



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2016년04월26일
(11) 등록번호 10-1612155
(24) 등록일자 2016년04월06일

(51) 국제특허분류(Int. Cl.)
G06F 13/14 (2006.01)
(21) 출원번호 10-2013-0022241
(22) 출원일자 2013년02월28일
심사청구일자 2016년01월20일
(65) 공개번호 10-2014-0076463
(43) 공개일자 2014년06월20일
(30) 우선권주장
13/711,885 2012년12월12일 미국(US)
(56) 선행기술조사문헌
US20100268904 A1
US20120042101 A1
US20070088928 A1
US20120089753 A1

(73) 특허권자
아바코 테크놀로지스 제너럴 아이피 (싱가포르)
퍼티이 리미티드
싱가포르 768923 이순 애비뉴 7 넘버 1
(72) 발명자
와이너 아담
미국 네바다주 89052 헨더슨 스모키 스카이 드라이브 2291
쉬필드 로버트 엘
미국 콜로라도주 80503 롱몬트 튜린 드라이브 1640
(뒷면에 계속)
(74) 대리인
제일특허법인

전체 청구항 수 : 총 13 항

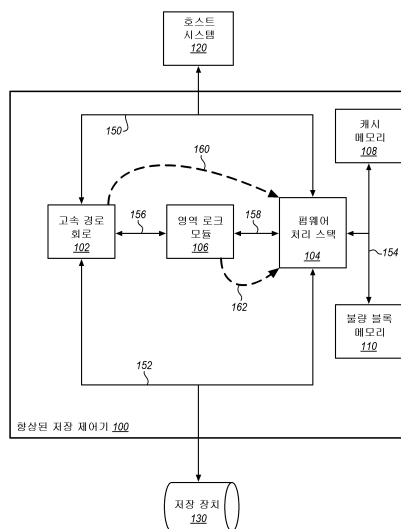
심사관 : 윤혜숙

(54) 발명의 명칭 다중 처리 스택을 구비한 저장 제어기의 I/O 요청을 우회시키기 위해 영역 로크를 사용하는 방법 및 구조

(57) 요약

제어기에서 부착된 호스트 시스템으로부터의 수신한 I/O 요청을 제어기에서 다수의 처리 스택들(102, 104) 중의 하나로 효율적으로 우회시키기 위하여 영역 로크를 사용하는 저장 제어기(100) 내 방법 및 구조를 개시한다. 제어기 내 영역 로크 모듈(106)은 각 처리 스택으로 하여금 저장 장치의 블록 주소 범위에 대한 영역 로크를 요청하게 한다. 우회형 로크 요청은 I/O 요청이 다중 처리 스택들 중의 특정한 일 스택으로 우회되어야 하는 블록 주소 범위를 식별하도록 설정될 수 있다.

대표도 - 도1



(72) 발명자

크리쉬나무르시 나빈

인도 560103 카르나타카 방갈로르 데바라비사나할리
리 블록 - 씨 마라타할리 아우터 링 로드 글로벌
테크놀로지 파크

순드라니 카필

인도 560103 카르나타카 방갈로르 데바라비사나할리
리 블록 - 씨 마라타할리 아우터 링 로드 글로벌
테크놀로지 파크

무르시 라지브 스리니바사

인도 560103 카르나타카 방갈로르 데바라비사나할리
리 블록 - 씨 마라타할리 아우터 링 로드 글로벌
테크놀로지 파크

나라야나무르시 어난드

인도 560103 카르나타카 방갈로르 데바라비사나할리
리 블록 - 씨 마라타할리 아우터 링 로드 글로벌
테크놀로지 파크

시미오네스쿠 호리아 크리스티안

미국 캘리포니아주 95131 산 호세 리더 파크 드라이브
1320

리조 제임스 에이

미국 텍사스주 78759 오스틴 에이에이유에스티 스위트
200 리아타 테라스 파크웨이 12331-비

명세서

청구범위

청구항 1

하나 이상의 호스트 시스템에 연결되도록 구성되는 저장 제어기(a storage controller)–상기 저장 제어기는 상기 저장 제어기와 연결된 하나 이상의 저장 장치 상의 데이터를 관리하도록 구성됨–로서,

부착된 호스트 시스템으로부터 수신한, 고속 경로 처리(fast path processing)를 위해 포맷된 I/O 요청을 처리하도록 구성된 고속 경로 회로(fast path circuit)와,

상기 저장 제어기의 프로세서 상에서 동작하며 임의의 포맷의 I/O 요청을 처리하도록 구성된 펌웨어 처리 스택(firmware processing stack)과,

상기 고속 경로 회로 및 상기 펌웨어 처리 스택과 연결되며, 상기 저장 제어기의 캐시 메모리 내의 지정된 블록 주소 범위를 로크(lock)하기 위해 상기 고속 경로 회로로부터의 로크 요청 신호를 처리하도록 구성되고, 현재 로크된 블록 주소 범위의 상태에 기초하여 수신한 고속 경로 포맷 I/O 요청이 상기 펌웨어 처리 스택으로 우회(divert)되어야 하는지 여부를 판정하도록 구성된 영역 로크 모듈(region lock module)을 포함하되,

상기 저장 제어기는 상기 고속 경로 포맷 I/O 요청의 처리가 상기 펌웨어 처리 스택으로 우회되어야 한다는 판정에 응답하여 처리를 위해 상기 고속 경로 포맷 I/O 요청을 상기 펌웨어 처리 스택으로 우회시키도록 구성되고,

상기 펌웨어 처리 스택은, 상기 블록 주소 범위의 블록에 액세스하려고 하는 고속 경로 포맷 I/O 요청이 상기 펌웨어 처리 스택에 의해 처리되어야 한다는 판정에 응답하여, 블록 주소 범위에 대한 우회형 영역 로크(divert-type region lock)를 설정하기 위해 상기 영역 로크 모듈로 로크 요청을 제공하도록 구성되고,

상기 영역 로크 모듈은, 상기 펌웨어 처리 스택으로부터의 상기 로크 요청의 수신에 응답하여, 블록 주소 범위에 대해 상기 우회형 영역 로크를 설정하도록 구성되고,

상기 저장 제어기는, 상기 고속 경로 포맷 I/O 요청의 고속 경로 회로 처리에 의해 영향받을 블록 주소 범위에 대해 상기 우회형 영역 로크가 설정되었음을 검출한 것에 응답하여, 수신한 고속 경로 포맷 I/O 요청을 상기 펌웨어 처리 스택으로 우회시키도록 더 구성되는

저장 제어기.

청구항 2

제1항에 있어서,

상기 캐시 메모리는 상기 펌웨어 처리 스택에 의한 기록 I/O 요청의 처리에 의해 영향받은 블록 주소에 대한 기록 데이터를 저장하도록 구성되며,

상기 펌웨어 처리 스택은, 상기 블록 주소 범위 내 블록 주소와 관련된 상기 캐시 메모리에 데이터를 저장하는 것에 응답하여, 블록 주소 범위에 대한 우회형 영역 로크 요청을 상기 영역 로크 모듈로 제공하도록 더 구성되는

저장 제어기.

청구항 3

제2항에 있어서,

상기 펌웨어 처리 스택은 상기 캐시 메모리를 라이트-백(write-back) 캐시 메모리로서 관리하는

저장 제어기.

청구항 4

제1항에 있어서,

상기 저장 장치의 잠재적 불량 블록(potentially bad blocks)의 블록 주소를 식별하는 불량 블록 정보를 저장하도록 구성되며, 상기 펌웨어 처리 스택과 연결된 불량 블록 메모리를 더 포함하되,

상기 펌웨어 처리 스택은, 블록 주소 범위 내 블록 주소를 식별하는 엔트리를 상기 불량 블록 메모리에 저장하는 것에 응답하여, 블록 주소 범위에 대한 우회형 영역 로크 요청을 상기 영역 로크 모듈로 제공하도록 더 구성되는

저장 제어기.

청구항 5

제1항에 있어서,

상기 고속 경로 회로는 상기 고속 경로 포맷 I/O 요청에 관한 상황 정보(context information)를 상기 영역 로크 모듈로 제공하고,

상기 영역 로크 모듈은 상기 상황 정보에 따른 처리를 위해 상기 고속 경로 포맷 I/O 요청을 상기 펌웨어 처리 스택으로 우회시키도록 더 구성되는

저장 제어기.

청구항 6

제1항에 있어서,

상기 고속 경로 회로는, 상기 고속 경로 포맷 I/O의 처리가 상기 펌웨어 처리 스택으로 우회되어야 한다는 것을 나타내는 상기 영역 로크 모듈로부터의 신호에 응답하여, 처리를 위해 상기 고속 경로 포맷 I/O 요청을 상기 펌웨어 처리 스택으로 우회시키도록 더 구성되는

저장 제어기.

청구항 7

하나 이상의 호스트 시스템 및 하나 이상의 저장 장치에 연결되도록 구성되며, 호스트 시스템으로부터 수신한 고속 경로 포맷 I/O 요청의 처리를 위한 고속 경로 회로 및 임의의 포맷의 I/O 요청을 처리하기 위한 펌웨어 처리 스택을 가지는 저장 제어기에서 동작될 수 있는 방법으로서,

상기 저장 제어기의 캐시 메모리 내의 블록 어드레스의 범위를 로크하기 위해 상기 고속 경로 회로로부터 로크 요청을 수신하는 단계와,

고속 경로 포맷 I/O 요청을 수신하는 단계와,

상기 저장 제어기의 영역 로크 모듈의 동작에 의해, 상기 캐시 메모리 내의 현재 로크된 블록 주소 범위의 상태에 기초하여 상기 수신한 고속 경로 포맷 I/O 요청이 상기 펌웨어 처리 스택으로 우회되어야 하는지 여부를 판정하는 단계와,

처리를 위해 상기 고속 경로 포맷 I/O 요청이 상기 펌웨어 처리 스택으로 우회되어야 한다는 판정에 응답하여, 처리를 위해 상기 고속 경로 포맷 I/O 요청을 상기 펌웨어 처리 스택으로 우회시키는 단계와,

상기 고속 경로 포맷 I/O 요청의 수신에 응답하여, 상기 고속 경로 회로로부터의 고속 경로 로크 요청 신호를

상기 영역 로크 모듈로 제공하는 단계—상기 고속 경로 로크 요청은 상기 고속 경로 포맷 I/O 요청에 관한 정보를 포함함—를 포함하되,

상기 판정하는 단계는,

영역 로크 모듈의 동작에 의해, 상기 고속 경로 로크 요청 신호와 함께 제공된 상기 정보에 기초하여 상기 수신한 고속 경로 포맷 I/O 요청이 로크된 블록 주소 범위 내 블록 주소로 향하는지 여부를 판정하는 단계와,

상기 펌웨어 처리 스택의 동작에 의해, 상기 범위에 액세스하려고 하는 고속 경로 포맷 I/O 요청이 상기 펌웨어 처리 스택으로 우회되어야 하는 블록 주소 범위를 식별하는 단계와,

상기 식별된 블록 주소 범위에 대한 우회형 로크의 설정을 요청하는 우회형 로크 요청 신호를 상기 펌웨어 처리 스택으로부터 상기 영역 로크 모듈로 제공하는 단계와,

상기 영역 로크 모듈의 동작에 의해, 상기 우회형 로크 요청 신호의 수신에 응답하여 상기 우회형 로크 요청 신호에서 식별된 블록 주소 범위에 대해 우회형 로크를 설정하는 단계를 포함하고,

상기 판정하는 단계는,

현재 로크된 블록 주소 범위의 상태에 기초하여 상기 고속 경로 포맷 I/O 요청의 처리가 상기 펌웨어 처리 스택으로 우회되도록, 상기 수신한 고속 경로 포맷 I/O 요청이 설정된 우회형 로크와 관련된 블록 주소 범위 내 블록 주소로 향하는지 여부를 판정하는 단계를 더 포함하는

방법.

청구항 8

제7항에 있어서,

상기 펌웨어 처리 스택은 상기 캐시 메모리를 라이트-백 캐시로서 관리하고,

상기 방법은,

상기 펌웨어 처리 스택의 동작에 의해, 상기 펌웨어 처리 스택에 의한 기록 I/O 요청의 처리에 응답하여 상기 캐시 메모리에 기록 데이터를 저장하는 단계를 더 포함하고,

상기 식별하는 단계는,

상기 캐시 메모리에 저장된 상기 기록 데이터와 관련된 블록 주소를 포함한 블록 주소 범위를 식별하는 단계를 더 포함하는

방법.

청구항 9

제7항에 있어서,

상기 저장 제어기는 상기 펌웨어 처리 스택과 연결된 불량 블록 메모리를 가지고,

상기 방법은,

상기 펌웨어 처리 스택의 동작에 의해, 상기 저장 장치 상의 잠재적 불량 블록의 블록 주소를 식별하는 정보를 상기 불량 블록 메모리에 저장하는 단계를 더 포함하고,

상기 식별하는 단계는,

상기 불량 블록 메모리에서 잠재적 불량 블록의 블록 주소를 포함한 블록 주소 범위를 식별하는 단계를 더 포함하는

방법.

청구항 10

컴퓨터에 의해 실행될 때 저장 제어기에서 동작가능한 방법을 수행하는 프로그램된 명령어를 유형적으로 구현하는 비일시적 컴퓨터 판독가능 매체로서,

상기 저장 제어기는 하나 이상의 호스트 시스템 및 하나 이상의 저장 장치에 연결되도록 구성되고, 호스트 시스템으로부터 수신한 고속 경로 포맷 I/O 요청의 처리를 위한 고속 경로 회로 및 임의 포맷의 I/O 요청을 처리하기 위한 펌웨어 처리 스택을 가지며,

상기 방법은,

상기 저장 제어기의 캐시 메모리 내의 블록 어드레스의 범위를 로크하기 위해 상기 고속 경로 회로로부터 로크 요청을 수신하는 단계와,

고속 경로 포맷 I/O 요청을 수신하는 단계와,

상기 저장 제어기의 영역 로크 모듈의 동작에 의해, 상기 캐시 메모리 내의 현재 로크된 블록 주소 범위의 상태에 기초하여 상기 수신한 고속 경로 포맷 I/O 요청이 상기 펌웨어 처리 스택으로 우회되어야 하는지 여부를 판정하는 단계와,

상기 고속 경로 포맷 I/O 요청의 처리가 상기 펌웨어 처리 스택으로 우회되어야 한다는 판정에 응답하여, 처리를 위해 상기 고속 경로 포맷 I/O 요청을 상기 펌웨어 처리 스택으로 우회시키는 단계와,

상기 고속 경로 포맷 I/O 요청의 수신에 응답하여, 상기 고속 경로 회로로부터의 고속 경로 로크 요청 신호를 상기 영역 로크 모듈로 제공하는 단계-상기 고속 경로 로크 요청은 상기 고속 경로 포맷 I/O 요청에 관한 정보를 포함함-를 포함하되,

상기 판정하는 단계는,

영역 로크 모듈의 동작에 의해, 상기 고속 경로 로크 요청 신호와 함께 제공된 상기 정보에 기초하여 상기 수신한 고속 경로 포맷 I/O 요청이 로크된 블록 주소 범위 내 블록 주소로 향하는지 여부를 판정하는 단계와,

상기 펌웨어 처리 스택의 동작에 의해, 상기 범위에 액세스하려고 하는 고속 경로 포맷 I/O 요청이 상기 펌웨어 처리 스택으로 우회되어야 하는 블록 주소 범위를 식별하는 단계와,

상기 식별된 블록 주소 범위에 대한 우회형 로크의 설정을 요청하는 우회형 로크 요청 신호를 상기 펌웨어 처리 스택으로부터 상기 영역 로크 모듈로 제공하는 단계와,

상기 영역 로크 모듈의 동작에 의해, 상기 우회형 로크 요청 신호의 수신에 응답하여 상기 우회형 로크 요청 신호에서 식별된 블록 주소 범위에 대해 우회형 로크를 설정하는 단계를 포함하고,

상기 판정하는 단계는,

현재 로크된 블록 주소 범위의 상태에 기초하여 상기 고속 경로 포맷 I/O 요청의 처리가 상기 펌웨어 처리 스택으로 우회되도록, 상기 수신한 고속 경로 포맷 I/O 요청이 설정된 우회형 로크와 관련된 블록 주소 범위 내 블록 주소로 향하는지 여부를 판정하는 단계를 더 포함하는

컴퓨터 판독가능 매체.

청구항 11

제10항에 있어서,

상기 펌웨어 처리 스택은 상기 캐시 메모리를 라이트-백 캐시로서 관리하고,

상기 방법은,

상기 펌웨어 처리 스택의 동작에 의해, 상기 펌웨어 처리 스택에 의한 기록 I/O 요청의 처리에 응답하여 상기 캐시 메모리에 기록 데이터를 저장하는 단계를 더 포함하고,

상기 식별하는 단계는,

상기 캐시 메모리에 저장된 상기 기록 데이터와 관련된 블록 주소를 포함한 블록 주소 범위를 식별하는 단계를 더 포함하는

컴퓨터 판독가능 매체.

청구항 12

제10항에 있어서,

상기 저장 제어기는 상기 펌웨어 처리 스택과 연결된 불량 블록 메모리를 가지고,

상기 방법은,

상기 펌웨어 처리 스택의 동작에 의해, 상기 저장 장치 상의 잠재적 불량 블록의 블록 주소를 식별하는 정보를 상기 불량 블록 메모리에 저장하는 단계를 더 포함하고,

상기 식별하는 단계는,

상기 불량 블록 메모리에서 잠재적 불량 블록의 블록 주소를 포함한 블록 주소 범위를 식별하는 단계를 더 포함하는

컴퓨터 판독가능 매체.

청구항 13

호스트 시스템으로부터 복수의 저장 장치로의 I/O(Input/Output) 요청을 처리하도록 동작가능한 저장 제어기로서,

상기 저장 제어기는,

상기 I/O 요청을 캐싱하도록 동작가능한 캐시 메모리와,

상기 I/O 요청을 처리하도록 동작가능한 펌웨어 처리 스택과,

상기 I/O 요청의 포맷에 기초하여 고속 경로 처리를 위한 상기 I/O 요청의 부분을 선택하도록 동작가능한 고속 경로 회로와,

상기 고속 경로 회로 및 상기 펌웨어 처리 스택에 연결된 영역 로크 모듈을 포함하고,

상기 영역 로크 모듈은, 상기 캐시 메모리 내의 지정된 블록 주소 범위를 로크하기 위해 상기 고속 경로 회로로부터의 로크 요청을 처리하여, 상기 로크 요청에 기초하여 고속 경로 처리를 위한 상기 I/O 요청의 상기 부분을 상기 펌웨어 처리 스택으로 우회시키도록 동작가능하고,

상기 펌웨어 처리 스택은, 상기 I/O 요청의 상기 부분이 상기 펌웨어 처리 스택으로 우회되는 것에 응답하여, 상기 고속 경로 회로를 대신하여 상기 저장 장치에 대한 상기 I/O 요청의 상기 부분을 처리하고,

상기 펌웨어 처리 스택은, 상기 블록 주소 범위의 블록에 액세스하려고 하는 고속 경로 포맷 I/O 요청이 상기 펌웨어 처리 스택에 의해 처리되어야 한다는 판정에 응답하여, 블록 주소 범위에 대한 우회형 영역 로크를 설정하기 위해 상기 영역 로크 모듈로 로크 요청을 제공하도록 구성되고,

상기 영역 로크 모듈은, 상기 펌웨어 처리 스택으로부터의 상기 로크 요청의 수신에 응답하여, 블록 주소 범위에 대해 상기 우회형 영역 로크를 설정하도록 더 구성되고,

상기 저장 제어기는, 상기 고속 경로 포맷 I/O 요청의 고속 경로 회로 처리에 의해 영향받을 블록 주소 범위에 대해 상기 우회형 영역 로크가 설정되었음을 검출한 것에 응답하여, 수신한 고속 경로 포맷 I/O 요청을 상기 펌웨어 처리 스택으로 우회시키도록 더 구성되는

저장 제어기.

발명의 설명

기술 분야

[0001] 본 발명은 일반적으로 저장 제어기에 관한 것으로, 특히 I/O 요청을 저장 제어기의 다중 I/O 처리 스택들 중의 일 스택으로 우회(divert)시키기 위해 영역 로크 구조(region lock structure)를 사용하는 방법 및 구조에 관한 것이다.

배경 기술

- [0002] <관련 출원의 상호 참조>
- [0003] 본 특허는 (여기서 집합적으로 "관련 특허"로 언급되는) 일반적으로 소유한 다음 미국 특허 및/또는 특허 출원과 관련있다.
- [0004] - 2010년 4월 14일 출원된 미국 특허출원 제12/760,415호, "APPARATUS AND METHODS FOR TREE MANAGEMENT ASSIST CIRCUIT IN A STORAGE SYSTEM"
- [0005] - 2010년 4월 14일 출원된 미국 특허출원 제12/760,434호, "APPARATUS AND METHODS FOR REGION LOCK MANAGEMENT ASSIST CIRCUIT IN A STORAGE SYSTEM"
- [0006] - 2005년 10월 19일 출원된 미국 특허출원 제7,529,902호, "METHODS AND SYSTEMS FOR LOCKING IN STORAGE CONTROLLERS"
- [0007] 이들 모두는 여기서 참조로서 병합된다.
- [0008] 저장 제어기는 하나 이상의 호스트 시스템으로부터 I/O 요청을 수신하며 저장 제어기와 연결된 저장 장치 상에 데이터를 저장 또는 검색하기 위해 수신한 이러한 요청을 처리하도록 구성된 장치이다. 예를 들면 RAID(Redundant Array of independent Drives) 저장 제어기는 하나 이상의 물리적 저장 장치의 일부를 각각 포함한 하나 이상의 논리 볼륨(logical volumes)을 관리한다. RAID 제어기는 논리 볼륨의 논리 블록 주소를 저장 장치의 대응한 물리적 블록으로 매핑(mapping)하고, (논리 볼륨을 위해 정의된 특정한 RAID 저장 관리 기법의 리던던시 및 스트라이핑 정책(redundancy and striping policies)에 따라서) 향상된 신뢰도 및/또는 성능을 위해 데이터의 리던던시 및/또는 스트라이핑을 제공한다.
- [0009] 고성능 저장 제어기의 경우에 I/O 요청을 처리시에 저장 제어기의 성능을 향상시키는데 캐시 메모리를 이용하는 것이 일반적이다. 예를 들면 호스트 시스템에 의해 논리 볼륨에 기록된 데이터는 저장 제어기의 캐시 메모리에 저장될 수 있으므로, 후속한 판독 I/O 요청은 논리 볼륨을 포함한 저장 장치로부터 요청 데이터를 검색하기보다는 캐시 메모리로부터 요청 데이터를 보다 더 신속하게 검색함으로써 만족될 수 있다. 일부 저장 제어기는 캐시 메모리와 저장 메모리로 데이터의 모두로 기록하는 "라이트-스루(write-through) 캐시"로서 캐시 메모리를 관리할 수 있다. 일부 제어기는 캐시 메모리로 데이터를 기록하고, 소정의 시간 후에(예를 들어 저장 제어기가 I/O 요청을 처리하기에 덜 바쁠 때) 데이터를 저장 장치로 단지 포스트(post)/플러싱(flushing)하는 "라이트-백(write-back)" 모드에서 캐시 메모리를 관리할 수 있다. 라이트-백 모드에서, 캐시 메모리에 상주하지만 아직 저장 장치에는 저장되지 않은 ("더티 데이터(dirty data)"로서 참조되는) 데이터가 있을 수 있다.
- [0010] 일부 저장 제어기(예를 들면 고성능 RAID 저장 제어기)는 I/O 요청을 처리하기 위해 저장 제어기 내에 다중 처리 경로 또는 스택을 제공한다. 예를 들면 LSI사의 소정 저장 제어기는 펌웨어 처리 스택(firmware processing stack)과 함께 동작가능한 "고속 경로(fast path)" 처리 회로(예를 들면 제1 처리 경로/스택)를 제공한다(예를 들어 제2/종래 처리 경로는 일반적으로 제어기의 범용 또는 특수목적 프로세서에 의해 실행되는 펌웨어 인스트럭션으로서 구현된다). LSI사 고속 경로 처리 회로는 고속 경로 회로가 요청을 신속하게 처리할 수 있도록 포맷된 I/O 요청을 보다 신속하게 처리하기 위해 호스트 시스템 상의 특수 드라이버와 함께 작동한다(예를 들면 호스트 시스템 향상된 드라이버는 기본 물리적 저장 장치를 직접 액세스하기 위해 RAID 매핑을 이해하여 고속 경로 요청을 생성함으로써 고속 경로 회로로 하여금 I/O 요청을 신속하게 처리하게 한다). 펌웨어 처리 경로는 (예를 들어 고속 경로 I/O 요청으로 인한 그의 포맷팅에 관계없이 I/O 요청을 처리하기 위해) 임의 호스트 시스템으로부터 수신한 임의 유형의 I/O 요청을 처리하는데 적합하다. 펌웨어 스택은 인스트럭션을 실행하는 범용 프로세서를 사용하며 전체 RAID 매핑 및 관리를 제공하는 보다 일반화된 처리를 다룬다. 따라서 보다 융통성있음에도 불구하고, 펌웨어 처리 스택은 I/O 요청을 고속 경로 회로에서 보다 더 느리게 처리한다.

[0011] 다중 처리 스택(예를 들어 고속 경로 회로 경로 및 펌웨어 처리 스택)을 가진 이러한 저장 제어기에서, 저장 제어기의 일부 초기 처리는 다중 처리 스택이 특정한 I/O 요청을 처리하기에 적절한 지를 판정할 필요가 있다. 이러한 초기 처리는 다수의 인자에 의존한 복잡한 처리일 수 있다. 예를 들어 만약 저장 제어기가 라이트-백 모드에서 그의 캐시 메모리를 이용하게 된다면, 고속 경로 처리 회로의 사용은 저장 제어기의 캐시 메모리에, 아직 저장 장치로 플러싱 또는 포스트되지 않은 더티 데이터가 현재 있는 경우에 데이터 무결성(data integrity) 문제를 일으킬 수 있다. 고속 경로 회로는 캐시 메모리의 콘텐츠를 알지 못하도록, 따라서 차후에 더티 데이터가 플러싱된다면 중복기록될 수 있는 저장 장치로 데이터를 기록할 수 있도록 이러한 제어기에서 최적으로 조정될 수 있다. 그렇지 않으면, 고속 경로 회로는 더티 데이터가 고속 경로 회로에 의해 액세스되는 블록에 대해 캐시 메모리에 현재 저장되어 있으므로 저장 장치로부터 최신이 아닌 데이터를 판독할 수 있다. 더욱이 예를 들어 만약 펌웨어 처리 스택이 특정한 저장 장치로부터 또는 논리 볼륨으로부터 데이터를 저장 또는 검색할 시에 피해야 할 불량 블록 리스트(list of bad blocks)를 유지한다면, 고속 경로 처리 회로는 불량 블록 리스트를 알지 못할 수 있고, 잠재적으로 불량 블록을 액세스하려고 함으로써 데이터 무결성 문제를 다시 일으킬 수 있다. 고속 경로 회로는 펌웨어 처리 스택만이 불량일 수 있음을 알고 있는 이러한 불량 블록으로 데이터를 판독 또는 기록하려고 할 수 있다. 다른 상황은 펌웨어 I/O 요청 처리 스택이 풀-스트라이프 기록(full-stripe writes)을 형성하도록 보다 작은 I/O 요청을 병합, 스트리밍 I/O 요청을 위한 최적화 등과 같은 소정의 최적화에 현재 관여할 시에 발생할 수 있다. 호스트의 요청을 인코딩시에 특정 프로세서를 선호하거나 또는 지명할 수 있지만, 상기 및 다른 이유로 인하여 저장 제어기의 일 I/O 요청 프로세서를 다른 I/O 요청 프로세서에 비해 선호하는 현상을 가져올 수 있다.

[0012] 다중 처리 스택이 I/O 요청을 처리하는데 최상으로 적합하다고 판정시에 고려하는 이들 다양한 조건으로는 시간 소비적이며 복잡한 절차를 들 수 있다. 따라서 본 기법은 소정 조건이 발생할 수 있다면 이들 및 다른 잠재적 문제를 피하기 위하여 다중 처리 스택 중의 하나를 간단히 디스에이블(disable)할 수 있다(예를 들면 LSI사 저장 제어기의 고속 경로 I/O 처리 회로의 사용을 디스에이블). 예를 들어 임의 더티 데이터가 (특정한 고속 경로 I/O 요청이 그 더티 데이터를 액세스할 수 있는 지의 여부에 관계없이) 현재 라이트-백 캐시 메모리에 저장 된다면, 또는 임의 블록이 (특정한 고속 경로 I/O 요청이 이들 불량 블록을 액세스할 수 있는 지의 여부에 관계없이) 펌웨어 처리 스택에 의해 "불량 블록"으로 지명된다면, 펌웨어 처리 스택은 고속 경로 처리 회로를 디스에이블하고, 그의 보다 저속이지만 보다 일반화된 처리 경로를 통해 모든 I/O 요청을 처리하는 것을 택한다. 전형적인 설계 접근 방안에서, LSI 고속 경로 I/O 처리회로는 고속 경로 회로로 향하는 I/O 요청의 처리에 의해 발생하는 이들 및 다른 문제를 피하도록 간단히 디스에이블될 수 있다. 고속 경로 I/O 처리회로의 사용을 디스에이블하게 되면 저장 제어기의 성능을 저하시킬 수 있다.

[0013] 따라서 특정한 I/O 요청을 처리하도록 구성된 I/O 처리 스택을 효과적으로 및 효율적으로 판정하고, 데이터 무결성 문제의 위험없이 선호되는 선택된 I/O 처리 스택으로 수신한 I/O 요청을 우회시키는 것이 진행중인 도전과제이다.

발명의 내용

과제의 해결 수단

[0014] 본 발명은 부착된 호스트 시스템으로부터 수신한 I/O 요청을 제어기에서 다중 처리 스택 중의 하나로 효율적으로 우회시키기 위해 영역 로크(region locks)를 사용하는 방법 및 구조를 저장 제어기 내에 제공함으로써, 상기 및 다른 문제를 해결하여 실용 기술의 상태를 향상시키려 한다. 제어기 내 영역 로크 모듈은 각 처리 스택으로 하여금 저장 장치의 블록 주소 범위에 대해 영역 로크를 요청하게 한다. 우회형(divert-type) 로크 요청은 I/O 요청을 다중 처리 스택 중의 특정한 한 스택으로 우회시켜야 하는 블록 주소 범위를 식별하도록 설정될 수 있다.

[0015] 고속 경로 포맷 I/O 요청을 처리하기 위한 고속 경로 회로, 그리고 임의 형태의 I/O 요청을 처리하기 위한 펌웨어 처리 스택을 가진 제어기의 일 예시적 실시예에서, 펌웨어 처리 스택은 영역 로크 모듈이 블록 주소 범위에 대해 우회형 영역 로크를 설정하도록 요청할 수 있다. 블록 주소 범위는 캐시 및/또는 잠재적 불량 블록 주소에서의 더티 데이터와 관련있다. 고속 경로 회로가 I/O 요청을 처리하기 위해 블록 범위를 로크하려고 할 때, I/O 요청을 위한 블록 범위의 임의 부분을 다루는 우회형 로크가 I/O 요청을 펌웨어 처리 스택으로 우회되도록 할 것이다.

[0016] 일 양상에서, 하나 이상의 호스트 시스템과의 연결에 적합하며, 저장 제어기와 연결된 하나 이상의 저장 장치 상에 데이터를 관리하도록 구성된 저장 제어기를 제공한다. 저장 제어기는 제어기의 프로세서 상에 동작하는 펌웨어 처리 스택과 부착된 호스트 시스템으로부터 수신한 것으로 고속 경로 처리를 위해 포맷된 I/O 요청을 처리하도록 구성된 고속 경로 회로, 그리고 제어기의 프로세서 상에서 동작하며 임의의 포맷 I/O 요청을 처리하도록 구성된 펌웨어 처리 스택을 포함한다. 제어기는 고속 경로 회로와 연결되며 펌웨어 처리 스택과 연결된 영역 로크 모듈을 더 포함한다. 영역 로크 모듈은 명시된 블록 주소 범위를 로크하기 위해 로크 요청 신호를 수신하는데 적합하며, 수신한 고속 경로 포맷 I/O 요청을 현재 로크된 블록 주소 범위의 상태를 기반으로 펌웨어 처리 스택으로 우회시켜야 하는지의 여부를 판정하는데 더 적합하다. 저장 제어기는 고속 경로 포맷 I/O를 펌웨어 처리 스택으로 우회시켜야 한다는 판정에 응답하여 처리를 위해 고속 경로 포맷 I/O 요청을 펌웨어 처리 스택으로 우회시키는데 더 적합하다.

[0017] 다른 양상은 방법을 구현하는 컴퓨터 판독가능 매체와 방법 그 자체를 제공한다. 방법은 하나 이상의 호스트 시스템과의 연결에 적합하고 하나 이상의 저장 장치와의 연결하도록 구성된 저장 제어기에서 작동될 수 있다. 저장 제어기는 호스트 시스템으로부터 수신한 고속 경로 포맷 I/O 요청의 처리를 위해 고속 경로 회로를 가지며, 임의의 포맷 I/O 요청을 처리하기 위해 펌웨어 처리 스택을 가진다. 방법은 고속 경로 포맷 I/O 요청을 수신하는 단계, 그리고 제어기의 영역 로크 모듈의 동작에 의해, 현재 로크된 범위의 블록 주소의 상태를 기반으로 수신한 고속 경로 포맷 I/O를 펌웨어 처리 스택으로 우회시켜야 하는지의 여부를 판정하는 단계를 포함한다. 그 다음, 방법은 고속 경로 포맷 I/O 요청을 펌웨어 처리 스택으로 우회시켜야 한다는 판정에 응답하여 처리를 위해 고속 경로 포맷 I/O 요청을 펌웨어 처리 스택으로 우회시킨다.

도면의 간단한 설명

[0018] 도 1은 제어기의 일 처리 스택으로부터 다른 처리 스택으로 I/O 요청의 처리를 우회시키기 위하여 우회형 영역 로크의 관리를 제공하는 특징 및 양상에 따른 향상된 예시적 저장 제어기의 블록도.

도 2 내지 도 8은 I/O 요청의 처리를 제어기의 일 처리 스택으로부터 다른 처리 스택으로 우회시키기 위해 우회형 영역 로크의 관리를 제공하는 특징 및 양상에 따르는 예시적 방법을 기술하는 흐름도.

도 9는 도 2 내지 도 8의 방법을 구현하는 컴퓨터 판독가능 매체를 수신하도록 구성된 도 1의 저장 제어기와 같은 저장 제어기의 예시적 컴퓨팅 장치의 블록도.

발명을 실시하기 위한 구체적인 내용

[0019] 도 1은 하나 이상의 호스트 시스템(120)과 하나 이상의 저장 장치(130)와 연결하도록 구성된 향상된 저장 제어기(100)를 구비한 시스템의 블록도이다. 호스트 시스템(120)은 저장 장치(130)를 액세스하기 위해 I/O 요청을 발생하는 임의의 적당한 컴퓨팅 장치 또는 다른 시스템일 수 있다. 저장 장치(130)는 자기 또는 광 디스크 드라이브, 고체상태 드라이브 등과 같이 데이터를 저장하기 위한 임의의 적당한 장치일 수 있다. 저장 제어기(100)는 인터페이스(150)에 의해 호스트 시스템(120)과 연결될 수 있다. 인터페이스(150)는 예를 들어 SAS(Serial Attachment SCSI), 병렬 SCSI, PATA(parallel Advanced Technology Attachment), SATA(Serial ATA), 주변 구성요소 상호연결부(PCI 익스프레스와 같이 PCI의 변형을 포함한 PCI) 등을 포함하는, 호스트 시스템(120)과 저장 제어기(100)를 연결하기 위한 임의의 적당한 통신 매체 및 프로토콜을 포함할 수 있다. 저장 제어기(100)는 인터페이스(152)를 통해 저장 장치(130)와 연결될 수 있다. 인터페이스(152)는 예를 들어 병렬 SCSI SAS, PATA, SATA, PCI, 섬유 채널 등을 포함하는, 저장 제어기(100)를 저장 장치(130)와 연결하기 위한 임의의 적당한 통신 매체를 포함할 수 있다.

[0020] 저장 제어기(100)는 저장 장치(130)를 액세스함으로써 부착된 호스트 시스템으로부터 수신한 I/O 요청을 처리하도록 구성된 고속 경로 회로(102) 및 펌웨어 처리 스택(104)을 포함한다. 회로(102)와 스택(104)의 모두는 처리 "스택"을 으로 언급될 수 있다. 따라서 제어기(100)는 I/O 요청을 처리하기 위한 다중 스택을 포함한다. 더욱이 임의의 수의 이러한 처리 스택은 설계 선택의 문제로서 제어기(100)의 실시예에 제공될 수 있다. 제어기(100)는 일반적으로 이후에 더 거론되는 바와 같이 설계 로크를 포함한 우회 정보를 기반으로 하여 일 처리 스택으로부터 다른 처리 스택으로 I/O 요청을 우회시키도록 동작할 수 있다. 일 처리 스택이 I/O 요청을 정확히 처리할 수 없다면, 영역 로크에 의한 이러한 우회가 잠재적 데이터 손상을 피하는데 사용될 수 있다. 또한 성능 향상 및 최적화를 위하여 일 처리 스택을 다른 처리 스택에 비해 우선하도록 우회 영역 로크를 사용할 수 있

다.

- [0021] 펌웨어 처리 스택(104)은 범용 또는 특수 목적 프로세서(도시되지 않음), 그리고 범용 또는 특수 목적 프로세서에 의한 실행을 위해 프로그래밍된 인스트럭션 및 데이터를 저장하는 관련 프로그램 및 데이터 메모리(도시되지 않음)를 포함한다. 펌웨어 처리 스택(104)의 프로그래밍된 인스트럭션은 임의의 적당한 커맨드 구조에 따라서 포맷된 호스트 시스템(120)으로부터 I/O 요청을 수신할 수 있고 저장 장치(130) 상의 정보를 액세스하도록 수신 I/O 요청을 처리할 수 있게 한다.
- [0022] 앞에서 언급한 바와 같이, 스택(104)은 논리 볼륨으로 향하는 I/O 요청과 관련된 RAID 저장 관리를 포함한 임의 및 모든 I/O 요청을 처리하는 융통성을 제공하지만, 스택(104)은 소정 유형의 I/O 요청의 처리에 원하는 것보다 더 느릴 수 있다. 따라서 저장 제어기(100)는 고속 경로 회로(102)를 더 포함한다. 고속 경로 회로(102)는 시스템(120) 내 향상된 드라이버로부터 수신한, 고속 경로 회로(102)에 의한 처리를 위해 포맷된 I/O 요청을 다루기 위해 특별히 설계된 전자 회로를 포함한다. LSI사의 소정 저장 제어기는 데이터를 액세스시에 저장 제어기(100)의 성능을 개선하기 위해 향상된 드라이버 모듈과 함께 이러한 고속 경로 회로를 사용하는 것을 예들 든다. 통상, 호스트 시스템(120)의 향상된 드라이버(도시되지 않음)는 RAID 관리를 위해 제어기(100)에 의해 유지되는 매핑 정보와 유사하게 저장 제어기(100)에 의해 관리되는 RAID 논리 볼륨에 대한 매핑 정보를 액세스 한다. 호스트(120)의 향상된 드라이버는 펌웨어 처리 스택(104)을 이용하는 저장 제어기(100)보다 더 신속하고 효율적으로 필수 논리적-물리적 매핑(requisite logical-physical mapping)을 선택적으로 수행할 수 있다. 이러한 경우에, 호스트 시스템(120)의 향상된 드라이버는 임의의 필수 논리적-물리적 매핑 또는 다른 필수 논리 볼륨 관리 계산에 그 자신의 처리력을 이용하는 고속 경로 포맷 I/O 요청을 준비한다. 그 후, 고속 경로 포맷 I/O 요청은 고속 경로 회로(102)에 의한 처리를 위해 저장 제어기(100)로 전송된다.
- [0023] 펌웨어 처리 스택(104)은 임의의 적당한 프로세서 또는 시스템 버스(154)에 의해 캐시 메모리(108) 및 불량 블록 메모리(110)와 연결된다. 캐시 메모리(108)와 불량 블록 메모리(110)의 각각은 예를 들면 DRAM(dynamic random access memory), 플래시 메모리 등을 포함한 임의의 적당한 메모리 구성요소를 포함할 수 있다. 본 기술 분야에 일반적으로 알려진 바와 같이, 펌웨어 처리 스택(104)은 저장 장치(130)로 예정된 기록 I/O 요청과 관련된 데이터를 저장하기 위해 캐시 메모리(108)를 이용한다. 캐시 메모리(108)를 관리하는 라이트-스루 모드에서, 펌웨어 처리 스택(104)은 캐시 메모리(108)에서 기록 I/O 요청과 관련된 기록 데이터를 저장하고, 또한 (본래 동시에) 저장 장치(130) 상에 기록 데이터를 저장한다. 그 후, 캐시 메모리(108)에 저장된 데이터는 캐시 메모리에 현재 저장된 데이터의 블록 주소로부터 데이터를 요청하는 후속한 판독 요청이 발생될 시에 보다 신속하게 완료하기 위해 펌웨어 처리 스택(104)에 의해 이용될 수 있다. 또한 펌웨어 처리 스택(104)은 잠재적 불량 블록일 수 있는 블록 주소의 범위 또는 특정 블록 주소를 식별할 수 있다. 펌웨어 처리 스택(104) 내의 임의의 적당한 기법이 "불량"으로 간주될 수 있는 블록을 식별하는데 이용될 수 있다. 펌웨어 처리 스택(104)은 I/O 요청의 처리 시에 회피(avoidance)를 위한 블록에 주목하기 위하여 불량 블록 메모리(110)에 이러한 블록 주소를 레코딩한다. 제어기(100) 내 이들 및 다른 기능은 우회형 영역 로크 특징 및 이의 양상으로부터 이룰 수 있다. 일 요청 처리 스택 또는 회로로 향하는 요청을 상이한 처리 스택 또는 회로로 우회해야 하는 제어기(100) 내 임의의 상황에서, 우회형 영역 로크 특징과 이의 양상은 최소 오버헤드 처리로써 효율적으로 이러한 관정을 하게 한다.
- [0024] 고속 경로 회로(102)와 펌웨어 처리 스택(104)의 모두는 (예를 들어 통신 경로 156, 158를 각각 통해) 제어기(100)의 영역 로크 모듈(106)과 상호작용한다. 영역 로크 모듈(106)은 펌웨어 처리 스택(104)으로부터 및 고속 경로 회로(102)로부터 로크 요청과 영역 로크 해제 요청을 수신하기 위한 임의의 적당한 회로를 포함한다. 영역 로크 요청은 일반적으로 I/O 요청 처리 스택이 액세스하기를 요구하는 블록 주소 범위를 식별한다(즉 판독을 위한 임시 공유 액세스, 소정 동작을 위한 배타적 액세스 등). 예를 들면 고속 경로 회로(102)가 수신한 고속 경로 포맷 I/O 요청을 처리하기 위해 필요한 영향받은(affected) 블록 주소 범위를 식별할 때, 이것은 경로(156)를 통해 영역 로크 모듈(106)로 로크 요청을 발생한다. 영역 로크 모듈(106)은 고속 경로 회로(102) 또는 펌웨어 처리 스택(104) 중의 하나로 현재 로크된 블록 범위를 식별하는 (도시되지 않은) 관련 영역 로크 메모리에서 데이터 구조를 유지한다. 새 로크 요청에 응답하여, 영역 로크 모듈(106)은 새 요청에 의해 식별된 블록 주소 범위가 현재 승인받은 영역 로크와 관련된 블록 주소 범위의 임의의 일부와 겹치는(overlap) 지의 여부를 판정한다. 이러한 겹침이 검출되면, 영역 로크 모듈(106)은 새 요청을 위한 블록 주소 범위가 현재 승인받은 로크와 관련된 블록 주소의 임의의 부분과 겹쳐지지 않는 시간때 까지 로크의 승인을 거부하거나 또는 지연시킨다. 이런 방식으로, 고속 경로 회로(102)와 펌웨어 처리 스택(104)은 또한 그들의 액세스 요구가 완료될 때에(즉 관련 I/O 요청이 처리되었을 때) 영역 로크 모듈(106)로 로크 해제 요청(lock release requests)을 발생한다.

소정 예시적 실시예에서, 영역 로크 모듈(106)은 또한 수신한 I/O 요청이 고속 경로 I/O 요청으로 포맷되었는지 또는 그렇게 포맷되지 않았는지의 여부를 검출하기 위해 초기 라우팅 구성요소로서의 기능을 할 수 있다. 이러한 예시적 실시예에서, 모듈(106)은 (적절한 통신 매체 및 프로토콜을 통해) 호스트(120)와 직접 연결될 수 있는 반면에, 회로(102) 및 스택(104)은 모듈(106)과 직접 연결된다(따라서 호스트 120과 간접적으로 연결된다). 제어기(100)의 요소의 상기 및 다른 구성과 배치는 설계 선택의 문제로서 본 기술분야에 통상의 지식을 가진 자에게 명백할 것이다.

[0025] 특징 및 이의 양상에 따라서, 펌웨어 처리 스택(104)은 (경로 158를 통해) 영역 로크 모듈(106)로 우회형 로크 요청을 발생할 수 있다. 우회형 로크 요청은 가능한 데이터 손상을 피하고, 그리고/또는 성능 향상을 위해 최적화를 가능하게 하기 위하여 식별된 블록 주소 범위에 영향을 줄 수 있는 모든 I/O 요청을 다중 I/O 처리 스택 중의 한 스택 또는 다른 스택(예를 들면 고속 경로 회로 102 또는 펌웨어 처리 스택 104)으로 우회시켜야 하는 블록 주소 범위를 식별한다. 특히, 예를 들어 펌웨어 처리 스택(104)은 캐시 메모리(108) 및 불량 블록 메모리(110)의 사용을 관리할 수 있다. 앞에서 언급한 바와 같이, 펌웨어 처리 스택(104)이 라이트-백 모드에서 캐시 메모리(108)를 관리할 때, 수신한 기록 I/O 요청과 관련된 기록 데이터가 캐시 메모리(108)에 저장되어 "더티 데이터"로서 표시될 수 있다. 펌웨어 처리 스택(104)이 캐시 메모리(108)에 이러한 더티 데이터를 저장할 때, 추가 I/O 요청을 펌웨어 처리 스택(104)으로 우회시켜야 하는 대응한 블록 주소 범위를 식별하기 위해 우회형 로크 요청을 영역 로크 모듈(106)로 발생할 수 있다. 그 다음, 고속 경로 회로(102)가 고속 경로 포맷 I/O 요청을 위해 블록 주소 범위를 로크하려고 할 때, 영역 로크 모듈(106)은 승인된 우회형 로크의 오버랩을 검출하고 추가 처리를 위해 펌웨어 처리 스택(104)으로 고속 경로 포맷 요청의 우회를 강요할 것이다. 따라서 펌웨어 처리 스택이 이미 우회형 영역 로크를 획득한 블록 주소 범위로 예정된 고속 경로 포맷 I/O 요청은 펌웨어 처리 스택에 의한 처리를 위해 고속 통화 회로(102)로부터 멀리 우회될 것이다. 앞에서 언급한 바와 같이 영역 로크는 펌웨어 처리 스택(104)에 의한 처리 동안에 고속 경로 회로(102)로부터 멀어지는 고속 경로 포맷 I/O 요청을 우회시키려는 임의의 수의 이유를 위해 펌웨어 처리 스택(104)에 의해 사용될 수 있다. 위에 언급한 바와 같이 데이터 손상의 회피 외에도, 보다 작은 동작을 보다 큰 동작으로 합치고, I/O 처리 스택들 간에 로드 균형, 스트리밍 I/O 최적화 등과 같이 이용가능한 다양한 최적화의 적절한 동작을 보장하기 위해, 펌웨어 처리 스택(104)은 우회 영역 로크를 이용할 수 있다.

[0026] 짧은 대시선 화살표(160, 162)는 펌웨어 처리 스택(104)에 의한 추가 처리를 위해 고속 경로 회로(102)에 수신된 고속 경로 포맷 I/O 요청을 우회하는 것을 나타낸다. 소정 예시적 실시예에서, 영역 로크 모듈(106)은 고속 경로 포맷 I/O 요청을 처리하기 전에 블록 주소 범위를 로크하려고 하는 고속 경로 회로(102)에 응답하여, (대시 화살표 162에 의해 표시된 바와 같이) 우회를 수행할 수 있다. 다른 예시적 실시예에서, 고속 경로 회로(102)는 고속 경로 포맷 I/O 요청에 의해 액세스될 블록 주소 범위에 대해 그의 액세스 로크 요청을 요청할 수 있고, 추가 처리를 위해 영향받은 블록 주소 범위를 펌웨어 처리 스택(104)으로 우회시켜야 한다는 것을 나타내는 거부 신호(denial signal)를 영역 로크 모듈(106)로부터 수신할 수 있다. 이러한 예시적 실시예에서, 고속 경로 회로(102) 그 자체는 대시 화살표(160)에 의해 표시된 바와 같이 펌웨어 처리 스택(104)으로 고속 경로 포맷 I/O 요청을 우회시키기 위한 처리를 수행할 수 있다.

[0027] 소정의 예시적 실시예에서, 영역 로크 모듈(106)은 잠재적으로 겹쳐지는 로크 요청의 위치를 알아내기 위해 신속한 검색이 가능하도록 트리 구조(tree structure)로 현재 승인된 로크(액세스 로크뿐만 아니라 우회형 로크)에 관한 정보를 유지한다. 트리 구조 엔트리는 영역 로크 모듈(106)에 의한 신속 검색을 허용하기 위해 블록 주소 범위에 의해 조직화된다. 트리 구조는 영역 로크 모듈(106) 내 적당히 설계된 주문형 회로에 의해 유지될 수 있거나, 혹은 다른 예시적 실시예에서 영역 로크 모듈(106)의 회로와 관련된 추가 주문형 회로 논리부에 의해 제공될 수 있다. 앞에서 언급한 관련 특허는 이러한 트리 구조의 구현 및, 이러한 트리 구조를 관리하도록 구성된 주문형 회로에 대한 예시적 상세사항을 제공한다. 이러한 트리 구조는 하나의 가능한 실시예를 나타낸다는 것을 알 것이다. 다수의 다른 데이터 구조 및 대응한 제어 로그를 사용하여 로크에 관한 정보를 레코딩 및 검색할 수 있다. 예를 들면 이들 목적을 위해 해시 테이블(hash table) 구조를 사용할 수 있다.

[0028] 고속 경로 포맷 I/O 요청의 우회와 우회형 영역 로크의 관련 관리에 관하여 저장 제어기(100)의 동작 방법의 추가적인 예시적 상세 사항은 다른 도면과 관련해 이후에 거론된다. 당해 기술분야에 통상의 지식을 가진 자는 우회형 영역 로크 특징 및 이의 양상이 (예를 들어 펌웨어/종래 처리 스택과 함께 LSI사의 고속 경로 회로와 같이) 적어도 두 처리 스택/프로세서를 가진 임의의 저장 제어기 구조에 적용될 수 있다는 것을 쉽게 알 것이다. 특징 및 이의 양상은 제어기의 (명목상 요청이 향하는) 일 처리 스택으로부터 다른 처리 스택으로 요청을 우회시키려는 신속한 판정이 바람직한 임의의 이러한 저장 제어기에 유리하게 적용될 수 있다. 더욱이 당해 기술분야

에 통상의 지식을 가진 자는 충분히 기능적인 저장 제어기에 존재할 수 있는 다수의 추가 및 증가의 요소를 쉽게 알 것이다. 이러한 추가적 증가 요소는 이러한 논의의 단순성과 간결성을 위해 여기서는 생략된다.

[0029]

도 2는 특징과 이의 양상에 따라서 고속 경로 포맷 I/O 요청을 저장 제어기의 고속 경로 처리 회로로부터 펌웨어 처리 스택으로 우회시키는 것을 관리하기 위한 예시적 방법을 기술하는 흐름도이다. 도 2 내지 도 8의 방법은 일반적으로 도 1의 저장 제어기(100)와 같은 향상된 저장 제어기에서 동작할 수 있다. 단계(200)에서, 향상된 저장 제어기는 고속 경로 포맷 I/O 요청을 수신한다. 이러한 요청은 향상된 저장 제어기의 고속 경로 회로에 직접 수신될 수 있거나, 또는 수신한 I/O 요청이 고속 경로 I/O 요청으로 포맷되는지의 여부를 판정하기 위해 동작가능한 제어기의 선택 모듈에 먼저 수신될 수 있고, 만약 그러하다면, 고속 경로 회로로 요청을 전송한다. 단계(202)에서, 향상된 저장 제어기는 고속 경로 포맷 I/O 요청이 승인된 로크의 현재 상태를 기반으로-이후에 더 거론되는 바와 같이 특히 우회형 로크를 기반으로- 펌웨어 처리 스택에 의해 처리되어야 하는 임의 블록 주소를 액세스하려고 시도중인지의 여부를 판정한다. 통상, 고속 경로 회로는 수신한 고속 경로 포맷 I/O 요청에서 식별된 블록 주소 범위에 대한 액세스를 요청하는 향상된 저장 제어기의 영역 로크 모듈로 로크 요청 신호를 적용할 것이다. 그 후, 영역 로크 모듈은 식별된 블록 주소 범위가 현재 승인된 임의 로크와 관련된 블록 주소 범위와 겹치는지의 여부에 대한 판정을 한다(단계202). 특히, 영역 로크 모듈은 요청된 블록 주소 범위가 영역 로크 모듈에 의해 이전에 승인된 임의 우회형 로크와 겹치는지의 여부를 판정할 것이다. 이러한 블록 주소 겹쳐짐이 확인되지 않는다면, 단계(204)는 고속 경로 회로의 일반 처리를 계속하여, 고속 경로 포맷 I/O 요청에 따라서 식별된 블록 주소 범위를 액세스한다(그리고 그 후에 영역 로크 모듈에 의해 승인된 액세스 로크를 해제한다). 이 대신에, 영역 로크 모듈이 고속 경로 포맷 I/O 요청에 의해 액세스할 블록 주소 범위가 이전에 승인된 우회형 로크 요청에서 블록 주소 범위의 임의 일부와 겹쳐진다고 판정하면, 단계(206)는 I/O 요청의 완료를 위해 펌웨어 처리 스택으로 고속 경로 포맷 I/O 요청을 우회시키도록 동작할 수 있다. 향상된 저장 제어기의 펌웨어 처리 스택으로 고속 경로 포맷 I/O 요청을 우회시킴으로써, 데이터 손상을 피할 수 있으며, (현재 실행되는 바와 같이) 향상된 저장 제어기의 고속 경로 회로를 완전히 디스에이블시키지 않고 펌웨어 처리 스택 최적화를 유리하게 이용할 수 있다. 따라서 펌웨어 처리 스택으로 우회시켜야 할 요청을 그와 같이 쉽게 식별할 수 있으며 고속 경로 회로에 의해 처리될 수 있는 고속 경로 포맷 I/O 요청에 대하여 향상된 저장 제어기의 바람직한 성능 레벨을 유지할 수 있다.

[0030]

도 3은 영역 로크를 사용하여 고속 경로 포맷 I/O 요청을 고속 경로 회로로부터 향상된 저장 제어기의 펌웨어 처리 스택으로 우회시키는 특징 및 이의 양상에 따른 방법의 예시적 추가 상세사항을 기술하는 흐름도이다. 도 3의 방법은 향상된 저장 제어기의 영역 로크 모듈과 함께 동작될 수 있는 향상된 저장 제어기의 고속 경로 회로에 의한 처리를 반영한다. 단계(300)에서, 호스트 시스템으로부터 고속 경로 포맷 I/O 요청을 수신한다(즉 향상된 저장 제어기의 예비 선택 구성요소를 통해 간접적으로 수신하거나 또는 호스트로부터 직접 수신). 그 후, 단계(302)는 고속 경로 포맷 I/O 요청에 의한 액세스를 위해 식별된 블록 주소 범위에 대한 액세스를 요청하는 영역 로크 모듈로 로크 요청 신호를 적용한다. 블록 주소 범위는 수신한 고속 경로 포맷 I/O 요청 내에서 정보에 의해 식별된다. 따라서 영역 로크 모듈로 적용되는 신호는 액세스 로크에 대한 요청뿐만 아니라 I/O 요청에 의해 영향받은 블록 주소 범위의 모두를 가리킬 수 있다. 소정의 예시적 실시예에서, 영역 로크 모듈로 적용되는 신호는 향상된 저장 제어기의 펌웨어 처리 스택으로 요청을 우회시키도록 요구할 수 있는 모든 정보를 영역 로크 모듈에 제공하기 위해 충분한 상황 정보를 더 포함할 수 있다. 단계(304)에서, 고속 경로 회로는 그의 요청된 로크가 승인되었는지의 여부를 판정한다. 소정의 예시적 실시예에서, 영역 로크 모듈은 요청된 로크를 승인 또는 거부되었음을 나타내는 신호를 고속 경로 회로로 반환할 수 있다. 요청된 로크가 승인되었다면, 단계(306, 308)는 고속 경로 회로의 동작에 의해 고속 경로 포맷 I/O 요청의 처리를 완료하도록(단계306), 그리고 영역 로크 모듈로 적절한 해제 신호를 적용함으로써 승인된 액세스 로크를 해제하도록(단계 310) 동작할 수 있다. 소정 실시예에서, 로크할 요청된 블록 범위가 펌웨어 처리 스택이 획득한 우회형 로크와 겹치므로 요청된 로크를 거부하는 경우, 영역 로크 모듈은 단계(302)에서 적용된 로크 요청 신호에 제공되는 상황 정보를 기반으로 펌웨어 처리 스택으로 I/O 요청을 우회시키는데 필요한 모든 처리를 수행한다. 이러한 실시예에서, 고속 경로 회로는 임의 추가 처리를 수행할 필요가 없고, 단지 다음 고속 경로 포맷 I/O 요청의 수신을 기다린다. 다른 예시적 실시예에서, 영역 로크 모듈은 그의 영역 로크 요청의 거부를 나타내며, 거부는 우회형 영역 로크 요청으로 이전에 승인되었기 때문임을 더 나타내는 신호를 고속 경로 회로로 반환한다. 이러한 예시적 실시예에서, 단계(312)는 펌웨어 처리 스택으로 고속 경로 포맷 I/O 요청을 우회시키거나 또는 전송시키도록 동작할 수 있다(예를 들면 I/O 요청은 영역 로크 모듈 그자체에 의해서라기 보다는 고속 경로 회로의 동작에 의해 펌웨어 처리 스택으로 우회된다).

[0031]

도 4는 향상된 저장 제어기 내 영역 로크 모듈(예를 들면 도 1의 저장 제어기의 영역 로크 모듈 106)의 처리의

추가적인 상세사항을 제공하는 흐름도이다. 도 4의 처리는 향상된 저장 제어기의 고속 경로 회로로부터 수신한 로크 요청 신호에 응답하여 영역 로크 모듈을 처리하는 것을 주로 기술한다. 펌웨어 처리 스택으로부터의 액세스 로크 요청의 수신에 응답하여 유사한 처리를 수행할 수 있다. 이러한 유사한 처리는 당해 기술분야에 통상의 지식을 가진 자에게 쉽게 명백할 것이며, 따라서 이 논의의 단순성과 간결성을 위해 여기서 생략된다. 고속 경로 회로로부터의 액세스 로크 요청의 수신에 응답하여, 단계(400)는 (각 엔트리와 관련된 블록 주소 범위에 의해 판정되어지는 적절한 장소에 추가된) 로크의 그의 트리 구조로 새 엔트리를 추가한다. 단계(402)는 로크 요청에서 식별되는 블록 주소 범위가 이전에 승인된 로크 요청(예를 들면 액세스 로크 또는 우회형 로크를 포함한 임의 유형의 미리 승인된 로크 요청)과 관련있는 블록 주소 범위의 임의 부분과 겹치는지의 여부를 판정한다. 겹치지 않으면, 단계(404)는 식별된 블록 주소 범위에 대해 고속 경로 회로로 승인되는 새 로크 엔트리를 표시하고, I/O 동작을 실행하기 위해 그의 동작을 계속하도록 고속 경로 회로로 시그널링한다. 단계(402)에서 로크할 요청 블록 주소 범위가 하나 이상의 이전 승인된 로크 요청과 관련된 블록 주소 범위와 겹친다고 판정하면, 단계(406)는 이전 승인된 로크의 임의 중복이 우회형 로크인지의 여부를 판정한다. 만약 그렇다면, 단계(408)는 펌웨어 처리 스택으로 우회된(곧 그렇게 될) I/O 요청에 대한 우회된 로크로서 새로이 추가된 로크 엔트리를 표시한다.

[0032] 단계(408)에 뒤이어, 또는 단계(406)에서 새 로크 요청이 승인된 우회형 로크와 겹치지 않는다고 판정한다면, 처리는 단계(410)으로 진행되어, 임의의 다른 것들(예를 들면 액세스 로크와 같은 비우회형)이 고속 경로 회로로부터 수신한 로크 요청의 블록 주소 범위와 겹칠 수 있는지의 여부를 판정한다. 만약 그렇다면, 단계(412)는 다른 이전에 승인된 로크의 모두가 해제되기를 기다린다.

[0033] 단계(412)에 뒤이어, 또는 단계(410)에서 이러한 다른 로크가 겹치지 않는다고 판정하면, 처리는 단계(414)로 진행되고, 여기서 로크 트리에 새로 추가된 엔트리가 이제 우회되는 것으로 표시되었는지의 여부를 판정한다. 만약 아니라면, 단계(404)는 로크 엔트리가 식별된 블록 범위를 위해 고속 경로 회로로 이제 승인되었다는 것을 표시하고, 고속 경로 회로에 승인을 시그널링하여, 고속 경로 회로가 고속 경로 포맷 I/O 요청의 처리를 계속하게 한다.

[0034] 단계(414)는 새 로크 엔트리가 이제 우회된 것으로 표시되었다고 판정하고, 단계(416)는 식별된 블록 주소 범위에 대해 펌웨어 처리 스택으로 승인되는 것으로 새로이 추가된 로크 엔트리를 표시한다. 따라서 고속 경로 포맷 I/O 요청은 적절한 때에 최종 처리를 위해 펌웨어 처리 스택으로 우회된다. 고속 경로 회로는 더 이상의 추가 요청 처리를 수행하지 않는다. 다른 예시적 실시예에서, 앞에서 언급한 바와 같이, 영역 로크 모듈은 그의 로크 요청이 거부되었으며 펌웨어 처리 스택으로 우회되어야 한다는 것을 고속 경로 회로에 간단히 신호함으로써, 고속 경로 회로 그자체가 요청을 고속 경로 처리 스택으로 우회시키는 처리를 수행하게 한다.

[0035] 또한 도 4에 도시된 방법은 이전에 승인된 로크 요청을 해제하기 위한 방법 처리 단계이다. I/O 요청 처리 스택(예를 들면 고속 경로 회로 또는 펌웨어 처리 스택)이 승인된 로크를 얻기 위한 요청의 처리를 완료하였을 때, 단계(420)는 처리 스택으로부터 수신한 적절한 해제 신호에 응답하여 이전에 승인된 로크를 해제한다.

[0036] 도 5, 도 6 및 도 7은 펌웨어 처리 스택 내 특정한 처리에 응답하여 우회형 영역 로크를 요청하고, 이전에 승인된 우회형 영역 로크를 해제하기 위하여, 향상된 저장 제어기의 펌웨어 처리 스택(예를 들면 도 1의 저장 제어기 100의 펌웨어 처리 스택 104) 내 처리의 예시적인 상세사항을 추가로 도시한다. 앞에서 언급한 바와 같이, 펌웨어 처리 스택이 라이트-백 모드에서 그의 캐시 메모리를 이용하는 기록 요청을 처리할 때, 그는 캐시 메모리에 상주하는 더티 데이터를 위한 블록 주소 범위에 대해 우회형 영역 로크가 승인되도록 요청한다(즉 아직 저장 장치로 플러싱 또는 포스트되지 않은 더티 데이터). 도 5는 그의 캐시 메모리에 더티 데이터를 저장하는 것에 응답하여 이러한 우회형 로크를 요청하기 위해 펌웨어 처리 스택의 처리의 예시적인 상세사항을 추가로 기술한다. 단계(500)에서, 펌웨어 처리 스택은 라이트-백 모드에서 그의 캐시 메모리에 데이터를 저장함으로써 기록 I/O 요청을 처리한다. 단계(502)에서, 펌웨어 처리 스택은 캐시 메모리가 아직 저장 장치로 포스트 또는 플러싱되지 않은 블록 주소에 대응한 데이터를 포함한다는 것을 의미하는 "더티 데이터"로서, 캐시 메모리에서 영향받은 블록을 표시한다. 단계(504)에서, 펌웨어 처리 스택은 캐시 메모리에 저장했던 더티 데이터를 위한 블록 주소 범위를 식별하는 우회형 로크 요청 신호를 영역 로크 모듈로 적용한다.

[0037] 도 6은 더티 데이터가 저장 장치로 성공적으로 플러싱 또는 포스트될 때 이전에 승인된 우회형 로크를 해제하기 위한 펌웨어 처리 스택의 추가 처리를 기술한다. 단계(600)에서, 펌웨어 처리 스택은 영구 저장을 위해 캐시 메모리로부터 저장 장치로 더티 데이터를 플러싱 또는 포스트하는 적절한 처리를 수행한다. 그 후, 단계(602)는 캐시 메모리에서 "더티 데이터"와 더 이상 관련없는 블록 주소 범위를 식별하는 영역 로크 모듈로 우회형 로

크 해제 신호를 적용한다. 그 다음, 영역 로크 모듈은 이전에 승인된 우회형 로크를 해제할 수 있다.

[0038] 도 7은 불량 블록을 나타낼 수 있는 블록 주소 범위에 대해 승인된 우회형 로크를 요청하는 펌웨어 처리 스택의 처리의 예시적 상세사항을 추가로 기술하는 흐름도이다. 앞에서 언급한 바와 같이, 펌웨어 처리 스택은 특정 한 블록 또는 블록 주소 범위를 이용시에 만나게 되는 다양한 유형의 오류 중의 임의 유형에 응답하여 블록을 "불량"으로 지명할 수 있다. 단계(700)에서, 펌웨어 처리 스택은 펌웨어 처리 스택과 관련된 불량 블록 메모리에 저장된 그의 불량 블록 리스트로 하나 이상의 블록 주소를 추가한다. 단계(702)에서, 펌웨어 처리 스택은 펌웨어 처리 스택의 처리에 의해 "불량" 블록으로서 식별되는 식별된 블록 주소 범위에 대해 승인된 우회형 영역 로크를 요청하는 우회형 로크 요청을 영역 로크 모듈로 적용한다.

[0039] 도 8은 (예를 들면 이전에 표시된 불량 블록으로 성공적으로 기록 시에) 이전에 불량으로 표시된 블록이 성공적으로 갱신되었을 때에 이전에 승인된 우회형 로크를 해제하기 위한 펌웨어 처리 스택의 추가 처리를 기술한다. 단계(800)에서, 펌웨어 처리 스택은 (예를 들어 불량 블록/블록들로의 성공적인 기록 동작의 완료와 관련하여) 불량 블록 리스트로부터 하나 이상의 블록을 제거하기 위해 적절한 처리를 수행한다. 그 후, 단계(802)는 갱신된 불량 블록 리스트에서 불량 블록과 더 이상 관련없는 블록 주소 범위를 식별하는 우회형 로크 해제를 영역 로크 모듈로 적용한다. 그 다음, 영역 로크 모듈은 이전에 승인된 우회형 로크를 해제할 수 있다.

[0040] 당해 기술분야에 통상의 지식을 가진 자는 도 2 내지 도 8에 관하여 기술한 방법과 같이 완전히 기능적인 방법에 존재할 수 있는 다수의 추가 및 등가의 단계를 쉽게 인식할 것이다. 이러한 추가 및 등가 단계는 이 논의의 단순성과 간결성을 위해 여기서 생략된다.

[0041] 여기에 개시된 실시예는 소프트웨어, 하드웨어, 펌웨어 또는 이들의 다양한 결합의 형태를 가질 수 있다. 일 특정한 실시예에서, 소프트웨어는 저장 장치의 처리 시스템(예를 들면 저장 제어기)에게 여기에 개시된 다양한 동작을 수행하게 하는데 사용된다. 도 9는 예시적 실시예에서 원하는 기능을 수행하기 위해 컴퓨터 판독가능 매체에 저장된 프로그래밍된 인스트럭션을 실행하도록 동작가능한, 도 1의 저장 제어기(100) 내 펌웨어 처리 스택(104)과 같은 처리 시스템(900)을 도시한다. 처리 시스템(900)은 컴퓨터 판독가능 저장매체(912) 상에 유형으로 구현된 프로그래밍된 인스트럭션을 실행함으로써 위의 동작을 수행하도록 동작할 수 있다. 이와 관련하여, 본 발명의 실시예는 컴퓨터 또는 임의의 다른 인스트럭션 실행 시스템에 사용되는 프로그램 코드를 제공하는 컴퓨터 판독가능 매체(912)를 통해 액세스될 수 있는 컴퓨터 프로그램의 형태를 가질 수 있다.

[0042] 이 설명을 위하여, 컴퓨터 판독가능 저장매체(912)는 컴퓨터에 사용되는 프로그램을 포함 또는 저장할 수 있는 임의의 것일 수 있다. 컴퓨터 판독가능 저장매체(912)는 전자, 자기, 광학, 전자기, 적외선 또는 반도체 장치일 수 있다. 컴퓨터 판독가능 저장매체(912)의 예는 고체상태 메모리(예를 들면 플래시 메모리), 자기 테잎, 착탈 가능 컴퓨터 디스켓, RAM(random access memory), ROM(read-only memory), 강성 자기 디스크 및 광 디스크를 포함한다. 광 디스크의 현 예는 CD-ROM(compact disk-read only memory), CD-R/W(compact disk-read/write) 및 DVD를 포함한다.

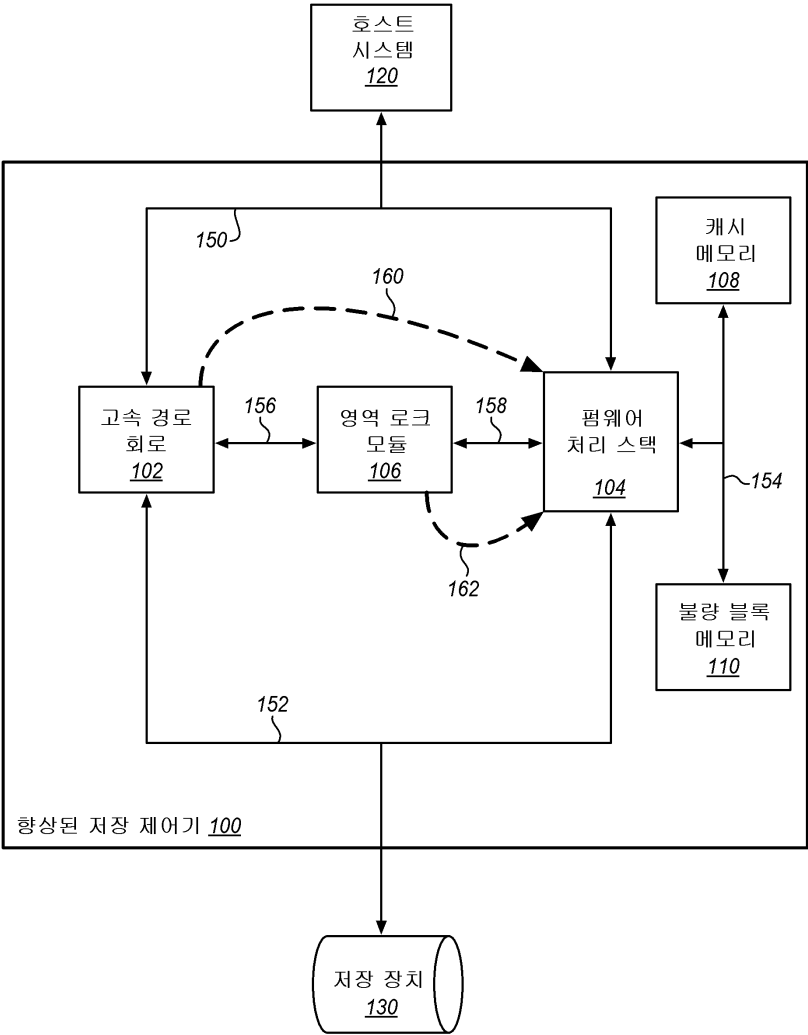
[0043] 프로그램 코드를 저장 및/또는 실행하는데 적당한 처리 시스템(900)은 시스템 버스(950)를 통해 프로그램 및 데이터 메모리(904)와 캐시 메모리 서브시스템(910)로 연결된 적어도 하나의 프로세서(902)를 포함한다. 프로그램 및 데이터 메모리(904)는 실행 동안에 벌크 저장소로부터 코드 및/또는 데이터를 검색하는 횟 수를 감소시키기 위하여 적어도 소정 프로그램 코드 및/또는 데이터의 임시 저장을 제공하는 캐시 메모리, 프로그램 코드 및 벌크 저장소의 실제 실행 동안에 사용되는 로컬 메모리를 포함할 수 있다.

[0044] 입력/출력 또는 I/O 장치(906)가 더 포함될 수 있다(예를 들어 저장 장치와 호스트 시스템과 연결을 위한 장치). 앞에서 거론한 바와 같이, 고속 경로 회로(908)와 영역 로크 모듈(914)은 또한 처리 시스템(900)이 고속 경로 포맷 I/O 요청을 신속하게 처리하고 블록 주소의 로킹을 조정하게 할 수 있도록 하기 위해 시스템과 통합될 수 있다.

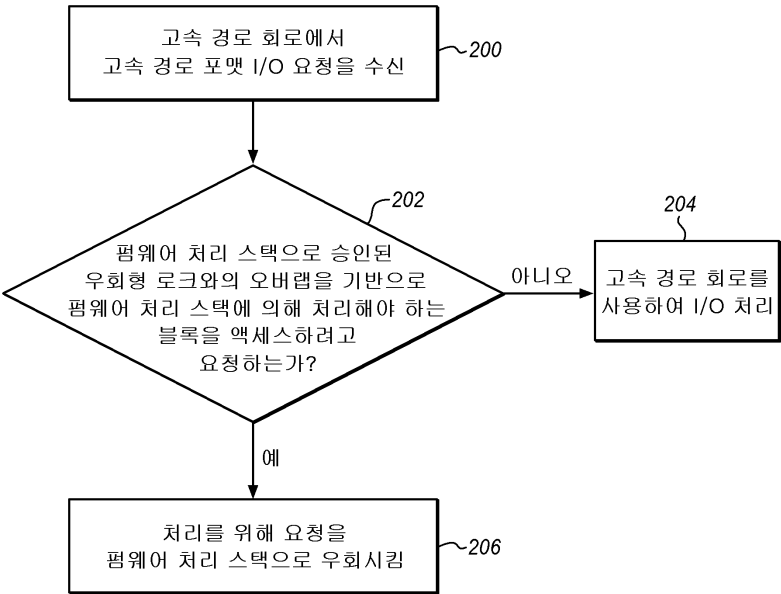
[0045] 본 발명을 도면 및 앞의 설명에 도시 및 기술하였지만, 이러한 도시 및 설명은 예로서 간주되며 특징을 제한하려는 것이 아니다. 본 발명의 일 실시예와 이의 작은 변경이 도시 및 기술되었다. 특히, 예시적인 소프트웨어 또는 펌웨어로서 도시 및 기술된 특징은 주문형 논리 회로로서 동등하게 구현될 수 있고, 그의 반대도 가능하다. 본 발명의 사상 내에 있는 모든 변경 및 변형에 대해 보호하는 것이 바람직하다. 당해 기술분야에 통상의 지식을 가진 자는 본 발명의 범주 내에 있는 전술한 실시예의 변경을 알 것이다. 결과적으로, 본 발명은 앞에서 거론한 특정 예 및 도시로 제한되지 않고 다음의 특허청구범위 및 그들의 등가물에 의해서만 제한된다.

도면

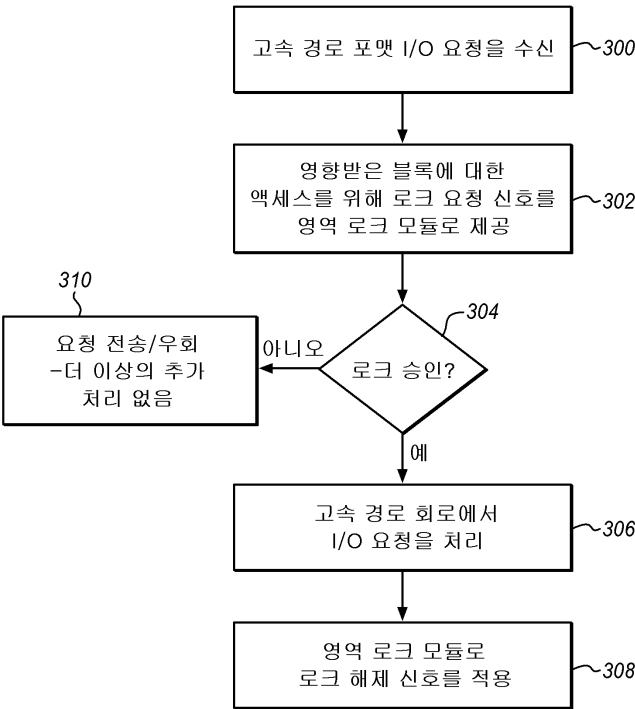
도면1



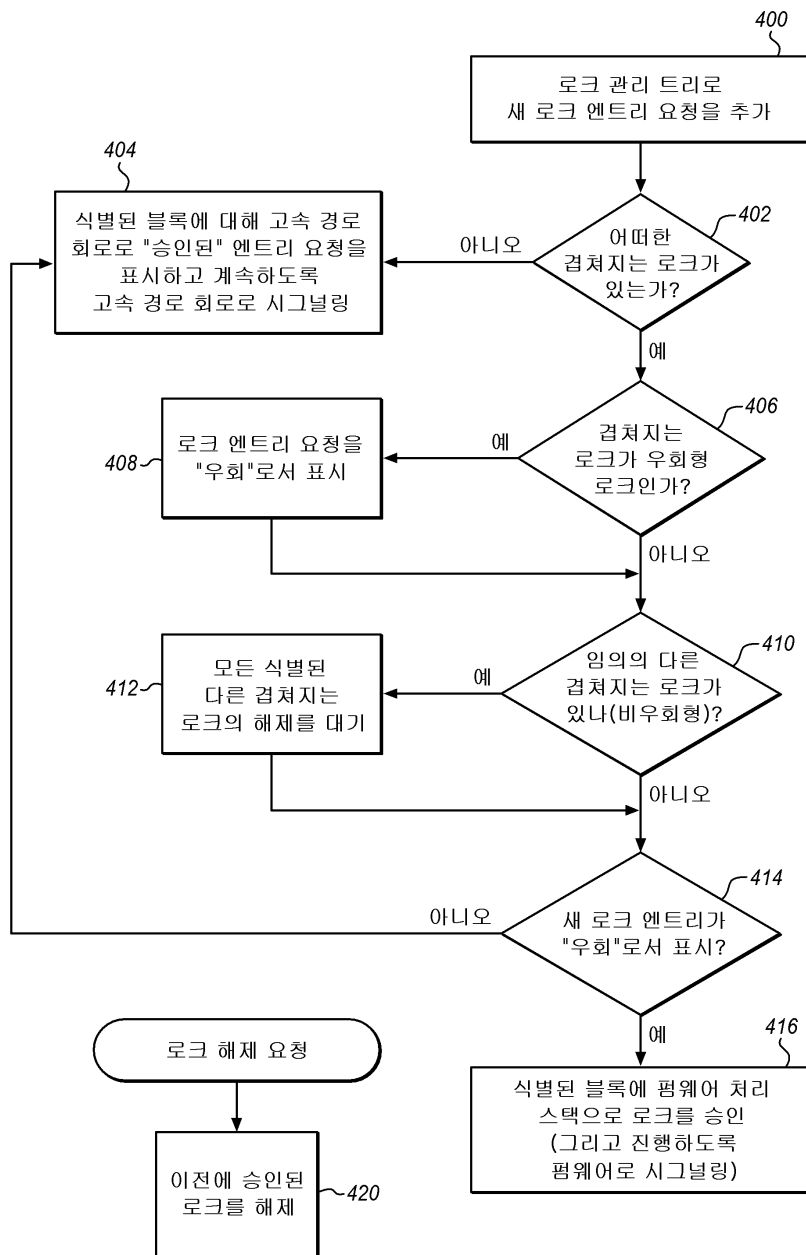
도면2



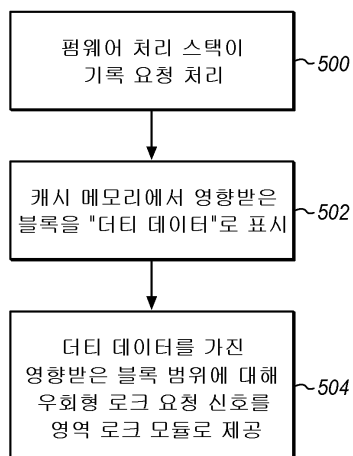
도면3



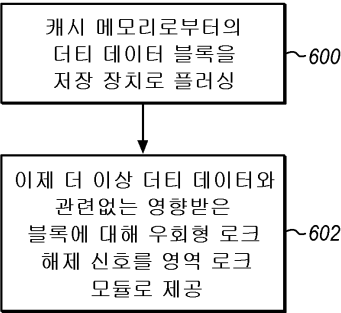
도면4



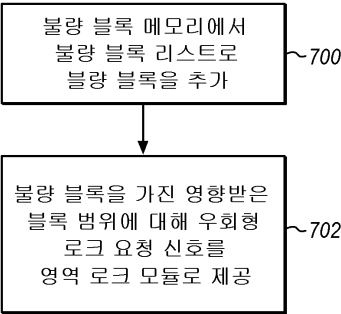
도면5



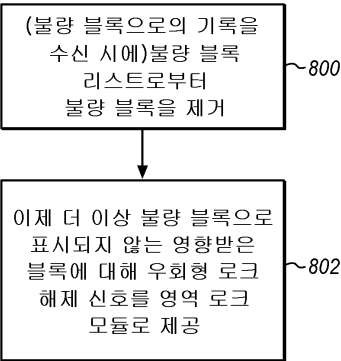
도면6



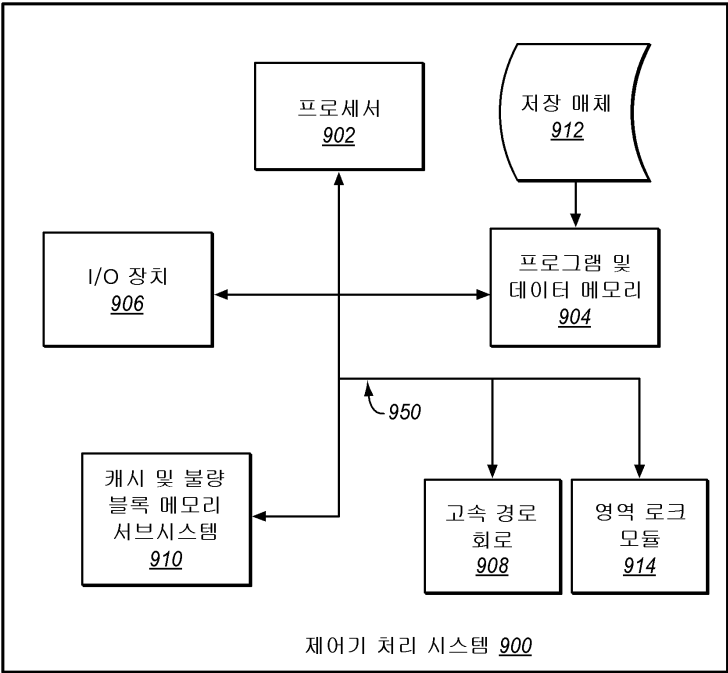
도면7



도면8



도면9



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 제13항

【변경전】

상기 블록 어드레스 범위

【변경후】

상기 블록 주소 범위