



병렬 TLC STT-MRAM 기반 대용량 LLC 설계

박태진*, 장우영**

Large-Scale Last-Level Cache Design Based on Parallel TLC STT-MRAM

Taejin Park*, Wooyoung Jang**

이 논문은 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (NRF-2017R1D1A1B03036353).

요약

최신의 고성능 멀티-코어 프로세서들은 대용량 LLC(Last-Level Cache)를 요구하고 있다. 차세대 메모리들 중의 하나인 TLC(Triple-Level Cell) STT(Spin-Transfer Torque)-MRAM(Magnetic Random Access Memory)는 높은 저장 밀도를 제공할 수 있지만, 다단계 읽기와 쓰기 동작으로 긴 메모리 지연과 높은 전력 소모를 유발한다. 본 논문에서는 병렬 TLC STT-MRAM으로 구성되는 LLC의 구조와 동작을 제안한다. 우리의 LLC는 셀 분할 매핑과 조건 블록 스와핑 기법들로 3-단계 읽기와 쓰기 발생을 최소화한다. 실험적으로, 제안된 LLC는 MARSSx86에서 두 개의 PARSEC 벤치마크들을 동시에 수행할 때, 직접 매핑 기법이 적용된 LLC보다 사이클 당 명령어 처리 횟수를 평균 11.03%, 전력 소모를 평균 12.54% 향상시킨다.

Abstract

State-of-the-art high-performance multi-core processors demand a large-scale last-level cache (LLC). Triple-level cell (TLC) spin-transfer torque (STT)-magnetic random access memory (MRAM), one of the emerging memories, can provide high storage density, but cause long memory latency and high memory power consumption due to multi-step read and write operations. In this paper, the architecture and operation of an LLC comprised of a parallel TLC STT-MRAM are proposed. Our LLC minimizes the occurrence of the three-step read and write operations via cell division mapping and conditional block swapping techniques. Experimental results show our proposed LLC achieves 11.03% more instructions per cycle and 12.54% lower power consumption, on average, than LLCs using direct mapping technique in the case that two PARSEC benchmarks are concurrently performed in MARSSx86.

Keywords

non-volatile memory, STT-MRAM, triple-level cell, last-level cache

* Dankook University School of Electronic and Electrical Engineering

- ORCID: <http://orcid.org/0000-0003-2040-5884>

** Dankook University School of Electronic and Electrical Engineering

- ORCID: <http://orcid.org/0000-0002-7262-0028>

• Received: Nov. 09, 2017, Revised: Dec. 01, 2017, Accepted: Dec. 04, 2017

• Corresponding Author: Wooyoung Jang

Dept. of Electronics and Electrical Engineering, Dankook University,
Yongin-si, Geonggi-do, Korea,

Tel.: +82-31-8005-3620, Email: wjjang@dankook.ac.kr

1. 서 론

최신의 컴퓨터 시스템들은 병렬 처리를 통한 성능 향상과 저전력 소모를 위하여 멀티-코어 프로세서를 사용하고 있다[1]. 그 코어들은 다양한 캐시(Cache)들을 계층적으로 내재하고 있다. 그러한 캐시들 중에서 LLC(Last-Level Cache)는 다수의 코어들에 의해 공유되기 때문에, 저장 용량이 충분하지 않으면 LLC의 효율이 크게 저하된다[2]. 이러한 문제점을 개선하기 위해서 멀티-코어 프로세서의 성능을 향상시키기 위한 대용량 온-칩 캐시들이 연구되고 있다.

최신의 멀티-코어 프로세서에서 SRAM(Static Random Access Memory)로 구현된 캐시들은 전체 컴퓨터 시스템의 성능, 비용, 전력 소모에 많은 영향을 주고 있다. 계속되는 반도체 미세 공정 발전으로 캐시의 저장 용량이 증가하여 성능은 향상되어 왔으나, 신뢰도와 전력 소모는 크게 악화되고 있다. 따라서, STT-MRAM(Spin Transfer Torque Magnetic Random Access Memory)[3][4], ReRAM(Resistive Random Access Memory)[5], PRAM(Phase-Change Random Access Memory)[6]와 같은 차세대 메모리들이 광범위하게 연구되고 있다. 그 중에서 MTJ(Magnetic Tunneling Junction)에 데이터를 저장하는 STT-MRAM은 SRAM을 대체할 수 있는 유력한 후보들 중의 하나로서 주목받고 있다. STT-MRAM은 SRAM과 비슷한 읽기 지연, SRAM보다 높은 저장 용량, 낮은 전력 소모를 제공할 뿐만 아니라, 전원을 공급하지 않아도 데이터가 지워지지 않는다. 또한, STT-MRAM은 다른 차세대 메모리들보다 우수한 성능과 낮은 전력 소모, 높은 내구성을 보여준다[7]. 하지만, STT-MRAM는 다른 차세대 메모리들보다 낮은 저장 밀도를 가지므로, 대용량 LLC에서 STT-MRAM의 사용을 주저하게 만든다. 따라서 STT-MRAM의 저장 밀도를 향상시켜, 차세대 멀티-코어 프로세서에서 요구하는 대용량, 고성능, 저전력 캐시를 구현하기 위한 연구들이 활발히 진행되고 있다.

최근에 플래시 메모리처럼 STT-MRAM 셀(Cell)에 다수의 비트(Bit)를 저장하는 기술들이 연구되고

있다[8]-[11]. MLC(Multi-Level Cell) STT-MRAM은 2-비트를 하나의 메모리 셀에 저장할 수 있고, TLC(Triple-Level Cell) STT-MRAM은 3-비트를 하나의 메모리 셀에 저장할 수 있다. MLC STT-MRAM을 LLC로 이용하기 위한 연구는 활발히 진행되었지만, 더 많은 코어들이 집적되어 LLC를 공유하게 될 것으로 예상됨에 따라, LLC의 끊임없는 저장 용량 증가를 요구하고 있다[8]-[10][12][13]. 따라서 본 논문에서는 병렬 TLC STT-MRAM을 기반으로 하는 LLC의 구조와 동작을 연구한다. LLC는 병렬 TLC STT-MRAM을 사용하여 매우 높은 저장 밀도를 제공할 수 있지만, 각 LLC 블록이 동일한 병렬 TLC STT-MRAM 셀의 3-비트를 모두 포함하면, 복잡한 3-단계의 읽기와 쓰기가 수행되어 LLC의 성능과 전력 소모가 악화된다.

제안하는 병렬 TLC STT-MRAM 기반 LLC는 셀 분할 매핑(Cell Division Mapping)과 조건 블록 스와핑(Conditional Block Swapping) 기법들을 적용하여 3-단계의 읽기와 쓰기 발생을 최소화한다. 셀 분할 매핑 기법은 각 LLC 블록이 병렬 TLC MTJ들의 동일한 영역으로 구성되게 한다. 병렬 TLC MTJ들의 하드 영역으로 구성된 LLC 블록은 읽기 지연이 작고, 쓰기 지연이 큰 반면에, 소프트 영역으로 구성된 LLC 블록은 읽기 지연이 크고, 쓰기 지연이 작다. 미디움 영역으로 구성된 LLC 블록의 읽기와 쓰기 지연은 하드와 소프트 영역으로 구성된 블록의 읽기와 쓰기 지연들의 사이에 있다. 셀 분할 매핑 기법이 적용된 LLC에서 데이터가 하드 영역으로 구성된 블록에 자주 저장되거나, 데이터가 소프트 영역으로 구성된 블록으로부터 자주 읽게 되면, LLC의 성능 저하와 전력 소모가 크게 증가될 수 있다. 제안하는 조건 블록 스와핑 기법은 데이터를 쓸 때, LLC 블록을 구성하는 영역들의 특징과 LLC 블록의 참조 빈도에 따라, LLC 블록들의 데이터를 서로 교환한다. 그 결과로 병렬 TLC MTJ의 소프트 영역으로 구성된 블록에서 읽기와 하드 영역으로 구성된 블록에서 쓰기 발생을 최소화한다. 이러한 셀 분할 매핑과 조건 블록 스와핑 기법들을 이용한 병렬 TLC STT-MRAM 기반 LLC는 메모리 접근 시간과 전력 소모를 크게 줄일 수 있다.

이 논문의 나머지 부분은 다음과 같이 구성된다. 2장에서는 STT-MRAM의 SLC와 TLC의 기본 동작과 STT-MRAM을 활용한 관련 기존 연구들을 논의한다. 3장에서는 제안하는 병렬 TLC STT-MRAM 기반 LLC 구조의 동작을 설명한다. 4장에서는 구현 방법을 설명하고, 실험결과를 분석한다. 마지막으로, 5장에서는 본 논문의 결론을 도출한다.

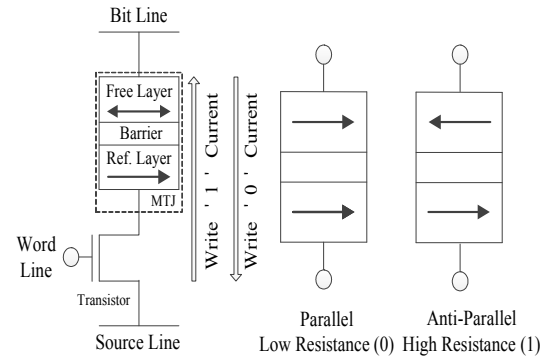
II. STT-MRAM의 기본 동작과 관련 연구

2.1 STT-MRAM의 기본 동작

그림 1(a)는 SLC STT-MRAM 셀의 구조를 보여준다[4]. 그 메모리 셀은 비트 라인과 소스 라인에 연결되고, 워드 라인에 연결된 트랜지스터에 의해 제어된다. MTJ는 자유 층과 참조 층이라 불리는 두 개의 강자성 물질이 산화물 장벽에 의해 분리된 구조를 가지고 있다. 참조 층의 자화 방향은 고정되어 있으며, 자유 층의 자화 방향은 스핀 분극 전류(Spin Polarized Current)에 의해 변경될 수 있다. 그림 1(b)는 MTJ의 두 가지 저항 상태를 보여준다. SLC MTJ의 참조 층과 자유 층의 자화 방향이 동일할 때, MTJ는 낮은 저항 상태를 가지며, 논리적 '0'을 나타낸다. 반면에, 두 층의 자화 방향이 서로 다를 때, MTJ는 높은 저항 상태를 가지며, 논리적 '1'을 나타낸다.

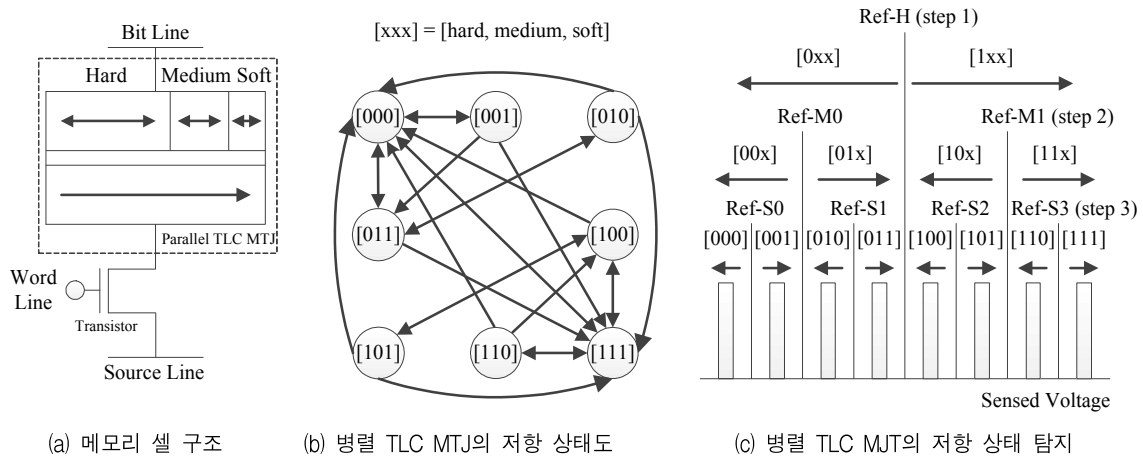
을 나타낸다.

그림 2(a)는 8개의 저항 상태들을 표현할 수 있는 병렬 TLC STT-MRAM 셀의 구조를 보여준다[14]. 병렬 TLC MTJ는 자유 층이 세 부분으로 나뉘져 있어서, 다양한 저항 상태를 표현할 수 있다. 세 부분은 소프트 영역, 미디움 영역, 그리고 하드 영역이라고 불리고, 각 영역은 1-비트 데이터를 저장할 수 있다. 병렬 TLC MTJ는 하나의 트랜지스터에 의해 제어되고, 세 종류의 전류가 하드, 미디움, 그리고 소프트 영역을 모두 통과하게 된다.



(a) 메모리 셀 구조 (b) MTJ의 병렬 및 비-병렬 상태들
그림 1. SLC STT-MRAM

Fig. 1. SLC STT-MRAM, (a) Memory cell structure, (b) Parallel and anti-parallel states of MTJ



(a) 메모리 셀 구조 (b) 병렬 TLC MTJ의 저항 상태도 (c) 병렬 TLC MTJ의 저항 상태 탐지

그림 2 병렬 TLC STT-MRAM

Fig. 2 Parallel TLC STT-MRAM, (a) Memory cell structure, (b) Resistive state diagram of parallel TLC MTJ, (c) Resistive state detection of parallel TLC MTJ

하드 영역의 자화 방향을 변환하는 전류는 미디움 영역의 자화 방향을 변환하는 전류보다 크고, 미디움 영역의 자화 방향을 변환하는 전류는 소프트 영역의 자화 방향을 변환하는 전류보다 크다. 그래서 병렬 TLC MTJ에서 하드 영역의 자화 방향이 변환되면, 미디움과 소프트 영역들의 자화 방향이 하드 영역과 동일하게 변환된다. 그리고, 미디움 영역의 자화 방향이 변환되는 경우에는 소프트 영역들의 자화 방향이 미디움 영역과 동일하게 변환된다.

그림 2(b)는 병렬 TLC MTJ의 저항 상태도 (Resistive state diagram)를 보여준다. 병렬 TLC MTJ 저항 상태 변환은 현재의 저항 상태와 최종 저항 상태에 의해 좌우된다. 3-단계 쓰기 동작은 병렬 TLC MTJ의 저항 상태가 $[0xx] \rightarrow [101]$ 와 $[1xx] \rightarrow [010]$ 로 변환되는 경우에 발생된다. 여기서, $[hms]$ 은 하드 영역의 논리 값이 h , 미디움 영역의 논리 값이 m , 그리고, 소프트 영역의 논리 값이 s 를 의미하고, x 는 무정의(Don't Care)를 의미한다. 3-단계 쓰기 동작에서는 우선 하드 영역에 쓰기 동작을 수행하여, 저항 상태를 $[0xx] \rightarrow [111]$ 와 $[1xx] \rightarrow [000]$ 으로 변환한다. 다음으로, 미디움 영역에 쓰기 동작을 수행하여, 저항 상태를 $[111] \rightarrow [100]$ 와 $[000] \rightarrow [011]$ 로 변환하고, 마지막으로, 소프트 영역에서 쓰기 동작을 수행하여, 저항 상태를 $[100] \rightarrow [101]$ 와 $[011] \rightarrow [010]$ 으로 변환한다. 2-단계 쓰기 동작은 병렬 TLC MTJ의 저항 상태가 $[0xx] \rightarrow [100]$, $[1xx] \rightarrow [011]$, $[0x0] \rightarrow [110]$, $[1x1] \rightarrow [001]$, $[h0x] \rightarrow [h10]$, 그리고 $[h1x] \rightarrow [h01]$ 로 변환되는 경우에 수행된다. 나머지 저항 상태 변환은 1-단계 쓰기 동작으로 수행될 수 있다.

그림 2(c)는 병렬 TLC MTJ의 저항 상태 탐지를 보여준다. 병렬 TLC MTJ에 저장된 3-비트 데이터를 모두 읽기 위하여 3-단계의 탐지 동작을 필요로 한다. 탐지 동작에서 입력 감지 전류에 의해 병렬 TLC MTJ에 전압이 유도되고, 그 전압은 7개 중에 3개의 레퍼런스 전압들과 비교된다. 감지 증폭기는 먼저 Ref-H와 전압을 비교하여 하드 영역의 저항 상태를 탐지하고, 그 결과에 따라 Ref-M0 또는 Ref-M1과 전압을 비교하여 미디움 영역의 저항 상태를 탐지한다. 마지막으로 하드와 미디움 영역들의 저항 상태 탐지 결과에 따라 Ref-S0, Ref-S1, Ref-S2, 또는 Ref-S3과 전압을 비교하여 소프트 영역의 저

항 상태를 탐지한다.

2.2 관련 연구

STT-MRAM들의 많은 장점에도 불구하고 쓰기 동작에서 증가된 지연과 높은 전력 소모는 해결해야 할 과제이다. 이것들은 STT-MRAM을 온-칩 캐시로 구현하는 것을 방해한다. 이 문제를 해결하기 위하여, 캐시 블록에 저장된 데이터와 저장하려는 데이터가 같을 경우 조기 종료하는 기법이 제안되었으며[15], 저장되는 데이터가 '0'일 경우, 쓰기를 생략하는 방법 등이 제안되었다[16]. 또한, SRAM과 STT-MRAM을 조합한 혼성 (Hybrid) 캐시 구조가 제안되었다[17][18].

MLC STT-MRAM은 높은 저장 밀도에 많은 주목을 받았지만, 2-단계 읽기와 쓰기로 긴 지연이 발생한다. [8]에서는 병렬 MLC STT-MRAM에서 자주 참조되는 데이터에 따라서 하드 영역으로 구성된 캐시 블록과 소프트 영역으로 구성된 캐시 블록을 이주시킨다. [9]에서는 직렬 MLC STT-MRAM에서 자주 참조되는 데이터가 하드 영역으로 구성된 캐시 블록에 저장되어 있으면, 소프트 영역으로 이주하고, 응용 프로그램의 특성에 따라 소프트 영역으로만 캐시 블록을 구성하였다. [11]에서는 응용 프로그램에 따라 64-바이트 캐시 블록과 32-바이트 캐시 블록을 구현하였다. 32-바이트 캐시 블록은 소프트 영역만을 사용하여 지연 시간과 전력 소모의 문제를 해결하려 하였다. [12]에서는 MLC MTJ에서 발생할 수 있는 내구성 문제를 해결하려 하였다. [13]에서는 2개의 MTJ 주변에 자화 방향을 갖는 자벽 자석(Domain wall magnet)을 구성하여 읽기와 쓰기 지연 시간을 해결하려 하였다. [19]에서는 MLC MTJ에서 에러 발생률이 가장 높은 저항 상태를 사용하지 않고, 3개의 상태만 사용하는 방법이 제시되었다. [20]에서는 멀티미디어 응용 프로그램을 위한 MLC STT-MRAM 제어기가 제안되었다.

미세공정의 발전으로 반도체 집적도가 점점 더 향상됨에 따라, 프로세서들은 더 많은 코어들을 내장하게 될 것이다. 이 코어들은 LLC를 공유하게 되므로, 더 큰 저장 용량의 LLC가 요구되고 있다. 최근에 TLC STT-MRAM이 개발되었지만, LLC로 사

용되기 위해서는 복잡한 쓰기과 읽기로 유발된 문제들을 해결해야 한다[14]. 따라서 본 논문에서는 병렬 TLC STT-MRAM 기반 고성능, 저전력, 대용량 LLC를 제안한다.

III. 병렬 TLC STT-MRAM 기반 LLC

본 논문에서는 병렬 TLC STT-MRAM 셀을 이용하여 대용량 LLC를 제안한다. 제안하는 LLC는 셀 분할 매핑 기법으로 구조되고, 조건 블록 스와핑 기법으로 관리된다.

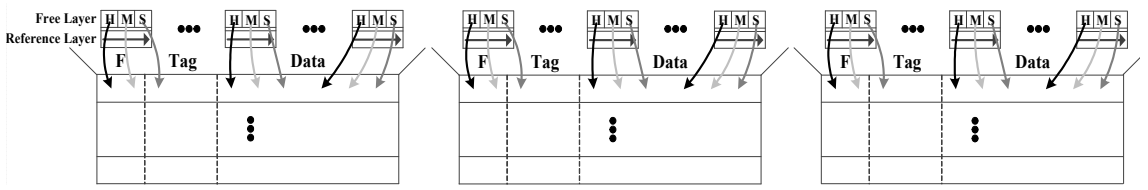
3.1 셀 분할 매핑 기법

그림 3은 두 종류의 3-웨이 집합-연관(Set-Associative) LLC들을 보여준다. 그 LLC들은 저장 용량 확장을 통해 캐시 적중을 향상시키기 위하여 SRAM 대신에 병렬 TLC STT-MRAM으로 설계되었다. 각 LLC 블록은 신호(Flag), 태그(Tag), 캐시 데이터를 저장한다.

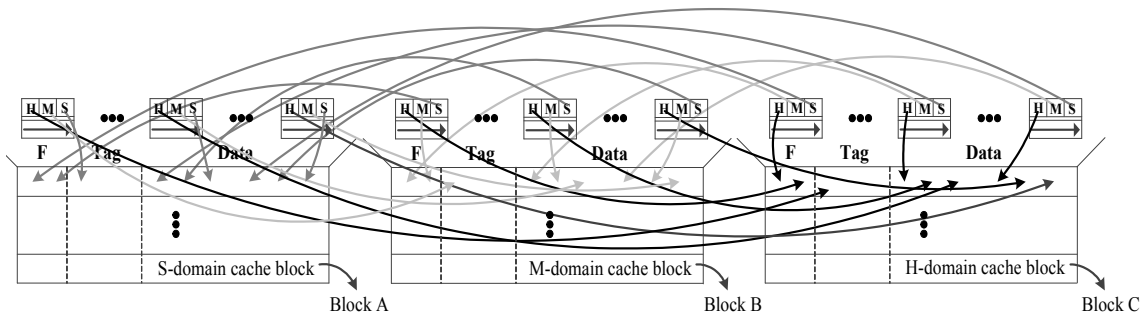
그림 3(a)는 직접 매핑 기법이 적용된 LLC를 보여준다. 각 LLC 블록은 병렬 TLC MTJ들의 하드

영역, 미디움 영역, 소프트 영역으로 구성되어 있다. 그러한 LLC 블록에 데이터를 쓸 때, 그 블록을 구성하는 각 병렬 TLC MTJ의 모든 영역의 저항 상태가 변환될 수 있다. 따라서 각 병렬 TLC MTJ의 현재 저항 상태에서 목표 저항 상태로 수행될 저항 상태 변환이 결정되어 수행된다. 직접 매핑 기법이 적용된 LLC 블록을 읽을 경우에, 그 LLC 블록에 구성하는 모든 병렬 TLC MTJ들의 저항 상태가 탐지되어야 한다. 각 병렬 TLC MTJ의 현재 저항 상태에 관계없이, LLC 블록을 구성하는 모든 병렬 TLC MTJ들에서 항상 3-단계 저항 상태 탐지가 수행된다.

직접 매핑 기법이 적용된 LLC의 성능과 전력 소모는 병렬 TLC MTJ의 복잡한 저항 상태 변환과 탐지에 의해 크게 저하될 수 있다. LLC 블록 쓰기 지연은 그 블록에 구성하는 모든 병렬 TLC MTJ들 중에서 최대 지연되는 것에 의해 좌우된다. 즉, LLC 블록에서 다수의 병렬 TLC MTJ들의 저항 상태가 1-단계 또는 2-단계로 변환될 수 있더라도, 어떤 병렬 TLC MTJ의 저항 상태가 3-단계로 변환된다면, 그 블록의 전체 쓰기 지연은 3-단계 저항 상태 변환을 수행하는 시간과 같다.



(a) 직접 매핑 기법을 적용한 LLC
(a) LLC using direct mapping technique



(b) 셀 분할 매핑 기법을 적용한 제안된 LLC
(b) Proposed LLC using cell division technique

그림 3. 3-웨이 집합-연관 LLC들
Fig. 3. 3-way set-associative LLCs

반면에, LLC 블록 읽기는 항상 병렬 TLC MTJ의 3-단계 저항 상태 탐지 시간과 같다. 따라서 이러한 병렬 TLC MTJ의 복잡한 저항 상태 변환과 탐지는 LLC의 성능을 저하시킬 수 있다. 또한, 블록에 접근할 때, 병렬 TLC MTJ의 다단계 쓰기 또는 읽기 동작이 자주 발생되어 전력 소모도 증가할 수 있다.

그림 3(b)는 제안된 셀 분할 매핑 기법이 적용된 LLC를 보여준다. 제안된 LLC에서 각 블록은 병렬 TLC MTJ들의 하드, 미디움, 소프트 영역들 중 동일한 영역으로 구성된다. 따라서 그러한 LLC 블록에서 데이터를 접근할 때, 병렬 TLC MTJ들의 모든 영역에서 쓰기 또는 읽기가 수행되는 직접 매핑 기법과 다르게, 셀 분할 매핑 기법에서는 병렬 TLC MTJ들의 일부 영역에서만 쓰기 또는 읽기 동작이 수행될 수 있다. 각 LLC 웨이는 동일한 병렬 TLC MTJ 영역으로 구성된 LLC 블록들로 구성된다.

제안된 LLC에서 각 블록은 병렬 TLC MTJ들의 동일한 영역으로 구성되기 때문에, 선택된 블록을 구성하는 병렬 TLC MTJ 영역에 의해 쓰기 동작들이 결정된다. 예를 들어, 그림 3(b)에서 데이터가 갱신, 또는 저장되는 LLC 블록이 A라고 하자. 블록 A는 병렬 TLC MTJ들의 소프트 영역으로 구성되어 있기 때문에, 소프트 영역에 1-단계 저항 상태로 LLC 블록 쓰기가 수행될 수 있다. 소프트 영역을 변환시키는 전류는 미디움과 하드 영역을 변환시키지 못한다. 만약 병렬 TLC MTJ들의 미디움 영역으로 구성된 블록 B가 선택된다면, 2-단계 저항 상태로 LLC 블록 쓰기가 수행될 수 있다. 2-단계 저항 상태 변환에서는 블록 B를 구성하는 미디움 영역에 쓰기 동작이 먼저 수행된다. 병렬 TLC MTJ에서 미디움 영역을 변환시키는 전류는 하드 영역을 변환시키지 못하기 때문에 블록 C는 원래 저장된 데이터를 유지할 수 있다. 하지만, 그 전류는 소프트 영역을 변환시킬 수 있어서 블록 A는 저장된 데이터를 잃어버릴 수 있다. 따라서 블록 A를 구성하는 병렬 TLC MTJ들의 소프트 영역을 복원하기 위한 쓰기 동작이 수행된다. 만약 병렬 TLC MTJ들의 하드 영역으로 구성된 블록 C가 선택된다면, 3-단계 저항 상태로 LLC 블록 쓰

기가 수행될 수 있다. 그 3-단계 저항 상태 변환에서는 블록 C를 구성하는 하드 영역에 쓰기 동작이 먼저 수행된다. 병렬 TLC MTJ에서 하드 영역을 변환시키는 전류는 미디움과 소프트 영역들을 변환시킬 수 있기 때문에, 블록 B와 A는 저장된 데이터를 손상될 수 있다. 따라서 블록 B를 구성하는 미디움 영역의 복원과 블록 A를 구성하는 소프트 영역의 복원하는 쓰기 동작이 순차적으로 수행된다.

셀 분할 매핑이 적용된 LLC에서 블록 읽기는 선택된 블록을 구성하는 병렬 TLC MTJ 영역에 따라 읽기 동작들이 결정된다. 그 이유는 병렬 TLC MTJ에서 하드 영역, 미디움 영역, 소프트 영역은 항상 순차적으로 읽여지기 때문이다. 예를 들어, 하드 영역으로 구성된 블록 C를 읽을 경우에는 1-단계 저항 상태 탐지가 수행된다. 즉, 병렬 TLC MTJ에서 하드 영역은 먼저 탐지되기 때문에, 불필요한 미디움과 소프트 영역의 탐지는 생략될 수 있다. 반면에, 미디움 영역으로 구성된 블록 B를 읽을 경우에는 2-단계 저항 상태 탐지가 수행된다. 불필요한 하드 영역을 먼저 탐지한 후, 미디움 영역을 탐지한다. 불필요한 소프트 영역 탐지는 생략될 수 있다. 또한, 소프트 영역으로 구성된 블록 A를 읽을 경우에는 3-단계 저항 상태 탐지가 수행된다. 즉, 필요하지 않은 하드와 미디움 영역을 탐지한 후, 소프트 영역을 탐지한다.

3.2 조건 블록 스와핑 기법

LLC는 상위 캐시에 의해 갱신되거나, 메인 메모리에서 읽어온 데이터를 저장한다. 나중-쓰기(Write-back)의 경우, 상위 캐시의 LRU(Least Recently Used) 알고리즘으로 선택된 블록이 LLC로 갱신되고, 즉시-쓰기(Write-through)의 경우에는 상위 캐시의 어떤 블록이 LLC로 갱신될지 예측하기 어렵다. 또한, 캐시 실패로 발생한 메인 메모리 데이터는 LLC의 LRU 알고리즘에 의해 선택된 블록에 저장된다. 일반적인 LRU 알고리즘은 셀 분할 매핑 기법이 적용된 LLC 블록들이 병렬 TLC MTJ의 어떤 영역으로 구성되는지 고려하지 않을 뿐만 아니라, 어떤 영역으로 구성된 블록이 자주 읽히는지 고려하지 않고,

저장할 블록을 선택한다. 만약, LRU 알고리즘이 소프트웨어 영역으로 구성된 LLC 블록의 데이터를 자주 읽도록 선택하고, 하드 영역으로 구성된 LLC 블록에 데이터를 자주 저장하도록 선택하면, 셀 분할 매핑 기법이 적용된 LLC의 성능과 전력 소모 향상은 크게 제한될 것이다. 따라서 셀 분할 매핑 기법이 적용된 LLC는 병렬 TLC MTJ의 소프트웨어 영역으로 구성된 블록에서 쓰기, 하드 영역으로 구성된 블록에서 읽기가 발생하도록 관리되어야 한다.

그림 4는 제안하는 조건 블록 스와핑 기법들의 순서도를 보여준다. 그림 4(a)는 상위 캐시의 갱신 데이터가 셀 분할 매핑 기법이 적용된 LLC에 저장되는 과정이다. 만약 갱신될 LLC 블록이 병렬 TLC MTJ의 소프트웨어 영역으로 구성되어 있다면, 갱신 데이터는 그 LLC 블록에 저장된다. 반면에, 만약 갱신될 LLC 블록이 병렬 TLC MTJ들의 미디움 영역으로 구성되어 있다면, 갱신 데이터는 그 미디움 영역의 병렬 TLC MTJ들을 공유하는 소프트웨어 영역으로 구성된 블록으로 바꿔 저장될 수 있다.

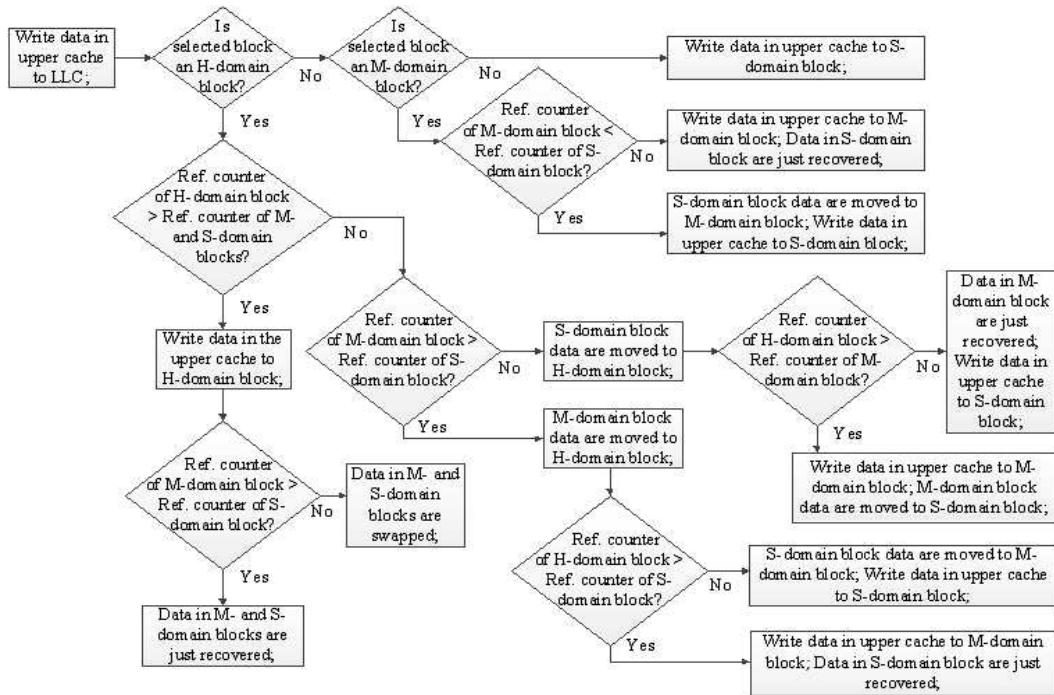
예를 들어, 그림 3(b)에서 블록 B가 갱신되는 경우, 갱신 데이터는 블록 A, 또는 B에 저장될 수 있다. 그 이유는 병렬 TLC MTJ들의 미디움 영역으로 구성된 블록의 데이터가 변경되면, 상응하는 소프트웨어 영역으로 구성된 블록의 데이터는 손상되어 복원이 필요하기 때문이다. 소프트웨어 영역으로 구성된 블록의 데이터와 갱신 데이터가 상위 캐시로 참조되는 횟수를 비교하여 그것들이 갱신 또는 복원되는 블록을 교환할 수 있다. 즉, 갱신 데이터가 소프트웨어 영역으로 구성된 블록의 데이터보다 상위 캐시로 참조되는 경우가 많으면, 읽기 속도가 더 빠른 미디움 영역으로 구성된 블록에 갱신 데이터를 저장하고, 소프트웨어 영역으로 구성된 블록의 데이터는 그 블록에 그대로 복원된다. 그렇지 않으면, 소프트웨어 영역으로 구성된 블록의 데이터는 미디움 영역으로 구성된 블록으로 이주되고, 갱신 데이터는 소프트웨어 영역으로 구성된 블록에 저장된다.

비슷하게, 만약 갱신될 LLC 블록이 병렬 TLC MTJ들의 하드 영역으로 구성되어 있다면, 갱신 데이터는 그 하드 영역의 병렬 TLC MTJ들을 공유하는 미디움과 소프트웨어 영역으로 구성된 블록으로 바

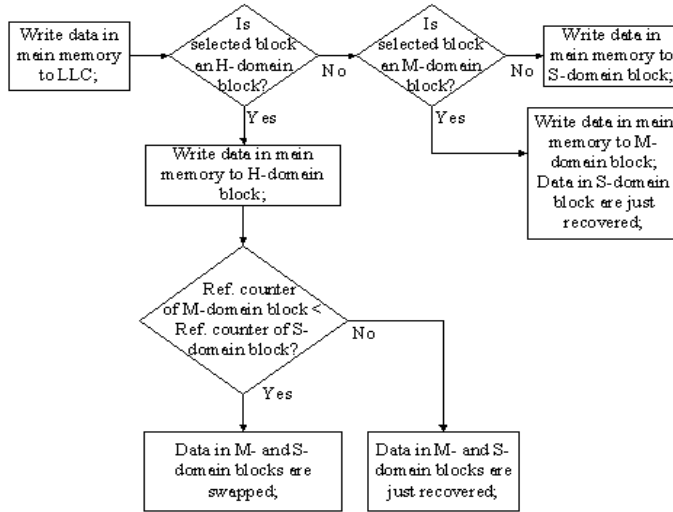
꿔 저장될 수 있다. 예를 들어, 그림 3(b)에서 블록 C가 갱신되는 경우, 갱신 데이터는 블록 A, B, 또는 C에 저장될 수 있다. 그 이유는 병렬 TLC MTJ들의 하드 영역으로 구성된 블록의 데이터가 변경되면, 상응하는 미디움과 소프트웨어 영역으로 구성된 블록의 데이터는 손상되어 복원이 필요하기 때문이다. 갱신 데이터와 미디움과 소프트웨어 영역으로 구성된 블록의 데이터들이 상위 캐시로 참조되는 횟수를 비교하여 그것들이 갱신 또는 복원되는 블록을 교환할 수 있다. 즉, 가장 자주 참조되는 데이터는 읽기 지연이 짧은 하드 영역으로 구성된 블록으로 이주하여 저장되고, 가장 드물게 참조되는 데이터는 소프트웨어 영역으로 구성된 블록으로 이주하여 저장된다. 이것은 낮은 단계의 저항 상태 탐지 과정이 수행되도록 만들어 읽기 지연을 줄일 수 있다.

그림 4(b)는 캐시 실패로 발생한 메인 메모리 데이터가 셀 분할 매핑 기법이 적용된 LLC에 저장되는 과정이다. 메인 메모리 데이터는 LRU 알고리즘으로 선택되는 LLC 블록에 저장된다. 만약 선택된 블록이 병렬 TLC MTJ들의 소프트웨어 영역으로 구성되어 있다면, 다른 블록의 데이터는 손상되지 않는다. 반면에, 선택된 블록이 미디움 영역으로 구성되어 있다면, 그 미디움 영역과 병렬 TLC MTJ들을 공유하는 소프트웨어 영역으로 구성된 블록의 데이터는 손상되어 복원해야 한다. 그리고 선택된 블록이 하드 영역으로 구성되어 있다면, 그 하드 영역의 병렬 TLC MTJ들을 공유하는 미디움 영역과 소프트웨어 영역으로 구성된 블록들의 데이터는 손실되어 복원해야 한다. 이 복원 과정에서 추가적인 비용 없이 두 블록들에 저장된 데이터들을 서로 교환해서 저장할 수 있다.

예를 들어, 그림 3(b)에서 블록 C에 메인 메모리 데이터가 저장된다면, 블록 A와 B의 데이터는 복원되어야 한다. 두 영역 중에 상위 캐시로 참조 횟수가 많은 영역을 미디움 영역으로 구성된 블록에 저장하고, 나머지 데이터를 소프트웨어 영역으로 구성된 블록에 저장한다. 이것은 그 블록이 읽어질 때, 낮은 단계의 저항 상태 탐지 과정이 수행하게 만들 것이다.



(a) LLC로 갱신 데이터를 쓰는 경우
(a) Dirty data written to LLC



(b) LLC로 캐시 실패 데이터를 쓰는 경우
(b) Cache-missed data written to LLC

그림 4. 조건 블록 스와핑 기법들의 순서도

Fig. 4. Process flow of conditional block swapping techniques

제한하는 조건 블록 스와핑 기법은 셀 분할 매핑 기법이 적용된 LLC에서 병렬 TLC MTJ들의 하드 영역으로 구성된 블록에서 쓰기, 소프트 영역으로 구성된 블록에서 읽기를 감소시킨다. 따라서 병렬

TLC MTJ들의 다단계 저항 상태 변화와 탐지의 발생을 최소화하여, 전체 메모리 접근 시간과 전력 소모를 크게 향상시킬 수 있다.

IV. 실험 결과

제안된 병렬 TLC STT-MRAM 기반 LLC의 성능과 전력 소모를 평가하기 위해 MARSSx86 [21] 시뮬레이터가 사용되었다. MARSSx86는 멀티-코어 x86 프로세서들을 위한 사이클 정확도를 가지는 시뮬레이터이다. 표 1은 프로세서와 캐시의 설정을 보여준다. 실험에서 사용된 프로세서는 듀얼-코어와 2-단계 캐시를 내재한다. 각 코어는 SRAM으로 구성된 전용 캐시를 가지며, 16KB의 4-웨이 집합-연관 명령-캐시(I-cache)와 16KB의 4-웨이 집합-연관 데이터-캐시(D-cache)를 가진다. 그 전용 캐시들은 64B 블록으로 구성되며, 나중-쓰기와 Pseudo LRU 방식을 사용한다. 두 코어들은 병렬 TLC STT-MRAM으로 구성된 3MB의 6-웨이 집합-연관 2차 캐시를 공유한다. 표 2는 NVSim[22] 시뮬레이터로 계산된 병렬 TLC STT-MRAM의 파라미터들을 보여준다. 그 공유 캐시는 64B 블록으로 구성되어 있고, 나중-쓰기와 Pseudo LRU 방식으로 동작한다. 본 실험에서는 메인 메모리의 영향으로 인한 LLC들의 성능 변동을 감소시키기 위해, 메인 메모리의 지연은 상수로 고정하였다.

표 1. 프로세서와 캐시의 설정

Table 1. Configuration of processor and caches

Processor	2.4GHz, dual-core, out-of-order, 2-issue per core,
L1 I- and D-cache (SRAM)	16KB, 64B block size, 4-way associativity, write-back, pseudo LRU
L2 cache (Parallel TLC MRAM)	3MB, 64B block size, 6-way associativity, write-back, pseudo LRU
Main memory	120 cycle

표 2. STT-MRAM의 파라미터들

Table 2. Parameters of STT-MRAM

Cell size	36F ² (4.5F transistor)
Read latency (Cycle)	S ^a : 8 M ^b : 5, H ^c : 3
Write latency (Cycle)	S: 19, M: 47, H: 70
Read energy (pJ/bit)	S: 1.094, M: 1.016, H: 0.938
Write energy (pJ/bit)	S: 2.5, M: 4.18, H: 5.72
Leakage (mW)	121.2

^a Soft domain, ^b Medium domain, ^c Hard domain.

기존에는 MLC STT-MRAM을 LLC로 이용하는 연구가 진행되었지만, TLC STT-MRAM을 LLC로 이용하는 연구는 진행되지 않았다. 이 이유는 TLC STT-MRAM의 구현이 어렵고, 오류가 많이 발생하여 메모리로 활용이 가능하지 않았기 때문이다. 하지만, [14]에서 그 문제들이 해결되어, TLC STT-MRAM을 LLC로 이용할 수 있게 되었다. 본 논문은 TLC STT-MRAM을 LLC로 이용한 최초의 논문이다. TLC STT-MRAM을 LLC로 이용하는 것은 MLC STT-MRAM을 LLC로 이용하는 것보다 고려해야 할 요소들이 더 많아서 MLC STT-MRAM을 LLC로 이용했던 방법들을 그대로 확장 적용하는 것은 효과적이지 않다. 따라서 본 논문에서는 TLC STT-MRAM을 LLC로 사용하기 위한 셀 분할 매핑과 조건 블록 스와핑 기법들을 제안하였다.

제안된 LLC에 조건 블록 스와핑 기법을 적용하기 위해서는 상위 캐시로 각 블록의 참조 횟수 정보가 필요하다. MARSSx86에서는 Pseudo LRU 알고리즘을 위한 참조 비트가 블록마다 존재한다. Pseudo LRU에서는 이 정보를 이용해서 메인 메모리로 퇴거되는 블록이 선택된다. 본 구현에서는 그 참조 비트를 사용하여 추가되는 구현 비용이 매우 적다.

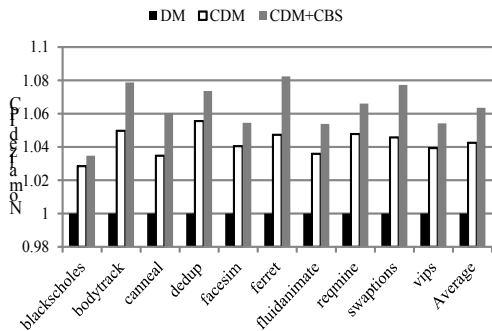
제안된 셀 분할 매핑과 조건 블록 스와핑 기법들을 적용된 LLC(CDM+CBS)는 성능과 전력 소모에서 직접 매핑 기법이 적용된 LLC (DM)와 비교된다. 또한, 제안된 각 기법이 성능과 전력 소모에 미치는 영향을 상세히 분석하기 위해, 셀 분할 매핑 기법만 적용된 LLC(CDM)와 비교된다. 이러한 LLC를 내재하는 각 MARSSx86 시뮬레이터는 하나 또는 두개의 PARSEC[23] 벤치마크들을 10억 사이클 동안 수행한다.

4.1 성능 비교

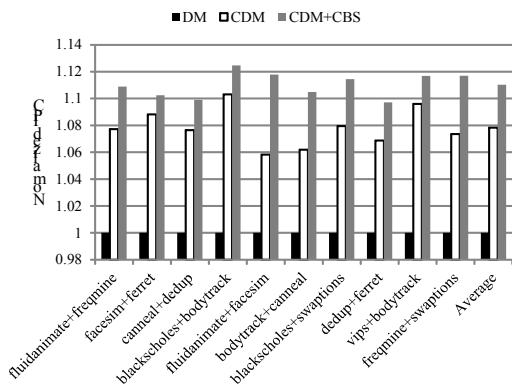
그림 5에서 DM, CDM, 또는 CDM+CBS를 적용한 LLC의 성능을 비교한다. 그림 5(a)는 벤치마크들이 개별적으로 수행되는 경우에 정규화된 Instruction Per Cycle (IPC)를 보여준다. 직접 매핑 기법이 적용된 LLC 블록은 데이터를 읽을 때, 병렬 TLC MTJ

의 모든 영역을 3-단계 저항 상태 탐지로 항상 감지해야 한다. 또한, 그 LLC 블록에 데이터를 쓸 때, 가장 긴 저항 상태 변환을 필요로 하는 병렬 TLC MTJ에 의해 전체 블록의 쓰기 지연이 결정된다. 반면에, 제안된 셀 분할 매핑 기법이 적용된 LLC 블록은 병렬 TLC MTJ들의 동일한 영역으로 구성되어 있어서, 선택된 블록을 구성하는 영역에 따라 읽기와 쓰기 시간이 결정된다. 실험에서는 셀 분할 매핑 기법만 적용된 LLC는 직접 매핑 기법이 적용된 LLC보다 평균적으로 4.25%의 IPC 향상을 보여준다.

제안된 LLC는 조건 블록 스와핑 기법으로 성능이 더 향상된다. 셀 분할 매핑 기법이 적용된 LLC는 하드 영역으로 구성된 블록에서 읽기, 소프트 영역으로 구성된 블록에서 쓰기를 선호한다.



(a) 싱글 벤치마크
(a) Single benchmark



(b) 듀얼 벤치마크
(b) Dual benchmark

그림 5. 성능 비교

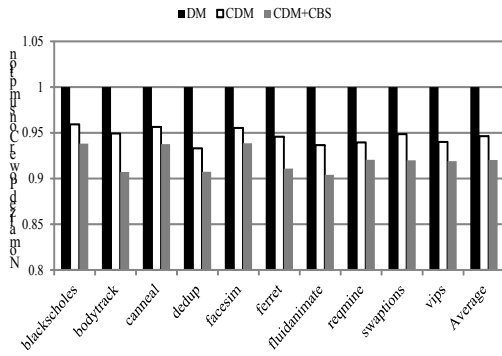
Fig. 5. Performance comparison

조건 블록 스와핑 기법은 상위 캐시와 메인 메모리로부터의 데이터를 LLC에 쓸 때, 블록을 구성하는 영역과 블록의 참조 빈도를 고려하여, 블록들의 데이터를 서로 교환한다. 그 결과, 셀 분할 매핑 기법이 적용된 LLC에서 하드 영역으로 구성된 블록이 자주 읽히고, 소프트 영역으로 구성된 블록에 데이터가 자주 쓰도록 만든다. 실험에서 셀 분할 매핑과 조건 블록 스와핑 기법들이 적용된 LLC가 셀 분할 매핑 기법만을 적용한 LLC보다 2.11% 높은 IPC를 보여준다. 셀 분할 매핑과 조건 블록 스와핑 기법들이 적용된 LLC는 직접 매핑 기법을 적용한 LLC보다 평균적으로 6.36%의 IPC 향상을 보여준다.

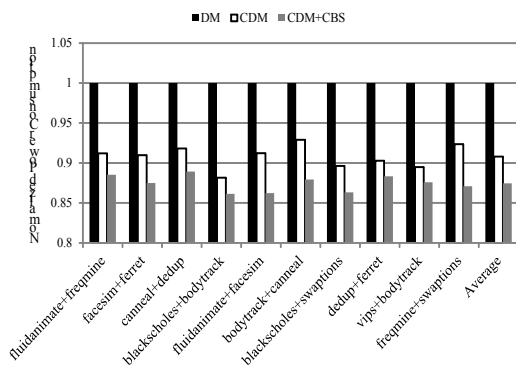
그림 5(b)는 두 개의 벤치마크들이 동시에 수행되는 경우에 정규화 된 IPC(Instruction Per Cycle)를 보여준다. 다수의 벤치마크들이 동시에 수행되면, LLC 실패가 증가한다. 따라서 메인 메모리 데이터를 LLC에 자주 쓰게 되고, 조건 블록 스와핑 기법도 자주 적용된다. 실험 결과에서 셀 분할 매핑 기법만 적용된 LLC는 직접 매핑 기법을 적용된 LLC보다 평균적으로 7.83%의 IPC 향상을 보여준다. 또한, 셀 분할 매핑과 조건 블록 스와핑 기법들이 적용된 LLC는 직접 매핑 기법을 적용한 LLC보다 평균적으로 11.03%의 IPC 향상을 보여준다. 제안된 LLC는 다수의 벤치마크들이 동시에 수행될수록 더 높은 성능을 기대할 수 있다.

4.2 전력 소모 비교

그림 6에서 DM, CDM, 또는 CDM+CBS를 적용한 LLC의 전력 소모를 비교한다. 그림 6(a)는 각 벤치마크가 개별적으로 수행되는 경우에 정규화 된 전력 소모를 보여준다. 제안된 LLC는 셀 분할 매핑과 조건 블록 스와핑 기법들로 전력 소모가 많은 다단계 저항 상태 탐지와 변환의 발생을 감소시킨다. 셀 분할 매핑 기법만 적용된 LLC는 직접 매핑 기법이 적용된 LLC보다 평균적으로 5.37%의 낮은 전력 소모를 보여준다. 셀 분할 매핑과 조건 블록 스와핑 기법들이 적용된 LLC는 직접 매핑 기법을 적용한 LLC보다 평균적으로 7.97%의 낮은 전력 소모를 보여준다.



(a) 싱글 벤치마크
(a) Single benchmark



(b) 듀얼 벤치마크
(b) Dual benchmark
그림 6. 전력 소모 비교

Fig. 6. Power consumption comparison

그림 6(b)는 두 개의 벤치마크들이 동시에 수행되는 경우에 정규화 된 전력 소모를 보여준다. 셀 분할 매핑 기법만 적용된 LLC는 직접 매핑 기법이 적용된 LLC보다 평균적으로 9.2%의 낮은 전력 소모를 보여준다. 셀 분할 매핑과 조건 블록 스와핑 기법들이 적용된 LLC는 직접 매핑 기법을 적용한 LLC보다 평균적으로 12.54%의 낮은 전력 소모를 보여준다. 따라서 제안된 LLC는 다수의 벤치마크들이 동시에 수행될수록 더 낮은 전력 소모를 기대할 수 있다.

V. 결론

인공지능 및 고품질 서비스들이 급격하게 요구됨에 따라, 멀티-코어 프로세서의 성능 향상을 위한 대용량, 고성능, 저전력 LLC들이 활발히 연구되고

있다. 최근에 주목을 받는 병렬 TLC STT-MRAM는 높은 저장 용량을 제공하지만, 복잡한 쓰기과 읽기 동작으로 낮은 성능과 높은 전력 소모 때문에, LLC로 사용이 크게 제한되었다. 이러한 문제들을 개선하기 위해서, 본 논문에서는 병렬 TLC STT-MRAM 기반 대용량, 고성능, 저전력 LLC를 제안하였다. 제안된 LLC는 셀 분할 매핑 기법으로 구조되고, 조건 블록 스와핑 기법으로 관리되어, 병렬 TLC STT-MRAM가 LLC로 사용으로 유발되는 문제들을 해결하였다. 실험 결과에서 제안된 LLC는 두 개의 PARSEC 벤치마크들을 동시에 수행할 때, 직접 매핑을 적용한 LLC보다 11.03%의 IPC 향상과 12.54%의 낮은 전력 소모를 보여준다. 따라서 제안된 병렬 TLC STT-MRAM 기반 LLC는 최신의 멀티-코어 프로세서에 내재되어, 더 향상된 인공지능 및 고품질 서비스 기회를 제공할 수 있다. 향후 연구로서, 구현이 용이하고, 오류가 적은 직렬 TLC STT-MRAM을 LLC로 사용하는 것을 고려하고 있다.

References

- [1] Young-kyu Kim, Seung-Ho Ok, and Byungin Moon, "Modeling and Performance Analysis of the Hardware Transactional Memory", Proceedings of KIIT Summer Conference, pp. 438-441, May 2010.
- [2] M. Chang, P. Rosenfeld, and S. Lu, "Technology comparison for large last-level caches(L3Cs): Low-Leakage SRAM, low write-energy STT-RAM, and refresh-optimized eDRAM", International Symposium on High Performance Computer Architecture (HPCA), pp. 143-154, Feb. 2013.
- [3] A. V. Khvalkovskiy, D. Apalkov, and S. Watts, et al., "Basic principles of STT-MRAM cell operation in memory arrays", Journal of Physics D: Applied Physics, Vol 46, No. 7, pp. 74001-74020, Jan. 2013.
- [4] M. Hosomi, H. Yamagishi, and T. Yamamoto, "A novel nonvolatile memory with spin torque transfer magnetization switching: spin-ram", International Electron Devices Meeting (IEDM), pp. 459-462, Dec. 2005.

- [5] H. Akinaga and H. Shima, "Resistive Random Access Memory (ReRAM) Based on Metal Oxides", *Proceedings of the IEEE*, Vol. 98, No. 12, pp. 2237-2251, Dec. 2010.
- [6] S. Raoux, G. W. Burr, and M. J. Breitwisch, "Phase-change random access memory: A scalable technology", *IBM Journal of Research and Development*, Vol. 52, No. 4.5, pp. 465-479, Jul. 2008.
- [7] X. Wu et al., "Hybrid cache architecture with disparate memory technologies", *International Symposium on Computer Architecture (ISCA)*, pp. 34-45, Jun. 2009.
- [8] L. Jiang, B. Zhao, and Y. Zhang, "Constructing large and fast multi-level cell STT-MRAM based cache for embedded processors", *Design Automation Conference (DAC)*, pp. 907-912, Jun. 2012.
- [9] X. Bi, M. Mao, and D. Wang, "Unleashing the potential of MLC STT-RAM caches", *International Conference on Computer-Aided Design (ICCAD)*, pp. 76-80, Feb. 2003.
- [10] J. Wang, P. Roy, and W. Wong, "Optimizing MLC-based STT-RAM caches by dynamic block size reconfiguration", *International Conference on Computer Design (ICCD)*, pp. 133-138, Oct. 2014.
- [11] Y. Chen, X. Wang, and W. Zhu, "Access Scheme of Multi-Level Cell Spin-Transfer Torque Random Access Memory and its optimization", *International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1109-1112, Aug. 2010.
- [12] Y. Chen, W. Wong, and H. Li, "Processor caches built using multi-level spin-transfer torque RAM cells", *International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 73-78, Aug. 2011.
- [13] M. Sharad, R. Venkatesan, and A. Raghunathan, "Multi-level magnetic RAM using domain wall shift for energy-efficient, high-density caches", *International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 64-69, Sept. 2013.
- [14] Z. Li, B. Yan, and L. Yang, "A new self-reference sensing scheme for TLC MRAM", *International Symposium on Circuits and Systems (ISCAS)*, pp. 593-596, May 2015.
- [15] P. Zhou, B. Zhao, and J. Yang, "Energy reduction for STT-RAM using early write termination", *International Conference on Computer-Aided Design (ICCAD)*, pp. 264-268, Nov. 2009.
- [16] J. W. Jung, Y. Nakata, and M. Yoshimoto, "Energy-efficient Spin-Transfer Torque RAM cache exploiting additional all-zero-data flags", *International Symposium on Quality Electronic Design (ISQED)*, pp. 216-222, Mar. 2013.
- [17] G. Sun, X. Dong, and Y. Xie, "A novel architecture of the 3D stacked MRAM L2 cache for CMPs", *International Symposium on High Performance Computer Architecture (HPCA)*, pp. 239-249, Feb. 2009.
- [18] X. Wu, J. Li, and L. Zhang, "Power and performance of read-write aware Hybrid Caches with non-volatile memories", *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 737-742, Apr. 2009.
- [19] S. K. Hong, J. M. Lee, and S. T. Kim, "Ternary cache: Three-valued MLC STT-RAM caches", *International Conference on Computer Design (ICCD)*, pp. 83-89, Oct. 2014.
- [20] W. Y. Jang, "Multi-level cell STT-RAM controller for multimedia applications", *Electronics Letters*, Vol. 53, No. 1, pp. 12-14, Jan. 2017.
- [21] A. Patel, F. Afram, and S. Chen, "MARSS: A full system simulator for multicore x86 CPUs", *Design Automation Conference (DAC)*, pp. 1050-1055, Jun. 2011.
- [22] X. Dong, C. Xu, and Y. Xie, "NVSim: A Circuit-Level Performance, Energy, and Area Model for Emerging Nonvolatile Memory", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 31, No. 7, pp. 994-1007, Jul. 2012.

- [23] C. Bienia, S. Kumar, and J. Singh et al., "The PARSEC benchmark suite; Characterization and architectural implications", International Conference on Parallel Architectures and Compilation Techniques (PACT), pp. 72-81, Oct. 2008.

저자소개

박 태 진 (Taejin Park)



2015년 2월 : 단국대학교
전자전기공학부(공학사)
2015년 ~ 현재 : 단국대학교
전자전기공학과(공학석사)
관심분야 : 컴퓨터 구조, 임베디드
시스템, SoC

장 우 영 (Wooyoung Jang)



1998년 2월 : 경희대학교
전파공학과(공학사)
2000년 2월 : 연세대학교
전기컴퓨터공학과(공학석사)
2011년 5월 : 텍사스 주립 대학교
(Univ. of Texas at Austin)
전기컴퓨터공학과(공학박사)

2000년 ~ 2012년 : 삼성전자 System LSI 사업부 SoC
개발실 책임연구원

2013년 ~ 현재 : 단국대학교 전자전기공학부 조교수
관심분야 : 컴퓨터 구조, 설계 자동화, 머신러닝