

논문 2011-06-48

이중 NAND 플래시 구조의 버퍼시스템에서 효율적 버퍼 크기

(The Efficient Buffer Size in A Dual Flash Memory Structure with Buffer System)

정 보 성, 이 정 훈*

(Bo-Sung Jung, Jung-Hoon Lee)

Abstract : As we know the effects of cache memory research, instruction and data caches can be separated for higher performance with Harvard CPUs. In this paper, we shows the efficiency of buffer system in the instruction and data flash storage medium. And we analyzed characteristics of the data and instruction flash and evaluated the performance. Finally, we propose the best buffer structure with an optimal block size and buffer size for the instruction and data flash.

Keywords : Flash memory, characteristics of instruction and data, dual structure, cache memory

1. 서론

오늘날 낮은 가격에 높은 집적도를 얻을 수 있고 충격에 강한 특성을 가진 플래시로 대표되는 비휘발성 메모리는 스마트폰, MP3, 스마트패드 등 다양한 개인용 기기에 널리 쓰이고 있으며, 이와 같은 소형기기는 물론 노트북 및 데스크톱 컴퓨터의 저장장치로도 주목받고 있다[1]. 이러한 플래시 메모리는 크게 NAND형과 NOR형으로 구분된다. NAND형과 NOR형 모두 비트를 저장하기 위한 셀의 구조는 동일하지만 셀의 집합을 어떻게 구성하느냐에 따라 차이가 있고, 이로 인해 각각 성능 면에서 장/단점이 존재한다[2].

NAND형과 NOR형의 플래시 메모리 읽기 성능은 초기 접근시간(initial access time)에서 가장 큰 차이를 보인다. NAND 플래시 메모리의 초기접근시

간은 25 μ s정도이며 NOR 플래시 메모리는 읽기 모드에 따라 50ns~100ns로 약 250배~500배 정도가 차이가 난다. 초기 접근 이후에 해당 주소에서 시작하는 데이터를 연속적으로 읽을 경우에는 NAND형, NOR형 모두 비슷한 수준의 시간이 걸린다. 이러한 특징은 NAND 플래시 메모리가 무작위 접근(random access)에 있어서 성능이 상당히 떨어짐을 의미한다. 이러한 특징으로 인해 NOR 플래시 메모리는 주로 실행을 위한 코드를 저장하는데 사용되고, NAND 플래시 메모리는 주로 데이터를 저장하는데 사용된다. 그러나 비용적인 측면에서 NAND 플래시 메모리는 NOR 플래시 메모리에 비해 40%정도 저렴한 장점을 보이고 있다[3].

이에 NAND 플래시에 SRAM/SDRAM을 이용하여 NOR형의 빠른 접근시간의 효과를 얻고자 하는 연구가 활발히 진행되고 있다. 그 대표적인 사례가 XIP(execution-in-place)로 특정 패턴을 분석하여 자주 참조가 일어나는 페이지를 SRAM/SDRAM에 저장하여 성능을 개선하고자하는 연구가 진행되었으며[4], NAND 플래시에 버퍼 시스템을 적용하여 플래시 메모리 셀에 직접 읽기 및 쓰기 수행 빈도를 줄여주는 연구를 수행하였다[5]. 특히 플래시 셀의 쓰기 빈도를 줄여주는 것은 플래시 메모리의 가장 큰 오버헤드인 지우기 모드(erase mode)를 줄

* 교신저자(Corresponding Author)

논문접수 : 2011. 07. 04., 수정일 : 2011. 08. 12.,
채택확정 : 2011. 08. 16.

정보성, 이정훈 :경상대학교 ERI 제어계측공학과

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

여주는 효과를 얻을 수 있으며, 플래시 셀 사용 제한의 단점(wear leveling problem)까지 극복할 수 있는 중요한 요인으로써 작용한다. 이처럼 가격적인 측면으로 인해 NAND형 플래시와 다양한 버퍼 시스템을 이용하여 NOR형 플래시를 대체하기 위한 고성능화 연구가 현재 활발히 진행되고 있는 실정이다. 그러나 국내/외 플래시 연구자들이 간과하고 있는 핵심적인 부분이 있다. 바로 하바드구조(Harvard architecture)와 폰노이만구조(Von Neumann architecture)의 캐쉬 메모리의 특성[6]을 플래시 메모리 구조 설계에 전혀 고려하지 않고 있다. 시스템의 모든 프로그램은 크게 데이터와 명령어로 나누어진다. 폰노이만구조의 캐쉬는 단일 캐쉬(unified cache) 구조로써 명령어와 데이터를 하나의 단일 모듈로 동작함에 반해 하바드구조는 명령어 캐쉬와 데이터 캐쉬를 나누어줌으로써 각각의 특성에 맞는 형태를 갖추게 되고 이로 인하여 고성능 및 많은 새로운 형태의 프로세서가 생겨나게 되었다. 특히 폰노이만 아키텍처 구조처럼 단일 구조는 명령어 처리 후 데이터 접근이 이루어지므로 고성능의 데이터 처리를 위한 파이프라인에 불리할 뿐만 아니라 동일한 버스를 사용하므로 명령어와 데이터간의 버스 점유에 의한 병목현상 등 성능 저하의 문제점을 가지고 있다.

현재 NAND 플래시는 하드디스크와 같은 최하위 저장매체로써 고려되고 있다[7]. 그러나 기존의 하드디스크 시스템은 하드디스크의 느린 접근 속도로 인해 시스템 성능을 떨어뜨리는 단점을 보인다. 그러나 플래시 메모리는 하드디스크와는 다른 구조와 특성을 가지고 있다. 그럼에도 불구하고 최하위 저장 매체로써만 인식되고 있는 플래시 메모리를 하드디스크의 특성에만 맞추어 현재까지 폰노이만 구조처럼 명령어/데이터의 특성과 상관없이 단일 구조로써 설계되어져 왔다.

그러므로 플래시를 기존의 단일 구조에서 명령어 플래시와 데이터 플래시로 나누어 구성할 경우 각각의 특성 분석이 요구되며 고성능을 위한 버퍼 사용시 버퍼의 크기 및 블록 크기에 따른 다양한 특성 분석은 기존의 온칩 캐쉬 구조처럼 하바드 구조의 플래시 연구에 대단히 중요하다.

II. 관련연구

기존의 플래시 메모리 연구는 대부분이 셀 연구에 기반을 두고 있으며, 매핑 알고리즘인 FTL 혹은 비교적 많은 시간이 소요되는 쓰기 연산을 줄여 성

능 향상을 이루고자하는 연구들이 주 연구 대상이었다. 더욱이 플래시 메모리와 캐쉬 구조의 버퍼 시스템을 이용한 NAND 플래시 메모리의 구조적 연구가 현재 활발한 상태이며, 명령어 특성 및 데이터 특성을 모두 고려한 플래시 연구는 현재 수행되지 못한 상태이다.

기존의 버퍼 시스템을 활용한 연구를 살펴보면, 스마트 버퍼 시스템[8]은 프로그램 수행 시 두 가지 지역성을 이용하여 플래시로부터 큰 블록을 인출하는 공간적 버퍼와 온칩 캐쉬로부터 버려지는 작은 블록을 저장하는 시간적 버퍼로 구성된다. 이 연구는 시간적/공간적 지역성을 이용하여 고성능 NAND 플래시를 이루고자하였다. Huang[9]은 하나의 쓰기 버퍼와 2개의 읽기 버퍼를 이용하여 플래시의 쓰기 연산을 줄여 성능 향상을 이루고자하였다. 쓰기 연산의 데이터 요청시 쓰기 버퍼에, 읽기에 요청된 데이터는 읽기 버퍼에 각각 저장하게 된다. Jo[10]는 휴대용 미디어 장치를 위한 정책을 제안하였으며 FAB(flash-aware-buffer)는 버퍼에서 플래시의 블록들 중 가장 많은 페이지를 가지는 버퍼 블록들을 희생 블록으로 선택한다. 이러한 버퍼의 희생 블록 선택은 한 번의 플래시 메모리 접근으로 많은 수의 쓰기 연산이나 지우기 연산을 줄여 성능 향상을 보였다. Park[11]은 기존의 명령어를 담당하던 NOR 플래시 메모리를 XIP(Excuse In Place)기반의 SRAM과 NAND 플래시로 대체하여 저비용의 플래시 패키지를 구현하였으며, 참조가 많은 플래시 메모리 셀은 희생버퍼에 저장되고 그렇지 못한 페이지는 DRAM에 저장하게 된다. 참조가 빈번한 페이지를 버퍼에 저장하므로 빠른 메모리 접근과 플래시 메모리의 낮은 소비전력을 이끌었다. 또한 실시간 내장형 시스템에 적합한 요구 페이지징(demand paging)을 버퍼를 이용하여 그림자(shadowing) 기법 연구[12]가 이루어졌다. 이 연구는 실시간 내장형 시스템에서 코드 및 데이터를 NOR 플래시의 대체인 SRAM/DRAM과 NAND 플래시 메모리에서의 요구 페이지징 기법에 대한 최악의 응답 시간 분석을 위한 연구이다.

그러나 공간적 스마트 버퍼[8], Huang[9], 그리고 Jo[10] 등의 연구는 데이터와 관련된 플래시 연구로 버퍼를 이용하여 플래시 메모리에서 가장 큰 문제점인 플래시 메모리 셀에 쓰기 연산을 줄이므로 전체 성능 향상을 이루고자한 연구였다. 반면, Pack[11]의 연구와 버퍼를 이용한 그림자 기법[12]의 경우 플래시 메모리에서 명령어에 대한 관련 연구로 Pack의 연구는 상대적으로 면적 대 비용

이 높은 큰 용량의 SRAM 사용과 이로 인한 높은 소비전력의 단점을 보이고 있다. 또한 페이지 단위의 저장으로 명령어의 특징인 공간적 지역성에 대하여 효과적인 알고리즘이지만 큰 블록 크기로 인해 버퍼의 블록 엔트리수가 줄어들므로 한번 참조된 명령어를 다시 참조 되는 분기 명령어 대해서 큰 단점을 보이고 있다.

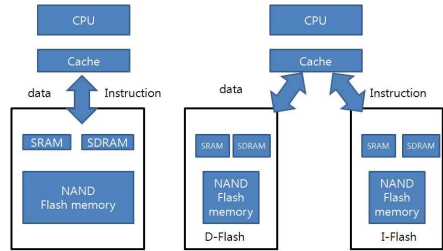
그럼에도 불구하고 기존의 NAND형 플래시 메모리에 대하여 RAM 버퍼의 사용은 성능 향상에 좋은 장점을 보이고 있다. 하지만 데이터와 달리 명령어에 대한 연구는 아직 미비한 수준이며, 더욱이 명령어 및 데이터 각각의 특성 분석과 이들 모두를 고려한 이중 NAND형 플래시 메모리에 대한 연구는 현재 진행되지 못하였다. 이에 오늘날 내장형 시스템의 저장매체로 사용하기 위하여 프로그램의 특성을 모두 반영한 하버드 아키텍처 기반의 플래시 연구는 반드시 필요하며 지속적으로 연구가 진행되어야 할 것이다.

III. 이중 플래시 구조에 대한 버퍼의 효과

기존의 NAND형 플래시 메모리의 핵심 연구 중 하나인 버퍼 시스템을 이용한 고성능 NAND 플래시 연구는 그림 1의 (a)처럼 하나의 단일 NAND 메모리 위에 SRAM/SDRAM 버퍼를 사용하여 패턴을 분석한다든지, 쓰기 버퍼로 사용한다든지, 시간적 지역성을 위한 버퍼와 공간적 지역성을 이용한 버퍼를 두어 고성능 NAND 플래시에 대한 연구를 수행하였으나 하버드 아키텍처처럼 최하위 저장 매체인 플래시 메모리를 명령어와 데이터 형태로 나누어 저장하고 그에 맞는 특성을 가지는 버퍼 구조에 대한 연구는 아직 보고된바가 없다. 그러나 캐시 메모리 연구의 효용성에서 알 수 있듯이 그림 1 (b) 처럼 명령어를 저장하는 플래시 메모리와 그에 맞는 버퍼 구조, 그리고 데이터를 저장하는 플래시 메모리와 데이터 특성을 반영한 버퍼 구조 설계는 보다 정확한 시간적/공간적 지역성을 이용할 수 있고 최소한의 비용으로 버퍼의 효용성을 극대화시킬 수 있다. 또한 플래시 설계시 고려되어야할 중요한 요인은 크게 세 가지로 구분할 수 있다.

① 플래시 메모리는 쓰기 전 지우기 연산이 선행되어야 한다는 것이다. 하드디스크 및 메모리의 경우 파일 업데이트 요청 시 그 파일을 덮어쓰게 되지만 플래시 메모리의 경우 파일 업데이트가 자

유롭지 못하다. 게다가 지우기 연산은 쓰기와 읽기 보다 훨씬 큰 단위인 플래시 블록 단위로 이루어진다. 그러므로 이러한 쓰기 및 지우기 연산을 줄이기



a) 단일 플래시 구조 b) 이중 플래시 구조
a) unified flash b) dual flash

그림 1. 이중 플래시 메모리 시스템

Fig 1. Dual flash memory system

위한 효과적인 방법이 선행되어야 한다.

② 플래시 메모리의 데이터 정보는 특정한 섹터에 대한 쓰기 연산의 정해진 횟수를 초과하게 되면 그 섹터에 대한 정보가 소실되거나 손상될 가능성이 있다. 하드디스크 및 메모리의 경우 특정섹터에 대한 사용횟수에 제한이 없으나 플래시 메모리의 경우 쓰기 횟수에 제한을 가진다. 이러한 제한을 최소화하기 위하여 플래시 메모리 셀의 접근을 줄일 필요가 있다.

③ 플래시 메모리에서 다양한 응용 프로그램을 수행함에 있어서 수행 속도가 매우 빨라야 한다. 특히 무작위 읽기 명령에 대하여 빠른 수행을 보장해야한다. 플래시 종류 중 NOR 플래시 메모리가 NAND 플래시 메모리에 비해 무작위 읽기 명령 수행 속도가 큰 차이를 보이지만 비용적인 측면에서 NAND가 저렴한 특징을 보인다. 그러므로 NAND 플래시 기본구조에 무작위 읽기 명령의 빠른 연산 속도 보장은 NAND 플래시 메모리 구조에서 가장 중요한 요인 중 하나로 고려되어야 할 것이다.

이러한 특성을 효과적으로 반영하는 방법 중의 하나가 바로 버퍼의 적중률을 높여줌으로써 해결할 수 있다. 만약 요청한 데이터 및 명령어의 경우 버퍼에서 적중이 일어나면 빠른 SRAM/SDRAM에서 인출이 일어나므로 수행속도가 매우 빨라진다. 특히 쓰기 동작의 경우 버퍼의 적중은 플래시 메모리 셀의 쓰기 대신 버퍼의 쓰기 동작으로 플래시 메모리의 쓰기 횟수를 줄여줌과 동시에 플래시 메

모리의 가장 큰 오버헤드인 쓰기 시간을 줄여줄 수 있다. 또한 쓰기 동작의 감소로 플래시 메모리의 지우기 동작 역시 감소되는 효과를 얻을 수 있다. 이러한 효과를 얻기 위하여 버퍼의 적중률을 높이는 것은 중요한 핵심사항이다. 버퍼의 적중률을 높이는 선행 연구는 캐쉬 메모리의 연구에서 이미 많이 진행되었다. 명령어의 경우 공간적 지역성이 매우 높으며 이를 위하여 버퍼의 블록사이즈를 키워 줄 필요가 있다 그러나 응용 군에 따른 버퍼 크기와 최적의 블록사이즈를 찾아야하는 할 것이다.

데이터의 경우에도 희생버퍼의 경우 간단한 구조와 메커니즘을 사용하여 고성능을 보장할 수 있는 구조로써 널리 사용되고 있다. 그러나 플래시 메모리의 경우 가장 중요한 특성은 데이터의 읽기 요청에 대한 적중보다 쓰기 동작에 대한 적중과 지우기 연산의 감소가 매우 중요하다. 이는 무작위 읽기 시간에 비해 쓰기 동작에 대한 시간이 무려 수십배 이상 차이가 나며, 지우기 연산의 경우 수백배의 차이가 나기 때문이다. 그러므로 쓰기 및 지우기 동작에 대한 적중률을 높일 수 있는 기존의 버퍼 구조에 대한 분석이 필요하다.

1. NAND형 플래시 메모리에서 명령어 특성

NAND형 플래시 메모리의 읽기/쓰기 동작은 페이지 단위로 이루어지며, 읽기/쓰기 레지스터 접근(순차적 접근) 시간에 비해 플래시 메모리 셀의 접근(무작위 접근) 시간은 순차적인 접근 시간의 약 1000배가 느린 특성을 가진다. 특히 명령어는 데이터와 달리 모두 읽기 동작만 수행됨으로 무작위 접근을 줄이는 것이 가장 효과적인 성능 향상을 이룰 수 있다. 기본적으로 명령어는 프로그램 수행시 순차적인 인출로 공간적 지역성이 강한 특성을 가지고 있다. 공간적 지역성의 특성은 큰 블록 크기를 가질수록 효과적으로 이용할 수 있다. 기존의 NAND형 플래시 메모리는 페이지 단위의 읽기 동작과 페이지 크기의 레지스터를 가지므로 공간적 지역성을 효과적으로 사용할 수 있다. 그러나 페이지 크기의 읽기/쓰기 레지스터는 순차적인 인출 명령어에서 공간적 지역성에 효과적이지만 단일 읽기/쓰기 레지스터로 인한 하나의 레지스터 바운드를 넘어서는 분기 명령어 에서는 플래시 메모리 셀에 다시 접근해야 하는 단점을 가진다.

그림 2은 기존의 NAND 플래시 메모리와 다양한 버퍼를 명령어 플래시에 적용한 경우 플래시 메모리의 평균 플래시 접근 시간(average flash access time)을 나타낸 그림이다. 각 구조의 버퍼

의 크기는 SRAM 4KB이며, 읽기/쓰기 레지스터로부터 버퍼로의 블록 크기는 32B를 사용하였다. 그림에서 보듯이 4KB의 간단한 직접사상 버퍼(direct-mapped buffer)만을 사용했을 경우에도 기존 플래시 메모리의 평균 접근 시간을 82%나 줄이는 효과를 보인다. 또한 기존 플래시 메모리에 비해 2-way 연관버퍼(2-way set associative buffer)를 가지는 플래시 메모리는 88%, 희생버퍼(victim buffer)와 완전연관버퍼(fully associative buffer)를 가지는 플래시 메모리에서는 약 92%의 평균 플래시 접근 시간을 줄였다. 희생버퍼와 완전연관버퍼는 기존의 캐쉬 메모리에서도 간단한 구조로 충돌 접근 실패(conflict miss)를 낮추므로 좋은 성능 향상을 보이는 구조이다. 기존 캐쉬 메모리에서 좋은 성능을 보이는 구조가 그림2에서 보듯이 플래시 메모리에 적용하였을 때 좋은 성능 향상을 가지는 것을 알 수 있다.

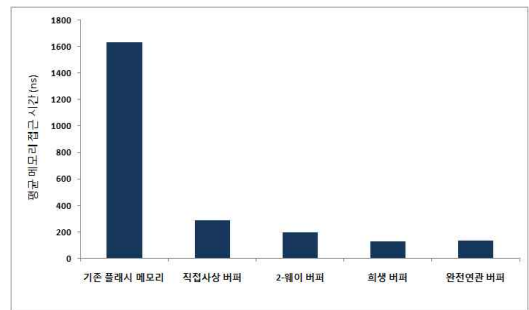


그림 2. 기존 NAND 플래시와 각 버퍼 구조를 가지는 NAND 플래시의 평균 플래시 접근 시간
Fig 2. Average flash access time of the conventional NAND and the NAND with various buffer structure

2. NAND형 플래시 메모리에서 데이터 특성

일반적으로 데이터는 명령어와 달리 읽기/쓰기 동작을 수행한다. 특히 NAND형 플래시 메모리는 기본 페이지 접근으로 한번 데이터가 적힌 페이지는 덮어 쓰기가 불가능하므로 무효 페이지에 데이터를 적어주어야만 한다. 쓰기 동작은 읽기 동작에 비해 매우 느린 수행 시간을 가진다. 만약 쓰기 동작을 위한 NAND형 플래시 메모리의 블록에 더 이상 무효 페이지가 없을 경우 블록 전체를 다른 무효 블록에 적어 주고, 그 블록은 무효 블록으로 처리하기 위한 지우기 동작이 이루어져야한다. 지우기 동작은 NAND형 플래시 메모리에서 가장 큰 오버

헤드를 가지는 동작이다. 따라서 데이터의 경우 명령어와 달리 NAND형 플래시 메모리의 쓰기 동작 및 지우기 동작을 효과적으로 줄여야 높은 성능 향상을 이룰 수 있다.

IV. 실험 결과

1. 명령어 플래시 버퍼 구조에 대한 성능 평가

본 연구에서는 명령어 플래시 메모리를 위한 효과적인 버퍼 구조와 블록 크기를 채택하기 위해 기존의 상용화된 캐쉬 구조를 이용하여 플래시 메모리에 접목하여 그 특성을 파악하였다. 선택된 기존의 캐쉬 구조는 직접사상 버퍼, 2-way 연관 버퍼, 희생버퍼 그리고 완전연관버퍼를 사용하였다. 효과적인 버퍼 구조를 선택하기 위해 사용된 버퍼의 크기는 4KB부터 256KB까지 크기를 증가시켜 실험하였다. 또한 공간적 지역성을 고려하여 버퍼 블록 크기 역시 32B부터 2048B까지 증가시키면서 실험을 하였다. 기본적으로 NAND 플래시 메모리의 한 페이지 단위는 대용량 추세에 맞추어서 2KB로 가정하였다. 연구에서 사용된 기존 버퍼 구조의 동작은 버퍼 접근 실패시 플래시 메모리 접근이 이루어지는 계층구조로 설계하였다. 기존 구조의 버퍼에서 접근 성공이 일어나면 cpu로 요청한 명령어를 보내게 되며, 만약 플래시 메모리의 읽기/쓰기 레지스터나 플래시 셀에서의 접근 성공이 발생한 경우 요청된 명령어가 포함된 블록 크기만큼 플래시 메모리 버퍼로 이동하게 된다.

본 연구에서 기존의 버퍼 구조의 최적 버퍼 크기와 블록 크기를 선택하기위한 비교 값으로 접근 실패율과 평균 플래시 메모리 접근 시간을 사용하였다. 최적의 버퍼 구조에서 비용 대 성능 향상의 비로 최적의 버퍼 크기와 블록 크기를 선택한다.

1.1 접근 실패율

기존의 상용화된 캐쉬 구조인 직접사상버퍼, 2-way연관버퍼, 희생버퍼 그리고 완전연관버퍼를 미디어벤치(media benchmark)에서 4KB~256KB 버퍼 크기를 변화시키면서 시뮬레이션을 수행하였으며, 버퍼 블록 크기 역시 32B~1024B 크기까지 다양한 시뮬레이션을 수행하였다. 분석결과를 요약하면 직접사상버퍼와 2-way연관버퍼 경우 64KB의 버퍼 크기 이후 모든 블록 크기와 상관없이 접근 실패율이 평준화를 이루는 것을 알 수 있었다. 완전연관버퍼와 Victim 버퍼에서도 동일한 결과를 보이

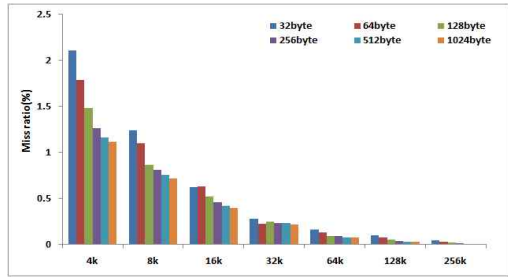


그림 3. 희생버퍼 접근 실패율
Fig 3. Miss ratio of the victim buffer

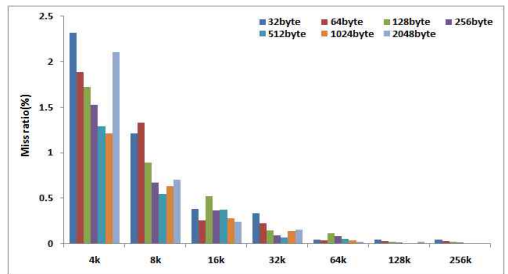


그림 4. 완전연관버퍼 접근 실패율
Fig 4. Miss ratio of the fully associative buffer

고 있다. 본 논문에서는 4가지 버퍼 구조 중 그림 2에서처럼 가장 성능이 좋고 기존의 플래시 버퍼 구조로써 많이 활용되고 있는 희생버퍼와 완전연관버퍼의 결과만을 제시한다.

그림 3 희생버퍼와 그림 4 완전연관버퍼에 대한 접근 실패율의 경우 희생버퍼는 직접사상 버퍼와 완전연관버퍼로 구성되며, 완전연관버퍼의 크기를 1KB로 고정하여 실험하였다(직접사상버퍼 4KB~256KB로 증가). 희생버퍼의 경우 블록 크기와 버퍼 크기가 클수록 좋은 성능 향상을 보이고 있으며, 비용 대 성능적인 측면에서 32KB-64B 버퍼가 최적으로 판단된다. 그림 4의 완전연관버퍼는 4KB의 버퍼 크기에서는 기존의 다른 버퍼 구조와 비슷한 접근 실패율을 보이고 있다. 하지만 8KB의 버퍼 크기에서는 512B의 블록 크기에서 가장 좋은 성능 향상을 보이며, 특히 16KB의 버퍼 크기에서는 64B의 블록 크기에서 가장 좋은 성능 향상을 보이고 있다. 이는 비용 대 성능적인 측면에서 16KB-64B 버퍼가 최적으로 판단된다.

결론적으로 명령어 플래시에서 버퍼를 사용할 경우 일반적으로 64KB이상의 버퍼 크기에서는 블록 크기에 관계없이 일정한 접근 실패율을 보이고

있으며, 직접사상버퍼는 64KB, 2-way연관버퍼와 희생 버퍼는 32KB, 그리고 완전연관버퍼의 경우 16KB의 버퍼 크기면 비용 대 성능 면에서 가장 효율성이 높은 것으로 판단된다.

1.2 평균 플래시 접근 시간

시스템의 성능 평가에서 대표적인 평가지표는 접근 실패율과 함께 평균 메모리 접근 시간을 사용한다. 평균 메모리 접근 시간은 상위계층의 접근 실패 후 하위계층 접근까지의 시간을 나타내는 지표로 전체 시스템 성능평가에 가장 많이 이용된다. 시뮬레이션을 위한 조건은 접근 실패율과 동일하며 결과는 희생버퍼와 완전연관버퍼만을 보인다.

그림 5와 그림 6은 희생버퍼와 완전연관버퍼의 평균 플래시 메모리 접근 시간을 나타낸 그림이다. 여기서 주목할 것은 작은 버퍼 크기(4KB~16KB)의 경우 접근 실패율과는 반대로 32B의 블록 크기에서 가장 좋은 성능 향상을 보이고 있으며, 오히려 공간적 지역성이 강한 1024B 또는 2048B의 블록 크기에서 가장 나쁜 성능을 보이고 있다. 이러한 특성은 기존의 CPU내의 캐쉬의 특성과 많이 틀리다. 캐쉬의 경우 미디어벤치마크의 경우 공간적 지역성이 크기 때문에 평균메모리 접근 시간이 그림 3~4의 접근실패율처럼 블록크기가 증가할수록 일정하게 성능향상을 보였다. 그러나 명령어 플래시에 버퍼의 평균메모리 접근시간은 이와 상반된 특성을 보이고 있다.

그림 5~6에서 보듯이 두 버퍼 모두 64KB이상의 버퍼 크기에서는 버퍼 크기와 블록 크기에 상관없이 일정한 평균 메모리 접근 시간을 보이고 있다. 구조적인 특성에서는 평균 메모리 접근 시간 역시 접근 실패율과 동일하게 완전연관버퍼가 다른 버퍼 구조보다 좋은 성능 향상을 보이고 있으며, 또한 비용 대 성능 향상 면에서 접근 실패율과 동일하게 직접사상버퍼는 64KB, 2-way연관버퍼와 희생버퍼는 32KB, 그리고 완전연관버퍼의 경우 16KB의 버퍼 크기면 비용 대 성능 면에서 가장 효율성이 높은 것으로 판단된다. 그러나 블록 크기에서는 작은 블록 크기인 32B 또는 64KB가 비용 대 성능면에서 가장 좋은 것으로 판단된다.

분석결과 평균 메모리 접근 시간과 접근 실패율에서 비용 대 성능 향상의 최적의 버퍼 크기는 직접사상버퍼는 64KB, 2-way 연관버퍼와 희생버퍼는 32KB, 그리고 완전연관버퍼의 경우 16KB의 버퍼 크기에서 모두 가장 좋은 성능 향상을 보이고 있다. 하지만 완전연관버퍼의 경우 그림 4에서 접근

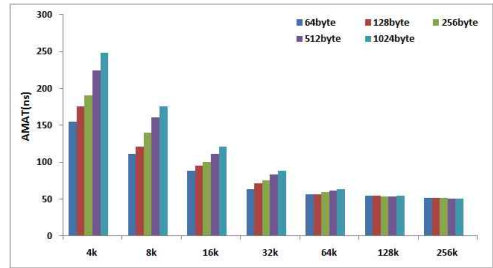


그림 5. 희생버퍼 평균 메모리 접근 시간
Fig 5. Average memory access time of the victim buffer

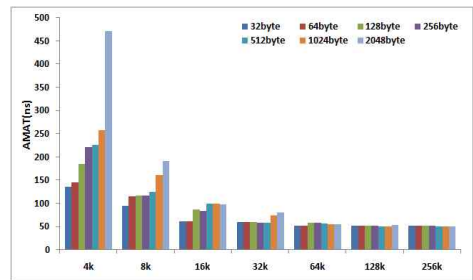


그림 6. 완전연관버퍼 평균 메모리 접근 시간
Fig 6. Average memory access time of the fully associative buffer

실패율은 1024B의 블록 크기에서 좋은 성능개선 효과가 있는 반면, 그림 6의 평균 메모리 접근 시간은 32B의 블록 크기에서 가장 좋은 성능개선을 보이고 있다. 이러한 결과는 접근 실패율은 단지 완전연관버퍼에서의 접근 실패율을 나타내는 반면, 평균 메모리 접근 시간은 전체 시스템의 성능평가를 나타내기 때문이다. 즉 버퍼의 접근 실패율이 높다고 해도 요청된 명령어의 페이지가 읽기/쓰기 레지스터에서 적중이 높기 때문이며, 새로운 플래시 셀의 접근이 아닌 레지스터 접근에 의한 결과이다. 이러한 결과를 바탕으로 소비전력이 중요한 응용 군에서는 접근실패율이 적은 블록 크기의 선택이 플래시 모듈 접근을 줄이는 효과를 얻을 수 있으며, 성능이 중요한 응용 군에서는 평균 접근 시간이 적은 블록을 선택하는 것이 유리하다고 할 수 있다.

2. 데이터 플래시 버퍼 구조에 대한 성능 평가
본 연구에서 NAND형 데이터 플래시 메모리에서 데이터 특성을 파악하기 위해 명령어 버퍼 구조와 동일한 방법으로 시뮬레이션을 수행하였다.

2.1 접근 실패율

그림 7~8은 미디어 벤치마크에 대한 희생버퍼와 완전연관버퍼의 평균 버퍼 접근 실패율을 나타낸 그림이다. 명령어와 달리 데이터의 경우 직접사상버퍼, 2-way 연관버퍼, 4-way 연관버퍼의 경우 작은 버퍼 크기에서 버퍼 접근 실패율이 매우 높았으며, 특히 적은 블록 크기에서 높은 접근 실패율을 보였다. 이에 공간적 지역성과 시간적 지역성을 모두 가지는 특성을 고려하여 각 버퍼들은 8KB~64KB의 버퍼 크기와 128B~2048B의 블록 크기까지의 시뮬레이션을 결과를 보인다.

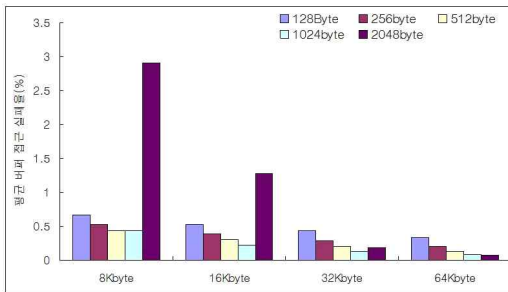


그림 7. 희생버퍼 평균 접근 실패율
Fig 7. Miss ratio of the victim buffer

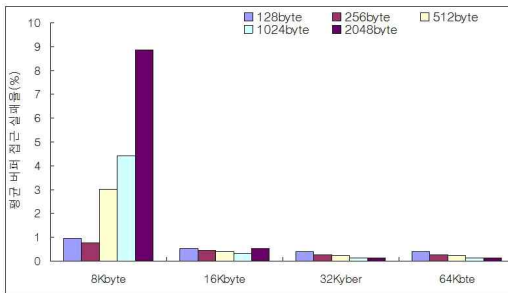


그림 8. 완전연관버퍼 평균 접근 실패율
Fig 8. Miss ratio of the fully associative buffer

희생버퍼 경우 8KB의 버퍼 크기에서 블록 크기가 증가할 수로 1024B를 제외하고 큰 변화가 없다. 그러나 완전연관버퍼의 경우 블록 크기가 클수록 접근 실패율이 매우 높아짐을 알 수 있다. 16KB의 버퍼 크기 이후에는 두 구조 모두 2048B를 제외하고는 블록 크기가 클수록 버퍼 접근 실패율은 좋은 성능향상을 보인다. 이는 버퍼의 크기가 증가할수록 공간적 지역성과 시간적 지역성을 효과적으로 사용

하기 위한 블록 크기와 충분한 엔트리를 가질 수 있기 때문이다. 그러므로 데이터의 경우 버퍼의 사용은 최소 16KB 이상이 되어야 블록 크기와 상관 없이 유사한 성능을 보임을 알 수 있다. 결론적으로 비록 버퍼의 구조에 따라 다소 약간의 차이는 보이지만, 16KB-1024B의 버퍼에서 가장 효과적인 비용 대 성능 비율을 보인다고 판단된다.

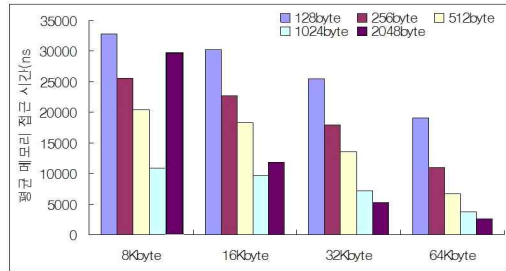


그림 9. 희생버퍼 평균 메모리 접근 시간
Fig 9. Average memory access time of the victim buffer

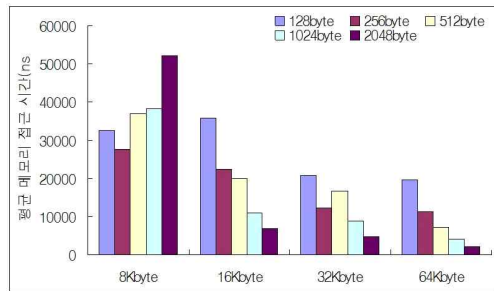


그림 10. 완전연관버퍼 평균 메모리 접근 시간
Fig 10. Average memory access time of the fully associative buffer

2.2 평균 플래시 접근 시간

그림 9~10는 평균 플래시 메모리 접근 시간을 나타낸 그림이다. 그림 9의 희생버퍼와 그림 10의 완전연관버퍼의 경우, 버퍼 접근 실패율과 평균 메모리 접근 시간의 그래프는 비슷한 모습을 하고 있다. 두 구조 모두 16KB의 경우, 높은 접근 실패율을 보이는 2048B의 블록 크기가 평균 메모리 접근 시간에서는 좋은 성능을 보이고 있다. 이는 NAND형 플래시 메모리의 한 페이지 크기인 2048B의 블록 크기가 128B 혹은 256B의 블록 크기보다 쓰기 및 지우기 동작을 많이 줄일 수 있기 때문이다. 따라서 NAND데이터 플래시 메모리에서 성능 향상을

위해서는 낮은 접근 실패율을 가지는 구조도 중요하지만 NAND형 플래시 메모리의 쓰기 및 지우기 동작을 효과적으로 줄여야 전체적으로 성능향상을 높일 수 있다.

V. 결 론

기존의 NAND 플래시 구조처럼 단일 구조에서 명령어와 데이터를 모두 저장할 경우 각각의 고유의 특성을 효과적으로 이용할 수 없다. 이에 본 연구에서는 각각의 특성을 분석하고 이를 효과적으로 반영할 수 있는 플래시 버퍼 구조에 대한 블록 크기와 버퍼 크기에 대하여 다양한 시뮬레이션을 통하여 수행하였다. 결론적으로 미디어 응용군을 위한 플래시 저장 기기의 경우 명령어의 경우 16KB-64B, 데이터의 경우 16KB-2048B의 완전 연관 버퍼를 이용할 경우 가장 좋은 결과를 얻을 수 있으며, 총 32KB 이상의 버퍼 사용은 불필요한 비용 증가를 초래한다고 할 수 있다.

참고문헌

- [1] C. H. Wu, T. W. Kuo, and C. L. Yang, "A Space-Efficient Caching Mechanism for Flash-Memory Address Translation." In Proc. ISORC'06, April 2006.
- [2] Samsung Elec., "NAND Flash Memory & SmartMedia Data Book," 2004.
- [3] <http://www.samsung.com/Products/Semiconductor/index.htm>
- [4] Seunghwan Hyun, Sehwan Lee, Sungyong Ahn, and Kern Koh, "Improving the Demand Paging Performance with NAND-type Flash Memory," In Proc. of the International Conference on Computational Sciences and Its Applications, pp. 157-163, July 2008.
- [5] 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.
- [6] John L. Hennessy and David A. Patterson, "Computer Architecture: A Quantitative Approach (4/E)," Morgan Kaufmann 2006.
- [7] 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
- [8] 정보성, 이정훈, "하드디스크 대용을 위한 공간적 스마트 버퍼 플래시 메모리 시스템," 한국컴퓨터정보학회 논문지, 제 14권, 제 3호, 41-50쪽, 2009년, 3월.
- [9] W. Huang, C. Chen, C. Chen, and C. Chen, "Energy-Efficient Buffer Architecture of Flash Memory," In Proc. of the Multimedia and Ubiquitous Engineering, pp.543-546, April. 2008.
- [10] 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, April, 2006
- [11] 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.
- [12] 이영호, 임성수, "플래시 메모리를 사용하는 demand paging 환경에서의 태스크 최악 응답 시간 분석", 한국컴퓨터정보학회 논문지, 제 11권, 제 6호, 113-123쪽, 2006년, 12월.

저 자 소 개

정보성



2006년 경상대학교 제어계측 공학과 학사.

2008년 경상대학교 제어계측 공학과 석사.

현재, 경상대학교 제어계측공학과 박사과정

관심분야:마이크로프로세서, 캐쉬 및 플래시메모리

Email:blueking80@gnu.ac.kr

이정훈



1999년 성균관대학교 제어계측공학과 학사.

2001년 연세대학교 컴퓨터과 학과 석사.

2004년 연세대학교 컴퓨터과 학과 박사.

현재, 국립경상대학교 ERI, 제어계측공학과 부교수

관심분야: 임베디드 하드웨어, SOC 시스템.

Email: leejh@gsnu.ac.kr