

FREE-p: Protecting Non-Volatile Memory against both Hard and Soft Errors

Doe Hyun Yoon[†]

Naveen Muralimanohar[‡]

Jichuan Chang[‡]

Parthasarathy Ranganathan[‡]

Norman P. Jouppi[‡]

Mattan Erez[†]



[†]Electrical and Computer Engineering
The University of Texas at Austin

[‡] Intelligent Infrastructure Lab.
Hewlett-Packard Labs.

The Problem

- **Challenge in emerging non-volatile memory**
 - Finite write endurance
 - PCRAM cells wear out after 10^8 writes on average
 - Actual advertised specification is only 10^6
 - Process variation exacerbates the problem
 - Some cells fail earlier than others
- **Limitations of Prior solutions**
 - Tolerate only hard cell errors
 - Can't cover soft cell errors, errors on the periphery, wires, package, ...
 - Require custom logic within NVRAM devices

Fine-grained Remapping with ECC and Embedded-pointer

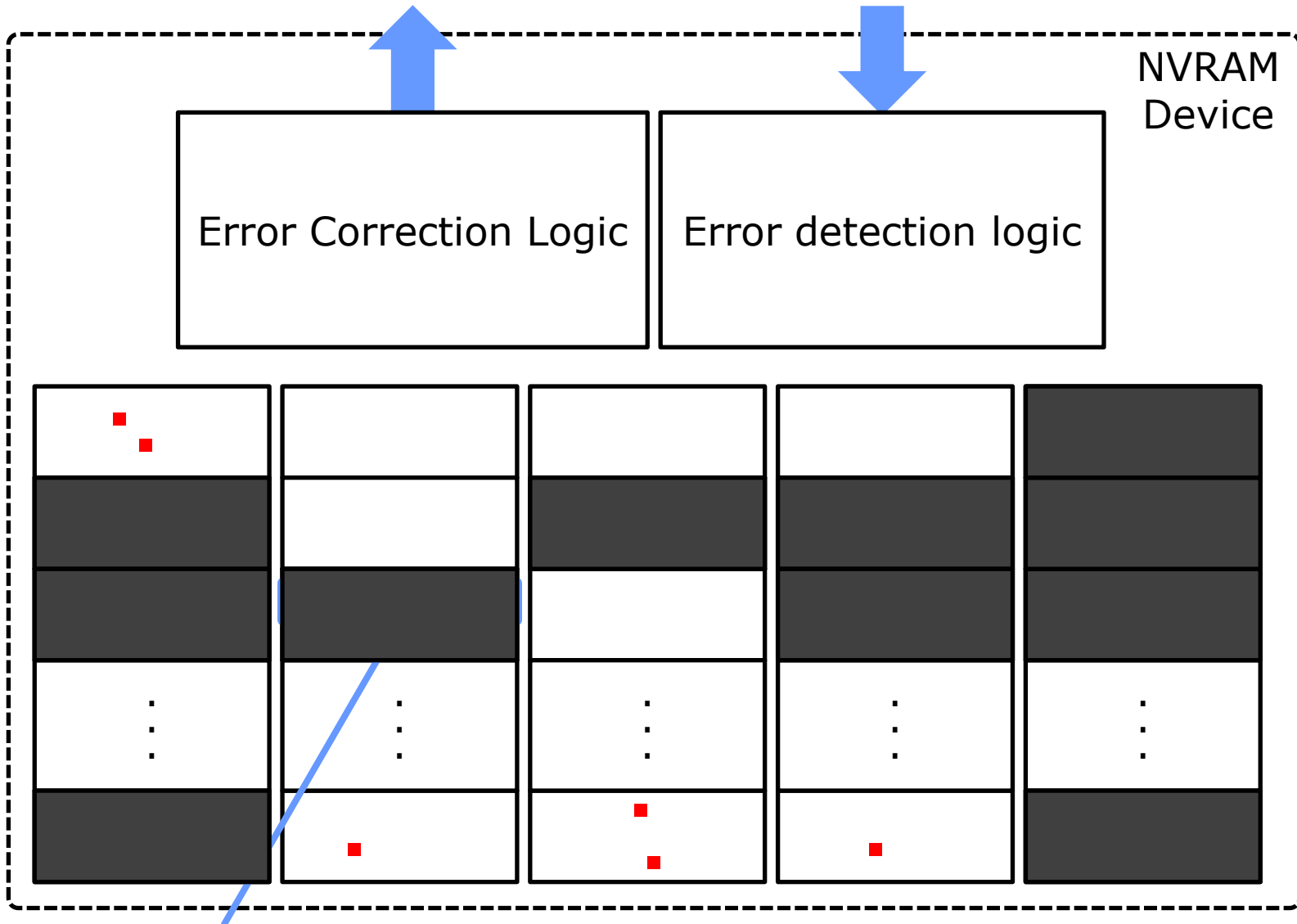
- **Multi-tiered ECC with fast and slow paths**
- **Fine-grained remapping**
 - Re-use a “dead” 64B block for storing a remap pointer
 - Architectural techniques to accelerate address remapping
- **Detection/correction at the memory controller**
 - Allow simple NVRAM devices
 - Tolerate hard/soft errors in the cell array, periphery, wire, ...
- **Trade off performance at the end of life with storage device cost and complexity**
 - Near-zero performance penalty initially
 - Less than 2% penalty, even at 7 years
- **Up to 26% longer lifetime**

Prior Solutions and Their Limitations

Prior solutions

- **Avoiding unnecessary writes**
- **Wear-leveling**
 - Widely used in FLASH memory
- ***Tolerating wear-out failures***
 - Built-in hard bit-error correcting mechanisms
 - Coarse-grained remapping
 - DRM: Dynamically Replicating Memory
 - ECP: Error Correcting Pointer
 - SAFER: Stuck-At-Failure Error Recovery

Prior Work: Tolerating Wear-out Failures



Intolerable Failure

Limitations of Prior Work

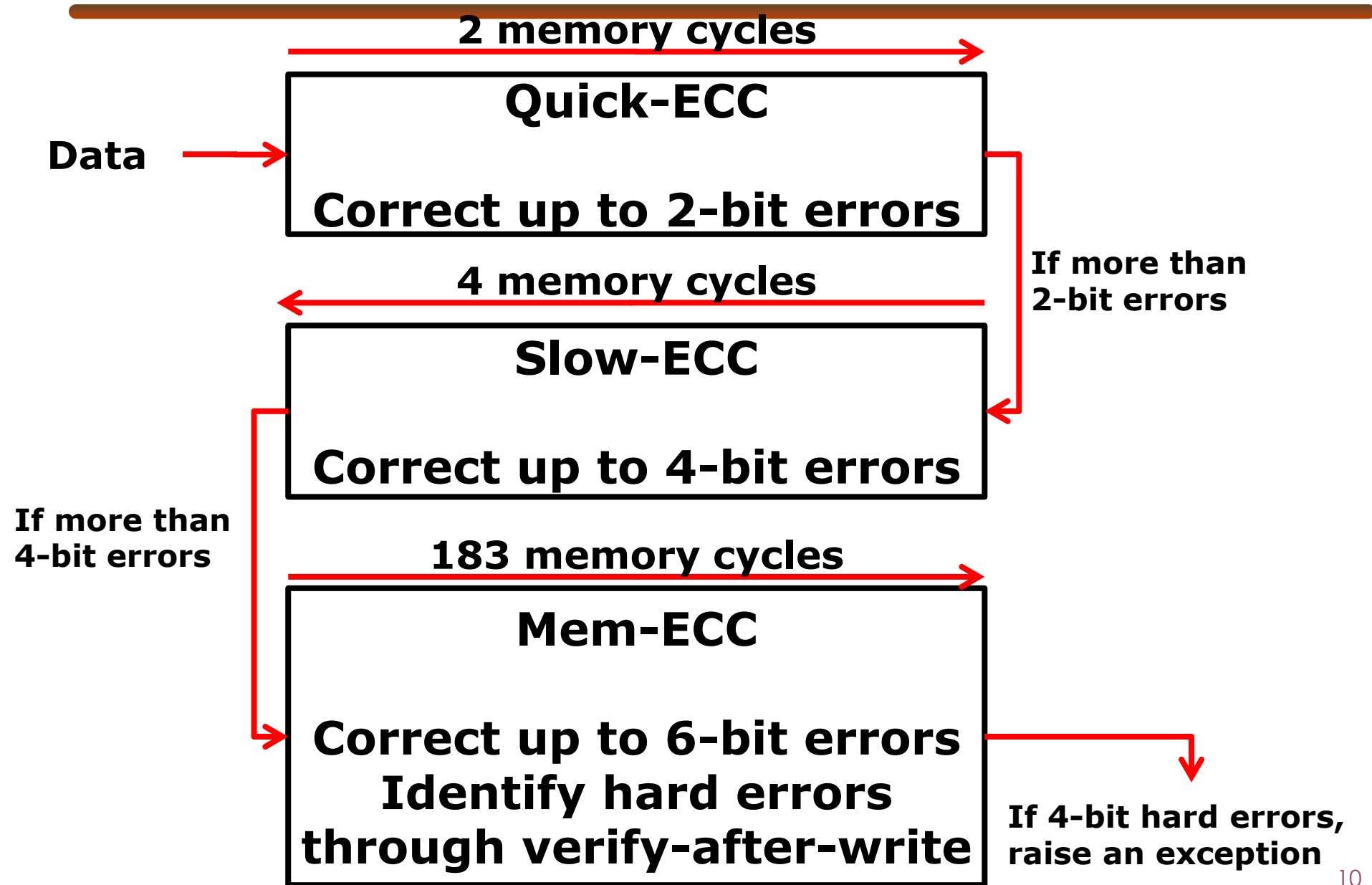
- **Only HARD cell errors**
 - Can't detect/correct SOFT errors
 - Resistance drift in PCRAM
 - Can't detect/correct errors on periphery, wires, and packaging
 - Supporting chipkill-correct?
- **Error detection/correction logic within NVRAM devices**
 - Memory industry favors SIMPLE and CHEAP devices
 - Error detection/correction in DRAM
 - DRAM chips only store data
 - Additional DRAM chips for storing redundant information
 - Error detection/correction is done at the memory controller
 - Better follow the same design strategy in NVRAM systems

FREE-p

Multi-tiered ECC

- **6EC-7ED BCH code**
 - 6-bit error correcting 7-bit error detecting
 - 61 bits for 64B data (less than 12.5% overhead)
- **Tolerate up to 4 bit wear-out failures and 2 bit soft errors**
- **Low-latency decoding for common case operations**
 - Quick-, Slow-, and Mem-ECC
 - Extend two-tiered decoding of Hi-ECC [Wilkerson ISCA'10]
 - Fast-path quick-ECC for most initial period

Multi-tiered ECC (Cont'd)

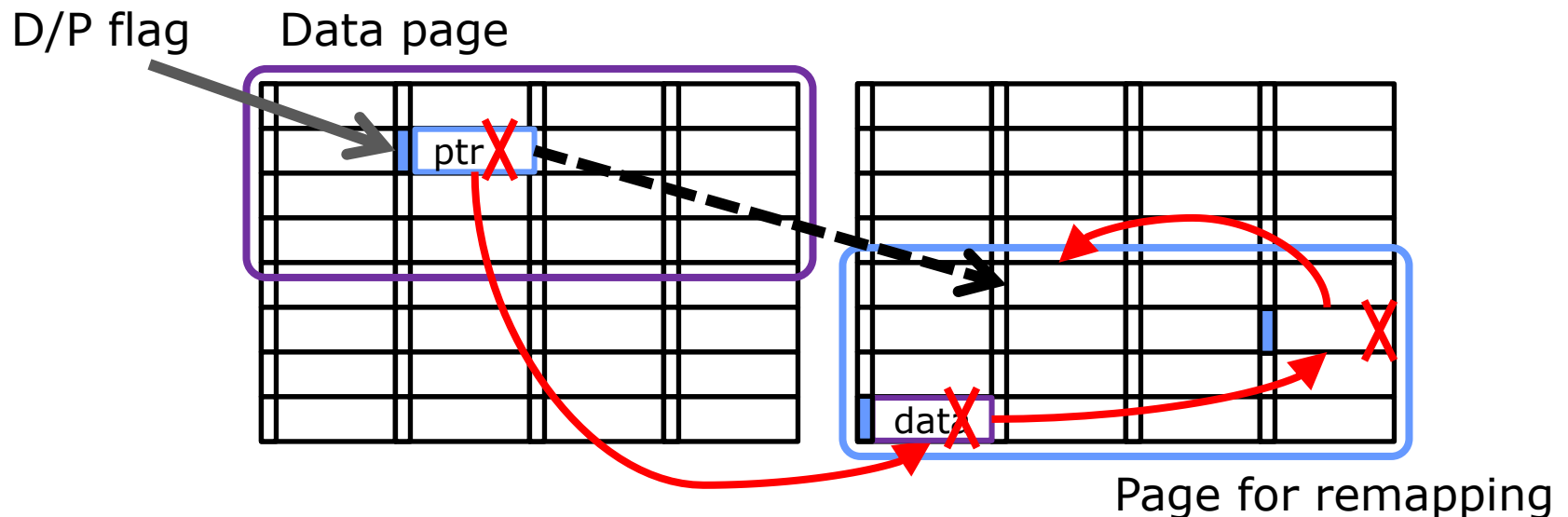


Dealing with Intolerable Failures

- **Eventually, some blocks become faulty**
 - More than 4 wear-out failures per block
- **Coarse-grained remapping (prior solutions)**
 - Leverage virtual-to-physical mapping
 - Mapping unit: 4kB or larger
 - A block with intolerable failure maps out the whole page
- **Fine-grained remapping**
 - Disable only a faulty block
 - Mapping unit: 64B
 - Effectively handle both random and concentrated errors

Fine-grained Remapping (FR) with Embedded-pointer

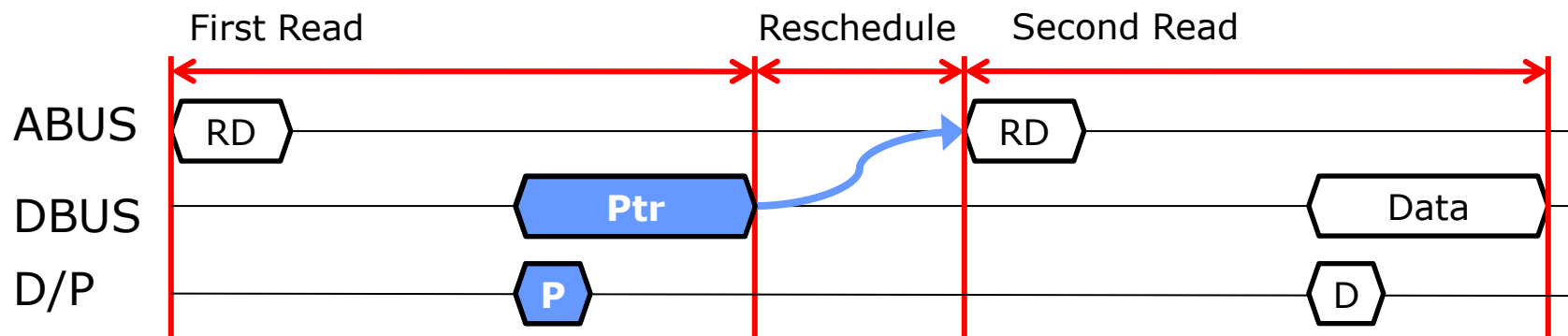
- Embed a 64-bit pointer within a faulty block
 - There are still-functional bits in a faulty block
 - Use 7-Modular Redundancy to tolerate the failures
- 1-bit D/P flag per 64B block
 - Identify a block is remapped or not
- Avoid chained remapping
 - Embed always the FINAL pointer



Read with FR

- **Read with FR**

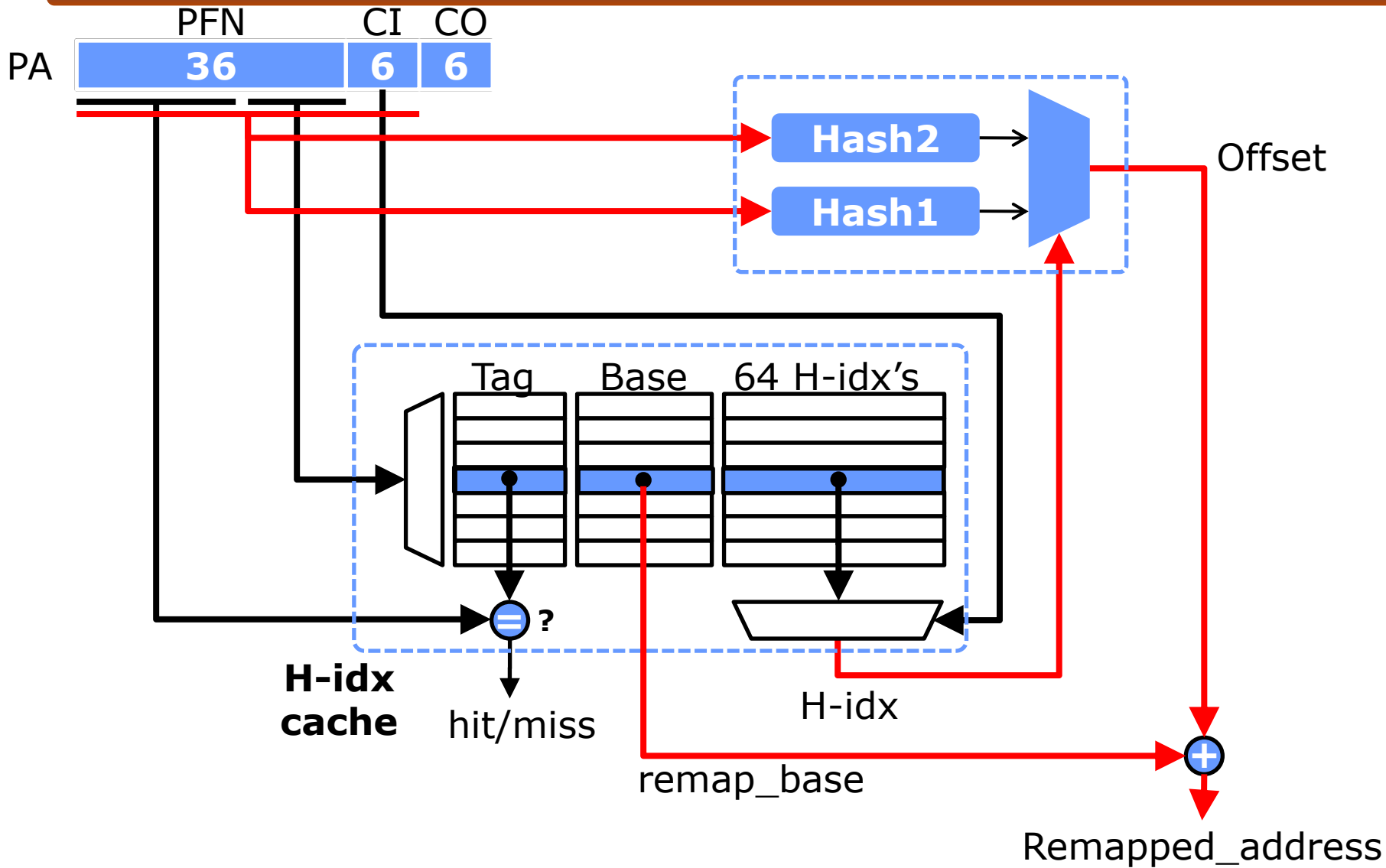
- Read data and D/P flag
- If a data block is remapped
 - Read the remapped block
 - Increase read latency, waste bandwidth
 - Penalty increases as NVRAM wear out



How to Mitigate This Penalty?

- **Remap pointer cache**
 - Cache remap pointers
 - Avoid reading remap pointer from NVRAM when cache hit
- **Hash based index cache**
 - Pre-defined hash functions for remapping
 - Compute, not cache, remap pointer
 - H-idx: which hash function is used for remapping?
 - 0: not remapped
 - 1 or 2: remapped using one of the hash functions
 - 3: hash collision
 - All candidate locations are already used for other blocks
 - Need to read the remap pointer
 - 2 bits per 64B block

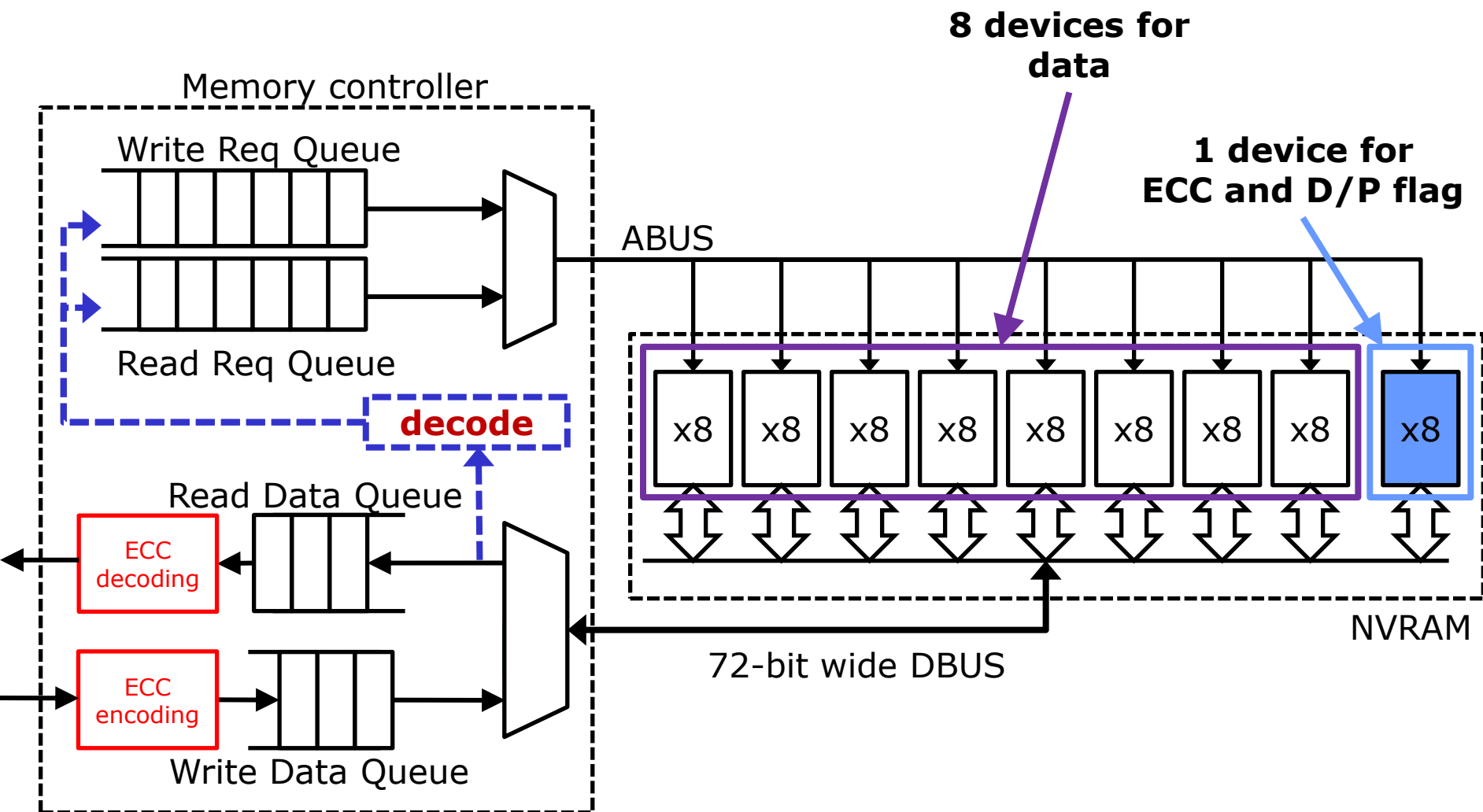
Hash-Based Index Cache



Index Cache (Cont'd)

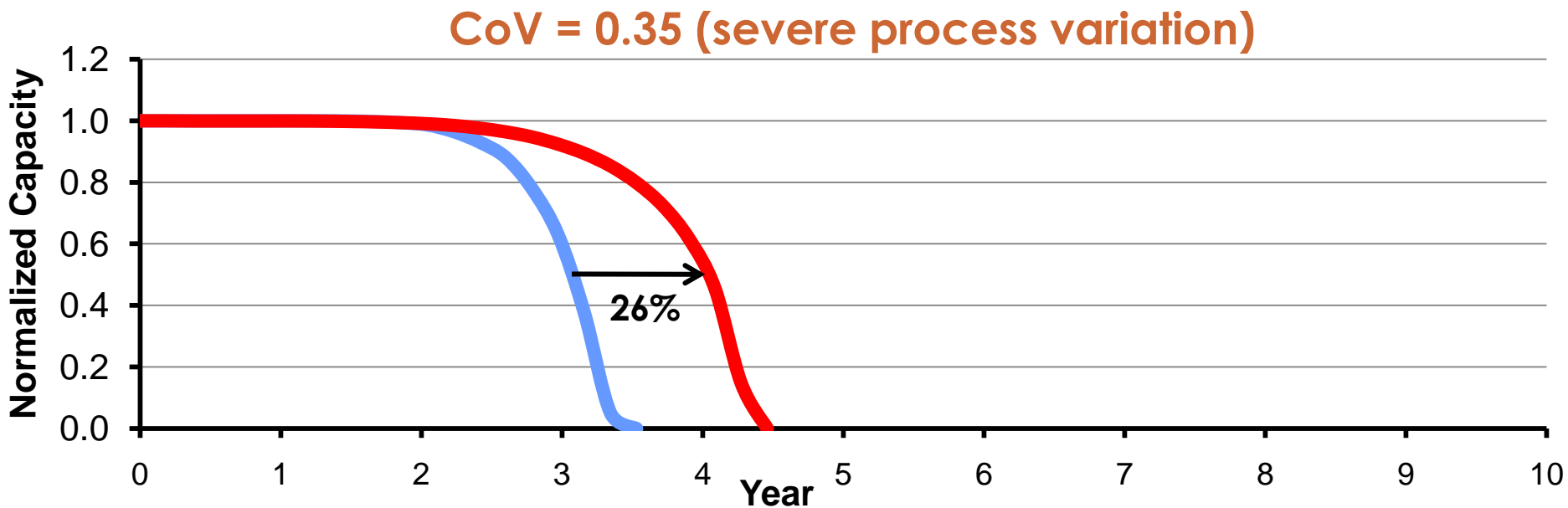
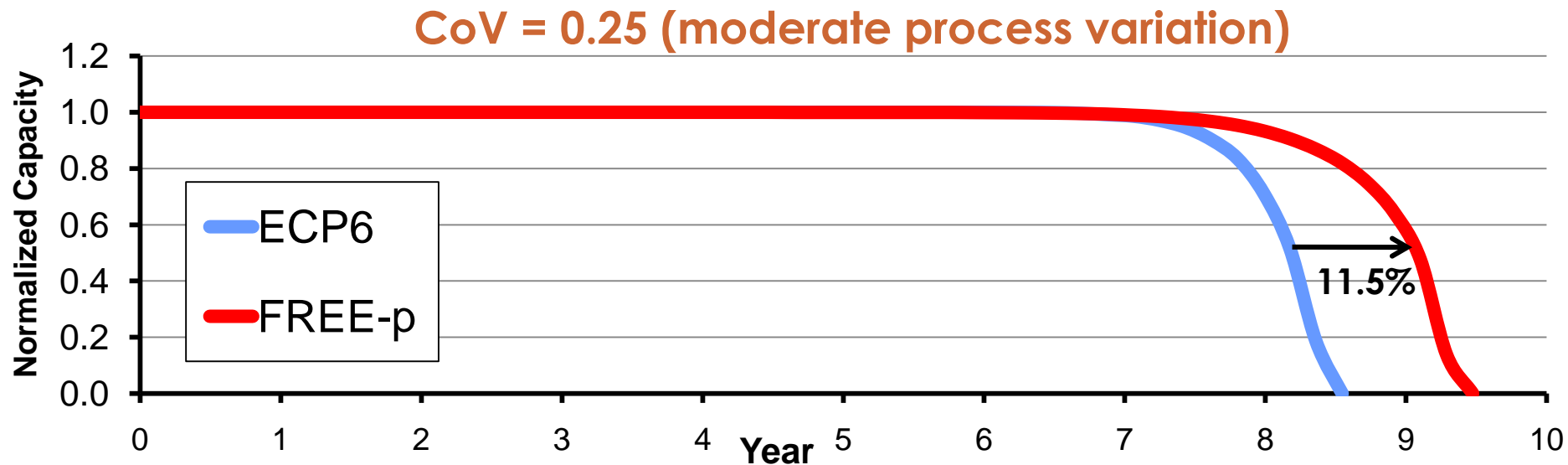
- **Fill/evict with TLB**
 - 100% read hit rate
- **Read remap pointers only when**
 - Dirty write-back misses the index cache (rare)
 - Hash collision (rare with good hash functions)
- **OS should be aware of hash functions**

Memory System Organization with FREE-p



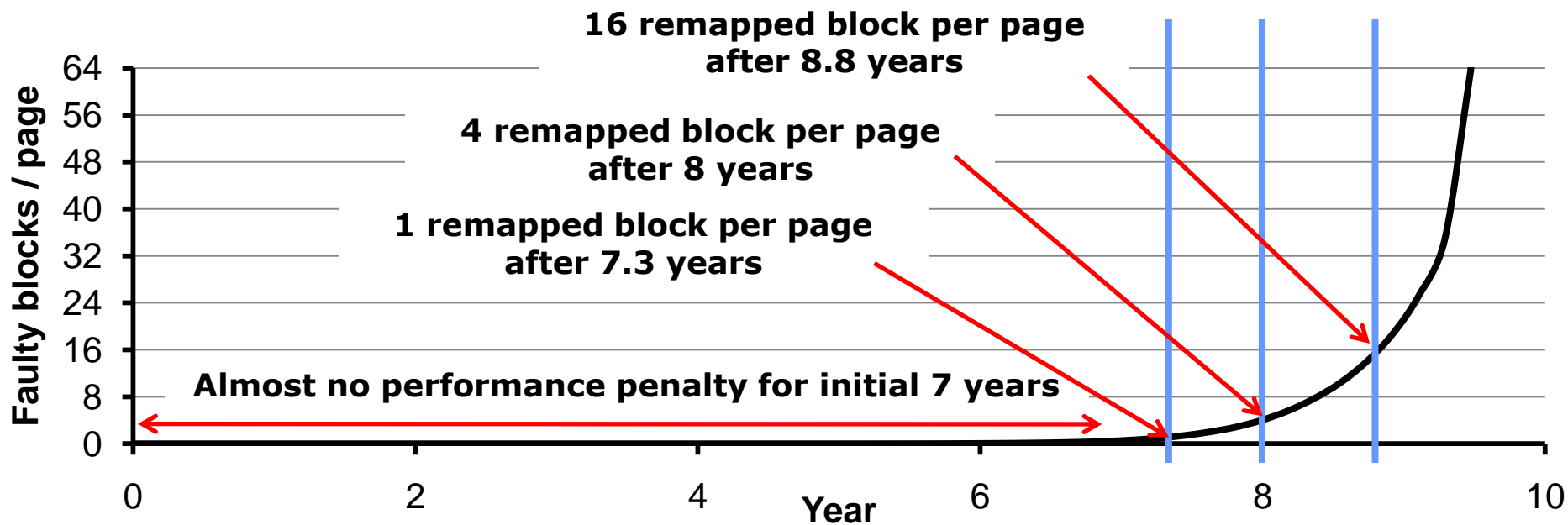
Evaluation

Capacity vs. Lifetime

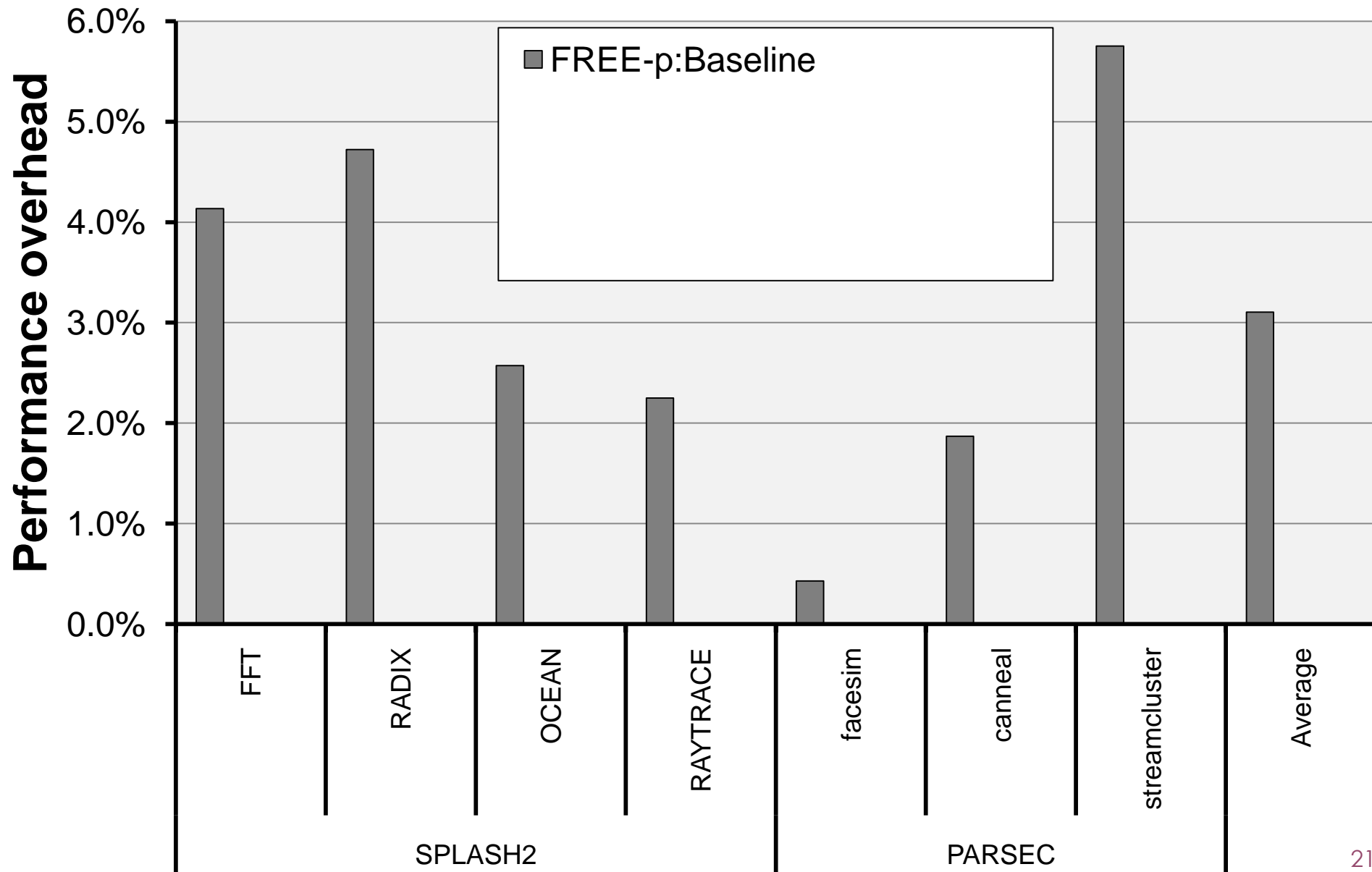


Performance Evaluation

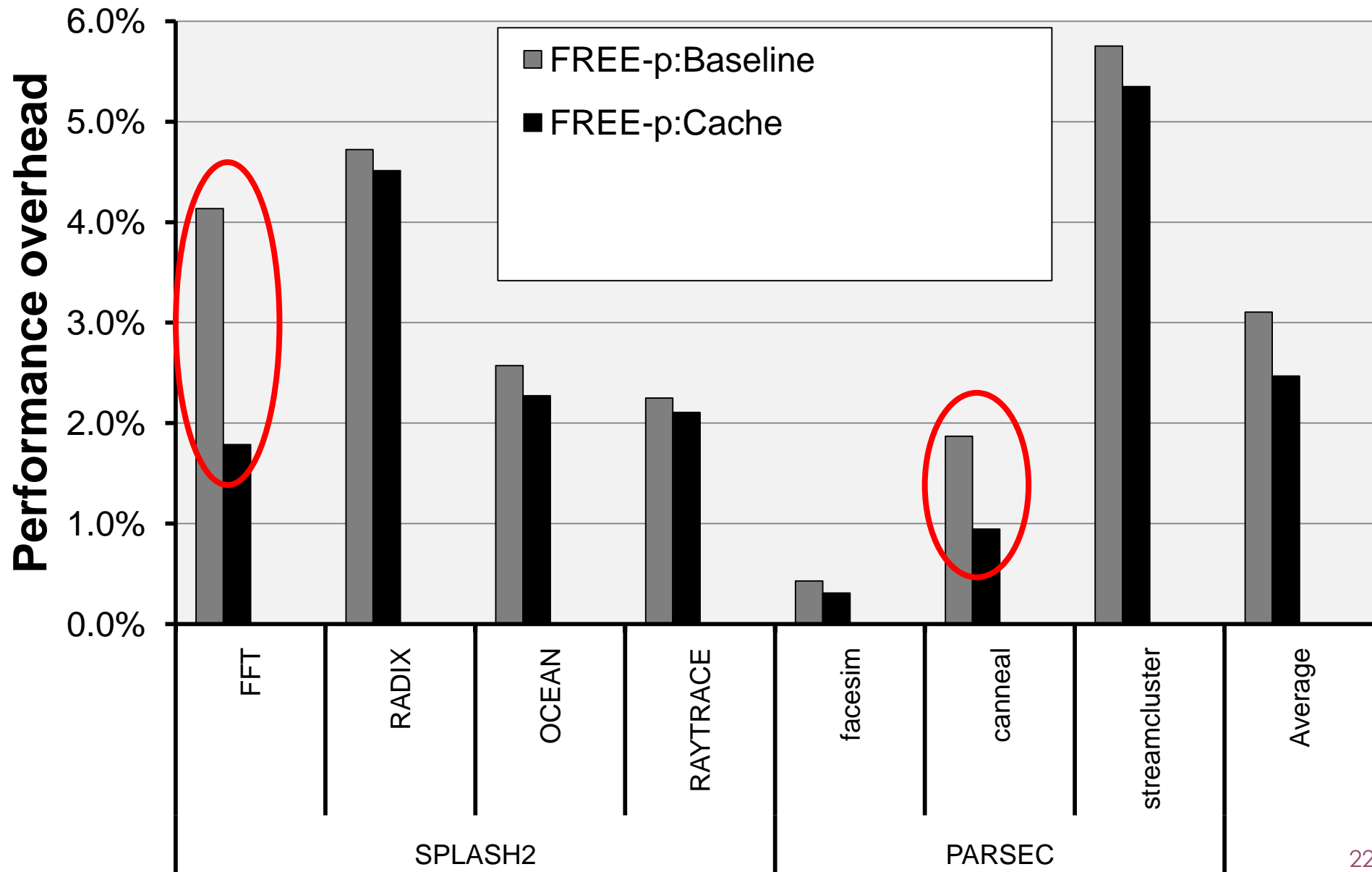
- In-order core with detailed NVRAM model
 - PCRAM with DDR3-like channel interface
- Performance depends on NVRAM wear-out status
 - Fault injection based on failure simulation



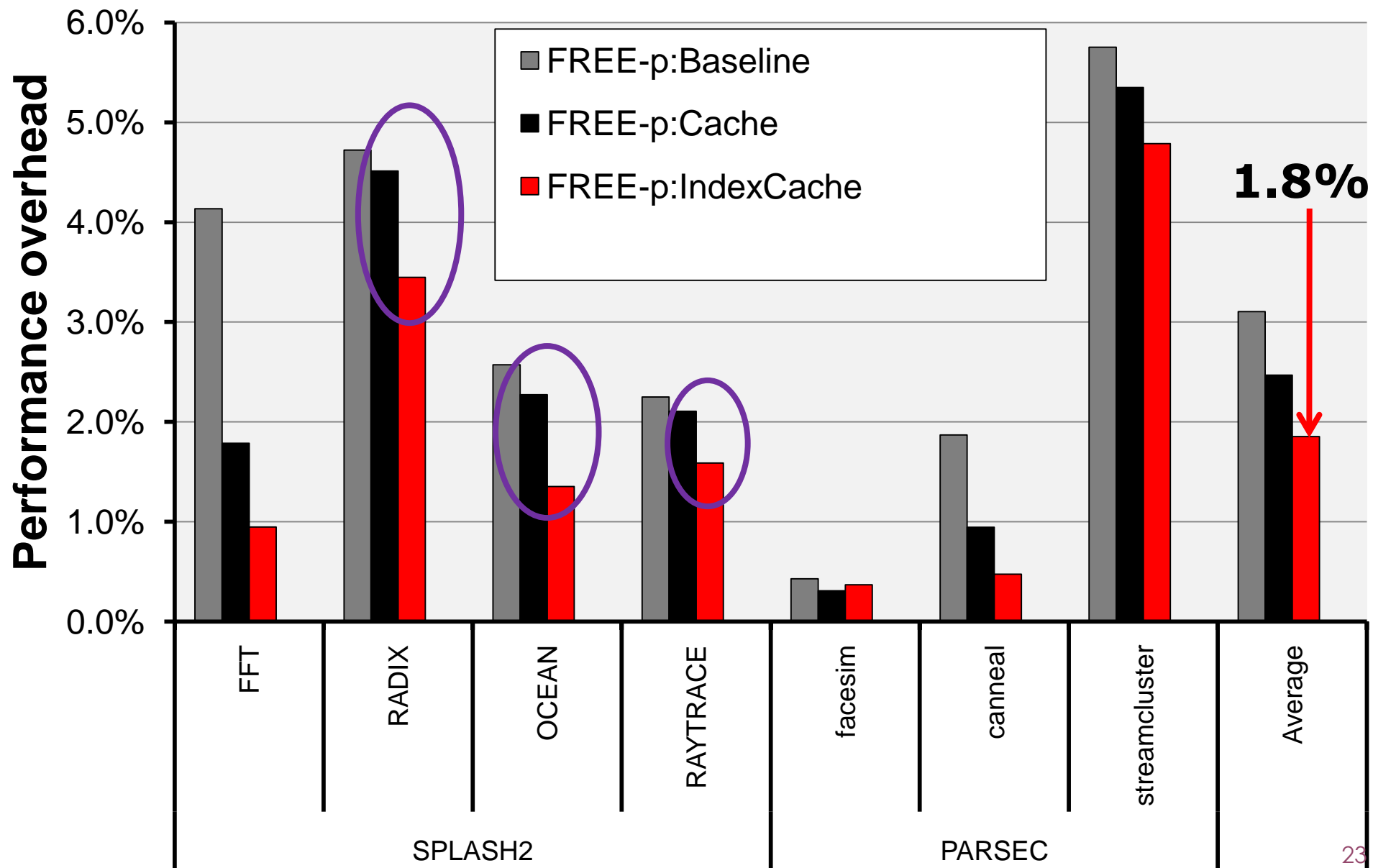
After 7.3 Years



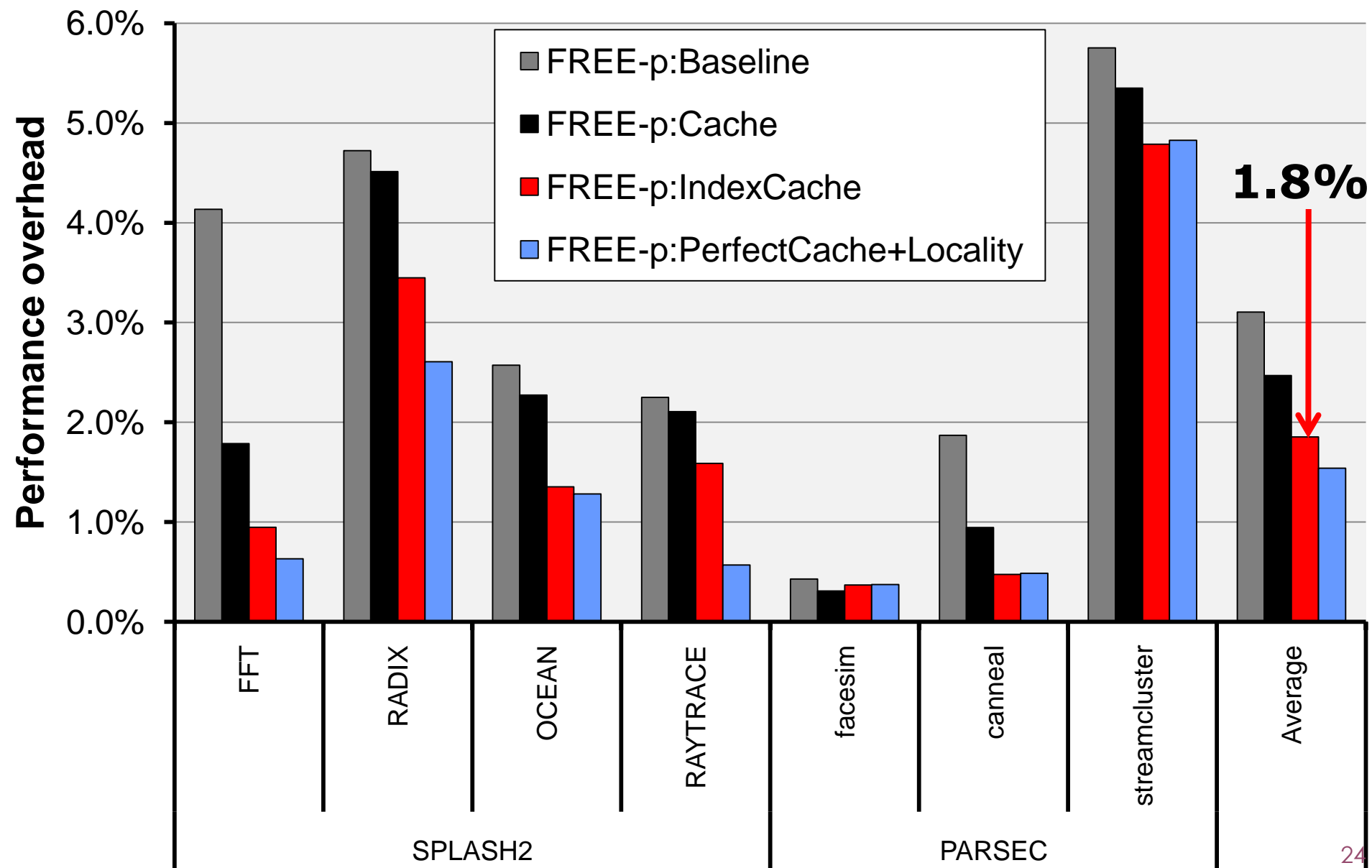
After 7.3 Years



After 7.3 Years



After 7.3 Years



Performance Impact (Cont'd)

- **After 8 years**
 - Index Cache: 3.5~6.7% penalty
- **After 8.8 years**
 - Simple caching doesn't work any more
 - > 30% penalty
 - Index Cache: 10.7~13% penalty

Conclusions

- **FREE-p combines FR and multi-tiered ECC**
 - Protect against both Hard AND Soft errors
 - 11.5% longer lifetime over ECP6
 - Less than 2% performance degradation, even at 7 years
 - 12.5% storage overhead (same as current DRAM protection)
- **Everything is implemented at the memory controller**
 - End-to-end protection
 - Hard and soft errors in the cell array
 - Errors on the periphery, wires, packaging, ...
 - System designers determine protection level
 - Can be extended to chipkill-correct
 - Simple and cheap (commodity) NVRAM devices

FREE-p: Protecting Non-Volatile Memory against both Hard and Soft Errors

Doe Hyun Yoon[†]

Naveen Muralimanohar[‡]

Jichuan Chang[‡]

Parthasarathy Ranganathan[‡]

Norman P. Jouppi[‡]

Mattan Erez[†]



[†]Electrical and Computer Engineering
The University of Texas at Austin

[‡] Intelligent Infrastructure Lab.
Hewlett-Packard Labs.