Elastic Adaptive Prefetching for Non-Volatile Cache in IoT Terminals

MaoNi1,2,3, LanChen1,3,a, XiaoranHao1,3, Chenji Liu1,2,3, YihengZhang1,3 and Ying Li1

Abstract STT-RAM with high storage density, near-zero leakage energy and CMOS compatibility is regarded as a replacement for SRAM to build large-sized cache, which can effectively alleviate the “memory wall” and improve computing power of IoT terminals. The state-of-the-art Near-Side Prefetch Throttling (NST) oriented to SRAM cache can effectively hide the access latency of off-chip memory. However, it also shows some inadaptability to the long write latency and high write energy of STT-RAM cache. The NST algorithm can not timely alleviate the cache congestion caused by STT-RAM long write latency, moreover, if the STT-RAM cache is congested, adjusting the prefetch distance is invalid to improve the prefetch timeliness. In response to the above problems, this paper novelty proposes a periodic and real-time complementary prefetch algorithm called ENCP for STT-RAM cache. Experiments show that, compared to the best-performed STREAM prefetcher, ENCP can reduce the write energy of STT-RAM cache by 8.3% on average and 23% the most and improve the CPU IPC performance by 0.46% on average and 3.1% the most. And the ENCP has better performance and lower dynamic energy compared with NST with almost the same hardware overhead.

Keywords: L2 cache; NVM; prefetcher; STT-RAM
Classification: Integrated circuits (memory, logic)

1. Introduction

In 2025, there will be 20 billion smart nodes connected to the network in China [1], and a large number of heterogeneous devices with different application requirements will produce enormous number of plural structure data, which significantly increases the load of cloud and network bandwidth. At the same time, the data privacy and real-time requirement of applications also put forward higher demand for the terminal processing ability [2]. However, terminals usually have limited power supply and volume. And the memory wall also hinders the improvement of terminal computing power [3]. Large-sized SRAM cache can effectively hide the access delay of off-chip storage, but its high leakage power and low storage density increase the chip area and energy. The emerging Non-volatile memory, such as PCM, RRAM, STT-RAM, etc., have advantages of near-zero leakage energy, high storage density, and high read-write speed. [4-7]. Among them, STT-RAM with excellent durability and CMOS compatibility [7-11] is regarded as a promising replacement for traditional SRAM cache. However, STT-RAM has some drawbacks such as high write energy and long write latency [11-13]. In order to mitigate the impact of its disadvantage, [14-15] reduce the STT-RAM write latency by upsizing the write access transistor or using differential bits. [16-17] propose a heterogeneous hybrid cache composed of STT-RAM and SRAM. [18-19] reduce the write latency and energy by changing the retention time of STT-RAM and proposed a hybrid cache composed of STT-MRAM and semi-volatile STT-RAM, in which the data location is decided by the data retention time in the cache.

Data prefetch is an effective technique to alleviate the effect of the “memory wall”. For the SRAM cache, some scholars [21-24] improve the cache performance by high prefetch coverage. However, prefetch write is one of the main write sources for the LLC [17], and especially for the STT-RAM cache, aggressive prefetch also increases the cache pressure because of STT-RAM long write latency and high write energy. This paper deeply analyzes the impact of existing prefetch algorithms on STT-RAM cache and innovatively proposes a prefetch method named ENCP, which alleviates the adverse impact of STT-RAM read-write asymmetry on cache performance. The main contributions of this article are as follows:

1. Experiments show that, although STT-RAM has long write latency, the STT-RAM L2 cache also can improve the CPU IPC performance by up to 38.4% and 3.1 % on average compared to the same sized SRAM L2 cache.

2. Aggressive prefetch improves the CPU IPC performance, but also amplifies the adverse impact of STT-RAM’s long write latency and high write energy which swallows its advantages of high storage density and low leakage energy.

3. Because of STT-RAM long write latency, aggressive write leads to STT-RAM cache congestion, which results in 8.7% prefetch increase compared to SRAM cache.

4. The state-of-the-art NST is valid for STT-RAM cache.
However, it also shows some inadaptability to STT-RAM long write latency. When the prefetch number is more than 9000 in a time window, adjusting prefetch distance is invalid to reduce MSHR hit rate. And periodically adjusting the prefetch distance only can not alleviate the cache congestion in time.

5. This paper proposes an elastic adaptive prefetch strategy named ENCP. This strategy adjusts the prefetch distance periodically according to the MSHR (Miss-status Handling Registers) hit ratio, and adjusts the prefetch aggressiveness in real time by the MSHR saturation. The ENCP can effectively alleviate the non-volatile cache congestion, which improves the CPU IPC performance and reduces the NVM write energy.

2. Related Works

To solve the read-write asymmetry of STT-RAM, there are a lot of works [16-20, 25-27] focusing on the research of STT-RAM cache. [16-17] propose a hybrid cache composed of SRAM and STT-RAM, which stores the write-frequent data into SRAM to avoid STT-RAM’s read-write asymmetry. However, this scheme needs to solve the problems of write-frequent data identification and data migration on different cache media. Korgaonkar [20] confirms the congestion of NVM cache and relieve the cache pressure by bypass writes. Prefetch write is an important write source of cache [16-17]. The FDP[26] and NST[27] oriented to SRAM cache can improve prefetch timeliness by adjusting prefetch distance, and these prefetch throttling are also valid for alleviating the long write latency and high write energy of STT-RAM cache.

As shown in ① of Fig.1, the data of address x + 3 arrives after the access demand, we call it late prefetch, so the prefetch distance should be increased. ③ of Fig.1 shows that the data of address M+n arrives earlier than access demand, which is easy to replace the valid data in the cache, which increases data movement between the cache and main memory, in this situation, the prefetch distance should be decreased. ② in Fig.1 means the ideal prefetch distance. In order to achieve prefetch data as timely as possible, the FDP adds 1 bit to each data block in the cache to mark whether it comes from prefetch, and counts all hit prefetch blocks in the cache. At the same time, it adds 1 bit to each entry in MSHR to mark whether the entry is a prefetch request or a read miss request, and counts total number of late prefetch. The prefetch distance is adjusted according to the ratio of the total late prefetch in MSHR to the hit prefetch in the cache in a time window. The MSHR is a miss status register in the cache, which is used to save cache read miss and prefetch requests. When the cache hits, the requested data will be directly obtained from the cache, and when the cache misses, it will check whether the same cache block is being serviced by an earlier memory request. If it is, discard it and if the entry is marked as prefetch, this prefetch is late prefetch. Otherwise, the MSHR entry will be allocated for the new request. Although FDP [26] can effectively adjust the prefetch distance, it has a large hardware overhead. While NST only needs about 1% hardware overhead of FDP to achieve the same performance. It is because the NST adjusts the prefetch distance only by the ratio of late prefetch in the MSHR to the total prefetch transmitted to the main memory from the MSHR. Both FDP and NST do not consider their adaptability to read-write asymmetry of STT-RAM cache.

![Fig. 1 Prefetch distance](image1)

To solve the above problem, this paper focuses on the impact of existing prefetch algorithms on STT-RAM cache and proposes an elastic adaptive prefetch strategy for read-write asymmetric NVM cache.

3. Background and Motivation

3.1. The Benefits of STT-RAM’s High Storage Density

Korgaonkar K [20] proposes that the capacity of STT-RAM is 3 times that of SRAM when the STT-RAM write latency is 10ns, and it can be 5 times when the latency is 30ns. We assume that the STT-RAM write latency is 15ns, and the STT-RAM capacity is about 4 times that of SRAM with the same chip area. In gem5 with system configurations shown in Table 3, the experimental results show that, compared to the same area SRAM cache(256KB), 1MB STT-RAM cache with 15ns write latency can still achieve the performance improvement by up to 38.4% as shown in Fig. 2. However, compared with SRAM cache with the same capacity, the performance of STT-RAM decreases significantly because of STT-RAM long write latency. So we’ll further analyze the impact of different write latency on the IPC performance with STT-RAM L2 Cache.

![Fig. 2 The IPC performance with large-capacity STT-RAM L2 cache](image2)

3.2. Effect of prefetching on STT-RAM cache performance

[20] found that the benefits of STT-RAM’s high storage density are gradually swallowed up with the increase of STT-RAM write latency, and our previous research work WANCP[28] also confirmed that when the write latency increases from 2ns to 30ns, the CPU IPC performance
decreases by 30.7%. At the same time, [28] also found that the aggressive prefetching can effectively improve the CPU IPC performance. However, the prefetch write is one of the cache write sources which accounts for more than 20% of L2 cache writes on average and more than 60% at most, and the aggressive prefetching would bring more prefetch writes. Fig. 3 shows that, the prefetch writes increases by up to 6 times as the prefetch aggressiveness increases from P1 to P3 shown in Table 1. A large number of prefetch writes would increase the pressure on STT-RAM cache port and amplify the impact of its high write energy consumption.

3.3 The Effect of Long Write Latency on Prefetch

![Fig. 3 Prefetch energy with different STREAM configurations](image)

Table 1. STREAM prefetcher configurations

| Prefetcher | Pre Distance | Pre Degree |
|------------|--------------|------------|
| P1         | 4            | 2          |
| P2         | 16           | 2          |
| P3         | 32           | 4          |

3.4 The Inadaptability of NST to STT-RAM Cache

The NST for SRAM cache improves the cache performance by adjusting the prefetch distance according to the MSHR hit rate. It reduces prefetch operations and improves the prefetch timeliness. However, we found that the NST algorithm is not fully applicable to STT-RAM cache with long write latency. When the number of prefetch issued in each time window (12.8ns) is greater than the threshold (shown in Fig.11), adjusting the prefetch distance is invalid to reduce the hit rate of MSHR. In addition, although the NST can reduce the prefetch periodically, it can not alleviate the cache congestion caused by long write latency in real time.

4. ENCP Architecture

Timely prefetch can effectively hide the long access latency of off-chip main memory. ENCP periodically adjusts the prefetch distance and degree according to the MSHR hit ratio and adjusts the prefetch degree in real time by the saturation of MSHR, which is complementary to the periodic prefetch distance adjustment. The proposed ENCP can effectively alleviate the STT-RAM cache congestion and reduce STT-RAM cache dynamic energy.

4.1 Monitoring of Prefetch Timelines

This paper uses MSHR to monitor the prefetch timeliness. In order to minimize the hardware overhead, we only add 1 bit to each entry of MSHR(pre_bit, 1 means a prefetch request, and 0 means a read miss request) and 3 registers (pres_in_MSHR, pre_laters, pre_issues) to monitor the MSHR hit ratio. When a prefetch request is stored in MSHR for the first time, the pre_bit is set to 1, and the pres_in_MSHR recording the prefetch number of MSHR is

Fig. 4 The prefetch issues under different cache write latency

Fig. 5 Total prefetch amount of SRAM and STT-RAM cache

Korgaonkar et al. [20] confirms that long write latency causes cache congestion. Our previous work WANCP [28] also confirms that aggressive prefetch make congestion more serious. In Fig. 10, the saturation peak of MSHR increasing along with the enhancement of prefetch aggressiveness also indicates the existence of cache congestion to some extent. Fig. 7 shows that, when MSHR tends to be saturated (t7-t19, t25-t34 in Fig. 4(a) and t1-t29 in Fig.4 (b)) with the most aggressive STREAM P3 prefetcher, the prefetch issues increases as the write latency increases from 2ns to 15ns, such as t7-t19, t25-t34 in Fig.4(a) and t1-t29 in Fig.4(b). For other time windows, such as t1-t7, t19-t25 in Fig. 4(a) and t29-t35 in Fig.4 (b), which have low MSHR saturation, although the write latency is different, the prefetch amount is roughly the same. From Fig. 5 we can see that under the same cache capacity and prefetch strategy, the prefetch amount of STT-RAM L2 is 8.7% more than that of SRAM cache on average. In other words, the prefetch amount increase of STT-RAM L2 cache mainly occurs when the cache is congested. This proves our intuition that the prefetched data from the main memory can not be written to the cache in time with cache congestion, which leads to more read miss and then more data prefetch.
increased by 1. When a prefetch request hits in MSHR, we set the pre_bit to 0, and reduce the pres_in_MSHR by 1, while increase the pre_laters by 1 which recording the number of the late prefetch. If the prefetch request in MSHR is sent to the bus, the pre_issues recording all prefetch sent to main memory is increased by 1, and if the pre_bit of this MSHR entry is 1, the pres_in_MSHR is reduced by 1. The ratio of pre_laters to pre_issues can well reflect the prefetch timeliness in the current time window.

4.2 Real Time Monitoring of MSHR Saturation
In our previous work [28], we found that the cache congestion caused by STT-RAM long write latency can be alleviated by adjusting the prefetch aggressiveness when MSHR is in saturation state. In this paper, ENCP not only takes into account the real-time MSHR saturation, but also considers the read miss in MSHR, because the read miss directly affect CPU pipeline. The MSHR saturation is represented by MSHR_allocated, and in order to reduce the algorithm overhead, the read miss number is equal to the difference of MSHR_allocated and pres_in_MSHR.

4.3 ENCP Algorithm

| Symbol | Value | Description            |
|--------|-------|------------------------|
| Dis    | Default 16 | Prefetch distance |
| Deg    | Default 4  | Prefetch degree        |
| h      | Read the value of Pre_laters register | Hit number of MSHR |
| a      | Read the value of Pre_issues register | The number of requests sent to bus from MSHR |
| S      | h/a    | Late prefetch ratio    |
| Win    | 12.8ns | Time window            |
| Mshr_u | Default 0 | The used entries of MSHR |
| Mshr_p | Read the value of Pres_in_MSHR | The number of prefetch requests in register | MSHR |

In response to the inapplicability of the state-of-the-art NST to the NVM cache, this paper proposes a periodic and real-time prefetch strategy. The left side of Fig.6 shows the progress of periodic adjustment to prefetch parameters, and the right side is the real-time adjustment process according to read miss request number in MSHR. As shown in the left side of Fig.6, when the system powers on, the prefetch distance Dis and prefetch degree Deg are set to 16 and 4 respectively. In each time window, monitor the lat late prefetch number (h) and the total prefetch issues (a) from MSHR to bus, and then calculate the late prefetch ratio S. If S is between 3% - 10%, the Dis remains unchanged. If S is less than 3%, Dis minuses 1 until down to 4. When S is greater than 10% and the prefetch amount is smaller than 9000, increasing the Dis can effectively decrease S as shown in Fig.11, so we increase the Dis by 4 each time until the Dis is equal to 32 (The experiments found that when Dis is larger than 32, the cache performance improvement is negligible); if S is greater than 10% and a is greater than 9000, increasing the Dis can not effectively reduce MSHR hit ratio, so we reduce the Deg to 3 and decrease the Dis, which can partially relieve the cache pressure. The pre_laters and pre_issues registers are cleared at the end of each time window.

As shown in the right side of Fig.6, in the real time adjustment process, when Mshr_u (used entries in MSHR) is less than 4, there is no need to adjust the Deg which holds the results of periodic adjustment. And when Mshr_u is larger than 4 and the read miss number is larger than 3, the Deg reduces to 2, otherwise, the Deg is set to 3. Multiple real time adjustments may be triggered within a time window. Adjusting the prefetch degree in real time is as a supplement to periodic adjustment of prefetch distance in ENCP, so as to alleviate the cache pressure.

4.4 Hardware Overhead
We assume that the L2 cache capacity is 1024KB and the MSHR has 18 entries. Compared with STREAM algorithm, ENCP only needs additional 4 bytes pre_laters and pre_issues registers, 5 bits Pres_in_MSHR register and 18 bits pre_bit for prefetch identification in MSHR. The total hardware overhead is 11 bytes except some simple logic overhead. With the same cache capacity, the FDP prefetcher needs 2072 bytes and the NST algorithm needs 12 bytes [27]. Therefore, ENCP and NST have almost the same hardware overhead, which is only 1% that of FDP.

5. Experimental Setup
This paper evaluates the performance of out-of-order processor with different STT-RAM L2 cache capacity in the gem5 simulator [29] which is a cycle-accurate simulator and provides an accurate cache model. We modify the SRAM cache model of gem5 according to [16] and add STT-RAM write latency parameter to the cache. Table 3 shows the
configuration parameters of gem5 and the read write latency of STT-RAM and SRAM. We select artificial intelligence, image and video compression and other general applications in SPEC CPU2006 [30] as our experiment benchmarks to simulate the cache behavior of IoT applications as real as possible. We run benchmarks for 800 million instructions to ensure that the occupancy rate of L2 cache is beyond 99% on average.

| Table 3. System configurations |
|-------------------------------|
| Parameter | Value |
| CPU | O3, 2GHz |
| L1 Cache | 32KB iCache,64KB dCache, 64B line size, 2-way set assoc, R/W delay 2cycles |
| L2 Cache | 64B line size, 8-way set assoc, read 4cycle; write 4-20-30-60cycles, cache size 256KB-1024KB, MSHR=18 |

6. Simulation Results

6.1 CPU IPC performance with ENCP
It can be seen from Fig.7, the IPC performance with ENCP is better than the NST and our previous work WANCP [28]. And compared with STREAM P3, the IPC performance is improved by 3.1% the most. However, for some applications, such as omnetpp, h264 and asar, the IPC performance with ENCP decreases by 0.1%, which means that although ENCP can improve the CPU IPC performance for most applications by alleviating the cache congestion, reducing the number of prefetch data also decreases the cache hit rate to a certain extent. Therefore, when the performance decline caused by cache hit rate reduction exceeds the performance improvement benefits from the cache congestion alleviation, the IPC performance declines.

![Fig. 7 IPC performance with ENCP](image)

6.2 Prefetch Energy with ENCP
The ENCP improves the IPC performance with STT-RAM L2 cache by periodically and real time adjusting the prefetch distance and prefetch degree as shown in Fig.7. The “Prefetch energy” of Fig.8 means the energy consumption of prefetch operations, which is calculated according to the number of write operations and dynamic write power consumption [17]. From Fig.8 we can see that the prefetch energy consumption is significantly lower than P3 and NST, however, it is slightly higher than WANCP.

![Fig. 8 Prefetch energy with ENCP](image)

6.3 Sensitivity of ENCP to Parameters

Compared with WANCP, more accurate prefetch adjustment slightly increases prefetch energy consumption, but gains more performance benefits. As shown in Fig. 9, compared with STREAM P3, ENCP can get 0.46% IPC performance gain on average, while WANCP only gets 0.19%. And ENCP can reduce prefetch energy consumption by 8.3% on average and 23% the most compared with STREAM P3. The decrease of prefetch operations can undoubtedly reduce the main memory bandwidth, and more important, this can effectively alleviate the dynamic energy consumption of STT-RAM cache.

![Fig. 9 IPC performance and cache energy consumption with different prefetchers](image)

In this paper, the depth of MSHR in L2 cache is 18. It can be seen from Fig.10, for all benchmarks the saturation peak of MSHR with STREAM P3 is between 10 and 18. The MSHR depth affects the hardware overhead of the ENCP. In ENCP algoritthm, when the prefetch amount in a time window is greater than 9000 (t1-t7 in the top of Fig.13 and t1-t23 in the bottom of Fig.11), the MSHR hit ratio is almost the same under different prefetch distance (8, 16, 32), so we can decrease the prefetch distance to alleviate the congestion of STT-RAM cache. And when the prefetch amount in a time window is smaller than 9000 (t7-t23 in the top of Fig.11), increasing prefetch distance can reduce the hit rate of MSHR. So in ENCP, we choose 9000 as the threshold to periodically update the prefetch parameters. The time...
window size (12.8ns) and MSHR hit rate threshold (3%-10%) in this paper refer to [27]. Limited to the length of this article, they will not be analyzed detailed.

![Fig.11: MSHR hit ratio sensitivity to different prefetch distance: top(leslie), bottom(sjeng)](image)

| MSHR hit ratio | issues/10000 |
|----------------|--------------|
| 0.3            |              |
| 0.5            |              |
| 0.7            |              |
| 1.0            |              |
| 1.3            |              |
| 1.5            |              |
| 1.7            |              |
| 2.0            |              |
| 2.5            |              |
| 3.0            |              |
| 3.5            |              |
| 4.0            |              |
| 4.5            |              |
| 5.0            |              |
| 5.5            |              |
| 6.0            |              |
| 6.5            |              |
| 7.0            |              |
| 7.5            |              |
| 8.0            |              |
| 8.5            |              |
| 9.0            |              |
| 9.5            |              |
| 10.0           |              |

7. Conclusions

STT-RAM with high storage density and near zero leakage power is a promising alternative to SRAM to build large-capacity cache. However, the SRAM-oriented prefetch algorithms didn’t consider the STT-RAM read-write asymmetry which brings adverse effect to CPU performance and energy. Experiments show that the aggressive STREAM algorithm can improve the CPU IPC performance, but it also increases the STT-RAM write number which enlarges cache pressure and energy consumption. The advanced prefetch algorithms such as FDP and NST can alleviate this problem to a certain extent, however, it is still not the best solution to STT-RAM cache. When the cache is saturated or even congested, increasing the prefetch distance has no use to solve prefetch late problem, moreover, periodically adjusting prefetch distance can not decrease the cache port pressure caused by long STT-RAM write latency in time. To solve these problems, this paper proposes a periodic and real-time complementary dynamic adaptive prefetching (ENCP). ENCP, compared to the best-performed STREAM P3, can reduce the write energy of STT-RAM L2 cache by 8.3% on average and 23% at most, and improve the IPC performance by 0.46% on average and 3.1% at most with extremely low hardware overhead.

Acknowledgments

This research was funded by the National Key R&D Program of China under Grant 2019YFB2102400.

References

[1] 2020 China Intelligent Internet of Things (AIoT) White Paper, 2020
[2] A. Tison, L. Roman, Vinu, et al. “Right-Provisioned IoT Edge Computing: An Overview”, GLS VLSI ’19, May 2019 Pages 531–536 doi:10.1145/3299874.3319338
[3] J. Ni., K. Liu., B. Wu B., et al. “Write Back Energy Optimization for STT-MRAM-based Last-level Cache with Data Pattern Characterization”, JETC, 2020, 16(3).
[4] H. Sun et al., “An Energy-Efficient and Fast Scheme for Hybrid Storage Class Memory in an AIoT Terminal System,” Electronics, (2020).
[5] C. Liu et al., “Fast cacheline-based data replacement for hybrid DRAM and STT-MRAM main memory,” IEICE Electro. Express, (2020).
[6] K. Kuan, T. Adegbi, “HALLS: An Energy-Efficient Highly Adaptable Last Level STT-RAM Cache for Multicore Systems,” IEEE Trans. Comp., vol. 68, no. 11, pp. 1623–1634, Nov 2019.
[7] A. Jorg et al., “Cache revive: Architecting volatile STT-RAM caches for enhanced performance in CMPs,” in Proc. DAC Design Autom. Conf., Jun. 2012, pp. 243–252.
[8] X. Dong et al., “Circuit and microarchitectural evaluation of 3D stacking magnetic RAM (MRAM) as a universal memory replacement”, DAC(2008), doi:10.1145/1391469.1391610
[9] Z. Sun et al., “Multi retention level STT-RAM cache designs with a dynamic refresh scheme”, MICRO(2011)
[10] K. Kuan, T. Adegbi, “Energy-Efficient Runtime Adaptable L1 STT-RAM Cache Design”, IEEE Trans. Comp. Aid. Des. Int. Circ., Sys., 2019.
[11] K. Kuan, et al. “A Study of Runtime Adaptive Prefetching for STT-RAM Caches,” ICCD(2020) (doi: 10.1109/ICCD50377.2020.00051)
[12] J. Ahn, et al., “Dasca: Dead write prediction assisted stt-ram cache architecture,” HPCA(2014).
[13] N. Sayed, et al. “Cross-layer adaptive approach for performance and power optimization in STT-RAM,” DATE(2018)
[14] M. Shihab, et al., “Couture: Tailoring STT-MRAM for Persistent Main Memory,” 4th Workshop Interact. NVM/Flash Operat. Sys. Workloads, 2016.
[15] H. Noguchi, et al., “7.2 4mb STT-RAM-based cache with memory-access-aware power optimization and write-verify-write/read-modify-write scheme,” ISSCC, (doi: 10.1109/ISSCC.2016.7417942).
[16] Q. Chen, “Design and Optimization of Hybrid Cache based on SRAM and STT-MRAM,” [D].
[17] Z. Wang, et al.: “Adaptive placement and migration policy for an STT-RAM based hybrid cache,” HPCA(2014).
[18] J. Park et al.: “MH Cache: A Multi-retention STT-RAM-based Low-power Last-level Cache for Mobile Hardware Rendering Systems,” ACM Trans. Arch. Code Optimization
[19] N. Sayed, et al.: “Compiler-Assisted and Profiling-Based Analysis for Fast and Efficient STT-MRAM On-Chip Cache Design,” ACM Trans. Des. Auto. Electron. Sys. (2019)
[20] K. Korgaonkar et al.: “Density Tradeoffs of Non-Volatile Memory as a Replacement for SRAM Based Last Level Cache,” ISCA(2018), (doi: 10.1109/ISCA.2018.00035).
[21] R. Bera et al.: ”DSPatch: Dual Spatial Pattern Prefetchor,” Proc. 52nd Ann. IEEE/ACM Internat. Symp. Micro architecture(2019).
[22] Y. Ishii et al.: ”Access Map Pattern Matching for High Performance Data Cache Prefetch,” J. Inst.-Level Parall., (2011).
[23] E. Bhatia et al.: ”Perceptron-Based Prefetch Filtering,” ISCA(2019)
[24] P. Michaud et al., ”Best-offset hardware prefetching,” HPCA(2016), (doi: 10.1109/HPCA.2016.7446087).
[25] P. K. Komalan et al.: ”System level exploration of a STT-MRAM based level 1 data-cache,” DATE(2015)
[26] S. R. Nath et al.: ”Feedback Directed Prefetching: Improving the Performance and Bandwidth-Effectiveness of Hardware Prefetchers,” (2006)
[27] H. Wim et al., “Near-side prefetch throttling: adaptive prefetching for high-performance many-core processors,” PACT(2018).
[28] M. Ni et al., ”Write-awareness prefetching for non-volatile cache in energy-constrained IoT device,” IEICE ELEX, Vol. 19, No. 3, 2022
[29] N. Binkert et al., ”The gem5 simulator,” SIGARCH Comput. Archit. News, vol. 39, no. 2, (2011).
[30] SPEC CPU 2006 Benchmarks. http://www.spec.org/cpu2006.