The Bleak Future of NAND Flash Memory*

Laura M. Grupp†, *John D. Davis*‡, *Steven Swanson*†

†*Department of Computer Science and Engineering, University of California, San Diego*

‡*Microsoft Research, Mountain View*

Abstract

In recent years, flash-based SSDs have grown enormously both in capacity and popularity. In high-performance enterprise storage applications, accelerating adoption of SSDs is predicated on the ability of manufacturers to deliver performance that far exceeds disks while closing the gap in cost per gigabyte. However, while flash density continues to improve, other metrics such as a reliability, endurance, and performance are all declining. As a result, building larger-capacity flash-based SSDs that are reliable enough to be useful in enterprise settings and high-performance enough to justify their cost will become challenging.

In this work, we present our empirical data collected from 45 flash chips from 6 manufacturers and examine the performance trends for these raw flash devices as flash scales down in feature size. We use this analysis to predict the performance and cost characteristics of future SSDs. We show that future gains in density will come at significant drops in performance and reliability. As a result, SSD manufacturers and users will face a tough choice in trading off between cost, performance, capacity and reliability.

1 Introduction

Flash-based Solid State Drives (SSDs) have enabled a revolution in mobile computing and are making deep inroads into data centers and high-performance computing. SSDs offer substantial performance improvements relative to disk, but cost is limiting adoption in cost-sensitive applications and reliability is limiting adoption in higher-end machines. The hope of SSD manufactures is that improvements in flash density through silicon feature size scaling (shrinking the size of a transistor) and storing more bits per storage cell will drive down costs and increase their adoption. Unfortunately, trends in flash technology suggest that this is unlikely.

While flash density in terms of bits/mm^2 and feature size scaling continues to increase rapidly, all other figures of merit for flash – performance, program/erase endurance, energy efficiency, and data retention time – decline steeply as density rises. For example, our data show each additional bit per cell increases write latency by 4× and reduces program/erase lifetime by 10× to 20× (as shown in Figure 1), while providing decreasing returns in density (2×, 1.5×, and 1.3× between 1-,2-,3- and 4-bit cells, respectively). As a result, we are reaching the limit of what current flash management techniques can deliver in terms of usable capacity – we may be able to build more spacious SSDs, but they may be too slow and unreliable to be competitive against disks of similar cost in enterprise applications.

This paper uses empirical data from 45 flash chips manufactured by six different companies to identify trends in flash technology scaling. We then use those trends to make projections about the performance and cost of future SSDs. We construct an idealized SSD model that makes optimistic assumptions about the efficiency of the flash translation layer (FTL) and shows that as flash continues to scale, it will be extremely difficult to design SSDs that reduce cost per bit without becoming either too slow or too unreliable (or both) as to be unusable in enterprise settings. We conclude that the cost per bit for enterprise-class SSDs targeting general-purpose applications will stagnate.

The rest of this paper is organized as follows. Section 2 outlines the current state of flash technology. Section 3 describes the architecture of our idealized SSD design, and how we combine it with our measurements to project the behavior of future SSDs. Section 4 presents the results of this idealized model, and Section 5 concludes.

*Correction to the original manuscript: The original version of this paper that appeared in the printed conference proceedings accidentally included results for a 40 GB baseline SSD rather than 320 GB. This version includes the correct values for a 320 GB drive.

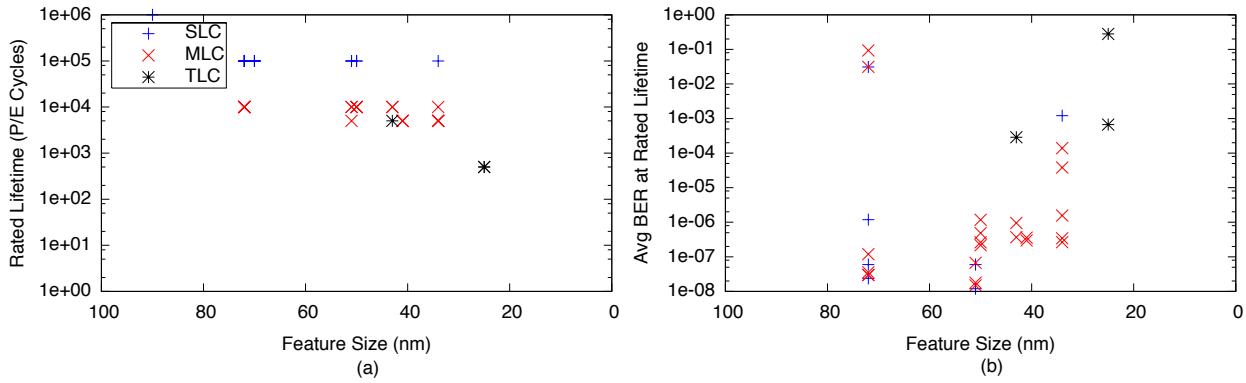

Figure 1: **Trends in Flash's Reliability** Increasing flash's density by adding bits to a cell or by decreasing feature size reduces both (a) lifetime and (b) reliability.

2 The State of NAND Flash Memory

Flash-based SSDs are evolving rapidly and in complex ways – while manufacturers drive toward higher densities to compete with HDDs, increasing density by using newer, cutting edge flash chips can adversely affect performance, energy efficiency and reliability.

To enable higher densities, manufacturers scale down the manufacturing feature size of these chips while also leveraging the technology's ability to store multiple bits in each cell. Most recently on the market are 25 nm cells which can store three bits each (called Triple Level Cells, or *TLC*). Before TLC came 2-bit, multi-level cells (MLC) and 1-bit single-level cells (SLC). Techniques that enable four or more bits per cell are on the horizon [12].

Figure 2, collects the trend in price of raw flash memory from a variety of industrial sources, and shows the drop in price per bit for the higher density chips. Historically, flash cost per bit has dropped by between 40 and 50% per year [3]. However, over the course of 2011, the price of flash flattened out. If flash has trouble scaling beyond 12nm (as some predict), the prospects for further cost reductions are uncertain.

The limitations of MLC and TLC's reliability and performance arise from their underlying structures. Each flash cell comprises a single transistor with an added layer of metal between the gate and the channel, called the floating gate. To change the value stored in the cell, the program operation applies very high voltages to its terminals which cause electrons to tunnel through the gate oxide to reach the floating gate. To erase a cell, the voltages are reversed, pulling the electrons off the floating gate. Each of these operations strains the gate oxide, until eventually it no longer isolates the floating gate, making it impossible to store charge.

The charge on the floating gate modifies the threhold voltage, V_{TH} of the transistor (i.e., the voltage at which the transistor turns on and off). In a programmed SLC cell, V_{TH} will be in one of two ranges (since programming is not perfectly precise), depending on the value the cell stores. The two ranges have a "guard band" between them. Because the SLC cell only needs two ranges and a single guard band, both ranges and the guard band can be relatively wide. Increasing the number of bits stored from one (SLC) to two (MLC) increases the number of distributions from two to four, and requires two additional guard bands. As a result, the distributions must be tighter and narrower. The necessity of narrow V_{TH} distributions increases programming time, since the chip must make more, finer adjustments to V_{TH} to program the cell correctly (as described below). At the same time, the narrow guard band reduces reliability. TLC cells make this problem even worse: They must accomodate eight V_{TH} levels and seven guard bands.

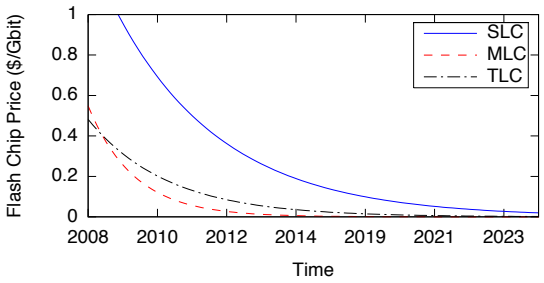

Figure 2: **Trends in Flash Prices** Flash prices reflect the target markets. Low density, SLC, parts target higher-priced markets which require more reliability while high density MLC and TLC are racing to compete with low-cost HDDs. Cameras, iPods and other mobile devices drive the low end.

We present empirical evidence of worsening lifetime and reliability of flash as it reaches higher densities. We collected this data from 45 flash chips made by six manufacturers spanning feature sizes from 72 nm to 25 nm. Our flash characterization system (described in [4]) allows us to issue requests to a raw flash chip without FTL interference and measure the latency of each of these operations with 10 ns resolution. We repeat this program-erase cycle (P/E cycle) until each measured

block reaches the rated lifetime of its chip.

Figure 1 shows the chips' rated lifetime as well as the bit error rate (BER) measured at that lifetime. The chips' lifetimes decrease slowly with feature size, but fall precipitously across SLC, MLC and TLC devices. While the error rates span a broad range, there is a clear upward trend as feature size shrinks and densities increase. Applications that require more reliable or longer-term storage prefer SLC chips and those at larger feature sizes because they experience far fewer errors for many more cycles than denser technology.

Theory and empirical evidence also indicate lower performance for denser chips, primarily for the program or write operation. Very early flash memory would apply a steady, high voltage to any cell being programed for a fixed amount of time. However, Suh et al. [10] quickly determined that the Incremental Step Pulse Programming (ISPP) would be far more effective in tolerating variation between cells and in environmental conditions. ISPP performs a series of program pulses each followed by a read-verify step. Once the cell is programmed correctly, programming for that cell stops. This algorithm is necessary because programming is a one-way operation: There is no way to "unprogram" a cell short of erasing the entire block, and overshooting the correct voltage results in storing the wrong value. ISPP remains a key algorithm in modern chips and is instrumental in improving the performance and reliability of higher-density cells.

Not long after Samsung proposed MLC for NAND flash [5, 6], Toshiba split the two bits to separate pages so that the chip could program each page more quickly by moving the cell only halfway through the voltage range with each operation [11]. Much later, Samsung provided further performance improvements to pages stored in the least significant bit of each cell [8]. By applying fast, imprecise pulses to program the fast pages, and using fine-grain, precise pulses to program the slow pages. These latter pulses generate the tight V_{TH} distributions that MLC devices require, but they make programming much slower. All the MLC and TLC devices we tested split and program the bits in a cell this way.

For SSD designers, this performance variability between pages leads to an opportunity to easily trade off capacity and performance [4, 9]. The SSD can, for example use only the fast pages in MLC parts, sacrificing half their capacity but making latency comparable to SLC. In this work, we label such a configuration "MLC-1" – an MLC device using just one bit per cell. Samsung and Micron have formalized this trade-off in multi-level flash by providing single and multi-level cell modes [7] in the same chip and we believe FusionIO uses the property in the controller of their SMLC-based drives [9].

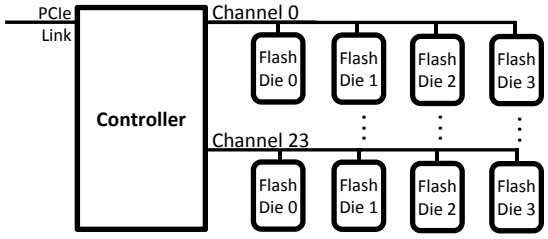

Figure 3: **Architecture of SSD-CDC** The architecture of our baseline SSD. This structure remains constant while we scale the technology used for each flash die.

Architecture Parameter	Value
Example Interface	PCIe 1.1x4
FTL Overhead Latency	30 μs
Channels	24
Channel Speed	400 MB/s [1]
Dies per Channel (DPC)	4
Baseline Parameter	**Value**
SSD Price	$7,800
Capacity	320 GB
Feature Size	34 nm
Cell Type	MLC

Table 1: **Architecture and Baseline Configuration of SSD-CDC** These parameters define the Enterprise-class, Constant Die Count SSD (SSD-CDC) architecture and starting values for the flash technology it contains.

3 A Prototypical SSD

To model the effect of evolving flash characteristics on complete SSDs we combine empirical measurement of flash chips in an SSD architecture with a constant die count called *SSD-CDC*. SSD-CDC's architecture is representative of high-end SSDs from companies such as FusionIO, OCZ and Virident. We model the complexities of FTL design by assuming optimistic constants and overheads that provide upper bounds on the performance characteristics of SSDs built with future generation flash technology.

Section 3.1 describes the architecture of SSD-CDC, while Section 3.2 describes how we combine this model with our empirical data to estimate the performance of an SSD with fixed die area.

3.1 SSD-CDC

Table 1 describes the parameters of SSD-CDC's architecture and Figure 3 shows a block representation of its architecture. SSD-CDC manages an array of flash chips and presents a block-based interface. Given current trends in PCIe interface performance, we assume that the PCIe link is not a bottleneck for our design.

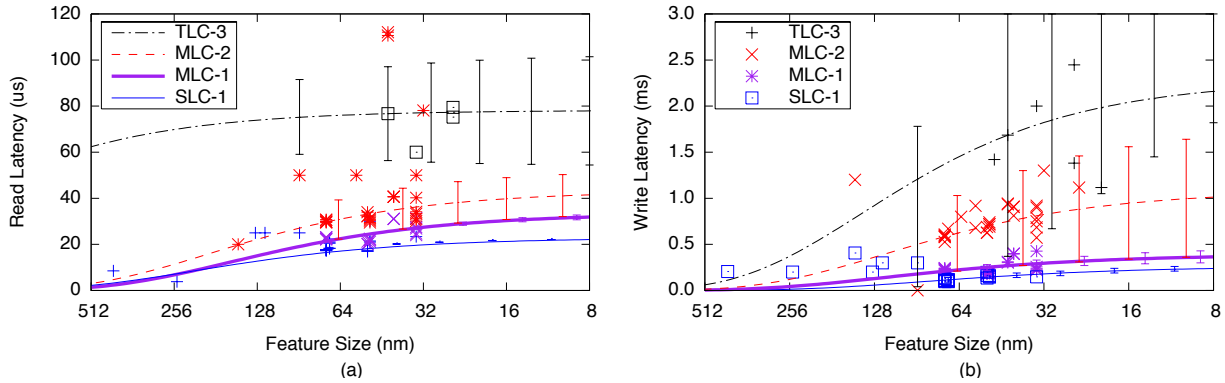

Figure 4: **Flash Chip Latency Trends** Fitting an exponential to the collection of data for each cell technology, SLC-1, MLC-1, MLC-2 and TLC-3, allows us to project the behavior of future feature sizes for (a) read latency and (b) write latency. Doing the same with one standard deviation above and below the average for each chip yields a range of probable behavior, as shown by the error bars.

Configuration	Read Latency (μs)		Write Latency (μs)	
	Equation	-1nm	Equation	-1nm
SLC-1	max $= 24.0e^{-3.5e-3f}$	0.36%	max $= 287.0e^{-1.1e-2f}$	1.07%
	avg $= 23.4e^{-3.2e-3f}$	0.32%	avg $= 262.6e^{-1.2e-2f}$	1.19%
	min $= 22.8e^{-2.9e-3f}$	0.29%	min $= 239.3e^{-1.3e-2f}$	1.34%
MLC-1	max $= 34.8e^{-6.9e-3f}$	0.69%	max $= 467.3e^{-1.0e-2f}$	1.01%
	avg $= 33.5e^{-6.3e-3f}$	0.63%	avg $= 390.0e^{-8.7e-3f}$	0.87%
	min $= 32.2e^{-5.6e-3f}$	0.57%	min $= 316.5e^{-7.0e-3f}$	0.70%
MLC-2	max $= 52.5e^{-4.5e-3f}$	0.45%	max $= 1778.2e^{-8.3e-3f}$	0.84%
	avg $= 43.3e^{-5.2e-3f}$	0.52%	avg $= 1084.4e^{-8.6e-3f}$	0.86%
	min $= 34.2e^{-6.6e-3f}$	0.66%	min $= 393.7e^{-9.9e-3f}$	1.00%
†TLC-3	max $= 102.5e^{-1.3e-3f}$	0.13%	max $= 4844.8e^{-1.1e-2f}$	1.12%
	avg $= 78.2e^{-4.4e-4f}$	0.04%	avg $= 2286.2e^{-7.1e-3f}$	0.71%
	min $= 54.0e^{9.9e-4f}$	-0.10%	min $= 2620.8e^{-4.6e-2f}$	4.67%

Table 2: **Latency Projections** We generated these equations by fitting an exponential ($y = Ae^{bf}$) to our empirical data, and they allow us to project the latency of flash as a function of feature size (f) in nm. The percentages represent the increase in latency with 1nm shrinkage. †The trends for TLC are less certain than for SLC or MLC, because our data for TLC devices is more limited.

Number	Metric	Value
1	$Capacity_{proj}$	$= Capacity_{base} \times \left(\frac{BitsPerCell_{proj}}{BitsPerCell_{base}}\right) \times \left(\frac{FeatureSize_{base}}{FeatureSize_{proj}}\right)^2$
2	SSD_BW_{proj}	$= ChannelCount \times ChannelBW_{proj}$
3	$ChannelBW_{proj}$	$= \frac{(DiesPerChannel-1)*PageSize}{DieLatency_{proj}}$, when $DieLatency_{proj} \leq BWThreshold$
4	$ChannelBW_{proj}$	$= ChannelSpeed$, when $DieLatency_{proj} > BWThreshold$
5	$TransferTime$	$= \frac{PageSize}{ChannelSpeed}$
6	$BWThreshold$	$= (DiesPerChannel - 1) \times TransferTime$
7	SSD_IOPs_{proj}	$= ChannelCount \times ChannelIOPs_{proj}$
9	$ChannelIOPs_{proj}$	$= \frac{1}{TransferTime}$, when $DieLatency_{proj} \leq IOPsThreshold$
8	$ChannelIOPs_{proj}$	$= \frac{(DiesPerChannel-1)}{DieLatency_{proj}}$, when $DieLatency_{proj} > IOPsThreshold$
10	$TransferTime$	$= \frac{AccessSize}{ChannelSpeed}$
11	$IOPsThreshold$	$= (DiesPerChannel - 1) \times TransferTime$

Table 3: **Model's Equations** These equations allow us to scale the metrics of our baseline SSD to future process technologies and other cell densities.

The SSD's controller implements the FTL. We estimate that this management layer incurs an overhead of 30 μs for ECC and additional FTL operations. The controller coordinates 24 channels, each of which connects four dies to the controller via a 400 MB/s bus. To fix the cost of SSD-CDC, we assume a constant die count equal to 96 dies.

3.2 Projections

We now describe our future projections for seven metrics of SSD-CDC: capacity, read latency, write latency, read bandwidth, write bandwidth, read IOPs and write IOPs. Table 1 provides baseline values for SSD-CDC and Table 2 summarizes the projections we make for the underlying flash technology. This section describes the formulas we use to compute each metric from the projections (summarized in Table 3). Some of the calculations involve making simplifying assumptions about SSD-CDC's behavior. In those cases, we make the assumption that maximizes the SSD's performance.

Capacity Equation 1 calculates the capacity of SSD-CDC, by scaling the capacity of the baseline by the square of the ratio of the projected feature size to the baseline feature size (34 nm). We also scale capacity depending on the number of bits per cell (BPC) the projected chip stores relative to the baseline BPC (2 – MLC). In some cases, we configure SSD-CDC to store fewer bits per cell than a projected chip allows, as in the case of MLC-1. In these cases, the projected capacity would reflect the *effective* bits per cell.

Latency To calculate the projected read and write latencies, we fit an exponential function to the empirical data for a given cell type. Figure 4 depicts both the raw latency data and the curves fitted to SLC-1, MLC-1, MLC-2 and TLC-3. To generate the data for MLC-1, which ignores the "slow" pages, we calculate the average latency for reads and writes for the "fast" pages only. Other configurations supporting reduced capacity and improved latency, such as TLC-1 and TLC-2, would use a similar method. We do not present these latter configurations, because there is very little TLC data available to create reliable predictions. Figure 4 shows each collection of data with the fitted exponentials for average, minimum and maximum, and Table 2 reports the equations for these fitted trends. We calculate the projected latency by adding the values generated by these trends to the SSD's overhead reported in Table 1.

Bandwidth To find the bandwidth of our SSD, we must first calculate each channel's bandwidth and then multiply that by the number of channels in the SSD (Equation 2). Each channel's bandwidth requires an understanding of whether channel bandwidth or per-chip bandwidth is the bottleneck. Equation 6 determines the threshold between these two cases by multiplying the transfer time (see Equation 5) by one less than the number of dies on the channel. If the latency of the operation on the die is larger than this number, the die is the bottleneck and we use Equation 3. Otherwise, the channel's bandwidth is simply the speed of its bus (Equation 4).

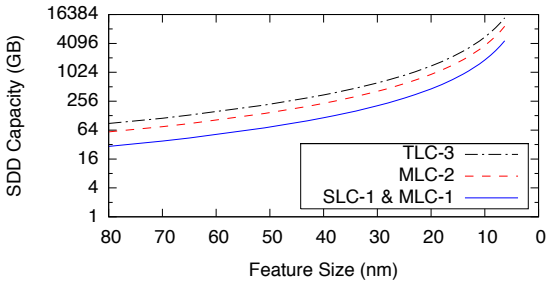

Figure 5: **Scaling of SSD Capacity** Flash manufacturers increase SSDs' capacity through both reducing feature size and storing more bits in each cell.

IOPs The calculation for IOPs is very similar to bandwidth, except instead of using the flash's page size in all cases, we also account for the access size since it effects the transfer time: If the access size is smaller than one page, the system still incurs the read or write latency of one entire page access. Equations 7-11 describe the calculations.

4 Results

This section explores the performance and cost of SSD-CDC in light of the flash feature size scaling trends described above. We explore four different cell technologies (SLC-1, MLC-1, MLC-2, and TLC-3) and feature sizes scaled down from 72 nm to 6.5 nm (the smallest feature size targeted by industry consensus as published in the International Technology Roadmap for Semiconductors (ITRS) [2]), using a fixed silicon budget for flash storage.

4.1 Capacity and cost

Figure 5 shows how SSD-CDC's density will increase as the number of bits per cell rises and feature size continues to scale. Even with the optimistic goal of scaling flash cells to 6.5 nm, SSD-CDC can only achieve capacities greater than ~4.6 TB with two or more bits per cell. TLC allows for capacities up to 14 TB – pushing capacity beyond this level will require more dies.

Since capacity is one of the key drivers in SSD design and because it is the only aspect of SSDs that improves consistently over time, we plot the remainder of the characteristics against SSD-CDC's capacity.

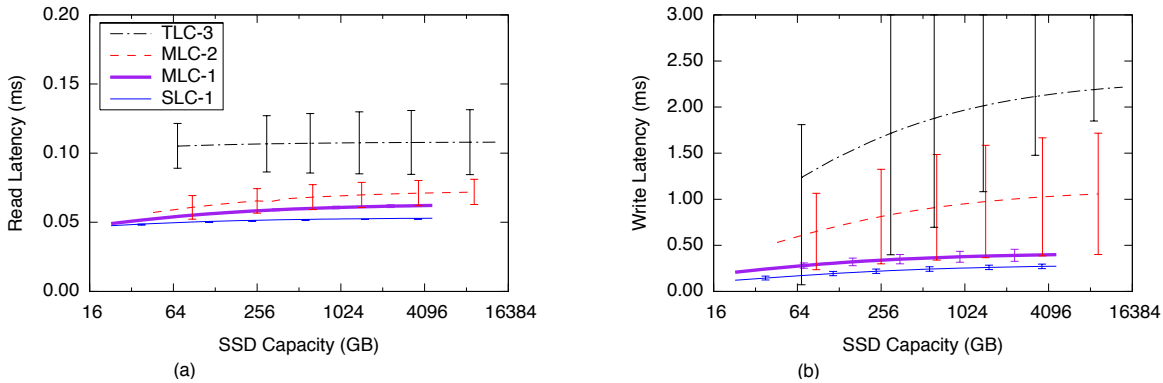

Figure 6: **SSD Latency** In order to achieve higher densities, flash manufacturers must sacrifice (a) read and (b) write latency.

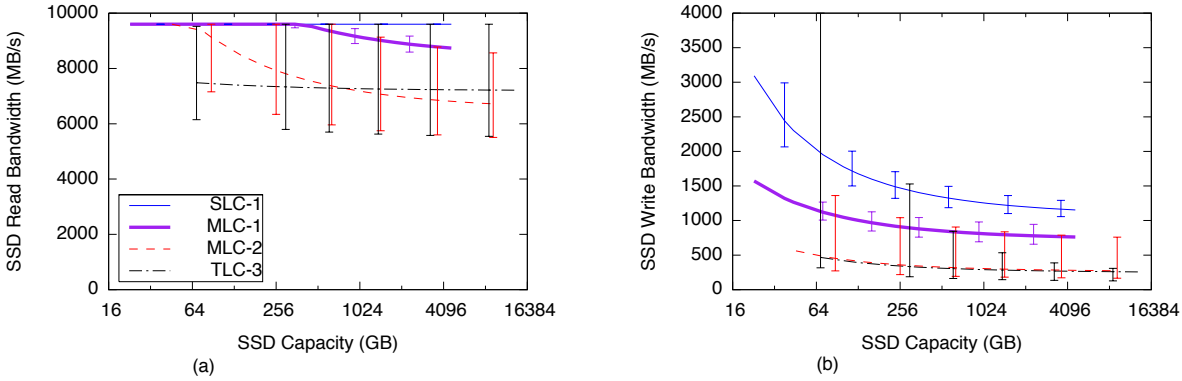

Figure 7: **SSD Bandwidth** SLC will continue to be the high performance option. To obtain higher capacities without additional dies and cost will require a significant performance hit in terms of (a) read and (b) write bandwidth moving from SLC-1 to MLC-2 or TLC-3.

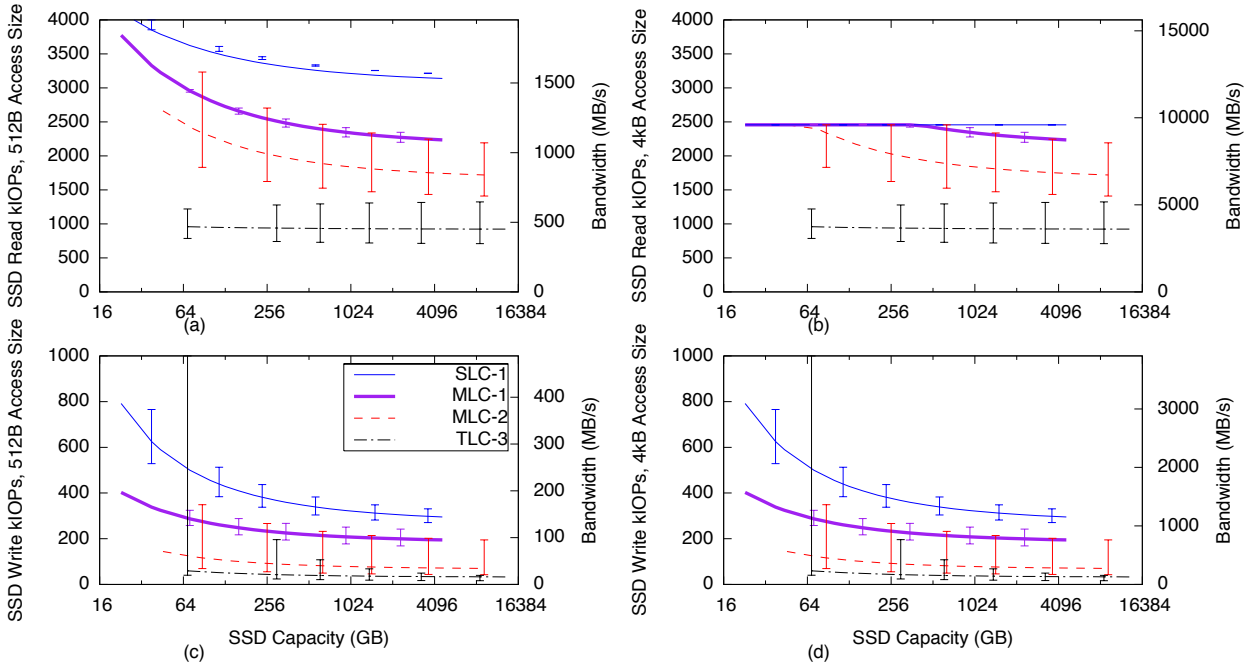

Figure 8: **SSD IOPS** With a fixed die area, higher capacities can only be achieved with low-performing MLC-2 and TLC-3 technologies, for 512B (a) reads and (c) writes and for 4kB (b) reads and (d) writes.

4.2 Latency

Reduced latency is among the frequently touted advantages of flash-based SSDs over disks, but changes in flash technology will erode the gap between disks and SSDs. Figure 6 shows how both read and write latencies increase with SSD-CDC's capacity. Reaching beyond 4.6 TB pushes write latency to 1 ms for MLC-2 and over 2.1 ms for TLC. Read latency, rises to least 70 μs for MLC-2 and 100 μs for TLC.

The data also makes clear the choices that SSD designers will face. Either SSD-CDC's capacity stops scaling at $\sim$4.6 TB or its read and write latency increases sharply because increasing drive capacity with fixed die area would necessitate switching cell technology from SLC-1 or MLC-1 to MLC-2 or TLC-3. With current trends, our SSDs could be up to 34$\times$ larger, but the latency will be 1.7$\times$ worse for reads and 2.6$\times$ worse for writes. This will reduce the write latency advantage that SSDs offer relative to disk from 8.3$\times$(vs. a 7 ms disk access) to just 3.2$\times$. Depending on the application, this reduced improvement may not justify the higher cost of SSDs.

4.3 Bandwidth and IOPs

SSDs offer moderate gains in bandwidth relative to disks, but very large improvements in random IOP performance. However, increases in operation latency will drive down IOPs and bandwidth.

Figure 7 illustrates the effect on bandwidth. Read bandwidth drops due to the latency of the operation on the flash die. Operation latency also causes write bandwidth to decrease with capacity.

SSDs provide the largest gains relative to disks for small, random IOPs. We present two access sizes – the historically standard disk block size of 512 B and the most common flash page size and modern disk access size of 4 kB. Figure 8 presents the performance in terms of IOPs. When using the smaller, unaligned 512B accesses, SLC and MLC chips must access 4 kB of data and the SSD must discard 88% of the accessed data. For TLC, there is even more wasted bandwidth because page size is 8 kB.

When using 4kB accesses, MLC IOPs drop as density increases, falling by 18% between the 64 and 1024 GB configurations. Despite this drop, the data suggest that SSDs will maintain an enormous (but slowly shrinking) advantage relative to disk in terms of IOPs. Even the fastest hard drives can sustain no more than 200 IOPs, and the slowest SSD configuration we consider achieves over 32,000 IOPs.

Figure 9 shows all parameters for an SSD made from MLC-2 flash normalized to SSD-CDC configured with currently available flash. Our projections show that the cost of the flash in SSD-CDC will remain roughly constant and that density will continue to increase (as long as flash scaling continues as projected by the ITRS). However, they also show that access latencies will increase by 26% and that bandwidth (in both MB/s and IOPS) will drop by 21%.

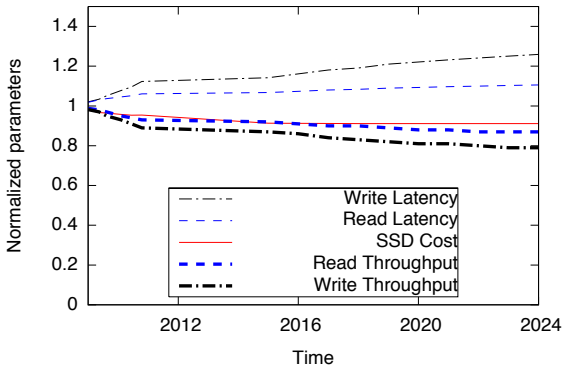

Figure 9: **Scaling of all parameters** While the cost of an MLC-based SSD remains roughly A constant, read and particularly write performance decline.

5 Conclusion

The technology trends we have described put SSDs in an unusual position for a cutting-edge technology: SSDs will continue to improve by some metrics (notably density and cost per bit), but everything else about them is poised to get worse. This makes the future of SSDs cloudy: While the growing capacity of SSDs and high IOP rates will make them attractive in many applications, the reduction in performance that is necessary to increase capacity while keeping costs in check may make it difficult for SSDs to scale as a viable technology for some applications.

References

[1] Open nand flash interface specification 3.0. http://onfi.org/specifications/.

[2] International technology roadmap for semiconductors: Emerging research devices, 2010.

[3] DENALI. http://www.denali.com/wordpress/index.php/dmr/2009/07/16/nand-forward-prices-rate-of-decline-will.

[4] GRUPP, L. M., CAULFIELD, A. M., COBURN, J., SWANSON, S., YAAKOBI, E., SIEGEL, P. H., AND WOLF, J. K. Characterizing flash memory: anomalies, observations, and applications. In *MICRO 42: Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture* (New York, NY, USA, 2009), ACM, pp. 24–33.

[5] JUNG, T.-S., CHOI, Y.-J., SUH, K.-D., SUH, B.-H., KIM, J.-K., LIM, Y.-H., KOH, Y.-N., PARK, J.-W., LEE, K.-J., PARK, J.-H., PARK, K.-T., KIM, J.-R., LEE, J.-H., AND LIM, H.-K. A 3.3 v 128 mb multi-level nand flash memory for mass storage applications. In *Solid-State Circuits Conference, 1996. Digest*

of Technical Papers. 42nd ISSCC., 1996 IEEE International (feb 1996), pp. 32 –33, 412.

[6] JUNG, T.-S., CHOI, Y.-J., SUH, K.-D., SUH, B.-H., KIM, J.-K., LIM, Y.-H., KOH, Y.-N., PARK, J.-W., LEE, K.-J., PARK, J.-H., PARK, K.-T., KIM, J.-R., YI, J.-H., AND LIM, H.-K. A 117-mm2 3.3-v only 128-mb multilevel nand flash memory for mass storage applications. *Solid-State Circuits, IEEE Journal of 31*, 11 (nov 1996), 1575 –1583.

[7] MAROTTA, G. E. A. A 3bit/cell 32gb nand flash memory at 34nm with 6mb/s program throughput and with dynamic 2b/cell blocks configuration mode for a program throughput increase up to 13mb/s. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International* (feb. 2010), pp. 444 –445.

[8] PARK, K.-T., KANG, M., KIM, D., HWANG, S.-W., CHOI, B. Y., LEE, Y.-T., KIM, C., AND KIM, K. A zeroing cell-to-cell interference page architecture with temporary lsb storing and parallel msb program scheme for mlc nand flash memories. *Solid-State Circuits, IEEE Journal of 43*, 4 (april 2008), 919 – 928.

[9] RAFFO, D. Fusionio builds ssd bridge between slc,mlc, july 2009.

[10] SUH, K.-D., SUH, B.-H., LIM, Y.-H., KIM, J.-K., CHOI, Y.-J., KOH, Y.-N., LEE, S.-S., KWON, S.-C., CHOI, B.-S., YUM, J.-S., CHOI, J.-H., KIM, J.-R., AND LIM, H.-K. A 3.3 v 32 mb nand flash memory with incremental step pulse programming scheme. *Solid-State Circuits, IEEE Journal of 30*, 11 (nov 1995), 1149 –1156.

[11] TAKEUCHI, K., TANAKA, T., AND TANZAWA, T. A multipage cell architecture for high-speed programming multilevel nand flash memories. *Solid-State Circuits, IEEE Journal of 33*, 8 (aug 1998), 1228 –1238.

[12] TRINH, C. E. A. A 5.6mb/s 64gb 4b/cell nand flash memory in 43nm cmos. In *Solid-State Circuits Conference - Digest of Technical Papers, 2009. ISSCC 2009. IEEE International* (feb. 2009), pp. 246 –247,247a.