

- 심사관 : 호진기

- 특허법인(유)남아이피그룹, 특허법인 남앤남

- (54) 발명의 명칭 메모리 액세스 동작 중지/재개

메모리 제어기는 호스트에 접속된 솔리드 스테이트 저장 디바이스의 메모리 어레이의 플래시 메모리 인터페이스를 통해 메모리 액세스 동작들을 관리한다. 메모리 제어기는 메모리 어레이에서 제1 메모리 액세스 동작을 실행한다. 제1 메모리 액세스 동작은 제1 우선순위를 갖는다. 메모리 제어기는, 메모리 어레이에서 실행하기 위해 이용가능하고 제1 우선순위보다 더 높은 우선순위를 갖는 중지 메모리 액세스 동작을 검출한다. 검출 동작은 중지 메모리 액세스 동작들과 중지하지 않은 메모리 액세스 동작들 사이를 구별한다. 메모리 제어기는, 메모리 어레이에서의 제1 메모리 액세스 동작의 실행을 중지하고, 제1 우선순위보다 더 높은 우선순위들을 갖고 메모리 어레이에서 실행하기 위해 이용가능한 하나 이상의 메모리 액세스 동작들을 실행한다. 메모리 제어기는 메모리 어레이에서 제1 메모리 액세스 동작의 실행을 재개한다.

- 1 -

(52) CPC특허분류

G06F 3/061 (2013.01)

G11C 16/0483 (2013.01)

G11C 16/10 (2013.01)

(72) 발명자

알후시엔, 압델하킴

미국 95031 캘리포니아 새너제이 에메랄드 힐스 서클 2288

하라트쉬, 에리히, 프란츠

미국 95135 캘리포니아 새너제이 크레센트 빌리지 서클 310 아파트먼트 1229

하우, 스티븐

미국 55902 미네소타 로체스터 메이오우드 커몬 서클 사우스웨스트 2951

(56) 선행기술조사문헌

KR1020080099336 A*

US06930925 B2*

US09176764 B1*

US20180004410 A1*

*는 심사관에 의하여 인용된 문헌

명세서

청구범위

청구항 1

호스트에 접속된 솔리드 스테이트 저장 디바이스(solid-state storage device)의 메모리 어레이의 플래시 메모리 인터페이스에 포함되는 메모리 제어기에 의해 수행되는 메모리 액세스 동작들을 관리하는 방법으로서,

상기 메모리 어레이에서 제1 메모리 액세스 동작을 실행하는 단계 - 상기 제1 메모리 액세스 동작은 제1 우선순위를 가짐 -;

상기 메모리 어레이에서 실행하기 위해 이용가능하고 상기 제1 우선순위보다 더 높은 우선순위를 갖는 적어도 하나의 중지 메모리 액세스 동작을 검출하는 단계 - 상기 검출 동작은 중지 메모리 액세스 동작들과 중지하지 않은(non-suspending) 메모리 액세스 동작들 사이를 구별함 -;

상기 검출 동작에 응답하여, 상기 메모리 어레이에서의 상기 제1 메모리 액세스 동작의 실행을 중지하는 단계;

상기 중지 동작에 응답하여, 상기 적어도 하나의 중지 메모리 액세스 동작을 실행하는 단계;

상기 제1 우선순위보다 높은 우선순위들을 갖고, 적어도 하나의 시한 중지 확장 윈도우(timed suspend extension window)의 만료 전에 상기 메모리 어레이에서 실행하기 위해 이용가능하게 되는 상기 메모리 어레이에서의 하나 이상의 새롭게 이용가능한 메모리 액세스 동작들을 실행하는 단계 - 상기 적어도 하나의 시한 중지 확장 윈도우는 상기 적어도 하나의 중지 메모리 액세스 동작을 실행하는 동작의 종료 후에 시작됨 -; 및

상기 실행 동작을 완료하는 것에 응답하여, 상기 메모리 어레이에서의 상기 제1 메모리 액세스 동작의 실행을 재개하는 단계를 포함하고,

상기 하나 이상의 새롭게 이용가능한 메모리 액세스 동작들을 실행하는 단계는 상기 재개 동작 전에 시작되는, 방법.

청구항 2

삭제

청구항 3

제1항에 있어서, 상기 메모리 어레이에서의 하나 이상의 새롭게 이용가능한 메모리 액세스 동작들을 실행하는 동작은, 상기 제1 메모리 액세스 동작이 중지된 채로 유지되는 동안 다수의 시한 중지 확장 윈도우들에 대해 반복적으로 수행되는, 방법.

청구항 4

제1항에 있어서, 상기 제1 메모리 액세스 동작은 구간들의 시퀀스로 나뉘어지고, 상기 제1 메모리 액세스 동작은 중지되기 전에 상기 구간들 중 실행 구간에서 실행되고, 상기 중지 동작은,

상기 제1 메모리 액세스 동작의 상기 실행 구간이 완료될 때까지 상기 제1 메모리 액세스 동작의 중지를 지연시키는 단계; 및

상기 제1 메모리 액세스 동작의 상기 실행 구간의 완료 후에 상기 메모리 어레이에서의 상기 제1 메모리 액세스 동작의 실행을 중지시키는 단계를 포함하는, 방법.

청구항 5

제4항에 있어서, 상기 구간들의 시퀀스 중 상기 실행 구간 뒤에 다음 구간이 뒤따르고, 상기 재개 동작은,

상기 시퀀스 중 상기 다음 구간에서 상기 메모리 어레이에서의 상기 제1 메모리 액세스 동작의 실행을 재개하는 단계를 포함하는, 방법.

청구항 6

삭제

청구항 7

제1항에 있어서, 상기 제1 메모리 액세스 동작은 프로그램 동작이고, 상기 프로그램 동작보다 더 높은 우선순위를 갖는 다른 메모리 액세스 동작들 중 적어도 하나는 호스트 판독 동작인, 방법.

청구항 8

제1항에 있어서, 다수의 레벨의 더 높은 우선순위 메모리 액세스 동작들에 대해 네스트된(nested) 레벨들의 중지 및 재개가 수행되는, 방법.

청구항 9

호스트에 접속된 솔리드 스테이트 저장 디바이스에서 메모리 액세스 동작들을 관리하기 위한 시스템으로서,

하나 이상의 메모리 어레이들; 및

상기 하나 이상의 메모리 어레이들에 커플링된 메모리 제어기를 포함하는 플래시 메모리 인터페이스를 포함하고, 상기 메모리 제어기는,

상기 하나 이상의 메모리 어레이들 중 하나의 메모리 어레이에서 제1 메모리 액세스 동작을 실행하도록 - 상기 제1 메모리 액세스 동작은 제1 우선순위를 가짐 -,

상기 메모리 어레이에서 실행하기 위해 이용가능하고 상기 제1 우선순위보다 더 높은 우선순위를 갖는 적어도 하나의 중지 메모리 액세스 동작을 검출하도록 - 상기 검출 동작은 상기 중지 메모리 액세스 동작들과 중지하지 않은 메모리 액세스 동작들 사이를 구별함 -,

상기 검출 동작에 응답하여, 상기 메모리 어레이에서의 상기 제1 메모리 액세스 동작의 실행을 중지하도록,

상기 제1 메모리 액세스 동작의 실행을 중지하는 것에 응답하여, 상기 적어도 하나의 중지 메모리 액세스 동작을 실행하도록,

적어도 하나의 재개 트리거 조건을 검출하도록, 그리고

상기 실행 동작의 완료 전에, 상기 적어도 하나의 재개 트리거 조건의 검출에 응답하여, 상기 메모리 어레이에서의 상기 제1 메모리 액세스 동작의 실행을 재개하도록 구성되고,

상기 적어도 하나의 재개 트리거 조건은,

중지 실행 동작 동안 실행되는 중지 메모리 액세스 동작들의 총 수가 미리 결정된 중지 메모리 액세스 동작들의 임계치를 충족 또는 초과하는 것; 또는

실행된 더 낮은 우선순위 메모리 액세스 동작들에 대한 실행된 더 높은 우선순위 메모리 액세스 동작들의 비가 미리 결정된 임계치를 충족하거나 초과하는 것인, 시스템.

청구항 10

제9항에 있어서, 상기 메모리 제어기는,

상기 적어도 하나의 메모리 액세스 동작을 실행하는 동작의 종료 후에 적어도 하나의 시한 중지 확장 윈도우를 시작하는 타이머를 포함하고, 상기 메모리 제어기는, 상기 재개 동작 전에, 상기 제1 우선순위보다 더 높은 우선순위들을 갖고, 상기 적어도 하나의 시한 중지 확장 윈도우의 만료 전에 상기 메모리 어레이에서 실행하기 위해 이용가능하게 되는 상기 메모리 어레이에서의 하나 이상의 새롭게 이용가능한 메모리 액세스 동작들을 실행하도록 추가로 구성되는, 시스템.

청구항 11

제10항에 있어서, 상기 제1 메모리 액세스 동작은 구간들의 시퀀스로 나뉘지고, 각각의 구간은 상기 적어도 하나의 시한 중지 확장 윈도우보다 시간이 더 길고, 상기 제1 메모리 액세스 동작은 중지되기 전에 상기 구간들 중 실행 구간에서 실행되고, 상기 메모리 제어기는,

상기 제1 메모리 액세스 동작의 상기 실행 구간이 완료될 때까지 상기 제1 메모리 액세스 동작의 중지를 지연시

키고,

상기 제1 메모리 액세스 동작의 상기 실행 구간의 완료 후에 상기 메모리 어레이에서의 상기 제1 메모리 액세스 동작의 실행을 중지시킴으로써 상기 중지 동작을 수행하도록 추가로 구성되는, 시스템.

청구항 12

제11항에 있어서, 상기 구간들의 시퀀스 중 상기 실행 구간 뒤에 다음 구간이 뒤따르고, 상기 메모리 제어기는, 상기 시퀀스 중 상기 다음 구간에서 상기 메모리 어레이에서의 상기 제1 메모리 액세스 동작의 실행을 재개함으로써 상기 제1 메모리 액세스 동작의 실행을 재개하도록 추가로 구성되는, 시스템.

청구항 13

제9항에 있어서, 상기 메모리 제어기는, 상기 제1 메모리 액세스 동작이 중지된 채로 유지되는 동안 다수의 시한 중지 확장 윈도우들에 대해 반복적으로 상기 메모리 어레이에서의 하나 이상의 새롭게 이용가능한 메모리 액세스 동작들을 실행하도록 추가로 구성되는, 시스템.

청구항 14

제9항에 있어서, 상기 제1 메모리 액세스 동작은 프로그램 동작이고, 상기 프로그램 동작보다 더 높은 우선순위를 갖는 다른 메모리 액세스 동작들 중 적어도 하나는 호스트 판독 동작인, 시스템.

청구항 15

삭제

청구항 16

제9항에 있어서, 더 높은 우선순위를 갖는 메모리 액세스 동작들은 더 낮은 우선순위를 갖는 메모리 액세스 동작들보다 빨리 이용가능하게 되는, 시스템.

청구항 17

호스트에 접속된 솔리드 스테이트 저장 디바이스의 메모리 어레이의 플래시 메모리 인터페이스를 통해 메모리 액세스 동작들을 관리하는 프로세스를 프로세서 상에서 실행하기 위한 프로세서 실행가능 명령어들을 인코딩하는 하나 이상의 유형적 프로세서 판독가능 저장 매체로서, 상기 프로세서는,

상기 메모리 어레이에서 제1 메모리 액세스 동작을 실행하는 것 - 상기 제1 메모리 액세스 동작은 제1 우선순위를 가짐 -;

상기 메모리 어레이에서 실행하기 위해 이용가능하고 상기 제1 우선순위보다 더 높은 우선순위를 갖는 적어도 하나의 중지 메모리 액세스 동작을 검출하는 것 - 상기 검출 동작은 상기 중지 메모리 액세스 동작들과 중지하지 않은 메모리 액세스 동작들 사이를 구별함 -;

상기 검출 동작에 응답하여, 상기 메모리 어레이에서의 상기 제1 메모리 액세스 동작의 실행을 중지하는 것;

상기 중지 동작에 응답하여, 상기 적어도 하나의 중지 메모리 액세스 동작을 실행하는 것;

상기 제1 우선순위보다 높은 우선순위들을 갖고, 적어도 하나의 시한 중지 확장 윈도우의 만료 전에 상기 메모리 어레이에서 실행하기 위해 이용가능하게 되는 상기 메모리 어레이에서의 하나 이상의 새롭게 이용가능한 메모리 액세스 동작들을 실행하는 것 - 상기 적어도 하나의 시한 중지 확장 윈도우는 상기 적어도 하나의 중지 메모리 액세스 동작을 실행하는 동작의 종료 후에 시작됨 -; 및

상기 실행 동작을 완료하는 것에 응답하여, 상기 메모리 어레이에서의 상기 제1 메모리 액세스 동작의 실행을 재개하는 것을 포함하고,

상기 하나 이상의 새롭게 이용가능한 메모리 액세스 동작들을 실행하는 것은 상기 재개 동작 전에 시작되는, 하나 이상의 유형적 프로세서 판독가능 저장 매체.

청구항 18

삭제

청구항 19

제17항에 있어서, 상기 제1 메모리 액세스 동작은 프로그램 동작이고, 상기 프로그램 동작보다 더 높은 우선순위를 갖는 다른 메모리 액세스 동작들 중 적어도 하나는 호스트 판독 동작인, 하나 이상의 유형적 프로세서 판독가능 저장 매체.

청구항 20

제17항에 있어서, 상기 제1 메모리 액세스 동작은 프로그램 동작이고, 상기 프로그램 동작보다 더 높은 우선순위를 갖는 다른 메모리 액세스 동작들 중 적어도 하나는 맵 판독 동작인, 하나 이상의 유형적 프로세서 판독가능 저장 매체.

발명의 설명

배경 기술

- [0001] 3D 플래시 메모리는 2D 플래시 메모리 기술들보다 더 긴 프로그램 시간들을 수반한다. 더 긴 프로그램 시간들은, 다수의 메모리 페이지들이 단일 동작으로 프로그래밍되는 원 샷(one shot) 프로그래밍의 실행에 의해 증가될 수 있으며, 이는 사용 중인 다이(busy die) 상에 상당한 병목현상들을 야기할 수 있다. 혼합된 판독/기록 작업부하들에서, 더 긴 프로그램 시간들은 다른 더 높은 우선순위 동작들에 대한 병목현상을 생성할 수 있다. 예를 들어, 판독 동작들은, 다이 상에서의 긴 프로그램 동작에 의해 차단될 때 긴 레이턴시를 경험할 수 있다.

발명의 내용

- [0002] 본 명세서에 개시된 구현예들은, 호스트에 접속된 솔리드 스테이트 저장 디바이스(solid-state storage device)의 메모리 어레이의 플래시 메모리 인터페이스를 통해 메모리 액세스 동작들을 관리하기 위한 시스템들 및 방법들을 제공한다. 메모리 제어기는 메모리 어레이에서 제1 메모리 액세스 동작을 실행한다. 제1 메모리 액세스 동작은 제1 우선순위를 갖는다. 메모리 제어기는, 메모리 어레이에서 실행하기 위해 이용가능하고 제1 우선순위보다 더 높은 우선순위를 갖는 적어도 하나의 중지 메모리 액세스 동작을 검출한다. 검출 동작은 중지 메모리 액세스 동작들과 중지하지 않은(non-suspending) 메모리 액세스 동작들 사이를 구별한다. 메모리 제어기는, 메모리 어레이에서의 제1 메모리 액세스 동작의 실행을 중지하고, 제1 우선순위보다 더 높은 우선순위들을 갖고, 제1 우선순위보다 더 높은 우선순위들을 갖는 어떠한 다른 메모리 액세스 동작들도 메모리 어레이에서 실행하기 위해 이용가능하지 않을 때까지 메모리 어레이에서 실행하기 위해 이용가능한 하나 이상의 메모리 액세스 동작들을 실행한다. 메모리 제어기는, 실행 동작의 완료에 응답하여, 메모리 어레이에서의 제1 메모리 액세스 동작의 실행을 재개한다.
- [0003] 이러한 발명의 내용은 발명을 실시하기 위한 구체적인 내용에서 하기에 추가로 기술되는 간소화된 형태로 개념들의 선택을 도입하기 위해 제공된다. 이러한 발명의 내용은 청구되는 발명의 요지의 중요한 특징부들 또는 본질적인 특징부들을 식별하도록 의도되는 것도 아니고, 청구되는 발명의 요지의 범주를 제한하기 위해 사용되도록 의도되는 것도 아니다. 이들 및 다양한 다른 특징들 및 이점들은 하기의 발명을 실시하기 위한 구체적인 내용을 읽음으로써 명백해질 것이다.
- [0004] 다른 구현예들이 또한 본 명세서에 기술되고 인용된다.

도면의 간단한 설명

- [0005] 도 1은 메모리 어레이에 대한 스케줄링된 메모리 액세스 동작 중지/재개 제어를 제공하기 위한 예시적인 메모리 제어기를 예시한다.
- 도 2는 데이터를 적어도 3 차원으로 저장하기에 적합한 메모리 어레이를 포함하는 메모리 디바이스의 일례를 예시한다.
- 도 3은 메모리 액세스 동작들의 중지들 및 재개들을 스케줄링하고 구현하기 위한 예시적인 동작들을 예시한다.
- 도 4는 메모리 액세스 동작 중지 및 재개를 관리하기 위한 예시적인 동작들을 예시한다.

도 5는 본 명세서에 기술된 기술을 구현하는 데 유용할 수 있는 예시적인 프로세싱 시스템을 예시한다.

발명을 실시하기 위한 구체적인 내용

- [0006] 도 1은 메모리 어레이(100)(데이터 어레이로도 지칭됨)에 대한 스케줄링된 메모리 액세스 동작 중지/재개 제어를 제공하기 위한 예시적인 메모리 제어기(126)를 예시한다. 메모리 어레이(100)는 상이한 ISPP(incremental step pulse program) 전압들을 통해 프로그래밍가능한 메모리 셀들(예컨대, 메모리 셀(102))을 포함한다. 메모리 어레이의 메모리 셀들은, 일반적으로 NAND 스트링들로 지칭되는, 직렬로 접속된 복수의 플로팅 게이트 트랜지스터 메모리 셀들을 포함한다. 이러한 유형의 메모리 어레이(100)는 NAND 플래시 메모리로서 알려져 있다. NAND 플래시는 단지 개시된 기술과 함께 사용하기에 적합한 하나의 예시적인 유형의 메모리임을 이해해야 한다.
- [0007] 각각의 NAND 스트링은 비트 라인(예컨대, 심볼이 출력 버스(예컨대, 버스(108))를 통해 판독될 수 있는 비트 라인(106))에 커플링된다. 인접한 NAND 스트링들의 셀들은, 메모리 셀들의 어느 행이 판독 및 프로그래밍 동작들에 의해 영향을 받는지를 선택하는 워드 라인들(예컨대, 워드 라인(110))을 통해 커플링된다. 선택 게이트들(112, 114)은 또한, 판독 및 프로그램 동작들 동안 NAND 스트링들을 각자의 비트 라인들 및 접지에 선택가능하게 커플링시킨다. 일반적으로, 각각의 워드 라인은, 가장 작은 물리적으로 주소지정 가능한 데이터 유닛인 하나 이상의 페이지들(예컨대, 페이지(들)(118))과 연관될 수 있다. (사용자 데이터 및 에러 정정 코드들(또는 ECC) 양쪽 모두를 포함할 수 있는) 비트들에서의 페이지의 크기는 워드 라인에서의 NAND 스트링들의 수에 대응한다. MLC 메모리 어레이는, 대응하는 비트들(예컨대, 각각의 페이지로부터의 모든 비트 0들)을 각각의 셀 내의 고유 전하 레벨로 인코딩함으로써 워드 라인 상에 다수의 페이지들을 저장한다.
- [0008] 플래시 메모리는 플로팅 게이트를 특정 전압으로 충전함으로써 상태를 유지한다. 이것은, 특정된 판독 전압이 인가될 때 채널을 통해 흐르는 미리정의된 양의 전류를 생성한다. 메모리 셀(102)의 확장된 뷰(101)는 플래시 메모리 셀 트랜지스터를 예시한다. 플래시 메모리 셀 트랜지스터는, 기판(136)의 표면에 위치한 소스 영역(134) 및 드레인 영역(132)을 포함한다. 게이트 구조물이 소스 영역(134)과 드레인 영역(132) 사이에 한정된 채널 영역 위에 정렬된다. 게이트 구조물은 플로팅 게이트(130) 및 제어 게이트(140)를 포함한다. 도시되지 않지만, 터널링 유전체 층이 플로팅 게이트(130)와 기판의 표면 사이에 개재되고, 다른 얇은 산화물 층 또는 제어 유전체가 플로팅 게이트(130)와 제어 게이트(140) 사이에 개재된다. 예시된 예에서, 드레인 전압(V_d)은 비트 라인(BL)으로부터 공급되고, 제어 게이트 전압(V_{cg})은 워드 라인(WL)으로부터 공급되고, 소스 전압(V_s)은 접지와 같은 기준 전위에 접속된다.
- [0009] 메모리 셀 트랜지스터를 초기 상태로부터 프로그래밍된 상태로 변경(프로그래밍)하기 위해, 파울러 노드하임(Fowler-Nordheim, FN) 터널링으로 알려진 프로세스가 이용된다. 간략하게, 제어 게이트(140)와 기판(136) 사이에 비교적 큰 양의 전위차가 생성되고, 기판(136)의 표면 상의 채널 내의 여기된 전자들이 플로팅 게이트(130)를 통과하게 하고 그 내에 트랩된다. 이러한 음으로 하전된 전자들은 제어 게이트(140)와 기판(136) 상의 채널 사이의 장벽으로서 작용하여, 메모리 셀 트랜지스터의 임계 전압을 증가시킨다. 메모리 셀 트랜지스터는, 제어 게이트(140)와 기판 사이에 큰 음의 전위차를 형성함으로써 그의 초기 상태로 되돌아갈 수 있는데, 이는 플로팅 게이트(130)와 기판(136) 사이의 얇은 산화물 층을 가로질러 전자들을 다시 끌어당기고, 이에 따라 전자 장벽을 제거하고 메모리 셀(102)의 임계 전압을 감소시킨다.
- [0010] SLC(single level cell) 저장 디바이스에서, 각각의 메모리 셀 트랜지스터는 2개의 가능한 메모리 상태들('0' 또는 '1') 중 하나에 대응하는 전압을 저장한다. 예를 들어, 메모리 상태 '0'은 전압들의 제1 범위에 대응할 수 있고, 메모리 상태 '1'은 전압들의 제2 범위에 대응할 수 있다.
- [0011] MLC(multi-level cell) 저장 디바이스에서, 각각의 메모리 셀 트랜지스터는 2 비트 이상의 데이터를 저장하고, 메모리 셀 트랜지스터는, 각각이 1들과 0들의 가능한 조합들 중 상이한 하나의 조합을 나타내는 일정 범위의 가능한 전압들 중 임의의 것을 취할 수 있다. 예를 들어, 각각의 메모리 셀 트랜지스터가 2 비트의 데이터를 저장하는 경우, 메모리 셀 트랜지스터는 상태들 00, 01, 10, 및 11에 각각 대응하는 4개의 상이한 전압 범위들 중 하나의 전압 범위 내에 속하는 전압을 가질 수 있다. 또한, TLC(triple-level cell) 저장 디바이스 내의 메모리 셀 트랜지스터는 셀당 3 비트를 저장하고, QLC(quad-level cell) 저장 디바이스 내의 메모리 셀 트랜지스터는 셀당 4 비트를 저장한다.
- [0012] ISPP는 MLC 및 SLC 트랜지스터들에서 메모리 상태들을 프로그래밍하기 위한 하나의 접근법이다. 프로그램 메모리 액세스 동작은 메모리에 대한 기록으로서 작용한다. ISPP에서, 프로그램 전압은 증가하는 크기의 짧은 연속

적인 프로그램 펄스들을 사용하여 선택된 메모리 셀의 제어 게이트에 인가된다. 예를 들어, 페이지가 프로그래밍 동작을 위해 선택되는 경우, 페이지의 메모리 셀들이 접속되는 워드 라인들에 바이어스 전압이 인가된다. ISPP 프로그래밍 방법은, 스텝 전압 단위로 워드 라인 바이어스 전압을 점진적으로 증가시키면서 선택된 페이지를 여러 번 프로그래밍한다.

[0013] 증가된 "스텝" 전압의 각각의 펄스 사이에서, 선택된 메모리 셀의 임계 전압이 프로그램 검증 전압 레벨을 초과하여 상승했는지 여부를 확인하기 위해 프로그램 검증 동작이 수행된다. ISPP 프로세스는, 선택된 메모리 셀들(예컨대, 페이지(104)의 셀들) 각각의 임계 전압이 프로그램 검증 레벨을 초과하여 상승할 때까지 계속된다.

[0014] 선택된 메모리 셀의 임계 전압이 프로그램 검증 전압 레벨을 초과할 것을 요구함으로써, ISPP 프로세스는, 선택된 메모리 셀의 각각의 가능한 프로그래밍된 상태에 대한 임계 전압 분포에 대한 하한(lower bound)을 효과적으로 제어한다. 프로그램 전압을 짧은 펄스들로 제어 게이트에 인가함으로써, ISPP 프로세스는 선택된 메모리 셀의 임계 전압이 프로그램 검증 전압 레벨 이상으로 상당히 증가하는 것을 방지하여, 이에 의해 메모리 셀의 각각의 프로그래밍된 상태에 대한 임계 전압 분포에 대한 적당한 상한을 유지한다.

[0015] 프로그램 동작들은, 특히 판독 동작들과 비교하여, 매우 길어질 수 있다. 따라서, 기술된 기술에서의 메모리 제어기(126)는, 다양한 유형의 판독 동작들(또는 다른 더 높은 우선순위 메모리 액세스 동작들)에 우호적으로 프로그램 동작(또는 다른 더 낮은 우선순위 메모리 액세스 동작들)의 중지 및 재개를 스케줄링한다. 이러한 방식으로, 더 높은 우선순위 동작들은, 길고 더 낮은 우선순위 동작들에 의해 차단될 가능성이 더 적다.

[0016] 3D NAND에서, 원 샷 프로그래밍 알고리즘이 종종 채용되는데, 여기서 워드 라인에서의 모든 페이지들은 본질적으로 함께 프로그래밍되고 있다. 따라서, 3D NAND에서의 프로그램 시간은 판독 시간과 비교하여 길 수 있고, 특히 평면형 또는 2D NAND보다 더 길 수 있다. 평면형 NAND에서, 단단계 프로그래밍 알고리즘이 전형적으로 채용되는데, 여기서 페이지들은 서로 독립적으로 프로그래밍된다. 3D NAND에서, 프로그래밍 시간은 셀당 저장된 비트들의 수에 따라 증가하는데, 그 이유는 워드 라인당 페이지들이 더 많기 때문이다. 따라서, TLC 및 QLC NAND에 대한 프로그램 시간들은 전형적으로 MLC 2 비트/셀 NAND의 경우보다 더 높다. 양호한 성능 및 서비스 품질 특성들을 달성하기 위해, 프로그램 중지 동작들이 후술되는 바와 같이 선점 판독 동작들을 허용한다.

[0017] 도 2는 호스트(230) 및 솔리드 스테이트 저장 디바이스(202)를 포함하는 예시적인 시스템(200)을 예시한다. 솔리드 스테이트 저장 디바이스(202)는, 데이터를 적어도 3 차원으로 저장하기에 적합한 하나 이상의 메모리 어레이들(예컨대, 데이터 어레이로도 지칭되는 메모리 어레이(204))을 포함한다. 이러한 유형의 메모리 블록은, 워드 라인들(예컨대, 워드 라인(210))을 통해 커플링된 인접한 NAND 스트링들의 셀들을 포함하는 3D NAND 블록을 포함하고, 또한 NAND 스트링들을 각자의 비트 라인들(예컨대, 비트 라인(211))에 선택가능하게 커플링시키는 선택 게이트들을 포함한다. 채널 스트링(212)으로 지칭되는, 메모리 셀들의 제3 차원은, 대응하는 비트 라인 및 대응하는 채널 선택 라인을 활성화시킴으로써 선택된다. 3D NAND 플래시에서, 워드 라인 방향의 집합은 또한 때때로 워드 라인 평면으로 지칭된다. 후술되는 스캐블링 방법론은 비트 라인(BL), 워드 라인(WL), 및 채널 선택 라인(CSL) 방향들 각각에서 3차원 데이터 랜덤화를 가져온다.

[0018] 호스트(230)는, 메모리 어레이(204)에 데이터를 판독/기록하기 위해 메모리 제어기(206)에, 판독, 기록, 및 소거 동작들과 같은 메모리 액세스 동작들에 대한 명령어들을 발행할 수 있다. 게다가, 메모리 제어기(206)는 또한, 호스트(230)로부터의 명시적 명령어들 없이 메모리 액세스 동작들을 실행할 수 있다. 예를 들어, 메모리 제어기(206)는 재시도 판독 동작들, 재순환 판독 동작들, 재맵핑 판독 동작들, 및 재맵핑 프로그램 동작들을 발행할 수 있다.

[0019] 각각의 유형의 동작, 및 일부 구현예들에서, 개별 메모리 액세스 동작들 자체는, 호스트에 의해서든 또는 메모리 제어기 구성에 의해서든 우선순위 레벨로 지정될 수 있다. 예를 들어, 메모리 제어기 구성은 소정의 판독 동작들을, 소정의 프로그램 동작들보다 더 높은 우선순위들을 갖는 것으로 지정할 수 있다. 이러한 방식으로, 메모리 제어기(206)는, 하나 이상의 판독 동작에 우호적으로 더 낮은 우선순위 프로그램 동작의 중지 및 재개를 스케줄링할 수 있다. 적어도 하나의 구현예에서, 임의의 메모리 액세스 구간에 도달하는(예컨대, 펌웨어에 의해) 높은 우선순위로 표시되고 메모리 어레이에 대한 임의의 호스트 판독 커맨드는, 그 메모리 어레이가 더 낮은 우선순위의 다른 메모리 액세스 동작으로 사용 중인 경우, 현재 메모리 액세스 구간이 완료된 후에 그 메모리 어레이에 대한 중지 커맨드를 트리거할 것이다.

[0020] 일 구현예에서, 메모리 제어기(206)는, 프론트 엔드 인터페이스 회로(220), 가상화 회로(222), 및 미디어 엔진 회로(224)를 포함하는, 회로부에서(예컨대, ASIC로서) 구현된다. 프론트 엔드 인터페이스 회로(220)는 호스트

(230)와 호스트 메모리 액세스 동작 명령어들 및 데이터를 통신한다. 가상화 회로(222)는 솔리드 스테이트 저장 디바이스의 많은 메모리 어레이들에서 가상 메모리 주소들과 물리적 메모리 주소들 사이를 변환한다. 미디어 엔진 회로(224)는 각각의 채널에 대한 개별 플래시 변환 인터페이스(FTI) 회로들 및 플래시 미디어 엔진(FME) 회로를 포함하며, 이는 솔리드 스테이트 저장 디바이스 내의 개별 메모리 어레이들에 대해 다양한 메모리 액세스 동작들에 대한 제어 및 데이터 통신을 제공한다. 예시적인 메모리 액세스 동작들은 판독, 프로그램, 및 소거 동작들을 제한 없이 포함할 수 있다.

[0021] 다양한 구현예들에서, 미디어 엔진 회로(224)는 본 명세서에 기술된 프로세스들에 따라 중지 및 재개 액션들을 스케줄링한다. 미디어 엔진 회로(224)는, 프로세스들에서 사용되는 타이밍들 및 카운트들을 조정하기 위해 디바이스 구성 메모리(226)에 저장된 디바이스 구성 파라미터들(예컨대, 그러한 메모리(226)에 저장된 디바이스 구성 파일들)에 액세스한다. 일부 구현예들에서, 중지/재개 기능을 제어하는 펌웨어는, 펌웨어를 재구축 및 재설치하지 않고서 중지/재개 기능의 동작에 대한 변경들을 허용하기 위해 그러한 디바이스 구성 파라미터들에 액세스할 수 있다. 예시적인 파라미터들은, 미리지정된 시간 동안 또는 미리지정된 수의 메모리 액세스 동작들에 대해 허용되는 최대 수의 중지들, 시한 중지 확장 윈도우(timed suspend extension window)의 크기, 메모리 액세스 동작 구간의 크기, 및 메모리 액세스 동작들의 유형들 또는 상이한 메모리 액세스 동작들의 상대적 우선순위를 제한 없이 포함할 수 있다.

[0022] 일부 구현예들에서, 중지 및 재개가 네스트될(nested) 수 있다는 것을 이해해야 한다. 예를 들어, 제1 메모리 액세스 동작이 중지되어서 제2(더 높은 우선순위) 메모리 액세스 동작이 실행되게 한다. 후속하여, 제1 메모리 액세스 동작이 중지되는 동안, 제2(더 높은 우선순위) 메모리 액세스 동작이 중지되어서 제3(훨씬 더 높은 우선순위) 메모리 액세스 동작이 실행되게 한다. 네스트된 중지 레벨들의 수는 임의의 특정 수로 제한될 필요는 없으며, 각각의 레벨의 중지는, 중지된 동작을 재개하기 전에 다수의 더 높은 우선순위 메모리 액세스 동작들이 완료되게 할 수 있다. 중지된 동작들을 재개하는 것은 중지하는 것과 비교하여 역순으로 수행된다. 중지/재개 레벨들의 수, 중지 시간, 중지들의 수 등에 대한 다른 제한들이 또한 채용될 수 있다. 일례에서, 제1 메모리 액세스 동작은 소거 동작이고, 제2 메모리 액세스 동작은 (소거 동작의 우선순위보다 더 높은 우선순위를 갖는) 프로그램 동작이고, 제3 메모리 액세스 동작은 (소거 및 판독 동작들의 우선순위들보다 더 높은 우선순위를 갖는) 판독 동작이다.

[0023] 도 3은 메모리 액세스 동작들의 중지들 및 재개들을 스케줄링하고 구현하기 위한 예시적인 동작들(300)을 예시한다. 개시 동작(302)은 현재 메모리 어레이에서 LUN(i)(Logical Unit Number)에 대한 메모리 액세스 동작을 개시한다. 메모리 액세스 동작은 우선순위에 기인하고 구간들의 시퀀스로 나뉘지는데, 이들은 각각의 구간의 완료 시에 메모리 액세스 동작의 안전한 중지를 허용하도록 설정되어 있다. 일부 구현예들에서, 구간들은 동일한 크기로 되어 있지만, 다른 구현예들은 상이한 크기의 구간들을 채용할 수 있다. 또한, 일부 구현예들에서, 구간들은, 구간의 시작과 종료를 설정하는 타이머에 기초하여 시간이 정해진다. 지연 동작(304)은, 개시 동작(302) 후에 현재 실행중인 메모리 액세스 동작 구간이 완료될 때까지 기다린다. 지연 동작(304)은, 메모리 액세스 동작이 그의 실행에서 안전하지 않은 지점에서 중지되지 않는 것을 보장한다. 동일한 크기의 구간들은 또한 판독 시간 레이턴시의 예측가능성을 제공하며, 이는 서비스 품질 성능에 유익하다.

[0024] 결정 동작(306)은, 적어도 하나의 더 높은 우선순위 메모리 액세스 동작이 현재 메모리 어레이에서 실행하기 위해 이용가능한지 여부를 결정한다. 일 구현예에서, 결정 동작(306)은, 현재 메모리 어레이에 대한 다양한 판독, 프로그램, 및 소거 요청들이 추가되는 큐(queue)를 평가할 수 있다. 일례에서, 개시 동작(302)은 제1 우선순위를 갖는 프로그램 동작을 실행할 수 있고, 결정 동작(306)은, 하나 이상의 판독 동작들이 큐에서 수신되었고 제1 우선순위보다 더 높은 우선순위를 갖는다고 결정할 수 있다. 어떠한 더 높은 우선순위 메모리 액세스 동작들도 현재 메모리 어레이에서 실행하기 위해 이용가능하지 않은 경우, 프로세싱은 메모리 액세스 동작의 실행을 계속하기 위해 다시 지연 동작(304)으로 되돌아간다. 다양한 메모리 액세스 동작들이, 예컨대, 더 높은 우선순위 메모리 액세스 동작들을 큐의 전방으로 밀어내면서 더 낮은 우선순위 메모리 액세스 동작들을 큐에서 다시 더 멀리 유지시키도록, 하나 이상의 큐들에서 재순서화될 수 있다는 것을 이해해야 한다. 이러한 방식으로, 더 높은 우선순위 메모리 액세스 동작들은 더 낮은 우선순위들을 갖는 메모리 액세스 동작들보다 빨리 이용가능하게 될 수 있다.

[0025] 하나 이상의 더 높은 우선순위 메모리 액세스 동작들이 현재 메모리 어레이에서 실행하기 위해 이용가능한 경우, 다른 결정 동작(308)은, 동일한 메모리 어레이에서 일정 기간에 걸쳐, 동일한 메모리 액세스 동작에 대한 너무 많은 중지들과 같은 너무 많은 중지들이 이 메모리 어레이에서 발행되었는지 여부 등을 결정한다. 만약 그렇다면, 프로세싱은 메모리 액세스 동작의 실행을 계속하기 위해 다시 지연 동작(304)으로 되돌아간다. 그렇

지 않으면, 중지 동작(310)은 LUN(i)에 대한 현재 실행중인 메모리 액세스 동작을 중지하고, 실행 동작(312)은 하나 이상의 더 높은 우선순위의 이용가능한 메모리 액세스 동작들을 실행한다. 결정 동작(314)은, 하나 이상의 더 높은 우선순위 메모리 액세스 동작들이 메모리 어레이에서 실행하기 위해 이용가능한지 여부를 결정한다. 만약 그렇다면, 프로세싱은 실행 동작(312)으로 되돌아간다.

[0026] 어떠한 더 높은 우선순위 메모리 액세스 동작들도 메모리 어레이에서 실행하기 위해 이용가능하지 않을 때, 다른 지연 동작(316)은 시한 중지 확장 윈도우가 만료되기를 기다린다. 다른 결정 동작(318)은, 하나 이상의 더 높은 우선순위 메모리 액세스 동작들이 메모리 어레이에서 실행하기 위해 이용가능한지 여부를 결정한다. 그러한 동작들은 "새롭게 이용가능한" 것으로 간주되는데, 그 이유는 그들이 시한 중지 확장 윈도우 동안 이용가능하게 되기 때문이다. 하나 이상의 새롭게 이용가능한 메모리 액세스 동작들이 현재 메모리 어레이에서 실행하기 위해 이용가능한 경우, 실행 동작(320)은 그러한 새롭게 이용가능한 메모리 액세스 동작들을 실행하고, 프로세싱은 지연 동작(316)으로 돌아온다. 그렇지 않으면, 재개 동작(322)은 더 낮은 우선순위 메모리 동작의 구간들의 시퀀스 중 다음 구간에서 더 낮은 우선순위 메모리 동작의 실행을 재개한다.

[0027] 동작 우선순위들의 적용은 설계 목적들에 기초하여 다를 수 있다. 예를 들어, 일부 구현예들에서, 임의의 관독 동작은 프로그램 또는 소거 동작보다 더 높은 우선순위를 가질 것이다. 다른 구현예들에서, 관독 재시도 동작들은 중지된 메모리 액세스 동작의 재개를 차단하지 않지만, 그 이외에는, 그들은 임의의 다른 관독 동작과 같이 처리된다. 많은 다른 변형예들이 고려된다. 메모리 액세스 동작들의 상대적 우선순위들은 호스트 및 저장 디바이스로부터의 디바이스 구성 파라미터들 및 명령어들에 따라 달라질 수 있다.

[0028] 다음은 상이한 우선순위들을 갖는 메모리 액세스 동작들이 어떻게 처리될 수 있는지의 일부 특정 예들을 제공한다. 다른 변형예들이 채용될 수 있다:

- [0029] • 큐에 존재하는 경우 프로그램 동작을 중지시키는 중지 메모리 액세스 동작들
- [0030] o 호스트 관독 동작들
- [0031] o 맵 관독 동작들(맵 엔트리가 캐싱되지 않은 경우), 여기서 "맵"은 로직 주소와 물리적 주소 사이의 맵핑을 저장하는 메타데이터를 지칭한다.
- [0032] o 전력 이벤트 또는 전력 실패로부터의 구동을 복구하는 것을 허용하는 체크포인트 관독 동작들(예컨대, 메타데이터의 관독 동작들)
- [0033] • 중지 동안 실행되는 중지하지 않은 메모리 액세스 동작들 - 그들이 중지 동안 큐에 도달하거나 그 내에 있는 경우(그러나 중지를 트리거하지 않을 것임)
- [0034] o 재순환 관독 동작들
- [0035] o 모든 에러 복구 및 플래시 정책 관독 동작들(예컨대, 백그라운드 스크립 동안의 관독 동작들 및 코드 레이트 테스트 동작들)
- [0036] o 관독 기준 전압들, 에러 정정 코드, 및 코드 레이트 설정들과 같은 SSD의 파라미터들을 교정 및 튜닝하여 제1 관독 효과들을 다루기 위한 백그라운드 관독 동작들 등.
- [0037] • 중지를 트리거하거나 중지 동안 실행하도록 허용되지 않고 중지되는 경우 프로그램 동작이 재개되게 하는 재개-트리거링(낮은 우선순위) 관독 동작들
- [0038] o 루트 파일 시스템 관독 동작들

[0039] 도 4는 메모리 액세스 동작 중지 및 재개를 관리하기 위한 예시적인 동작들(400)을 예시한다. 실행 동작(402)은 메모리 어레이에서 제1 메모리 액세스 동작을 실행한다. 제1 메모리 액세스 동작은 제1 우선순위를 갖는다. 검출 동작(404)은, 메모리 어레이에서 실행하기 위해 이용가능하고 제1 우선순위보다 더 높은 우선순위를 갖는 적어도 하나의 중지 메모리 액세스 동작을 검출한다. 검출 동작(404)은 또한, 중지 메모리 액세스 동작들과 중지하지 않은 메모리 액세스 동작들(그리고 일부 구현예들에서 재개-트리거링 메모리 액세스 동작들) 사이를 구별한다. 중지 동작(406)은, 적어도 하나의 더 높은 우선순위 중지 메모리 액세스 동작들이 메모리 어레이에서 실행하기 위해 이용가능하다고 결정되는 경우, 현재 실행중인 메모리 액세스 동작의 실행을 중지한다. 하나 이상의 중지하지 않은 메모리 액세스 동작들이, 이용가능하고 제1 우선순위보다 더 높은 우선순위를 갖는 것으로 검출되지만, 어떠한 중지 메모리 액세스 동작들도, 이용가능하고 제1 우선순위보다 더 높은 우선순위들을 갖는

것으로 검출되지 않는 경우, 중지 동작(406)은 실행되지 않는다. 일 구현예에서, 중지 동작(406)은, 중지를 개시하기 전에 메모리 액세스 동작의 현재 실행중인 구간이 완료될 때까지 기다린다.

[0040] 실행 동작(408)은, 중지 동작(406)이 완료된 후에 이용가능하게 되는 새로운 더 높은 우선순위 메모리 액세스 동작들을 포함하는, 이용가능한 더 높은 우선순위 메모리 액세스 동작들을 실행한다. 실행 동작(408)은, 어떠한 추가적인 더 높은 우선순위 메모리 액세스 동작들도 메모리 어레이에서 실행하기 위해 이용가능하지 않을 때 완료된다. 다른 실행 동작(410)은, 시한 중지 확장 윈도우의 만료 후에, 시한 중지 확장 윈도우들 동안 이용가능하게 된 임의의 새롭게 이용가능한 더 높은 우선순위 메모리 액세스 동작들을 실행한다. 실행 동작(410)은 또한, 시한 중지 확장 윈도우들의 종료 시에 어떠한 더 높은 우선순위 메모리 액세스 동작들도 이용가능하지 않을 때까지 다수의 시한 중지 확장 윈도우들을 통해 반복될 수 있다. 재개 동작(412)은 더 낮은 우선순위 메모리 액세스 동작의 실행을 재개한다. 일부 구현예들에서, 실행 동작(410)은 생략될 수 있다.

[0041] 일부 구현예들에서, 실행 동작(408) 또는 실행 동작(410) 중 어느 하나의 실행 동작의 완료 전에, 하나 이상의 재개 트리거 조건들은 중지된 메모리 액세스 동작을 재개하도록 설정될 수 있다. 예를 들어, 재개 트리거 조건은 타이머의 만료로 설정될 수 있어서, 메모리 제어기가 어떠한 추가적인 더 높은 우선순위 메모리 액세스 동작들도 메모리 어레이에서 실행하기 위해 이용가능하지 않음을 검출할 때보다는 오히려, 메모리 제어기가 타이머의 만료를 검출할 때 실행 동작(408)이 완료되게 한다. 다른 예에서, 재개 트리거 조건은 최대 더 높은 우선순위 메모리 액세스 동작 카운트로 설정될 수 있어서, 메모리 제어기가 어떠한 추가적인 더 높은 우선순위 메모리 액세스 동작들도 메모리 어레이에서 실행하기 위해 이용가능하지 않음을 검출할 때보다는 오히려, 메모리 제어기가 중지 동안 실행되는 더 높은 우선순위 메모리 액세스 동작들의 수가 최대 더 높은 우선순위 메모리 액세스 동작 카운트를 충족하거나 초과함을 검출할 때 실행 동작(408)이 완료되게 한다. 또 다른 예에서, 재개 트리거 조건은, (예컨대, 일정 기간에 걸친) 실행된 더 높은 우선순위 메모리 액세스 동작들 대 (예컨대, 일정 기간에 걸친) 더 낮은 우선순위 메모리 액세스 동작들의 원하는 비로 설정될 수 있어서, 메모리 제어기가 어떠한 추가적인 더 높은 우선순위 메모리 액세스 동작들도 메모리 어레이에서 실행하기 위해 이용가능하지 않음을 검출할 때보다는 오히려, 메모리 제어기가, 실행된 더 높은 우선순위 메모리 액세스 동작들 대 실행된 더 낮은 우선순위 메모리 액세스 동작들의 비가 미리결정된 최대 임계치를 충족하거나 초과함을 검출할 때 실행 동작(408)이 완료되게 한다.

[0042] 실행 동작(408)의 완료 전에, 재개-트리거링 메모리 액세스 동작이 메모리 어레이에서 실행하기 위해 이용가능한 것으로 검출되는 경우, 심지어 그 우선순위가 중지 동안 실행되는 더 높은 우선순위 메모리 액세스 동작보다 낮은 경우에도, 실행중인 메모리 액세스 동작은 종료되고, 제1 메모리 액세스 동작의 실행은 재개 동작(412)에서 재개된다.

[0043] 그러한 재개-트리거링 조건들은 또한, 실행 동작(410)을 선점하거나 조기에 종료하고 재개 동작(412)을 트리거하기 위해 적용될 수 있다. 다른 선점 재개 조건들이 고려된다.

[0044] 도 5는 기술된 기술을 구현하는 데 유용할 수 있는 예시적인 프로세싱 시스템(500)을 예시한다. 프로세싱 시스템(500)은, 컴퓨터 프로세스를 실행하기 위해 유형적 컴퓨터 판독가능(또는 프로세서 판독가능) 저장 매체에서 구현되는 컴퓨터 프로그램 제품을 실행할 수 있다. 데이터 및 프로그램 파일들은, 하나 이상의 프로세서들(예컨대, CPU들, GPU들, ASIC들)을 사용하여 파일들을 판독하고 그 내의 프로그램들을 실행하는 프로세싱 시스템(500)에 입력될 수 있다. 프로세싱 시스템(500)의 요소들 중 일부가 도 5에 도시되어 있는데, 여기서 프로세서(502)는 입력/출력(I/O) 섹션(504), 하나 이상의 프로세서들(506), 및 메모리(508)를 갖는 것으로 도시되어 있다. 하나 이상의 프로세서들(506)은 하나 이상의 프로세싱 디바이스들 또는 코어들을 포함할 수 있다. 프로세서들은 단일 코어 또는 멀티코어 프로세서들일 수 있다. 프로세싱 시스템(500)은 종래의 컴퓨터, 분산형 컴퓨터, 제어기 보드, ASIC, 또는 임의의 다른 유형의 프로세싱 시스템일 수 있다. 기술된 기술은 선택적으로, 메모리(508), 저장 유닛(512)에 로딩되고/되거나, 반송파 신호(예컨대, 이더넷, 3G 무선, 5G 무선, LTE(Long Term Evolution)) 상에서 유선 또는 무선 네트워크 링크(514)를 통해 통신되는 소프트웨어로 구현되어, 이에 의해 도 5에서의 프로세싱 시스템(500)을 기술된 동작들을 구현하기 위한 특수 목적 기계로 변환한다. 프로세싱 시스템(500)은 본 명세서에 개시된 디스크 드라이브 처리량 밸런싱 시스템을 지원하도록 구성된 애플리케이션 특정 프로세싱 시스템일 수 있다.

[0045] I/O 섹션(504)은 하나 이상의 사용자 인터페이스 디바이스들(예컨대, 키보드, 터치 스크린 디스플레이 유닛(518) 등) 또는 저장 유닛(512)에 접속될 수 있다. 기술된 기술에 따라 시스템들 및 방법들을 달성하기 위한 메커니즘들을 포함하는 컴퓨터 프로그램 제품들이 그러한 시스템(500)의 저장 유닛(512) 상에 또는 메모리(508)

내에 상주할 수 있다.

[0046] 통신 인터페이스(524)는, 컴퓨터 시스템이 반송파에서 구현되는 명령어들과 데이터를 수신할 수 있게 하는 네트워크 링크(514)를 통해 프로세싱 시스템(500)을 기업 네트워크에 접속할 수 있다. 로컬 영역 네트워킹(LAN) 환경에서 사용될 때, 프로세싱 시스템(500)은, 통신 디바이스의 하나의 유형인 통신 인터페이스(524)를 통해 로컬 네트워크에 (유선 접속에 의해 또는 무선으로) 접속된다. 광역 네트워킹(WAN) 환경에서 사용될 때, 프로세싱 시스템(500)은 전형적으로 모뎀, 네트워크 어댑터, 또는 광역 네트워크를 통해 통신을 확립하기 위한 임의의 다른 유형의 통신 디바이스를 포함한다. 네트워크화된 환경에서, 프로세싱 시스템(500) 또는 그의 일부분들에 대해 도시된 프로그램 모듈들은 원격 메모리 저장 디바이스에 저장될 수 있다. 도시된 네트워크 접속들은 통신 디바이스들의 예들이고, 컴퓨터들 사이의 통신 링크를 확립하는 다른 수단이 사용될 수 있다는 것이 이해된다.

[0047] 예시적인 구현예에서, 메모리 제어기 및 다른 모듈들을 제어하는 펌웨어는, 메모리(508) 및/또는 저장 유닛(512)에 저장되고 프로세서(502)에 의해 실행되는 프로세서 실행가능 명령어들에 의해 구현될 수 있다. 또한, 저장 제어기는 RAID 구현예를 지원하는 것을 돕도록 구성될 수 있다. RAID 저장소는, 범용 컴퓨터 및 특화된 소프트웨어(예컨대, 서버 실행 서비스 소프트웨어), 특수 목적 컴퓨팅 시스템 및 특화된 소프트웨어(예컨대, 모바일 디바이스 또는 네트워크 기기 실행 서비스 소프트웨어), 또는 다른 컴퓨팅 구성들을 사용하여 구현될 수 있다. 또한, 구간 크기들, 우선순위들, 판독/프로그램 데이터, 및 다른 데이터 및 파라미터들이 메모리(508) 및/또는 저장 유닛(512)에 저장되고 프로세서(502)에 의해 실행될 수 있다.

[0048] 프로세싱 시스템(500)은, 사용자 디바이스, 저장 디바이스, IoT 디바이스, 데스크톱, 랩톱, 컴퓨팅 디바이스와 같은 디바이스에서 구현될 수 있다. 프로세싱 시스템(500)은, 사용자 디바이스 내에서 실행되는 또는 사용자 디바이스 외부에 있는 저장 디바이스일 수 있다.

[0049] 프로세싱 시스템(500)은 여러가지 유형적 프로세서 판독가능 저장 매체들 및 무형적 프로세서 판독가능 통신 신호들을 포함할 수 있다. 유형적 프로세서 판독가능 저장소는, 프로세싱 시스템(500)에 의해 액세스될 수 있는 임의의 이용가능한 매체들에 의해 구현될 수 있고, 휘발성 및 비휘발성 저장 매체들, 제거가능 및 제거불가능한 저장 매체들 양쪽 모두를 포함한다. 유형적 프로세서 판독가능 저장 매체들은 무형적 통신 신호들을 배제하고, 프로세서 판독가능 명령어들, 데이터 구조들, 프로그램 모듈들 또는 다른 데이터와 같은 정보의 저장을 위한 임의의 방법 또는 기술로 구현되는 휘발성 및 비휘발성, 제거가능 및 제거불가능한 저장 매체들을 포함한다. 유형적 프로세서 판독가능 저장 매체들은, RAM, ROM, EEPROM, 플래시 메모리 또는 다른 메모리 기술, CDROM, 디지털 다기능 디스크(DVD) 또는 다른 광학 디스크 저장소, 자기 카세트들, 자기 테이프, 자기 디스크 저장소 또는 다른 자기 저장 디바이스들, 또는 원하는 정보를 저장하는 데 사용될 수 있고 프로세싱 시스템(500)에 의해 액세스될 수 있는 임의의 다른 유형적 매체를 포함하지만 이로 제한되지 않는다. 유형적 프로세서 판독가능 저장 매체들과는 대조적으로, 무형적 프로세서 판독가능 통신 신호들은 프로세서 판독가능 명령어들, 데이터 구조들, 프로그램 모듈들 또는 변조된 데이터 신호에 상주하는 다른 데이터, 예컨대 반송파 또는 다른 신호 전달 매커니즘을 구현할 수 있다. 용어 "변조된 데이터 신호"는, 신호 내의 정보를 인코딩하는 것과 같은 방식으로 설정되거나 변경된 그의 특성들 중 하나 이상을 갖는 신호를 의미한다. 제한이 아닌 예로서, 무형적 통신 신호들은, 유선 네트워크 또는 직접-유선 접속과 같은 유선 매체들, 및 음향, RF, 적외선, 및 다른 무선 매체들과 같은 무선 매체들을 통해 이동하는 신호들을 포함한다.

[0050] 일부 구현예들은 제조 물품을 포함할 수 있다. 제조 물품은 로직을 저장하기 위한 유형적 저장 매체를 포함할 수 있다. 저장 매체의 예들은, 휘발성 메모리 또는 비휘발성 메모리, 제거가능한 또는 제거불가능한 메모리, 소거가능한 또는 소거불가능한 메모리, 기록가능한 또는 재기록가능한 메모리 등을 포함하는, 전자 데이터를 저장할 수 있는 하나 이상의 유형의 컴퓨터 판독가능 저장 매체들을 포함할 수 있다. 로직의 예들은, 소프트웨어 컴포넌트들, 프로그램들, 애플리케이션들, 컴퓨터 프로그램들, 애플리케이션 프로그램들, 시스템 프로그램들, 기계 프로그램들, 운영 체제 소프트웨어, 미들웨어, 펌웨어, 소프트웨어 모듈들, 루틴들, 서브루틴들, 동작 세그먼트들, 방법들, 절차들, 소프트웨어 인터페이스들, 애플리케이션 프로그램 인터페이스들(API), 명령어 세트들, 컴퓨팅 코드, 컴퓨터 코드, 코드 세그먼트들, 컴퓨터 코드 세그먼트들, 단어들, 값들, 심볼들, 또는 이들의 임의의 조합과 같은 다양한 소프트웨어 요소들을 포함할 수 있다. 일 구현예에서, 예를 들어, 제조 물품은, 컴퓨터에 의해 실행될 때, 컴퓨터로 하여금 기술된 실시예들에 따른 방법들 및/또는 동작들을 수행하게 하는 실행가능한 컴퓨터 프로그램 명령어들을 저장할 수 있다. 실행가능한 컴퓨터 프로그램 명령어들은, 소스 코드, 컴파일된 코드, 해석된 코드, 실행가능한 코드, 정적 코드, 동적 코드 등과 같은 임의의 적합한 유형의 코드를 포함할 수 있다. 실행가능한 컴퓨터 프로그램 명령어들은, 컴퓨터가 소정 동작 세그먼트를 수행하도록 지시하기 위해, 미리정의된 컴퓨터 언어, 방식 또는 신택스에 따라 구현될 수 있다. 명령어들은 임의의 적합한 하이 레

벨, 로우 레벨, 객체 지향, 시각적, 컴파일된 및/또는 해석된 프로그래밍 언어를 사용하여 구현될 수 있다.

[0051]

데이터 저장 및/또는 메모리는 다양한 유형의 프로세서 관독가능 저장 매체들, 예컨대 하드 디스크 매체들, 다수의 저장 디바이스들을 포함하는 저장 어레이, 광학 매체들, 솔리드 스테이트 드라이브 기술, ROM, RAM, 및 다른 기술에 의해 구현될 수 있다. 동작들은, 마이크로프로세서, 마이크로프로세서 코어, 마이크로제어기, 특수 목적 회로부, 또는 다른 프로세싱 기술들에 의해 실행되든지 또는 보조되든지 간에, 펌웨어, 소프트웨어, 하드 와이어드 회로부, 게이트 어레이 기술 및 다른 기술들에서 프로세서 실행가능 명령어들로 구현될 수 있다. 기록 제어기, 저장 제어기, 데이터 기록 회로부, 데이터 관독 및 복구 회로부, 분류 모듈, 및 데이터 저장 시스템의 다른 기능 모듈들이 시스템 구현 프로세스를 수행하기 위한 프로세서 관독가능 명령어들을 프로세싱하기 위한 프로세서를 포함하거나 그와 협력하여 작동할 수 있다는 것을 이해해야 한다.

[0052]

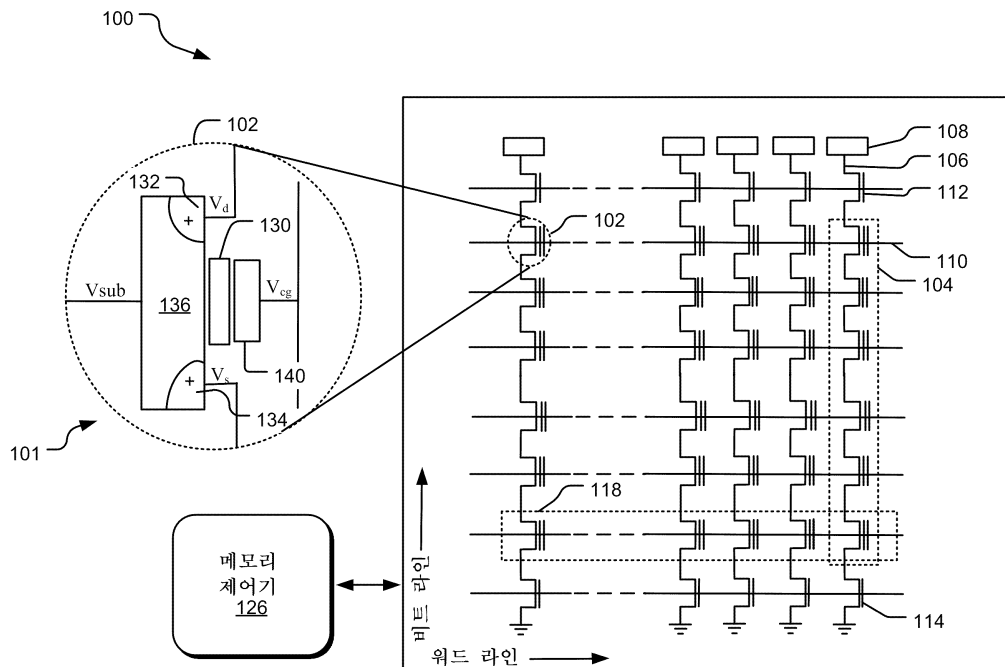
본 명세서에 기술된 개시된 기술의 실시예들은 하나 이상의 컴퓨터 시스템들에서 로직 단계들로서 구현된다. 현재 개시된 기술의 로직 동작들은 (1) 하나 이상의 컴퓨터 시스템들에서 실행되는 프로세서 구현 단계들의 시퀀스로서, 그리고 (2) 하나 이상의 컴퓨터 시스템들 내의 상호접속 기계 또는 회로 모듈들로서 구현된다. 구현예는 개시된 기술을 구현하는 컴퓨터 시스템의 성능 요건들에 따른 선택의 문제이다. 따라서, 본 명세서에 기술된 개시된 기술의 실시예들을 구성하는 로직 동작들은 동작들, 단계들, 객체들, 또는 모듈들로서 다양하게 지칭된다. 더욱이, 달리 명시적으로 주장되지 않거나 특정 순서가 청구항의 언어로 고유하게 필요하지 않다면, 로직 동작들은 임의의 순서로, 원한다면 추가 및 생략하여, 수행될 수 있는 것을 이해해야 한다.

[0053]

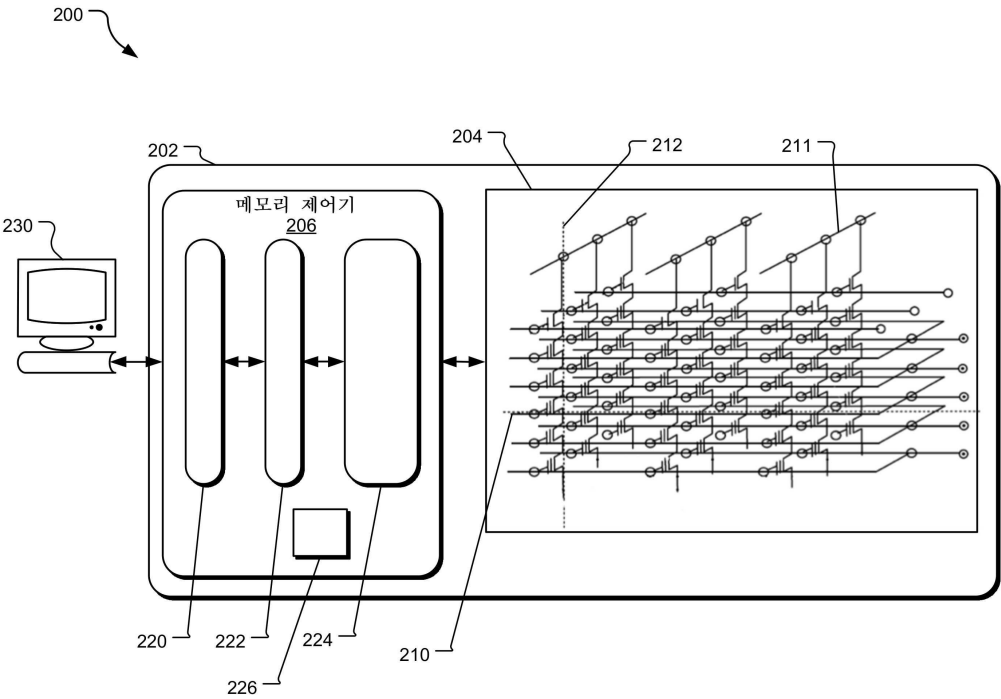
상기 명세서, 예들, 및 데이터는 구조의 완전한 설명 및 개시된 기술의 예시적인 실시예들의 사용을 제공한다. 개시된 기술의 많은 실시예들이 개시된 기술의 사상 및 범주로부터 벗어남이 없이 이루어질 수 있기 때문에, 개시된 기술은 이하 첨부된 청구범위 내에 존재한다. 게다가, 상이한 실시예들의 구조적 특징부들은 인용되는 청구범위로부터 벗어남이 없이 또 다른 실시예에 조합될 수 있다.

도면

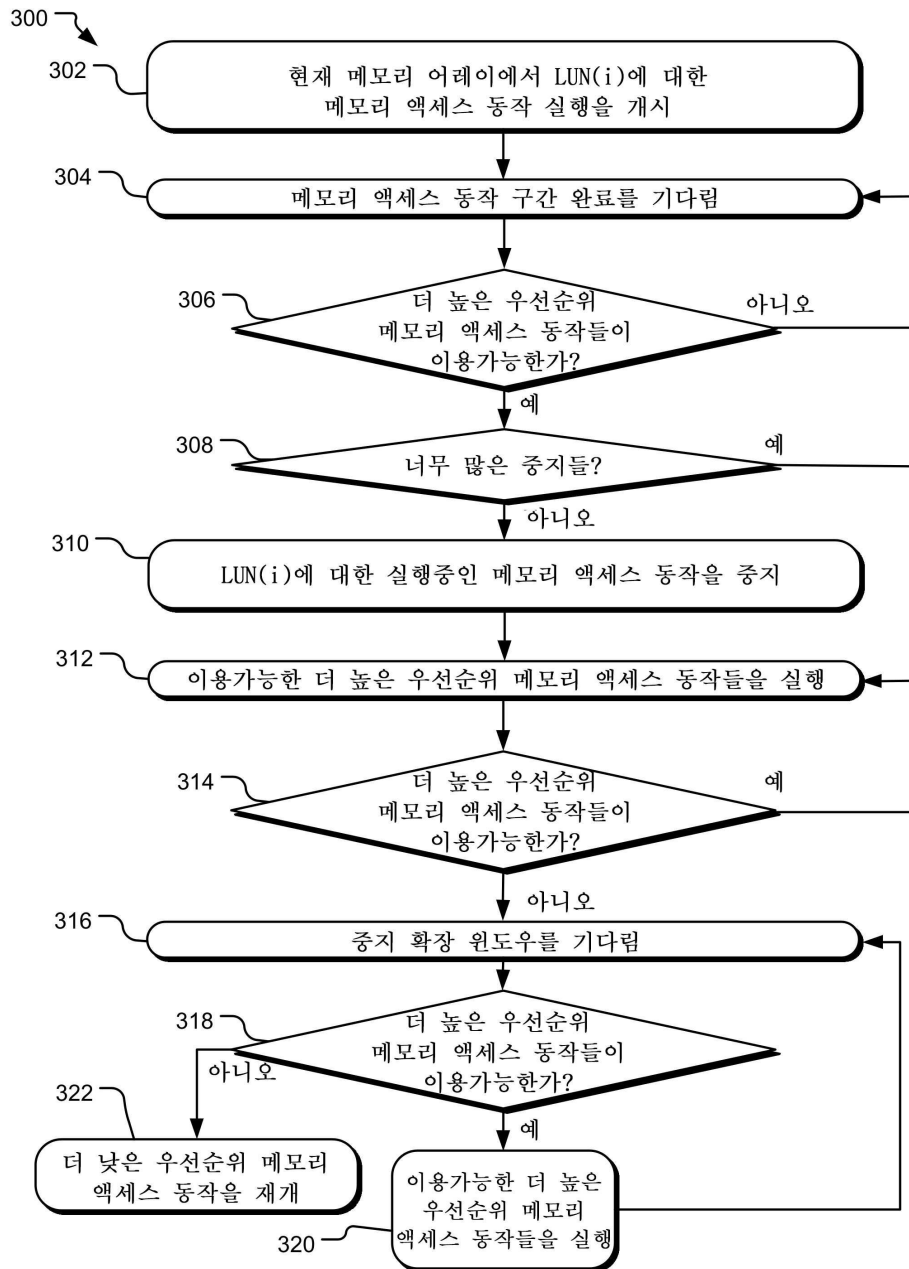
도면1



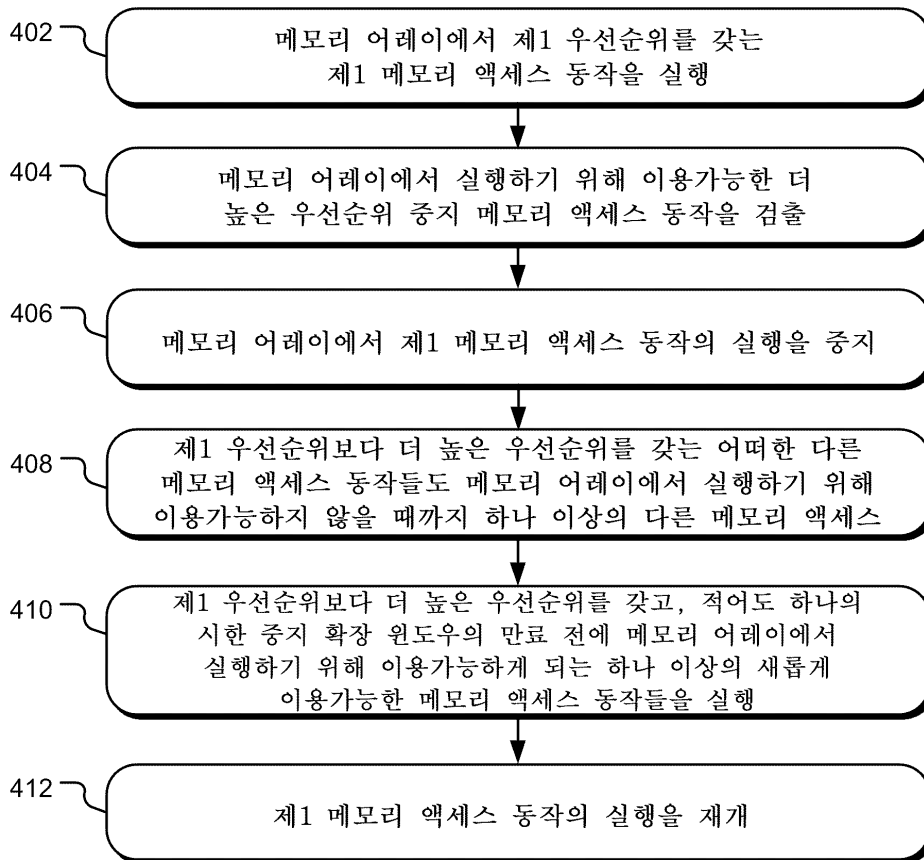
도면2



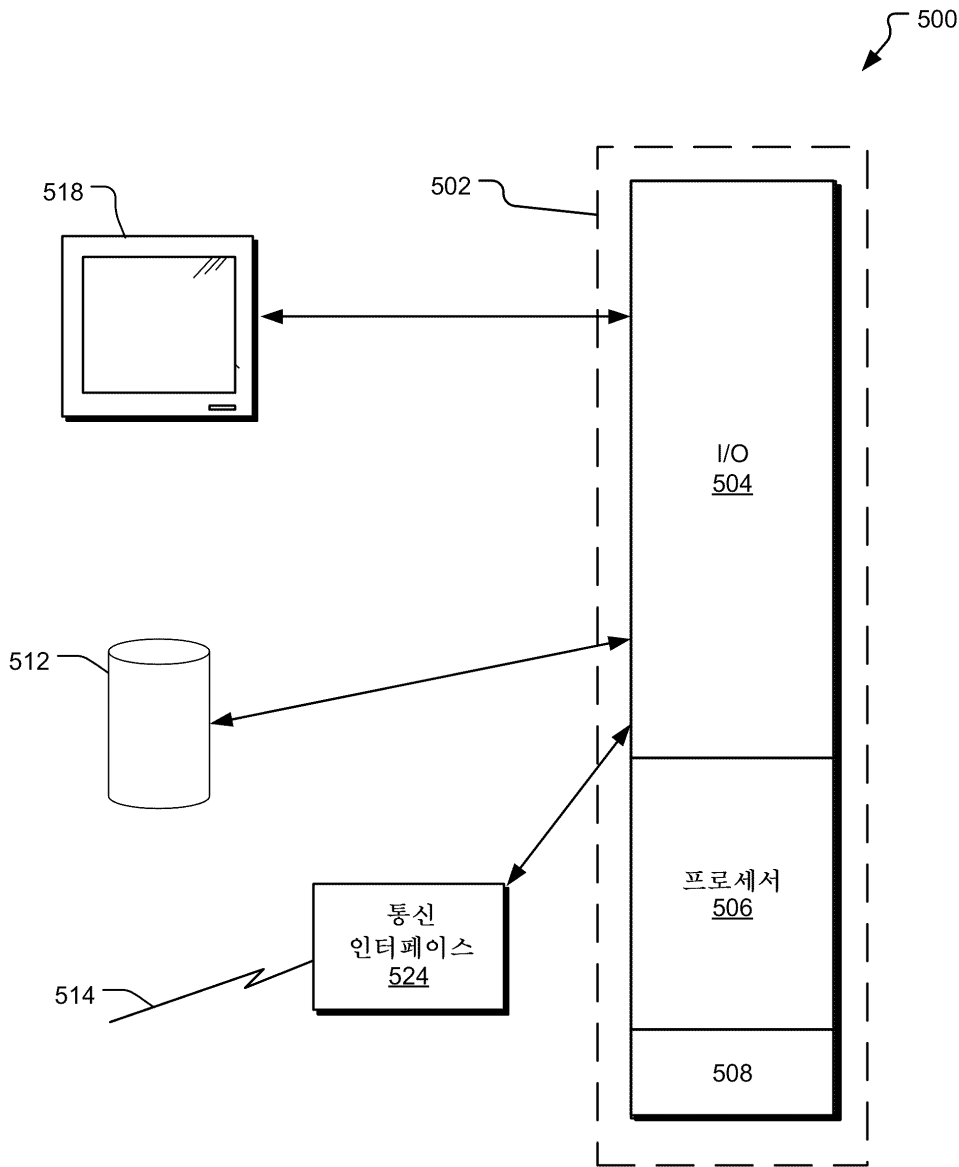
도면3



도면4



도면5



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 청구항 7

【변경전】

제1항에 있어서, 상기 제1 메모리 액세스 동작은 프로그램 동작이고, 상기 프로그램 동작보다 더 높은 우선순위를 갖는 상기 다른 메모리 액세스 동작들 중 적어도 하나는 호스트 판독 동작인, 방법.

【변경후】

제1항에 있어서, 상기 제1 메모리 액세스 동작은 프로그램 동작이고, 상기 프로그램 동작보다 더 높은 우선순위를 갖는 다른 메모리 액세스 동작들 중 적어도 하나는 호스트 판독 동작인, 방법.

【직권보정 2】

【보정항목】 청구범위

【보정세부항목】 청구항 14

【변경전】

제9항에 있어서, 상기 제1 메모리 액세스 동작은 프로그램 동작이고, 상기 프로그램 동작보다 더 높은 우선순위를 갖는 상기 다른 메모리 액세스 동작들 중 적어도 하나는 호스트 판독 동작인, 시스템.

【변경후】

제9항에 있어서, 상기 제1 메모리 액세스 동작은 프로그램 동작이고, 상기 프로그램 동작보다 더 높은 우선순위를 갖는 다른 메모리 액세스 동작들 중 적어도 하나는 호스트 판독 동작인, 시스템.

【직권보정 3】

【보정항목】 청구범위

【보정세부항목】 청구항 19

【변경전】

제17항에 있어서, 상기 제1 메모리 액세스 동작은 프로그램 동작이고, 상기 프로그램 동작보다 더 높은 우선순위를 갖는 상기 다른 메모리 액세스 동작들 중 적어도 하나는 호스트 판독 동작인, 하나 이상의 유형적 프로세서 판독가능 저장 매체.

【변경후】

제17항에 있어서, 상기 제1 메모리 액세스 동작은 프로그램 동작이고, 상기 프로그램 동작보다 더 높은 우선순위를 갖는 다른 메모리 액세스 동작들 중 적어도 하나는 호스트 판독 동작인, 하나 이상의 유형적 프로세서 판독가능 저장 매체.

【직권보정 4】

【보정항목】 청구범위

【보정세부항목】 청구항 20

【변경전】

제17항에 있어서, 상기 제1 메모리 액세스 동작은 프로그램 동작이고, 상기 프로그램 동작보다 더 높은 우선순위를 갖는 상기 다른 메모리 액세스 동작들 중 적어도 하나는 맵 판독 동작인, 하나 이상의 유형적 프로세서 판독가능 저장 매체.

【변경후】

제17항에 있어서, 상기 제1 메모리 액세스 동작은 프로그램 동작이고, 상기 프로그램 동작보다 더 높은 우선순위를 갖는 다른 메모리 액세스 동작들 중 적어도 하나는 맵 판독 동작인, 하나 이상의 유형적 프로세서 판독가능 저장 매체.