

낸드 플래시 메모리 기반 저장 장치를 위한 낸드 플래시 메모리의 사양 최적화

*김동건, 원삼규, 박상훈, 서혁준, 정의영
연세대학교 전기전자공학부

e-mail: *digikim@dtl.yonsei.ac.kr, imref@yonsei.ac.kr,*
soskhong@dtl.yonsei.ac.kr, jjsky7@dtl.yonsei.ac.kr, eychung@yonsei.ac.kr

Optimization of NAND Flash Memory Specification for NAND Flash Memory Based Storage Devices

*Dong-gun Kim, Sam Kyu Won, Sang-Hoon Park,
Hyukjun Seo and Eui-Young Chung
School of Electrical and Electronic Engineering
Yonsei University

Abstract

A NAND flash memory based storage devices like Solid State Drive(SSD) is quickly replacing position of conventional storage devices like Hard Disk Drive(HDD). As time goes by, the specifications of NAND flash memory like page size and block size are growing larger. In this paper, we will investigate effect of specification changes and propose method to determine optimized NAND flash memory specifications for various workloads.

I. 서론

SSD와 같은 낸드 플래시 메모리를 이용한 저장 장치가 빠르게 HDD를 대체하고 있다. 낸드 플래시 메모리는 기계적 구조의 배제로 접근 시간 면에서 HDD보다 훨씬 유리하며 무게가 가볍고 충격에 강하다. 또한 비휘발적 특성으로 전원이 없어도 저장된 데이터를 유지할 수 있다. 앞으로 낸드 플래시 메모리를 기반으로 하는 저장 장치가 HDD의 뒤를 이을 것으로 예상된다.

하지만 아직까지는 SSD가 뛰어난 성능에도 불구하고 HDD를 완전히 대체하지 못하게 하는 단점이 존재한다. 높은 단위 용량 당 가격이 대표적이며 다른 단점으로는 덮어쓰기가 불가능하다는 점이다. 각각의 cell은 한 번 기록되면 지우는 과정을 거쳐야 다시 기록이 가능하다.

이 특징은 읽고 쓰는 이외에 추가적인 작업을 발생시키기 때문에 성능적인 측면에서 악재로 작용한다. 게다가 읽기, 쓰기에 비해 지우기는 10배에 가까운 시간이 소요된다. 한 개의 셀을 읽고 쓰고 지우는 시간이 길기 때문에 낸드 플래시 메모리에서는 셀의 집합인 페이지와 페이지의 집합인 블록을 작업 단위로 사용한다. 또한 즉시 덮어쓰기가 불가능하기 때문에 OS의 논리적 주소와 낸드 플래시 메모리의 물리적 주소 사이에 매핑이 필요하다. Flash Translation Layer(FTL)가 이 역할을 담당하며 BAST^[1], 페이지 매핑^[2] 등이 있다.

낸드 플래시 메모리는 셀 당 1비트를 저장할 수 있는 Single Level Cell(SLC)과 2비트를 저장할 수 있는 Multi Level Cell(MLC)이 있으며 현재는 셀 당 3비트까지 저장할 수 있는 TLC도 상용화되고 있다. 단위 셀 당 저장용량의 차이로 인해 MLC가 훨씬 저렴하지만 표.1과 같이 작업 속도와 소비 전력에서 불리한 면이 있다.

표.1 SLC와 MLC의 사양 비교

	SLC	MLC
tPROG (μ s)	200	1200
tR (μ s)	25	60
tWC (μ s)	0.03	0.06
tRC (μ s)	0.03	0.06
tBERS (μ s)	2000	3000
Icc1 (mA)	15	30
Icc2 (mA)	15	30
Icc3 (mA)	15	30

낸드 플래시 메모리는 읽기, 쓰기 작업의 단위로 페이지, 지우기의 단위로 블록을 사용한다. 페이지의 크기는 보통 1~8KB로 이루어지며 한 개의 블록은 16~256 페이지로 구성된다. 최근의 추세를 보았을 때 그림.1와 같이 페이지와 블록의 크기는 계속 커지고 있다.[15] 하지만 이 추세가 성능과 비용, 소모 전력 면에서 모두 이득을 가져올 지에 대해서는 아직 연구된 바가 없다.

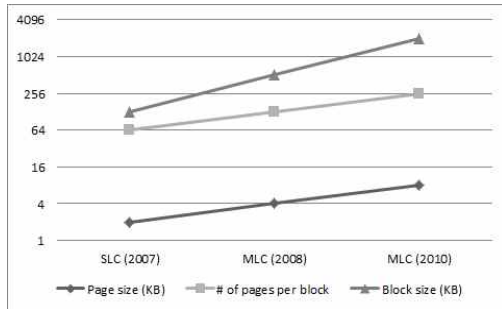


그림.1 낸드 플래시 메모리 스펙의 변화

덮어쓰기가 일어날 때 기존 페이지는 무효화되고 새로운 페이지만이 유효하게 된다. 이런 경우 무효화된 페이지는 지워질 때까지 쓰레기 값을 가진 채 용량을 차지하게 된다. 여기서 물리적 사용 가능 용량과 논리적 사용 가능 용량에 차이가 발생하는데 이를 정리하는 과정을 가비지 컬렉션이라 하며 기본적인 읽기, 쓰기 작업 이외에 페이지 복사, 블록 삭제와 같은 추가적인 작업 수행을 필요로 한다. 페이지와 블록의 크기에 따라 가비지 컬렉션의 빈도, 수행 시간과 효율 모두 변하게 되며 이는 성능의 상승, 하락에 영향을 미친다.

최근 수요가 급증하고 있는 모바일, 휴대용 제품들은 사용할 수 있는 자원에 대한 제약이 크다. SRAM, DRAM과 같은 자원부터 면적과 소모 전력에 이르기까지 많은 제한 속에서 제품을 설계해야 한다. 따라서 이 논문에서는 낸드 플래시 메모리 스펙의 변화에 따른 시스템의 성능 변화를 분석하여 사용자가 원하는 응용 프로그램 또는 OS에 따라 적당한 낸드 플래시 메모리 스펙을 선택할 수 있는 방법을 제안할 것이다. 성능과 비용, 전력 사이에서 적절한 트레이드오프를 통해 출시되어 있는 제품들 중 가장 최적에 가까운 낸드 플래시 메모리 제품을 선택하거나 해당 사양의 제품을 새롭게 생산할 수 있을 것이다.

II. 본론

2.1 전송률 측정

최근의 낸드 플래시 메모리 기반 저장 장치는 채널, 웨이를 사용함으로써 그림.2의 (a), (b)와 같은 주소 변환[13]을 통한 인터리빙을 이용한다. 동시에 여러 개의 낸드

드 플래시 메모리에 순차적인 내용을 기록한다면 채널, 웨이를 늘려 그에 비례하는 성능 향상을 이룰 수 있다. 하지만 그에 따른 추가적인 하드웨어 또한 필요하며 이것은 직접적인 비용 증가로 이어져 성능과 비용 사이에서 적절한 트레이드오프를 요구하게 된다.

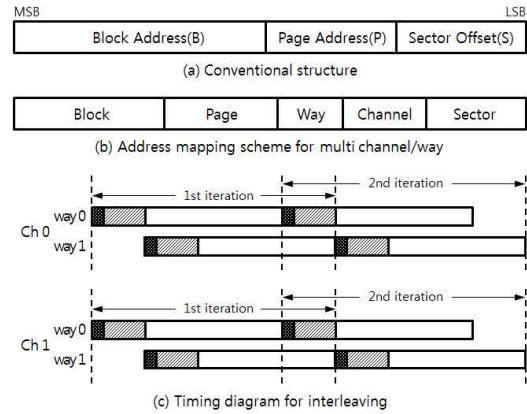


그림.2 주소변환을 이용한 인터리빙

그리고 SSD에서 인터리빙을 사용한다는 것은 동시에 여러 개의 낸드 플래시 메모리에서 작업이 실행되고 있다는 것을 의미하므로 전송률 계산에 있어 인터리빙에 대한 고려가 필요하다. 정해진 용량을 전송하는데 걸리는 시간인 t_{OP} 와 데이터의 크기 S_{data} 를 이용해 전송률 T 를 구할 수 있는데 t_{OP} 는 다음과 같이 나타낸다.[14]

$$t_{OP, write} = \lceil \alpha \rceil t_{CMD} + \alpha \cdot t_{TRANS} + t_{PROG} + K(\lceil \alpha \rceil - 1)$$

$$t_{OP, read} = \lceil \alpha \rceil t_{CMD} + \lceil \alpha \rceil \cdot t_{TRANS} + \lceil \alpha/W \rceil (t_R - (W-1)t_{CMD})$$

$$T = S_{data}/t_{OP}$$

where $K = \max(t_{PROG}(W-1)(t_{CMD} + t_{TRANS}), 0)$

이 식에서 t_{CMD} 와 t_{TRANS} 는 낸드 컨트롤러에 명령을 전달하는 시간과 데이터를 전송하는 시간이며 α , W 는 요청으로 전달된 데이터의 페이지 수, 장치의 웨이 수이다. 그리고 K 는 그림.2 (c)와 같이 인터리빙이 반복될 때 이후의 반복이 얼마나 빨리 시작할 수 있는가이다.

2.2 비용(면적) 측정

낸드 플래시 메모리의 내부 구조는 그림.3과 같다. 커맨드는 커맨드 인터페이스 로직으로 전달되고 주소는 디코더로 전달되어 낸드 플래시 셀에 접근하게 된다. 낸드 플래시 메모리는 랜덤 접근이 불가능하기 때문에 셀의 집합인 페이지 단위로 페이지 버퍼에 저장된 후에 접근한다. 따라서 페이지 크기가 달라지면 페이지 버퍼의 크기도 같이 변하게 되고 여기서 페이지 오버헤드 (O_p)가 변화한다. 지우는 작업의 경우에는 블록 단위로 수행되기 때문에 블록의 크기와 관련이 있으며 블록 오

버헤드(O_b)가 변하게 된다. 셀 이외의 저장 공간과 추가적인 면적, I_{cc} 역시 변화가 생긴다.

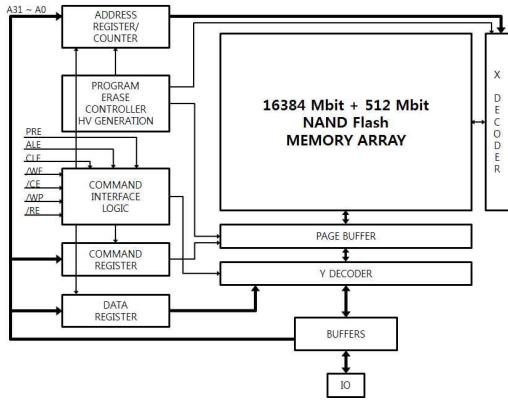


그림.3 낸드 플래시 메모리 내부 구조[5]

낸드 플래시 메모리의 비용은 면적과 비례한다. 페이지와 블록의 크기에 따라 변하는 면적을 나타낸 식을 bit cost equation이라고 하자. Bit cost equation을 유도하기 위해서는 먼저 블록 당 비용을 C_b 로 정의하고 다음과 같이 계산한다.

$$C_b = S_p(N_p/T + O_b) + O_p/N_b$$

S_p 는 페이지의 크기(KB)이며 N_p 는 블록 당 페이지 수이며 N_b 는 웨이 당 블록 수를 나타낸다. T 는 셀 타입으로 SLC일 경우는 1, MLC는 2, TLC는 3으로 정의한다. O_p 는 페이지 크기가 512B, 1KB, 2KB, 4KB, 8KB와 같이 증가함에 따라 같이 증가하며 1~9의 값을 가진다. 또한 O_b 는 블록이 커짐에 따라, 즉 블록 당 페이지의 개수가 32, 64, 128, 256과 같이 증가함에 따라 2~5의 값을 가진다.

블록 당 비용을 계산하였으면 전체 저장 장치의 비용을 계산해야 한다. 최근 거의 모든 저장 장치들이 멀티 채널, 웨이를 사용하고 있으며 이를 위해서는 추가적인 비용이 발생하게 된다. 한 개의 채널, 웨이로 인해 발생하는 비용을 각각 C_c , C_w 라고 하고 저장 장치의 채널 수와 채널 당 웨이 수는 각각 N_c , N_w 라고 하자. 또한 버퍼, 캐시 등의 용도로 SRAM이나 DRAM 등을 낸드 플래시 메모리 외부에 추가한다고 하면 비용 C_r 이 추가로 발생하며 모든 비용 C 는 다음과 같이 나타낸다.

$$C = C_b \cdot N_b + N_c \cdot C_c \cdot N_w \cdot C_w + C_r$$

2.3 전력 측정

낸드 플래시 메모리 기반 저장 장치는 HDD보다 훨씬 적은 전력을 소모한다. 하지만 모바일, 휴대용 제품에서는 최대한 적은 전력을 소모하는 것이 유리하다. 따라서 저장 장치의 전력 소모량 또한 중요한 사안이다. 낸드 플래시 메모리의 기본 작업에 소모되는 전력량은 다음과 같이 구할 수 있다.

$$P_{read} = V_{CC}((I_{CC3} - I_{CC4}) \cdot t_R + I_{transfer} \cdot t_{RC})$$

$$P_{program} = V_{CC}((I_{CC2} - I_{CC4}) \cdot t_{PROG} + I_{transfer} \cdot t_{WC})$$

$$P_{erase} = V_{CC}((I_{CC3} - I_{CC4}) \cdot t_{BETS})$$

$I_{transfer}$ 는 낸드 플래시 메모리에 쓸 데이터 혹은 읽은 데이터를 전송하는 데 필요한 전류로 일반적으로 50mA이며 t_{RC} , t_{WC} 는 그에 필요한 시간이다. I_{CC4} 는 standby 전류이며 보통 1mA이다. 다음으로 모든 요청을 처리하면서 발생한 낸드 플래시 커맨드의 개수를 N_{read} , $N_{program}$, N_{erase} 라 하면 소모한 전력량 P_{req} 는 다음과 같다.

$$P_{req} = (N_{read} \cdot P_{read} + N_{program} \cdot P_{program} + N_{erase} \cdot P_{erase})$$

이 외에 저장 장치가 유휴 상태일 때도 전력을 소모하게 되는데 유휴 상태의 소비 전류는 I_{CC5} 라 하고 0.05mA 정도이다. m 번의 유휴 상태 중 k 번째의 시간이 $t_{IDLE,k}$ 일 때 소비 전력량을 유도할 수 있다.

$$P_{idle} = (V_{CC} \cdot I_{CC5}) \cdot \sum_{k=1}^m t_{IDLE,k}$$

이제 n 개의 요청이 처리될 때 전체 소비 전력을 구할 수 있다. 처리 사이에 $t_{idle,1} \sim t_{idle,n}$ 의 길이를 가지는 유휴 상태를 거친다고 하면 소비 전력량 P 는 다음과 같다.

$$P = P_{req} + P_{idle}$$

2.4 최적화

낸드 플래시 메모리의 최적화는 앞에서 2.1~2.3에서 구한 전송률 T , 비용 C , 전력 P 를 통해 이루어진다. T 는 높을수록, C 와 P 는 낮을수록 좋기 때문에 최적화 상수 O 를 다음과 같이 정의한다. 성능을 살리면서도 최적화 상수 O 를 최대로 하는 낸드 플래시 메모리 스펙을 선택하면 된다.

$$O_{T,PC} = T/(PC)$$

$$O_{T,C} = T/C$$

III. 구현 및 실험 결과

3.1 구현 및 실험 환경

최적화 결과를 검증하기 위해 C++로 트레이스를 이용, SSD의 동작을 시뮬레이션 하는 시뮬레이터를 개발하였다. 시뮬레이터 내부에서 낸드 플래시 메모리 스펙과 FTL의 변경이 가능하며 DiskMon_[7]으로 추출한 트레이스를 요청으로 입력받아 낸드 커맨드와 작업 시간, 전송률, 비용과 전력을 결과로 내놓게 된다. 이 시뮬레이션에서 사용한 트레이스의 정보는 표.2와 같다.

표.2 실험에서 사용된 트레이스

Characteristics	R/W ratio (%)	Avg. Length (Sectors)	# of request
(a) CrystalMark _[8] Sequential	5.91 /94.09	0.74 /1821.64	10000
(b) CrystalMark Random	0.05 /99.95	0.01 /8.27	33740
(c) General usage (Windows 7)	52.34 /47.66	35.5 /15.08	633714

3.2 실험 결과 및 분석

위에서 설정한 환경과 트레이스를 이용하여 실험을 진행하였다. FTL은 BAST와 페이지 매핑을 사용하였으며 서론에서 보였던 3가지의 낸드 플래시 메모리의 파라미터를 이용하여 비교하였다. 실험을 통해 T와 P, C를 구하여 최적화 상수 O를 구한 결과는 아래와 같다.

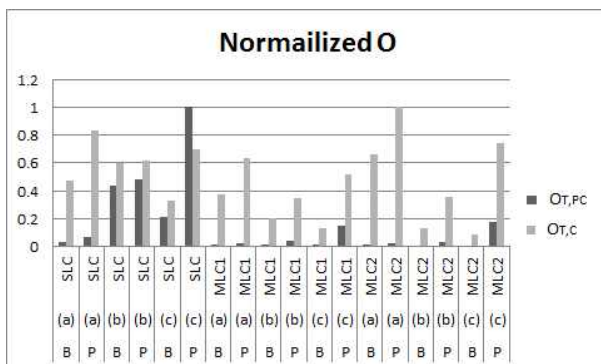


그림.4 노멀라이즈된 최적화 상수 O

B: BAST, P: 페이지 매핑

SLC: 2KB×64, MLC1: 4KB×128, MLC2: 8KB×256 (pages)

그림.4의 그래프는 싱글 채널, 웨이에서의 결과이다. 멀티채널, 웨이를 사용할 경우는 C_c 와 C_w 의 값을 C에 반영하여 최적화 상수를 구하면 된다. C_r 역시 정확한 메모리들의 가격을 반영하여 적용할 수 있다. 먼저 $O_{T,PC}$ 에서 알 수 있는 것은 추가 비용을 제외한다면 페이지 매핑이 확실히 모든 면에서 우수하며 트레이스마다 유리한 낸드 플래시 메모리의 스펙이 다르다는 것이다. 페이지 크기가 커지면 I_{cc} 의 크기가 비례해서 커지기 때문에 전력 면에서 불리함과 동일한 시간에 많은

데이터를 쓸 수 있다는 장점을 동시에 가진다. MLC는 SLC에 비하여 성능과 전력에서 불리하며 비용에서는 유리하다. 한편 소비 전력은 성능과 반대의 결과를 보이는데 성능이 좋다는 것은 같은 데이터를 쓰는 과정에서 부가적인 작업의 발생이 적다는 것이기 때문에 성능을 향상시키는 것은 소비 전력량 면에서도 유리하다. 최적화 과정에서는 용도에 따라 $O_{T,PC}$ 와 $O_{T,C}$ 를 이용할 수 있다. 순차적인 트레이스(a)에서는 MLC2가, 랜덤한 트레이스(b)에서는 SLC가 유리하다. 트레이스 (c)와 같은 실제 사용에서 $O_{T,PC}$ 와 $O_{T,C}$ 를 모두 고려한다면 페이지의 크기가 큰 MLC2에서 페이지 매핑을 사용하는 것이 비교적 좋아 보이지만 성능에서 중점을 둔다면 SLC가 더 좋을 수 있으며 비용 절감이 더 중요한 비중이라면 결과는 다를 수 있다.

IV. 결론 및 향후 연구 방향

실험 결과에서 살펴 볼 수 있듯이 최적인 낸드 플래시 메모리의 스펙은 용도에 따라 다르다. 본 논문에서 제안한 방식이 최적화에 도움이 될 수 있을 것이다. 향후 보다 다양한 조건에서 여기서 고려되지 않은 매개변수들의 영향을 모델링하고 최적화 요소에 가중치를 추가하여 최적화된 스펙을 찾을 수 있도록 하겠다.

Acknowledgement: 이 논문은 하이닉스 및 2010년 정부(교육과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임. (No. 2010-0025423)

참고문헌

- [1] Jesung Kim, Jong Min Kim, Sam H. Noh, Sang Lyul Min and Yookun Cho, A Space-Efficient Flash Translation Layer for Compactflash Systems, IEEE TCE, Vol. 48, No. 2, pp. 366-375, May 2002
- [2] Atsuo Kawaguchi, Shingo Nishioka and Hiroshi Motoda, A Flash-Memory Based File System, USENIX Technical Conference, 1995
- [3] Sang-Hoon Park, Seung-Hwan Ha, Kwanhu Bang and Eui-Young Chung, Design and Analysis of Flash Translation Layers for Multi-Channel NAND Flash-Based Storage Devices, IEEE TCE, Vol. 55, No. 3, pp. 1392-1400, August 2009
- [4] Sam Kyu Won, Seung-Hwan Ha and Eui-Young Chung, Fast Performance Analysis of NAND Flash-Based Storage Devices, IET, Vol. 45, No. 24, pp. 1219-1221, November 2009
- [5] NAND Flash Datasheet, <http://www.hynix.com/>
- [6] DRAMSim2, <http://www.ece.umd.edu/dramsim/>
- [7] DiskMon, <http://technet.microsoft.com/>
- [8] CrystalMark, <http://crystalmark.info/>