RELIABILITY OF RESISTIVE MEMORIES

Mahdi Nazm Bojnordi

Assistant Professor

School of Computing

University of Utah

Overview

- Upcoming deadlines
  - April 6th: student paper presentation

- This lecture
  - Hard errors in resistive memories
  - Increasing reliability by replication, ECP, SAFER, FREE-p
  - Resistive computing

Recall: Resistive vs. Dynamic RAM

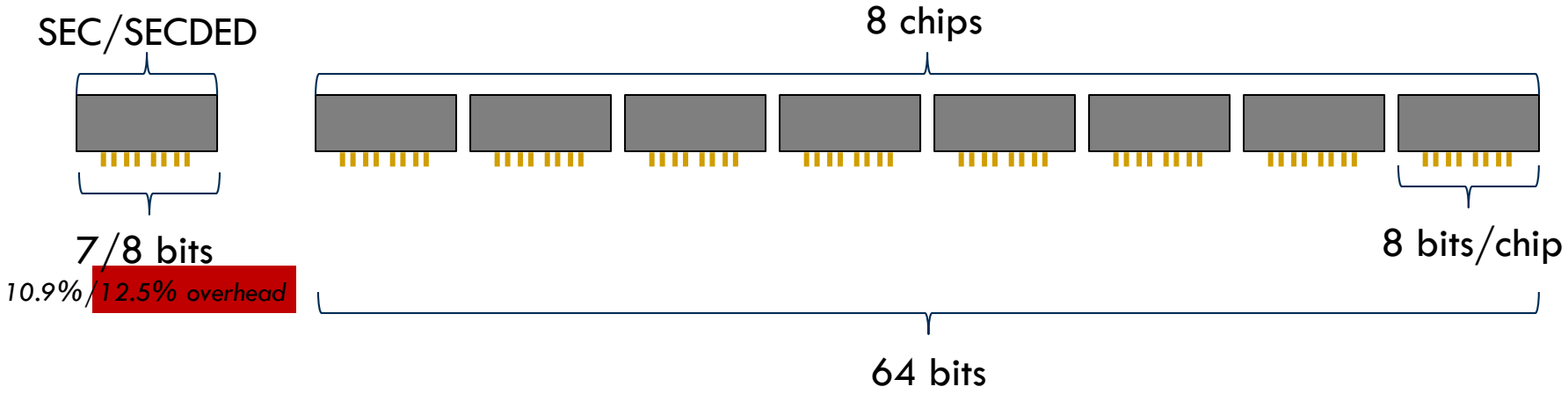
- Phase-Change RAM
  - Nonvolatile
  - Projected to be more scalable
  - Cells may be written individually
  - Slower, with more energy intensive writes
  - Susceptible to **hard** errors
- DRAM
  - Volatile, charge based
  - Difficult to further scale down the capacitor
  - All of the accesses are through row buffer
  - Faster, with acceptable energy consumption
  - Vulnerable to **soft** errors

Solutions to Memory Hard Errors

- Accept failure of some fraction of pages
  - Map failed pages out of logical memory
- Wear-level data pages/blocks, and within blocks
  - Shift/rotate data randomly (intervals/locations)
- Differential writes
  - Write only cells with values that change
- Correct errors when possible
  - Error correction techniques

Error Correction Techniques

- No correction (detection only)
  - Inefficient
  - A page must be retired when the first cell fails
- SECDED ECC
  - With a 12.5% memory overhead

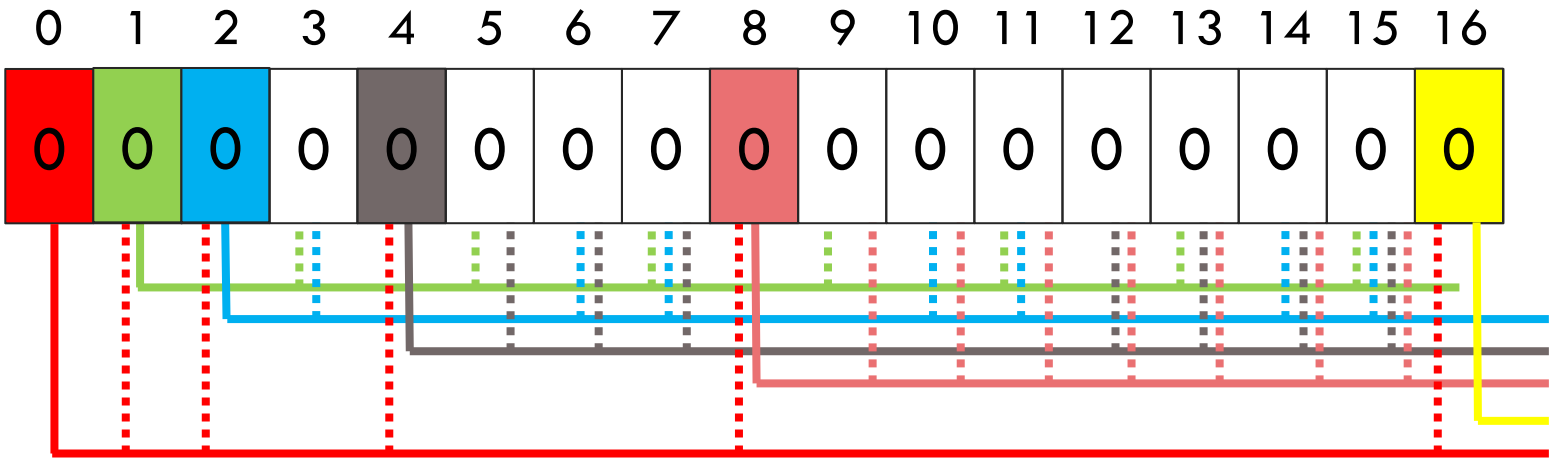


Error Correction Techniques

- No correction (detection only)
  - Inefficient
  - A page must be retired when the first cell fails
- SECDED ECC
  - With a 12.5% memory overhead
  - A page must be retired when a block within the page suffers a second error

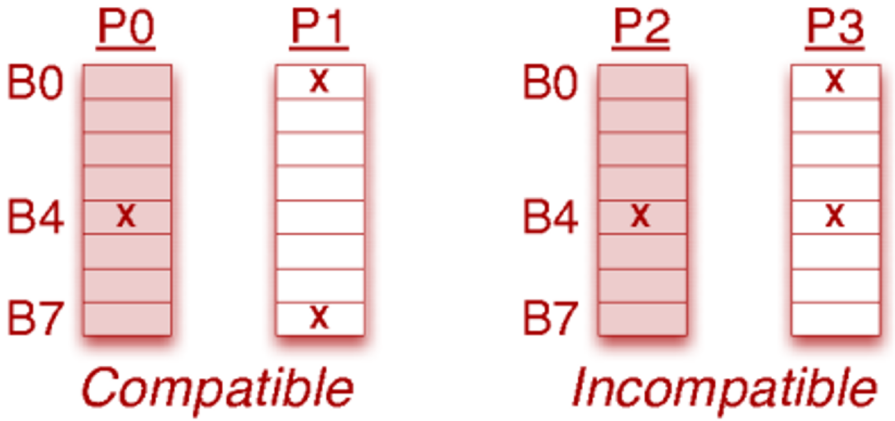
Error Correction Codes

- Good for soft errors
  - Transient errors
- Not good for hard errors
  - ECC has high entropy and can hasten wear-out
  - Flipping just one data bit changes about half of ECC bits



Dynamically Replicated Memory

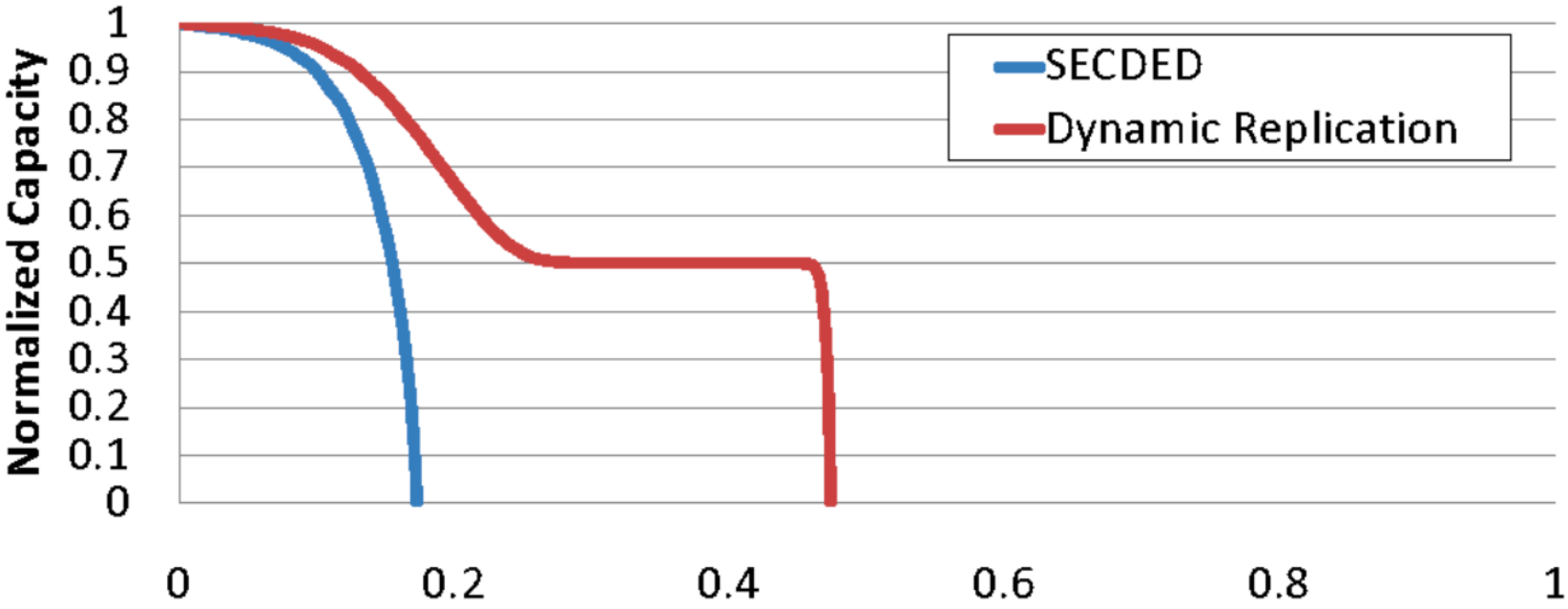
- Goal: handle hard errors by pairing two pages that have faults in different locations; replicate data across the two pages
- How: errors are detected with parity bits; replica reads are issued if the initial read is faulty



[ASPLOS'10]

Dynamically Replicated Memory

- Improve the lifetime of PCM by up to 40x over conventional error-detection techniques



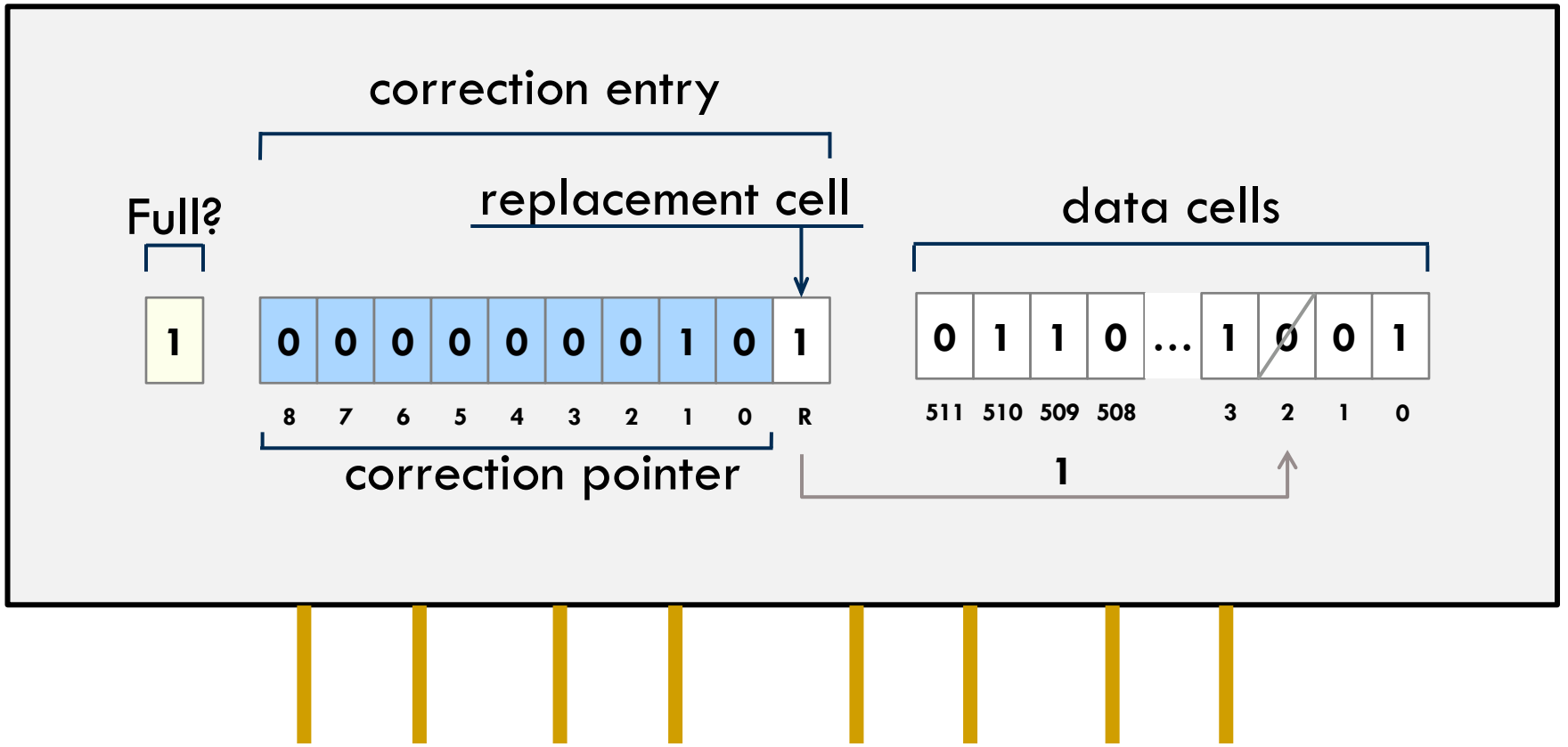
[ASPLOS'10]

Error Correction Pointers

- Key idea: instead of using ECC to handle a few transient faults in DRAM, use error-correcting pointers to handle hard errors in specific locations
- For a 512-bit line with 1 failed bit, maintain a 9-bit field to track the failed location and another bit to store the value in that location
- Can store multiple such pointers and can recover from faults in the pointers too

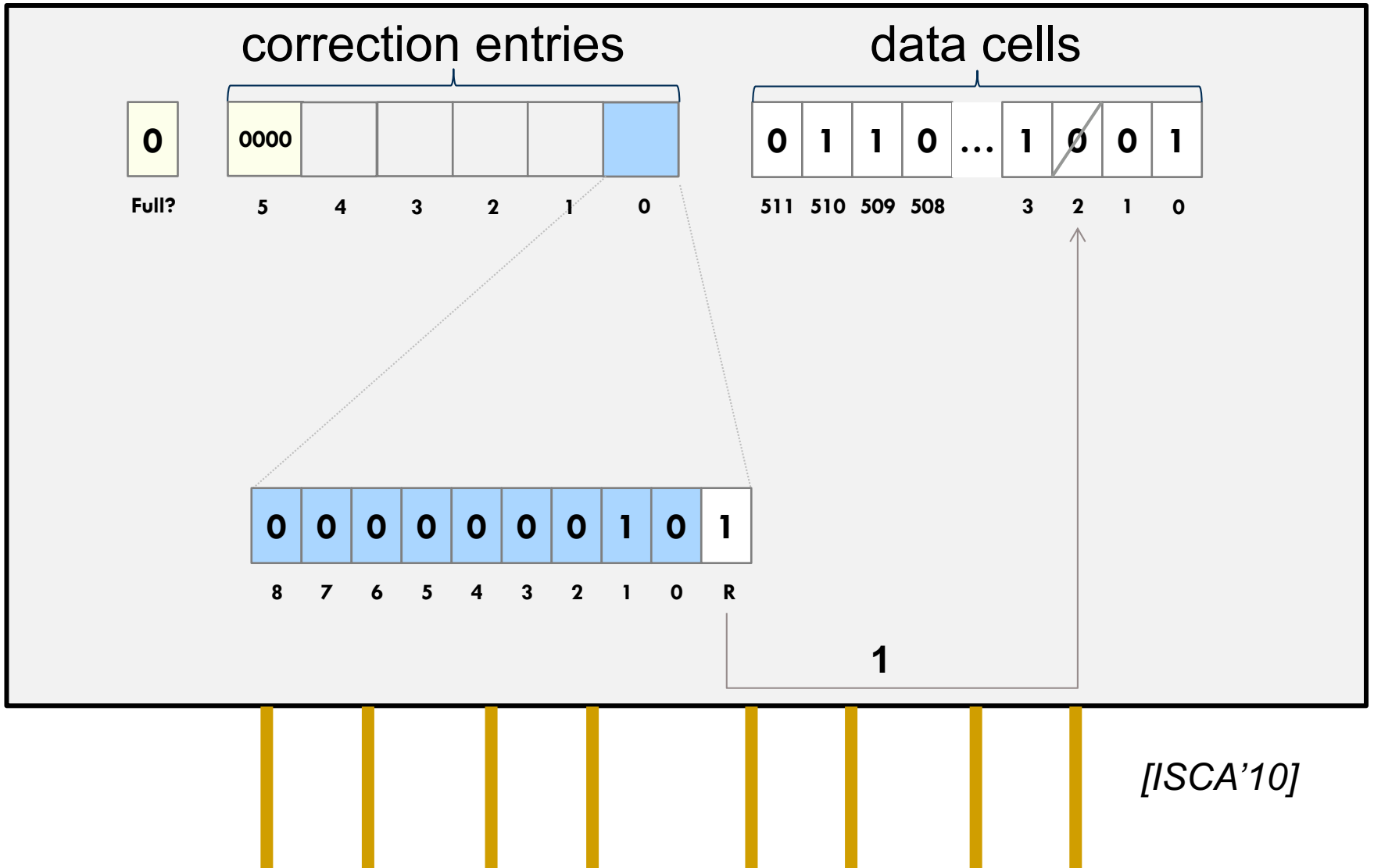
[ISCA'10]

Error Correction Pointers



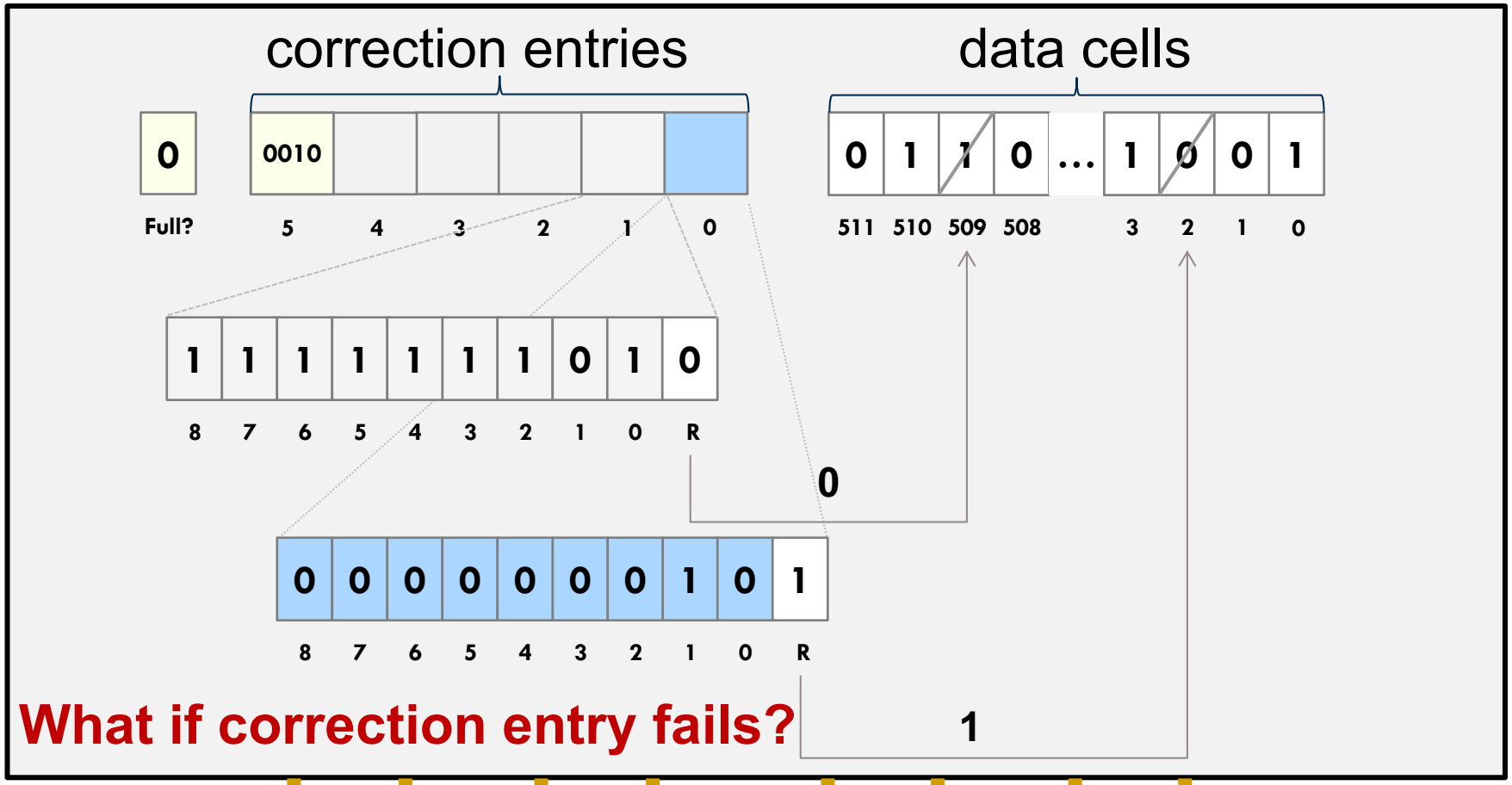
[ISCA'10]

Error Correction Pointers



[ISCA'10]

Error Correction Pointers



[ISCA'10]

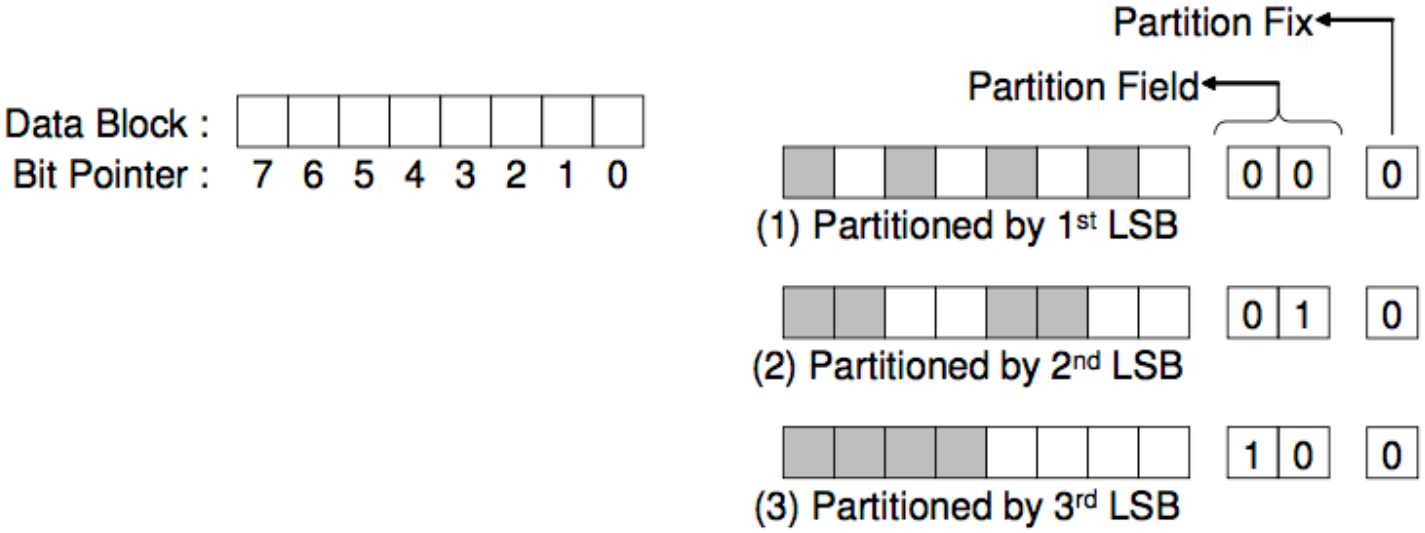
Stuck-At-Fault Error Recovery

- Observation: a failed cell with a stuck-at value is still readable
- Goal: either write the word or its flipped version so that the failed bit is made to store the stuck-at value
- For multi-bit errors, the line can be partitioned such that each partition has a single error
- Errors are detected by verifying a write; recently failed bit locations are cached so multiple writes can be avoided

[MICRO'10]

Stuck-At-Fault Error Recovery

- Three partition candidates in SAFER

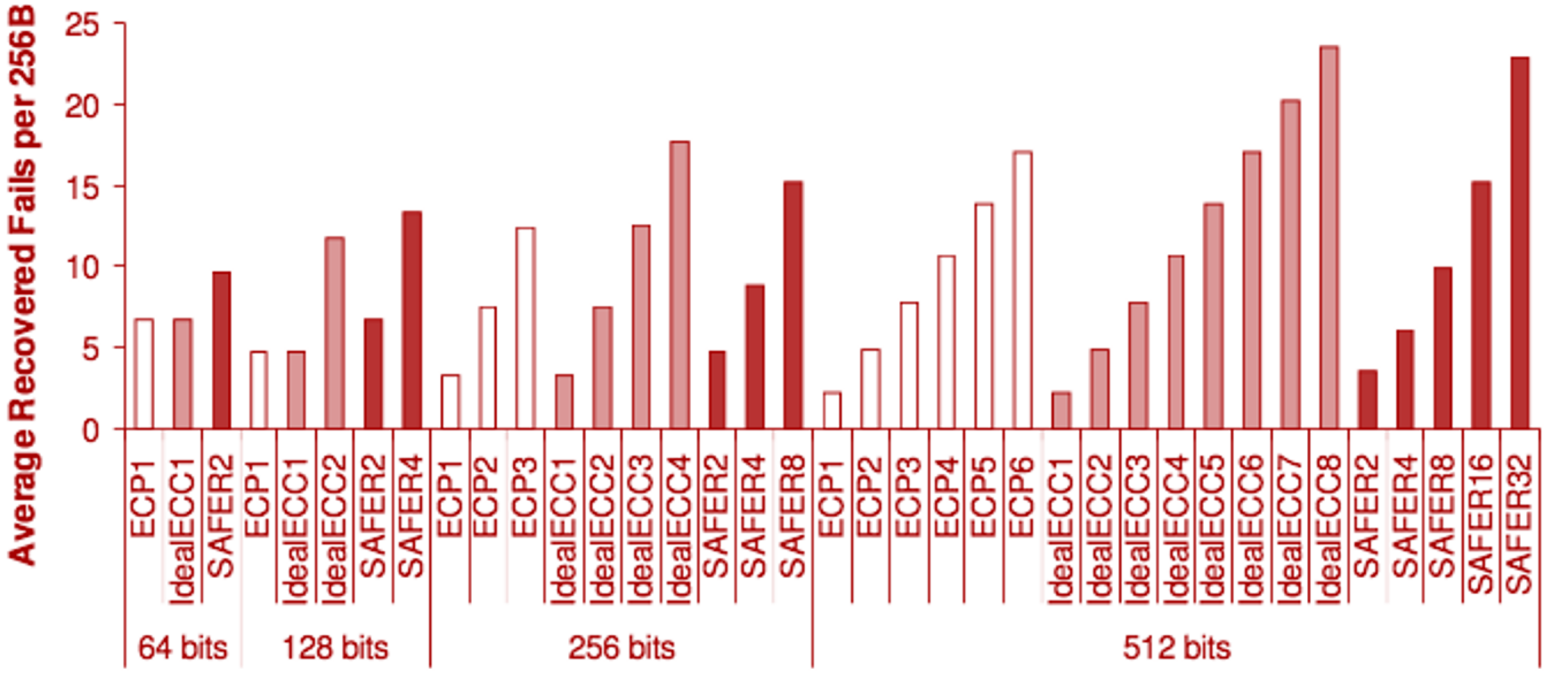


How to detect two fails?
(read the paper)

[MICRO'10]

Stuck-At-Fault Error Recovery

- Fail recovery



[MICRO'10]

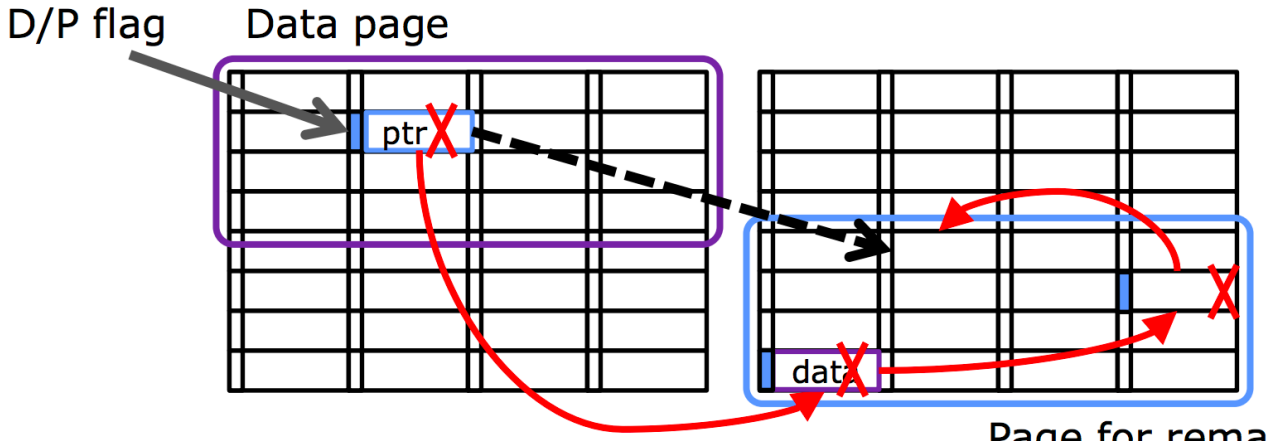
Multi-tiered ECC for Hard/Soft Errors

- FREE-p: fine-grained remapping with ECC and embedded pointer
  - Re-use a "dead" 64B block for storing a remap pointer
  - Architectural techniques to accelerate address remapping
- Detection/correction at the memory controller
  - Allow simple NVRAM devices
  - Tolerate hard/soft errors in the cell array, periphery, etc.

[HPCA'11]

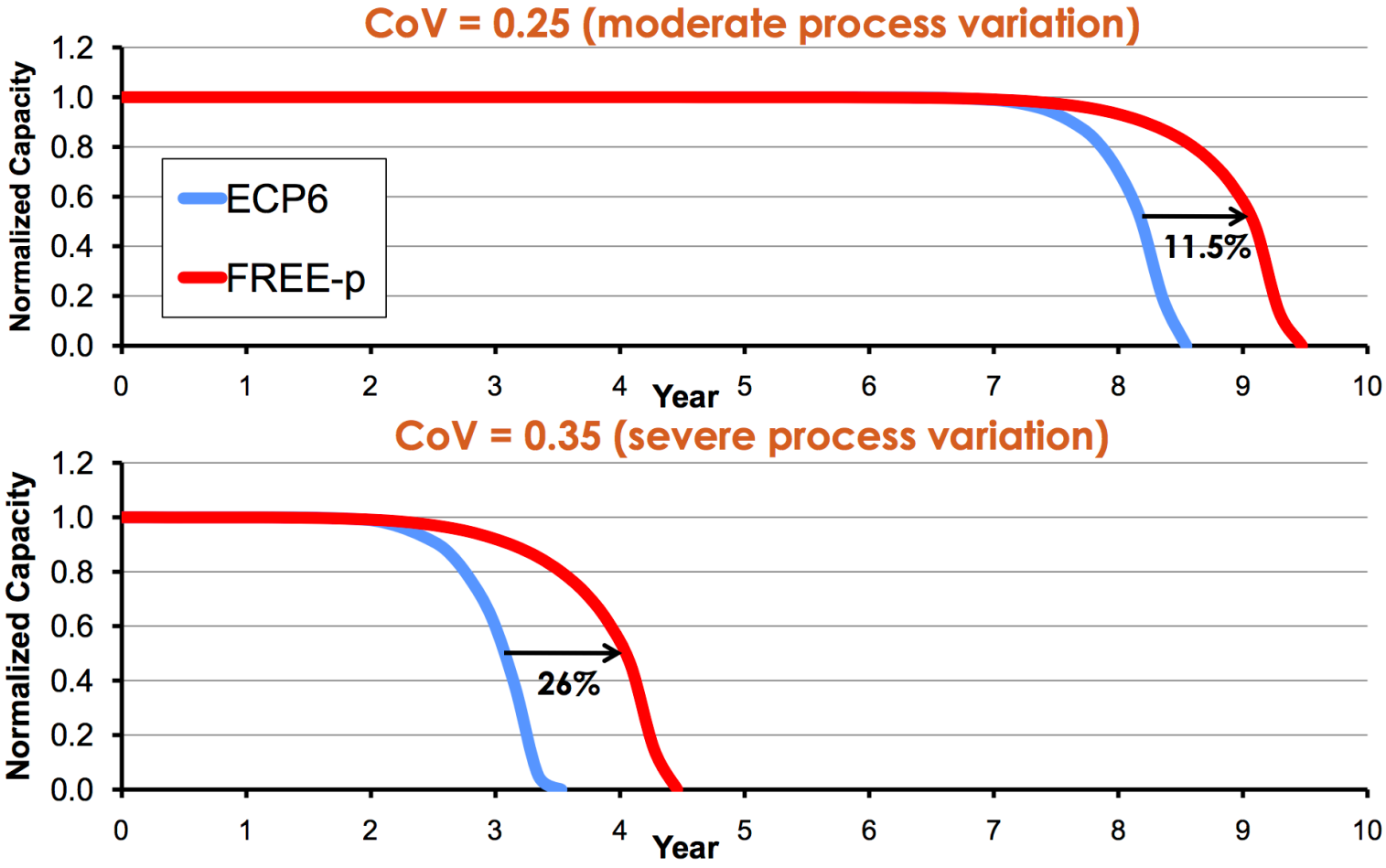
FREE-p

- Embed a 64-bit pointer within a faulty block
  - There are still-functional bits in a faulty block
  - 1-bit D/P flag per 64B block
    - Identify a block is remapped or not
  - Avoid chained remapping
    - Embed always the FINAL pointer



[HPCA'11]

Capacity vs. Lifetime



[HPCA'11]

Resistive Computation

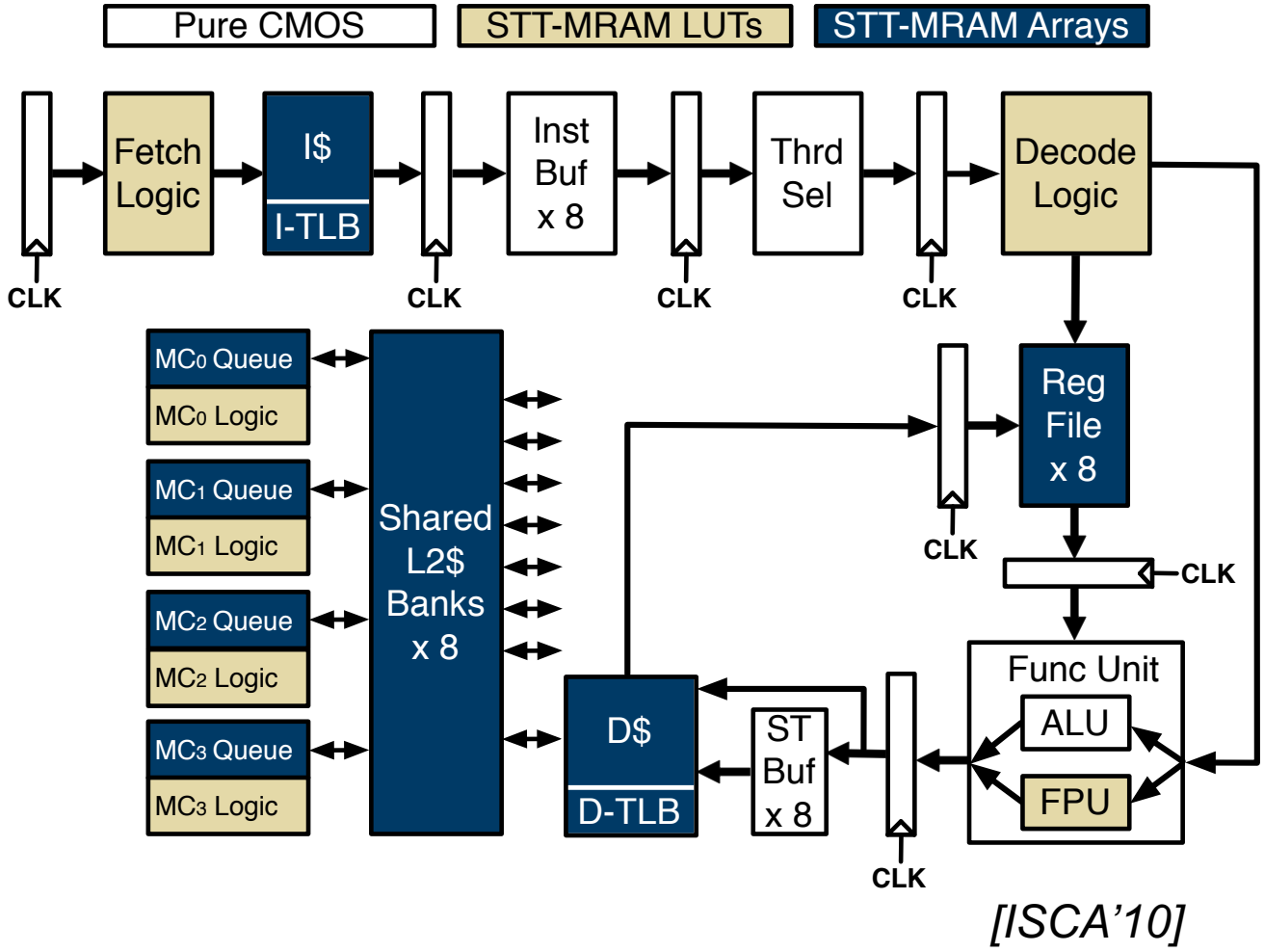
- Leverage STT-MRAM for energy efficiency
  - Near-zero leakage power
  - Low-energy read operation

- Goal: selectively migrate on-chip storage and combinational logic to STT-MRAM to reduce power
  - On-chip storage: caches, TLBs, register files, queues
  - Combinational logic: lookup-table (LUT) based computing

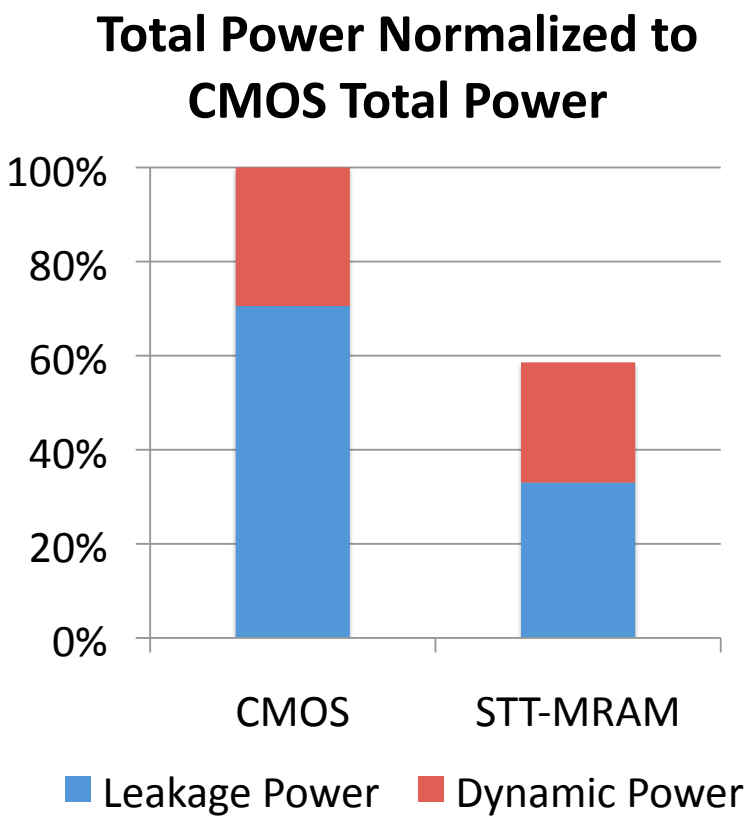
[ISCA'10]

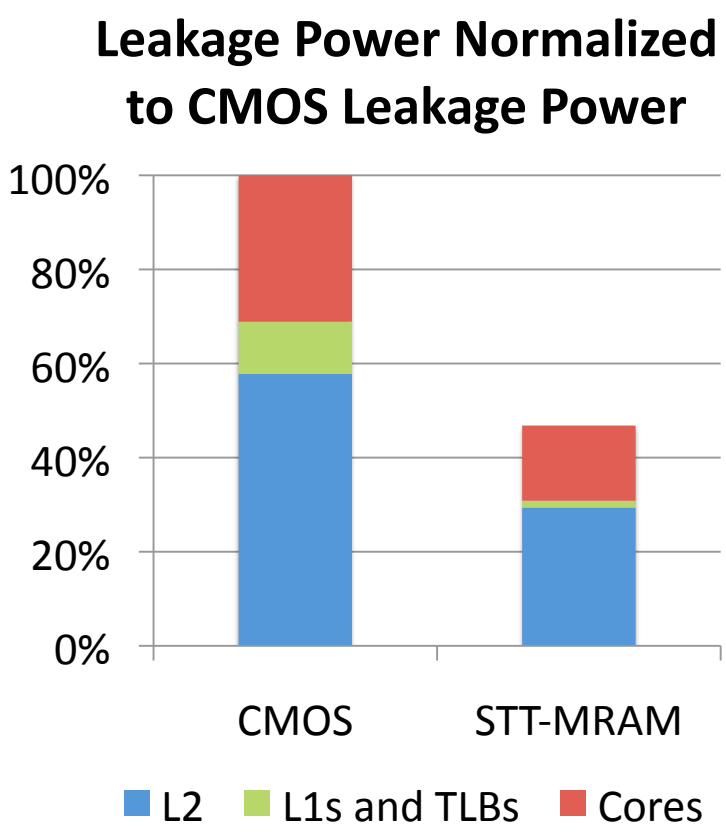
Hybrid CMT Pipeline

- Small arrays and simple logic in CMOS
- Large arrays and complex logic in STT-MRAM



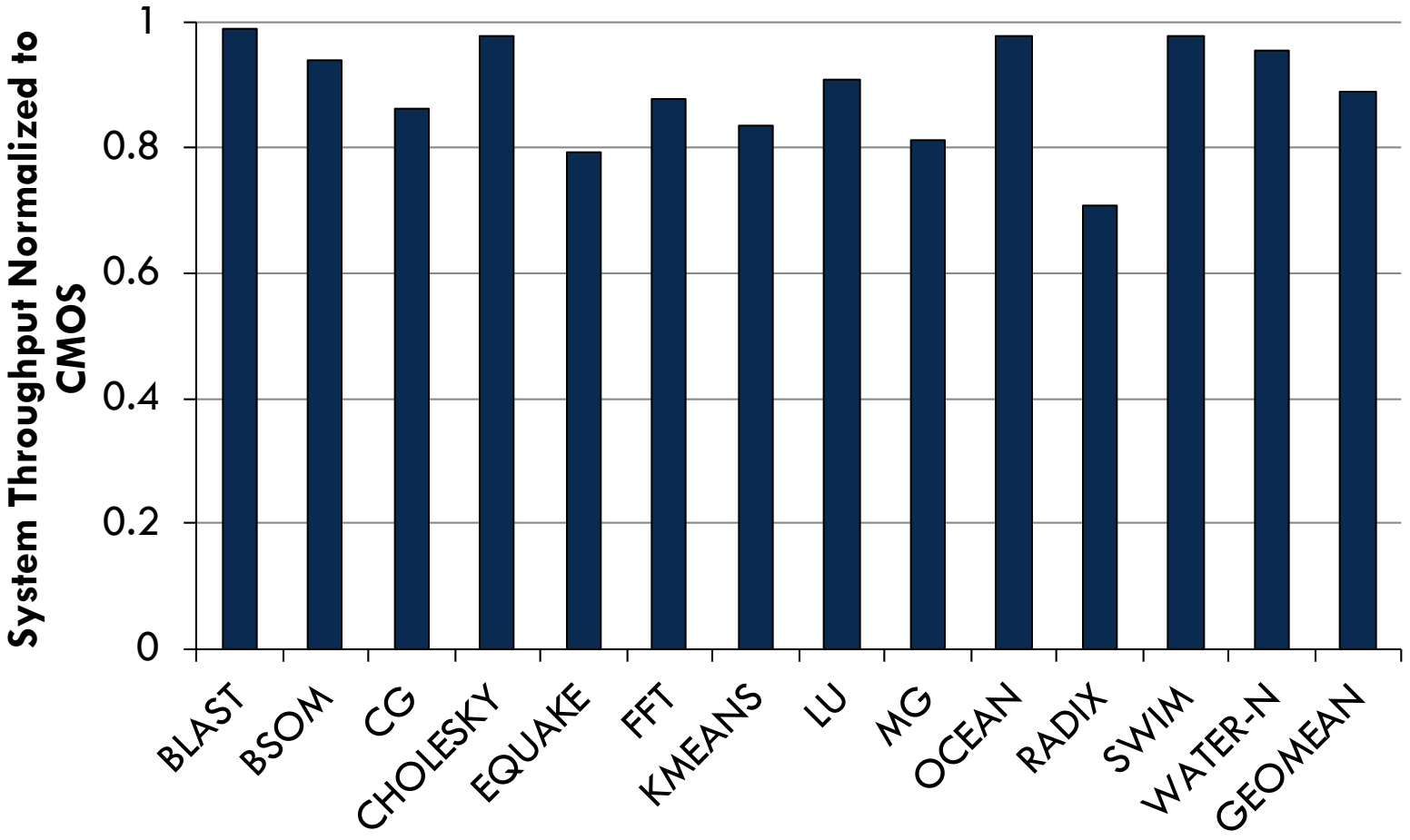
System Power





[ISCA'10]

System Performance



[ISCA'10]