LSI
Storage. Networking. Accelerated.

SandForce DRIVEN By LSI

Why Variable-Size Matters:
Beyond Page-Based Flash Translation Layers

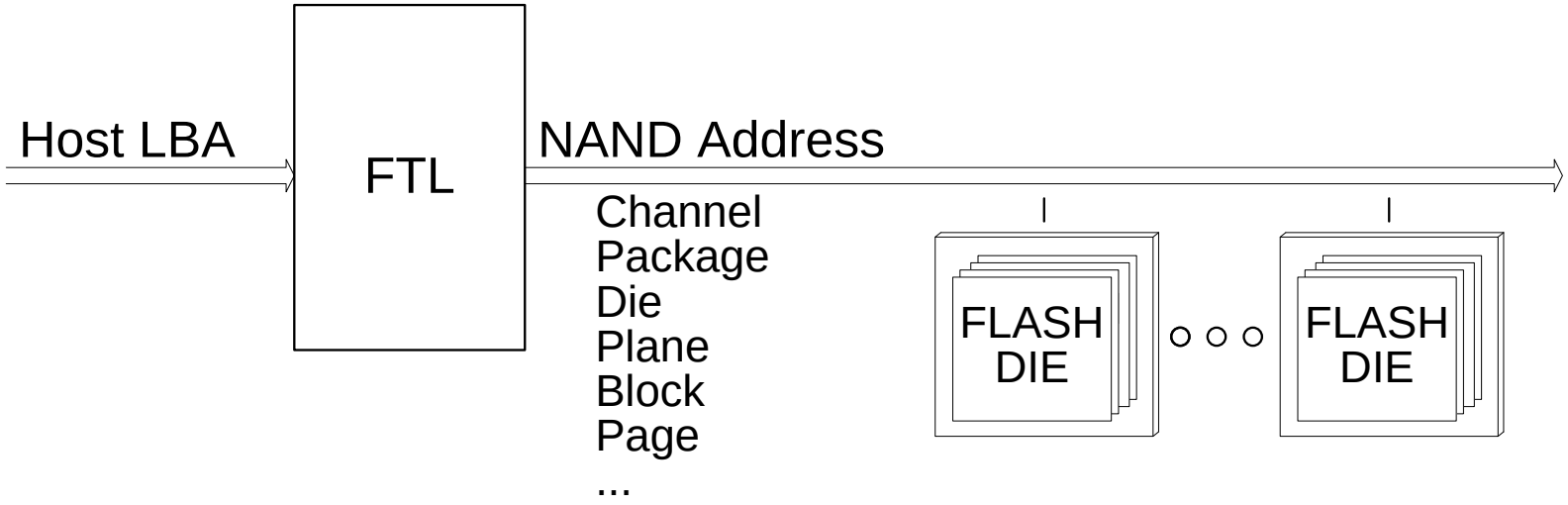
Earl T. Cohen

Flash Components Division

LSI Corporation

Introduction

- Flash Translation Layers (FTLs)
  - Provide the dynamic mapping from Host Logical Block Addresses (LBAs) to addresses in flash
  - In a desired granularity (e.g., block, page, …)



Some FTL Aspects

- Mapping of LBA (algorithm, granularity, …)
- Recycling (garbage collection)
- Wear-Leveling (Block Picking, Data Placement)
- Bad-Block Handling
- Performance / Overhead
- Checkpoint and Recovery Algorithms

Some FTL Performance Factors

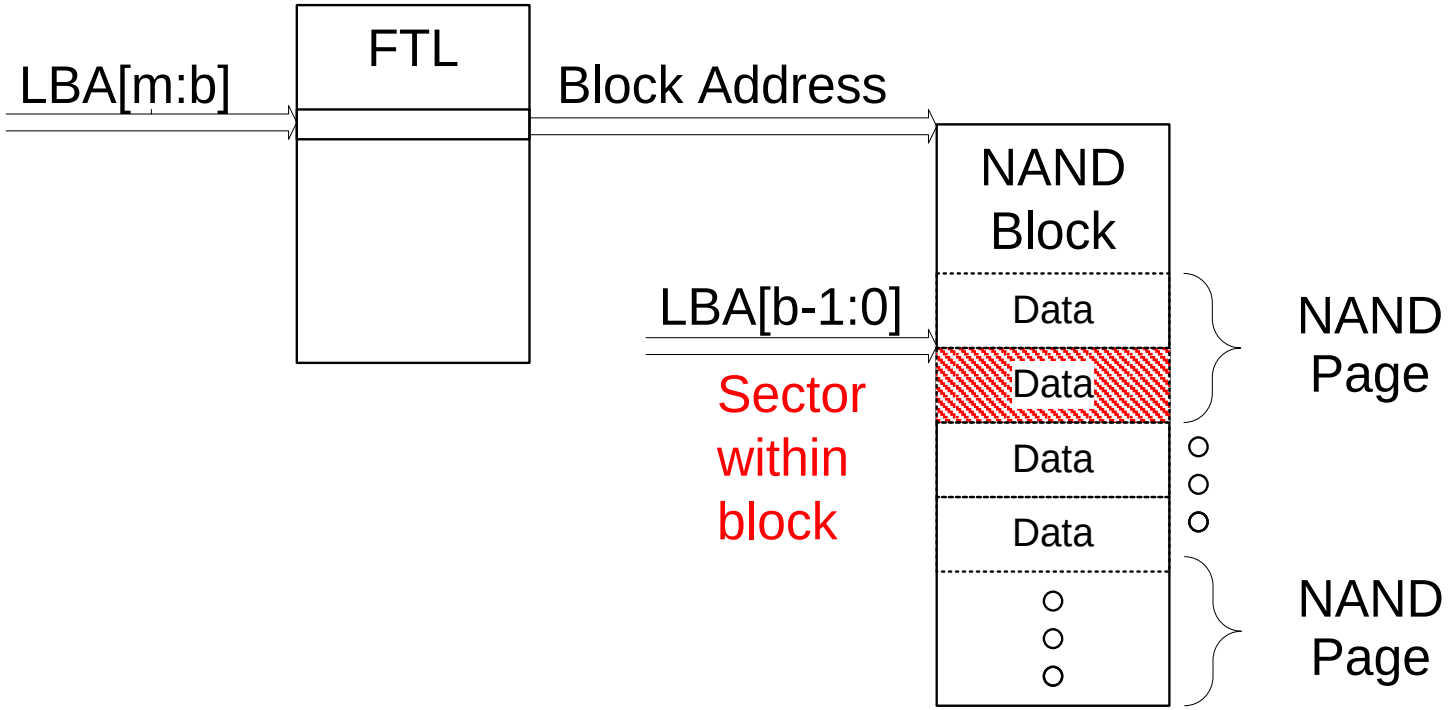
- Read/Write overhead to use/maintain FTL
  - Flash bandwidth consumption, incl. write amplification
- CPU processing overhead
- Wake-up time (what must be loaded at power-on)
  - How quickly after power-on is user data accessible?
- Unsafe shutdown recovery time
  - How long to restore the FTL after a power fail?

Typical FTL Components

- Map
  - A means to convert LBAs to NAND addresses
- Meta-data
  - Generally stored in-band with user data, such as storing the LBA with the data itself
  - LBA stored with data acts as a “reverse map” – used for recovery
- Logs/Journals
  - Used in some FTLs to record changes to the map

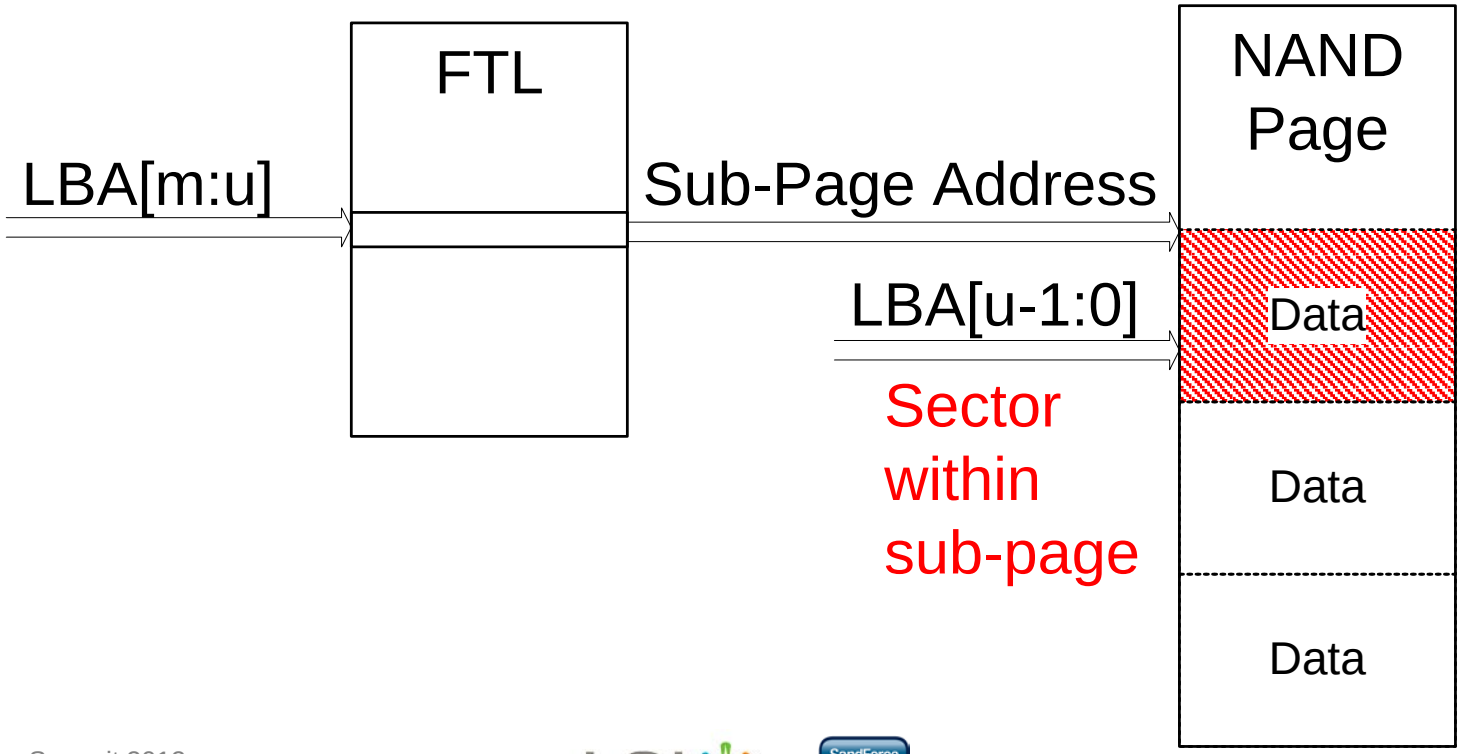
Traditional Block-Based FTLs

- LBA MSBs -> Block
- LBA LSBs -> within block



Newer (Sub-)Page-Based FTLs

- LBA MSBs -> unit within page (e.g., 4KB)
- LBA LSBs -> within unit (e.g., sector)

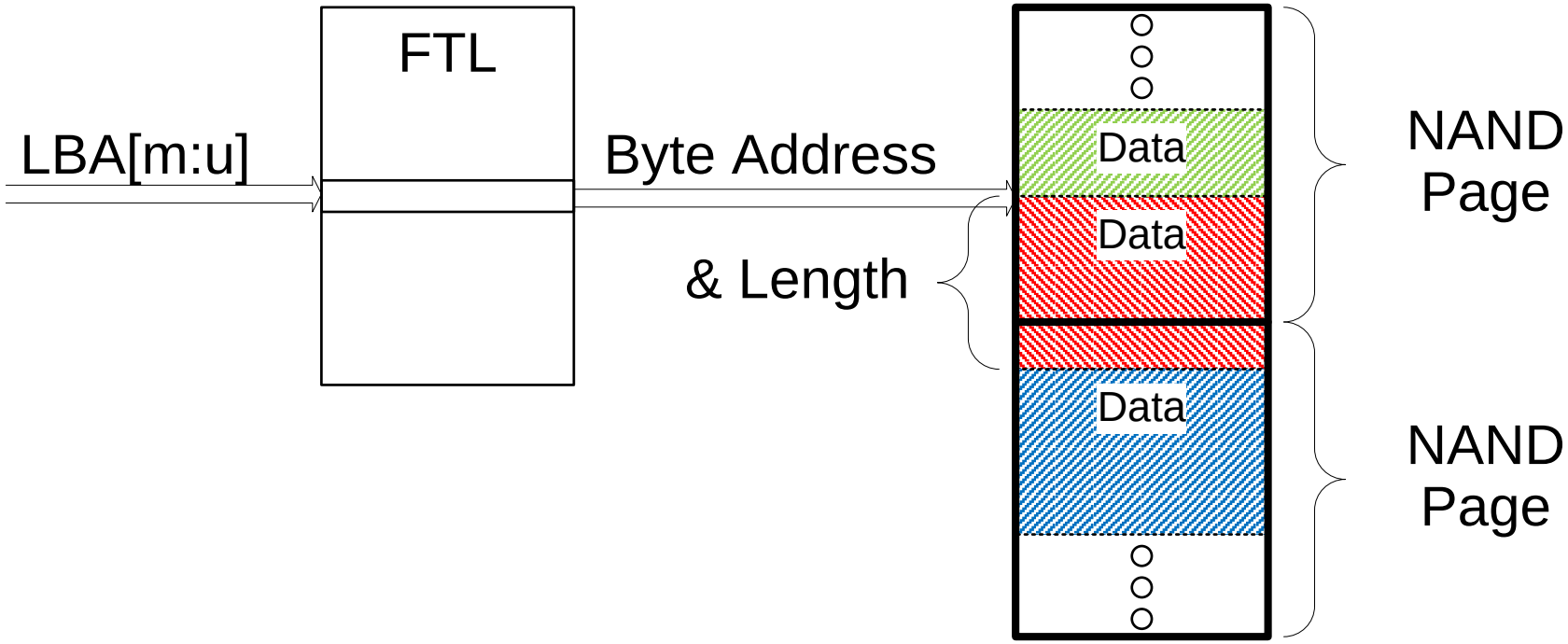


Issues with these Approaches…

- Block-based too coarse a granularity
  - Write amplification way too high – writing 4KB re-writes an entire block
- Sub-page-based ties the data size to be an integer divisor of the flash page size
  - What happens with non-512B sectors?
  - What if the data size is variable?
  - What happens if ECC requirements differ from the spare size (either more or less)?

Solution: Variable-Sized FTLs (VFTLs)

- LBA MSBs -> Byte Address and Length
- LBA LSBs -> used to extract data

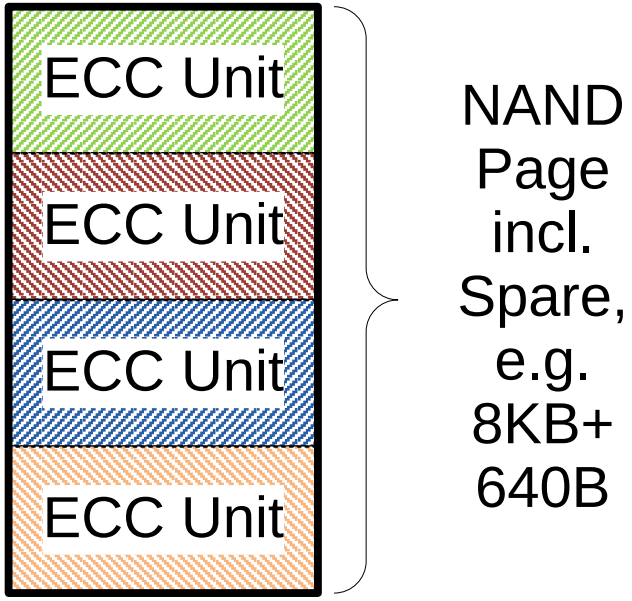


Byte Address and Length???

- Byte address and length is a lot of bits!
  - 256GB device has 38-bit byte address
  - 4KB granularity requires 12-bit length
  - 50 bits total per map entry
- Compared to 4KB granularity sub-page map: 38-12 = 26 bits per map entry
- To close this gap, use ECC unit address!
  - Rather than byte address

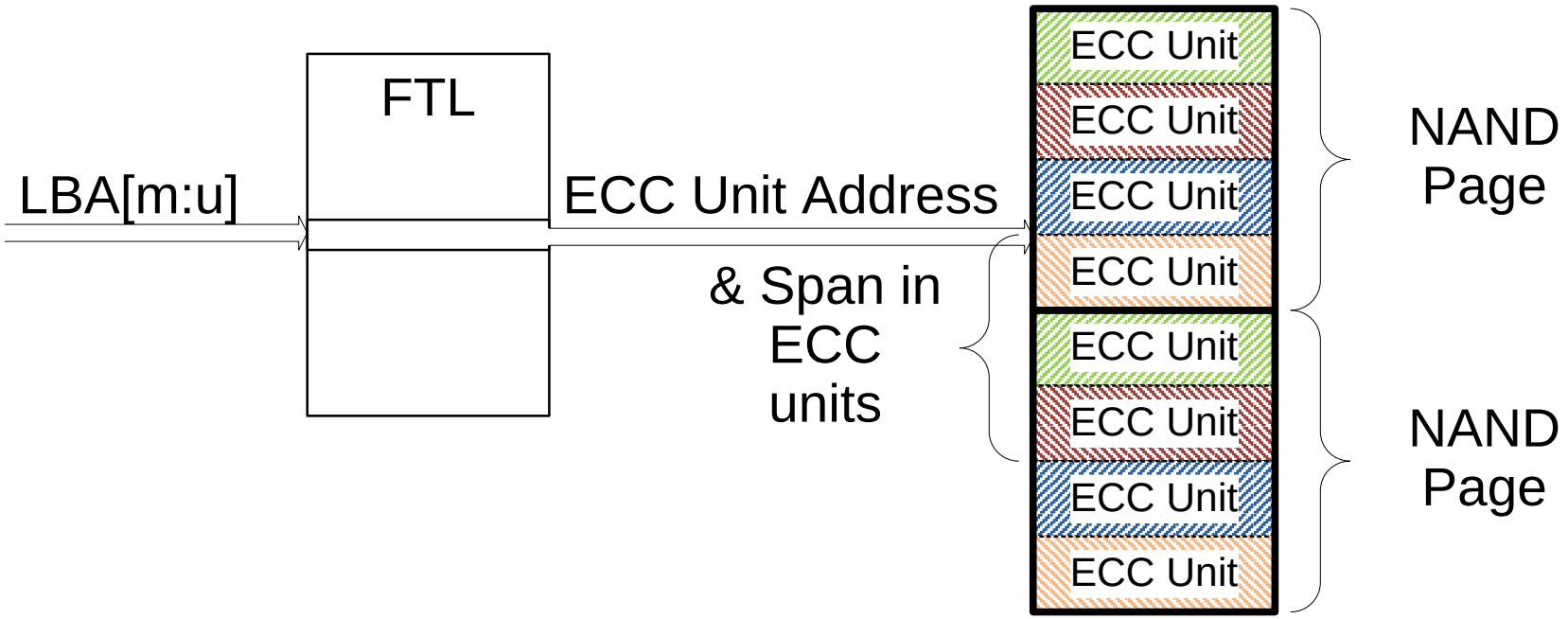
What is an ECC Unit?

- Flash page sizes are larger than the Error-Correcting Code (ECC) sizes used
  - 8KB or 16KB flash pages
  - vs. typical 1KB BCH codes
- Multiple ECC units (codewords) per flash page
  - The ECC unit is the smallest portion that can be independently read and corrected



Mapping Scheme for VFTLs

- 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

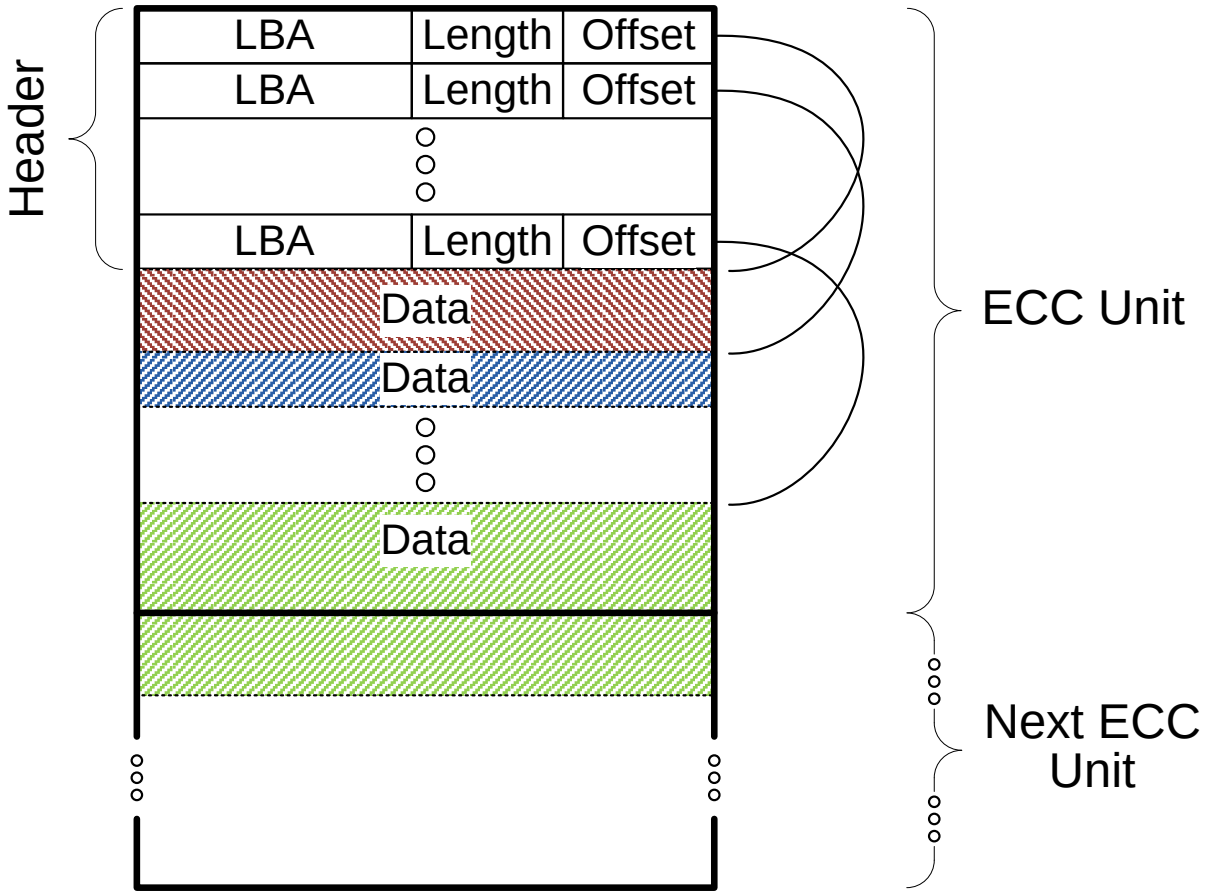


ECC Unit Address and Span

- For 256GB drive, byte address and length for a 4KB unit was 50 bits
- With 4KB mapping, 2KB ECC units, span in ECC units is just 1 bit
  - Map cost is 38-11+1 = 30 bits
  - vs. 26 bits for the simple, sub-page map
- But we still need to find the data in the ECC units…
  - Data is preferably *not* aligned with ECC units

Extracting the Data in the ECC Units

- Header at start of ECC unit defines data in that ECC unit
- Data can span ECC units



Nothing is Free: Costs of VFTLs

- Map entry size increase
- Writing variable-sized data (span flash pages)
- Finding data within ECC units
- Free space tracking complexities
- Recycling interactions
- …
- The SSD Processor must be designed from the ground up to support Variable-Sized FTLs

Benefits of VFTLs

- Great flexibility in flash support
  - Non-binary user portion of flash page
  - Great flexibility in how spare is used and how much or little is used for ECC (decouples ECC code rate)
- Enables non-power-of-two sector sizes
- Enables data reduction (e.g., compression)
  - The storage used by an LBA can be arbitrary
    - No wasted space – every byte is used
  - Decreases write amplification, increases longevity

LSI is Accelerating Mobility at the Flash Memory Summit 2012!

- **Visit us at the booth 624-626 to:**
  - Experience the latest SandForce Driven™ Ultrabook™ systems
  - See live demos of LSI SandForce Driven SSDs
  - Discuss the latest trends in flash and SSD solutions
  - Enter to win **SandForce Driven SSDs** and the **Grand Prize – a SandForce Driven Ultrabook!**