

SSD Test Shell 제작 프로젝트

챕터의 포인트

- 가상 SSD 제작
- Test Shell Application
- Test Script 작성하기

가상 SSD 제작

SSD 제품을 테스트 할 수 있는 Test Shell 을 제작

1. SSD 를 가상으로 프로그래밍으로 구현한다.
2. Test Shell 프로그램을 제작하여 SSD 동작을 테스트 할 수 있다.
3. 다양한 Test Script를 제작한다.



Test 수행



1. SSD

- HW 를 Software로 구현한다.

2. Test Shell Application

- 테스트 프로그램

3. Test Sciprt

- 테스트 프로그램 內 Test Code

저장할 수 있는 공간

- 저장할 수 있는 최소 공간의 사이즈는 4KB
(한 글자 = 약 1 Byte 으로 간주했을 때, 4,096 글자 저장 가능 공간)
- 각 공간마다 LBA (Logical Block Address) 라는 주소를 가짐
- SSD 는 OS로부터
Read / Write / Unmap 등 다양한 명령어를
전달받는다.



최소화된 기능 수행

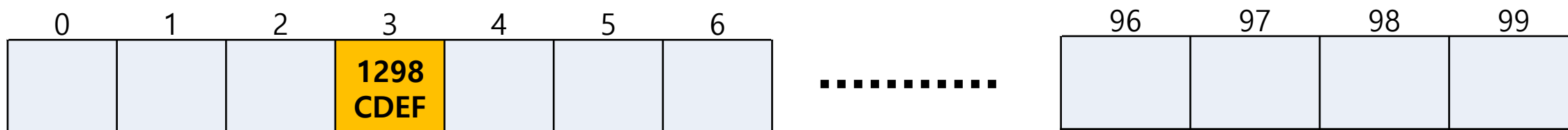
- Read 명령어와 Write 명령어만 존재
- LBA 단위는 4 Byte (실제로는 4KB 이지만, 우리가 만들 최소 저장공간 사이즈는 4 Btye)
- LBA 0 ~ 99 까지 100 칸을 저장할 수 있다.

총 400 Byte를 저장 할 수 있는 가상 SSD 를 구현



- “ssd”

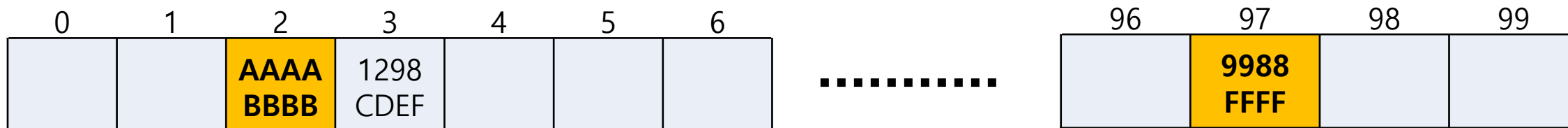
- **ssd W 3 0x1298CDEF** : 3 번 LBA 영역에 값 0x1298CDEF 를 저장한다.



Write 명령어 사용 예시 2

- `ssd W 2 0xAAAABBBB`
- `ssd W 97 0x9988FFFF`

출력결과 : 없음 (저장만 수행)



Read 명령어 사용 예시

- `ssd R 2`

→ 출력결과 : 0xAAAABBBB

- `ssd R 97`

→ 출력결과 : 0x9988FFFF



nand.txt 파일을 생성

- 사용자가 Write 할 때 마다, SSD 내부 (Nand) 에 기록이 된다.
- 이를 모사하여, nand.txt 파일에 값을 저장 해 둔다.



화면 출력하지 않고, result.txt 파일에 결과를 저장

- **Write 명령어 수행시**
 - result.txt 파일 건드리지 않음
 - Write는 내부적으로 기록만 수행한다.
- **Read 명령어 수행시**
 - result.txt 파일 내용이 교체 된다.

Read 명령 시 결과값이 result.txt 파일에 저장

Write 명령 시 nand.txt 파일에 값을 저장

입력	출력
W 20 0x1289CDEF	-
R 20	0x1289CDEF
R 19	0x00000000
W 10 0xFF1100AA	
R 10	0xFF1100AA

PC 용 저장장치 (SSD)

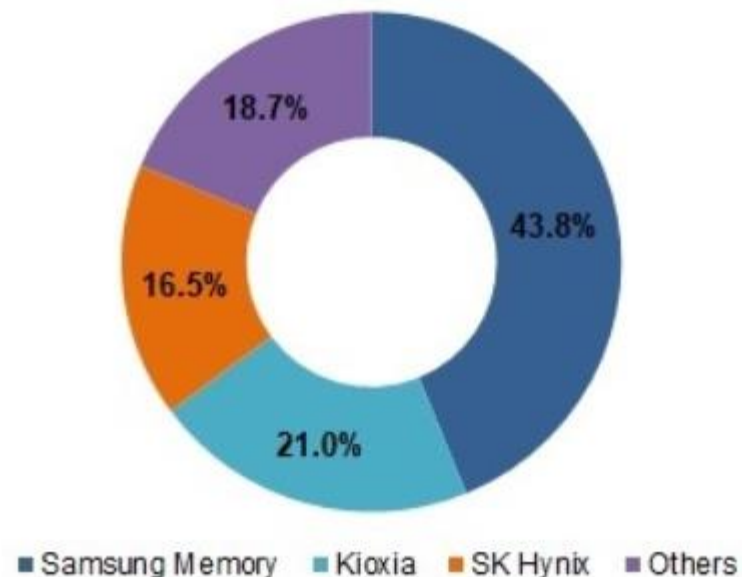
- 세계 1위 : 삼성전자 (B2B, B2C)
- 세계 5위 : SK Hynix (B2B)



Mobile용 저장장치 (emmc / ufs)

- 삼성전자 : 세계 1위
- SKHynix 세계 4위

Smartphone NAND Flash Market Revenue Share Q1 2020

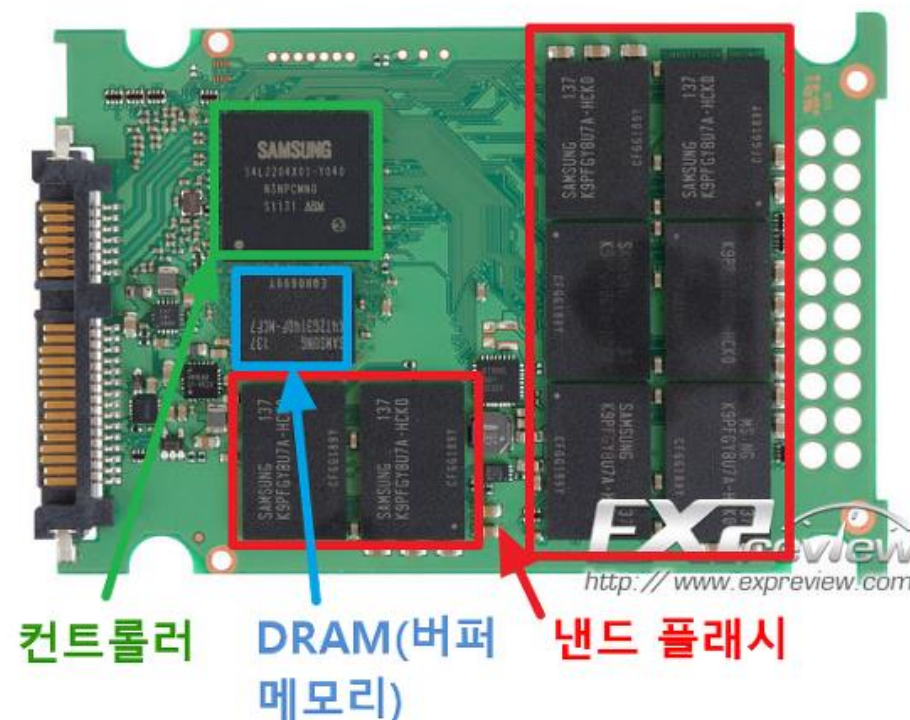


컨트롤러 역할

- OS 는 Filesystem를 거쳐 LBA 주소에 특정 값 R/W 요청
- 컨트롤러가 이 명령어들을 받아 Nand Flash Memory 에 저장

NAND Flash Memory

- 실제로 값들이 저장되는 곳
- 컨트롤러에 의해 값이 저장이 된다.



데이터 범위

- LBA : 0 ~ 99, 10진수
- 값 : 항상 0x가 붙으며 10 글자로 표기한다. (0x00000000 ~ 0xFFFFFFFF)

Read 명령어

- ssd R [LBA]
- result.txt 에 읽은 값이 적힌다. (기존 데이터는 사라진다.)
- 한번도 안적은 곳은 0x00000000 으로 읽힌다.

Write 명령어

- ssd W [LBA] [값]
- nand.txt 에 저장한 값이 기록된다.

Test Shell Application

SSD를 테스트 할 수 있는 프로그램

- Shell 이 동작하여 사용자 입력을 받는다.
- 사용 가능 명령어
 - write
 - read
 - exit
 - help
 - fullwrite
 - fullread

read / write 명령어 수행시,
제작한 "ssd" app을 실행시켜 값 읽기 / 저장 명령을 수행한다.

사용자 입력 예시

- **write 3 0xAAAABBBB**
 - 3번 LBA 에 0xAAAABBBB 를 기록한다.
 - ssd 에 명령어를 전달한다.
- **read 3**
 - 3번 LBA 를 읽는다.
 - ssd 에 명령어를 전달한다.
 - **result.txt 에 적힌 결과를 화면에 출력한다.**

exit 명령어

- Shell 이 종료된다.

help 명령어

- 각 명령어당 사용 방법을 출력한다.

fullwrite 명령어

- LBA 0 번부터 99 번 까지 Write를 수행한다.
- ssd 전체에 값이 써진다.
- ex) fullwrite 0xABCDFFFF
→ 모든 LBA에 값 0xABCDFFF 가 적힌다.

fullread 명령어

- LBA 0 번부터 99 번 까지 Read를 수행한다.
- ssd 전체 값을 모두 화면에 출력한다.
- ex) fullread
→ 모든 LBA의 값들이 화면에 출력된다.

기능 구현시 유의사항

- 입력받은 매개변수가 유효성 검사 수행
 - 파라미터의 Format이 정확해야 함
 - LBA 범위는 0 ~ 99
 - A ~ F, 0 ~ 9 까지 숫자 범위만 허용
- 없는 명령어를 수행하는 경우 "INVALID COMMAND" 을 출력
 - 어떠한 명령어를 입력하더라도 segment fault가 나오면 안된다.

Test Script 작성하기

실제 Test를 수행하는 프로그램을 제작

- Test 를 수행하는 명령어의 집합을 Test Script라고 한다.
- 주로 “검증”팀에서 Test Script를 제작한다.

TestApp1 제작하기

- Test Shell 에서 “testapp1” 명령어를 입력하면 Script가 수행된다.
- 먼저 fullwrite를 수행한다.
- fullread를 하면서, write 한 값대로 read가 되는지 확인한다.
 - SSD 정상 동작하는지 확인하는 프로그램

TestApp2 제작하기

- 0 ~ 5 번 LBA 에 0xAAAABBBB 값으로 총 30번 Write를 수행한다.
- 0 ~ 5 번 LBA 에 0x12345678 값으로 1 회 Over Write를 수행한다.
- 0 ~ 5 번 LBA Read 했을 때 정상적으로 값이 읽히는지 확인한다.