# The Nibbles and Bits of SSD Data Integrity

Earl T. Cohen
Flash Components Division
LSI Corporation

# What is Data Integrity?

1. Maintaining and assuring the accuracy and consistency of data over its entire life-cycle.
    http://en.wikipedia.org/wiki/Data_integrity

2. Don’t “foul” up the data!
    Earl

# Aspects of Data Integrity

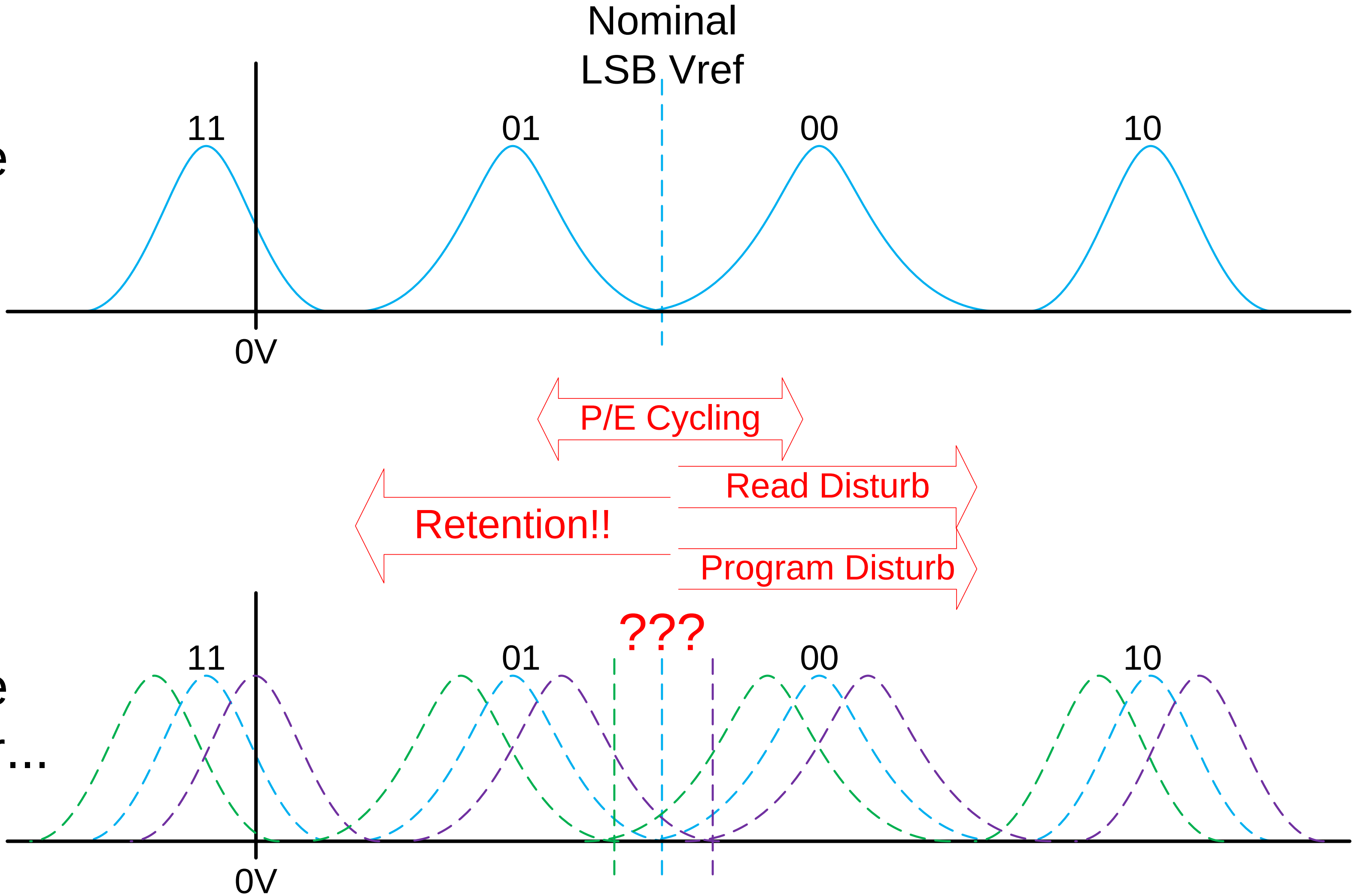

- Knowing there was an error!
  - End-To-End Integrity Checking
  - Avoid silent data corruption, misdirected writes, …
    - Internal ECC/parity, address corruption checks, …
  - Issues here common to all storage devices
- Preventing/Correcting Errors
  - Robust Error Correction – Beat the UBER
  - But watch out for performance suffering!
    - Sometimes the cost of getting your data is high…

- This talk: preventing/correcting errors

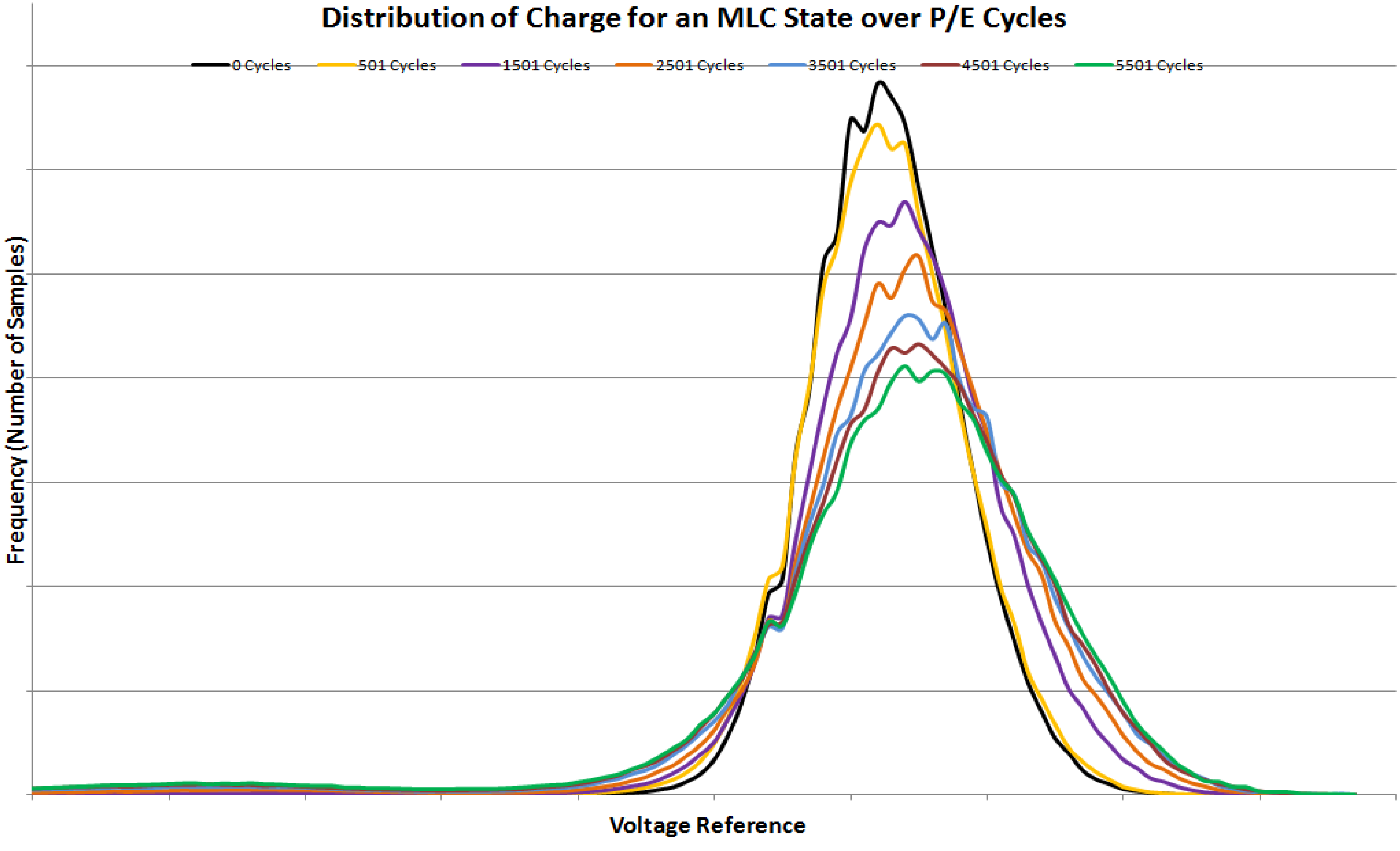
# Why ECC – Where's My Data?

What we wrote...

What we find later...

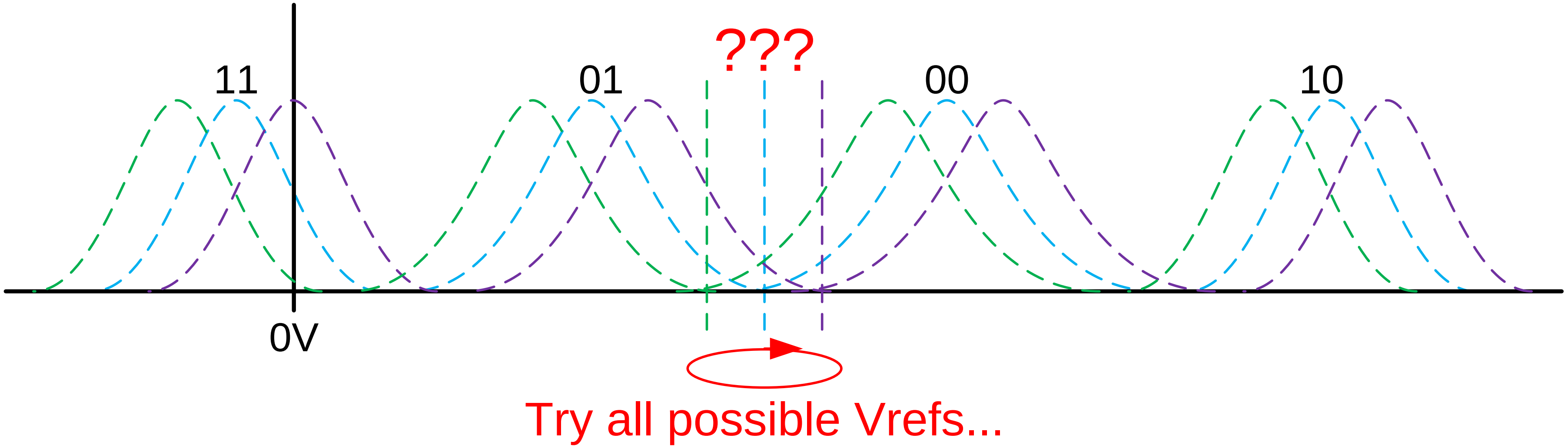


# What Your Data Really Looks Like



# Read Retry – Finding Your Data!

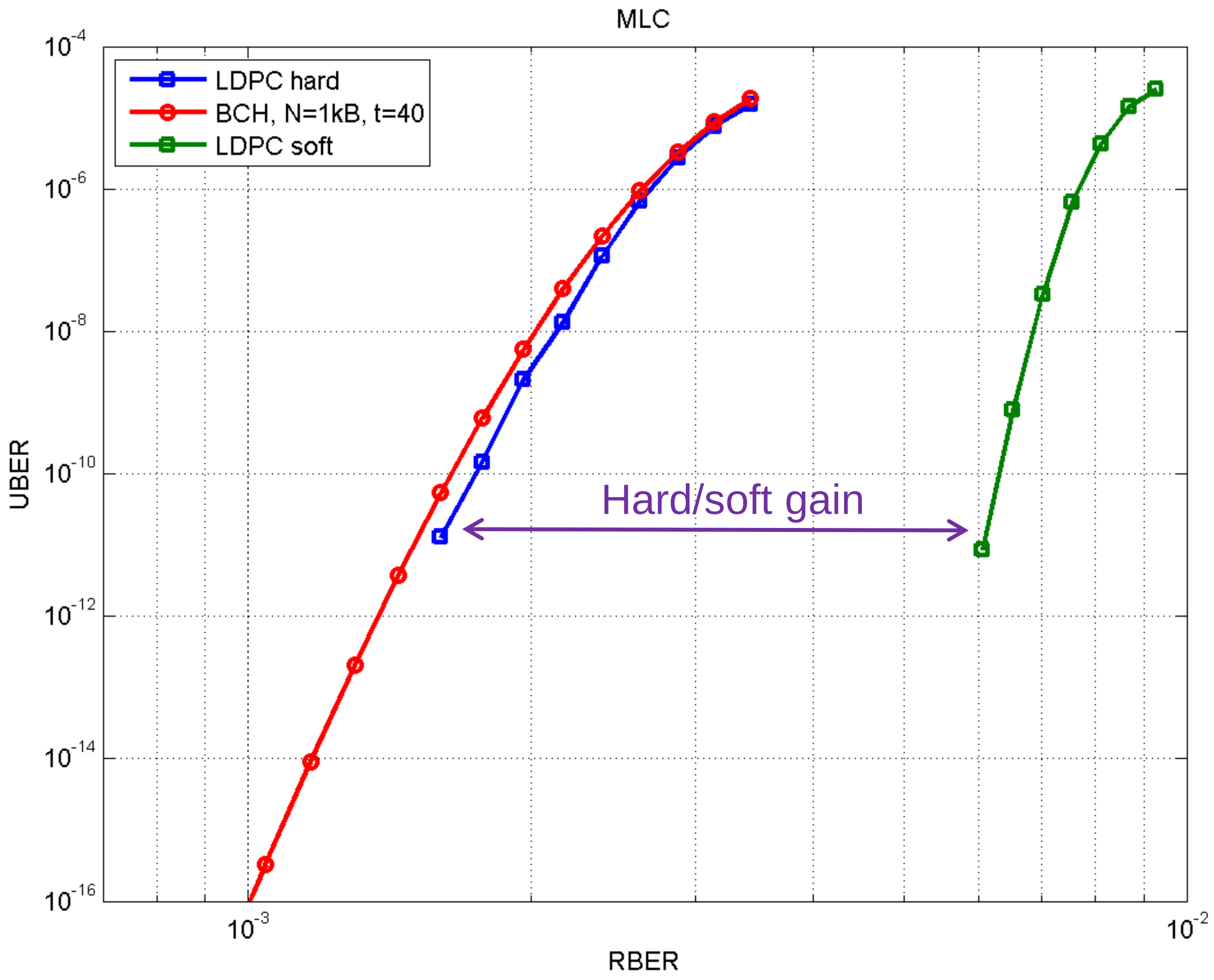
- Adjust Vref until you can recover data
  - Naïve approach – linear search
  - Sophisticated approaches…
    - Tracking, interpolation, …
- How long will it take you to find your data?



# LDPC – Coding Headroom

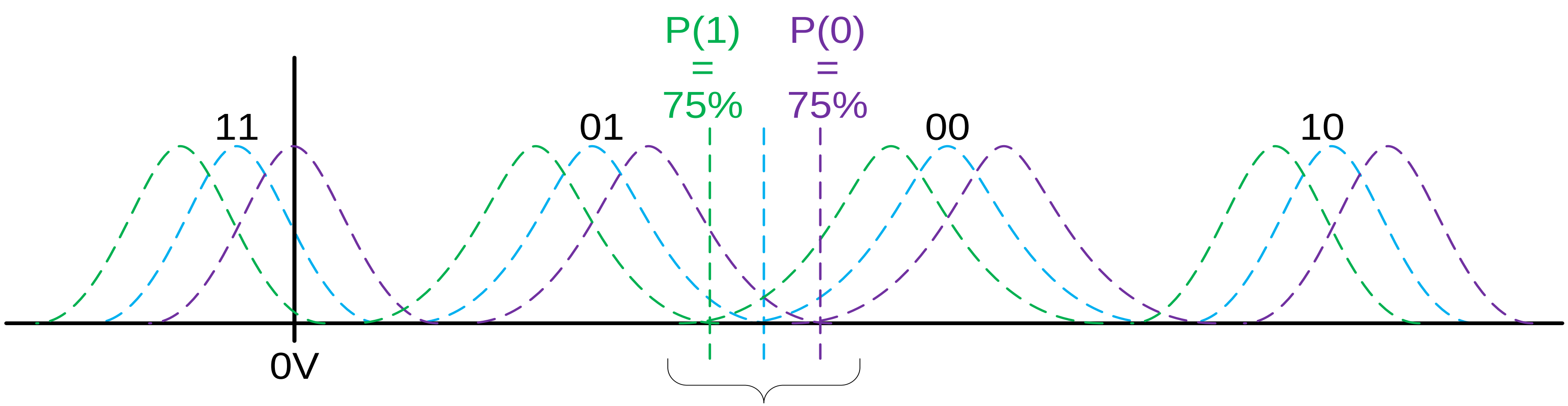

- LDPC is an iterative coding technique
  - More run-time ⇨ better correction
    - But lower throughput
  - More information (read retry) ⇨ better correction
    - BCH: binary use of individual read retries
    - LDPC: soft-decision use of *all* read retry information
- Optimize for throughput
  - But be able to use coding headroom when needed

# What Coding Headroom Looks Like



# LDPC – More Efficient Read Retry

- Time to data is a key metric
  - How many read retries are required to "find" data?
  - Soft-decision decoding using information from read retries can reduce time to data

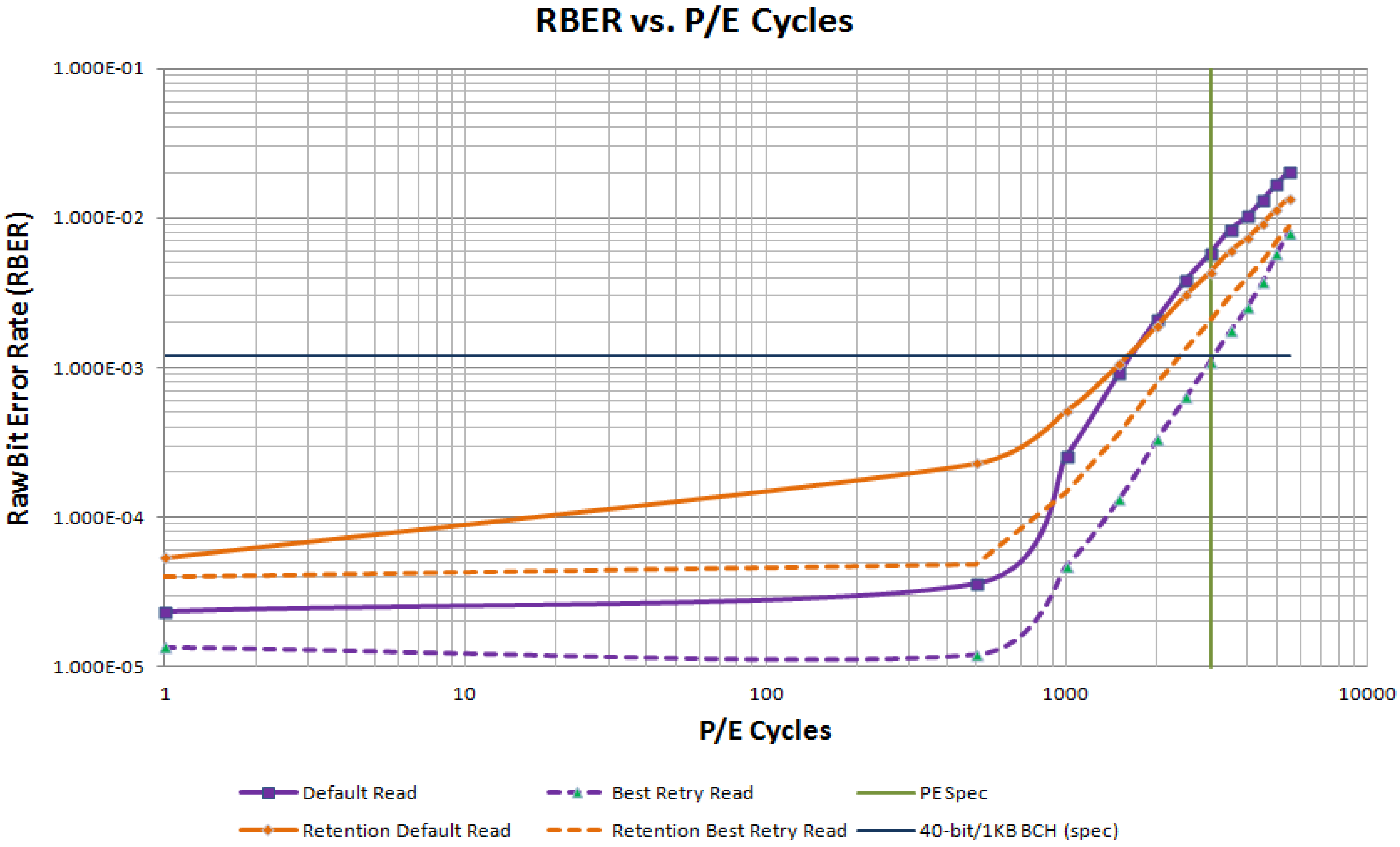



Convert small number of read retries to LLR (Log Likelihood Ratio)

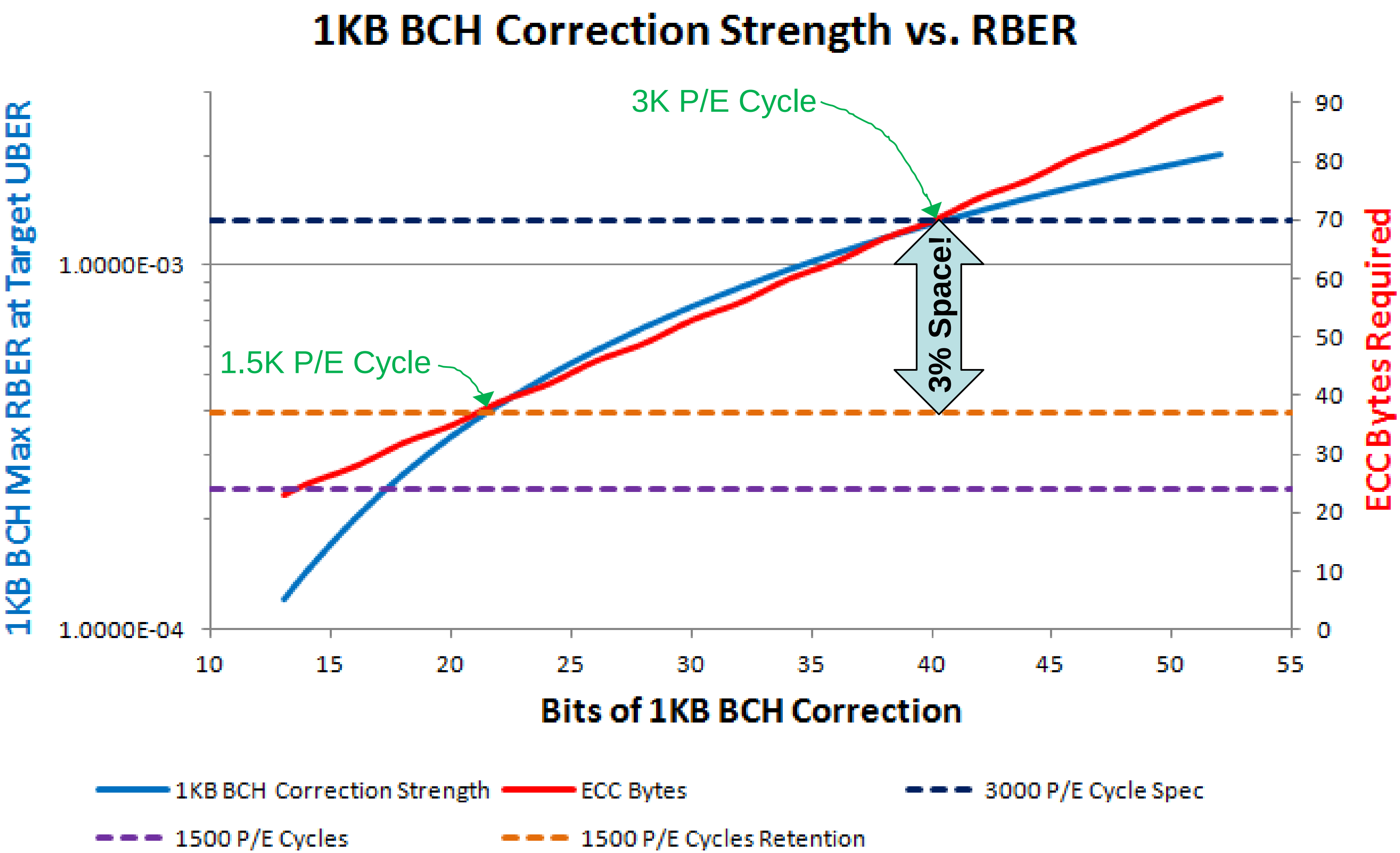
# So When *Do* We Need Strong ECC?

- That depends on …
  - How often you want to read retry
    - And performance consequences thereof
    - Pay one Tr per read retry!
    - But it may let you find a point with fewer errors
- For a good fraction of the P/E cycle lifetime
  - We *don't* need very strong ECC
  - But late in life, read retry may be required!
  - Is there cost in having ECC constant over lifetime?

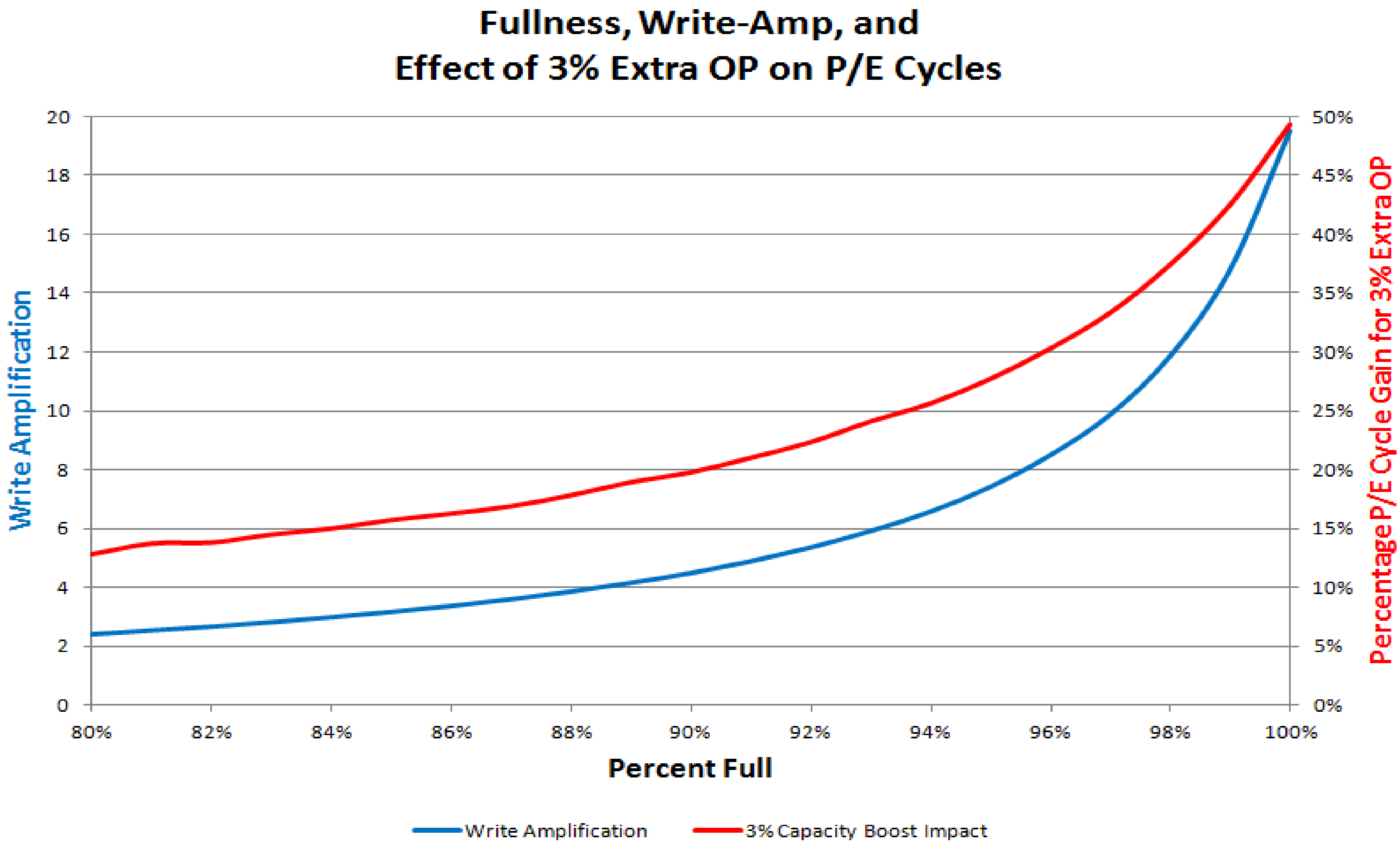
# When Do You Need (Strong) ECC?



# How Much ECC and When?



# The Power of 3%

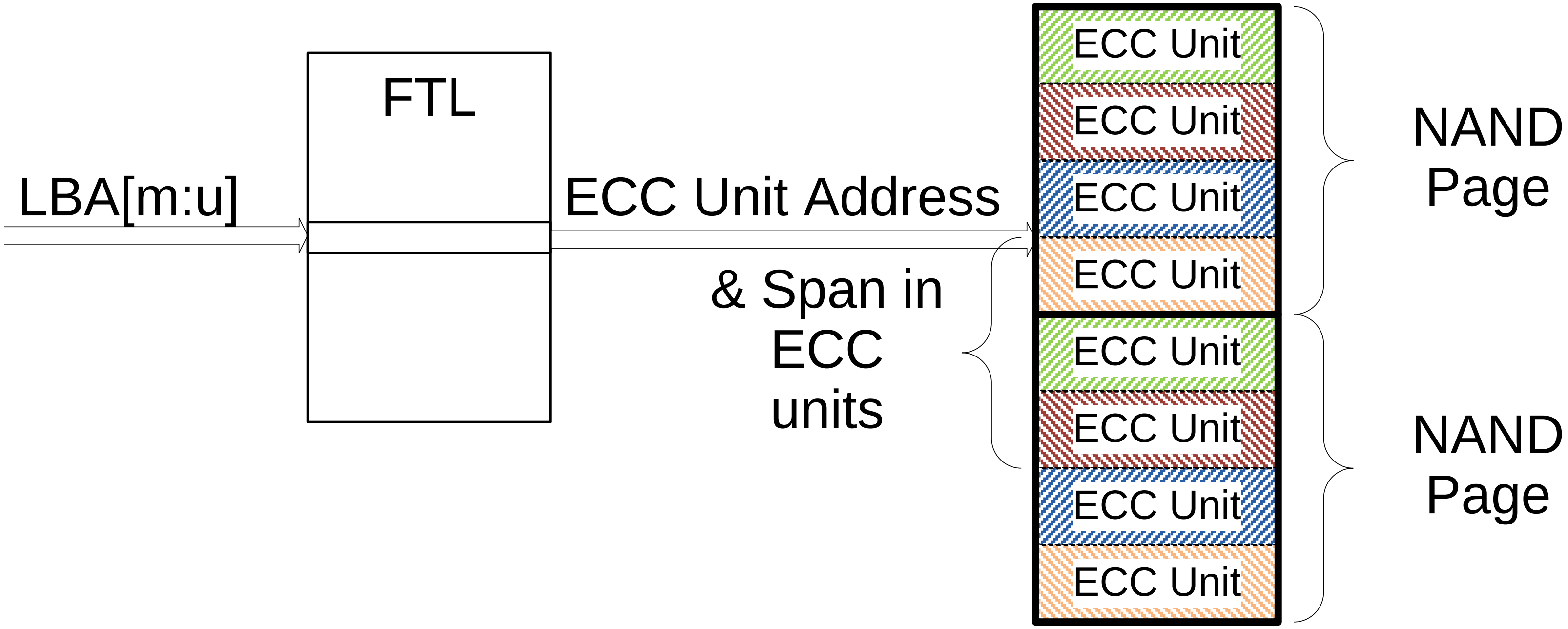


# You Want the 3%!
# FTL Implications…

- Goal: maximize use of flash page for user data
  - User Data vs. ECC changes over lifetime
  - User Data vs. ECC changes for …
    - Stronger and weaker blocks/pages/…
- Problem: typical 4KB write doesn't pack nicely into flash pages any more…
  - User portion of flash page size "borrows" some of the spare normally used for ECC

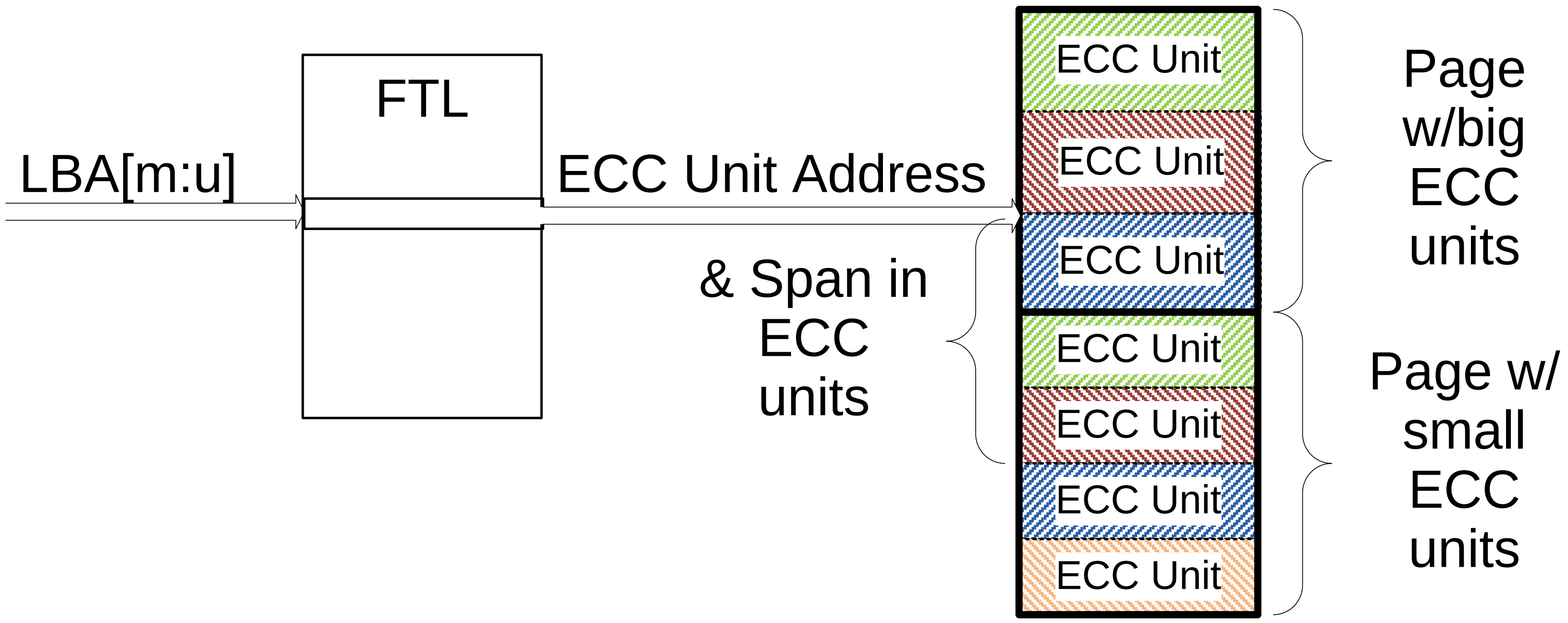
# Mapping Scheme for VFTLs (FMS 2012)

- How to map the LBA to data location in flash?
  - Any access must read an integer # of ECC units
  - Only need to point to first one and how many



# Mapping Scheme for VFTLs with Variable User Flash Page Size

- Data spans any number of ECC units
- Number of ECC units per page and/or amount of data per ECC unit can vary



# Summary

- NAND flash error rates continue to increase while datasheet lifetimes decrease
- ECC needs of NAND vary over lifetime
- Design to take advantage of this:
  - User powerful coding with headroom
  - Design your FTL to optimize for this variability
- Maintain data integrity while maximizing throughput and performance

# LSI is Accelerating Flash Storage Innovation


- **Attend the LSI keynote on Thursday 8/15 @ 11:30am**
  - *Optimizing Flash Controller Technology for Next-Gen Flash*

    Greg Huff, LSI Senior Vice President and CTO

- **Visit us at booth #402**
  - Experience new LSI flash storage innovations
  - See live demos of LSI SandForce Driven SSDs
  - Enter to win SandForce Driven SSDs