

## 비휘발성 메모리 소자를 이용한 낸드 플래시 파일 시스템의 부팅시간 개선 기법

전병길<sup>0,1,2</sup> 김은기<sup>1,2</sup> 신형중<sup>1,2</sup> 원유집<sup>1</sup>

<sup>1</sup>한양대학교 전자통신컴퓨터공학부 분산 멀티미디어 연구실

<sup>2</sup>삼성 전자

{ugfman<sup>0</sup>, zerobit, newbell, yjwon}@ece.hanyang.ac.kr

### Boosting Up the file system mount latency using Byte Addressable NVRAM

Byung-gil Jeon<sup>0,1,2</sup> Eun-ki Kim<sup>1,2</sup> Hyungjong Shin<sup>1,2</sup> Youjip won<sup>1</sup>

<sup>1</sup>DMC Lab, Dept. of Electronics and Computer Engineering, Hanyang University

<sup>2</sup>Samsung Electronics, Co., Ltd

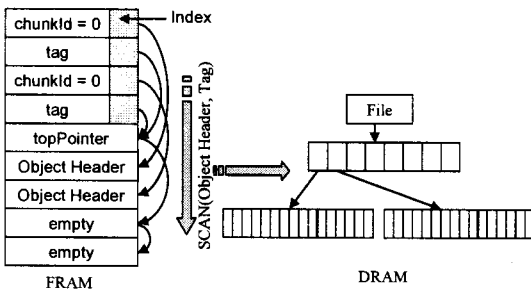
오늘날 휴대용 멀티미디어 장치의 저장 매체로서 가장 각광 받고 있는 낸드 플래시 메모리는 MP3, 디지털 카메라, 휴대폰, 휴대용 오라기, 노트북, 등에 사용되고 있다. 그런데, 낸드 플래시 메모리 장치의 고유 특성인 Erase 동작 및 제한적인 Erase 횟수, Erase 후 쓰기 특성 등으로 인하여 시스템에 장착하여 사용하기 위해 FTL, YAFFS, JFFS 등과 같은 소프트웨어와 같이 사용되고 있다. 이들 가운데 YAFFS나 JFFS는 성능 면에서 뛰어나지만 마운트 시간이 많이 소모된다는 단점이 있다. 이것은 낸드 플래시 메모리 저장장치의 용량이 증가하면 할수록 더 많은 마운트 시간을 필요로 한다. 본 연구에서는 YAFFS와 같은 LFS 구조의 파일 시스템의 마운트 시간을 단축시키기 위하여 메타 데이터를 비휘발성, 저 소비전력, 고속 읽기/쓰기, random access 등의 특성을 갖는 FRAM[1]에서 관리 하도록 구현 하였다. 이전 연구인 YAFFS를 기반으로 하는 FRASH[2](FRASH1.0 으로 명명함)에서는 마운트 시간은 기존 YAFFS 대비 5 배 이상을 감소 시켰으나, 파일의 생성과 삭제의 성능 저하가 발생 하였다. 본 논문에서는 이를 개선하기 위하여 FRAM에 저장되는 메타 데이터를 효율적으로 관리 할 수 있는 최적의 NVRAM 자료구조를 도입하여 성능 저하를 최소화 하고, FRAM을 사용함에 따라 불필요한 기능들을 제거함으로써 마운트 시간 측면에서도 FRASH1.0 에 비해 개선을 한 FRASH1.5 를 구현 하였다.

이전 연구인 FRASH1.0 은 낸드 플래시 메모리의 Spare 영역과 Object Header 영역을 FRAM에 지정된 Tag 영역과 Object Header 영역에 각각 저장을 하였다. FRAM에 저장된 Tag는 낸드 플래시 메모리의 데이터 영역과 일대일로 대응되지만, Object Header는 파일의 수에 따라 가변적이기 때문에 고정된 영역으로 지정을 할 수 없었다. 따라서 각각 Tag에 추가적인 Object Header Pointer 영역을 생성하여 Tag의 Chunk Id가 "0"일 때 Object Header Pointer가 Object Header 영역을 지정하도록 구성 하였다. 이렇게 구성된 FRASH1.0 은 마운트 성능은 향상 되었지만, 파일 생성시 Object Header를 저장하기 위하여 FRAM에 지정된 Object Header 영역의 빈 공간을 찾는데 추가적인 시간이 필요하기 때문에 기존 YAFFS 대비 5~10%의 성능 감소가 발생 하였다. 또한 FRAM에 저장된 Tag의 구동은 낸드 플래시의 Tag 구조를 그대로 사용하여 구동하였기 때문에 이 부분을 FRAM에 적합하게 최적화 할 필요가 있었다.

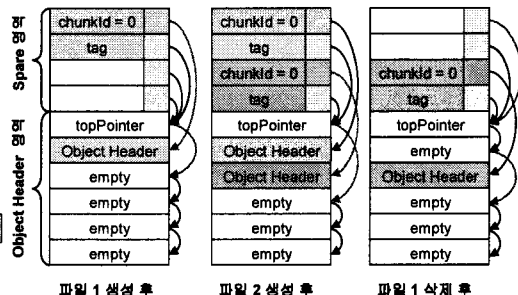
본 논문에서는 FRASH1.0 에서 미진한 부분인 FRAM 자료구조를 최적화 하고, 파일 생성/삭제 성능 저하를 개선한 FRASH1.5 를 소개한다. FRASH1.0 에서의 성능 저하를 개선하기 위하여 다음의 두 가지 측면에서 접근 하였다. 첫 번째는 FRAM과 같은 NVRAM은 RAM의 특성을 지원하기 때문에 낸드 플래시 메모리와 달리 결함을 포함하지 않는 무결성을 제공한다. 따라서 기존 YAFFS에서 낸드 플래시 메모리 사용에 적합하도록 구성된 Spare 영역내의 각 항목 가운데 FRAM을 사용함으로써 불필요한 항목을 제거하여 메모리 사용 효율을 증가 시키고, Workload를 감소시키는 측면에서 접근 하였다. 기존 YAFFS에서 낸드 플래시의 Spare 영역에 구성된 메타 데이터의 구조에서 무결성을 지원하는 FRAM에 메타 데이터를 저장 할 경우 필요 없는 tag ecc를 제거하고, 이것을 Object Header의 위치를 지정하는 index로 활용 하였다.[그림 1] 두 번째는 FRASH1.0 은 FRAM에서 Object Header 정보를 읽거나 쓸 경우 반드시 Tag, Object Header Pointer, Object Header, 3 개의 영역을 구동 시켜야 한다. 이것은 마운트를 위한 스캔 동작뿐만 아니라 파일 생성 및 삭제 시에 성능을 저하시키는 요인으로 작용한다.

또한, 파일 생성시 Object Header 영역에서 빈 공간을 찾기 위한 스캔은 파일 개수에 따라 그 성능 저하가 더욱 크게 나타나게 된다. 따라서 FRAM을 Spare 영역과 Object Header 영역으로 나누고, Object Header 영역의 시작 주소를 topPointer로 지정하였다. index는 Object Header를 지정하도록 구성하였으며, 그것을 지정하고 있지 않을 경우에는 항상 Object Header 영역의 topPointer를 지정하게 구성되어 있다. topPointer 주소 공간에는 항상 빈 Object Header 주소를 지정하고 있다. 각 Object Header들은 topPointer를 기준으로 모두 연결 되어 있으며, 빈 Object Header가 Object Header 영역의 어디에 위치하든 관계없이 topPointer 에서 한번의 Workload로 지정 될 수 있다.

FRASH1.5 의 마운트는 FRASH1.0 과 유사하지만 Object Header를 찾기 위하여 Object Header Pointer를 추가로 읽어야 하는 과정이 필요 없고, tag ecc를 제거함으로써 모든 tag에서 수행되는 tag ecc 확인 과정이 삭제되었다. 따라서 마운트에 필요한 최소의 동작만 수행하기 때문에 많은 시간을 감소시킬 수 있었다. 파일 생성은 Spare 영역에 Tag 정보가 기록되고, chunkId=0 으로 설정된 Tag 영역에 있는 index는 topPointer가 가지고 있는 empty 영역을 Object Header로 지정하고, topPointer는 이 empty 영역이 가지고 있던 다음 empty 영역의 위치를 지정하도록 수정되고. Object Header 정보 및 Tag 정보를 기록한다. 하나의 파일이 생성된 후에 또 다른 새로운 파일 2 의 생성이 진행 될 경우, topPointer는 이미 Object Header 영역의 empty 영역을 지정 하고 있기 때문에 첫 번째 파일 생성과 동일한 시간으로 진행 될 수 있다. 파일을 삭제 할 경우 Spare 영역의 정보를 수정하고, topPointer는 삭제된 Object Header 영역을 지정하고, topPointer가 가지고 있던 empty 영역의 정보를 삭제된 Object Header 영역에 넘겨준다. 따라서 삭제된 Object Header 영역은 empty 상태가 되고, 그 다음 empty 영역을 지정하게 된다. 이후에 Tag 영역의 index가 topPointer를 지정 하도록 수정 한다.[그림 2]



[그림 1] FRASH1.5 구조 및 마운트 동작



[그림 2] FRASH1.5 파일 생성/삭제

이렇게 구성된 FRASH1.5 는 SMDK2440 보드에 구성하여 성능을 평가한 결과 마운트 동작 시간은 이전 연구인 FRASH1.0 대비 2 배 이상 단축시킬 수 있었고, 파일 생성 성능은 FRASH1.0 대비 3 ~ 8% 향상 되었다. 특히, 파일 크기가 크고, 개수가 많을 경우는 기존 YAFFS 대비 성능 저하 없이 마운트 시간을 8 배 이상 감소시킬 수 있었다.

본 연구의 목표는 낸드 플래시용 파일 시스템의 고질적인 문제점인 파일 시스템 마운트 지연시간을 개선하는 것이다. 이를 위하여 바이트 접근성을 가지는 비휘발성 메모리 소자와 낸드 플래시로 계층적 저장시스템을 구성하고, 바이트 접근성을 가지는 비휘발성 메모리 소자 상에 파일 시스템 메타 데이터를 위치시키는 기법을 개발하였다. 비휘발성 메모리 소자에 메타 데이터를 위치시킴으로써 낸드 플래시에 저장된 메타 데이터를 관리하기 위해 Spare 영역과 Object Header 영역에 구성된 항목가운데 불필요한 항목을 제거할 수 있었다. Object Header의 자료구조를 효율적으로 설계하여 Object Header를 스캔 하는 단계를 제거하였다. 본 연구 결과는 FRAM이 낸드 플래시 저장장치의 성능 향상에 크게 기여하는 한 단면을 보여준 것으로 향후 많은 분야에 응용될 수 있는 가능성을 보여 주고 있다.

**참고 문헌**

[1] Kang, Y.M. 외 다수, "World Smallest 0.34um<sup>2</sup> COB Cell 1T1C 64Mb FRAM with New Sensing Architecture and Highly Reliable MOCVD PZT Integration Technology", Symposium on VLSI Technology Digest of Technical Papers, pp.124-125, 2006  
 [2] Eun-ki Kim, Hyungjong Shin, Byung-gil Jeon, Seokhee Han, Jaemin Jung, Youjip Won, "FRASH: Hierarchical File System for FRAM and Flash", ICCSA 2007, Malaysia, pp. 238-251, 2007