

# 다중 섹터 사이즈를 지원하는 낸드 플래시 메모리 저장장치의 FTL 매핑 관리 방법

임승호<sup>o</sup>,  
한국외국어대학교 디지털정보공학과,  
[slim@hufs.ac.kr](mailto:slim@hufs.ac.kr),

최민  
원광대학교 컴퓨터공학과  
[mchoi@wonkwang.ac.kr](mailto:mchoi@wonkwang.ac.kr)

## FTL Mapping Management for Multiple Sector Size based NAND Flash Memory Storage

Seung-Ho Lim<sup>o</sup>, Min Choi  
Department of Digital Information Engineering, Department of Computer Engineering  
HanKuk University of Foreign Studies, Wonkwang University

### 1. 서론

컴퓨터 시스템에서 호스트와 저장장치간의 데이터 최소 전송 단위는 섹터 단위인데, 최근 이러한 섹터 사이즈가 컴퓨터 시스템의 데이터 안정성을 더욱 더 보장하기 위해서 일반 유저 데이터와 데이터 보호를 위한 ECC 데이터를 포함하는 경우가 있으며 이로 인해서 섹터 사이즈가 컴퓨터 시스템에 따라 다르게 설정된다. 이러한 다중 섹터 사이즈의 지원은 아주 데이터의 신뢰성을 보장해 주어야 하는 서버 시스템에서 주로 사용된다.

최근 컴퓨터 시스템의 저장장치 소자로 각광받고 있는 소자가 낸드 플래시 메모리이며 이를 이용해서 나온 저장장치가 SSD(Solid State Disk)이다. 낸드 플래시 메모리는 특성상 플래시 변환 계층(Flash Translation Layer, FTL)이라고 하는 플래시 소프트웨어 계층을 이용하여 관리가 된다. 플래시 변환 계층은 플래시 논리 주소를 물리주소로 변환시켜주는 매핑 관리를 통해서 낸드 플래시 메모리의 물리적 한계점을 극복하고 저장장치로서 사용될 수 있도록 인터페이스를 제공하는 소프트웨어 계층이라고 할 수 있다. 특히, FTL의 설계는 낸드 플래시 메모리 기반의 저장장치의 성능을 결정짓기 때문에 FTL의 매핑 관리는 플래시 소프트웨어의 핵심적인 역할을 한다고 할 수 있다. 낸드 플래시 메모리는 물리적 특성상 데이터를 읽고 쓰는 단위가 페이지라고 하는 단위로 정의되게 되어 있다. 이러한 FTL의 매핑 단위 및 동작 단위는 그림 1과 같이 낸드 플래시 메모리의 페이지를 기본으로 하지만 호스트 시스템의 전송 단위인 섹터와 아주 밀접한 상관관계가 있다. 만약, 섹터 사이즈와 페이지 사이즈의 상관관계가 정수 비례단위가 되지 않은 경우에는 FTL 매핑 관리 및 동작 관리에 많은 오버헤드가 수반된다. 바꾸어 말하면, 다중 섹터 사이즈를 지원하도록 FTL의 매핑 관리 및 동작 관리가 이루어지지 않으면 다중 섹터를 요구하는 컴퓨터 시스템의 성능은 좋을 수가 없다. 본 논문에서는 낸드 플래시 메모리 기반의 저장장치에서 효율적인 다중 섹터 사이즈를 지원하는 FTL 매핑 관리 기법을 제안하고, 그 관리 방법과 성능에 관하여 분석하여 본다.

### 2. 본론

본 논문에서 제안하는 방식은 그림 2에 나와 있는 방식으로 FTL을 관리 한다. 즉, 본 논문에서 제안하는 방식은 페이지 그룹(Page Group, PG) 기반의 FTL 매핑 관리 방법이다. 그림 2에서 보는 바와 같이 페이지 그룹 기반의 FTL 매핑 관리 방법은 먼저, 페이지 사이즈에 맞아 떨어지지 않아서 나누어지는 분리 섹터의 수를 최소화하기 위해서 페이지를 그룹(PG)으로 묶는다. 그 후, PG의 첫 페이지부터 섹터를 할당해 나간다. 만약 하나의 섹터가 현재 페이지에 완전히 저장되지 않는 경우에는 그 나머지 부분을 다음 페이지에 연속적으로 저장하며 이 때 분리 섹터가 발생하게 된다. 이런 방식으로 계속 섹터를 할당하면, 한 페이지에 저장되는 완전한 섹터의 개수(N)은 (1)과 같이 나타낼 수 있다. 수식에서

P는 페이지 사이즈며 S는 섹터 사이즈이다

$$N = P/S \quad (1)$$

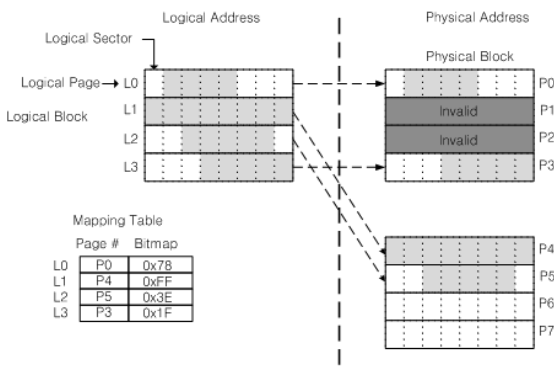


그림 2 페이지 단위의 매핑 관리방식의 FTL

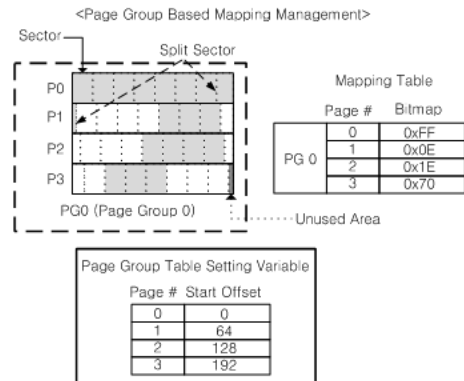


그림 1 제안된 Page Group 기반의 FTL

즉, PG내의 마지막 페이지를 제외한 모든 페이지는 한 페이지 내에서 분리되지 않고 저장되는 완전한 섹터의 개수가 N개가 유지되는 페이지의 수로 그룹은 생성된다 한 PG의 모든 페이지는 분리 섹터를 가지게 되는데, 각 페이지의 분리 섹터의 오프셋(OP<sub>n</sub>)은 모든 PG에서 동일한 값을 가지며 (2)와 같이 계산되며, PG내의 페이지 수(PN)은 (3)과 같이 계산된다.

$$OP_n = \begin{cases} 0, & (n = 0) \\ OP_{n-1} + ((N+1)S - P), & (0 < n < PN) \end{cases} \quad (2)$$

$$PN = OP_n + NS > P \text{인 } n \quad (3)$$

이와 같은 구성에 의하면 페이지 그룹 내에서 맨 마지막 페이지를 제외한 페이지들은 매 페이지당 섹터의 개수는 완전한 섹터(N)과 분리 섹터 하나, 즉, (N+1)의 섹터를 가지게 된다 마지막 페이지는 N개의 완전한 섹터로만 구성되며 그 이후의 영역은 사용하지 않는 영역이다 또한, 페이지 그룹 내의 모든 페이지에 대해서 매 페이지의 시작 섹터 오프셋은 수식(2)와 같이 계산하여 Page Group Table을 생성해 낼 수 있다. 즉, FTL 매핑 테이블을 구성할 때 낸드 플래시 메모리의 페이지 사이즈가 정해지고 호스트 시스템에서 사용할 섹터 사이즈가 정해지면위의 수식에 의해서 페이지 그룹(PG)을 생성할 수 있으며, 페이지 그룹 내에서 각 페이지의 섹터 비트맵을 구성할 수 있다그림 2에서 예를 들은 바와 같이, PG0는 네 개의 페이지로 구성되며 각 페이지 당 섹터 개수는 P0, P1, P2는 8개이며, P3는 7개의 섹터로 구성된다 그리고 각 페이지 당 자신에게 포함된 섹터의 시작 섹터 오프셋(OP<sub>n</sub>)은 Page Group Table과 같이 구성된다.

이렇게 구성된 페이지 그룹을 이용하여 보다 효율적인 낸드 플래시 메모리 연산을 수행할 수가 있게 된다. 즉, 한 페이지 그룹 내에서 페이지 당 섹터 비트맵을 일정하게 구성할 수 있으며, 각 섹터당 시작 섹터 오프셋은 Page Group Table을 통해서 바로 참조할 수 있기 때문에 해당 섹터의 위치를 찾는 오버헤드가 들지 않게 되어 기존의 섹터 비트맵 연산과 유사한 연산 오버헤드를 수반한다기와 동시에, 사용하지 않는 영역이 PG내의 마지막 페이지 일부분이기 때문에 전체 낸드 플래시 메모리 영역에 비해서는 극히 일부분이 될 수 있으므로 용량에 대한 오버헤드가 적다

### 3. 결론

본 논문은 다중 섹터 사이즈를 효율적으로 지원할 수 있는 페이지 그룹 기반의 FTL 매핑 관리 방법을 제안함으로써 FTL의 매핑 관리 및 동작 방식에 오버헤드를 최소화함과 동시에 다중 섹터를 지원해 줌으로써 보다 안정성 높은 낸드 플래시 메모리 기반의 저장장치의 구현이 가능하다