A Survey of Low Power Design Techniques for Last Level Caches

Emmanuel Ofori-Attah′, Xiaohang Wang*, Michael Opoku Agyeman′
′Faculty of Art, Science and Technology, Univeristy of Northampton, UK.
*South China University of Technology, China.

***Abstract*—The end of Dennard scaling has shifted the focus of performance enhancement in technology to power budgeting techniques, specifically in the nano-meter domain because, leakage power depletes the total chip budget. Therefore, to meet the power budget, the number of resources per die could be limited. With this emerging factor, power consumption of on-chip components is detrimental to the future of transistor scaling. Fortunately, earlier research has identified the Last Level Cache (LLC) as one of the major power consuming element. Consequently, there have been several efforts towards reducing power consumption in LLCs. This paper presents a survey of recent contribution towards reducing power consumption in the LLC.**

I. Introduction

Multi-level Cache Architectures (MCA) have become increasingly popular for mitigating the disparity between memory and processors trading-off power consumption. MCA (Fig. 1) consumes a significant amount of power and affects the chip's total power ($P_{total} = P_{dynamic} \times P_{leakage}$). Particularly, the Last Level Cache (LLC) is said to consume most of the power and occupies 50% of the chip area due to its large size [1], [2]. With leakage power set to dominate power consumption in the near future, a reduction in LLC power and area can increase the number of components which can be activated through the Dark-Silicon solution. Fig. 1 a depicts a multi-level cache architecture in a typical heterogeneous many-core system. The L1 cache is the closest to the processor, small in size and thus, it is the fastest. In contrast, the L3 cache (LLC) is the furthest away from the processor and thus, is slower. However, it is much bigger, holds a large amount of data and thus, consumes most of the cache power.

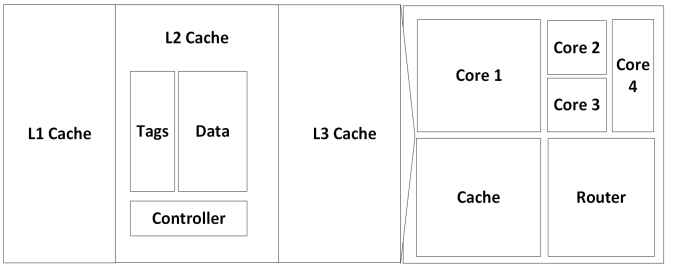

Fig. 1. Cache Architecture

In modern technology, on-chip caches are made up of Static Random-Access Memory (SRAM) technology, which has improved performance, but is expensive and suffers excessively from leakage power as technology scales down below 40 nm [3], [4]. Previously, Dynamic Random-Access Memory (DRAM) was used to design caches but have been overlooked due to the desirable properties (Low Access Latency and very high write endurance) of SRAM (Table I). DRAM is slower and thus cannot respond quickly to the demands of the cores (Fig. 1). With the number of core integration per chip escalating above its 100's, large and fast caches capable of handling large data will be required. Therefore, to avoid high access latency, the capacity of SRAM LLCs have to be increased proportionate to the number of the cores in System-on-Chip (SoC) for an assured Quality of Service (QoS). However, increase of SRAM LLCs multiplies the power consumption making it an undesirable technology for future embedded systems [5]. What this entails is an increase in the cost of implementation as well as the leakage power consumption in cache architectures. Therefore, efficient alternative design for SRAM technology is highly demanded for the scaling trend in transistors to continue.

Alternatively, to reduce the power consumption in caches, power-gating techniques are used to power-off idle parts of the cache during run-time since not all workloads require full access to the cache. However, these techniques degrade the performance of the system and therefore, minimizing cache power and finding a balance between power efficiency and high performance have become an interesting research area. Nonetheless, several architectural design techniques have been proposed to overcome this challenge.

This paper presents a survey of recent contributions towards reducing power consumption in cache architectures. Particularly, to reduce a significant amount of power, we target LLCs since it consumes majority of on-chip power. The rest of the paper is organised as follows, Section II presents surveys that also look into low power design techniques for caches. Section III presents an overview of Non-Volatile memory technologies while Section IV introduces monitoring cache behaviour as a technique for reducing power consumption in caches. Section V presents cache resizing techniques. Section VI summarises the techniques presented and finally, Section VIII concludes the paper.

II. Related Work

Cache power consumption is increasingly becoming a constraint for SoC. Although performance is enhanced by the in-

troduction of MCA, high power consumption and chip temperature becomes a problem [6], [7], [8], [9], [10]. To overcome this issue, several surveys [11], [12], [13] have presented low power consumption design techniques for caches. However, to the best of our knowledge, LLCs have not received much attention. Ofori-Attah et al. [13] conducted a survey on recent techniques for reducing the power consumption in Network-on-Chip (NoC) and Caches. In their work, techniques for leakage and dynamic power of caches have been addressed.

Similarly, Artes et al. [14] presented techniques for instruction memory organisations. Wei Zang et al. [11] evaluated the pros and cons of offline static and online dynamic cache tuning techniques. Mittal et al. [12] presented architecture level techniques on improving cache power management during runtime. They focussed on optimizing power efficiency. In [15], Mittal et al. also presented a comprehensive study of memory technologies and techniques to overcome the challenges in caches. Contrary to the work above, this work emphasises on reducing power consumption in LLCs.

III. Hybrid Architectures

One possible solution to this issue is the emerging Non-Volatile Memory (NVM) technologies (Spin-Transfer Torque RAM (STT-RAM) [16], Phase Change Memory (PCM), Resistive RAM (R-RAM) and Magnetic RAM (MRAM)). Memory itself can either be volatile or non-volatile. Volatile Memory (VM) requires power for it to function, and loses its content when the memory is powered-off. NVM on the other hand does not require power to store data. In addition to this, data is retained when the memory is shutdown. Although VM is faster, NVM are desirable because of their low cost of implementation, high speed, high density, scalability and ability to hold data under low power [17], [18].

Unfortunately, NVM technologies suffer from low write endurance (RRAM (10 λ^{11}), PCM (10 λ^{8}), STT-RAM (10 λ^{15})) and incur high energy during write operations which degrades their performance and use, due data being stored in the form of change in physical state [15]. Especially, during write intensive workloads. Therefore, implementing NVM trades-off leakage power for dynamic power. Furthermore, although MRAM implementation provides larger memory for the same die of footprint as SRAM, its write latency and energy are higher [19].

Table I presents a comparative evaluation of SRAM and NVM technologies. From the table, we can conclude that, STT-RAM mirrors characteristics (high data storage, fast speed) close to the properties of SRAM in terms of performance and power even though it suffers from dynamic power. However, in comparison to other technologies such as DRAM and PRAM, the cost is affordable. For example, although DRAM does not consume a lot of power, its high read and write latency can have a negative impact on the performance of the system as more cores are integrated. This even gets worse as workload increases. PRAM is also a good alternative for SRAM, however, its write latency is too high to consider implementing the LLC with.

TABLE I
Memory Technology Comparison [15]

Characteristics	SRAM	STT-RAM	DRAM	PRAM
Read Latency	Very Low	Low	High	High
Write Latency	Very Low	High	High	Very High
Read Power	Low	Low	Average	Average
Write Power	Low	High	Average	Low
Leakage Power	High	Low	Average	Low
Dynamic Power	Low	High	Average	High

However, as identified by [20], the high write energy of STT-RAM as a standalone technology in LLC makes it difficult to implement. For this purpose, hybrid memories have been proposed to exploit the low leakage power of STT-RAM and high performance of SRAM. In such an architecture, SRAM is used for write-intensive workloads while the STT-RAM is used for read-intensive workloads.

A. SRAM and STT-RAM Architectures

In this section, we discuss the techniques which have been employed in STT-RAM and SRAM hybrid architectures.

Li et al. [20] proposed a scheme for a hybrid architecture comprised of STT-RAM and SRAM. Firstly, a Neighbourhood Group Caching (NCU) technique is used for neighbouring cores to share their private STT-RAM groups with each other. This allows data to be shared among each core reducing latency and power consumption. During write misses, target blocks are loaded in the SRAM banks since it consumes less power and it is quicker. Furthermore, during a cache read, a request hit will be made available to the neighbouring groups or a copy of the target block will be made available for a future read. Compared to state-of-art architectures, the proposed architecture reduces power consumption by 40%.

Kim et al. [21] proposed a hybrid exclusive LLC cache architecture. Exclusive caches behave, perform, and operate differently from inclusive caches. In an exclusive cache architecture, cache blocks are inserted into the LLC after it has been evicted from the lower-level caches which is contrary to inclusive caches where, data is duplicated in each of the memory hierarchy. The proposed architecture has been implemented with many STT-RAM blocks and a few SRAM blocks. To reduce high write energy latency, a reuse distance predictor is used to determine which data needs to be placed in the SRAM or STT-RAM region of the LLC. The main idea is to eliminate the use of writing data that is less likely to be used again in the caches. By utilizing this technique, the power consumption in the LLC is reduced by 55%.

Similarly, Cheng et al. [22] proposed LAP, a technique which combines both non-inclusive and exclusive designs to manage the way the caches are handled in the LLCs. By using exclusive properties in the cache policy, the LAP is able to cache only the required data of the upper-level data in the LLC to reduce redundant writes. This technique reduces unnecessary writes in NVM memory technology reducing high energy writes. The characteristics of LAP are: only write

non-duplicate data, duplicate only useful clean data and no redundant LLC data-fill in the LLC.

Fanfan et al. [23] propose a technique called Feedback Learning Based Dead Write Termination (FLDWT) for eliminating dead blocks from inclusive STT-RAM LLCs. The proposed architecture works by classifying blocks into two categories (dead and live) based on access behaviour. Dead blocks are blocks which are have not been referenced for a long time. The technique works by discarding the dead blocks from being written to LLC before they are evicted to save power. The proposed technique reduces power consumption by 44.6%.

Safayenikoo et al. [24] proposed a hybrid cache memory for 3D CMPs comprised of STT-RAM and SRAM banks. To reduce the power consumption of high energy writes in STT-RAM cache banks, the number of writes to STT-RAM is monitored. Data is migrated to SRAM banks when the number of writes to the STT-RAM increases. This is done by employing a counter to count the number of accesses and writes for each bank. Mittal et al. [25] propose AYUSH, a technique which swaps an STT-RAM block to an SRAM block if the data stored in the SRAM block is old. This is done by using a least recently used parameter to determine if the data-item stored in the SRAM is likely to be used. If it is likely to be used, a NVM block data which just got inserted will be swapped for that SRAM block since it is likely to be used in the future to prevent high energy writes. Sukarn et al. [26] deals with high energy writes in STT-RAM based hybrid caches by restricting the number of writes associated with private blocks.

Aluru et al. [27] reduces the high current write in STT-RAM by splitting the cache line into many parts and writes them in different locations in the cache. The proposed solution reduces power consumption and reduces the errors that occur. Sato et al. [28] on other hand reduces power consumption by merging two adjacent lines and then writes them back to the STT-RAM LLC as one line instead of two writes and minimizes latency.

TABLE II
LLC SRAM AND STT-RAM TECHNOLOGY COMPARISONS

Characteristics	SRAM	STT-RAM	Hybrid
Read Latency	Low	Low	Low
Write Latency	Low	High	Average
Read Power	Low	Low	Low
Write Power	Low	High	Average
Leakage Power	High	Low	Average
Dynamic Power	Low	High	Average

B. Data Compression Schemes

Safayenikoo et. al proposed a compression method to reduce the number of write count which in turns reduces the power consumption in the LLCs by 78%. The proposed compression scheme reduces the number of repetitive words (zero).

Similarly, Liu et al. [29] proposed two compression schemes to reduce the power consumption in Multi-Level Cell (MLC) STT-RAM. MLC STT-RAM stores soft-bits and hard-bits but takes longer during read and write operations even though they offer better performances than single-level STT-RAMs. Unfortunately, during hard-bit accesses, it takes longer and consumes power and therefore downgrades the performance of the system. To overcome this, the first data compression technique reduces the size of the cache lines and reduces the time it takes to access it. The second technique allows an additional line to be stored in the hard-bit region. The proposed architecture reduces power consumption by 19%.

IV. MONITORING CACHE BEHAVIOUR

One of the most effective ways to reduce power consumption in on-chip caches is by monitoring cache blocks. Along these lines, two different approaches can be implemented (bypass predictions and dead blocks). Although the principal of locality is used to make decisions about data exchange in caches, majority of cache blocks in the LLCs are never referenced again and thus, useless/dead blocks dominate the LLC. For an LLC architecture which incorporates STT-RAM technology, bypassing writes to its bank could reduce the high write energy.

A. Bypass Predictions

Park et al. [30] proposed a Bypass First Policy (BFP) which reduces the number of blocks which are less likely to be used. This is done by bypassing blocks that are less likely to be used by default thus reducing useless blocks in the LLC which consumes a lot of power and space. The proposed solution reduces power consumption by 57%.

Hameed et al. [31] proposed a shared Row Buffer (RB) Organisation in an STT-RAM cache architecture which exploits the Row Buffer Locality (RBL) to reduce row buffer power consumption. In STT-RAM, each bank consists of a RB which stores the row that was recently accessed. If a data is required in the same row in the near future, this data is fetched from the row buffer instead of accessing the STT-RAM bit cell. This saves time and power. Unfortunately, RB conflict occurs when a current row in the RB is evicted and replaced with a new one. This increases the access latency. To reduce the RB conflict and misses, a shared RB organisation is proposed. Each bank is divided into different groups. Each group share the RB resources available to that group. This increases the RB hit rate because each group now has an RB assigned to it. In addition to this, a write-back bypass policy is proposed to reduce low RBL insertions into the RB and to also bypass the RB for write-back requests.

Azad et al. [4] proposed an Error-Correcting Code (ECC) protection technique which protects cache blocks by partitioning them into different groups. Instead of using worst-case protection for all cache blocks, the proposed technique categorises blocks into groups with different level of protection depending on the write requests. In comparison with conventional non-uniform ECC, the proposed algorithm reduces power consumption by 50% and guarantees the same level protection.

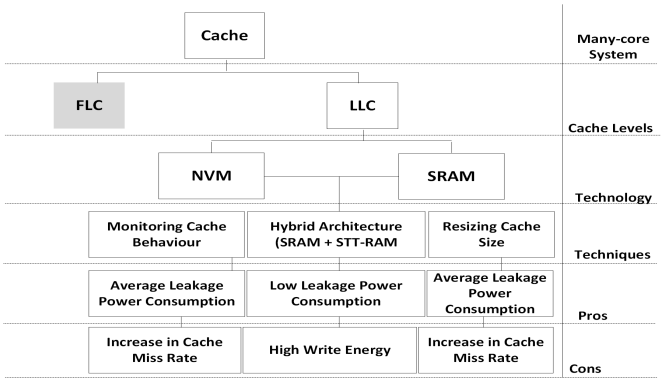

Fig. 2. Summary of LLC design techniques

B. Dead Blocks

Mnivannan et al. [32] proposed RADAR, a technique which eliminates dead blocks from the LLCs. RADAR works by using a Look-ahead and Look-back scheme to predict the regions that will be accessed. Based on the information gathered, RADAR evicts cache blocks in dead regions.

Das et al. [33] proposed a Sub-Level Insertion Policy (SLIP) to manage the movement of cache lines from one location to the other. SLIP is used to place cache lines into groups with similar cache access energy. This technique reduces power consumption in the LLCs by 22%.

Kurian et al. [34] proposed a technique that only replicates the high locality cache lines in the LLC slice and bypasses replicating low locality cache lines. The number of times a cache line is accessed is tracked and depending on the number of times it has been accessed, a replica will be made in the LLC. Similarly, Chaturvedi et al. [35] propose a technique that dynamically replicate high usage cache lines in the local banks close to the requesting core in non-uniform cache architectures.

Agawal et al. [26] on the hand proposed a hybrid architecture that reduces the number of writes by only storing the tags and directory entry of private blocks in the LLC. Private blocks are blocks that have exclusive permission to only one core and thus, they are different from the copies in the LLC. Therefore, these copies are useless and are only useful when a write-back occurs. To reduce this, only tags and directory entries are stored instead of data to reduce the write energy.

V. Resizing Cache Size

Another widely used technique to reduce power is to shutdown parts of the cache which are idle. However, shutting down idle parts of the cache affects performance. Particularly, when there is a sudden overshoot in the workloads. Therefore, power-gating techniques should consider performance degradation when resizing the cache (cache banks and ways are powered-off) [36].

Chakraborty et al. [36] proposed a bank shutdown technique to reduce the power consumption in the LLCs. The banks that are less likely to be used are powered-off and their future requests redirected to neighbouring banks with average utilization. High active banks will not be selected to respond to the requests of the shutdown banks because they will not be able to handle it and will have a negative impact on the performance of the system. Furthermore, there is a limit to the number of banks that can be powered-off to prevent performance loss. Consequently, because not all banks can be powered-off, lightly used banks will have some of their ways turned off. To choose which bank to shutdown, a counter is attached to monitor accesses.

Park et al. [37] proposed a technique which improves the performance of LLCs when some cache ways are powered-off to reduce the power consumption by 34%. In the proposed architecture, all tag ways can be accessed in parallel. Additionally, a partial tag-based way filter is proposed and attached to each cache way. The proposed algorithm dynamically activates the number of ways that is required.

Cheng et al. [38] proposed a mechanism for turning off cache slices. To turn off a cache slice, a power management unit is employed to store information about the cache access from the previous epoch. This information is then used to decide the capacity required by the workload.

Choi et al. [39] proposed a cache way allocation scheme which effectively allocates SRAM and NVM ways by considering the impact of NVM writes by the landfill operation. Unlike other schemes which allocates write-intensive blocks to the SRAM ways, this scheme reduces the NVM write counts through a two-step approach. The proposed approach reduces power consumption with approximate range of 28.6% - 37%.

Fig. 2 presents an overview of the techniques presented in this paper where the grey rectangle represents the First Level Cache (FLC) and the white rectangle represents the LLC[1]. The techniques discussed so far can be categorised into three main parts (Monitoring cache behaviour, hybrid architecture and resizing the cache size. Although all the techniques reduce leakage power consumption, the hybrid architecture reduces a large amount.

VI. Conclusion

To address the power challenges in Multi-Level Caches, various solutions have been proposed over the past years. However, comprehensive work that evaluates low power techniques for LLC design is limited. In this paper, we presented low power design techniques for LLCs. Our findings demonstrate that, integrating caches with STT-RAM and SRAM provides an effective solution to the leakage power consumption dominating modern technology. By creating a hybrid memory, STT-RAM banks can be used for read-intensive data while SRAM is used for write-intensive workloads. In addition to this, LLC power consumption can also be reduced by data compression schemes, eliminating dead blocks, and resizing cache size.

[1]Though FLC power consumption is important, we focus on the LLC because they have not received much attention and consumes more power and area.

REFERENCES

[1] D. Wendel, R. Kalla, R. Cargoni, J. Clables, J. Friedrich, R. Frech, J. Kahle, B. Sinharoy, W. Starke, S. Taylor, S. Weitzel, S. G. Chu, S. Islam, and V. Zyuban, "The implementation of power7tm: A highly parallel and scalable multi-core high-end server processor," in *IEEE International Solid-State Circuits Conference - (ISSCC)*, 2010.

[2] G. Gammie, A. Wang, H. Mair, R. Lagerquist, M. Chau, P. Royannez, S. Gururajarao, and U. Ko, "Smartreflex power and performance management technologies for 90 nm, 65 nm, and 45 nm mobile application processors," *Proceedings of the IEEE*, 2010.

[3] X. Bi, M. Mao, D. Wang, and H. H. Li, "Cross-layer optimization for multilevel cell stt-ram caches," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2017.

[4] Z. Azad, H. Farbeh, A. M. H. Monazzah, and S. G. Miremadi, "An efficient protection technique for last level stt-ram caches in multi-core processors," *IEEE Transactions on Parallel and Distributed Systems*, 2017.

[5] N. A. Kurd, S. Bhamidipati, C. Mozak, J. L. Miller, T. M. Wilson, M. Nemani, and M. Chowdhury, "Westmere: A family of 32nm ia processors," in *IEEE International Solid-State Circuits Conference - (ISSCC)*, 2010.

[6] M. A. Awan and S. M. Petters, "Enhanced race-to-halt: A leakage-aware energy management approach for dynamic priority systems," in *23rd Euromicro Conference on Real-Time Systems*, 2011.

[7] W. B. E. Ofori-Attah and M. O. Agyeman, "Architectural techniques for improving the power consumption of noc-based cmps: A case study of cache and network layer," in *Emerging Network-on-Chip Architectures for Low Power Embedded Systems*, 2017.

[8] J. Dai, M. Guan, and L. Wang, "Exploiting early tag access for reducing l1 data cache energy in embedded processors," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2014.

[9] A. Ranjan, S. Venkataramani, Z. Pajouhi, R. Venkatesan, K. Roy, and A. Raghunathan, "Staxcache: An approximate, energy efficient stt-mram cache," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2017.

[10] Z. Gan, M. Zhang, Z. Gu, and J. Zhang, "Minimizing energy consumption for embedded multicore systems using cache configuration and task mapping," in *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, 2016.

[11] W. Zang and A. Gordon-Ross, "A survey on cache tuning from a power/energy perspective," *ACM Comput. Surv.*, 2013.

[12] S. Mittal, "A survey of architectural techniques for improving cache power effi- ciency. sustainable computing: Informatics and systems," *Sustainable Computing: Informatics and Systems (SUSCOM)*, 2014.

[13] E. Ofori-Attah, W. Bhebhe, and M. O. Agyeman, "Architectural techniques for improving the power consumption of noc-based cmps: A case study of cache and network layer," *Journal of Low Power Electronics and Applications*, vol. 7, 2017.

[14] J. A.Artes, J.Ayala and F.Catthoor, "Survey of low-energy techniques for instruction memory organisations in embedded systems," in *Journal of Signal Processing Systems 70(1):1-19 January*, 2013.

[15] S. Mittal, J. S. Vetter, and D. Li, "A survey of architectural approaches for managing embedded dram and non-volatile on-chip caches," *IEEE Transactions on Parallel and Distributed Systems*, 2015.

[16] D. Apalkov, A. Khvalkovskiy, S. Watts, V. Nikitin, X. Tang, D. Lottis, K. Moon, X. Luo, E. Chen, A. Ong, A. Driskill-Smith, and M. Krounbi, "Spin-transfer torque magnetic random access memory (stt-mram)," *J. Emerg. Technol. Comput. Syst.*, 2013.

[17] H. Noguchi, K. Kushida, K. Ikegami, K. Abe, E. Kitagawa, S. Kashiwada, C. Kamata, A. Kawasumi, H. Hara, and S. Fujita, "A 250-mhz 256b-i/o 1-mb stt-mram with advanced perpendicular mtj based dual cell for nonvolatile magnetic caches to reduce active power of processors," in *Symposium on VLSI Circuits*, 2013.

[18] H. Noguchi, K. Ikegami, N. Shimomura, T. Tetsufumi, J. Ito, and S. Fujita, "Highly reliable and low-power nonvolatile cache memory with advanced perpendicular stt-mram for high-performance cpu," in *Symposium on VLSI Circuits Digest of Technical Papers*, 2014.

[19] S. Senni, L. Torres, G. Sassatelli, A. Bukto, and B. Mussard, "Exploration of magnetic ram based memory hierarchy for multicore architecture," in *IEEE Computer Society Annual Symposium on VLSI*, 2014.

[20] J. Li, C. J. Xue, and Y. Xu, "Stt-ram based energy-efficiency hybrid cache for cmps," in *IEEE/IFIP 19th International Conference on VLSI and System-on-Chip*, 2011.

[21] N. Kim, J. Ahn, W. Seo, and K. Choi, "Energy-efficient exclusive last-level hybrid caches consisting of sram and stt-ram," in *IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, 2015.

[22] H. Y. Cheng, J. Zhao, J. Sampson, M. J. Irwin, A. Jaleel, Y. Lu, and Y. Xie, "Lap: Loop-block aware inclusion properties for energy-efficient asymmetric last level caches," in *ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, 2016.

[23] F. Shen, Y. He, J. Zhang, N. Jiang, Q. Li, and J. Li, "Feedback learning based dead write termination for energy efficient stt-ram caches," *Chinese Journal of Electronics*, 2017.

[24] P. Safayenikoo, A. Asad, M. Fathy, and F. Mohammadi, "Exploiting non-uniformity of write accesses for designing a high-endurance hybrid last level cache in 3d cmps," in *IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2017.

[25] S. Mittal and J. S. Vetter, "Ayush: Extending lifetime of sram-nvm way-based hybrid caches using wear-leveling," in *IEEE 23rd International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, 2015.

[26] S. Agarwal and H. K. Kapoor, "Restricting writes for energy-efficient hybrid cache in multi-core architectures," in *IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, 2016.

[27] R. K. Aluru and S. Ghosh, "Droop mitigating last level cache architecture for sttram," in *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2017.

[28] M. Sato, Z. Sakai, R. Egawa, and H. Kobayashi, "An adjacent-line-merging writeback scheme for stt-ram last-level caches," in *IEEE Symposium in Low-Power and High-Speed Chips (COOL CHIPS)*, 2017.

[29] L. Liu, P. Chi, S. Li, Y. Cheng, and Y. Xie, "Building energy-efficient multi-level cell stt-ram caches with data compression," in *22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2017.

[30] J. J. K. Park, Y. Park, and S. Mahlke, "A bypass first policy for energy-efficient last level caches," in *International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation (SAMOS)*, 2016.

[31] F. Hameed and M. B. Tahoori, "Architecting stt last-level-cache for performance and energy improvement," in *17th International Symposium on Quality Electronic Design (ISQED)*, 2016.

[32] M. Manivannan, V. Papaefstathiou, M. Pericas, and P. Stenstrom, "Radar: Runtime-assisted dead region management for last-level caches," in *IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2016.

[33] S. Das, T. M. Aamodt, and W. J. Dally, "Slip: Reducing wire energy in the memory hierarchy," in *ACM/IEEE 42nd Annual International Symposium on Computer Architecture (ISCA)*, 2015.

[34] G. Kurian, S. Devadas, and O. Khan, "Locality-aware data replication in the last-level cache," in *IEEE 20th International Symposium on High Performance Computer Architecture (HPCA)*, 2014.

[35] "Selective cache line replication scheme in shared last level cache," *Procedia Computer Science*, pp. 1095 – 1107, 2015.

[36] S. Chakraborty and H. K. Kapoor, "Static energy reduction by performance linked dynamic cache resizing," in *IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, 2016.

[37] J. Park, J. Lee, and S. Kim, "A way-filtering-based dynamic logicalassociative cache architecture for low-energy consumption," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2017.

[38] H. Y. Cheng, M. Poremba, N. Shahidi, I. Stalev, M. J. Irwin, M. Kandemir, J. Sampson, and Y. Xie, "Eecache: Exploiting design choices in energy-efficient last-level caches for chip multiprocessors," in *IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, 2014.

[39] J. Choi and G. H. Park, "Nvm way allocation scheme to reduce nvm writes for hybrid cache architecture in chip-multiprocessors," *IEEE Transactions on Parallel and Distributed Systems*, 2017.