Adaptive Tag Comparison for Hybrid Cache

Seong Jae Eom
Dept. of Electrical and Electronic Engineering
Yonsei University
Seoul, Republic of Korea
esj2804@yonsei.ac.kr

Eui-young Chung
Dept. of Electrical and Electronic Engineering
Yonsei University
Seoul, Republic of Korea
eychung@yonsei.ac.kr

Abstract— **This paper presents a proposal for an STT-RAM (Spin Transfer Torque Magnetic Random Access Memory) and SRAM (Static Random Access Memory) hybrid cache architecture and adaptive tag comparison for this architecture, considering the drawbacks of high-power consumption and the high cost of cache size expansion in conventional SRAM-based cache architectures for performance enhancement. Although the hybrid cache structure allows for an increase in memory size relative to the area, it has the drawback of increased tag comparison. The proposed adaptive tag comparison method involves dynamically changing the flag associated with the memory state and the value of threshold, which determines the memory state based on access count, in real-time to make decisions about tag comparison. By adopting the hybrid cache architecture, an average power reduction of 51.01% is achieved. And applying adaptive tag comparison, further power consumption is reduced by an average of 34.12% without degradation of performance, through an average reduction of 53.14% in tag comparison overhead compared to the hybrid cache scheme.**

Keywords; Hybird Cache; GPGPU; Adaptive Tag Comparison;

I. Introduction

In GPUs, the cache structure includes private L1 caches for each SM (streaming multiprocessor) and a shared L2 cache among the SMs. To improve the performance of GPUs, the exploration of cache capacity enhancement is underway. Since, increasing the size of the L1 cache in GPUs is costly, there is a need to explore changes in the structure of the L2 cache.

The L2 cache is designed as a set-associative cache, and simply increasing its size would result in longer tag array search times. Additionally, the L2 cache is composed of SRAM, which presents challenges in increasing its capacity due to area and cost constraints. Furthermore, increasing the SRAM capacity leads to higher leakage power, which affects data integrity. To mitigate this power consumption, the focus has shifted towards NVM (non-volatile memory) technologies, including PRAM and STT-RAM. Among these NVMs, STT-RAM emerges as a promising choice due to its durability and performance. However, it has disadvantages such as write penalty, meaning it incurs high energy and timing consumption during write operations compared to SRAM. Hence, a hybrid cache architecture combining SRAM and STT-RAM is proposed.

By integrating STT-RAM and SRAM, the cache size can be increased in proportion to the occupied area. While power consumption is reduced compared to employing SRAM alone, there are associated overheads related to enlarged cache search area and increased access time resulting from the augmented capacity. In previous research [1][2][3], performance improvement was achieved by reducing tag comparisons through a partial tag comparison technique tailored for SRAM cache or additional Bloom Filters. Hence, suitable partial tag comparison method is needed for hybrid cache architecture.

In this paper, adaptive tag comparison methods, not the partial tag comparison, tailored for hybrid cache are proposed to suit the characteristics of a hybrid cache and aim to enhance its power consumption.

II. Hybrid cache & Adaptive tag comparison

A. Hybrid cache scheme

Like previous research, the design of the hybrid cache architecture distinguishes SRAM and STT-RAM based on the way. The hybrid cache adopts a shared set methodology, allowing access to both SRAM and STT-RAM memories for read and write operations. The implemented hybrid cache showcases a 3.75MB L2 cache structure comprising 20 ways, with 4 ways allocated for SRAM and 16 ways for STT-RAM, presenting a capacity four times that of SRAM. And the allocation & migration method for hybrid cache scheme is shown as Figure 1.

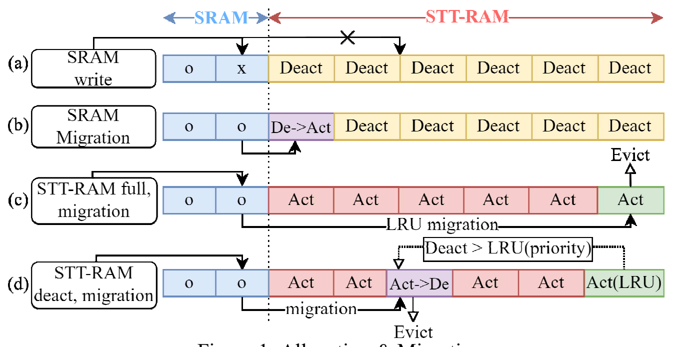

Figure 1. Allocation & Migration

B. Activate / Deactivate Algorithm

STT-RAM is distinguished by two states: activate and deactivate. Activate signifies the presence of valid data, while deactivate indicates the state of holding invalid data. This distinction is facilitated by the addition of a 1-bit flag in the tag array. Furthermore, a count bit is introduced for each way of STT-RAM to monitor the access frequency and dynamically adjust the flag's state accordingly. For each way, if the access count of a block surpasses a predetermined threshold value (Act_threshold), the corresponding block is deemed unnecessary for further read operations. As a result,

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-00754, Software Systems for AI Semiconductor Design)

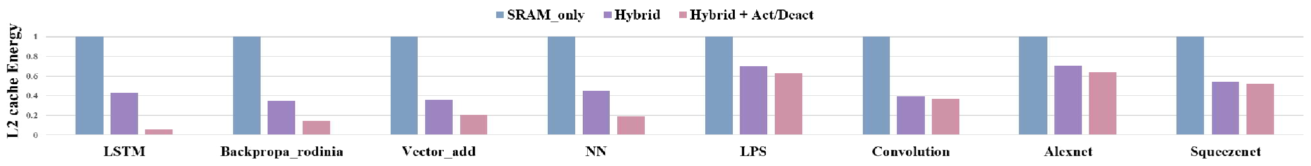

Figure 4. L2 cache energy (normalized to SRAM_only)

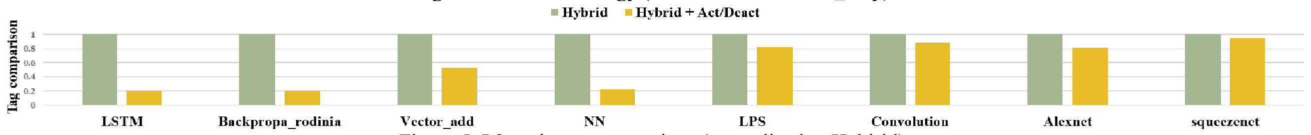

Figure 5. L2 cache tag comparison (normalized to Hybirid)

the flag is changed to Deactivate, the count is reset to 0, and the data is evicted. By utilizing the activate / deactivate flag, only the tag arrays of the activated STT-RAM blocks need to be accessed, reducing tag array comparison time and range as the cache size increases shown as Figure 2.

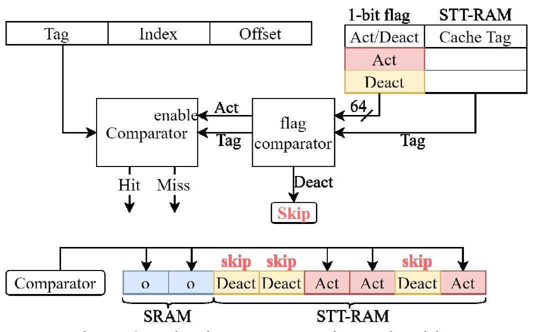

Figure 2. Adaptive tag comparison Algorithm

C. *Act_threshold Determination Algorithm*

To dynamically adjust the memory state according to the workload, it is necessary to modify the value of Act_threshold, which determines the memory state. Increasing the Act_threshold value extends the memory lifespan (maximum access count) of the corresponding set, delaying the deactivation time. Decreasing the Act_threshold value shortens the memory lifespan, accelerating the deactivation time.

Figure 3 provides an algorithm for determining the Act_threshold value. If the count of the accessed line exceeds Act_threshold, the line is evicted, and the count is reset. For hit or miss events, the Act_threshold value is incremented or decremented based on the number of deactivated memory lines.

Algorithm 1 Act_threshold determination algorithm

```
function threshold_determine(Tag_compare, line, set_index, STT_RAM_way)
    if Tag_compare == Hit then
        if deact_num[set_index] == 0 then
            Act_threshold[set_index] -= line->STT_RAM_count
        else
            Act_threshold[set_index] += line->STT_RAM_count
        line->STT_RAM_count = 1
    else //Miss
        if deact_num[set_index] > STT_RAM_way / 2 then
            Act_threshold[set_index] += Avg_STT_RAM_count
        else
            Act_threshold[set_index] -= Avg_STT_RAM_count
        line->STT_RAM_count += 1
end function
```

Figure 3. Act_threshold determination algorithm

III. EVALUATION

The hybrid cache structure was implemented using the GPGPU-sim [3], and the power consumption of the cache was measured using NVSIM [4]. The L2 cache has a size of 3.75MB, with 750KB allocated for SRAM and 3MB for STT-RAM, in a ratio of 1:4. In the case of an L2 cache composed solely of SRAM, it was configured to have the same size as the hybrid cache, which is 3.75KB. Figure 4 represents the normalized energy consumption of the L2 cache compared to the SRAM-only architecture. In the case of the hybrid cache, there is an average reduction of 51.01% in power consumption compared to SRAM-only. Additionally, the application of Act/Deact further reduces power consumption by an additional 15%. By applying Act/Deact and performing adaptive tag comparison, there is a 58.35% reduction in overall tag comparisons shown as Figure 5. It tends to exhibit a higher probability of STT-RAM deactivation when utilizing memory to an extent that migration to STT-RAM is not warranted or in cases of high cache miss rates which determines the value of Act_threshold. The proposed algorithm leads to a reduction in tag comparison proportional to the size of STT-RAM.

IV. CONCLUSION

In conclusion, this paper proposes adaptive tag comparison, allocation, and migration techniques suitable for the STT-RAM and SRAM hybrid cache scheme. Although there is a memory overhead (504 Bytes) corresponding to 0.02% of the L2 cache size for adaptive tag comparison, these techniques achieved a reduction of approximately 58% in tag comparisons, resulting in a power reduction of about 34% compared to the power consumption of the hybrid cache architecture. And compared to SRAM-based architecture, proposed algorithm saves 66% of L2 cache energy without any performance degradation.

REFERENCES

[1] Abed, Sa'ed, et al. "Hybrid approach based on partial tag comparison technique and search methods to improve cache performance." IET Computers & Digital Techniques 10.2 (2016): 69-76.

[2] Min, Rui, et al. "Partial tag comparison: A new technology for power-efficient set-associative cache designs." 17th International Conference on VLSI Design. Proceedings.. IEEE, 2004.

[3] Zhang, Jie, Myoungsoo Jung, and Mahmut Kandemir. "Fuse: Fusing stt-mram into gpus to alleviate off-chip memory access overheads." 2019 IEEE International Symposium on High Performance Computer Architecture (HPCA). IEEE, 2019.

[4] Bakhoda, Ali, et al. "Analyzing CUDA workloads using a detailed GPU simulator." 2009 IEEE international symposium on performance analysis of systems and software. IEEE, 2009.

[5] Dong, Xiangyu, et al. "Nvsim: A circuit-level performance, energy, and area model for emerging nonvolatile memory." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 31.7 (2012): 994-1007.