



A Toshiba Group Company

# Enhancing System Architecture by Modeling the Flash Translation Layer

Robert Sykes  
Sr. Dir. Firmware

OCZ Storage Solutions

A Toshiba Group Company

Audio-Visual Sponsor

Santa Clara, CA  
August 2014



Shannon Systems



# Introduction

- *This presentation will discuss the benefits of creating a standalone software representation of a real Flash Translation Layer that will be used in an end product*
- *How generating this model will help in ensuring the system will meet the workloads as set out in the IDEMA specification will be discussed, as well as how the FTL model will allow flexibility in looking at the system dynamics for custom workloads such as specific dirty performance characterization, write amplification characteristics, garbage collection and reclamation, and the impact that over-provisioning has on the system dynamics*

Audio-Visual Sponsor

Santa Clara, CA  
August 2014



Shannon Systems

## What is an FTL Model ?

- The FTL is the only major software components of an SSD
  - Whilst it heavily interacts with the hardware, the hardware elements could be simulated resulting in a software only representation of a complete system
    - Thus a software model of an FTL could be created for analyzing such interactions with its surrounding hardware to help with the architecture and system design before any RTL has been written
- Consideration should be given in the design of such a model that will allow the code to be easily transported to the real system
- A model could be created using a compiler on a desktop / laptop allowing for quick simulations / prototyping to be run

Audio-Visual Sponsor

Santa Clara, CA  
August 2014



Shannon Systems



# Why Create an FTL Model?

- Allows for early prototyping of FTL algorithms
- Enables easy experimentation
- Proves validity of system level concepts
- Easier debugging
- Useful to keep and extend for future generations of the FTL
- Better, faster test platform for FTL development

Audio-Visual Sponsor

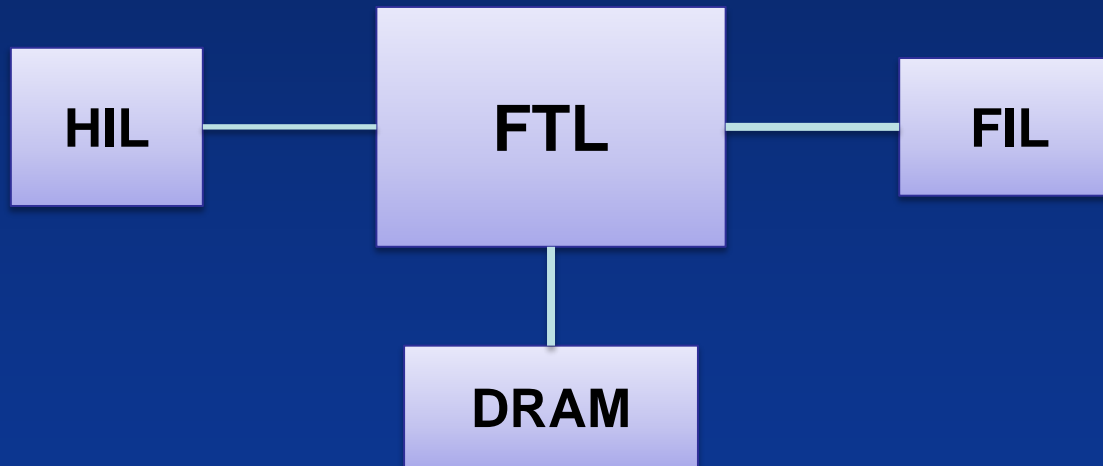
Santa Clara, CA  
August 2014



Shannon Systems

# Scope of the FTL Model

- FTL is bounded by:
  - The HIL and FIL
  - DRAM (and other utilities)



Audio-Visual Sponsor

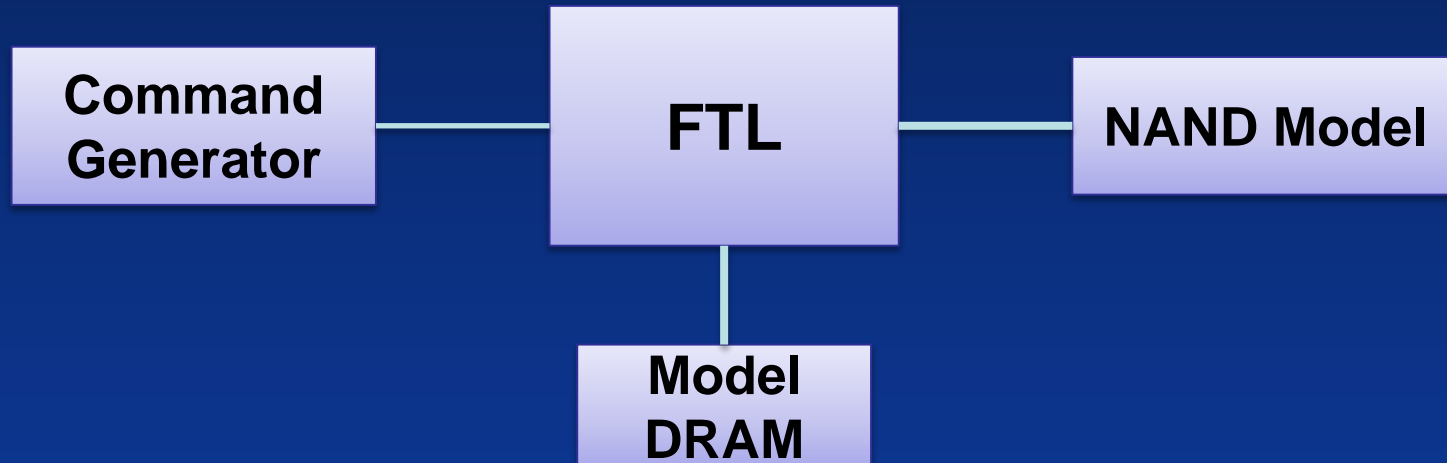
Santa Clara, CA  
August 2014



Shannon Systems

# Model Boundaries

- Model attempts to use these same boundaries



Audio-Visual Sponsor

Santa Clara, CA  
August 2014



Shannon Systems



# FTL Model

- Compression
- Packing of data into pages, and the layout of that data on the NAND Map Table
- Reclaim
  - Testing the reclaim features including wear levelling and keeping erase counts close to nominal etc.
  - Testing different trigger points for GC and wear levelling – i.e. choosing a block with low erase count or low data or an old block with growing error rate

Audio-Visual Sponsor

Santa Clara, CA  
August 2014



Shannon Systems



## FTL Model

- Injecting errors onto the software tables and recovering
- PFail, testing recovery mechanism, analysing how much data can be backed up with different algorithms, DRAM sizes, bank counts etc.
- Accuracy, want model to be as close as possible to the real thing (in terms of number of banks and then throughput) to create the most accurate tests

Audio-Visual Sponsor

Santa Clara, CA  
August 2014



Shannon Systems





# FTL Model Advanced

- Model NAND should model flash controller too
  - Add channel bottlenecks and latencies for read write or erase commands – how many errors or retries before performance is affected? Max/average latencies?
- Model host needs to be able to take many different sequences, 4KB random, 128KB sequential, mixture of compressible and uncompressible data, mix of slower commands too, the JEDEC standards for different commands
  - WA can then be analysed
- The model should be able to run on desktop PC for speed (and regression testing) but also on development platform for easy cross-platform testing. Consideration of the HW environment should be driving the tool development.

Audio-Visual Sponsor

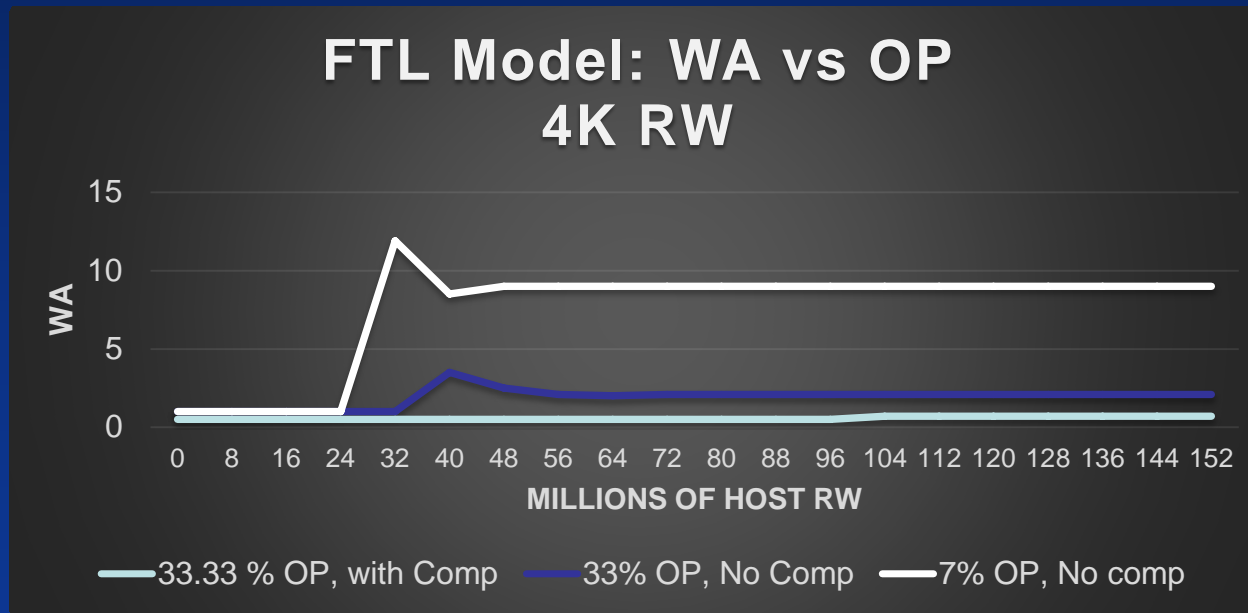
Santa Clara, CA  
August 2014



Shannon Systems

# FTL Statistics

- Model can store interesting information for validating concepts



Audio-Visual Sponsor

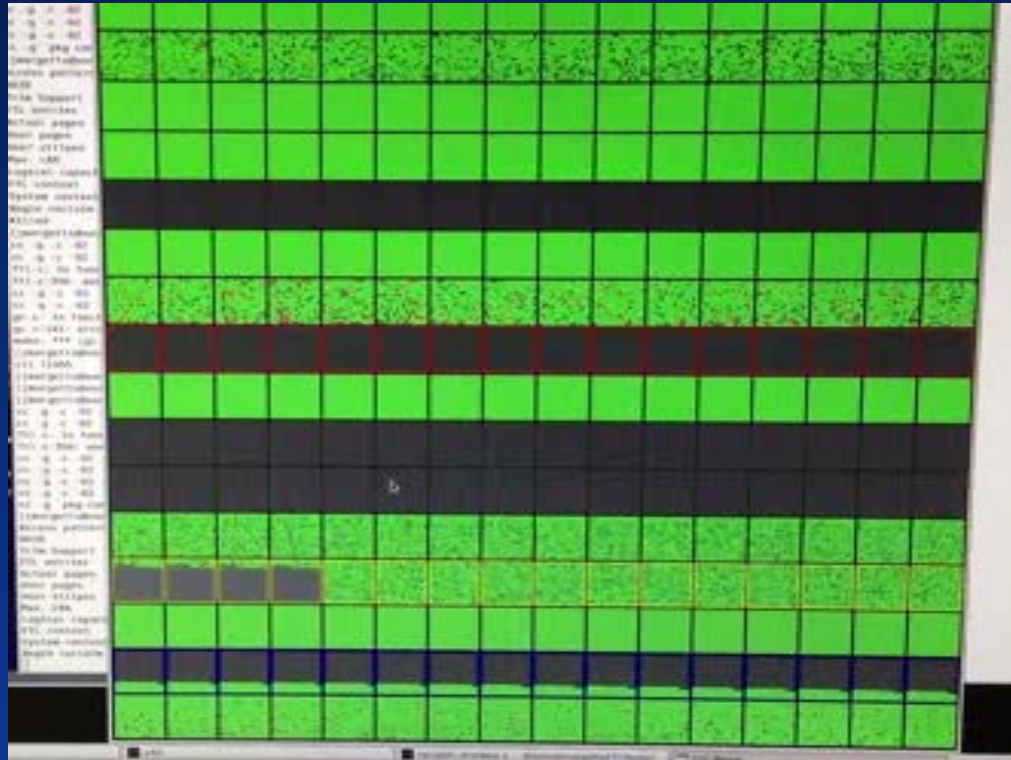
Santa Clara, CA  
August 2014



Shannon Systems



# Graphical Reclaim Simulation



Audio-Visual Sponsor

Santa Clara, CA  
August 2014



Shannon Systems



# FTL Model HIL Basics

- Start with a very simple command generator
  - Random LBA accesses
  - Random read/write command mix
  - Random compression ratio
- Adding more flexible tests, traffic patterns etc.

Audio-Visual Sponsor

Santa Clara, CA  
August 2014



Shannon Systems

# FTL Model HIL advanced

- Create different traffic profiles
  - Vary OP – different OPs will affect NAND utilization, WA, performance, change the bottlenecks
  - Compression – testing different compression algorithms, data patterns, mechanisms for packing data into NAND pages – split onto multiple pages, limit number of slices per page etc.
  - NAND configuration – when does the number of banks become critical to the traffic profile – could use it to measure throughput and with good models the system performance could estimated

Audio-Visual Sponsor

Santa Clara, CA  
August 2014



Shannon Systems



# JEDEC standards

- Enterprise requirement:

- 512 bytes (0.5k) 4%
- 1024 bytes (1k) 1%
- 1536 bytes (1.5k) 1%
- 2048 bytes (2k) 1%
- 2560 bytes (2.5k) 1%
- 3072 bytes (3k) 1%
- 3584 bytes (3.5k) 1%
- 4096 bytes (4k) 67%
- 8192 bytes (8k) 10%
- 16,384 bytes (16k) 7%
- 32,768 bytes (32k) 3%
- 65,536 bytes (64k) 3%

- The workload shall be distributed across the SSD such that the following is achieved:

- 1) 50% of accesses to first 5% of user LBA space (LBA group a)
- 2) 30% of accesses to next 15% of user LBA space (LBA group b)
- 3) 20% of accesses to remainder of user LBA space (LBA group c)

\*\*JESD218, Solid State Drive (SSD) Requirements and Endurance Test Method

Audio-Visual Sponsor

Santa Clara, CA  
August 2014



Shannon Systems

# Command Generator

- Now we have a specification of what to test we need to create the specific data. This can be done in two ways:
  - Create a file on the host PC that has the exact data type you want and feed this into the systemor
  - Create a data generator:
    - This would be similar in creating a packet header for the type of data you want the system to generate i.e. create a list of requirements you want the data to be, i.e. LBA address, random data, compression ratio etc.
    - This will create data on-the-fly, obviously adds delay whilst the data is generated but that's OK as we don't need to time anything until the data is created

– **The advantage of this method is that it provides a more powerful tool in simulating quickly different data pattern**

Audio-Visual Sponsor

Santa Clara, CA  
August 2014



Shannon Systems



## FTL Model FIL

- Basic FIL
  - Start with a simple flash queue interface
  - Simple storage of commands
  - Test-out synchronous command completion
- Advanced FIL
  - Real flash queue
  - Modeled NAND channels include latencies
  - Async status completions back to FTL

Audio-Visual Sponsor

Santa Clara, CA  
August 2014



Shannon Systems





## FTL Model FIL advanced

- NAND arbitration (round robin or some other technique based on traffic profile?) Size of a RAID block?
- NAND page size – linked to number of channels, if you make the page larger you affect performance
- Measure WA over extended periods of time, see the influence that different traffic has on it
- Buffer management - management of the host data ring buffers, reclaim buffers, others? Cache hits?

Audio-Visual Sponsor

Santa Clara, CA  
August 2014



Shannon Systems



## FTL Model FIL cont...

- Test NAND migration
  - T\_Prog simulation
  - Read Retry algorithms

Audio-Visual Sponsor

Santa Clara, CA  
August 2014



Shannon Systems

## FTL Model DRAM

- Simple DRAM model could use static arrays to simulate large parts of DRAM
- Advanced features:
  - Create a real system interface
  - Add statistics – how many accesses etc.
  - Optimizations, caching locally in SRAM

Audio-Visual Sponsor

Santa Clara, CA  
August 2014



Shannon Systems



## Limitations

- Hard to keep close to HW design
- Different timings and latencies cause different bugs in real world
- Hard to get accurate models
- Extra maintenance

Audio-Visual Sponsor

Santa Clara, CA  
August 2014



Shannon Systems



# Thank You!

Audio-Visual Sponsor

Santa Clara, CA  
August 2014



Shannon Systems