Flash Correct-and-Refresh

Retention-Aware Error Management for Increased Flash Memory Lifetime

Yu Cai[1] Gulay Yalcin[2] Onur Mutlu[1] Erich F. Haratsch[3]
Adrian Cristal[2] Osman S. Unsal[2] Ken Mai[1]

[1] Carnegie Mellon University
[2] Barcelona Supercomputing Center
[3] LSI Corporation

SAFARI

Carnegie Mellon

Executive Summary

- NAND flash memory has low endurance: a flash cell dies after 3k P/E cycles vs. 50k desired → Major scaling challenge for flash memory
- Flash error rate increases exponentially over flash lifetime
- Problem: Stronger error correction codes (ECC) are ineffective and undesirable for improving flash lifetime due to
  - diminishing returns on lifetime with increased correction strength
  - prohibitively high power, area, latency overheads
- Our Goal: Develop techniques to tolerate high error rates w/o strong ECC
- Observation: Retention errors are the dominant errors in MLC NAND flash
  - flash cell loses charge over time; retention errors increase as cell gets worn out
- Solution: Flash Correct-and-Refresh (FCR)
  - Periodically read, correct, and reprogram (in place) or remap each flash page before it accumulates more errors than can be corrected by simple ECC
  - Adapt "refresh" rate to the severity of retention errors (i.e., # of P/E cycles)
- Results: FCR improves flash memory lifetime by 46X with no hardware changes and low energy overhead; outperforms strong ECCs

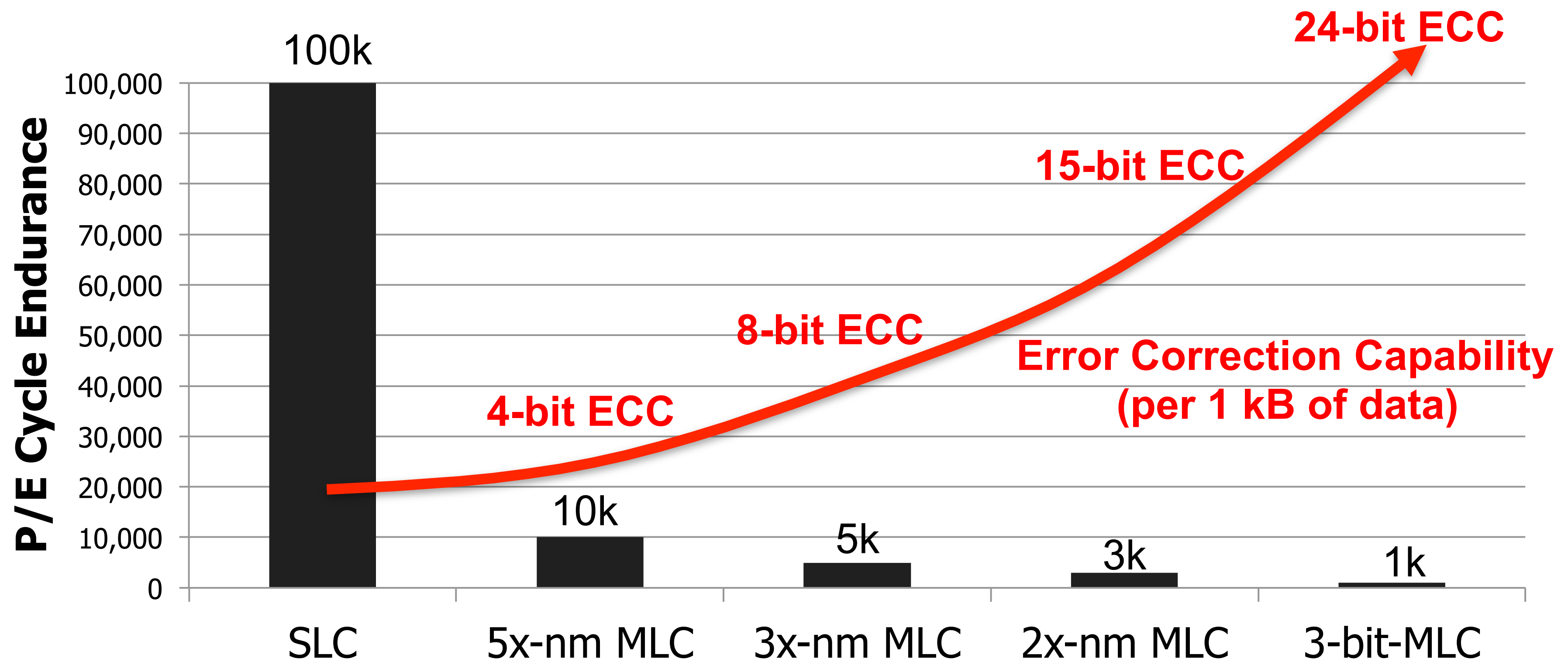
Outline

- Executive Summary
- The Problem: Limited Flash Memory Endurance/Lifetime
- Error and ECC Analysis for Flash Memory
- Flash Correct and Refresh Techniques (FCR)
- Evaluation
- Conclusions

Problem: Limited Endurance of Flash Memory

- NAND flash has limited endurance
  - A cell can tolerate a small number of Program/Erase (P/E) cycles
  - 3x-nm flash with 2 bits/cell → 3K P/E cycles
- Enterprise data storage requirements demand very high endurance
  - >50K P/E cycles (10 full disk writes per day for 3-5 years)
- Continued process scaling and more bits per cell will reduce flash endurance
- One potential solution: stronger error correction codes (ECC)
  - Stronger ECC not effective enough and inefficient

Decreasing Endurance with Flash Scaling



Ariel Maislos, "A New Era in Embedded Flash Memory", Flash Summit 2011 (Anobit)

- Endurance of flash memory decreasing with scaling and multi-level cells
- Error correction capability required to guarantee storage-class reliability (UBER < 10^{-15}) is increasing exponentially to reach *less* endurance

UBER: Uncorrectable bit error rate. Fraction of erroneous bits after error correction.

The Problem with Stronger Error Correction

- Stronger ECC detects and corrects more raw bit errors → increases P/E cycles endured

- Two shortcomings of stronger ECC:

  1. High implementation complexity

     → Power and area overheads increase super-linearly, but correction capability increases sub-linearly with ECC strength

  2. Diminishing returns on flash lifetime improvement

     → Raw bit error rate increases exponentially with P/E cycles, but correction capability increases sub-linearly with ECC strength

Outline

- Executive Summary
- The Problem: Limited Flash Memory Endurance/Lifetime
- Error and ECC Analysis for Flash Memory
- Flash Correct and Refresh Techniques (FCR)
- Evaluation
- Conclusions

Methodology: Error and ECC Analysis

- Characterized errors and error rates of 3x-nm MLC NAND flash using an experimental FPGA-based flash platform
  - Cai et al., "Error Patterns in MLC NAND Flash Memory: Measurement, Characterization, and Analysis," DATE 2012.
- Quantified Raw Bit Error Rate (RBER) at a given P/E cycle
  - Raw Bit Error Rate: Fraction of erroneous bits without any correction
- Quantified error correction capability (and area and power consumption) of various BCH-code implementations
  - Identified how much RBER each code can tolerate

    → how many P/E cycles (flash lifetime) each code can sustain

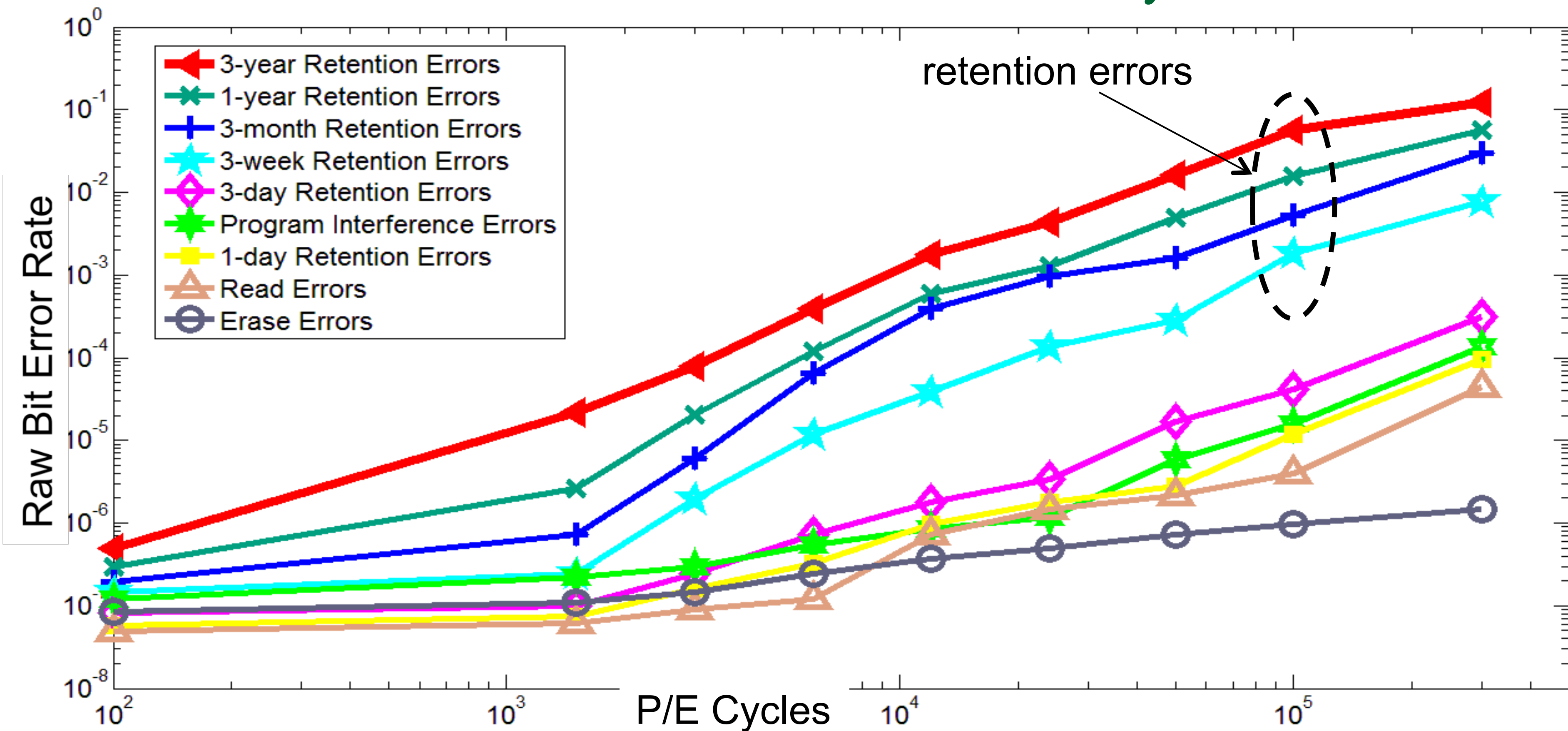
NAND Flash Error Types

- Four types of errors [Cai+, DATE 2012]

- Caused by common flash operations
  - Read errors
  - Erase errors
  - Program (interference) errors

- Caused by flash cell losing charge over time
  - Retention errors
    - Whether an error happens depends on required retention time
    - Especially problematic in MLC flash because voltage threshold window to determine stored value is smaller

Observations: Flash Error Analysis



- Raw bit error rate increases exponentially with P/E cycles
- Retention errors are dominant (>99% for 1-year ret. time)
- Retention errors increase with retention time requirement

Methodology: Error and ECC Analysis

- Characterized errors and error rates of 3x-nm MLC NAND flash using an experimental FPGA-based flash platform
  - Cai et al., "Error Patterns in MLC NAND Flash Memory: Measurement, Characterization, and Analysis," DATE 2012.
- Quantified Raw Bit Error Rate (RBER) at a given P/E cycle
  - Raw Bit Error Rate: Fraction of erroneous bits without any correction
- Quantified error correction capability (and area and power consumption) of various BCH-code implementations
  - Identified how much RBER each code can tolerate
    → how many P/E cycles (flash lifetime) each code can sustain

ECC Strength Analysis

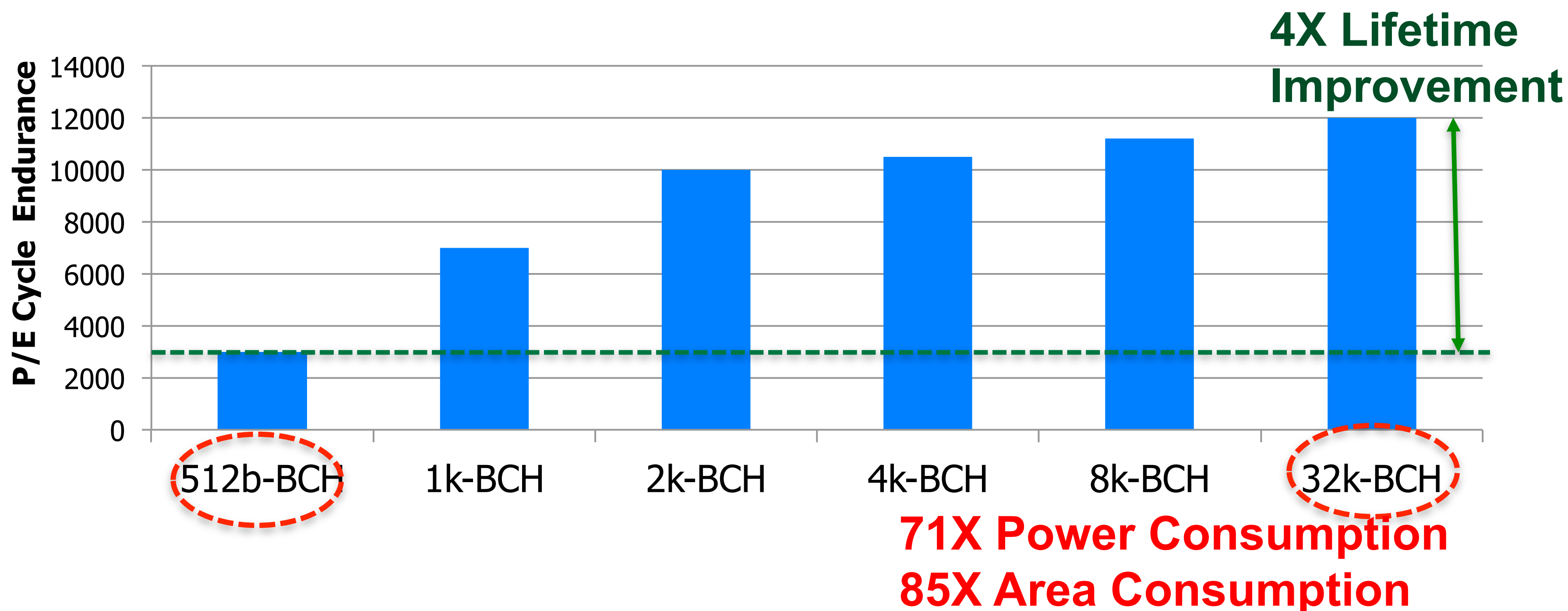
Error correction capability increases sub-linearly

Power and area overheads increase super-linearly

Code length (n)	Correctable Errors (t)	Acceptable Raw BER	Norm. Power	Norm. Area
512	7	1.0×10^{-4} (1x)	1	1
1024	12	4.0×10^{-4} (4x)	2	2.1
2048	22	1.0×10^{-3} (10x)	4.1	3.9
4096	40	1.7×10^{-3} (17x)	8.6	10.3
8192	74	2.2×10^{-3} (22x)	17.8	21.3
32768	259	2.6×10^{-3} (26x)	71	85

Resulting Flash Lifetime with Strong ECC

- Lifetime improvement comparison of various BCH codes



Strong ECC is very inefficient at improving lifetime

Our Goal

Develop new techniques
to improve flash lifetime
without relying on stronger ECC

Outline

- Executive Summary
- The Problem: Limited Flash Memory Endurance/Lifetime
- Error and ECC Analysis for Flash Memory
- Flash Correct and Refresh Techniques (FCR)
- Evaluation
- Conclusions

Flash Correct-and-Refresh (FCR)

- Key Observations:
  - Retention errors are the dominant source of errors in flash memory [Cai+ DATE 2012][Tanakamaru+ ISSCC 2011]
    → limit flash lifetime as they increase over time
  - Retention errors can be corrected by "refreshing" each flash page periodically

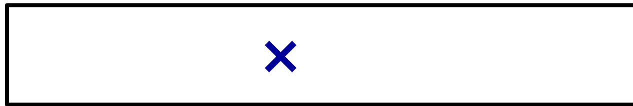
- Key Idea:
  - Periodically read each flash page,
  - Correct its errors using "weak" ECC, and
  - Either remap it to a new physical page or reprogram it in-place,
  - Before the page accumulates more errors than ECC-correctable
  - Optimization: Adapt refresh rate to endured P/E cycles

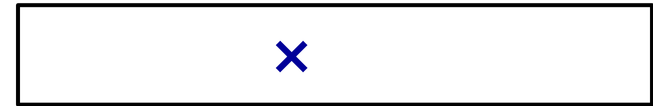
FCR Intuition

Errors with
No refresh

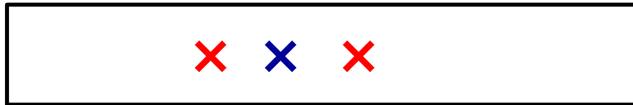
Errors with
Periodic refresh

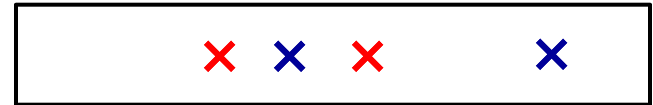
Program Page



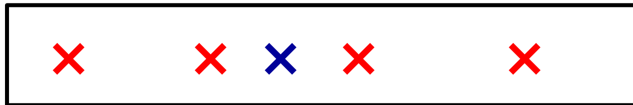


After time T

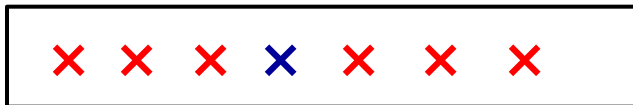




After time 2T



After time 3T



× **Retention Error** × **Program Error**

FCR: Two Key Questions

- How to refresh?
  - Remap a page to another one
  - Reprogram a page (in-place)
  - Hybrid of remap and reprogram
- When to refresh?
  - Fixed period
  - Adapt the period to retention error severity

Outline

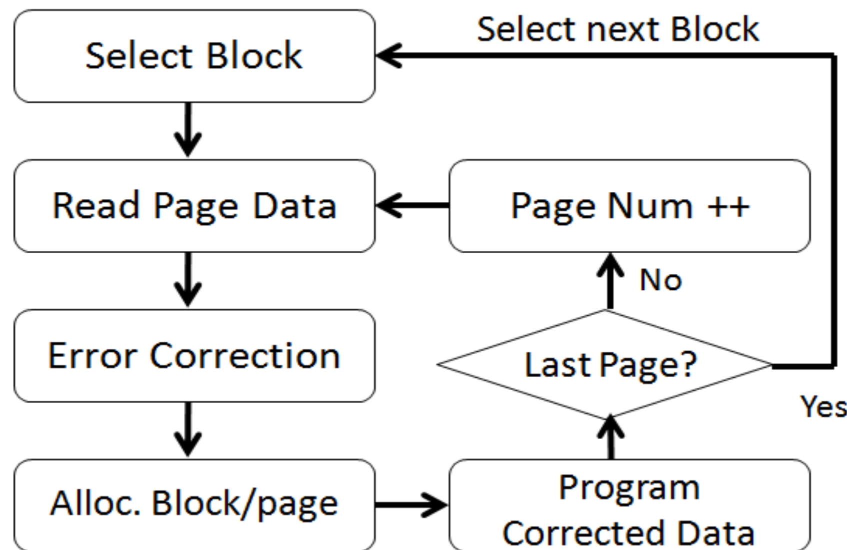
- Executive Summary
- The Problem: Limited Flash Memory Endurance/Lifetime
- Error and ECC Analysis for Flash Memory
- Flash Correct and Refresh Techniques (FCR)
  1. Remapping based FCR
  2. Hybrid Reprogramming and Remapping based FCR
  3. Adaptive-Rate FCR
- Evaluation
- Conclusions

Outline

- Executive Summary
- The Problem: Limited Flash Memory Endurance/Lifetime
- Error and ECC Analysis for Flash Memory
- Flash Correct and Refresh Techniques (FCR)
  1. Remapping based FCR
  2. Hybrid Reprogramming and Remapping based FCR
  3. Adaptive-Rate FCR
- Evaluation
- Conclusions

Remapping Based FCR

- Idea: Periodically remap each page to a different physical page (after correcting errors)
  - Also [Pan et al., HPCA 2012]
  - FTL already has support for changing logical → physical flash block/page mappings
  - Deallocated block is erased by garbage collector



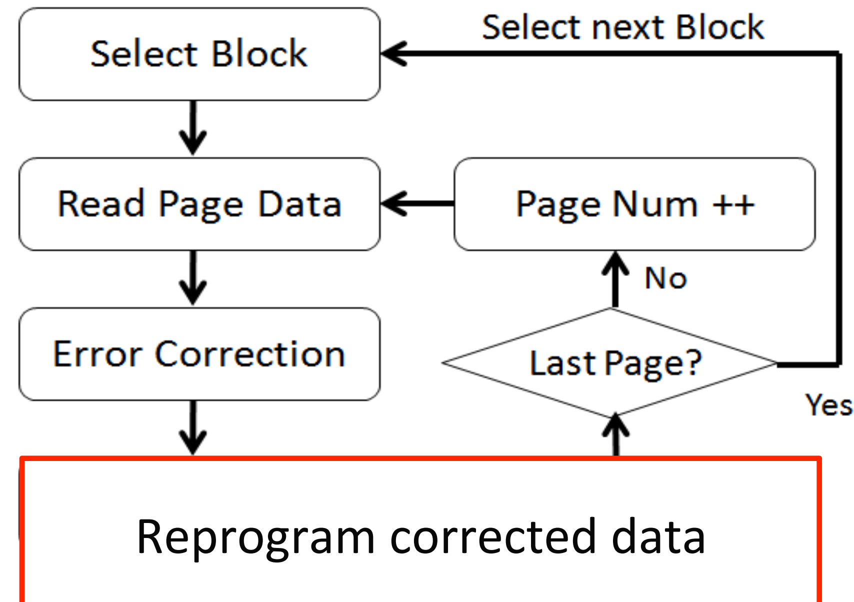
- Problem: Causes additional erase operations → more wearout
  - Bad for read-intensive workloads (few erases really needed)
  - Lifetime degrades for such workloads (see paper)

Outline

- Executive Summary
- The Problem: Limited Flash Memory Endurance/Lifetime
- Error and ECC Analysis for Flash Memory
- Flash Correct and Refresh Techniques (FCR)
  1. Remapping based FCR
  2. Hybrid Reprogramming and Remapping based FCR
  3. Adaptive-Rate FCR
- Evaluation
- Conclusions

In-Place Reprogramming Based FCR

- Idea: Periodically reprogram (in-place) each physical page (after correcting errors)
  - Flash programming techniques (ISPP) can correct retention errors in-place by recharging flash cells

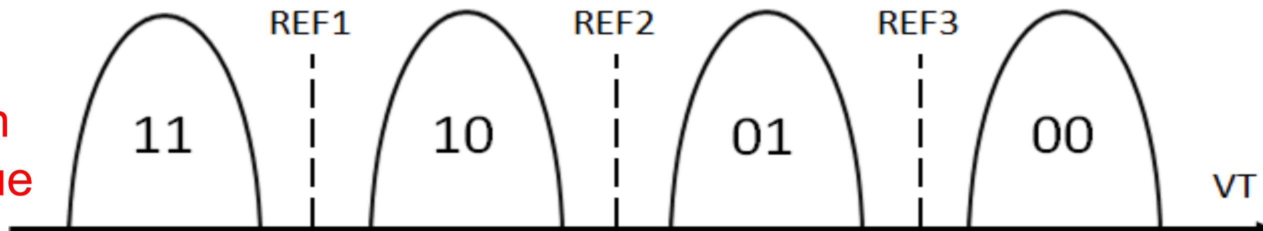


- Problem: Program errors accumulate on the same page → may not be correctable by ECC after some time

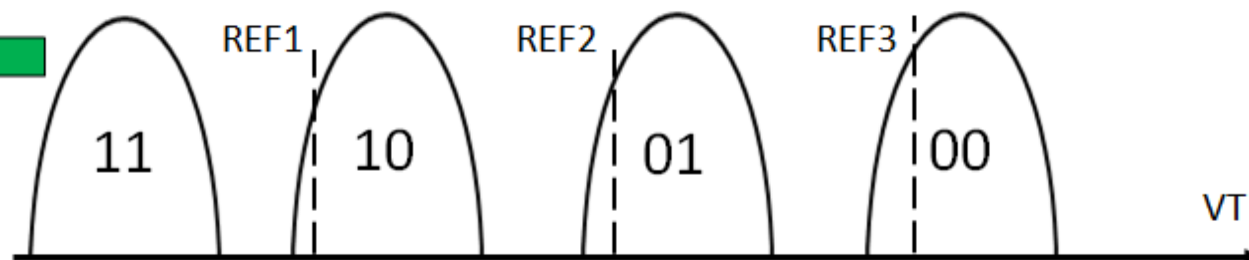
In-Place Reprogramming of Flash Cells

Floating Gate

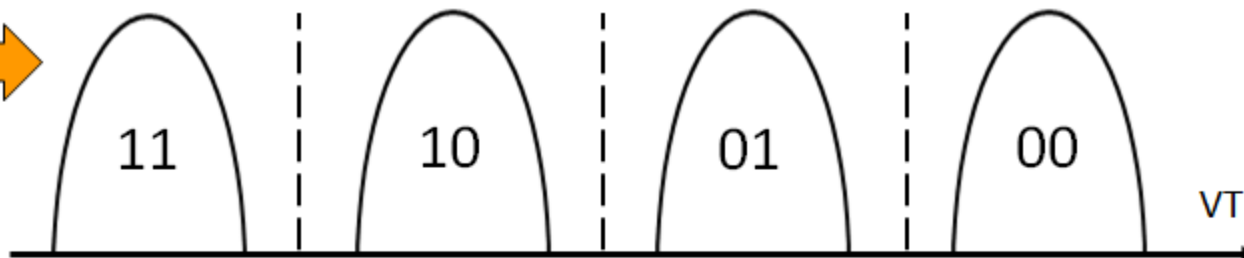
Floating Gate Voltage Distribution for each Stored Value



Retention errors are caused by cell voltage shifting to the left



ISPP moves cell voltage to the right; fixes retention errors



- Pro: No remapping needed → no additional erase operations
- Con: Increases the occurrence of program errors

Program Errors in Flash Memory

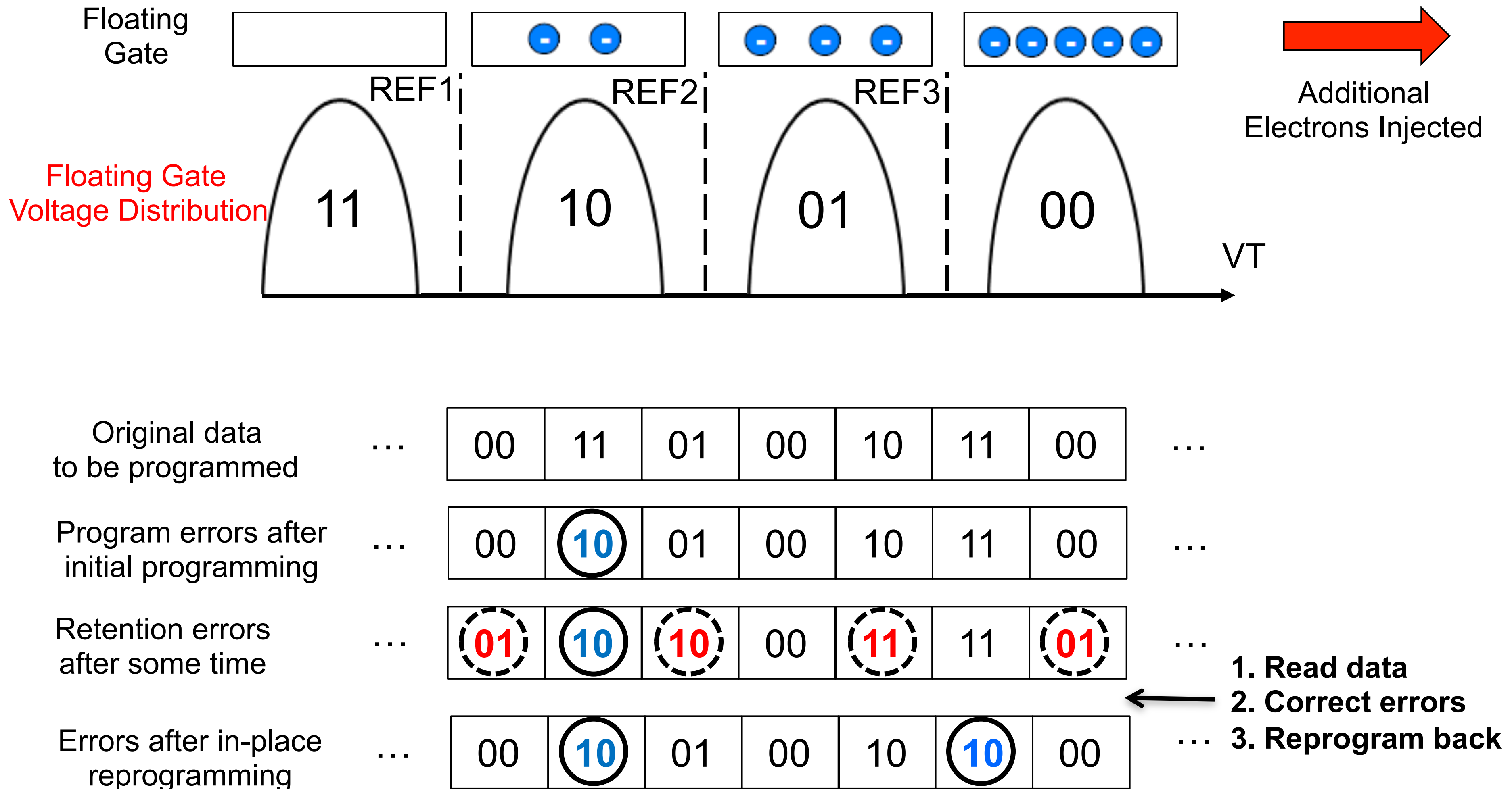
- When a cell is being programmed, voltage level of a neighboring cell changes (unintentionally) due to parasitic capacitance coupling

  → can change the data value stored

- Also called program interference error

- Program interference causes neighboring cell voltage to shift to the right

Problem with In-Place Reprogramming



Problem: Program errors can accumulate over time

Hybrid Reprogramming/Remapping Based FCR

- Idea:
  - Monitor the count of right-shift errors (after error correction)
  - If count < threshold, in-place reprogram the page
  - Else, remap the page to a new page

- Observation:
  - Program errors much less frequent than retention errors → Remapping happens only infrequently

- Benefit:
  - Hybrid FCR greatly reduces erase operations due to remapping

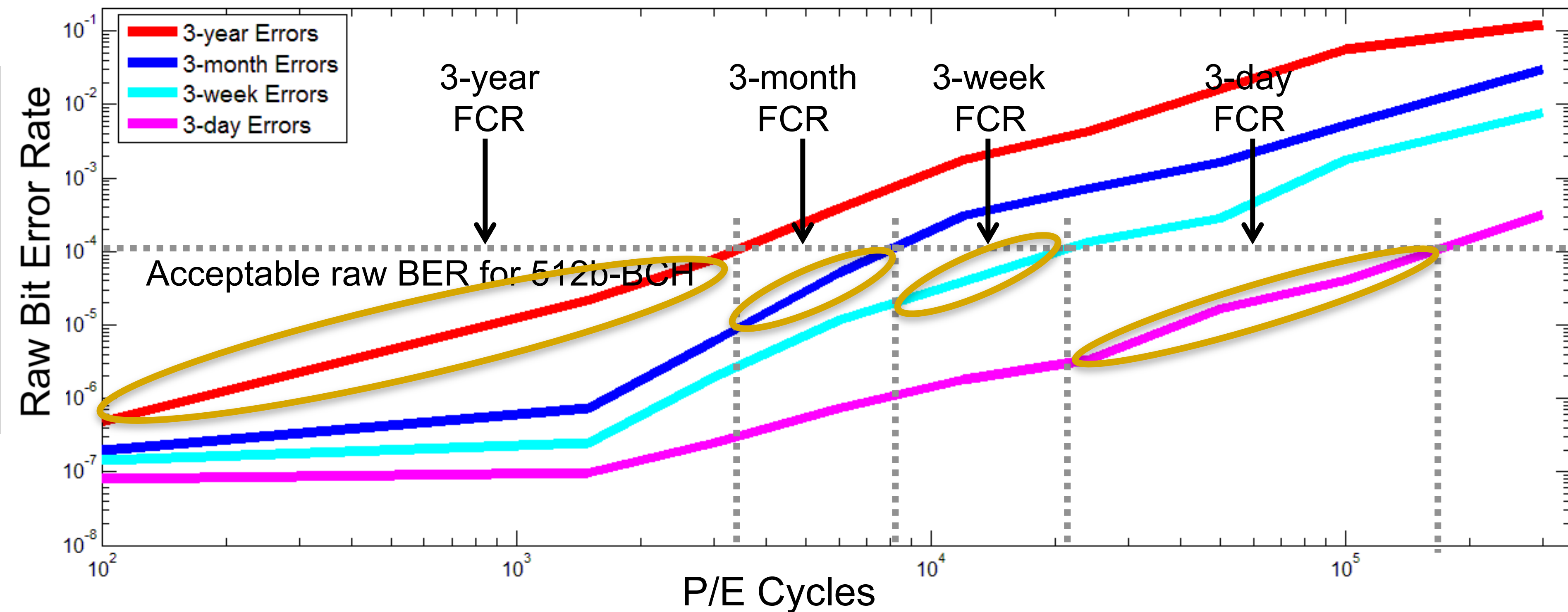
Outline

- Executive Summary
- The Problem: Limited Flash Memory Endurance/Lifetime
- Error and ECC Analysis for Flash Memory
- Flash Correct and Refresh Techniques (FCR)
    1. Remapping based FCR
    2. Hybrid Reprogramming and Remapping based FCR
    3. Adaptive-Rate FCR
- Evaluation
- Conclusions

Adaptive-Rate FCR

- Observation:
  - Retention error rate strongly depends on the P/E cycles a flash page endured so far
  - No need to refresh frequently (at all) early in flash lifetime
- Idea:
  - Adapt the refresh rate to the P/E cycles endured by each page
  - Increase refresh rate gradually with increasing P/E cycles
- Benefits:
  - Reduces overhead of refresh operations
  - Can use existing FTL mechanisms that keep track of P/E cycles

Adaptive-Rate FCR (Example)



Select refresh frequency such that error rate is below acceptable rate

Outline

- Executive Summary
- The Problem: Limited Flash Memory Endurance/Lifetime
- Error and ECC Analysis for Flash Memory
- Flash Correct and Refresh Techniques (FCR)
  1. Remapping based FCR
  2. Hybrid Reprogramming and Remapping based FCR
  3. Adaptive-Rate FCR
- Evaluation
- Conclusions

FCR: Other Considerations

- Implementation cost
  - No hardware changes
  - FTL software/firmware needs modification
- Response time impact
  - FCR not as frequent as DRAM refresh; low impact
- Adaptation to variations in retention error rate
  - Adapt refresh rate based on, e.g., temperature **[Liu+ ISCA 2012]**
- FCR requires power
  - Enterprise storage systems typically powered on

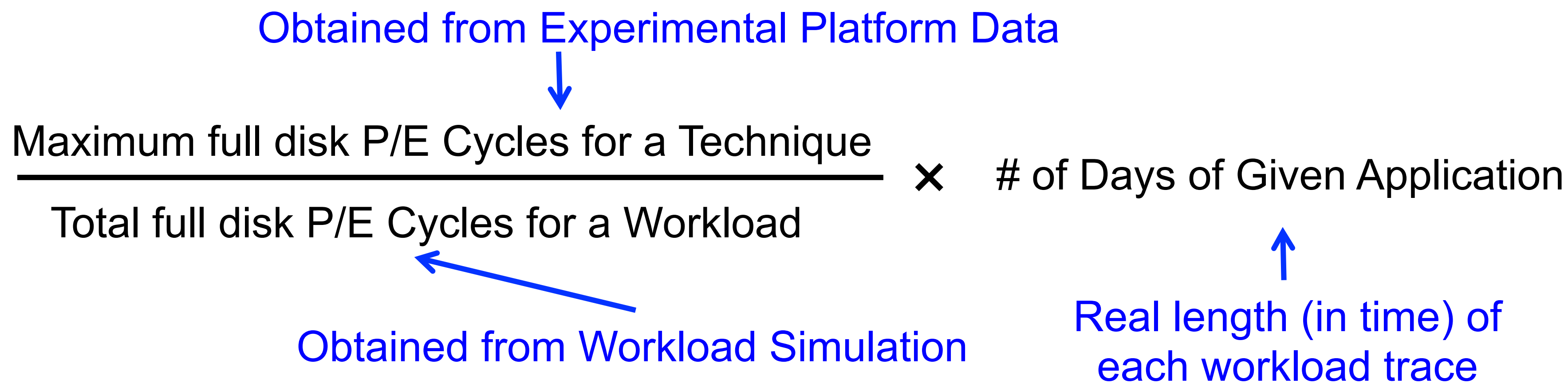
Outline

- Executive Summary
- The Problem: Limited Flash Memory Endurance/Lifetime
- Error and ECC Analysis for Flash Memory
- Flash Correct and Refresh Techniques (FCR)
- Evaluation
- Conclusions

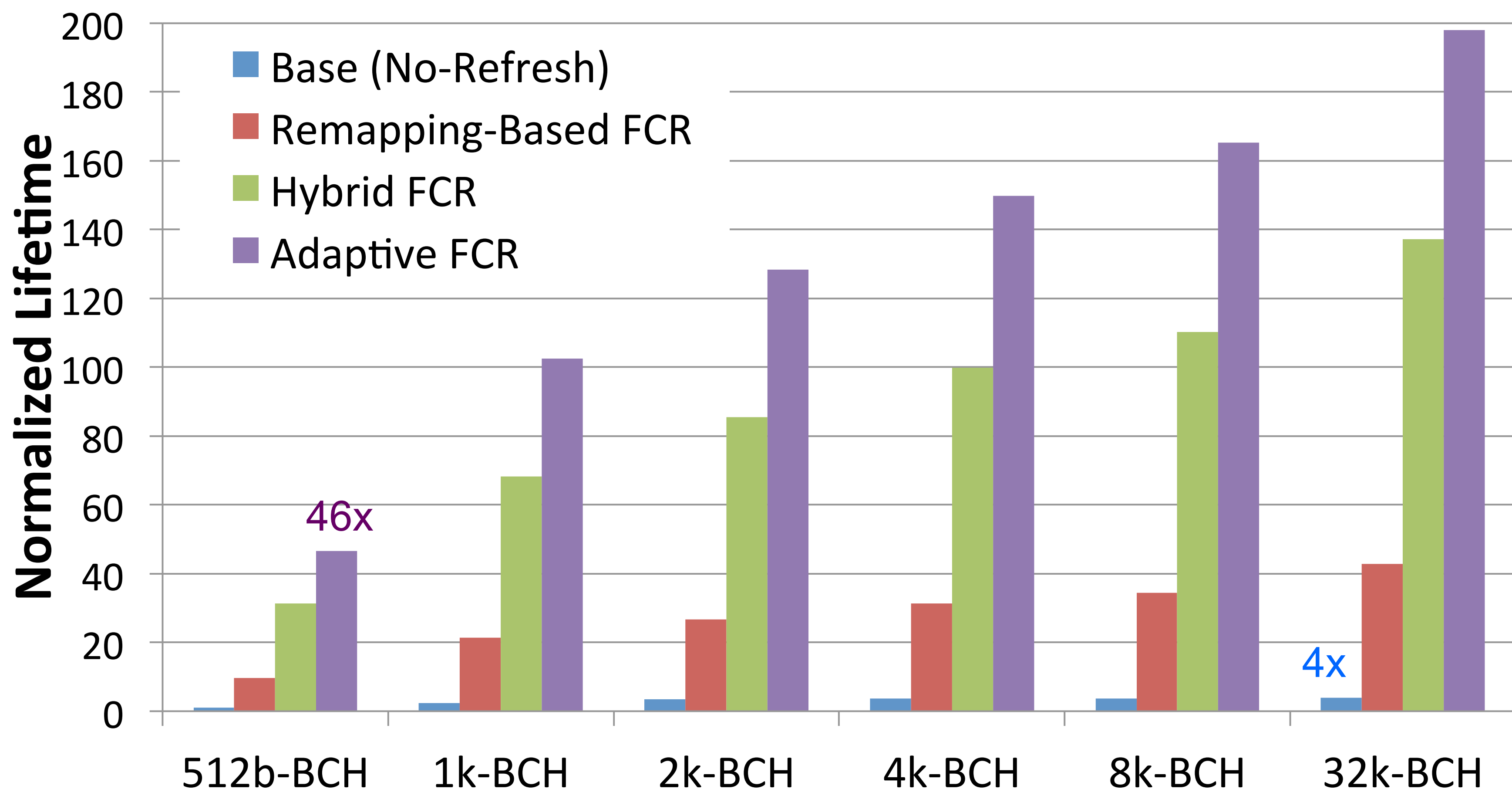
Evaluation Methodology

- Experimental flash platform to obtain error rates at different P/E cycles **[Cai+ DATE 2012]**
- Simulation framework to obtain P/E cycles of real workloads: DiskSim with SSD extensions
- Simulated system: 256GB flash, 4 channels, 8 chips/channel, 8K blocks/chip, 128 pages/block, 8KB pages
- Workloads
  - File system applications, databases, web search
  - Categories: Write-heavy, read-heavy, balanced
- Evaluation metrics
  - Lifetime (extrapolated)
  - Energy overhead, P/E cycle overhead

Extrapolated Lifetime



Normalized Flash Memory Lifetime

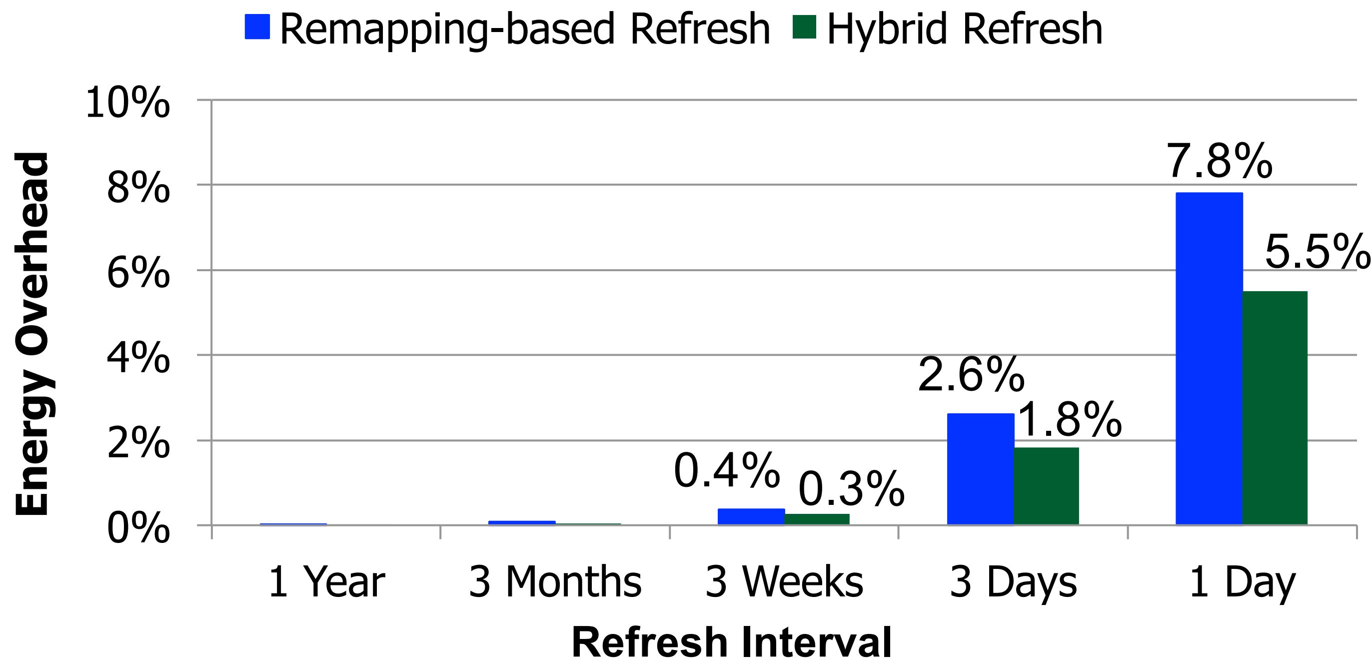


Lifetime of FCR much higher than lifetime of stronger ECC

Lifetime Evaluation Takeaways

- Significant average lifetime improvement over no refresh
  - Adaptive-rate FCR: 46X
  - Hybrid reprogramming/remapping based FCR: 31X
  - Remapping based FCR: 9X
- FCR lifetime improvement larger than that of stronger ECC
  - 46X vs. 4X with 32-kbit ECC (over 512-bit ECC)
  - FCR is less complex and less costly than stronger ECC
- Lifetime on all workloads improves with Hybrid FCR
  - Remapping based FCR can degrade lifetime on read-heavy WL
  - Lifetime improvement highest in write-heavy workloads

Energy Overhead



- Adaptive-rate refresh: <1.8% energy increase until daily refresh is triggered

Overhead of Additional Erases

- Additional erases happen due to remapping of pages

- Low (2%-20%) for write intensive workloads
- High (up to 10X) for read-intensive workloads

- Improved P/E cycle lifetime of all workloads largely outweighs the additional P/E cycles due to remapping

More Results in the Paper

- Detailed workload analysis
- Effect of refresh rate

Outline

- Executive Summary
- The Problem: Limited Flash Memory Endurance/Lifetime
- Error and ECC Analysis for Flash Memory
- Flash Correct and Refresh Techniques (FCR)
- Evaluation
- Conclusions

Conclusion

- NAND flash memory lifetime is limited due to uncorrectable errors, which increase over lifetime (P/E cycles)
- Observation: Dominant source of errors in flash memory is retention errors → retention error rate limits lifetime
- Flash Correct-and-Refresh (FCR) techniques reduce retention error rate to improve flash lifetime
  - Periodically read, correct, and remap or reprogram each page before it accumulates more errors than can be corrected
  - Adapt refresh period to the severity of errors
- FCR improves flash lifetime by 46X at no hardware cost
  - More effective and efficient than stronger ECC
  - Can enable better flash memory scaling

Thank You.

Flash Correct-and-Refresh

Retention-Aware Error Management for Increased Flash Memory Lifetime

Yu Cai[1] Gulay Yalcin[2] Onur Mutlu[1] Erich F. Haratsch[3]
Adrian Cristal[2] Osman S. Unsal[2] Ken Mai[1]

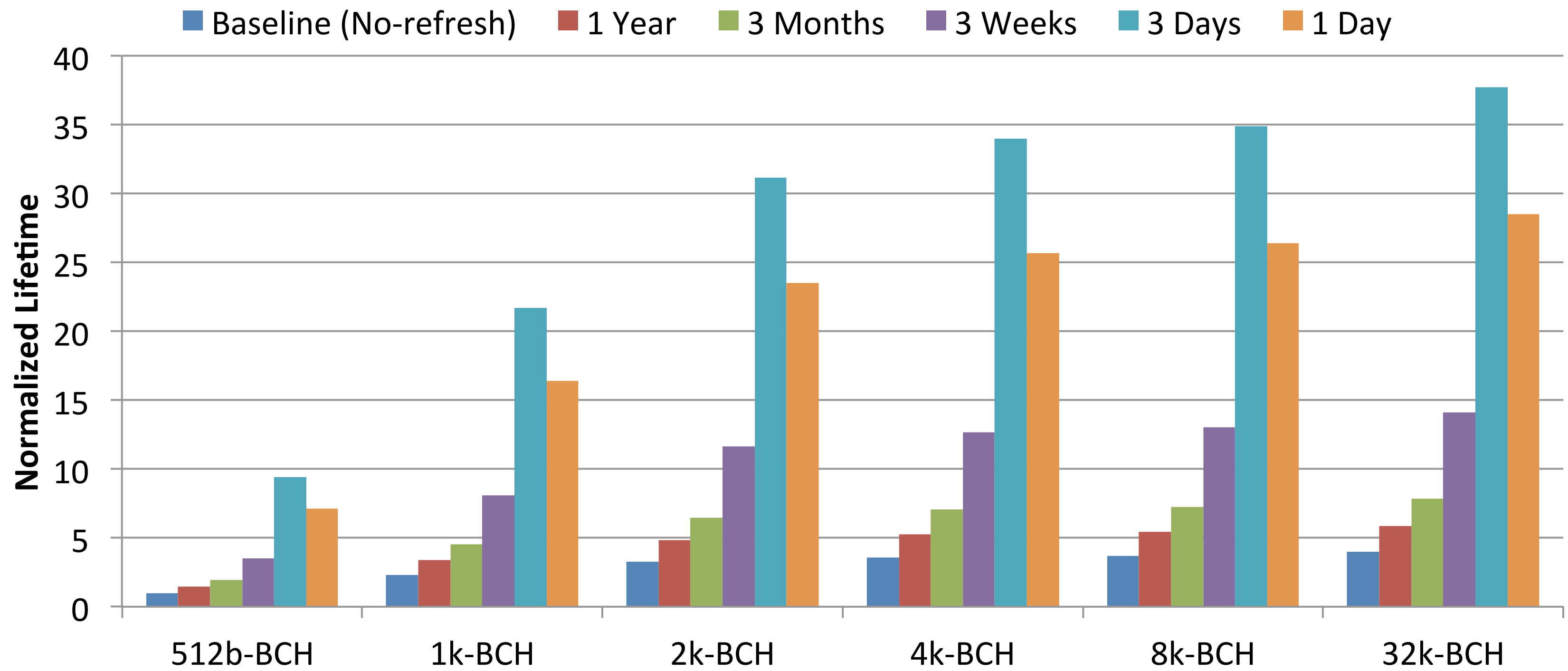
[1] Carnegie Mellon University
[2] Barcelona Supercomputing Center
[3] LSI Corporation

SAFARI

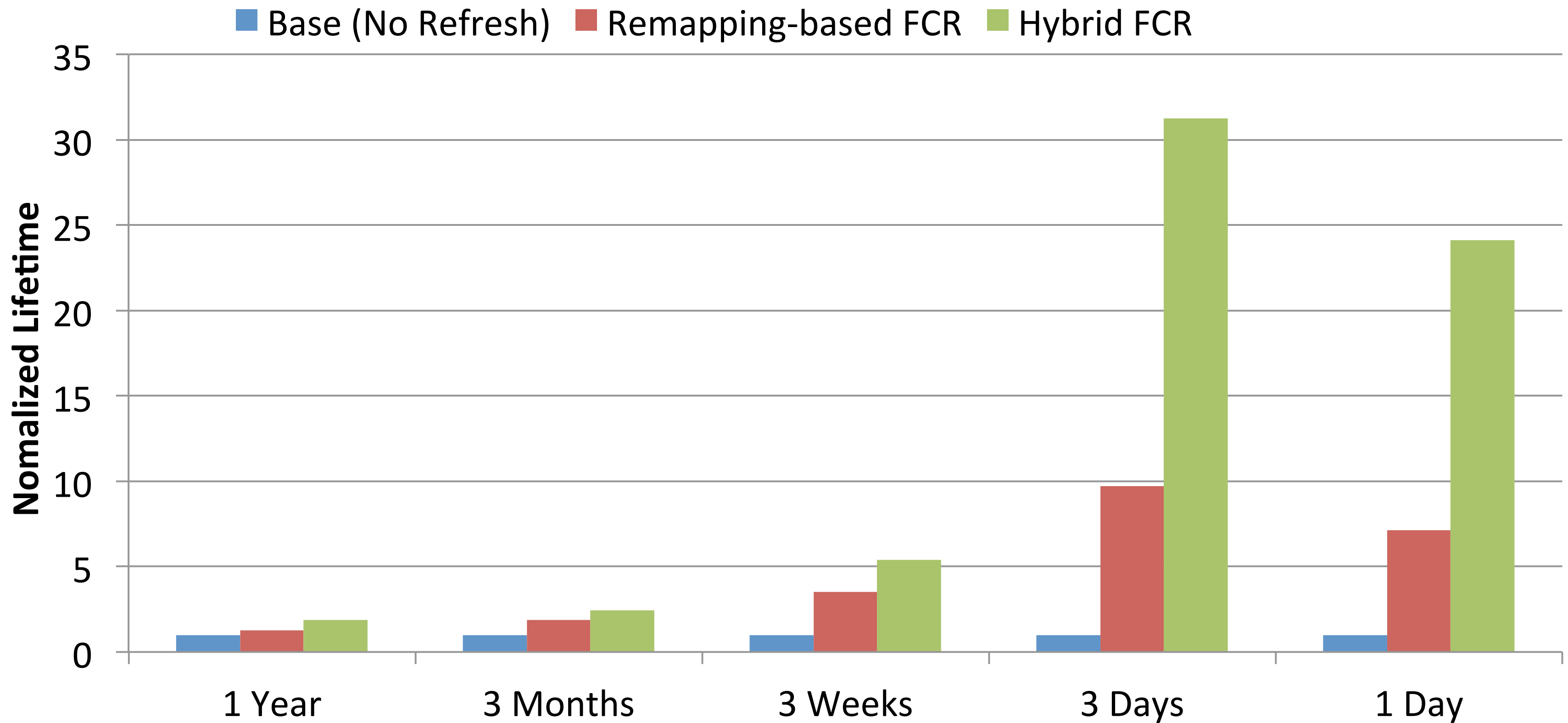
Carnegie Mellon

Backup Slides

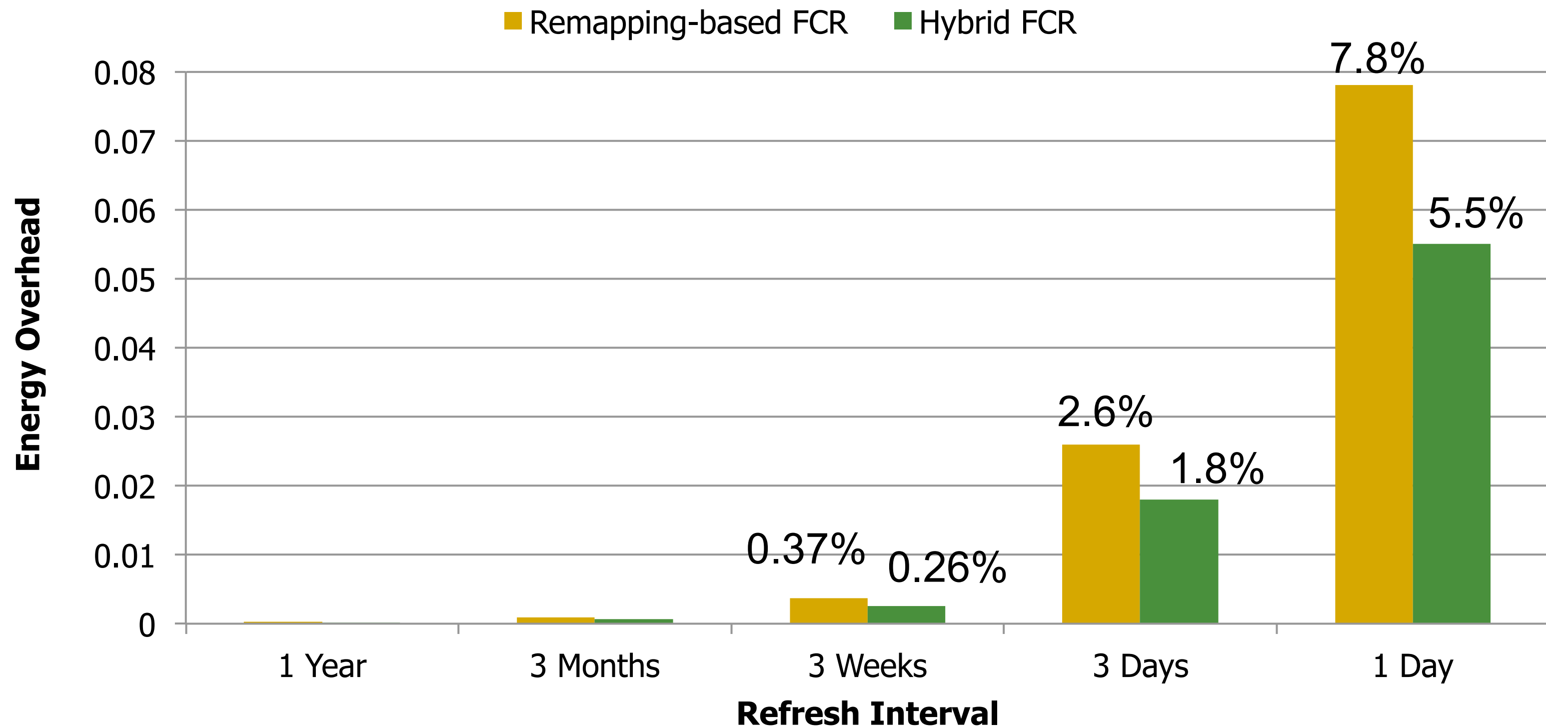
Effect of Refresh Rate on Lifetime



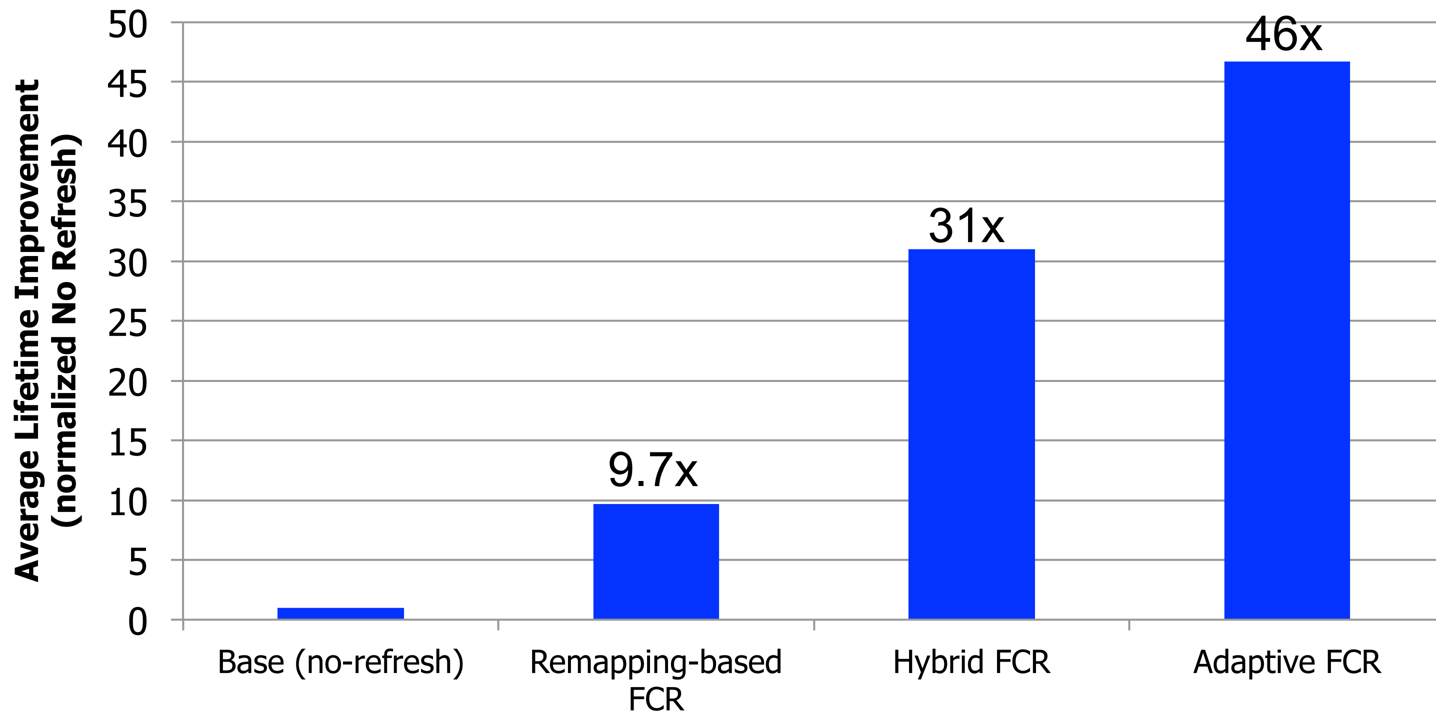
Lifetime: Remapping vs. Hybrid FCR



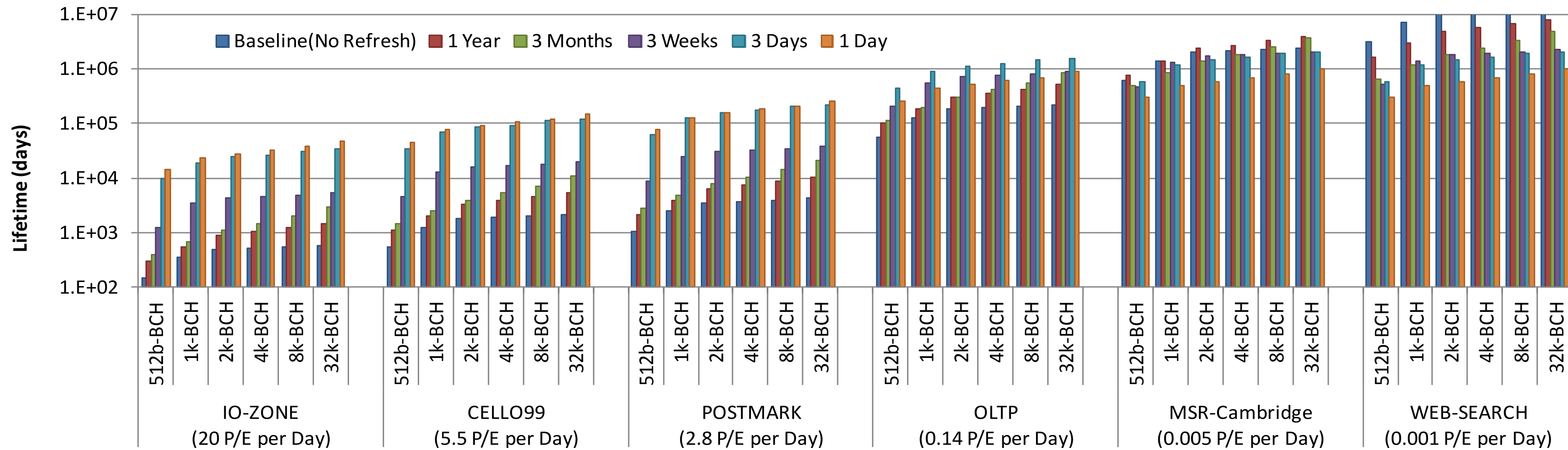
Energy Overhead



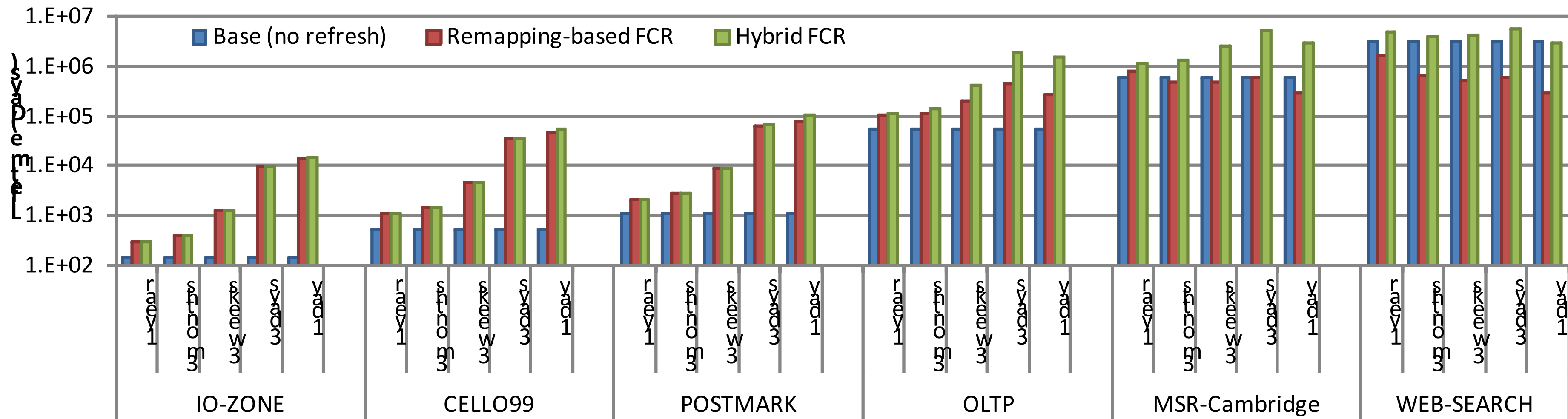
Average Lifetime Improvement



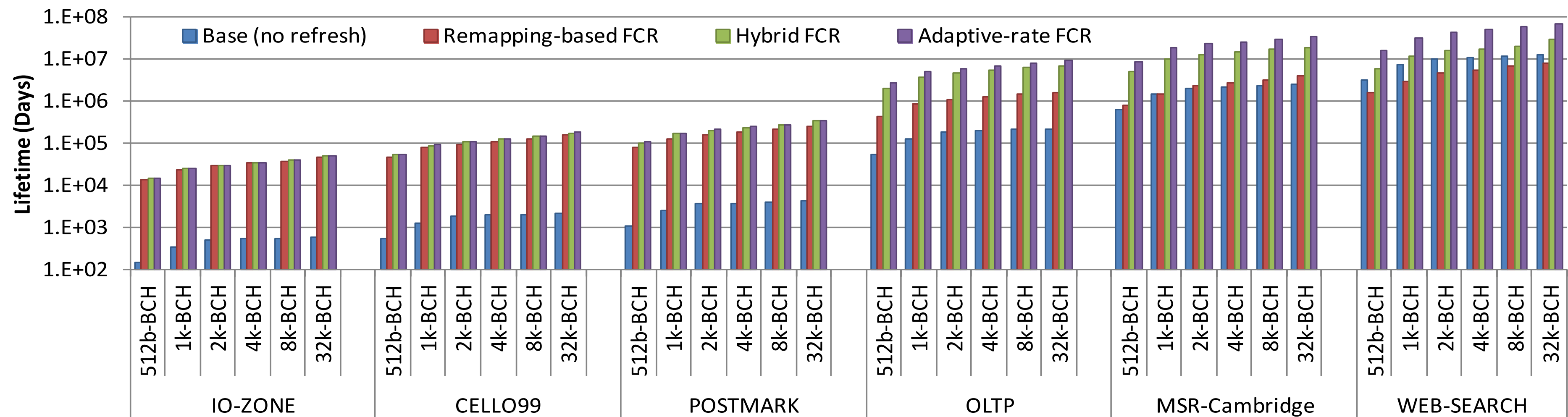
Individual Workloads: Remapping-Based FCR



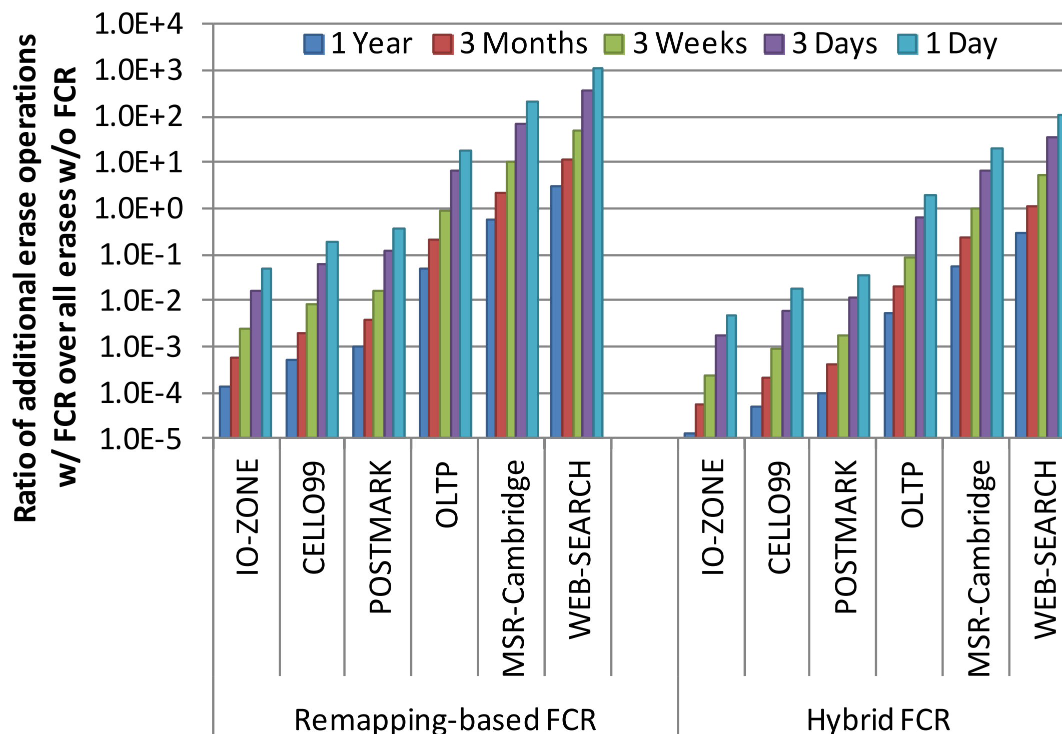
Individual Workloads: Hybrid FCR



Individual Workloads: Adaptive-Rate FCR

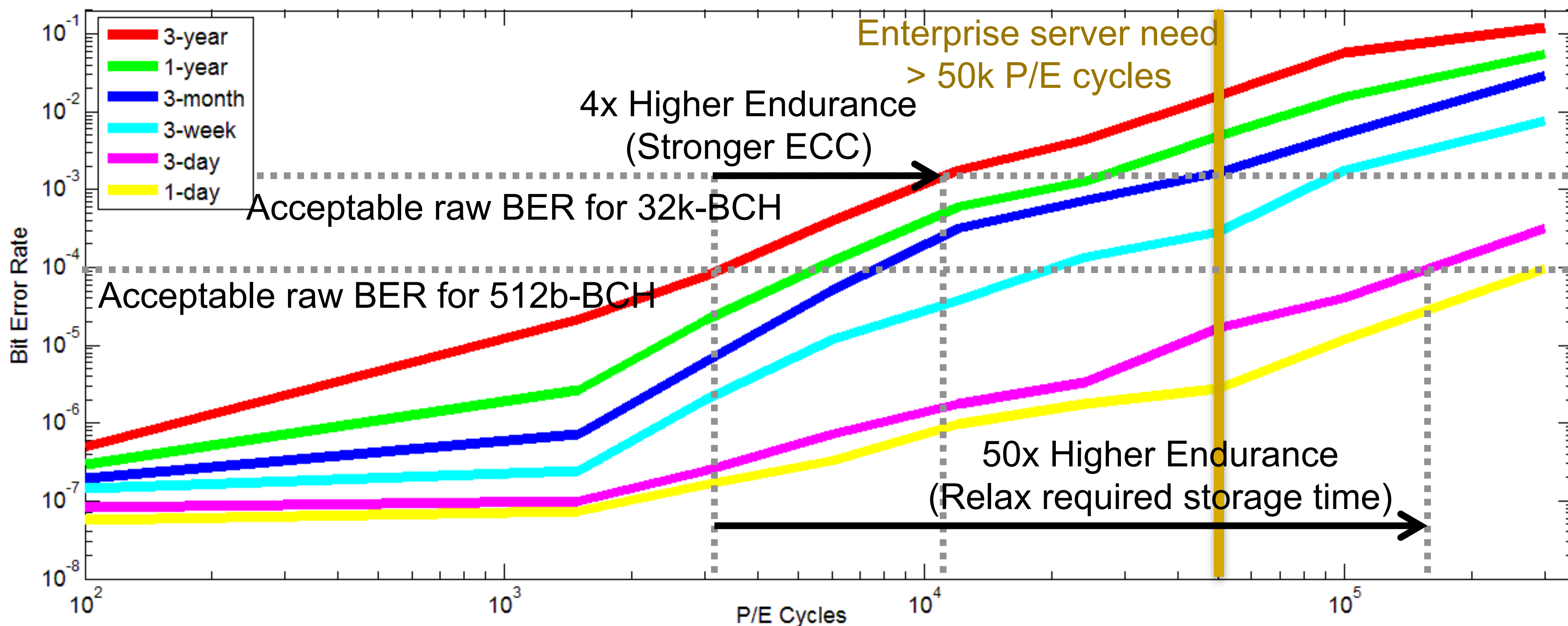


P/E Cycle Overhead



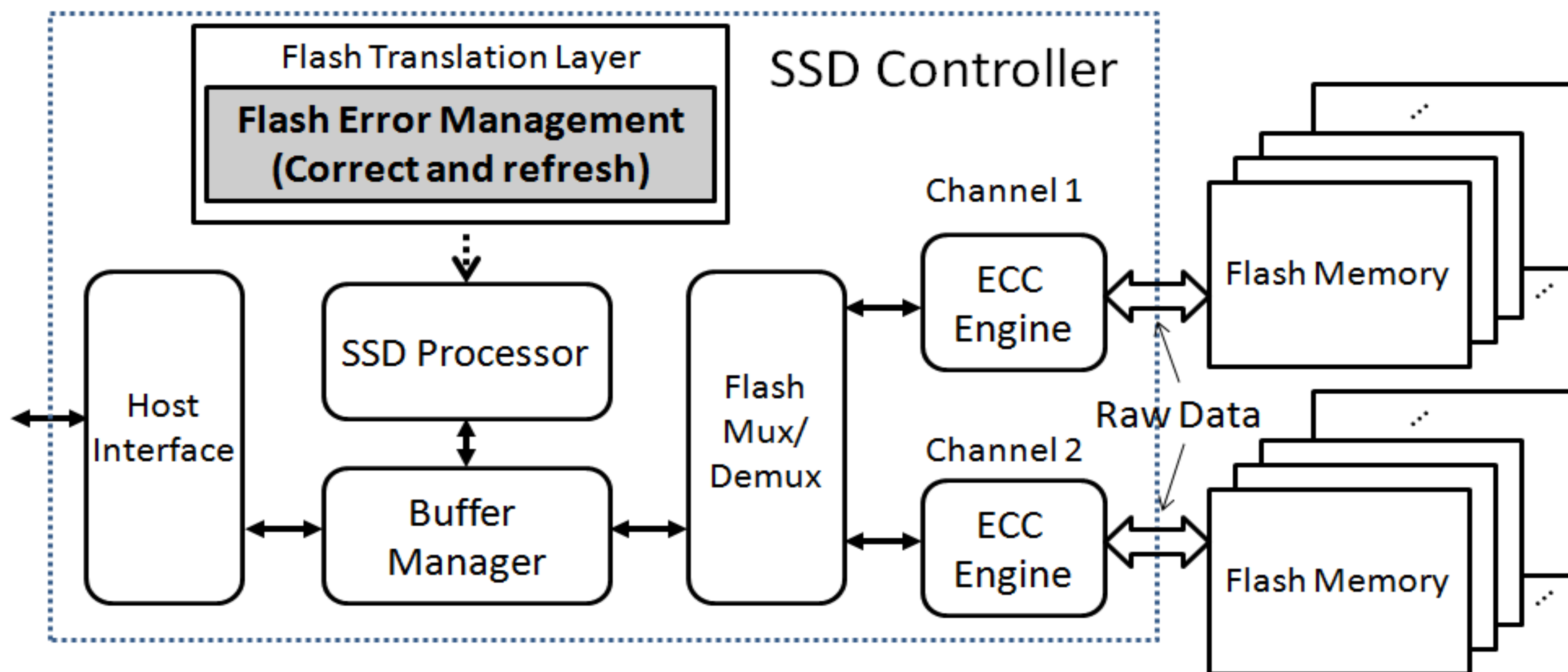
- P/E cycle overhead of hybrid FCR is lower than that of remapping-based FCR
- P/E cycle overhead for write-intensive applications is low
  - Remapping-based FCR (20%), Hybrid FCR (2%)
- Read-intensive applications have higher P/E cycle overhead

Motivation for Refresh: A Different Way



- NAND flash endurance can be increased via
  - Stronger error correction codes (4x)
  - Tradeoff guaranteed storage time for one write for high endurance (> 50x)

FTL Implementation



- FCR can be implemented just as a module in FTL software

Flash Cells Can Be Reprogrammed In-Place

- Observations:
  - Retention errors occur due to loss of charge
  - Simply recharging the cells can correct the retention errors
  - Flash programming mechanisms can accomplish this recharging

- ISPP (Incremental Step Pulse Programming)
  - Iterative programming mechanism that increases the voltage level of a flash cell step by step
  - After each step, voltage level compared to desired voltage threshold
  - Can inject more electrons but cannot remove electrons