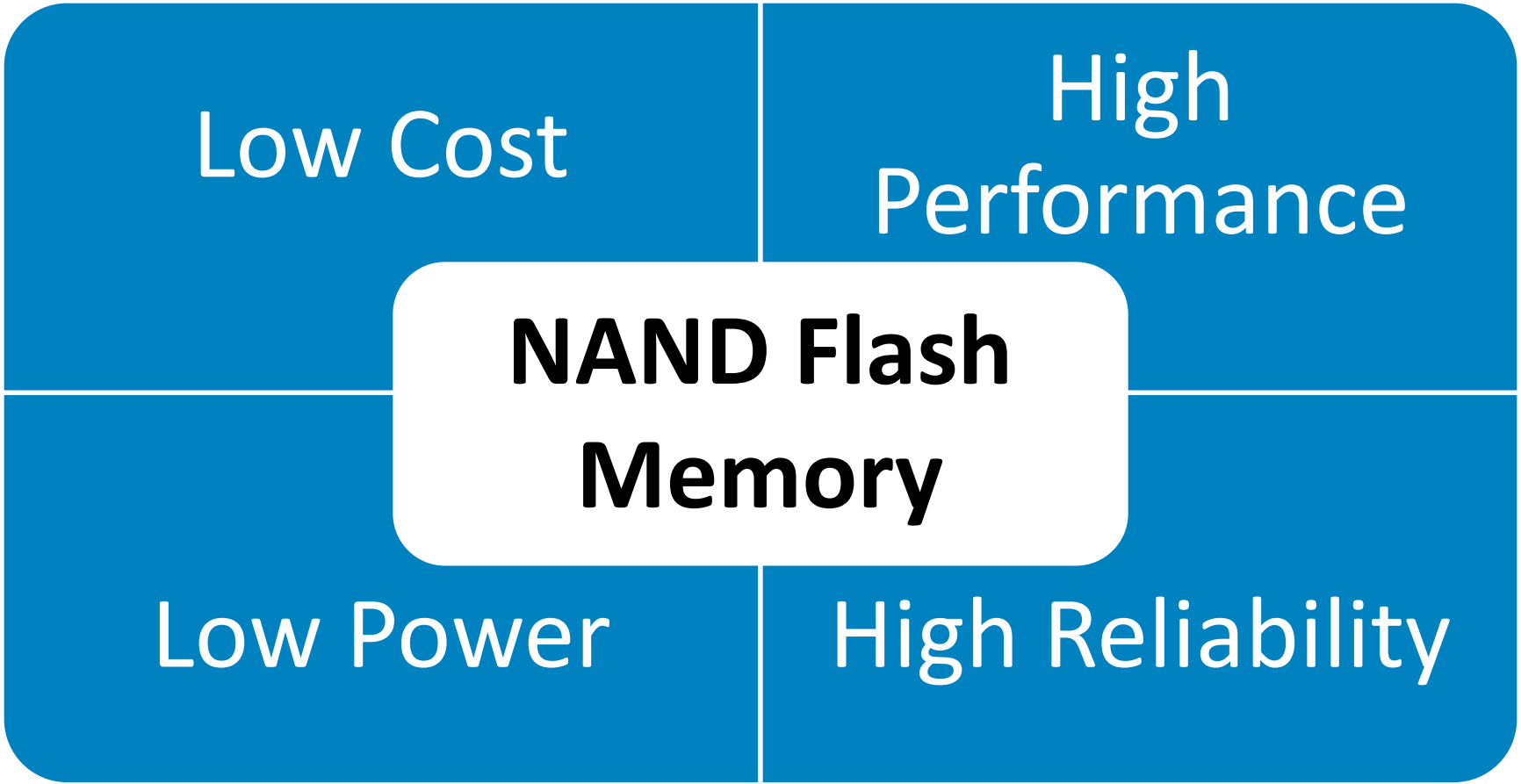
Towards Building Energy Efficient, Reliable and Scalable NAND Flash Based Storage Systems

Vidyabhushan Mohan

Ph.D. Dissertation Defense

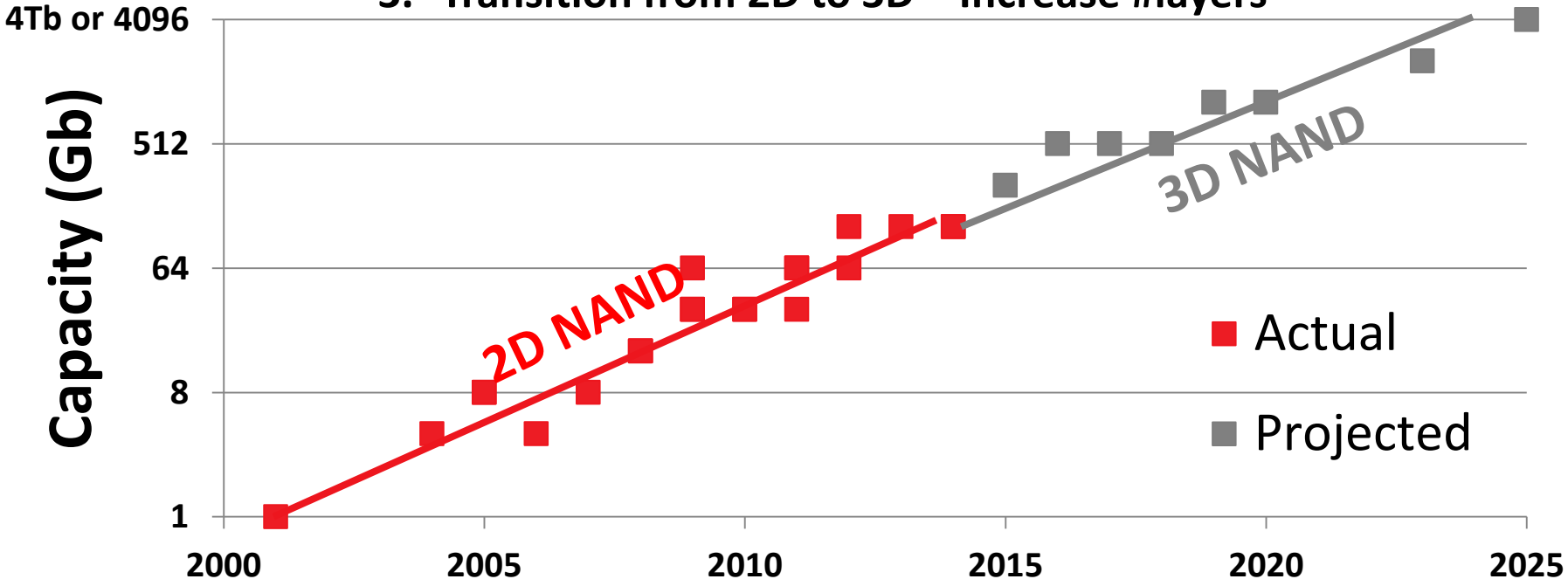
Department of Computer Science, University of Virginia

10/16/2015


Low Cost
High Performance
NAND Flash Memory
Low Power
High Reliability

NAND Flash Scalability

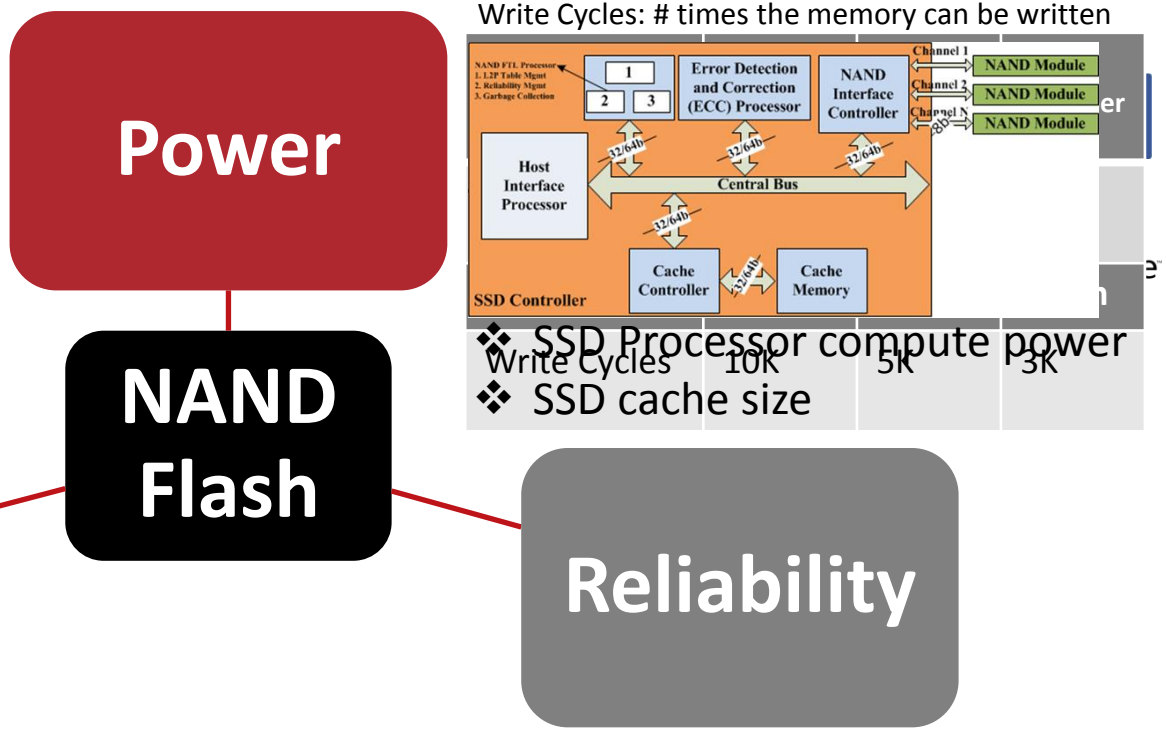
1. Geometric Scaling – shrinking transistors
2. Logical scaling – multiple bits per cell
3. Transition from 2D to 3D – Increase #layers



Actual data from ISSCC proceedings and projections from ITRS 2013

Major NAND Flash Storage System Challenges

- ❑ System heterogeneity
- ❑ Impact of memory scaling
- ❑ Sub-optimal architectures
- ❑ Lack of modeling tools



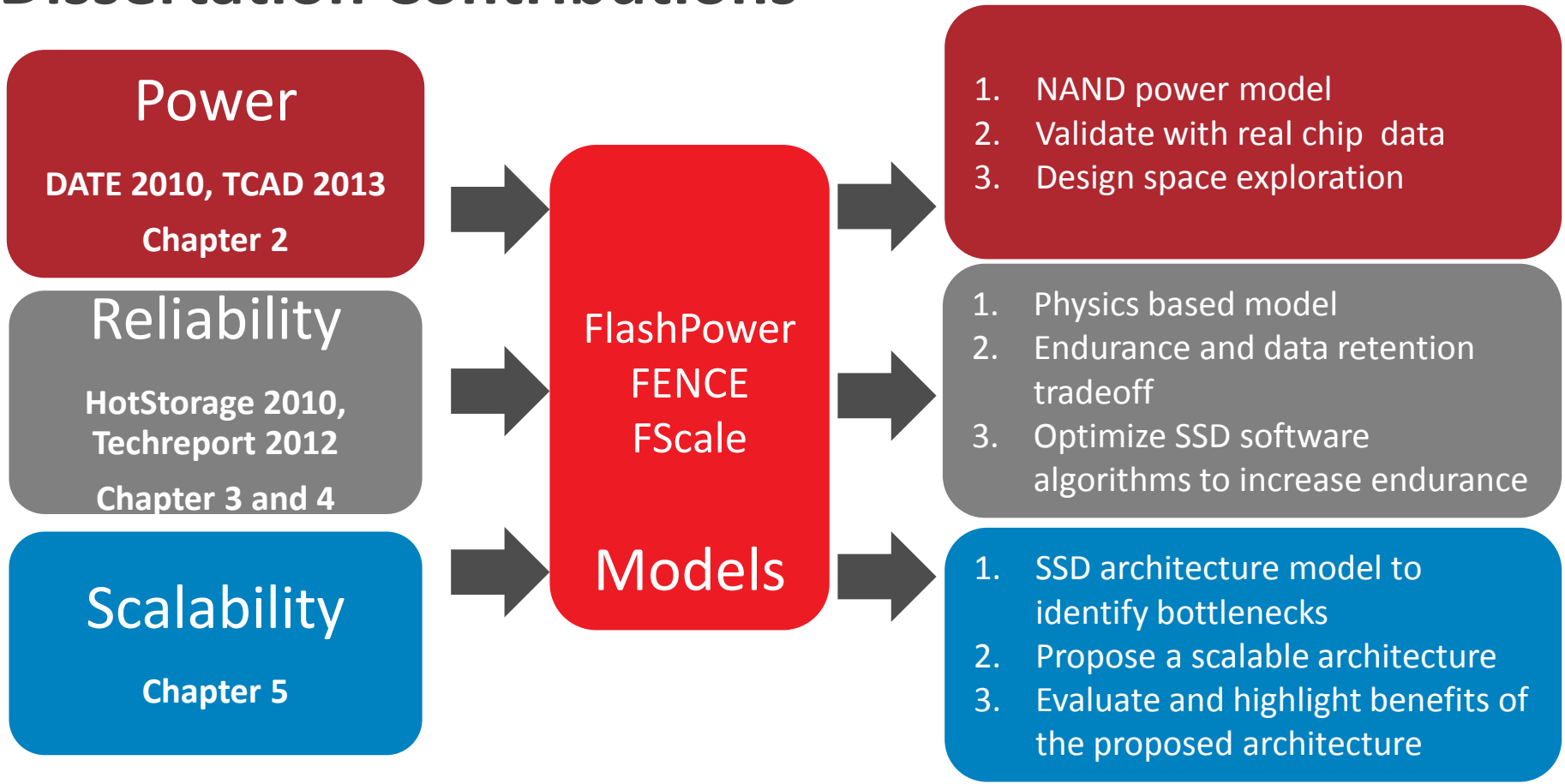
System Heterogeneity: Flash in cameras optimized for absolute power. Flash in data centers optimized for power efficiency

[1] http://www.flashmemorysummit.com/English/Collaterals/Proceedings/2011/20110810_T1A_Maislos.pdf

Dissertation Goals

- Develop architecture level models to study power, reliability and performance of flash based storage systems
- Explore dependencies and tradeoffs between various metrics
- Design algorithms and architectures to develop application optimal storage systems

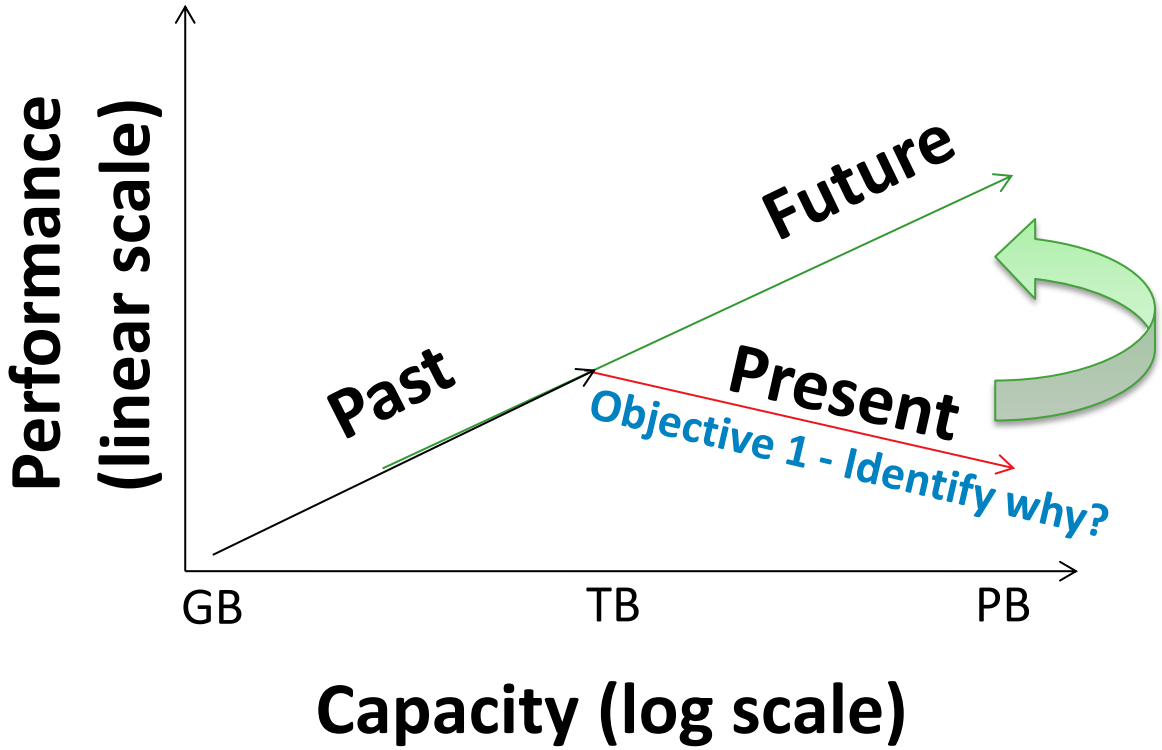
Dissertation Contributions



Outline

- Scalability
- Reliability
- Power

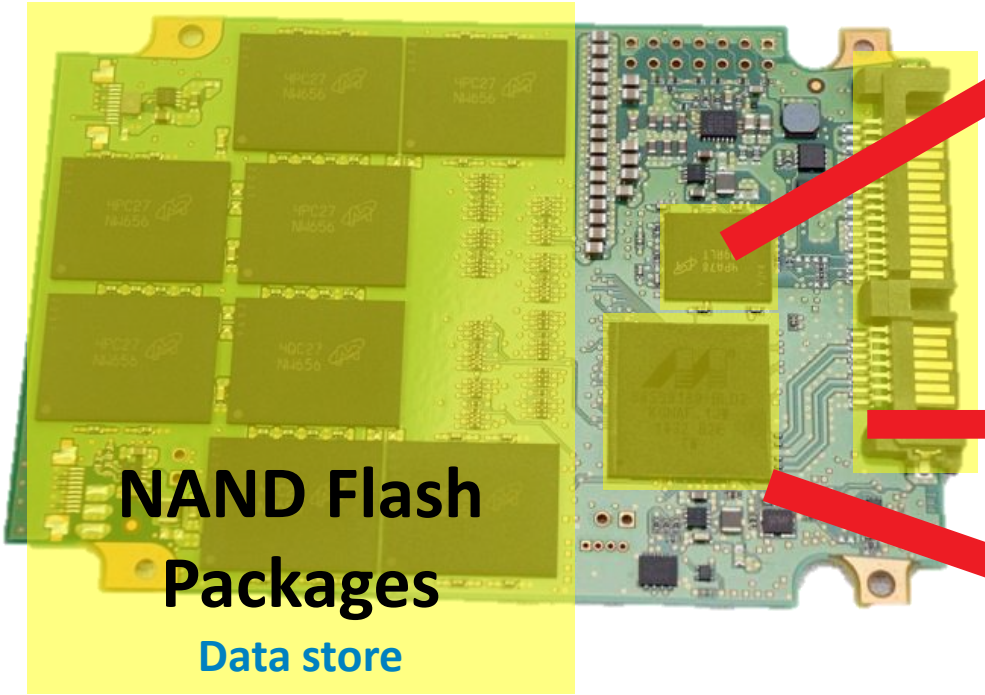
Growth in Storage System Performance – Past, Present and Future



Objective 2 of this work

Source: HGST[1], SanDisk[1,2], Toshiba[1], Samsung[1,2], Micron[1]

Conventional SSD Hardware Architecture



NAND Flash Packages

Data store

DRAM Cache

Cache Logical to physical (L2P) address translation table

Host Interface

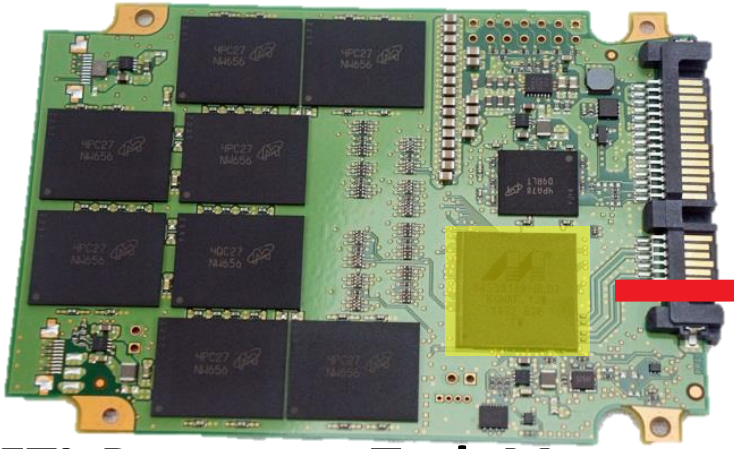
SATA, SAS or PCIe

SSD Controller

Executes SSD's secret sauce, the Flash Translation Layer (FTL)

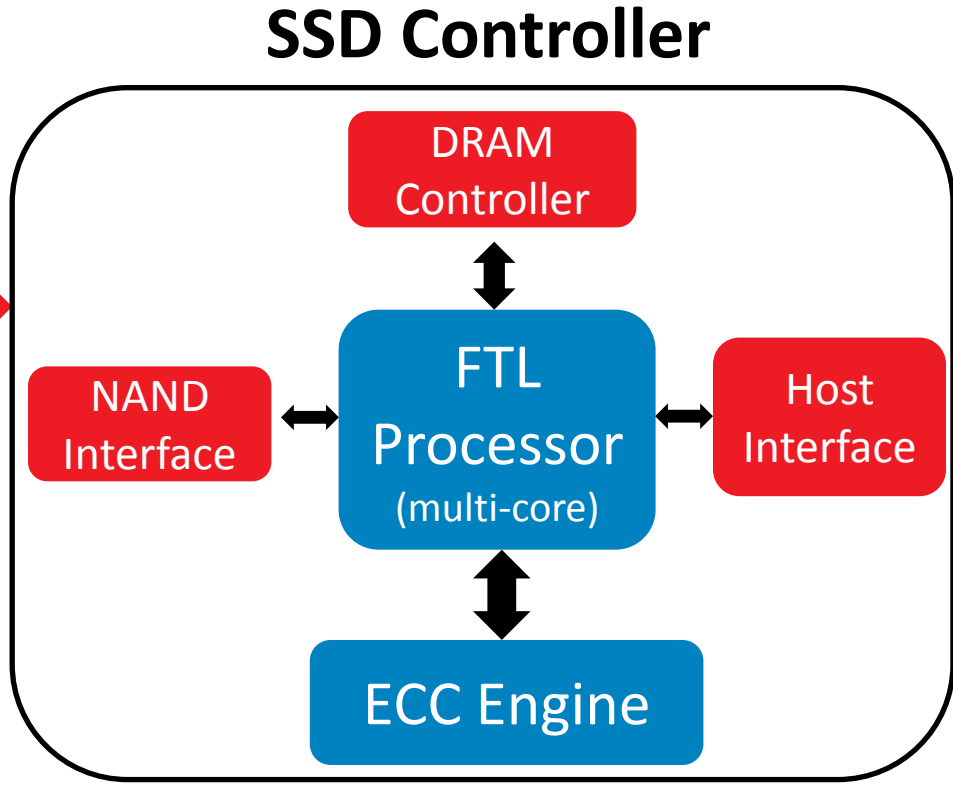
Image source: http://www.thinkcomputers.org/crucial-mx100-512gb-solid-state-drive-review/2/

SSD Controller Internals



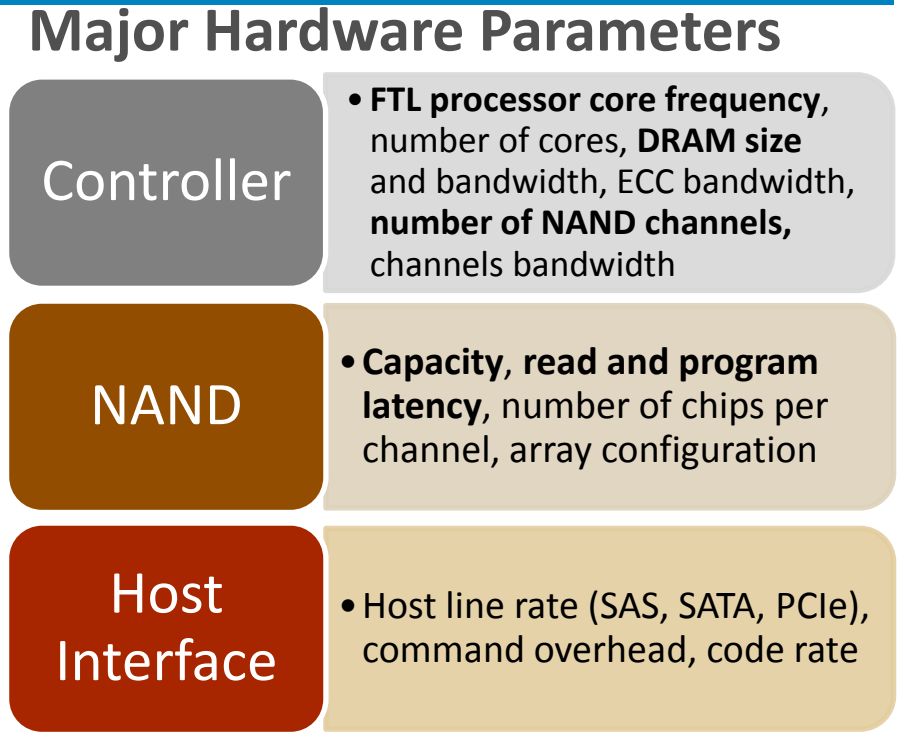
FTL Processor Task Management

1. Host I/O
2. Logical to physical (L2P) table
3. Garbage Collection (GC)
4. Wear leveling (WL)



SSD Hardware Model

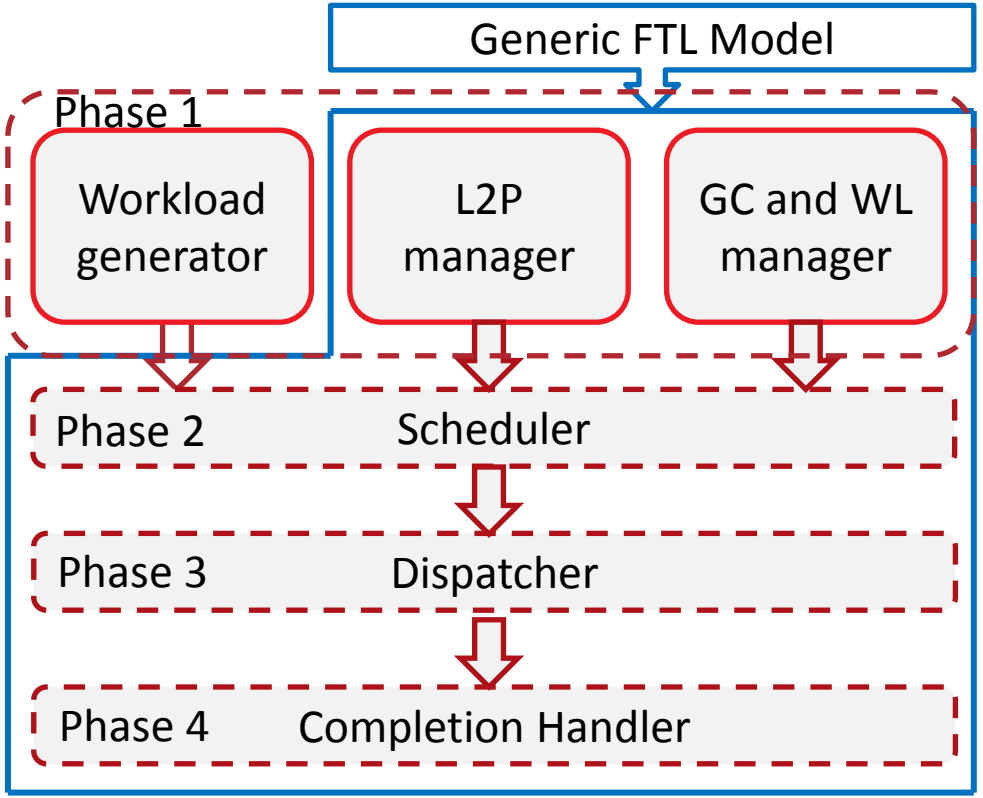
- Developed a generic SSD model with low effort to change and high observability
  - Enables architectural exploration
  - Tool: Intel Cofluent Studio
  - System C based
  - Functional Model



Important parameters in **bold**

SSD Software (aka) Flash Translation Layer (FTL) Model

- SSD's Secret Sauce
  - Existing work(s) treat FTL processor as a black box
  - Generic FTL model that abstracts out the details
- Common FTL Tasks
  - Host I/O management
  - Logical to physical (L2P) address table translation management
  - Garbage collection (GC)
  - Wear leveling (WL)



Flash Translation Layer (FTL) and Workload Model

- FTL Model
  - FTL Task latency for each phase and task (measured in processor cycles)
  - Measured from enterprise drives with varying capacity 100GB to 1TB
- Workload Model
  - A synthetic workload generator similar to IoMeter
  - Used by several customers to evaluate SSD performance
  - Focus mainly on random I/O workload performance

Major FTL and Workload Parameters

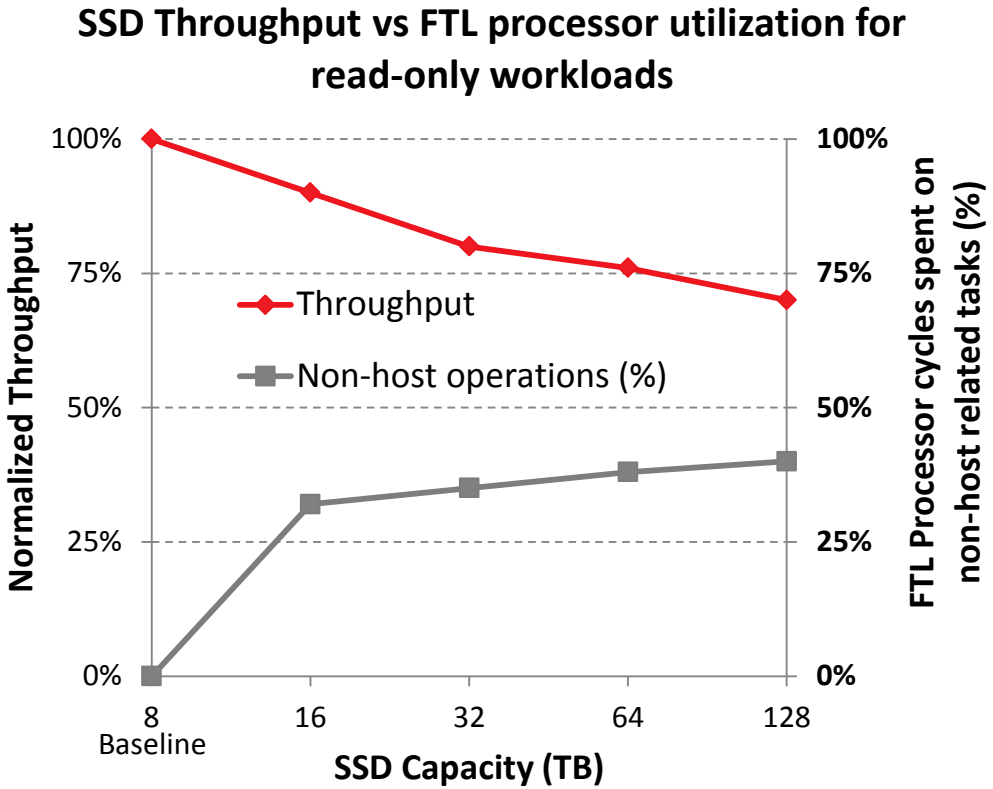
FTL	• **Logical page size**, L2P table hit rate, **write amplification (WA) factor**, cycles per FTL operation type and phase
Workload (Iometer)	• **Read/Write ratio**, IO request size (4K, 8K, and so on), **Queue depth**

Important parameters in **bold**

Metrics, Workloads and Baseline

- Metric: Normalized Throughput (I/O operations per second (IOPS))
  - Baseline: 8TB SSD with 8GB DRAM cache
  - L2P table fits in the cache
- Workload: 100% random workload (based on Iometer)
  - Read/write ratio: 0% (write only) to 100% (read only)
  - Host queue depth: 1 to 256

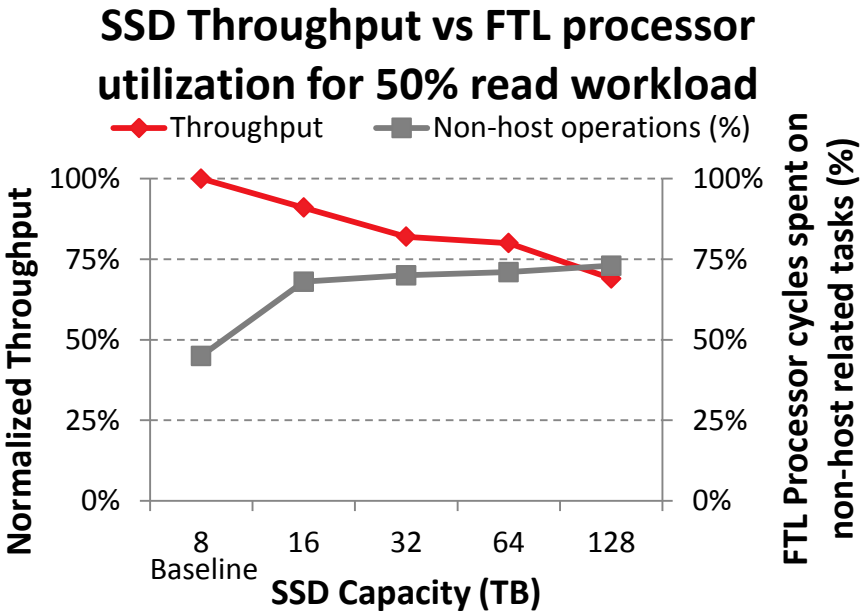
Read-only workload performance in conventional SSDs

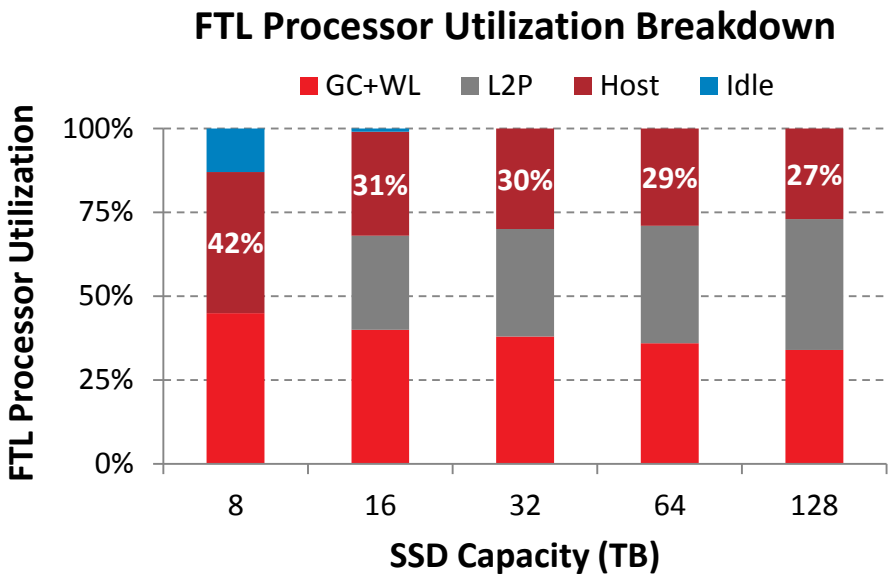


Observations

- Performance of the drive decreases as SSD capacity increases
- Performance impact due to L2P table size resulting in high cache miss rate
  - Minimal GC or WL operations for read-only operations

Mixed workload performance in conventional SSDs





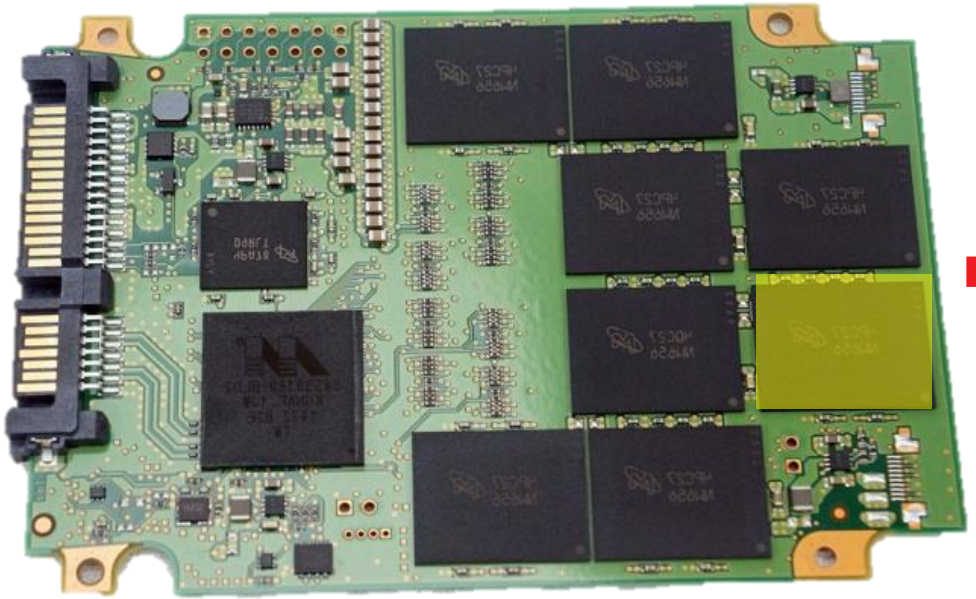
Observations:

1. Performance decreases with SSD capacity
   - ❑ % of FTL processor cycles spent on host operations reduces with capacity

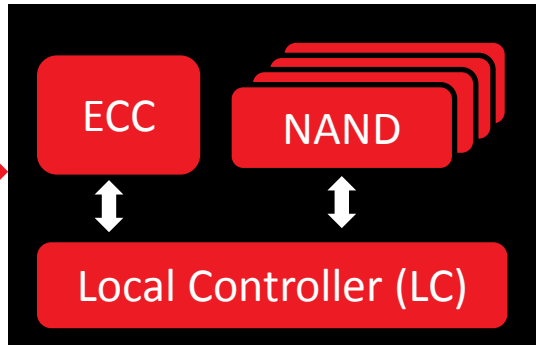
FScale: A Performance Scalable SSD Architecture

- Motivation
  - Reduce L2P table size
  - Increase FTL processor compute power
- FScale: A distributed processor based SSD architecture
  - Hardware architecture scales FTL processor power
    - Replace passive NAND packages with Active NAND packages
  - Software architecture
    - reduces L2P table size by converting the table into a hierarchy
    - distributes operations to take advantage of distributed processor

FScale: Distributed Hardware Architecture

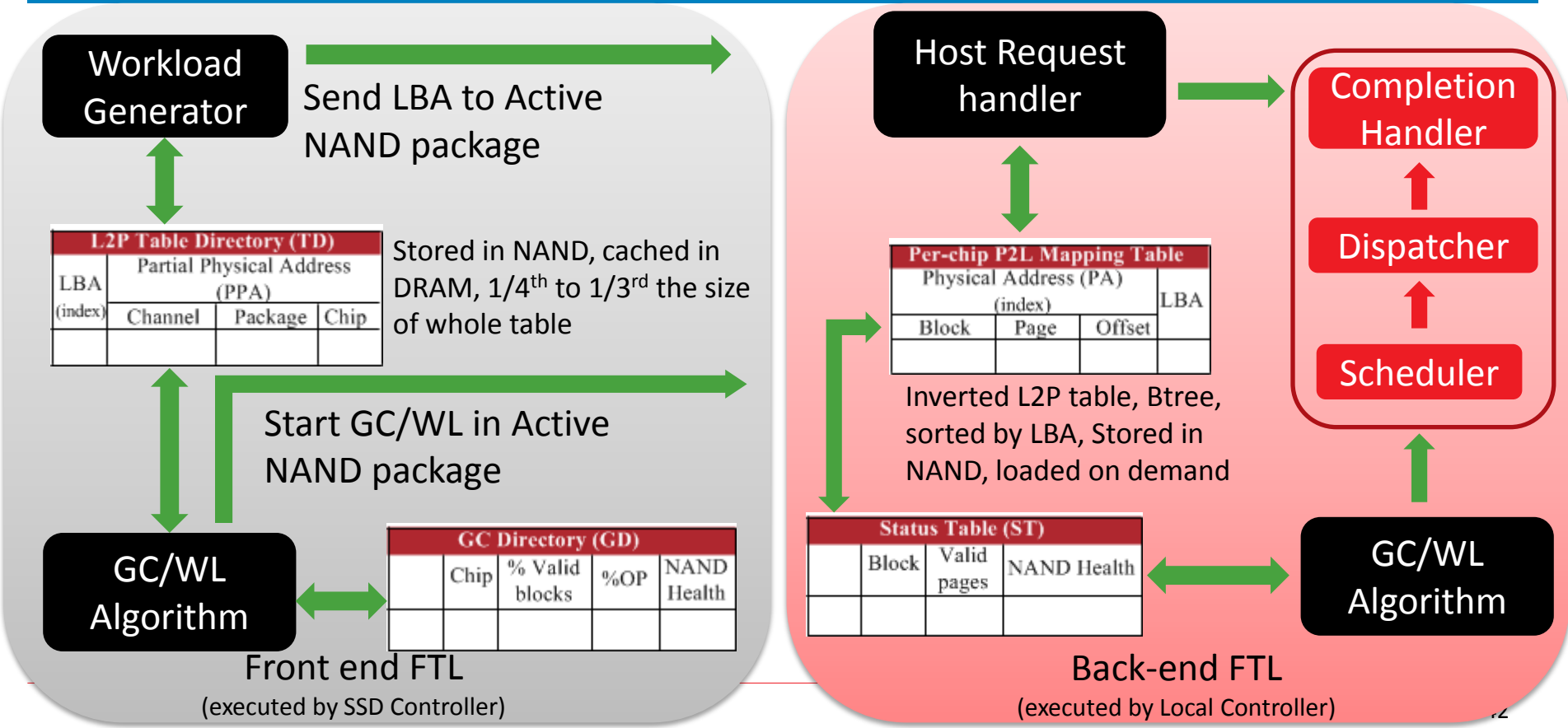


Active NAND Package

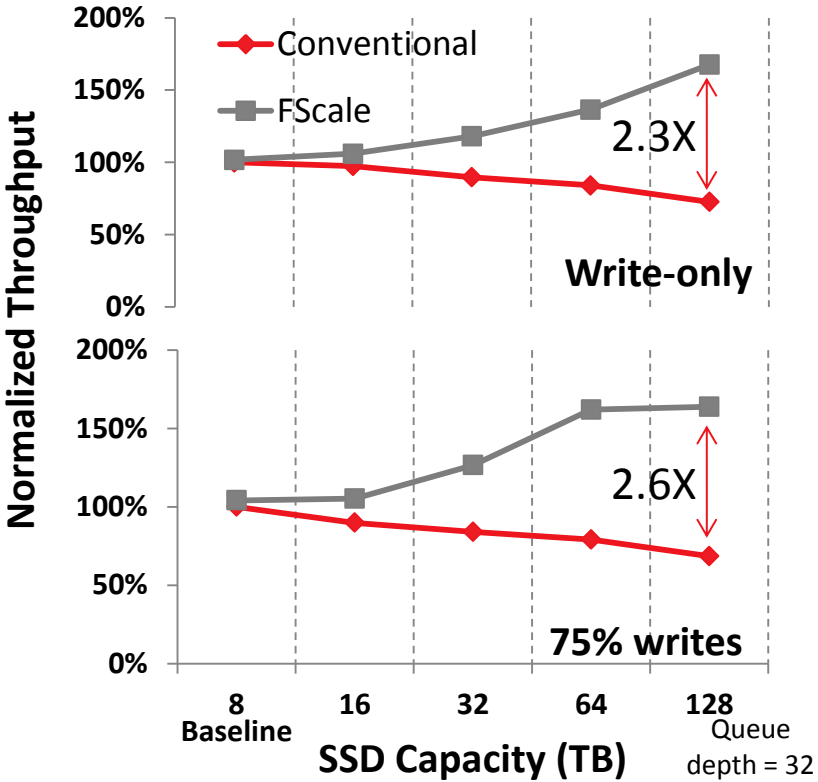


1. Low power local controller (LC)
2. Ability to correct NAND errors with mini-ECC engine
3. L2P address translation (in LC)
4. GC and WL (in LC)

FScale: Distributed FTL Architecture



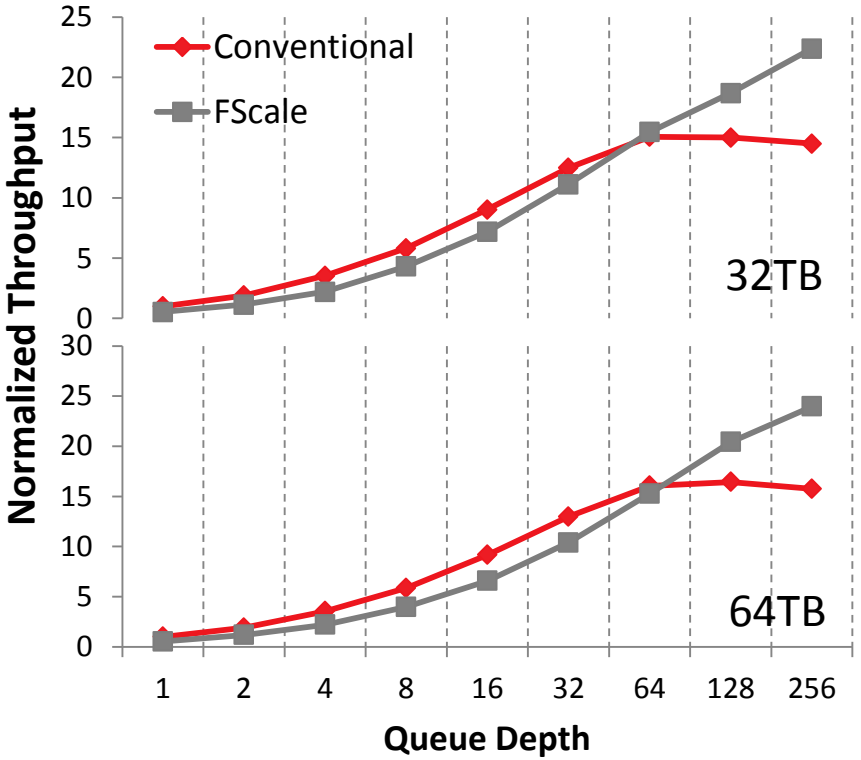
FScale Architecture Performance Evaluation
Write-intensive Workloads



Observations

- ❑ FScale performance scales with capacity and better than conventional SSD performance by offloading GC and WL to local controllers
- ❑ More than 2X increase in SSD performance

FScale Architecture Read-only Workload Performance at Various Queue Depths



- **Observations**
- Low Queue Depth (QD<=32) performance lower than conventional
  - Latency impact of P2L table access
- High queue depth (QD>32) performance higher than conventional architecture
  - Pipelining and handling more requests hides latency impact

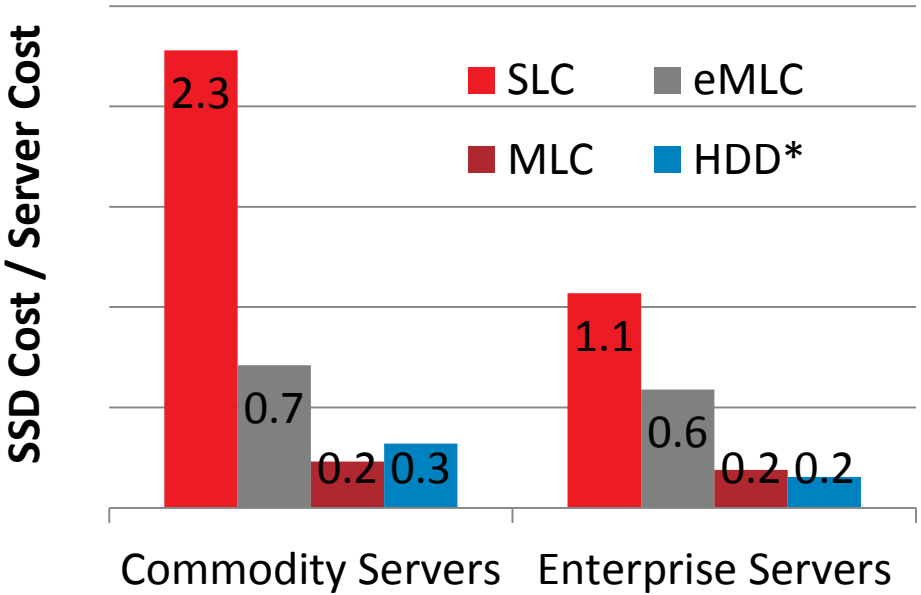
Summary

- Developed a SSD hardware and FTL model
  - Using measurements from real drives
- Show that FTL processor compute power and DRAM cache size are primary bottlenecks for scaling SSD performance
- Propose and evaluate FScale, a scalable distributed processor based architecture to overcome the bottlenecks
- Work done at SanDisk
- Presented a subset at SanDisk Technical Conference (internal)
- Filed 4 patent applications based on this work
- To submit to USENIX ATC

Outline

- Scalability
- Reliability
- Power

Cost of SSDs in Datacenters



Type of SSD	$/GB [1]	Relative Endurance @ 3xnm
SLC	20	8x
eMLC	6	2x
MLC	2	1x

MLC NAND flash is the most cost efficient…….

But have the least reliability

1 bit per cell - Single Level Cell (SLC)
2 bit per cell - Multi-Level Cell (MLC)
2 bit per cell – enhanced Multi-Level-Cell (eMLC)

[*] Assumes 4 HDD per SSD to attain equivalent performance
[1] As of 2011, but the cost ratios are similar in 2015

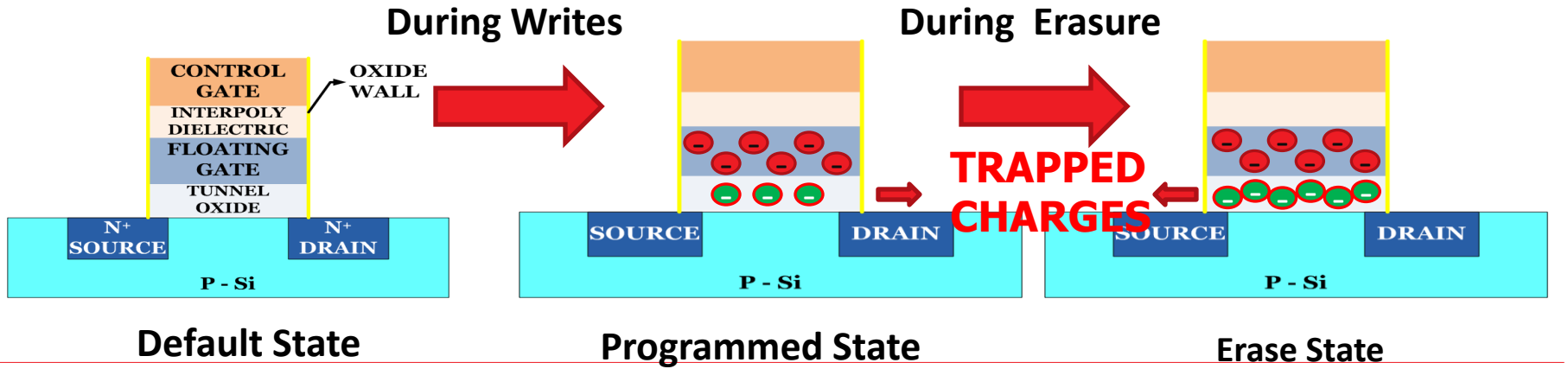
Motivation

How to make MLC SSDs **reliable enough** to use in data centers?

Specifically, how to **increase endurance** of MLC SSDs in data centers?

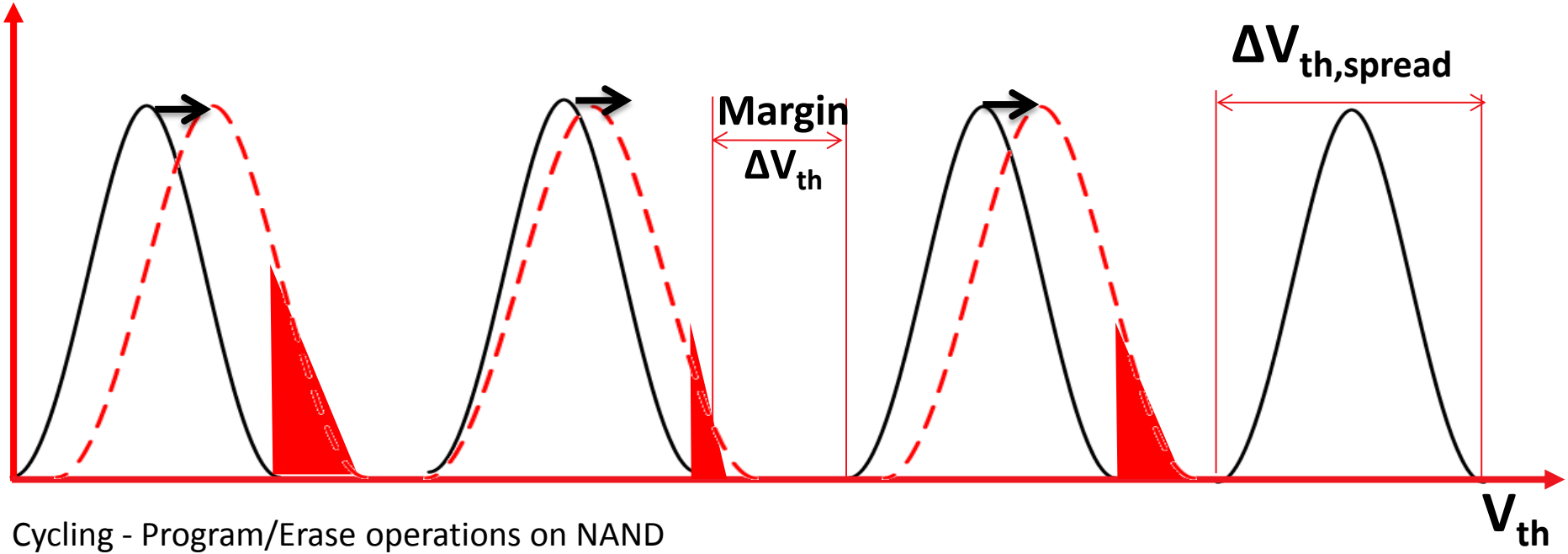
Cycling

- Writes and Erase – stress events
- Side effect of cycling
  - Trapped charges
  - Increase in threshold voltage ($\Delta V_{th,s}$)



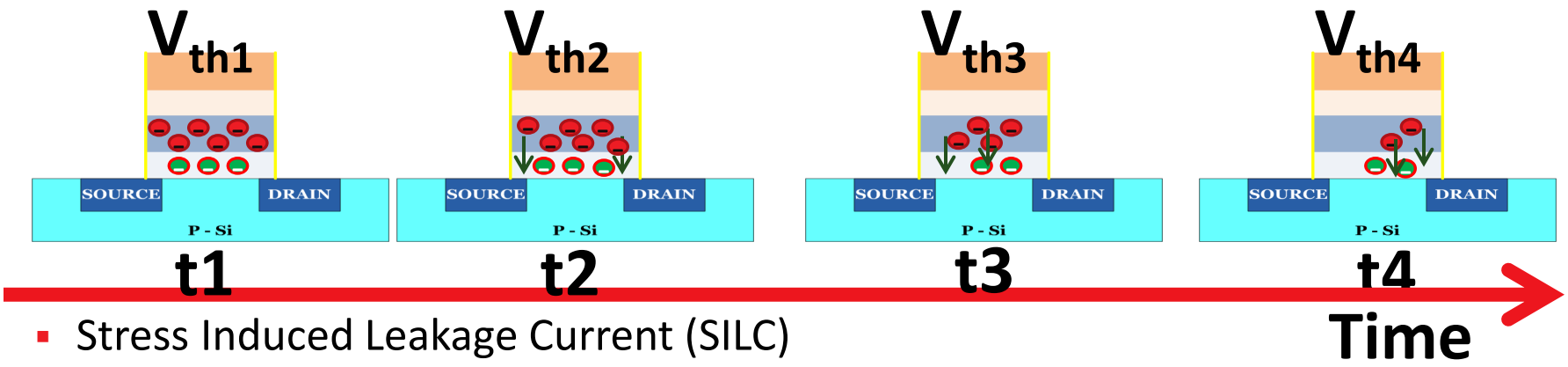
Change in Threshold voltage Distribution with Cycling

Program/Erase cycles increase charge trapping in tunnel oxide which reduces available margin by moving distributions to the right



Cycling - Program/Erase operations on NAND

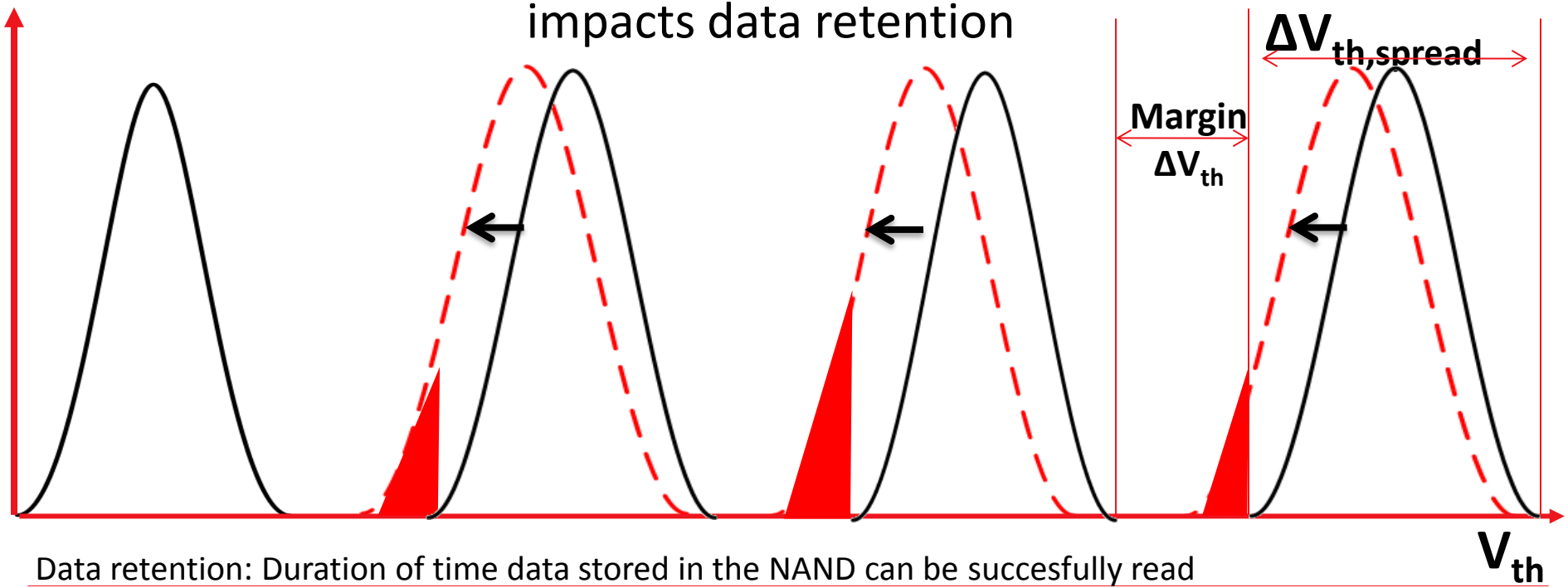
Data Retention



- Stress Induced Leakage Current (SILC)
  - Charge leakage due to trap assisted tunneling
- $\delta V_{th} = V_{th1} - V_{th4}$
  - δV_{th} == Margin => Data retention failure
- Data Retention Time ($t_{retention}$)= t4 –t1
- Exacerbated by temperature

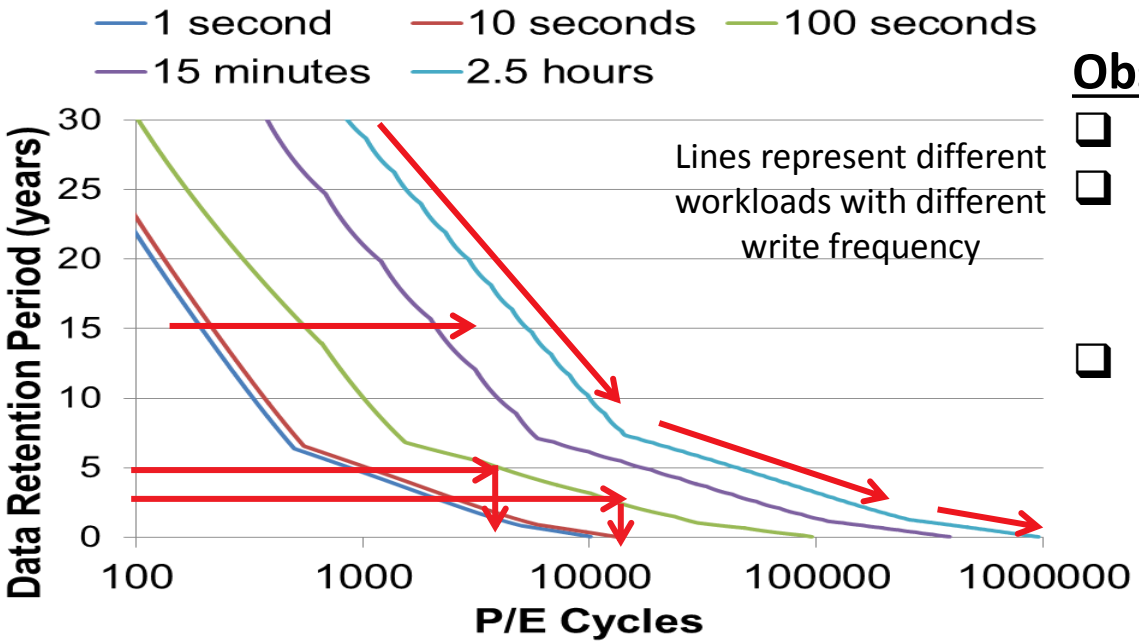
Change in Threshold Voltage Distribution with Stress Induced Leakage Current (SILC)

SILC reduces available margin by moving distributions to the left and impacts data retention



Data retention: Duration of time data stored in the NAND can be succesfully read

Cycling vs Retention for 2-bit MLC Flash



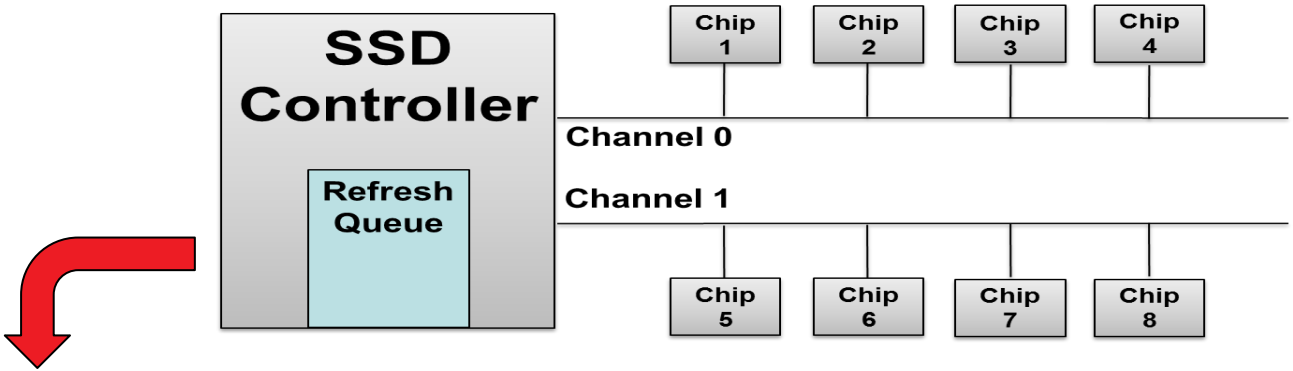
Observations

- ❑ Slope of decay varies with cycles
- ❑ Higher recovery periods increase SSD endurance and data retention
- ❑ Relaxing data retention requirement (y-axis) increases flash endurance (x-axis). Exploit this in datacenters SSDs!

Major Parameters

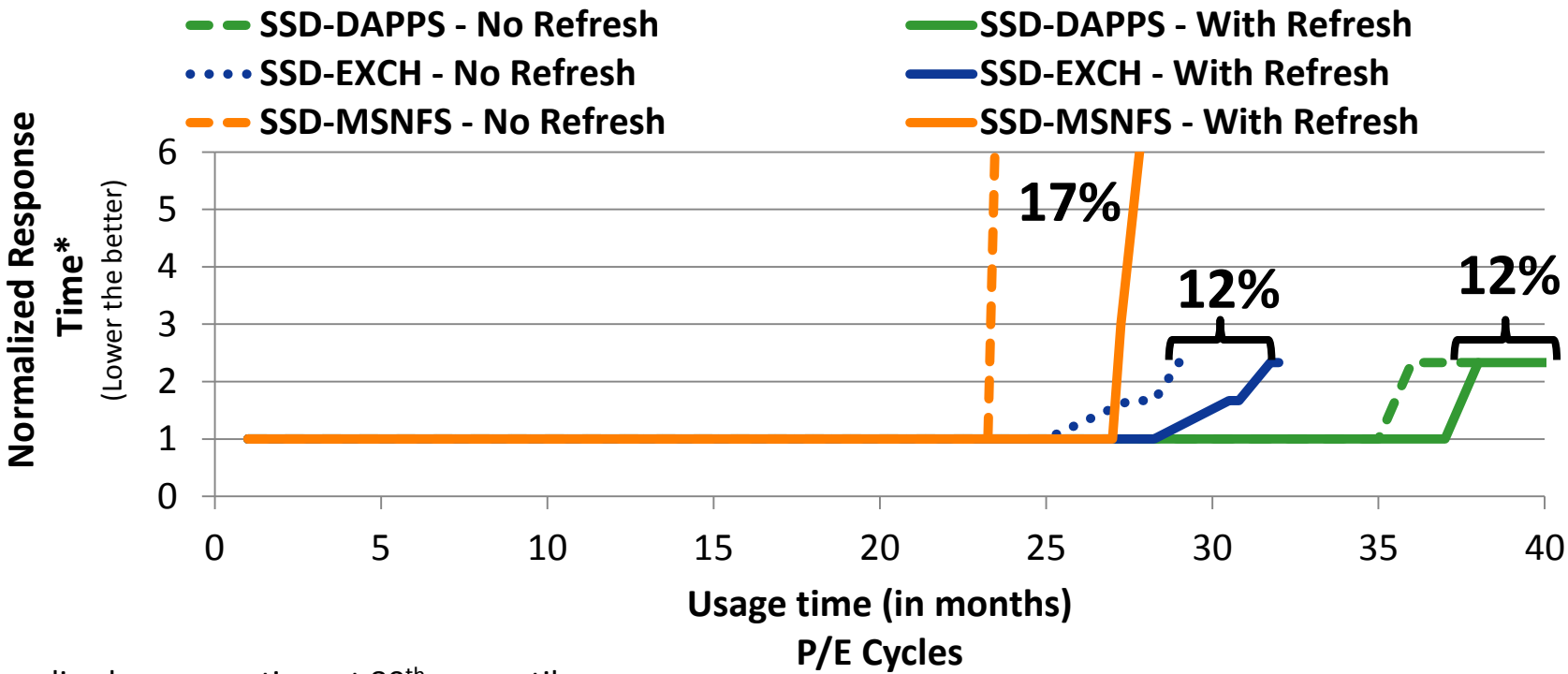
F =80nm, Temperature = 30C, Values for model parameters derived from empirical data collected from device level experiments. References in slide 70

reFresh SSDs: Architecture



- Useful for enterprise applications which do not require high data retention.
  - Tradeoff retention for higher endurance
- Refresh Queue
  - Managed by the SSD controller
  - Queue entries – Pointers to physical flash blocks that have valid data
  - Priority queue – Sorted by block lifetime
  - FENCE model estimate used to determine queue ordering

Evaluating reFresh SSDs with 1 month Data Retention



*Normalized response time at 80th percentile

Summary
Published at Hot Storage 2010 and UVa Tech Report 2012

- Physics based model for NAND endurance and data retention
- Abstracts low level reliability issues into application and system level reliability
- Major parameters
  - Cycling: Number of program and erase operations
  - Recovery period: Time between cycles
  - Drive operation temperature
  - Flash technology
- Quantify the tradeoff between endurance and data retention
- Propose new firmware algorithm to exploit the tradeoff and increase SSD endurance

Outline

- Scalability
- Reliability
- Power

Motivation

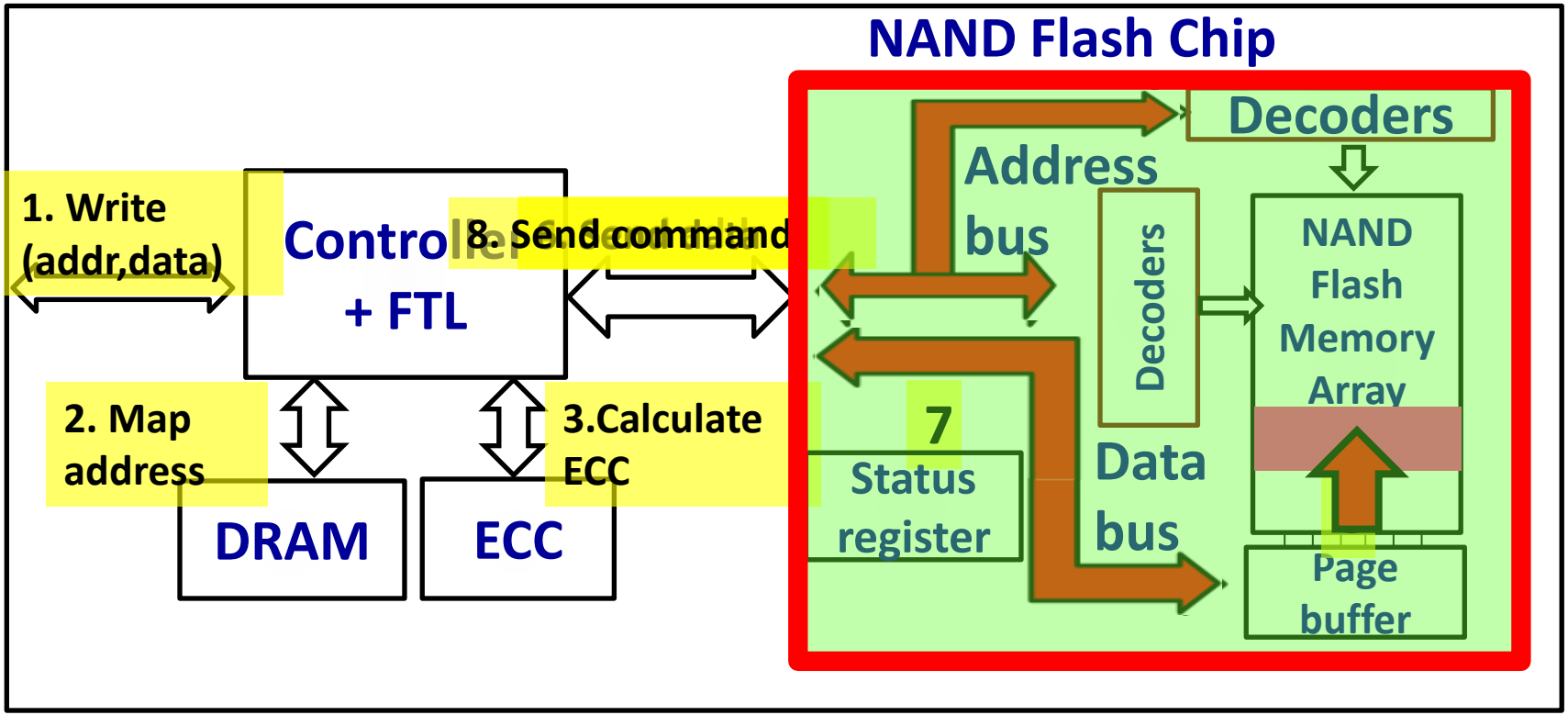
- System Heterogeneity severely impacts power
- No publicly available tools to study power
  - Design space exploration
  - Study impact of various parameters
- FlashPower fills this void
  - Analytical Model based on NAND operation
  - Validated with measurements from real chips
  - Highly parameterized to enable design space exploration

SD Cards

Embedded storage

SSDs

NAND Flash Memory Operation



FlashPower

- Based of CACTI – Architectural simulator for memory systems
- 6 Major parameter categories, 35 parameters total
  - Micro-architectural: e.g. Capacity, #blocks, #pages, etc.
  - Timing: e.g. Read/Write/Erase latency.
  - Device: e.g. Feature size, oxide thickness, coupling ratio, etc.
  - Voltage: e.g. Read/Program/Erase voltage, etc.
  - Workload: e.g. Distribution of 1s and 0s in data.
  - Policy: e.g. #verify cycles, write/erase optimization flags, etc.

MLC Flash Chip Configuration

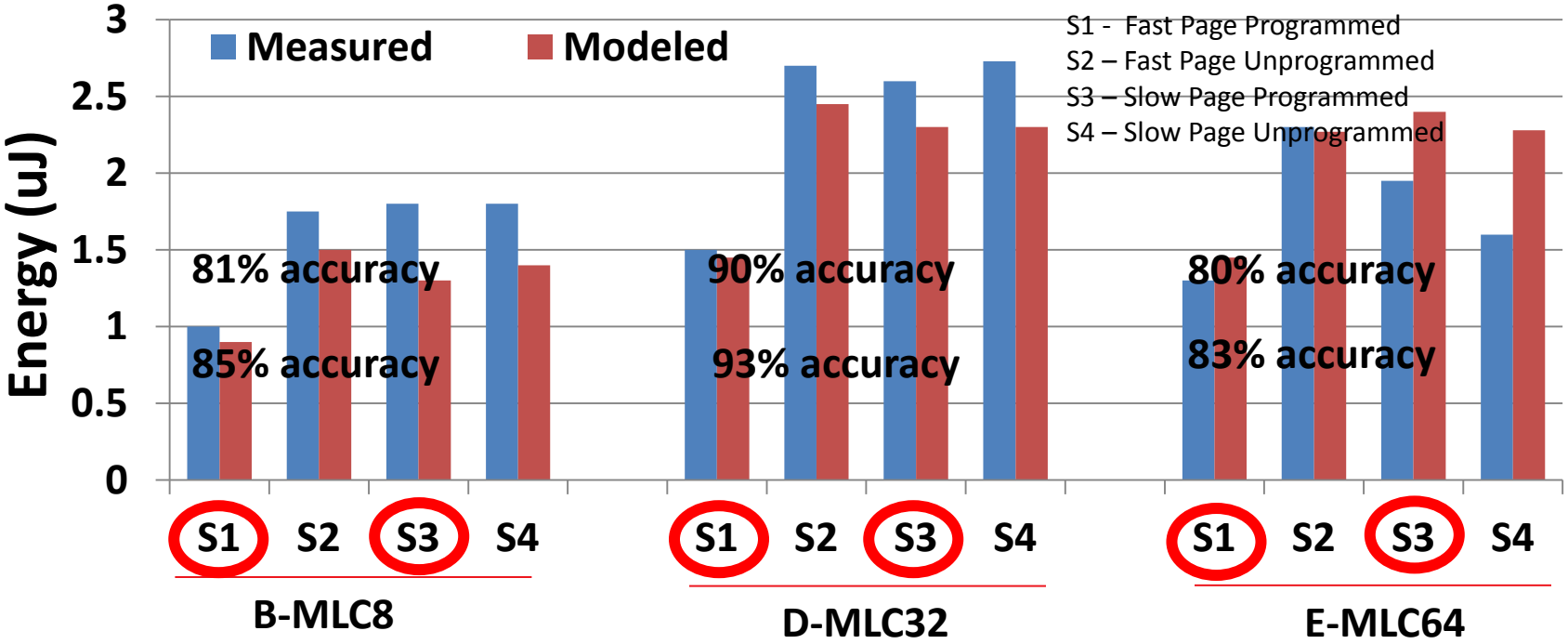
- 3 Chips from different manufacturers

Chip Name	Capacity (Gb)	Page size (KB)	Pages/ Block	Blocks/ Plane	Planes/ Die	Dies	F (nm)
B-MLC8	8.25	2	128	2048	2	1	72
D-MLC32	33.77	4	128	2096	2	2	80*
E-MLC64	66	4	128	2048	4	2	51

Experimental measurement provided by NVSL team in UC SanDiego.

*Estimated feature size

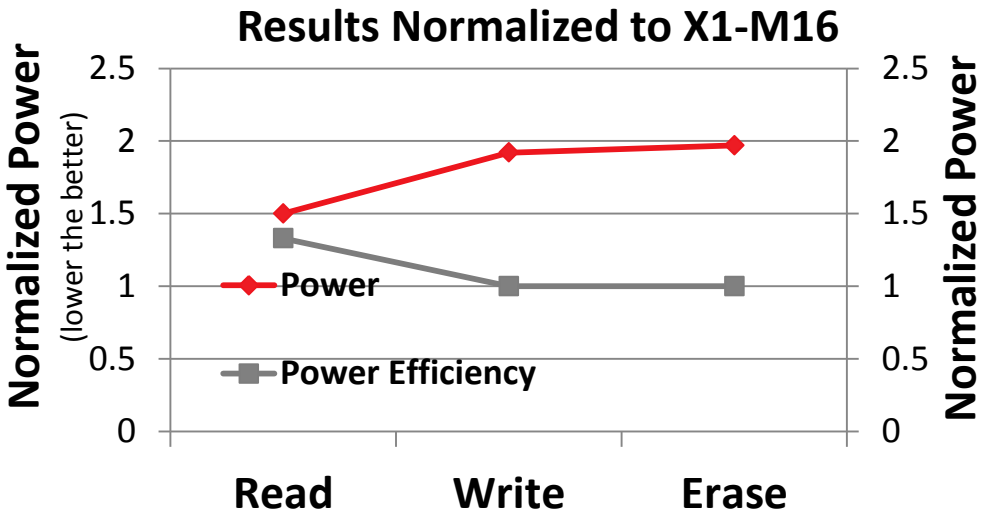
Modeling Results for MLC Flash: Read Operation



Overall accuracy > 80%, Accuracy of reading programmed pages (common case) > 87%
Accuracy >90% with parameter fine tuning.

Design Space Exploration Using FlashPower

Chip Name	Page Size (KB)	Pages / Block	Blocks / Plane	Planes/ Die	Dies/ Chip	Optimal for:
X1 – M16	2	128	2048	2	2	Random IO
X2 – M16	4	128	2048	2	1	Sequential IO



Observations

- X1-M16 lower power than X2-M16
  - Smaller page size
- X2-M16 has higher power efficiency for reads alone

Summary
Published at DATE'2010 and TCAD'2013

- Developed FlashPower, an analytical Model based on NAND read, program and erase operation
- Validated FlashPower, with measurements from real chips
- Highly parameterized
- Enables NAND flash chip architects to study tradeoffs between various array configurations
- Used by universities (SNU, RPI, Penn State), and companies (Micron)

Dissertation Statement

This dissertation addresses the power, reliability and scalability challenges of NAND flash based storage systems by building an set of tools to **model the metrics, exploring the inter-relationship between metrics** and **evaluating the design space to build optimal** NAND flash based storage systems.

Other Work

- STT-RAM memory evaluation (at UVa)
  - HPCA 2011: Architecting processor caches with STT-RAM with relaxed retention
  - ISPLED 2011: Thermal noise model for simulation statistical variations in Magnetic Tunnel Junctions (MTJs)
- Gigascale cache design with STT-RAM (Internship at Rambus Labs)
  - Propose and evaluate a new architecture for designing gigabyte scale STT-RAM last level caches
  - Virtual Cache (VCache) Architecture: Decouple tag and data based cache design by introducing a new address space to manage VCaches
- SSD reliability evaluation (Internship at Google)
  - Built statistical models to evaluate SSD failure mechanisms and predict lifetime of SSDs deployed in data centers while running real world workload

Publications

1. **Vidyabhushan Mohan**, Sudhanva Gurumurthi and Mircea R. Stan. FlashPower: A Detailed Power Model for NAND Flash Memory. Design, Automation and Test in Europe (DATE), Dresden, Germany. March 2010.
2. **Vidyabhushan Mohan**, Taniya Siddiqua, Sudhanva Gurumurthi and Mircea R. Stan. How I Learned to Stop Worrying and Love Flash Endurance. 2nd Workshop on Hot Topics in Storage and File Systems (HotStorage), Co-located with USENIX Annual Technical Conference, Boston, MA. June 2010.
3. **Vidyabhushan Mohan**, Sriram Sankar and Sudhanva Gurumurthi. reFresh SSDs: Enabling High Endurance, Low Cost Flash in Datacenters. University of Virginia, Technical Report, CS-2012-05. May 2012 and Flash Memory Summit, August 2012.
4. **Vidyabhushan Mohan**, Trevor Bunker, Laura Grupp, Sudhanva Gurumurthi, Mircea R. Stan and Steven Swanson. Modeling Power Consumption of NAND Flash Memories Using FlashPower. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), Issue 7, July 2013.

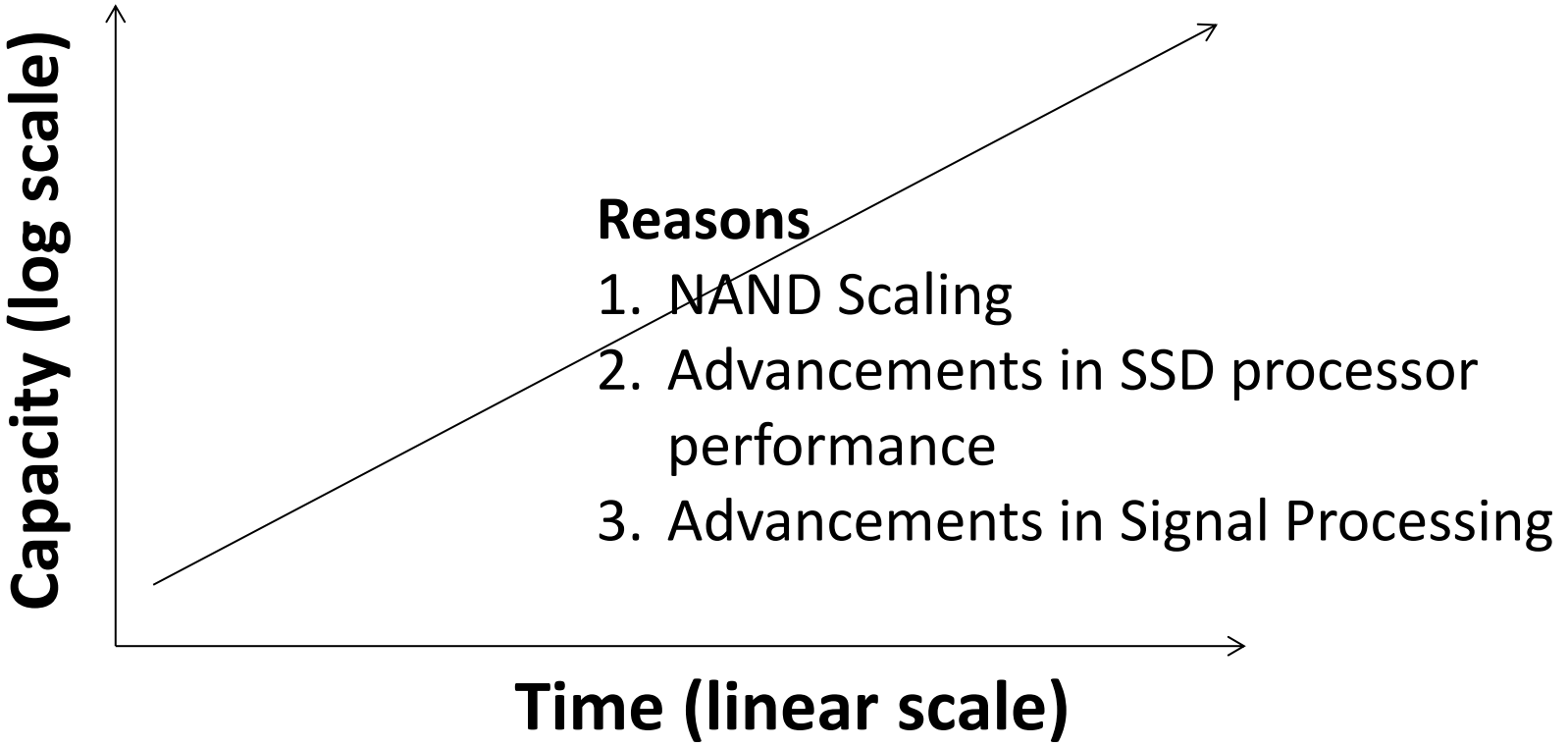
Publications

5. Clint Smullen, **Vidyabhushan Mohan**, Anurag Nigam, Sudhanva Gurumurthi and Mircea R. Stan. Relaxing Non-Volatility for Fast and Energy-Efficient STT-RAM Caches. The 17th IEEE Symposium on High Performance Computer Architecture (HPCA-17), February 2011.
6. Anurag Nigam, Clint Smullen, **Vidyabhushan Mohan**, Eugene Chen, Sudhanva Gurumurthi and Mircea R. Stan. Delivering on the Promise of Universal Memory for Spin-Transfer Torque RAM (STT-RAM). International Symposium on Low Power Electronics and Design (ISLPED). Fukuoka, Japan. August 2011.

Questions

Backup for Scalability Section

Growth in Storage System capacity



Growth in Storage System capacity

Captive flash supply **AND** in-house controller expertise are prerequisites to build cost efficient and high performance SSDs

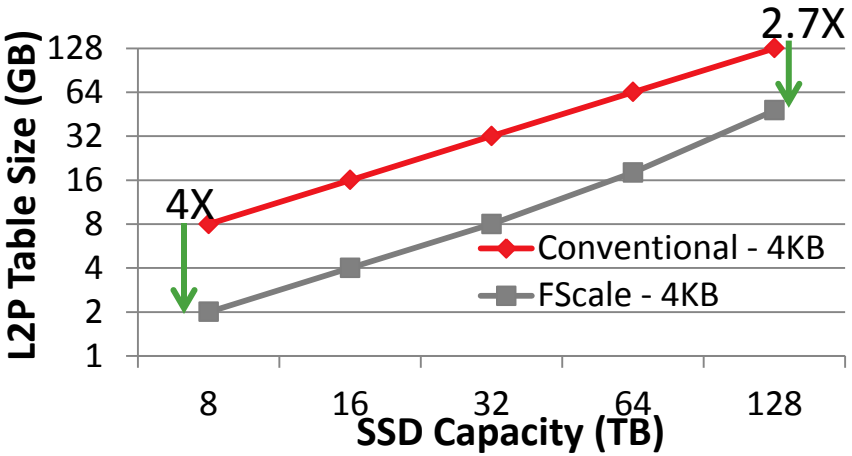
FScale Architecture

- Rationale
  - Capacity scales by adding more storage – i.e. NAND chips
  - Add more processing capability for each NAND package
    - Package: a group of NAND chips
    - Manage cost - since we build a special package only for applications that require high scalability

Why performance doesn't scale with capacity?

- As capacity increases,
  - L2P table size increases which results in higher cache miss rate
  - FTL processor cycles spent on non-host operations increases
    - To service L2P table misses, Garbage Collection (GC) and Wear Leveling (WL)
- Drawbacks with existing solutions
  - Reduce L2P Table size by increasing the logical page size
    - Increases write amplification (WA), higher stress on NAND memory
  - Increase processor frequency with capacity
    - Increases cost (non-NAND)
    - Delays time-to-market (design and development time with each new processor)

FScale Architecture: Impact on L2P table size



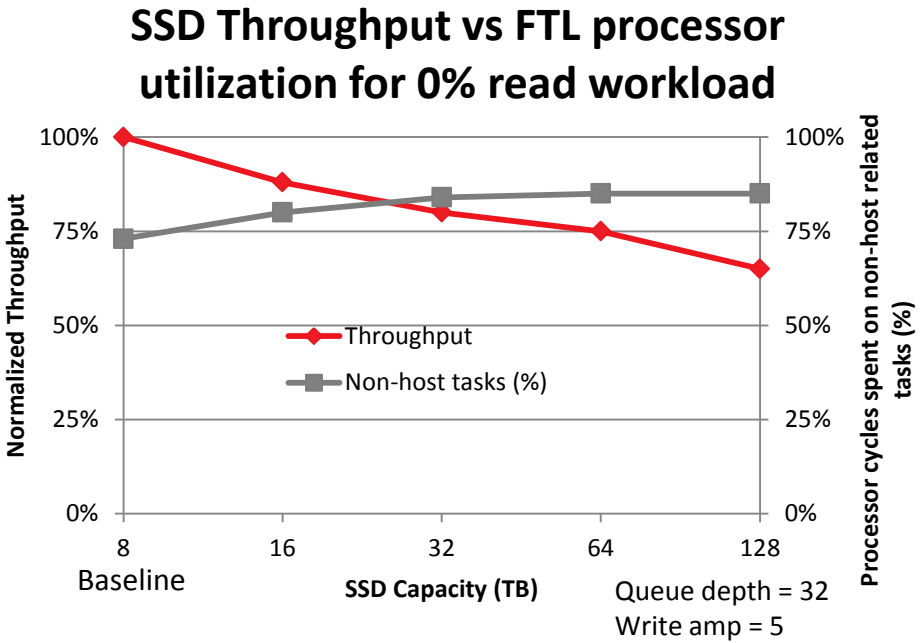
Cache Latency Metric	Conventional	FScale		
		Min	Typ	Max
Hit Latency #DRAM, NAND access	1,0	1,0	1,3	1,7
Miss latency #DRAM, NAND access	2,1	2,1	2,4	2,8

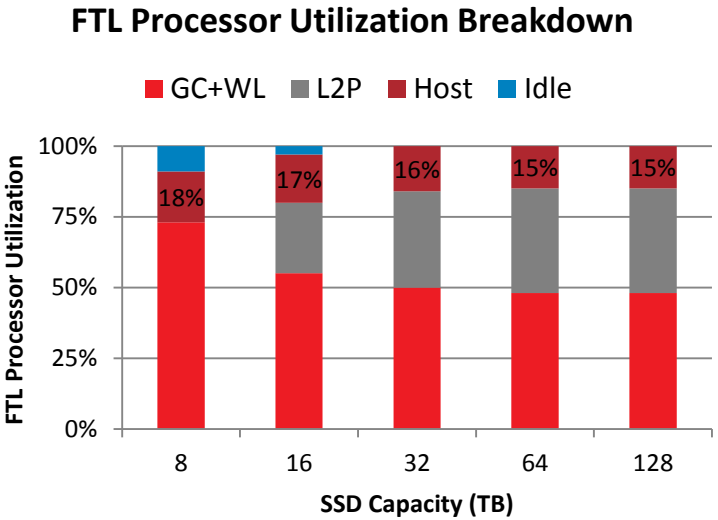
Observations

1. 4X to 2.7X reduction in L2P table size withFScale architecture -i.e. For a fixed cache size, equivalent improvement in hit rate
   - ❑ Only the L2P Table Directory is cached

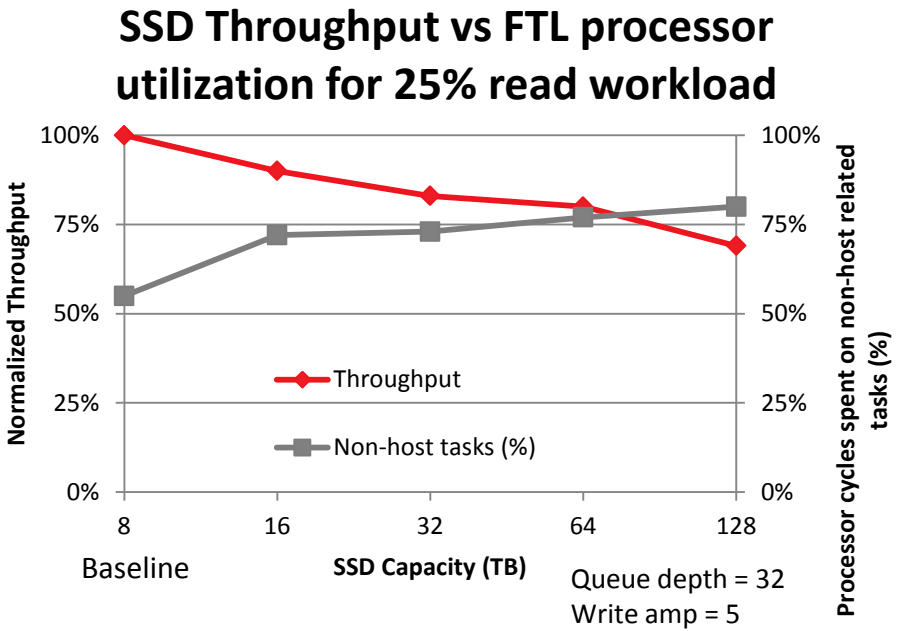
2. However, cache latencies (hit and miss) are higher for FScale architecture
   - ❑ Traversing the P2L map table requires several NAND accesses
   - ❑ Btree and sorted P2L table fixes the latency bounds

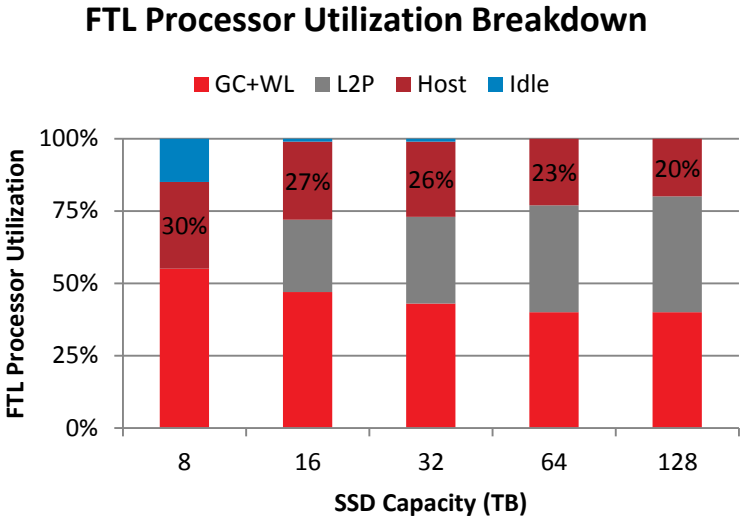
0% Read Workload performance in conventional SSDs



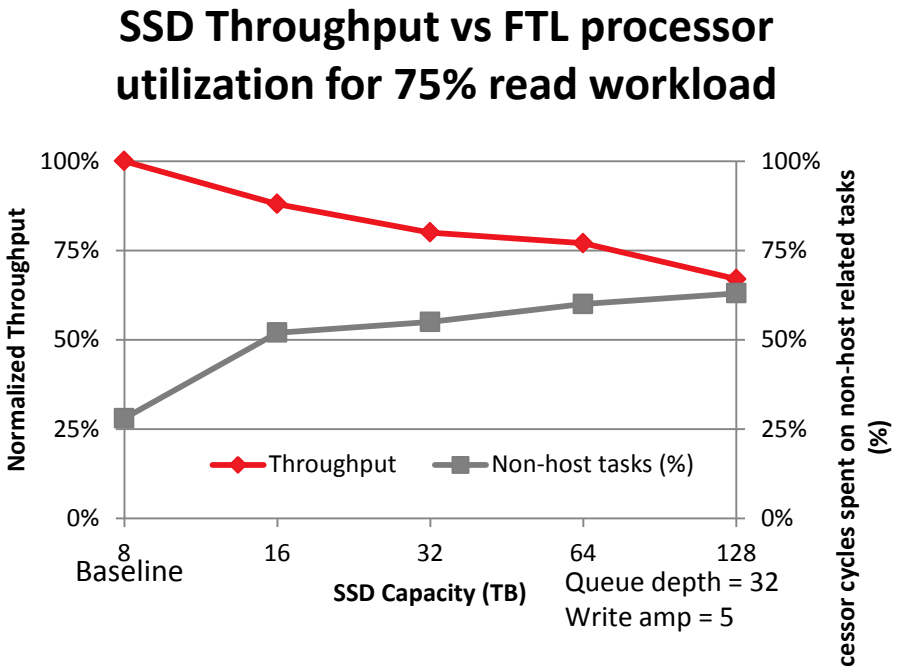


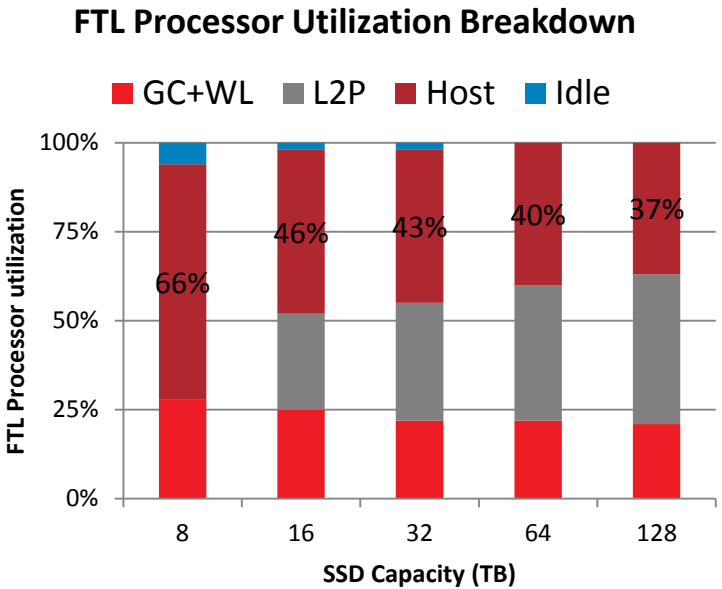
25% Read Workload performance in conventional SSDs



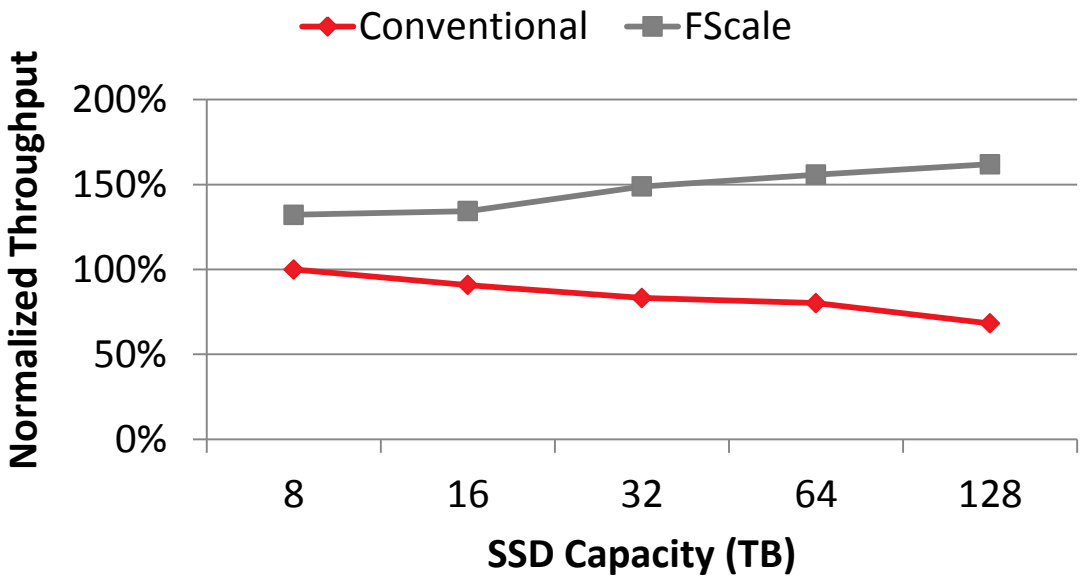


75% Read Workload performance in conventional SSDs

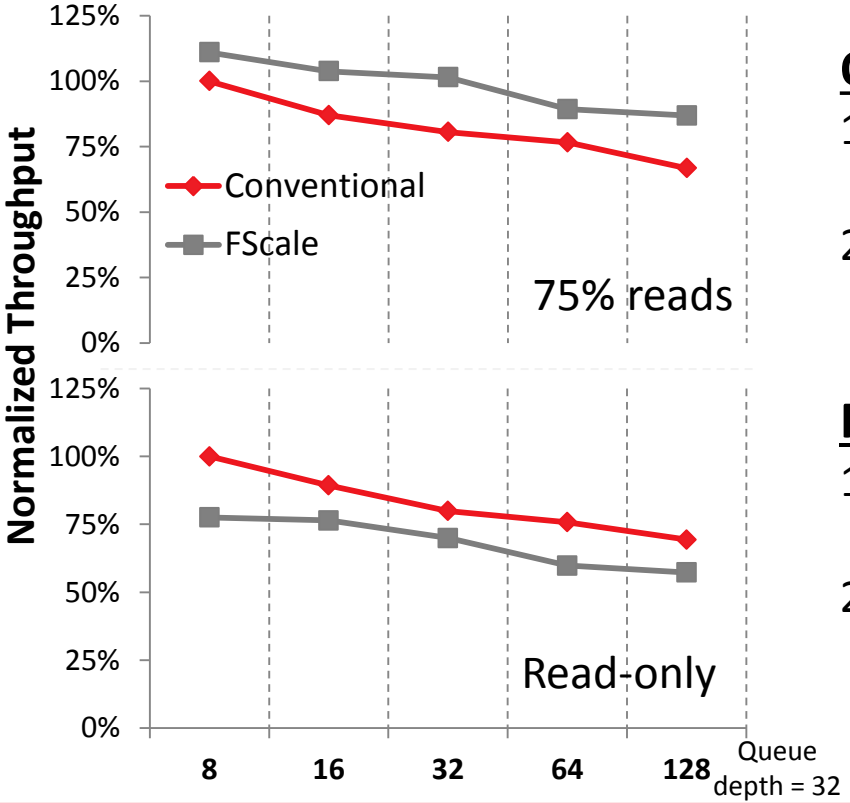




50% write workload – Performance with FScale



FScale Architecture Performance Evaluation
Read-intensive Workloads



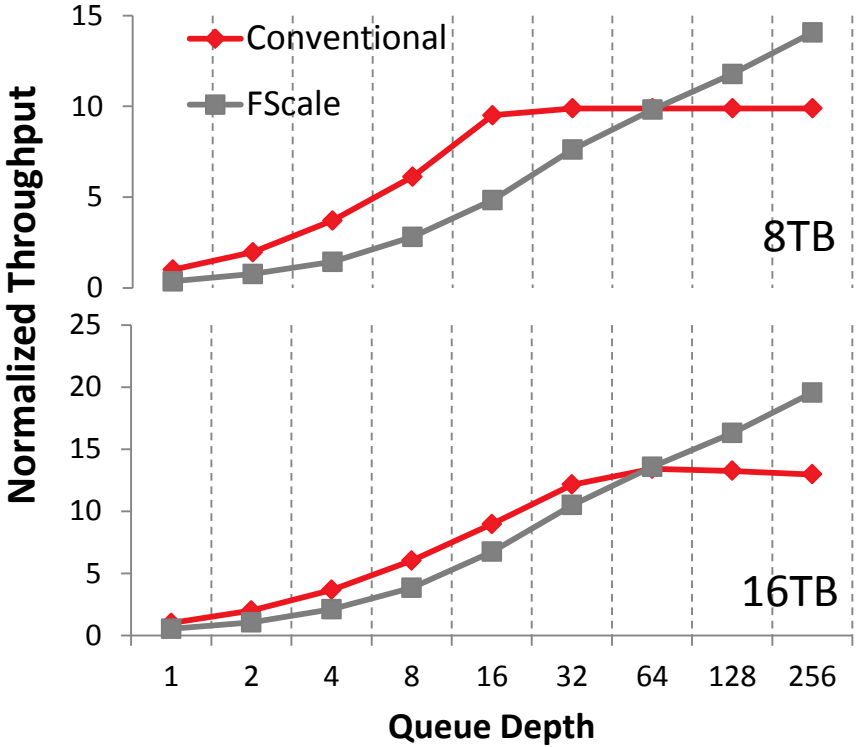
Observations

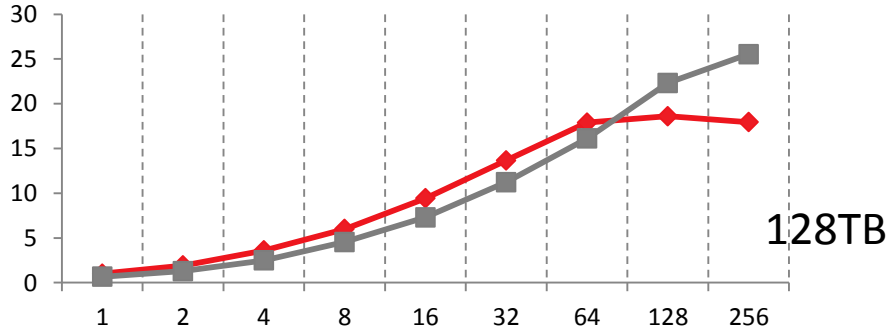
1. FScale performance does not scale for read-intensive workloads
2. For read-only workload, FScale performance lower than conventional SSDs

Reason:

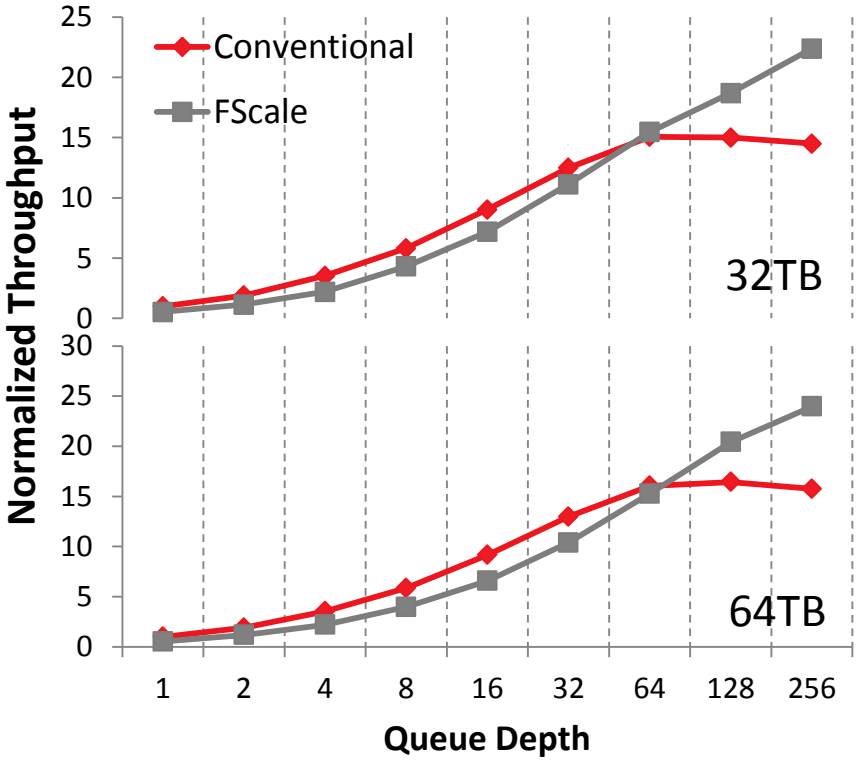
1. Effectiveness of LC decreases due to low GC and WL
2. Performance affected by latency penalty due to L2P table management at local controller (LC)

FScale Architecture Read-only Workload Performance at Various Queue Depths for capacities 8TB, 16TB and 128TB





FScale Architecture Read-only Workload Performance at Various Queue Depths



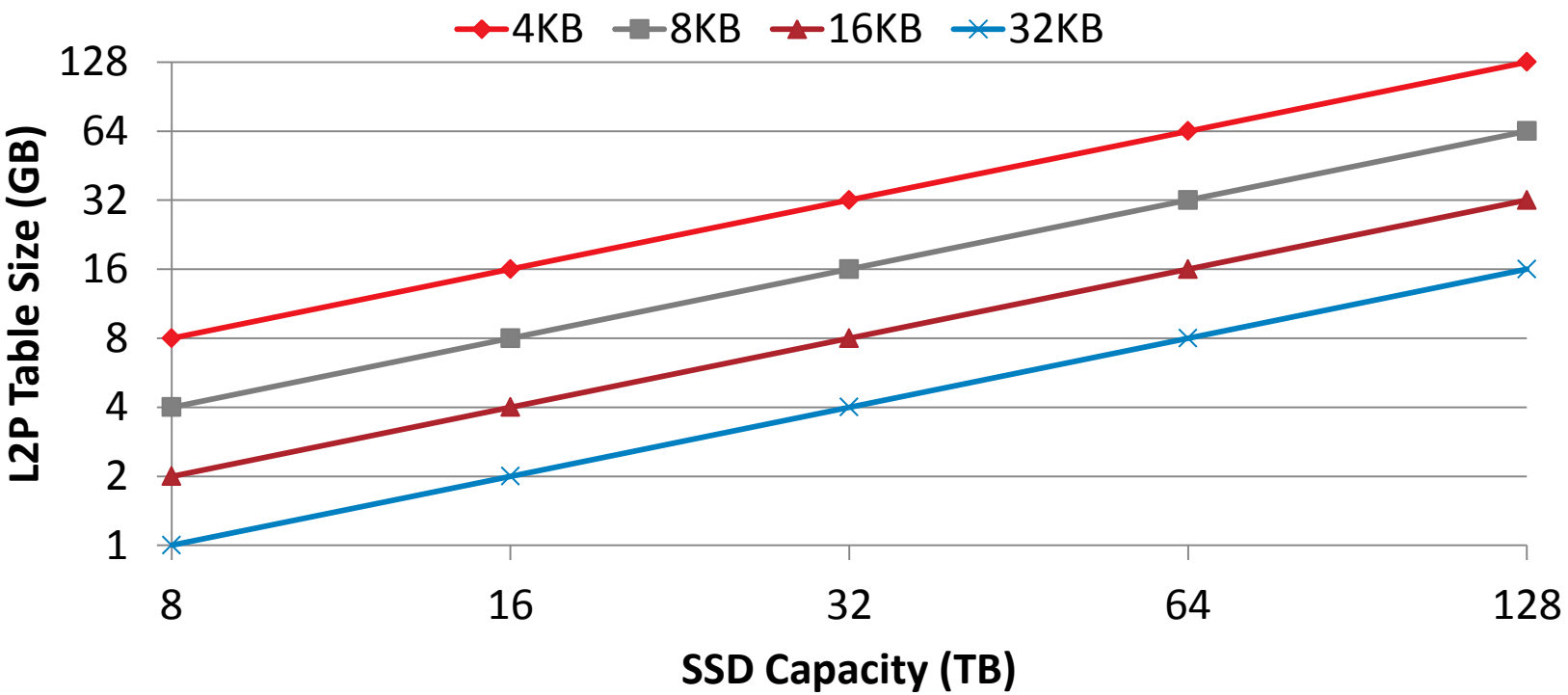
Observations

As queue depth increases, the performance of read-only workload increases and exceeds conventional SSD architecture performance

Reason

When the number of pending host I/O request increases, we can effectively hide the latencies of L2P table management at the local controller (LC)

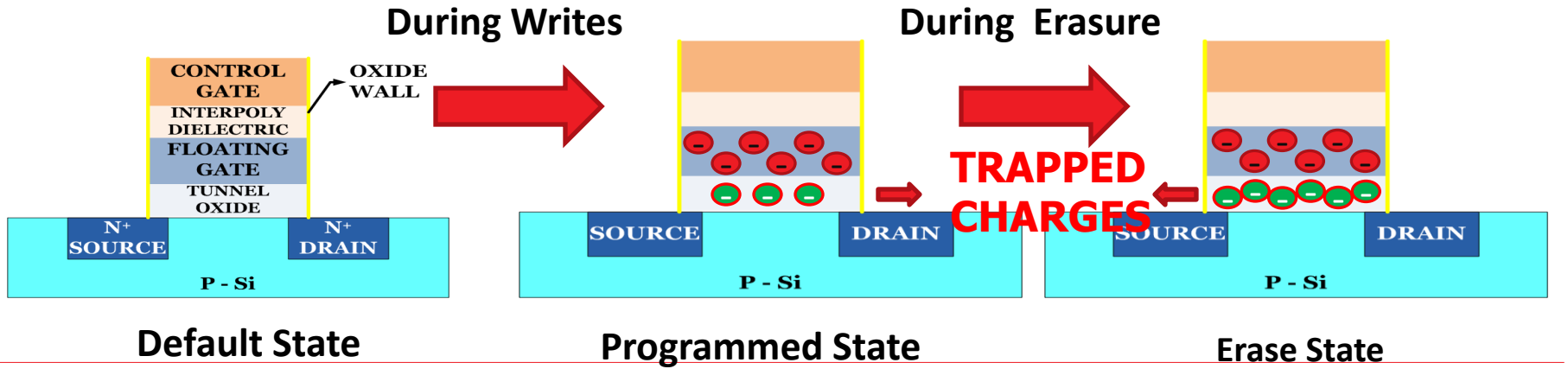
SSD Capacity vs L2P table size for conventional architectures



Backup for reliability section

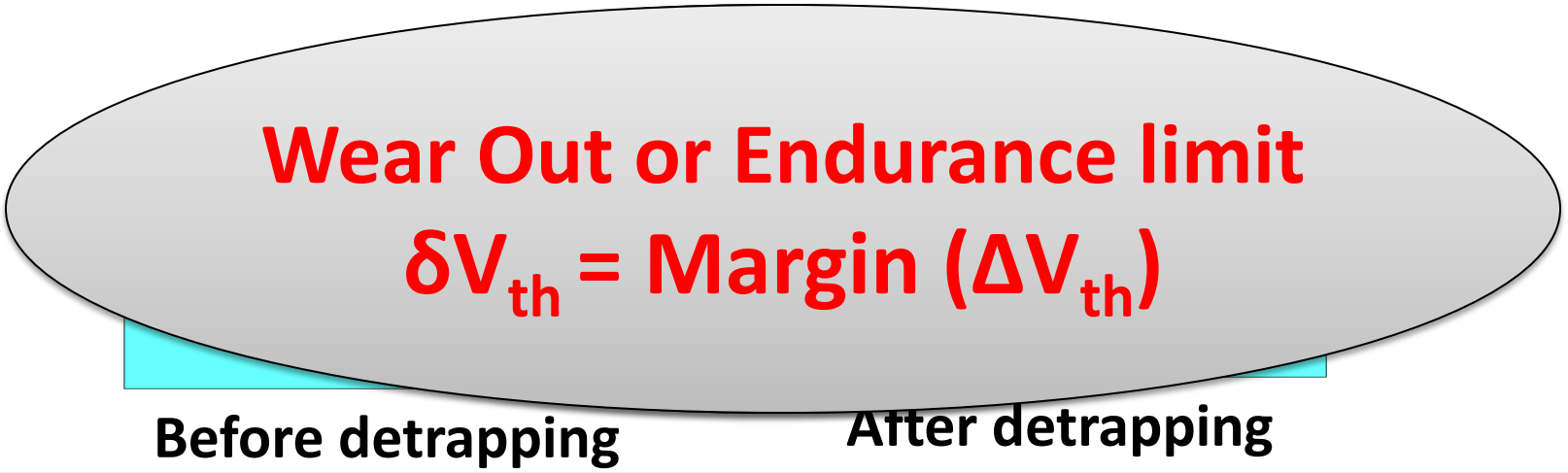
Cycling

- Writes and Erase – stress events
- Side effect of cycling
  - Trapped charges
  - Increase in threshold voltage ($\Delta V_{th,s}$)

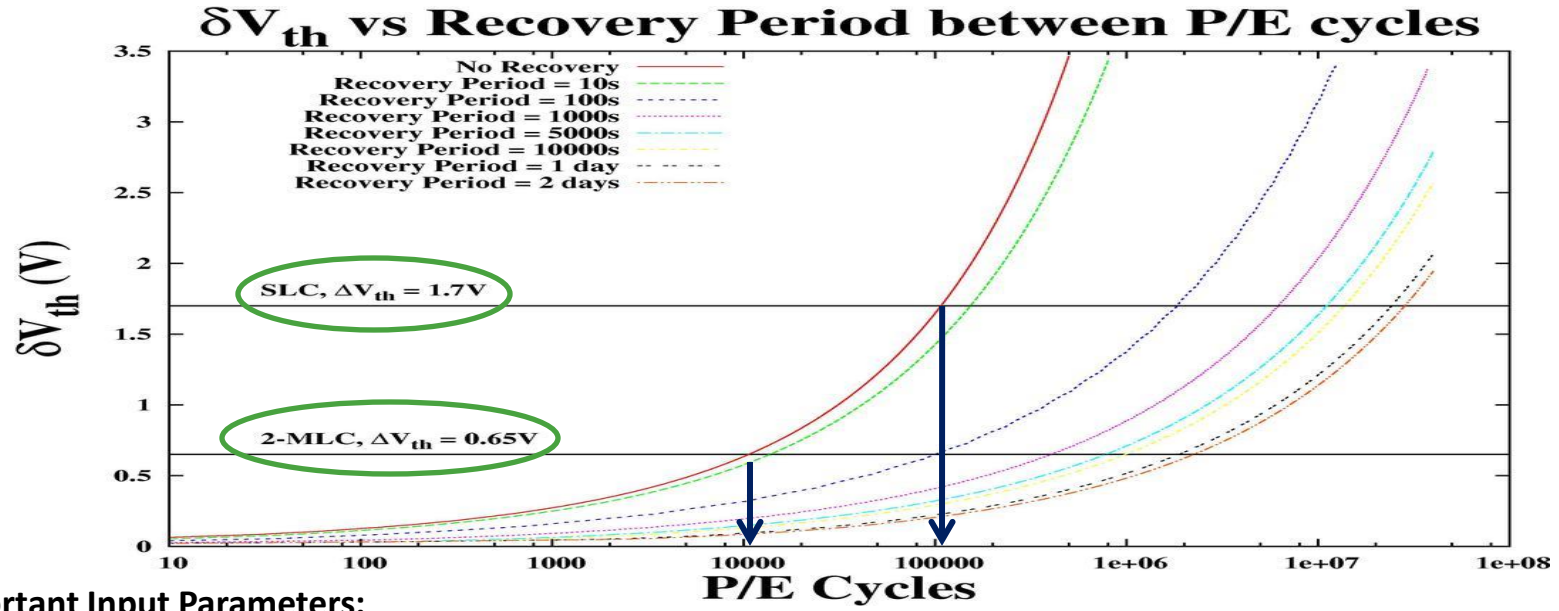


Cycling - Recovery Model

- Quiescent period between stresses
- Some charges get **detrapped**
- Reduces the threshold voltage ($\Delta V_{th,r}$)



Impact of Recovery Period on Cycling

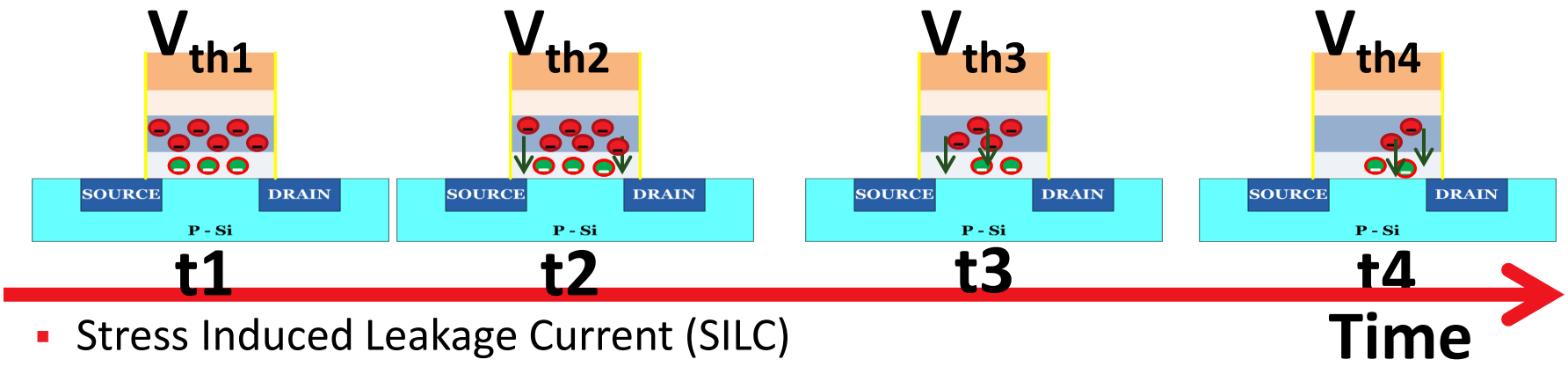


Important Input Parameters:

F =80nm, Temperature = 30C

Values for model parameters derived from empirical data collected by experiments at device/circuit level

Data Retention



- Stress Induced Leakage Current (SILC)
  - Charge leakage due to trap assisted tunneling
- $\delta V_{th} = V_{th1} - V_{th4}$
  - δV_{th} == Margin => Data retention failure
- Data Retention Time ($t_{retention}$)= t4 –t1
- Exacerbated by temperature

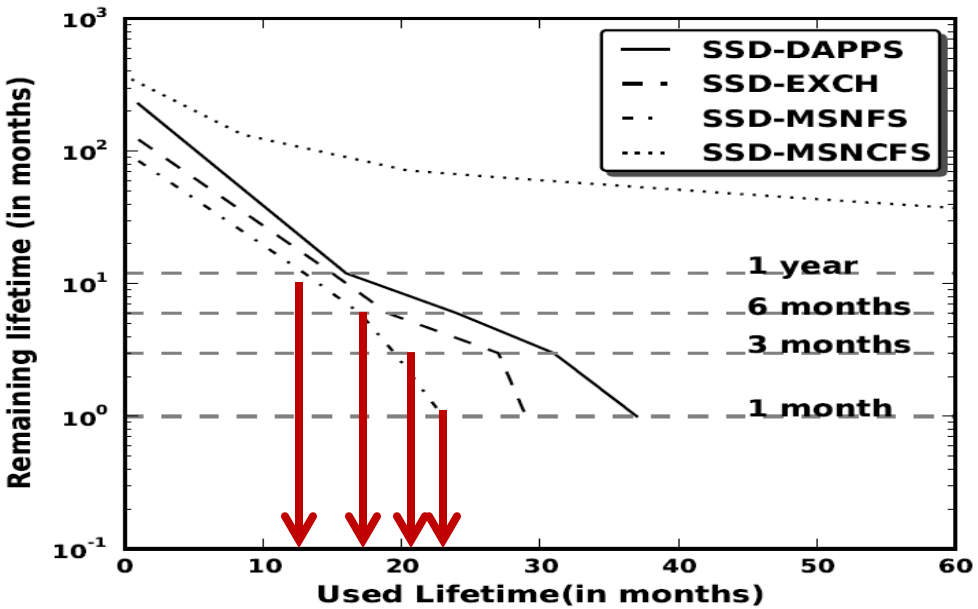
Workload Traces

Workload [3]	Total I/Os (millions)	Read/Write Ratio
Display Ads Platform Payload Server (SSD-DAPPS)	10.9	1:1.2
Exchange Server (SSD-EXCH)	22	1:2.2
MSN File Server (SSD-MSNFS)	15.54	1:1.2
MSN Metadata Server (SSD-MSNCFS)	7.8	1:0.64

SSD traces extrapolated from HDD I/O traces of enterprise workloads

[3] HDD Traces from IOTTA Trace Repository from SNIA - http://iotta.snia.org/

Baseline: How long do Enterprise SSDs last?

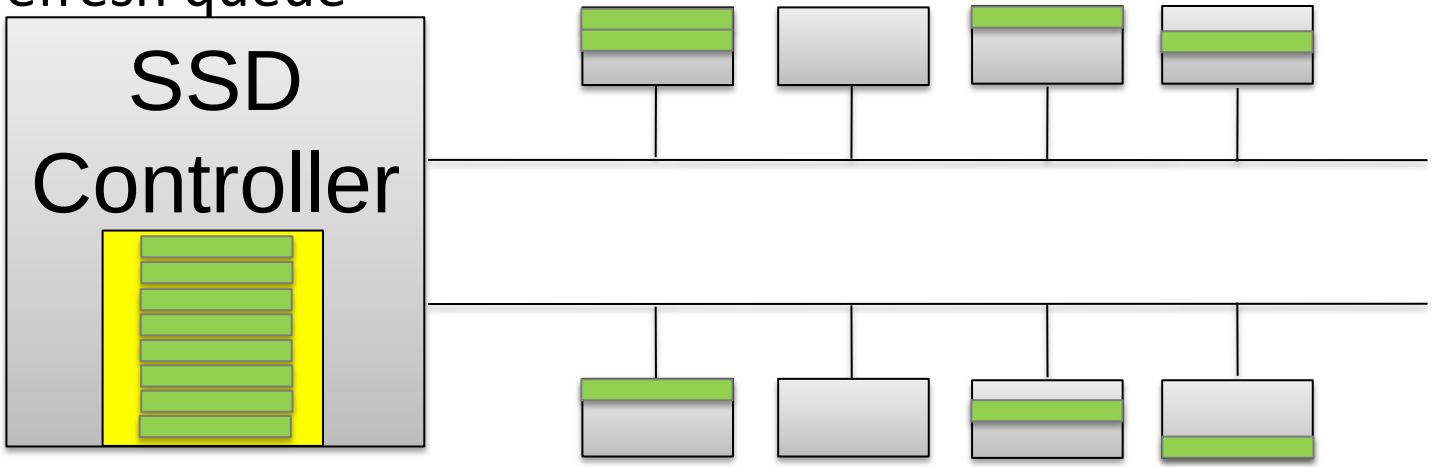


- Using FENCE along with Disksim
- 64GB, 2-bit MLC SSD
- F= 80nmm
- Temperature = 30C
- 5 year service life
- Varying data retention requirements
- Enterprise workloads from Microsoft

☛Even with reduced retention, SSDs do not last for their service life

reFresh SSDs: Operation

Refresh operation invoked at regular intervals on blocks in the refresh queue



Unlike wear leveling, refresh operations are triggered to handle a immediate deadline (PBRP < VRP)

FENCE – Stress and Recovery Model

- Increase in threshold voltage due to stress -$\Delta V_{th,s} = C_1 * cycle^{0.62} + C_2 * cycle^{0.30}$

- Decrease in threshold voltage due to recovery - $\Delta V_{th,r} = C_3 * \ln(\Delta V_{th,s}) * \ln(t)$

- Effective increase in threshold voltage - $\delta V_{th} = \Delta V_{th,s} - \Delta V_{th,r}$

FENCE – SILC Model

Larcher et al. SILC effects on *E*2PROM memory cell reliability

- J_{SILC} = J tr(t) + Jss
- Jss >> J tr(t)
- JSILC = A_{SILC} x F^2_{ox} x exp(-B_{SILC}/F_{ox})
- ASILC = C x J^b_{str} x exp (-D/Q^a_{inj})

(A,B,C,D, a,b are constants)

Another distinct trend that can be observed from the above figures is that every line has 4 distinct slopes depending on the amount of cycling the cell has experienced. This behavior is in accordance with studies by [?]. According to [15], for P/E cycles less than 10^3, the drop is threshold voltage is dominated by A_{SILC} more than the cycle count, while for P/E cycles greater than 10^5, the threshold voltage saturates because the oxide field across the tunnel oxide decreases and the leakage current also decreases. Between 10^3 and 10^4 cycles, a combination of these two effects results in a different slope. Because of the difference in the slope of the curve, the rate of change in retention period also varies significantly.

Moazzami et al observed that for thinner tunnel oxide ($< 13nm$), the steady state component dominates the transient component [21]. As [5, 14, 28] show, this steady state component predominantly originates from a trap-assisted tunneling mechanism, where presence of interface and bulk traps increase the leakage current. Hence, to model J_{SILC}, it is sufficient to model the steady state component (J_{ss}). So, Equation (12) can be modified to

$$J_{SILC} = J_{ss} \tag{13}$$

Because the tunnel oxide thickness is smaller than 13nm for many generations of NAND flash [11], Equation (13) provides a good estimate of SILC. According to Larcher et al, the steady state component is modeled by using a Fowler-Nordheim (FN) like expression, as given below [15].

$$J_{SILC} = J_{ss} = A_{SILC} \cdot F_{OX}^2 \cdot \exp\left(-\frac{B_{SILC}}{F_{OX}}\right) \tag{14}$$

$$A_{SILC} = C \cdot J_{STR}^{\beta} \cdot \exp\left(-D \cdot Q_{INJ}^{\alpha}\right) \tag{15}$$

The barrier height used to calculate the exponential factor B_{SILC} is between $0.8 - 1.1eV$. F_{OX} is the electric field across the tunnel oxide during stress events and is considered to be $3.8MV/cm$. The values for constants C, β, D, α are available in [15]. The term J_{STR} represents the current density across the tunnel oxide during stress events and is a function of applied program or erase voltage. We assume an operating voltage of 16V for program and erase operations based on [11]. Q_{INJ} is the total amount of charge exchanged across the tunnel oxide and is a function of P/E cycles. Incorporating the cycling term in Q_{INJ} helps to calculate the leakage current as a device is cycled over a period of time. Based on [15], Q_{INJ} can be defined as,

$$Q_{INJ} = \Delta Q_{INJ} \cdot N_C \tag{16}$$

$$\Delta Q_{INJ} = \Delta V_{th} \cdot C_{CG} \tag{17}$$

where ΔV_{th} is the threshold voltage difference between programmed and erased state and C_{CG} is the pacitance between the control gate and floating gate of a FGT.

Equation (14) represents the SILC due to the presence of trapped charges in the tunnel oxide. Assuming t $Q_{th,spread}$ to be the total charge stored in the floating gate corresponding to a logical bit, δV_{th} to be total arge trapped in the tunnel oxide (calculated from Equation (11)), and J_{SILC} to be the leakage current, we n calculate the time taken for the charges to leak from the floating gate ($t_{retention}$) to be,

$$t_{retention} = \frac{(Q_{th,spread} - \delta V_{th})}{J_{SILC}} \tag{18}$$

References for FENCE

1. Mielke et al. Recovery Effects in the Distributed Cycling of Flash Memories
2. Mielke et al. Bit error rate in nand flash memories
3. Yamada et al. A novel analysis method of threshold voltage shift due to detrap in a multi-level flash memory
4. Yang et al. Reliability issues and models of sub-90nm NAND flash memory cells.
5. de Blauwe et al. SILC-related effects in flash *E*2PROM's-Part I and II: A quantitative model for steady-state SILC.
6. Larcher et al. SILC effects on *E*2PROM memory cell reliability

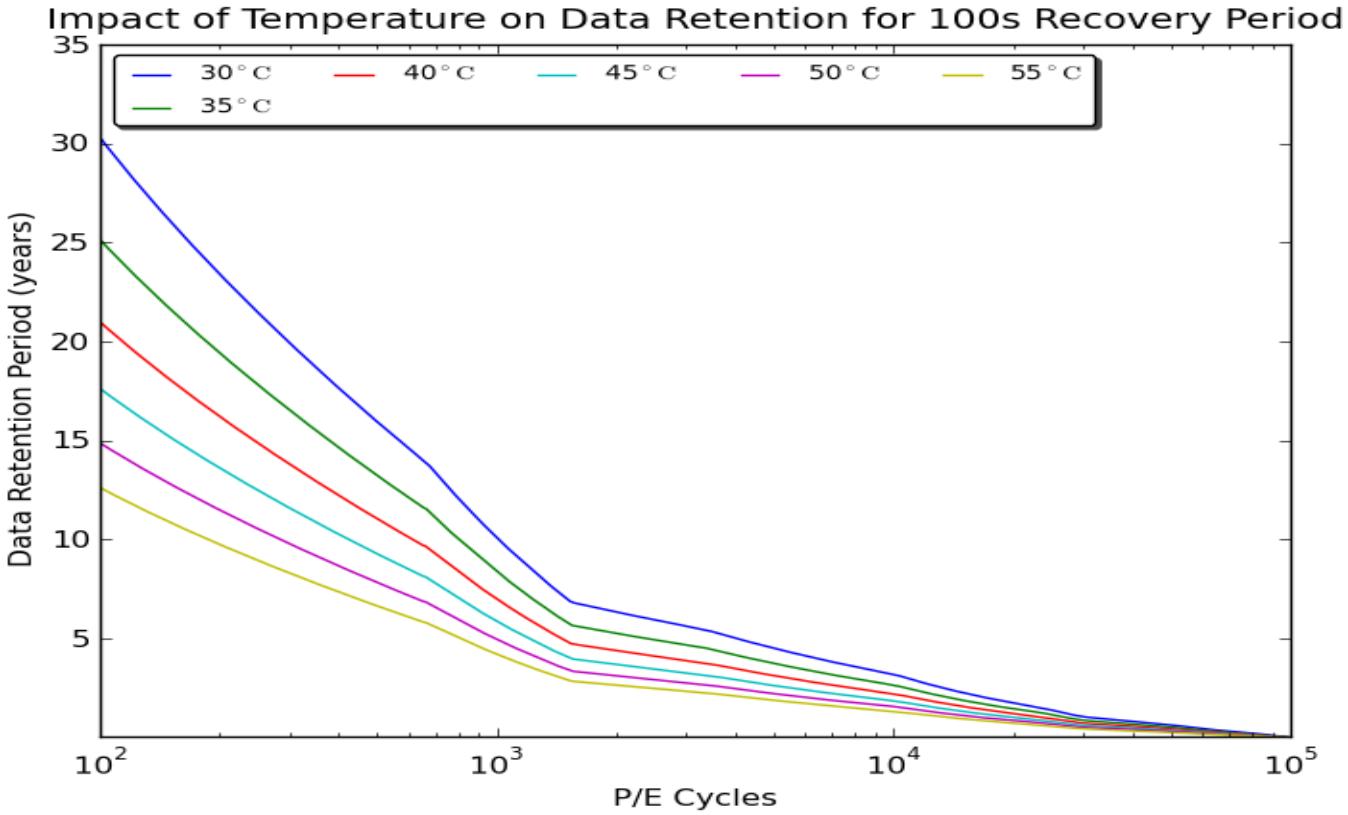
Estimating SSD Lifetime

- Reliability – long term effect
- Traces – Disk activity over 1 day and are from HDD based systems
  - Extrapolate HDD trace into SSD trace
  - Extrapolate 1 day behavior to disk service life.
- Capture temporal and spatial lifetime of SSD
  - Block B_i at time T_i has remaining lifetime L_i after P_i cycles

Cross Validation for SSD workloads

Workload	Mean			Std deviation		
	M	E	D	M	E	D
DAPPS ($i = 1$)	6.51	6.52	0.01	0.34	0.27	0.07
DAPPS ($i = 2$)	4.49	4.52	0.03	0.33	0.23	0.10
Exchange ($i = 1$)	2.91	3.37	0.46	0.33	0.80	0.47
	3.78	5.37	1.59	0.19	2.98	2.79
Exchange ($i = 2$)	1.07	1.31	0.24	0.15	0.63	0.48
	1.73	3.54	1.81	0.13	1.93	1.8
MSNFS ($i = 1$)	4.39	4.45	0.06	0.33	0.24	0.09
MSNFS ($i = 2$)	2.31	2.33	0.02	0.31	0.23	0.08

Impact of Temperature on SSD reliability



Future Research Directions

1. Hardware architectures
2. Software System architecture
3. Emerging memories: Materials and Circuit Design

Future Research Directions – Hardware architectures

- In-Storage Compute
  - How to effectively use the processing power of a storage device?
  - Increase performance and energy efficiency by reducing data movement across storage interfaces
- Software Defined SSDs
  - How much FTL is too much?
  - Software configurability of SSDs
  - Optimize data layout and increase performance and reliability

Future Research Directions - Software Architectures

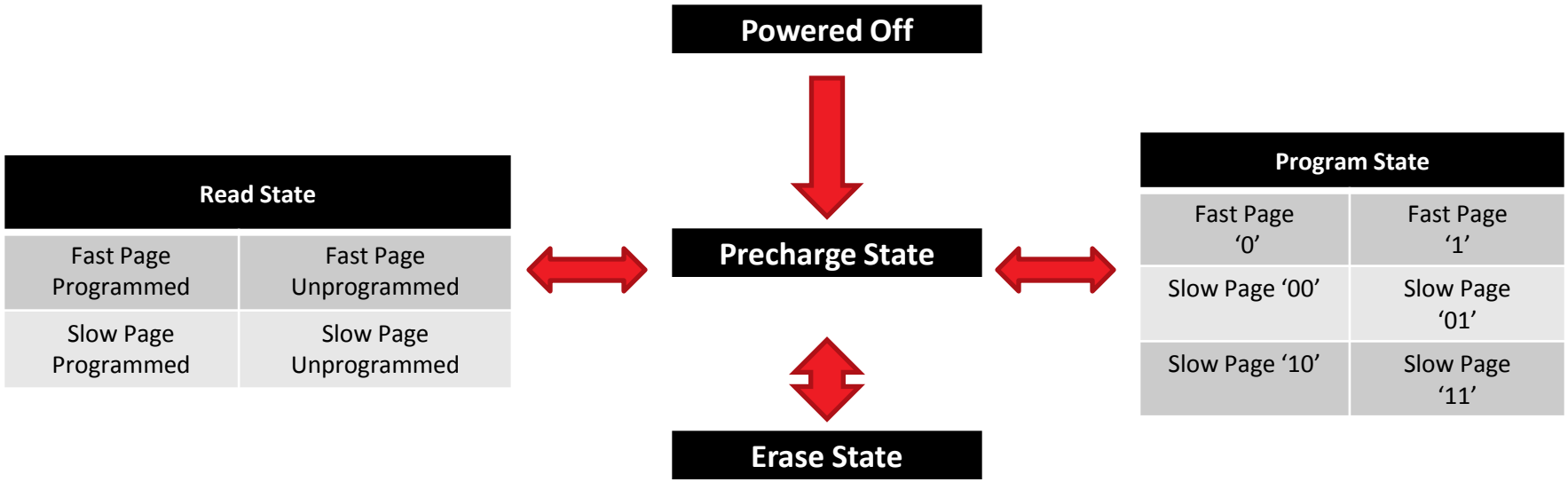
- Exploiting in-storage compute
  - Rewrite applications to take advantage of in storage compute
  - Example: Send a search request (map-phase) instead of read/write I/O requests
  - Process result from the storage in the reduce phase
- With new storage class memories (SCMs), the software latency is no longer negligible
  - Optimize software based on SCM technology

Future Research Directions - Materials and Circuit Design

- Storage Class Memories: Bridge gap between memory and storage in the memory hierarchy
  - PCRAM (e.g. Intel/Micron's 3D Xpoint)
  - ReRAM (SanDisk/Samsung/etc.)
  - STT-RAM
- 3D NAND: Challenges as we go to hundreds of layers
- Circuit and Package Design: Cost effective methodology to design a chip where features can be easily modified based on application
  - System heterogeneity: Personal electronics, Cars and Data centers
  - Cost reduction and higher chances of success when a memory is used for many applications

Backup results for power section

Power State Machine for MLC Flash



Definitions

1. Multi-Level Cell (MLC) Flash : 2 bits are stored per memory cell
2. Page: Smallest granularity of read/write operation
3. Fast & Slow Page: Page types categorized based on read/write access latency

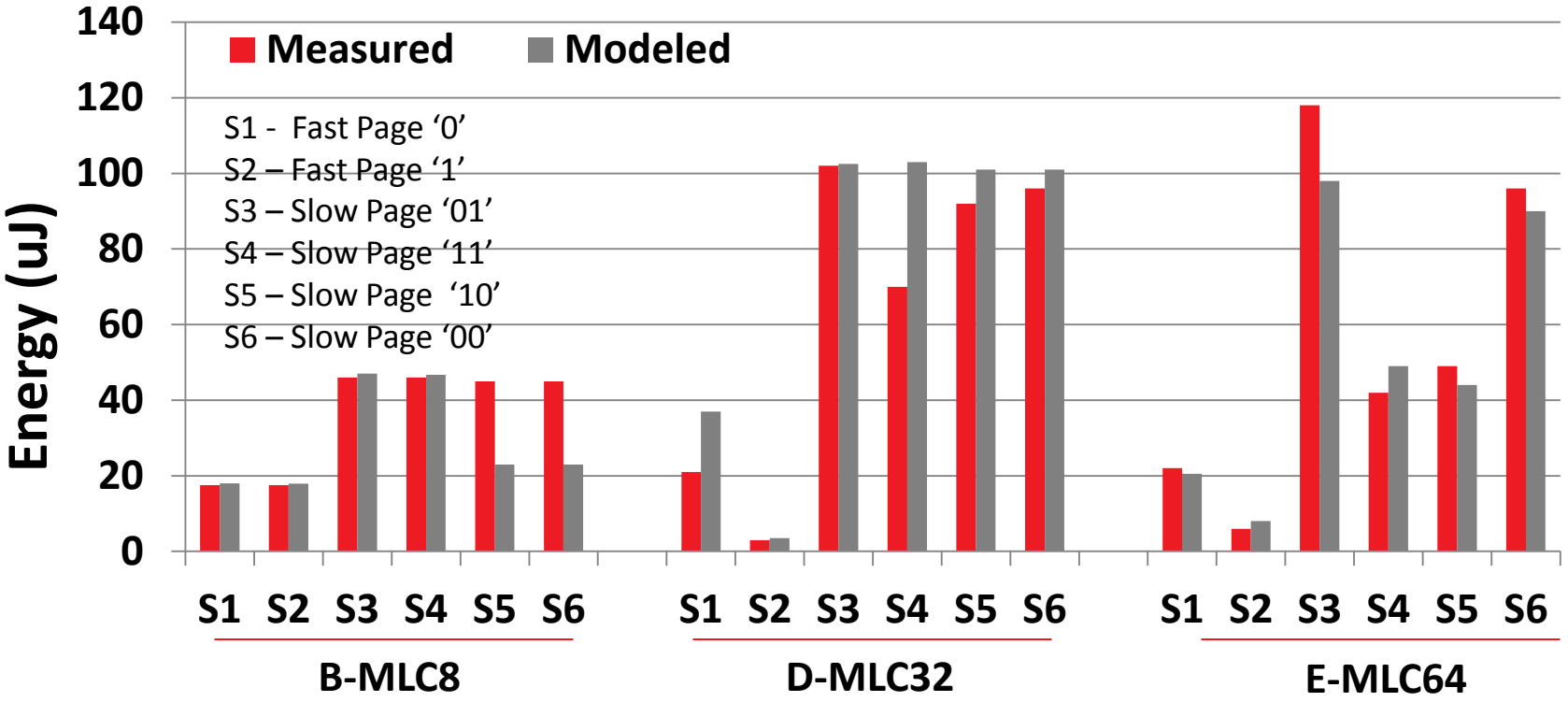
Complete Parameter List for FlashPower

Microarchitectural Parameters	
$N_{pagesize}$ (R)	Size (in bytes) for the data area in each page.
$N_{sparebytes}$ (R)	Size (in bytes) for the spare area in each page.
N_{pages} (R)	Number of pages per block.
N_{brows} (R)	Number of rows of blocks in a plane.
N_{bcols} (O)	Number of columns of blocks in a plane. Defaults to 1.
N_{planes} (O)	Number of planes per die. Defaults to 1.
N_{dies} (O)	Number of dies per chip. Defaults to 1.
$tech$ (R)	Feature size of FGTs.
$Bits\_per\_cell$ (R)	Number of bits per FGT.
Device-level Parameters	
N_A, N_D (O)	Doping level of P-well and N-well. Defaults to $10^{15}cm^{-}3$ and $10^{19}cm^{-}3$.
β (O)	Capacitive coupling between control gate and P-well. Defaults to 0.8.
GCR (O)	Ratio of control gate to total floating gate capacitance. Obtained from ITRS.
t_{ox} (R)	Thickness of tunnel oxide in FGTs.
$\frac{W}{L}$ (O)	Aspect ratio of FGTs. Defaults to 1.

Timing Parameters	
$t_{program}$ (R)	Latency to program a page for SLC flash (fast page in MLC flash).
t_{read} (R)	Latency to read a page in SLC flash (fast page in MLC flash).
t_{erase} (R)	Latency to erase a flash block.
$t_{program,slow}$ (O)	Latency to program a slow page in MLC flash. Defaults to $t_{program}$ * 2.
$t_{read,slow}$ (O)	Latency to read a slow page in MLC flash. Defaults to t_{read} * 2.
Bias Parameters	
V_{dd} (R)	Maximum operating voltage of the chip.
V_{read} (O)	Read voltage for SLC flash (fast page in MLC flash). Defaults to 4.5V.
$V_{read,slow}$ (O)	Read voltage for slow page in MLC flash. Defaults to 2.4V.
$V_{bl,drop\_[0/1]}$ (O)	Bit-line swing for read operation. Defaults to 0.7V.
V_{pgm} (O)	Program voltage for selected page. Obtained from ITRS.
V_{era} (O)	Erase voltage to bias the substrate. Same as V_{pgm}.
$V_{bl,pre}$ (O)	Bit-line precharge voltage. Defaults to 3/5th of V_{dd}.
V_{step} (O)	Step voltage used for program and erase. Defaults to 0.3V.

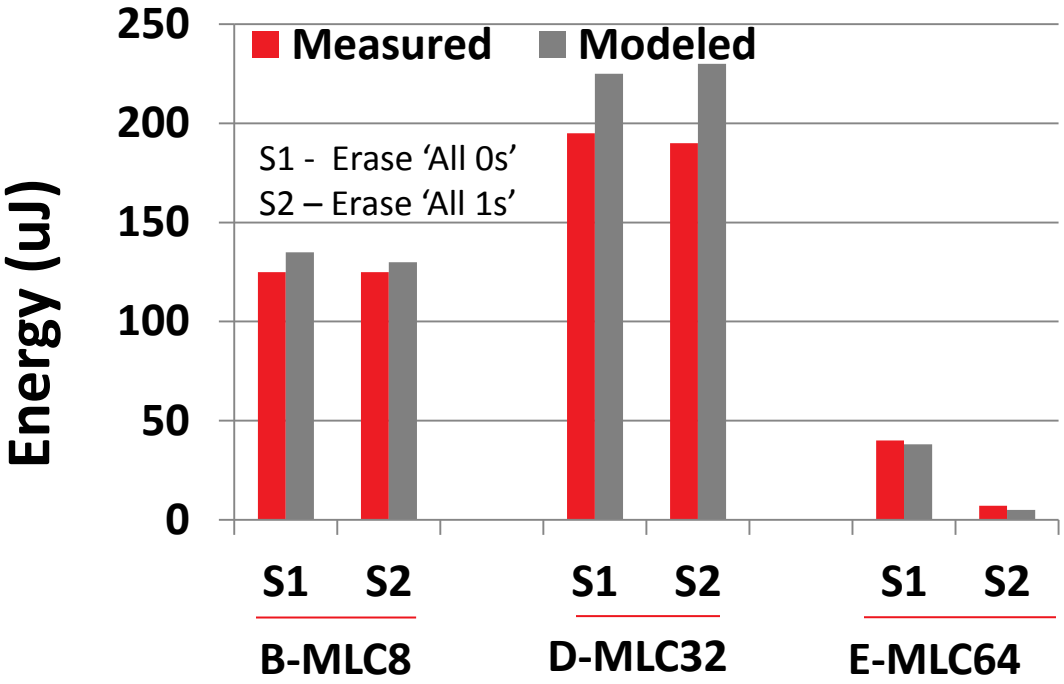
Workload Parameters	
N_{bits_1} (O)	Number of 1's to be read, or programmed.
Policy Parameters	
$N_{read\_verify\_cycles}$ (R)	Number of read verify cycles for a program operation.
N_{erase_cycles} (R)	Number of erase pulses required to erase a block.
$\Delta V_{th,slc}$ (O)	Threshold voltage difference between programmed and erased state for SLC flash. Defaults to 3V.
$\Delta V_{th,mlc}$ (O)	Threshold voltage difference between adjacent programmed states for MLC flash. Defaults to 0.9V.
$Optimize\_Write$ (O)	Boolean flag enabling optimization to certain threshold level transitions in MLC flash. Defaults to false.
$Optimize\_Erase$ (O)	Boolean flag enabling optimization if the threshold level of a FGT is unchanged after programming. Defaults to false.

Modeling Results for MLC Program Operation



B-MLC8: 72% accurate, D-MLC32: 74% accurate, E-MLC64: 85% accurate

Modeling Results for MLC Erase Operation



B-MLC8: 94% accurate, D-MLC32: 82% accurate, E-MLC64: 83% accurate

Backup for related work

Related Work - Power

SSD Power Characterization

System level

Yoo et al. HotStorage 2011
Bjørling et al. IEEE Data Eng. 2010
Hui et al. APSCC 2011

Memory modeling & measurement

Architecture level

Grupp et al. MICRO 2009
Jung et al. MSST 2012
Dong et al. ICCAD 2009
Thozhiyoor et al. ISCA 2008
Smullen et al. HPCA 2011

This work

Charge Transport & Chip design

Device/Circuit level

Lenzlinger et al. IEEE TED'64
Tanaka et al. ISSCC'94
Suh et al. ISSCC'95

Related Work - Reliability

Reliability management in a storage array

Kadav et al. SIGOPS 2010
Soundararajan et al. FAST 2010
Yang et al. HPCA 2011

System level

Architecture level

Pan et al. HPCA 2012
Boboila et al. FAST 2010
Sun et al. SNAPI 2011

SSD level modeling and measurement

This work

Device level

Mielke et al. IRPS 2006, 2008
Larcher et al. Trans. Devices, Material reliability 2002

Device & chip modeling measurements

Related Work - Scalability

This work

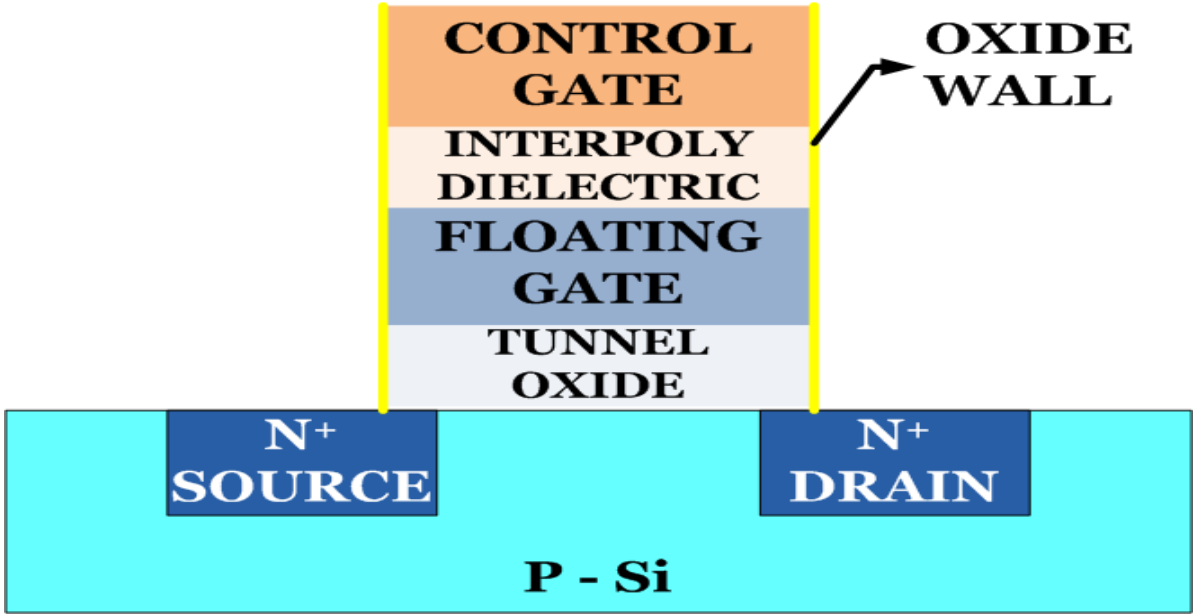
Architecture level

Ouyang et al. ASPLOS'14
Tiwari et al. FAST'13
Li et al. Usenix ATC'14

NAND Flash Primer

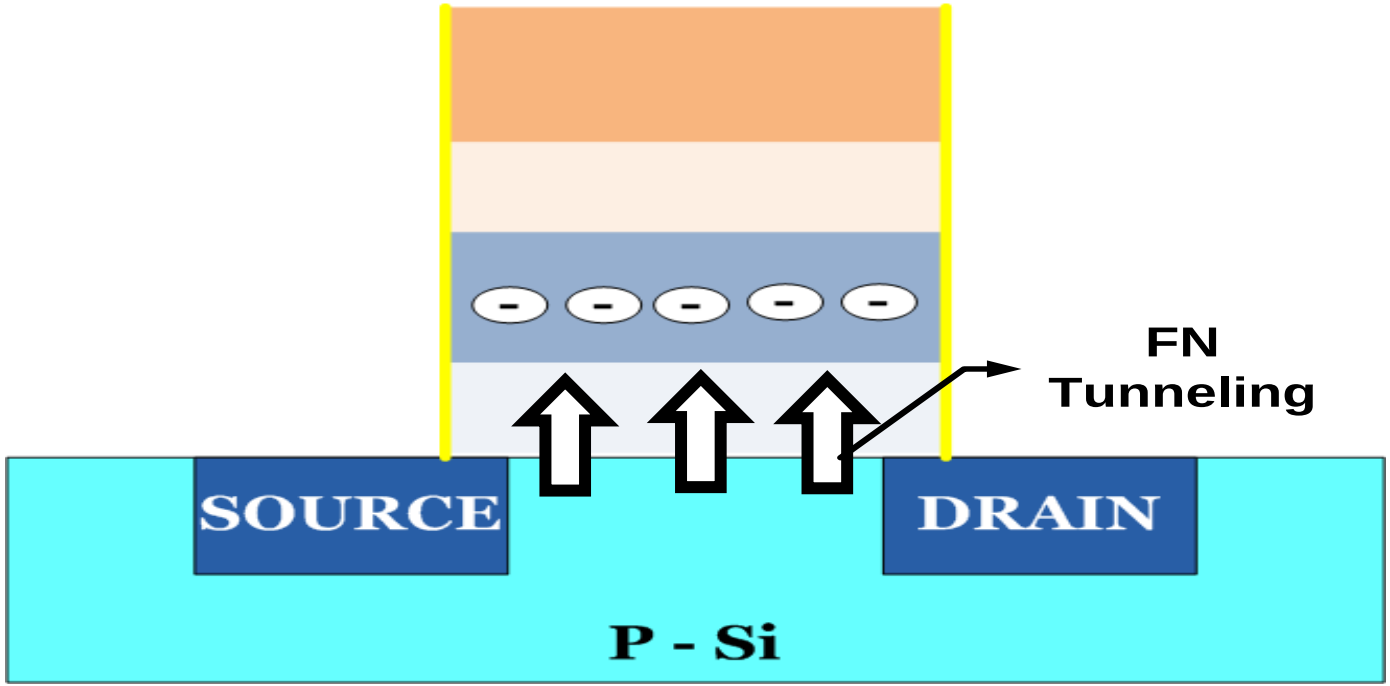
Floating Gate Transistor (FGT)

- Programmable Threshold Voltage

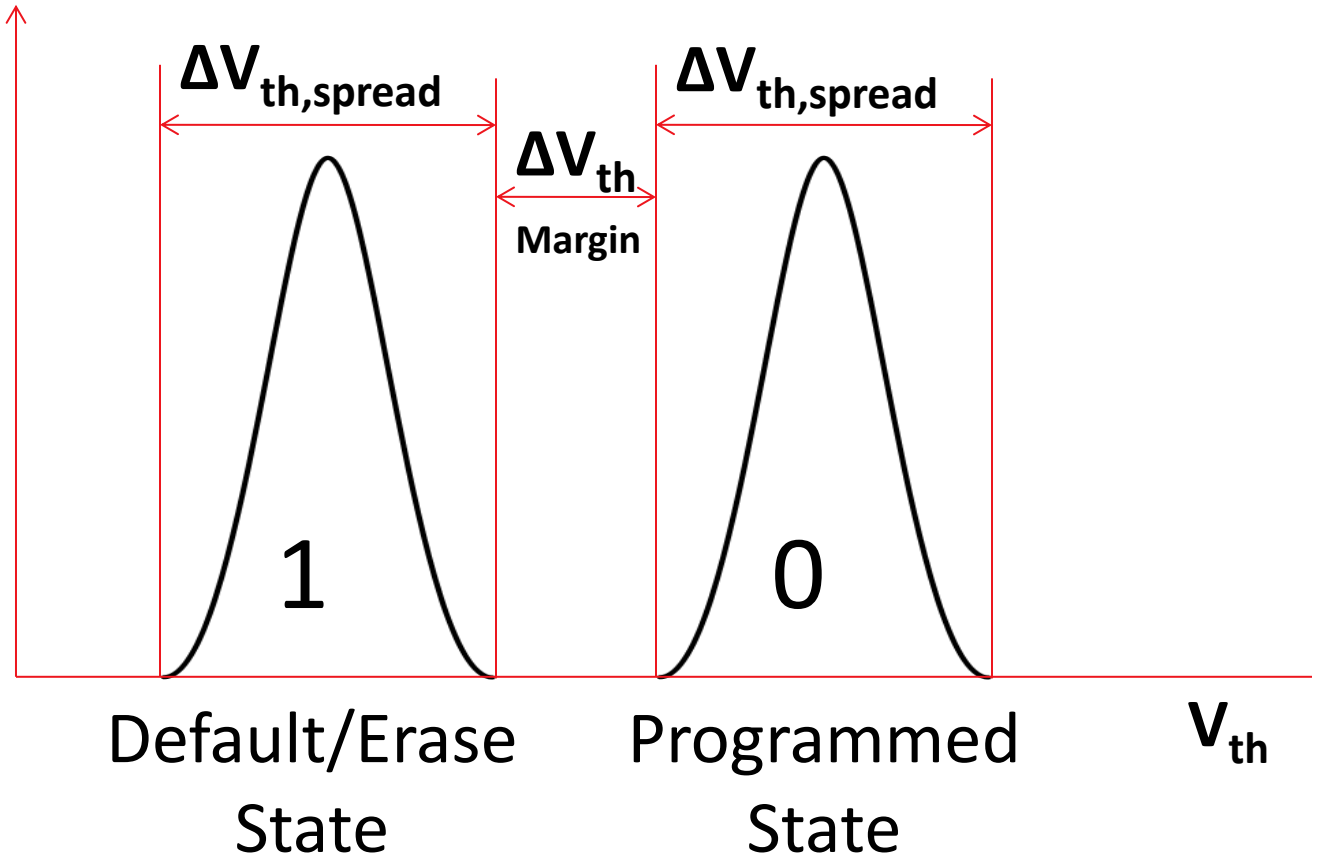


Write/Program

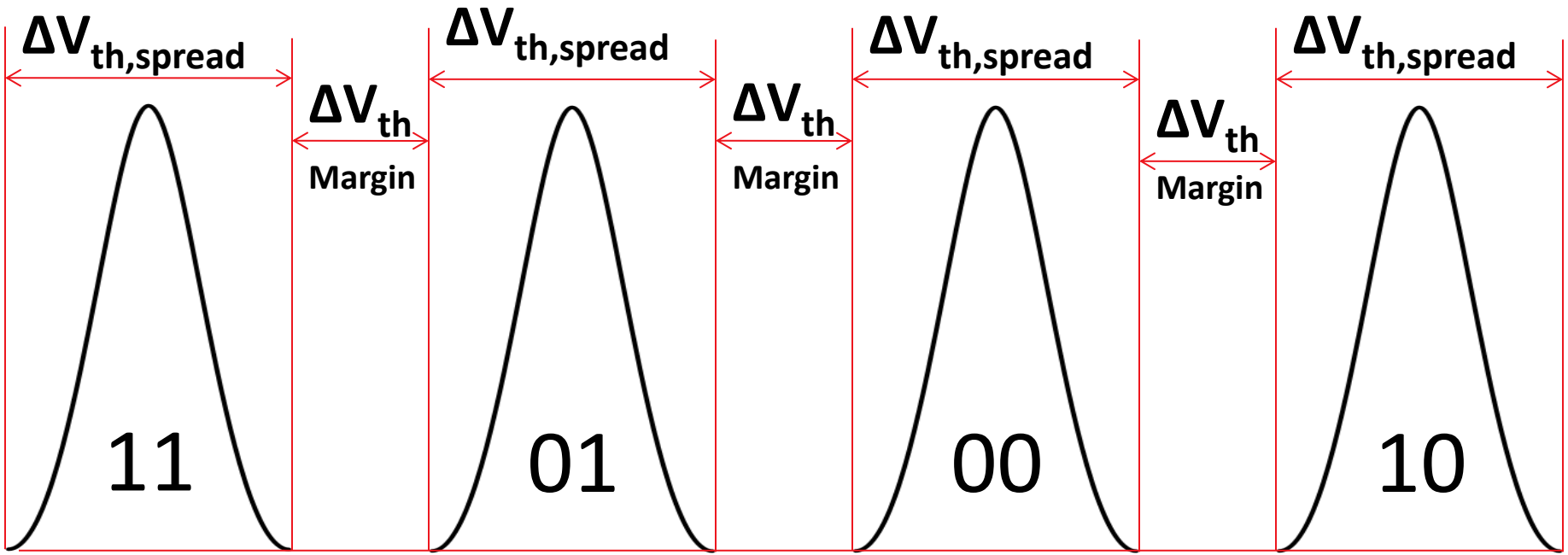
- Writes increase threshold voltage



Threshold voltage distribution for SLC flash

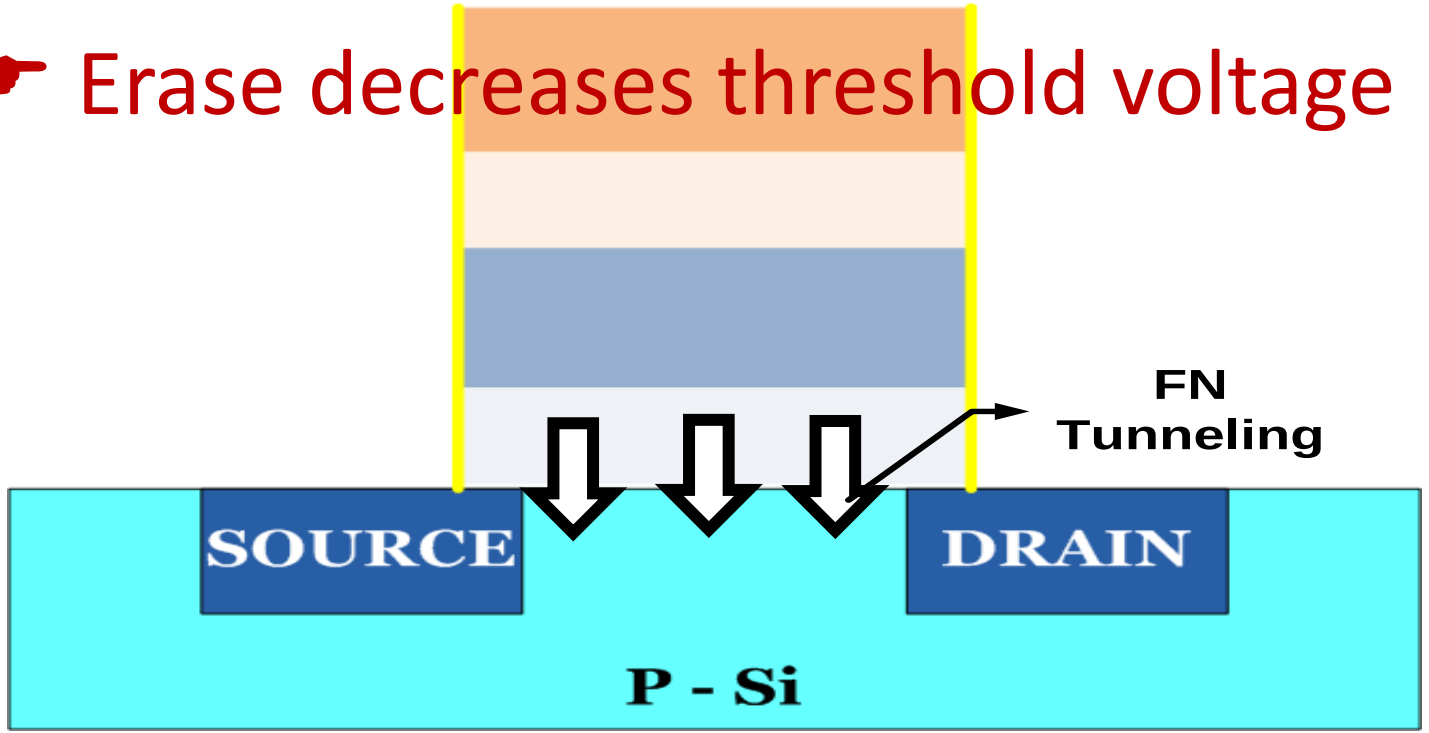


Threshold voltage distribution for MLC flash



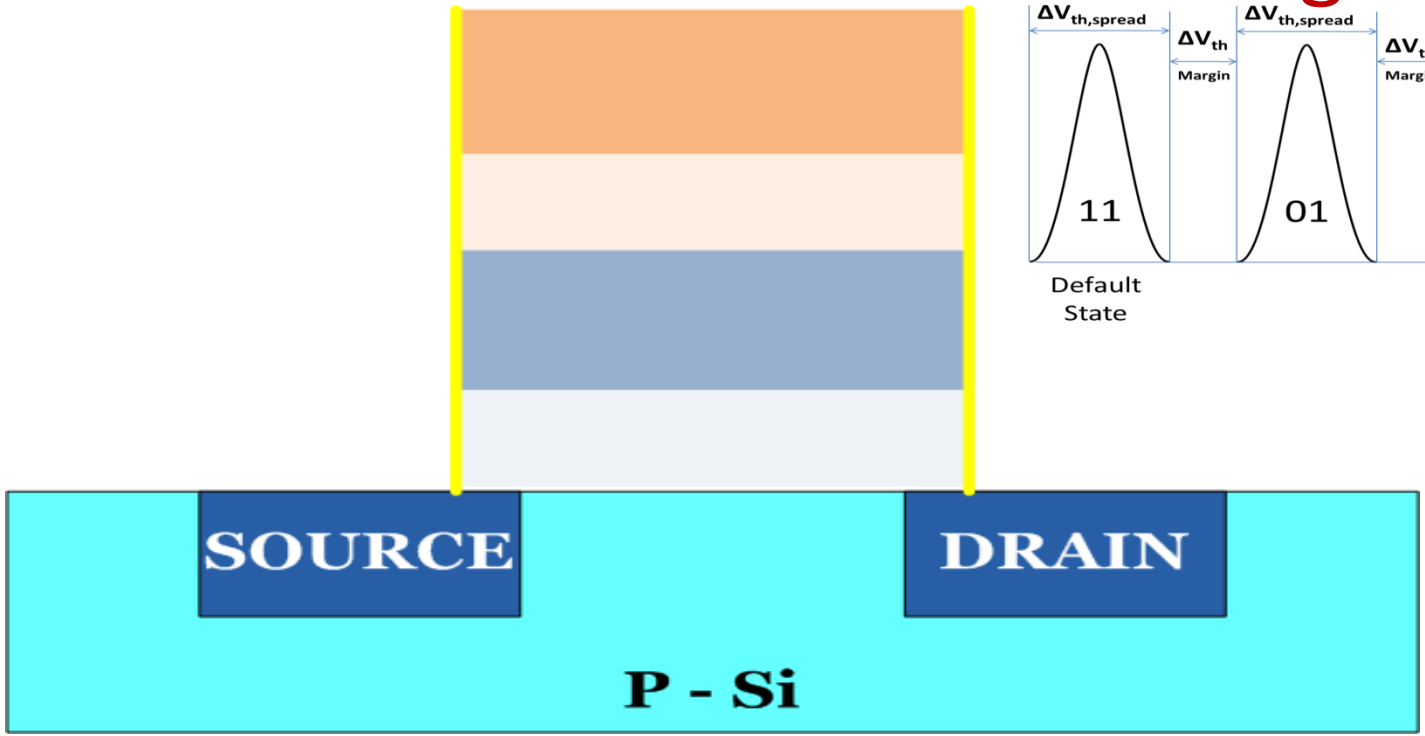
Erase – Switch back to default state

- Erase decreases threshold voltage

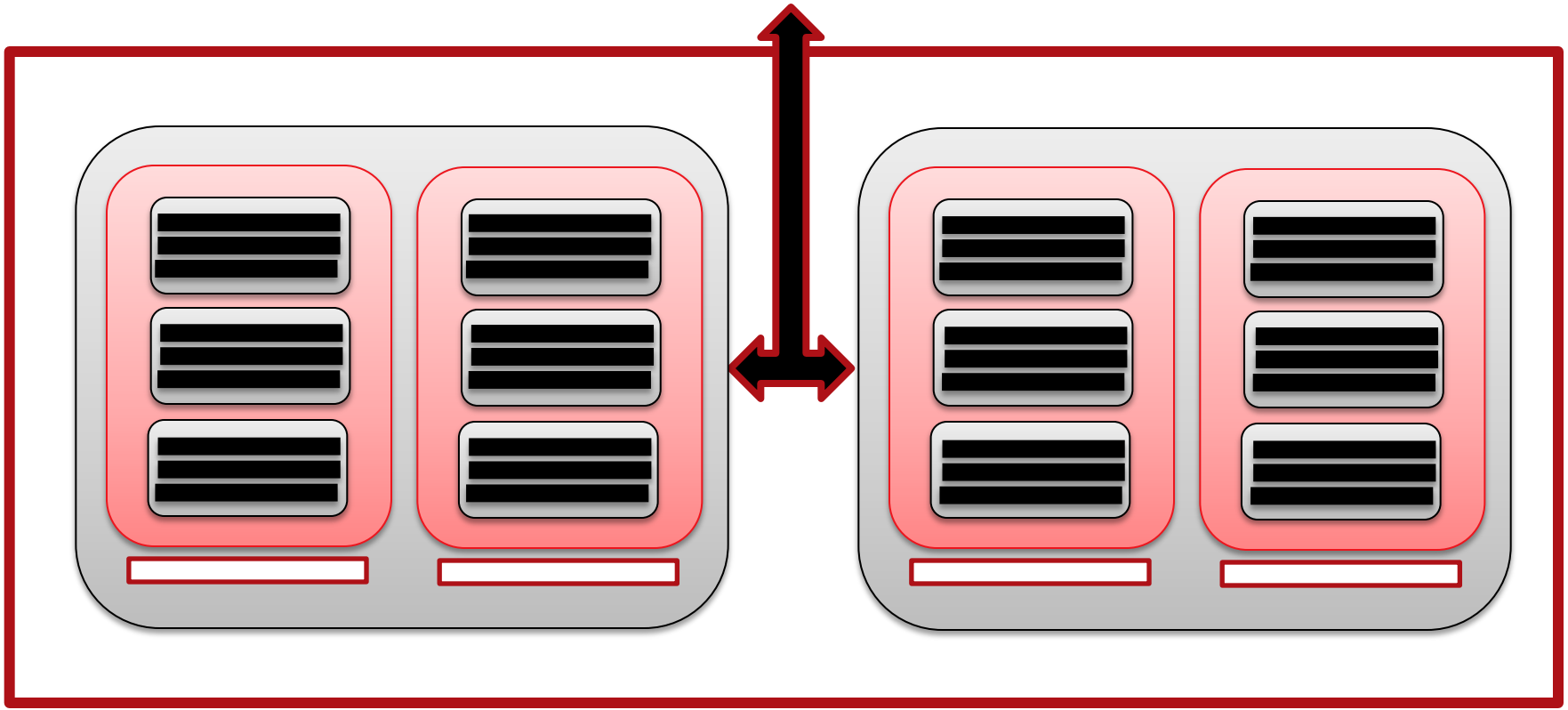


Reads – Sense Threshold Voltage

- Reads sense threshold voltage

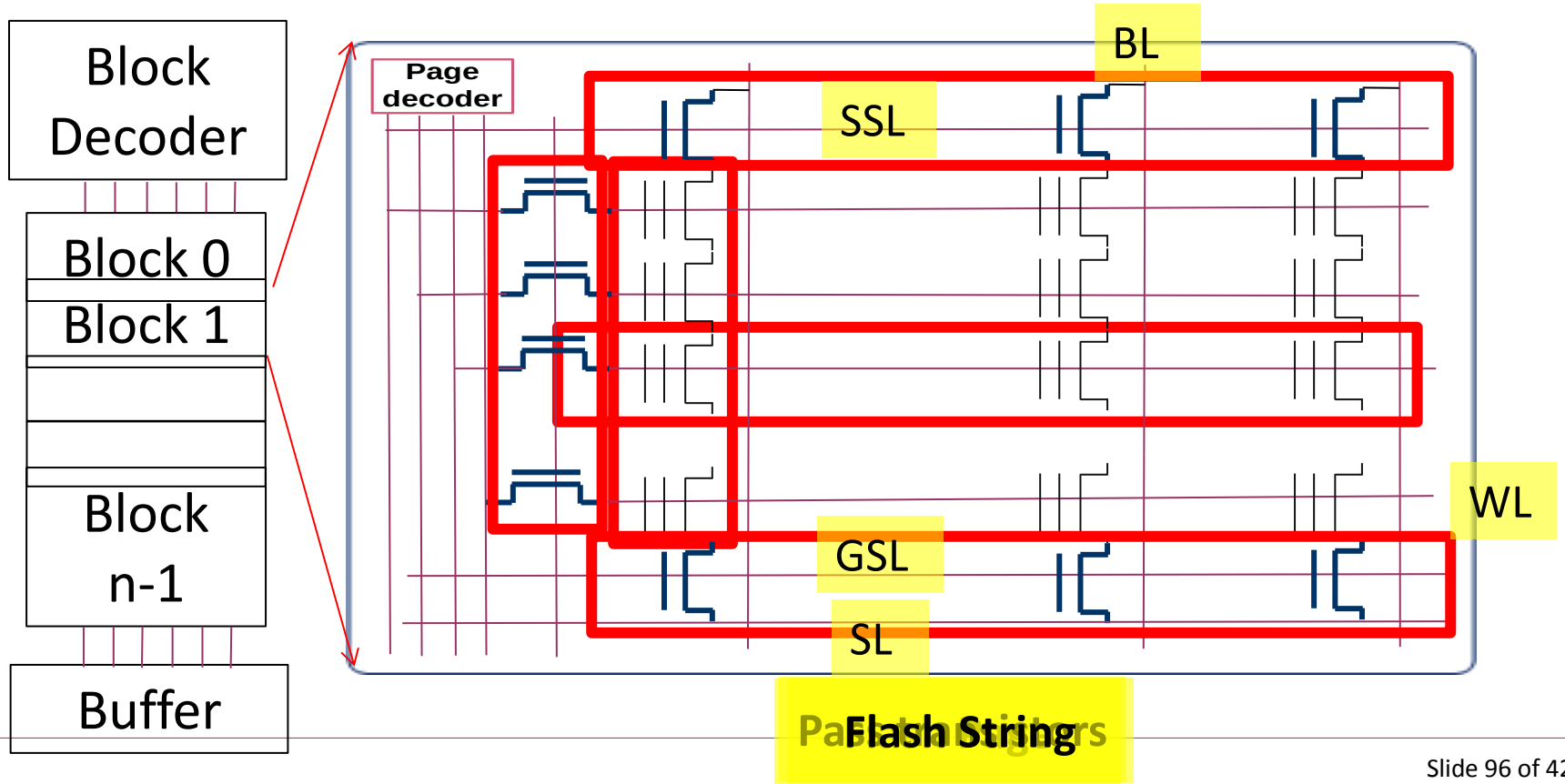


NAND Flash Memory Organization



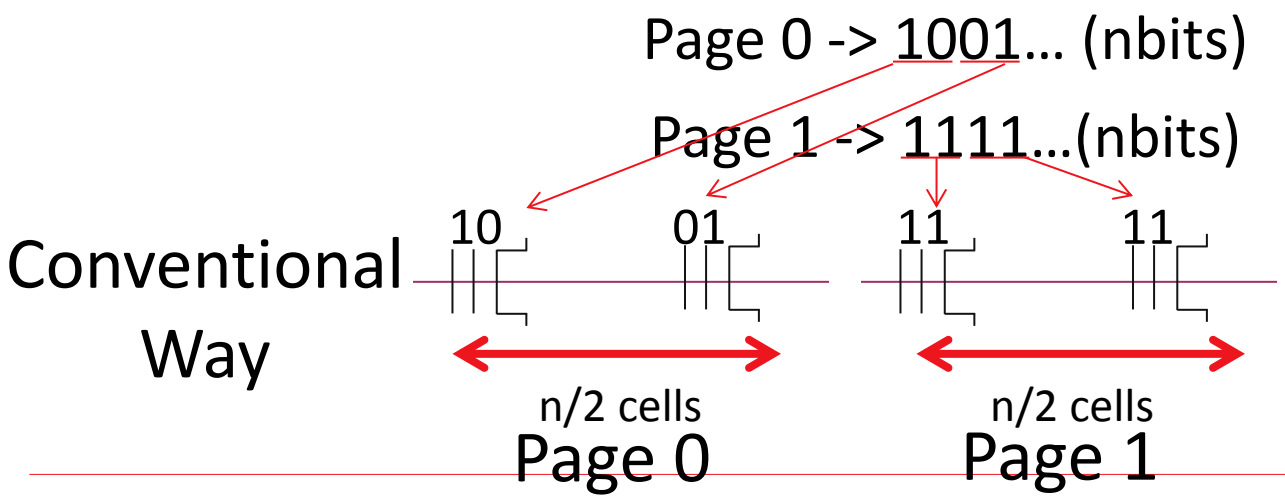
Package =>Dies =>Planes + Buffer =>Block=>Pages

NAND Flash Plane



Multi-page Architecture for MLC flash

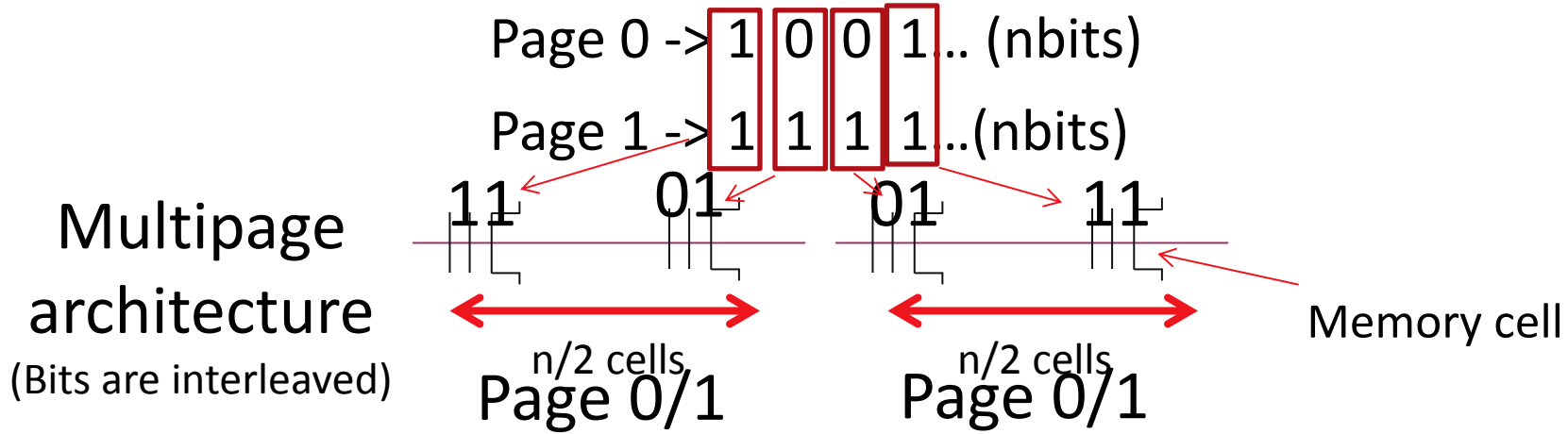
- Multi-page architecture defines how voltage levels are mapped to bits
- Lets assume 2-bit MLC



Latency per page
Latency to sense the second bit in a cell. Every read is a slow read.

Multi-page Architecture for MLC flash

- Multi-page architecture defines how voltage levels are mapped to bits
- Lets assume 2-bit are stored per memory cell



Fast Page: Latency to sense the first bit in a `cell (page 0 bits)

Slow Page: Latency to sense the second bit in a cell (page 1 bits)

General backup

Dissertation Contributions

Power **DATE 2010, TCAD 2013** **Chapter 2**	• Develop a detailed analytical model for NAND energy dissipation and evaluate the impact of various parameters on NAND energy dissipation
Reliability **HotStorage 2010, Techreport 2012** **Chapter 3 and 4**	• Build a NAND reliability model to model the impact of practical usage conditions on SSD reliability and propose firmware level algorithms to increase SSD endurance
Scalability **Chapter 5**	• Quantify the impact of conventional architectures on SSD performance and propose a new scalable SSD architecture to increase performance with capacity

SanDisk Optimus SAS SSDs

Optimus Eco™ 2.5" SAS SSDs Preliminary Specifications subject to change	
Performance	
Interface	6Gb/s SAS
Interface Ports	Dual/Wide
Sequential Read/Write (MB/s)**	Up to 530/530 MB/s[1] Up to 1 GB/s Wide-Port
Random Read/Write (IOPS)	Up to 90K/35K IOPS[2]
Capacity	
19nm eMLC User Capacities*	400GB, 800GB, 1.6TB, 2TB
Reliability	

Source [1]

Optimus MAX™ 2.5" SAS SSD	
Performance**	
Interface	SAS 6Gb/s
Interface Ports	Dual
Sequential Read/Write**	Up to 500/500 MB/s[1] per port
Random Read/Write (IOPS)	Up to 85K/11K[2]
Capacity	
19nm eMLC*	4TB
Reliability	

Source [1]