SSD Characterization: From Energy Consumption's Perspective

Balgeun Yoo Youjip Won
Seokhei Cho Sooyong Kang
Hanyang University, Korea

Jongmoo Choi
Dankook University, Korea

Sungroh Yoon
Korea University, Korea

Abstract

In this work, we perform μsec time scale analysis on energy consumption behavior of the SSD Write operation and exploit this information to extract key technical characteristics of SSD internals: channel utilization policy, page allocation strategy, cluster size, channel switch delay, way switch delay, etc. We found that some SSDs adopt a multi-page *cluster* as a write unit instead of a *page*. We found that SSDs adopt significantly different ways of exploiting channel-level and way-level parallelism to maximize write throughput, which governs the peak current consumption. The X25M(Intel) emphasizes the performance aspect of SSDs and linearly increases the channel parallelism as the IO size increases. The MXP(Samsung) puts more emphasis on energy consumption aspect of SSD and controls the degree of parallelism to reduce the peak current consumption. Cluster size of the X25M and the MXP correspond to one and eight pages, respectively. The current consumed when writing a page to NAND flash varies significantly depending on the NAND model(17 mA- 35 mA).

1 Introduction

NAND flash based Solid State Drives(SSDs) promise to address the technical issues which legacy HDD based storage devices have suffered from: energy consumption, acoustics, bandwidth, IOPS, etc. State of the art SSD controllers adopt sophisticated schemes, (e.g., Flash Translation Layer, wear-leveling, garbage collection, storage signal processing, over-provisioning, compression, deduplication) to address the issues which modern NAND flash devices face: endurance, retention and page program time. To enhance the IO performance, state of art SSDs adopt a multi-channel, multi-way architecture with a large DRAM buffer. To fully exploit the potential of a storage device, it is mandatory to a acquire a comprehensive understanding of its behavior (e.g., how data is written to the storage media). The traditional way of characterizing internals of a hard disk is via examining IO latency [1]. In characterizing SSDs, we explore an entirely different method. We analyze the energy consumption behavior of the NAND flash program operation on a very fine time scale($10\mu sec$).

A number of previous works characterize of SSDs from an energy consumption perspective. Shin et al. [7] and Seong et al. [6] applied different workloads(random, sequential, etc.) to SSDs and measured the power consumption. Their work aimed to identify the relationship between workload characteristics and aggregate power consumption. Grupp et al. [2] examined the average power consumption(W) and energy(J) of SSDs across different workloads for eleven SSD models. Seo et al. [5] did a similar study, but they also varied the file systems in their measurement study. Mohan et al. [4] developed a detailed power model for the NAND flash chip itself. These works mostly focused on identifying aggregate energy consumption behavior for a given set of workloads. None of these works attempted to extract internal details of SSD behavior.

In this work, we develop a methodology to extract key technical characteristics of SSDs from an architectural aspect and determine the way in which the SSD controller utilizes its NAND flash devices. The key technical characteristics include page allocation strategy, cluster size(the size of minimum IO unit to/from NAND flash), channel switch delay, way switch delay, peak current consumption, and NAND flash programming current. None of this information is publicly available.

The conclusion of this work is quite open ended but puts forth an important thesis. The SSD vendors improve the IO performance by increasing the number of channels and the number of ways in an IO operation. There exist performance vs. energy consumption trade-offs in SSDs. The energy consumption aspect of SSDs should be emphasized equally with the performance aspects. We propose a *Power budget*(Peak current consumption of SSD) as an index to characterize SSDs. The *Power budget* can

also be used as a configuration parameter to govern the parallelism degree of an SSD.

2 SSD Organization and Energy Consumption

2.1 SSD Organization

The main components of SSDs are flash chips, the SSD controller, DRAM, and host interface, along with some others. SSD controller consists of a processor, flash controller (one for each channel), buffer (DRAM) controller, ECC/CRC module, internal memory (SRAM), and hardware accelerator for performance. Fig. 1 shows the conceptual architecture of an SSD.

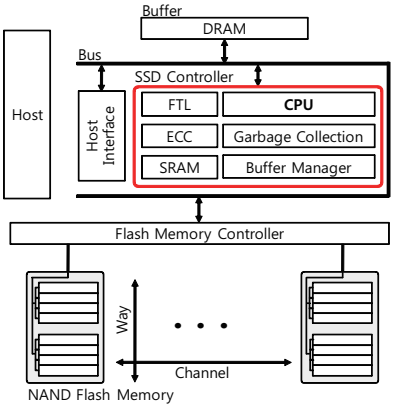


Figure 1: SSD Organization

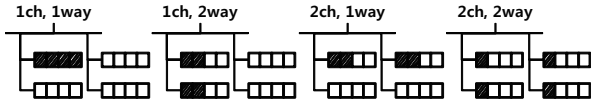


Figure 2: Page Allocation Strategies

A *Write* operation consists of sending a command to the Command register(C), sending data to the Data register(D) and program(P). To parallelize the I/O operations on multiple NAND flash devices, flash memory chips are organized into multiple channels and ways. Each channel has its own data register and command register, and the flash chips in different channels can be accessed in parallel. When we access a number of chips in the same channel, the access request is served in a time interleaved manner.

When an SSD has a multi-channel and multi-way architecture, the page allocation strategy becomes rather complicated and becomes the crux of the SSD performance. A number of issues, including write performance, cell wear-leveling, and peak current consumption, are interwined in the page allocation algorithm. Fig. 2 illustrates an SSD with two channels and two NAND flash chips in a channel. It illustrates four ways to write four pages. On one end of the spectrum, the SSD can write the pages to a single flash memory chip. At the other extreme, the SSD controller distributes four pages across four flash chips in two channels. Among the four page allocation methods in Fig. 2, distributing the pages into four chips yields the best performance but will incur the highest peak current.

2.2 Energy Consumption in a Multi-channel and Multi-way Write

The diagrams in Fig. 3 illustrate the relationship between the degree of parallelism and the energy consumption in writing three pages. We schematically represent the current of only the Program operation. In Fig. 3(a), three pages are written to a single chip. The peak current of the write operation is the lowest among the three and the total duration of program is the greatest. In Fig. 3(b), pages are written to three chips attached to the same channel. We assume that the Command and Data registers become available as soon as the NAND chip starts programming. The time to transfer the command and data to the Command and Data registers constitutes the *way switch* delay. In Fig. 3(b), the maximum degree of parallelism is two. Fig. 3(c) illustrates the situation where each page is written to a different channel. Due to architectural limitations (e.g., bus width), it is not practically possible to transfer the command and data to three channels precisely at the same time. This is because *channel switch* delay exists. In Fig. 3(c), three NAND flash chips are programmed simultaneously. The total programming duration is the shortest, but the peak current is also the highest.

3 Measurement Methodology

We measured the energy consumption behavior of a total of four SSDs: Forte+ (Hanamicron), X25M (Intel), MXP (Samsung), and Vertex (OCZ), only two of which are presented in Section 4 due to space limitations. Measurement hardware consists of an oscilloscope(Model: Tektronix TDS3032), a high resolution current probe(Model: Tektronix TCP202), a host system and the target SSD. We attached the current probe to the power line(V_{dd}) of the SSD. The host system was loaded with Linux 2.6. We opened the SSD as a *raw* device to minimize the measurement noise caused by the filesystem. The program time of SLC and MLC NAND flash is around 200 μsec and 900 μsec, respectively. To precisely measure the programming behavior of the SSD, it is critical that the sampling granularity is finer than the NAND flash programming time. In this study, we set the sampling interval to 10μsec. For accuracy of measurement, the DRAM buffer(write buffer) was turned off. To avoid garbage collection during our measurement, we factory-reset the SSD before each measurement round.

If write operations of different IO sizes yield identical energy consumption behavior, the number of chips

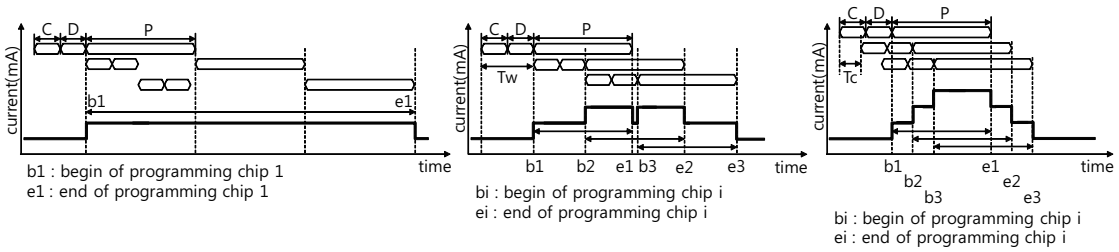


Figure 3: Page Allocation Strategy vs. Current Consumption

involved in these operations are the same. This phenomenon is used to identify the *cluster* size. Fig. 4(a) illustrates the energy consumption of 4, 8, 16, and 32 KByte writes. Energy consumption for 4, 8 and 16 KByte writes is identical. On the other hand, the peak current of a 32 KByte write is larger than the peak current of 16 KByte write by approximately 40 mA. In this case, we infer that the cluster size is 16 KByte. Observing the peak current, we can determine the number of flash chips involved in the IO operation. Fig. 4(b) illustrates the energy consumption behavior for 16 KByte and 32 KByte writes. As we double the IO size from 16 KByte to 32 KByte, peak current increases. We conjecture that the degree of parallelism has increased. Without any interleaving, when we double the IO size, the programming duration doubled. Fig. 4(c) illustrates the energy consumption behavior for 256 KByte and 512 KByte writes(Intel X25M). For these two IO sizes, the peak currents are identical(500 mA), but the programming duration is doubled.

4 Case Study

4.1 Case Study 1: Intel X25M

We vary the IO size from 4 KByte to 160 KByte(Δ = 4 KByte) and measure the energy consumption behavior of the write operation for the Intel X25M. Fig. 5(a) and Fig. 5(b) illustrate energy consumption behavior of IOs from 4KByte to 80 KByte and from 80 KByte to 160 KByte, respectively. In Fig. 5(a), we can clearly see that each increment of 4 KByte in IO size causes an increase in the peak current(approximately 17 mA). From this, we can conjecture that each 4 KByte is written to a different NAND flash device.

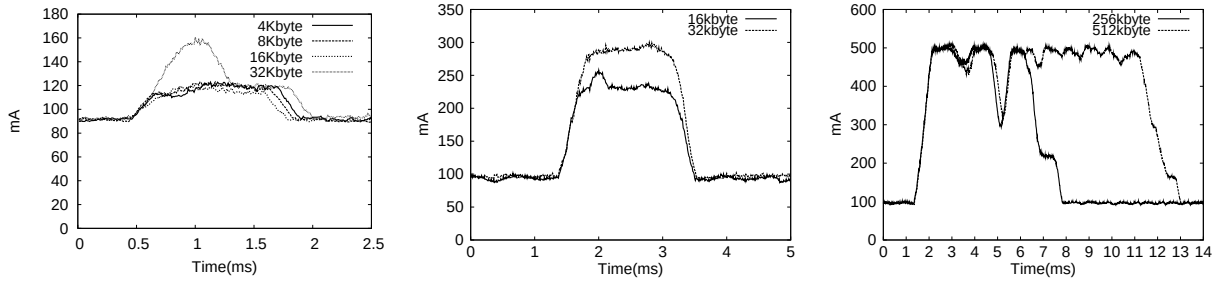
The next step is to identify the interleaving method for a multi-page write. In an 80 KByte write, current consumption reaches the peak at around 1.1 $msec$ on the X-axis. Programming one more page increases the total programming time by 30 μsec (i.e. 600 $\mu sec \div$ 20). Given that in most SSDs the channel speed is 40 MByte/sec, it takes 100 μsec to transfer 4KByte to the data register. Therefore, the way switch delay cannot be shorter than 100 μsec. We conjecture that the X25M uses simple *channel* interleaving to allocate pages. Fig. 5(b) illustrates the energy consumption behavior for write IO from 80 KByte(20 pages) to 160KByte(40 pages) write. Since the X25M has ten channels and two chips per channel, a maximum of 20 pages(80 KByte) can be programmed in parallel. However, writing 21 pages(84 KByte) incurs a significant increase in programming latency. Let us number the pages from P_0 to P_{20}. Based on our conjecture, P_0, P_{10}, and P_{20} will be allocated to the same channel and P_0 and P_{20} are allocated to the same chip. A Program operation of P_{20} can start only after the SSD finishes programming P_0. The current consumption observed in Fig. 5(b) precisely matches our expectation. We conclude that the X25M uses simple channel interleaving for page allocation and a cluster size of one page.

We can precisely identify the amount of current used in page programming. Peak current increases by 17 mA as we increase the write size by 4 KByte. A single NAND flash chip consumes 17 mA to program a page. The NAND flash chips in the X25M bear relatively good energy characteristics compared to those used in the MXP(25 - 35 mA to program a page).

4.2 Case Study 2: SAMSUNG MXP

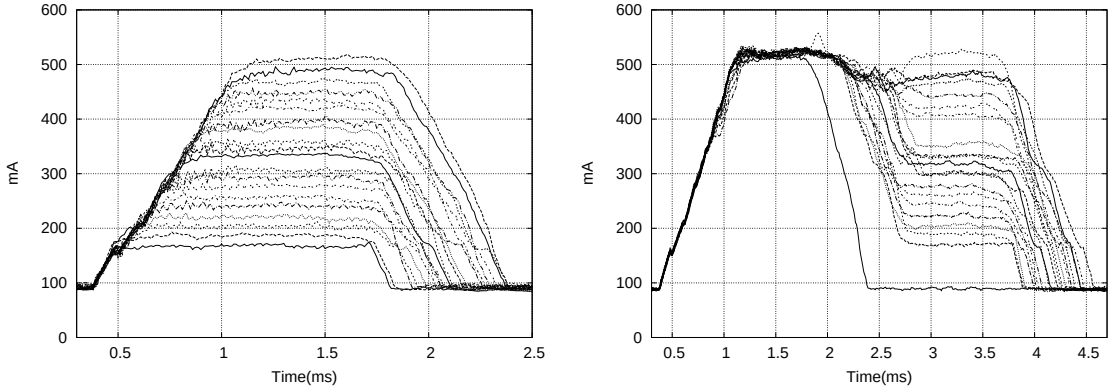
We measured the current consumption behavior for the MXP(Samsung). We varried IO size from 4 KByte to 96 KByte. TheMXP has eight channels. There are two packages for each channel and each package has four chips. A total of 8 chips are attached to one channel. According to our measurement, write operations from 4 KByte to 96 KByte can be categorized into three categories with respect to energy consumption behavior: from 4 KByte to 32 KByte(Fig. 6(a)), from 36 KByte to 64 KByte(Fig. 6(b)) and from 68KByte to 96 KByte(Fig. 6(c)). Write operations in the same category yield similar(if not identical) energy consumption behavior. Given this, we infer that the cluster size is 8 pages in the MXP.

As we increase the cluster size, we can observe the increase in the peak current, but the entire programming duration does not increase significantly. From this, we infer that MXP increases the degree of parallelism in units of a cluster.



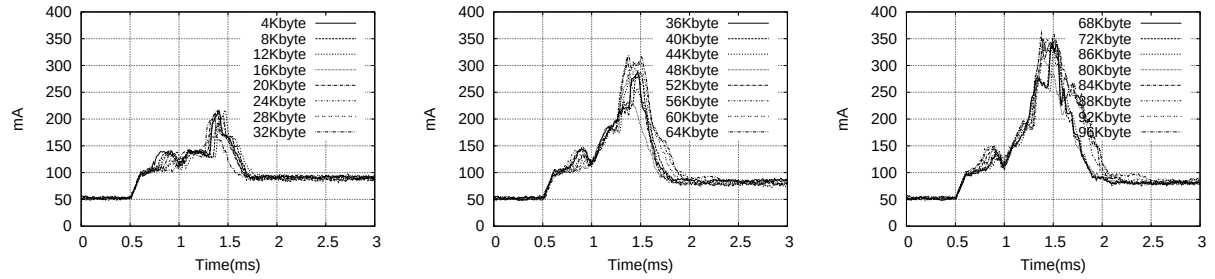
(a) Cluster Size(Hanamicro Forte+) (b) Increase in degree of parallelism: 16 KB vs. 32 KB(Intel X25M) (c) Single Chip IO: 256 KB vs 512 KB(Intel X25M)

Figure 4: Identifying Cluster Size, Parallelism Degree and Operation on the Same Chip



(a) Data Size: from 4KB to 80KB (b) Data Size: from 80KB to 160KB IO

Figure 5: Energy Consumption of Write Operation: Intel X25M under varying IO sizes(10 channels, 2 chips/channel, 4Gbyte/chip, Δ = 4KByte)



(a) from 4KB to 32 KB (b) from 36 KB - 64 KB (c) from 68KB - 96 KB

Figure 6: Energy Consumption behavior of Write Operation under varying IO sizes: Samsung MXP(8 Channel, two packages/chip, four chips/packages, 2GByte/chip, Δ = 4KByte)

5 Power Budget

Excessive peak power consumption of a multi-channel SSD can cause problems for the SSD device itself as well as for the host. If a multi-channel, SSD programs too many NAND chips in parallel, excessive peak current can cause supply voltage drop, ground bounce, signal noise, black-out, etc. and can lead to unreliable SSD operation[3]. The current for programming 16 flash chips in parallel can reach as high as the current consumption of a modern enterprise class HDD. This issue will compound when we configure RAID system with SSD.

We propose a metric called *Power Budget*, which specifies the maximum tolerable peak current for SSD operation. It can be defined by the host or by the SSD controller. It enables the host to specify the maximum energy consumption for the device in very flexible manner. Based upon this "Power Budget", the SSD can adapt its internal mechanisms (e.g., channel allocation and the maximum number of chips programmed in parallel) to properly incorporate the performance/power trade-offs. Fig. 7 schematically illustrates the power-performance trade-offs. The X and Y axes denote the time and the current consumption, respectively. In Fig. 7, The Power Budget limits the maximum tolerable peak current, and Performance Bound specifies maximum tolerable latency. Assuming an 8 channel with 8-ways/channel SSD, feasible page allocation strategies are 2×4(four channel, 2-way/channel) or 4×2 (two channel, 4-way/channel).

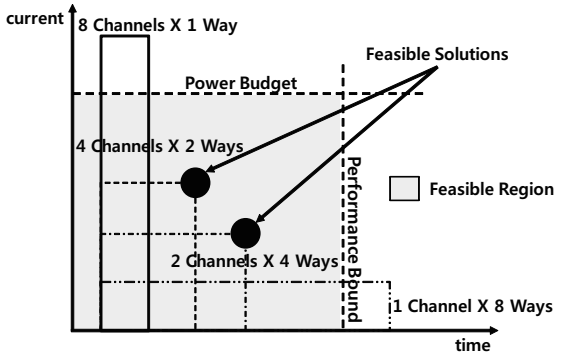


Figure 7: Power Budget and Feasible Page Allocations

6 Conclusion

SSD vendors do not publicize the internal details of their products. As more channels are used for sequential data striping, not only does the parallelism of I/O increase but so does the instantaneous power consumption. As more flash memory chips, in a channel, are used for data interleaving, the power consumption for an I/O operation sustains for a longer time. Page/block allocation policy also affects the energy consumption behavior of an SSD.

In this study, we developed an elaborate methodology to identify the internal details of SSDs and propose the *Power Budget* metric, which can be used for SSD characteristics, design criterion for power consumption constraint-based page/block allocation policy, or as a configuration parameter which governs the degree of parallelism dynamically. Using our forensic method, we successfully verified the page/block allocation policies of commodity SSD.

7 Acknowledgements

We thank the anonymous reviewers and John Strunk(our shepherd) for their valuable feedback on our work. We would also like to thank Jaehyuk Cha for his insightful comment. This work is sponsored by the IT R&D Program(large scale hyper-MLC SSD development), No. 10035202 of MKE/KEIT, Korea.

References

[1] GIM, J., AND WON, Y. Extract and infer quickly: Obtaining sector geometry in modern hard disk drives. *ACM Transactions on Storage 6*, 2 (July 2010).

[2] GRUPP, L., CAULFIELD, A., COBURN, J., SWANSON, S., YAAKOBI, E., SIEGEL, P., AND WOLF, J. Characterizing flash memory: Anomalies, observations, and applications. In *Proc. of MICRO 2009* (New York, USA, Dec. 2009), pp. 24–33.

[3] HONG, G. Analysis of peak current consumption for large-scale, parallel flash memory. In *Workshop for Operating System Support for Non-Volatile RAM(NVRAMOS 2011 Spring) (Jeju, Korea, April 2011).*

[4] MOHAN, V., GURUMURTHI, S., AND STAN, M. Flashpower: A detailed power model for nand flash memory. In *Proc. of Date '10* (Dresden, Germany, March 2010), pp. 502–507.

[5] SEO, E., PARK, S. Y., AND URGAONKAR, B. Empirical analysis on energy efficiency of flash-based SSDs. In *Proc. of HotPower'08* (San Diego, CA, USA, Dec. 2008).

[6] SEONG, Y., NAM, E., YOON, J., KIM, H., CHOI, J., LEE, S., BAE, Y., LEE, J., CHO, Y., AND MIN, S. Hydra: A Block-Mapped Parallel Flash Memory Solid-State Disk Architecture. *IEEE Transactions on Computers 59*, 7 (2010), 905–921.

[7] SHIN, S., AND SHIN, D. Power Analysis for Flash Memory SSD. In *Work- shop for Operating System Support for Non-Volatile RAM(NVRAMOS 2010 Spring) (Jeju, Korea, April 2010).*