



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0119921
(43) 공개일자 2015년10월26일

- (51) 국제특허분류(Int. Cl.)
G06F 12/08 (2006.01) G06F 12/02 (2006.01)
- (52) CPC특허분류
G06F 12/0891 (2013.01)
G06F 12/0238 (2013.01)
- (21) 출원번호 10-2015-7025301
- (22) 출원일자(국제) 2014년02월12일
심사청구일자 없음
- (85) 번역문제출일자 2015년09월15일
- (86) 국제출원번호 PCT/US2014/015994
- (87) 국제공개번호 WO 2014/130317
국제공개일자 2014년08월28일
- (30) 우선권주장
13/772,400 2013년02월21일 미국(US)

- (71) 출원인
퀄컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
- (72) 발명자
동, 시양유
미국 92121 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
- (74) 대리인
특허법인 남앤드남

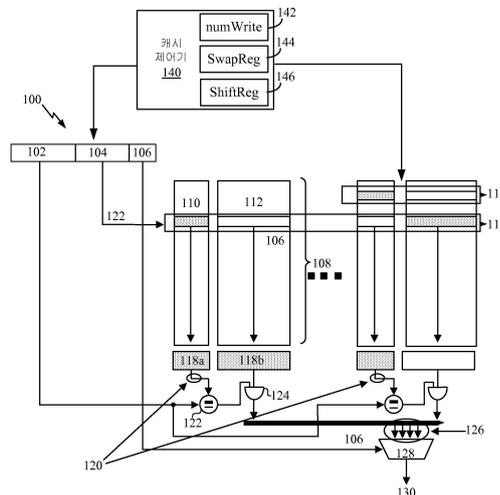
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 제한된 기록 내구성을 갖는 캐시들에 대한 세트-간 웨어-레벨링

(57) 요약

캐시 제어기는, 캐시 메모리 내의 다수의 캐시 세트들에 대한 모든 각각의 메모리 위치 스와핑 연산 이후에 업데이트하고, 모든 각각의 N-1 메모리 위치 스와핑 연산들을 리셋하는 제 1 레지스터를 포함한다. N은 캐시 메모리 내의 캐시 세트들의 수이다. 메모리 제어기는 또한, 모든 각각의 N-1 메모리 위치 스와핑 연산들 이후에 업데이트하고, 모든 각각의 (N²-N) 메모리 위치 스와핑 연산들을 리셋하는 제 2 레지스터를 갖는다. 제 1 및 제 2 레지스터는 캐시 세트들의 논리 위치들과 물리 위치들 사이의 관계를 추적한다.

대표도 - 도1



(52) CPC특허분류

G06F 12/0802 (2013.01)

G06F 12/0864 (2013.01)

G06F 2212/7211 (2013.01)

명세서

청구범위

청구항 1

캐시 메모리의 웨어-레벨링을 위한 방법으로서,

상기 캐시 메모리에의 메모리 기록 연산들의 수가 임계 값에 도달할 때 캐시 세트들에 대해 복수의 메모리 위치 스와핑 연산들을 수행함으로써 상기 캐시 메모리의 캐시 세트들을 동적으로 로테이션시키는 것 - 각각의 스와핑 연산은 단지 스와핑된 캐시 세트들로부터의 컨텐츠들을 클리어하는 반면, 다른 캐시 세트들의 메모리 컨텐츠들을 그대로 남겨두는 것을 포함함 - ; 및

논리 캐시 세트 수를 물리 캐시 세트 수로 변환하기 위해 상기 스와핑된 캐시 세트들을 추적하는 것을 포함하는,

캐시 메모리의 웨어-레벨링을 위한 방법.

청구항 2

제 1 항에 있어서,

상기 캐시 세트들을 동적으로 로테이션시키는 것은,

글로벌 카운터로 상기 캐시 메모리에의 메모리 기록 연산들의 수를 카운트하는 것;

상기 글로벌 카운터가 상기 임계 값에 도달할 때 상기 복수의 메모리 위치 스와핑 연산들을 개시하는 것; 및

상기 글로벌 카운터를 리셋하는 것을 더 포함하는,

캐시 메모리의 웨어-레벨링을 위한 방법.

청구항 3

제 2 항에 있어서,

상기 캐시 세트들을 동적으로 로테이션시키는 것은,

스왑 레지스터(SwapReg)를 $(\text{SwapReg} + 1) \bmod (N-1)$ 로 세팅하는 것 - N은 상기 캐시 메모리 내의 상기 캐시 세트들의 수임 - ; 및

상기 SwapReg가 0일 때, 시프트 레지스터(ShiftReg)를 $(\text{ShiftReg} + 1) \bmod N$ 으로 세팅하는 것을 포함하는,

캐시 메모리의 웨어-레벨링을 위한 방법.

청구항 4

제 1 항에 있어서,

상기 캐시 메모리에의 각각의 메모리 기록 연산에 대해 글로벌 카운터를 증분시킴으로써, 상기 글로벌 카운터로 상기 캐시 메모리에의 메모리 기록 연산들의 수를 카운트하는 것;

스와핑 연산을 수행하고, 상기 글로벌 카운터가 상기 임계 값과 동일할 때 스왑 카운터를 증분시키는 것 - 상기 스와핑 연산을 수행하는 것은 상기 스와핑된 캐시 세트들을 리맵핑하는 것을 포함함 - ;

시프트 카운터를 증분시키고, 스왑 카운터 값이 N-1과 동일할 때, 상기 스왑 카운터를 리셋하는 것 - N은 상기 캐시 메모리 내의 상기 캐시 세트들의 수임 - ; 및

시프트 카운터 값이 N과 동일할 때, 상기 시프트 카운터를 리셋하는 것을 더 포함하는,

캐시 메모리의 웨어-레벨링을 위한 방법.

청구항 5

제 4 항에 있어서,
 상기 스와핑된 캐시 세트들을 리맵핑하는 것은,
 기록 백 버퍼에 상기 스와핑된 캐시 세트들 내의 더티(dirty) 데이터를 저장하는 것; 및
 상기 스와핑된 캐시 세트들 내의 다른 데이터를 무효화하는 것을 포함하는,
 캐시 메모리의 웨어-레벨링을 위한 방법.

청구항 6

제 4 항에 있어서,
 입력 캐시 세트 수를 출력 캐시 세트 수로 변환하는 것은,
 입력 논리 세트 수가 상기 스왑 카운터 값과 동일할 때, 출력 물리 세트 수를 상기 시프트 카운터 값으로 세팅하는 것;
 상기 입력 논리 세트 수가 상기 스왑 카운터 값보다 클 때, 상기 출력 물리 세트 수를 (상기 입력 논리 세트 수 + 상기 시프트 카운터 값) modulo N으로 세팅하는 것; 및
 상기 입력 논리 세트 수가 상기 스왑 카운터 값보다 적을 때, 상기 출력 물리 세트 수를 (상기 입력 논리 세트 수 + 상기 시프트 카운터 값 + 1) modulo N으로 세팅하는 것을 포함하는,
 캐시 메모리의 웨어-레벨링을 위한 방법.

청구항 7

제 4 항에 있어서,
 상기 스와핑된 캐시 세트들을 추적하는 것은,
 입력 물리 세트 수가 상기 시프트 카운터 값과 동일할 때, 출력 논리 세트 수를 상기 스왑 카운터 값으로 세팅하는 것;
 (상기 입력 물리 세트 수 - 상기 시프트 카운터 값) modulo N이 상기 스왑 카운터 값보다 클 때, 상기 출력 논리 세트 수를 (상기 입력 물리 세트 수 - 상기 시프트 카운터 값) modulo N으로 세팅하는 것; 및
 그렇지 않으면, 상기 출력 논리 세트 수를 (상기 입력 물리 세트 수 - 상기 시프트 카운터 값 - 1) modulo N으로 세팅하는 것을 포함하는,
 캐시 메모리의 웨어-레벨링을 위한 방법.

청구항 8

제 1 항에 있어서,
 모바일 폰, 셋탑 박스, 뮤직 플레이어, 비디오 플레이어, 엔터테인먼트 유닛, 네비게이션 디바이스, 컴퓨터, 핸드-헬드 PCS(personal communication systems) 유닛, 휴대용 데이터 유닛 및/또는 고정 위치 데이터 유닛으로 상기 캐시 메모리를 통합시키는 것을 더 포함하는,
 캐시 메모리의 웨어-레벨링을 위한 방법.

청구항 9

캐시 제어기로서,
 캐시 메모리의 복수의 캐시 세트들에 대한 각각의 메모리 위치 스와핑 연산 이후에 업데이트하고, 각각의 N-1 메모리 위치 스와핑 연산들에서 리셋하는 제 1 레지스터 - N은 상기 캐시 메모리 내의 캐시 세트들의 수임 - ; 및
 상기 캐시 메모리의 복수의 캐시 세트들에 대한 모든 각각의 N-1 메모리 위치 스와핑 연산들 이후에 업데이트하

고, 모든 각각의 (N^2-N) 메모리 위치 스와핑 연산들을 리셋하는 제 2 레지스터를 포함하고,

상기 제 1 레지스터 및 상기 제 2 레지스터는 상기 캐시 세트들의 논리 위치들과 물리 위치들 사이의 관계를 추적하는,

캐시 제어기.

청구항 10

제 9 항에 있어서,

상기 캐시 제어기는 2개의 메모리 위치들을 동적으로 스와핑하고, 상기 제 1 레지스터 및 상기 제 2 레지스터를 이용하여 논리 캐시 세트 수를 물리 캐시 세트 수로 변환하기 위해 상기 캐시 세트들의 스와핑 연산들을 추적하도록 동작가능한,

캐시 제어기.

청구항 11

제 9 항에 있어서,

상기 제 1 레지스터는 메모리 스와핑 연산이 수행될 때, $(\text{SwapReg} + 1) \bmod (N-1)$ 로 세팅된 스왑 레지스터 (SwapReg)인,

캐시 제어기.

청구항 12

제 11 항에 있어서,

상기 제 2 레지스터는 상기 SwapReg가 0과 동일할 때, $(\text{ShiftReg} + 1) \bmod N$ 으로 세팅된 시프트 레지스터 (ShiftReg)인,

캐시 제어기.

청구항 13

제 9 항에 있어서,

상기 제 1 레지스터는 메모리 스와핑 연산이 수행될 때, $(\text{SwapReg} + 1) \bmod (N-1)$ 로 세팅된 스왑 레지스터 (SwapReg)이고,

상기 제 2 레지스터는 상기 SwapReg가 0과 동일할 때, $(\text{ShiftReg} + 1) \bmod N$ 으로 세팅된 시프트 레지스터 (ShiftReg)인,

캐시 제어기.

청구항 14

제 13 항에 있어서,

상기 캐시 제어기는 2개의 메모리 위치들을 동적으로 스와핑하고, 상기 SwapReg 및 상기 ShiftReg를 이용하여 논리 캐시 세트 수를 물리 캐시 세트 수로 변환하기 위해 상기 캐시 세트들의 스와핑 연산들을 추적하도록 동작가능한,

캐시 제어기.

청구항 15

제 9 항에 있어서,

상기 캐시 제어기는 기록 카운터 레지스터가 임계 값과 동일할 때, 메모리 위치 스와핑 연산을 개시하도록 동작가능한,

캐시 제어기.

청구항 16

제 9 항에 있어서,

상기 캐시 제어기는 모바일 폰, 셋탑 박스, 뮤직 플레이어, 비디오 플레이어, 엔터테인먼트 유닛, 네비게이션 디바이스, 컴퓨터, 핸드-헬드 PCS(personal communication systems) 유닛, 휴대용 데이터 유닛 및/또는 고정 위치 데이터 유닛으로 통합되는,

캐시 제어기.

청구항 17

캐시 메모리의 웨어-레벨링을 위한 캐시 제어기로서,

상기 캐시 메모리에의 메모리 기록 연산들의 수가 임계 값에 도달할 때 캐시 세트들에 대해 복수의 메모리 위치 스와핑 연산들을 수행함으로써 상기 캐시 메모리의 캐시 세트들을 동적으로 로테이션시키기 위한 수단 - 각각의 스와핑 연산은 단지 스와핑된 캐시 세트들로부터의 콘텐츠들을 클리어하는 반면, 다른 캐시 세트들의 메모리 콘텐츠들을 그대로 남겨두는 것을 포함함 - ; 및

논리 캐시 세트 수를 물리 캐시 세트 수로 변환하기 위해 상기 스와핑된 캐시 세트들을 추적하기 위한 수단을 포함하는,

캐시 메모리의 웨어-레벨링을 위한 캐시 제어기.

청구항 18

제 17 항에 있어서,

상기 캐시 제어기는 모바일 폰, 셋탑 박스, 뮤직 플레이어, 비디오 플레이어, 엔터테인먼트 유닛, 네비게이션 디바이스, 컴퓨터, 핸드-헬드 PCS(personal communication systems) 유닛, 휴대용 데이터 유닛 및/또는 고정 위치 데이터 유닛으로 통합되는,

캐시 메모리의 웨어-레벨링을 위한 캐시 제어기.

청구항 19

캐시 메모리의 웨어-레벨링을 위한 방법으로서,

상기 캐시 메모리에의 메모리 기록 연산들의 수가 임계 값에 도달할 때 캐시 세트들에 대해 복수의 메모리 위치 스와핑 연산들을 수행함으로써 상기 캐시 메모리의 캐시 세트들을 동적으로 로테이션시키는 단계 - 각각의 스와핑 연산은 단지 스와핑된 캐시 세트들로부터의 콘텐츠들을 클리어하는 반면, 다른 캐시 세트들의 메모리 콘텐츠들을 그대로 남겨두는 것을 포함함 - ; 및

논리 캐시 세트 수를 물리 캐시 세트 수로 변환하기 위해 상기 스와핑된 캐시 세트들을 추적하는 단계를 포함하는,

캐시 메모리의 웨어-레벨링을 위한 방법.

청구항 20

제 19 항에 있어서,

모바일 폰, 셋탑 박스, 뮤직 플레이어, 비디오 플레이어, 엔터테인먼트 유닛, 네비게이션 디바이스, 컴퓨터, 핸드-헬드 PCS(personal communication systems) 유닛, 휴대용 데이터 유닛 및/또는 고정 위치 데이터 유닛으로 상기 캐시 메모리를 통합시키는 단계를 더 포함하는,

캐시 메모리의 웨어-레벨링을 위한 방법.

발명의 설명

기술분야

[0001] 본 개시는 일반적으로 메모리들 및 캐시들에 관한 것이다. 더 상세하게는, 본 개시는 제한된 기록 내구성(endurance)을 갖는 캐시들에 대한 세트-간 웨어-레벨링에 관한 것이다.

배경기술

[0002] 무선 통신들 또는 다른 애플리케이션을 위해 이용될 수 있는 고속 디지털 전자 기술(electronics)에 대해, 비-휘발성 메모리들이 이용된다. 그러나, ReRAM(resistive random access memory) 및 PCRAM(phase-change random access memory)과 같은 비-휘발성 메모리들은 제한된 기록 내구성을 갖는다. 기록 내구성은, 저장 매체들이 신뢰성이 없어지게 되기 전에 메모리의 블록에 적용될 수 있고, 얼마나 자주 그리고 얼마나 완전히 메모리가 이용되는지를 추정함으로써 통상적으로 계산되는 프로그램/사이클들의 수로서 정의될 수 있다. 다시 말해서, 기록 내구성은 특정 타입의 저장 매체들의 서비스 연한(service life)을 측정한다.

[0003] 웨어-레벨링은 저장 매체들의 기록 내구성(예를 들어, 서비스 연한)을 연장하기 위해 이용되는 기법이며, 캐시 설계의 부분이다. 하나의 웨어-레벨링 접근법은 재기록들이 저장 매체에 걸쳐 균등하게 분포되도록 데이터를 배열한다. 이러한 방식으로, 어떠한 단일 블록도 고 집중의 기록 사이클들로 인하여 실패하지는 않는다. 웨어-레벨링에 대한 다른 접근법들은 기록이 발생할 때마다 맵을 동적으로 업데이트하는 것을 포함할 수 있는데, 맵은 후속적으로, 기록된 블록을 새로운 블록에 링크시킨다. 또 다른 접근법은 동일한 블록들을 그들을 대체하지 않고 정적으로 유지하지만, 블록들을 주기적으로 로테이션(rotate)시켜서, 이들이 다른 데이터에 의해 이용될 수 있게 한다.

[0004] (예를 들어, 컴퓨터들에 대한 메인 메모리들에서 또한 이용될 수 있는) 비-휘발성 메모리들에 대한 웨어-레벨링은 잘 알려져 있으며, 잘 탐구되고 있다(well explored). 그럼에도 불구하고, 온-칩 캐시들에 대한 웨어-레벨링을 이용할 때, 비-휘발성 메모리들에 대해 통상적으로 이용되는 전통적 웨어-레벨링 접근법들은 아주 과한 성능 오버헤드를 나타낸다. 따라서, 고성능 오버헤드는 제한된 기록 내구성을 갖는 캐시들에 대한 웨어-레벨링 기법들의 유효성(effectiveness)을 억제한다.

발명의 내용

[0005] 본 개시의 일 양상에 따라, 캐시 메모리를 세트-간 웨어-레벨링하기 위한 캐시 제어기가 설명된다. 캐시 제어기는 캐시 메모리의 캐시 세트들에 대한 각각의 메모리 위치 스와핑 연산 이후에 업데이트하고, 각각의 N-1 메모리 위치 스와핑 연산들에서 리셋하는 제 1 레지스터를 포함한다. N은 캐시 메모리 내의 캐시 세트들의 수이다. 캐시 제어기는 캐시 메모리의 캐시 세트들에 대한 모든 각각의 N-1 메모리 위치 스와핑 연산들 이후에 업데이트하고, 모든 각각의 (N²-N) 메모리 위치 스와핑 연산들을 리셋하는 제 2 레지스터를 더 포함한다. 제 1 레지스터 및 제 2 레지스터는 캐시 세트들의 논리 위치들과 물리 위치들 사이의 관계를 추적할 수 있다.

[0006] 본 개시의 또 다른 양상에 따라, 캐시 메모리의 세트-간 웨어-레벨링을 위한 방법이 설명된다. 방법은, 캐시 메모리에의 메모리 기록 연산들의 수가 임계 값에 도달할 때, 캐시 세트들에 대해 메모리 위치 스와핑 연산들을 수행함으로써 캐시 메모리의 캐시 세트들을 동적으로 로테이션시키는 것을 포함한다. 각각의 스와핑 연산은 단지 스와핑된 캐시 세트들로부터의 콘텐츠들을 클리어하는 반면, 다른 캐시 세트들의 메모리 콘텐츠들을 그대로 남겨두는 것을 포함할 수 있다. 방법은 또한, 논리 캐시 세트 수를 물리 캐시 세트 수로 변환하기 위해 스와핑된 캐시 세트들을 추적하는 것을 포함한다.

[0007] 본 개시의 추가 양상에 따라, 캐시 메모리의 세트-간 웨어-레벨링을 위한 캐시 제어기가 설명된다. 캐시 제어기는 캐시 메모리에의 메모리 기록 연산들의 수가 임계 값에 도달할 때, 캐시 세트들에 대해 메모리 위치 스와핑 연산들을 수행함으로써 캐시 메모리의 캐시 세트들을 동적으로 로테이션시키기 위한 수단을 포함한다. 각각의 스와핑 연산은 단지 스와핑된 캐시 세트들로부터의 콘텐츠들을 클리어하는 반면, 다른 캐시 세트들의 메모리 콘텐츠들을 그대로 남겨두는 것을 포함할 수 있다. 캐시 제어기는 논리 캐시 세트 수를 물리 캐시 세트 수로 변환하기 위해 스와핑된 캐시 세트들을 추적하기 위한 수단을 더 포함한다.

[0008] 본 개시의 또 다른 양상에 따라, 캐시 메모리의 세트-간 웨어-레벨링을 위한 방법이 설명된다. 방법은 캐시 메모리에의 메모리 기록 연산들의 수가 임계 값에 도달할 때, 캐시 세트들에 대해 메모리 위치 스와핑 연산들을 수행함으로써 캐시 메모리의 캐시 세트들을 동적으로 로테이션시키는 단계를 포함한다. 각각의 스와핑 연산은 단지 스와핑된 캐시 세트들로부터의 콘텐츠들을 클리어하는 반면, 다른 캐시 세트들의 메모리 콘텐츠들

을 그대로 남겨두는 것을 포함할 수 있다. 방법은 또한, 논리 캐시 세트 수를 물리 캐시 세트 수로 변환하기 위해 스와핑된 캐시 세트들을 추적하는 단계를 포함한다.

[0009]

[0009] 위의 설명은, 다음의 상세한 설명이 더 양호하게 이해될 수 있도록 본 개시의 특징들 및 기술적 이점들을 상당히 광범위하게 요약하였다. 본 개시의 추가적인 특징들 및 이점들이 이하에서 설명될 것이다. 당업자들은 본 개시가 본 개시의 동일한 목적들을 수행하기 위해 다른 구조들을 수정 또는 설계하기 위한 기초로서 쉽게 이용될 수 있다는 것을 인식하여야 한다. 또한, 당업자들은 이러한 등가의 구조들이 첨부된 청구항들에 기술되는 본 개시의 교시들을 벗어나지 않는다는 것을 인지하여야 한다. 추가적인 목적들 및 이점들과 함께 본 개시의 구성 및 동작 방법 둘 다에 대해, 본 개시의 특징으로 여겨지는 신규한 특징들은 첨부한 도면들과 관련하여 고려될 때 다음의 설명으로부터 더 잘 이해될 것이다. 그러나, 도면들 각각은 단지 예시 및 설명을 목적으로 제공되며, 본 개시의 제한들에 대한 한정으로서 의도되지 않는다는 것이 명백하게 이해될 것이다.

도면의 간단한 설명

[0010]

[0010] 본 개시의 더 완전한 이해를 위해, 첨부한 도면들과 함께 취해진 다음의 설명들에 대한 참조가 이제 이루어진다.

[0011] 도 1은 본 개시의 양상에 따른, 캐시 메모리의 웨어-레벨링을 위한 캐시 제어기를 포함하는 예시적 캐시 메모리의 도면이다.

[0012] 도 2는 본 개시의 양상에 따라 웨어-레벨링하기 위한 예시적 로테이션을 도시하는 도면이다.

[0013] 도 3은 본 개시의 양상에 따라 웨어-레벨링하는 연산을 도시하는 로직 흐름도이다.

[0014] 도 4는 본 개시의 양상에 따라 웨어-레벨링하는 연산을 도시하는 로직 흐름도이다.

[0015] 도 5는 본 개시의 양상에 따라 웨어-레벨링하기 위한 방법을 예시하는 프로세스 흐름도이다.

[0016] 도 6은 본 개시의 양상에 따라 웨어-레벨링하기 위한 방법을 예시하는 프로세스 흐름도이다.

[0017] 도 7은 본 개시의 구성이 유리하게 이용될 수 있는 예시적 무선 통신 시스템을 도시하는 블록도이다.

[0018] 도 8은 일 구성에 따른, 반도체 컴포넌트의 회로, 레이아웃 및 로직 설계에 대해 이용되는 설계 워크스태이션을 예시하는 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0011]

[0019] 첨부된 도면들과 관련하여 아래에서 기술되는 상세한 설명은 다양한 구성들의 설명으로서 의도되며, 본원에 설명되는 개념들이 실시될 수 있는 구성들만을 표현하는 것으로 의도되지 않는다. 상세한 설명은 다양한 개념들의 완전한 이해를 제공하기 위해 특정 세부사항들을 포함한다. 그러나, 이러한 개념들이 이러한 특정 세부사항들 없이 실시될 수 있음이 당업자들에게 명백해질 것이다. 일부 경우들에서, 공지된 구조들 및 컴포넌트들은 이러한 개념들을 모호하게 하는 것을 회피하기 위해 블록도 형태로 도시된다. 본원에서 설명되는 바와 같이 "및/또는"이라는 용어의 이용은 "포괄적 또는"을 표현하는 것으로 의도되고, "또는"이라는 용어의 이용은 "배타적 또는"을 표현하는 것으로 의도된다.

[0012]

[0020] SRAM(static random access memory) 및 eDRAM(embedded dynamic RAM)과 같은 메모리들은 보통, 현대 마이크로프로세서들에서 온-칩 캐시 설계를 위해 이용된다. 현대 컴퓨터들 및 디바이스들은 또한, 더 큰 온-칩 캐시들을 특정하지만, 전통적 SRAM 또는 eDRAM 캐시들의 확장성(scalability)은 누설 전력 및 셀 밀도와 같은 기술 제한들에 의해 점점 더 제약된다. 최근에, 예를 들어, 위상-변화 RAM(random access memory), 스핀-토크 전달 RAM 및 저항성 RAM과 같은 새로운 NVM(non-volatile memory) 기술들은 온-칩 캐시들에 대해 이용될 유망한 대안적 메모리 기술들로서 탐구되었다. SRAM 및 eDRAM와 같은 전통적 메모리들과 비교하여, 이러한 신흥 비-휘발성 메모리 기술들은 높은 밀도, 낮은 대기 전력, 낮은 전압, 더 양호한 확장성 및 비-휘발성의 공통 이점들을 갖는다. 그러나, 이들의 채택은 이들의 제한된 기록 내구성에 의해 방해될 수 있다. 이 문제는 이러한 정책들이 기록 변화를 인지하지 못하기 때문에, 캐시 블록들에 대한 불균형적 기록 트래픽을 초래하는 기존의 캐시 관리 정책들에 의해 증폭될 것이다. 이러한 정책들은 원래 SRAM 캐시들에 대해 설계되었으며, 캐시 블록들에의 기록에 관하여 상당한 비-균일성을 초래하는데, 이는 과도하게(heavily)-기록된 캐시 블록들이 대부분의 다른 블록들보다 훨씬 더 빠르거나 더 앞서지 못하게 한다.

[0013]

[0021] 많은 웨어-레벨링 기법들은 비-휘발성 메모리 기술들의 수명을 연장하도록 제안되었지만, 캐시와 메인

메모리 연산 메커니즘들 사이의 차는 비-휘발성 메모리들에 대한 기존의 웨어-레벨링 기법들이 비-휘발성 캐시들에 대해 부적당하게 만든다. 이러한 문제들을 다루고, 세트-간 기록 변화들을 감소시키기 위해, 스왑-시프트 방식은 비-휘발성 메모리 캐시들에 대해 캐시 세트-간 기록 변화들을 감소시키도록 제공된다. 방식은, 단지 하나의 글로벌 카운터 및 2개의 글로벌 레지스터들만을 이용하는 아주 작은 하드웨어 오버헤드를 갖는다. 이 방식을 채택함으로써, 저-레벨 온-칩 비-휘발성 메모리 캐시들의 수명이 향상될 수 있다.

[0014] [0022] 기록 변화는 제한된 기록 내구성을 갖는 비-휘발성 메모리들을 이용하는 임의의 캐시 또는 메모리 서브시스템을 설계하는 것에 있어서의 중요한 관심사이다. 단지 최악의 경우의 기록 트래픽을 경험하는 메모리의 작은 서브세트만이 심지어 다수의 셀들이 웨어-아웃(wear-out)으로부터 멀어질 때에도 데드 캐시 또는 메모리 서브시스템을 초래할 수 있기 때문에, 큰 기록 변화는 제품 수명을 크게 저하시킬 수 있다.

[0015] [0023] 도 1은 본 개시의 양상에 따른 캐시 메모리(100)의 세트-간 웨어-레벨링을 위한 캐시 제어기(140)를 포함하는 예시적 캐시 메모리(100)의 도면이다. 캐시 메모리(100)는 페이지 수(102), 세트 수(104), 바이트 수(106), 캐시 웨이(108), 태그 부분(110), 데이터 부분(112), 캐시 블록(114), 캐시 세트들(116), 태그 감지 증폭기(118a), 데이터 감지 증폭기(118b), 태그 출력(120), 비교기(122), 로직 게이트(124), 캐시 그룹(126), 선택 회로(128) 및 워드 출력(130)을 포함한다.

[0016] [0024] 캐시 메모리(100) 내의 어드레스는 페이지 수(102), 세트 수(104) 및 바이트 수(106)를 포함할 수 있다. 일 구현에서, 페이지 수(102)는 가상 페이지 수일 수 있다. 세트 수(104)는 캐시 세트들(116) 중 하나에 대응한다. 캐시 블록(114)은 태그 부분(110) 및 데이터 부분(112)을 포함한다. 태그 부분(110)은 데이터 부분(112) 내의 실제 데이터의 어드레스의 부분 또는 데이터 부분(112) 내의 데이터를 로케이팅하기 위한 다른 식별 정보를 포함할 수 있다. 데이터 부분(112)은 실제 데이터를 포함한다. 도 1의 수평 그룹핑에 의해 보여질 수 있는 바와 같이, 캐시 세트들(116) 중 하나는 캐시 블록들(114)의 하나의 세트이다. 도 1에 보여질 수 있는 바와 같이, 캐시 웨이(108)는 캐시 블록들(114)의 또 다른 그룹이지만, 수직 그룹핑 내에 있다. 태그 감지 증폭기(118a) 및 데이터 감지 증폭기(118b)는 캐시 엔트리들로부터의 로직 레벨들을 감지하여서, 데이터가 출력될 때 적절히 (1 또는 0으로서) 해석되게 한다.

[0017] [0025] 태그 감지 증폭기(118a)의 출력인 태그 출력(120)에서의 데이터는 페이지 프레임 수, 유효 비트 및 코히런스 비트들을 포함할 수 있다. 그 다음, 태그 출력(120)으로부터의 데이터는 비교기(122)에 의해 페이지 수(102)와 비교되고, 이는 2개의 값들이 동일하지 여부를 나타낸다. 그 값들이 동일하고, 히트(hit)가 존재하면, 비교기(122)의 출력은 데이터 감지 증폭기(118b)의 출력과 함께, 로직 게이트(124)로 입력된다. 로직 게이트(124)의 출력은 캐시 그룹(126)에 나타난다. 일 구현에서, 캐시 그룹들(126) 중 하나는 다수의 워드들을 포함한다. 캐시 그룹(126)은 선택 입력으로서 바이트 수(106)를 이용하는 선택 회로(128)로 입력된다. 선택 입력으로서 바이트 수(106)를 이용하는 선택 회로(128)의 출력은 워드 출력(130)이다.

[0018] [0026] 도 1은 또한 n-웨이 세트-연관적 캐시에 대한 예시적 블록도이고, 본 개시에 대해 이용되는 다른 타입들의 캐시들이 존재할 수 있다. 세트-연관적 캐시는 병렬로 동작되는 몇몇 다이렉트-맵핑된 캐시들로 구성될 수 있다(예를 들어, 하나의 다이렉트-맵핑된 캐시는 태그 부분(110) 및 데이터 부분(112)을 포함하는 캐시 엔트리일 수 있음). 데이터 관독은 페이지 수(102) 뿐만 아니라 블록-유효 비트(이는 태그 또는 메타데이터 엔트리의 일부일 수 있음) 및 페이지 허가들(페이지 수(102)의 부분)과의 태그 비교에 의해 제어될 수 있다. 캐시 칼럼 크기는 또한, 가상 메모리 페이지 크기와 동일할 수 있고, 캐시 인덱스는 페이지 수(102) 또는 가상 페이지 수로부터의 피트들(pits)을 이용하지 않을 수 있다.

[0019] [0027] 캐시 블록들(114) 중 개별 캐시 블록들은 하나의 방향에서 캐시 세트들(116) 중 하나로 그리고 또 다른 방향에서 캐시 웨이(108)로 그룹핑된다. 캐시 블록들(114)은 또한, 이들이 얼마나 종종 기록되는지에 따라, 로드 분배들을 가질 수 있다. 캐시 블록들(114) 중 일부는 과도하게 기록되고, 캐시 블록들(114) 중 나머지는 거의 기록되지 않는다. 따라서, 이것은 세트-간 기록 변화를 야기한다. 세트-간 기록 변화는 기록 활동에 관하여 변경되는 캐시 세트들(116) 사이에서 발생한다. 즉, 캐시 세트들(116) 중 하나의 캐시 세트로부터의 캐시 블록들(114)은 캐시 세트들(116) 중 또 다른 캐시 세트로부터의 캐시 블록들(114)보다 많거나 적게 기록될 수 있다. 게다가, 상이한 캐시 세트들(116)이 캐시 세트들(116) 중 나머지와 비교하여 전체로서 기록될 때, 세트-간 기록 변화가 또한 발생한다.

[0020] [0028] 이 구성에서, 캐시 제어기(140)는 기록 트래픽을 캐시 웨이들(108)에 대한 상이한 행들로 그리고 상이한 열들로 균등하게 분배한다. 도 1에서 캐시 웨이들(108)은 열들을 점유하는 것으로서 도시되고, 캐시 세트들(116)은 행들을 점유하는 것으로서 도시되지만, 캐시 메모리(100)의 구현이 이 구성에 제한되는 것은 아니다.

특히, 캐시 웨이들(108)은 캐시 메모리(100) 내의 열들이 아닐 수 있는 행들 또는 다른 구조들을 점유할 수 있고, 캐시 세트들(116)은 캐시 메모리(100) 내의 행들이 아닐 수 있는 열들 또는 다른 구조들을 점유할 수 있다.

[0021] [0029] 메인 메모리 웨어-레벨링 기법들은 통상적으로, 어드레스 리맵핑을 구현하기 위해 데이터 이동을 이용한다. 이것은, 메인 메모리에서, 데이터가 손실될 수 없으며, 각각의 리맵핑 이후에 새로운 포지션으로 이동될 수 있기 때문이다. 그럼에도 불구하고, 데이터 이동 연산들은 항상 영역(area) 및 성능 오버헤드를 발생시킨다. 첫째, 데이터 이동은 데이터를 수신하기 위해 일시적 데이터 저장 위치를 필요로 한다. 둘째, 하나의 캐시 세트 이동은 몇몇 블록 판독 및 기록 연산들을 포함한다. 따라서, 캐시 포트는 데이터 이동 동안 차단되고, 시스템 성능이 결과적으로 저하된다. 하나의 예시적 데이터 이동 방식이 메모리 캐시들에 적용되도록 메인 메모리로부터 확장될 때, 하나의 추가 캐시 세트(갭 세트)가 부가되고, 하나의 세트로부터 갭 세트로의 데이터는 주기적으로 이동된다. 캐시 기법들이 더 많은 성능 감지식이기 때문에, 메인 메모리 웨어-레벨링 기법들은 직접적으로 이용될 수 없다. 따라서, 데이터 이동의 이용은 캐시 세트-간 웨어-레벨링 기법들을 설계할 때 재고려될 수 있다.

[0022] [0030] 비-휘발성 메모리 캐시들에 대한 세트 어드레스 리맵핑을 구현하기 위한 또 다른 옵션은 데이터 무효화를 수행하는 것이다. 캐시들 내의 데이터가 하위-레벨 메모리들로부터 추후에 다시 판독될 수 있기 때문에, 캐시 라인 무효화들이 이용될 수 있다. 캐시들의 이러한 특별한 특징은 캐시 세트-간 웨어-레벨링 기법들을 설계하기 위한 새로운 기회를 제공한다.

[0023] [0031] 데이터 이동과 비교하여, 무효화들은 임의의 영역 오버헤드를 발생시키지 않는다. 따라서, 본 개시의 일 양상은 이전의 메인 메모리 웨어-레벨링 기법들을 수정하며, 무효화들을 이용하여 비-휘발성 메모리 캐시들의 세트-간 기록 변화를 감소시키기 위해 스왑-시프트 웨어-레벨링 방식을 이용함으로써 이전의 메인 메모리 웨어-레벨링 기법들을 강화한다.

[0024] [0032] 비-휘발성 메인 메모리들에 대한 기존의 웨어-레벨링 기법들과는 대조적으로, 스왑-시프트 방식은 비-휘발성 메모리 캐시들에 대해 설계된다. 스왑-시프트 방식은 영역 및 성능 오버헤드 둘 모두를 감소시키기 위해 세트 어드레스 맵핑을 변경할 때, 데이터 이동 대신 데이터 무효화를 이용한다.

[0025] [0033] 스왑-시프트 방식의 일 구성은 세트들 간의 저장된 데이터를 로테이션시키기 위해 캐시 물리 세트들의 맵핑을 시프트한다. 그럼에도 불구하고, 한번에 모든 캐시 세트들을 시프트하는 것은 상당한 성능 오버헤드를 초래한다. 이 문제를 해결하기 위해, 캐시 제어기(140)의 스왑-시프트 방식은 단지 한 번에 2개의 세트들의 맵핑을 스와핑하고, 모든 캐시 세트들은 완전한 스왑 로테이션 이후 한 단계 시프트될 수 있다.

[0026] [0034] 이 구성에서, 캐시 제어기(140)는 캐시에의 메모리 기록 연산들의 수를 저장하기 위해 스왑-시프트 방식에서 이용되는 글로벌 카운터(142)를 포함하고, 이는 변수 이름 "numWrite"에 의해 표시된다. 캐시 제어기(140)는 또한, 현재 스와핑 값을 저장하기 위해 이용되는 스왑 레지스터(144)(SwapReg)를 포함한다. SwapReg는 초기에 0으로 셋팅되며, 순환적으로 0으로부터 $N-1$ 로 변경되는데, 여기서, N 은 캐시 내의 세트들의 수이다. 캐시 제어기(140)는 현재 시프트 값을 저장하는 시프트 레지스터(146)(ShiftReg)를 더 포함한다. ShiftReg는 순환적으로 0으로부터 N 으로 변경된다. 이 2개의 값들, SwapReg 및 ShiftReg는 시프트-스왑 방식에서의 두 타입들의 로테이션들, 스왑 로테이션 및 시프트 로테이션을 제어하기 위해 캐시 제어기(140)에 의해 이용된다.

[0027] [0035] 첫째, 스왑 로테이션이 설명된다. numWrite가 특정한 미리 정의된 임계치("threshold")와 동일할 때, SwapReg는 1만큼 증분되고, SwapReg가 $N-1$ 단계들만큼 이동될 때, 하나의 스왑 로테이션이 발생한다. 따라서, 하나의 스왑 로테이션은 $N-1$ 회의 스와핑들로 구성된다.

[0028] [0036] 둘째, 시프트 로테이션이 설명될 것이다. ShiftReg는 각각의 스왑 로테이션 이후 1만큼 증분되고, ShiftReg가 N 단계들만큼 이동될 때, 하나의 시프트 로테이션이 발생한다. 따라서, 각각의 시프트 로테이션은 N 회의 스왑 로테이션들로 구성된다.

[0029] [0037] 도 2는 본 개시의 양상에 따라 웨어-레벨링하기 위한 예시적 로테이션을 도시하는 도면(200)이다. 도면(200)은 캐시 구조(202) 및 로테이션 박스들(204, 206, 208, 210, 212, 214, 216 및 218)의 세트를 포함한다. 캐시 구조(202)는 캐시 블록들(114), 캐시 웨이들(108) 또는 캐시 세트들(116)을 포함하는 임의의 구조이고, 간략함을 위해 캐시 구조(202)가 캐시 세트들(202a, 202b, 202c 및 202d)의 그룹을 갖는 것으로 표현되지만, 캐시 구조(202)는 캐시 메모리(100) 자체의 전체를 포함할 수 있다. 캐시 구조(202)의 서브-구조들 또는 캐시 세트들의 수가 4에 제한되는 것은 아니지만, 제 1 캐시 세트(202a), 제 2 캐시 세트(202b), 제 3 캐시 세트(202c) 및 제 4 캐시 세트(202d)는 도면(200)의 로테이션 예에 대해 도시된 예시적 캐시 세트들이다. 로테이션 프로세스

스가 이제 설명된다.

- [0030] [0038] 로테이션 박스(204)에서, 캐시 구조(202)는 자신의 초기 포지션 내에 있고, 여기서, 제 1 캐시 세트(202a)는 "0"로 라벨링되고, 제 2 캐시 세트(202b)는 "1"로 라벨링되며, 제 3 캐시 세트(202c)는 "2"로 라벨링되고, 제 4 캐시 세트는 "3"으로 라벨링된다. 캐시 세트들(202a, 202b, 202c 및 202d) 각각 내에 저장된 데이터가 또한 존재할 수 있다. SwapReg 카운터는 캐시 세트들이 스와핑된 횟수를 카운트한다. 카운터는 0으로 초기화될 수 있다. ShiftReg 카운터는 캐시 구조(202)의 전체가 포지션들을 시프트한 횟수를 카운트하고, (캐시 구조(202) 내의 캐시 세트들 모두가 하나의 포지션을 이동했다는 점에서), 또한 초기화되고, 0으로 세팅될 수 있다. 로테이션에 대한 종래의 접근법은 일시적 블록을 이용하며, 시간이 지남에 따라 모든 캐시 세트들 또는 캐시 구조들을 클리어링한다. 그럼에도 불구하고, 도면(200)에 도시된 구현에 있어서, 일시적 블록이 이용되지 않거나, 캐시 세트들의 데이터 콘텐츠들이 삭제되거나 플러싱되지 않는다. 이것은, 데이터가 실제로 스와핑되지 않으며 동일한 위치 또는 포지션 내에서 유지되기 때문에, 트루(true)이다. 단지 캐시 세트들의 포지션들만이 스와핑되고, 단지 스와핑된 캐시 세트들로부터의 콘텐츠들만이 클리어링되어서, 모든 다른 캐시 세트들의 콘텐츠들이 그대로 남겨진다. 캐시 세트들 각각은 또한 이들의 새로운 위치로 이동되고, 새로운 데이터는 어떠한 데이터도 이동시킬 필요없이 새로운 캐시 세트들로 자동-로딩된다. 다시, 스와핑된 유일한 것은 캐시 세트들의 위치이다. 모든 것은, 데이터를 실제로 이동시키는 것 대신, 스와핑 연산들에 의해 이루어진다.
- [0031] [0039] 로테이션 박스(206)에서, 제 1 캐시 세트(202a)(0)의 포지션은 제 2 캐시 세트(202b)(1)의 포지션과 스와핑된다. 결과적으로, 제 2 캐시 세트(202b)(1)는 이제 최초(very first) 또는 상위(top) 캐시 세트이고, 제 1 캐시 세트(202a)(0)는 제 1 캐시 세트(202a)(0) 다음 또는 아래의 제 2 캐시 세트가 된다. 일 구현에서, 이 스와핑이 발생하면, 새로운 데이터는, 그들의 새로운 포지션들에서 한 번 새로운 캐시 세트들로 자동-리로딩될 것이다. 일 구현에서, 스와핑된 캐시 세트들은 그들의 데이터를 보유할 것이어서, 캐시 세트들이 스와핑되면, 새로운 데이터는 그 캐시 세트들로 자동-리로딩되지 않을 것이다. 캐시 세트들이 스와핑된 횟수를 카운트하는 SwapReg 카운터는 또한 1만큼 증분될 수 있다. 로테이션 박스(208)에서, (현재 제 2 포지션, 제 2 캐시 세트(202b)(1)의 이전(old) 포지션에서의) 제 1 캐시 세트(202a)(0)의 포지션은 제 3 캐시 세트(202c)(2)의 포지션과 스와핑된다. 이제, 제 3 캐시 세트(202c)(2)는 제 2 포지션(제 2 캐시 세트(202b)(1)의 이전(former) 포지션)이 되고, 제 1 캐시 세트(202a)(0)는 제 3 포지션(제 3 캐시 세트(202c)(2)의 이전 포지션)이 된다. 그 다음, SwapReg 카운터는 다시 1만큼 증분될 수 있다.
- [0032] [0040] 로테이션 박스(210)에서, (현재 제 3 포지션, 제 3 캐시 세트(202c)(2)의 이전 포지션에서의) 제 1 캐시 세트(202a)(0)의 포지션은 제 4 캐시 세트(202d)(3)의 포지션과 스와핑된다. 이제, 제 4 캐시 세트(202d)(3)는 제 3 포지션(제 3 캐시 세트(202c)(2)의 이전 포지션)이 되고, 제 1 캐시 세트(202a)(0)는 제 4 포지션(제 4 캐시 세트(202d)(3)의 이전 포지션)이 된다. 모든 캐시 세트들이 시프트되기 때문에, SwapReg 카운터는 그 다음 초기화될 수 있고, 스와핑의 세 라운드들 이후, 캐시 구조(202) 내에서의 모든 레지스터들이 하나의 포지션만큼 시프트되기 때문에, ShiftReg 카운터가 증분된다.
- [0033] [0041] 로테이션 박스(212)로 계속 진행하면, 제 2 캐시 세트(202b)(1)의 포지션은 제 3 캐시 세트(202c)(2)의 포지션과 스와핑된다. 이제, 제 2 캐시 세트(202b)(1)는 제 2 포지션(제 3 캐시 세트(202c)(2)의 이전 포지션)이 되고, 제 3 캐시 세트(202c)(2)는 제 1 포지션(제 2 캐시 세트(202b)(1)의 이전 포지션)이 된다. SwapReg 카운터는 또한, 1만큼 증분된다. 로테이션 박스(214)에서, 제 2 캐시 세트(202b)(1)의 포지션은 제 4 캐시 세트(202d)(3)의 포지션과 스와핑된다. 이제, 제 4 캐시 세트(202d)(3)는 제 2 포지션(제 2 캐시 세트(202b)(1)의 이전 포지션)이 되고, 제 4 캐시 세트(202d)(3)는 제 2 포지션(제 2 캐시 세트(202b)(1)의 이전 포지션)이 된다. SwapReg 카운터는 또한 1만큼 증분한다. 로테이션 박스(216)에서, 제 2 캐시 세트(202b)(1)의 포지션은 제 1 캐시 세트(202a)(0)의 포지션과 스와핑된다. 이제, 제 2 캐시 세트(202b)(1)는 제 4 포지션(제 1 캐시 세트(202a)(0)의 이전 포지션)이 되고, 제 1 캐시 세트(202a)(0)는 제 3 포지션(제 2 캐시 세트(202b)(1)의 이전 포지션)이 된다. SwapReg 카운터는 그 다음 초기화될 수 있고, 스와핑의 또 다른 세 라운드들 이후, 캐시 구조(202) 내에서의 모든 레지스터들이 2개의 포지션들만큼 시프트되기 때문에, ShiftReg 카운터는 1만큼 증분된다.
- [0034] [0042] 일 구현에서, 모든 것이 스와핑 연산들에 의해 수행되기 때문에, 전체 캐시의 콘텐츠들은 플러싱되지 않는다. 게다가, 스와핑 연산들은 임의의 성능 저하를 포함하지 않는다. 스와핑을 수행하기 위해 스와핑 연산들을 이용함으로써, "시프트 업" 연산들 및/또는 추적이 발생하지 않는다. 일 구현에서, 데이터는 단지 그 포지션들에서 스와핑되지 않고, 데이터는 그 다음, 새로운 스와핑된 위치로 리로딩된다. 일 구현에서, 무효 캐시 세트들과 같은 미이용된 포지션들에서의 데이터는 단순히 폐기된다. 일 구현에서, 하나의 스와핑에 대해,

SwapReg 카운터 포지션(예를 들어, 캐시-레지스터[SwapReg])에 대응하는 캐시 세트의 콘텐츠들 및 SwapReg 카운터 플러스 1(예를 들어, 캐시-레지스터[SwapReg + 1])에 대응하는 캐시 세트의 콘텐츠들이 폐기될 수 있다. 그 다음, 이들의 물리적 위치들이 스와핑된다. N-1회의 스와핑들 이후 - N은 캐시 구조에서의 캐시 세트들의 수임 -, 캐시 구조 내에서의 모든 물리 위치들(캐시 세트들 각각)은 모두 1만큼 시프트된다.

[0035] [0043] 일 구현에서, SwapReg 및 ShiftReg 카운터들은, 도 1에 도시된 바와 같이, 데이터를 저장하기 위한 캐시 제어기(140)의 스왑 레지스터(144) 및 시프트 레지스터(146)로서 구현될 수 있다. 이러한 경우, 레지스터들은 이들의 각각의 카운터들에 대한 값의 추적을 유지하는 어떤 종류의 수치 데이터로서 구현된다.

[0036] [0044] 도 3은 본 개시의 양상에 따라 캐시 메모리를 웨어-레벨링하기 위한 캐시 제어기의 연산을 도시하는 논리 흐름도(300)이다. 도 3은 ShiftReg 및 SwapReg 카운터들이 기록 활동에 관하여 웨어-레벨링 시스템에서 어떻게 업데이트되는지를 도시한다. 블록(302)에서, 캐시 기록이 존재하는지 여부가 결정된다. 캐시 기록이 존재하면, 블록(304)에서, "numWrite"로 알려진 캐시 기록 카운터는 1만큼 증분된다. 캐시 기록이 존재하지 않으면, 프로세스는 블록(302) 전으로 다시 진행한다. numWrite 카운터는 또한, 스와핑할 시기를 트리거링한다. 캐시 세트의 스와핑을 개시하기에 얼마나 많은 기록들이 충분한지를 결정하는 일부 미리 결정된 임계 값이 또한 존재한다. 예를 들어, 블록(306)에서, numWrite 카운터가 미리 결정된 임계치와 동일한지 여부가 결정된다. 만약 그렇다면, 블록(308)에서, 스와핑이 개시되어, numWrite 카운터를 0 값으로 업데이트하고, SwapReg 카운터를 $(\text{SwapReg} + 1) \bmod (N-1)$ 의 값으로 업데이트한다. Mod는 모듈로 연산자이고, N은 선택된 캐시 구조에서의 캐시 세트들의 총 수이다. 블록(306)에서, numWrite 카운터가 미리 결정된 임계치와 동일하지 않으면, 프로세스는 블록(304)으로 다시 진행한다. 블록(310)에서, SwapReg 카운터가 0과 동일한지 여부가 결정된다. 만약 그렇다면, 블록(312)에서, ShiftReg 카운터가 $(\text{ShiftReg} + 1) \bmod N$ 의 값으로 업데이트된다. 이것은 전체 캐시 구조를 1만큼 시프트한다. 블록(310)에서, SwapReg 카운터의 값이 0과 동일하지 않으면, 프로세스는 블록(308)으로 다시 진행한다.

[0037] [0045] 도 3은 또한, 다음의 의사 코드에 의해 표현될 수 있다:

```
If (there is a cache write),
then: numWrite++;
If (numWrite == threshold),
then: numWrite = 0;
then: SwapReg = (SwapReg + 1) mod (N-1);
if (SwapReg == 0),
then: ShiftReg = (ShiftReg + 1) mod (N).
```

[0038]

[0039] [0046] 도 4는 본 개시의 양상에 따라 캐시 메모리를 웨어-레벨링을 위한 캐시 제어기의 연산을 도시하는 로직 흐름도(400)이다. 도 4는 기록들의 수를 표현하는 numWrite 카운터를 저장하는 글로벌 카운터(404)를 도시하고, 각각의 기록은 캐시 기록 동작(402)에 의해 표현된다. ("SwapReg"로서 반영되는) SwapReg 카운터 및 ("ShiftReg"로서 표현되는) ShiftReg 카운터는 SwapReg 및 ShiftReg 값들을 각각 저장하기 위한 데이터 레지스터들로서 구현될 수 있다.

[0040] [0047] 도 3에 관하여 설명된 바와 같이, 캐시 기록(402)이 존재하면, 글로벌 카운터(404)는 증가된다. 406에서, 글로벌 카운터가 미리 결정된 임계 값과 동일하면, 블록 408에서, 스와핑이 발생하여, SwapReg 카운터를 $(\text{SwapReg} + 1) \bmod (N-1)$ 의 값으로 업데이트하고, 여기서, mod는 modulo 연산자이고, N은 총 캐시 세트들의 수이다. SwapReg가 0이면(410), 블록(412)에서, ShiftReg 카운터는 $(\text{ShiftReg} + 1) \bmod N$ 의 값으로 업데이트된다.

[0041] [0048] 논리 세트(LS) 수가 논리 세트 수 입력(414)으로서 입력될 때, 물리 세트(PS) 수는 3개의 상이한 상황들에 기초하여 물리 세트 수 출력(418)으로서 컴퓨팅될 수 있다.

[0042] [0049] 첫째, 로직 박스(416)에 도시된 바와 같이, 논리 세트 수 입력(414)이 SwapReg 값과 동일하면, 그것은, 이 논리 세트가 정확하게, 이 로테이션에서 스와핑되어야 하는 캐시 세트라는 것을 의미한다. 따라서, 물리 세

트는 ShiftReg의 현재 시프트 값에 맵핑되며, 물리 세트 수 출력(418)으로서 출력된다.

- [0043] [0050] 둘째, 로직 박스(416)에 또한 도시된 바와 같이, 논리 세트 수 입력(414)이 SwapReg 값보다 크면, 그것은 이 캐시 세트가 이 로테이션에서 시프트되지 않고, 마지막 로테이션과 동일한 맵핑을 유지하는 것을 의미한다. 따라서, 물리 세트는 LS + ShiftReg에 맵핑된다. 일 구현에서, 맵핑은 (LS + ShiftReg) 값을 취하고, 캐시 세트들의 수인 N을 이용하여 modulo 연산을 수행하며, 그 다음, 물리 세트에 결과적 값을 할당함으로써 이루어진다.
- [0044] [0051] 셋째, 로직 박스(416)에 또한 도시된 바와 같이, 논리 세트 수 입력(414)이 SwapReg 값보다 작으면(else 절), 그것은 이 캐시 세트가 이 로테이션에서 시프트되는 것을 의미한다. 따라서, 물리 세트는 LS + ShiftReg + 1에 맵핑된다. 일 구현에서, 맵핑은 (LS + ShiftReg + 1) 값을 취하고, 캐시 세트들의 수인 N을 이용하여 연산을 수행하며, 그 다음 물리 세트에 결과적 값을 할당함으로써 이루어진다.
- [0045] [0052] 위에서 논의된 연산들과 유사한 3가지 연산들은 로직 박스(422)에서 발생하나, 논리 세트 수 출력(420)을 컴퓨팅하기 위해 물리 세트 입력 수(424)에 대해 발생한다.
- [0046] [0053] 캐시 라인이 하위 레벨 메모리로 다시 기록될 필요가 있을 때, 논리 세트 어드레스는 재생성된다. 물리 세트로부터 논리 세트로의 맵핑이 대칭적이다. 이 맵핑 방식은 또한, 도 2에 나타낼 수 있는 바와 같이 검증될 수 있다. SwapReg 및 ShiftReg가 증가하는 기록 카운트들과 함께 변경되기 때문에, 논리 세트와 물리 세트 간의 맵핑은 항상 변화하고 있고, 이는 상이한 물리 세트들로의 기록들이 균형적인 것을 보장하여, 기록 변화들을 감소시킨다.
- [0047] [0054] 종래의 캐시 아키텍처들과 비교하여, 시프트-스왑 웨어-레벨링 방식에서의 세트 인덱스 변형(translation)은 단지 단순한 산술적 연산을 부가하며, 행-디코더로 머지(merge)될 수 있다. 또한, 이 하나의-사이클 레이턴시 오버헤드는 단지, 하위-레벨 캐시들에 액세스하는 상위-레벨 캐시 미스들에 대해 지불된다.
- [0048] [0055] 도 5는 본 개시의 양상에 따라 웨어-레벨링하기 위한 방법을 예시하는 프로세스 흐름도(500)이다. 블록(502)에서, 메모리 기록 연산들의 수는 모든 각각의 메모리 기록 연산에 대해 글로벌 카운터를 1만큼 증분시킴으로써 글로벌 카운터에 의해 카운트된다. 블록(504)에서, 글로벌 카운터가 미리 결정된 임계 값과 동일할 때, 스와핑 연산이 수행되고, 스왑 카운터가 1만큼 증분된다. 또한, 스왑 연산을 수행하는 것은 스와핑된 캐시 세트들을 리맵핑하는 것을 포함하는데, 이는 차례로, 기록 백 버퍼에 2개의 스와핑된 캐시 세트들 내의 더티 데이터를 배치하는 것 및 2개의 스와핑된 캐시 세트들 내의 다른 데이터를 무효화하는 것을 포함한다.
- [0049] [0056] 블록(506)에서, 스왑 카운터가 N-1과 동일할 때, 시프트 카운터는 1만큼 증분되고, 스왑 카운터는 0으로 다시 세팅된다. N은 캐시 메모리 내의 캐시 세트들의 수이다. 블록(508)에서, 시프트 카운터가 N과 동일할 때, 시프트 카운터는 0으로 다시 세팅된다. 블록(510)에서, 입력 캐시 세트 수는 출력 캐시 세트 수로 변환된다. 입력 또는 출력 캐시 세트 수는 논리 세트 수 또는 물리 세트 수일 수 있다.
- [0050] [0057] 도 6은 본 개시의 양상에 따라 웨어-레벨링하기 위한 방법을 예시하는 프로세스 흐름도(600)이다. 블록(610)에서, 캐시 메모리로의 메모리 기록 연산들의 수(예를 들어, numWrites)가 미리 결정된 임계치에 도달하였는지 여부가 결정된다. 만약 그렇다면, 블록(612)에서, 메모리 위치 스와핑 연산이 2개의 캐시 세트들에 대해 수행된다. 블록(614)에서, 2개의 스와핑된 캐시 세트들 중 하나로부터의 콘텐츠들이 클리어링된다. 블록(616)에서, 스와핑된 2개의 캐시 세트들 중 다른 하나의 캐시 세트의 메모리 콘텐츠들은 그대로 남겨진다. 블록(618)에서, 캐시 세트의 스와핑이 추적된다. 블록(620)에서, 논리 캐시 세트 수는 물리 캐시 세트 수로 변환된다. 기록 연산들의 수가 임계치에 도달되지 않으면, 프로세스는 블록(610)에서 유지된다.
- [0051] [0058] 스왑-시프트 웨어-레벨링 방식에서, 세트-간 기록 변화 감소는 실험적 시간 동안의 시프트 로테이션들의 수에 관련된다. 캐시 내에 N개의 세트들이 존재한다고 가정하면, 하나의 시프트 로테이션은 N회의 스왑 로테이션들을 포함하고, 스왑-시프트 방식에서의 하나의 스왑 로테이션은 N-1회의 스와핑들을 필요로 한다. 각각의 시프트 로테이션 이후에, 모든 캐시 세트들은 N 단계들만큼 시프트되고, 논리 세트 인덱스들은 이들의 원래 포지션들에 맵핑된다. 따라서, 캐시가 시프트되는 라운드들이 많아질수록, 기록 액세스들이 더 균등하게 각각의 캐시 세트에 분배된다.
- [0052] [0059] 본 개시의 추가 양상에 따라, 캐시 메모리의 웨어-레벨링을 위한 캐시 제어기가 설명된다. 캐시 제어기는 캐시 메모리에의 메모리 기록 연산들의 수가 임계 값에 도달할 때, 캐시 세트들에 대해 복수의 메모리 위치 스와핑 연산들을 수행함으로써 캐시 메모리의 캐시 세트들을 동적으로 로테이션시키기 위한 수단을 포함한다. 각각의 스와핑 연산은 단지 스와핑된 캐시 세트들로부터의 콘텐츠들을 클리어하는 반면, 다른 캐시 세트들의 메

모리 콘텐츠를 그대로 남겨두는 것을 포함할 수 있다. 동적으로 로테이션시키는 수단은 캐시 제어기(140)일 수 있다. 장치는, 논리 캐시 세트 수를 물리 캐시 세트 수로 변환하기 위해 스와핑된 캐시 세트들을 추적하기 위한 수단을 더 포함한다. 추적 수단은 캐시 제어기(140), 글로벌 카운터(142), 스왑 레지스터(144) 및/또는 시프트 레지스터(146)일 수 있다. 또 다른 양상에서, 전술된 수단은 전술된 수단에 의해 기술되는 기능들을 수행하도록 구성되는 임의의 모듈 또는 임의의 장치일 수 있다.

[0053] [0060] 도 7은 본 개시의 양상이 유리하게 이용될 수 있는 예시적 무선 통신 시스템(700)을 도시하는 블록도이다. 예시를 목적으로, 도 7은 3개의 원격 유닛들(720, 730 및 750) 및 2개의 기지국들(740)을 도시한다. 무선 통신 시스템들이 많은 더 원격 유닛들 및 기지국들을 가질 수 있다는 것이 인식될 것이다. 원격 유닛들(720, 730 및 750)은 개시된 캐시 메모리를 포함하는 IC 디바이스들(725A, 725C 및 725B)을 포함한다. 다른 디바이스들이 또한, 기지국들, 스위칭 디바이스들 및 네트워크 장비와 같이 개시된 캐시 메모리를 포함할 수 있다는 것이 인식될 것이다. 도 7은 기지국(740)으로부터 원격 유닛들(720, 730 및 750)로의 순방향 링크 신호들(780) 및 원격 유닛들(720, 730 및 750)로부터 기지국들(740)로의 역방향 링크 신호들(790)을 도시한다.

[0054] [0061] 도 7에서, 원격 유닛(720)은 모바일 전화로서 도시되고, 원격 유닛(730)은 휴대용 컴퓨터로서 도시되며, 원격 유닛(750)은 무선 로컬 루프 시스템에서 고정 위치 원격 유닛으로서 도시된다. 예를 들어, 원격 유닛들은 모바일 폰들, 핸드-헬드 PCS(personal communication systems) 유닛들, 휴대용 데이터 유닛들, 이를테면, 개인용 데이터 보조기들, GPS(global positioning system) 인에이블 디바이스들, 네비게이션 디바이스들, 셋탑 박스들, 뮤직 플레이어들, 비디오 플레이어들, 엔터테인먼트 유닛들, 고정 위치 데이터 유닛들, 이를테면, 미터 판독 장비, 또는 데이터 또는 컴퓨터 명령들, 또는 이들의 결합들을 저장 또는 리트리브하는 다른 디바이스들일 수 있다. 도 7은 본 개시의 양상들에 따른 원격 유닛들을 예시하지만, 본 개시는 이 예시적 예시된 유닛들에 제한되는 것은 아니다. 본 개시의 양상들은 많은 디바이스들에서 적합하게 이용될 수 있는데, 이는 개시된 캐시 메모리를 포함한다.

[0055] [0062] 도 8은 회로, 레이아웃 및 위에서 개시된 캐시 메모리와 같은 반도체 컴포넌트의 로직 설계에 대해 이용되는 설계 워크스테이션을 예시하는 블록도이다. 설계 워크스테이션(800)은 운영 시스템 소프트웨어, 지원 파일들 및 Cadence 또는 OrCAD와 같은 설계 소프트웨어를 포함하는 하드 디스크(801)를 포함한다. 설계 워크스테이션(800)은 또한, 회로(810) 또는 캐시 메모리와 같은 반도체 컴포넌트(812)의 설계를 가능하게 하기 위한 디스플레이(802)를 포함한다. 저장 매체(804)는 회로 설계(810) 또는 반도체 컴포넌트(812)를 유형적으로 저장하기 위해 제공된다. 회로 설계(810) 또는 반도체 컴포넌트(812)는 GDSII 또는 GERBER와 같은 파일 포맷으로 저장 매체(804) 상에 저장될 수 있다. 저장 매체(804)는 CD-ROM, DVD(digital versatile disc), 하드 디스크, 플래시 메모리 또는 다른 적절한 디바이스일 수 있다. 게다가, 설계 워크스테이션(800)은 저장 매체(804)로부터의 입력을 수용하거나, 저장 매체(804)에 출력을 기록하기 위한 드라이브 장치(803)를 포함한다.

[0056] [0063] 저장 매체(804) 상에 레코딩된 데이터는 로직 회로 구성들, 포토리소그래피 마스크들에 대한 패턴 데이터 또는 전자 빔 리소그래피와 같은 직렬 기록 툴들에 대한 마스크 패턴 데이터를 특정할 수 있다. 데이터는 로직 검증 데이터, 이를테면, 로직 시뮬레이션들과 연관된 타이밍도들 또는 넷 회로들을 더 포함할 수 있다. 저장 매체(804) 상에서 데이터를 제공하는 것은 반도체 웨이퍼들을 설계하기 위한 프로세스들의 수를 감소시킴으로써 회로 설계(810) 또는 반도체 컴포넌트(812)의 설계를 가능하게 한다.

[0057] [0064] 펌웨어 및/또는 소프트웨어 구현에 있어서, 방법들은 본원에 설명된 기능들을 수행하는 모듈들(예를 들어, 프로시저들, 함수들 등)로 구현될 수 있다. 명령들을 유형으로 구현하는 기계 판독가능한 매체는 본원에 설명된 방법들을 구현하는데 이용될 수 있다. 예를 들어, 소프트웨어 코드들은 메모리에 저장되고, 프로세서 유닛에 의해 실행될 수 있다. 메모리는 프로세서 유닛 내부에서 또는 프로세서 유닛 외부에서 구현될 수 있다. 본원에 이용되는 바와 같이, "메모리"라는 용어는 롬 텀, 쇼트 텀, 휘발성, 비휘발성 또는 다른 메모리의 타입들을 지칭하며, 특정 타입의 메모리 또는 특정 수의 메모리들 또는 메모리가 저장되는 임의의 타입의 매체들에 제한되는 것은 아니다.

[0058] [0065] 펌웨어 및/또는 소프트웨어로 구현되는 경우, 기능들은 컴퓨터 판독가능한 매체 상에 하나 또는 둘 이상의 명령들 또는 코드로서 저장될 수 있다. 예들은 데이터 구조를 이용하여 인코딩되는 컴퓨터 판독가능한 매체들 및 컴퓨터 프로그램을 이용하여 인코딩되는 컴퓨터 판독가능한 매체들을 포함한다. 컴퓨터 판독가능한 매체들은 물리적 컴퓨터 저장 매체들을 포함한다. 저장 매체는 컴퓨터에 의해 액세스될 수 있는 임의의 이용가능한 매체일 수 있다. 제한이 아닌 예로서, 이러한 컴퓨터 판독가능한 매체들은 RAM, ROM, EEPROM, CD-ROM 또는 다른 광 디스크 스토리지, 자기 디스크 스토리지 또는 다른 자기 저장 디바이스들, 또는 원하는 프로그램 코드를

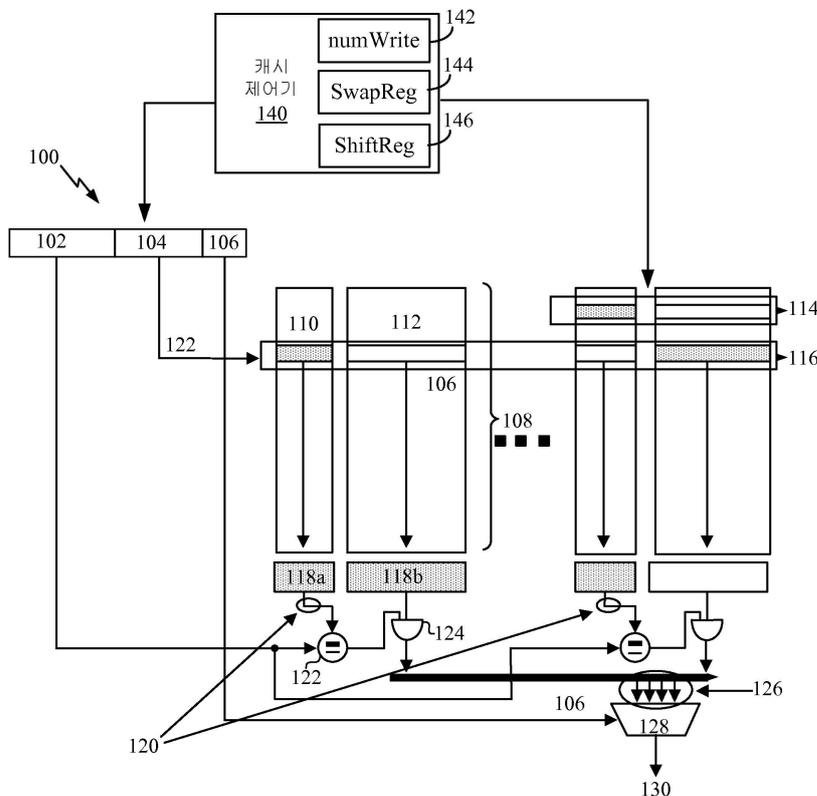
명령들 또는 데이터 구조들의 형태로 저장하는데 이용될 수 있고, 컴퓨터에 의해 액세스될 수 있는 다른 매체를 포함할 수 있고; 본원에서 이용되는 바와 같은 디스크(disk 및 disc)는 CD(compact disc), 레이저 디스크(disc), 광 디스크(disc), DVD(digital versatile disc), 플로피 디스크(disk) 및 블루-레이 디스크(disc)를 포함하며, 여기서 디스크(disk)들은 통상적으로 데이터를 자기적으로 재생하는 반면, 디스크(disc)들은 레이저들을 이용하여 데이터를 광학적으로 재생한다. 위의 것들의 결합들은 또한, 컴퓨터 판독가능한 매체들의 범위 내에 포함되어야 한다.

[0059] [0066] 컴퓨터 판독가능한 매체 상에서의 저장과 더불어, 명령들 및/또는 데이터는 통신 장치에 포함되는 송신 매체들 상에 신호들로서 제공될 수 있다. 예를 들어, 통신 장치는 명령들 및 데이터를 표시하는 신호들을 가지는 트랜시버를 포함할 수 있다. 명령들 및 데이터는 하나 또는 둘 이상의 프로세서로 하여금 청구항들에서 약속되는 기능들을 구현하게 하도록 구성된다.

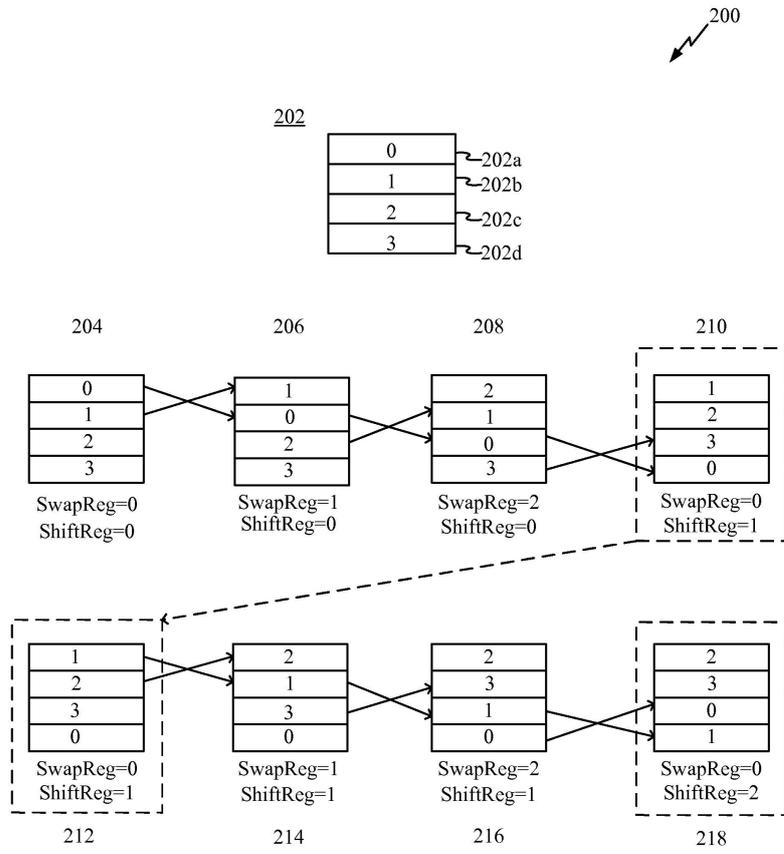
[0060] [0067] 본 개시 및 그 이점들은 상세하게 설명되었지만, 다양한 변화들, 치환들 및 변경들이 첨부된 청구항들에 의해 정의되는 바와 같은 본 개시의 기술로부터 벗어나지 않으면서 본원에서 이루어질 수 있다는 것이 이해되어야 한다. 예를 들어, "위" 및 "아래"와 같은 상관적 용어들은 기관 또는 전자 디바이스에 대해 이용된다. 물론, 기관 또는 전자 디바이스가 반전(invert)되면, 위는 아래가 되고, 아래는 위가 된다. 추가적으로, 측면으로 지향되면, 위 및 아래는 기관 또는 전자 디바이스의 면(side)들을 지칭할 수 있다. 더욱이, 본 출원의 범위는 본 명세서에서 설명된 프로세스, 기계, 제조품, 물질의 조성물들, 수단, 방법들 및 단계들에 대한 특정 구성들에 제한되는 것으로 의도되지 않는다. 당업자가 본 개시로부터 쉽게 인식할 바와 같이, 본원에 설명된 대응하는 구성들과 동일한 기능을 실질적으로 수행하거나, 본원에 설명된 대응하는 구성들과 동일한 결과를 실질적으로 달성하는 현재 존재하거나 추후에 개발될 프로세스들, 기계들, 제조품, 물질의 조성물들, 수단, 방법들 또는 단계들은 본 개시에 따라 이용될 수 있다. 따라서, 첨부된 청구항들은 그 범위 내에 이러한 프로세스들, 기계들, 제조품, 대상의 구성품, 수단, 방법들 또는 단계들을 포함하는 것으로 의도된다.

도면

도면1

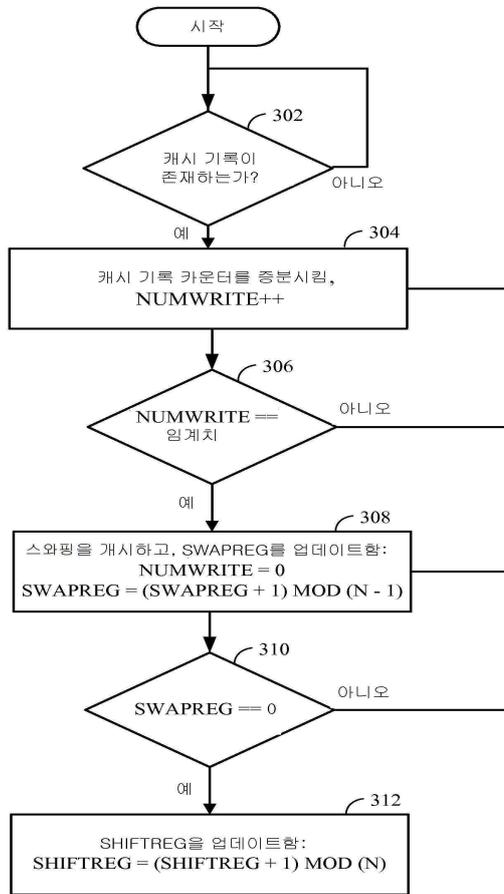


도면2

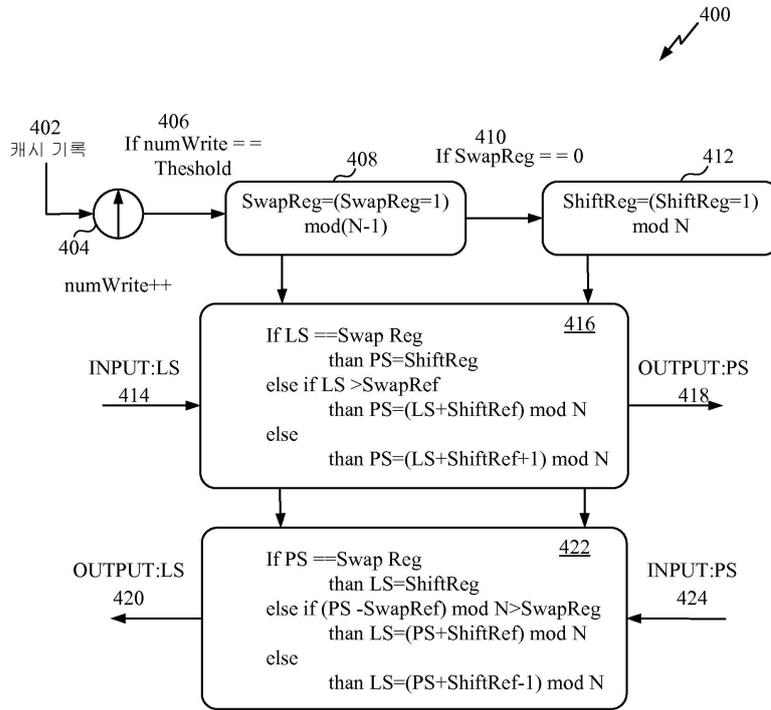


도면3

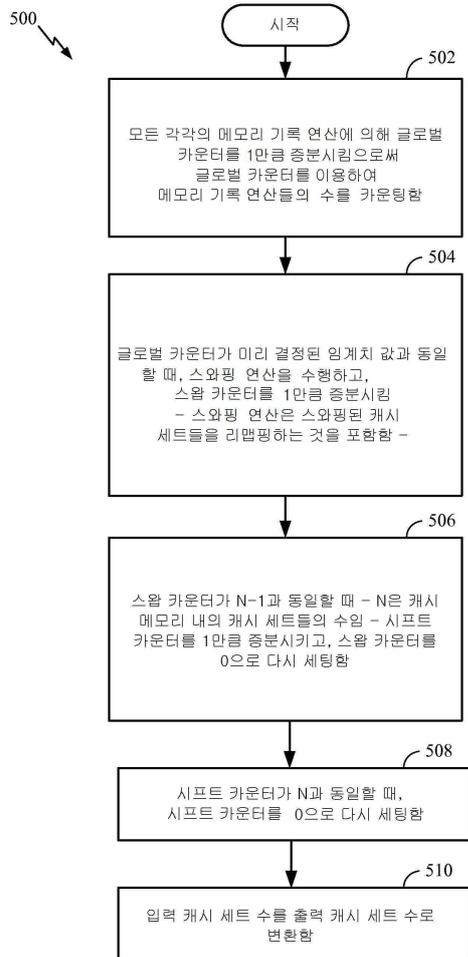
300 ↘



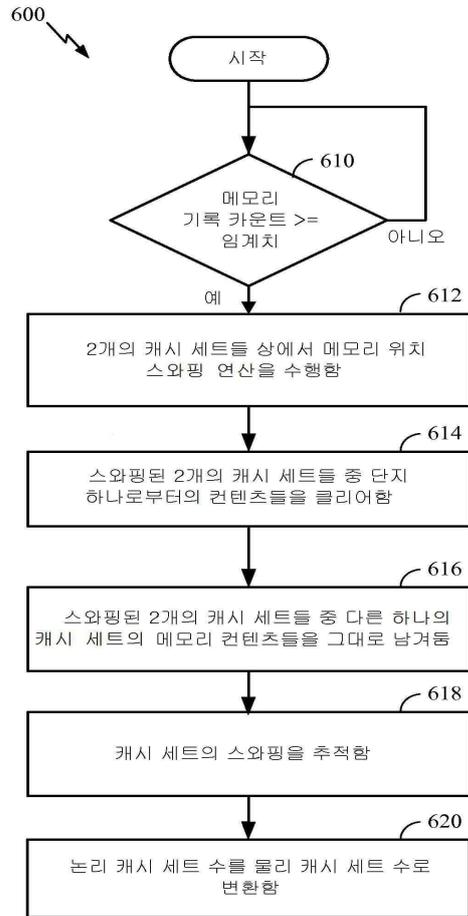
도면4



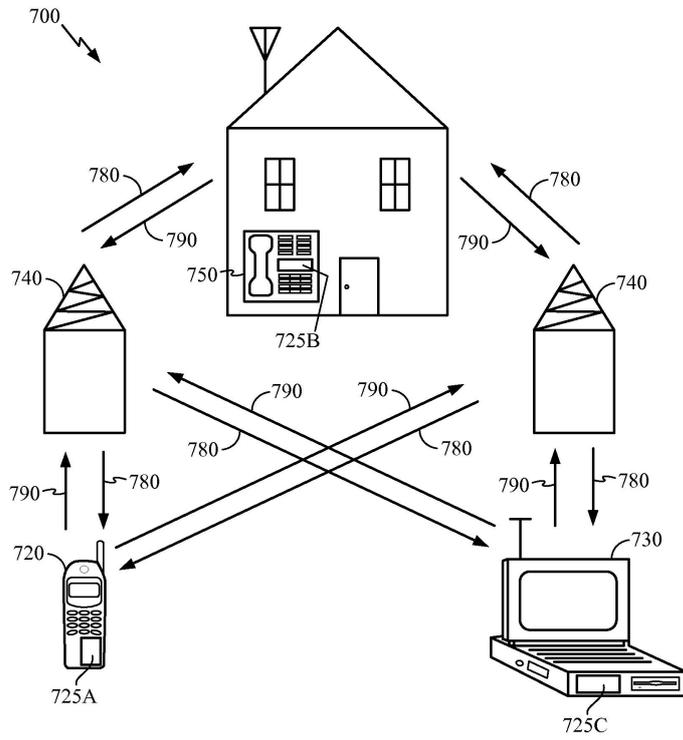
도면5



도면6



도면7



도면8

