

1. 요구분석

추가 요구사항 설명-

((정적 섹터 매핑, 동적 섹터 매핑, 정적 블록 매핑, 동적 블록 매핑 구현))

2. 설계

Flowchart의 순서에 따른 과정 함수로 설명

전체 Flowchart

세부 Flowchart FTL_read, FTL_write

(내부 추가함수 가능, 하지만 FTL_read, FTL_write 내부에서 수행)

동적 섹터/블록 매핑 알고리즘에서 해당 알고리즘을 방법을 선택한 이유에 대해 꼭 설명할 것!!!!!!!

3. 구현

디바이스 설명

Flash_read, Flash_write, Flash_erase (1차 보고서 복사해도 됨)

(단, 1차 보고서와 변경사항이 있을 시에는 표기)

각 함수 설명

예) FTL_write (a, b)

함수 역할

중요 소스코드 라인 설명

4. 테스트 결과

각 workload 수행

결과 스크린 샷

결과 분석 (문제점 에러사항 등)

5. 요구사항에 따른 분석

- 정적/동적 섹터 매핑, 정적/동적 블록 매핑의 차이점
- 각 알고리즘에서 매핑 테이블을 생성하기 위한 DRAM 용량 계산 식으로 분석 보여주기

2. FTL (정적/동적 섹터/블록 매핑 구현)

정적/동적 섹터/블록 FTL 구현

1. Input 파일에서 일렬로 명령어 수행

2. Main () 함수 안에서 다음과 같은 명령어 수행

init 혹은 **trace** 혹은 **view**

((동적, 정적, 블록, 섹터 매핑을 초이스로 만들어도 되고 따로 함수 파일로 제공해도됨))

init : init () 플래시 메모리 생성

trace: FTL_read () ;

 FTL_write () ;

 //다음의 명령어를 실행하여 전체 read, write, erase 횟수를 카운트 해야함.

 Flash_read ();, Flash_write ();, Flash_erase ();

(FTL_read/FTL_write 외에 다른 함수를 편의상 만들어도 상관없음 허나 모두 FTL_read, FTL_write 내부에서 함수를 불러오거나 실행해야함.)

결과값:

init 의 ouput: x megabytes flash memory ;

trace 의 ouput: read, write, erase 횟수

 (Flash_read, Flash_write, Flash_erase 의 수행 횟수)

view 의 output: 매핑 테이블 정보 상태