Introduction

- Current challenges in SSD test
- Concept of a Test Primitive
- The life cycle of SSD testing
- How the Primitive concept can be applied to the life cycle test phases
Current Challenges in SSD Test

- Need to test each phase of SSD development
- Each phase has unique test focus

- Is there a way to combine this?
Concept of a Test Primitive

- Combination of:
  - Firmware/software board
  - Power supply board
- The Primitive insert into different tester frameworks to address various testing needs
Test Primitive is the Heart of Testing

- FPGA
- Embedded Linux
- Firmware control
- Software control
- Embedded OS
- Example of software/firmware board
- Supports N devices
- Example of power supply board
- Power control for each device individually
- UL Software
Can use the same:
  • User-level software
  • System-level control

Across different tester types of the SSD life cycle

Scalable to configurations across life cycle
The Life Cycle of SSD Test
Test Requirements

- Different Test Methods
- Different Form Factors
- Different Protocols
- Different Speeds
- Enterprise/Consumer
- Manual/Automated
- Different Temperatures

SSDs
The Primitive for Development

- Primitive in small tester
  - Develop test programs
  - Doesn’t burn power like large systems
  - Focused on basic functionality verification

Would use one primitive

Example of a development type of tester and its loadboard
The Primitive for Reliability Testing

- Primitive in environmental chamber
  - Use same software
  - Use development test programs
- Test end of life of devices and corner case exposure of the controller/NAND interaction
Scalability of Primitive for RDT

- Same primitive as development tester can be multiplied for an RDT tester
  - Could use up to 16 primitives for this type of system

Would use two primitives per loadboard
The Primitive for High Volume Manufacturing

- Primitive in rack system
  - Use high-density power supply and firmware boards for the primitive
  - Use same software
  - Use same development test programs
- Shorter tests for device confirmation for manufacturing quality
High-Density Primitive

- Same embedded Linux and FPGA architecture
  - No need to change how tests are run
  - Same method, just larger parallelism
High-Density Primitive Examples

Example of HD firmware/software board

FPGA

Embedded Linux

Firmware control

Embedded OS

Software control

Example of HD power supply board

Still could have power control for each device individually

Supports 2N devices
Would use one high-density primitive per shelf.

Example of an HVM type of tester. Three racks with four shelves each.
The Primitive with Automation

- HVM tester with primitive
  - Can add automation to drive insertion and removal
  - No change to primitive required
The Primitive with BIST

- Simple firmware and power supply boards for primitive
  - Not as much control or performance needed
- Can be combined with burn-in testing
- Concept of Primitive scalability remains
Revisit Test Requirements

- Different Test Methods
- Different Form Factors
- Different Protocols
- Different Speeds
- Enterprise/Consumer
- Manual/Automated
- Different Temperatures

SSDs
Concept of the Primitive as Applied to Life Cycle Testing

- Same primitive concept used across many tester types
  - Reuse software, firmware, embedded OS across:
    - Test phases
    - Tester types
    - Form factors
    - Protocols
Implications of this Solution

- Meet cost targets of each individual phase of life cycle testing
- Make the heart of tester portable and scalable to address challenge of testing the life cycle of an SSD