Memory Scaling is Dead, Long Live Memory Scaling

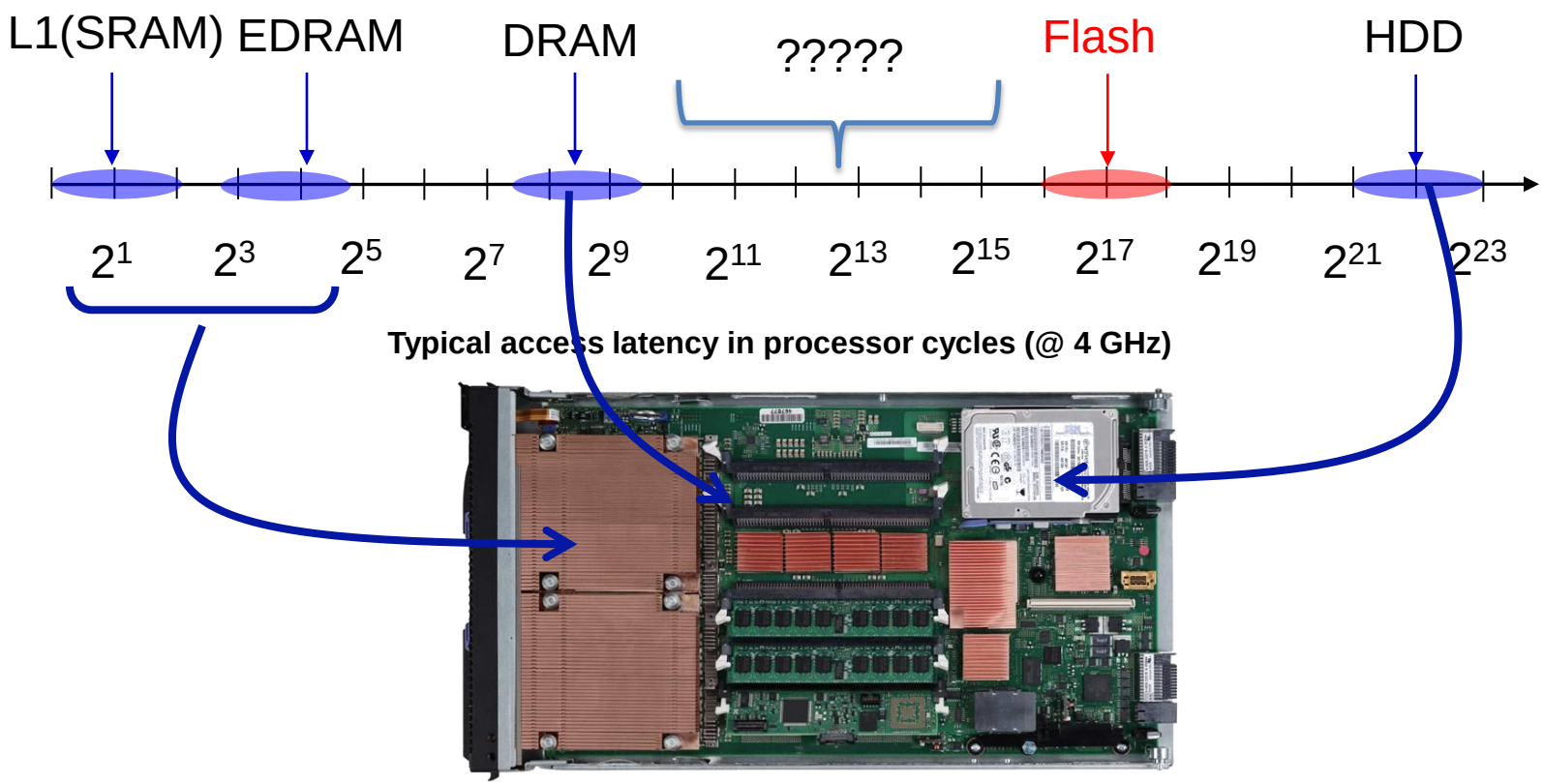
Le Memoire Scaling est mort, vive le Memoire Scaling!

Moinuddin K. Qureshi

ECE, Georgia Tech

At Yale's "Mid Career" Celebration at University of Texas at Austin, Sept 19 2014

The Gap in Memory Hierarchy

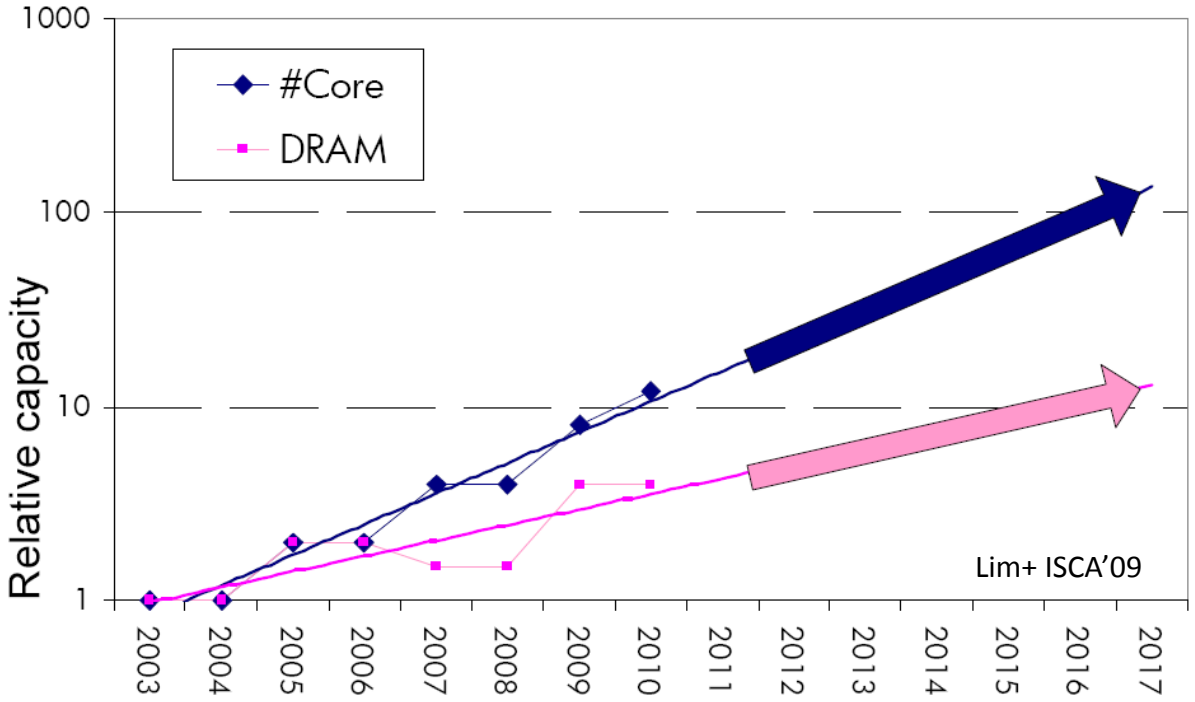


Misses in main memory (page faults) degrade performance severely

Main memory system must scale to maintain performance growth

The Memory Capacity Gap

Trends: Core count doubling every 2 years.
DRAM DIMM capacity doubling every 3 years

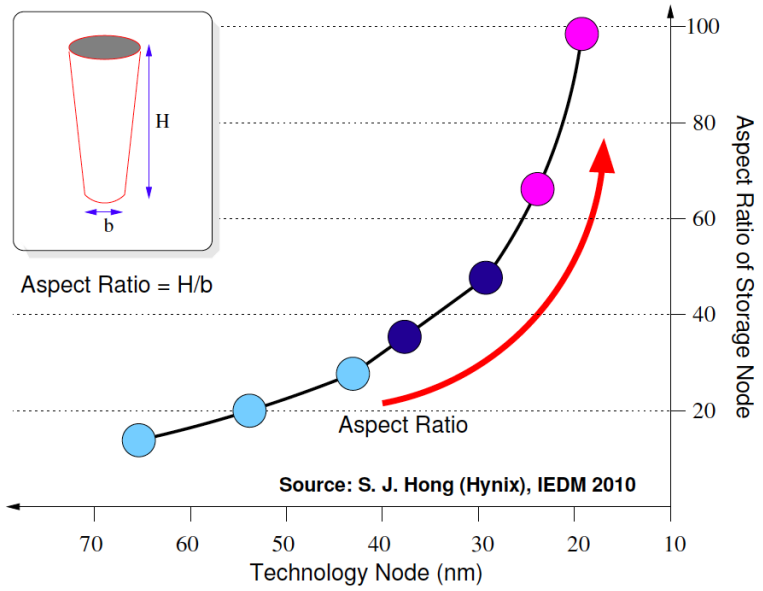


Memory capacity per core expected to drop by 30% every two years

Challenges for DRAM: Scaling Wall

Scaling wall

DRAM does not scale well to small feature sizes (sub 1x nm)



Increasing error rates can render DRAM scaling infeasible

Two Roads Diverged …

Architectural support for DRAM scaling and to reduce refresh overheads

Find alternative technology that avoids problems of DRAM

DRAM challenges

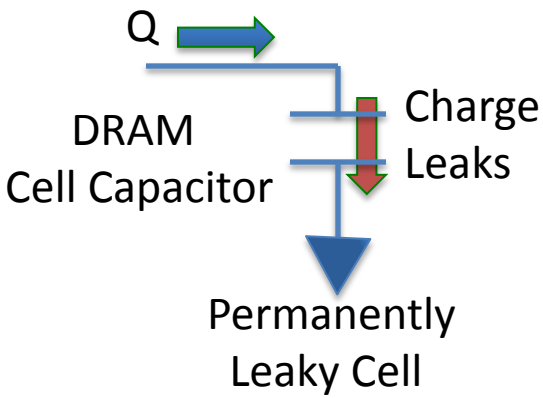
Important to investigate both approaches

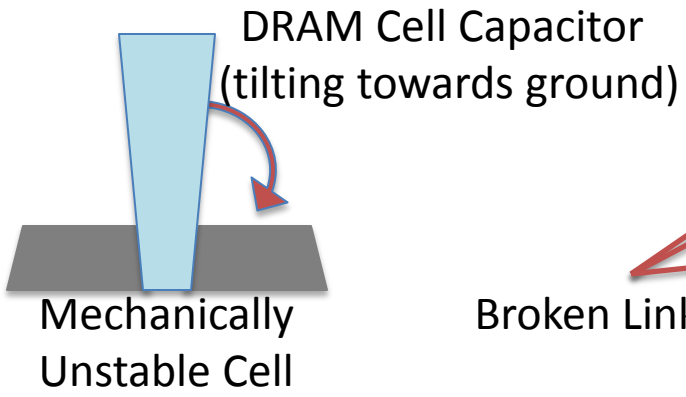
Outline

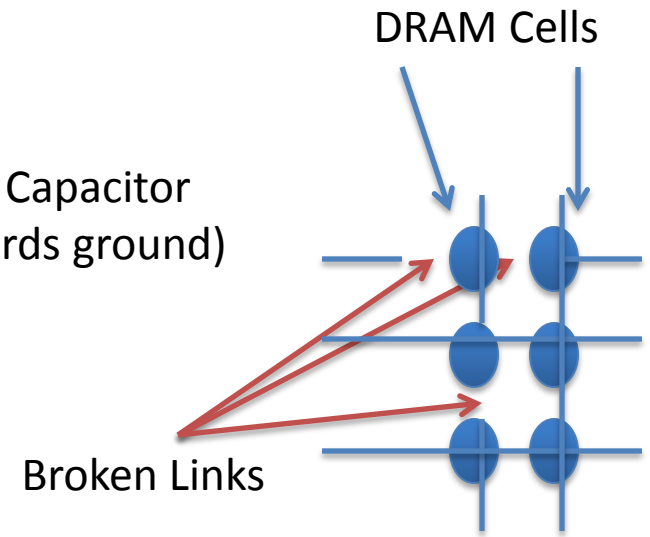
- Introduction
- ArchShiled: Yield Aware (arch support for DRAM)
- Hybrid Memory: reduce Latency, Energy, Power
- Adaptive Tuning of Systems to Workloads
- Summary

Reasons for DRAM Faults

- Unreliability of ultra-thin dielectric material
- In addition, DRAM cell failures also from:
  - Permanently leaky cells
  - Mechanically unstable cells
  - Broken links in the DRAM array



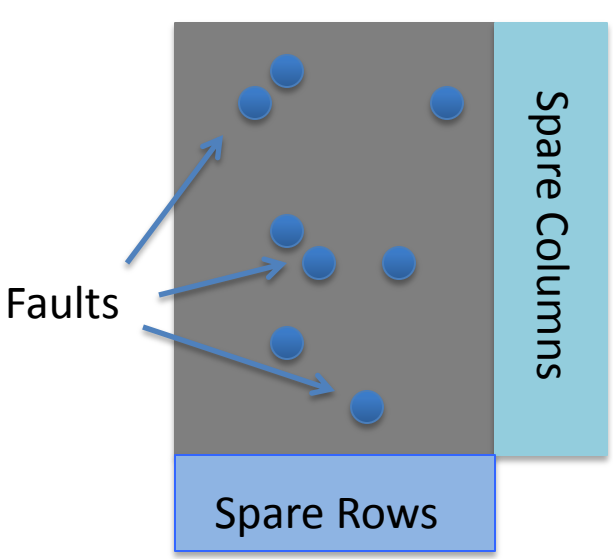




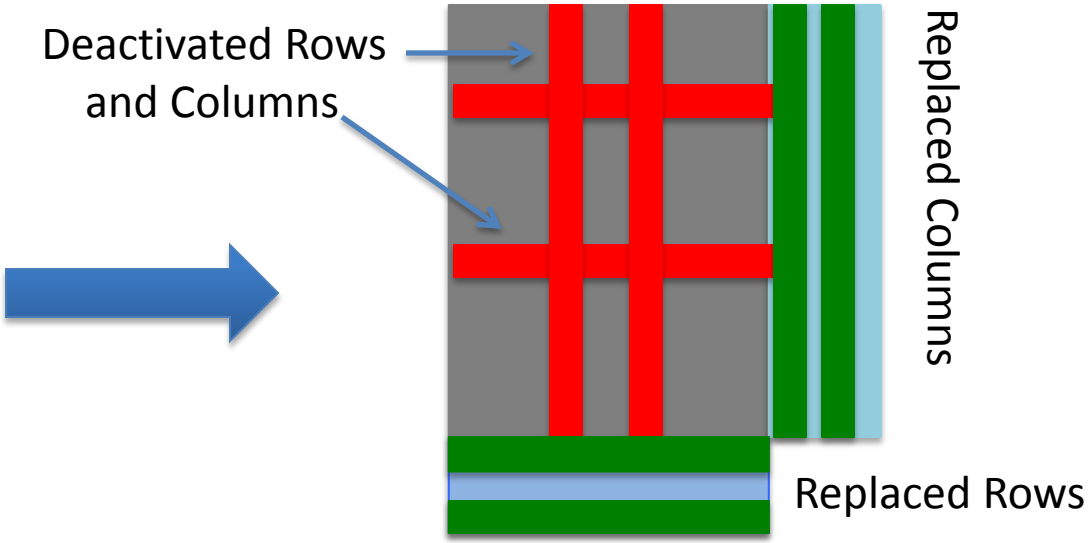
Permanent faults for future DRAMs expected to be much higher

Row and Column Sparing

- DRAM chip (organized into rows and columns) have spares



DRAM Chip:
Before Row/Column Sparing



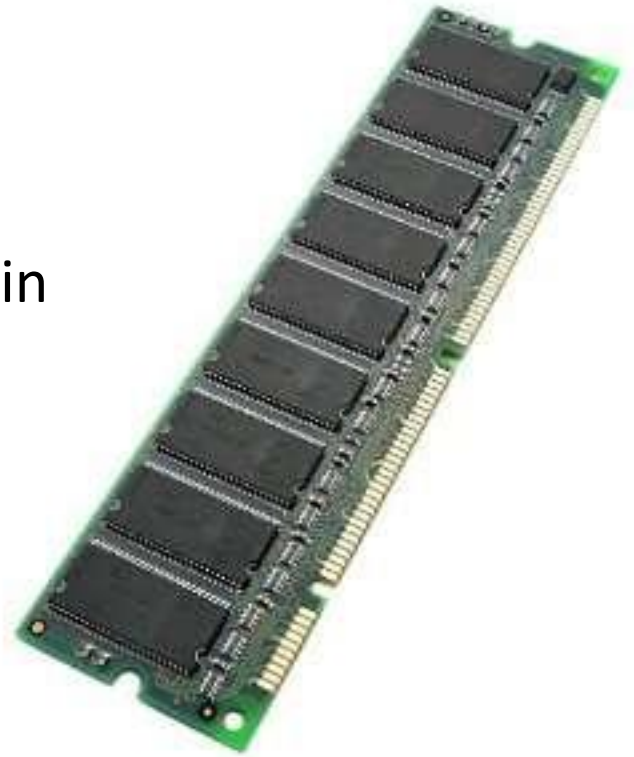
DRAM Chip:
After Row/Column Sparing

- Laser fuses enable spare rows/columns
- Entire row/column needs to be sacrificed for a few faulty cells

Row and Column Sparing incurs large area overheads

Commodity ECC-DIMM

- Commodity ECC DIMM with SECDED at 8 bytes (72,64)
- Mainly used for soft-error protection
- For hard errors, high chance of two errors in same word (birthday paradox)



For 8GB DIMM ➔ 1 billion words

Expected errors till double-error word

= 1.25*Sqrt(N) = 40K errors ➔ 0.5 ppm

SECDED not enough at high error-rate (what about soft-error?)

Dissecting Fault Probabilities

At Bit Error Rate of 10^{-4} (100ppm) for an 8GB DIMM (1 billion words)

Faulty Bits per word (8B)	Probability	Num words in 8GB
0	99.3%	0.99 Billion
1	0.7%	7.7 Million
2	26×10^{-6}	28 K
3	62×10^{-9}	67
4	10^{-10}	0.1

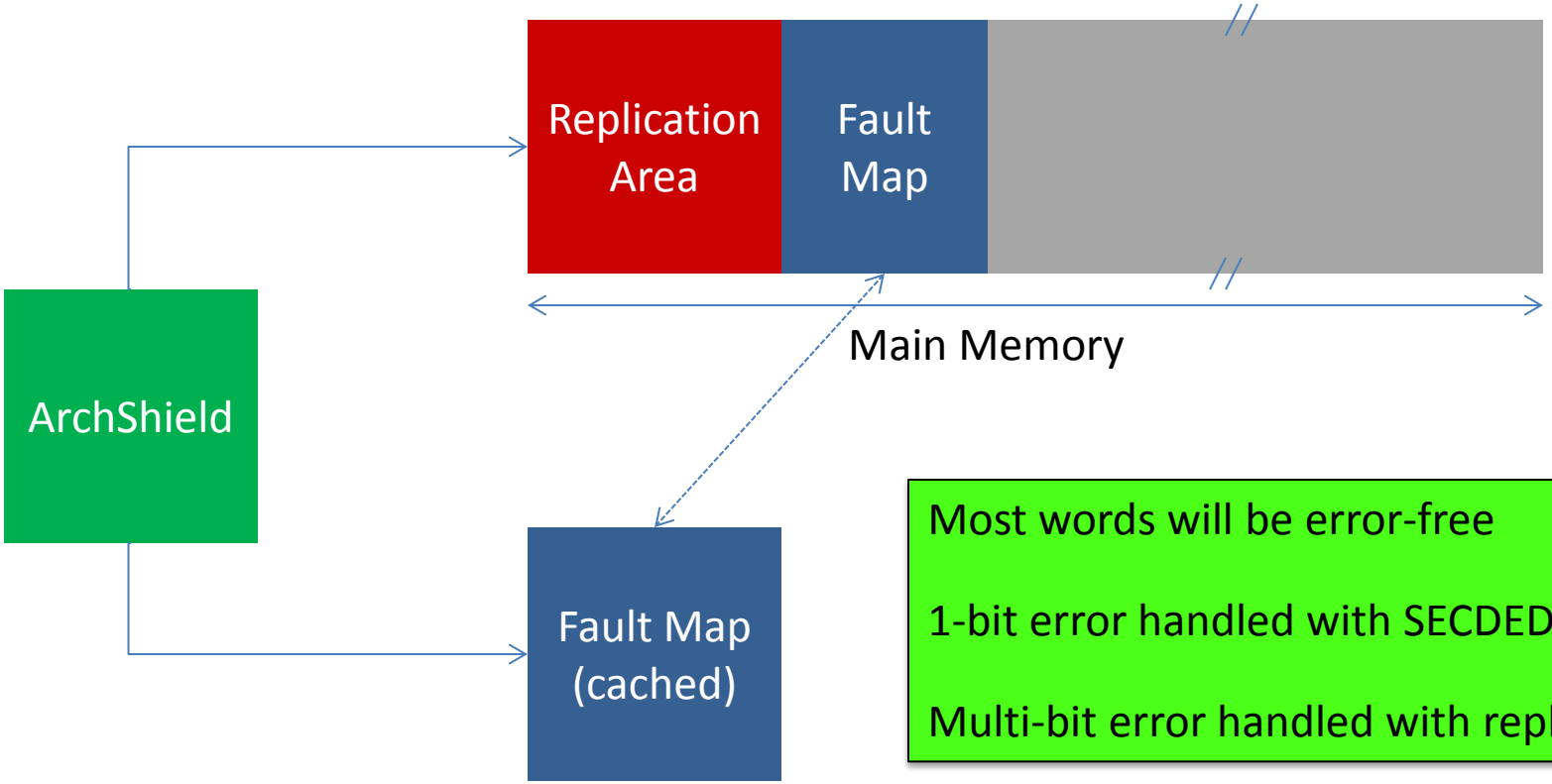
Most faulty words have 1-bit error ➔ The skew in fault probability can be leveraged for low cost resilience

Tolerate high error rates with commodity ECC DIMM while retaining soft-error resilience

ArchShield: Overview

Inspired from Solid State Drives (SSD) to tolerate high bit-error rate

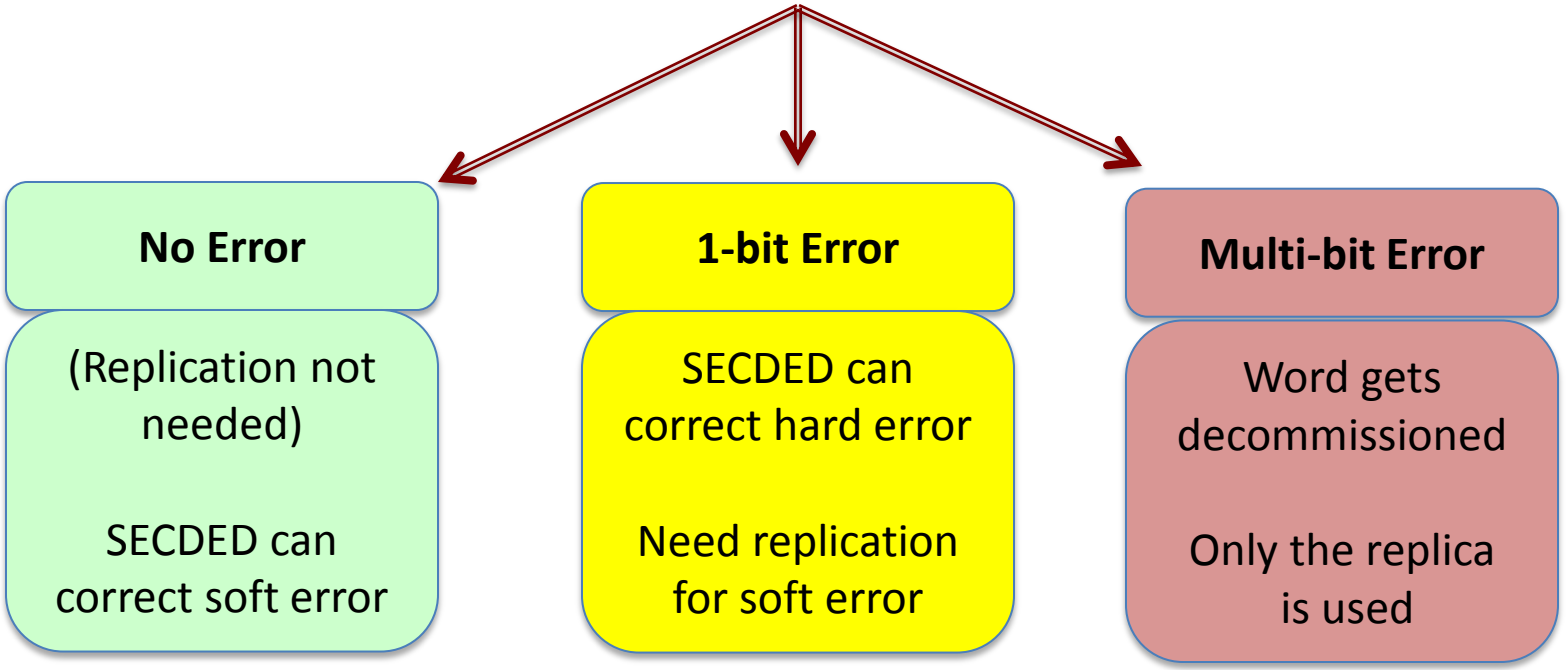
Expose faulty cell information to Architecture layer via runtime testing



ArchShield stores the error mitigation information in memory

ArchShield: Yield Aware Design

When DIMM is configured, runtime testing is performed. Each 8B word gets classified into one of three types:



(classification of faulty words can be stored in hard drive for future use)

Tolerates 100ppm fault rate with 1% slowdown and 4% capacity loss

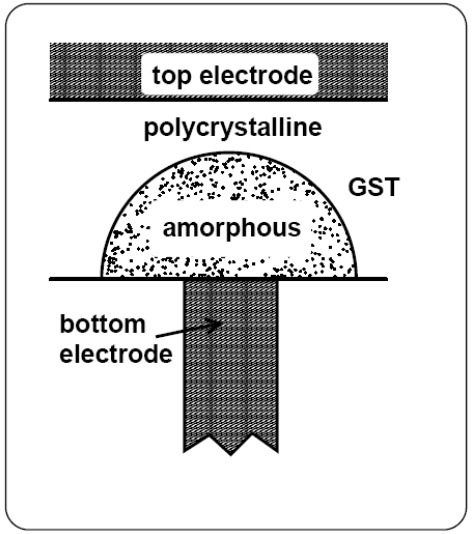
Outline

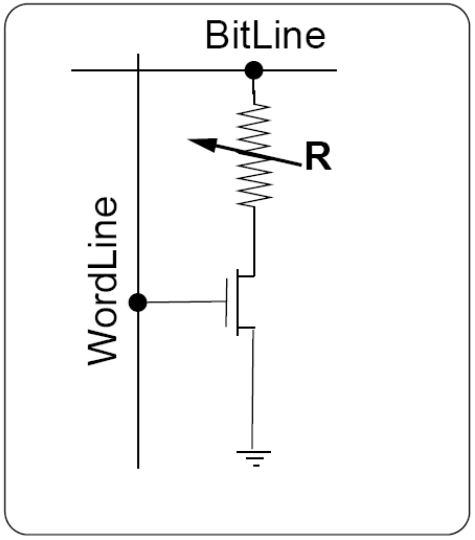
- Introduction
- ArchShiled: Yield Aware (arch support for DRAM)
- Hybrid Memory: reduce Latency, Energy, Power
- Adaptive Tuning of Systems to Workloads
- Summary

Emerging Technology to aid Scaling

Phase Change Memory (PCM): Scalable to sub 10nm

Resistive memory: High resistance (0), Low resistance (1)





Advantages: scalable, has MLC capability, non volatile (no leakage)

PCM is attractive for designing scalable memory systems. But …

Challenges for PCM

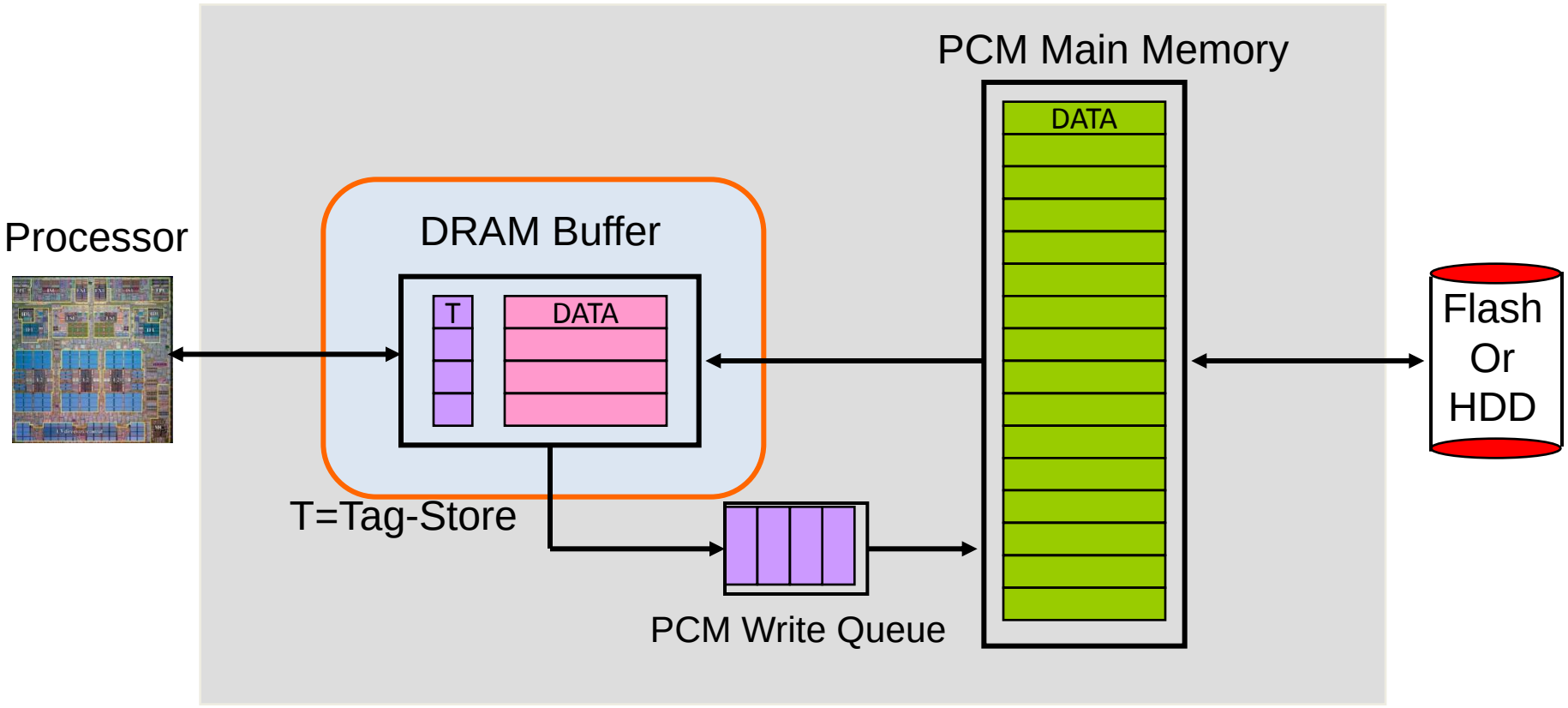
Key Problems:

1. Higher read latency (compared to DRAM)
2. Limited write endurance (~10-100 million writes per cell)
3. Writes are much slower, and power hungry

Replacing DRAM with PCM causes:
High Read Latency,
High Power
High Energy Consumption

How do we design a scalable PCM without these disadvantages?

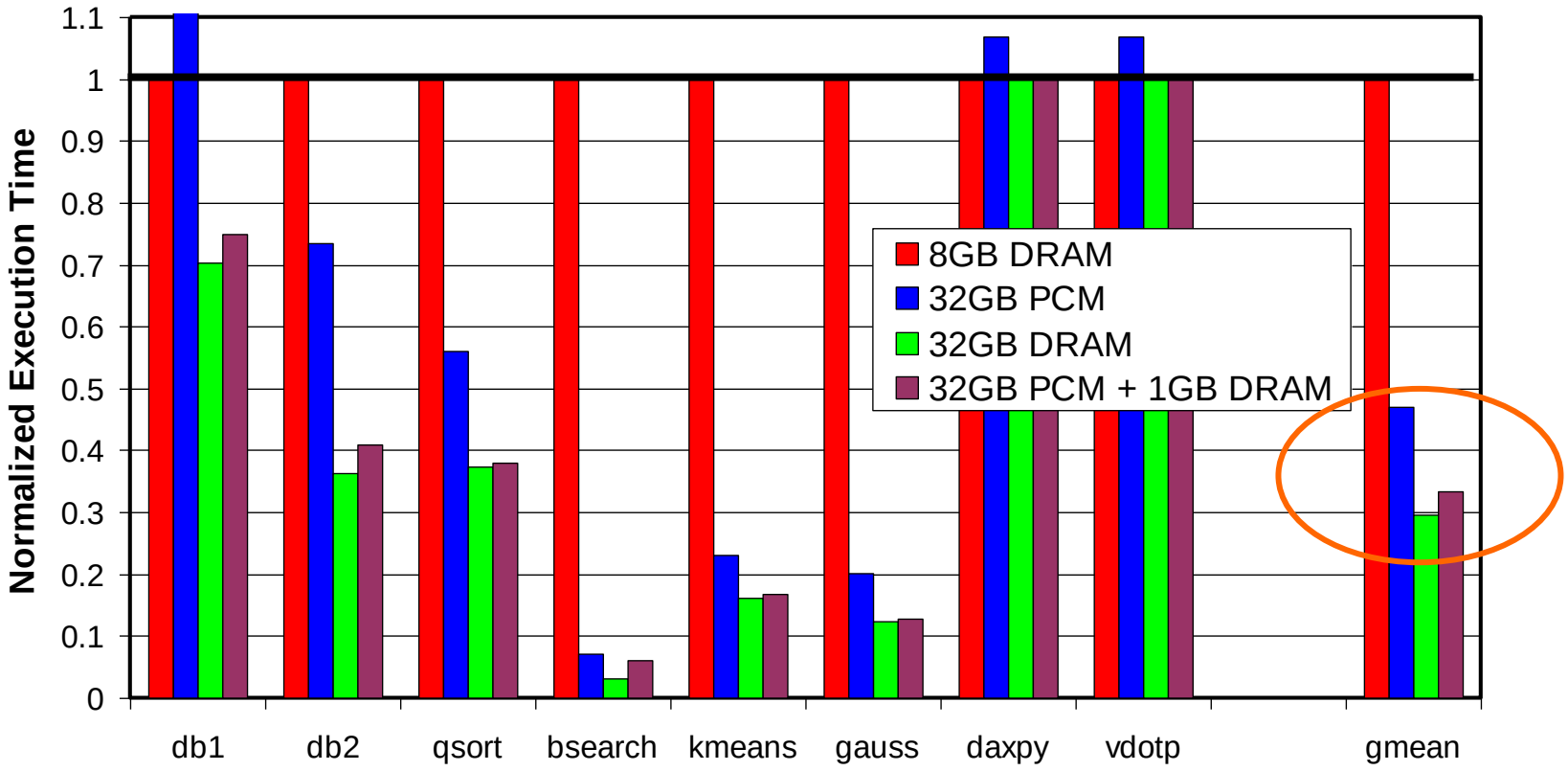
Hybrid Memory: Best of DRAM and PCM



Hybrid Memory System:

1. DRAM as cache to tolerate PCM Rd/Wr latency and Wr bandwidth
2. PCM as main-memory to provide large capacity at good cost/power
3. Write filtering techniques to reduces wasteful writes to PCM

Latency, Energy, Power: Lowered



Hybrid memory provides performance similar to iso-capacity DRAM
Also avoids the energy/power overheads from frequent writes

Outline

- Introduction
- ArchShiled: Yield Aware (arch support for DRAM)
- Hybrid Memory: reduce Latency, Energy, Power
- Adaptive Tuning of Systems to Workloads
- Summary

Workload Adaptive Systems

Different policies work well for different workloads

1. No single replacement policy works well for all workloads
2. Or, the prefetch algorithm
3. Or, the memory scheduling algorithm
4. Or, the coherence algorithm
5. Or, any other policy (write allocate/no allocate?)

Unfortunately: systems are designed to cater to average case
(a policy that works good enough for all workloads)

Ideal for each workload to have the policy that works best for it

Adaptive Tuning via Runtime Testing

Say we want to select between two policies: P0 and P1

Divide the cache in three:

- Dedicated P0 sets
- Dedicated P1 sets
- Follower sets (winner of P0,P1)

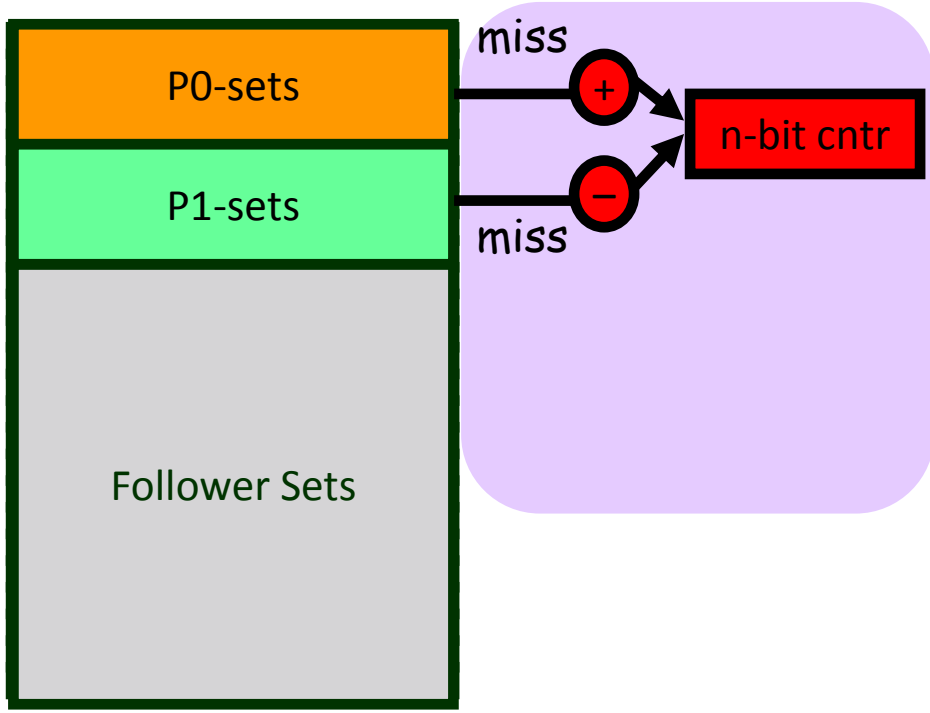
n-bit saturating counter

misses to P0-sets: counter++

misses to P1-set: counter--

Counter decides policy for Followers:

- MSB = 0, Use P0
- MSB = 1, Use P1



monitor → choose → apply
(Set Dueling: using a single counter)

Adaptive Tuning can allow dynamic policy selection at low cost

Outline

- Introduction
- ArchShiled: Yield Aware (arch support for DRAM)
- Hybrid Memory: reduce Latency, Energy, Power
- Adaptive Tuning of Systems to Workloads
- Summary

Challenges for Computer Architects

End of: Technology Scaling, Frequency Scaling, Moore's Law, ????

How do we address these challenges:

The solution for all computer architecture problems is:

Yield Awareness

Hybrid memory: Latency, Energy, Power reduction for PCM

Workload adaptive systems: low cost "Adaptivity Through Testing"

Challenges for Computer Architects

End of: Technology Scaling, Frequency Scaling, Moore's Law, ????

How do we address these challenges:

The solution for all computer architecture problems is:

Yield Awareness

Hybrid memory: Latency, Energy, Power reduction for PCM

Workload adaptive systems: get low cost Adaptivity Through Testing

Happy 75th Yale !