

논문 2007-44SD-3-7

지연 이중 버퍼링: OneNAND 플래시를 이용한 페이지 반입 비용 절감 기법

(Delayed Dual Buffering: Reducing Page Fault Latency in Demand
Paging for OneNAND Flash Memory)

주 용 수*, 박 재 현**, 정 성 우***, 정 의 영****, 장 래 혁*

(Yongsoo Joo, Jaehyun Park, Sung Woo Chung, Eui-Young Chung, and Naehyuck Chang)

요 약

NAND와 NOR 플래시의 장점을 결합한 OneNAND 플래시가 출시되면서 기존의 NAND 플래시를 빠르게 대체하게 되었다. 하지만 기존의 NAND 플래시 기반 요구 페이징 시스템에서는 OneNAND 플래시의 기능들이 제대로 활용되지 않았다. 본 연구에서는 OneNAND 플래시의 임의 접근 기능과 이중 페이지 버퍼를 활용하는 새로운 OneNAND 플래시 기반 요구 페이징 기법인 지연 이중 버퍼링 기법을 제안하였다. 이 기법은 요구된 페이지를 페이지 버퍼로부터 주기억장치로 이동하는 데 걸리는 시간을 효과적으로 절감함으로써 페이지 반입 비용을 절감하였다. 실험 결과, 본 연구에서 제안한 기법은 평균 28.5%의 수행 시간 절감 효과와 4.4%의 페이징 시스템 에너지 절감 효과를 보였다.

Abstract

OneNAND flash combines the advantages of NAND and NOR flash, and has become an alternative to the former. But the advanced features of OneNAND flash are not utilized effectively in demand paging systems designed for NAND flash. We propose delayed dual buffering, a demand paging system which fully exploits the random-access I/O interface and dual page buffers of OneNAND flash demand paging system. It effectively reduces the time of page transfer from the OneNAND page buffer to the main memory. On average, it achieves and 28.5% reduction in execution time and 4.4% reduction in paging system energy consumption.

Keywords: OneNAND, Flash, Demand paging, eExecute-In-Place

I. 서 론

내장형 시스템에서는 ROM과 같은 비휘발성 저장장치에 프로그램 코드를 저장한다. NOR 플래시는 ROM과 유사한 구조를 가지고 있어 워드 단위의 개별적인

접근이 가능하며 임의 접근(random access)에 적합한 입출력 인터페이스를 제공한다. 따라서 플래시에 저장된 프로그램을 직접 접근, 실행하는 것이 가능하며 이러한 기능을 XIP(eExecute-In-Place)라고 한다.

NAND 플래시는 단위 면적 당 집적도가 높아 저비용으로 대용량의 저장 공간을 제공할 수 있어 파일 시스템이나 사용자 데이터를 저장하는 용도에 적합하다. 하지만 NAND 플래시는 XIP를 지원하지 않기 때문에 NAND 플래시에 저장된 프로그램을 실행하기 위해서는 이를 XIP를 지원하는 SRAM 등의 주기억장치로 먼저 복사한 후 주기억장치에서 실행해야 한다. 이를 코드 섀도잉(code shadowing)이라 하는데, 프로그램이 항상 *고속의 주기억장치에서 실행되기 때문에 실행 속

* 정회원, ** 학생회원, 서울대학교 전기컴퓨터공학부 (School of Computer Science and Engineering, Seoul National University)

*** 정회원, 고려대학교 컴퓨터통신공학부 (Division of Computer and Communication Engineering, Korea University)

**** 정회원, 연세대학교 전기전자공학부 (School of Electrical & Electronic Engineering, Yonsei University)

접수일자: 2006년10월13일, 수정완료일: 2007년2월7일

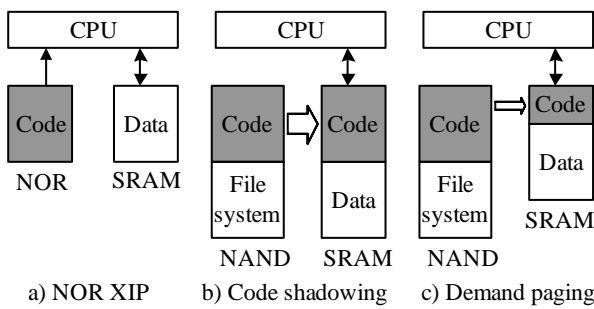


그림 1. 플래시 메모리 시스템 구조

Fig. 1. Flash memory system architectures.

도가 빠르지만 사용 빈도가 낮은 프로그램도 항상 주기억장치에 상주하기 때문에 주기억장치의 용량이 낭비되며 시스템 시동 시에 NAND 플래시에 저장된 모든 프로그램을 주기억장치로 복사해야 하므로 지연 시간이 발생하는 단점이 있다.

초기의 내장형 시스템에서는 내장되는 프로그램의 크기가 그리 크지 않아 구현이 간단하고 별도의 메모리 관리 시스템이 필요 없는 NOR 플래시 기반의 XIP 시스템(그림 1.a)이 주로 사용되었다. 이후 내장형 시스템에 저장되는 프로그램이나 사용자 데이터의 용량이 증가하고 NAND 플래시의 용량 대비 비용이 저렴해짐에 따라 내장형 시스템에도 NAND 플래시가 기본적으로 포함되면서 NAND 플래시 기반의 코드 쉐도잉 시스템(그림 1.b)으로 발전하였다.

최근에는 관련 기술의 급격한 발전과 더불어 내장형 시스템의 적용 분야가 기존의 산업, 의료 분야 등에 국한되지 않고 휴대용 멀티미디어 기기 등의 새로운 응용으로 확대되어 가면서 내장형 시스템의 하드웨어 사양, 실행되는 프로그램의 종류, 제약 조건 등의 제반 사항에도 큰 변화가 일어나게 되었다. 우선, 고품질의 사용자 인터페이스, 디지털 저작권 관리 등의 부가 기능이 포함되면서 내장형 시스템에 포함되는 프로그램의 크기가 급격히 증가하게 되었다. 또한 저전력, 저비용, 경량화 등의 제약 조건을 만족시키기 위해 시스템의 온-칩화(off-chip) 메모리 대신 전력 소모가 적은 온-칩(on-chip) 메모리의 사용이 보편화되었고 프로그램 크기에 비해 주기억장치의 용량 제약은 더욱 심해졌다.

이러한 변화와 더불어 내장형 시스템을 위한 새로운

플래시 메모리 시스템의 필요성이 급증하게 되었고, NAND 플래시 기반의 요구 페이징 시스템(그림 1.c)이 그 대안으로 고려되기 시작하였다. 요구 페이징 기법은 전체 프로그램 중에서 실행을 위해 요청되는 페이지만을 주기억장치에 반입하여 실행하기 때문에 프로그램 크기의 증가 문제와 주기억장치의 용량 제약을 동시에 해결할 수 있다.

NAND 플래시 기반 요구 페이징 기법에 관한 연구 결과도 꾸준히 발표되었는데, NAND 플래시의 페이지 쓰기를 최소화하여 에너지 절감 및 플래시의 수명을 연장시키는 페이지 교체 정책이 제시되었다^[1]. 또한 운영 체제를 사용하지 않는 내장형 시스템에서 컴파일러 기법을 활용한 요구 페이징 구현 기법도 연구되었다^[2].

최근 들어 NOR 플래시와 NAND 플래시의 장점을 결합시킨 새로운 형태의 하이브리드 플래시인 OneNAND 플래시가 발표되었다. OneNAND 플래시는 NOR 인터페이스를 채택하여 NAND 플래시의 입출력 성능을 개선하였고 페이지 버퍼에 대한 XIP 지원, 이중 페이지 버퍼 등의 다양한 기능이 추가되었다. 하지만 기존의 NAND 플래시 기반 요구 페이징 시스템에서는 OneNAND 플래시의 추가 기능이 고려되지 않았다. 요구 페이징 시스템을 수정하지 않고 NAND 플래시를 단순히 OneNAND 플래시로 교체하는 경우에는 페이지 버퍼의 XIP와 같은 기능들이 전혀 활용되지 않으며 OneNAND 플래시의 동기(synchronous) NOR 인터페이스로 인해 비동기(asynchronous) 인터페이스를 가진 NAND 플래시보다 페이지 버퍼 읽기 속도가 빨라지는 효과만을 이용하고 있다.

본 논문에서는 NAND 플래시 기반 요구 페이징 시스템을 OneNAND 플래시에 적합하도록 개선하여 추가의 성능 향상을 얻을 수 있는 자연 이중 버퍼 기법을 제안한다. 제안된 기법은 OneNAND 플래시의 XIP 기능과 이중 버퍼를 활용하여 요구 페이징 시스템의 페이지 반입 시간을 획기적으로 절감할 수 있도록 고안되었다. 모의실험 결과 본 논문에서 제안한 기법은 전체 시스템 수행 시간을 평균 28.6%, 페이징 시스템 에너지를 평균 4.5% 절감할 수 있음을 확인하였다.

II. 본 론

1. OneNAND 플래시

OneNAND 플래시는 NAND 플래시 코어에 NOR 인터페이스와 두 개의 SRAM 페이지 버퍼, 하나의 부팅

* 100MHz로 동작하는 SRAM과 SDRAM의 초기 접근 지연 시간은 각각 10ns, 30ns이며 (SDRAM의 경우 row hit, cas latency=3 가정) Intel Strata NOR 플래시는 동작 속도와 상관없이 85ns이다.

용 페이지 버퍼를 결합시킨 하이브리드 플래시 메모리이다. 그림 2는 NAND 플래시와 OneNAND 플래시의 내부 구조를 보여준다. OneNAND 플래시는 NAND 플래시와 비교하여 다음과 같은 장점을 가지고 있다.

- **부트램(bootRAM) 제공:** 시스템 시동 시에 부트-로더 프로그램 실행을 위한 별도의 마스크-롬(mask-ROM)이나 NOR 플래시가 필요하지 않다.
- **읽기 속도:** 동기(synchronous) NOR 인터페이스로 인해 기존의 비동기(asynchronous) NAND 인터페이스보다 읽기 속도가 4배 이상 빠르다.
- **XIP 지원:** SRAM 페이지 버퍼와 NOR 인터페이스가 적용되어 페이지 버퍼의 내용에 대해 임의접근(random access)이 가능하다.
- **이중 버퍼 지원:** 두 개의 페이지 버퍼를 내장하고 있어 각각의 페이지 버퍼가 독립적으로 동작 가능하다.

기존의 NAND 플래시는 하나의 페이지 버퍼만을 장착하고 있어 외부에서 페이지 버퍼의 내용을 읽는 중에 새로운 페이지를 페이지 버퍼로 적재하는 것이 불가능

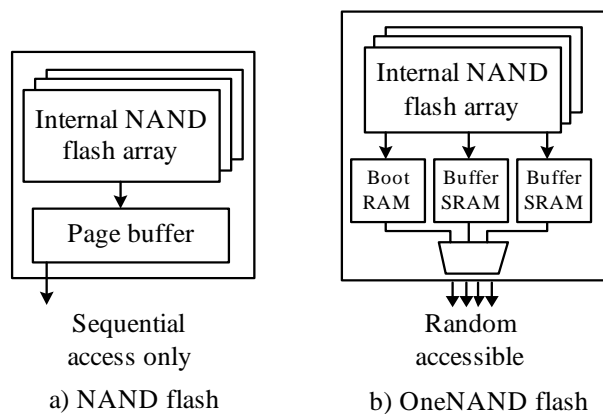


그림 2. NAND 플래시와 OneNAND 플래시의 내부 구조
Fig. 2. Internal structures of NAND flash and OneNAND flash.

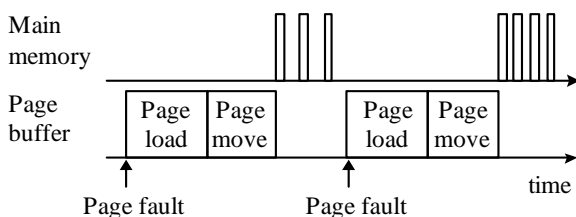


그림 3. NAND 플래시 기반의 요구 페이징 시스템
Fig. 3. NAND flash based demand paging system.

하다. 반면에 OneNAND 플래시는 두 개의 SRAM 페이지 버퍼를 장착하고 있고 각각의 페이지 버퍼가 독립적으로 동작할 수 있으므로 외부에서 하나의 페이지 버퍼를 접근하는 동안 다른 하나의 페이지 버퍼에 새로운 페이지를 적재하는 것이 가능하다.

2. NAND 플래시 기반 요구 페이징 시스템

NAND 플래시 기반의 요구 페이징 시스템(그림 3)에서는 페이지 부재 결함이 발생할 경우 다음과 같은 절차를 거쳐 페이지 반입이 이루어진다.

- **페이지 적재(Page load):** 요구된 페이지는 NAND 플래시 내부의 제어 회로에 의해 NAND 어레이로부터 NAND 플래시의 페이지 버퍼로 옮겨진다.
- **페이지 이동(Page move):** NAND 플래시는 페이지 버퍼에 대한 임의접근을 지원하지 않기 때문에 페이지 버퍼에 올라온 페이지는 CPU의 접근을 위해 곧바로 시스템의 주기억장치로 이동된다.
- **페이지 접근(Page access):** 주기억장치로 이동이 완료된 페이지는 페이지 테이블에 등록되고 CPU가 접근할 수 있게 된다.

하지만 OneNAND 페이지 버퍼는 페이지 이동 시에만 순차적으로 접근되며 따라서 OneNAND 페이지 버퍼의 XIP 기능은 사용되지 않는다. 또한 OneNAND 플래시에는 독립적으로 동작할 수 있는 두 개의 페이지 버퍼가 있지만 그림 3의 요구 페이징 시스템에서는 하나의 버퍼만 활용되고 나머지 버퍼는 유휴 상태에 머물러 있게 된다.

NAND 플래시 기반의 요구 페이징 시스템에서 이중 버퍼가 활용되지 않는 이유는 요구 페이징 시스템의 다음과 같은 특성 때문이다. 우선, 페이지의 반입 과정은 페이지 적재와 페이지 이동의 두 단계로 구성되는데 각각의 절차는 반드시 순차적으로 처리되어야 하기 때문에 이중 버퍼로 동시에 처리될 수 없다. 또 다른 방안으로는 인접한 페이지 부재 결함에 대해서 선행 페이지에 대한 페이지 이동과 후행 페이지에 대한 페이지 적재에 대해 이중 버퍼링을 적용하는 방안을 고려할 수 있다. 즉, 선행 페이지가 이동을 시작할 때 동시에 후행 페이지를 적재하는 방안인데, 요구 페이징 시스템에서는 다음 페이지 부재 결함이 발생되기 전에는 미리 적재되어야 할 페이지가 어떤 페이지인지 알 수 있는 방법이 없다. 하지만 다음 페이지 부재 결함이 발생되려면 선행 페이지의 이동이 완료되어야 하기 때문에 후행 페이지

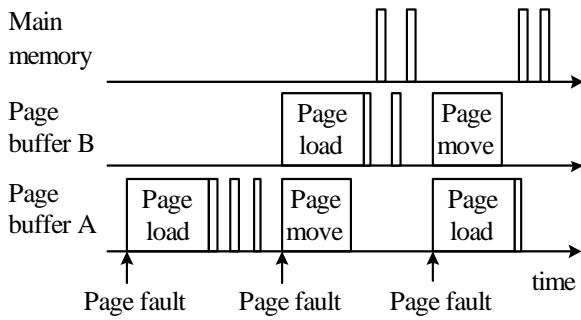


그림 4. OneNAND 플래시 기반의 요구 페이징 시스템
Fig. 4. OneNAND flash based demand paging system.

적재는 선행 페이지 이동과 동시에 발생할 수 없다. 따라서 이중 버퍼링의 적용이 역시 불가능하다.

3. 지연 이중 버퍼링 기법

위에서 살펴본 바와 같이 NAND 플래시 기반의 요구 페이징 시스템을 수정하지 않고 단순히 NAND 플래시를 OneNAND 플래시로 교체하는 경우에는 OneNAND 플래시의 기능을 충분히 활용하지 못하는 것을 알 수 있다. 본 연구에서 제안하는 지연 이중 버퍼링 기법에서는 OneNAND의 XIP 기능과 이중 버퍼를 활용하여 추가의 성능 개선을 얻기 위해 그림 4와 같은 새로운 요구 페이징 시스템을 고안하였다. 지연 이중 버퍼링 기법에서는 페이지 부재 결함이 발생했을 경우 다음과 같은 절차로 페이지 반입이 이루어진다.

- **페이지 적재:** 요구된 페이지는 NAND 어레이로부터 페이지 버퍼로 옮겨지며 이는 기존의 요구 페이징 시스템과 동일하다.
- **페이지 버퍼를 통한 페이지 접근:** 페이지 적재가 완료되면 CPU가 바로 동작을 재개하며 OneNAND 버퍼의 XIP 기능을 이용하여 OneNAND 플래시의 버퍼로부터 직접 페이지를 읽는다. 해당 페이지는 다음 페이지 부재 결함이 발생할 때까지 계속 OneNAND 페이지 버퍼에 위치하면서 XIP로 접근된다.
- **이중 버퍼링을 통한 페이지 이동:** 다음 페이지 부재 결함이 발생되면 이중 버퍼를 이용하여 새로운 페이지의 적재와 기존 페이지의 이동이 동시에 시작된다. 이 때 새로 적재되는 페이지는 비어 있는 다른 페이지 버퍼를 사용하여 기존 페이지와의 충돌을 피한다.
- **주기억장치를 통한 페이지 접근:** 페이지 이동이 완료된 페이지는 OneNAND 페이지 버퍼에서 제거되며 이후의 접근은 주기억장치를 통해 이루어진다.

위에서 설명된 바와 같이 지연 이중 버퍼링 기법은 페이지 적재 후 페이지 이동을 즉시 수행하는 대신에 이를 다음 페이지 부재 결함이 발생할 때까지 지연시키고 그 동안의 페이지 접근 요청은 페이지 버퍼의 XIP 기능을 통해 처리함으로써 페이지 이동을 다음 번 페이지 적재와 동시에 수행하는 이중 버퍼링을 가능하게 한다. 일반적인 플래시 메모리의 경우 페이지 적재 시간보다 페이지 이동 시간이 짧기 때문에 페이지 이동은 페이지 적재가 끝나기 전에 완료되며, 따라서 페이지 반입 시간이 페이지 이동 시간만큼 줄어드는 효과를 기대할 수 있다.

하지만 페이지 버퍼를 통한 페이지 접근 단계에서는 고속의 주기억장치 대신에 상대적으로 성능이 낮은 OneNAND 페이지 버퍼를 통해 페이지를 접근하게 되므로 이 단계에서의 페이지 접근 회수가 많을수록 페이지 접근 시간이 늘어나게 되어 지연 이중 버퍼링 기법의 효과를 감소시키게 된다.

다음 장에서는 모의실험을 통해 지연 이중 버퍼링 기법의 효과를 수행 시간과 에너지 소모 측면에서 검증한다.

III. 성능 평가

1. 성능 측정 지표

OneNAND 기반 요구 페이징 시스템의 수행 성능 및 에너지 소모 분석을 위해 페이징 시스템에서 발생할 수 있는 메모리 접근 단위에 대해 수행 시간과 에너지 소모 파라미터를 표 1에 정의하였다. 또한 전체 프로그램 수행 시간 동안 발생하는 페이징 시스템의 메모리 접근 통계에 대한 파라미터를 표 2에 정의하였다. 본 실험에서는 OneNAND 플래시로 쓰기 동작은 고려하지 않았으며 프로그램의 읽기 전용 영역인 .text와 .rodata에 대해서만 요구 페이징을 적용하고 .data나 .bss와 같은 읽기/쓰기 영역에 대해서는 코드 쉐도잉 기법을 적용하여 항상 주기억장치에서 접근된다고 가정하였다.

표 1과 표 2의 파라미터들로부터 페이징 시스템의 수행시간 t_{paging} 을 다음과 같이 정의할 수 있다.

$$t_{paging} = t_{pl} \cdot N_{pl} + t_{pm} \cdot N_{pm} + t_{rdmain} \cdot N_{rdmain} + t_{rdxip} \cdot N_{rdxip} + t_{wrmain} \cdot N_{wrmain} \quad (1)$$

또한 프로그램의 전체 수행 시간 t_{total} 은 다음과 같다.

표 1. 요구 페이징 시스템의 수행 성능 및 에너지 파라미터

Table 1. Performance and energy parameters of the demand paging system.

파라미터	설명
t_{idle}	CPU의 요청이 연속적으로 캐시에서 적중하여 페이지 접근이 없는 유휴 시간
t_{pl}	페이지 적재 시에 필요한 시간
t_{pm}	페이지 이동 시에 필요한 시간
t_{rdmain}	주기억장치에서 캐시 블록을 읽는 시간
t_{wrmain}	주기억장치에 캐시 블록을 쓰는 시간
t_{rdxip}	OneNAND 페이지 버퍼에서 캐시 블록을 읽는 시간
e_{pl}	페이지 적재 시에 소모되는 에너지
e_{pm}	페이지 이동 시에 소모되는 에너지
e_{rdmain}	주기억장치에서 캐시 블록을 읽을 때 소모되는 에너지
e_{wrmain}	주기억장치에 캐시 블록을 쓰는데 소모되는 에너지
e_{rdxip}	OneNAND 페이지 버퍼에서 캐시 블록을 읽을 때 소모되는 에너지

표 2. 요구 페이징 시스템의 메모리 접근 통계
Table 2. Memory access statistics of the demand paging system.

파라미터	설명
N_{pl}	총 페이지 적재 회수
N_{pm}	총 페이지 이동 회수 (이중 버퍼링 미적용)
N_{pmdual}	총 페이지 이동 회수 (이중 버퍼링 적용)
N_{rdmain}	주기억장치에서 읽은 캐시 블록의 총 개수
N_{wrmain}	OneNAND 페이지 버퍼에서 읽은 캐시 블록의 총 개수
N_{rdxip}	주기억장치에 쓴 캐시 블록의 총 개수

$$t_{total} = t_{paging} + t_{idle} \quad (2)$$

마찬가지로 페이징 시스템의 전체 소모 에너지 e_{paging} 은 다음과 같이 정의할 수 있다.

$$e_{paging} = e_{pl} \cdot N_{pl} + e_{pm} \cdot (N_{pm} + N_{pmdual}) + e_{rdmain} \cdot N_{rdmain} + e_{rdxip} \cdot N_{rdxip} + e_{wrmain} \cdot N_{wrmain} \quad (3)$$

2. 실험 환경

본 연구에서는 제안된 기법의 효과 검증을 위해

OneNAND 기반 요구 페이징 시스템에 대한 모의실험 환경을 구축하였다.

모의실험 환경에서 사용된 OneNAND 플래시 메모리는 삼성전자의 KFG5616(16Mx16, 1.8V)이며 1KB의 페이지 크기를 가진다^[3]. 또한 주기억장치로는 삼성전자의 0.18um SRAM 라이브러리 중에서 64KB high-density single port synchronous static RAM을 선택하였다^[4].

선택된 OneNAND 플래시 메모리와 SRAM에 해당하는 요구 페이징 시스템의 수행 성능 및 에너지 파라미터(표 1 참조)는 사이클 별 메모리 전력 소모 측정 도구^[5]를 사용하여 계산하였고 그 결과는 표 3과 같다. OneNAND 플래시는 50MHz, SRAM은 100MHz로 동작한다고 가정하였다.

모의실험의 입력으로 사용된 페이지 접근 순서는 MiBench^[6]에서 선택된 3가지의 프로그램으로부터 추출되었으며 각 프로그램의 이미지 크기는 표 4와 같다. 페이지 접근 순서는 시스템 수준 전력 시뮬레이터^[7]를 사용하여 추출되었다. 페이지 접근 순서 추출을 위한 시스템 수준 전력 시뮬레이터의 시스템 파라미터는 다음과 같이 설정되었다. 우선, CPU 클럭 주파수는 400MHz를, L1 캐시는 명령어 캐시와 데이터 캐시에 대해 각각 4KB 4-way set을 적용하였다. 또한 캐시 블록 크기는 8워드로 설정하였다. 시스템 버스 클럭 주파수는 100MHz로 설정하였으며 SRAM과 OneNAND 플래시는 각각 100MHz, 50MHz로 설정하였다. 페이지 접근

표 3. 모의실험에 쓰인 수행 성능 및 에너지 파라미터

Table 3. Performance and energy parameters for the simulation.

파라미터	값 (단위: μs)	파라미터	값 (단위: nJ)
t_{pl}	29.33	e_{pl}	1295.48
t_{pm}	12.86	e_{pm}	1056.21
t_{rdmain}	0.04	e_{rdmain}	1.79
t_{rdxip}	0.22	e_{rdxip}	15.24
t_{wrmain}	0.04	e_{wrmain}	2.04

표 4. 모의실험에 사용된 프로그램 정보

Table 4. Benchmark applications for the simulation.

프로그램	전체 크기 (단위: KB)	읽기 전용 영역 크기 (단위: KB)
djpeg	453	158
cjpeg	595	462
fft	32	32

회수는 4종의 프로그램에 대해 공통적으로 100,000회를 추출하였다.

3. 실험 결과

표 5는 jpeg 프로그램에 대해 지연 이중 버퍼링 기법을 적용하기 전과 후의 수행 시간 및 에너지 소모를 비교한 결과를 보여준다.

성능 지표 중에서 페이지 적재 회수, 즉 페이지 부재 결합 회수를 의미하는 N_{pl} 값이 기법 적용 후에 약 11% 감소하였음을 확인할 수 있는데, 이는 이중 지연 버퍼링 기법이 항상 OneNAND의 페이지 버퍼에 한 페이지를 할당하기 때문에 코드를 실행할 수 있는 메모리의 용량이 OneNAND 페이지 버퍼 크기인 1KB만큼 증가하는 효과를 나타내기 때문이다.

페이지 이동과 관련한 수행 성능의 영향을 살펴보면,

표 5. 제안된 기법의 성능 분석 (프로그램: jpeg, 주기억장치 크기: 8KB)

Table 5. Performance of the proposed method (jpeg, 8KB main memory).

성능 지표	기법 적용 전	기법 적용 후
N_{pl}	9557	8526
N_{pm}	9557	0
N_{pmdual}	0	8525
N_{rdmain}	70232	41295
N_{rdxip}	0	28937
N_{wrmain}	29768	29768
수행 성능 (ms)	기법 적용 전	기법 적용 후
t_{idle}	20.17	20.17
$t_{pl} \cdot N_{pl}$	280.31	250.07
$t_{pm} \cdot N_{pm}$	122.90	0
$t_{rdmain} \cdot N_{rdmain}$	2.81	1.65
$t_{rdxip} \cdot N_{rdxip}$	0	6.37
$t_{wrmain} \cdot N_{wrmain}$	1.19	1.19
t_{paging}	407.21	259.28
t_{total}	427.38	279.45
에너지 소모 (mj)	기법 적용 전	기법 적용 후
$e_{pl} \cdot N_{pl}$	12.38	11.05
$e_{pm} \cdot (N_{pm} + N_{pmdual})$	10.10	9.00
$e_{rdmain} \cdot N_{rdmain}$	0.13	0.07
$e_{rdxip} \cdot N_{rdxip}$	0	0.44
$e_{wrmain} \cdot N_{wrmain}$	0.06	0.06
e_{paging}	22.67	20.62

기법 적용 전에는 N_{pmdual} 의 값이 0이고 기법 적용 후에는 N_{pm} 값이 0이 되는데, 이를 통해 지연 이중 버퍼링 기법이 모든 페이지 이동을 이중 버퍼링을 통해 처리하였음을 알 수 있다. 따라서 수행시간에서 $t_{pm} \cdot N_{pm}$ 에 해당하는 페이지 이동 시간이 절감되는 결과를 얻게 된다. 하지만 페이지 이동이 이중 버퍼링에 의해 처리되더라도 이로 인해 에너지 소모가 추가로 줄어들지는 않는다.

한편, 지연 이중 버퍼링 기법을 적용할 경우 전체 페이지 접근 중 일부가 OneNAND의 페이지 버퍼를 통해 이루어지게 된다. 주기억장치보다 OneNAND 페이지 버퍼의 읽기 속도가 느리기 때문에 이는 페이지징 시스템의 수행 성능 및 에너지를 증가시키는 부작용을 가져오게 되는데, 표 5의 실험 결과에서는 이와 관련하여 수행 시간이 5.21ms, 에너지가 0.38mJ 만큼 증가하였다.

정리해보면, 지연 이중 버퍼링 기법의 효과는 다음과 같이 세 가지로 요약된다.

- 코드 실행 메모리의 증가 효과로 인한 수행 시간 및 에너지 감소
- 페이지 이동 시간 절감에 의한 수행 시간 감소
- 페이지 버퍼 직접 접근에 의한 수행 시간 및 에너지 증가

위의 효과는 프로그램의 특성과 요구 페이지징 시스템에 할당된 주기억장치의 용량에 영향을 받는다. 이를 살펴보기 위해 그림 5에서 주기억장치의 용량을 변화시키면서 지연 이중 버퍼링 기법의 효과를 살펴보았다. y축의 값은 기법 적용 전의 값을 100%로 하여 표시하였다.

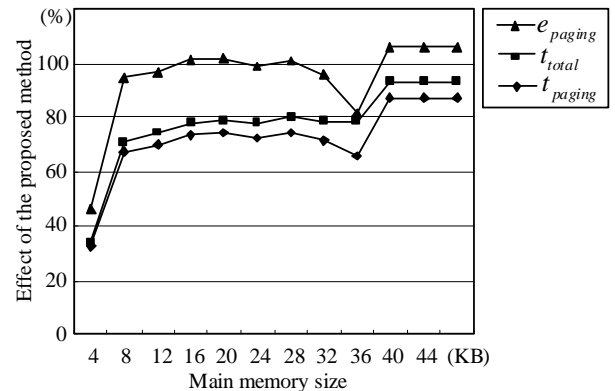


그림 5. 주기억장치 용량에 따른 성능 변화 (jpeg)
Fig. 5. Performance variation according to the main memory size. (jpeg)

그림 5에서 주기억장치의 용량이 40KB 이상인 경우에는 기법의 효과에 거의 변화가 없는 것을 볼 수 있는데 이는 cjpeg 프로그램의 작업집합(working set) 크기가 약 40KB이기 때문이다. 40KB 이상의 주기억장치 용량이 확보되는 경우 cjpeg의 작업집합 페이지들이 주기억장치에 상주하기 때문에 페이지 부재 결함 발생 빈도가 급격히 줄어들며 프로그램 수행시간은 코드 웨딩 기법과 큰 차이를 보이지 않게 된다. 즉, 이 구간에서는 요구 페이징 시스템이 사실상 비활성 상태에 머무르게 되며 지연 이중 버퍼링 기법의 효과 역시 제한적일 수밖에 없다.

주기억장치 용량이 40KB 미만인 경우에는 지연 이중 버퍼링 기법의 효과가 확연히 증가하는 현상을 볼 수 있다. 이 구간에서는 cjpeg 프로그램의 작업집합 페이지들이 동시에 주기억장치에 상주하지 못하기 때문에 빈번한 페이지 부재 결함이 발생하며 페이징 시스템이 활성화된다. 따라서 지연 이중 버퍼링 기법 또한 활성화되어 수행 시간 및 에너지 절감 효과를 보이게 된다.

주기억장치 용량이 4KB와 36KB인 경우 수행 시간 및 에너지 절감 효과가 커지는 현상이 나타난다. 이는 지연 이중 버퍼링 기법의 코드 실행 메모리 증가 효과가 주기억장치 용량에 따라 다르게 나타나기 때문이다. 주기억장치의 용량이 지나치게 작은 4KB 지점에서는 불과 1KB의 코드 메모리 증가 효과만으로도 페이지 부재 결함 발생 회수가 크게 줄어들 수 있다. 또한 주기억장치의 용량이 프로그램의 작업 집합 크기에 근접하지만 작업 집합 전체를 수용하지 못하는 36KB 지점에서 1KB의 코드 메모리 증가 효과가 페이지 부재 결함을 크게 줄이는 결과를 가져온다.

여러 가지 프로그램에 대한 지연 이중 버퍼링 기법의

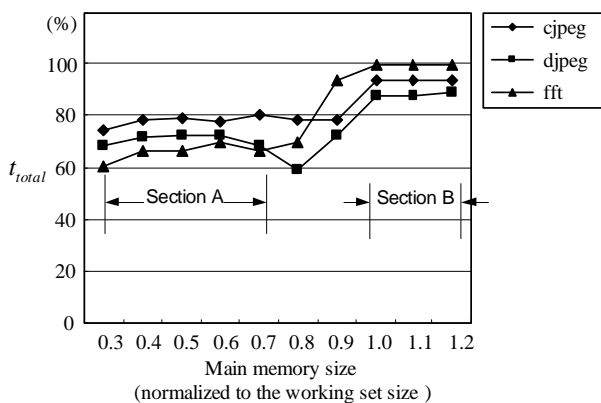


그림 6. 수행 시간 절감 효과 (cjpeg, djpeg, fft)
Fig. 6. Reduction of the total execution time. (cjpeg, djpeg, fft)

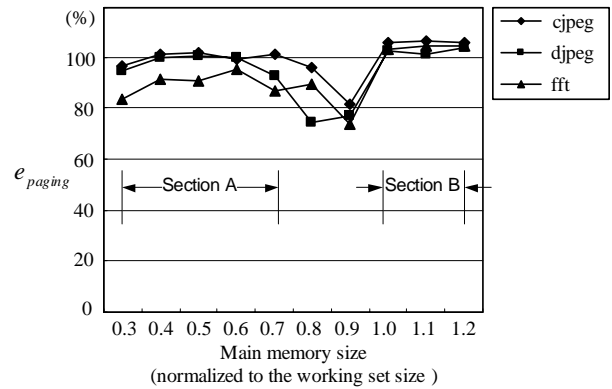


그림 7. 페이징 시스템 에너지 변화 (cjpeg, djpeg, fft)
Fig. 7. Variation of the paging system energy (cjpeg, djpeg, fft).

표 6. 각 프로그램 별 작업집합 크기
Table 6. Working set size of each benchmark application.

프로그램	읽기 전용 영역 크기 (단위: KB)	작업집합 크기 (단위: KB)
djpeg	158	31
cjpeg	462	40
fft	32	18

표 7. 제안된 기법의 평균 성능
Table 7. Overall performance of the proposed method.

구간	작업집합 크기 대비 주기억장치 용량 비율	평균 수행 시간 t_{total} (%)	평균 에너지 소모 e_{paging} (%)
A	0.3~0.7	71.5	95.6
B	1.0~1.2	93.6	104.0

효과를 확인하기 위해 각 프로그램 별 작업집합 크기 (표 6 참조)를 기준으로 주기억장치 용량의 비율을 0.3에서 1.2까지 0.1씩 증가시키면서 전체 시스템 수행 시간과 페이징 시스템 에너지 소모에 대한 기법의 효과를 그림 6과 그림 7에 나타내었다. y축의 값은 기법 적용 전의 값을 100%로 하여 표시하였다.

구간 A는 프로그램의 작업 집합 크기보다 주기억장치 용량이 작아 페이징 시스템이 활성화되는 경우를, 구간 B는 주기억장치 용량이 충분하여 페이징 시스템이 비활성화되는 경우를 대표한다. 지연 이중 버퍼링 기법은 세 가지 프로그램에 대해 공통적으로 구간 A에서 높은 수행 시간 절감 효과를 보이는 것을 알 수 있다. 구간 별 수행 시간 및 페이징 시스템 에너지 소모의 평균치는 표 7과 같다.

실제로 요구 페이징 시스템을 구현할 경우 시스템 수행 성능을 고려한다면 주기억장치의 용량이 프로그램의 작업집합 크기보다 크도록 설정하는 것이 유리하다. 하지만 요구 페이징 기법이 사용되는 환경에서는 이미 주기억장치 용량에 대해 심한 제약을 받고 있을 가능성이 높다고 볼 수 있고, 다중 프로세스 등의 상황을 가정한다면 프로세스 간의 주기적인 문맥 교환으로 인해 페이지 부재 결함이 꾸준히 발생한다고 가정해야 한다. 따라서 실제 요구 페이징 시스템은 대부분 구간 A에서 동작한다고 볼 수 있으며, 본 논문에서 제안한 지연 이중 버퍼링 기법은 이에 대해 평균 28.5%의 수행시간 절감 및 4.4%의 페이징 시스템 에너지 소모 절감 효과를 나타내었다.

IV. 결 론

본 연구에서는 NAND 플래시 기반 요구 페이징 시스템을 OneNAND 플래시에 적합하도록 최적화하는 지연 이중 버퍼링 기법을 제안하고 그 성능을 평가하였다. 제안된 기법은 OneNAND 플래시의 임의 접근 기능과 이중 페이지 버퍼를 활용하여 페이지 반입 시간을 효과적으로 절감할 수 있도록 고안되었다. 모의실험 결과, 지연 이중 버퍼링 기법은 전체 시스템 수행 시간을 평균 28.5% 절감하였으며 페이징 시스템 에너지 소모에 대해서도 평균 4.4%의 절감 효과를 보였다.

요구 페이징 시스템은 페이지 부재 결함 시에 성능 저하가 발생하기 때문에 실시간 응답이 중요한 멀티미디어 코덱 등은 요구 페이징을 적용하지 않고 주기억장치에 고정하여 실행되며, 성능 저하를 감수할 수 있는 부가 프로그램들에 대해 여분의 주기억장치를 활용하여 요구 페이징 시스템을 적용하는 절충안이 사용되고 있다. 따라서 요구 페이징 시스템의 효율이 낮아지더라도 주기억장치를 추가로 할당할 수 없는 경우가 자주 발생하며 이러한 경우 페이지 부재 결함이 지나치게 많이 발생하여 부가 프로그램의 수행 성능이 심각하게 저하된다.

본 연구의 의의는 이러한 상황에서 OneNAND 플래시의 기능을 최대한 활용하여 효과적으로 페이지 반입 비용을 절감하는 기법을 제시함으로써 프로그램의 수행 성능을 개선하거나 요구 페이징 시스템에 더 많은 양의 프로그램 코드를 수용할 수 있는 유연성을 확보하는 데 있다. 특히, 별도의 하드웨어 수정이나 추가 비용이 들지 않으며 페이징 시스템의 소프트웨어적인 수정만으로

기존 내장형 시스템에 적용이 가능하기 때문에 기법의 효과는 더욱 크다고 할 수 있다.

참 고 문 헌

- [1] C. Park, J.-U. Kang, S.-Y. Park, and J.-S. Kim, "Energy-aware demand paging on NAND flash-based embedded storages," in *Proceedings of the 2004 International Symposium on Low Power Electronics and Design (ISLPED'04)*, pp. 338-343, Newport, USA, August 2004.
- [2] C. Park, J. Lim, K. Kwon, J. Lee, and S. L. Min, "Compiler-assisted demand paging for embedded systems with flash memory," in *Proceedings of the 4th ACM International Conference on Embedded Software (EMSOFT'04)*, pp. 114-124, Pisa, Italy, September 2004.
- [3] KFG5616x1Ax16 OneNAND Specification, Samsung Electronics, Co. Ltd., <http://samsungelectronics.com>, December 2005.
- [4] STD130(Rev. 2.1) 0.18um 1.8V CMOS Standard Cell Library for Pure Logic Products, Samsung Electronics, Co. Ltd., <http://www.samsungelectronics.com>, February 2004.
- [5] SECM: SNU Energy Characterizer for Memory Devices, <http://elpl.snu.ac.kr/measurement/sec.htm>, 2005.
- [6] M. Guthaus, J. Ringenberg, D. Ernst, T. Austin, T. Mudge, and R. Brown, "Mibench: a free, commercially representative embedded benchmark suite," in *Proceedings of Fourth IEEE Workshop Workload Characterization (WWC-4)*, pp. 10-22, Austin, USA, December 2001.
- [7] I. Lee, Y. Choi, Y. Cho, Y. Joo, H. Lim, H. G. Lee, H. Shim, and N. Chang, "Web-based energy exploration tool for embedded systems," *IEEE Design and Test of Computers*, vol. 21, no. 6, pp. 572-586, 2004.

저 자 소 개



주 용 수(정회원)
 2000년 서울대학교 컴퓨터공학과
 학사 졸업.
 2002년 서울대학교 전기컴퓨터
 공학부 석사 졸업.
 2002년~현재 서울대학교 전기
 컴퓨터공학부 박사과정.
 <주관심분야 : 내장형 시스템, 저전력 시스템>



박 재 현(학생회원)
 2006년 서울대학교 전기공학부
 학사 졸업.
 2006년~현재 서울대학교 전기
 컴퓨터공학부 석사과정.
 <주관심분야 : 내장형 시스템, 저
 전력 시스템>



정 성 우(정회원)
 1996년 서울대학교 컴퓨터공학과
 학사 졸업.
 1998년 서울대학교 컴퓨터공학과
 석사 졸업.
 2003년~2005년 삼성전자
 책임연구원.
 2005년~2006년 University of Virginia 연구원.
 2006년~현재 고려대학교 컴퓨터통신공학부
 조교수.
 <주관심분야 : 저전력 시스템, 내장형 시스템,
 SoC>



정 의 영(평생회원)
 1988년 고려대학교 전자공학과
 학사 졸업.
 1992년 고려대학교 전자공학과
 석사 졸업.
 2002년 Stanford University
 Electrical Engineering
 박사 졸업.
 1990년~2005년 삼성전자 수석연구원.
 2005년~현재 연세대학교 전기전자공학부
 부교수.
 <주관심분야 : 저전력 시스템, VLSI CAD, 내장
 형 시스템, 컴퓨터구조>



장 래 혁(정회원)
 1989년 서울대학교 제어계측
 공학과 학사 졸업.
 1992년 서울대학교 제어계측
 공학과 석사 졸업.
 1996년 서울대학교 제어계측
 공학과 박사 졸업.
 1997년 미국 University of Michigan 연구원.
 2005년 미국 Arizona State University 방문교수.
 1997년~현재 서울대학교 컴퓨터공학과 부교수.
 <주관심분야 : 저전력 시스템, 고속 디지털 시스
 템, 내장형 시스템>