

다중 메모리로 구성된 저장장치에서 데이터 탐색 비용을 줄이기 위한 메모리 매핑 기법

이현섭*

백석대학교 첨단IT학부 교수

A Memory Mapping Technique to Reduce Data Retrieval Cost in the Storage Consisting of Multi Memories

Hyun-Seob Lee*

Professor, Division of Advanced IT, Baekseok University

요약 최근 메모리 기술의 급격한 발전으로 개발되고 있는 다양한 종류의 메모리는 데이터 관리 시스템에서 처리 속도 향상을 위해 활용되고 있다. 특히 NAND 플래시 메모리는 전원이 차단되어도 데이터를 유지할 수 있는 비휘발성 특성이 있으므로 메모리 기반 저장장치의 데이터 저장용 주요 미디어로 활용되고 있다. 그러나 최근 연구되고 있는 메모리 기반 저장장치는 NAND 플래시 메모리뿐만 아니라 MRAM과 PRAM 등 다양한 종류의 메모리로 구성되어 있고 추가로 새로운 특성이 있는 다양한 종류의 메모리가 개발되고 있다. 따라서 특성이 서로 다른 이종의 메모리들로 구성된 저장 시스템에서 미디어의 데이터 처리 성능과 효율 향상을 위한 메모리 관리 기술의 연구가 필요하다. 본 논문에서는 데이터 관리를 위해 다양한 메모리로 구성된 저장장치에서 데이터를 효율적으로 관리하기 위한 메모리 사상 기법을 제안한다. 제안하는 아이디어는 서로 다른 이종 메모리를 하나의 사상 테이블을 활용하여 관리하는 방법이다. 이 방법은 데이터의 주소 체계를 통일할 수 있고 데이터 티어링(tiering)을 위해 서로 다른 메모리에 분할 저장된 데이터의 탐색 비용을 감소시킬 수 있다.

주제어 : 메모리, 데이터 티어링, 저장 시스템, 데이터 관리 정책, 사상 테이블

Abstract Recently, with the recent rapid development of memory technology, various types of memory are developed and are used to improve processing speed in data management systems. In particular, NAND flash memory is used as a main media for storing data in memory-based storage devices because it has a nonvolatile characteristic that it can maintain data even at the power off state. However, since the recently studied memory-based storage device consists of various types of memory such as MRAM and PRAM as well as NAND flash memory, research on memory management technology is needed to improve data processing performance and efficiency of media in a storage system composed of different types of memories. In this paper, we propose a memory mapping scheme thought technique for efficiently managing data in the storage device composed of various memories for data management. The proposed idea is a method of managing different memories using a single mapping table. This method can unify the address scheme of data and reduce the search cost of data stored in different memories for data tiering.

Key Words : memory, data tiering, storage system, data management scheme, mapping table

*이 논문은 2022학년도 백석대학교 학술연구비 지원을 받아 작성되었음

*교신저자 : 이현섭(hyunseob@bu.ac.kr)

접수일 2022년 12월 16일

수정일 2023년 1월 17일

심사완료일 2023년 1월 19일

1. 서론

최근 메모리 기술의 급격한 발전으로 다양한 종류의 메모리가 개발되고 있다. 메모리는 외부 충격에 상대적으로 강하며 저전력 소비로 고속의 데이터 저장이 가능하여서 이동 장비에서 대용량 엔터프라이즈 저장 시스템까지 다양한 저장장치의 데이터 저장 및 관리를 위해 활용되고 있다. 특히 SSD(solid state disk)는 저비용과 높은 집적도(degree of integration)의 양산성이 있는 NAND 플래시 메모리를 저장 미디어로 사용하고 있다[1-4]. 그러나 플래시 메모리의 쓰기 전 지우기(erase-before-write) 특징과 비대칭의 단위로 데이터 쓰기 지우기 연산을 처리해야 하는 독특한 특징이 있다. 그리고 최근에는 이러한 특성을 개선하기 위한 다양한 메모리들이 개발되고 있다[5, 6]. 따라서 저장 시스템에서 데이터를 효율적으로 관리하기 위해 NAND 플래시 메모리뿐만 아니라 DRAM과 SRAM처럼 다양한 용도로 새로운 특징을 가지고 있는 메모리들의 활용이 연구되어야 한다. 본 논문에서는 데이터 관리를 위해 다양한 메모리로 구성된 저장장치에서 데이터를 효율적으로 관리하기 위한 메모리 사상 기법을 제안한다. 제안하는 아이디어는 서로 다른 이종 메모리를 하나의 사상 테이블을 활용하여 관리하는 방법이다. 이 방법은 데이터의 주소 체계를 통일하여 데이터를 효율적으로 관리할 수 있다. 마지막으로 실험을 통해 제안하는 아이디어가 DRAM 버퍼의 탐색 비용을 감소시키는 장점이 있음을 증명한다.

2. 배경 및 문제점

2.1 메모리의 개발과 발전

메모리의 발전은 휘발성 메모리인 SRAM과 DRAM 그리고 비휘발성 메모리인 NOR와 NAND 플래시 메모리의 한계를 넘어 지속해서 발전해 오고 있다. 대표적인 메모리는 PRAM, MRAM, FeRAM, 3D-Xpoint 메모리가 있다.

MRAM은 자성 소자를 이용한 비휘발성 메모리이다. 각 셀의 저항 변화를 이용하여 데이터를 구분하므로 지속적인 리프레시(Refresh)동작 없이 데이터를 유지할 수 있는 특징이 있다. 또한, NAND 플래시 메모리와 비교하여 속도가 빠르고 전력 소비가 적은 장점이 있다. 그러나 자성을 이용한 읽기 쓰기 동작을 수행하기 때문에 집적도가 높아질수록 반도체 내부에서 인접 셀 간의 간섭이

발생할 수 있는 한계가 있다[7, 8].

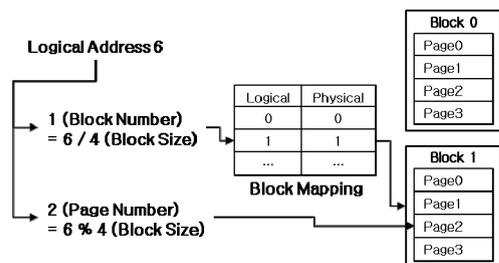
PRAM은 셀의 상 변화를 이용하여 데이터를 저장한다. 결정과 비결정 상태에 따라 다른 저항값을 갖는 특징이 있는 GST(GeSbTe)에 전류를 인가하여 600도 이상의 열을 발생시킨 후 저항값에 따라 발생하는 전압의 차이 구분하여 데이터를 식별한다. 비휘발성이라는 특징과 빠른 데이터 접근 속도의 특징이 있으나 데이터 처리를 위해 발생시켜야 하는 발열 문제와 셀을 많이 사용할수록 저항 비가 상승하여 고집적도 메모리 양산이 어려운 한계가 있다[9, 10].

FeRAM은 전압을 가하면 물질 내부에 분극이 유지되는 특징이 있는 강유전체(ferroelectric)를 이용하여 셀을 설계하였다. 또한, 직렬방식의 데이터 전송 구조로 되어있는 SRAM과 같이 고속으로 1bit 쓰기 동작이 가능한 비휘발성 메모리의 특징이 있다[11, 12].

3D-Xpoint는 메모리를 제어하는 워드 라인과 컬럼 라인을 교차하는 위치에 셀을 배치하여 다층으로 쌓은 구조로 되어있다. 또한, 비휘발성 메모리이면서 NAND 플래시 메모리보다 내구성이 우수하고 응답시간은 DRAM만큼 우수한 특징이 있다. 그러나 고비용의 양산 문제가 있다[13, 14].

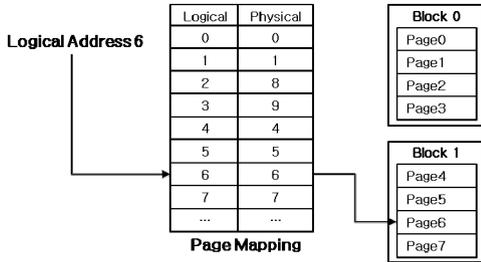
2.2 NAND 플래시 메모리의 매핑 기법

최근 저장장치의 주요 미디어로 사용되고 있는 NAND 플래시 메모리의 칩 구조는 여러 개의 블록으로 구성되어 있고 각 블록은 여러 개의 페이지를 포함하고 있다. 그러나 일반적인 저장 미디어와 비교하여 읽기/쓰기는 페이지 단위로 처리되는 반면 지우기는 블록 단위로 처리되는 특징이 있다. 따라서 이러한 특성을 고려하지 않으면 불규칙한 성능 저하 등 여러 문제를 일으킬 수 있다. 이러한 문제를 해결하기 위해 적용된 FTL(flash transfer layer)은 매핑 기법을 활용하여 NAND 플래시 메모리의 특성을 숨긴다.



[Fig. 1] Block Mapping Algorithm

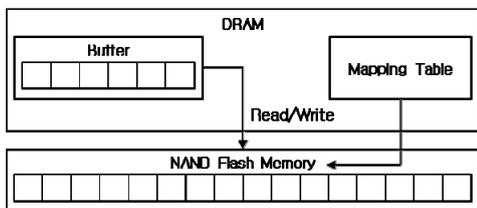
Fig. 1은 블록 매핑 알고리즘을 보여주고 있다. 블록 매핑 알고리즘은 블록 단위로 논리적인 주소와 물리적인 주소를 매핑한다. 그리고 논리적인 블록의 주소는 블록 크기로 나누는 몫을 통해 식별하고 블록 내의 페이지 위치는 블록 크기로 모드 연산한 오프셋(offset) 결과를 통해 식별한다. 그림의 예에서 논리적 주소 6의 논리적 블록 번호는 블록 크기 4로 나누는 연산을 통해 1이라는 것을 확인할 수 있다. 그리고 논리적 블록 1은 물리적 블록 1에 매핑 되어있다. 그다음 블록 내의 페이지 오프셋은 연산을 통해 2임을 확인할 수 있다. 블록 매핑 방법은 논리적인 주소를 물리적인 주소로 변환하여 NAND 플래시 메모리의 특성을 감추는 효과가 있으나 데이터가 저장된 페이지에 접근하기 위해 추가적인 연산을 수행해야 하는 단점이 있다.



[Fig. 2] Page Mapping Algorithm

Fig. 2는 페이지 매핑 알고리즘을 보여주고 있다. 페이지 매핑은 모든 논리적 주소에 대한 물리적 페이지 주소를 매핑한다. 그림의 예에서 논리적 주소 6은 물리적 페이지 6에 매핑 되어있는 것을 확인할 수 있다. 따라서 추가적인 연산 없이 매핑이 되어있는 페이지에 빠르게 접근하는 장점이 있다. 그러나 페이지 매핑은 모든 페이지 매핑 정보를 유지해야 하므로 메모리 자원을 많이 소비하는 단점이 있다.

2.3 다중 메모리 저장장치에서 탐색 비용



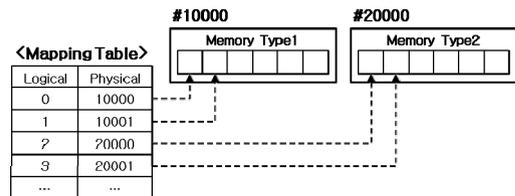
[Fig. 3] Example of Multi Memories

Fig. 3는 독립된 관리체계를 가지고 있는 DRAM과 NAND 플래시 메모리로 구성된 저장장치의 구조를 보여주고 있다. 그림의 예에서는 사용자의 데이터를 NAND 플래시 메모리에 관리한다. 그리고 DRAM에서는 NAND 플래시 메모리와 호스트 간에 전송 속도를 극복하기 위한 버퍼와 FTL의 매핑 테이블을 유지하고 있다. Fig. 3의 예제와 같이 NAND 플래시 메모리는 비휘발성 특징과 용량 대비 가격의 우수성 때문에 대용량의 저장 미디어로 활용된다. 대용량 저장장치에서 FTL의 매핑 테이블은 데이터 탐색을 위해 적극적으로 활용되고 있다. 반면 휘발성 특징과 데이터 접근 속도가 빠른 DRAM은 소량의 데이터를 유지할 수 있는 버퍼 공간을 사용하므로 탐색을 위한 별도의 관리체계 없이 순차적인 탐색을 통해 데이터를 탐색한다. 현재까지의 저장장치 설계에서는 소량의 고속 메모리를 탐색하는 비용이 많이 들지 않았으나 메모리가 다양화되고 있는 기술 발전 환경에서 새로운 타입의 대용량 메모리를 적용할 경우 별도의 관리체계 없이 메모리를 순차적으로 탐색하는 것은 비효율성을 증가시킨다. 또한, 탐색 속도 향상을 위해 별도의 관리체계를 추가하는 것은 비용을 증가시키는 문제를 발생시킬 수 있다.

3. 이중 메모리 매핑 기법

3.1 핵심 아이디어

본 논문에서는 서로 다른 타입의 다중 메모리로 구성된 저장장치에서 하나의 매핑 테이블을 이용하여 메모리를 매핑하는 기법을 소개한다. 이 방법은 속도 차이가 있는 이중 메모리를 버퍼로 활용할 때 탐색 비용을 줄이는 장점이 있다.



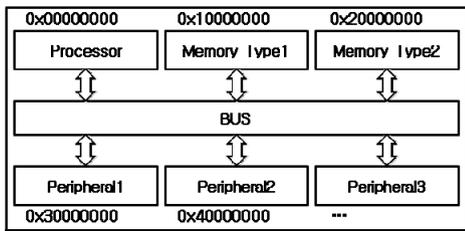
[Fig. 4] Key Idea

Fig. 4는 제안하는 핵심 아이디어를 보여주고 있다. 그림에서는 두 개의 서로 다른 메모리 타입의 칩이 10000번 주소와 20000 주소에 위치에 있다. 그리고 하

나의 매핑 테이블에서 각각의 메모리를 구분 없이 매핑하고 있다. 예를 들어 논리적 주소 0번과 1번의 데이터는 메모리 타입1의 10000번 주소와 10001번 주소에 매핑되어 있다. 그리고 논리적 주소 2번과 3번의 데이터는 메모리 타입2의 20001번 주소와 20001번 주소에 매핑되어 있다. 이 방법은 다중 메모리로 구성된 저장장치에서 메모리 관리를 일원화하여 단순화시키는 장점이 있다.

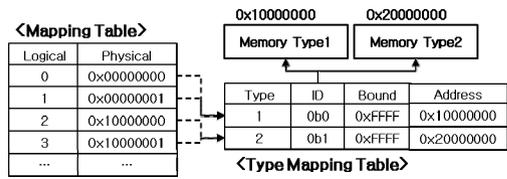
3.2 이종 메모리 타입 식별 정책

본 논문에서 제안하는 방법은 서로 다른 이종 메모리를 하나의 매핑 테이블에서 관리하는 방법이다. 그러나 이종 메모리는 특성이 달라서 메모리를 구분하여 관리하기 위한 메모리 타입 식별 방법이 필요하다. 본 절에서는 주소를 기반으로 메모리를 구분하는 정책을 소개한다.



[Fig. 5] Example of Memory Map

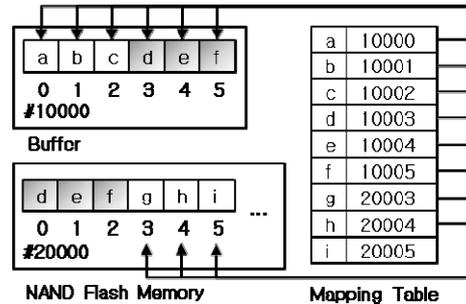
Fig. 5는 메모리 기반 저장장치에서 사용하는 메모리 맵의 간략한 예제를 보여주고 있다. 일반적으로 저장장치의 각 모듈은 AXI(advanced extensible interface) 버스를 중심으로 연결되어 있다. 그리고 이러한 모듈은 버스에서 각각의 식별을 위해 고유의 절대적 주소를 가지고 있다. 그러나 각 모듈의 상대적 주소는 절대적 주소와 관계없이 0번지에서 시작한다. 예를 들어 각 메모리 칩의 블록과 페이지들은 0번 블록의 0번 페이지에서 시작한다. 그리고 일반적으로 단일 메모리를 관리하는 매핑 테이블은 메모리 칩이 연결된 상대적 순서에 따라 블록의 주소와 페이지 번호를 식별하여 매핑한다. 이러한 매핑 테이블의 주소 체계는 절대적 주소가 다른 다양한 메모리 칩을 매핑했을 때 각각의 메모리 타입과 칩을 식별하는데 혼란을 준다. 따라서 각각의 절대적 주소를 가지고 있는 메모리 칩과 칩에서 관리하는 메모리 영역을 1차로 매핑하고, 이 매핑 정보를 메모리 매핑 테이블의 상위 비트에 연동하는 기법을 설계하였다.



[Fig. 6] 1st Mapping Method on Chips

Fig. 6은 서로 다른 64KB의 이종의 메모리를 0x10000000번지와 0x20000000번지에 구성된 저장장치에서 칩 간에 1차 매핑을 하고 매핑한 ID를 이용하여 메모리의 매핑 정보를 구분하는 방법을 보여주고 있다. 그림의 예제에서는 1차 매핑에서 0x10000000번지의 메모리 타입1을 0번 ID로 매핑하고 0x20000000번지의 메모리 타입2를 1번 ID로 매핑하였다. 그리고 각 ID는 2진수로 변환하여 매핑 테이블의 물리적 주소에서 사용하지 않는 상위 비트에 적용하였다. 예를 들어 논리적 주소 0과 1번 페이지는 메모리 타입 1번의 물리적 주소 0과 1번 페이지에 매핑 되어있고, 논리적 주소 2와 3번 페이지는 메모리 타입 2번의 물리적 주소 0과 1번 페이지에 매핑이 되어있는 것을 의미한다.

3.3 버퍼 탐색 속도 향상을 위한 응용



[Fig. 7] Example of Performance Improvement

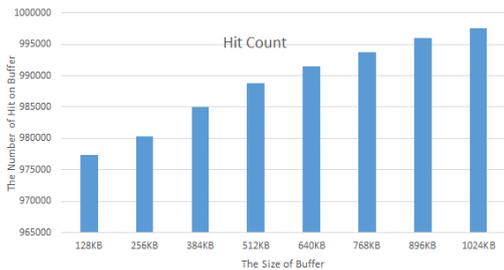
Fig. 7은 서로 특성이 다른 DRAM 버퍼와 NAND 플래시 메모리로 구성된 저장 시스템에서 이종 메모리 매핑 방법을 적용한 예제를 보여주고 있다. 예제에서는 NAND 플래시 메모리의 0, 1, 2, 3, 4, 5번 주소에 d, e, f, g, h, i, 데이터가 저장되어 있다. 그리고 버퍼에는 a, b, c, d, e, f의 데이터가 적재되어 있다. 이중 d, e, f의 데이터는 버퍼에서 최신 데이터를 유지하는 중복된 데이터이다. 만약 버퍼를 관리하기 위한 별도의 체계가 없는 경우 버퍼에서는 순차 탐색을 이용하여 데이터를

찾는다. 즉, d, e, f의 데이터를 찾기 위해 각각 4, 5, 6회의 탐색이 필요하다. 그리고 만약 탐색 성능을 향상하기 위해 별도의 관리체계를 추가할 경우 추가적인 비용이 필요하다. 그러나 제안하는 아이디어는 NAND 플래시 메모리를 관리하기 위한 매핑테이블에서 추가로 버퍼를 관리할 수 있으므로 오버헤드 없이 다른 타입의 메모리를 식별할 수 있는 주소 체계를 유지할 수 있고, 버퍼의 탐색 비용을 줄일 수 있다.

4. 실험 및 평가

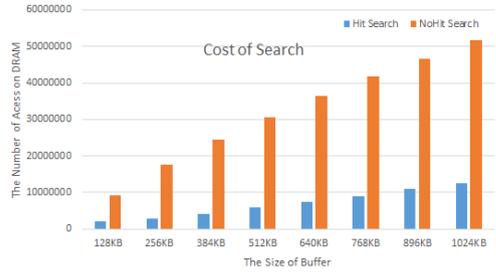
본 논문에서는 이종의 메모리를 하나의 매핑 테이블로 관리하는 방법을 제안했다. 제안한 아이디어를 버퍼 관리 정책에 적용했을 때 얻을 수 있는 장점을 분석하기 위해 시뮬레이션을 통해 실험 평가하였다. 시뮬레이션 실험은 버퍼의 크기를 128KB에서 1MB까지 증가하며 약 100만 개의 트레이스 데이터를 LRU(least recently used)[15, 16]로 관리했을 때 발생하는 버퍼 접근 횟수와 읽기 성능을 측정 및 분석하였다.

Fig. 8은 버퍼를 128KB에서 1MB까지 증가해 가며 트레이스 데이터를 LRU에서 관리했을 때 발생한 정중 횟수이다. 버퍼의 크기가 증가할수록 적재할 수 있는 데이터의 수가 증가하여 전반적으로 적중횟수가 증가하는 것을 확인할 수 있었고, 128KB에서는 977,391회 1M에서는 997,586회 적중하였다.



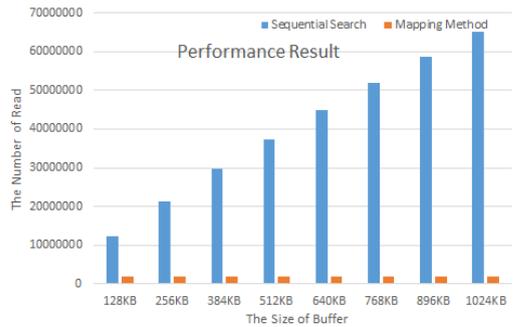
[Fig. 8] Hit Count

Fig. 9는 버퍼에서 적중된 데이터와 적중되지 않은 데이터를 탐색할 때 발생한 탐색 비용을 분석한 결과이다. 전반적으로 버퍼에 적중된 데이터보다 적중되지 않은 데이터를 탐색하는 비용이 약 75~84% 높았다. 128KB에서 2,158,303, 9,174,609회의 메모리 접근을 시도했고, 1M에서 12,456,804회, 51,739,950회 접근을 시도했다.



[Fig. 9] Cost of Search

Fig. 10은 매핑 기법이 순차적인 탐색과 비교하여 발생한 탐색 성능 차이를 보여주고 있다. 그림에서 보여주는 결과와 같이 매핑 기법은 일정한 성능을 보여주고 있으나 순차 탐색 기법은 버퍼의 크기가 증가할수록 탐색 비용이 증가하였다. 전반적으로 약 92~98%의 성능 차이가 있었고 128KB에서는 매핑 기법이 92.06%, 1MB에서는 98.46%의 성능향상 효과가 있었다.



[Fig. 10] Performance Result

5. 결론

본 논문에서는 데이터 관리를 위해 다양한 메모리로 구성된 저장장치에서 데이터를 효율적으로 관리하기 위한 메모리 사상 기법을 제안하였다. 제안한 아이디어는 서로 다른 이종 메모리를 하나의 사상 테이블을 활용하여 관리하는 방법이다. 이 방법은 데이터의 주소 체계를 통일하여 자원의 추가 없이 데이터를 효율적으로 관리할 수 있다. 또한, 실험을 통해 서로 다른 메모리를 버퍼와 저장 미디어로 분할 관리할 때 저장된 데이터의 탐색 비용을 약 90%이상 감소시킬 수 있음을 증명하였다. 제안하는 방법은 다양한 타입의 메모리를 적용하여 연구 중인 차세대 저장장치에서 메모리를 효율적으로 관리할 수

있을 것으로 기대된다. 향후 예는 분할된 메모리를 저 자원으로 관리하는 기법뿐만 아니라 분할된 메모리를 효과적으로 관리하기 위한 연구를 진행할 예정이다.

REFERENCES

- [1] H.S.Lee, "A Prediction-Based Data Read Ahead Policy using Decision Tree for improving the performance of NAND flash memory based storage devices," *The Korea Internet of Things Society*, Vol.8, No.4, pp.9-15, 2022.
- [2] H.S.Lee, "A Safety IO Throttling Method Inducting Differential End of Life to Improving the Reliability of Big Data Maintenance in the SSD based RAID," *The Society of Digital Policy & Management*, Vol.20, No.5, pp.593-598, 2022.
- [3] H.S.Lee, "Performance analysis and prediction through various over-provision on NAND flash memory based storage," *The Society of Digital Policy & Management*, Vol.20, No.3, pp.343-348, 2022.
- [4] H.S.Lee, "A method for optimizing lifetime prediction of a storage device using the frequency of occurrence of defects in NAND flash memory," *The Korea Internet of Things Society*, Vol.7, No.4, pp.9-14, 2021.
- [5] K.Parat and A.Goda, "Scaling Trends in NAND Flash", *2018 IEEE International Electron Devices Meeting (IEDM)*, pp.211-214, 2018.
- [6] G.H.Lee, S.M.Hwang, J.S.Yu and H.J.Kim, "Architecture and Process Integration Overview of 3D NAND Flash Technologies," *Open Access* Vo.11, No.15, p.6703, 2021.
- [7] T.Na, S.H.Kang and S.O.Jung, "STT-MRAM Sensing: A Review," *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol.68, No.1, pp.12-18, 2021.
- [8] Q.Shao, Z.Wang and J.J.Yang, "Efficient AI with MRAM," *Nature Electronics*, Vol.5 pp.67-68, 2022.
- [9] K.W.Lee, H.K.Park and S.O.Jung, "Adaptive Sensing Voltage Modulation Technique in Cross-Point OTS-PRAM," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol.29, No.4, pp.631-642, 2021.
- [10] H.K.Park, T.H.Choi, H.K.Ahn and S.O.Jung, "Thermoelectric Cooling Read for Resolving Read Disturb With Inrush Current Issue in OTS-PRAM," *IEEE Transactions on Nanotechnology*, Vol.18, pp.421-431, 2019.
- [11] Y.C.Luo, J.Hur, Z.Wang, W.Shim, A.I.Khan and S.Yu, "A Technology Path for Scaling Embedded FeRAM to 28 nm and Beyond With 2T1C Structure," *IEEE Transactions on Electron Devices*, Vol.69, No.1, pp.109-114, 2022.
- [12] S.C.Chang and U.E.Avci, "Hafnium-based FeRAM for Next-generation High-speed and High-Density Embedded Memory," *2022 IEEE Silicon Nanoelectronics Workshop (SNW)*, pp.1-2, 2022.
- [13] A.A.Abud, G.L.Miotto and R.Sipos, "Experience and Performance of Persistent Memory for the DUNE Data Acquisition System," *IEEE Transactions on Nuclear Science*, Vol.68, No.8, pp.2159-2164, 2021.
- [14] J.Guo, H.Wang, W.Jing, H.Li, Y.Du, Z.Song and B.Chen, "Subsystem under 3D-Storage Class Memory on a chip," *Computers & Electrical Engineering*, Vo.74, pp.47-58, 2019.
- [15] A.A.Titinch and N.Halasa, "FPGA implementation of simplified Fuzzy LRU replacement algorithm," *16th International Multi-Conference on Systems, Signals & Devices (SSD)*, pp.657-662, 2019.
- [16] Q.Zheng, T.Yang, Y.Kan, X.Tan, J.Yang, and X.Jiang, "On the Analysis of Cache Invalidation With LRU Replacement," *IEEE Transactions on Parallel and Distributed Systems*, Vol.33, No.3, pp.654-666, 2022.

이 현 섭(Hyun-Seob Lee)

[종신회원]



- 2013년 2월 : 한양대학교 컴퓨터 공학과 (공학 박사)
- 2012년 3월 ~ 2021년 2월 : 삼성전자 책임연구원
- 2021년 3월 ~ 현재 : 백석대학교 첨단IT학부 조교수

<관심분야>

인공지능, 저장시스템, 임베디드 시스템