Dual-Port Pre-emptive Cache Buffer Flushing for High Performance NAND Flash-based Storage Devices

Sang-Hoon Park
School of Electrical and Electronic Engineering
Yonsei University
Seoul, Republic of Korea
Email: soskhong@dtl.yonsei.ac.kr

Eui-Young Chung
School of Electrical and Electronic Engineering
Yonsei University
Seoul, Republic of Korea
Email: eychung@yonsei.ac.kr

***Abstract*—The cache buffer (CB) is now an essential part for high performance NAND flash-based storage devices and its capacity has been continuously growing. In this paper, a pre-emptive flushing method using an additional port of CB is proposed. It utilizes the additional port to write back dirty data while the other port handles host requests so that future requests will not be affected by time-consuming flushing operations. The experimental results indicate that the proposed method shows 44% and 17% shorter latency than single-port CB of same and double capacity.**

I. Introduction

The cache buffer (CB) has become essential for modern high performance NAND flash-based storage devices (NFSDs) to compromise latency drawbacks and unique characteristics of NAND flash memories (NFMs). The capacity of CBs is also growing and most of modern commercial NFSDs are equipped with 1-2MB DRAM as CB for 1GB of NFMs. The increasing capacity shows that the capacity is not sufficient and the insufficiency causes frequent flushing operations, which write back dirty data in CB to NFMs.

In this paper, therefore, we propose a flushing method that minimizes performance degradation. More specifically, the method is based on the CB with one additional port (dual-port DRAM) instead of increasing its capacity that enables an NFSD to flush data to NFMs in a pre-emptive way. It writes back data by exploiting the additional port of CB before it is demanded to hide flushing latency from the host.

II. Dual-port pre-emptive flushing

Fig. 1 shows the exploitation of the dual-port CB by the proposed method. HIT/MISS stand for the request hit/miss at CB and MISS invokes eviction depicted as EV, since MISS requires vacated space of CB for incoming data. With dual-port, the eviction can be done before it is demanded for future requests without overhead. For the proposed method, two things should be modified from the ordinary NFSD, which are depicted as gray boxes in Fig. 2.

First, it employs a dual-port DRAM as CB. The additional port (P_{ADD}) is utilized to flush data to NFMs in pre-emptive way. We assume that both ports can access any banks in DRAM but one bank cannot be accessed by two ports

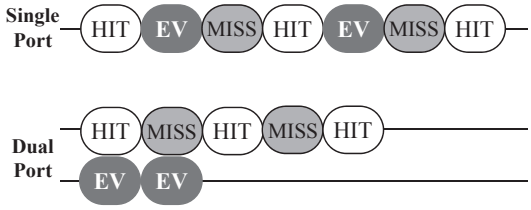

Fig. 1. Timing diagram of single and dual-port CB

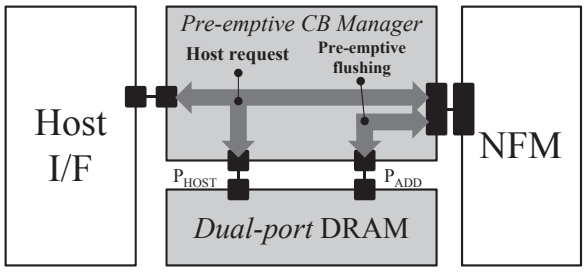

Fig. 2. Block diagram of NFSD with proposed method

simultaneously, like the dual-port DRAM in [1] to prevent data corruption. The hardware overhead for P_{ADD} is not significant, since the major portion of a DRAM chip is taken by memory cells whereas peripheral logics required by the additional port occupy under 10% of the chip [2]. In other words, adding P_{ADD} to DRAM is much cheaper than doubling the capacity.

Second, pre-emptive CB manager (PCBM) utilizes two ports of DRAM with fixed roles and priorities. The first port (P_{HOST}) is dedicated to the host requests and used in the same way the only port of ordinary CB is used. PCBM always gives higher priority to P_{HOST} than P_{ADD} when target banks of their accesses are same. On the other hand, it activates pre-emptive flushing using P_{ADD} only when three conditions are met and the conditions are 1) the current host request is hit at CB, 2) no other pre-emptive flushing is being executed and 3) dirty data with sufficiently high eviction priority according to CB management policy exist.

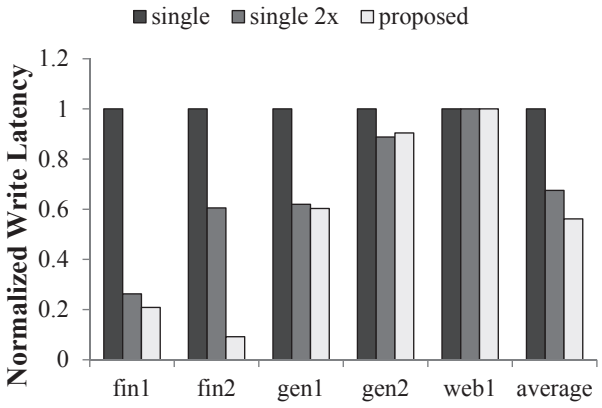

Fig. 3. Normalized write latency

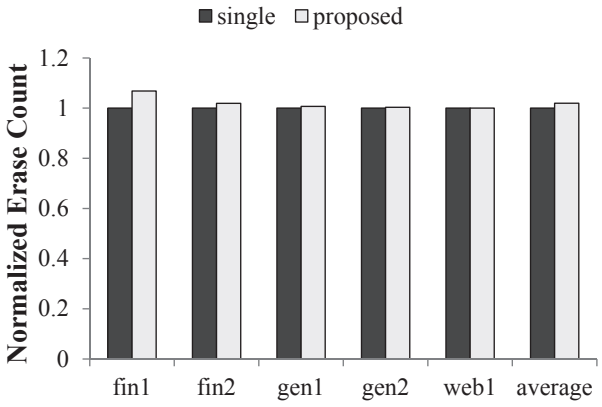

Fig. 4. Normalized erase count of NFMs

The first and second conditions prevent degradation in performance due to interference between pre-emptive flushing and host requests that need NFMs. The third one controls aggressiveness of flushing that affects both performance and life-time of NFMs. Also, it is designed to co-work with various CB management policies. In this paper, we assume that PCBM checks data dirtiness in the descending order of eviction priority until the amount of checked data reaches 5% of the total CB capacity.

III. Evaluation

To evaluate the proposed method, we implement a SystemC-based trace-driven NFSD simulator. It well describes hardware parallelism essential to simulate multiple ports and modules. Input traces are obtain from [3] (fin1, fin2, web1) and from our daily PC usage (gen1, gen2). The proposed method is compared with the single-port CB of the same capacity (*single*), which is 16MB, and doubled capacity (*single2x*). The page-level least recently used policy and page-level flash translation layer [4] are used for CB and NFM management. NFM and DRAM timing parameters are from [5] and [6]. The capacity of NFM is set as 16GB.

As Fig. 3 depicted, the proposed method shows 44% and 17% shorter write latency than *single* and *single2x* on average. It even outperforms *single2x* except for gen2 and web1. It implies that, in spite of more chances to exploit locality by larger capacity, the efficient flushing method should be adopted since on-demand flushing degrades performance eventually.

Fig. 4 shows how many additional erases are required when pre-emptive flushing, which is speculative method, is used. On average, it induces 1.9% (7% at most) more erase operations than *single* which may be considered negligible compared to performance improvement.

Additionally, the leakage power consumption that is important for mobile devices with DRAM is analyzed using CACTI [7] and the proposed method requires 49% lower power that *single2x* due to its smaller cell area. The power saving may also compensate the increased dynamic power consumption incurred by additional erase operations of NFMs.

IV. Conclusion

This paper proposes a pre-emptive flushing method using dual-port DRAM for CB equipped by high-performance NFSDs. The proposed method flushes dirty data from CB to NFMs by exploiting the additional port of CB to minimize on-demand flushing. Using a trade-driven simulator, we show that the proposed method even outperforms a single-port CB with doubled capacity. As a future work, the proposed method will be modified to utilize idle periods of NFSDs and parameterized for better flexibility.

Acknowledgment

This work (Grant No. C0146555) was supported by Business for Cooperative R&D between industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2013, by IDEC (IC Design Education Center) and by SK-Hynix Semiconductor Inc.

References

[1] D. Kim, K. Bang, S.-H. Ha, S. Yoon, and E.-Y. Chung, "Architecture exploration of high-performance pcs with a solid-state disk," *Computers, IEEE Transactions on*, vol. 59, no. 7, pp. 878–890, 2010.

[2] Y.-C. Bae, J.-Y. Park, S. J. Rhee, S. B. Ko, Y. Jeong, K.-S. Noh, Y. Son, J. Youn, Y. Chu, H. Cho *et al.*, "A 1.2 v 30nm 1.6 gb/s/pin 4gb lpddr3 sdram with input skew calibration and enhanced control scheme," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*. IEEE, 2012, pp. 44–46.

[3] *UMass Trace Repository*, 2007. [Online]. Available: http://traces.cs.umass.edu/

[4] H.-j. Kim and S.-g. Lee, "A new flash memory management for flash storage system," in *Computer Software and Applications Conference, 1999. COMPSAC'99. Proceedings. The Twenty-Third Annual International*. IEEE, 1999, pp. 284–289.

[5] Micron Technology Inc., *64Gb, 128Gb, 256Gb, 512Gb Asynchronous/Synchronous NAND Features*. Micron Technology Inc., 2009.

[6] JEDEC Solid State Technology Association, "Jedec standard: Ddr2 sdram specification," *JESD79-2A (Revision of JESD79-2)*, Jan. 2004.

[7] S. Thoziyoor, N. Muralimanohar, J. H. Ahn, and N. P. Jouppi, "Cacti 5.1," *HP Laboratories, April*, vol. 2, 2008.