

# NAND 플래시 메모리의 효율적 사용을 위한 접근계층의 설계 및 구현

박정태<sup>o</sup> 최운선 김성조  
중앙대학교 컴퓨터공학부  
{blueidev<sup>o</sup>, windchoi, sjkim}@konan.cse.cau.ac.kr

## Design and Implementation of NAND Flash Memory Access Layer

Jeong Tae Park<sup>o</sup> Mun Seon Choi Sung Jo Kim  
School of Computer Science and Engineering, Chung-Ang University

### 요 약

최근 소형 모바일 기기들이 대중화되고 그 종류가 다양해지면서 플래시 메모리가 기본 저장 매체로서 많이 사용되고 있다. 플래시 메모리는 기존의 하드디스크 같은 자기 매체에 비해서 크기가 작고, 전력소모도 적으며 내구성도 높다. 멀티미디어 데이터를 다루는 기기들이 증가하면서 플래시 메모리 중에서도 비용이 저렴하고 단일 칩으로도 대용량을 가지는 NAND형 플래시 메모리를 저장장치로 사용하는 기기들이 계속해서 늘어나고 있다. NAND 플래시 메모리는 기존에 많이 사용되던 NOR 플래시 메모리와는 다른 많은 특징이 있다. 따라서 NAND 플래시 메모리에 적합한 저장 기법을 설계하기 위해서는 NAND 플래시 메모리의 특징을 잘 이해하고 이용해야 한다. 이에 본 논문에서는 NAND 플래시 메모리를 효율적으로 사용할 수 있도록 해주는 접근계층을 설계, 구현하고 이에 대한 구조와 세부 특징에 대해서 살펴본다. 본 논문에서 구현한 접근계층은 하드웨어에 종속적이지 않으며 NAND 플래시 메모리가 제공하는 다양한 기능을 상위 계층에서 충분히 활용할 수 있도록 설계되었다.

### 1. 서 론

플래시 메모리는 높은 신뢰성과 집적도, 낮은 소비전력 등의 특징을 가지는 반도체 형태의 저장매체로 장치의 크기나 전원공급장치의 한계, 이동성 등의 특징을 가지는 무선 및 모바일, 핸드헬드, 임베디드 기기의 저장매체로 많이 사용되고 있다. 또한 플래시 메모리는 디지털 카메라나 MP3 플레이어, 미디어 플레이어 같은 멀티미디어 장비의 저장매체로도 많이 사용되며 다양한 형태의 제품이 널리 사용되고 있다.

플래시 메모리는 EEPROM(Electrically Erasable Read Only Memory)의 한 형태로 현재 주로 사용되고 플래시 메모리는 NOR형과 NAND형이 있다. 이 두 가지 형태의 플래시 메모리는 몇 가지의 공통점과 차이점을 가지고 있는데 블록단위의 내부구성, 삭제/쓰기 연산의 제약 등이 플래시 메모리로서 가지는 공통점이고 단위 셀의 크기 및 집적도, 외부 인터페이스, 전력소비, 배드블록 존재여부 등의 차이점을 가지고 있다.

최근의 많은 응용기기들은 대용량의 저장매체를 필요로 하는 경우가 많다. 예를 들어, 셋톱박스(Set-Top Box)는 각종 소프트웨어, 시스템 또는 사용자 데이터(사용자 프로파일, 설정, 시스템 레지스터, 업데이트 가능한 시스템 파일 등) 그리고 비디오 스트림 등을 저장하기 위해 많은 저장 공간을 필요로 한다. 또한 휴대폰이나 PDA와 같은 이동형 기기들에도 멀티미디어 데이터에 대한 수요가 늘어남에 따라 이를 저장하기 위해 많은 저장 공간을 필요로 하고 있으며 다양한 기능을 내장하기 위해 보다 많은 시스템 자원을 저장할 저장 장치의 필요가 증대하고 있다. NOR 플래시 메모리는 집적도가 낮고 고가이기 때문에 이러한 요구에 적합하지 못하다. 반면에 NAND 플래시 메모리는 단일 칩으로도 대용량이며 비용도 저렴하다. 따라서 NAND 플래시 메모리를 이용하여 응용기기를 만들게 되면 기본 저장용량을 크게 할 수 있으며 기기의 비용도 낮출 수 있게 된다. 또한 최근의 출시되는 많은 수의 마이크로 컨트롤러들은 NAND 플래시 메모리를 제어할 수 있는 기능을 탑재하고 있어 NAND 플래시 메모리로부터 직접 부팅이 가능하므로 기기는 부가

적인 부트장치 없이 NAND 플래시 메모리만으로 구성할 수 있다.

본 논문에서는 NAND 플래시를 시스템의 기본 저장 장치로 사용한 기기들에서 이용할 수 있는 NAND 플래시 메모리 접근계층의 설계 및 구현에 관해 논의한다. 이 접근계층은 다양한 NAND 플래시 메모리를 지원하며 NAND 플래시 메모리만이 가지는 독특한 기능을 상위계층에 제공하여 NAND 플래시 메모리를 효과적으로 사용할 수 있도록 한다. 설계 및 구현한 계층은 파일시스템과 플래시 메모리 사이의 미들웨어인 플래시 변환계층(Flash Translation Layer, FTL)[1,6]보다 플래시 메모리 전용 파일시스템[5,7,9]에 적합하도록 설계되었다.

본 논문의 구성은 다음과 같다. 2장에서는 NAND 플래시 메모리와 그것의 특징적인 기능에 대해 알아보고 3장에서는 설계 및 구현한 NAND 플래시 메모리 접근계층에 대해 살펴본 후 4장에서는 결론을 맺고 향후 연구과제에 대해 이야기한다.

### 2. NAND 플래시 메모리

NOR형 플래시 메모리는 일반적인 메모리처럼 시스템의 주소 버스와 데이터 버스에 바로 연결할 수 있는 버스형태의 외부인터페이스를 가지고 있으므로 일반적인 메모리에 접근하는 것과 같은 방식으로 직접액세스가 가능하고 코드를 직접 실행하는 것도 가능하다. 바이트단위 프로그래밍이 가능하며 빠른 속도의 임의 접근을 제공하므로 시스템에서 주로 코드를 저장하고 실행하는 용도로 사용된다. 하지만 단위 셀의 크기가 커서 NAND 플래시 메모리에 비해 집적도가 낮고 비용이 높아 대용량 저장장치를 필요로 하는 응용에 적합하지 않다.

NAND 플래시 메모리는 NOR 플래시 메모리와 마찬가지로 삭제 연산의 기본단위인 블록으로 나뉘어 진다. 블록들의 크기는 모두 같고 블록들은 읽기/쓰기 연산의 기본단위인 256바이트, 512바이트, 2048바이트의 페이지(Page)로 좀더 세분화된다. 또한 각 페이지는 8바이트/16바이트/64바이트의 저장 영역을 추가로 가지는데 이

잉여 영역(Spare Area)은 많은 응용에서 메타데이터(metadata)나 에러정정코드(error correction code, ECC)를 저장하는데 사용된다.

NAND 플래시 메모리는 NOR 플래시 메모리에 비해 블록단위의 삭제 연산속도가 빠르며 쓰기 연산의 속도도 NOR 플래시 메모리에 비해 빠르다. 하지만 임의 접근이 느리고 읽기/쓰기 동작이 페이지 단위로 이루어진다는 특성을 가진다. 또한 NAND 플래시 메모리는 출하시 약간의 배드 블록을 가지고 있을 수 있으며 삭제연산의 횟수가 증가함에 따라 마모도가 증가하여 배드 블록이 발생할 수도 있다. 따라서 NAND 플래시 메모리를 사용하는 응용에서는 이러한 부분에 대한 처리도 필요하다.

NAND 플래시 메모리는 데이터와 주소를 위한 단일 8비트 혹은 16비트 입출력 선을 사용한다. NAND 플래시 메모리가 I/O의 형태의 인터페이스를 가지고 있기 때문에 시스템의 주소 버스와 데이터 버스에 바로 연결할 수 없고 이러한 구성을 위해서는 CPLD(Complex Programmable Logic Device)를 사용하거나 간단한 회로가 추가해야 한다. 하지만 비교적 간단한 외부 인터페이스를 장점으로 대용량의 비휘발성 저장매체를 필요로 하는 멀티미디어 장비에 최적의 해결책이 되기도 한다.

최근에 주로 사용되는 NAND 플래시 메모리는 페이지 크기가 512바이트 혹은 2048바이트인 것이다. 페이지 크기에 따른 분류보다는 표 1에서처럼 플래시 메모리가 제공하는 명령어 집합에 따라 분류하는 것이 NAND 플래시 메모리의 특성을 잘 나타낼 수 있으며 접근계층의 설계에 유리하다. 설계한 접근계층은 명령어 집합 4와

명령어 집합 5에 해당하는 것은 2048바이트(1024워드)의 페이지 크기를 가지며 64개의 페이지가 하나의 블록을 구성한다.

NAND 플래시 메모리는 데이터와 주소를 위한 단일 입출력 선을 사용하기 때문에 연산 수행이나 제어가 다소 복잡하다. 명령어 집합 4에 속하며 8비트 입출력 선을 가지는 NAND 플래시 메모리를 예로 작동과정을 살펴보자. 페이지 읽기와 쓰기 연산은 I/O 버퍼와 메모리 사이에 존재하는 528바이트의 페이지 레지스터를 이용한다. 쓰기 연산의 경우, 80h 명령어를 입출력 선에 입력한 후 4사이클의 주소를 입력한 후 데이터를 입력한다. 이를 통해 8비트 단위로 레지스터 버퍼가 채워지고 이후에 쓰기 확인 명령어를 입력하면 메모리로 데이터가 입력된다. 읽기 연산의 경우도 비슷한 과정을 거치게 된다. 페이지는 명령어에 의해 3개의 영역으로 나누어지는데 00h 명령어는 전체 페이지의 앞쪽 절반 영역 즉 0~255바이트를 가리키며 01h 명령어는 나머지 절반 영역 즉 256~511바이트를 가리킨다. 50h 명령어는 잉여 공간 즉, 512~527바이트에 해당하는 공간에 대한 연산을 수행한다는 것을 의미한다. 이외에도 블록 삭제 연산, 상태 읽기 연산 등이 있으며 명령어 집합 4에 속하는 NAND 플래시 메모리의 경우 멀티플레인 연산에 대한 명령어도 제공한다. 또한 외부메모리를 사용하지 않고 페이지의 내용을 복사할 수 있는 copy-back 쓰기 명령어를 제공하기도 한다. 이러한 연산과정에서 입출력 선을 제외한 제어선은 GPIO나 CPLD에 의해 제어되며 이러한 제어선에는 CE(Chip Enable), RE(Read enable), WE(Write Enable), WP(Write Protect) 등이 있다.

표 1. 명령어 집합에 따른 NAND 플래시 메모리 분류

명령어 집합 1	4Mbit	[Read, Reset, Frame Program, Block Erase, Status Read, Read ID]
명령어 집합 2	64Mbit - 256Mbit	[Read1, Read2, Read ID, Reset, Page Program, Block Erase, Read Status]
명령어 집합 3	256Mbit - 512Mbit	[Command Set 2] + [Copy-Back Program, Lock, Unlock, Lock-tight, Read Block Lock Status]
명령어 집합 4	512Mbit - 1Gbit	[Command Set 2] + [Page Program (dummy), Copy-Back Program (dummy), Multi-Plane Block Erase, Read Multi-Plane Status]
명령어 집합 5	1Gbit - 8Gbit	[Read, Read for Copy Back, Read ID, Reset, Page Program, Cache Program, Block Erase, Random Data Input, Random Data Output, Read Status]

명령어 집합 5에 해당하는 NAND 플래시 메모리를 지원한다. 명령어 집합 4에 속하는 NAND 플래시 메모리는 512바이트(256워드)의 페이지 크기를 가지며 32개의 페이지로 하나의 블록을 구성하는 반

### 3. NAND 플래시 메모리 접근계층

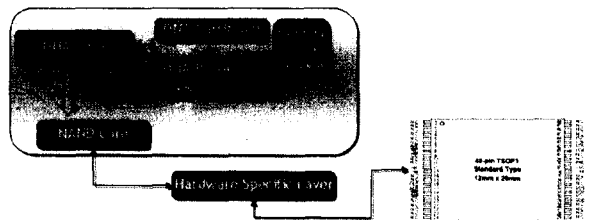


그림 1. NAND 플래시 메모리 접근계층 전체 구성

그림 1은 NAND 플래시 메모리 접근계층의 전체적인 구성을 나타내고 있다. 접근계층은 DNA(Direct NAND Access) Layer로 명명되어 있다. Hardware Specific Layer는 플래시 메모리의 하드웨어 연결 상태에 따라 사용자가 정의할 수 있는 영역이다. 컨트롤러의 GPIO를 통해 NAND 플래시 메모리의 제어선에 신호를 입력하는 경우 또는 CPLD를 사용하는 경우에 이 계층에 해당하는 코드를 작성하여 NAND 플래시 메모리를 제어할 수 있게 된다. DNA Layer는 크게 네 부분으로 구성되어 있다. NAND Core는 Hardware Specific Layer가 제공하는 플래시 메모리 제어기능을 이용하여 플래시 메모리의 기본연산을 수행하게 된다. DNA Core는 DNA Partition을 등록을 담당하고 있으며 등록된 DNA Partition은 DNA Layer를 사용하는 상위계층에 페이지 읽기, 페이지 쓰기, 블록 삭제 등의 기본연산과 잉여 영역에 대한 읽기, 쓰기 및 copy-back 쓰기 연산을 제공한다. 위와 같은 연산은 플래시 변환계층이나 플래시 전용 파일시스템과 같은 상위 계층이 NAND 플래시 메모리를 쉽게 사용할 수 있도록 해준다.

DNA Layer는 명령어 집합 4와 5에 속하는 NAND 플래시 메모리가 제공하는 기능을 수용하고 있다. 앞서 설명한 바와 같이 명령어 집합 4에 속한 플래시 메모리는 멀티플레인으로 구성되어 있어 멀티플레인 연산을 수행할 수 있으며 이는 페이지 크기가 512바이트인 NAND 플래시 메모리가 2048바이트의 논리페이지 파티션을 가질 수 있도록 하며 이는 플래시 변환계층이나 플래시 전용 파일시

시스템의 설계물 2048바이트 크기의 페이지에 최적화할 수 있는 기반을 제공한다. 또한 명령어 집합 4, 5에 속하는 명령어는 copy-back 쓰기 연산을 제공하는데 이는 유효한 데이터를 가지고 있는 블록을 대상으로 가비지 컬렉션을 수행할 때 성능향상을 가져올 수 있으며 DNA Layer는 이러한 연산을 위한 인터페이스를 제공한다.

이미 언급한 바와 같이 NAND 플래시 메모리는 EEPROM의 한 종류로 데이터를 쓰기 위해서는 삭제 연산을 선행해야 하며 또한 쓰기 연산 단위인 페이지보다 큰 블록 단위의 삭제 연산만 허용된다. 이러한 제약으로 인해 갱신되는 데이터는 플래시 메모리의 같은 곳에 쓰이지 않고 다른 위치에 쓰이게 된다. 따라서 이전의 데이터는 무효화되며 계속되는 갱신은 특정 블록이 무효화된 데이터를 가지게 한다. 무효화된 데이터만을 가지고 있는 블록의 경우 가비지 컬렉션시 삭제 연산의 대상이 되고 삭제 연산이 수행된 후 자유 블록으로 변환된다. 하지만 플래시 메모리의 마모도 평준화(wear levelling)와 가비지 컬렉션시 비운론제로 인해 유효한 데이터를 가지고 있는 블록도 삭제 연산의 대상이 되기도 한다. 이 경우 블록내의 유효한 데이터는 플래시 메모리의 할당중인 다른 블록으로 복사되어야 하며 이는 플래시 메모리 저장기법의 성능을 저하할 수 있다. Copy-back 쓰기 연산은 외부 메모리 사용하지 않고 특정 페이지에 있는 데이터를 다른 페이지로 복사할 수 있게 하는 기능으로 플래시 메모리 저장기법이 NAND 플래시 메모리가 제공하는 copy-back 쓰기 연산을 수행할 수 있도록 이에 대한 인터페이스를 제공하는 것은 상위 계층인 플래시 메모리 저장기법의 성능 향상에 도움이 된다.

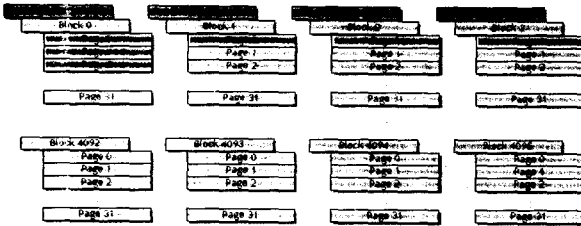


그림 2. NAND 플래시 메모리 멀티플레인 구성

명령어 집합 4에 속하는 플래시 메모리는 메모리 셀의 구성이 멀티플레인으로 되어 있어 다른 플레인에 속하는 여러 페이지에 동시에 쓰기 연산, 블록삭제 연산을 수행할 수 있다. 이러한 멀티플레인 연산을 활용하여 DNA Layer는 512바이트 페이지 크기의 NAND 플래시 메모리도 2048바이트의 논리 페이지 파티션을 생성할 수 있도록 한다. 멀티플레인 형태로 구성된 플래시 메모리는 그 크기에 따라 4개에서 8개의 플레인으로 구성된다. 그림 2에서처럼 각각의 플레인은 1024개의 블록과 528바이트의 페이지 레지스터를 가진다. 이러한 구성으로 인해 각 플레인에 속하는 페이지나 블록에 동시에 쓰기 연산이나 블록 삭제 연산을 수행할 수 있게 된다. 2048바이트 크기의 논리 페이지를 가지는 파티션의 페이지 쓰기 연산이 수행되면 그림 3과 같은 주소변환 과정을 거쳐 멀티플레인 쓰기 연산이 수행된다. 128 메가바이트 크기의 NAND 플래시 메모리는 총 27비트의 주소를 사용하는데 하위 8비트는 컬럼 주소로 나머지는 페이지 주소로 사용한다. 그림 3에서처럼 이러한 주소를 컬럼 주소, 플레인 비트, 페이지 주소, 블록 주소로 세분화하고 플레인 비트와 페이지 주소를 치환하게 되면 2048바이트 페이지 쓰기 연산 주소는 하나의 블록 내에서의 연속된 페이지 주소가 아닌 플레인에 걸쳐 블록별로 하나의 페이지를 가리키는 주소로 변환되고 이 주소를 이용해 멀티플레인 쓰기 연산을 수행하게 된다. 그림 2에서처럼 첫 번째 블록의 네 페이지에 쓰여야 하는 데이터는 첫 번째 블록에서부터 네 번째 블록까지의 첫 번째 페이지에 쓰이게 된다. 이러한 멀티플레인 쓰기 연산은 각 플레인이 가지고 있는 페이지 레지스터를 이용하기 때문에 512바이트 쓰기 연산을 수행하는 것과 비슷한 시간이 걸린

다. 따라서 이를 이용하는 것은 플래시 메모리 저장기법의 쓰기 성능을 크게 향상 시키게 된다. 또한 명령어 집합 4에 해당하는 NAND 플래시 메모리도 2648바이트 크기의 페이지에 기반을 두어 설계된 플래시 변환 계층이나 플래시 전용 파일시스템을 적용할 수 있게 되므로 상위계층의 설계를 간단히 할 수 있게 된다.

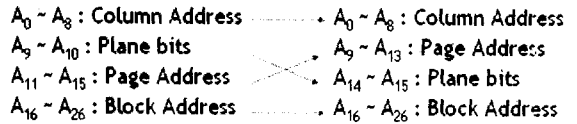


그림 3. 멀티플레인 쓰기 연산의 주소변환

#### 4. 결론

본 논문은 대용량 저장매체를 필요로 하는 기기에서 기본 저장매체로 많이 쓰이고 있는 NAND 플래시 메모리의 특성을 살펴보고, 이를 효율적으로 사용할 수 있도록 하기 위한 접근계층의 설계 및 구현에 대해 설명하였다. 구현된 접근계층은 하드웨어에 종속적이지 않으며 NAND 플래시 메모리의 기본연산을 충실히 제공함은 물론 NAND 플래시 메모리가 제공하는 다양한 기능을 상위 계층에서 충분히 활용할 수 있도록 하고 있다. NAND 플래시 메모리가 다양한 기기의 기본 저장매체로서 사용되고 멀티미디어 데이터에 대한 수요가 늘어남에 따라 향후 과제로서 NAND 플래시 메모리에 특성을 활용하여 시스템 데이터는 물론 멀티미디어 데이터의 저장에도 효율적인 저장기법에 대한 연구가 필요할 것으로 예상된다.

#### 참고문헌

- [1] Intel Corporation, "Understanding the flash translation layer(FTL) specification," <http://developer.intel.com>.
- [2] M. Wu, and W. Zwaenepoel, "eNVy: A Non-Volatile, Main Memory Storage System," In *Proceedings of the 6th International Conference on Architectural Support for Programming Languages and Operating Systems*, pp.86-97, 1994.
- [3] A. Kawaguchi, S. Nishioka, and H. Motoda, "A Flash Memory Based File System," In *Proceedings of the USENIX 1995 Winter Technical Conference*, pp.155-164, 1995.
- [4] Samsung Electronics, "Data sheets of all products of NAND Flash Memories," <http://www.samsungelectronics.com/>.
- [5] D. Woodhouse, "JFFS: The Journaling Flash File System," Ottawa Linux Symposium, 2001.
- [6] J. Kim, J.M. Kim, S.H. Noh, S.L. Min, and Y. Cho, "A Space-Efficient Flash Translation Layer for CompactFlash Systems," *IEEE Transactions on Consumer Electronics*, Vol. 48, No. 2, pp.366-375, 2002.
- [7] Aleph One Company, "Yet Another Flash Filing System," <http://www.aleph1.co.uk/yaffs/>.
- [8] MTD, "Memory Technology Device (MTD) sub-system for Linux," <http://www.inlinux-mtd.infradad.org/>.
- [9] JFFS2, <http://sources.redhat.com/jffs2/>.