

NAND 플래시 메모리에서 쓰기/지우기 연산을 줄이기위한 버퍼 관리 시스템

정 보 성*, 이 정 훈**

The buffer Management system for reducing write/erase operations in NAND flash memory

Bo-Sung Jung*, Jung-Hoon Lee**

요 약

NAND 플래시 메모리는 저전력, 저렴한 가격, 그리고 대용량임에도 불구하고 페이지 단위의 쓰기 및 블록 단위의 지우기 연산은 큰 문제점을 가지고 있다. 특히 NAND 플래시 메모리 특성상 덮어쓰기가 불가능하므로 쓰기동작 후 수반되는 지우기 동작은 전체 성능저하의 원인이 된다. 기존의 NAND 플래시 메모리를 위한 SRAM 버퍼는 간단하면서도 NAND 플래시 메모리의 쓰기 동작을 효과적으로 줄여줄 수 있을 뿐 아니라 빠른 접근 시간을 보장 할 수 있다. 본 논문에서는 작은 용량의 SRAM을 이용하여 NAND 플래시 메모리의 가장 큰 오버헤드인 지우기/쓰기 동작을 효과적으로 줄일 수 있는 버퍼 관리 시스템을 제안한다. 제안된 버퍼는 큰 페칭 크기를 가지는 공간적 버퍼와 작은 페칭 크기를 가지는 시간적 버퍼인 완전연관 버퍼로 구성된다. 시간적 버퍼는 공간적 버퍼에서 참조된 작은 페칭을 가지며, NAND 플래시 메모리에서 쓰기 및 지우기 수행시 시간적 버퍼내에 존재하는 같은 페이지 혹은 블록에 포함된 페칭 블록을 찾아 동시에 처리한다. 따라서 NAND 플래시 메모리에서 쓰기 및 지우기 동작을 획기적으로 줄였다. 시뮬레이션 결과에 따르면 제안된 NAND 플래시 메모리 버퍼 시스템은 2배 크기의 완전연관 버퍼에 비해 접근 실패율 관점에서는 높았지만, 쓰기 동작과 지우기 동작은 평균적으로 각각 58%, 83% 정도를 줄였으며, 결론적으로 평균 플래시 메모리 접근 시간은 약 84%의 성능 향상을 이루었다.

▶ Keyword : NAND 플래시 메모리, 버퍼 시스템, 시간적/공간적 지역성, 쓰기/지우기 연산

Abstract

There are the large overhead of block erase and page write operations in NAND flash memory, though it has low power consumption, cheap prices and a large storage. Due to the physical characteristics of NAND flash memory, overwrite operations are not permitted at the same

• 제1저자 : 정보성 교신저자 : 이정훈

• 투고일 : 2011. 04. 19, 심사일 : 2011. 05. 10, 게재확정일 : 2011. 07. 25

* 국립경상대학교 제어계측공학과 박사과정 (Dept. of C.I.E, GyeongSang National University)

** 국립경상대학교 제어계측공학과(공학연구원) 부교수(Dept. of C.I.E, GyeongSang National University)

※ 이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2009-0065096)

location, so rewriting operation require after erase operation. it cause performance decrease of NAND flash memory. Using SRAM buffer in traditional NAND flash memory, it can not only reduce effective write operation but also guarantee fast memory access time. In this paper, we proposed the small SRAM buffer management system for reducing overhead of NAND flash memory, that is, erase and write operations. The proposed buffer system in a NAND flash memory consists of two parts, i.e., a fully associative temporal buffer with the small fetching block size and a fully associative spatial buffer with the large fetching block size. The temporal buffer have small fetching blocks that referenced from spatial buffer. When it happen write operations or erase operations in NAND flash memory, the related fetching blocks in temporal buffer include a page or a block are written in NAND flash memory at the same time. The writing and erasing counts in NAND flash memory can be reduced. According to the simulation results, although we have high miss ratios, write and erase operations can be reduced approximately 58% and 83% respectively. Also the average memory access times are improved about 84% compared with the fully associative buffer with two sizes.

▶ Keyword : NAND flash memory, buffer memory, temporal-spatial locality, write/eraser operation

1. 서 론

오늘날, 낮은 가격에 높은 집적도를 얻을 수 있고 충격에 강한 특성을 가진 NAND 플래시 메모리는 스마트폰, 스마트패드 등 다양한 개인용 기기에 널리 쓰이고 있으며, 이와 같은 소형기기는 물론 노트북 및 데스크톱 컴퓨터의 최하위 저장장치로 사용하기 위한 많은 연구가 이루어지고 있다 [1,2]. 특히 고성능 대용량을 요구하는 오늘날의 모바일 기기에서는 고집적도 및 저전력의 NAND 플래시 메모리의 내장은 필수적이며 그 수요가 급증하고 있다.

하지만 하드웨어의 급속한 발전으로 프로세서와 하드디스크와의 데이터 처리속도 차이가 더욱 커지고 있다. 비록 NAND 플래시 메모리가 하드디스크에 비해 접근 시간(Seek time)이 빠르기 때문에 읽기/쓰기의 데이터 처리속도가 우수하지만 NOR 플래시 메모리와 달리 byte단위의 접근이 불가능하다[3]. 따라서 페이지 단위의 접근(쓰기/읽기) 그리고 이미 기록된 페이지에 덮어 쓰기가 불가능하기 때문에 블록의 쓰기 동작 후 지우기 동작이 수반되어야 하는 큰 문제점을 가지고 있다. 더욱이 NAND 플래시 메모리는 읽기 동작에 비해 쓰기 동작과 지우기 동작은 큰 오버헤드를 가질 뿐 아니라 동일한 주소의 쓰기 동작은 횟수가 제한된다. 특히 제한된 횟수 이상 쓰기 동작이 반복 될 경우 해당 블록을 사용 못하게 되는 구조적 특징을 가지고 있다.

이러한 NAND 플래시 메모리의 단점을 보완하고 성능향

상을 위해 많은 연구가 이루어지고 있다. 플래시 셀에서의 빠른 접근과 효과적인 쓰기 동작을 위한 FTL(Flash Translation Layer) 연구[4]와 지우기 동작의 블록 Wear leveling으로써 플래시 메모리 셀의 페이지 쓰기와 블록 지우기의 성능 향상을 위한 연구[5]이다. 하지만 FTL과 Wear leveling의 연구는 플래시 메모리 페이지와 블록에 대한 쓰기/지우기 연산에서 효과적인 성능 개선을 보이지만 여전히 느린 랜덤 접근 시간을 가지는 문제점이 있다. 그리고 기존의 NAND 플래시 메모리에 SRAM 버퍼를 이용하여 NOR 플래시 메모리의 빠른 접근 시간 및 쓰기 연산을 줄여 NAND 플래시 메모리의 성능향상을 이루는 방법이 제시되었다[6,7]. 이는 기존 NAND 플래시 메모리에 간단한 메커니즘을 활용한 버퍼의 사용으로 플래시 메모리 셀에 직접 읽기 및 쓰기 수행 빈도를 60% 이상 줄여줌으로써 많은 성능 향상을 보여 주고 있다[7].

하지만 하드 디스크와 NAND 플래시 메모리는 기본적으로 접근 단위에서 다른 특성을 가진다. 바이트 단위의 하드 디스크와 달리 페이지 단위의 NAND 플래시 메모리에 SRAM 버퍼를 사용할 경우 일반적으로 한 페이지의 크기에서 참조된 작은 페칭 크기를 저장하므로 작은 용량의 버퍼로부터 동일한 페이지에 대한 많은 횟수의 쓰기 동작을 요구 할 수 있다는 문제점을 초래한다. 또한 SRAM 버퍼의 경우 프로그램의 특성에 따라 버퍼의 구조, 참조 정책 그리고 버퍼의 크기 및 페칭 크기에 따라 성능의 차이를 가진다.

본 논문에서는 NAND 플래시 메모리의 특성에 적합한 작

은 용량의 SRAM 버퍼를 이용하여 빠른 접근 시간과 쓰기/지우기 동작을 줄여 전체 시스템의 성능향상을 위한 버퍼 구조와 운용방법을 제안한다. 제안된 버퍼는 큰 페칭 크기를 가지는 공간적 버퍼와 작은 페칭 크기를 가지는 시간적 버퍼로 모두 완전연관 버퍼로 구성된다. 또한 제안된 버퍼의 쓰기 동작을 줄이기 위하여 버퍼로부터 NAND 플래시 메모리의 쓰기 동작 요청시 버퍼내의 요청 페이지에 해당하는 변경된 데이터를 일괄적으로 찾아서 한 번에 한 페이지를 쓰게 된다. 제안된 버퍼 관리 시스템은 2배 크기의 비교 버퍼들, 예로 8-웨이 집합연관버퍼(8-way set associative buffer), 희생버퍼(Victim buffer), 완전연관버퍼(Fully associative buffer)에 비해 높은 버퍼 접근 실패율을 보였지만, 쓰기 동작과 지우기 동작의 접근을 획기적으로 줄여줌으로써 전체적으로 평균 플래시 메모리 접근 시간에서 높은 성능 향상을 보였다.

이 논문의 구성은 다음과 같다. II장에서는 NAND 플래시 메모리 버퍼 시스템의 관련연구에 대해 설명한다. III장에서는 제안된 플래시 메모리의 구조와 제안된 알고리즘 및 동작에 대해 설명한다. IV장에서는 시뮬레이션 결과 및 평가 그리고 V장에서 결론을 맺는다.

II. 관련 연구

기존의 캐쉬 메모리는 빠른 접근 속도와 메모리 접근 실패율을 줄이기 위해 다양한 구조와 동작 알고리즘들이 연구되어 왔다. 이러한 기존의 캐쉬 정책을 NAND 플래시 메모리의 버퍼로 사용한 연구 또한 현재 진행되고 있는 추세이다.

Park[8]은 NOR 플래시에서 담당하던 명령어 부분을 SRAM과 NAND 플래시 메모리를 이용하여 실시간 응용 프로그램을 원활히 수행하기 위한 연구로써 NOR 플래시 메모리를 제거시킨 저비용의 플래시 메모리 패키지를 구현하였다. 희생(Victim)버퍼 구조를 가지며, NAND 타입 플래시 메모리의 페이지 여유 공간(spare area)의 참조 패턴을 이용하여 희생 버퍼 혹은 SDRAM에 선택적으로 저장하게 된다.

Jo는[9] 휴대용 미디어 장치를 위한 FAB(Flash Aware Buffer)정책을 제안하였다. FAB는 DRAM 버퍼에서 요청 페이지 수용 공간이 없을 때, NAND 플래시 메모리의 블록들 중 가장 많은 페이지를 가지는 버퍼의 데이터를 희생 데이터로 선택한다. 이러한 희생 블록의 선택은 버퍼의 많은 여유 공간을 확보할 뿐 아니라 동일한 NAND 플래시 메모리의 블록에 쓰기 접근/ 지우기 동작을 줄이는 효과를 가져다준다. 하지만 위의 연구들은 버퍼의 페칭 크기가 NAND 플래시 메모리의 페이지 단위와 동일한 페칭 크기를 가진다. 큰 페칭

크기를 가지는 버퍼는 공간적 지역성에 효과적이지만, 시간적 지역성이 높은 데이터 혹은 참조 패턴 주기가 큰 데이터의 경우 페칭 크기가 큰 버퍼는 큰 용량을 사용해야만 한다. 큰 용량의 버퍼를 사용하므로 전체 시스템의 성능향상에 도움이 되지만 고비용과 높은 소비전력을 요구하는 단점을 가지고 있다.

Lee[7]는 프로그램 수행시 적합한 공간적 지역성과 시간적 지역성을 이용하여 작은 용량의 버퍼를 이용하여 NAND 플래시 메모리의 성능향상을 이루었다. 이 플래시 메모리 패키지는 시간적 지역성을 위한 희생 버퍼와 공간적 지역성을 위한 공간적 버퍼 그리고 동적 페칭 유닛으로 구성된다. 시간적 지역성을 위한 희생 버퍼는 쓰기 동작이 수행된 데이터가 L1 캐쉬로부터 플래시에 쓰기 동작을 수행하기 위한 데이터를 저장하므로 기존의 L2 캐쉬처럼 다시 읽기 및 쓰기 동작을 수행하는 데이터에 효과적이지만 지우기 연산을 줄이기 위한 메커니즘이 존재하지 않는 단점을 가진다.

따라서 본 논문에서는 작은 용량의 버퍼를 사용하여 NAND 플래시 메모리의 가장 큰 문제점인 쓰기 및 지우기 동작을 줄이며, 버퍼에서 쓰기 동작 요청 혹은 NAND 플래시 메모리에서 지우기 동작시 효과적인 동작을 수행하는 버퍼 관리 시스템을 제안한다.

III. 제안된 NAND 플래시 메모리 버퍼 관리 시스템

이 장에서는 연구 개발 동기에 대하여 설명하고, 아울러 SRAM 버퍼를 내장한 NAND 플래시 메모리에 대한 구조적 모델 및 공간적 지역성에 효과적인 구조와 메커니즘의 구체적인 동작에 대해서 설명한다.

3.1 제안 동기 및 방법

본 연구의 목적은 NAND 플래시 메모리에 작은 용량의 캐쉬 메모리 버퍼를 이용하여 NAND 플래시 메모리의 가장 큰 단점인 쓰기 및 지우기 동작에 대한 데이터 버퍼 관리 시스템을 설계하는 것이다.

본 논문에서는 NAND 플래시 메모리를 위한 가장 효과적인 버퍼의 구조와 크기 그리고 페칭 크기를 위해서 선행 연구로 기존의 상용화된 캐쉬 메모리 버퍼를 NAND 플래시 메모리에 접목하여 버퍼의 접근 실패율을 평가하였다. 선행 연구로 TLB(translation lookaside buffer)처럼 작은 용량으로 좋은 성능을 보이는 완전연관 버퍼를 사용하여 버퍼 크기와 성능의 상관관계에서 가장 좋은 버퍼의 크기 및 페칭 크기를

선택하였다. 완전연관 버퍼는 작은 용량으로 가장 좋은 접근 실패율을 가지는 구조이다.

본 논문에서는 완전연관 버퍼의 크기를 8Kbyte에서 128Kbyte까지 증가시켰으며, 버퍼 페칭 크기는 일반적인 L1 캐쉬의 페칭 크기인 32byte부터 NAND 플래시 메모리의 한 페이지 크기인 2048byte까지의 접근 실패율을 확인하였다. 결론적으로 본 논문에서는 버퍼의 크기당 가장 좋은 성능의 향상을 보이는 16Kbyte의 완전연관버퍼를 기본 버퍼의 크기로 선택하였다.

NAND 플래시 메모리에서 가장 큰 문제점은 동일 셀에 대하여 덮어 쓰기가 불가능하다. 하나의 쓰기 동작은 차후 블록의 지우기 동작이 수반된다. 특히 지우기 동작은 블록 단위로 이루어지며, 이는 많은 동작 시간을 요구하게 되므로 전체 시스템에 가장 큰 성능저하의 원인이 된다.

그림 1은 본 논문에서 선택한 16Kbyte의 완전연관 버퍼에서 다양한 버퍼의 페칭 크기를 가질 때 NAND 플래시에 쓰기 동작 비율을 나타낸 그림이다. 16Kbyte 완전연관 버퍼에서 시간적 지역성과 공간적 지역성을 고려하여 페칭 크기는 256byte부터 2048byte까지 실행하였다. 16Kbyte의 완전연관 버퍼에서 NAND 플래시 메모리에 가장 적은 쓰기 동작 요청에 대한 페칭 크기는 1024byte이다. 버퍼 접근 실패율 역시 동일한 페칭 크기에서 가장 좋은 결과를 보였다.

cjpeg와 rasta의 경우, 짧은 시간의 공간적 지역성과 높은 시간적 지역성의 특징을 가지므로 2048byte의 페칭 크기에서 높은 쓰기 동작 비율을 보이고 있다. 반면 epic과 unepic의 경우, 공간적 지역성이 강한 특성으로 페칭 블록이 클수록 좋은 쓰기 동작의 성능을 보이고 있다.

djpeg와 mpegde의 경우, 시간적 지역성과 공간적 지역성에 대하여 한쪽의 특성에 치우치지 않는 특성을 보이지 있다. 256byte와 1024byte의 페칭 크기는 512byte와 2048byte의 페칭 크기에 비해 좋은 성능을 보이고 있다. 이는 djpeg와 mpegde는 공간적 지역성과 시간적 지역성의 특성을 비슷한 비율로 분포되어 있기 때문이다.

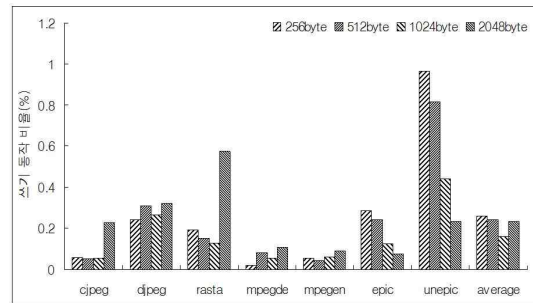


그림 1. 다양한 페칭 크기를 가지는 완전연관 버퍼(16Kbyte)의 NAND 플래시 메모리에 쓰기연산 비율

Fig. 1. Write operation counts of fully associative buffer with various fetching sizes in NAND Flash memory

시뮬레이션 결과처럼 데이터는 프로그램의 특성에 따라 공간적 지역성과 시간적 지역성의 특성이 서로 다르게 나타난다. 공간적 지역성은 순차적인 접근에 강한 특성으로 최근 참조되었던 데이터의 이웃 데이터가 참조 되어질 확률이 높은 것으로 큰 페칭 크기에 효과적이며, 시간적 지역성은 한번 참조되었던 데이터는 가까운 시간에 다시 참조 되어질 확률이 높다는 것을 의미하며, 버퍼의 엔트리 수가 많을수록 효과적이다.

완전연관 버퍼는 접근 실패율이 가장 좋은 구조임에도 불구하고 프로그램의 가장 큰 특성인 공간적/시간적 지역성을 효과적으로 이용하지 못하는 구조이다. 하지만 NAND 플래시 메모리에 작은 페칭 크기를 가지는 버퍼의 이용은 시간적 지역성을 높여줄 수 있지만 이는 오히려 NAND 플래시 메모리의 많은 쓰기 동작을 초래할 수 있다.

본 논문에서는 시간적 지역성과 공간적 지역성을 이용할 수 있는 큰 페칭 크기를 가지는 공간적 버퍼와 작은 페칭 크기를 가지는 시간적 버퍼로 이중 구조의 버퍼를 사용하였다. 큰 페칭 크기를 가지는 공간적 버퍼에서 참조가 이루어진 작은 페칭 크기는 시간적 버퍼에 저장하게 된다. 하지만 작은 페칭 크기의 시간적 버퍼는 동일한 페이지에 많은 쓰기 동작을 요청 할 수 있다. 따라서 본 논문에서는 NAND 플래시 메모리에 쓰기 동작 요청시 시간적 버퍼에 존재하는 동일한 페이지의 쓰기 데이터를 모두 동시에 NAND 플래시 메모리에 쓰기를 수행할 뿐 아니라 지우기 동작시 NAND 플래시 메모리의 블록에 존재하는 데이터 검색하여 동시에 적용하게 된다.

3.2 구조적 특징과 동작모델

제안된 버퍼 시스템에서 공간적/시간적 버퍼 모두 완전연관 버퍼로 구성된다. 공간적 버퍼의 동작은 기존의 완전연관

버퍼와 동일하게 FIFO로 구성하였다. 공간적 버퍼의 페칭 크기는 n 개의 작은 페칭 크기의 총합과 같다. 예로 시간적 버퍼가 128byte의 페칭 크기일 때 공간적 버퍼의 페칭 크기가 $n * 128\text{byte}$ 의 페칭 크기를 가진다. 따라서 공간적 버퍼에 n 개의 hit 비트와 dirty 비트를 가진다. 만약 공간적 버퍼가 1024byte의 페칭 크기 그리고 시간적 버퍼가 128byte라면, 공간적 버퍼에 각 페칭 블록들은 8개의 hit 비트와 8개의 dirty 비트를 가지게 된다.

공간적 버퍼에서 버퍼 접근이 성공하면, 참조 되어진 공간적 버퍼의 페칭 블록에서 n 개의 hit 비트와 dirty 비트들 중 참조되어진 작은 페칭 블록의 비트들이 갱신된다. 즉 읽기 요청의 경우 접근 성공시 해당 작은 블록의 데이터를 CPU로 전송함과 동시에 hit 비트를 "1"로 셋하게되며, 쓰기 요청의 경우 접근 성공시 해당 작은 블록의 데이터를 해당 작은 블록에 씌고 동시에 hit 비트와 dirty 비트 모두를 "1"로 셋하게 된다. 시간적 버퍼의 경우에는 dirty 비트만 존재하게 되며 구동 방식은 공간적 버퍼와 동일하다.

공간적 버퍼에 더 이상 페칭 블록을 저장할 무효 페칭 블록이 없을 경우, FIFO 동작으로 제일 먼저 들어온 페칭 블록을 추출하게 된다. 이때 추출 되어지는 공간적 버퍼의 페칭 블록에서 한번 참조되어진 작은 페칭 블록, 즉 hit bit가 "1"로 셋 되어진 작은 블록을 시간적 버퍼에 선택적으로 저장하게 된다.

시간적 버퍼 역시 완전연관 버퍼로 FIFO 동작으로 이루어진다. NAND 플래시 메모리로 쓰기 동작은 시간적 버퍼에서 추출 되어지는 수정 데이터에서만 이루어진다. 시간적 버퍼에 더 이상 무효 페칭 블록이 없고 추출 되어지는 시간적 버퍼의 페칭 블록이 이미 데이터가 갱신되었다면(dirty 비트 "1"), 시간적 버퍼에서 동일한 NAND 플래시 페이지에 속하는 모든 작은 페칭 블록들을 시간적 버퍼에서 찾아 NAND 플래시 메모리에 저장하게 된다. 이는 플래시 메모리 내의 한 페이지에 해당하는 레지스터가 2Kbyte 임으로 한번 쓰기 동작시 동일한 페이지 데이터를 모아서 한 번에 쓸 경우 잦은 쓰기 동작을 줄일 수 있다.

또한 NAND 플래시 메모리에서 지우기 연산이 발생하면, 시간적 버퍼는 지우기 동작이 발생한 NAND 플래시 메모리의 블록에 포함되는 쓰기 동작이 수행된(dirty 비트 "1") 페칭 블록들을 검색하게 되며, NAND 플래시 메모리의 지우기 동작이 수행되는 블록이 무효 블록으로 데이터 이동시 선택되어진 시간적 버퍼의 페칭 블록들을 모두 관련 페이지에 저장하게 된다. 이때 선택되어진 시간적 버퍼의 페칭 블록들 dirty 비트를 "0"으로 갱신하게 된다.

본 논문에서는 버퍼에서 쓰기 동작의 지연을 위해 시간적 버퍼에서 공간 확보를 위한 메커니즘을 추가 하였다. 만약 공간적 버퍼가 프로그램 수행시 공간적 지역성의 패턴 주기보다 작은 용량의 버퍼를 사용하게 되면, 충분한 공간적 지역성을 위한 참조가 일어나지 못하게 된다. 그리고 공간적 버퍼에 추출 되는 페칭 블록들중 참조가 발생한 작은 페칭 블록들은 시간적 버퍼에 저장하게 된다. 이때 가까운 시간에 다시 추출된 공간적 버퍼의 페칭 블록중 시간적 버퍼에 저장되지 못한 데이터의 접근이 발생하게 되면, 시간적 버퍼는 공간적 버퍼에 포함되는 데이터를 가지게 된다. 이러한 데이터 저장은 시간적 버퍼의 공간을 낭비함과 동시에 잦은 NAND 플래시 메모리의 접근의 원인이 된다. 또한 시간적 버퍼의 데이터가 이미 쓰기 동작이 갱신되었다면(dirty 비트 "1") 메모리 일관성(coherence)이 깨지는 원인이 된다.

따라서 제안된 버퍼 시스템에서는 데이터의 일관성과 시간적 버퍼의 공간 확보를 위해 두 버퍼에서 모두 접근 실패가 발생시, NAND 플래시 메모리로부터 공간적 버퍼로 페칭 블록을 저장될 때, 시간적 버퍼에서 공간적 버퍼의 페칭 블록에 포함되는 페칭 블록이 있는지 찾게 된다. 만약 시간적 버퍼에서 공간적 버퍼에 포함되는 페칭 블록들이 존재한다면, 그 페칭 블록들은 NAND 플래시 메모리로부터 가져오는 공간적 페칭 블록에 덮어 쓰게 된다. 이때 공간적 버퍼에 포함된 시간적 버퍼의 페칭 블록들은 모두 무효 블록으로 처리된다. 이러한 동작은 시간적 버퍼의 공간을 효과적으로 사용할 뿐 아니라 NAND 플래시 메모리에 쓰기/지우기에 대한 지연과 데이터 일관성을 보장 할 수 있다.

그림 2는 제안된 버퍼 시스템의 구조 및 버퍼 접근 실패시 동작을 나타낸 그림이다. 예로 공간적 버퍼의 한 엔트리는 4개의 시간적 버퍼의 페칭 블록 크기의 합과 같고, NAND 플래시 메모리의 한 블록은 4개의 페이지(A, D, E, H...)를 가지고 있다고 가정한다. 그림 2(a)는 공간적 버퍼와 시간적 버퍼에 각 페칭 블록들을 나타낸 그림이다. A, B, C, D는 NAND 플래시 메모리의 각 페이지를 나타내며, (1), (2)..은 한 페이지에 속하는 시간적 지역성의 페칭 블록을 나타낸다. 만약 두 버퍼에서 접근 실패가 발생했다면, 공간적 버퍼는 NAND 플래시 메모리로부터 페이지 A(1),(2),(3),(4)를 요청하게 된다. 공간적 버퍼에서 가장 먼저 들어온 페칭 블록 중 참조가 발생한 A(5)가 시간적 버퍼에 저장되며, 시간적 버퍼에서는 D(1)이 추출되어 진다. 이때 D(1)은 이미 쓰기동작이 이루어진 페칭 블록이므로(dirty 비트 "1") NAND 플래시 메모리에 쓰기 동작이 수행되어야한다. 이때 시간적 버퍼에서 동일한 페이지에 속하는 D(5)의 페칭 블록

역시 NAND 플래시 메모리에 D(1)과 함께 쓰기 동작이 이루어진다. 이와동시에 D(5)의 dirty 비트는 “0”으로 변경된다. D(3)의 경우 같은 페이지에 속하는 페칭 블록이지만 쓰기 동작이 일어나지 않는 페칭 블록임으로 본 동작에 관여하지 않는다.

또한 시간적 버퍼의 A(1)와 A(3)는 NAND 플래시 메모리부터 공간적 버퍼로 저장되는 페칭 블록에 포함되는 페칭 블록으로 시간적 버퍼의 공간 활용과 데이터의 일관성을 위해 공간적 버퍼에 A(1)와 A(3)를 덮어 쓰게 된다. 이때 hit 비트와 dirty 비트는 동일한 값을 가진다. 그림 2(b)는 NAND 플래시 메모리에 쓰기 동작 후, 공간적 버퍼와 시간적 버퍼의 상태를 나타낸 그림이다.

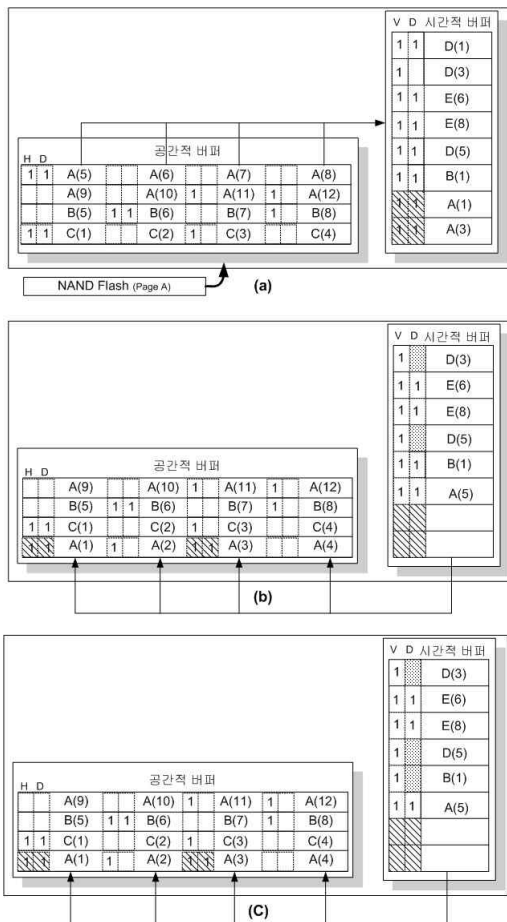


그림 2 제안된 버퍼 시스템의 쓰기 및 지우기 동작 알고리즘
 Fig. 2. write and erase operation Algorithm of Proposed buffer system.

그림 2(c)는 NAND 플래시 메모리가 지우기 동작이 수행 될 때, 시간적 버퍼와 공간적 버퍼의 상태를 나타낸 그림이다. 만약 그림 2(a)에서 추출 되는 페칭 블록 D(1)이 NAND 플래시 메모리에 쓰기 동작시 NAND 플래시 메모리의 해당 블록에 더 이상 쓰기 동작을 위한 무효 페이지가 없다면, NAND 플래시 메모리의 다른 무효 블록으로 찾아 페이지(A,B,C,D)들을 쓰게 된다. 이때 시간적 버퍼는 NAND 플래시 메모리의 지우기 동작이 수행되는 블록에 포함되는 페이지중 쓰기 동작이 이루어진(dirty 비트 “1”) 페칭 블록들을 모두 찾는다. 선택되어진 시간적 버퍼의 페칭 블록들은 NAND 플래시 메모리에 저장되고 dirty 비트를 “0”으로 갱신된다.

즉, 제안된 버퍼 관리 시스템은 완전연관 버퍼인 시간적/공간적 버퍼로 구성되며, 시간적 버퍼로부터 NAND 플래시 메모리에 쓰기 동작이 발생할 때, NAND 플래시 메모리의 페이지에 포함되는 이미 쓰기 동작이 이루어진 시간적 페칭 블록들을 모두 저장하게 되며, 더욱이 쓰기 동작을 위한 더 이상 무효 페이지가 없을 경우, NAND 플래시 메모리의 해당 블록은 다른 무효 블록에 페이지들을 쓰게 된다. 이때 시간적 버퍼는 NAND 플래시 메모리 블록에 포함되는 이미 쓰기 동작이 이루어진 모든 시간적 버퍼의 페칭 블록을 찾아 NAND 플래시 메모리에 적어주게 된다. 이때 선택 되어진 시간적 버퍼의 페칭 블록들은 모두 dirty 비트가 “0”으로 갱신된다.

또한 시간적 버퍼의 효과적인 공간 활용과 두 버퍼의 접근 실패시 데이터 일관성 그리고 쓰기 동작의 지연을 위해 두 버퍼의 접근 실패시, 시간적 버퍼는 NAND 플래시 메모리부터 저장되는 공간적 버퍼의 페칭 블록에 포함되는 페칭 블록을 찾아 공간적 버퍼에 덮어 쓰게 된다. 이때 해당되는 공간적 버퍼의 hit 비트는 “1”로 갱신되며, dirty 비트는 시간적 버퍼의 상태에 따라 갱신이 이루어진다.

IV. 시뮬레이션을 통한 성능평가

이 장에서는 시뮬레이션 환경 및 성능 평가에 대해 자세히 설명한다. 사용된 벤치마크는 멀티미디어 응용을 대표하는 미디어벤치마크를 이용하여 성능 평가를 하였다[10]. 이 벤치마크들의 트레이스를 생성하기 위해 SimpleScalar3.0을 [11] 사용하여 각각 1억 개의 명령을 수행하는 동안 데이터 참조 주소를 모니터링 하여 사용하였다. 시뮬레이션을 위한 시스템으로 본 연구에서는 다양한 버퍼 시스템을 가지는 플래시 메모리 자체 성능을 평가하였다. 기본 시뮬레이션 환경 변수는 표 1과 같다[12].

표 4. 시뮬레이션 변수들
Table 1. Simulation Parameters

System Parameters	Value
Page size	2kbyte
Block size	128kbyte(64page)
Erase operation time	1.5ms
Write operation time	200μs
Random read time	20μs
Serial read time	20ns
Buffer access time	50ns

4.1 제안된 버퍼시스템 성능평가

본 연구에서는 NAND 플래시 메모리의 성능 평가를 위해 하이브리드 맵핑(hybrid mapping)[13] 알고리즘을 사용하였다. 제안된 NAND 플래시 메모리의 페이지 크기는 2Kbyte이며, 블록 크기는 128Kbyte로 64개의 페이지로 구성하였다.

NAND 플래시 메모리 시스템의 성능 평가는 버퍼 접근 실패율(miss ratio)과 평균 메모리 접근 시간(Average Memory Access Time)을 제안된 버퍼관리 시스템과 다른 기존의 비교버퍼 시스템을 가진 NAND 플래시 메모리 성능을 비교, 분석하였다. 여기서 버퍼 접근 실패율은 NAND 플래시 메모리를 위한 버퍼에서의 접근 실패율을 의미하며, 평균 메모리 접근 시간은 버퍼 시스템의 접근 실패로부터 요청한 페칭 블록이 NAND 플래시 메모리로부터 버퍼 저장 될 때까지 지연된 시간을 의미한다.

제안된 버퍼 시스템에서 공간적 버퍼는 3Kbyte 크기로 페칭 블록 크기는 비교 버퍼들의 가장 좋은 성능을 보였던 1024byte 페칭 블록 크기 선택하였다. 시간적 버퍼는 5Kbyte 크기의 128byte의 페칭 블록 크기를 가진다. 5Kbyte의 시간적 버퍼의 페칭 크기는 선행 연구에서 4Kbyte의 완전연관 버퍼에서 가장 좋은 성능을 보인 크기로 제안된 시간적 버퍼와 비슷한 크기로 128byte를 선택하였다.

성능 평가를 위한 비교 버퍼로 기존의 캐쉬 메모리에서 접근 실패율이 좋은 구조인 8-웨이 연관사상 버퍼(8-웨이), 희생(Victim) 버퍼, 그리고 완전연관 버퍼를 사용하였다. 각 비교 버퍼의 페칭 크기와 버퍼의 크기의 선정은 선행연구로 버퍼크기는 4Kbyte~128Kbyte 그리고 페칭 크기는 32byte~2048byte까지 시뮬레이션 결과 버퍼 크기 면에서 접근 실패율이 가장 효과적인 버퍼 크기와 페칭 크기를 선택하였다. 8-웨이 연관사상 버퍼, 희생 버퍼 그리고 완전연관 버퍼 모두 16Kbyte의 버퍼 크기에서 1024byte의 페칭 크기를 가질 때 가장 좋은 성능을 보였다. 따라서 비교 버퍼의 크

기는 모두 1024byte의 페칭 크기를 가지는 16Kbyte의 버퍼 크기로 선정하였다. 단 희생 버퍼의 경우 완전연관 버퍼의 페칭 블록 수는 4개로 선정하였다.

그림 3은 버퍼 접근 실패율을 나타낸 그림이다. 제안된 버퍼 시스템은 다른 비교 버퍼에 비해 다소 높은 접근 실패율을 보이고 있다. cjpeg, djpegd 그리고 epic에서 비교 버퍼와 제안된 버퍼의 접근 실패율이 가장 큰 차이를 보이고 있다. 이는 cjpeg, djpeg 그리고 epic의 경우 참조 패턴이 제안된 버퍼 시스템의 용량에 비해 크기 때문에 비교버퍼와 큰 차이를 보인다. 특히 공간적 지역성을 고려할 때 제안된 버퍼는 비교 버퍼에 비해 5배의 작은 크기를 가진다. 그럼에도 불구하고 나머지 벤치마크에 대해서는 큰 성능 저하를 보이지 않고 있다. 물론 제안된 구조에 대하여 비교버퍼들과 동일한 버퍼 크기(16Kbyte)를 기준으로 시뮬레이션을 수행한 결과 8-웨이 연관사상 버퍼, 희생(Victim) 버퍼, 그리고 완전연관 버퍼에 비해 각각 평균 32%, 30%, 5%의 성능향상을 보였다. 그러나 제안된 구조에 대하여 8Kbyte의 사이즈를 선택한 이유는 플래시 메모리의 성능 평가가 쓰기 및 지우기 연산에 매우 영향을 많이 받기 때문에 후에 설명할 평균 메모리 접근 시간을 보면 버퍼 접근 실패율이 전체 성능에 큰 영향을 미치지 않음을 알 수 있다. 그러므로 비용 및 전력적인 측면에서 본 논문은 작은 버퍼 사이즈를 최종적으로 선택하게 되었다.

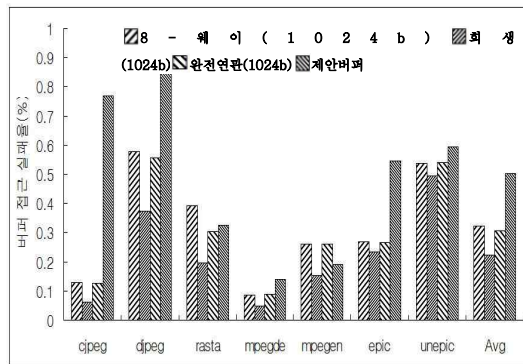


그림 3. 각 버퍼 시스템의 버퍼 접근 실패율
Fig. 3. Miss ratio of various buffer systems

그림 4는 각 버퍼들의 NAND 플래시 메모리에 쓰기 동작 감소율(a)과 지우기 동작 감소율(b)을 나타내고 있다. 쓰기 동작과 지우기 동작의 감소율은 완전연관 버퍼를 기준으로 평균화하여 나타냈다. 이는 제안된 버퍼의 구조가 모두 완전연관 버퍼로 구성되기 때문이다. 제안된 버퍼 시스템은 다른 비교 버퍼에 비해 다소 높은 버퍼 접근 실패율을 가짐에도 불구하고 쓰기 동작과 지우기 동작에서 좋은 성능향상을 보였다.

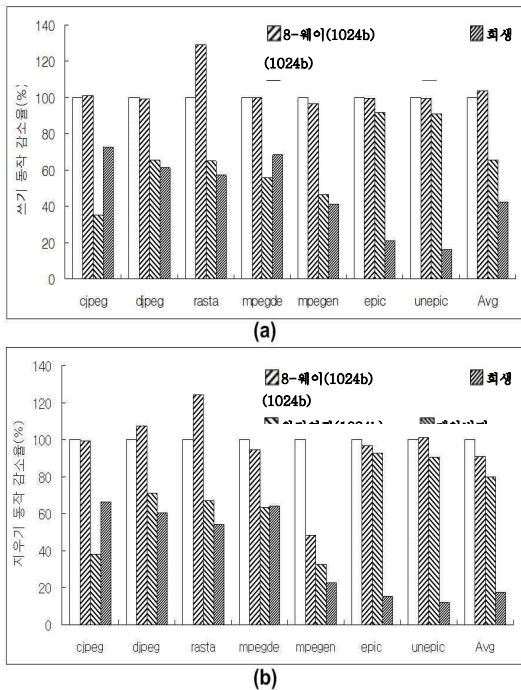


그림 4. NAND 플래시 메모리의 쓰기 동작(a)과 지우기 동작(b) 감소율
 Fig. 4. write operation (a) and erase operation decrease of NAND flash memory

그림 4(a)의 쓰기 동작 감소율에서 제안된 버퍼는 cjpeg, djpeg 그리고 mpegde는 버퍼 접근 실패율과 달리 제안된 버퍼 시스템은 희생 버퍼를 제외하고는 좋은 성능향상을 보이고 있다. 이는 cjpeg, djpeg 그리고 mpegde는 한 페칭 블록에 대해 연속적인 접근의 공간적 지역성 보다는 시간 간격에 따른 접근의 공간적 지역성이 높기 때문이다. 즉 한 페칭 블록에 대해 시간 간격에 따른 공간적 지역성은 시간적 지역성과 유사하게 버퍼의 엔트리수가 많을수록 유리하다. 제안된 버퍼는 공간적 지역성을 위한 엔트리보다 비교 버퍼들은 약 5배 높은 엔트리를 가지지만, 128byte의 페칭 크기를 가지는 시간적 버퍼를 가지므로 cjpeg, djpeg, mpegde에서 제안된 쓰기 동작을 줄이는 알고리즘으로 높은 쓰기 동작 감소율을 가질 수 있었다. 그 외 다른 벤치마크들에서도 제안된 버퍼 시스템이 비교 버퍼들에 비해 높은 쓰기 동작 감소율을 보이고 있다. 이는 제안된 버퍼 구조가 공간적 지역성과 시간적 지역성을 효과적으로 사용할 수 있으며, 특히 공간적 지역성에 대한 데이터들은 제안된 쓰기 동작 알고리즘으로 NAND 플래시 메모리의 쓰기 동작시 좋은 성능향상을 이룰 수 있기 때문이다. 따라서 제안된 버퍼 시스템의 쓰기 동작이 작은 용량

의 버퍼에서 효과적이다. 특히 epic과 unepic의 경우, 8웨이 연관사상 버퍼와 희생 버퍼에 비해 평균 77%의 낮은 쓰기 동작을 보였다.

또한 cjpeg(희생 버퍼 제외), djpeg 그리고 mpegde의 경우 쓰기 동작 감소율과 같이 지우기 동작에서는 제안된 버퍼 시스템이 좋은 성능을 보였다. 이는 제안된 버퍼는 NAND 플래시 메모리의 블록 지우기 동작시 시간적 버퍼에서 NAND 플래시 메모리에 관련된 이미 쓰기 동작이 이루어진 페칭 블록들을 모두 적어주게 된다. 최상의 경우, 만약 시간적 버퍼의 모든 엔트리가 다른 페이지의 페칭 블록들로 저장되어져 있고, 모두 쓰기 동작이 이루어진 상태라면, 제안된 버퍼 시스템은 한번의 지우기 동작에서 시간적 버퍼의 엔트리 수 만큼의 NAND 플래시 메모리 지우기 횟수를 줄일 수 있기 때문이다.

쓰기 동작 및 지우기 동작 감소율에서 cjpeg는 희생 버퍼에 비해 낮은 성능향상을 보이고 있다. 이는 희생 버퍼는 16Kbyte의 직접사상 버퍼와 4Kbyte의 완전연관 버퍼로 총 20Kbyte의 크기를 가진다. 더욱이 공간적 지역성에 효과적인 큰 페칭 블록을 가질 뿐 아니라 직접사상 버퍼와 공간적 버퍼의 스왑 알고리즘으로 시간적 지역성에도 효과적인 구조이기 때문이다.

결과적으로, 제안된 버퍼 시스템은 완전연관 버퍼에 비해 58%, 8웨이 버퍼에 비해 60% 그리고 희생 버퍼에 비해 36%의 쓰기 동작 감소율을 보였으며, 지우기 동작에서는 완전연관 버퍼에 비해 83% 그리고 8웨이 버퍼와 희생 버퍼에서 81%, 78%의 감소율을 이루었다.

그림 5는 가장 중요한 성능 평가 지표인 NAND 플래시 메모리의 평균 메모리 접근 시간을 나타낸 그림이다.

평균 메모리 접근 시간은 그림 4의 쓰기와 지우기 동작의 감소율 결과와 비슷한 그래프 형태를 보이고 있다. 역시 쓰기 동작 및 지우기 동작 감소율에서와 같이 epic과 unepic에서 평균 메모리 접근 시간이 가장 좋은 성능향상을 보이고 있다. 이는 epic과 unepic는 순차적인 공간적 지역성이 강한 특성을 가지며, 제안된 버퍼 시스템 동작은 공간적 지역성이 높은 데이터에 대해서 효과적으로 쓰기 동작과 지우기 동작을 줄일 수 있기 때문이다.

비록 버퍼 접근 실패율에서 제안된 버퍼가 cjpeg와 djpeg에서 다른 비교 버퍼에 비해 높은 실패율을 보였지만, 지우기 동작에서 평균 각각 25%와 8%를 줄이므로 평균 메모리 접근 시간에서는 비교 버퍼들과 비슷한 성능을 보이고 있다.

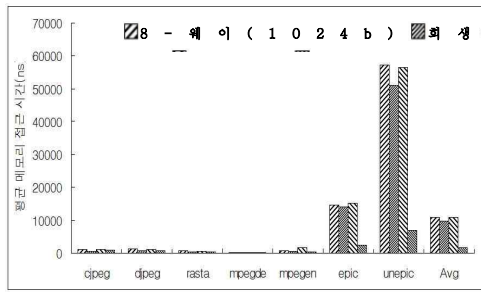


그림 5. NAND 플래시 메모리의 평균 메모리 접근 시간
Fig. 5. Average memory access time of NAND flash

실제 NAND 플래시 메모리 시스템에서 버퍼 접근 시간 (70ns)에 비해 NAND 플래시 메모리의 쓰기 동작(20us)와 지우기 동작(1.5ms)이 큰 오버헤드를 가진다. 즉, NAND 플래시 메모리에 쓰기 동작과 지우기 동작을 줄이므로 NAND 플래시 메모리의 전체 성능향상을 이룰 수 있다. 시뮬레이션 결과, 그림 5에서 보듯이 평균적으로 비교 버퍼에 비해 평균 메모리 접근 시간을 약 80%의 성능향상을 이루었다.

V. 결론

본 논문에서는 NAND 플래시 메모리에 효과적으로 쓰기 동작과 지우기 동작을 위한 버퍼 관리시스템을 제안하였다. 제안된 시스템은 완전연관 버퍼인 시간적 버퍼와 공간적 버퍼로 구성되어진다. NAND 플래시 메모리의 쓰기 동작은 작은 페칭 블록을 가지는 시간적 버퍼에서 이미 쓰기 동작이 이루어진 페칭 블록이 추출될 때 이루어진다.

본 논문에서는 NAND 플래시 메모리에 쓰기 동작시 NAND 플래시 메모리에 접근하는 블록에 페이지를 쓰기 위한 무효 페이지가 있다면, 시간적 버퍼는 NAND 플래시 메모리에 포함되는 이미 쓰기 동작이 이루어진(dirty 비트 "1") 페칭 블록을 찾아서 모두 NAND 플래시 메모리에 적어준다. 만약 NAND 플래시 메모리에 접근한 블록에 페이지를 쓰기 위한 유효공간이 없다면, NAND 플래시 메모리의 블록을 다른 무효 블록에 저장할 때, 시간적 버퍼는 NAND 플래시 메모리에 접근한 블록의 페이지들에 포함되는 이미 쓰기 동작이 이루어진 모든 페칭 블록을 NAND 플래시 메모리에 적게 된다. 이때 시간적 버퍼에서 선택되어진 페칭 블록들의 dirty 비트들은 모두 "0"으로 갱신된다.

또한 시간적 버퍼에서 쓰기 지연 및 데이터의 일관성을 위해 NAND 플래시 메모리로부터 공간적 버퍼에 저장되는 페칭 블록이 시간적 버퍼에 존재하는지 찾게 된다. 만약 시간적

버퍼에 공간적 버퍼에 저장되는 페칭 블록중 일부가 존재한다면, 시간적 버퍼의 페칭 블록은 공간적 버퍼의 페칭 블록에 덮어 쓰게 된다. 이때, 공간적 버퍼의 해당 hit 비트는 "1"로 갱신되며, dirty 비트는 시간적 버퍼의 dirty 비트에 따라 갱신이 된다.

제안된 버퍼 시스템에서는 버퍼가 없는 기존 NAND 플래시 메모리에 비해서는 쓰기 동작과 지우기 동작을 각각 99%를 줄였다. 또한 두 배 크기의 비교 버퍼를 사용한 NAND 플래시 메모리에 비해 다소 높은 버퍼 접근 실패율을 보이지만, 8웨이 연관버퍼에 비해 60%, 희생 버퍼에 비해 36% 그리고 완전연관 버퍼에 비해 58%의 쓰기 동작 감소율을 보였다. 또한 지우기 동작에서도 비교 버퍼들에 비해 평균 80%의 성능향상을 이루었으며, 전체적인 평균 플래시 메모리 접근 시간에서는 비교 버퍼들에 비해 평균 83%의 성능 향상을 이루었다.

참고문헌

- [1] N. Ballard, "State of PDAs and Other Pen-Based Systems," In Pen Computing Magazine, pp.14-19, Aug. 1994.
- [2] B. March, F. Dougliis, and P. Krishnan, "Flash Memory File Caching for Mobile Computer," In Proc. of the 27 Hawaii International conference on System Science, pp.451-460, Jan. 1994.
- [3] Needham & Company, LCC, NAND vs. Hard Disk Drives: Hype, Myth and Reality, Oct. 2005.
- [4] Jung-Wook Park, Seung-Ho Park, Charles C. Weems, Shin-Dug Kim, "A hybrid flash translation layer for SLC-MLC flash memory based multibank solid state disk", Microprocessors and Microsystems, Vol.35, Issue 1, pp.48-59, Feb. 2011.
- [5] D. Jung, Y.H. Chae, H. Jo, J.S. Kim and J. Lee, "A Group-Based Wear-Leveling Algorithm for Large-Capacity Flash Memory Storage Systems", Proc. The 2007 international conference on Compilers, architecture, and synthesis for embedded systems, pp. 160-164, 2007.
- [6] B. Jung, J. Lee, "The Smart buffer NAND Flash memory System for a hard-disk substitution." Journal of The Korea Society of Computer and Information, Vol 14, No 3, pp. 41-50, Mar. 2009.
- [7] J. Lee, G. Park, and S. Kim, "A New NAND-type

Flash Memory Package with Smart Buffer System for Spatial and Temporal Localities," Journal of System Architecture, Vol.51 No.2 pp.111-123, Feb. 2005.

[8] C. Park, J. Seo, S. Bae, H. Kim, S. Kim and Bumsoo Kim, "A low-cost memory architecture with NAND XIP for mobile embedded systems," In Proc. of the 1st CODES-ISSS'03, pp.138-143, Oct. 2003.

[9] H. S. Jo, J. U Kang, S. Y Pack, " FAB: Flash-aware buffer management policy for portable media players," Consumer Electronics, IEEE Transactions on, pp.485-493, Apr. 2006

[10] Samsung Elec., "NAND-type Flash Memory," <http://www.samsungelectronics.com/semiconductor/flash/Falsh.html>

[11] Mediabench, <http://euler.slu.edu/~fritts/mediabench/>

[12] D. Burger and T. M. Austin, "The SimpleScalar tool set, version 2.0, Technical Report TR-97-1342," University of Wisconsin-Madison, 1997.

[13] S. Lee, D. Park and H. Song, "A Log Buffer-Based Flash Translation Layer Using Fully-Associative Sector Translation," ACM Transactions on Embedded Computing Systems, Vol. 6, No. 3, July, 2007.

저자 소개



정 보 성
 2008년 2월: 경상대학교 제어 측공
 학과(석사)
 2008년~현재: 경상대학교 제어 측공
 학과 박사과정
 관심분야: 마이크로프로세서, 캐쉬 및
 플래시 메모리
 e-mail : blueking80@gnu.ac.kr



이 정 훈
 2004년 2월: 연세대학교 컴퓨터 학과
 (박사)
 2004년~현재: 경상대학교 제어 측공
 학과 부교수
 관심분야: 고성능 컴퓨팅, 내장형 시
 스템 및 SOC 시스템
 e-mail: leejh@gsnu.ac.kr