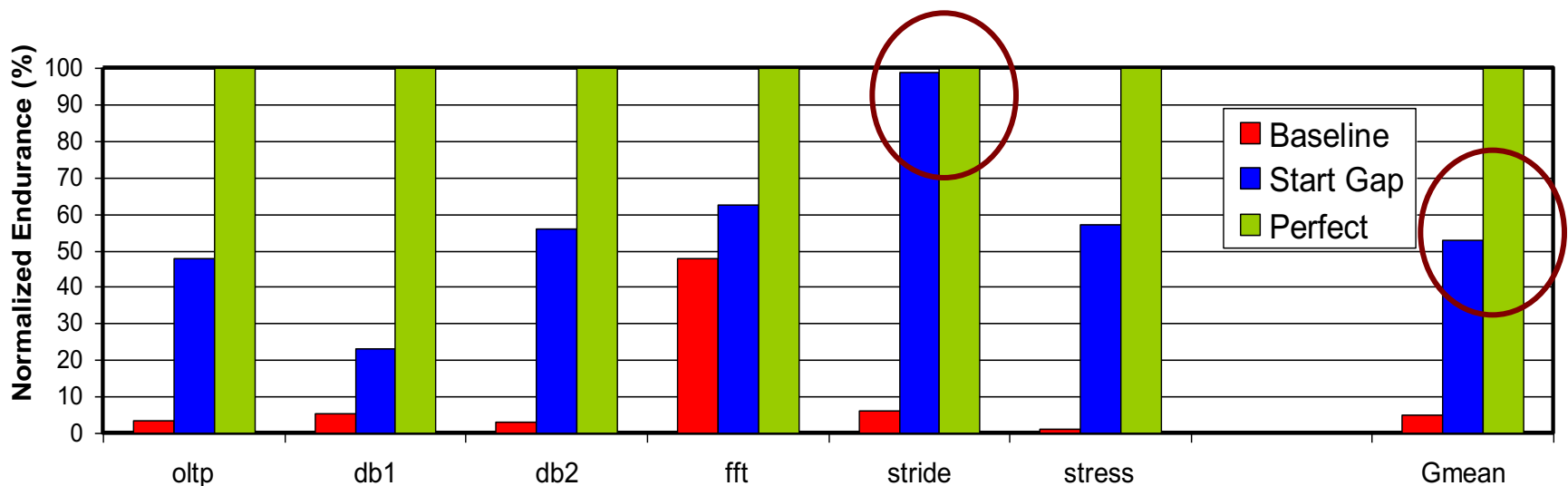

PCMAddr = (Start+Addr); (PCMAddr >= Gap) PCMAddr++)

Storage overhead: less than 8 bytes (GapLine taken from spares)
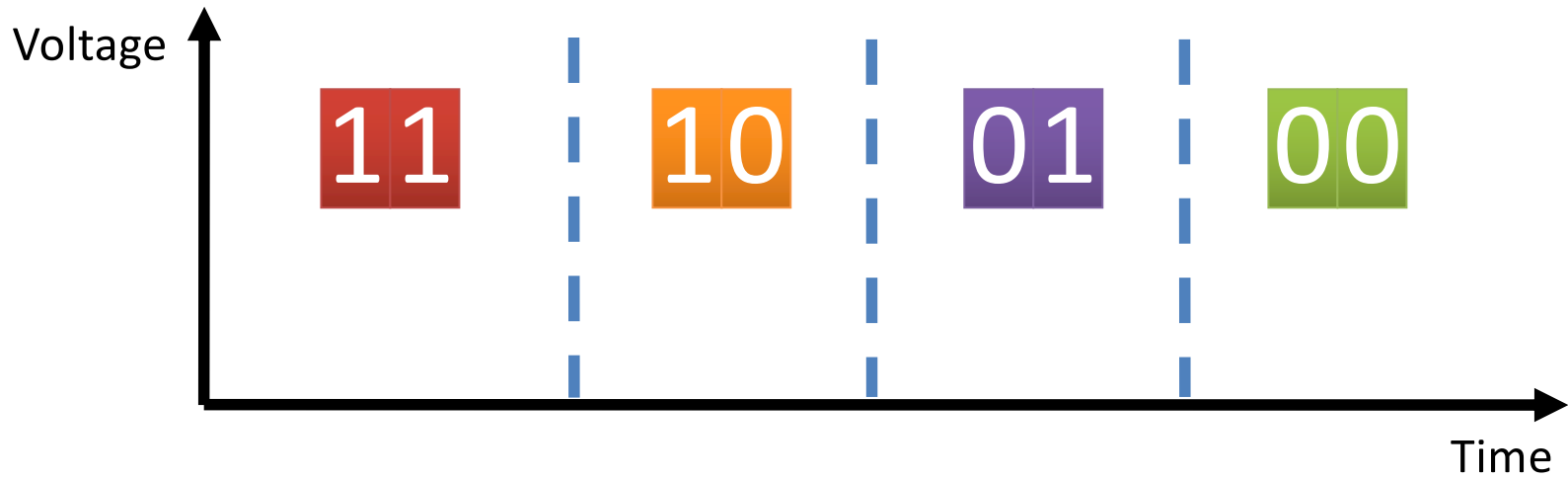Latency: Two additions (no table lookup)
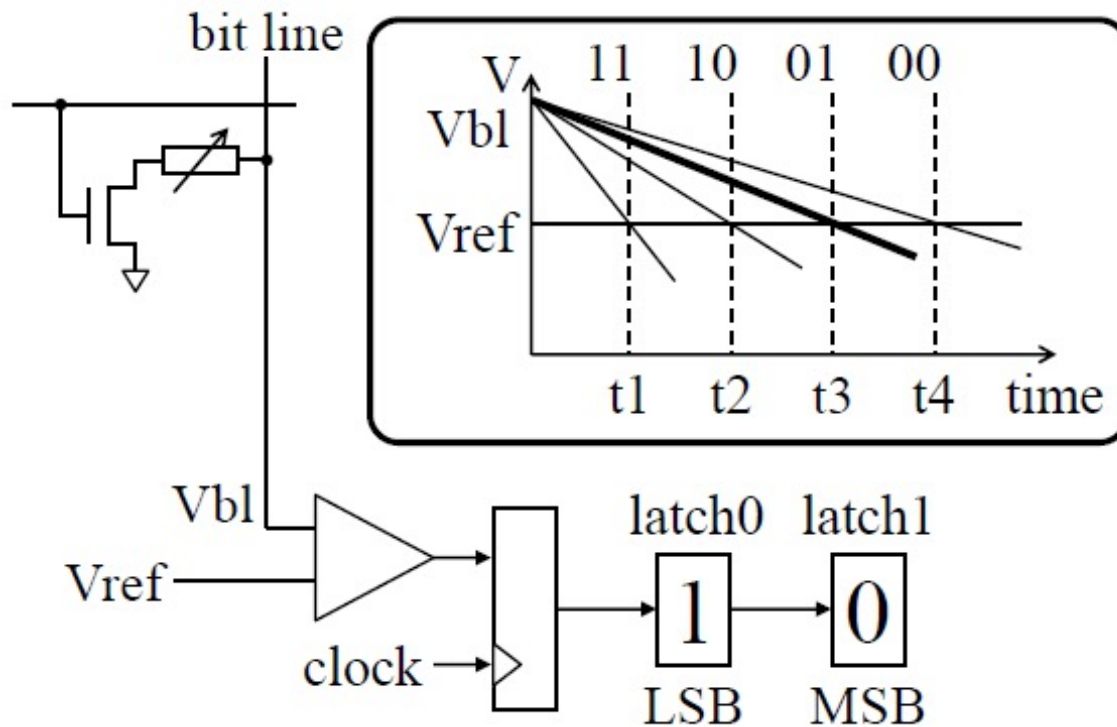Write overhead: One extra write every 100 writes ➔ 1%

[Qureshi'09]

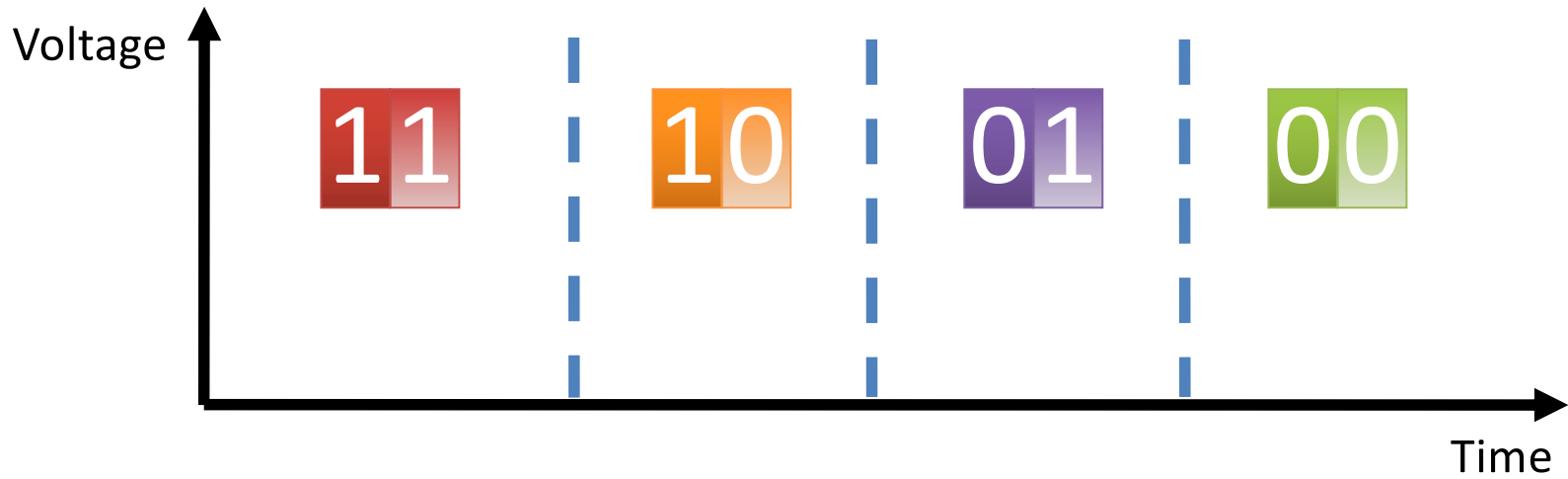# Start-Gap Results

☐ On average, Start-Gap gets 53% normalized endurance

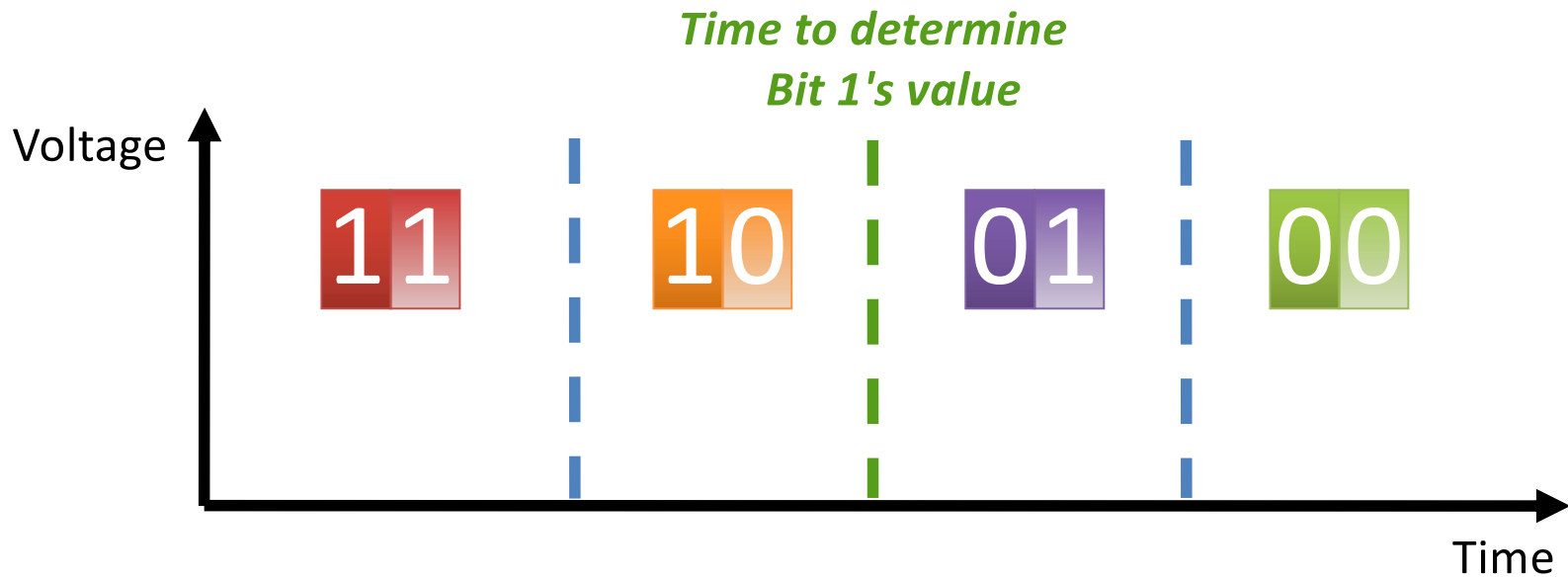
[Qureshi'09]

Multi-Level Cells

[Yoon'14]

Sensing Multi-level Cells

[Yoon'14]

Multi-Level Cells

[Yoon'14]

Multi-Level Cells

[Yoon'14]

Multi-Level Cells

[Yoon'14]

Decoupled Bit Mapping

MLC-PCM cell

Bit 1 (fast read) →
Bit 0 (fast write) →
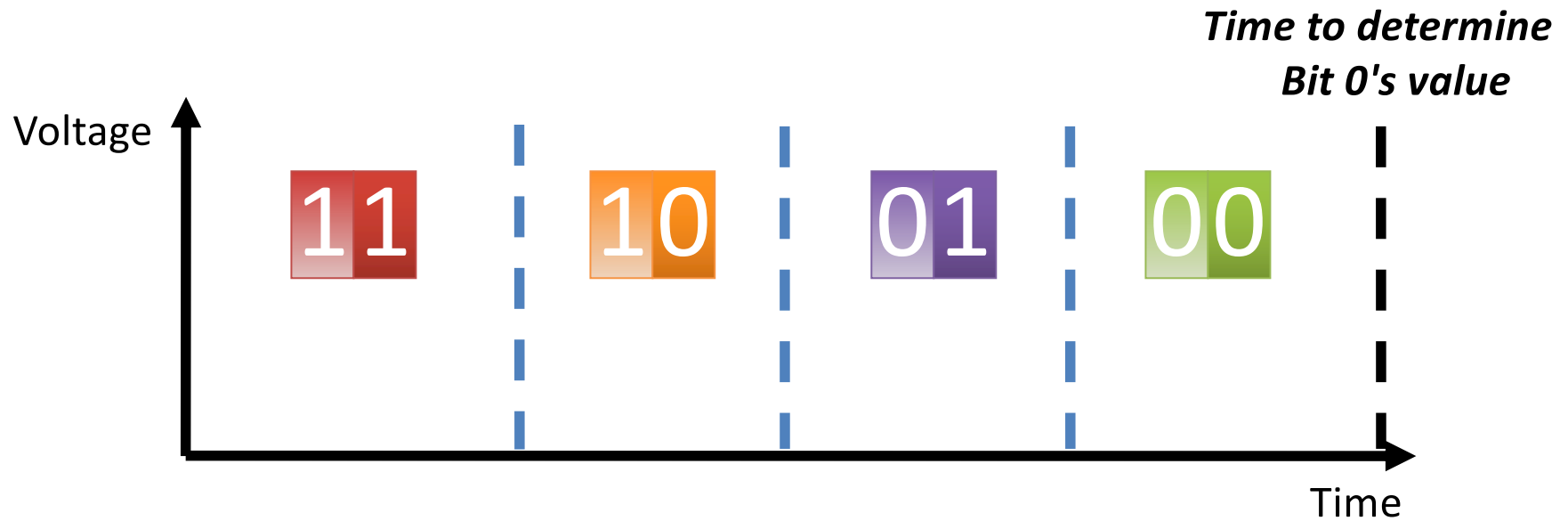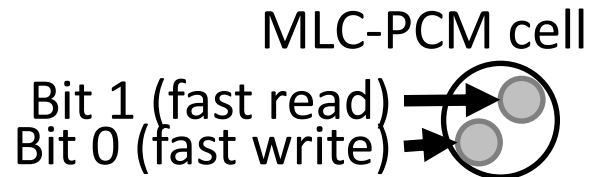
Coupled (baseline): Contiguous ***bits*** alternate between FR and FW
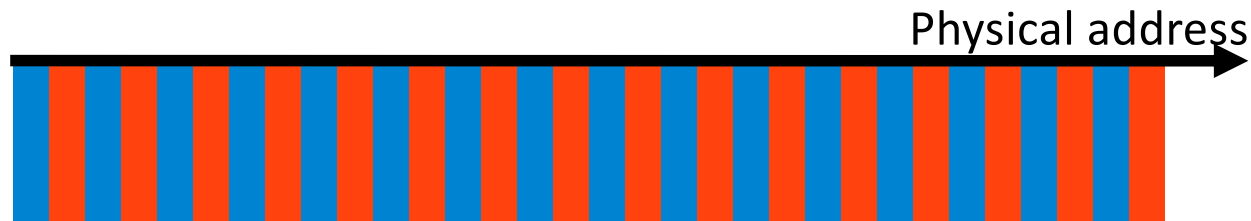

Decoupled: Contiguous ***regions*** alternate between FR and FW

[Yoon'14]

Decoupled Bit Mapping

- By decoupling, we've created regions with distinct characteristics
  - We examine the use of 4KB regions (e.g., OS page size)

Physical address

Fast read page **Fast write page**

- Want to match *frequently read data to FR pages* and vice versa
- Toward this end, we propose a new **OS page allocation** scheme

[Yoon'14]

Performance Results

Normalized Speedup

Legend:
- Conventional
- All fast write
- All fast read
- DBM
- DBM+APM+SRB
- Ideal

Bar labels: +10%, +16%, +13%, +19%, +31%

[Yoon'14]