

<http://dx.doi.org/10.7236/JIIBC.2014.14.1.15>

JIIBC 2014-1-3

소프트 에러에 대한 캐시 메모리의 태그 비트 신뢰성 향상 기법

Reliability Improvement of the Tag Bits of the Cache Memory against the Soft Errors

김영웅*

Young-Ung Kim*

요약 반도체 공정 기술의 발달로 인하여 프로세서 내에 적재할 수 있는 캐시 메모리의 용량은 증가하였지만 높은 트랜지스터 집적율은 프로세서를 소프트 에러에 대해 더 취약하게 만들었으며, 이는 설계 고려사항 중 신뢰성의 비중이 점점 더 높아짐을 의미한다. 이러한 취약성을 극복하기 위하여 캐시 메모리의 데이터에 대한 다양한 신뢰성 기법이 제안되었으나, 태그 비트에 대한 연구는 제한적이다. 본 연구는 캐시 메모리 중 태그 비트에 대해 Temporal Locality 특성을 만족하지 않는 write-back 동작에 대한 보호율을 분석하고, 이를 극복할 수 있는 방안을 제안한다. 실험을 통해 제안된 기법으로 기존의 write-back에 대한 보호율을 59.0%에서 76.8%까지 성능 저하 없이 증가시킬 수 있다.

Abstract Due to the development of manufacturing technology scaling, more transistors can be placed on a cache memories of a processor. However, processors become more vulnerable to the soft errors because of highly integrated transistors, the reliability of cache memory must consider seriously at the design level. Various researches are proposed to overcome the vulnerability of soft error, but researches of tag bit are proposed very rarely. In this paper, we reevaluate the reliability improvement technique for tag bit, and analyse the protection rate of write-back operation, which is a typical case of not satisfying temporal locality. We also propose the methodology to improve the protection rate of write-back operation. The experiments of the proposed scheme shows up to 76.8% protection rate without performance degradations

Key Words : Tag Bits, Reliability, Soft Error, Protection Rate

1. 서 론

나노 스케일 반도체 공정 기술은 수 조에서 수십 조 개 이상의 트랜지스터들을 하나의 칩 안에 집적할 수 있는 기술 수준으로 발전하였다. 이러한 트랜지스터의 집

적도 증가는 프로세서가 소프트 에러에 대한 취약성의 원인이 되었으며, 이는 프로세서 신뢰성의 중요도가 크게 증가되고 있다는 것을 의미한다.

소프트 에러는 외부 요인에 의하여 SRAM으로 구성된 메모리 비트 값이 반전되는 현상이다. 이는 불안정한

*정회원, 한성대학교 컴퓨터공학과
접수일자 2013년 12월 3일, 수정완료 2014년 1월 6일
게재확정일자 2014년 2월 7일

Received: 3 December, 2013 / Revised: 6 January, 2014

Accepted: 7 February, 2014

*Corresponding Author: yukim@hansung.ac.kr

Dept. of Computer Engineering, Hansung University, Korea

공급 전원, 타이밍 에러 또는 외부로부터의 알파 입자 유입 등으로 인하여 발생한다^[1]. 프로세서는 이와 같은 발생원인들 중 알파 입자 유입으로 인한 소프트 에러에 가장 취약하며 또한 예측하기가 어렵다. 물리적인 특성 상 메인 메모리에 사용되는 커패시터는 알파 입자에 의한 소프트 에러에 강한 특성을 보이는 반면, 트랜지스터를 사용하는 온칩 캐시 메모리(On-chip Cache Memory)는 소프트 에러에 상대적으로 취약하다. 미세한 프로세서 공정의 발전과 고성능 프로세서에 대한 수요로 인해 현대의 프로세서는 더 큰 온칩 캐시 메모리를 탑재하고 있으며, 이러한 양상은 네트워크 프로세서부터 스마트폰과 같은 고성능 소형 임베디드 디바이스까지 동일하게 나타나고 있다^[2]. 소프트 에러를 극복하기 위하여 캐시 메모리의 데이터 비트에 대한 많은 보호 기법이 제안되었으나, 태그 비트의 소프트 에러를 해결하기 위한 기법은 상대적으로 취약하다. 태그 비트에 대한 소프트 에러 역시 잠재적으로 시스템에 치명적인 에러를 유발할 수 있으며 이에 대한 위험도는 평가절하 되어있다.

대부분의 태그 비트들은 동일한 값을 가진 다른 태그 비트를 다른 캐시 Set에 존재한다. 이것은 하나의 캐시 블록이 접근되거나 교체될 때 인접한 캐시 Set으로부터 동일한 값의 태그 비트를 찾을 수 있다는 것을 의미한다. 이러한 태그 비트 유사성(tag bits similarity)은 소프트 에러를 방지하기 위해 사용될 수 있는데, 예를 들면, 전통적인 패리티 확인 방식으로 에러가 검출되면 인접한 캐시 Set으로부터 동일한 태그 비트를 가져온 후 에러가 발생한 태그 비트를 덮어 써서 수정할 수 있다.

SimTag 기법^[3]은 이러한 태그 비트 유사성을 이용한 소프트 에러 방지 기법이다. 캐시 블록이 교체되어 새로운 데이터 블록이 적재될 때, 상응하는 태그 비트와 동일한 태그 비트가 바로 인접한 캐시 Set에 존재하는지 검사한 후 존재하면 추가 비트에 표시를 하고 추후 에러가 검출되었을 때 이 정보를 활용하여 복구한다. 에러가 발생하였을 때 복구할 수 있는 보호율을 매 사이클 마다 캐시 메모리의 전체 태그 비트가 자신과 동일한 태그 비트를 가지고 있는지 그 수를 검사하여 구하였으며 보호율은 97.9%로 높은 수준의 보호를 수행할 수 있음을 보였다.

하지만 본 연구에서는 이러한 보호율 산출법이 태그 비트의 에러 영향을 잘 반영하지 않을 것으로 판단하여 동일한 기법을 다른 방식으로 측정하였다. 이에 따른 에러 보호율이 어떻게 변하는지 재평가하였으며 기존 방식

의 문제점을 도출하였다. Dirty 상태인 캐시 블록에 대하여 read, write 및 write-back 시 각각의 보호율은 87.3%, 92.0% 및 59.0%이며, 이는 SimTag가 dirty 상태인 데드 블록^[4]에 대하여 취약하다는 것을 의미한다. 데드 블록에 대한 보호율은 시스템의 복잡도가 증가할수록 중요해지는데, 이는 어플리케이션간의 공유 자원 증가 및 처리된 정보의 재사용률이 증가하기 때문이다. 이러한 취약성은 동일한 태그를 찾는 캐시의 Set 범위를 확장하는 것으로 완화할 수 있다. 인접한 위 또는 아래 캐시 블록만 찾지 않고 2개의 Set, 3개의 Set 그리고 5개의 Set을 검사하여 동일한 태그를 찾았을 경우에는 각각 write-back에 대한 보호율이 68.2%, 71.9% 그리고 76.8%로 증가하였다. 이때 성능은 저하되지 않는다.

본 논문의 구성은 다음과 같다. 2장에서는 소프트 에러에 대하여 기술하고 소프트 에러가 태그 비트에 발생하였을 경우 야기되는 에러에 대해 소개한다. 3장에서는 제안된 기법에 대하여 상세히 기술하며, 4장에서는 제안된 기법의 실험 및 결과에 대하여 기술하였다. 5장에서는 결론을 맺는다.

II. 배경

1. 소프트 에러

소프트 에러는 캐시 내 트랜지스터에 저장된 비트 값이 불안정한 공급 전원, 인터커넥션 노이즈, 타이밍 에러 및 외부로부터의 알파 입자 유입으로 인하여 그 값이 0에서 1 또는 1에서 0으로 반전되는 현상을 말하며, 에러가 발생한 값은 데이터 패스(data path)에서 읽거나 다음 레벨 메모리로 쓰기가 수행될 때 영향을 미치게 된다. 이러한 에러는 일시적으로만 발생하므로 소프트 에러(Soft Error or Transient Error)라고 하며, 사용자나 시스템으로부터 관측이 불가능하기 때문에 높은 잠재적 위험을 가진다.

캐시 메모리 내에 발생한 에러를 검출하기 위해 일반적으로 에러 수정 코드(Error Correction Code: ECC)를 추가로 삽입한다. 레벨 1 캐시는 성능에 민감하므로 패리티 비트를 사용하여 에러를 검출하고 LLC(Last Level Cache)에서는 SECCED(Single Error Correction Double Error Detection)와 같은 에러 검출 코드를 사용하여 에러를 검출한다. 만약 에러가 검출된 캐시 블록의 값이 아

직 수정되지 않은 clean 상태라면 단지 다음 레벨의 메모리에서 다시 값을 가져오는 것만으로 복원할 수 있지만 이미 수정된 dirty 상태라면 에러로부터 복구될 수 없다^[8].

2. Tag Bits Errors

소프트 에러는 특별한 로직 설계 없이는 감지되지 않으므로 시스템 결과의 부정확성을 초래할 수 있다. 태그 비트는 일반적으로 패리티나 SECCDED와 같은 에러 수정 코드로 보호되어 있다.

태그 비트에서 발생하는 소프트 에러는 Pseudo-hits, Pseudo-miss, Replacement-error 그리고 Multi-hits 현상이 있다. 그림 1(A)는 Pseudo-hit의 예를 나타낸다. Pseudo-hit는 원래 캐쉬 미스가 발생해야 하나 소프트 에러로 인하여 태그 비트 값이 바뀌면서 히트되어 해당 데이터가 사용되는 현상을 말한다. Pseudo-hit는 에러 수정 코드로 보호되어 있는데, 태그 비트가 읽혀질 때 에러 수정 코드로도 함께 읽어 해당 태그 비트가 올바른 값인지 검사하여 알 수 있다.

Pseudo-miss는 원래 캐쉬 히트가 발생해야 하지만 소프트 에러가 발생하여 미스가 되어 사용되지 않는 현상을 말한다. 그림 1(B)는 Pseudo-miss의 예를 나타낸다. Pseudo-miss는 데이터 캐쉬가 dirty 블록인 경우 캐쉬 미스가 발생하면 다음 레벨 메모리에서 새로 데이터를 적재해야 하고, 이 적재된 데이터는 업데이트되기 이전의 데이터이기 때문에 데이터 무결성 문제를 야기할 수 있다. 만약 Pseudo-miss가 발생한 데이터 캐쉬가 clean 상태라면 아무런 문제도 발생하지 않지만, 캐쉬 라인이 수정된 이후 태그 비트가 소프트 에러에 의해 손상된다면 이후 write-back시 캐쉬의 Set 인덱스와 태그 비트를 합쳐 메인 메모리의 주소를 결정하므로 잘못된 위치에 저장될 수 있다. 이러한 종류의 태그 비트 에러를 Replacement-error라고 하며 그림1(C)와 같다.

캐쉬 메모리가 연관 사상 구조(Set-associative Structure)로 되어있을 경우 소프트 에러에 의하여 Multi-hit 에러가 발생할 수 있다. 그림 1(D)는 4-way 연관 캐쉬 메모리상에서의 Multi-hit 에러의 예를 보여준다. 캐쉬 히트인 경우에 소프트 에러가 발생하여 동일한 캐쉬 Set에서 한개 이상 way 히트가 발생하는 것이다. 이 상황에서는 어떠한 결과가 발생할지 장담할 수 없다. Multi-hit 에러는 캐쉬 히트가 된 상황에서 해당 Set의 다른 way에 존재하는 태그 비트에 대해 비트 반전 현상

이 발생하고, 반전된 값이 인접한 태그 비트와 정확히 같아야 하기 때문에 그 확률은 극히 낮다.

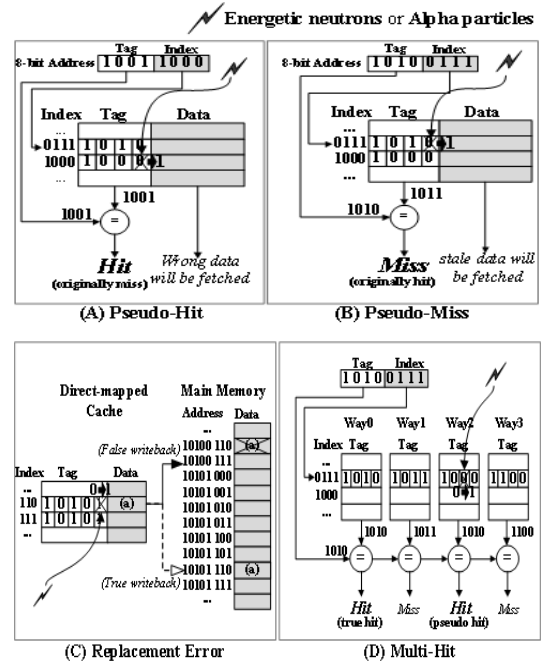


그림 1. 태그 비트에 대한 소프트 에러 영향
Fig. 1. Effects of the Soft Errors on Tag Bits

III. 제안 기법

1. SimTag

SimTag는 하나의 특정 캐쉬 태그 비트와 동일한 태그 비트가 인접한 캐쉬 Set에 존재할 확률이 높은 Spatial Locality 현상을 이용하여 태그 비트에 대한 소프트 에러를 극복하는 기법이다. 그림 2는 메인 메모리의 Spatial Locality와 캐쉬 메모리의 태그 비트에 대한 Spatial Locality를 보여준다. 캐쉬 메모리는 메인 메모리의 전체 공간을 다 반영할 수 없으므로 메모리 주소의 하위 부분을 캐쉬 메모리의 Set Index로 사용하며 나머지 상위 부분은 태그 비트로 저장한다. 어플리케이션 특성상 메인 메모리의 데이터들은 순차적으로 사용될 확률이 높으므로(Spatial Locality) 캐쉬 내 태그 비트들은 서로 유사할 가능성이 높다. 이를 태그 비트 유사성(Tag Bits Similarity)이라고 한다.

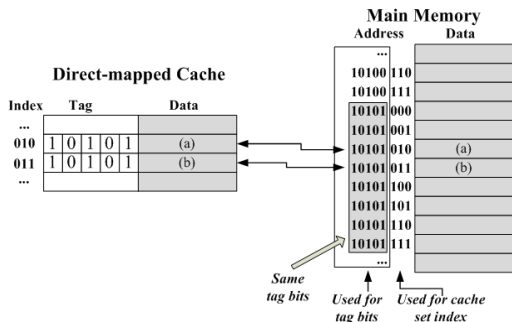


그림 2. 메모리간의 Locality 관계
Fig. 2. Relationship of Locality between Memories

그림 3은 2-Way 연관 사상 구조로 된 캐쉬에서의 SimTag 동작을 나타낸다. 메인 메모리로부터 캐쉬 메모리로 데이터가 적재될 때 SimTag 기법에서는 위 또는 아래 Set을 순차적으로 검색하여 동일한 태그 비트가 있는지 검사한다. 만약 동일한 태그 비트가 존재하면 해당 태그 비트의 위치를 STI(Same Tag Information) 비트에 저장하는데, 2-Way 연관 사상 캐쉬의 경우 캐쉬 라인 당 3비트가 추가로 필요하다. SimTag에서는 에러 검출은 Parity 또는 SECDED와 같은 에러 검출 코드로 수행하며, 에러가 검출 될 경우 STI 비트가 가리키고 있는 위치에서 태그 비트를 읽은 후 에러가 발생한 캐쉬 블록의 태그 비트를 교체한다.

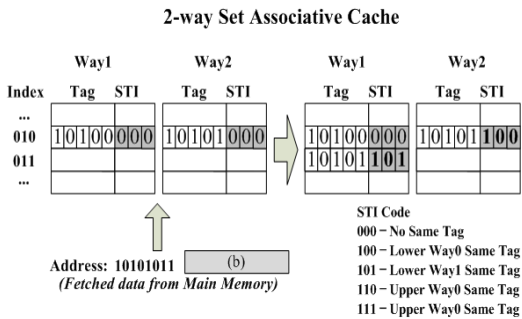


그림 3. SimTag 동작 예제
Fig. 3. Example of SimTag Operation

2. Error Protection Coverage 재평가

기존 SImTag 연구에서는 소프트 에러에 대한 보호율을 프로세서가 동작하는 동안 캐쉬 메모리 내의 전체 블록에 대하여 보호받고 있는지 아닌지를 검사하는 방식으로 측정한다. 즉, 캐쉬 메모리의 Read, Write, Replacement 등의 동작과는 상관없이 매 사이클마다 모

든 블록에 대해 동일한 태그 비트를 참조하고 있는지 아닌지를 검사한다. 기존 연구에서 SimTag는 벤치마크^[5]와 시뮬레이터^[6]를 통한 실험을 수행하였는데 이와 같은 환경에의 제안된 측정 방법은 문제점이 있다. 매 사이클마다 캐쉬의 전체 상태를 관찰하는 방법은 기준을 캐쉬 메모리 자체에 두고 해당 설계가 외부의 다양한 에러 패턴에 대하여 보호 및 복구 능력을 측정해야 하므로 Error-Injection 등과 같은 방법으로 에러의 패턴을 생성하여 각 패턴에 대해 검사해야 한다. 벤치마크를 통하여 측정한다는 것은 다양한 캐쉬 메모리의 동작 형태를 가진 응용 소프트웨어의 특성을 반영하여 각 응용들에 대해 얼마나 유연하게 보호 및 복구를 수행하는지를 측정하는 것으로, 이와 같은 실험에서는 캐쉬 메모리 동작인 Read, Write, Replacement, Write-back의 동작에 대응한 실험 결과가 필요하다. 이에 본 연구에서는 캐쉬에서 dirty 상태인 캐쉬 블록에 대한 Read, Write 그리고 Write-back이 발생하였을 때, 해당 블록이 동일한 다른 태그 비트로부터 보호를 받고 있는지 여부를 관측하여 보호율을 재측정하였다.

3. STI Range 확장

기존 SimTag 연구에서는 바로 인접한 캐쉬의 Set에서 동일한 태그 비트를 찾는 것만으로 97% 이상의 보호율을 보였다. 하지만 캐쉬의 동작별로 보면 이와 같은 구조는 캐쉬 블록의 Read/Write 동작에는 높은 보호율을 보여주나, Write-back에는 취약할 수 있다. 응용 소프트웨어는 Temporal 및 Spatial Locality 특성이 있기에 한번 접근된 메모리 공간은 짧은 시간 동안 여러 번 접근되며 그 이후에는 잘 사용되지 않으며, 무작위로 접근되지 않고 메모리 주소의 순서대로 접근되는 특성이 있다. 이와 같은 상황에서 캐쉬 메모리의 Read/Write 동작은 Spatial Locality와 Temporal Locality를 모두 만족하지만 Write-back은 Temporal Locality를 만족하지 않는다. 어떠한 캐쉬 블록이 Write-back되는 시점은 해당 데이터가 필요한 것이 아니라 새로 적재될 캐쉬 블록에 의하여 강제로 다음 레벨 메모리로 이동하기 때문이다. 따라서 Write-back되는 데이터는 바로 인접한 캐쉬의 Set에 동일한 태그 비트가 존재할 확률이 Read/Write 동작보다 낮다.

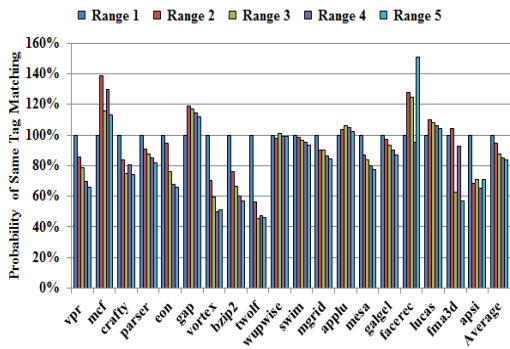


그림 4. STI Range 별 Tag Matching 확률
Fig. 4. Probability of Tag Matching per STI Range

이러한 단점은 동일한 태그 비트를 찾는 Set 범위를 확장함으로써 극복할 수 있다. 이 범위를 STI Range라고 한다. 그림 4는 Set 위치에 따른 동일 태그 비트의 존재 확률을 나타낸다. 예를 들어, Range 1은 기존 SimTag 기법을 나타내며 캐쉬 블록이 캐쉬에 적재될 때 Set Index +1 / -1 만큼의 위치 (바로 인접한 위/아래 위치)에 동일한 태그가 있을 확률이다. Range 2는 Set Index +2 / -2 의 위치에 존재할 확률이며 이러한 Range를 30까지 확장하여 실험하였으나 본 논문에는 Range 5까지의 결과만을 기술하였다. SimTag에서 인접한 캐쉬 Set에서 동일한 태그를 찾을 수 있는 확률을 1이라고 하였을 때 Range의 값이 각각 2, 3, 4, 5일 경우 동일한 태그 비트를 찾을 수 있는 확률은 각각 94.7%, 87.3%, 85.2%, 83.8%으로 나타났다. 또한 Range 6의 확률은 78.5%이며 이후 순차적으로 감소하여 Range 30일 때는 40.4%를 보였다. 이는 STI Range의 확장이 Temporal Locality를 만족하지 않는 Write-back 동작에 대한 보호율을 높여줄 수 있는 단서가 된다.

IV. 실험 및 결과

1. 실험 환경

실험은 SimpleScalar 3.0e^[5] 시뮬레이터와 SPEC CPU2000^[5] 벤치마크를 사용하여 측정하였다. 10억 사이클을 Fast-forward 시킨 후 10억 사이클 동안 실행하였으며, 프로세서는 현재 활발히 사용되고 있는 임베디드 프로세서인 ARM Cortex-A8^[7]을 기본으로 구성하였다. 표 1은 본 논문의 실험에서 설정한 시뮬레이터 구성의 상

세이다.

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
Memory	80 cycles

2. 소프트 에러 보호율

앞에서 기술한 바와 같이 캐쉬 메모리의 Read/Write 동작은 Temporal 및 Spatial Locality를 모두 만족하기 때문에 STI Range의 변화에 민감하게 반응하지 않을 것으로 예측하였다. 그림 5에서 보는 바와 같이 dirty 상태의 캐쉬 블록 Read시 Range의 값이 1, 2, 3, 5일 때의 보호율은 각각 87.3%, 89.9%, 90.5%, 91.2%로 나타났으며, Range 값이 1일 때와 5일 때의 차이가 단지 3.9% 밖에 발생하지 않음을 알 수 있다.

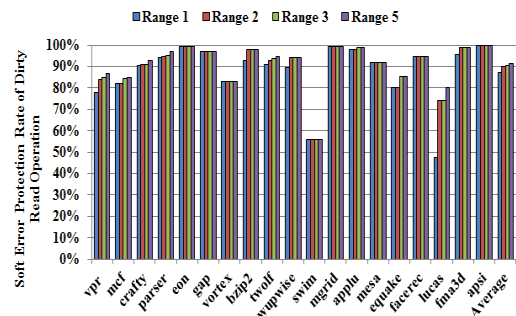


그림 5. Dirty Read 동작에 대한 보호율
Fig. 5. Protection Rate of Dirty Read Operation

그림 6은 dirty 상태의 캐쉬 블록 Write 동작시 보호율이다. Range 값이 1, 2, 3, 5일 때 보호율은 각각 92.0%, 93.6%, 94.2%, 94.8%로 나타났고, 역시 Range 값이 1일 때와 5일 때의 차이는 2.8% 수준이다.

그림 7은 Write-back 동작시 보호율을 보여준다. 기존

의 SimTag 기법은 Write-back 동작에 대하여 단지 59.0%의 보호율을 제공한 반면, Range 2, 3, 5일 때의 보호율은 각각 68.2%, 71.9%, 76.8%이며 이는 기존 기법 대비 9.2%, 12.9%, 17.8% 높은 보호율을 제공한다. 이는 STI Range 확장이 기존 구조 대비 저비용으로 Write-back까지 보호 할 수 있음을 의미한다.

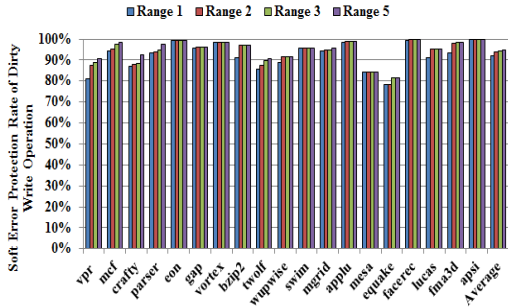


그림 6. Dirty Write 동작에 대한 보호율
Fig. 6. Protection Rate of Dirty Write Operation

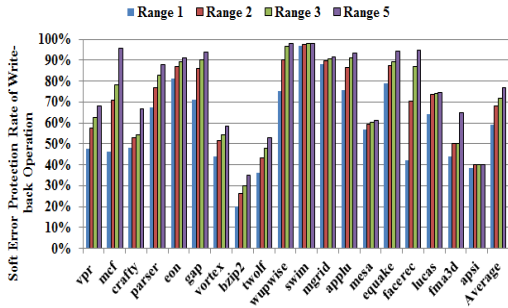


그림 7. Write-back 동작에 대한 보호율
Fig. 7. Protection Rate of Write-back Operation

3. 성능 오버헤드

SimTag는 캐시 블록이 교체되는 시점에서 인접한 캐시 Set을 검사하여 동일한 태그 비트를 찾는다. 이 과정은 캐시 메모리가 다음 레벨의 메모리에서 데이터를 적재하는 동안인 stall 사이클 안에 이루어지므로 기본적으로 성능 저하는 발생하지 않는다. 본 연구에서는 다음 레벨 메모리인 DRAM에서 캐시 메모리로 데이터가 적재되는 사이클이 80 사이클이라고 가정하였으므로 제안된 기법으로 STI Range를 확장하여 검사하여도 성능 저하는 발생하지 않는다.

V. 결론

SimTag는 태그 비트 유사성을 이용한 대표적인 소프트웨어 에러 방지 기법이다. 캐시 블록이 교체되어 새로운 데이터 블록이 적재될 때, 상응하는 태그 비트와 동일한 태그 비트가 바로 인접한 캐시 Set에 존재하는지 검사한 후 존재하면 추가 비트에 표시하고 추후 에러가 검출되었을 때 이 정보를 활용하여 복구한다.

하지만 기존의 연구에서 사용한 소프트웨어 에러 보호을 산출법은 태그 비트에 대한 소프트웨어 에러의 영향 정도를 제대로 나타내지 못하였다고 판단하여 본 연구에서는 기존의 연구에서 사용하였던 보호을 산출법보다 더 세밀한 방법을 사용하여 재평가하였다.

재평가된 실험에서 기존의 SimTag 기법은 Temporal Locality를 만족하지 않는 동작인 Write-back 동작에 취약함을 보였으며, 본 연구에서 제안된 방법으로 STI Range 값을 기존 1로 하였을 때의 Write-back 보호율이 59.0%에서 5로 확장하였을 때 76.8%까지 상승하였으며, 이는 기존 대비 17.8% 가량 보호율이 향상되었다.

References

- [1] O. Ergin et al, "Exploiting narrow values for soft error tolerance," IEEE Computer Architecture Letters, 2006.
- [2] J. Baek and H. Kim, Soft error correction controller for FPGA Configuration memory, Journal of the Korea Academia-Industrial cooperation Society, Vol. 13, No. 11, pp. 5465 - 5470, 2012.
- [3] 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, Automation and Test in Europe, pp. 8 - 12, Mar, 2010.
- [4] W. Zhang, S. Gurumurthi, M. Kandemir, and A. Sivasubramaniam, ICR: In-cache replication for enhancing data cache reliability," in Proc. Int. Conf. Depend. Syst. Netw, pp. 291 - 300, 2003.
- [5] The Standard Performance Evaluation Corporation. Spec CPU2000 suite.

<http://www.specbench.org/osg/cpu2000/>.

- [6] D. Burger and T. M. Austin. The SimpleScalar Tool Set, Version 2.0. Computer Architecture News, pp.13 - 25, June 1997.
- [7] ARM Cortex A8 processor, "<http://www.arm.com/products/processors/cortex-a/cortex-a8.php>".
- [8] Y Kim, Improving the Reliability of the Last Level Cache with Low Energy and Low Area Overhead, The Journal of The Institute of Webcasting, Internet and Telecommunication, Vol 12, No 2, pp.35 - 41, Apr, 2012.

저자 소개

김 영 응(정회원)



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

※ 본 연구는 한성대학교 교내연구비 지원 과제임.