

SSD의 PC적용을 위한 시스템 수준의 환경 구축

*김동, 방관후, 정의영
연세대학교 전기전자공학과

e-mail : *charikim@yonsei.ac.kr, khbang@dtl.yonsei.ac.kr, eychung@yonsei.ac.kr*

Establishment of System Level environment to apply SSD to PC

*Dong Kim, Kwanhu Bang, Eui-Young Chung
School of Electrical and Electronic Engineering
Yonsei University

Abstract

In this paper, we propose a establishment of system level environment to exploit PC system with SSD (Solid State Disk) by using TLM (Transaction Level Modeling) method with SystemC language. The reason why we choose this modeling method is that it eases RTL (Register Transfer Level) modeling burdens and we can accurately estimate the performance about different architectural changes. Also, it provides simulation speed which is relatively faster than RTL modeling method. The baseline architectural platform we implemented showed that SSD's internal transfer time is a dominant factor, so we need to improve that part and it is expected to be a good simulator to measure the system's overall performance by exploiting SSD's internal architectures.

I. 서론

최근 저전력, 무소음, 내구성, 작고 가벼운 폼팩터 등에 대한 요구가 커짐에 따라 낸드 플래시 메모리를 이용한 SSD(Solid State Disk)는 기존 HDD(Hard Disk

Drive)의 대안으로 떠오르고 있으며, 이와 관련된 업체들은 SSD의 연구 개발에 박차를 가하고 있다. 그러나 현재까지 개발된 SSD의 성능은 기존의 HDD의 속도를 크게 상회하지 못하는 등 전반적으로 만족스럽지 못한 수준에 머무르고 있다. 이에 SSD 설계 프로우 변화에 대한 중요성을 인식하여, 고성능 아키텍처의 개발 및 성능 예측을 효율적으로 하기 위한 상위 수준의 모델링 및 설계 그 환경을 구축할 필요가 있다. 이러한 환경은 SSD 내부의 다양한 구조 변화에 대해 빠른 시간 안에 그 성능을 예측할 수 있도록 도와줌으로써 개발에 들이는 시간과 노력을 덜어줄 수 있다.

II. 본론

2.1 SSD 내부 구조

SSD의 내부 구조는 제조사마다 차이가 존재하지만, 일반적인 내부 구조는 그림 1과 같다. SSD는 크게 wear-leveling 위해 FTL을 구동시키는 프로세서, 호스트(PC)와 데이터를 주고 받는 PATA(SATA) 호스트 인터페이스, 빠른 접근을 돕는 캐시 버퍼, 데이터를 저장하는 낸드 플래시 메모리 및 각각의 메모리들을 위한 컨트롤러가 존재한다. 특히 DRAM으로 구성되는 캐시 버퍼는 SSD의 성능을 높이는 데 큰 역할을 할 수가 있다. 즉 캐시 버퍼에 저장된 데이터가 캐시 hit이 일어나거나, 또는 캐시 miss가 일어나는 경우에 따라 SSD의 데이터 접근 시간은 현격하게 차이가 날 수 있다.

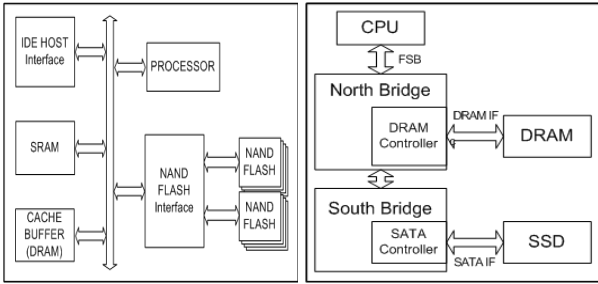


그림 1. SSD 내부 구조 그림 2. PC시스템 구조

2.2 PC 시스템 구조

그림 2에서는 본 연구에서 모델링 된 80x86 시스템의 구조를 보여준다. PC 시스템에서는 크게 North bridge와 South bridge로 나뉘며 North bridge에는 일반적으로 CPU, Main memory, Video accelerator와 같은 고속의 장치들이 연결되며, South bridge에는 일반적으로 하드디스크(또는 SSD), 프린터, USB interface와 같은 비교적 저속의 장치들이 연결된다. CPU는 North bridge를 통해 South bridge에 접근하여 하드디스크(또는 SSD)와 주 메모리와의 정보를 주고받도록 하는 역할을 한다.

III. 구현

PC시스템 및 SSD의 모델링에 사용된 언어는 전송 수준 방식의 모델링에 적합한 SystemC[1]를 이용하였다. PC시스템의 경우 인텔의 965 칩셋과 ICH8 칩셋 스펙[2],[3]을 참고하여 모델링을 하였으며, 특히 North bridge와 South bridge 간 데이터 전송의 핵심이 되는 PCI-express 스펙의 필수 기능들을 지원하도록 모델링 하였다. SSD의 경우 SATA 인터페이스[4]로 채택하였으며, 전송 명령은 현재 대부분 사용되고 있는 DMA 전송만을 지원 가능하도록 모델링하였다. 특히 SSD 내부 구조의 경우, channel 수, way 수, 낸드 플래시 메모리의 종류(SLC/MLC) 및 캐시 버퍼의 크기를 각각 파라미터화 하여 내부 구조 변경에 대한 성능의 변화를 쉽게 알아 볼 수 있도록 하였다.

IV. 실험 결과

그림 3은 기존 SSD를 모델링한 16KB DMA 전송에 대한 실험 결과로 각각 SSD로의 쓰기, 캐시 버퍼 읽기 miss, 캐시 버퍼 읽기 hit에 대한 데이터 전송 시간 및 각 구간 별 차지하는 전송 시간을 보여준다. 캐시 버퍼 읽기 hit의 경우 SSD 내부에서 차지하는 데이터 전송 시간의 비율은 약 37%로 낸드 플래시 메모리까

지의 접근 없이 비교적 빠른 속도로 읽기가 가능하나 쓰기 모드와 캐시 버퍼 읽기 모드에서는 SSD의 내부에서 차지하는 데이터 전송 시간의 비율이 각각 72%, 77%로 16KB DMA 전송 시간의 대부분을 차지하여 전체 전송 시간을 지체시키는 주요 원인을 잘 보여 준다.

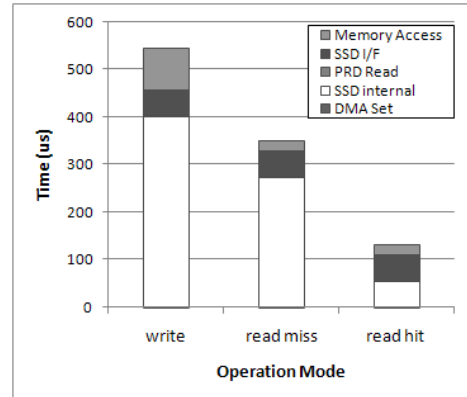


그림 3. 16KB DMA 전송 결과

V. 결론 및 향후 연구 방향

실험 결과를 통해 알 수 있듯이 SSD 내부의 구조 탐색을 통한 속도 개선이 PC 시스템의 전반적인 데이터 전송 시간을 줄여줄 수 있음을 보여 준다. 이에 본 연구에서 구축된 시스템 수준의 시뮬레이터를 통해 향후 SSD 내부의 channel 또는 way 수, FTL 알고리즘의 종류, 캐시 버퍼의 사이즈 및 정책 등 다양한 구조적인 탐색이 비교적 빠른 시간 안에 가능해질 것으로 기대된다.

Acknowledgement : 본 논문은 삼성전자의 지원 및 2007년 정부의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임 (KRF-2007-313-D00578)

참고문헌

- [1] IEEE Standard SystemC Language Reference Manual: <http://www.systemc.org>
- [2] Intel corporation: 'Intel 965 Express Chipset Family', data sheet, July 2006
- [3] Intel corporation: 'Intel I/O Controller Hub 8 (ICH8) Family' data sheet, July 2006
- [4] SATA Specification: Revision 1.0a, Jan. 2003