반도체디스플레이기술학회지 제20권 제4호(2021년 12월)
Journal of the Semiconductor & Display Technology, Vol. 20, No. 4. December 2021.

Low Power Scheme Using Bypassing Technique for Hybrid Cache Architecture

Juhee Choi[*†]

[*†]Dept. of Smart Information Communication Engineering, Sangmyung University

ABSTRACT

Cache bypassing schemes have been studied to remove unnecessary updating the data in cache blocks. Among them, a statistics-based cache bypassing method for asymmetric-access caches is one of the most efficient approach for non-voliatile memories and shows the lowest cache access latency. However, it is proposed under the condition of the normal cache system, so further study is required for the hybrid cache architecture. This paper proposes a novel cache bypassing scheme, called hybrid bypassing block selector. In the proposal, the new model is established considering the SRAM region and the non-volatile memory region separately. Based on the model, hybrid bypassing decision block is implemented. Experiments show that the hybrid bypassing decision block saves overall energy consumption by 21.5%.

Key Words : Bypass Algorithm, Non-Volatile Memories, Hybrid Cache Architecture, Low Power

1. INTRODUCTION

NVM, which static power is extremely low, has been researched as an alternative to SRAM [1,2,3]. As the process technology evolves, the portion of the static power consumption over the total power consumption is tremendously increased in the nano-scale circuits [4]. Although NVM is considered as a solution for leakage problems, there are two shortcomings in terms of write operation for the LCC to be adopted directly. The write energy consumption of NVM is higher than that of SRAM and it has longer time to finish the write operations.

A bypassing scheme for cache system has been suggested to alleviate the data contamination caused by unnecessary data allocation. Some cache blocks are deactivated when it is written into the cache immediately. It implies more useful cache blocks are evicted due to these dead blocks. Therefore, the capacity of the cache is wasted, which is called cache pollution. To overcome the problem, cache bypassing schemes have been studied to decide these dead blocks and avoid storing them in the cache [5,6,7]. Even though several works have been carried out, they ignore the difference between the write operations and the read operations because their works were assumed the SRAM-based cache. Therefore, the novel schemes are devised to optimize the dynamic energy consumption for the NVM-based LLC [8,9]. Even though SBAC is one of the most outstanding bypassing techniques for NVMs, it has out of consideration of HCA environment [8]. To alleviate this limitation, this paper proposes a novel scheme, which is called hybrid bypassing block selector (HBS), to enable cache bypassing for the HCA. This method applies different bypassing schemes for SRAM and NVM by keeping track of the write counts of SRAM and NVM, respectively. To do this, the extended bypassing decision selector contains the extra counters and logics. The experimental results show that the HBS improves the power efficiency by 21.5%.

[†]E-mail: jhplus@smu.ac.kr

2. RELATED WORKS

2.1 Cache Bypassing

There are several aspects of cache management such as set associativity, write-through/write-back, allocation policy, and replacement policy. Among them, for a good replacement policy, the extensive studies have been proposed by avoiding unnecessary cache block placing, which is called cache bypassing [5,6,7,10]. It stores the cache block from the upper level caches only if there is a benefit for the current level of cache.

Dybdahl et. al designed a method to decide the cache bypassing based on the program counter of the memory instruction [5]. Insertion and bypassing scheme for the exclusive cache hierarchy is introduced by Gaur et. al [6]. Access counters are utilized to predict by a table indexed by a hash function for the instruction address [7]. Another bypassing algorithm predicts the utilization the access counter [10].

2.2 Spin Torque Transfer Random Access Memory

STT-RAM, as one of the most promising solutions for next generation LLC [11], has significantly higher density and extremely low leakage power compared with SRAM. It is natural consequence that STT-RAM is being an alternative to SRAM. An STT-RAM cell consists of one transistor and one magnetic tunnel junction (MTJ) as shown in Fig. 1; 1T1J using for keeping a bit information. An MTJ has two ferromagnetic layers. There are a pinned layer and a free layer. Also, there is a tunnel barrier layer made of MgO between two layers. The magnetization direction of the free layer can be changed, while the magnetization direction of the pinned layer is persistent. When an MTJ cell has the anti-parallel magnetization directions in two layers, it has the high resistance so that it indicates '1'. For the case of that the two layers have the parallel directions each other in the MTJ cell, the cell becomes '0' due to the low resistance of the MTJ. Since there is no electric source used to maintain the information, the static power consumption is significantly low in the LLC composed of STT-RAM. However, it requires to a long latency and high energy consumption for the write operations. They come from the fact that switching the direction of the layer in the MTJ cell requires more energy than switching the bit information in the SRAM cell.

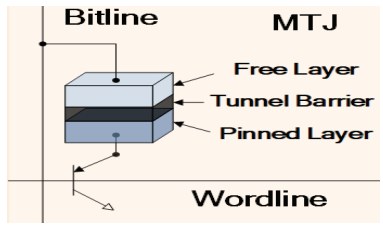

Fig. 1. An illustration of an STT-RAM cell.

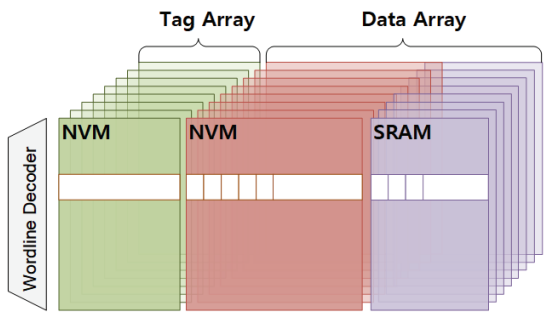

Fig. 2. Overall architecture of hybrid cache architecture. Two kinds of memory (NVM and SRAM) are combined into a single cache hierarchy.

2.3 Hybrid Cache Architecture

To overcome these problems related to write operations of NVM, many research groups have approached a novel architecture, which is called hybrid cache architecture (HCA) [12][13][14]. These schemes have merged two types of memory into a single cache system. As discussed above, the drawbacks of NVM are derived from the fundamental characteristics of NVM. In other words, the write counts of NVM is the dominant factor for the dynamic energy consumption because the write energy consumption of NVM occupies much larger portion compared with the portion of the dynamic energy consumption for the read operations. Thus, the main themes for HCA studies have been finding the method to decrease the number of write operations of the NVM area by placing the write-intensive blocks into the SRAM area.

3. HYBIRD BYPASSING BLOCK SELECTOR

Based on the discussion above, hybrid bypassing block selector (HBS) is proposed. This section provides the model

for the HBS derived from the previous work. In addition, the implementation of the proposal is given.

3.1 Model

The statistics based cache bypassing method (SBAC) is proposed with the help of a theoretical model [8]. The authors of the SBAC focused on the costs of cache bypassing. Thus, they suggested a novel decision logic for the proper bypassing based on data reuse count (DRC) probability. Note that LLC is assumed that L3 cache in this section.

Their presentative theoretical achievements are described [8] as follows:

$$\mathrm{EN_{avg}} = \sum_{i=0}^{\infty}\{P_i[R_{LLC} + W_{L2} + (i+1)\,R_{LLC}]\} \quad (1)$$

where $\mathrm{EN_{avg}}$ denotes the average access energy of the data without bypassing, $\mathrm{P_i}$ is the probability of DRC=i, $\mathrm{R_{LLC}}$ means the read energy of LLC, and $\mathrm{W_{L2}}$ is the write energy of L2 cache.

If all data is stored in the L2 cache, the cache access energy consumption is noted as $\mathrm{EB_{avg}}$ [8]:

$$\mathrm{EB_{avg}} = \mathrm{P_0}\big(R_{LLC} + R_{L2tag}\big) + \sum_{i=0}^{\infty}\{P_i[2R_{LLC} + R_{L2tag} + W_{L2} + i\,R_{L2}]\} \quad (2)$$

where $\mathrm{EB_{avg}}$ denotes the average access energy of the data with the bypassing scheme and $\mathrm{R_{L2tag}}$ means the energy of reading L2 tag.

Another important equation they invented is to calculate the bypassing depth [8]. It implies that data with DRC less than bypassing depth will be bypassed in the cache.

$$\frac{P_{d-1}}{1-\sum_{j=0}^{d-2}P_j} > \frac{R_{LLC}+R_{L2tag}-R_{L2}}{W_{L2}+R_{LLC}} \quad (3)$$

where d denotes the bypassing depth.

The bypassing control logic can make decision based on the runtime bypassing depth as follows:

$$\frac{N_{\geq d}-N_{\geq d+1}}{N_{\geq d}} > \lambda \quad (4)$$

$$\frac{N_{\geq d-1}-N_{\geq d}}{N_{\geq d-1}} < \lambda \quad (5)$$

where λ means the bypassing feature of the system. When Equation (4) is satisfied, the bypass depth is added by one, while the bypass depth is decreased by one when Equation (5) is met. Please, refer to [8] for the detail information.

The new model for HBS is defined by modifying the existing model. The first approach starts by dividing the energy consumption of the total access for the LLC into the energy consumption of the SRAM area and the NVM area as follows:

$$\begin{aligned}\mathrm{EHN_{avg}} = {} & \sum_{i=0}^{\infty}\{PS_i[SR_{LLC} + (i+1)\,SR_{LLC}]\} \\ & + \sum_{i=0}^{\infty}\{PN_i[NR_{LLC} + (i+1)\,NR_{LLC}]\} \\ & + \sum_{i=0}^{\infty}\{(PS_i + PN_i)W_{L2}\}\end{aligned} \quad (6)$$

where $\mathrm{EHN_{avg}}$ denotes the average access energy of the data for HCA without bypassing, $\mathrm{PS_i}$ is the probability of DRC=i for the SRAM area, $\mathrm{PN_i}$ is the probability of DRC=I for the NVM area, $\mathrm{SR_{LLC}}$ means the read energy of the SRAM area, and $\mathrm{NR_{LLC}}$ means the read energy of the NVM area.

Next, the average energy of the data with the bypassing scheme ($\mathrm{EHB_{avg}}$) is revised.

$$\begin{aligned}\mathrm{EHB_{avg}} = {} & (PS_0 + PN_0)\big(SR_{LLC} + NR_{LLC} + R_{L2tag}\big) \\ & + \sum_{i=0}^{\infty}\{2PS_iSR_{LLC} + 2PN_iNR_{LLC}\} \\ & + \sum_{i=0}^{\infty}\{(PS_i + PN_i)[R_{L2tag} + W_{L2} + i\,R_{L2}]\}\end{aligned} \quad (7)$$

Note that the energy consumption variables for the L1 cache or the L2 cache is not divided since they consist of SRAM. On the other hand, the LLC is composed of the HCA unlike the Equation (2)(3).

Finally, the calculation for the bypassing depth for the HCA is obtained as follows:

$$\frac{PS_{d-1} + PN_{d-1}}{1 - \sum_{j=0}^{d-2}(PS_j + PN_j)} > \frac{SR_{LLC}+NR_{LLC}+R_{L2tag}-R_{L2}}{W_{L2}+SR_{LLC}+NR_{LLC}} \quad (8)$$

3.2 Implementation

Fig. 3 illustrates the overall architecture of the HBS. The main components of the proposal are the hybrid bypass decision block (HBDB), the counters for SRAM and NVM, and the selection logics for arbitration. The HBDB is extended from the bypass decision block in the SBAC to consider the characteristics of the SRAM and the NVM, respectively. To aim it, the counters are separated into the

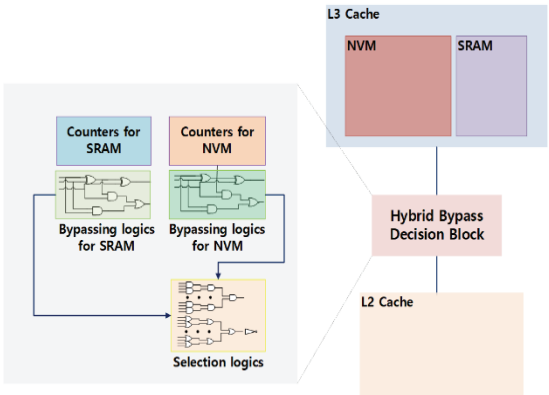

Fig. 3. Overall architecture of hybrid bypass block selector.

counters for SRAM and the counters for NVM. Also, the bypassing decision logics are modified for both types of memory. For the final decision, extra selection logics are inserted.

Assuming a scenario that the bypassing depth is 2 and all DRC bits are initialized to zero. When a request arrives to the L1 cache, the DRC bit of the cache block in the LLC is increased by one. The incoming cache block is bypassed to the L1 cache without storing the data in the LLC on top of bypassing the L2 cache. In contrast to the previous work, only the tag array is updated instead of both updating the tag array and the data array. If the cache block is accessed again, the corresponding DRC bit becomes 2. Then, the counters in the HBDB are increased according to the type of the cache block. When the data is read for the third time, the cache block is stored to the L2. Plus, the data array in the L3 cache is finally updated.

4. PERFORMANCE EVALUATION

4.1 Simulation Environments

The gem5 simulator [15], which is one of the most widely used full-system simulator, is selected to implement the proposal. Table 1 presents the system parameters for the experiments. The capacity of the L1 instruction and data caches are 32KB and their associativity is four, respectively. The 8-way 256KB L2 cache resides between the L1 cache and the LLC. The hybrid LLC cache is a 2MB 16-way cache with 4-way SRAM. The size of a cache block is 64B. Some programs from SPEC CINT2006 and SPEC CFP2006 suite [16] are chosen as benchmarks. The NVSim provides the parameters of memories [17]. The Base in the figures in the section means the results of the SBAC [8].

Table 1. Processor configurations

Core Type	x86, out-of-order, 2GHz
I-Cache	32KB, 4-way, 64B, 2 cycles
D-Cache	32KB, 4-way, 64B, 2 cycles
L2 Cache	256KB, 8-way, 64B, 6 cycles
LLC Config.	1MB(4-way SRAM and 12-way STT-RAM)
LLC Latency	SRAM : 6 / 6 cycles STT-RAM : 6 / 23 cycles
LLC Energy	SRAM Read / Write : 0.246nJ STT-RAM Read : 0.239nJ STT-RAM Write : 1.750nJ

4.2 Simulation Results

The analysis of the performance evaluation starts by showing the change of the write counts for each application. Fig. 4 provides the normalized write counts of the proposal compared with the baseline. From our experimental results, it is proven that substantial improvements of write count reduction. On average, the reduction in the write count is about 31.6%. It is also found that the result varies depending on the program. For instance, in case of *libquan*, the write counts are reduced by 42.5%, while only a 18% reduction is shown for *sjeng*.

Fig. 5 presents the comparison of the dynamic energy consumption of the HBS with the previous work. Generally, the trend of the energy reduction is similar to the reduction in write counts. Because the energy of write operations is much higher than read operations. The average reduction

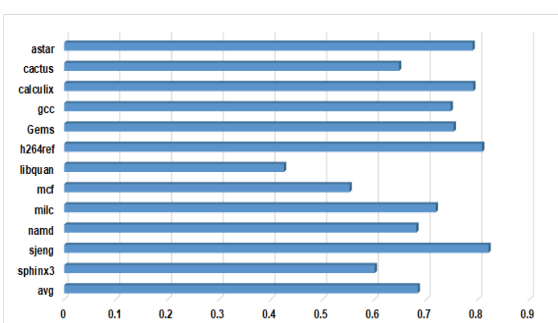

Fig. 4. Normalized write counts of the proposal compared with the baseline.

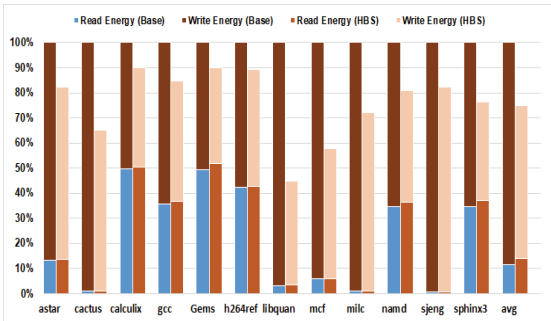

Fig. 5. Reduction ratio of the dynamic energy consumption.

ratio is 25.1%.

However, the results seem to need further investigating since the reduction in the dynamic energy consumption is not perfectly proportional to the reduction in the write counts. The normalized value is 0.80 and the reduction ratio of the dynamic energy consumption is 10.4% for *h264ref*, while the normalized value of *sjeng* is 0.82 and a 17.7% of energy savings achieved. This phenomenon is explained by the portion of the write energy consumption over the total energy consumption. To reveal this factor, the graph in Fig. 5 is divided into the read energy consumption and the write energy consumption.

To sum up, the HBS achieved a 25.1% of average energy savings by cutting down a 31.6% of the write counts.

5. CONCLUSIONS

This paper proposed a hybrid bypassing block selector (HBS) scheme to remove unnecessary write accesses in the NVM area in HCA-based LLC. Since the existing researches have no consideration of HCA, the HBS focused on reducing the write counts of the NVM region with bypassing scheme. Extra counters and logics are inserted logics for tracking the information of the SRAM area and the NVM area to determine whether which block is bypassed. To evaluate the proposal, the experiments were performed with the HCA-based LLC which consists of STT-RAM and SRAM. The results showed the proposed scheme removes the unnecessary write counts by 31.6%, which led to a 25.1% reduction in the total dynamic energy consumption.

References

1. T. Kim, O. Yang, and J. Yeon, “Design of Asynchronous Non-Volatile Memory Module Using NAND Flash Memory and PSRAM,” Journal of the Semiconductor & Display Technology, vol. 19, no. 5, pp. 118-123, 2020.
2. J. Sung, J. Jeong, and G. Lee, “Reliability Analysis by Lateral Charge Migration in Charge Trapping Layer of SONOS NAND Flash Memory Devices,” Journal of the Semiconductor & Display Technology, vol. 18, no. 4, pp. 138-142, 2019.
3. S. Yoon, and J. Nah, “Hybrid Memory Adaptor for OpenStack Swift Object Storage,” Journal of the Semiconductor & Display Technology, vol. 19, no. 3, pp. 61-67, 2020.
4. S. Rodriguez and B. Jacob. “Energy/power breakdown of pipelined nanometer caches (90nm/65nm/45nm/32nm),” In Proceedings of the 2006 international symposium on Low power electronics and design, pp. 25-30. 2006.
5. H. Dybdahl and P. Stenstrom. “Enhancing last-level cache performance by block bypassing and early miss determination,” In Asia-Pacific Conference on Advances in Computer Systems Architecture, pp. 52-66, 2006.
6. J. Gaur, M. Chaudhuri, and S. Subramoney. “Bypass and insertion algorithms for exclusive last-level caches,” ACM SIGARCH Computer Architecture News, no. 39, pp. 81-92, 2011.
7. M. Kharbutli and D. Solihin. “Counter-based cache replacement and bypassing algorithms,” Computers, IEEE Transactions on, no. 57, pp. 433-447, 2008.
8. C. Zhang, G. Sun, P. Li, T. Wang, D. Niu, and Y. Chen. “Sbac: a statistics based cache bypassing method for asymmetric-access caches,” In Proceedings of the international symposium on Low power electronics and design, pp. 345-350, 2014.
9. Xu, Y., Xu, Y., Tang, M., Zhang, L., and Lan, Y. “Asymmetry & Locality-Aware Cache Bypass and Flush for NVM-Based Unified Persistent Memory,” In 2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), pp. 168-175, 2019.
10. S. Gupta, H. Gao, and H. Zhou. “Adaptive cache bypassing for inclusive last level caches.” In Parallel & Distributed Processing (IPDPS), IEEE International Symposium on, pp. 1243-1253, 2013.
11. Z. Diao, Z. Li, S. Wang, Y. Ding, A. Panchula, E. Chen, L.-C. Wang, and Y. Huai. “Spin-transfer torque switching in

magnetic tunnel junctions and spin-transfer torque random access memory," Journal of Physics: Condensed Matter, no. 16, pp. 165209, 2007.

12. J. Li, L. Shi, C. J. Xue, C. Yang, and Y. Xu, "Exploiting set-level write non-uniformity for energy-efficient nvm-based hybrid cache," in Embedded Systems for Real-Time Multimedia (ESTIMedia), IEEE Symposium on, pp. 19-28, 2011.
13. B. Quan, T. Zhang, T. Chen, and J. Wu, "Prediction table based management policy for stt-ram and sram hybrid cache," in Computing and Convergence Technology (ICCCT), IEEE International Conference on, pp. 1092-1097, 2012.
14. Z. Wang, D. A. Jimenez, C. Xu, G. Sun, and Y. Xie, "Adaptive placement and migration policy for an stt-ram-based hybrid cache," in High Performance Computer Architecture (HPCA), IEEE International Symposium on, pp. 13-24, 2014.
15. J. Power, J. Hestness, M. S. Orr, M. D. Hill, and D. A. Wood, "gem5-gpu: A heterogeneous cpu-gpu simulator," IEEE Computer Architecture Letters, vol. 14, no. 1, pp. 34-36, 2015.
16. J. Henning, "Spec cpu2006 benchmark descriptions," SIGARCH Comput. Archit. News, vol. 34, no. 4, p. 1-17, 2006.
17. X. Dong, C. Xu, Y. Xie, and N. P. Jouppi, "Nvsim: A circuit-level performance, energy, and area model for emerging nonvolatile memory," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 31, no. 7, pp. 994-1007, 2012.

접수일: 2021년 9월 10일, 심사일: 2021년 12월 8일, 게재확정일: 2021년 12월 14일