

논문 2009-46C1-4-3

퓨전 플래시 메모리의 다중 블록 삭제를 위한 Erase Group Flash Translation Layer

(Erase Group Flash Translation Layer for Multi Block Erase of Fusion
Flash Memory)

이 동 환*, 조 원 희*, 김 덕 환**

(Dong-Hwan Lee, Won-Hee Cho, and Deok-Hwan Kim)

요 약

OneNAND™와 같이 NAND와 NOR 플래시 메모리의 장점을 혼합한 퓨전 플래시 메모리는 대용량과 빠른 읽기/쓰기 및 XIP(eXecute-In-Place)를 지원하여 고성능 휴대용 임베디드 시스템을 위한 유틸리티저장장치로 각광받고 있다. 또한 OneNAND™는 혼합형 구조의 장점뿐만 아니라 다수의 블록을 한 번에 삭제할 수 있는 다중 블록 삭제 기능을 제공하여 플래시 메모리의 느린 삭제 성능을 향상시켰다. 하지만 기존의 플래시 메모리 주소 변환 계층에서는 다수의 블록을 한 번에 삭제할 수 있다는 점을 고려하지 않고, 소수의 블록들을 가비지 컬렉션의 희생 블록으로 선택하여 삭제하므로 다중 블록 삭제 기능의 효율적인 사용이 어렵다. 본 논문에서는 다중 블록 삭제의 사용을 개선할 수 있는 EGFTL(Erase Group Flash Translation Layer)를 제안한다. EGFTL은 가비지 컬렉션 성능이 뛰어난 Superblock scheme과 다수의 무효 블록들을 관리하는 무효 블록 관리자를 통하여 다수의 블록들을 한 번에 삭제할 수 있도록 한다. 또한 균집형 해시 테이블을 적용하여 Superblock scheme의 주소 변환 성능을 개선하였다. 실험 결과 본 논문에서 제안한 EGFTL이 다른 주소 변환 계층 보다 가비지 컬렉션 성능을 30% 이상 향상시켰으며, Superblock scheme의 주소 변환 성능을 5%이상 향상시켰다.

Abstract

Fusion flash memory such as OneNAND™ is popular as a ubiquitous storage device for embedded systems because it has advantages of NAND and NOR flash memory that it can support large capacity, fast read/write performance and XIP(eXecute-In-Place). Besides, OneNAND™ provides not only advantages of hybrid structure but also multi-block erase function that improves slow erase performance by erasing the multiple blocks simultaneously. But traditional NAND Flash Translation Layer may not fully support it because the garbage collection of traditional FTL only considers a few block as victim block and erases them. In this paper, we propose an Erase Group Flash Translation Layer for improving multi-block erase function. EGFTL uses a superblock scheme for enhancing garbage collection performance and invalid block management to erase multiple blocks simultaneously. Also, it uses clustered hash table to improve the address translation performance of the superblock scheme. The experimental results show that the garbage collection performance of EGFTL is 30% higher than those of traditional FTLs, and the address translation performance of EGFTL is 5% higher than that of Superblock scheme.

Keywords : FTL, Multi Block Erase, Fusion flash memory, garbage collection, erase group

* 학생회원, ** 정회원, 인하대학교 전자공학과

(Department of Electronic Engineering, Inha University)

※ 이 논문은 2008년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임 (KRF-2008-313-D00822).

※ 본 연구는 지식경제부와 한국산업기술 재단의 전략기술인력양성사업으로 수행된 연구결과임.

※ 본 논문은 정보통신부 출연금으로 ETRI, SOC산업 진흥센터에서 수행한 IT SOC 핵심설계인력양성사업의 연구 결과입니다.

접수일자: 2009년5월29일, 수정완료일: 2009년7월2일

I. 서 론

비휘발성 메모리 저장장치인 플래시 메모리는 작고 가벼운 무게와 적은 전력 소모 그리고 충격에 강한 특성을 가지고 있어 휴대용 임베디드 시스템을 위한 유비쿼터스 저장장치로 많이 활용된다. NOR 플래시 메모리의 경우 빠른 읽기 성능과 바이트 단위로 읽기가 가능한 램(RAM) 형태의 인터페이스, XIP(eXecute In Place)를 지원하여 주로 코드 영역을 저장하기 위한 저장장치로 사용되며, 집적도가 높은 구조적 특성과 NOR 플래시 메모리보다 빠른 쓰기/삭제 성능을 갖고 있는 NAND 플래시 메모리는 주로 데이터 저장을 위한 저장장치로 사용된다. 대부분의 임베디드 시스템에서는 이와 같이 용도에 따라 NOR 플래시 메모리와 NAND 플래시 메모리를 구분하여 함께 사용하지만 이 경우 시스템 구조가 복잡해지고 비용이 상승하게 되는 문제를 갖게 된다. 이러한 문제점을 해결하기 위해 NAND 플래시 메모리의 대용량, 빠른 쓰기 속도와 NOR 플래시 메모리의 빠른 읽기 속도, XIP를 모두 지원하는 퓨전 플래시 메모리가 제안되었다. 대표적인 퓨전 플래시 메모리 중의 하나인 OneNAND™는 NAND 플래시 메모리와 제어 로직을 결합해 RAM 인터페이스를 사용할 수 있도록 하였다^[1]. RAM 인터페이스를 통하여 OneNAND™는 NOR 플래시 메모리와 NAND 플래시 메모리를 혼합한 구조의 장점뿐만 아니라 다중 블록 삭제(Multi Block Erase) 기능을 제공하여, 플래시 메모리의 단점으로 지적되었던 느린 삭제 성능을 향상시켰다. 다중 블록 삭제는 다수의 블록을 한 번에 삭제하는 기능으로 최대 64개의 블록을 4ms내에 삭제할 수 있다. 이는 기존의 NAND 플래시 메모리의 블록 당 삭제 시간이 2ms에 비해 최대 32배 빠르므로, 다중 블록 삭제를 사용하여 가비지 컬렉션의 삭제 시간에 대한 부하를 크게 줄일 수가 있다. 다중 블록 삭제 기능은 특히 64개의 블록을 한 번에 삭제할 경우 최대의 효과를 보일 수 있다^[2].

이러한 특징 외에 OneNAND™는 NOR 플래시 메모리와 NAND 플래시 메모리와 마찬가지로 블록 당 삭제 횟수 제한 및 쓰기, 읽기, 삭제 단위와 속도가 다른 비대칭적인 연산 성능과 제자리 갱신이 불가능하고 쓰기 전에 블록이 삭제되어야 하는 특성(erase-before-write)을 갖고 있다. 따라서 이러한 특성을 극복하기 위해 기존의 플래시 메모리와 같이 플래시 변환 계층(Flash

Translation Layer)를 사용한다^[3]. 플래시 변환 계층은 논리적인 주소를 물리적인 주소로 사상하는 방법을 제공하고, 균등한 쓰기를 보장하기 위한 마모도 평준화 방법(Wear-leveling)과 무효 데이터를 처리하기 위한 가비지 컬렉션(Garbage collection) 기능을 제공한다. 특히 주소 사상 방법과 가비지 컬렉션 성능에 따라 플래시 메모리의 성능이 좌우된다. OneNAND™는 다중 블록 삭제 기능을 제공하므로 가비지 컬렉션의 삭제 시간을 단축시켜 플래시 메모리의 성능을 높일 수 있는 가능성을 제공하지만 기존의 플래시 변환 계층은 다수의 블록을 한 번에 삭제할 수 있는 점을 고려하지 않고 있기 때문에 다중 블록 삭제 기능을 효율적으로 사용하지 못하고 있다. 유비쿼터스 컴퓨팅 환경에서 플래시 메모리를 대용량 저장장치로 활용하기 위해서는 수행시간이 오래 걸리는 삭제 연산 문제를 해결해야 멀티미디어 데이터 처리 및 저장 등의 QoS를 보장할 수 있다.

본 논문에서는 퓨전 플래시 메모리의 다중 블록 삭제 기능을 효율적으로 사용하는 방안에 대하여 연구하며, EGFTL(Erase Group Flash Translation Layer)를 제안한다. EGFTL은 3단계 주소 사상을 사용하는 플래시 변환 계층인 Superblock scheme과 빠른 주소 변환을 위한 군집형 해시 테이블, 그리고 다수의 블록을 삭제하기 위한 무효 블록 관리자를 사용하여 가비지 컬렉션 비용을 최소화 할 수 있도록 한다.

본 논문의 순서는 먼저 II장에서 관련 연구에 대해 소개하고 III장에서 군집형 해시 테이블과 무효 블록 관리자를 사용한 EGFTL(Erase Group FTL)의 설계 및 구현에 대해 설명한다. IV장에서는 기존의 FTL과 EGFTL을 적용한 FTL의 성능을 비교한다. 마지막으로 V장에서 결론을 맺는다.

II. 관련 연구

1. 플래시 변환 계층(Flash translation layer)

플래시 메모리는 데이터를 기록하기 전에 데이터 영역이 지워져 있어야 하는 제약이 있으며, 비대칭적인 읽기, 쓰기, 삭제 연산의 처리속도, 블록 당 소거 횟수의 제한과 같은 특징을 지닌다. 위와 같은 단점을 극복하고 플래시 메모리를 효율적으로 사용하기 위해서 플래시 변환 계층이라는 미들웨어가 제안되었다^[3]. 플래시 변환 계층의 주요 기능은 논리적인 주소를 물리적인 주소로 사상하는 방법과 무효 데이터들을 처리하기 위한

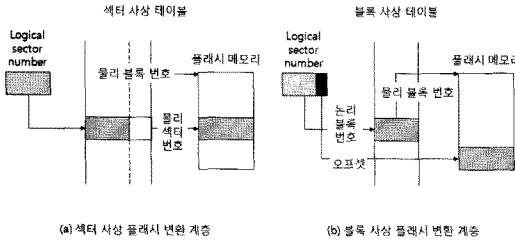


그림 1. FTL의 주소 사상 방법
Fig. 1. The address translation scheme in FTL.

가비지 컬렉션 기능 그리고 균등한 쓰기를 보장하기 위한 마모도 평준화 기능을 제공한다.

플래시 변환 계층의 주소 사상 방법은 섹터 사상 (sector mapping), 블록 사상(block mapping), 혼합 사상(hybrid mapping)으로 나눌 수 있다. 그림 1(a)의 섹터 사상은 논리주소와 물리주소를 섹터 단위로 1:1 주소 변환을 하기 때문에 주소 변환 성능이 빠르다. 하지만 섹터 사상은 테이블 관리를 위한 메모리 요구량이 큰 단점이 있다. 이를 해결하기 위해 그림 1(b)의 블록 사상은 블록 단위로 사상하는 방법을 제공하여 메모리 요구량을 감소시켰다. 하지만 겹쳐 쓰기(Overwrite)가 발생할 경우 블록 내의 모든 데이터를 다른 블록으로 복사해야 하므로 성능이 저하되는 단점이 있다. 따라서 섹터 사상과 블록 사상을 혼합한 혼합 사상 방법이 제안되었다. 대표적인 혼합 사상 방법으로 로그 블록을 사용하는 BAST(Block Associative Sector Translation layer)와 FAST(Fully Associative Sector Translation layer)가 있다^[4-5].

BAST와 FAST는 블록 사상의 단점인 겹쳐 쓰기에 대한 불필요한 복사를 줄이기 위해 로그 블록(Log block)이라는 일종의 쓰기 캐시 블록을 사용한다. 로그 블록의 사용방법에 따라 BAST와 FAST를 구분한다. BAST는 겹쳐 쓰기가 일어날 경우 데이터 블록과 관련된 로그 블록을 두어 데이터 블록의 오프셋과 동일한 위치에 데이터를 저장한다. BAST는 순차적인 데이터 갱신에 좋은 성능을 보이지만 랜덤쓰기가 발생할 경우 로그 블록이 모두 사용되기 전에 새로운 로그 블록을 할당하는 블록 스레싱(block thrashing) 문제가 발생한다^[4]. 또한 새로운 로그 블록에 갱신되기 전의 로그 블록의 유효 페이지를 복사해야하는 추가적인 쓰기 연산이 발생하여 플래시 메모리의 성능이 저하된다. BAST의 이러한 단점을 보완하기 위해 모든 데이터 블록에 대해 오프셋에 상관없이 데이터 갱신이 발생하면 로그

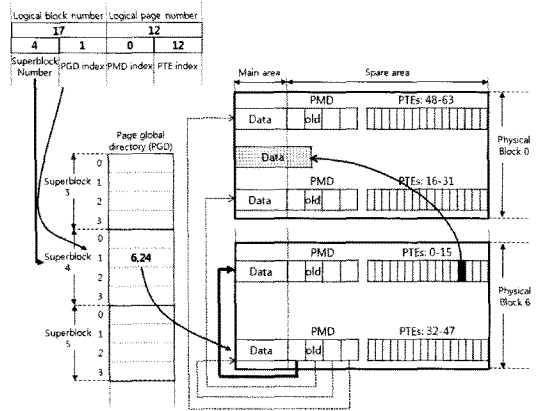


그림 2. Superblock scheme의 주소 사상
Fig. 2. The address translation in the superblock scheme with three-level page-mapping table.

블록의 빈 공간에 데이터를 순차적으로 저장하는 FAST가 제안되었다. FAST를 통하여 랜덤 쓰기에 대해 성능 저하를 줄일 수 있었지만 Hot data와 같이 자주 쓰이는 데이터가 존재할 경우 가비지 컬렉션의 성능이 떨어지는 단점이 존재한다. Hot data에 대한 문제를 해결하기 위한 방법으로 Superblock scheme가 있다.

2. Superblock scheme

일반적인 저장장치 데이터 특성이 블록 수준의 시간 지역성과 공간 지역성을 갖는 점을 고려하여, 그림 2는 인접한 논리 블록들의 집합인 Superblock을 사용하는 주소 사상 방법을 보여준다. 여기서 Superblock은 블록 수준의 주소 사상 방법을 사용하고 Superblock내의 블록들은 섹터 수준의 주소 사상 방법을 사용한다^[6]. Superblock scheme은 Superblock 단위로 갱신 블록을 관리하며, Superblock 내의 데이터는 오프셋에 상관없이 쓰인 순서대로 저장된다. 따라서 자주 접근되는 데이터와 접근되지 않는 데이터가 자동적으로 분리되어 가비지 컬렉션에서의 추가적인 쓰기를 감소시킬 수 있다.

Superblock scheme은 3단계 주소 사상을 사용하여 Superblock 내의 블록들에 대하여 섹터 사상이 가능하도록 하였다. 3단계 주소 사상은 PGD(Page Global Directory), PMD(Page Middle Directory), PTE(Page Table Entry)의 3단계 주소 변환 테이블을 사용하며, PTE에 섹터 사상 테이블을 저장하여 섹터사상과 같은 자유로운 주소 변환이 가능하다. 또한 PGD만 랩 영역

에 저장하고 나머지 테이블은 플래시 메모리에 저장하여 주소 사상 테이블에 소모되는 메모리 요구량을 줄일 수 있다.

하지만 Superblock scheme은 주소 사상 테이블을 플래시 메모리의 스페어 영역에 저장하기 때문에 원하는 데이터를 읽기 위해 플래시 메모리를 두 번 읽는 오버헤드가 발생할 수 있다. 이를 해결하기 위해 PMD와 PTE의 일부를 저장하는 LRU 테이블을 사용하지만 테이블 내에 요청된 주소가 없거나 연속된 주소 요청이 발생할 경우 주소 변환 성능이 저하되는 단점이 발생한다. 또한 가비지 컬렉션의 상황에 따라 선택되는 희생 블록의 수가 일정하지 않기 때문에 다중 블록 삭제의 효율성이 떨어진다. 따라서 효율적인 주소 변환 방법 및 다중 블록 삭제를 위한 다수의 희생 블록들을 선택하여 삭제하는 방법이 필요하다.

III. EGFTL의 설계 및 구현

본 논문에서 제안하는 EGFTL은 로그 블록을 사용한 기존의 혼합 사상 보다 불필요한 복사가 적은 Superblock scheme과 다중 블록 삭제를 위한 무효 블록 관리자를 통해 가비지 컬렉션 성능을 향상시킨다. 또한 기존의 Superblock scheme의 주소 변환 성능을 향상시키기 위해 LRU/PMD 군집형 해시 테이블을 사용하는 방법을 제안한다.

1. 주소 변환 방법

그림 3은 EGFTL의 주소 변환 방법의 전체 구조를 설명한다. 일반적으로 Superblock scheme의 3단계 주소 변환은 요청된 논리 주소에 대한 물리 주소를 검색하기 위해 플래시 메모리의 스페어 영역을 두 번 읽게 되어 성능 저하 문제가 발생한다. 이를 해결하기 위해 Superblock scheme는 일부의 PTE에 대해 LRU를 적용한 캐시 정책을 사용하여 보완하였다^[6]. 하지만 캐시 테이블에 요청한 주소가 없을 경우와 연속적인 주소 요청이 많이 발생할 경우 LRU 테이블의 갱신이 많이 이루어져 플래시 메모리의 성능을 감소시킨다. EGFTL의 주소 변환은 이와 같은 플래시 메모리를 읽는 오버헤드를 줄이기 위해 최근에 참조된 PMD와 관련된 페이지들을 섹터 사상을 하여 플래시 메모리의 스페어 영역의 참조를 최소화 한다.

EGFTL은 LRU 군집형 해시 테이블, PMD 군집형

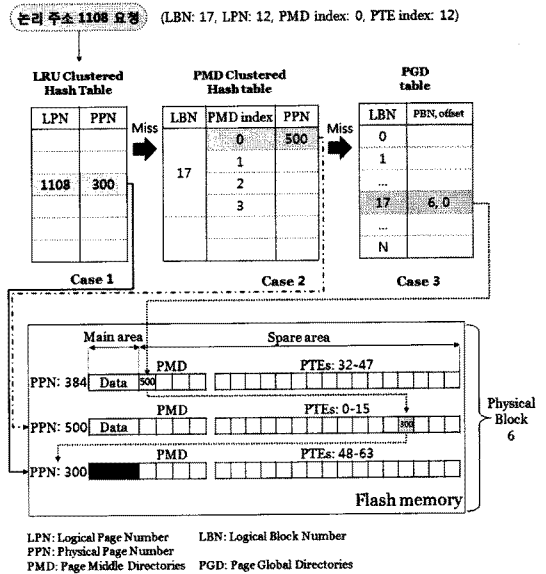


그림 3. EGFTL의 주소 사상

Fig. 3. The address translation in the EGFTL.

해시 테이블, PGD 테이블로 구성된다. LRU 군집형 해시 테이블은 공간 지역성을 고려하여 최근에 사용된 PMD의 PTE를 섹터 사상하며, 연속된 논리 주소를 빠르게 처리할 수 있게 한다. LRU 군집형 해시 테이블이 모두 사용된 경우 PMD 해시 테이블로 강동한다. PMD 군집형 해시 테이블은 PMD를 빠르게 요청할 수 있게 한다.

논리 주소가 요청되면 먼저 LRU 군집형 해시 테이블을 검색하여 요청된 논리 주소에 대한 물리 페이지 주소가 있을 경우 플래시 메모리의 스페어 영역의 참조 없이 물리 주소를 반환하여 요청된 논리 주소에 대한 데이터를 찾는다. 이는 그림 3의 Case 1에 해당하며 메모리에서 주소 요청이 바로 이루어지므로 빠른 주소 변환이 가능하다. LRU 군집형 해시 테이블에 요청된 논리 주소에 대한 물리 주소가 존재하지 않을 경우 그림 3의 Case 2와 같이 PMD 군집형 해시 테이블을 검색하여 요청된 논리 주소에 대한 PMD 데이터가 있는지 검색하여 플래시 메모리의 스페어 영역의 참조를 한 번으로 줄인다. 요청된 논리 주소가 LRU 군집형 해시 테이블과 PMD 군집형 해시 테이블에 존재하지 않을 경우 그림 3의 Case 3과 같이 PGD 테이블을 검색하여 플래시 메모리의 스페어 영역의 PMD와 PTE를 차례로 읽어 데이터를 검색한다.

1. 1. LRU 군집형 해시 테이블

EGFTL은 Superblock scheme의 연속된 주소 요청에 대한 성능을 향상시키기 위해 LRU 군집형 해시 테이블을 사용한다.

Superblock scheme는 주소 변환을 위해 최악의 경우 플래시 메모리를 두 번 읽어야 하는 오버헤드가 존재한다. EGFTL은 지역성을 고려하여 최근에 사용된 주소와 그와 인접한 주소들을 섹터 사상으로 관리하여 연속된 주소 요청을 빠르게 할 수 있도록 한다. 또한 주소 변환 성능을 향상시킬 수 있도록 선형 탐색 시간을 갖는 그림 4와 같은 군집형 해시 테이블을 사용하여 주소를 관리한다.

군집형 해시 테이블은 과거에 Sun Microsystems에서 설계한 군집형 페이지 테이블 기법을 FTL 상에서 구현한 것으로 해시 테이블에서 Subblocking 기법을 이용한 확장 방식이다^[7, 11]. 군집형 해시 테이블에서 버킷은 Subblock의 집합을 의미하며 각 Subblock의 개

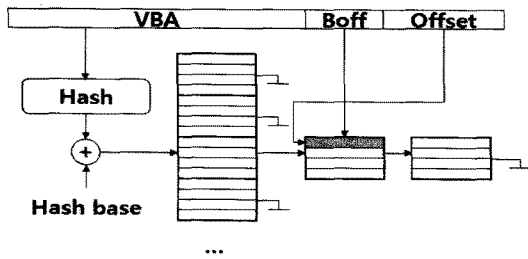


그림 4. 군집형 해시 테이블
Fig. 4. Clustered Hash Table.

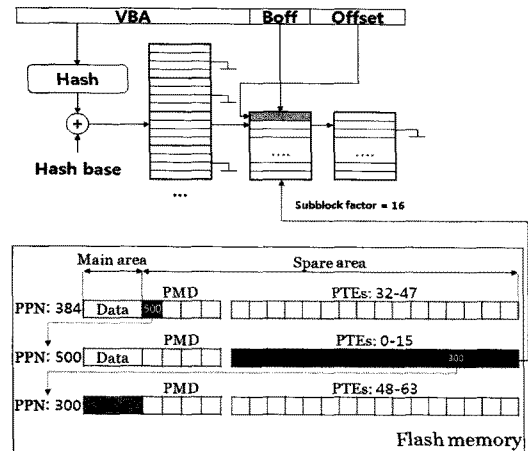


그림 5. LRU 군집형 해시 테이블
Fig. 5. LRU Clustered Hash Table.

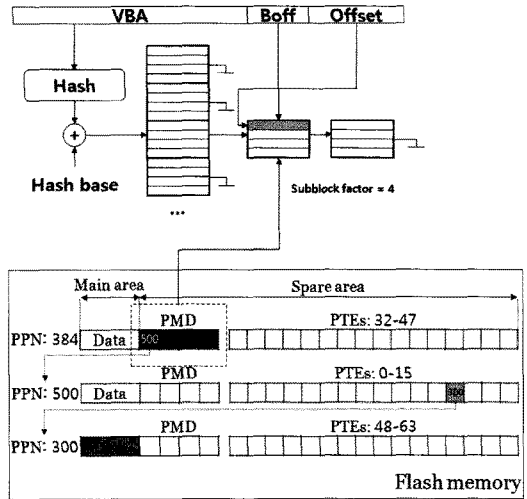


그림 6. PMD 군집형 해시 테이블
Fig. 6. PMD Clustered Hash Table.

Subblock factor라고 일컫는다.

LRU 군집형 해시 테이블에서는 Subblock factor를 16으로 하여 그림 5와 같이 참조된 PMD와 관련된 연속적인 주소들인 PTE를 저장하여 주소 변환 성능을 향상시킨다. 최근에 참조된 주소와 이 주소와 관련된 연속적인 주소를 저장하므로 시간, 공간 지역성을 고려하여 적중률을 증가시킬 수 있다. LRU 군집형 해시 테이블을 모두 사용한 경우 LRU 기법에 따라 자주 사용되지 않았던 주소를 PMD 군집형 해시 테이블로 강등한다.

1.2. PMD 군집형 해시 테이블

LRU 군집형 해시 테이블에서 강등된 주소는 사용되었지만 추후 사용될 가능성이 있는 주소이기 때문에 PMD 군집형 해시 테이블에 PMD 주소를 섹터 사상으로 관리한다. 이 때 한 블록당 4개의 PMD 값을 가지므로 Subblock factor는 4로 설정한다.

그림 6의 PMD 군집형 해시 테이블은 LRU 군집형 해시 테이블에서 강등된 주소와 관련된 PMD 값을 저장하므로 LRU 군집형 해시 테이블보다 많은 주소들을 캐시할 수 있다. 요청된 논리 주소를 PMD 군집형 해시 테이블에서 찾을 경우 플래시 메모리의 PTE를 한 번만 참조하여 주소 변환을 할 수 있다. 또한 PMD 군집형 해시 테이블을 통하여 읽은 PTE들을 LRU 군집형 해시 테이블에 저장하여 다음 주소 변환이 빠르게 이루어지도록 하여 LRU 군집형 해시 테이블 구성을 보완한

다. PMD 군집형 해시 테이블에서 주소 검색이 실패했을 경우 기존의 Superblock scheme와 같이 PGD 테이블부터 검색하며, 플래시 메모리에 저장된 PMD와 PTE를 검색하여 주소를 반환한다. 요청된 주소에 대한 PTE를 LRU 군집형 해시 테이블에 저장하여 다음 주소 변환에 대해 빠른 수행이 가능하게 한다.

2. 가비지 컬렉션

EGFTL의 가비지 컬렉션은 무효 블록 관리자를 통해 이루어진다. 무효 블록 관리자는 다중 블록 삭제를 사용하여 삭제될 희생 블록들을 관리하기 위한 삭제 그룹 리스트(Erase group list)와 무효 블록들을 효율적으로 수집하기 위한 무효 블록 수집기로 나누어진다. 또한 빠른 무효 블록 수집을 위해 2단계 갱신 블록을 사용한다. 삭제 그룹 리스트가 완성된 경우 가비지 컬렉션이 발생하여 삭제 그룹 리스트의 정보를 통해 다수의 블록을 삭제한다. 다수의 블록들을 가비지 컬렉션의 삭제 단위로 사용하므로 삭제시간을 감소시킬 수 있다.

2.1 무효 블록 관리자

다중 블록 삭제 기능을 통해 가비지 컬렉션 성능을 높이기 위해 다음과 같이 두 가지를 고려해야 한다. 첫 번째, 다중 블록 삭제 성능을 최대화하기 위해서는 한번에 삭제할 수 있는 최대 블록 수 단위로 가비지 컬렉션이 실행되어야 한다. 두 번째, 삭제하기 위한 다수의 블록들을 선택하기 위한 탐색시간을 최소화하여야 한다. 기존의 NAND 플래시 변환 계층에서는 블록 수에 비례하여 삭제 시간이 증가하였으므로, 이러한 점을 고려할 필요가 없었다. 따라서 다중 블록 삭제 기능을 효율적으로 사용하기 위해 무효 블록 관리자를 사용한다.

무효 블록 관리자는 그림 7 같이 무효 블록 수집기를 통하여 수집된 무효 블록들을 삭제 그룹 리스트로 관리하여 리스트가 모두 찬 경우 가비지 컬렉션을 실행하여 삭제 시간을 최소화 한다.

2.2 무효 블록 수집기

무효 블록 수집기는 2단계 갱신 블록을 사용하여 무효 블록들을 탐색하는 시간을 줄인다.

2단계 갱신블록은 단일 갱신 블록과 다중 갱신 블록으로 구성된다. 단일 갱신 블록은 데이터 블록의 데이터가 한번 갱신되는 경우 사용하는 블록이다. Superblock에 할당된 블록과 동일한 수의 블록을 사용한다.

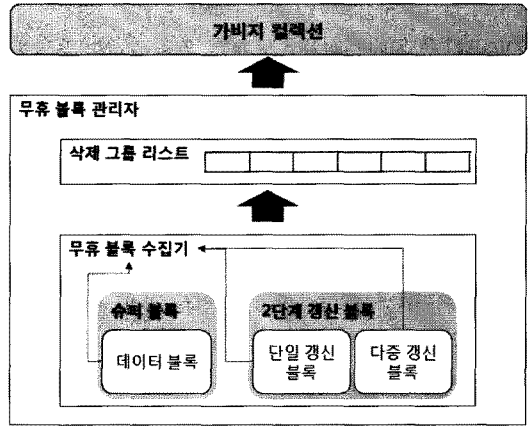


그림 7. 무효 블록 관리자
Fig. 7. Manager of invalid blocks.

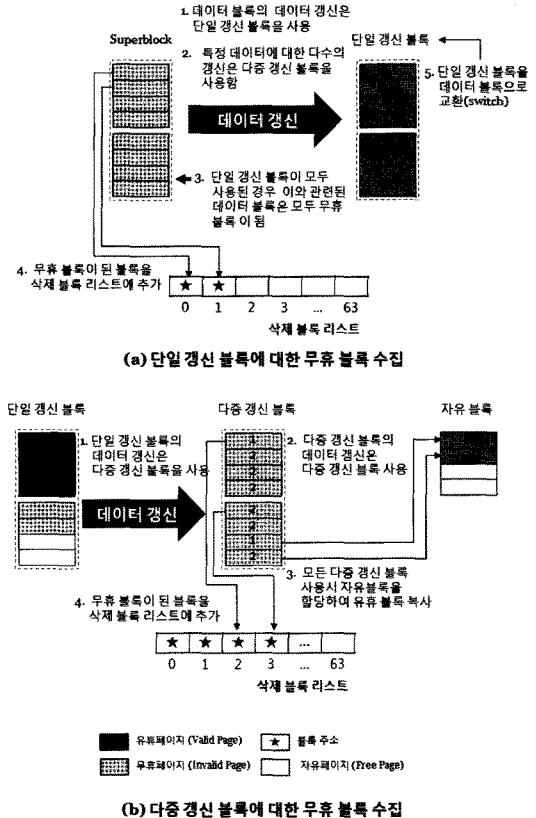


그림 8. 무효 블록 수집
Fig. 8. Method of invalid block collection.

다중 갱신 블록은 단일 갱신 블록과 다중 갱신 블록에서 데이터 갱신이 발생할 경우 사용하는 블록이다. 다중 갱신 블록을 통해 Hot 데이터를 구분하여 관리할 수

있다.

그림 8은 2단계 갱신 블록을 사용하여 무효 블록을 수집하는 방법을 나타낸다. 무효 블록 수집 시기는 단일 갱신 블록이 모두 사용된 경우와 다중 갱신 블록이 모두 사용된 경우에 수집한다.

단일 갱신 블록은 데이터가 한번 갱신이 되고 데이터 블록 수가 동일하므로 데이터 블록의 데이터가 모두 한번씩 갱신된 경우, 즉 데이터 블록이 모두 무효 데이터가 된 경우 단일 갱신 블록이 모두 사용이 된다. 따라서 그림 8(a)와 같이 단일 갱신 블록이 모두 사용 되었을 때, 데이터 블록으로 사용되었던 블록들의 주소를 삭제 그룹 리스트에 추가한다.

다중 갱신 블록은 자주 갱신되는 데이터가 저장되므로 다중 갱신 블록이 모두 사용된 경우 무효 데이터 비율이 높다. 따라서 그림 8(b)와 같이 다중 갱신 블록이 모두 사용된 경우 다중 갱신 블록에 존재하는 유효 데이터들을 자주 갱신되는 Hot 데이터로 간주하고 자유

블록에 복사하여 다중 갱신 블록으로 활용한다. 기존에 다중 갱신 블록으로 사용된 블록은 모두 무효 블록이 되므로 삭제 블록 리스트에 추가한다.

2단계 갱신 블록을 사용하기 위해 EGFTL은 그림 9 (c)의 PBMT를 이용한다. Superblock scheme는 스페어 영역의 용량 한계의 제약으로 인해 Superblock 마다 8개의 블록 번호를 PBMT(Physical block mapping 테이블)에 저장하여 block index를 통해 그에 해당하는 블록 주소를 사용한다. EGFTL은 PBMT를 세 영역으로 구분하여 2단계 갱신 블록을 사용한다. PBMT의 PBN 0~PBN2는 데이터 블록, PBN3~PBN6은 단일 갱신 블록, PBN7은 다중 갱신 블록으로 사용한다. 단일 갱신 블록이 모두 사용되면 데이터 블록을 가리키는 PBN 0~PBN2가 무효 블록 수집기에 의해 삭제 그룹 리스트에 저장된다. 다중 갱신 블록이 모두 사용되면 다중 갱신 블록을 가리키는 PBN7이 삭제 그룹 리스트에 저장 된다.

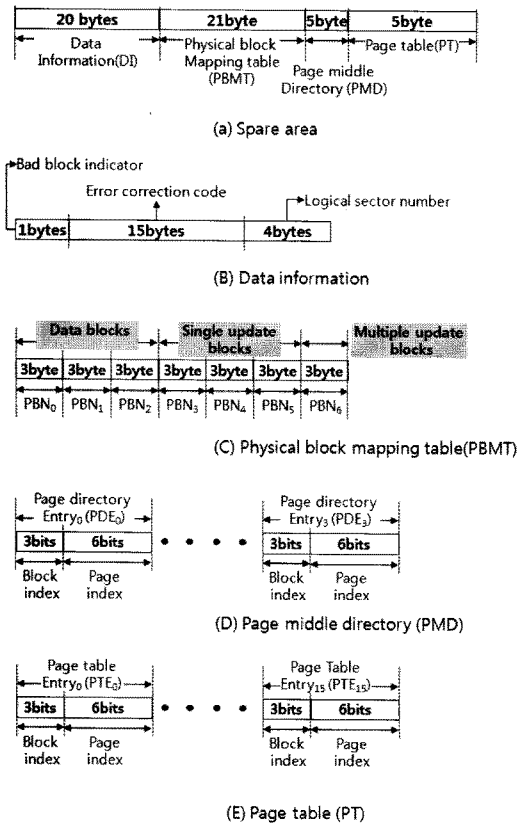


그림 9. EGFTL의 스페어 영역의 구조
Fig. 9. The format of the spare area in the EGFTL.

2.3 가비지 컬렉션 시기

EGFTL의 가비지 컬렉션 시기는 세 가지 경우로 나눈다. 첫 번째 삭제 블록 리스트가 모두 찬 경우, 두 번째 단일 갱신 블록이 모두 사용된 경우, 세 번째, 가용 공간이 부족한 경우로 나눈다.

삭제 블록 리스트가 모두 찬 경우 리스트에 있는 무효 블록들을 다중 블록 삭제를 사용하여 한 번에 삭제한다. 두 번째와 같이 단일 갱신 블록이 사용률이 적을 경우 플래시 메모리의 사용 가능한 자유 블록 수를 감소시키므로 단일 갱신 블록의 총 개수를 제한하여 관리한다. 단일 갱신 블록의 수가 제한된 값을 넘게 되면, 무휴 시간(idle time)에 단일 갱신 블록 수가 많은 Superblock을 검색하여, 단일 갱신 블록과 데이터 블록을 병합한다. 병합 후 생성된 무효 블록들을 희생 블록 리스트에 추가하며, 희생 블록 리스트가 완성될 때까지 반복한다. 세 번째는 데이터를 쓰기 위한 용량이 부족한 경우로 먼저 희생 블록 리스트의 블록을 삭제하며, 필요에 따라 단일 갱신 블록과 데이터 블록을 병합하여 희생 블록 리스트를 구성하여 가용 공간을 확보한다.

IV. 성능 평가

EGFTL의 성능은 주소 변환 성능, 가비지 컬렉션 성능으로 나누어 평가한다. 실험은 리눅스 2.6.17 환경에

표 1. OneNAND™ 에뮬레이터 정보
Table 1. Information of OneNAND™ emulator.

	OneNAND™ 에뮬레이터
Flash size	256MB
Page size	2KB
OOB area size	64bytes
Pages number	131072
Block number	2048
Pages/sector	64

서 실행하였으며, MTD(Memory Technology Device)에서 제공하는 표 1의 OneNAND™ 에뮬레이터와 앤드류 벤치마크(Andrew benchmark), 포스트마크(Postmark)를 사용하여 평가하였다^[8~10].

1. 주소 변환 성능 평가

Superblock scheme와 EGFTL은 플래시 메모리의 스페어 영역에 주소 변환 테이블을 저장하는 플래시 변환 계층이다. 따라서 스페어 영역을 읽는 오버헤드가 존재하며 이러한 오버헤드를 얼마나 효율적으로 제거했는지에 대한 비교 대상으로 주소 변환 테이블을 램에 유지하는 플래시 변환 계층이 필요하다. 본 논문에서는 혼합 사상을 사용하고 주소 변환 테이블을 램에 유지하는 플래시 전환 계층 중 성능이 뛰어나고 널리 알려진 BAST와 FAST와의 비교를 통해 EGFTL의 성능을 비교한다^[4~5].

EGFTL의 주소 변환 성능을 평가하기 위해 스토리지

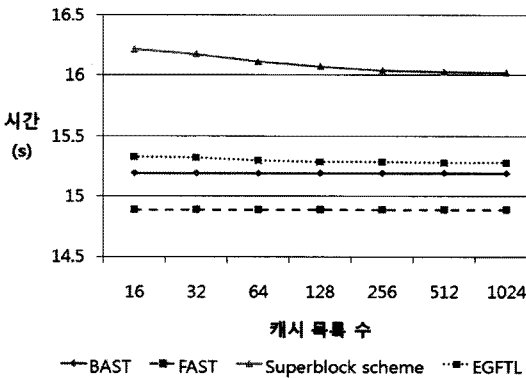
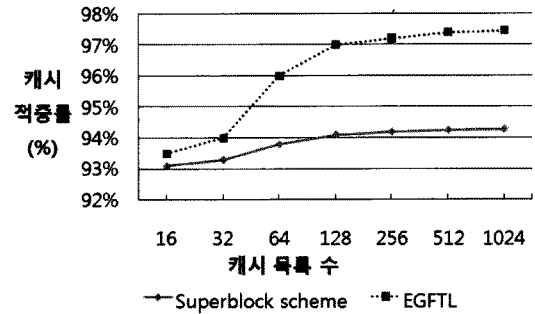


그림 10. 앤드류 벤치마크 실험 결과
Fig. 10. Result of Andrew benchmark.

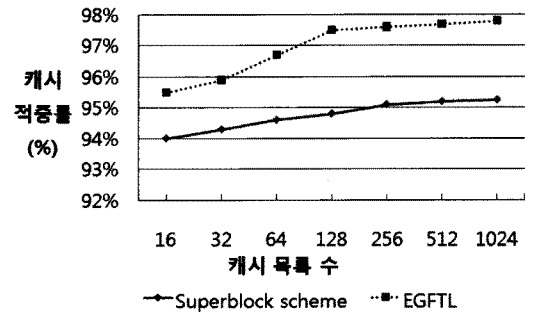
시스템에서 성능 벤치마킹 방식으로 잘 알려진 앤드류 벤치마크를 사용하였으며, 앤드류 벤치마크 완료 시간 및 캐시 적중률에 대해 비교하였다. 캐시 적중률은 주소 요청에 대한 EGFTL과 Superblock scheme에 사용된 LRU 캐시 테이블 목록에 적중된 비율이다. 앤드류 벤치마크는 다음과 같으며 주소 변환 성능이 읽기/쓰기 성능에 미치는 영향을 확인하는데 효과적이다^[9].

- 첫째, 약 100개의 디렉터리 생성
- 둘째, 멀티미디어 데이터를 반복 복사
- 셋째, 재귀적으로 디렉터리 상태 확인
- 넷째, 각 파일의 내용을 검색
- 다섯째, 컴파일하기

EGFTL과 Superblock scheme의 캐시 목록 수를 달리 하여 앤드류 벤치마크 테스트를 10회 실행한 결과 EGFTL이 BAST와 FAST 기법과 3% 이하의 거의 비슷한 성능을 보이는 것을 그림 10의 실험 결과를 통해 알 수 있다. 이는 EGFTL의 해시 테이블을 통해 스페



(a) 쓰기 요청에 대한 캐시 적중률



(b) 읽기 요청에 대한 캐시 적중률

그림 11. 캐시 적중률 결과
Fig. 11. Result of cache ratio.

어 영역의 참조 횟수를 감소시켜 램에 주소 변환 테이블을 사용하는 플래시 전환 계층인 BAST, FAST와 유사하게 동작할 수 있음을 보여준다. BAST, FAST와 성능의 차이는 스페어 영역을 읽는 부하와 해시 테이블을 통해 연산되는 과정으로 인해 발생하는 차이지만 효과는 미비함을 알 수 있다.

EGFTL은 Superblock scheme보다 5% 빠른 성능을 갖으며, 이는 그림 11에 나타난 캐시 적중률 향상의 결과로 볼 수 있다. 캐시 적중률은 최대 3% 높지만 캐시 적중률이 높을수록 플래시 메모리를 두 번 읽는 부하를 크게 감소시킬 수 있으므로 주소 변환 성능 향상이 높음을 알 수 있다. 예를 들어 주소 변환을 100번 실행할 경우 적중률 1%의 차이는 스페어 영역의 읽기 횟수의 차가 2회가 나므로 비동기 읽기 모드를 기준으로 152ns의 시간이 차이가 나게 된다.

2. 가비지 컬렉션 성능 평가

가비지 컬렉션 성능을 평가하기 위해 포스트마크 테스트를 사용하여 삭제 성능을 비교한다.

포스트마크는 작은 크기의 파일들에 대한 쓰기, 읽기, 변경, 삭제 등의 트랜잭션을 수행한다. 파일의 개수는 100개이며, 수행할 트랜잭션은 각각 300, 500로 설정하였다^[10]. 포스트 마크를 통하여 삭제 횟수와 쓰기 횟수, 삭제 시간, 가비지 컬렉션 시간을 측정하였다. 삭제 횟수는 다중 블록 삭제와 단일 블록 삭제 명령 횟수 합을 의미하며, 삭제 시간은 해시 테이블을 통해 연산되는 과정을 포함하지 않고 다음과 같이 계산된다.

$$\text{삭제 시간} = \text{다중 블록 삭제 명령 요청 횟수} * 4\text{ms} + \text{단일 블록 삭제 명령 요청 횟수} * 2\text{ms}$$

또한 삭제 횟수 단축 및 다중 블록 사용 횟수를 증가시켜 삭제 시간을 BAST와 FAST 보다 최대 94%, Superblock scheme보다 최대 90% 단축시켰다.

그림 13은 쓰기 횟수를 나타낸 실험 결과이다. EGFTL은 3단계 주소 변환과 2단계 갱신 블록을 통하여 병합 횟수를 줄이고 Hot 데이터를 분리하여 쓰기 횟수를 감소시킬 수 있으며, 이를 그림 13의 쓰기 횟수를 나타낸 실험 결과로 알 수 있다. EGFTL이 BAST와 FAST보다 최대 32% 쓰기 횟수를 감소시켰으며, Superblock scheme보다 20% 쓰기 횟수를 감소시켰다.

그림 14은 가비지 컬렉션 시간을 나타낸다. EGFTL

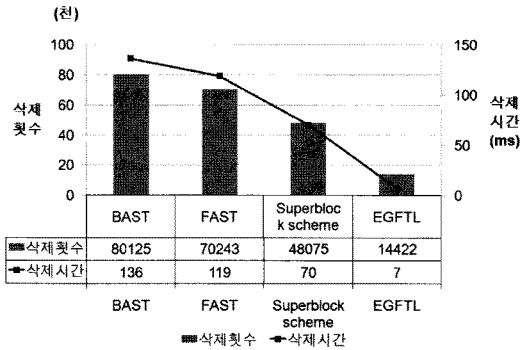


그림 12. 삭제 횟수 및 삭제 시간 결과
Fig. 12. Result of the number of erase operation.

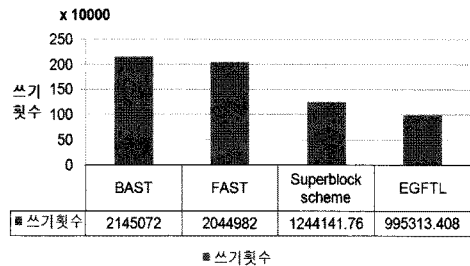


그림 13. 쓰기 횟수 결과
Fig. 13. Result of the number of write operations.

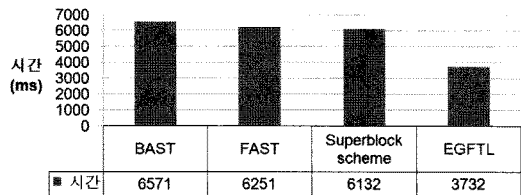


그림 14. 가비지 컬렉션 소모 시간
Fig. 14. The time of garbage collection.

이 BAST와 FAST보다 43%, Superblock scheme보다 39% 향상되었다. 삭제 횟수와 삭제 시간 그리고 쓰기 횟수를 줄여 가비지 컬렉션 소모 시간을 줄일 수 있음을 알 수 있다.

IV. 결 론

퓨전 플래시 메모리는 기존의 NAND 플래시 메모리의 단점을 극복하여 휴대용 임베디드 시스템을 위한 유비쿼터스 저장장치로 기대되고 있다. 대표적인 퓨전 플래시 메모리인 OneNAND™는 혼합형 구조의 특징

뿐만 아니라 다중 블록 삭제 기능을 추가하여 삭제 성능을 향상시켰다. 하지만 다수의 희생 블록을 관리하는 방법에 대한 연구가 미비하여 다중 블록 삭제 기능을 효율적으로 사용하기 어렵다. 본 논문에서는 3단계 주소 사상 방법을 사용하는 Superblock scheme과 군집형 해시 테이블 무효 블록 관리자를 통해 빠른 주소 변환과 가비지 컬렉션 성능을 제공하였다. 실험 결과 Superblock scheme보다 주소 변환 성능을 5% 향상시켰으며, 가비지 컬렉션 성능을 기존보다 39%이상 향상시켰다. EGFTL를 통해 가비지 컬렉션으로 인해 발생하는 플래시 메모리의 성능 저하를 감소시킬 수 있을 것을 알 수 있다.

참고 문헌

- [1] Samsung Electronics Co. Ltd. "OneNANDTM Features & Performance," http://www.samsung.com/Products/Semiconductor/Memory/appnote/onenand_features_performance_051104.pdf, 2005.
- [2] 이동환, 조원희, 김덕환, "다중 블록 지우기 기능을 적용한 퓨전 플래시 메모리의 FTL 성능 측정 도구 설계 및 구현," 대한전자공학회, 제31권, 제1호, pp.647-648, 2008년 6월
- [3] Intel Corporation, "Understanding the Flash Translation Layer Specification," <http://developer.intel.com>, 1998.
- [4] Jesung Kim, Jong Min Kim, Sam H. Noh, Sang Lyul Min, and Yookun Cho, "A space-efficient flash translation layer for compact flash systems," IEEE Transactions on Consumer Electronics, vol. 48, no. 2, pp. 366-375, 2002.
- [5] Sang-Won Lee and Dong-Joo Park, "A log buffer based flash translation layer using fully associative sector translation," ACM Transactions on Embedded Computing Systems, vol. 6, no. 3, 2007.
- [6] Jeong-Uk Kang, Heeseung Jo, Jin-Soo Kim, and Joonwon Lee, "A superblock-based flash translation layer for NAND flash memory," ACM Transactions on Embedded Computing Systems, vol. 6, no. 3, 2007.
- [7] Madhusudhan Talluri, Mark D. Hill and Yousef A. Khalidi, "A new page table for 64-bit address spaces," In proceedings of Symposium of Operating System Principles, Dec 1995.
- [8] MTD, "Memory Technology Device subsystem for Linux," <http://www.linux-mtd.infradead.org>
- [9] John H. Howard, Michael L. Kazar, Sherri G. Menees, David A. Nichols, M. Satyanarayanan, Robert N. Sidebotham, and Michael J. West, "Scale and Performance in a Distributed File System," ACM Transactions on Computer Systems, vol. 6, pp.51-81, Feb 1988.
- [10] Postmark, <http://packages.debian.org/stable/utils/postmark>
- [11] Kwanghee Park, Junsik Yang, Joon-Hyuk Chang, and Deok-Hwan Kim, "Anticipatory I/O Management for Clustered Flash Translation Layer in NAND Flash Memory," ETRI Journal, vol.30, no.6, Dec. 2008, pp.790-798

저 자 소 개



이동환(학생회원)
2008년 인하대학교 전자공학과
학사 졸업.
2008년~현재 인하대학교
전자공학과 석사과정
<주관심분야 : 임베디드시스템,
스토리지 시스템>



김덕환(정회원)-교신저자
2003년 한국과학기술원 컴퓨터
공학 박사.
2006년~현재 인하대학교
전자공학부 부교수
<주관심분야 : 시각정보처리, 스
토리지 시스템, 임베디드 시스템>



조원희(학생회원)
2007년 명지대학교 정보통신공학
부 통신공학과 학사 졸업.
2007년~현재 인하대학교
전자공학과 석사과정
<주관심분야 : 추천시스템, 시스
템 소프트웨어>