정량적인 성능 평가를 위한 Solid-State Disk 시뮬레이션 환경 구축

*하승환, 방관후, 전민제, 김동, 박상훈, 정의영
연세대학교 전기전자공학과
e-mail : shha@dtl.yonsei.ac.kr, khbang@dtl.yonsei.ac.kr, jjuninho@dtl.yonsei.ac.kr, charikim@dtl.yonsei.ac.kr, soskhong@dtl.yonsei.ac.kr, eychung@yonsei.ac.kr

Solid-State Disk Simulation Environment Establishment for Quantitative Performance Analysis

*Seung-Hwan Ha, Kwanhu Bang, Minje Jun,
Dong Kim, SangHoon Park, Eui-Young Chung
School of Electrical and Electronic Engineering
Yonsei University

Abstract

Recently, SSD (Solid-State Disk) is actively studied in both academia and industry as an alternative to HDD. However, most of existing works are related to software without consideration about hardware effect of SSD. We implemented a cycle-accurate simulation environment of SSD for accurate performance estimation. We expect that quantitative performance analysis with our SSD simulation environment will help the researches on SSD from both hardware and software perspectives.

I. 서론

최근 HDD를 대체할 저장매체로서 SSD가 주목을 받고 있으며, 이와 관련된 연구가 활발히 진행 중이다. 하지만 대부분의 경우 SSD의 HW 특성을 고려하지 않은 SW 개발에 치중하여 그 기법에 대한 정량적인 비교 결과가 부족했다. 이에 각종 기법들에 대한 정량적인 비교 결과를 산출할 수 있는 시스템 수준의 정확도를 갖는 SSD 시뮬레이션 환경을 구축하였다. 구축된 환경은 HW를 고려한 FTL 개발과 SSD 내부구조 탐색에 도움을 주어 개발에 들이는 시간과 노력을 덜어줌과 동시에 개발된 기법들의 효과를 정량적으로 판단할 수 있을 것이다.

II. 본론

SSD의 일반적인 구조는 그림 1과 같다. SSD는 호스트와 연결되는 SATA 인터페이스, FTL 알고리즘과 Map Table 등을 저장하는 SRAM, 이를 연산하는 CPU, 데이터의 빠른 접근을 위한 캐시 버퍼와 컨트롤러, 데이터를 저장 하는 낸드 플래시 메모리와 이에 대한 제어 및 웨이 인터리빙을 지원하는 낸드 플래시 컨트롤러로 구성된다.

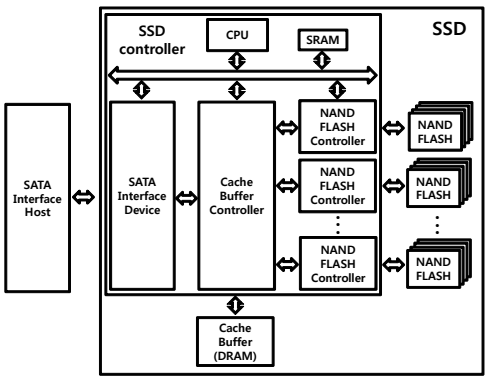

그림 1 Target SSD Architecture

Ⅲ. 구현

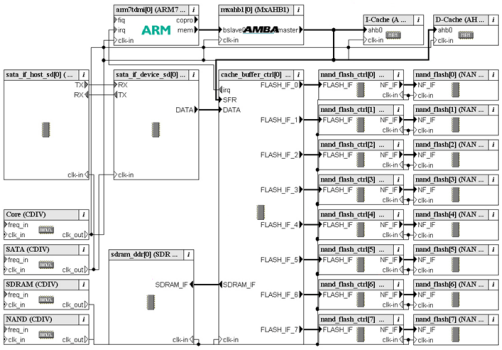

그림 2 구현된 SSD 시뮬레이션 환경

2.1 컴포넌트 모델링

컴포넌트 모델링에 사용된 언어는 SystemC 로서 Cycle-Accuracy를 만족하도록 하였다[1],[2]. 모델링은 각 컴포넌트가 갖는 기능 구현과 더불어 각종 변수를 파라미터 화하여 높은 Flexibility를 갖추는 것에 중점을 두었다[3],[4],[5].

2.2 시뮬레이션 과정

시뮬레이션은 SATA 인터페이스에서 Trace Driven 하여 읽기/쓰기, LSA, length의 정보가 입력되는 것으로 시작된다.

쓰기의 경우 캐시 버퍼 컨트롤러에서 CPU에 요청이 들어왔음을 알리고, 이후 FTL 알고리즘이 수행되어 데이터의 목적지나 Merge를 위한 정보 등을 캐시 버퍼 컨트롤러로 보내준다. 캐시 버퍼 컨트롤러는 이 정보를 통해 명령 또는 데이터를 해당 채널/웨이의 낸드 플래시 컨트롤러로 보내준다. 낸드 플래시 컨트롤러는 전달받은 명령 또는 데이터를 처리한다. 이후 데이터 전송이 완료가 되면 캐시 버퍼 컨트롤러가 그 정보를 SATA 인터페이스로 전달하여 다음 Trace를 처리하게 된다.

읽기의 경우 쓰기와 유사하게 우선 캐시 버퍼 컨트롤러에서 CPU에 요청이 들어왔음을 알리고, 이후 FTL 알고리즘이 수행되어 읽을 데이터의 위치에 대한 정보를 캐시 버퍼 컨트롤러에 보내준다. 캐시 버퍼 컨트롤러는 그 명령은 해당 채널/웨이의 낸드 플래시 컨트롤러로 보내준다. 낸드 플래시 컨트롤러가 읽어온 데이터는 FTL의 연산 결과의 순서에 따라 SATA 인터페이스로 전달되고 데이터 전송이 완료 되면 다음 Trace를 처리하게 된다.

Ⅳ. 실험 결과

그림 3은 SLC 낸드 플래시 메모리를 멀티 채널/웨이를 갖는 SSD의 Sequential Write에 대한 실험 결과를 보여준다. 실험에서 사용된 FTL은 채널/웨이 인터리빙 기능만을 구현된 것으로 채널/웨이에 대한 HW의 영향만을 확인하였다.

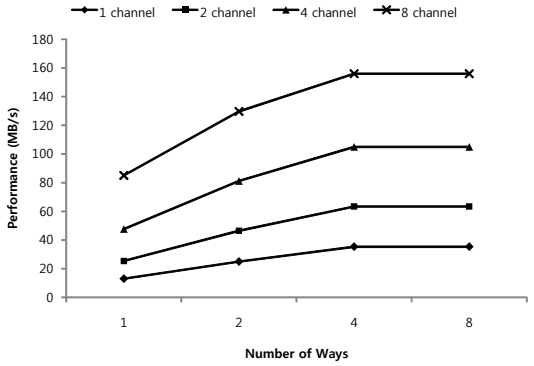

그림 3 Sequential Write 실험 결과

1 채널 1 웨이와 비교하여 1 채널 4 웨이가 270% 성능이 향상된 것에 반해 4 채널 1 웨이가 364% 성능이 향상되어 채널의 효과가 웨이보다 큰 것을 확인 할 수 있다.

4 웨이 와 8 웨이 의 성능이 거의 일치한 점에서 SLC 낸드 플래시 메모리의 경우, 4 웨이 에서 웨이 인터리빙의 효과가 한계에 달하는 것을 확인할 수 있다.

Ⅴ. 결론 및 향후 연구 방향

구축된 시뮬레이션 환경은 정량적인 결과를 보여주어, 각종 기법들에 대한 신뢰성 있는 결과 및 분석을 가능하게 한다. 이를 통해 향후 다양한 FTL에 대한 정량적인 평가, SSD HW를 고려한 FTL 구현, SSD 내부 구조 탐색을 수행할 계획이다.

Acknowledgement : 이 논문은 2008년 정부(교육과학기술부)의 재원으로 한국학술진흥재단 의 지원 및 IDEC 의 지원을 받아 수행된 연구임.

참고문헌

[1] IEEE Standard SystemC Language Reference Manual : http://www.systemc.org
[2] SoC Designer : http://carbondesignsystems.com/
[3] SATA Specification : Revision 1.0a, Jan. 2003
[4] ONFI (Open NAND Flash Interface) Specification : Revision 2.0 Feb. 2008
[5] NAND Flash Datasheet : http://www.hynix.com/