



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0077609
(43) 공개일자 2023년06월01일

- | | |
|---|---|
| <p>(51) 국제특허분류(Int. Cl.)
G06F 9/48 (2018.01) G06F 12/0862 (2016.01)
G06F 3/06 (2006.01) G06F 9/50 (2018.01)</p> <p>(52) CPC특허분류
G06F 9/4881 (2013.01)
G06F 12/0862 (2013.01)</p> <p>(21) 출원번호 10-2022-0054905
(22) 출원일자 2022년05월03일
심사청구일자 2022년05월03일</p> <p>(30) 우선권주장
1020210164973 2021년11월25일 대한민국(KR)</p> | <p>(71) 출원인
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)</p> <p>(72) 발명자
싱라젠드라
경기도 수원시 영통구 삼성로 129
송재원
경기도 수원시 영통구 삼성로 129
김재섭
경기도 수원시 영통구 삼성로 129</p> <p>(74) 대리인
팬코리아특허법인</p> |
|---|---|

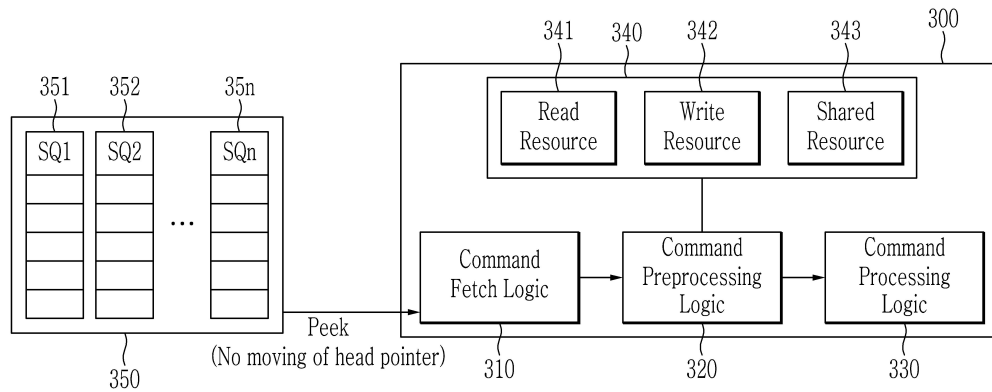
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 비휘발성 메모리 기반의 저장 장치, 장치 컨트롤러 및 방법

(57) 요약

비휘발성 메모리를 포함하는 저장 장치의 장치 컨트롤러가 제출 큐로부터 명령을 피크하며, 장치 리소스의 상태에 기초해서 피크한 명령이 실행 가능한지를 판단할 수 있다. 장치 컨트롤러는 명령이 실행 가능하다는 판단에 응답하여 명령을 폐치하고, 폐치한 명령에 장치 리소스를 할당하고 실행할 수 있다.

대표도



(52) CPC특허분류

G06F 3/0659 (2013.01)

G06F 9/5016 (2013.01)

G06F 9/5077 (2013.01)

명세서

청구범위

청구항 1

비휘발성 메모리를 포함하는 저장 장치의 장치 컨트롤러로서,

장치 리소스,

복수의 제출 큐 중에서 제1 제출 큐로부터 제1 명령을 피크(peek)하며, 상기 제1 명령이 실행 가능하다는 판단에 응답하여 상기 제1 명령을 페치(fetch)하는 명령 페치 로직,

상기 장치 리소스의 상태에 기초해서 상기 명령 페치 로직에서 피크한 상기 제1 명령이 실행 가능한지를 판단하는 명령 전처리 로직, 그리고

상기 제1 명령이 실행 가능하다는 판단에 응답하여, 상기 명령 페치 로직에서 페치한 상기 제1 명령에 상기 장치 리소스를 할당하고 상기 제1 명령을 실행하는 명령 처리 로직

을 포함하는 장치 컨트롤러.

청구항 2

제1항에서,

상기 명령 페치 로직은 상기 제1 명령의 피크 시에 상기 제1 제출 큐의 헤드 포인터를 이동하지 않고, 상기 제1 제출 큐의 상기 헤드 포인터를 이동하여 상기 제1 명령을 페치하는, 장치 컨트롤러.

청구항 3

제1항에서,

상기 명령 페치 로직은 상기 복수의 제출 큐 중 제2 제출 큐로부터 제2 명령을 피크하고,

상기 명령 전처리 로직은 상기 장치 리소스의 상태에 기초해서 상기 제2 명령이 실행 가능한지를 판단하고, 상기 제2 명령이 실행 가능하지 않다는 판단에 응답하여 상기 제2 제출 큐를 상기 장치 리소스를 대기하는 대기 상태로 지정하는

장치 컨트롤러.

청구항 4

제3항에서,

상기 명령 페치 로직은 상기 제2 명령이 실행 가능하지 않다는 판단에 응답하여 상기 제2 명령을 페치하지 않고 드롭하는, 장치 컨트롤러.

청구항 5

제3항에서,

상기 명령 전처리 로직은 상기 장치 리소스의 상태에 기초해서 상기 대기 상태로 지정된 상기 제2 제출 큐가 처리 가능한지를 판단하고,

상기 제2 제출 큐가 처리 가능하다는 판단에 응답하여, 상기 명령 페치 로직은 상기 제2 제출 큐로부터 상기 제2 명령을 피크하고, 상기 명령 전처리 로직은 상기 제2 명령이 실행 가능한지를 판단하는

장치 컨트롤러.

청구항 6

제5항에서,

상기 명령 전처리 로직은, 상기 장치 리소스 중에서 상기 제2 명령이 대기하는 리소스가 사용 가능한 경우, 상기 제2 제출 큐가 처리 가능하다고 판단하는, 장치 컨트롤러.

청구항 7

제6항에서,

상기 명령 전처리 로직은 상기 제2 제출 큐를 상기 복수의 제출 큐 중 나머지 제출 큐에 비해 우선해서 처리하는, 장치 컨트롤러.

청구항 8

제1항에서,

펜딩 명령 큐를 더 포함하며,

상기 명령 폐치 로직은 상기 제1 제출 큐로부터 상기 제1 명령과 제2 명령을 피크하고,

상기 명령 전처리 로직의 상기 제1 명령이 실행 가능하고 상기 제2 명령이 실행 가능하지 않다는 판단에 응답하여, 상기 명령 폐치 로직은 상기 제1 명령과 상기 제2 명령을 폐치하고, 상기 명령 전처리 로직은 상기 제2 명령을 상기 펜딩 명령 큐에 저장하는

장치 컨트롤러.

청구항 9

제8항에서,

상기 명령 전처리 로직은 상기 장치 리소스의 상태에 기초해서 상기 펜딩 명령 큐에 저장된 제3 명령이 실행 가능한지를 판단하고,

상기 펜딩 명령 큐의 상기 제3 명령이 실행 가능하다는 판단에 응답하여, 상기 명령 처리 로직은 상기 펜딩 명령 큐의 상기 제3 명령을 실행하는

장치 컨트롤러.

청구항 10

제9항에서,

상기 명령 전처리 로직은 상기 펜딩 명령 큐를 상기 복수의 제출 큐에 비해 우선해서 처리하는, 장치 컨트롤러.

청구항 11

제1항에서,

상기 명령 폐치 로직은 상기 제1 제출 큐로부터 상기 제1 명령과 제2 명령을 피크하고,

상기 명령 전처리 로직의 상기 제1 명령이 실행 가능하고 상기 제2 명령이 실행 가능하지 않다는 판단에 응답하여, 상기 명령 폐치 로직은 상기 제1 제출 큐에서 상기 제1 명령을 폐치하고, 상기 제2 명령을 제1 제출 큐에서 패킹(packing)하는

장치 컨트롤러.

청구항 12

제11항에서,

상기 명령 폐치 로직은 상기 제1 제출 큐의 헤드 포인터를 상기 제1 명령의 수만큼 이동하고, 상기 헤드 포인터가 이동된 엔트리부터 상기 제2 명령을 패킹하는, 장치 컨트롤러.

청구항 13

제11항에서,

상기 명령 전처리 로직은 상기 장치 리소스의 상태에 기초해서 패킹된 상기 제1 제출 큐가 처리 가능한지를 판단하고,

상기 제1 제출 큐가 처리 가능하다는 판단에 응답하여, 상기 명령 폐치 로직은 상기 제1 제출 큐로부터 상기 제2 명령을 피크하고, 상기 명령 전처리 로직은 상기 제2 명령이 실행 가능한지를 판단하는

장치 컨트롤러.

청구항 14

제13항에서,

상기 장치 리소스 중에서 상기 제2 명령이 대기하는 리소스가 사용 가능한 경우, 상기 명령 전처리 로직은 상기 제1 제출 큐를 상기 복수의 제출 큐 중 나머지 제출 큐에 비해 우선해서 처리하는, 장치 컨트롤러.

청구항 15

제1항에서,

상기 명령 전처리 로직은 상기 장치 리소스 중에서 상기 제1 명령을 실행하기 위한 리소스가 사용 가능한 경우 상기 제1 명령이 실행 가능하다고 판단하고, 상기 제1 명령을 폐치하는, 장치 컨트롤러.

청구항 16

제15항에서,

상기 장치 리소스는 읽기 명령에 전용인 읽기 리소스, 쓰기 명령에 전용인 쓰기 리소스 및 상기 읽기 명령과 상기 쓰기 명령이 공유하는 공유 리소스를 포함하며,

상기 명령 전처리 로직은 상기 읽기 리소스와 상기 쓰기 리소스 중 상기 제1 명령의 입출력 유형에 해당하는 리소스와 상기 공유 리소스가 사용 가능한 경우, 상기 제1 명령을 실행하기 위한 리소스가 사용 가능하다고 판단하고, 상기 제1 명령을 폐치하는

장치 컨트롤러.

청구항 17

제15항에서,

상기 장치 리소스는 상기 제1 명령을 실행하는데 사용되는 복수의 유형의 리소스를 포함하며,

상기 명령 전처리 로직은 상기 복수의 유형의 리소스 중 일부 유형의 리소스가 사용 가능하지 않은 경우, 상기 제1 명령을 실행하기 위한 리소스가 사용 가능하지 않다고 판단하는

장치 컨트롤러.

청구항 18

제15항에서,

상기 명령 전처리 로직은 상기 장치 리소스 중에서 상기 제1 명령을 실행하기 위한 리소스가 사용 가능하지 않은 경우, 상기 제1 명령에 상기 장치 리소스를 할당하지 않는, 장치 컨트롤러.

청구항 19

비휘발성 메모리, 그리고

상기 비휘발성 메모리로의 액세스를 제어하며, 제1 제출 큐로부터 상기 제1 제출 큐의 헤드 포인터의 이동 없이 제1 명령을 가져오고, 장치 리소스의 상태에 기초해서 상기 제1 명령이 실행 가능한지를 판단하고, 상기 제1 명령이 실행 가능하다는 판단에 응답하여 상기 제1 제출 큐의 상기 헤드 포인터를 이동하는 장치 컨트롤러

를 포함하는 저장 장치.

청구항 20

비휘발성 메모리를 포함하는 저장 장치에 의해 실행되는 방법으로서,
 제출 큐로부터 명령을 피크하는 단계,
 장치 리소스의 상태에 기초해서 상기 명령이 실행 가능한지를 판단하는 단계,
 상기 명령이 실행 가능하다는 판단에 응답하여 상기 명령을 폐치하는 단계,
 상기 명령이 실행 가능하다는 판단에 응답하여, 상기 명령에 상기 장치 리소스를 할당하고 상기 명령을 실행하는 단계
 를 포함하는 방법.

발명의 설명**기술 분야**

[0001] 개시 내용은 비휘발성 메모리 기반의 저장 장치, 장치 컨트롤러 및 방법에 관한 것이다.

배경 기술

[0002] 솔리드 스테이트 드라이브(solid-state drive, SSD)와 같은 비휘발성 메모리 기반 저장 장치의 리소스 풀에는 읽기 리소스와 쓰기 리소스가 별도로 존재할 수 있다. 읽기 리소스와 쓰기 리소스는 각각 특정 유형의 입출력(input/output, I/O)를 처리하는데 사용될 수 있다. 또한 리소스 풀은 모든 유형의 I/O가 공유하는 공유 리소스를 포함할 수 있다.

[0003] 공유 리소스가 모두 사용되면, 공유 리소스를 다시 사용할 수 있을 때까지 저장 장치는 모든 I/O 유형의 명령 처리를 일시적으로 중지한다. 특히, 저장 장치가 많은 수의 쓰기 명령을 폐치하면, 거의 모든 공유 리소스가 쓰기 명령에 의해 점유될 수 있다. 대체적으로 쓰기 명령을 처리하는데 시간이 오래 걸리므로, 공유 리소스가 더 오래 점유될 수 있다. 따라서, 읽기 명령이 읽기 리소스를 사용할 수 있지만, 공유 리소스가 모두 소진되어 저장 장치가 읽기 명령을 처리할 수 없을 수 있다. 또한, 쓰기 리소스가 소진된 경우에 저장 장치가 쓰기 명령을 가지고 오는 경우, 쓰기 리소스가 사용 가능할 때까지 저장 장치는 공유 리소스를 차단할 수 있다. 그러면 저장 장치가 읽기 명령을 가지고 오더라도 읽기 명령을 처리할 수 없을 수 있다. 이와 같이, 일부 리소스의 소진에 의해 다른 명령이 대기하고 대기 명령은 명령 처리 파이프라인에 병목 현상을 일으킬 수 있으므로, 명령의 대기 시간이 늘어나서 명령 처리 효율성이 저하될 수 있다.

발명의 내용**해결하려는 과제**

[0004] 어떤 실시예는 명령 처리 효율성을 높일 수 있는 비휘발성 메모리 기반 저장 장치, 그 장치 컨트롤러 또는 방법을 제공할 수 있다.

과제의 해결 수단

[0005] 한 실시예에 따르면, 비휘발성 메모리를 포함하는 저장 장치의 장치 컨트롤러가 제공될 수 있다. 상기 장치 컨트롤러는 장치 리소스, 명령 폐치 로직, 명령 전처리 로직 및 명령 처리 로직을 포함할 수 있다. 상기 명령 폐치 로직은 복수의 제출 큐 중에서 제1 제출 큐로부터 제1 명령을 피크하며, 상기 제1 명령이 실행 가능하다는 판단에 응답하여 상기 제1 명령을 폐치할 수 있다. 상기 명령 전처리 로직은 상기 장치 리소스의 상태에 기초해서 상기 명령 폐치 로직에서 피크한 상기 제1 명령이 실행 가능한지를 판단할 수 있다. 상기 명령 처리 로직은 상기 제1 명령이 실행 가능하다는 판단에 응답하여, 상기 명령 폐치 로직에서 폐치한 상기 제1 명령에 상기 장치 리소스를 할당하고 상기 제1 명령을 실행할 수 있다.

[0006] 어떤 실시예에서, 상기 명령 폐치 로직은 상기 제1 명령의 피크 시에 상기 제1 제출 큐의 헤드 포인터를 이동하지 않고, 상기 제1 제출 큐의 상기 헤드 포인터를 이동하여 상기 제1 명령을 폐치할 수 있다.

[0007] 어떤 실시예에서, 상기 명령 폐치 로직은 상기 복수의 제출 큐 중 제2 제출 큐로부터 제2 명령을 피크할 수 있

다. 상기 명령 전처리 로직은 상기 장치 리소스의 상태에 기초해서 상기 제2 명령이 실행 가능한지를 판단하고, 상기 제2 명령이 실행 가능하지 않다는 판단에 응답하여 상기 제2 제출 큐를 상기 장치 리소스를 대기하는 대기 상태로 지정할 수 있다.

- [0008] 어떤 실시예에서, 상기 명령 폐치 로직은 상기 제2 명령이 실행 가능하지 않다는 판단에 응답하여 상기 제2 명령을 폐치하지 않고 드롭할 수 있다.
- [0009] 어떤 실시예에서, 상기 명령 전처리 로직은 상기 장치 리소스의 상태에 기초해서 상기 대기 상태로 지정된 상기 제2 제출 큐가 처리 가능한지를 판단할 수 있다. 상기 제2 제출 큐가 처리 가능하다는 판단에 응답하여, 상기 명령 폐치 로직은 상기 제2 제출 큐로부터 상기 제2 명령을 피크하고, 상기 명령 전처리 로직은 상기 제2 명령이 실행 가능한지를 판단할 수 있다.
- [0010] 어떤 실시예에서, 상기 명령 전처리 로직은, 상기 장치 리소스 중에서 상기 제2 명령이 대기하는 리소스가 사용 가능한 경우, 상기 제2 제출 큐가 처리 가능하다고 판단할 수 있다.
- [0011] 어떤 실시예에서, 상기 명령 전처리 로직은 상기 제2 제출 큐를 상기 복수의 제출 큐 중 나머지 제출 큐에 비해 우선해서 처리할 수 있다.
- [0012] 어떤 실시예에서, 상기 장치 컨트롤러는 펜딩 명령 큐를 더 포함할 수 있다. 상기 명령 폐치 로직은 상기 제1 제출 큐로부터 상기 제1 명령과 제2 명령을 피크할 수 있다. 상기 명령 전처리 로직의 상기 제1 명령이 실행 가능하고 상기 제2 명령이 실행 가능하지 않다는 판단에 응답하여, 상기 명령 폐치 로직은 상기 제1 명령과 상기 제2 명령을 폐치하고, 상기 명령 전처리 로직은 상기 제2 명령을 상기 펜딩 명령 큐에 저장할 수 있다.
- [0013] 어떤 실시예에서, 상기 명령 전처리 로직은 상기 장치 리소스의 상태에 기초해서 상기 펜딩 명령 큐에 저장된 제3 명령이 실행 가능한지를 판단할 수 있다. 상기 펜딩 명령 큐의 상기 제3 명령이 실행 가능하다는 판단에 응답하여, 상기 명령 처리 로직은 상기 펜딩 명령 큐의 상기 제3 명령을 실행할 수 있다.
- [0014] 어떤 실시예에서, 상기 명령 전처리 로직은 상기 펜딩 명령 큐를 상기 복수의 제출 큐에 비해 우선해서 처리할 수 있다.
- [0015] 어떤 실시예에서, 상기 명령 폐치 로직은 상기 제1 제출 큐로부터 상기 제1 명령과 제2 명령을 피크할 수 있다. 상기 명령 전처리 로직의 상기 제1 명령이 실행 가능하고 상기 제2 명령이 실행 가능하지 않다는 판단에 응답하여, 상기 명령 폐치 로직은 상기 제1 제출 큐에서 상기 제1 명령을 폐치하고, 상기 제2 명령을 제1 제출 큐에서 패킹할 수 있다.
- [0016] 어떤 실시예에서, 상기 명령 폐치 로직은 상기 제1 제출 큐의 헤드 포인터를 상기 제1 명령의 수만큼 이동하고, 상기 헤드 포인터가 이동된 엔트리부터 상기 제2 명령을 패킹할 수 있다.
- [0017] 어떤 실시예에서, 상기 명령 전처리 로직은 상기 장치 리소스의 상태에 기초해서 패킹된 상기 제1 제출 큐가 처리 가능한지를 판단할 수 있다. 상기 제1 제출 큐가 처리 가능하다는 판단에 응답하여, 상기 명령 폐치 로직은 상기 제1 제출 큐로부터 상기 제2 명령을 피크하고, 상기 명령 전처리 로직은 상기 제2 명령이 실행 가능한지를 판단할 수 있다.
- [0018] 어떤 실시예에서, 상기 명령 전처리 로직은 상기 제1 제출 큐를 상기 복수의 제출 큐 중 나머지 제출 큐에 비해 우선해서 처리할 수 있다.
- [0019] 어떤 실시예에서, 상기 명령 전처리 로직은 상기 장치 리소스 중에서 상기 제1 명령을 실행하기 위한 리소스가 사용 가능한 경우 상기 제1 명령이 실행 가능하다고 판단하고, 상기 제1 명령을 폐치할 수 있다.
- [0020] 어떤 실시예에서, 상기 장치 리소스는 읽기 명령에 전용인 읽기 리소스, 쓰기 명령에 전용인 쓰기 리소스 및 상기 읽기 명령과 상기 쓰기 명령이 공유하는 공유 리소스를 포함할 수 있다. 상기 명령 전처리 로직은 상기 읽기 리소스와 상기 쓰기 리소스 중 상기 제1 명령의 입출력 유형에 해당하는 리소스와 상기 공유 리소스가 사용 가능한 경우, 상기 제1 명령을 실행하기 위한 리소스가 사용 가능하다고 판단하고, 상기 제1 명령을 폐치할 수 있다.
- [0021] 어떤 실시예에서, 상기 장치 리소스는 상기 제1 명령을 실행하는데 사용되는 복수의 유형의 리소스를 포함할 수 있다. 상기 명령 전처리 로직은 상기 복수의 유형의 리소스 중 일부 유형의 리소스가 사용 가능하지 않은 경우, 상기 제1 명령을 실행하기 위한 리소스가 사용 가능하지 않다고 판단할 수 있다.

[0022] 어떤 실시예에서, 상기 명령 전처리 로직은 상기 장치 리소스 중에서 상기 제1 명령을 실행하기 위한 리소스가 사용 가능하지 않은 경우, 상기 제1 명령에 상기 장치 리소스를 할당하지 않을 수 있다.

[0023] 다른 실시예에 따르면, 비휘발성 메모리 및 장치 컨트롤러를 포함하는 저장 장치가 제공될 수 있다. 상기 장치 컨트롤러는 상기 비휘발성 메모리로의 액세스를 제어하며, 제1 제출 큐로부터 상기 제1 제출 큐의 헤드 포인터의 이동 없이 제1 명령을 가져오고, 장치 리소스의 상태에 기초해서 상기 제1 명령이 실행 가능한지를 판단하고, 상기 제1 명령이 실행 가능하다는 판단에 응답하여 상기 제1 제출 큐의 상기 헤드 포인터를 이동할 수 있다.

[0024] 또 다른 실시예에 따르면, 비휘발성 메모리를 포함하는 저장 장치에 의해 실행되는 방법이 제공될 수 있다. 상기 방법은 제출 큐로부터 명령을 피크하는 단계, 장치 리소스의 상태에 기초해서 상기 명령이 실행 가능한지를 판단하는 단계, 상기 명령이 실행 가능하다는 판단에 응답하여 상기 명령을 폐지하는 단계, 그리고 상기 명령이 실행 가능하다는 판단에 응답하여, 상기 명령에 상기 장치 리소스를 할당하고 상기 명령을 실행하는 단계를 포함할 수 있다.

도면의 간단한 설명

[0025] 도 1은 한 실시예에 따른 스토리지 시스템의 예시 블록도이다.

도 2는 다른 실시예에 따른 스토리지 시스템의 예시 블록도이다.

도 3은 한 실시예에 따른 저장 장치의 장치 컨트롤러의 예시 블록도이다.

도 4는 한 실시예에 따른 장치 컨트롤러의 장치 리소스를 예시하는 도면이다.

도 5 및 도 6은 도 3에 도시한 장치 컨트롤러의 동작의 한 예를 설명하는 도면이다.

도 7은 한 실시예에 따른 저장 장치의 장치 컨트롤러의 동작을 예시하는 흐름도이다.

도 8은 다른 실시예에 따른 저장 장치의 장치 컨트롤러의 예시 블록도이다.

도 9는 또 다른 실시예에 따른 저장 장치의 장치 컨트롤러를 예시하는 도면이다.

도 10은 도 9에 도시한 장치 컨트롤러의 동작의 한 예를 설명하는 도면이다.

도 11은 또 다른 실시예에 따른 저장 장치의 장치 컨트롤러의 동작을 예시하는 흐름도이다.

도 12는 또 다른 실시예에 따른 저장 장치의 장치 컨트롤러를 예시하는 도면이다.

도 13은 도 12에 도시한 장치 컨트롤러의 동작의 한 예를 설명하는 도면이다.

도 14는 또 다른 실시예에 따른 저장 장치의 장치 컨트롤러의 동작을 예시하는 흐름도이다.

도 15는 한 실시예에 따른 컴퓨터 시스템을 나타내는 예시 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0026] 아래에서는 첨부한 도면을 참고로 하여 본 발명의 실시예에 대하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다.

[0027] 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다. 도면을 참고하여 설명한 흐름도에서, 동작 순서는 변경될 수 있고, 여러 동작들이 병합되거나, 어느 동작이 분할될 수 있고, 특정 동작은 수행되지 않을 수 있다.

[0028] 또한, 단수로 기재된 표현은 "하나" 또는 "단일" 등의 명시적인 표현을 사용하지 않은 이상, 단수 또는 복수로 해석될 수 있다. 제1, 제2 등과 같이 서수를 포함하는 용어는 다양한 구성요소를 설명하는데 사용될 수 있지만, 구성요소는 이러한 용어에 의해 한정되지는 않는다. 이들 용어는 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로 사용될 수 있다.

[0029] 도 1은 한 실시예에 따른 스토리지 시스템의 예시 블록도이다.

[0030] 도 1을 참고하면, 스토리지 시스템(100)은 호스트 장치(110)와 저장 장치(120)를 포함할 수 있다.

- [0031] 호스트 장치(110)는 저장 장치(120)와 인터페이스할 있는 컴퓨팅 플랫폼(또는 컴퓨팅 장치)을 포함할 수 있다. 호스트 장치(110)는 예를 들면 데스크톱 컴퓨터(desktop computer), 서버 컴퓨터, 태블릿 컴퓨터(tablet computer), 랩톱 컴퓨터(laptop computer), 스마트폰(smartphone)과 같은 이동 전화기(mobile phone), 멀티미디어 플레이어(multimedia player), 게임 콘솔(game console), 텔레비전, 다양한 유형의 사물 인터넷(internet of thing, IoT) 기기 등을 포함할 수 있으며, 이에 한정되지 않는다.
- [0032] 호스트 장치(110)는 프로세서(111)와 호스트 메모리(112)를 포함할 수 있다. 프로세서(111)는 명령어를 실행하여 다양한 동작(예를 들면, 연산, 로직, 제어, 입출력 등의 연산)을 수행할 수 있다. 프로세서(111)는 예를 들면 중앙 처리 장치(central processing unit, CPU), 그래픽 처리 장치(graphics processing unit, GPU), 마이크로프로세서 또는 어플리케이션 프로세서(application processor, AP)일 수 있다. 호스트 메모리(112)는 프로세서(111)에 의해 액세스되어 사용되는 시스템 메모리일 수 있다. 호스트 메모리(112)는 예를 들면 동적 랜덤 액세스 메모리(dynamic random access memory, RAM)와 같은 랜덤 액세스 메모리를 포함할 수 있다.
- [0033] 저장 장치(120)는 비휘발성 메모리 기반 저장 장치로, 장치 컨트롤러(121)와 비휘발성 메모리(122)를 포함할 수 있다. 장치 컨트롤러(121)는 비휘발성 메모리(122)로의 액세스를 제어할 수 있다. 어떤 실시예에서, 장치 컨트롤러(121)는 비휘발성 메모리(122)에 액세스하기 위한 프로토콜로 NVMe(non-volatile memory express) 프로토콜을 지원할 수 있다. 어떤 실시예에서, 비휘발성 메모리는 플래시 기반 메모리, 예를 들면 NAND 플래시 메모리를 포함할 수 있다. 어떤 실시예에서, 비휘발성 메모리는 예를 들면 상변화(phase-change) 메모리, 저항성(resistive) 메모리(memory), 자기저항(magnetoresistive) 메모리, 강유전체(ferroelectric) 메모리 또는 폴리머(polymer) 메모리를 포함할 수 있다.
- [0034] 어떤 실시예에서, 저장 장치(120)는 호스트 장치(110)에 탈착 가능하도록 연결될 수 있다. 어떤 실시예에서, 저장 장치(120)는 호스트 장치(110) 내에 통합되거나 호스트 장치(110)에 탈착 가능하지 않게 연결될 수 있다.
- [0035] 호스트 장치(110)는 저장 장치(120)로 명령을 제출(submit)하여 다양한 동작을 수행할 수 있다. 예를 들면, 호스트 장치(110)는 저장 장치(120)로 읽기 명령을 제출하여 저장 장치(120)로부터 저장 장치(120)로부터 데이터를 읽거나, 저장 장치(120)로 쓰기 명령을 제출하여 저장 장치(120)에 데이터를 저장할 수 있다.
- [0036] 호스트 장치(110)가 저장 장치(120)로부터 데이터를 읽거나 쓰기 위해, 프로세서(111)는 명령(예를 들면, 읽기 명령 또는 쓰기 명령)을 생성하고, 호스트 메모리(112)의 제출 큐(submission queue, SQ)(112a)에 명령을 제출(또는 기입)할 수 있다. 어떤 실시예에서, 제출 큐(112a)는 NVMe 프로토콜을 지원할 수 있다. 장치 컨트롤러(121)는 호스트 메모리(112)의 제출 큐(112a)에 제출된 명령을 피크(peek)하고, 장치 컨트롤러(121)의 장치 리소스 상태에 기초해서 명령이 실행 가능한지를 판단하고, 명령이 실행 가능하다는 판단에 응답하여 명령을 페치(fetch)하여 명령을 실행할 수 있다. 어떤 실시예에서, 명령의 "피크"는 제출 큐로부터 명령을 장치 컨트롤러로 가져왔지만 해당 명령이 다시 가져올 상태로 제출 큐에 남아 있도록 하는 동작을 의미할 수 있다. 따라서, 장치 컨트롤러(121)는 제출 큐(112a)의 헤드 포인터 이동 없이 제출 큐(112a)로부터 명령을 가져옴으로써 명령을 피크하고, 명령의 피크 후에 제출 큐(112a)의 헤드 포인터를 이동함으로써 제출 큐(112a)로부터 명령을 페치(즉, 명령 페치 동작을 완료)할 수 있다. 장치 컨트롤러(121)는 읽기 명령을 실행함으로써 비휘발성 메모리(122)로부터 호스트 메모리(112)로 데이터를 전달하거나, 쓰기 명령을 실행함으로써 호스트 메모리(112)로부터 메모리 장치(122)로 데이터를 전달할 수 있다.
- [0037] 도 2는 다른 실시예에 따른 스토리지 시스템의 예시 블록도이다.
- [0038] 도 2를 참고하면, 스토리지 시스템(200)은 호스트 장치(210)와 저장 장치(220)를 포함할 수 있다.
- [0039] 호스트 장치(210)는 프로세서(211)와 호스트 메모리(212)를 포함할 수 있다. 저장 장치(220)는 장치 컨트롤러(221)와 비휘발성 메모리(222)를 포함할 수 있다. 도 1을 참고로 하여 설명한 실시예와 달리, 장치 컨트롤러(221)는 호스트 장치(210)로부터의 명령이 제출(또는 기입)될 제출 큐가 형성되는 메모리 버퍼(221a)를 더 포함할 수 있다. 어떤 실시예에서, 메모리 버퍼(221a)는 NVMe 프로토콜을 지원하는 컨트롤러 메모리 버퍼(controller memory buffer, CMB)일 수 있다.
- [0040] 호스트 장치(210)가 저장 장치(220)로부터 데이터를 읽거나 쓰기 위해, 프로세서(211)는 명령을 생성하고, 메모리 버퍼(221a)의 제출 큐에 명령을 제출할 수 있다. 장치 컨트롤러(221)는 메모리 버퍼(221a)의 제출 큐에 제출된 명령을 피크하고, 장치 컨트롤러(221)의 장치 리소스 상태에 기초해서 명령이 실행 가능한지를 판단하고, 명령이 실행 가능하다는 판단에 응답하여 명령을 페치하여 명령을 실행할 수 있다. 장치 컨트롤러(221)는 제출 큐의 헤드 포인터 이동 없이 제출 큐로부터 명령을 피크하고, 제출 큐의 헤드 포인터를 이동하여 제출 큐로부터

명령을 폐치할 수 있다. 장치 컨트롤러(221)는 읽기 명령을 실행함으로써 메모리 장치(222)로부터 호스트 메모리(212)로 데이터를 전달하거나, 쓰기 명령을 실행함으로써 호스트 메모리(212)로부터 메모리 장치(222)로 데이터를 전달할 수 있다.

[0041] 도 3은 한 실시예에 따른 저장 장치의 장치 컨트롤러의 예시 블록도이며, 도 4는 한 실시예에 따른 장치 컨트롤러의 장치 리소스를 예시하는 도면이고, 도 5 및 도 6은 도 3에 도시한 장치 컨트롤러의 동작의 한 예를 설명하는 도면이다.

[0042] 도 3을 참고하면, 장치 컨트롤러(300)는 명령 폐치 로직(310), 명령 전처리 로직(320), 명령 처리 로직(330) 및 장치 리소스(340)를 포함할 수 있다.

[0043] 어떤 실시예에서, 장치 컨트롤러(300)는 메모리 버퍼(350)를 더 포함하고, 메모리 버퍼(350)는 복수의 제출 큐(351, 352, ..., 35n)를 포함할 수 있다. 어떤 실시예에서, 복수의 제출 큐(351 ~ 35n)는 호스트 장치의 호스트 메모리(예를 들면, 도 1의 112)에 형성될 수 있다. 복수의 제출 큐(351, 352, ..., 35n)를 각각 SQ1, SQ2, ..., SQn으로 표현할 수 있다. 호스트 장치는 명령을 생성하고, 생성한 명령을 제출 큐(35i)에 넣을 수 있다. 어떤 실시예에서, 호스트 장치는 명령을 제출 큐(35i)에 넣은 후에, 장치 컨트롤러(300)의 제출 큐(35i)에 대응하는 레지스터(예를 들면, SQ 테일 도어벨 레지스터)를 새로운 테일 포인터로 갱신할 수 있다.

[0044] 명령 폐치 로직(310)은 제출 큐(35i)로부터 명령을 피크할 수 있다. 제출 큐(35i)로부터 명령을 피크하는 경우, 명령 폐치 로직(310)은 제출 큐(35i)의 헤드 포인터를 이동하지 않을 수 있다. 명령 폐치 로직(310)은 피크한 명령을 장치 컨트롤러(300)의 버퍼 또는 장치 컨트롤러(300)를 포함하는 저장 장치의 버퍼에 임시로 저장할 수 있다. 어떤 실시예에서, 명령 폐치 로직(310)은 복수의 제출 큐(351 ~ 35n)에서 타겟 제출 큐(35i)를 선택하고, 선택한 제출 큐(35i)로부터 명령을 피크할 수 있다. 명령 폐치 로직(310)은 예를 들면 장치 컨트롤러(300)가 지원하는 중재(arbitration) 메커니즘에 기초해서 타겟 제출 큐(35i)를 선택할 수 있다. 어떤 실시예에서, 명령 폐치 로직(310)은 중재 버스트(arbitration burst) 크기와 제출 큐(35i)에서 이용할 수 있는 명령의 수에 기초해서 제출 큐(35i)로부터 피크할 명령의 수를 결정할 수 있다. 중재 버스트 크기는 제출 큐로부터 한번에 가져올 수 있는 명령의 최대 수를 의미할 수 있다. 명령 폐치 로직(310)은 예를 들면 중재 버스트 크기와 제출 큐(35i)에서 이용할 수 있는 명령의 수 중 적은 수를 제출 큐(35i)로부터 피크할 명령의 수로 결정할 수 있다. 제출 큐(35i)에서 이용할 수 있는 명령의 수는 제출 큐(35i)의 헤드(head)에서 테일(tail) 이전까지의 명령의 수에 해당할 수 있다. 헤드는 제출 큐(35i)에서 헤드 포인터가 지시하는 엔트리이고, 테일은 제출 큐(35i)에서 테일 포인터가 지시하는 엔트리일 수 있다. 따라서, 명령 폐치 로직(310)은 제출 큐(35i)의 헤드부터 결정한 수만큼의 명령을 피크할 수 있다.

[0045] 명령 전처리 로직(320)은 장치 리소스(340)의 상태에 기초해서 피크한 명령의 실행 가능성을 판단할 수 있다. 명령 전처리 로직(320)은 피크한 명령을 실행하기 위한 장치 리소스(340)가 사용 가능한지를 판단하여서 실행 가능성을 판단할 수 있다. 어떤 실시예에서, 장치 리소스(340)는 읽기 명령에 전용인 읽기 리소스(341) 및 쓰기 명령에 전용인 쓰기 리소스(342)를 포함할 수 있다. 장치 리소스(340)는 읽기 명령과 쓰기 명령이 공유하는 공유 리소스(343)를 더 포함할 수 있다.

[0046] 명령 전처리 로직(320)은 장치 리소스(340) 중에서 피크한 명령의 실행을 위한 리소스가 사용 가능하다고 판단한 경우 해당 명령이 실행 가능하다고 판단할 수 있다. 어떤 실시예에서, 피크한 명령의 입출력(input/output, I/O) 유형에 따라 리소스의 종류가 달라질 수 있으므로, 명령 전처리 로직(320)은 피크한 명령의 I/O 유형, 공유 리소스(343), 그리고 읽기 리소스(341)와 쓰기 리소스(342) 중 I/O 유형에 대응하는 리소스에 기초해서 피크한 명령의 처리를 위한 리소스가 사용 가능한지를 판단할 수 있다. 어떤 실시예에서, 피크한 명령의 크기에 따라 필요한 리소스의 양이 달라질 수 있으므로, 명령 전처리 로직(320)은 피크한 명령의 I/O 유형 및 크기에 기초해서 해당 명령의 처리를 위한 리소스가 사용 가능한지를 판단할 수 있다. 명령 전처리 로직(320)은 장치 리소스(340) 중에서 피크한 명령의 처리를 위한 리소스가 다른 I/O 처리를 위해 점유되어 사용할 수 없는 경우, 해당 명령이 실행 가능하지 않다고 판단할 수 있다. 예를 들면, 명령 전처리 로직(320)은 읽기 리소스(341) 또는 공유 리소스(343)가 점유되어 사용할 수 없는 경우 읽기 명령을 실행할 수 없다고 판단하고, 읽기 리소스(341)와 공유 리소스(343)가 사용 가능한 경우 읽기 명령을 실행할 수 있다고 판단할 수 있다. 또한, 명령 전처리 로직(320)은 쓰기 리소스(342) 또는 공유 리소스(343)가 점유되어 사용할 수 없는 경우 쓰기 명령을 실행할 수 없다고 판단하고, 쓰기 리소스(342)와 공유 리소스(343)가 사용 가능한 경우 쓰기 명령을 실행할 수 있다고 판단할 수 있다.

[0047] 피크한 명령이 실행 가능하다고 판단한 경우, 명령 전처리 로직(320)은 명령 폐치 로직(310)을 통해 해당 명령

을 폐치하고 명령 처리 로직(330)으로 폐치한 명령을 실행할 것을 지시할 수 있다. 명령 폐치 로직(310)은 제출 큐(35i)에서 헤드 포인터를 폐치한 명령(엔트리)의 수만큼 이동함으로써 제출 큐(35i)로부터 명령을 폐치할 수 있다. 이에 따라, 제출 큐(35i)의 헤드 포인터는 폐치한 명령의 다음 엔트리로 이동될 수 있다.

[0048] 명령 처리 로직(330)은 폐치한 명령에 장치 리소스(340)를 할당하고 폐치한 명령을 실행할 수 있다. 어떤 실시예에서, 명령 처리 로직(330)은 폐치한 명령을 명령 처리 파이프라인에 넣어 실행할 수 있다. 명령 처리 로직(330)은 폐치한 명령을 명령 처리 파이프라인의 명령 처리 큐에 넣고, 명령 처리 큐로부터 명령을 꺼내서 명령을 실행할 수 있다. 명령 처리 파이프라인은 장치 리소스(340)를 사용하여 명령 처리(실행)를 수행하는 일련의 처리 엔진을 포함할 수 있다.

[0049] 명령 전처리 로직(320)이 피크한 명령이 실행 가능하지 않다고 판단한 경우, 명령 처리 로직(330)은 피크한 명령을 폐치하지 않을 수 있다. 따라서, 제출 큐(35i)의 헤드 포인터는 이동되지 않고 유지될 수 있다. 명령 전처리 로직(320)은 명령 처리 로직(330)으로 명령을 실행할 것을 지시하지 않고, 피크한 명령을 폐치하지 않고 드롭(drop)할 수 있다. 따라서, 명령 처리 로직(330)은 피크한 명령에 리소스를 할당하지 않고, 피크한 명령을 명령 처리 파이프라인에 넣지 않을 수 있다. 또한, 명령 전처리 로직(320)은 제출 큐(35i)를 리소스 대기 상태로 지정할 수 있다. 어떤 실시예에서, 리소스 대기 상태는 대기 중인 리소스를 지시하는 정보를 포함할 수 있다. 대기 중인 리소스를 지시하는 정보는 예를 들면 대기 중인 리소스의 종류와 대기 중인 리소스의 양을 포함할 수 있다.

[0050] 타겟 제출 큐(35i)를 처리한 후에, 장치 컨트롤러(300)는 복수의 제출 큐(351 ~ 35n)에서 다음 타겟 제출 큐(35j)를 선택하고, 타겟 제출 큐(35j)에서 명령을 피크하고 피크한 명령이 실행 가능한지를 판단하는 과정을 반복할 수 있다.

[0051] 어떤 실시예에서, 명령 전처리 로직(320)은 장치 리소스(340)의 상태를 모니터링하고, 장치 리소스(340)의 상태에 기초해서 대기 상태로 지정된 제출 큐(35i)가 처리 가능한지를 판단할 수 있다. 어떤 실시예에서, 명령 전처리 로직(320)은 대기 상태로 지정된 제출 큐(35i)의 리소스가 사용 가능한 상태로 되는 경우 제출 큐(35i)가 처리 가능하다고 판단할 수 있다. 제출 큐(35i)가 처리 가능하다고 판단한 경우, 명령 전처리 로직(320)은 제출 큐(35i)를 사용 가능한 제출 큐(35i)로 지정할 수 있다. 이에 따라, 앞서 설명한 것처럼, 제출 큐(35i)에 명령을 피크하여 실행 가능한지를 판단하는 동작이 수행될 수 있다. 어떤 실시예에서, 대기 상태로 지정된 제출 큐(35i)의 리소스가 사용 가능한 상태로 된 경우, 명령 전처리 로직(320)은 제출 큐(35i)에서 명령을 피크하는 동작을 수행하지 않고, 제출 큐(35i)에서 명령을 폐치하고 명령 처리 로직(330)에 명령을 실행할 것을 지시할 수도 있다. 어떤 실시예에서, 리소스가 사용 가능한 상태로 된 경우, 명령 전처리 로직(320)은 대기 상태로 지정된 제출 큐(35i)를 다른 제출 큐에 비해 우선하여 처리할 수 있다. 예를 들면, 명령 전처리 로직(320)은 대기 상태로 지정된 제출 큐(35i)를 사용 가능한 제출 큐의 리스트의 헤드에 추가할 수 있다.

[0052] 어떤 실시예에서, 명령 처리 로직(330)이 특정 I/O 유형의 명령의 실행을 완료하는 경우, 해당 I/O 유형의 리소스가 사용 가능한 상태로 될 수 있다. 명령 처리 로직(330)이 명령 처리 파이프라인에서 특정 유형의 처리를 완료하는 경우, 해당 유형의 리소스가 사용 가능한 상태로 될 수 있다. 예를 들면, 명령 처리 로직(330)이 읽기 명령의 실행을 완료하는 경우 읽기 리소스(341)와 공유 리소스(343)가 사용 가능한 상태로 되고, 쓰기 명령의 실행을 완료하는 경우 쓰기 리소스(342)와 공유 리소스(343)가 사용 가능한 상태로 될 수 있다.

[0053] 도 4를 참고하면, 어떤 실시예에서, 읽기 리소스(341), 쓰기 리소스(342) 및 공유 리소스(343)는 각각 다양한 유형의 리소스를 포함할 수 있다. 읽기 리소스(341)는 읽기 버퍼(341a) 또는 직접 메모리 액세스(direct memory access, DMA) 읽기 디스크립터(descriptor)(341b) 중 적어도 하나를 포함하고, 쓰기 리소스(342)는 쓰기 버퍼(342a) 또는 DMA 쓰기 디스크립터(342b) 중 적어도 하나를 포함할 수 있다.

[0054] 읽기 버퍼(341a)는 비휘발성 메모리(예를 들면, 도 1의 122 또는 도 2의 222)에서 읽은 데이터를 임시로 저장하고, DMA 읽기 버퍼(341b)는 저장 장치에서 호스트 메모리(예를 들면, 도 1의 112 또는 도 2의 212)로의 DMA 전송과 관련된 정보를 저장할 수 있다. 쓰기 버퍼(342a)는 비휘발성 메모리(122 또는 222)에 기입될 데이터를 임시로 저장하고, DMA 쓰기 디스크립터(342b)는 호스트 메모리(112 또는 212)로부터 저장 장치로의 DMA 전송과 관련된 정보를 저장할 수 있다.

[0055] 공유 리소스(343)는 복수의 유형의 공유 리소스를 포함하며, 복수의 유형의 공유 리소스는 어드레스 변환 캐시(address translation cache, ATC) 리소스(343a), 호스트 연결 리소스(343b), 태그 리소스(343c), 채널 리소스(343d), 코어 리소스(343e), 어드레스 변환 리소스(343f), 물리적 영역 페이지(physical region page, PRP)/분

산 수집 리스트(scatter gather list, SGL) 버퍼(343g) 또는 명령 완료 리소스(343h) 중 적어도 하나를 포함할 수 있다.

[0056] 어드레스 변환 캐시(343a)는 명령에서 파싱된 PRP 또는 SGL에 의해 참조되는 가상 어드레스를 호스트 메모리(112 또는 212)의 물리적 어드레스로 변환할 때 사용되는 어드레스 변환 데이터를 저장할 수 있다. 호스트 연결 리소스(343b)는 호스트와의 연결에 사용되는 인터페이스(예를 들면, PCIe(peripheral component interconnect express) 인터페이스) 관련 리소스를 포함할 수 있다. 태그 리소스(343c)는 명령 처리 파이프라인 내에서 명령의 유일한 식별을 위해 할당되는 리소스(태그)를 포함할 수 있다. 채널 리소스(343d)는 장치 컨트롤러(300)와 비휘발성 메모리(122 또는 222) 사이의 채널 관련 리소스를 포함할 수 있다. 코어 리소스(343e)는 명령 처리 로직에서 명령의 처리를 위한 프로세싱 코어 리소스를 포함할 수 있다. 어드레스 변환 리소스(343f)는 명령에서 파싱된 논리적 어드레스(예를 들면, LBA(logical block address))를 비휘발성 메모리(122 또는 222)의 물리적 어드레스(예를 들면, PPN(physical page number)로 변환하는 리소스를 포함할 수 있다. PRP/SGL 버퍼(343g)는 명령에서 파싱된 PRP 또는 SGL을 임시로 저장할 수 있다. 명령 완료 리소스(343h)는 명령의 완료 처리를 하기 위한 리소스를 포함할 수 있다.

[0057] 어떤 명령을 실행하기 위해 다양한 유형의 리소스가 필요한 경우, 일부 유형의 리소스가 사용 가능하지만 다른 일부 유형의 리소스가 사용 가능하지 않은 경우, 명령 전처리 로직(320)은 해당 명령이 실행 가능하지 않다고 판단하고, 해당 명령을 폐기하지 않을 수 있다. 또한, 복수의 유형의 공유 리소스(343a-343h) 중에서 일부 유형의 공유 리소스(예를 들면, 태그 리소스(343c))는 사용 가능하지만, 다른 일부 유형의 공유 리소스(예를 들면, 어드레스 변환 리소스(343f))가 사용 가능하지 않은 경우, 명령 전처리 로직(320)은 공유 리소스(343)가 사용 가능하지 않다고 판단할 수 있다.

[0058] 예를 들면, 도 5 및 도 6에 도시한 것처럼, 제출 큐(SQ1)에 두 개의 쓰기 명령과 두 개의 읽기 명령이 입력되고, 제출 큐(SQ2)에 네 개의 쓰기 명령과 한 개의 읽기 명령이 입력되고, 제출 큐(SQ3)에 네 개의 읽기 명령과 한 개의 쓰기 명령이 입력되고, 제출 큐(SQ1, SQ2, SQ3)에서 헤드 포인터는 첫 번째 엔트리를 포인트하고, 중재 버스트 크기는 4인 것으로 가정한다. 도 5 및 도 6에서 쓰기 명령을 "W"로, 읽기 명령을 "R"로 표시한다. 또한, 장치 리소스(340) 중에서 쓰기 리소스(342)가 소진되어 사용 가능하지 않은 상태인 것으로 가정한다.

[0059] 이 경우, 전형적인 장치 컨트롤러는 제출 큐(SQ1)에서 쓰기 명령을 폐기하고, 폐기한 쓰기 명령의 실행을 위한 쓰기 리소스(342)가 사용 가능해질 때까지 대기할 수 있다. 또한, 장치 컨트롤러는 쓰기 리소스가 사용 가능해질 때까지 공유 리소스(343)를 차단할 수 있다. 장치 컨트롤러는 쓰기 리소스(342)가 사용 가능해지면 제출 큐(SQ1)에서 폐기한 쓰기 명령 및 읽기 명령을 실행한 후에, 다음 제출 큐(SQ2)에서 명령을 폐기할 수 있다. 따라서, 리소스가 사용 가능해질 때까지 모든 명령의 실행을 기다리므로, 대기 시간이 길어질 수 있다. 또한, 쓰기 명령이 대기 중인 상태에서도 쓰기 명령은 공유 리소스(343)를 점유하므로, 읽기 리소스(341)가 사용 가능한 상태에서 읽기 명령이 폐기되더라도 공유 리소스(343)가 사용 가능해질 때까지 읽기 명령이 실행될 수 없을 수 있다.

[0060] 어떤 실시예에서, 도 5에 도시한 것처럼, 명령 폐치 로직(310)은 복수의 제출 큐(SQ1, SQ2, SQ3) 중에서 타겟 제출 큐(SQ1)를 선택하고, 제출 큐(SQ1)로부터 두 개의 쓰기 명령과 두 개의 읽기 명령을 피크할 수 있다. 명령 전처리 로직(320)은 쓰기 명령의 실행을 위한 쓰기 리소스(342)가 사용 가능하지 않은 상태이므로, 제출 큐(SQ1)를 리소스 대기 상태로 지정할 수 있다. 이 경우, 제출 큐(SQ1)는 명령이 피크된 상태이므로, 제출 큐(SQ1)의 헤드 포인터는 이동하지 않고 유지될 수 있다. 명령 폐치 로직(310)은 다음 제출 큐(SQ2)로부터 네 개의 쓰기 명령을 피크할 수 있다. 마찬가지로, 명령 전처리 로직(320)은 쓰기 명령의 실행을 위한 쓰기 리소스(342)가 사용 가능하지 않은 상태이므로, 제출 큐(SQ2)를 리소스 대기 상태로 지정할 수 있다. 이 경우, 제출 큐(SQ2)도 명령이 피크된 상태이므로, 제출 큐(SQ2)의 헤드 포인터는 이동하지 않고 유지될 수 있다.

[0061] 도 6에 도시한 것처럼, 명령 전처리 로직(320)은 다음 제출 큐(SQ3)로부터 네 개의 읽기 명령을 피크할 수 있다. 읽기 명령의 실행을 위한 읽기 리소스(341)와 공유 리소스(343)가 사용 가능한 상태이므로, 명령 폐치 로직(310)은 읽기 명령을 폐기하고, 명령 처리 로직(330)은 폐기한 읽기 명령을 실행할 수 있다. 이 경우, 제출 큐(SQ3)의 명령은 폐기되었으므로, 제출 큐(SQ3)의 헤드 포인터가 이동될 수 있다. 따라서, 점유 중인 리소스(쓰기 리소스(342))의 해제를 기다리지 않고 즉시 실행 가능한 명령을 먼저 실행할 수 있으므로, 대기 시간을 줄이고, 이에 따라 명령 처리 효율성을 높일 수 있다.

[0062] 도 7은 한 실시예에 따른 저장 장치의 장치 컨트롤러의 동작을 예시하는 흐름도이다.

- [0063] 도 7을 참고하면, 장치 컨트롤러는 사용 가능한 제출 큐가 존재하는 경우(S730), 사용 가능한 제출 큐에서 타겟 제출 큐를 선택할 수 있다(S735). 어떤 실시예에서, 사용 가능한 제출 큐는 대기 상태로 표시되지 않은 제출 큐일 수 있다. 사용 가능한 제출 큐가 존재하지 않는 경우(S730), 장치 컨트롤러는 동작을 종료할 수 있다. 타겟 제출 큐에 명령이 존재하는 경우(S740), 장치 컨트롤러는 타겟 제출 큐에서 다음 명령을 피크할 수 있다(S745). 장치 컨트롤러는 타겟 제출 큐에서 명령을 피크하고, 헤드 포인터를 이동하지 않고 유지할 수 있다(S745). 어떤 실시예에서, 장치 컨트롤러는 중재 버스트 크기와 타겟 제출 큐에서 이용할 수 있는 명령의 수 중 최소 수에 해당하는 명령을 타겟 제출 큐로부터 피크할 수 있다(S745).
- [0064] 장치 컨트롤러는 장치 리소스의 상태에 기초해서 피크한 명령을 실행 가능성을 판단할 수 있다(S750). 어떤 실시예에서, 장치 컨트롤러는 장치 리소스의 상태, 명령의 I/O 유형 및 명령의 크기에 기초해서 명령의 실행 가능성을 판단할 수 있다(S750). 명령이 실행 가능하다고 판단한 경우(S750), 장치 컨트롤러는 해당 명령을 폐지하고, 명령 처리 파이프라인(또는 명령 처리 큐)에 넣어 명령을 실행할 수 있다(S755). 장치 컨트롤러는 타겟 제출 큐에서 헤드 포인터를 이동하여 명령을 폐지할 수 있다(S755). 다음, 장치 컨트롤러는 타겟 제출 큐에 다음 명령이 존재하는지를 다시 확인할 수 있다(S740).
- [0065] 명령이 실행 가능하지 않다고 판단한 경우(S750), 장치 컨트롤러는 타겟 제출 큐를 리소스 대기 상태로 표시하고(S760), 사용 가능한 제출 큐가 존재하는지를 확인할 수 있다(S730). 어떤 실시예에서, 리소스 대기 상태는 대기 중인 리소스를 지시하는 정보를 포함할 수 있다. 대기 중인 리소스를 지시하는 정보는 예를 들면 대기 중인 리소스의 종류 또는 대기 중인 리소스의 양 중 적어도 하나를 포함할 수 있다.
- [0066] 타겟 제출 큐에 명령이 존재하지 않는 경우(S740), 장치 컨트롤러는 사용 가능한 제출 큐에서 다음 타겟 제출 큐를 선택하고(S730, S735), S740 내지 S760을 참고로 하여 설명한 동작을 반복할 수 있다.
- [0067] 어떤 실시예에서, 대기 상태로 표시된 제출 큐를 처리하기 위해, 장치 컨트롤러는 대기 상태로 지정된 제출 큐가 존재하는지를 확인할 수 있다(S710). 대기 상태로 지정된 제출 큐가 존재하지 않는 경우(S710), 장치 컨트롤러는 사용 가능한 제출 큐 중에서 타겟 제출 큐를 선택하고(S730, S735), S740 내지 S760을 참고로 하여 설명한 동작을 반복할 수 있다.
- [0068] 대기 상태로 지정된 제출 큐가 존재하는 경우(S710), 장치 컨트롤러는 대기 상태로 지정된 제출 큐 중에서 제출 큐를 선택할 수 있다(S715). 장치 컨트롤러는 장치 리소스의 상태에 기초해서 선택한 제출 큐의 대기 상태를 해제할 수 있는지를 판단할 수 있다(S720). 어떤 실시예에서, 장치 컨트롤러는 장치 리소스의 상태와 대기 상태의 정보에 기초해서 선택한 제출 큐의 대기 상태를 해제할 수 있는지를 판단할 수 있다(S720). 예를 들면, 대기 중인 리소스의 종류와 대기 중인 리소스의 양에 해당하는 리소스가 장치 리소스에서 사용 가능한 경우, 장치 컨트롤러는 선택한 제출 큐의 대기 상태를 해제할 수 있다. 대기 상태를 해제할 수 없는 경우, 장치 컨트롤러는 대기 상태로 지정된 제출 큐가 추가로 존재하는지를 확인할 수 있다(S710).
- [0069] 대기 상태를 해제할 수 있는 경우, 장치 컨트롤러는 선택한 제출 큐를 사용 가능한 제출 큐의 리스트에 추가할 수 있다(S725). 어떤 실시예에서, 장치 컨트롤러는 선택한 제출 큐가 중재 메커니즘에 따라 먼저 선택될 수 있도록 사용 가능한 제출 큐의 리스트의 헤드에 추가할 수 있다. 다음, 장치 컨트롤러는 대기 상태로 표시된 제출 큐가 추가로 존재하는지를 확인하고(S710), S715 내지 S725를 참고로 하여 설명한 동작 또는 S730 내지 S760을 참고로 하여 설명한 동작을 반복할 수 있다.
- [0070] 다음, 어드레스 변환 리소스를 공유 리소스로 사용하는 실시예에 대해서 도 8을 참고로 하여 설명한다.
- [0071] 도 8은 다른 실시예에 따른 저장 장치의 장치 컨트롤러의 예시 블록도이다.
- [0072] 도 8을 참고하면, 장치 컨트롤러(800)는 명령 폐치 로직(810), 명령 전처리 로직(820), 명령 처리 로직(830) 및 어드레스 변환 회로(840)를 포함할 수 있다.
- [0073] 어떤 실시예에서, 장치 컨트롤러(800)는 메모리 버퍼(850)를 더 포함하고, 메모리 버퍼(850)는 복수의 제출 큐(851, 852, ..., 85n)를 포함할 수 있다. 어떤 실시예에서, 복수의 제출 큐(851 ~ 85n)은 호스트 장치의 호스트 메모리(예를 들면, 도 1의 112)에 형성될 수 있다.
- [0074] 어드레스 변환 회로(840)는 어드레스 변환 캐시(841) 및 제어 로직(842)를 포함할 수 있다. 어드레스 변환 캐시(841)는 복수의 엔트리를 갖고, 각 엔트리는 가상 어드레스를 호스트 메모리(예를 들면, 도 1의 112 또는 도 2의 212)의 물리적 어드레스로 변환할 때 사용되는 어드레스 변환 데이터를 저장할 수 있다. 가상 어드레스는 명령의 PRP 또는 SGL에 의해 참조되는 어드레스일 수 있다. 제어 로직(842)은 명령(즉, 명령의 가상 어드레스)이

어드레스 변환 캐시(841)에서 히트(hit)하는지를 확인할 수 있다. 명령이 히트인 경우, 제어 로직(842)은 히트한 엔트리에 기초해서 명령의 가상 어드레스를 물리적 어드레스로 변환할 수 있다. 명령이 어드레스 변환 캐시(841)에서 미스(miss)인 경우, 제어 로직(842)은 호스트 장치(예를 들면, 도 1의 110 또는 도 2의 210)로 명령의 가상 어드레스에 해당하는 물리적 어드레스를 요청할 수 있다. 어떤 실시예에서, 제어 로직(842)은 어드레스 변환 서비스(address translation service, ATS)/페이지 요청 인터페이스(page request interface, PRI)를 통해 물리적 어드레스를 요청할 수 있다.

[0075] 명령 페치 로직(810)은 복수의 제출 큐(851 ~ 85n)에서 타겟 제출 큐(85i)를 선택하고, 선택한 제출 큐(85i)로부터 명령을 피크할 수 있다.

[0076] 명령 전처리 로직(820)은 피크한 명령이 어드레스 변환 캐시(841)에서 히트인지에 기초해서 명령의 실행 가능성을 판단할 수 있다. 명령이 어드레스 변환 캐시(841)에서 히트인 경우, 명령 전처리 로직(820)은 해당 명령이 실행 가능하다고 판단하고, 명령 페치 로직(810)을 통해 해당 명령을 페치하고, 명령 처리 로직(830)으로 해당 명령을 실행할 것을 지시할 수 있다. 명령 처리 로직(830)은 해당 명령을 명령 처리 파이프라인에 넣고, 어드레스 변환 회로(840)를 통해 명령의 가상 어드레스를 물리적 어드레스로 변환하여 명령을 실행할 수 있다. 피크한 명령이 어드레스 변환 캐시(841)에서 히트이더라도, 피크한 명령을 실행하기 위한 다른 리소스가 부족한 경우에 명령 전처리 로직(820)은 해당 명령이 실행 가능하지 않다고 판단할 수 있다.

[0077] 피크한 명령이 어드레스 변환 캐시(841)에서 미스인 경우, 어드레스 변환 회로(840)의 제어 로직(842)은 피크한 명령의 가상 어드레스에 해당하는 물리적 어드레스를 호스트 장치로 요청하고, 명령 전처리 로직(820)은 제출 큐(85i)를 리소스 대기 상태로 지정할 수 있다. 제어 로직(842)이 호스트 장치로부터 물리적 어드레스를 수신하여 어드레스 변환 캐시(841)가 사용 가능해지는 경우, 제출 큐(85i)의 대기 상태를 해제하고 제출 큐(85i)의 명령을 실행할 수 있다.

[0078] 도 9는 또 다른 실시예에 따른 저장 장치의 장치 컨트롤러를 예시하는 도면이며, 도 10은 도 9에 도시한 장치 컨트롤러의 동작의 한 예를 설명하는 도면이다.

[0079] 도 9를 참고하면, 장치 컨트롤러(900)는 명령 페치 로직(910), 명령 전처리 로직(920), 명령 처리 로직(930), 장치 리소스(940) 및 펜딩 명령 큐(pending command queue)(960)를 포함할 수 있다.

[0080] 어떤 실시예에서, 장치 컨트롤러(900)는 메모리 버퍼(950)를 더 포함하고, 메모리 버퍼(950)는 복수의 제출 큐(951, 952, ..., 95n)를 포함할 수 있다. 어떤 실시예에서, 복수의 제출 큐(951 ~ 95n)는 호스트 장치의 호스트 메모리(예를 들면, 도 1의 112)에 형성될 수 있다.

[0081] 어떤 실시예에서, 펜딩 명령 큐(960)는 실행 가능하지 않은 읽기 명령을 저장하는 펜딩 읽기 명령 큐(961)과 실행 가능하지 않은 쓰기 명령을 저장하는 펜딩 쓰기 명령 큐(962)를 포함할 수 있다. 어떤 실시예에서, 펜딩 명령 큐의 크기는 중재 버스트 크기, 제출 큐(951 ~ 95n)의 개수 및 장치 컨트롤러(900)에서 동시에 실행할 수 있는 명령의 최대 개수에 기초해서 설정될 수 있다.

[0082] 명령 페치 로직(910)은 제출 큐(95i)로부터 명령을 피크할 수 있다. 제출 큐(95i)로부터 명령을 피크하는 경우, 명령 페치 로직(910)은 제출 큐(95i)의 헤드 포인터를 이동하지 않을 수 있다. 어떤 실시예에서, 명령 페치 로직(910)은 복수의 제출 큐(951 ~ 95n)에서 타겟 제출 큐(95i)를 선택하고, 선택한 제출 큐(95i)로부터 명령을 피크할 수 있다.

[0083] 명령 전처리 로직(920)은 장치 리소스(940)의 상태에 기초해서 피크한 명령의 실행 가능성을 판단할 수 있다. 어떤 실시예에서, 장치 리소스(940)는 읽기 명령을 처리하기 위한 읽기 리소스(941) 및 쓰기 명령을 처리하기 위한 쓰기 리소스(942)를 포함할 수 있다. 장치 리소스(940)는 읽기 명령과 쓰기 명령이 공유하는 공유 리소스(943)를 더 포함할 수 있다.

[0084] 명령 전처리 로직(920)은 피크한 명령이 실행 가능하다고 판단한 경우, 명령 페치 로직(910)을 통해 해당 명령을 페치하고, 명령 처리 로직(930)으로 페치한 명령을 실행할 것을 지시할 수 있다. 명령 페치 로직(910)은 제출 큐(95i)의 헤드 포인터를 이동함으로써 제출 큐(95i)로부터 명령을 페치할 수 있다. 명령 처리 로직(930)은 페치한 명령에 장치 리소스(940)를 할당하고 페치한 명령을 실행할 수 있다. 피크한 명령이 실행 가능하지 않다고 판단한 경우, 명령 전처리 로직(920)은 실행 가능하지 않은 명령을 리소스 할당 없이 펜딩 명령 큐(960)에 넣고, 명령 페치 로직(910)을 통해 해당 명령을 페치할 수 있다. 즉, 실행 가능하지 않은 명령을 펜딩 명령 큐(960)로 이동함에 따라, 명령 페치 로직(910)은 제출 큐(95i)에서 헤드 포인터를 이동할 수 있다.

- [0085] 어떤 실시예에서, 명령 전처리 로직(920)은 피크한 명령 중 일부 명령이 실행 가능하고 나머지 일부 명령이 실행 가능하지 않다고 판단한 경우, 명령 처리 로직(930)으로 실행 가능한 명령을 실행할 것을 지시하고, 실행 가능하지 않은 명령을 펜딩 명령 큐(960)에 넣을 수 있다. 예를 들면, 명령 폐치 로직(910)이 제출 큐(95i)에서 피크한 명령이 읽기 명령과 쓰기 명령을 포함하고, 읽기 리소스(941)는 사용 가능하지만 쓰기 리소스(942)가 소진된 경우, 명령 전처리 로직(920)은 쓰기 명령을 펜딩 쓰기 명령 큐(962)에 넣고, 명령 처리 로직(930)은 읽기 명령을 실행할 수 있다. 펜딩 명령 큐(960)가 꽉 찬(full) 경우, 명령 전처리 로직(920)은 피크한 명령을 펜딩 명령 큐(960)에 넣지 않고 드롭할 수 있다. 어떤 실시예에서, 피크한 명령을 드롭한 경우, 명령 전처리 로직(920)은 도 3 내지 도 8을 참고로 하여 설명한 것처럼, 제출 큐(95i)를 리소스 대기 상태로 지정할 수 있다.
- [0086] 타겟 제출 큐(95i)를 처리한 후에, 장치 컨트롤러(900)는 복수의 제출 큐(951 ~ 95n)에서 다음 타겟 제출 큐(95j)를 선택하고, 선택한 제출 큐(95j)에서 명령을 폐치하고 폐치한 명령이 실행 가능한지를 판단하는 과정을 반복할 수 있다.
- [0087] 어떤 실시예에서, 명령 전처리 로직(920)은 장치 리소스(940)의 상태를 모니터링하고, 명령 펜딩 큐(960)에 저장된 명령을 실행하기 위한 리소스가 사용 가능한 상태로 된 경우, 명령 펜딩 큐(960)에 저장된 명령을 폐치하고, 명령 처리 로직(930)에 해당 명령을 실행할 것을 지시할 수 있다. 어떤 실시예에서, 명령 전처리 로직(920)은 명령 펜딩 큐(960)에 저장된 명령을 피크하고, 피크한 명령이 실행 가능한지를 판단하는 과정을 수행할 수 도 있다.
- [0088] 예를 들면, 도 10에 도시한 것처럼, 제출 큐(SQ1)에 두 개의 쓰기 명령(W0, W1)과 한 개의 읽기 명령(R0)이 입력되고, 제출 큐(SQ2)에 한 개의 읽기 명령(R1)과 두 개의 쓰기 명령(W2, W3)이 입력되고, 제출 큐(SQ3)에 두 개의 쓰기 명령(W4, W5)과 한 개의 읽기 명령(R2)이 입력되고, 제출 큐(SQ1, SQ2, SQ3)에서 헤드 포인터는 첫 번째 엔트리를 포인트하고, 중재 버스트는 3인 것으로 가정한다. 또한, 장치 리소스(940) 중에서 쓰기 리소스(942)가 소진되어 사용 가능하지 않은 상태인 것으로 가정한다.
- [0089] 명령 폐치 로직(910)은 복수의 제출 큐(SQ1, SQ2, SQ3) 중에서 타겟 제출 큐(SQ1)를 선택하고, 제출 큐(SQ1)로부터 명령(W0, W1, R0)을 폐치할 수 있다. 명령 전처리 로직(920)은 쓰기 리소스(942)가 사용 가능하지 않은 상태이므로 쓰기 명령(W0, W1)을 펜딩 쓰기 명령 큐(962)에 넣을 수 있다. 또한 읽기 리소스(941)과 공유 리소스(943)가 사용 가능한 상태이므로, 명령 전처리 로직(920)은 명령 처리 로직(930)으로 읽기 명령(R0)을 실행할 것을 지시하고, 명령 처리 로직(930)은 읽기 명령(R0)을 실행할 수 있다.
- [0090] 명령 폐치 로직(910)은 다음 타겟 제출 큐(SQ2)를 선택하고, 제출 큐(SQ2)로부터 명령(R1, W2, W3)을 폐치할 수 있다. 명령 전처리 로직(920)은 명령 처리 로직(930)으로 읽기 명령(R1)을 실행할 것을 지시하고, 명령 처리 로직(930)은 읽기 명령(R1)을 실행할 수 있다. 또한, 명령 전처리 로직(920)은 쓰기 리소스(942)가 사용 가능하지 않은 상태이므로 쓰기 명령(W2, W3)을 펜딩 쓰기 명령 큐(962)에 넣을 수 있다.
- [0091] 명령 폐치 로직(910)은 다음 타겟 제출 큐(SQ3)를 선택하고, 제출 큐(SQ3)로부터 명령(W4, W5, R2)을 폐치할 수 있다. 명령 전처리 로직(920)은 쓰기 리소스(942)가 사용 가능하지 않은 상태이므로 쓰기 명령(W4, W5)을 펜딩 쓰기 명령 큐(962)에 넣고, 명령 처리 로직(930)으로 읽기 명령(R2)을 실행할 것을 지시할 수 있다.
- [0092] 쓰기 리소스(942)가 사용 가능한 상태로 되는 경우, 명령 전처리 로직(920)은 펜딩 쓰기 명령 큐(962)에서 쓰기 명령(W0-W5)을 폐치하고, 명령 처리 로직(930)으로 쓰기 명령(W0-W5)을 실행할 것을 지시할 수 있다.
- [0093] 도 11은 또 다른 실시예에 따른 저장 장치의 장치 컨트롤러의 동작을 예시하는 흐름도이다.
- [0094] 도 11을 참고하면, 장치 컨트롤러는 사용 가능한 제출 큐가 존재하는 경우(S1130), 사용 가능한 제출 큐에서 타겟 제출 큐를 선택할 수 있다(S1135). 사용 가능한 제출 큐가 존재하지 않는 경우(S1130), 장치 컨트롤러는 동작을 종료할 수 있다. 타겟 제출 큐에 명령이 존재하는 경우(S1140), 장치 컨트롤러는 타겟 제출 큐에서 다음 명령을 피크할 수 있다(S1145). 장치 컨트롤러는 타겟 제출 큐에서 명령을 피크하고, 헤드 포인터를 이동하지 않고 유지할 수 있다(S1145).
- [0095] 장치 컨트롤러는 장치 리소스의 상태를 포함하는 정보에 기초해서 피크한 명령의 실행 가능성을 판단할 수 있다(S1150). 피크한 명령 중 실행 가능한 명령이 존재하는 경우(S1150), 장치 컨트롤러는 해당 명령을 폐치하고, 명령 처리 파이프라인(또는 명령 처리 큐)에 넣어 명령을 실행할 수 있다(S1155). 장치 컨트롤러는 타겟 제출 큐에서 헤드 포인터를 이동하여 명령을 폐치할 수 있다(S1155). 폐치한 명령 중 실행 가능하지 않은 명령이 존재하는 경우(S1150), 장치 컨트롤러는 실행 가능하지 않은 명령을 펜딩 명령 큐에 넣을 수 있다(S1160). 어떤

실시예에서, 장치 컨트롤러는 실행 가능하지 않은 명령을 펜딩 명령 큐에 넣고, 해당 명령을 제출 큐에서 폐치할 수 있다(S1160). 장치 컨트롤러는 타겟 제출 큐에서 헤드 포인터를 이동하여 명령을 폐치할 수 있다(S1160).

[0096] 장치 컨트롤러는 타겟 제출 큐에 다음 명령이 존재하는지를 확인할 수 있다(S1140). 타겟 제출 큐에 다음 명령이 존재하는 경우(S1140), 장치 컨트롤러는 S1145 내지 S1160을 참고로 하여 설명한 동작을 반복할 수 있다.

[0097] 타겟 제출 큐에 명령이 존재하지 않는 경우(S1140), 장치 컨트롤러는 사용 가능한 제출 큐에서 다음 타겟 제출 큐를 선택하고(S1130, S1135), S1140 내지 S1160을 참고로 하여 설명한 동작을 반복할 수 있다.

[0098] 어떤 실시예에서, 펜딩 명령 큐를 처리하기 위해, 장치 컨트롤러는 펜딩 명령 큐에 명령이 존재하는지를 확인할 수 있다(S1110). 펜딩 명령 큐에 명령이 존재하지 않는 경우(S1110), 장치 컨트롤러는 사용 가능한 제출 큐 중에서 타겟 제출 큐를 선택하고(S1130, S1135), S1140 내지 S1160을 참고로 하여 설명한 동작을 반복할 수 있다.

[0099] 펜딩 명령 큐에 명령이 존재하는 경우(S1110), 장치 컨트롤러는 펜딩 명령 큐의 명령을 실행할 수 있는지를 판단할 수 있다(S1120). 어떤 실시예에서, 장치 컨트롤러는 장치 리소스의 상태와 펜딩 명령 큐의 I/O 유형에 기초해서 펜딩 명령 큐의 명령을 실행할 수 있는지를 판단할 수 있다(S1120). 예를 들면, 장치 컨트롤러는 공유 리소스와 읽기 리소스가 사용 가능한 경우 펜딩 읽기 명령 큐의 명령을 실행할 수 있다고 판단하고, 공유 리소스와 쓰기 리소스가 사용 가능한 경우 펜딩 쓰기 명령 큐의 명령을 실행할 수 있다고 판단할 수 있다. 펜딩 명령 큐의 명령을 실행할 수 있는 경우, 장치 컨트롤러는 펜딩 명령 큐의 명령을 폐치해서 실행할 수 있다(S1125).

[0100] 펜딩 명령 큐의 명령을 실행할 수 없는 경우(S1120), 장치 컨트롤러는 사용 가능한 제출 큐 중에서 타겟 제출 큐를 선택하고(S1130, S1135), S1140 내지 S1160을 참고로 하여 설명한 동작을 반복할 수 있다.

[0101] 도 12는 또 다른 실시예에 따른 저장 장치의 장치 컨트롤러를 예시하는 도면이며, 도 13은 도 12에 도시한 장치 컨트롤러의 동작의 한 예를 설명하는 도면이다.

[0102] 도 12를 참고하면, 장치 컨트롤러(1200)는 명령 폐치 로직(1210), 명령 전처리 로직(1220), 명령 처리 로직(1230) 및 장치 리소스(1240)를 포함할 수 있다.

[0103] 어떤 실시예에서, 장치 컨트롤러(1200)는 메모리 버퍼(1250)를 더 포함하고, 메모리 버퍼(1250)는 복수의 제출 큐(1251, 1252, ..., 125n)를 포함할 수 있다. 어떤 실시예에서, 복수의 제출 큐(1251 ~ 125n)는 호스트 장치의 호스트 메모리(예를 들면, 도 1의 112)에 형성될 수 있다.

[0104] 명령 폐치 로직(1210)은 제출 큐(125i)로부터 명령을 피크할 수 있다. 제출 큐(125i)로부터 명령을 피크하는 경우, 명령 폐치 로직(1210)은 제출 큐(125i)의 헤드 포인터를 이동하지 않을 수 있다. 어떤 실시예에서, 명령 폐치 로직(1210)은 복수의 제출 큐(1251 ~ 125n)에서 타겟 제출 큐(125i)를 선택하고, 선택한 제출 큐(125i)로부터 명령을 피크할 수 있다.

[0105] 명령 전처리 로직(1220)은 장치 리소스(1240)의 상태에 기초해서 피크한 명령의 실행 가능성을 판단할 수 있다. 어떤 실시예에서, 장치 리소스(1240)는 읽기 명령을 처리하기 위한 읽기 리소스(1241) 및 쓰기 명령을 처리하기 위한 쓰기 리소스(1242)를 포함할 수 있다. 장치 리소스(1240)는 읽기 명령과 쓰기 명령이 공유하는 공유 리소스(1243)를 더 포함할 수 있다.

[0106] 명령 전처리 로직(1220)은 피크한 명령이 실행 가능하다고 판단한 경우, 명령 처리 로직(1230)으로 피크한 명령을 폐치하고, 폐치한 명령을 실행할 것을 지시할 수 있다. 명령 폐치 로직(1210)은 제출 큐(125i)의 헤드 포인터를 이동함으로써 제출 큐(125i)로부터 명령을 폐치할 수 있다. 명령 처리 로직(1230)은 폐치한 명령에 장치 리소스(1240)를 할당하고 폐치한 명령을 실행할 수 있다. 어떤 실시예에서, 피크한 명령이 실행 가능하지 않다고 판단한 경우, 명령 전처리 로직(1220)은 명령 폐치 로직(1210)에 실행 가능하지 않은 명령을 제출 큐(125i)로 다시 푸시(push)할 것을 지시할 수 있다. 어떤 실시예에서, 명령 폐치 로직(1210)은 제출 큐(125i)의 헤드 포인터를 조작하여 실행 가능하지 않은 명령을 제출 큐(125i)에서 패킹(packing)할 수 있다. 이에 따라, 제출 큐(125i)에서는 순서에서 벗어난(out of order) 명령 처리가 수행될 수 있다.

[0107] 어떤 실시예에서, 명령 폐치 로직(1210)은 제출 큐(125i)의 헤드부터 테일까지 명령을 차례로 피크하고, 명령 전처리 로직(1220)은 피크한 명령이 실행 가능한지를 판단할 수 있다. 피크한 명령 중 일부 명령이 실행 가능하고 나머지 일부 명령이 실행 가능하지 않다고 판단한 경우, 명령 전처리 로직(1220)은 명령 처리 로직(1230)으로 실행 가능한 명령을 실행할 것을 지시하고, 실행 가능하지 않은 명령을 제출 큐(125i)에서 다시 패킹할 수 있다. 명령 폐치 로직(1210)이 제출 큐(125i)에서 피크한 명령이 읽기 명령과 쓰기 명령을 포함하고, 읽기 리소

스(1241)는 사용 가능하지만 쓰기 리소스(1242)가 소진된 경우, 명령 처리 로직(1230)은 읽기 명령을 실행할 수 있다. 명령 페치 로직(1210)은 제출 큐(125i)의 헤드 포인터를 페치된 읽기 명령의 수만큼 이동하고, 이동한 헤드 포인터가 지시하는 헤드 엔트리부터 쓰기 명령을 패키징할 수 있다.

[0108] 어떤 실시예에서, 제출 큐(125i)가 호스트 메모리(112)에 형성된 경우, 명령 페치 로직(1210)은 호스트 장치로 제출 큐(125i)의 쓰기 권한을 요청하고, 호스트 장치로부터 쓰기 권한을 받은 후에 제출 큐(125i)의 명령을 패키징할 수 있다.

[0109] 어떤 실시예에서, 명령 전처리 로직(1220)은 명령을 패키징한 제출 큐(125i)를 도 3 내지 도 7을 참고로 하여 설명한 것처럼 리소스 대기 상태로 지정할 수 있다. 어떤 실시예에서, 명령 전처리 로직(1220)은 패키징한 제출 큐(125i)가 대기하는 리소스 정보를 표시할 수 있다. 리소스 정보는 예를 들면 대기하는 리소스의 종류 또는 대기하는 리소스의 양 중 적어도 하나를 포함할 수 있다.

[0110] 도 13에 도시한 것처럼, 예를 들면, 제출 큐(SQ)에 쓰기 명령(W0, W1), 읽기 명령(R0), 쓰기 명령(W2), 읽기 명령(R1) 및 쓰기 명령(W3) 순으로 명령이 입력된 것으로 가정한다. 또한, 장치 리소스(1240) 중에서 쓰기 리소스(1242)가 소진되어 사용 가능하지 않은 상태인 것으로 가정한다.

[0111] 명령 페치 로직(1210)은 제출 큐(SQ)에서 명령을 차례로 피크하고, 읽기 리소스(1241)과 공유 리소스(1243)가 사용 가능한 상태이므로, 읽기 명령(R0, R1)을 페치할 수 있다. 명령 처리 로직(1230)은 페치한 읽기 명령(R0, R1)을 리소스를 할당하고 실행할 수 있다. 또한, 명령 페치 로직(1210)은 두 개의 읽기 명령(R0, R1)을 페치하였으므로, 제출 큐(SQ)의 헤드 포인터를 h에서 (h-2)로 이동하고, 헤드 포인터 (h-2)가 지시하는 제출 큐(SQ)의 엔트리부터 쓰기 명령(W0, W1, W2, W3)을 다시 패키징할 수 있다.

[0112] 어떤 실시예에서, 명령 전처리 로직(1220)은 장치 리소스(1240)의 상태를 모니터링하고, 명령이 다시 패키징된 제출 큐(125i)의 리소스가 사용 가능한 상태로 되는지를 확인할 수 있다. 리소스가 사용 가능한 상태로 된 경우, 명령 전처리 로직(1220)은 제출 큐(125i)를 사용 가능한 제출 큐로 지정할 수 있다. 이에 따라, 앞서 설명한 것처럼, 제출 큐(125i)에 명령을 피크하여 실행 가능한지를 판단하는 동작이 수행될 수 있다. 어떤 실시예에서, 리소스가 사용 가능한 상태로 된 경우, 명령 전처리 로직(1220)은 명령이 다시 패키징된 제출 큐(125i)를 다른 제출 큐에 비해 우선하여 처리할 수 있다. 예를 들면, 명령 전처리 로직(1220)은 명령이 다시 패키징된 제출 큐(125i)를 사용 가능한 제출 큐의 리스트의 헤드에 추가할 수 있다.

[0113] 어떤 실시예에서, 명령 전처리 로직(1220)은 제출 큐(125i)의 명령을 처리한 후에, 복수의 제출 큐(1251 ~ 125n)에서 다음 타겟 제출 큐(125j)를 선택하고, 선택한 제출 큐(125j)에서 명령을 피크하고 피크한 명령이 실행 가능한지를 판단하는 과정을 반복할 수 있다.

[0114] 도 14는 또 다른 실시예에 따른 저장 장치의 장치 컨트롤러의 동작을 예시하는 흐름도이다.

[0115] 도 14를 참고하면, 장치 컨트롤러는 사용 가능한 제출 큐가 존재하는 경우(S1430), 사용 가능한 제출 큐에서 타겟 제출 큐를 선택할 수 있다(S1435). 사용 가능한 제출 큐가 존재하지 않는 경우(S1430), 장치 컨트롤러는 동작을 종료할 수 있다. 타겟 제출 큐에 명령이 존재하는 경우(S1440), 장치 컨트롤러는 타겟 제출 큐에서 다음 명령을 피크할 수 있다(S1445). 장치 컨트롤러는 타겟 제출 큐에서 명령을 피크하고, 헤드 포인터를 이동하지 않고 유지할 수 있다(S1445).

[0116] 장치 컨트롤러는 장치 리소스의 상태를 포함하는 정보에 기초해서 피크한 명령을 실행 가능성을 판단할 수 있다(S1450). 피크한 명령이 실행 가능하지 않다고 판단한 경우(S1450), 장치 컨트롤러는 타겟 제출 큐에 다음 명령이 존재하는지를 다시 확인할 수 있다(S1440). 피크한 명령이 실행 가능하다고 판단한 경우(S1450), 장치 컨트롤러는 해당 명령을 페치하고, 명령 처리 파이프라인(또는 명령 처리 큐)에 넣고 명령을 실행할 수 있다(S1455). 다음, 장치 컨트롤러는 타겟 제출 큐에 다음 명령이 존재하는지를 다시 확인할 수 있다(S1440).

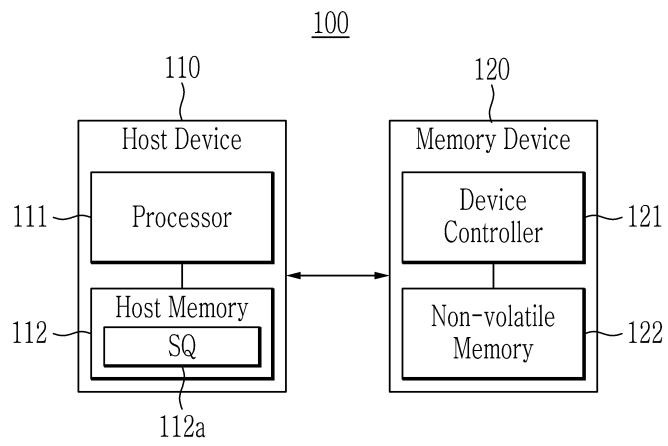
[0117] 타겟 제출 큐의 테일까지 피크한 후(S1440), 장치 컨트롤러는 타겟 제출 큐에서 실행 가능하지 않은 명령이 존재하는지를 확인할 수 있다(S1460). 실행 가능하지 않은 명령이 존재하지 않으면, 장치 컨트롤러는 사용 가능한 제출 큐에서 다음 타겟 제출 큐를 선택할 수 있다(S1430, S1435). 실행 가능하지 않은 명령이 존재하면, 장치 컨트롤러는 실행 가능하지 않은 명령을 타겟 제출 큐에서 패키징할 수 있다(S1465). 장치 컨트롤러는 타겟 제출 큐의 헤드 포인터를 페치된 명령의 수만큼 이동하고, 이동한 헤드 포인터가 지시하는 헤드 엔트리부터 실행 가능하지 않은 명령을 패키징할 수 있다. 어떤 실시예에서, 장치 컨트롤러는 패키징한 제출 큐를 리소스 대기 상태의 제출 큐로 표시할 수 있다. 다음, 장치 컨트롤러는 사용 가능한 제출 큐에서 다음 타겟 제출 큐를 선택하고

(S1430, S1435), S1440 내지 S1465을 참고로 하여 설명한 동작을 반복할 수 있다.

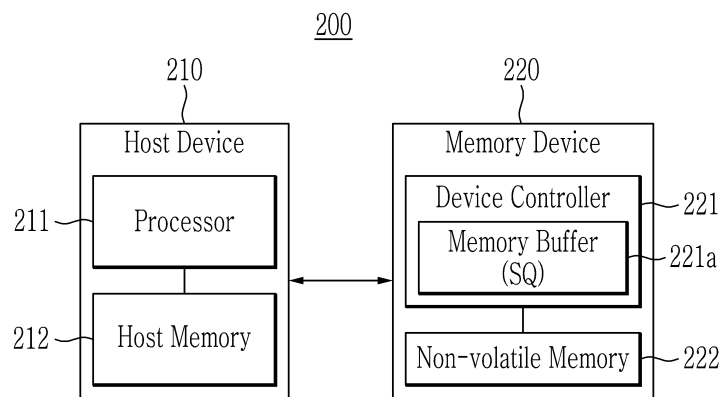
- [0118] 어떤 실시예에서, 패킹된 제출 큐를 우선해서 처리하기 위해, 장치 컨트롤러는 패킹된 제출 큐가 존재하는지를 확인할 수 있다(S1410). 어떤 실시예에서, 장치 컨트롤러는 리소스 대기 상태의 제출 큐가 존재하는지를 확인할 수 있다(S1410). 패킹된 제출 큐가 존재하지 않는 경우(S1410), 장치 컨트롤러는 사용 가능한 제출 큐 중에서 제출 큐를 선택하고(S1430, S1435), S1440 내지 S1465를 참고로 하여 설명한 동작을 반복할 수 있다.
- [0119] 패킹된 제출 큐가 존재하는 경우(S1410), 장치 컨트롤러는 패킹된 제출 큐 중에서 제출 큐를 선택할 수 있다(S1415). 장치 컨트롤러는 장치 리소스의 상태에 기초해서 선택한 제출 큐를 처리할 수 있는지를 판단할 수 있다(S1420). 어떤 실시예에서, 장치 컨트롤러는 장치 리소스의 상태와 선택한 제출 큐가 대기하는 리소스의 정보에 기초해서 선택한 제출 큐를 처리할 수 있는지를 판단할 수 있다(S1420). 예를 들면, 대기하는 리소스의 종류와 양에 해당하는 리소스가 장치 리소스에서 사용 가능한 경우, 장치 컨트롤러는 선택한 제출 큐를 처리할 수 있다고 판단할 수 있다. 선택한 제출 큐를 처리할 수 없는 경우, 장치 컨트롤러는 패킹된 제출 큐가 추가로 존재하는지를 확인할 수 있다(S1410).
- [0120] 선택한 제출 큐를 처리할 수 있는 경우, 장치 컨트롤러는 선택한 제출 큐를 사용 가능한 제출 큐의 리스트에 추가할 수 있다(S1425). 어떤 실시예에서, 장치 컨트롤러는 선택한 제출 큐가 중재 메커니즘에 따라 먼저 선택될 수 있도록 사용 가능한 제출 큐의 리스트의 헤드에 추가할 수 있다. 선택한 제출 큐를 처리할 수 없는 경우, 장치 컨트롤러는 패킹된 제출 큐가 추가로 존재하는지를 확인하고(S1410), S1415 내지 S1425를 참고로 하여 설명한 동작 또는 S1430 내지 S1465를 참고로 하여 설명한 동작을 반복할 수 있다.
- [0121] 도 15는 한 실시예에 따른 컴퓨터 시스템을 나타내는 예시 블록도이다.
- [0122] 도 15를 참고하면, 컴퓨팅 장치(1500)는 프로세서(1510), 메모리(1520), 메모리 컨트롤러(1530), 저장 장치(1540), 통신 인터페이스(1550) 및 버스(1560)를 포함한다. 컴퓨팅 장치(1500)는 다른 범용적인 구성 요소를 더 포함할 수 있다.
- [0123] 프로세서(1510)는 컴퓨팅 장치(1500)의 각 구성의 전반적인 동작을 제어한다. 프로세서(1510)는 CPU(central processing unit), AP(application processor), GPU(graphic processing unit) 등의 다양한 프로세싱 유닛 중 적어도 하나로 구현될 수 있다.
- [0124] 메모리(1520)는 각종 데이터 및 명령을 저장한다. 메모리 컨트롤러(1530)는 메모리(1520)로의 및 메모리(1520)로부터의 데이터 또는 명령의 전달을 제어한다. 어떤 실시예에서, 메모리 컨트롤러(1530)는 프로세서(1510)와 별도의 칩으로 제공될 수 있다. 어떤 실시예에서, 메모리 컨트롤러(1530)는 프로세서(1510)의 내부 구성으로 제공될 수 있다.
- [0125] 저장 장치(1540)는 프로그램 및 데이터를 비임시적으로 저장한다. 어떤 실시예에서, 저장 장치(1540)는 도 1 내지 도 14를 참고로 하여 설명한 저장 장치로 구현될 수 있다. 통신 인터페이스(1550)는 컴퓨팅 장치(1500)의 유무선 인터넷 통신을 지원한다. 또한, 통신 인터페이스(1550)는 인터넷 통신 외의 다양한 통신 방식을 지원할 수도 있다. 버스(1560)는 컴퓨팅 장치(1500)의 구성 요소간 통신 기능을 제공한다. 버스(1560)는 구성 요소간의 통신 프로토콜에 따라 적어도 하나의 유형의 버스를 포함할 수 있다.
- [0126] 어떤 실시예에서, 도 1 내지 도 14를 참고로 하여 설명한 각 로직 또는 둘 이상의 로직의 조합은 디지털 회로, 프로그램 가능한 또는 프로그램할 수 없는 로직 장치 또는 어레이, 응용 주문형 집적 회로(application specific integrated circuit, ASIC) 등으로 구현될 수 있다.
- [0127] 이상에서 본 발명의 실시예에 대하여 상세하게 설명하였지만 본 발명의 권리범위는 이에 한정되는 것은 아니고 다음의 청구범위에서 정의하고 있는 본 발명의 기본 개념을 이용한 당업자의 여러 변형 및 개량 형태 또한 본 발명의 권리범위에 속하는 것이다.

도면

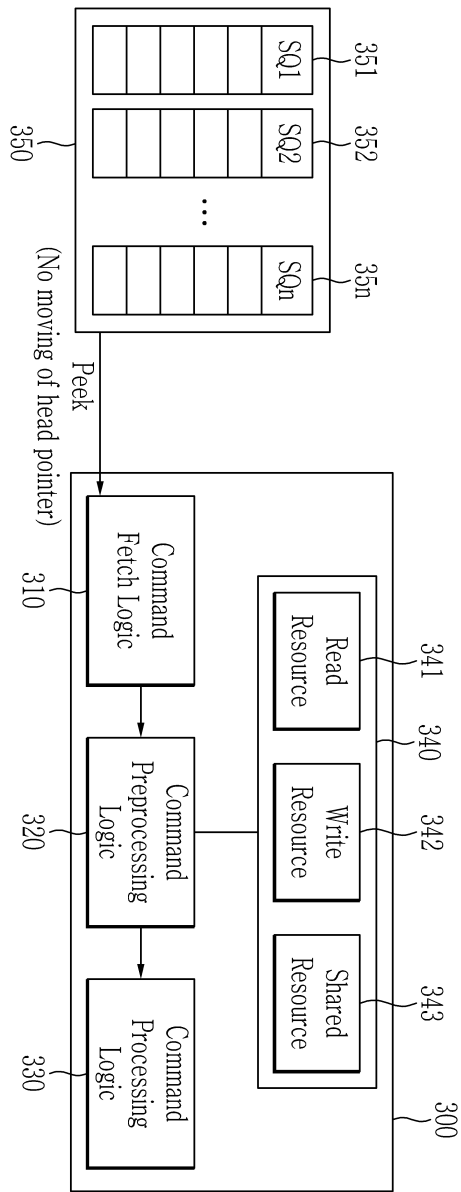
도면1



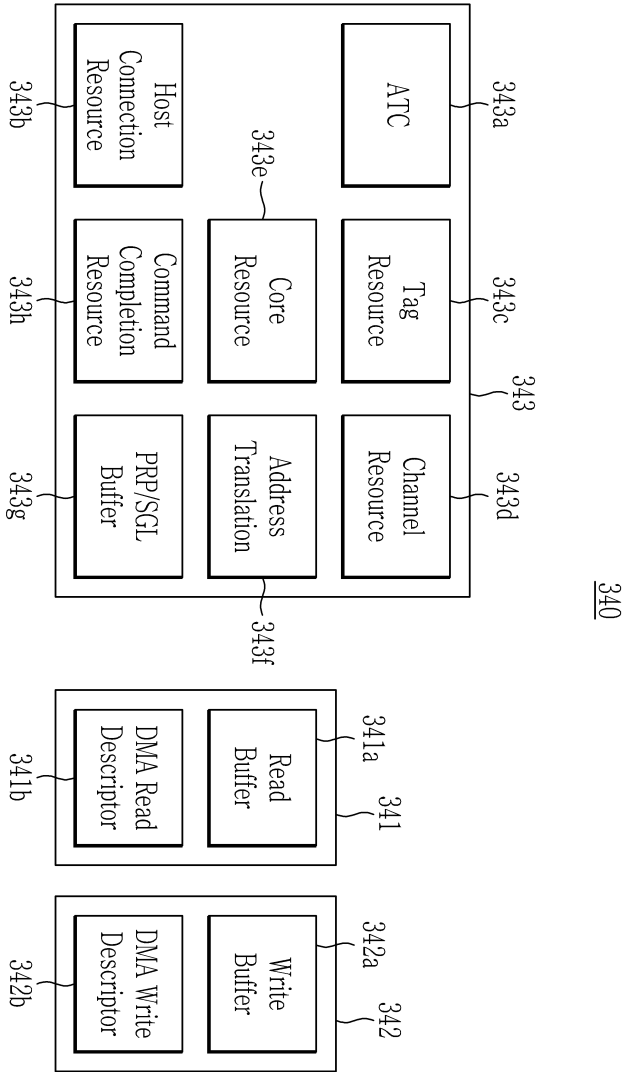
도면2



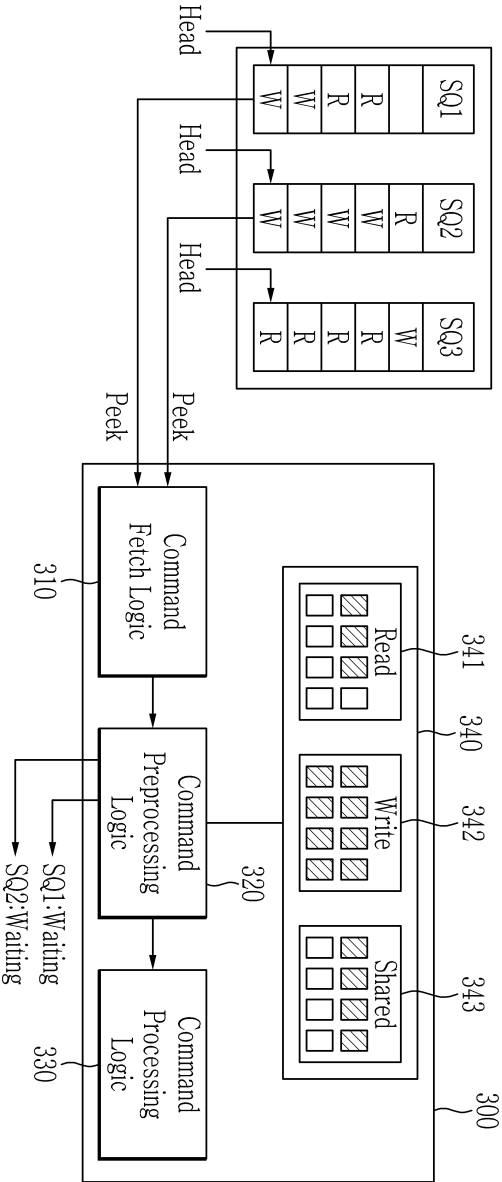
도면3



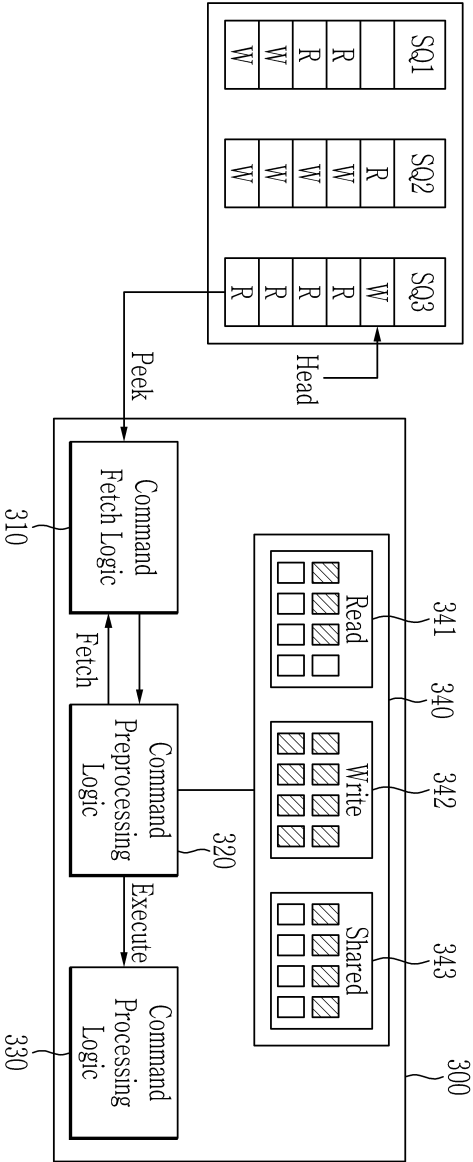
도면4



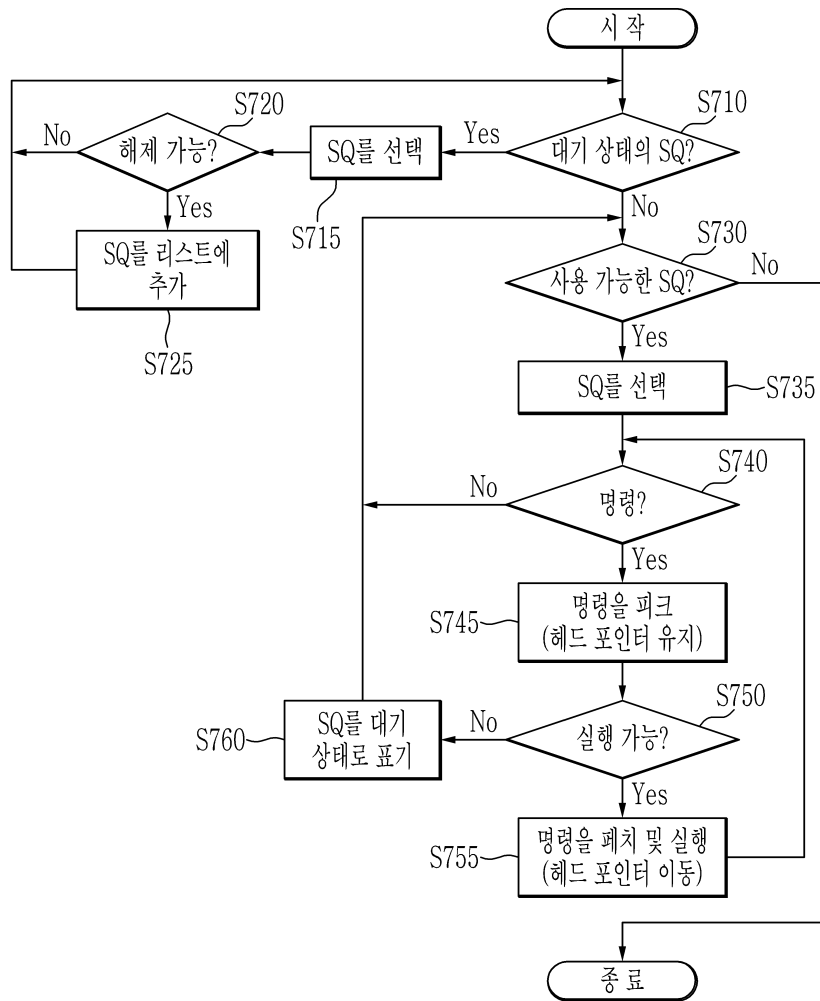
도면5



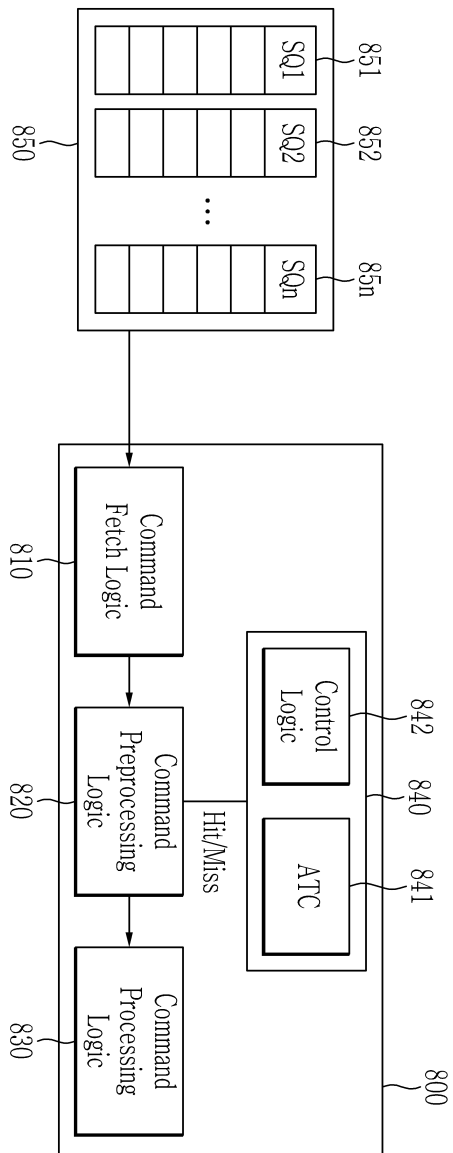
도면6



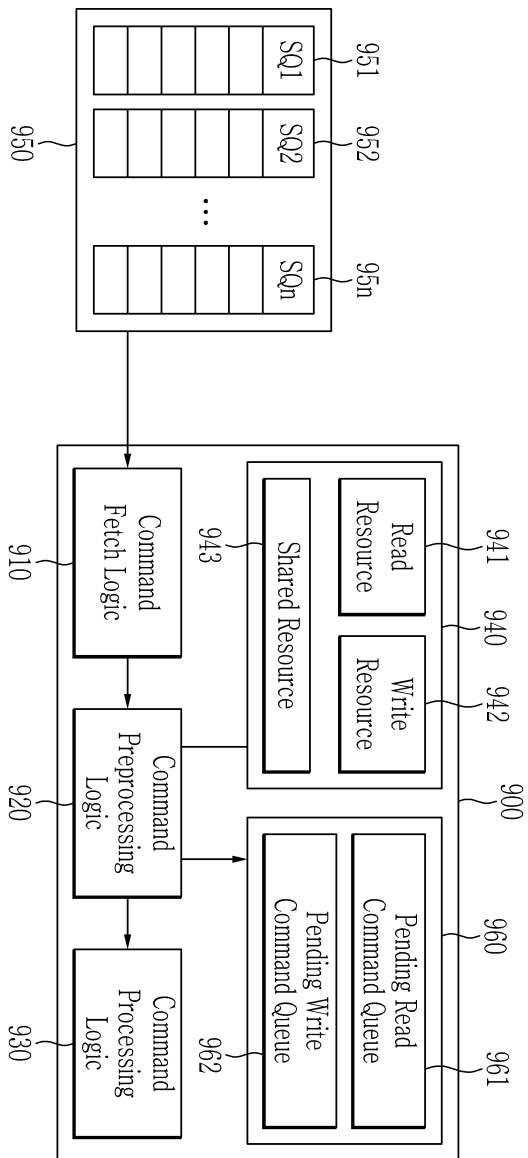
도면7



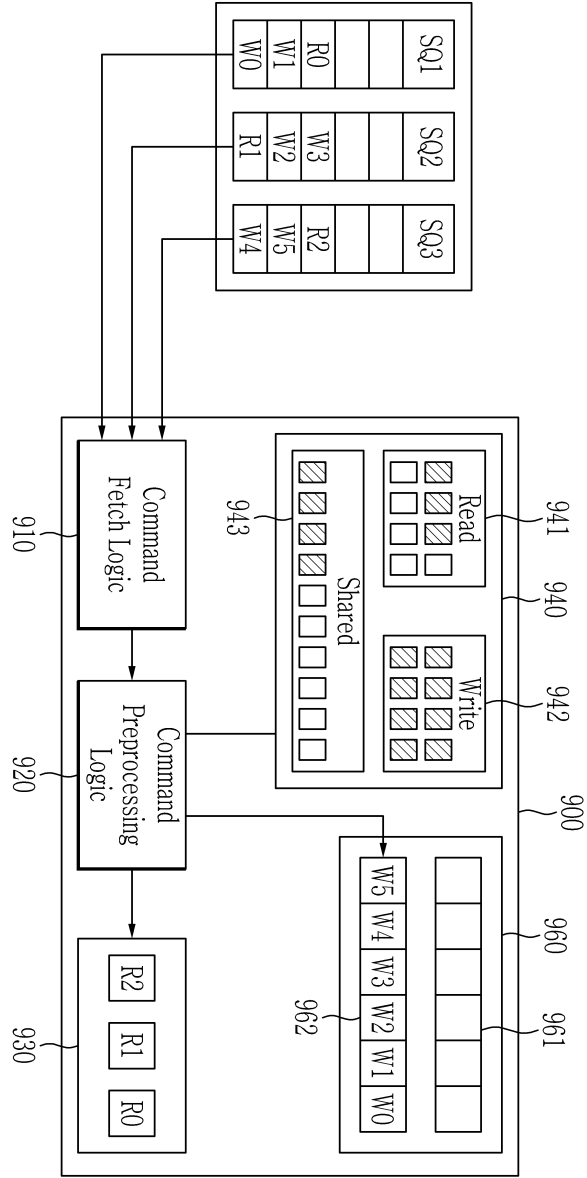
도면8



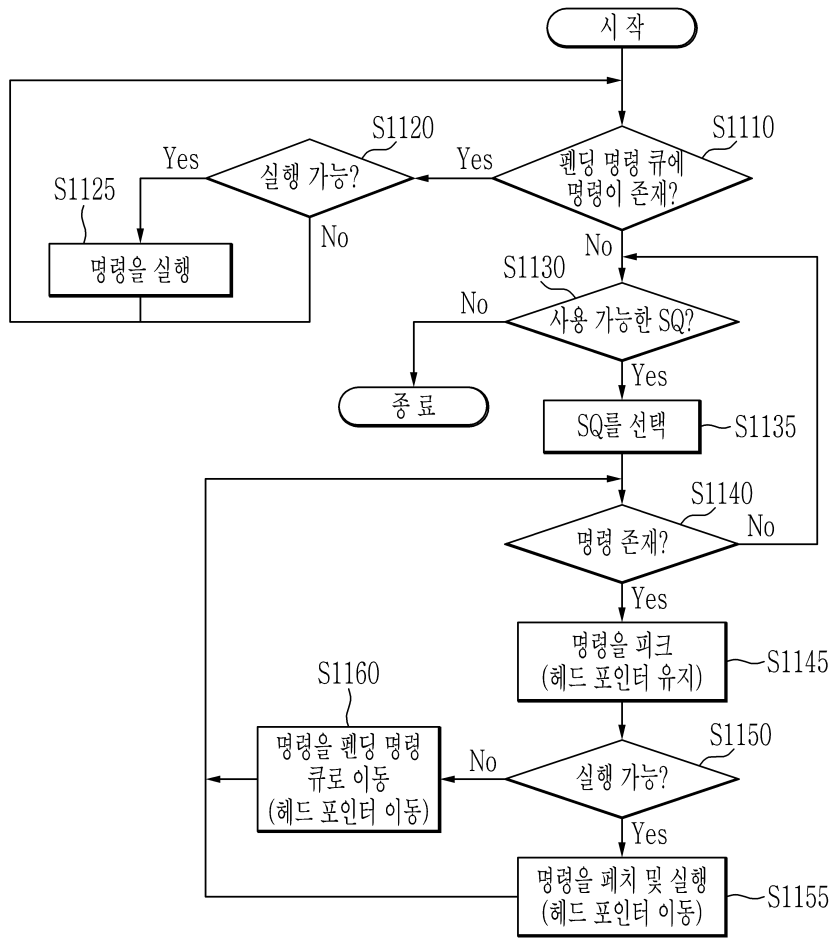
도면9



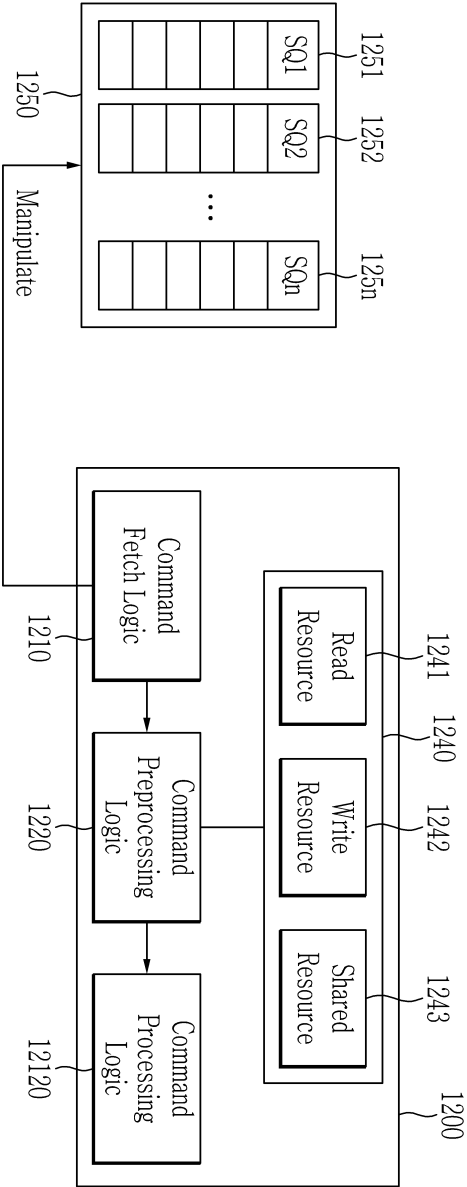
도면10



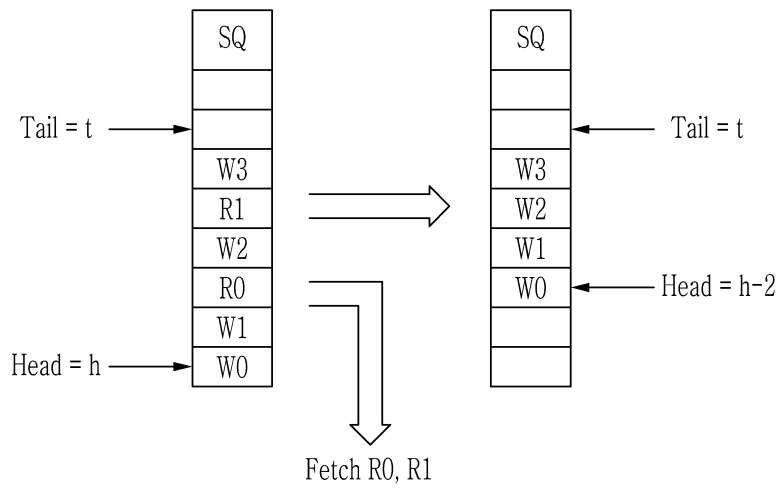
도면11



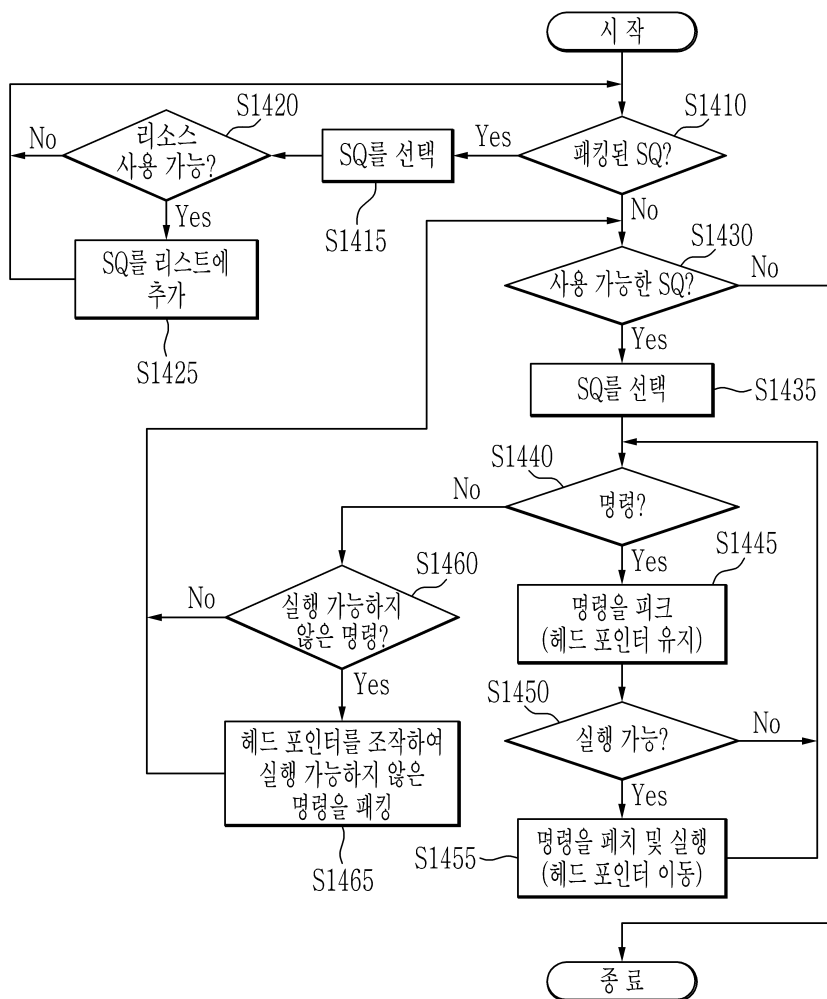
도면12



도면13



도면14



도면15

