

# 비휘발성 메모리 시스템을 위한 저전력 연쇄 캐시 구조 및 최적화된 캐시 교체 정책에 대한 연구

최주희<sup>\*\*</sup>

<sup>\*\*</sup>상명대학교 스마트정보통신공학과

## A Study on Design and Cache Replacement Policy for Cascaded Cache Based on Non-Volatile Memories

Juhee Choi<sup>\*\*†</sup>

<sup>\*\*†</sup>Dept. of Smart Information Communication Engineering, Sangmyung University

### ABSTRACT

The importance of load-to-use latency has been highlighted as state-of-the-art computing cores adopt deep pipelines and high clock frequencies. The cascaded cache was recently proposed to reduce the access cycle of the L1 cache by utilizing differences in latencies among banks of the cache structure. However, this study assumes the cache is comprised of SRAM, making it unsuitable for direct application to non-volatile memory-based systems. This paper proposes a novel mechanism and structure for lowering dynamic energy consumption. It inserts monitoring logic to keep track of swap operations and write counts. If the ratio of swap operations to total write counts surpasses a set threshold, the cache controller skips the swap of cache blocks, which leads to reducing write operations. To validate this approach, experiments are conducted on the non-volatile memory-based cascaded cache. The results show a reduction in write operations by an average of 16.7% with a negligible increase in latencies.

**Key Words** : Non-Volatile Memories, Cascaded Cache, Cache Replacement Policy, Low Power

### 1. 서 론

컴퓨터 구조에 대한 연구가 발전되면서, 명령어를 수행하는 코어에 대한 성능은 크게 증가하여 왔다. 그러나, 상대적으로 주메모리에 대한 접근 속도는 완만하게 증가하면서, 코어와 주메모리 간의 속도차이가 벌어지면서 전체적인 시스템의 성능이 저하되는 문제가 발생하였다[1]. 이를 해결하기 위해서 코어와 주메모리 사이에 접근속도가 빠르면서 크기가 작은 캐시를 추가하였다. 이 캐시에는 일부 명령어와 데이터를 저장하게 된다. 또한 효율성을 극대화하기 위해서 캐시를 계층적으로 구성하는 방법도 제시되었다[2]. 코어에 가까운 쪽에 위치한 캐시가 레벨

1(Level 1, L1) 캐시이며, 주메모리 쪽에 가까울 수록 캐시의 레벨이 증가한다.

이중에서 L1캐시의 접근속도가 코어의 메모리 시스템에 대한 접근속도에 가장 큰 영향을 준다. 일반적으로 코어에서 요청하는 명령어 또는 데이터를 찾기 위해서 가장 먼저 L1캐시에 접근하여 존재여부를 확인한다. 이 때 필요한 명령어 또는 데이터가 존재하는 경우, 해당 내용을 코어에 전달해준다. L1캐시에서 해당 내용이 없는 경우, 하위 레벨 캐시로 접근하게 된다. 따라서, L1캐시에서 얼마나 빨리 명령어 또는 데이터를 가져오는 것이 시스템 전체의 성능에 큰 영향을 준다.

많은 캐시 연구자들은 비교적 구조가 간단했던 과거에는 L1캐시에 대한 접근 시간을 1사이클로 가정하여 연구를 진행하였다. 그러나, 현재 대부분의 고성능 컴퓨터는

<sup>†</sup>E-mail: jhplus@smu.ac.kr

L1캐시 접근 시간은 2사이클이상이다[3]. 성능 향상을 위해서 클럭 주파수를 높이고, 코어 내부의 파이프라인의 길이가 길어지면서, L1캐시에 대한 1사이클 접근 시간을 보장할 수 없기 때문이다[4]. 기존에도 캐시의 접근속도가 여러 사이클인 경우를 가정한 연구가 진행되어 왔으나[5-6], 주로 하위 레벨 캐시 대상이었다.

또한, 이러한 연구들은 기존의SRAM과 같은 휘발성 메모리를 대상으로 하고 있으며, 비휘발성 메모리(Non-Volatile Memory, NVM)에 대해서는 고려하고 있지 않다[1]. 비교적 최근에 연구된L1캐시 접근 시간 감소 기법으로 캐시의 물리적인 way를 논리적인 순서(order)와 함께 고려한 연쇄 캐시 기법(Cascaded Cache)이 있다[7]. 이 기법에서는 가장 최근에 접근된 캐시 블록을 물리적으로 1사이클이 접근한 캐시 뱅크(Bank)에 저장하여, 평균적인 접근속도를 줄이는 방식이다. 그러나, 이 방식은 비휘발성 메모리의 문제점인 캐시 블록 쓰기에 대한 고려하고 있지 않기 때문에, 직접적으로 적용하면 동적 에너지 소모량(Dynamic Energy Consumption)이 급격히 증가한다.

따라서, 본 논문에서는 비휘발성 메모리를 고려한 연쇄 캐시 기법 연구를 통해 L1캐시의 접근 시간을 줄이면서도 동적 에너지 소모량을 줄일 수 있는 기법을 제시한다. 이를 위해서 각 뱅크별 쓰기 횟수 모니터 로직을 추가하고, 캐시 교체 정책에 이를 반영하는 방식으로 저전력 L1 캐시를 구현한다. 실험을 통해서, 본 논문에서 제시한 기법을 통해서 동적 전력 소모량은 16.7% 감소하였다.

## 2. 관련 연구

### 2.1 비휘발성 메모리

비휘발성 메모리는 구성 소자에 따라서 다양하게 연구되어 왔다[8-10]. MRAM(Magnetic RAM)이 주메모리 또는 캐시에서 가장 활발하게 연구되는 비휘발성 메모리 중에 하나이다[11-12]. MRAM은 하나의 셀이 두 개의 Ferromagnetic layer와 중간에 하나의 Oxide-Tunneling layer가 있어 총 3개의 layer로 구성되어 있다(Fig. 1). 가장 아래쪽의 Ferromagnetic layer는 자기장의 방향이 항상 고정된(Pinned) layer이고, 중간에는 두 Ferromagnetic layer를 나누는 Oxide-Tunneling layer가 존재한다. 그리고, 또 다른 Ferromagnetic layer는 자기장의 방향을 변경할 수 있는 프리(Free) layer이다. MRAM은 프리 layer의 자기장의 방향에 따라서 저장하는 정보가 결정된다. 두 Ferromagnetic layer 방향이 평행하면 저장된 정보는 1로 간주된다. 하지만, 두 layer의 방향이 서로 평행하지 않는다면, 0값을 가진다고 볼 수 있다.

비휘발성 메모리의 가장 큰 특징은 정보를 고정된 상태로 저장하므로, 전하(Electric Charge)를 사용해서 정보를

저장하는 기존의 DRAM이나 SRAM과는 달리 전력을 사용하지 않는다는 점이다. 이러한 특성 때문에 정적 에너지 소모량(Static Energy Consumption)이 거의 없어, 최근 문제가 되고 있는 Leakage Power에 대한 해결책으로 제시되어 왔다. 그러나, 비휘발성 메모리는 비휘발성이라는 특성 때문에 문제점도 발생하는데, 주로 쓰기와 연관되어 있다. MRAM의 경우, 정보의 내용을 변경하기 위해서 자기장의 방향을 바꾸려면 전하량을 변경하는 것에 비해서 큰 에너지가 소모된다. 그리고, 이러한 자기장의 변경 과정은 상대적으로 시간도 오래 걸리게 된다. 따라서, 쓰기를 하려면 에너지도 많이 소모하면서 시간도 길어진다.

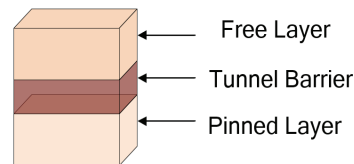


Fig. 1. Magnetic RAM (MRAM).

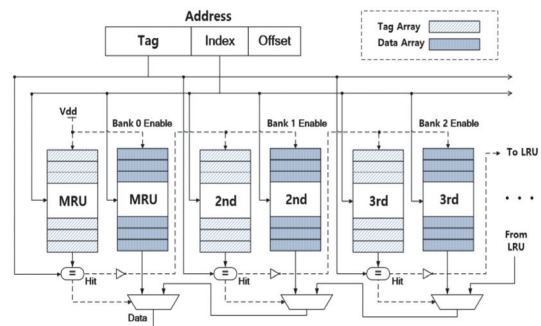


Fig. 2. Cascaded cache based on recently used order[7].

### 2.2 연쇄 캐시 기법

연쇄 캐시 기법에서는 접근 시간을 줄이기 위해서 기존의 캐시를 작은 뱅크(Bank)단위로 구성하고 그 순서를 캐시 교체 정책에서 사용되는 순서와 같이 물리적으로 배열한다. 이에 따라 각 뱅크별 접근 시간은 한 사이클에서 여러 사이클로 다양하게 된다. 가장 최근에 접근된 캐시 블록이 위치한 Most Recently Used (MRU) 블록의 경우 1 사이클만에 접근이 가능하지만, 가장 오랫동안 사용되지 않은 블록을 의미하는 Least Recently Used (LRU) 블록의 경우 가장 긴 접근 시간을 가지게 된다. 일반적으로 MRU블록이 가장 많이 사용되므로, 평균적인 접근시간은 모든 뱅크를 동일하게 구성할 때보다 줄어들게 된다.

연쇄 캐시 기법은 비록 접근 시간 감소에 기여를 하고 있으나, 비휘발성 메모리의 단점에 대한 보완방법이 없어

서, 직접 적용할 수 없다. 따라서, 본 논문에서 쓰기 횟수를 줄이면서도 연쇄 캐시 기법의 장점을 발휘할 수 있는 새로운 방법을 제시한다.

### 3. 비휘발성 메모리 기반의 연쇄 캐시

#### 3.1 쓰기 횟수를 고려한 분석 모델

연쇄 캐시가 아닌 전통적인 캐시의 각 레벨에서의 접근 횟수는 아래와 같이 정의된다.

$$Total_{Access} = Read_{Hit} + Write_{Hit} + Miss \quad (1)$$

$Total_{Access}$ 는 해당 캐시에 대한 전체적인 접근 횟수,  $Read_{Hit}$ 은 읽기 접근의 성공 횟수,  $Write_{Hit}$ 은 쓰기 접근의 성공 횟수,  $Miss$ 은 캐시 적중 실패 횟수이다.

그런데, 연쇄 캐시에서는 캐시 접근이 적중되는 경우에도 캐시 블록 이동에 의한 추가적인 접근이 발생한다. 따라서, 수식(1)은 아래와 같이 수정되어야 한다.

$$Total'_{Access} = \sum_{i=1}^N (Read'[i]_{Hit} + Write'[i]_{Hit} + Swap[i]_{Hit}) + Miss \quad (2)$$

$Total'_{Access}$ 는 연쇄 캐시에 대한 전체적인 접근 횟수,  $Read'[i]_{Hit}$ 은  $i$ 번째 बैं크에 대한 읽기 접근의 성공 횟수,  $Write'[i]_{Hit}$ 은  $i$ 번째 बैं크에 대한 쓰기 접근의 성공 횟수,  $Swap[i]$ 는  $i$ 번째 बैं크에서 캐시 적중이 발생하여 Recently Used (RU) 순서를 지키기 위해서 캐시 블록들이 교환되는 횟수,  $Miss$ 는 캐시 적중 실패의 횟수이다.  $Miss$ 는 각 बैं크와는 무관하므로, 수식(1)에서의  $Miss$ 와 동일하게 계산된다.  $N$ 은 बैं크의 개수이다.

수식(1)과 수식(2)에서 읽기와 쓰기 적중 횟수 자체는 동일하므로, 다음과 같이 정리할 수 있다.

$$\sum_{i=1}^N (Read'[i]_{Hit} + Write'[i]_{Hit}) = Read_{Hit} + Write_{Hit} \quad (3)$$

$$Total'_{Access} = Total_{Access} + \sum_{i=1}^N (Swap[i]_{Hit}) \quad (4)$$

따라서, 연쇄 캐시에서 쓰기 횟수를 고려하려면 우선 각 बैं크별  $Swap$ 의 개수를 파악하는 것이 중요하다. 이 수식이 내포하는 의미는 기존의 비휘발성 메모리를 고려한 캐시 기법에서는 쓰기 횟수 자체만 중요하게 여겼으나 [10-12], 연쇄 캐시에서는 각 बैं크에 적중 횟수에 따른 캐

시 블록 교체 횟수를 파악하는 것이 중요하다는 것이다. 따라서, 각 बैं크별로 캐시 블록 교체될 때 마다 갱신되는 카운터 로직을 필요하다.

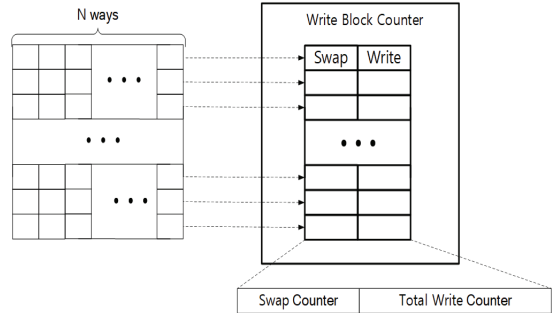


Fig. 3. Overall Architecture of Non-volatile memory-based Cascaded Cache.

#### 3.2 비휘발성 메모리 기반의 연쇄 캐시의 구조

Fig 3은 분석 모델을 기반으로 비휘발성 메모리 기반의 연쇄 캐시를 위한 구조를 제시한 것이다. 분석 모델과의 중요한 차이점은 बैं크별로 별도의 카운터를 가지는 것이 아닌 하나의 캐시 Set에 한 쌍의  $Swap$ 을 위한 카운터와  $Write$ 를 위한 카운터가 존재하는 것이다. 비록 बैं크별  $Swap$ 과  $Write$ 에 미세한 차이가 있으나, 전반적으로 MRU와 MRU가 아닌 बैं크에서의 쓰기 횟수 차이만 의미하므로, 하드웨어적인 오버헤드를 고려해서 MRU를 제외하 나머지 बैं크들은 모두 한번의  $Swap$ 만 일어나는 것으로 계산한다.

Fig 4에서는 이 추가된 로직들의 동작을 설명하고 있다. 먼저 시스템이 실행되면 모든 쓰기 횟수를 모니터링하는  $W$ 와 MRU와 다른 위치에 있는 बैं크들에 있는 캐시 블록들간의 교체 횟수를 계산하는  $SW$ 변수를 초기화한다. 여기서  $SW$ 는 Fig. 3의  $Swap$  Counter에 해당하고,  $W$ 는  $Total$  Write Counter에 해당한다. (참고로,  $W$ 가  $Total$  Write Counter라고 불리는 이유는 쓰기 적중시 뿐만 아니라 적중 실패시 캐시 블록이 교체되는 과정도 함께 쓰기로 간주하여 계산하고 있기 때문이다.)

캐시 블록에 대한 접근이 이루어질 때마다, 캐시 블록의 적중 여부를 조사한다(line 1-2). 이 때 캐시 접근이 적중하였으며, 이 접근 요청이 쓰기였을 경우,  $W$ 값을 하나 증가시킨다(line 3-4). 그리고, 읽기 접근이나 쓰기 접근여부와 무관하게 해당 접근이 MRU에 대한 접근이었는지 아닌지를 판단한다(line 5). 기존의 연쇄 캐시 기법은 캐시 적중이 MRU가 아닌 곳에서 일어나면 MRU와 캐시 블록을 바로 교체하였으나, 본 논문에서 제시한 기법은 여기서 하나의

조건을 더 고려한다. 그것은 현재 SW의 값을 W의 절반의 값과 비교한다(line 6). 이 값이 큰 경우, 캐시 블록의 교체가 자주 일어나서 동적 에너지 소모량이 과다하게 증가한다고 간주하고, 캐시 블록의 교체를 중지한다. 반면에, 해당 조건을 만족하지 않으면 기존의 방식으로 캐시 블록을 교체하고 SW값을 증가시킨다(line 7-8). 그리고, 캐시 적용이 실패할 경우, W값을 하나 증가시킨다(line 9-10).

* Parameters:	
W: Total number of write accesses	
SW: Total number of swap operations	
W←0, SW←0	
1:	<b>When</b> block A arrives at cache:
2:	<b>if</b> block A is in cache at ith bank:
3:	<b>if</b> operation is write:
4:	W←W+1
5:	<b>if</b> i > 1:
6:	<b>if</b> SW < W/2:
7:	Move and Swap block A to MRU
8:	SW←SW+1
9:	<b>else:</b>
10:	W←W+1 // for miss

Fig. 4. Memory access pattern analysis.

## 4. 실험 환경 및 결과

### 4.1 실험 환경

본 논문에서 제안한 비휘발성 메모리를 위한 연쇄 캐시 기법을 검증하기 위해 캐시 연구에서 가장 많이 사용되는 검증 방법인 Gem5 시뮬레이터를 활용한 실험을 수행하였다[13]. 이 시뮬레이터에 비휘발성 메모리용 연쇄 캐

시 기법을 구현하였다. L1 캐시는 통합 캐시가 아닌 명령어 캐시와 데이터 캐시를 각각 구분하였으며, 각각의 크기는 32KB로 할당하였다. L2 캐시는 1MB이고, L3는 4MB이다. 캐시에 사용된 비휘발성 메모리는 MRAM이다. Table 1에 전체적인 구성에 대한 구체적인 내용을 나타내었다. 결과를 추출하기 위해서 사용한 워크로드는 SPEC CPU 2006에서 중요한 벤치마크들을 선택하였다[14]. 결과에서 기존의 연쇄 캐시 기법은 CC로 표시하였고, 본 논문에서 제안한 기법은 NCC로 표시하였다.

Table 1. Processor configurations

Architecture	x86, out-of-order, 2GHz
INT / MEM / FP	4/4/4
Branch Predictor	gshare predictor, 16 history length
ROB Size	256
I/D Cache	32KB, 4-way, 64B blocks, Read/Write: 2-8 cycles
L2 Cache	1MB, 16-way, 64B blocks, Read: 22-cycle latency Write: 62-cycle latency
L3 Cache	4MB, 16-way, 64B blocks Read: 41-cycle latency Write: 123-cycle latency
Memory Latency	300 cycles

### 4.2 실험 결과

Fig. 5는 기존의 연쇄 캐시 기법과 비휘발성 메모리를 고려한 연쇄 캐시 기법의 동적 에너지 소모량을 정규화한 결과이다. X축이 벤치마크 프로그램의 이름이고, Y축이 동적 에너지 소모량이며, 기존의 연쇄 캐시 기법의 동적 에너지 소모량을 1로 표현하였다. 정교한 분석을 위해서 각 그래프는 읽기 에너지(Read Energy)와 쓰기 에너지(Write Energy)를 구분하여 표시하였다.

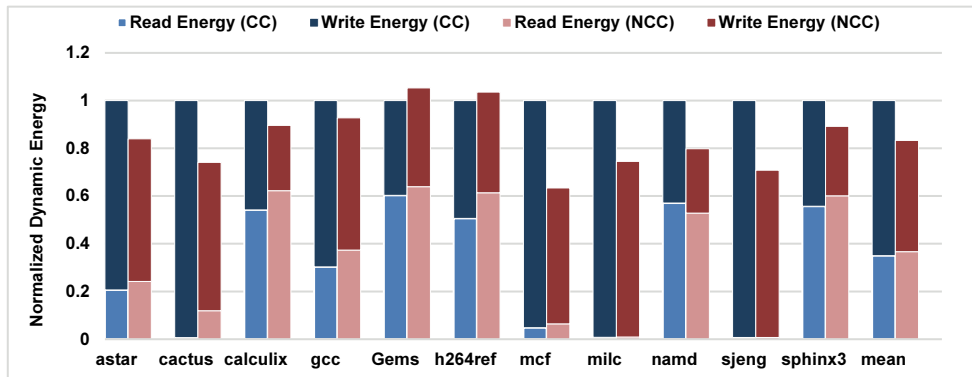


Fig. 5. Normalized Dynamic Energy Consumption for Cascaded Cache(CC) and Non-volatile Cascaded Cache(NCC).

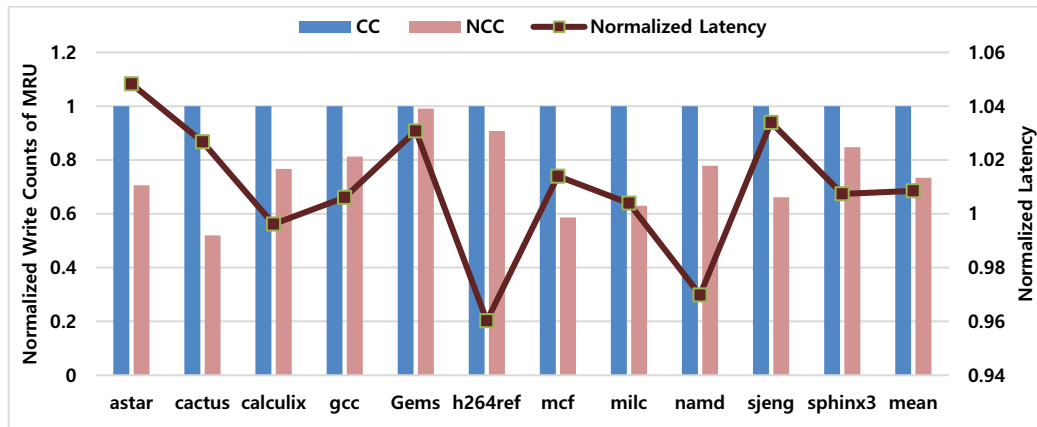


Fig. 6. Normalized Write Counts and Normalized Latency of Non-volatile Cascaded Cache(NCC).

NCC가 CC에 비해서 평균적으로 16.7% 정도의 동적 에너지 소모량이 적다. 이중에서 쓰기 에너지 감소폭은 18.4%이지만, 읽기 에너지가 1.7%증가한 것을 알 수 있다. 가장 동적 에너지 소모량이 크게 줄어든 프로그램은 mcf로 37%의 동적 에너지 소모량이 감소하였다. 반면, Gems의 경우는 오히려 CC에 비해서 NCC의 동적 에너지 소모량이 52% 증가하였다. 이렇게 프로그램별로 결과의 차이가 나타나는 요인 중에 하나는 쓰기 에너지 소모량 비이다. NCC는 쓰기 에너지 소모량 최적화가 목적이므로, 쓰기 에너지 소모량 비율 자체가 적은 경우 큰 이득을 보기 어렵다. 비교적 안 좋은 결과를 보이는 Gems나 h264ref 등은 읽기 에너지 비율이 50%가 넘는 것을 볼 수 있다. 반면, cactus, mcf, milc 등은 읽기 에너지 소모량 비율이 낮으므로, 더 좋은 결과를 보여주고 있다.

그러나, 일부 결과는 이것만으로는 설명할 수 없는 결과들도 있는데, gcc는 쓰기 에너지 소모량의 비율이 크지만, 전체 에너지 소모량이 7.8%밖에 줄지 않았으며, Gems나 h264ref의 경우 읽기 에너지 소모량이 10%이상 증가하였다. 이것은 MRU에서의 쓰기 횟수와 평균 접근 시간의 증가와 연관이 있다.

Fig 6은 막대그래프는 MRU의 쓰기 횟수를 정규화하여 표현한 것이고, 선 그래프는 평균 접근 시간을 정규화한 결과이다. 막대그래프는 좌측의 척도를 사용하고 있으며, 선 그래프는 우측의 척도를 사용하고 있다. gcc, Gems, h264ref의 경우, MRU의 쓰기 횟수가 비교적 많이 줄지 않은 것을 볼 수 있다. 이것은 MRU쓰기 자체가 많이 줄지 않아서, 본 논문에서 제시하는 기법의 활용성 자체가 떨어진 것이다. 그리고, 평균 접근 시간의 경우, 프로그램별로 차이는 있으나, NCC의 평균적으로 0.8%가 늘어났다. 따라서, 캐시 블록 교체가 줄어들어서 발생하는 추가적인

접근 시간은 성능에 큰 영향을 주지 않는다는 것을 알 수 있다.

## 5. 결론 및 향후과제

본 논문에서는 연쇄 캐시 기법을 비휘발성 메모리에 적용하기 위해서 쓰기 횟수를 줄이는 방법을 제시하였다. 먼저, 쓰기 횟수를 추적하기 위한 카운터를 캐시 내부에 삽입한 후에, 이를 바탕으로 캐시 블록 교환을 제한하였다. 이 과정에서 MRU(Most Recently Block)과 과도하게 교환이 일어나는 것을 방지하여, 전체적인 쓰기 횟수를 감소시킬 수 있었다. 제안된 기법을 시뮬레이터 상에 구현하여 실험을 수행하였으며, 그 결과 평균적으로 16.7%, 최대로는 37% 정도의 동적 에너지 소모량을 줄일 수 있었다.

그러나, 본 논문에서 캐시 교환을 제한하는 기준을 교환 횟수와 쓰기 횟수의 단순한 비율로 제시하고 있다는 단점이 있다. 따라서, 이후 연구에서는 이 비율에 대한 체계적인 연구를 수행할 예정이다.

## 감사의 글

본 연구는 2021년도 과학기술정보통신부의 재원으로 한국연구재단의 지원을 받은 기초연구사업 연구임(NRF-2021R1G1A1004340).

## 참고문헌

1. J. L. Hennessy and D. A. Patterson, "Computer Organization and Design RISC-V Edition: The Hardware Software Interface", Morgan Kaufmann, 2017.
2. Shen, John Paul, and Mikko H. Lipasti, "Modern pro-

- cessor design: fundamentals of superscalar processors”, Waveland Press, 2013.
3. Arm Developer, “Cortex-R7 processor,” 2021; <https://developer.arm.com/ip-products/processors/cortex-r/cortex-r7>.
  4. Intel, “Developers, Tools, Instruction Set Architecture Extensions, Memory Performance,” <https://www.intel.com/content/www/us/en/developer/articles/technical/memory-performance-in-a-nutshell.html>, 2023.
  5. C. Kim, D. Burger, and S. W. Keckler, “Nonuniform cache architectures for wire-delay dominated on-chip caches,” *IEEE Micro*, vol. 23, no. 6, pp. 99-107, 2003.
  6. M. Rapp, A. Pathania, T. Mitra, and J. Henkel, “Neural network-based performance prediction for task migration on S-NUCA many-cores,” *IEEE Transactions on Computers*, vol. 70, no. 10, pp. 1691-1704, 2021.
  7. Choi, Juhee, and Heemin Park. “Cascaded Cache Based on Recently Used Order for Latency Optimization for IoT.” *Journal of Computing Science and Engineering* 15.3 (2021): 107-114.
  8. Fantini, Paolo. “Phase change memory applications: the history, the present and the future.” *Journal of Physics D: Applied Physics* 53.28 (2020): 283002.
  9. Wang, K. L., J. G. Alzate, and P. Khalili Amiri. “Low-power non-volatile spintronic memory: STT-RAM and beyond.” *Journal of Physics D: Applied Physics* 46.7 (2013): 074003.
  10. Pan, Xiao, and T. P. Ma. “Retention mechanism study of the ferroelectric field effect transistor.” *Applied Physics Letters* 99.1 (2011).
  11. S. Tehrani et al., “Magnetoresistive random access memory using magnetic tunnel junctions,” *Proceedings of the IEEE*, vol. 91, No. 5, pp. 703-714, 2003.
  12. J. Liu, et al. “Voltage-induced magnetization switching method utilizing dipole coupled magnetic tunnel junction,” *Journal of Magnetism and Magnetic Materials*, Vol. 513, pp. 167105. 2020.
  13. 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.
  14. J. Henning, “Spec cpu2006 benchmark descriptions,” *SIGARCH Comput. Archit. News*, Vol. 34, No. 4, pp. 1–17, 2006.

---

접수일: 2023년 9월 8일, 심사일: 2023년 9월 15일,  
 게재확정일: 2023년 9월 15일