



SSD 테스트 범위의 극대화 방안

홍성희 (하이닉스반도체)

I. Abstract

Solid State Drive(SSD)는 NAND Flash Memory와 이를 제어하는 Controller로 구성되는 차세대 대용량 저장매체로써 기존의 HDD에 비해 상대적으로 읽기 및 쓰기가 빠르고, 동작 중 전력소비가 적으며, 또한 발열이 낮고, 강한 내구성을 가지고 있기 때문에 PC에서 HDD보다 더 적합한 저장매체라 할 수 있다. 하지만, 가격이 비싸다는 점과 아직 안정성이 검증되지 않은 이유로 HDD를 대체하는데 시간이 더 필요하다. 안정성을 높이기 위해 많은 개발자들이 노력을 하고 있음에도 불구하고 공통적으로 겪는 문제점 중에 하나가 바로 객관적인 성능 및 품질을 판단하는 테스트 기준이 없다는 것이다. 본 논문에서 다루고자 하는 내용은 기존 HDD 및 SSD 테스트의 한계점을 찾아보고 이를 보완한 테스트 기준을 제안함으로써, 많은 개발자들이 SSD 개발에 있어 좀 더 정확한 테스트로 SSD의 성능 및 안정성을 높이는데 기여할 수 있도록 SSD 테스트 관점에서의 Key word를 결론으로 제시하고자 한다.

II. 서론

HDD에 비해 상대적으로 빠른 읽기 및 쓰기 동작 중 낮은 전력소비로 SSD는 PC 시장에서 분명 HDD보다 성능 측면에서는 한 수 위인 것만은 확실하다. 그럼에도 불구하고 많은 PC 제조사들이 초기 기대와는 달리 SSD 채택에 있어 많은 고민을 하고 있다. 이는 분명 가격적인 문제가 주요 원인인긴 하나 또 다른 큰 문제가 하나 있다. 바로 SSD의 성능 및 품질을 판단하기 위한 객관적인 테스트 기준이 없다는 것이다. 기준이 없다는 것은 곧, 성능 및 품질을 비교할 수 없다는 것으로 SSD가 실제 HDD에 비해 정말 Field에서 우수한가? 또는 어떤 SSD가 우수한가? 를 판단하기 어렵다고 해석 할 수 있다.

앞에서 언급한 이유 때문에 SSD 개발업체에서는 다음의 3가지 테스트에 많은 인력과 시간을 소비한다.

- (1) 읽기 및 쓰기 성능 측정
- (2) Firmware 검증
- (3) 호환성 검증

위에서 언급한 3가지의 테스트는 SSD의 성능 및 품질을 판단함에 있어 가장 중요한 항목들이라고 할 수 있다. 즉, 어느 개발업체의 SSD가 가장 빠르고, 안정적이며, PC에서 오래 사용할 수 있는지를 비교할 수 있는 테스트인 것이다.

그러나 앞서 언급한 것처럼 객관적인 기준이 없기 때문에 이 주요 3가지 주요 항목을 테스트하더라도 어떤 SSD가 우수한지 비교한다는 것은 상당히 어렵다. 그렇다면 왜 그런 기준이 오랫동안 만들어지지 못했는지 많은 테스트를 하면서 확인한 결과 다음과 같은 한계점을 찾을 수 있었다.

먼저, 읽기 및 쓰기 성능 측정에 있어 가장 필요한 것은 그것을 측정하는 프로그램이다. 일반적으로 기존 HDD의 읽기, 쓰기 성능을 측정하는 프로그램을 SSD에서도 그대로 사용한다. 이는 오랜 시간 HDD 개발자 및 사용자가 많은 제품의 읽기, 쓰기 성능을 비교하고자 그 성능에 영향을 미치는 많은 요소들을 변경할 수 있게 환경을 미리 만들어 놓았기 때문이다. 예를 들자면, OS나 File System 종류, Cache의 사용유무, Data Pattern 등이 그런 것들이다. 그런 요소들을 변경하면서 HDD의 성능을 측정하면 때론 그 결과에서 큰 차이를 보일 때도 있다. 그런 요소들을 반영하여 만들어진 프로그램 종류만 20가지 안팎이며, 큰 업체에서 주로 사용하는 프로그램만 10가지 내외가 된다. 하지만, 그 많은 종류의 프로그램으로 SSD의 성능을 측정하면서 기존의 프로그램들이 가지고 있는 몇몇 가지 한계점을 발견하게 되었다. 그 한계점은 바로 다양한 OS와 Application 프로그램, 하드웨어 등을 반영하기가 거의 불가능하다는 것이다. 이런 한계점은 동일한 프로그램임에도 불구하고 사용자가 가진 환경에 따라 성능 측정의 결과가 차이를 보이게 되는 결과를 만들게 된다.

다음으로, 성능 측정에 있어 다른 모든 환경을 일치시키더라도 Firmware에 따라 성능 및 안정성의 큰 차이가 생길 수 있는데, Firmware 검증은 다른 테스트에 비해 객관적인 수치로 나타내는 것이 상당히 어려운 테스트이다. Sudden Power Loss Recovery, Wear-Leveling, Bad Block Management 등이 대표적인 Firmware 검증 항목들이다. 왜냐하면, SSD 외부에서 위 3가지 검증 항목들의 값을 보기 위해서는 SSD 자체에서 SMART나 특정 명령어를 통해 값을 볼 수 있도록 SSD 제작사에서 제공을 해야 하는데 보통은 그러지 않기 때문에 객관적인 기준을 세운다는 것은 거의 불가능하며, 가능한 많은 환경에서 정상적으로 동작을 하는지 여부를 확인하는 정도가 Firmware의 성능과 안정성을 보는 전부가 되고 있다.

마지막으로 다양한 OS와 BIOS, Chipset에 대한 호환성에 대한 한계점이다. OS와 BIOS만 해도 각각 10여 가지가 넘고, Chipset은 수십 가지가 된다. 그렇다면, SSD를 개발하면서 어떻게 하면 처음부터 그렇게 많은 환경의 호환성을 보장할 것인가에 대한 의문이 생기게 되는데, 당장 모든 환경에서 호환성을 검증하기 위해서는 다양한 OS와 BIOS가 설치된 상당한 양의 PC 종류가 필요하고, 그것을 모두 테스트하기 위해 적게는 일주일 길게는 한 달 이상의 시간이 필요하다. 또한 Firmware나 PCB 등이 바뀔 때마다 매번 이를 진행한다는 것은 상당한 무리가 따르게 된다.

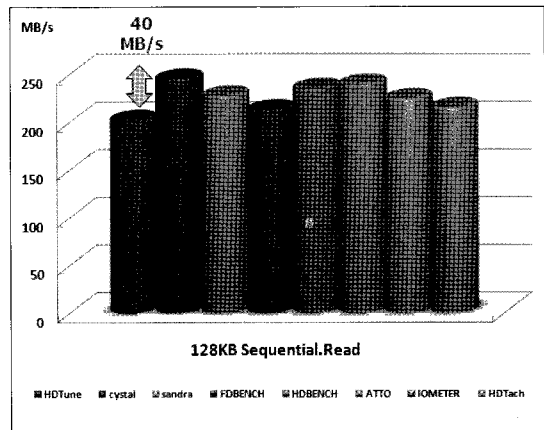
III. 본 론

1. 읽기 및 쓰기 성능 측정

현재 SSD 개발업체에서 사용하고 있는 읽기

및 쓰기 성능을 측정하는 프로그램의 종류와 각각의 프로그램에서 성능에 영향을 줄 수 있는 요소들 중에 어떤 것을 지원하는지 <표 1>에 간략히 정리하였다. 그리고 각각의 성능 측정 프로그램에서 동일한 SSD가 읽기와 쓰기 성능 측정 결과에서 어떻게 다른 결과를 보이는지 <그림 1>과 <그림 2>에 나타내었다.

앞서 서론에서도 언급하였지만 SSD의 읽기 및 쓰기의 성능에 영향을 주는 요소들은 상당히 많이 존재한다. 그 중에서 Transfer Size는 한번에 얼마의 Data를 주고받을지를 결정하는 것으로 NAND Flash Memory로 구성된 SSD에서는 상당히 중요한 요소 중에 하나이다. 기본적으로 Sector Size와 NAND Flash의 Page Size가 다르고, NAND Flash의 특성상 읽기와 쓰기는 Page 단위로 수행되고, 지우기는 Block 단위로 수행되는 것 때문에 Transfer Size가 NAND



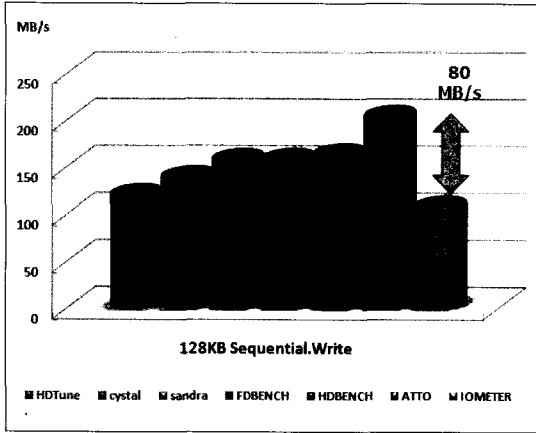
<그림 1> 동일한 SSD가 다양한 측정 Tool에서 다른 읽기 성능을 보이는 예

Flash의 Page Size와 다를 경우 수시로 일어나는 Background Garbage Collection과 Merge Operation에 의해 성능이 저하된다. 하지만, 가능한 Page Size 또는 Block Size와 일치하는 Transfer Size로 Data를 주고받으면, 그만큼

<표 1> 다양한 읽기 및 쓰기 성능 측정 프로그램 및 특징

	IOMeter	PCMark	H2Bench	ATTO	HD Tach	HD Tune	HDBench
Transfer Size	Variable	Fix (random)	Fix	Variable	Fix (64KB)	Variable	Fix (seq: 128KB, Ran: 64KB)
DATA Pattern	06:RANDOM 08:2bit repetition	Almost ZERO pattern	Repetitive pattern	ALL ZERO	---	ALL ZERO	Routine RANDOM Pattern
CMD Read/Write	Read/write Seq/Ran	Read/Write Sea/Ran	Read/Write	Read/Write	Read Seq.	Read/Write Seq.	Read/Write Sea/Ran
OS	Windows Linux Netware	Windows	Windows (Win32 Console App)	Windows	Windows	Windows	Windows
Partition	physical/Logical	logical	physical	logical	Physical/ logical	Physical/ Logical(read)	logical
Latency	V		V		V	V	
Multi process	V	V					
Queue Depth	V			V			
Precondition	V						
CMD Duty cycle	V						
B.G.C	V						
Remark	*freezing *MB/s, IOPS	*Real home usage *Application test	*DATA Integrity check *application test				*detail graph

- Quick hit BM Tool / Simple point and click
- Not consistent results / just Simple Read/Write test
- Not support specific & critical option
- Not give a larger perspective of performance



〈그림 2〉 동일한 SSD가 다양한 측정 Tool에서 다른 쓰기 성능을 보이는 예

Background Garbage Collection, Merge Operation과 같은 내부 동작의 빈도수가 줄어들게 되어 읽기 및 쓰기 성능을 향상시킬 수 있다.

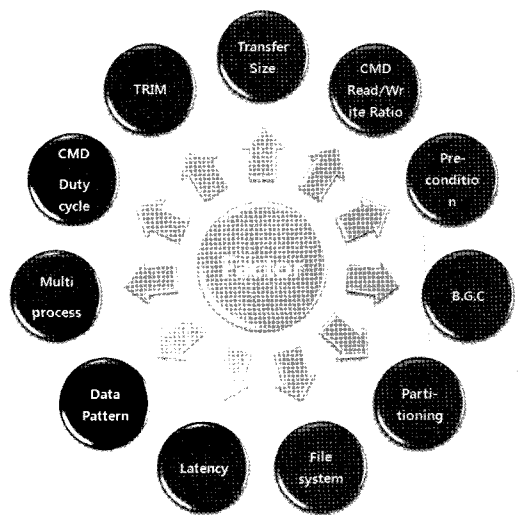
또한 Precondition의 정도에 따라 읽기 및 쓰기의 성능이 차이가 날 수 있는데, Precondition은 SSD의 NAND Flash Memory의 상태를 의미하는 것으로 Clean 또는 Dirty 상태로 나뉜다. Clean은 Flash Memory의 Block에 Data가 없는 상태로 쓰기 동작이 일어나도 이전에 쓰인 Data가 없어서 소거 및 Background Garbage Collection, Merge Operation이 없어도 된다. 즉, 쓰기 동작이 빨리 끝난다는 것이다. 반대로 Dirty 상태는 Flash Memory의 Block이 Valid 혹은 Invalid Data로 채워져 있어서 쓰기 동작 전에 앞서 말한 동작들이 필요하기 때문에 실제 쓰기 동작이 느려진다. Precondition은 Random 쓰기와 Sequential 쓰기로 만들 수 있는데, 일반적으로 Random 쓰기를 했을 경우 SSD의 내부 상태가 더 Dirty한 상태가 된다. 즉 조각난 Data들이 여기저기 많이 흩어지게 되므로 Background에서 해야 할 동작이 많이 생긴다.

정리하자면 Precondition이 성능에 영향을 주

는 것 역시 동일한 Physical Address에 덮어쓰기가 되지 않고 Data를 쓰기 전에 이전 Data의 소거가 선행되어야 한다는 NAND Flash의 특성 때문에 그렇다. Invalid Data라 할지라도 어떤 Data가 채워져 있으면, 쓰기 동작을 위해 무조건 소거 작업이 선행되어야 하며, 그 과정에서 Merge와 Background Garbage Collection으로 인한 Overhead가 생기며, 직접적으로 읽기 및 쓰기 성능에 지대한 영향을 줄 수 있다. 때문에 SSD의 상태가 Clean이나 Dirty나에 따라 성능 측정 결과의 차이가 발생하고 있으며, 정확한 성능 측정을 위해서는 Precondition의 정도를 알고 측정하여야 하는 것이다.

그 외에도, 읽기와 쓰기의 비율, Back Ground Garbage Collection, File System, Latency, Data Pattern, Multi-process, Command Duty Cycle, Trim Command 등이 읽기와 쓰기 성능에 많은 영향을 미친다. 〈그림 3〉

이렇듯 읽기와 쓰기 성능에 영향을 주는 많은 요소들을 하나하나 변경해 가면서 정확한 측정



〈그림 3〉 읽기 및 쓰기 성능에 영향을 주는 요소

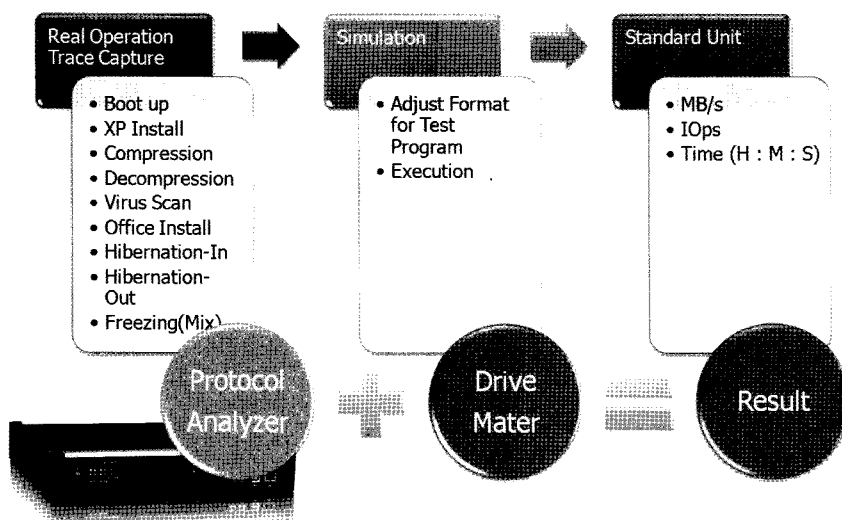
을 하기란 상당한 시간과 노력이 필요하고, 다른 하드웨어적인 차이로 인해 객관성도 떨어지게 된다. 이를 배제하고 읽기와 쓰기 성능을 객관적으로 측정 가능한 방법을 많은 시행착오 끝에 찾았으며, 다음과 같이 제시할 수 있다. <그림 4>

위에서 보여주는 테스트를 위해서 기본적으로 Host와 SSD간에 주고받는 Data와 Command의 정보를 확인할 수 있는 Protocol Analyzer가 필요하고, Drivemaster라고 하는 Host 대신 ATA Command를 생성 할 수 있는 Windows 기반 프로그램이 필요하다. 이 두 가지를 가지고 User 환경에서 일어날 수 있는 일상적인 동작의 Trace를 Capture 하고, Simulation하여 실제 상황에서 얼마의 성능을 가지는지 확인할 수 있다. 즉, 다양한 환경에서 달라질 수 있는 많은 요소를 배제하고, 단지 실제 환경만 고려함으로써 그 상황에서 어떤 SSD의 성능이 우수한지를 확인할 수 있는 것이다. 물론 Drivemaster를 사용함으로써 하드웨어적인 영향을 최대한 배제할 수 있다.

먼저, Protocol Analyzer를 사용하여 Boot-up, OS Installation, Compression, Virus Scan 등 등 사용자들이 PC에서 많이 하는 일상 동작을 그대로 진행하면서 Trace를 Capture한다. 다음으로 Capture된 Trace를 바탕으로 Drivemaster에서 그대로 Simulation할 수 있도록 Drivemaster용 Script를 작성한다. 마지막으로 Drivemaster에 SSD를 연결해서 동작시키면 되는 것이다. 이 방법을 사용 하게 되면 앞서 언급한 성능에 영향을 줄 수 있는 요소들이 SSD에 그대로 적용되어 객관적인 결과를 얻을 수 있게 된다. 물론 다양한 시나리오를 만들어서 실제의 동작을 대신 할 수 있는 다양한 Script를 작성하여, 일상 동작에 대한 성능 검증이 되어야 성능의 향상을 이끌어 낼 수 있다.

2. Firmware 검증

NAND 플래시 메모리는 물리적인 특성으로 인하여 데이터를 갱신하기 전에 해당 페이지가



<그림 4> 환경에 영향을 받지 않는 읽기 및 쓰기 측정 방법

속한 블록을 소거하는 동작이 선행되어야 한다. 이러한 소거 동작을 감추기 위해 플래시 변환 계층(이하 FTL : Flash Translation Layer)이라는 소프트웨어가 파일시스템과 NAND 플래시 메모리 사이에 존재 하게 된다. FTL은 소거 동작을 감추기 위한 기능 외에 NAND 플래시 메모리의 각 블록에 대한 소거 횟수가 특정 제약 값을 넘지 않도록 모든 블록들의 소거 횟수를 일정하게 유지하는 소거 연산 평준화(Wear Leveling) 기능과 NAND 플래시 메모리의 배드 블록들을 관리 하는 기능 등의 다양한 기능들을 수행하게 된다. 이와 같은 기능들을 사용하여 SSD의 동작을 제어하는 FTL 소프트웨어는 SSD 개발 업체마다 다른 알고리즘을 사용하여 여러 형태로 구현 된다. 하지만 SSD 개발 업체 별로 상이한 알고리즘을 적용하여 구현한 SSD FTL 소프트웨어의 성능과 안정성을 정확하게 테스트하고 검증 할 수 있는 환경이 현재로서는 부족한 실정이다. 실제로 SSD 개발 업체의 자체 SSD 성능 및 안정성 검증이 완료 되어 당사로 납품된 SSD 제품들에 대하여 당사의 SSD 성능 및 안정성 검사를 수행한 결과, SSD 읽기 명령을 수행하는 도중 NAND 플래시 메모리 읽기 동작 에러가 발생할 경우, 이에 수반되는 Host 리셋 명령어 수행 후의 읽기 동작들에 대하여 SSD가 UNC를 반환하는 오동 작을 하는 경우들이 검출된 사례가 있었다.

이와 같이 다양한 FTL의 기능들에 대해 상이한 알고리즘이 적용된 SSD의 소프트웨어들을 검증하기 위해서는 WHQL, ATACT, DriveMaster 등의 검증 도구들 외에 각 SSD 업체에서 사용하는 다양한 검증 방안 및 도구들에 대한 취합이 필요하다. 또한 NAND 플래시의 동작을 정교하게 모방하여 NAND 플래시 메모리의 고유한 특

징으로 발생하는 소거/쓰기/읽기 동작 등의 에러 상황들에 대처하기 위한 SSD 소프트웨어의 회복 기능들이 정상적으로 동작하는지 여부를 검증할 수 있는 환경이 갖추어져야 한다. 또한 NAND 플래시 메모리의 각 블록들에 대하여 수행된 소거 횟수와 각 블록들에 대한 소거 동작 후 다음 소거가 발생하기 전까지 발생한 읽기 횟수 등의 정보들을 모니터링하여 NAND 플래시의 물리적인 특성들을 만족하기 위한 FTL의 소거 연산 평준화 및 읽기 Disturb Protection등의 기능들이 정상적으로 동작하는지를 검증 할 수 있는 환경들이 갖추어져야 한다.

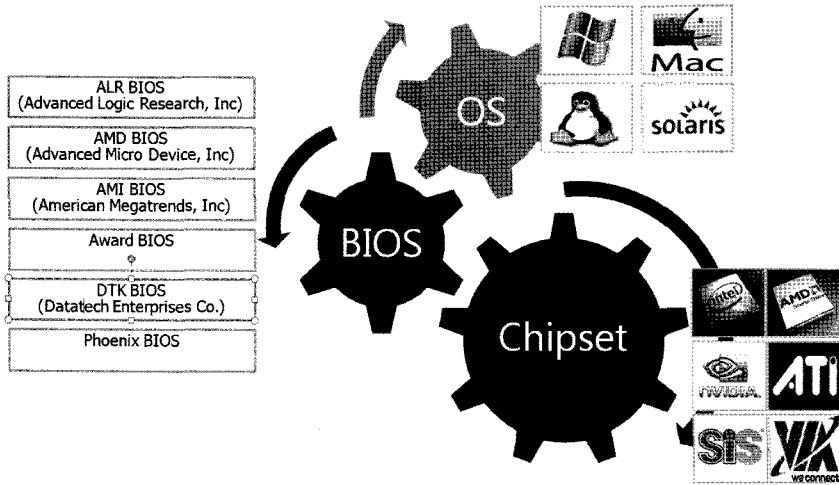
3. OS, BIOS와 Chipset의 호환성 검증

SSD가 사용되는 PC는 하드웨어 이외에 상당히 많은 OS와 BIOS, Chipset의 종류를 가지고 있다. 이런 OS는 읽기와 쓰기의 성능에도 영향을 주지만, BIOS와 Chipset과 더불어 호환성에도 영향을 미친다. 예를 들자면, Windows XP에서는 정상적으로 동작하던 SSD가 Windows 7에서는 동작에 문제가 발생하거나 느려지거나 하는 그런 경우를 드물게 경험할 수 있다. 이는 Windows XP에는 없던 기능을 Windows 7에서 지원하게 되면서 미처 Drive가 해당 기능을 지원하지 못해 생기는 문제라고 할 수 있다. 실제 TRIM의 기능이 그러한 문제를 일으키기도 했다.

<그림 5>는 OS와 BIOS, Chipset의 종류와 업체를 표시하였다.

다음에서 보는 것과 같이 OS는 크게 MS에서 만든 Windows, Apple에서 만든 MAC, Sun에서 만든 Solaris 외에 일반적으로 많이 사용하는 Linux 등 4가지가 있으며, 그 안에서도 각각의 다른 많은 버전들이 존재한다. 또한 BISO 역시

There are many kind of model & version, configuration, feature.
But We have to considerate all of them.



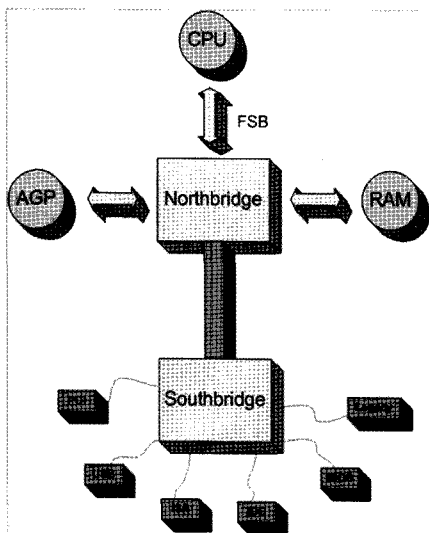
〈그림 5〉 OS와 BIOS, Chipset의 종류

크게 6개 업체에서 많은 버전의 것들을 만들고 있으며, Chipset은 6개의 큰 업체에서 수십 가지의 모델을 가지고 있다. 여기서 Chipset은 〈그림 6〉에서 보듯이 South-bridge를 의미하는 것으로 SSD가 직접 연결되는 Chipset을 뜻한다. 이들 Chipset에서 기본적으로 잘 동작해야 SSD의

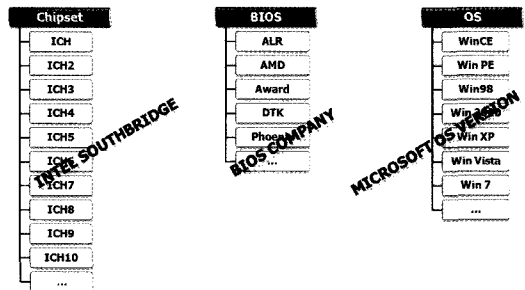
호환성에 문제가 없다고 할 수 있고, PC에서 정상동작 한다고 한다.

〈그림 7〉은 호환성을 고려한 OS와 BIOS, Chipset의 조합으로 Chipset과 OS는 한 업체의 모델만을 나열한 것이다. 그렇다면 모든 업체의 모델을 나열하면 얼마나 많은 경우의 수가 나오는지 예상할 수 있다.

호환성 테스트를 위해 OS, BIOS, Chipset을 모두 고려하여 진행하기엔 이 역시 상당한 노력과 시간이 필요하고, 비용 적으로도 많은 금액이 필요할 수밖에 없다. 이를 해결하기 위해 다음



〈그림 6〉 Diagram of Motherboard Chipset



〈그림 7〉 Combination Chipset/BIOS/OS

과 같은 테스트를 생각하게 되었고 진행하고 자 한다.

테스트 방법은 다음과 같다.

- 주요 Chipset에서 주고받는 Command Trace를 Protocol Analyzer로 Capture하고 Timing/ Sequence/ Data Pattern 정리
- 주요 OS에서 주고받는 Command Trace를 Protocol Analyzer로 Capture하고 Timing/ Sequence/ Data Pattern 정리
- Protocol Analyzer에 있는 Emulator 기능으로 Simulation 후 위에서 정리된 내용과 비교 및 분석

위 각 Step이 선행되고 Pass와 Fail의 Criterion을 설정하고, Data를 수집하기 위해, 초기 분석 및 정리하는 작업에 시간이 일부 소요될 수 있으나 향후에 Data가 쌓이게 되면 점점 호환성 검증을 위한 절차가 간편해져 시간, 노력, 비용이 크게 절감 될 것이다. 그리고 무엇보다 다양하고 객관적인 호환성 검증이 가능해지게 된다.

IV. 결론

실제 User가 체감할 수 있는 SSD의 성능 및 품질 향상을 위해 무엇보다 객관적으로 그것을 표현 할 수 있는 수치가 있어야 하며, 그 수치를 바탕으로 비교 평가를 통해 취약한 부분의 보완이 이루어져야 한다. 그 중에서도 읽기 및 쓰기 성능 측정, Firmware 검증, 호환성 검증은 SSD의 전반적인 성능과 안정성을 판단하는데 가장 중요한 테스트 항목들이며, 정확하게 검증이 되어야 하는 것들이다.

그러한 테스트 환경을 구축하기 위해서는 무

엇보다 다양한 환경에서도 객관적인 결과를 얻을 수 있도록 의도한 요소 이외의 것들에 최대한 영향을 받지 않아야 한다. SSD만의 성능과 안정성을 확인하기 위해, SSD를 제외한 다른 하드웨어 또는 인위적인 간섭이 영향을 주어서도 안되며, 사용자가 사용하는 다양한 Operating System에 의해서도 변화가 있으면 안 된다. 또한 당연히 사용하는 다양한 Application 프로그램도 정상적으로 동작시켜야 한다. 이에 본 논문은 최대한 그런 환경을 배제하고 객관적인 결과를 얻을 수 있는 테스트 방법에 대해 언급하였다.

아직은 본 논문에서 제시한 테스트 방법들이 성능 및 호환성에 영향을 주는 모든 환경을 배제시킬 수는 없다. 앞으로 많은 보완이 이루어져 완성도를 높여야 할 것이며, 현재 Hynix가 가진 테스트들이 SSD 개발업체의 개발 및 성능 향상에 도움을 줄 수 있었으면 하는 바람이다. 또한 많은 SSD 개발업체들의 정확한 테스트를 위한 노력이 필요할 것이다.

저자소개



홍 성 희

1988년 2월 인하대학교 전자공학과 학사
1988년 2월~2001년 2월 현대전자
2001년 3월~현재 하이닉스 반도체

주관심 분야 : 플래시 응용복합제품(SD/eMMC/SSD등)