

논문 2009-46SD-1-2

고속 처리가 가능한 다중처리 Nand 플래시 Controller

(High Performance Nand Flash Controller using Multi- Processing Scheme)

강 신 욱*, 이 동 우**, 정 성 훈**, 이 용 석***

(Shin Wook Kang, Dong Woo Lee, Seong Hun Jeong, and Yong Surk Lee)

요 약

NAND 플래시 메모리를 이용한 카드가 보편화되어 이제는 대량의 멀티미디어 데이터를 모두 저장할 수 있는 수준에 이르렀다. 하지만 NAND 플래시 셀(cell)의 느린 동작으로 인하여 대량의 데이터를 빠르게 전송하기에는 많이 부족한 수준이다. 즉 대량의 멀티미디어 데이터를 NAND 플래시 메모리 카드로 전송할 경우 많은 시간이 걸리는 단점이 있다. 이에 본 논문에서는 데이터 전송률을 높이기 위한 새로운 하드웨어 및 소프트웨어의 구조를 제안한다. 제안하는 구조에서는 기존의 직렬 처리(serial processing) 기법과 다른, 다중 처리(multiprocessing) 기법을 사용하였다. 제안된 구조를 이용하여 VIP(Virtual IP) 환경에서 시뮬레이션 하고 FPGA 보드환경에서 최종 실험하였다. 실험 결과 VIP환경에서는 160MB/s의 다운로드 성능을 볼 수 있었으며, FPGA 보드환경에서는 85.3MB/s의 다운로드 성능을 볼 수 있었다.

Abstract

Lately, NAND flash cards have been used to store massive amounts of multimedia data. However, these nand flash cells itself has a slow operation time and by that, the nand flash cards are not appropriate for high performance massive data transfer. Indeed, most flash card products have a disadvantage in that they require plenty of time to transfer massive amounts of data. Therefore, we propose a new architectural design for the hardware and software of the NAND flash cards by improving their data transfer rate. Our design is based on a multiprocessing which is different from the conventional serial processing method. We simulated our design under the VIP (Virtual IP) environment, and verified our work using FPGA test platforms. As a result, the downloading performances was approximately 160MB/s on VIP and 85.3MB/s on FPGA.

Keywords : NAND Flash, Controller, Pipeline, Multi Channel, Virtual Platform, FTL

I. 서 론

멀티미디어 환경으로의 변화와 함께 사용자가 이용하는 저장 매체도 큰 발전을 거듭하여 왔다. Tape에서 CD, DVD로 발전하여 이제는 HD-DVD, BD(Blue-ray Disk)와 같은 대량의 데이터를 저장하는 매체가 등장하여 사용자는 보다 화려한 이미지와 현장감 있는 음질을

들으며 생활할 수 있게 되었다.

만약 우리가 사용하는 대량의 저장 매체(HD-DVD, BD)를 NAND 플래시의 저장 매체가 대체하고 NAND 플래시의 느린 동작을 극복하여 보다 빠르게 읽고 쓰기 하여 사용한다면 우리의 생활에 또 다른 패러다임의 변화가 올 수 있다. 그림 1은 NAND 플래시 메모리 카드를 위한 사용 시나리오를 나타낸 것이다. 지금까지 비디오 대여점에 들러 새로운 영화를 대여 받으러 갈 필요 없이, 사람의 이동이 많은 곳에서 서버 단말기(kiosk)를 통해 플래시 메모리 카드로 기록하여 받은 다음, 개인 단말기(personal player)나 A/V기기를 통해 플래시 메모리 카드의 멀티미디어 데이터를 사용할 수 있게 된다. 이렇게 받은 멀티미디어 데이터는 일정 시간

* 학생회원, *** 평생회원, 연세대학교 전기전자공학과
(Department of Electrical and Electronic Engineering, Yonsei University)

** 정회원, 삼성전자 DM연구소
(Digital Media R&D Center, Samsung Electronic Co., Ltd.)

접수일자: 2008년7월7일, 수정완료일: 2009년1월5일

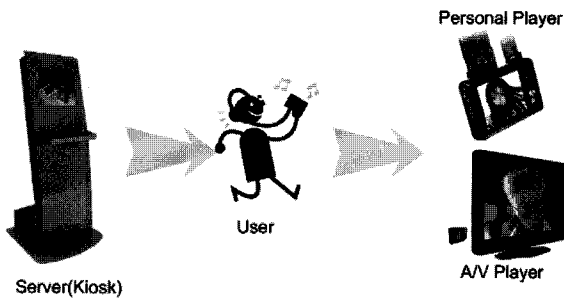


그림 1. Flash card 사용 시나리오
Fig. 1. Flash card user scenario.

후 자동으로 사용할 수 없도록 하여 반납의 불편함도 없앨 수 있다. 배급자(contents provider) 또한 많은 곳에서 새로운 멀티미디어 데이터를 보다 쉽게 보급할 수 있기 때문에 이득이 될 수 있다. 이렇게 기존의 오프라인(off-line)의 배급 방식에서 NAND 플래시 메모리를 이용한 대량 저장 매체가 보급된다면 사용자의 편의성과 함께 배급자의 이익 또한 증가되리라 본다.

위와 같은 시나리오 상에서 HD(High Definition)급 영화를 플래시 메모리 카드로 받을 경우를 생각해 보자. H.264 video codec을 사용한 HD급 영화의 경우 8~20Mbps(Mega bit per second)의 대역(bandwidth)이 필요하다.^[1] 만약 12Mbps, H.264로 encoding된 2시간짜리 HD급 영화의 경우 약 18GB(Giga Bytes)[20Mbps × 60초 × 60분 × 2시간]의 용량을 갖는다. 이를 40MB/s(Mega Bytes per second) 쓰기 성능을 가지는 콤팩트 플래시 카드(CF card: Compact Flash card)^[2]로 저장할 경우 약 450초(7분 30초) 정도 걸린다. 이는 위의 시나리오의 상황을 고려한다면 사용자가 서버 단말기(kiosk)로 HD급 영화를 받을 경우 불편함을 초래할 수 있다.

여기서 인터페이스의 성능은 전송률(transfer rate)일 뿐이며, 실제 NAND 플래시로의 읽고 쓰기 속도는 NAND 플래시 카드의 내부 특성에 따라 다소 느리다. 예를 들어 USB 2.0의 경우 480Mbps^[3], PCIe(PCI express) 2.0은 500MB/s^[4], SATA(Serial ATA) 2.0은 300MB/s^[5]까지의 전송률을 갖고 있으며 이는 인터페이스에서의 전송률이다.

그림 2는 그림 1과 같은 사용 시나리오에서 사용자가 서버를 통해 NAND 플래시 메모리 카드로 데이터를 받을 경우 각 구성요소를 개괄적으로 나타낸 것이다. 서버 단말기(server platform)와 NAND 플래시 메모리 카드(storage platform) 그리고 이를 연결하는 인터페이스 부분으로 구성된다.

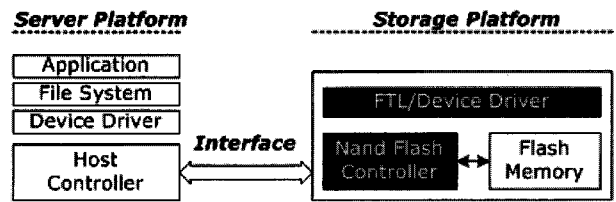


그림 2. 서버와의 연결 시 세부 구성도
Fig. 2. Detail diagram of interfacing with server.

NAND 플래시 메모리 카드의 성능 향상을 위해서는 그림 2에 있는 각 구성 요소의 모든 부분을 고려하여 개선하여야 한다. 하지만 본 논문에서는 NAND 플래시 메모리 카드 중 특히 NAND 플래시 controller 및 FTL(File Translation Layer)에 대한 부분에 초점을 맞춰 진행하였다. 나머지 서버 단말기(server platform) 및 인터페이스는 실험을 위하여 쉽게 구현할 수 있는 방법을 사용하였다.

본문에서는 NAND 플래시 메모리 카드에서 사용하는 기존의 방법 및 구조에 대하여 알아보고, 제안하는 방법에서의 하드웨어 및 소프트웨어에 대하여 설명한다. 실험에서는 그림 2와 같은 구성으로 VIP 시뮬레이션 환경과 FPGA보드 환경을 구성하여 NAND 플래시 메모리 카드(storage platform)의 성능을 실험하였으며, 결론에서는 이 실험 결과를 비교 분석하였다.

II. 본 론

1. 하드웨어 구조

가. NAND 플래시 기본 구성 및 동작,

NAND 플래시 제어를 위한 기존의 방법

NAND 플래시 메모리는 가장 큰 단위인 블록(block)으로 구성되어 있다. 1개의 블록에는 보통 64개의 페이지(page)가 있으며 1개 페이지에는 512 bytes ~ 4K byte의 데이터 저장 공간과 16 bytes ~ 128 bytes의 여분의 영역(spare area)으로 구성되어 있다.

NAND 플래시 메모리를 읽고 쓰기 위한 동작은 다음과 같이 구성된다. 먼저, NAND 플래시 메모리로 읽고 쓰기 위한 명령을 전송하고 다음으로 읽고 쓰기 위한 어드레스를 전송한 후 마지막으로 NAND 플래시 메모리로부터 데이터의 전송이 이뤄진다. NAND 플래시 메모리의 지우기 동작은 위의 읽고 쓰기 동작 중 데이터 전송 부분만 제외하고 같은 구성으로 동작한다.

이 중 가장 많은 시간이 소요되는 쓰기 동작의 경우

를 살펴보면 다음과 같다. 명령과 어드레스 그리고 기록할 데이터를 플래시 메모리로 전송하는 시간을 셋업 시간(setup time)이라고 하며 이 전송된 데이터가 메모리 셀(cell)로 기록되는 시간을 프로그램 시간(program time)이라고 한다. 여기서 명령과 address를 전송하는 시간은 제외하고 플래시 메모리로 데이터를 1byte씩 페이지 레지스터(page register)로 저장하는 셋업 시간(tWC:25ns, 1 페이지가 2K bytes라면 $25ns \times 2048 =$ 약 51.2us)과 이 페이지 레지스터의 내용을 셀로 기록하는 프로그램 시간 (tPROG : 200us(최소))을 합하면, 전체 NAND 플래시 메모리의 사용시간은 상당한 부분을 차지한다(setup time + program time = 약 251.2us이상).^[6] (단, 이때 기타시간을 제외하여 계산함.) 즉 2K bytes 데이터를 기록하는데 최소 251us이상의 시간이 소요된다.

이렇게 느린 NAND 플래시 메모리의 성능을 개선하기 위하여 기존에는 인터리빙(interleaving)을 이용하는 방법, 멀티채널(multi channel)을 이용한 방법 그리고 인터페이스상의 구조나 동작을 개량하는 방법이 사용되었다.

인터리빙 방법에서는 NAND 플래시 셀(cell)을 플레인(plane)을 나누고, 서로 다른 플레인(plane)간 인터리빙(이를 또 다른 표현으로 웨이[way] 또는 뱅크[bank] 또는 어레이[array] 인터리빙이라고도 함)이 일어나게 하거나 멀티 칩(multi chip)을 패키지(package)하여 칩 인터리빙(chip interleaving)을 하는 방법^[7] 또는 멀티다이(multi die)로 다이(die)간 인터리빙하는 방법^[8]을 제공하여 최대 성능을 얻도록 하고 있다.^[9-10] 만약 NAND 플래시로의 쓰기 성능을 최대한 높이기 위하여 프로그램 시간(tPROG)을 보이지 않도록 하고 셋업 시간(tWC가 25ns이고 2K bytes 페이지의 SLC에서 대략 51.2us정도임)만 보이도록 구성된다면 40MB/s (1byte/25ns)의 최대 성능을 얻을 수 있다. 이는 앞서 얘기한 서로 다른 플레인(plane)이나 다이(die) 또는 칩(chip)간 인터리빙 방법을 통해 NAND 플래시를 사용하는데 있어 높은 성능을 얻을 수 있다.

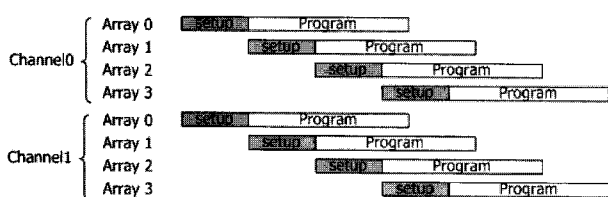


그림 3. 2채널 4웨이 인터리빙 프로그램 동작
Fig. 3. 2-channel 4-way interleaving program.

멀티채널(multi channel)을 이용한 방법은 플래시 메모리를 여러 개 동시에 액세스하여 한 번에 읽고 쓰는 데이터양이 많도록 하는 것이다.^[11] 그림 3은 2채널 4웨이(way) 인터리빙의 방법을 나타낸 것으로 2개의 NAND 플래시 칩을 액세스함과 동시에 웨이(way)간 인터리빙을 통해 높은 성능을 얻도록 하고 있다.^[11] 서버에서 보내어진 데이터는 channel 0와 channel 1의 첫 번째 어레이(array 또는 plane 또는 bank 또는 way)에 동시에 기록된다. 다음으로 보내어진 서버의 데이터는 인터리빙을 통해 각 채널의 두 번째 어레이에 기록되는 순서로 진행된다. 따라서 각 채널의 동시 동작으로 인하여, 서버에서 보내어지는 데이터 크기는 한 채널에서 처리하는 크기(size) × 채널의 개수가 된다. 쓰기를 예로 들면, 한 개의 채널에서 인터리빙을 통해 얻을 수 있는 최대 성능이 40MB/s이므로 이 경우 80MB/s (40MB/s x 2 채널)가 된다. 여기서 채널을 더 늘린다면 40MB/s x N 채널의 성능을 얻을 수 있다. 하지만 만약 서버의 파일시스템에서 사용하는 데이터 블록(FAT의 경우 클러스터)을 페이지 크기와 동일하게 설정한다면 클러스터 갭(cluster gap)문제가 발생할 수 있으며, 그렇지 않을 경우 FTL(File Translation Layer)에서의 연산량(block merge와 같은 FTL의 동작)이 더 많아지는 문제점이 발생할 수 있다.

마지막으로 인터페이스 상에서의 성능향상을 위한 방법이 있다. 데이지 체인(daisy-chain)방식으로 플래시 메모리를 연결한 방법과 DDR SDRAM(Double Data Rate Synchronous DRAM)의 인터페이스를 이용하여 double data rate를 얻도록 구성한 방법^[10]이 있다. 데이지 체인 방식은 해당 플래시 메모리에 직렬(serial)로 명령과 데이터를 전송할 수 있게 되어 빠른 동작이 가능하고 새로운 입출력 핀의 추가 없이 확장도 용이하다.^[12] DDR 인터페이스를 사용한 방법은 서버와의 클럭 속도 향상에 초점이 맞춰진 것으로, double data rate를 사용하여 단위 시간 동안 전송하는 데이터양이 많아질 수 있다. 하지만 double data rate를 사용하는 방법에서는 반응 시간(latency)이 커지게 되어 버스 효율성이 떨어지는 것이 문제점으로 존재한다. 이를 위해 보통 클럭 속도를 아주 빠르게 하고 단위 시간동안 처리하는 데이터양을 많게 하여 버스 사용에서의 비효율을 보상하는 방식으로 사용한다.

나. 제안하는 NAND 플래시 제어 방법

기존 방식에서의 멀티채널(multi channel), 멀티웨이

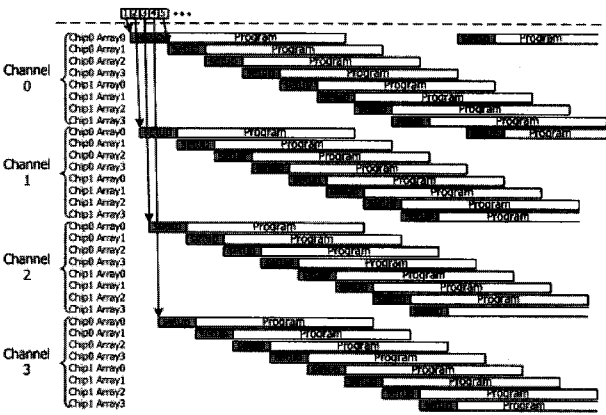


그림 4. 확장된 멀티채널 파이프라인 프로그램 동작
Fig. 4. Enhanced multi-channel pipeline program.

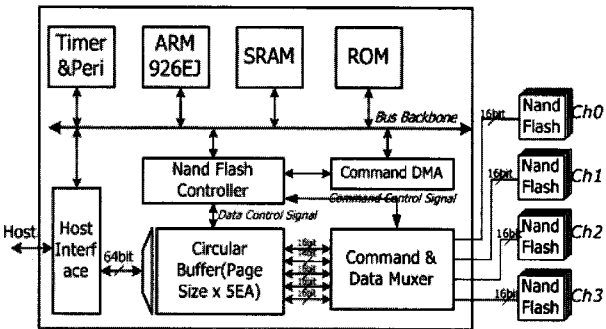


그림 5. Flash card 내부 구성도
Fig. 5. Flash card architecture.

(multi way) 인터리빙 동작(interleaving operation)을 확장하여 채널 간 동작이 동시에 일어나면서 각 채널에서 웨이(또는 플레인, 뱅크, 어레이)간 인터리빙이 일어나도록 하는 방식을 제안한다. 즉 채널 간 다중처리(multi processing)이 가능하도록 한 방식이다.

그림 4에서는 서버에서 보내어진 데이터(페이지 크기의 데이터 또는 섹터[sector:512 bytes] 크기의 데이터)가 각각의 멀티채널로 어떻게 기록되어지를 나타낸 것이다. 서버에서 보내어진 첫 번째 데이터를 channel 0의 첫 번째 어레이에 기록하고 두 번째 데이터를 channel 1의 첫 번째 어레이에 기록한다. 이렇게 동시에 오버랩(overlap)하면서 셋업 시간(setup time)이 계속하여 진행하게 되면, 기존의 멀티채널에서의 액세스 페이지의 크기 문제(멀티채널에서는 채널 개수만큼 액세스하는 페이지 크기가 커지게 된다)를 해결할 수 있고 채널확장도 용이할 수 있게 된다. 각 채널에서는 어레이 인터리빙과 함께 멀티 다이(multi die)인 경우 다이(die) 인터리빙도 행하여 질 수 있으며, 멀티 칩 패키지(multi chip package)인 경우 칩 인터리빙도 행하여 질 수 있다. 또한 각 채널에서 기존의 멀티채널 방식처럼 두 개 또는 여러 개를 동시에 액세스하여 채널 한

개의 성능도 높일 수 있다. 즉 기존의 방식을 접목하여 사용할 수 있을 뿐만 아니라 병렬(parallel)로 진행하므로 전체적인 성능향상을 얻을 수 있게 된다.

그림 5는 제안하는 플래시 카드의 내부 구성도이다. 확장된 멀티채널 파이프라인 동작(enhanced multi-channel pipeline operation)을 지원하기 위해 controller에서는 circular 버퍼(버퍼 크기 = NAND 플래시 페이지 크기 × (채널 개수 + 1))를 이용하여 연속적으로 각 채널에 데이터를 전송할 수 있게 된다. circular 버퍼의 크기가 채널의 수 보다 1개 더 많도록 한 이유는 4개 버퍼를 통해 4개 채널로 동시 읽거나 쓰기 동작을 하면서 나머지 1개의 버퍼를 통해 서버로의 데이터 전송을 하기 위함이다. 또한 FTL(File Translation Layer)에서의 각 채널 동작 명령을 SRAM에 기록 후 command DMA로 읽어 NAND 플래시 controller가 파이프라인 동작(pipeline operation)을 보다 빠르게 진행할 수 있도록 하였다.

2. 소프트웨어 구조

가. FTL(File Translation Layer)의 역할

NAND 플래시 셀 특성으로 인해 기록 동작 시 HDD 처럼 해당 위치에 바로 기록이 되지 않는다. 또한 지우기 동작 시 블록(block)단위로만 지울 수 있기 때문에 이와 관련한 여러 가지 동작이 필요하다. 이런 동작을 지원하는 소프트웨어가 FTL이며 아래와 같은 큰 특징을 갖는다.^[13~14]

- Mapping logical address to physical address
- Garbage collection
- Wear leveling

Address mapping의 여러 방법 중 본 논문에서는 log block mapping^[15]을 사용하였으며 garbage collection은 merge, switch^[15, 17]과 함께 compaction기법을 사용하였다. wear leveling에서는 다수의 블록을 존(zone)으로 묶은 다음 erase counter를 두어 관리하도록 하였고, 여기에 static wear leveling까지 포함하여 동작하도록 하였다.^[16~17]

나. 제안하는 FTL의 구조 및 방법

본 논문에서는 그림 6에서처럼 멀티채널 파이프라인 특성에 따라 멀티채널을 동시 처리하는 병렬구조의 FTL을 사용하였다. 즉 FTL에서도 다중처리(multi processing)를 지원하는 구조를 사용하였다. FTL은 크

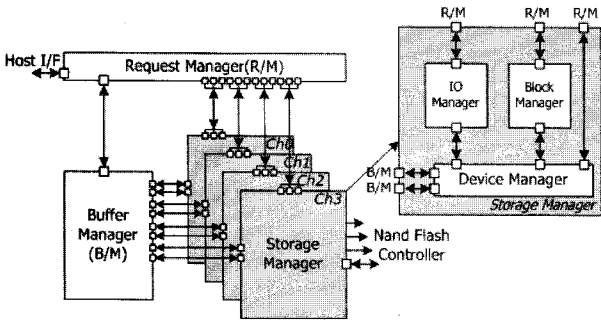


그림 6. FTL의 구성도
Fig. 6. FTL structure.

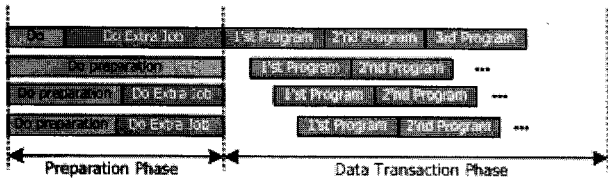


그림 7. Command pipeline을 위한 FTL 선행 처리 작업
Fig. 7. FTL preparation for command pipeline.

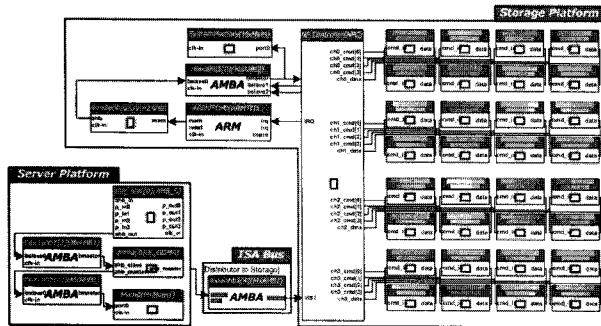


그림 8. VIP의 구성도
Fi. 8. VIP block diagram.

게 다음과 같은 서브블록으로 구성되어 있다. 서버에서 보내어진 논리 주소(logical address)를 받아 처리하는 Request Manager부분과 circular 버퍼를 관리하고 모니터링하는 Buffer Manager, 각 멀티채널의 동작을 관리하는 Storage Manager부분으로 구성된다. Storage Manager에서는 블록단위의 처리(블록할당 및 해제, bad 블록처리, merge)를 담당하는 Block Manager와 page단위의 처리(page map table생성 및 page단위의 동작과 관련한 전반적인 request처리)를 담당하는 IO Manager 그리고 NAND 플래시 controller의 동작을 담당하는 Device Manager로 구성된다.

기존 방식의 FTL에 더하여 NAND 플래시 controller의 효율을 높이기 위한 방법이 추가되었다. 그림 7에서 처럼 서버로부터의 명령(command)을 처리하기 전에 FTL에서의 동작 중 일부를 사전 처리하도록 하였다. 이런 사전처리 작업으로(Do Preparation, Do Extra

Job), 서버로부터의 명령을 처리하는 시간동안에는 모든 동작이 파이프라인(pipeline)으로 이뤄지게 되어 빠른 동작이 가능하다. Do Preparation은 load map table, garbage collection, update map table과 관련한 작업이다. 그리고 Do Extra Job은 특정 채널에서 이미 할당된 preparation time까지 시간이 남을 경우 지우기 또는 다음에 있을 데이터 전송에 필요한 map table을 읽고 Tm는 동작과 관련한 작업이다.

III. 실험

1. Test를 위한 환경

NAND 플래시의 성능을 테스트하기 위해서는 서버에서의 명령(command)요청과 서버와의 데이터 전송이 필요하다. 또한 NAND 플래시 카드와의 고속 인터페이스도 필요하다. 서버와의 인터페이스는 USB나 SATA(Serial ATA) 또는 PCIe(PCI express)가 쓰일 수 있다. 하지만 이 인터페이스 부분은 우리가 중점을 두는 NAND 플래시메모리 카드의 높은 성능을 위한 부수적인 이슈이므로 본 실험에서는 데이터 bit을 확장하여 64bit 데이터 버스, modified ISA protocol(보드 테스트 시 전송에 대한 에러를 줄이기 위해 ISA protocol에 클럭을 추가하여 사용함)을 사용하여 진행하였다. 또한 페이지 데이터의 전송에서는 DMA 모드를 사용하여 대량(Burst)의 데이터가 연속적으로 전송되도록 하였다. 이와 같은 큰 구성을 갖고 VIP(Virtual IP)를 통하여 FTL에서의 알고리즘 분석 및 하드웨어 성능의 선행 측정을 실시하였고 최종적으로 FPGA 보드로 NAND 플래시 controller 및 FTL의 성능을 측정하였다.

2. VIP(Virtual IP) platform

Virtual IP를 사용하여 NAND 플래시 하드웨어부분을 emulation하고 FTL 알고리즘 테스트를 진행하였다. 그림 8은 VIP platform의 구성을 나타낸 것으로 크게 서버 단말기(server platform), ISA bus, 플래시 메모리 카드(storage platform)로 구성되어 있다. ISA bus를 통하여 서버의 읽기/쓰기/지우기 요청이 플래시 메모리 카드로 전송된 다음 플래시 메모리 카드로의 data전송이 이뤄지도록 하였다.

VIP에서의 하드웨어 부분은 NAND 플래시 controller내의 CDMA(Command DMA) 처리 방법, 플래시 메모리 카드(storage platform)내에서의 버스 사용을 그리고 각 채널에서의 NAND 플래시 controller 처

리 속도를 확인하는데 사용되었다. 또한 FTL에서 CDMA(Command DMA)와의 명령처리 확인, Buffer Manager의 동작 및 Block Manager의 merge 및 copy-back operation 등 high level에서의 FTL 알고리즘을 테스트하고 emulation하는데 사용하였다.

3. FPGA Test 환경

FPGA환경에서는 서론에서 얘기한 대로 server, interface, storage부분을 모두 구성하여, 실제와 비슷한 상황에서 테스트를 진행하였다. VIP에서 실험 완료한 FTL을 FPGA 보드에 올려 전체 구성에서의 성능을 측정하였다. 또한 하드웨어 동작을 검증하여 높은 성능에서의 NAND 플래시 카드에 대한 문제점 등을 파악하고 실제 동작에서 생기는 오류를 검증하여 최종 성능을 측정하는데 사용되었다.

그림 9는 FPGA 테스트 환경을 개괄적으로 나타낸 것이다. 서버(server platform)의 MS(Memory Stick card) 카드에 저장된 약 1GB의 HD급 H.264 Video 데이터를 1GB DRAM으로 먼저 로딩(loading)한 다음, Modified ISA Bus(64Bit 데이터 버스)를 통하여 플래시 메모리 카드(storage platform)로 전송하였다. 최종적으로

로 전송된 데이터의 이상 유무를 확인하기 위하여, 플래시 메모리 카드(storage platform)로 저장된 1GB의 HD급 H.264 Video 데이터를 USB를 통해 x86 보드에서 재생하여 보았다.

서버(server platform)에서의 ARM CPU는 플래시 메모리 카드(storage platform)로의 명령(읽기/쓰기/지우기) 및 데이터 전송과 같은 파일 시스템 동작과 관련한 일을 한다. 플래시 메모리 카드(storage platform)에서의 ARM CPU는 block management, address mapping, wear leveling, power management와 같은 FTL의 동작과 관련한 일을 수행한다. 서버에서는 파일 시스템으로 TFS4™ (Transactional File System 4™)^[18]를 올려 동작시켰고 하위 레벨의 디바이스 드라이버는 특정 동작에 맞춰 구성하도록 하였다. board1과 board2의 동작 주파수(system clock)는 24.5Mhz이고 modified ISA bus도 64Bit 데이터 버스, 24.5Mhz로 동작하도록 하였다. 따라서 인터페이스에서의 버스 대역(bandwidth)은 196MB/s(8byte × 24.5Mhz) 이다.

IV. 결 론

1. 실험 결과

가. VIP를 이용한 성능 측정 결과

VIP에서는 기존 방법과 본 논문에서 제시하는 방법, 두 가지를 구현하여 실험하였다. 기존의 방법에서는 그림 8과 같이 4채널 4웨이 인터리빙으로 동작하도록 하였고, 본 논문에서 제시한 방법에서는 4채널 4웨이 구조에서 각 채널이 오버랩(overlap)하면서 동시에 동작하도록 하였다. 이에 대한 실험 조건 및 결과를 표 1에 나타내었다.

기존 방법에서는 4개의 채널을 동시에 액세스하므로 서버에서는 8K bytes 페이지(한개 채널에서 2K bytes 사용)를 기본 단위로 전송하지만 본 논문의 방법에서는 각 채널을 8bit NAND 플래시 칩 2개(1개 채널의 데이터 버스는 16bit가 됨)로 구성하여 4K bytes 페이지(2K bytes × 2개 칩)를 사용한다. 실험에서는 총 1024개의 페이지를 사용하였다. 즉 기존 방법에서는 8 K bytes × 1024, 제안하는 방법에서는 4 K bytes × 1024개의 데이터를 서버에서 플래시 메모리 카드로 전송하였다. 단, 이때 플래시 메모리 카드는 포맷된 상태에서 데이터 전송(즉 초기 지우기 동작 없이)이 이뤄졌다. 이런 조건에서 표 2에 기록된 바와 같이 FTL과 파일 시스템(TFS4™) 없이 NAND 플래시 controller 자체(N/F C.)만의 데이터 전송률과

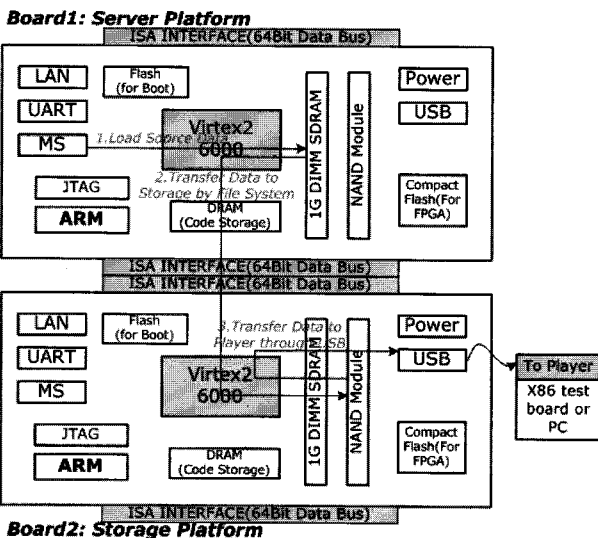


그림 9. FPGA 보드구성
Fig. 9. FPGA test environment block diagram.

표 1. VIP에서의 테스트 조건
Table 1. Constraint of VIP.

	기존 방법 (Multi channel multi way)	제안하는 방법(Enhanced multi channel multi way)
서버에서의 데이터 전송량	8KB x 1024	4KB x 1024
Architecture	4channel 4way	4channel 4way

표 2. VIP에서의 성능 측정 결과
Table 2. Test result of VIP.

	기존 방법에서의 성능 (Multi channel multi way)	제안하는 방법에서의 성능 (Enhanced multi channel multi way)
N/F C.	130.13 MB/s	301.48 MB/s
FTL + N/F C.	110.44 MB/s	240.42 MB/s
FTL/TFS4™ + N/F C.	71.24 MB/s	160.04 MB/s

NAND 플래시 controller와 FTL을 함께(FTL + N/F C.) 테스트한 결과의 전송률 그리고 NAND 플래시 controller, FTL, 파일 시스템(TFS4™) 모두(FTL/TFS4™ + N/F C.)를 올릴 경우를 비교 실험하였다.

플래시 메모리 카드(storage platform)의 NAND 플래시 controller와 FTL, 서버(server platform)에서의 파일 시스템(TFS4™)이 모두 동작할 때 기존 방법에서는 71.24MB/s, 제안하는 방법에서는 160.04MB/s의 결과를 볼 수 있다. 즉 제안하는 방법에서 기존 방법에 비하여 적어도 2배 이상의 성능향상을 기대할 수 있다고 볼 수 있다.

나. FPGA를 이용한 성능 측정 결과

표 3는 앞서 언급한 방법대로 FPGA환경에서 1GB 데이터를 서버(server platform)에서 플래시 메모리 카드(storage platform)로 다운로드(download)했을 때의 경과 시간을 나타낸 것이다. 플래시 메모리 카드에서 FTL없이 NAND 플래시 controller의 device driver만 동작하고 서버에서는 이미 약속된 DMA 전송을 위한 소프트웨어만 동작시킨 결과 약 8초(포맷된 초기상태에서 진행)의 시간이 나왔다. 두 번째로 FTL을 플래시 메모리 카드에 올려 서버의 상태를 그대로 두고 1GB 데이터를 다운로드한 결과 9초의 시간을 얻었다. 최종적으로 서버에 파일 시스템(TFS4™)를 올려 다운로드한 결과 12초 정도의 시간을 얻을 수 있었다. 파일 시스템과 함께 서버와의 환경이 모두 동작할 때를 생각한다면 표3에서 보는 봐와 같이 85.33MB/s(초기 erase동작 제외)와 80.38MB/s(초기 erase동작 포함)의 결과를 얻는다. 이는 FPGA에서 clock을 24Mhz에서 테스트한 결과

표 3. FPGA board에서 1GB 다운로드 테스트 결과
Table 3. Test result of downloading 1GB at FPGA board.

	소요시간(Sec)				측정 성능(MB/s)	
	데이터 기록량(MB)	데이터 기록	삭제동작	소계	erase제외	erase 포함
N/F C.	1,024	8	0.74	8.74	128.00	117.16
FTL+N/F C.	1,024	9	0.74	9.74	113.78	105.13
FTL/TFS4™+N/F C.	1,024	12	0.74	12.74	85.33	80.38

로 플래시 메모리 카드(storage platform)에서의 NAND 플래시 controller가 24Mhz에서 동작했음을 뜻한다. 따라서 실제 ASIC화하여 사용할 경우 VIP에서의 결과에 서처럼 상당한 성능을 기대할 수 있다.

FPGA에서는 기존 방법을 Test하지 않고 제안하는 방법으로만 성능 측정을 실시하였다. 또한 FPGA환경으로 인해 ASIC에서 얻을 수 있는 성능을 볼 수는 없었다. 다만 VIP에서 측정된 결과와 FPGA에서 측정된 성능을 이용하여 상용화에서의 최대 성능을 예상할 수 있다.

2. 실험 평가 및 앞으로의 과제

본 논문에서의 FPGA 실험 환경을 그대로 실시한다면, 서론에서 언급한 것처럼 20Mbps, H.264로 encoding된 2시간짜리 영화(18GB)를 18×12초=216초(3분 36초)에 플래시 메모리 카드로 저장할 수 있다. 이는 FPGA 동작환경에서 측정될 수 있는 결과일 뿐이며 ASIC에서는 이보다 훨씬 빠른 결과를 예상할 수 있다.

본 논문에서는 이렇게 빠른 성능을 얻기 위해 기존과 다른 하드웨어 및 소프트웨어에서의 병렬 처리(parallel processing) 개념을 도입하여 적용해 보았다.

실험에서는 NAND 플래시가 모두 초기화된 상태에서 성능측정을 실시하였다. 하지만 NAND 플래시 메모리 카드를 사용할 때 overwrite가 많이 발생하며, 이 때 FTL의 동작이 전체 카드 성능의 결정적 요소로 작용한다. 따라서 앞으로 실제와 같은 환경에서 추가적인 테스트를 통해 병렬 처리(parallel processing)을 위한 하드웨어 및 소프트웨어 구조 및 방법을 더 찾아 개선할 필요가 있다.

참 고 문 헌

- [1] Mauricio Alvarez, Esther Salamí, Alex Ramirez, Mateo Valero, "A Performance Characterization of High Definition Digital Video Decoding using H.264/AVC", Workload Characterization Symposium, 2005. Proceedings of the IEEE International, pp. 24-33, Oct 2005.
- [2] [http://www.sandisk.com/Products/Catalog\(1191\)-SanDisk_Extreme_IV_Compactflash.aspx](http://www.sandisk.com/Products/Catalog(1191)-SanDisk_Extreme_IV_Compactflash.aspx)
- [3] www.usb.org
- [4] PCI Express Base Specification 2.0
- [5] www.sata-io.org
- [6] Samsung Electronic Co., Ltd., SLC NAND Flash Memory Data Sheet, <http://www.samsung.com>

- [7] Micron Technology, Inc., SLC NAND flash memory data Sheet, <http://www.micron.com>
- [8] Lexar Media, Inc., "Non-volatile Memory control", US Patent, no. 7,215,580 8 May 2007.
- [9] Horii, Takashi Yoshida, Keiichi Nozoe, Atsushi, "Memory system and Memory card", US Patent, no. 7,290,109, 2007.
- [10] ONFI_WhitePaper.pdf, <http://www.onfi.org>
- [11] Chansik Park, Prakash Talawar, Daesik Won, Myungjin Jung, JungBeen Im, Suksan Kim, Youngjoon Choi, "A High Performance NAND Flash Controller for NAND Flash-based Solid State Disk", Non-Volatile Semiconductor Memory Workshop, 2006. IEEE NVSMW 2006. 21st, pp, 17-20, 2006.
- [12] Schuetz, R. HakJune Oh, Jin-Ki Kim Hong-Beom Pyeon, Przybylski, S.A., Gillingham, P. "HyperLink NAND flash Architecture for Mass Storage Applications", Non-Volatile Semiconductor Memory Workshop, 2007 22nd IEEE, pp, 3-4, Aug 2007.
- [13] Mei-Ling Chiang, Paul C. H Lee, and Rwei-Chuan Chang, "Managing Flash Memory in Personal Communication Devices," in Proceedings of the 1997 International Symposium on Consumer Electronics (ISCE'97), pp, 177-182, 1997.
- [14] Intel Corporation, "Understanding the flash translation layer (FTL) specification," APPLICATION NOTE (AP-684), <http://developer.intel.com/>.
- [15] Jesung Kim, Jong Min Kim, Noh, S.H., Sang Lyul Min, and Yookun Cho, "A space-efficient flash translation layer for Compact flash systems," Consumer Electronics, IEEE transactions on, vol. 48, no. 2, May 2002.
- [16] M-Systems, "TrueFFS? Wear-Leveling Mechanism," Technical Note (TN-DOC-017), May, 2002.
- [17] Dawoon Jung, Yoon-Hee Chae, Heeseung Jo, Jin-Soo Kim, and Joonwon Lee, "A Group-Based Wear-Leveling Algorithm for Large-Capacity flash memory storage Systems," in Proceedings of the 2007 international conference on Compilers, architecture, and synthesis for embedded systems (CASES'07), pp, 160-164, 2007.
- [18] TFS4™ (transactional File System 4™), http://www.samsung.com/global/business/semiconductor/products/flash/Products_flashSoftware.html.

 저 자 소 개



강 신 욱(정회원)
2001년 단국대학교 전자공학과
학사 졸업.
2001년~현재 삼성전자
DM연구소 선임 연구원.
2008년 연세대학교 전기전자
공학과 석사과정.

<주관심분야 : SoC, 영상신호 처리, 마이크로 프로세서>



이 동 우(정회원)
2001년 건국대학교 전자공학과
학사 졸업.
2001년~현재 삼성전자
DM연구소 선임 연구원.
<주관심분야 : SoC, 영상 및 음성
신호 처리, HW/SW
Co-Verification>



정 성 훈(정회원)
2003년 연세대학교 컴퓨터과학과
학사 졸업.
2005년 연세대학교 컴퓨터과학과
석사 졸업.
2005년~현재 삼성전자
DM연구소 선임 연구원

<주관심분야 : 3D그래픽, FTL, File System>



이 용 석(평생회원)
1973년 연세대학교 전자공학과
학사 졸업.
1977년 University of Michigan
Electrical Engineering
석사 졸업.
1981년 University of Michigan
Electrical Engineering
박사 졸업.

1991년~현재 연세대학교 전기전자공학과 교수.
<주관심분야 : 마이크로 프로세서 설계, VLSI 설
계, DSP 프로세서 설계, 고성능 연산기 설계>