A Solid-State Disk Simulator for Quantitative Performance Analysis and Optimization

연세대학교 전기전자공학부

정 의 영

Contents

- Introduction
- Related Works
- Solid-State Disk Simulator Implementation
- What to do w/ Solid-State Disk Simulator
- Solid-State Disk Simulator Experimental Results
- Ongoing & Future Works
- Summary
- Demo of SSD Simulator

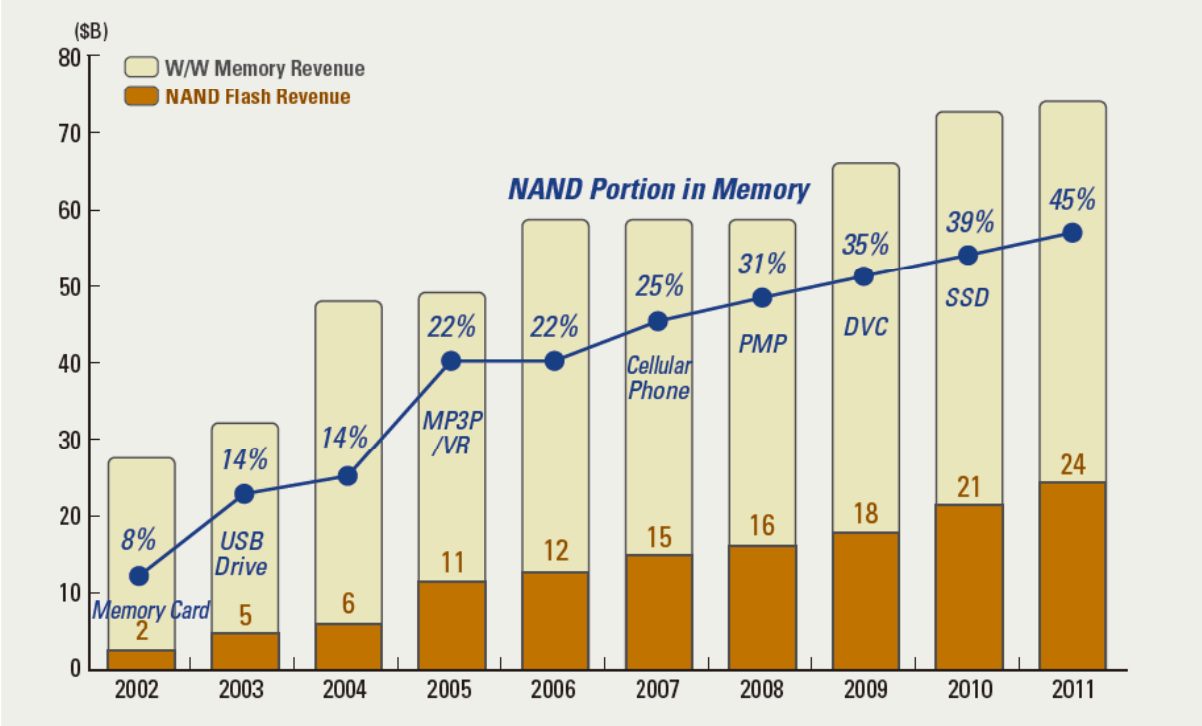
Why SSD?

- **Nice features compared to HDD**
  - **Small form factor**
  - **Shock resistance**
  - **Light weight**
  - **Low power consumption**
  - **High performance**

- **Worth to pay more?**
  - **NAND flash is getting cheaper**
  - **More data-intensive applications**

NAND Flash Memory Market Forecast

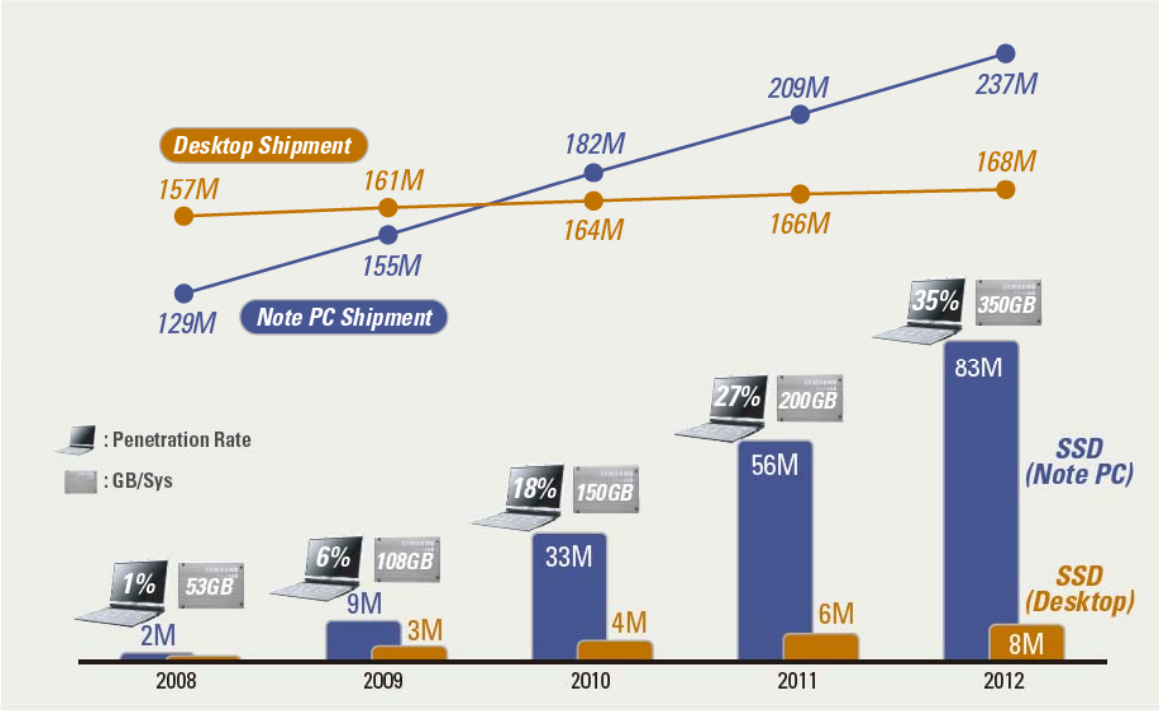
- **NAND becomes dominant in memory market**



Source : WSTS'07, iSuppli('08~), SEC('08~)

SSD Market Forecast

- Currently, the market is in the opening stage
- Note PC is the major market



Source : iSuppli (June, 2008)

To boost up the market growth

- **Reduce the disadvantages**
  - **Hopefully by Multi-level cell (MLC) NAND**
    - **Price**
      - **Integrate more bits to a single cell**
    - **Reliability**
      - **Improve the noise immunity by a novel ECC scheme**
    - **Performance**
      - **Hide the poor characteristics by the smart controller**
- **Enhance the strengths**
  - **Higher performance**
    - **Controller / Host IF**
    - **Especially for random read / write**
  - **And others**

Why SSD simulator?

- For the challenges in the previous slide
  - Need a tool for analysis and exploration
- System-level simulator
  - Host computer + SSD
    - Host computer architecture
    - Memory hierarchy
    - Interaction between File system and SSD
- SSD simulator
  - Architecture
  - Flash Translation Layer (FTL)
  - Also their combined effects

Purpose of this talk

- Not to propose a new architecture
- Not to propose a fancy FTL algorithm

- But to propose a method for the analysis and exploration of SSD in a quantitative manner

Contents

- Introduction
- Related Works
- Solid-State Disk Simulator Implementation
- What to do w/ Solid-State Disk Simulator
- Solid-State Disk Simulator Experimental Results
- Ongoing & Future Works
- Summary
- Demo of SSD Simulator

Simulator (I)

- **System-level simulator**
  - Need to model both the host and SSD
  - Need to raise the abstraction level
    - For simulation speed
    - But, loosing accuracy
  - Not actively researched
- **SSD simulator**
  - Need to explore the internal architecture of SSD
  - Must consider both the HW and SW features
    - Cycle-accurate level would be a good choice
  - Not many works satisfy the above requirements

Simulator (II)

- **Merits of SSD Simulator**
  - Quantitative performance analysis
  - HW & SW co-simulation
  - FTL research considering HW characteristics
  - HW optimization for specific algorithms

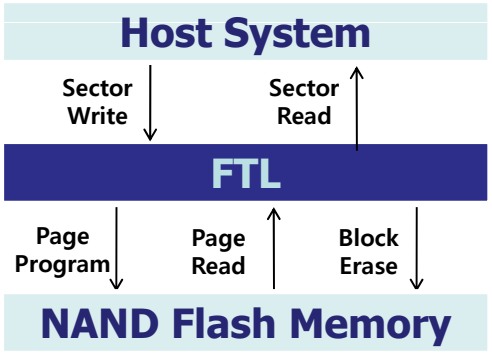
- **Real board is another solution, but …**
  - Hard to prepare
  - Less flexibility
  - Less controllable

FTL perspective (I)

- **Purpose of FTL**
  - **Improve flash memory access pattern**
- **Typical Flash characteristics**
  - **Operation unit**
    - **read / write (program) : page unit**
    - **erase : block unit**
  - **Erase-before-Write**
  - **Asymmetric Operation Latencies**
  - **Erase Count Limitation**
    - **SLC (~100,000), MLC (~10,000)**
- **Have used HW-independent metrics**
  - **Hard to evaluate FTL for various architectures**
  - **Cannot consider its overhead**

FTL perspective (II)

- **HW-dependent FTL Operations**
  - **Mapping**
    - **Logical Address → Physical Address**
    - **Block-level, Page-level, Hybrid**
  - **Garbage Collection**
    - **Merge : Switch, Partial, Full**
  - **Interleaving**
  - **Wear Leveling**
    - **Greedy, Rate-Limiting, Migration**
  - **Bad Block Management**
  - **Meta Data**
  - **Power-Off Recovery**



FTL perspective (III)

- **Various FTL algorithms**
  - **Mostly focus on reducing merging frequencies**
    - **Log-block based FTL (BAST)**
      - **Log block is organized by an out-of-place scheme**
    - **FAST : Fully-Associative Sector Translation**
      - **Log block is shared by all the data blocks**
    - **Super-block based FTL**
      - **A set of adjacent logical blocks that share data blocks and log blocks**
    - **LAST : Locality-Aware Sector Translation**
      - **Using Locality Detector**
      - **Reduces merge by partitioning Hot / Cold log blocks for random writes**
    - **μ-FTL : Minimally Updated FTL**
      - **Using μ-Tree & Bitmap cache**
- **FTL performance Metrics**
  - **R/P/E count**
  - **All metrics are HW independent**
  - **Operation overhead is not considered**
  - **Limited HW architectures**
    - **Single channel and single way**
    - **Ignores memory organization**

HW perspective

- **Major topics**
  - **Channel / Way optimization**
  - **DRAM cache buffer management**
  - **Hybrid Flash memory arrays**
    - **MLC + SLC**
- **Apparently, closely related to FTL**
  - **Not many works have touched this point**

Contents

- Introduction
- Related Works
- Solid-State Disk Simulator Implementation
- What to do w/ Solid-State Disk Simulator
- Solid-State Disk Simulator Experimental Results
- Ongoing & Future Works
- Summary
- Demo of SSD Simulator

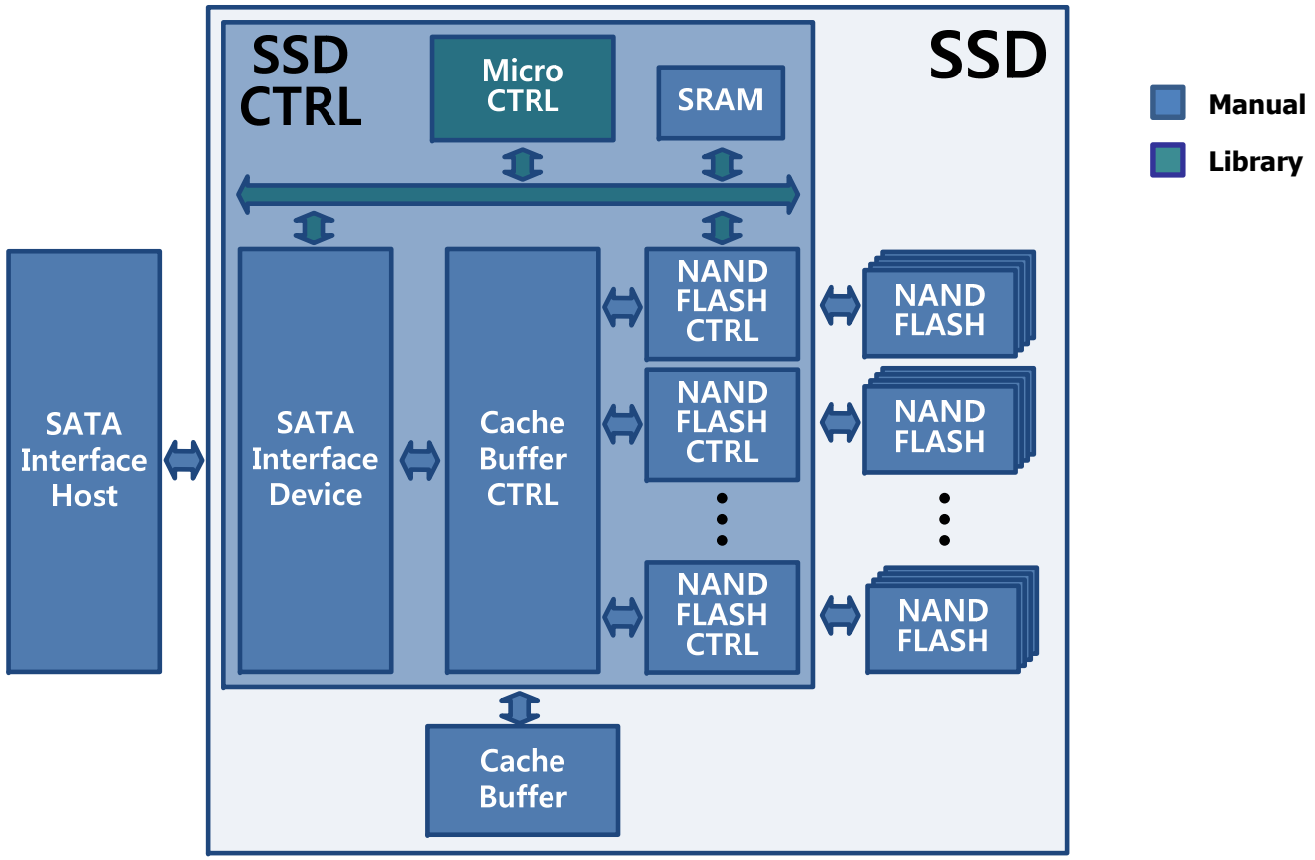
Simulation Environment

- SoC Designer™ (MaxSim™)
  - System-Level Modeling Tool by Carbon Design Systems
  - Fast platform builder using SystemC
    - Simulation speed over 200kcps
    - Easy to build various architectures
  - Component Library is another plus
    - Core / Bus / Memory / Peripheral / …
    - Cycle Accuracy
  - HW & SW co-simulation
  - GUI - Monitor / Profile / Waveform
  - RTL co-simulation

Modeling strategies

- **Cycle-Accurate Modeling**
  - Enough for quantitative performance analysis
- **Flexibility**
  - Parameterization of important variables
- **Simulation speed over 200kcps**
  - SW engineers will complain, but
  - HW engineers will appreciate
- **Trace Driven Simulation**
  - Performance analysis with Real workloads
  - Synthetic workloads for specific cases

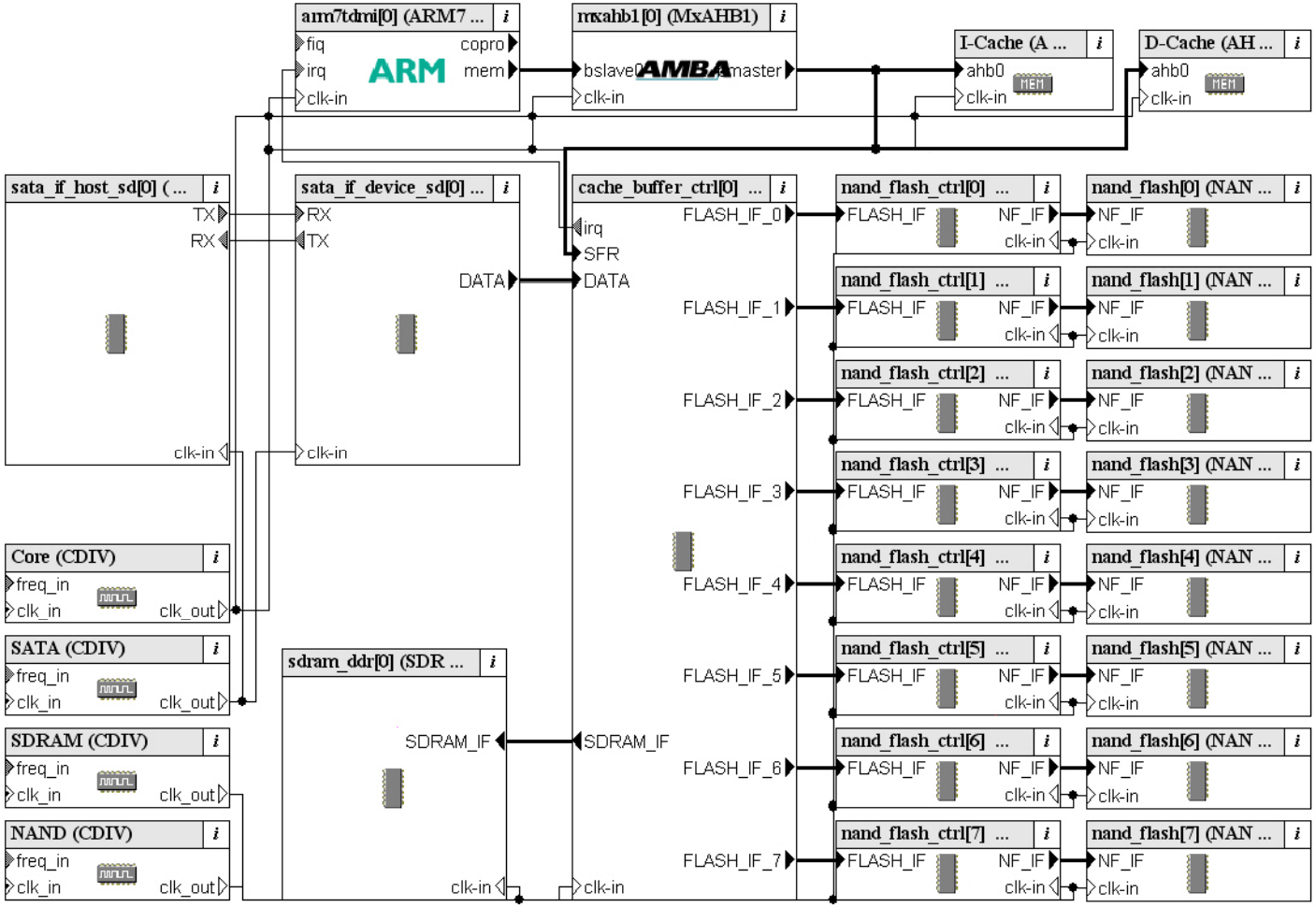
Target Architecture



Component Modeling

- **SATA interface**
  - Trace input
- **SRAM & Controller**
  - FTL code, map table
- **SDRAM & Controller**
  - Cache buffer
  - Decoder : Channel striping
- **NAND Flash Memory & Controller**
  - Controls flash accesses
  - Way interleaving

Completed Target SSD Simulator



Contents

- Introduction
- Related Works
- Solid-State Disk Simulator Implementation
- What to do w/ Solid-State Disk Simulator
- Solid-State Disk Simulator Experimental Results
- Ongoing & Future Works
- Summary
- Demo of SSD Simulator

FTL Optimization / Analysis

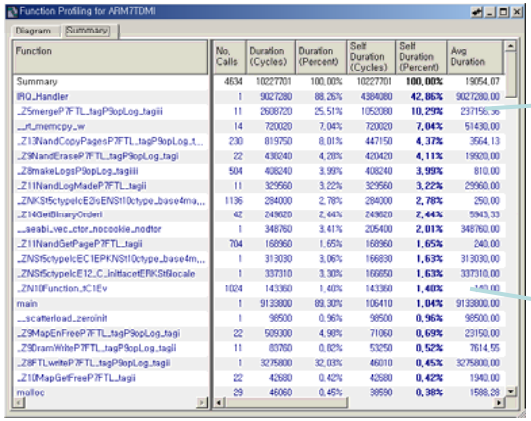
- **Advanced FTL development**
  - **HW-dependant function profiling**
    - **Optimize the functions in the performance bottleneck**
  - **Specialize FTL to the target HW architecture**

- **Fair comparison of various FTL algorithms**
  - **Comparison for a common architecture**
    - **With various architectural variants**
  - **Consideration of their computing overheads**
  - **Comparison for various classes of workloads**

HW/SW Partitioning

Micro Controller

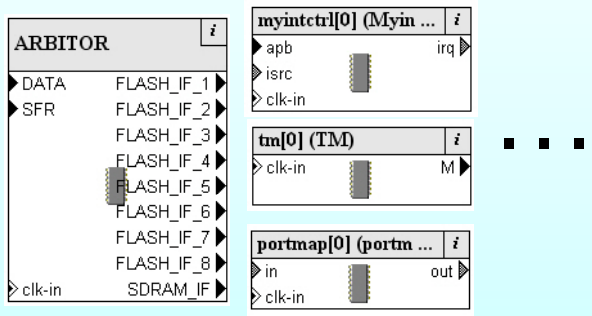
Virtual Profiling



Frequently Used function
& Possible to make HW accelerator

Only SW

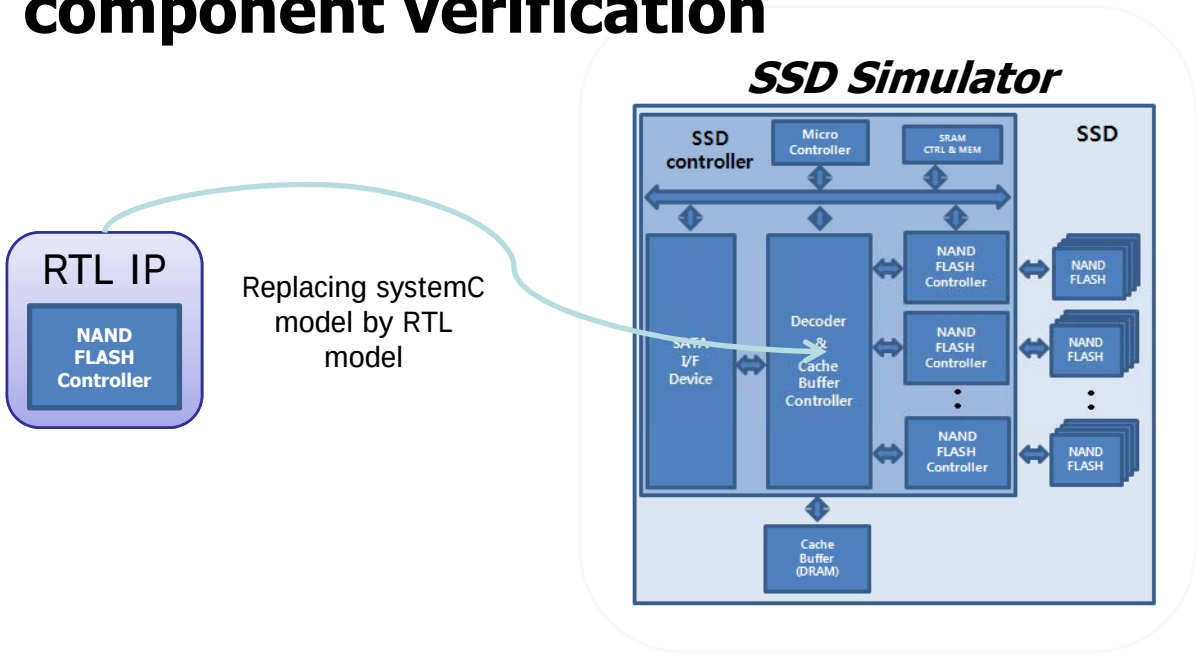
Make Function Component



Performance Improvement !

RTL verification in SSD

- **Vera / Specman / …**
  - **Component-level verification**
- **SSD simulator**
  - **Golden reference of RTL verification**
  - **SSD-level component verification**



Contents

- Introduction
- Related Works
- Solid-State Disk Simulator Implementation
- Analysis using Solid-State Disk Simulator
- Solid-State Disk Simulator Experiment
- Ongoing & Future Works
- Summary
- Demo of SSD Simulator

Experimental Results

❑ Environment Setting

Core	SATA	Cache Buffer	NAND
100 MHz	300 MHz	200 MHz	40 MHz

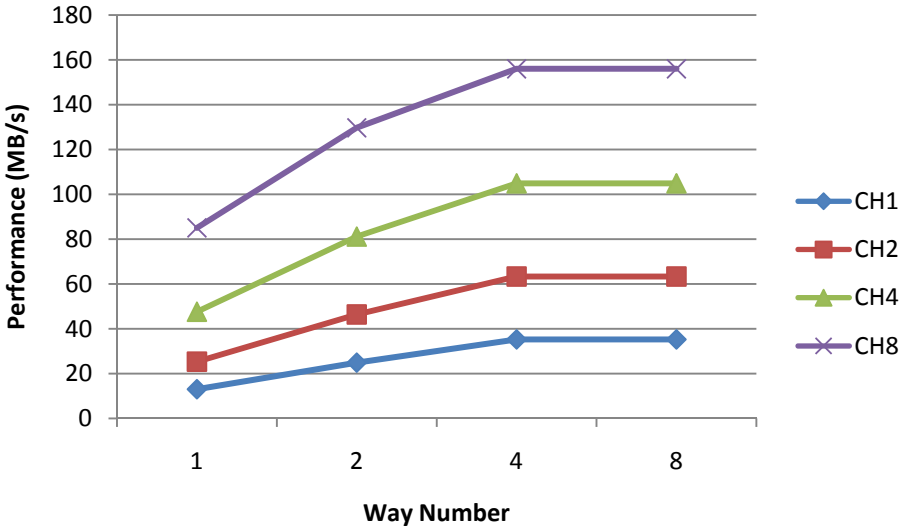
Trace	Size
Random Write / Read	4 KB
Sequential Write / Read	0.5 MB

NAND Flash Memory			SDRAM		
Parameter	Value	Unit	Parameter	Value	Unit
Ways	4		Bit Width/Module	64	Bit
Blocks/Way	4096		Ranks	2	
Planes/Way	1		Banks/Rank	4	
Pages/Block	64		Burst Length	4	
Page Data Size	4096	Byte	Refresh Period	64	ms
Page Spare Size	128	Byte	Size	16	MB
tADL	4	Cycle	tCL	3	Cycle
tBERS	80000	Cycle	tDQSS	1	Cycle
tCCS	3	Cycle	tRCD	6	Cycle
tCEA	3	Cycle	tRFC	51	Cycle
tIEBSY	20	Cycle	tRFCI	3120	Cycle
tIPBSY	20	Cycle	tRP	6	Cycle
tPCBSY	120	Cycle	tWR	6	Cycle
tPROG	8000	Cycle			
tR	1000	Cycle			
tRCBSY	120	Cycle			
tWB	4	Cycle			

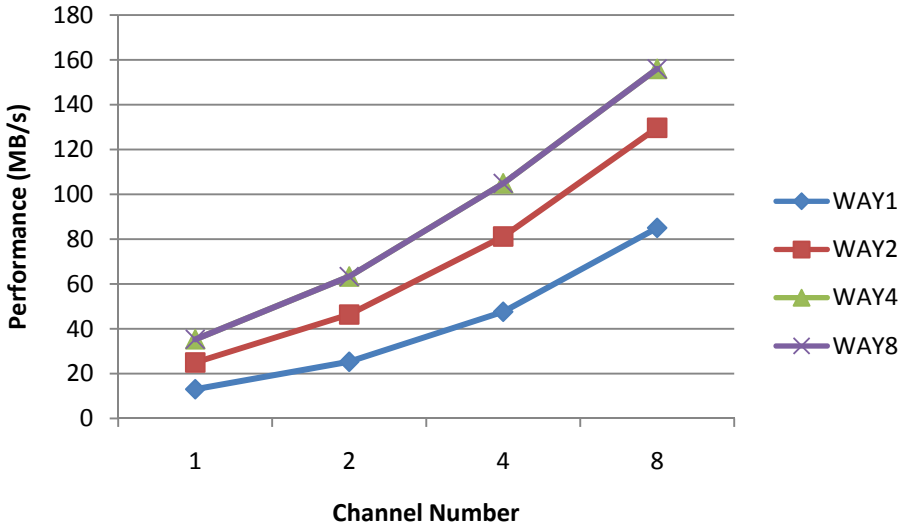
*** NAND Flash cells are initially all erased**

Experimental Results

Channel Effect



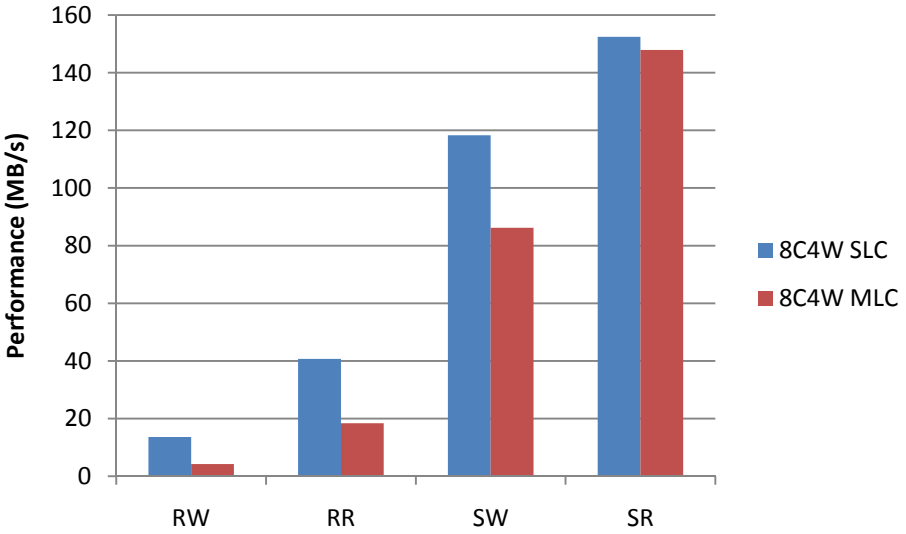
WAY Effect



- **Trace: Sequential Write**
- **NAND Flash: SLC**
- **Channel effect dominates way effect**
- **Saturated at way=4**
- **Providing quantitative analysis**

Experimental Results (cont.)

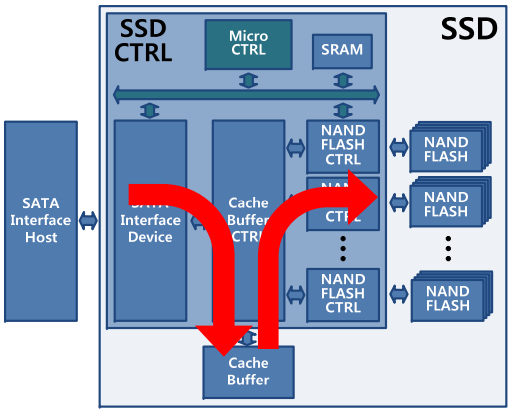
SLC & MLC



- Sequential operation
  - -MLC can catch up SLC in multi channel & way organization
    - -Thanks to channel striping and way interleaving
- Random operation
  - - Cannot benefited by multi channel & way organization

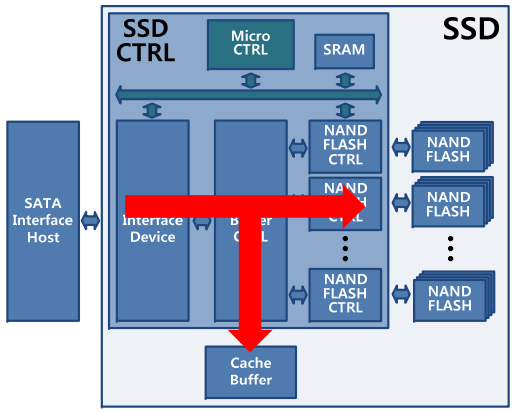
Experimental Results (cont.)

For locality exploitation

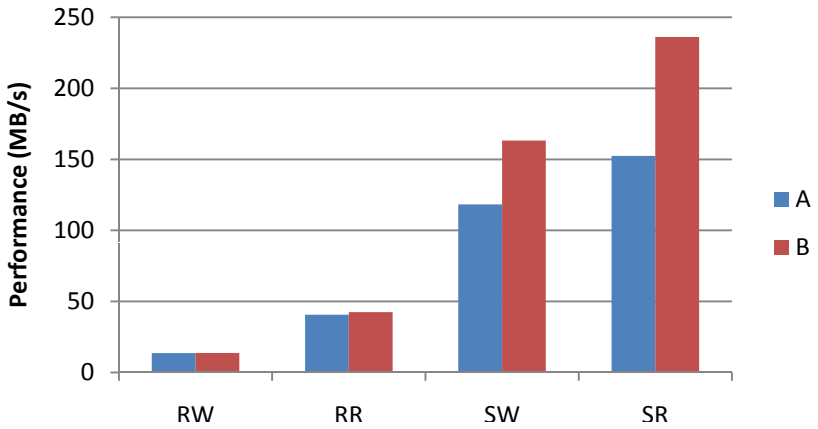


A

Parallelizing sequential ops.



B

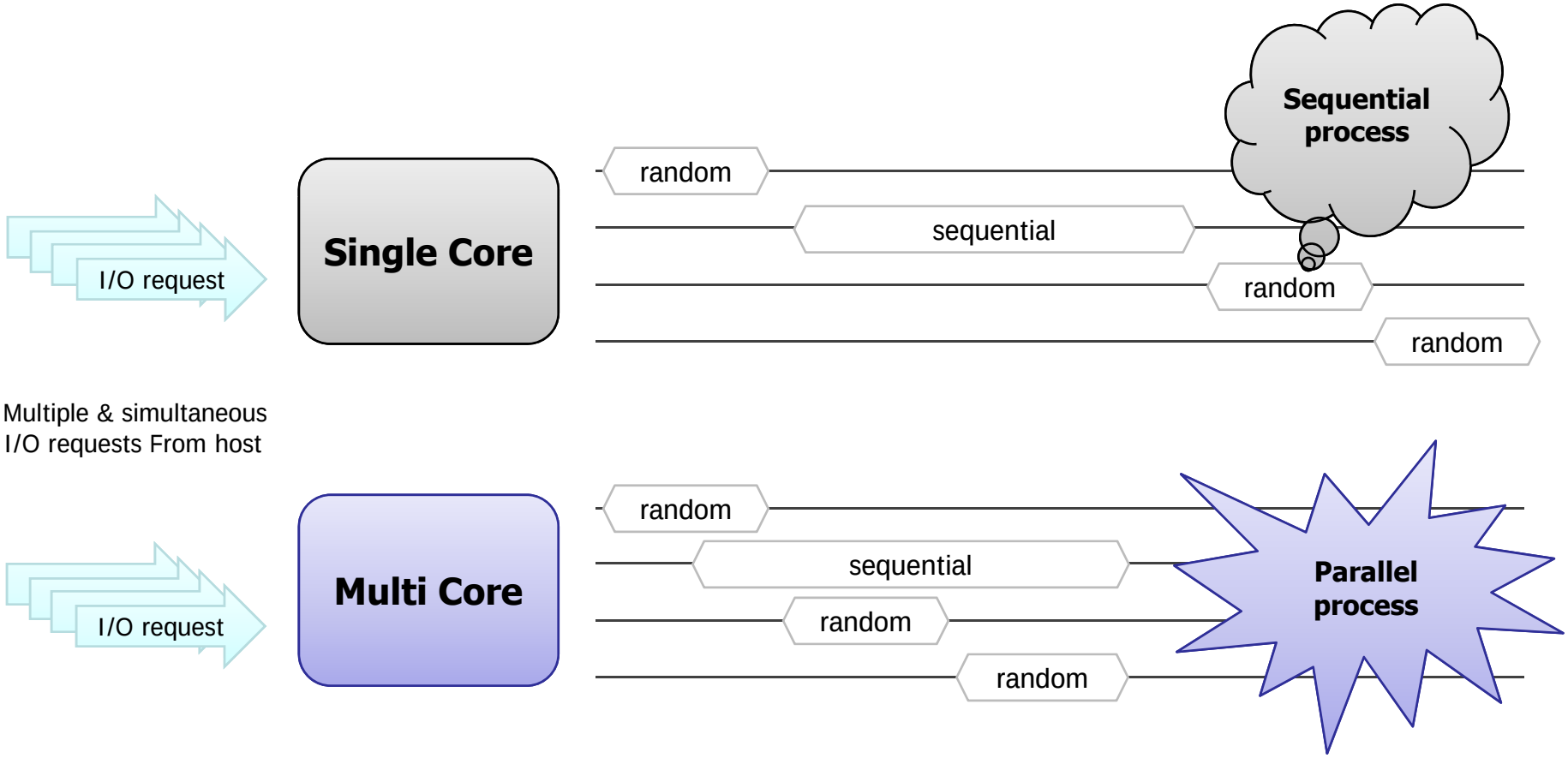


- 8 channel 4 way
- Sequential Requests has benefit at B (direct access)
- In A, we do not perform any locality exploitation

Contents

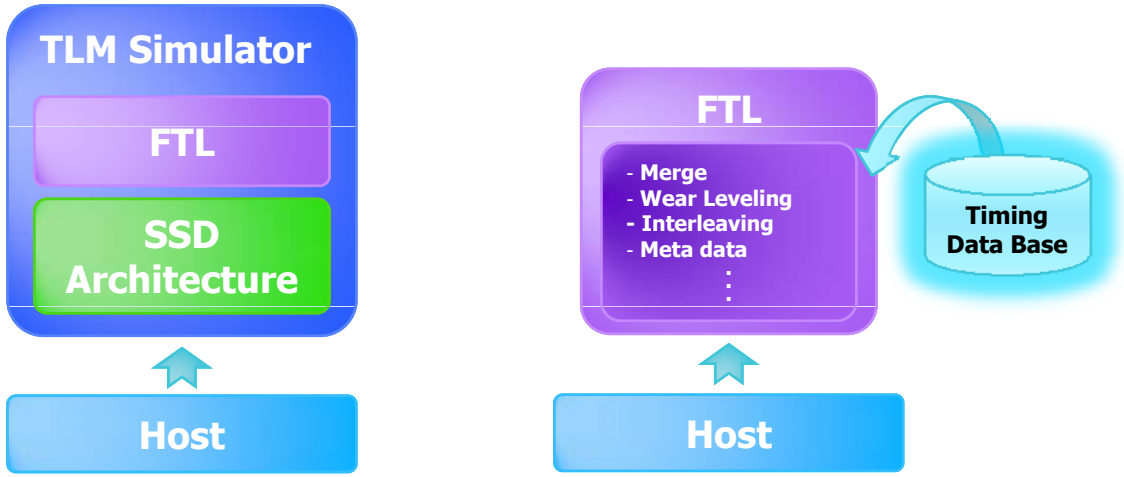
- Introduction
- Related Works
- Solid-State Disk Simulator Modeling
- Solid-State Disk Simulator Usage
- Solid-State Disk Simulator Experiment
- Ongoing & Future Works
- Summary
- Demo of SSD Simulator

Multi-Core SSD Controller



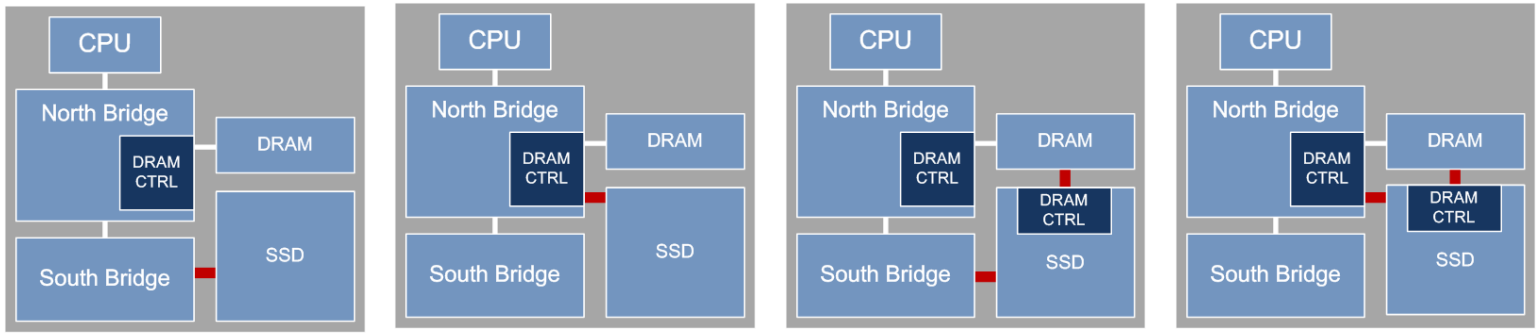
Speed Up SSD Simulator

- **Cycle-aware FTL simulator**
  - **Similar accuracy to TLM**
  - **3 or 4 orders of magnitude faster than TLM**
- **Performance characterization of each operation for each target architecture**



Integration with PC-System simulator

- **PC System Simulator**
  - **To understand the benefit of SSD in PC**
  - **Implementation: SystemC**
  - **Propose architectural enhancement of PC w/ SSD**
    - **Location: South Bridge (SB) or North Bridge (NB)?**
    - **Interface: SATA or other protocols?**
    - **Concurrency: Single port or dual port?**



A) SP SB Architecture B) SP NB Architecture C) DP SB Architecture D) DP NB Architecture

Contents

- Introduction
- Related Works
- Solid-State Disk Simulator Modeling
- Solid-State Disk Simulator Usage
- Solid-State Disk Simulator Experiment
- Ongoing & Future Works
- Summary
- Demo of SSD Simulator

Summary

- Growth of NAND Flash Memory & SSD Market
- FTL has been studied w/o HW characteristics
- Limited HW architecture were explored
- SSD Simulator is advantageous for
  - Quantitative performance and analysis
  - Architecture exploration
  - FTL comparison and optimization
  - HW/SW partitioning
  - Golden reference for RTL design

Contents

- Introduction
- Related Works
- Solid-State Disk Simulator Modeling
- Solid-State Disk Simulator Usage
- Solid-State Disk Simulator Experiment
- Ongoing & Future Works
- Summary
- Demo of SSD Simulator

Demo of SSD Virtual Platform

- **SoC Designer Canvas**
  - **Easy to use like schematic capture system**
  - **Flexibility : Easy to explore architecture**
- **SoC Designer Simulator**
  - **HW/SW Co-simulation & Debug**
    - **HW : Profiling, Tracing, Waveform, Monitor**
    - **SW : RVDS**