

<http://dx.doi.org/10.7236/IIBC.2015.15.2.137>

IIBC 2015-2-19

내로우 값을 이용한 상변화 메모리상에서의 에너지 소모 절감 기법

Reducing Method of Energy Consumption of Phase Change Memory using Narrow-Value Data

김영웅*

Young-Ung Kim *

요 약 과거 30년 동안 메모리 생산의 경제성 이유로 주 메모리 핵심 제품으로 사용된 DRAM의 단점을 극복할 수 있는 대체 기술로 PRAM 기술이 제안되어 왔다. 본 논문에서는 PRAM의 내로우 값을 이용하여 쓰기 동작을 줄임으로써 에너지 소모를 절감할 수 있는 기법을 제안한다. 이를 위해 내로우 값을 이용한 데이터 압축 방법을 기술하고, PRAM의 아키텍처 구조를 설정하고, SimpleScalar 3.0e 시뮬레이터와 SPEC CPU2000 벤치마크를 사용하여 실험한다. 본 연구의 실험 결과에 의하면 제안된 기법을 사용할 경우 PRAM의 데이터 히트율은 39.4%에서 67.7%로 증가하였으며, 에너지 소모율은 9.2% 감소하였다. 제안된 기법을 사용하기 위해서는 공간 오버헤드가 워드 당 3.13% 발생하며 약간의 추가적인 하드웨어 모듈이 필요하다.

Abstract During the past 30 years, DRAM has been used for the reasons of economic efficiency of the production. Recently, PRAM has been emerged to overcome the shortcomings of DRAM. In this paper, we propose a technique that can reduce energy consumption by use of a narrow values to the write operation of PRAM. For this purpose, we describe the data compression method using a narrow value and the architecture of PRAM, We also experiment under the SimpleScalar 3.0e simulator and SPEC CPU2000 benchmark environments. According to the experiments, the data hit rate of PRAM was increased by 39.4% to 67.7% and energy consumption was reduced by 9.2%. In order to use the proposed technique, it requires 3.12% of space overhead per word, and some additional hardware modules.

Key Words : Phase Change Memory(PRAM), Narrow value, Data hit rate, Energy consumption

1. 서 론

주 메모리는 용량, 파워 소모, 성능 측면에서 급격한 성장을 이루어왔지만 멀티코어 환경 및 가상화 컴퓨팅 등의 등장으로 주 메모리는 더 빠른 발전을 요구받고 있다. 주 메모리는 시스템 전체 중 통상 42%의 파워를 소모

한다. 이와 같은 상황에서 시스템이 더 고속화되고 동시에 주 메모리에 대한 요구사항이 증가할수록 메모리에 대한 파워 소모, 발열에 대한 중요도는 점점 더 증가할 것이다.

지난 30년간 DRAM은 주 메모리의 핵심으로 다뤄져 왔는데, 이는 DRAM이 대체 가능한 다른 기술인 SRAM,

*정회원, 한성대학교 컴퓨터공학과
접수일자 : 2014년 12월 8일, 수정완료 : 2015년 2월 4일
게재확정일자 : 2015년 4월 10일

Received: 8 December, 2014 / Revised: 4 February, 2015 /

Accepted: 10 April, 2015

*Corresponding Author: yukim@hansung.ac.kr

Dept. of Computer Eng., Hansung University, Korea

MRAM 및 플래시 등의 기술보다 더 싼 가격으로 대용량 고속 메모리 설계를 가능하게 하였기 때문이다. 즉, DRAM의 주요 장점은 집적도, 에너지 효율성, 속도 그리고 비트 당 원가가 저렴하다고 할 수 있다.

하지만 이러한 장점들에도 불구하고 DRAM에는 다음과 같은 단점들이 있다. 첫째, DRAM은 읽기 동작을 수행한 후 데이터 셀에 저장되어 있던 값이 지워지기 때문에 DRAM에서 읽기 동작을 수행하고 나면 해당 셀에 저장되어 있던 값을 다시 복구시켜줘야 한다. 둘째, DRAM 셀에 한번 저장된 값은 짧은 순간만 유지되기 때문에 수 밀리 초에 한 번씩 캐퍼시터를 다시 충전시켜주는 리프레시 동작이 필요하게 되며 이로 인하여 많은 양의 대기 전력을 소모하게 된다. 마지막으로, DRAM은 소프트웨어에 취약하다. 외부에서 유입된 알파 입자 또는 노이즈 간섭 등으로 DRAM 셀에 저장된 값이 바뀌어 에러를 유발할 수 있다.

이러한 DRAM의 단점을 극복할 수 있는 대체 기술로 상변화 메모리(Phase Change Memory: PRAM) 기술이 제안되어 왔다^{[1][2][3][4]}. PRAM은 소재의 물리적인 특성 변화를 이용하여 값을 저장하는 방식이다. PRAM의 각 셀은 결정질 상태 또는 무정형 상태로 변할 수 있다. 각각의 상태에서는 특정한 전기적 저항을 갖게 되며 이를 이용하여 0과 1을 저장한다. 물리적인 상태 변화를 유발하는 방식으로 인하여 PRAM에서는 읽기와 쓰기 시 서로 다른 파워 소모 및 동작 시간을 갖게 된다. 읽기 동작 시에는 아주 적은 양의 파워만을 소모하여 빠른 시간에 읽을 수 있는 반면, 쓰기 동작 시에는 많은 파워와 시간을 소모한다^[4]. 이와 같은 PRAM의 특성 때문에 쓰기 동작의 횟수를 줄이면 PRAM에서 소모되는 에너지를 감소시킬 수 있다.

본 논문에서는 PRAM의 쓰기 동작을 내로우 값(narrow value)을 이용하여 줄임으로써 에너지 소모를 절감할 수 있는 기법을 제안한다. 내로우 값은 데이터가 전체 비트의 절반만으로 표현될 수 있는 값을 말한다. 예를 들어, 하나의 데이터 값이 32비트로 표현될 경우, 값이 양의 수이고 그 크기가 작다면 하위 16비트만으로 값을 표현할 수 있을 것이고 상위 16비트는 모두 0값을 가질 것이다. PRAM에 데이터가 적재하기 위해 디스크로부터 읽혀오는 값들 중 내로우 값이 많다면, 내로우 값들의 상위 비트들을 버리고 하위 비트값만을 저장하여도 원래의 값을 표기할 수 있다. 이렇게 내로우 값들의 상위 비

트를 버림으로써 PRAM 내에는 더 많은 저장 공간을 확보할 수 있게 되며 이 경우 PRAM의 데이터 미스율 및 교체율(replacement rate)이 감소하게 된다. PRAM의 데이터가 교체되면 데이터를 디스크로부터 새로 읽은 후 PRAM에 적재하게 되는데, 이 횟수를 줄이면 그만큼 쓰기 동작 횟수를 감소시킬 수 있고 결과적으로 에너지 소모를 절감할 수 있다.

본 연구의 실험 결과에 의하면 제안된 기법을 사용할 경우 PRAM의 데이터 히트율은 39.4%에서 67.7%로 증가하였으며, 에너지 소모율은 9.2% 감소하였다. 제안된 기법을 사용하기 위해서는 공간 오버헤드가 워드 당 3.13% 발생하며 약간의 추가적인 하드웨어 모듈이 필요하다.

본 논문의 구성은 다음과 같다. 제 2장에서는 PRAM에 대하여 설명하고 내로우 값에 대해 소개한다. 제 3장에서는 제안된 기법에 대하여 상세히 설명하며, 제 4장에서는 제안된 기법의 실험 및 결과에 대하여 기술하고, 제 5장에서 결론을 맺는다.

II. 배경

1. 상변화 메모리

상변화 메모리는 안정되면서도 서로 다른 특징을 갖는 특정 물질의 특성을 이용한 메모리이다. PRAM에서 이 물질은 보통 게르마늄, 안티모니, 그리고 텔루리움으로 이루어진 칼코겐 화합물 합금(Ge₂Sb₂Te₅, GST)으로 구성되며, 이 물질은 물리적인 특성이 서로 다른 두 가지 상태로 변이될 수 있다^[2]. GST의 두 가지 상태 중 첫 번째는 무정형 상태(amorphous state)이다. 이 상태에서는 적은 양의 빛만 투과되며 높은 전기적 저항을 갖는다. 두 번째 상태는 결정질 상태(crystalline state)로 이 상태에서는 높은 빛 투과율과 낮은 전기적 저항을 가진다. 100 나노 초 이하의 빠른 결정질 상태 변이 속도와 상태 변이에 따른 큰 저항 변화는 GST를 이용하여 PRAM을 구현케 하는 중요한 요소이다.

PRAM에서는 이러한 전기적 저항의 차이를 가지고 0과 1을 기록한다. 이러한 특성 덕분에 PRAM에서의 읽기 동작은 매우 적은 전류를 흘려 값을 확인해 보는 것만으로도 가능하며 수행 속도도 매우 빠르다는 장점이 있다. 이와는 반대로 쓰기 동작은 조금 복잡하다. GST는 무정

형 상태에서 결정형성점의 온도에 도달하면 결정화하여 결정질 상태가 된다. 그리고 결정질 상태에 도달한 GST에 용해점까지 열을 가하면 다시 무정형 상태가 된다. 이 과정에서 열을 가하기 위한 전류가 소모되며 이는 읽기 동작에 소모되는 양에 비해 훨씬 큰 에너지를 소모한다. 또한 상태 변화에 소모되는 시간 및 변화 후 GST 냉각에 소모되는 시간으로 인하여 읽기 동작보다 훨씬 더 많은 시간을 필요로 하게 된다. 이러한 이유로 PRAM에서는 쓰기 연산이 에너지 소모 및 동작 시간의 병목점이 된다. 즉, PRAM에서의 쓰기 연산을 줄일 수 있다면 에너지 소모 및 성능 개선을 이룰 수 있다.

2. 내로우 값

내로우 값(narrow value)은 워드로 표현되는 데이터 값 중 하위 절반 비트만으로 표현할 수 있는 값을 말한다^{[5][6][10]}. 그림 1은 내로우 값의 예시를 나타낸다. 예를 들어, 16비트가 1워드로 구성되는 시스템에서 그림 1(나)와 같이 정수 값이 5인 경우 16비트 중 하위 절반인 8비트만으로 표현할 수 있다 이러한 경우 해당 값을 내로우 값이라고 한다. 하지만 그림 1(가)와 같이 값이 큰 데이터의 경우는 하위 절반 8비트만으로는 표현할 수 없으며 상위 8비트 모두 필요로 하게 된다. 이 경우 해당 값은 내로우 값이 아니다.

PRAM은 DRAM과는 달리 저장된 데이터에 대해 바이트 단위 접근이 가능한 구조이다^[4]. PRAM의 상위 레벨 메모리인 캐시 메모리로부터 전달되는 데이터에 대해 바이트 단위로 접근하여 PRAM 쓰기를 수행할 수 있으며 이러한 구조적 특징으로 인하여 PRAM에서 내로우 값을 사용한 저장이 가능하다^[6]. 이와는 달리 주 메모리를 DRAM으로 구성할 경우 뱅크 단위로 데이터에 일괄적인 접근하는 특성으로 인해 특정 워드만 활성화시켜 내로우 값을 쓰는 구조는 불가능하다.

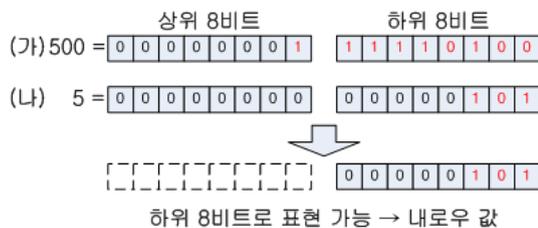


그림 1. 내로우 값 예제
 Fig. 1. Example of Narrow-value

III. 제안 기법

1. 내로우 값을 이용한 데이터 압축

본 연구에서는 시스템에서 사용되는 프로그램의 데이터 중 내로우 값을 지닌 데이터가 많다는 특성을 이용하여 PRAM에서 적재되는 데이터를 압축하여 적재하는 방법을 제안한다. 적재되는 데이터를 압축하면 동일한 공간에 기존 대비 더 많은 데이터를 적재할 수 있게 되며, 이는 결과적으로 페이지 교체율을 감소시켜 더 적은 쓰기 동작을 유발하게 된다.

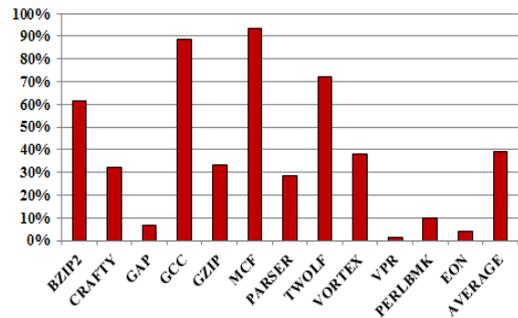


그림 2. SPEC 2000 내 내로우 값의 비율
 Fig. 2. Portion of Narrow-values in SPEC 2000

그림 2는 SPEC 2000 벤치마크 프로그램에서 사용되는 데이터들에 대한 내로우 값 비율을 나타낸다. 적재되는 데이터들 중 내로우 값의 비율은 약 46.8%이다. 즉, PRAM에 적재되는 데이터들 중 절반에 가까운 수가 데이터 워드의 절반 공간만으로 표현이 가능하다는 것이며, 이 절반 공간을 압축하여 저장한다면 PRAM 메모리 크기의 20% ~ 25%에 해당하는 데이터들을 추가로 더 적재할 수 있다.

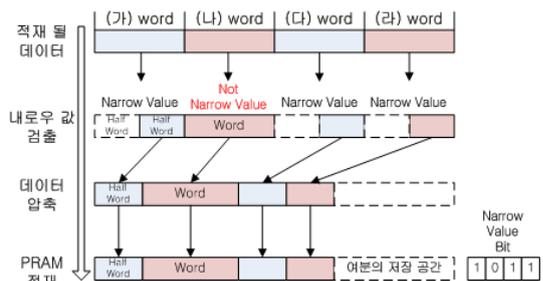


그림 3. 내로우 값을 이용한 데이터 압축 방법
 Fig. 3. A Data Compression Method via Narrow-values

그림 3은 PRAM에 데이터가 적재될 때 내로우 값을 검출하여 압축 저장하는 방법에 대한 논리를 나타내고 있다. 하위 메모리로부터 읽혀진 데이터들은 그 값이 내로우 값인지 아닌지를 워드 단위로 검사된다. 데이터가 내로우 값으로 판명되면 워드의 하위 절반 비트만을 사용하여 저장되며, 이 값이 내로우 값이라는 것을 표시하기 내로우 값 비트에 표기를 한다. 여기서 내로우 값 비트는 원래의 PRAM에는 없는 구조지만 본 연구에서 제안하는 PRAM 구조에 추가하였다. 만약 적재될 워드가 내로우 값이 아니라면 원래의 워드 크기만큼 PRAM에 적재한다. PRAM에 적재된 데이터를 상위 메모리인 캐시 메모리로 보낼 때에는 반대의 동작을 수행한다. 우선 내로우 값 비트를 읽어 각 워드가 압축이 되었는지를 판별한 후, 압축되었다면 상위 비트를 복원하여 원래의 워드를 만든다. 내로우 값 데이터가 아니라면 복원 과정을 생략하고 상위 메모리로 보낸다.

2. 상세 아키텍처

그림 4는 본 연구에서 제안한 PRAM 구조의 상세 아키텍처를 나타낸다. PRAM 데이터에 대한 압축 및 복원을 수행하려면 제로 검출기, 비트 쉬프트 로직, 그리고 내로우 값을 나타내는 추가 비트가 필요하다.

PRAM에 적재되는 데이터는 상위 절반 워드에 대해 제로 검출기를 이용하면 내로우 값 유무를 간단히 알 수 있다. 본 연구에서는 내로우 값을 상위 절반 비트가 모두 0인 경우로 가정하였지만, 경우에 따라 음수까지 포함하여 상위 절반 비트가 모두 1인 경우도 고려할 수 있다. 하지만 그럴 경우 내로우 값을 표기해 주는 내로우 값 비트를 1비트가 아닌 2비트로 확장해야 하고, 이는 워드 당 1비트의 추가적인 공간 오버헤드를 유발하므로 내로우 값을 상위 비트가 0인 경우로 제한하였다. 또한 본 연구에서는 상위 절반 비트가 1, 즉 음수인 내로우 값의 비율을 측정하였으나, 그 비율이 극히 미미하였다.

내로우 값 비트는 적재된 각 데이터 워드가 내로우 값 유무를 표시하는 비트이다. 데이터가 PRAM에 적재될 때 값이 기재되며, 상위 레벨 메모리에서 PRAM으로부터 데이터를 가져갈 때 이 값을 읽어 PRAM 데이터를 압축된 형태로부터 원래의 형태로 복원한 후 데이터를 넘겨준다.

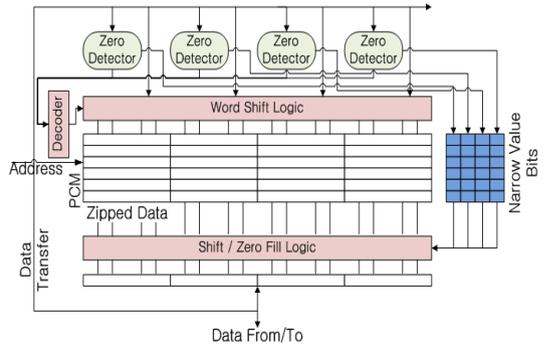


그림 4. 제안된 아키텍처의 상세 구조
Fig. 4. Detailed Architecture of Suggested Method

데이터를 압축하거나 복원할 때 각 워드들은 내로우 값 유무에 따라 데이터의 크기가 달라진다. 예를 들어, 워드의 크기가 4바이트인 시스템에서 하나의 라인 크기가 32바이트인 PRAM이 있다고 가정할 경우, 8개의 워드 데이터가 모두 내로우 값이라면 16바이트로 저장될 것이고, 반대로 모두 내로우 값이 아니라면 32바이트 크기 그대로 저장될 것이다. 기존의 구조에서는 하나의 워드 또는 하나의 라인을 읽거나 쓸 때 그 위치가 항상 일정하였으나, 이러한 상황에서는 PRAM에서 데이터를 읽을 때 어느 바이트 위치에서 시작하여 어디까지 읽어야 하는지가 정해져 있지 않다. 이를 위하여 비트를 이동시킬 수 있는 쉬프트 로직이 필요하다. 데이터를 PRAM에 쓸 경우 제로 검출기로 검사한 내로우 값 검출 결과를 쉬프트 로직의 입력 값으로 사용하여 각 워드들을 얼마나 이동시켜 저장할지 결정한다. 반대로 데이터를 읽어 상위 비트로 전달해야 할 경우에는 내로우 값 비트에 저장된 값을 읽은 후 쉬프트 로직의 입력 값으로 사용하여 압축된 데이터를 원래의 형태로 복원시킨다. 쉬프트 로직은 쉬프트와 가산기를 혼합하여 간단히 구현할 수 있으며 이 때 추가되는 연산 속도는 크게 증가되지 않으므로 성능에 영향을 주지 않는다고 가정하였다.

또한 PRAM에서 데이터를 압축하여 적재한 후 남은 여분의 공간에 추가 데이터를 적재하여야 하는데 이는 데이터를 상위 레벨 메모리로 전달한 후 PRAM이 접근되지 않는 여분의 시간에 수행해도 충분하다. 따라서 이 부분은 PRAM 하위 시스템 외부의 메인 컨트롤러에서 수행한다고 가정하였으며 추가되는 하드웨어 로직은 없고 동작 방식의 변경만 있다고 가정하였다.

IV. 실험 및 결과

1. 실험 환경

실험은 SimpleScalar 3.0e^[7] 시뮬레이터와 SPEC CPU2000^[8] 벤치마크를 사용하여 측정하였다. 10억 사이클을 fast-forward 시킨 후 10억 사이클 동안 실행하였으며, 프로세서 구성은 ARM Cortex-A8^[9]을 기본으로 구성하였다. 표 1은 본 논문의 실험에서 설정한 프로세서 구성 내역이다. 일반적인 DRAM 주 메모리는 기가바이트 급의 용량을 가지고 있지만 PRAM의 경우 주로 128K와 256K로 구성되어 있어 PRAM의 크기를 256K로 제한하였다.

표 1. 시뮬레이터 구성
 Table 1. Simulator Configuration

Processor Core	
Datapath Width	2
Functional Units	2 IALU, 1 IMULT/DIV 1 FPALU, 1 FPMULT/DIV
Branch Predictor	
Predictor	2 Level Globla, 8-entry RAS
BTB	512-entry, 2-way
Memory	
L1 D-Cache	16KB, 4 ways 64B blocks, 2 cycles
PRAM	256K, 4KB memory line, 16-entry write buffer read latency: 60 cycles, write latency: 160 cycles per 16Byte

2. PRAM 데이터 히트율

그림 5는 상위 레벨 메모리인 캐시 메모리에서 PRAM으로 데이터를 요청할 때의 히트율을 나타낸 것이다. Zipped Date Hit는 압축되어 저장된 데이터에서 히트가 발생한 비율이고, Extra Zipped Data Hit는 압축되고 남은 공간에 추가로 적재한 데이터에서 히트가 발생한 비율이다. Miss는 데이터를 찾지 못한 경우로 이 경우 PRAM 라인에 대해 교체가 발생하게 된다.

그림 5는 3장에서 기술한 그림 2와 대체로 비슷한 결과를 보여주고 있다. GCC와 MCF 벤치마크의 경우 적재되는 시점에 내로우 값 데이터들이 많을뿐더러 PCM 미스가 발생하는 빈도 또한 거의 없으므로 높은 커버리지를 보여준다. GZIP과 VPR, 그리고 EON 벤치마크의 경우는 다른 벤치마크들과 다른 양상을 보이는데, 이는 최

초에 PRAM에 데이터가 적재될 때는 데이터가 내로우 값이 아니었으나 데이터가 처리되면서 내로우 값 데이터의 비율이 증가한 경우이다. 예를 들어, GZIP 벤치마크의 경우 ZIP 알고리즘으로 압축된 데이터에 대해 압축해제를 수행하게 되는데, 압축이 해제되면서 내로우 값이 아니었던 데이터가 내로우 값으로 바뀌게 되는 경우이다.

전체 PRAM 데이터 히트율은 67.7%이며 이 중 Extra Zipped Data Hit에 의해 상승된 히트율은 28.4%로, 본 제안 기법을 사용함으로써 약 30%에 달하는 PRAM 미스를 낮추고 PRAM 라인 교체를 감소시켰다.

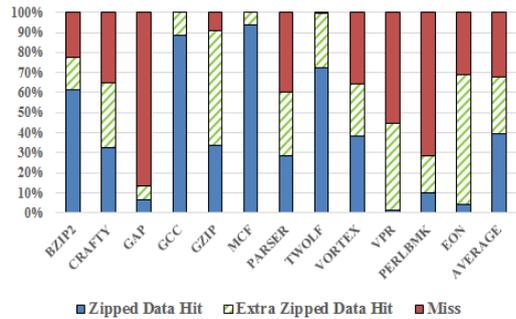


그림 5. PRAM 데이터 히트율
 Fig. 5. PRAM Data Hit Rate

3. 에너지 소모

표 2는 PRAM과 DRAM의 읽기, 쓰기 시 소모되는 에너지를 나타낸 것이다. 표에서 볼 수 있는 것처럼, PRAM은 쓰기 시의 에너지 소모가 약 7배 정도 많다.

표 2. PRAM 에너지 소모량
 Table 2. Energy Consumption of PRAM

	PRAM 동작	
	읽기	쓰기
Energy per bit	2.47 pJ	16.82 pJ

그림 6은 제안한 기법으로 인해 절감된 에너지 소모를 보여준다. GZIP 벤치마크 프로그램의 경우 49.92%의 절감률을 보여주어 제안된 기법이 잘 동작함을 알 수 있으며, GCC와 ART는 거의 효과가 없음을 알 수 있다. GCC와 ART는 PRAM 메모리의 라인 교체 횟수가 현저히 적어 제안된 기법이 잘 적용되지 않는다. 그림 5에서도 알

수 있듯이, 데이터를 압축하여 추가적인 데이터를 더 적재함으로써 PRAM 데이터에 대한 미스가 거의 발생하지 않았으며, 제안 기법을 적용하기 전에도 미스율은 현저히 낮아 교체에 따른 에너지 소모가 크지 않았다. EON의 경우 제안 기법에 따른 감소율이 나타나지 않는데, 이는 제안 기법을 적용함으로써 교체가 전혀 발생하지 않았기 때문이다. 그림 5를 보면 제안된 기법을 적용하더라도 EON의 미스율은 31%인데, 이는 콜드 미스로 인하여 발생한 것으로 이것은 교체를 유발하지 않는다.

제안된 기법을 적용하면 원래의 PRAM 구조 대비 90.8%의 에너지 소모율을 보여주었으며, 이는 9.2%의 절감 효과가 있는 것으로 판단할 수 있다.

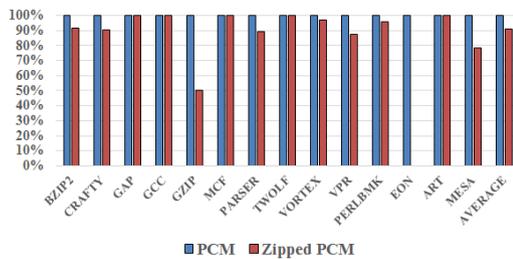


그림 6. 제안된 구조에서의 에너지 소모
Fig. 6. Energy Consumption of Suggested Architecture

4. 공간 오버헤드

본 제안 기법을 구현하기 위해서는 두 개의 쉬프트로직과 적재되는 워드 당 한 개의 제로 검출기가 필요하다. 또한 워드가 내로우 값인지 아닌지를 저장하기 위해서는 워드 당 1비트의 추가 비트가 필요하며, 이는 전체 PRAM 셀 공간 대비 3.12%의 공간 오버헤드를 유발한다.

V. 결론

지난 30년간 DRAM은 주 메모리 구성 기술의 핵심으로 다뤄져 왔는데, 이는 DRAM이 다른 메모리 기술에 비해 고집적도, 고에너지 효율성, 빠른 속도 그리고 비트 당 원가가 저렴한 것에 기인한다. 하지만 이러한 장점들에도 불구하고 DRAM은 읽기 동작을 수행한 후 데이터 셀에 저장되어 있던 값이 지워진다는 점, 셀에 한번 저장된 값은 짧은 순간만 유지될 수 있다는 점, 그리고 소프트웨어에 취약하다는 단점이 있다. 이러한 단점을 극복할

수 있는 대체 기술로 PRAM 기술이 제안되어왔다. PRAM은 소재의 물리적인 특성 변화를 이용하여 값을 저장하는 방식으로, 그 특성상 읽기보다는 쓰기에 더 많은 시간과 에너지가 소모되는 단점이 있다.

본 연구에서는 PRAM의 쓰기 동작을 내로우 값을 이용하여 줄임으로써 에너지 소모를 절감할 수 있는 기법을 제안한다. 내로우 값들의 상위 비트들을 버리고 하위 비트값만을 PRAM에 저장하는 방식으로 PRAM 교체율을 줄이고 이에 따른 쓰기 동작을 줄임으로써 에너지 소모를 절감하였다. 제안된 기법을 통하여 PRAM 데이터 히트율을 28.4% 증가시켰으며, 에너지 소모는 9.2% 감소하였다.

References

- [1] Z. Shao, "Utilizing PCM for Energy Optimization in Embedded Systems," VLSI (ISVLSI), 2012 IEEE Computer Society Annual Symposium on, 398-403 pages, 19-21 Aug. 2012.
- [2] A. Mirhoseini, M. Potkonjak, and F. Koushanfar, "Coding-based energy minimization for phase change memory," DAC '12 Proceedings of the 49th Annual Design Automation Conference, Pages 68-76, 2012.
- [3] S. Cho and H. Lee, "Flip-N-Write: A simple deterministic technique to improve PRAM write performance, energy and endurance," 42nd Annual IEEE/ACM International Symposium, pages, 12-16 Dec. 2009.
- [4] T. Liu, Y. Zhao, C. Xue, and M. Li, "Power-aware variable partitioning for DSPs with hybrid PRAM and DRAM main memory," Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE, pages, 5-9 Jun. 2011.
- [5] O. Ergin et al, "Exploiting narrow values for soft error tolerance," IEEE Computer Architecture Letters, 2006.
- [6] J Kim, S Kim, Y Lee, SimTag: exploiting tag bits similarity to improve the reliability of the data caches, Proceedings of the Conference on Design,

and Test in Europe, 08-12, Mar. 2010.

- [7] D. Burger and T. M. Austin. The SimpleScalar Tool Set, Version 2.0. Computer Architecture News, pages 13 - 25, Jun. 1997.
- [8] The Standard Performance Evaluation Corporation. Spec CPU2000 suite. <http://www.specbench.org/osg/cpu2000/>.
- [9] ARM Cortex A8 processor, "<http://www.arm.com/products/processors/cortex-a/cortex-a8.php>"
- [10] Y. Kim, "Improving Reliability of the Last Level Cache with Low Energy and Low Area Overhead" The Journal of The Institute of Internet, Broadcasting and Communication(IIBC), pages, 35-41, Apr. 2012.

저자 소개

김 영 응(정회원)



- 1993년 : KAIST 전산학과 박사
 - 1984년 ~ 1997년 : KT 통신망연구소
 - 1997년 ~ 현재 : 한성대학교 컴퓨터공학과 교수
- <주관심분야 : 소프트웨어 신뢰도, 소프트웨어 설계, 데이터 모델링>

※ 본 연구는 한성대학교 교내연구비 지원 과제임.