



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0109286
(43) 공개일자 2024년07월10일

- | | |
|---|--|
| <p>(51) 국제특허분류(Int. Cl.)
 <i>G06F 12/109</i> (2016.01) <i>G06F 12/1036</i> (2016.01)
 <i>G06F 12/14</i> (2006.01) <i>G06F 21/53</i> (2013.01)
 <i>G06F 21/74</i> (2013.01) <i>G06F 9/455</i> (2018.01)</p> <p>(52) CPC특허분류
 <i>G06F 12/109</i> (2013.01)
 <i>G06F 12/1036</i> (2013.01)</p> <p>(21) 출원번호 10-2024-7021069
 (22) 출원일자(국제) 2022년04월28일
 심사청구일자 없음
 (85) 번역문제출일자 2024년06월24일
 (86) 국제출원번호 PCT/GB2022/051073
 (87) 국제공개번호 WO 2023/099860
 국제공개일자 2023년06월08일
 (30) 우선권주장
 2117274.7 2021년11월30일 영국(GB)</p> | <p>(71) 출원인
 에이알엠 리미티드
 영국 캠브리지 씨비1 9엔제이 체리턴 폴번로드 110</p> <p>(72) 발명자
 그리센스화이트, 리처드 로이
 영국 씨비1 9엔제이 케임브리지셔 케임브리지 폴본 로드 110 씨/오 에이알엠 리미티드</p> <p>(74) 대리인
 남호현</p> |
|---|--|

전체 청구항 수 : 총 25 항

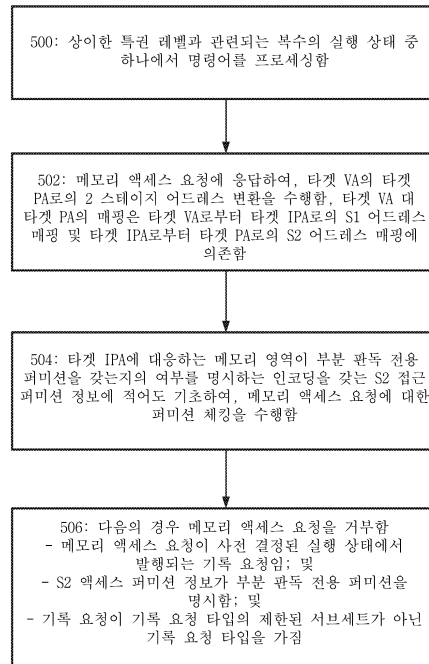
(54) 발명의 명칭 2 스테이지 어드레스 변환

(57) 요약

메모리 관리 회로부(28)는 스테이지 1 및 스테이지 2 변환 테이블 구조에 기초하는 2 스테이지 어드레스 변환을 지원한다. 스테이지 2 변환 테이블 엔트리에 의해 명시되는 스테이지 2 액세스 퍼미션 정보는, 프로세싱 회로부(4)가 사전 결정된 실행 상태에 있을 때 발행되는, 타겟 중간 어드레스에 대응하는 메모리 영역에 대한 기록 요

(뒷면에 계속)

대표도 - 도10



청이 기록 요청 타입의 제한된 서브세트(변환 테이블 엔트리의 액세스 추적 메타데이터를 업데이트하기 위한 메타데이터 업데이트 기록 요청을 포함함)에 대해 허락되지만 그러나 다른 기록 요청 타입에 대해서는 금지된다는 것을 나타내는 부분 판독 전용 퍼미션을 대응하는 메모리 영역이 갖는지를 명시하는 인코딩을 갖는다. 메모리 관리 회로부(28)는, 대응하는 스테이지 2 변환 테이블 엔트리의 스테이지 2 액세스 퍼미션 정보가 부분 판독 전용 퍼미션을 명시하고 메모리 액세스 요청이 사전 결정된 실행 상태에서 발행되는, 기록 요청 타입의 제한된 서브세트가 아닌, 기록 요청인 경우 메모리 액세스 요청을 거부한다.

(52) CPC특허분류

G06F 12/1441 (2013.01)
G06F 12/1475 (2013.01)
G06F 12/1483 (2013.01)
G06F 12/1491 (2013.01)
G06F 21/53 (2013.01)
G06F 21/74 (2013.01)
G06F 9/45558 (2013.01)
G06F 2009/45583 (2019.08)
G06F 2212/657 (2013.01)

명세서

청구범위

청구항 1

장치로서,

상기한 레벨의 특권과 관련되는 복수의 실행 상태 중 하나에서 명령어를 프로세싱하는 프로세싱 회로부; 및

메모리 액세스 요청에 의해 명시되는 타겟 가상 어드레스를 상기 메모리 액세스 요청에 응답하여 액세스될 메모리 시스템 위치를 나타내는 타겟 물리적 어드레스로 변환하는 메모리 관리 회로부를 포함하고; 상기 메모리 관리 회로부는 스테이지 1 변환 테이블 엔트리를 포함하는 스테이지 1 변환 테이블 구조 및 스테이지 2 변환 테이블 엔트리를 포함하는 스테이지 2 변환 테이블 구조에 기초하여 2 스테이지 어드레스 변환(two-stage address translation)을 지원하도록 구성되고, 상기 타겟 가상 어드레스로부터 상기 타겟 물리적 어드레스로의 매핑은 스테이지 1 어드레스 매핑 및 스테이지 2 어드레스 매핑에 의존하고, 상기 스테이지 1 어드레스 매핑은 상기 타겟 가상 어드레스로부터 상기 타겟 가상 어드레스에 대응하는 대응 스테이지 1 변환 테이블 엔트리에 의해 명시되는 타겟 중간 어드레스로의 매핑을 포함하고, 그리고 상기 스테이지 2 어드레스 매핑은 상기 타겟 중간 어드레스로부터 상기 타겟 중간 어드레스에 대응하는 대응 스테이지 2 변환 테이블 엔트리에 의해 명시되는 상기 타겟 물리적 어드레스로의 매핑을 포함하고;

상기 메모리 관리 회로부는 상기 대응 스테이지 2 변환 테이블 엔트리에 의해 명시되는 스테이지 2 액세스 퍼미션 정보에 적어도 기초하여 상기 메모리 액세스 요청에 대한 퍼미션 체크를 수행하도록 구성되고, 상기 스테이지 2 액세스 퍼미션 정보는, 상기 프로세싱 회로부가 상기 복수의 실행 상태 중 사전 결정된 실행 상태에 있을 때 발행되는, 상기 타겟 중간 어드레스에 대응하는 상기 메모리 영역에 대한 기록 요청이 기록 요청 타입의 제한된 서브세트에 대해 허락되지만 그러나 상기 제한된 서브세트 이외의 기록 요청 타입에 대해서는 금지된다는 것을 나타내는 부분 판독 전용 퍼미션(partially-read-only permission)을 상기 타겟 중간 어드레스에 대응하는 메모리 영역이 갖는지의 여부를 명시하는 인코딩을 가지며, 상기 기록 요청 타입의 제한된 서브세트는 변환 테이블 엔트리의 액세스 추적 메타데이터를 업데이트하기 위해 상기 메모리 관리 회로부에 의해 생성되는 메타데이터 업데이트 기록 요청을 포함하고; 그리고

상기 메모리 관리 회로부는 상기 메모리 액세스 요청이 상기 사전 결정된 실행 상태에서 발행되는 기록 요청이라는 것을 결정하는 것에 응답하여 상기 메모리 액세스 요청을 거부하도록 구성되고, 상기 스테이지 2 액세스 퍼미션 정보는 상기 타겟 중간 어드레스에 대응하는 상기 메모리 영역이 상기 부분 판독 전용 퍼미션을 갖는다는 것을 명시하고, 상기 메모리 액세스 요청은 상기 기록 요청 타입의 제한된 서브세트 이외의 기록 요청 타입인, 장치.

청구항 2

제1항에 있어서, 상기 기록 요청 타입의 제한된 서브세트는 상기 사전 결정된 실행 상태에서 상기 프로세싱 회로부에 의해 실행되는 적어도 하나의 사전 결정된 타입의 변환 테이블 엔트리 업데이트 명령어에 응답하여 발행되는 기록 요청을 또한 포함하는, 장치.

청구항 3

제1항 또는 제2항에 있어서, 상기 기록 요청 타입의 제한된 서브세트는 상기 사전 결정된 실행 상태에서 상기 프로세싱 회로부에 의해 실행되는 적어도 하나의 타입의 저장 명령어에 응답하여 발행되는 기록 요청을 배제하는, 장치.

청구항 4

제1항 내지 제3항 중 어느 한 항에 있어서, 각각의 스테이지 1 변환 테이블 엔트리는, 그 스테이지 1 변환 테이블 엔트리가, 보호된 엔트리로서 인코딩되지 않는 스테이지 1 변환 테이블 엔트리와 비교하여, 업데이트가 제한되지 않는 보호된 엔트리인지의 여부를 명시하는 인코딩을 갖는, 장치.

청구항 5

제4항에 있어서, 주어진 스테이지 1 변환 테이블 엔트리가 보호된 엔트리로서 마킹되는지의 여부에 기초하여, 상기 주어진 스테이지 1 변환 테이블 엔트리에 의해 명시되는 액세스 퍼미션 또는 어드레스 매핑에 대한 업데이트를 제한할지의 여부를 체크하는 보호 엔트리 체크 회로부(protected-entry-checking circuitry)를 포함하는, 장치.

청구항 6

제4항 또는 제5항에 있어서, 상기 프로세싱 회로부가 주어진 메모리 시스템 위치에 대한 업데이트를 요청하기 위한 보호 엔트리 체크 타입의 판독 체크 기록 명령어(read-check-write instruction)를 실행하는 것에 응답하여:

상기 주어진 메모리 시스템 위치로부터 데이터를 판독하고,

상기 주어진 메모리 시스템 위치로부터 판독되는 상기 데이터가 보호된 엔트리로서 명시되는 스테이지 1 변환 테이블 엔트리와 일치하는 값을 갖는지의 여부를 체크하고, 그리고

상기 주어진 메모리 시스템 위치로부터 판독되는 상기 데이터가 보호된 엔트리로서 명시되는 스테이지 1 변환 테이블 엔트리와 일치하는 값을 갖는다는 것을 결정하는 것에 응답하여, 상기 스테이지 1 변환 테이블 엔트리의 비트의 업데이트 제한 서브세트가 상기 판독 체크 기록 명령어에 응답하여 업데이트되는 것을 방지하는

보호 엔트리 체크 회로부를 포함하는, 장치.

청구항 7

제6항에 있어서, 상기 보호 엔트리 체크 타입의 판독 체크 기록 명령어에 응답하여 상기 주어진 메모리 시스템 위치에 대한 상기 업데이트가 수행되는 경우, 상기 보호 엔트리 체크 회로부는 상기 주어진 메모리 시스템 위치로부터의 상기 데이터의 상기 판독, 상기 데이터가 보호된 엔트리로서 명시되는 스테이지 1 변환 테이블 엔트리와 일치하는 값을 갖는지의 여부를 상기 체크, 및 원자성 동작(atomic operation)으로서 수행될 상기 주어진 메모리 시스템 위치의 상기 업데이트를 제어하도록 구성되는, 장치.

청구항 8

제6항 또는 제7항에 있어서, 상기 프로세싱 회로부가 상기 사전 결정된 상태에 있을 때 실행되는 상기 보호 엔트리 체크 타입의 판독 체크 기록 명령어에 응답하여 상기 주어진 메모리 시스템 위치에 대한 상기 업데이트가 수행되는 경우, 상기 보호 엔트리 체크 회로부는 상기 주어진 메모리 시스템 위치에 대한 업데이트가, 상기 주어진 메모리 시스템 위치에 저장되는 스테이지 1 변환 테이블 엔트리가 보호된 엔트리로서 명시되는지의 여부를 변경하는 것을 금지하도록 구성되는, 장치.

청구항 9

제6항 내지 제8항 중 어느 한 항에 있어서, 상기 보호 엔트리 체크 타입의 판독 체크 기록 명령어에 응답하여 발행되는 기록 요청은 상기 기록 요청 타입의 제한된 서브세트 중 하나인, 장치.

청구항 10

제6항 내지 제9항 중 어느 한 항에 있어서, 상기 비트의 업데이트 제한 서브세트는:

상기 스테이지 1 변환 테이블 엔트리의 모든 비트; 및

상기 스테이지 1 변환 테이블 엔트리가 보호된 엔트리로서 명시되는지의 여부를 인코딩하기 위해 사용되는 하나 이상의 비트를 포함하는, 상기 스테이지 1 변환 테이블 엔트리의 비트의 적절한 서브세트

중 하나를 포함하고,

상기 비트의 적절한 서브세트에 어떤 비트가 있는지의 선택은: 제어 레지스터에 저장되는 소프트웨어 프로그래밍 가능 선택 값에 따라,

프로그래밍 불가능한 것; 및

프로그래밍 가능한 것
중 하나인, 장치.

청구항 11

제6항 내지 제10항 중 어느 한 항에 있어서,

각각의 유효한 스테이지 1 변환 테이블 엔트리는 그 스테이지 1 변환 테이블 엔트리가 보호된 엔트리인지의 여부를 명시하는 인코딩을 가지며; 그리고

상기 주어진 메모리 시스템 위치로부터 판독되는 상기 데이터가 무효한 스테이지 1 변환 테이블 엔트리와 일치하는 값을 갖는다는 것을 결정하는 것에 응답하여, 상기 보호 엔트리 체크 회로부는, 상기 주어진 메모리 시스템 위치에서의 상기 데이터로 하여금 보호된 엔트리로서 명시되는 유효한 스테이지 1 변환 테이블 엔트리가 되게 하는 업데이트가 아니라, 상기 무효한 스테이지 1 변환 테이블 엔트리의 업데이트를 허락하도록 구성되는, 장치.

청구항 12

제6항 내지 제10항 중 어느 한 항에 있어서, 유효한 및 무효한 스테이지 1 변환 테이블 엔트리 둘 모두는 그 스테이지 1 변환 테이블 엔트리가 보호된 엔트리인지의 여부를 명시하는 인코딩을 가지며;

상기 주어진 메모리 시스템으로부터 판독되는 상기 데이터가 보호된 엔트리로서 명시되는 무효한 스테이지 1 변환 테이블 엔트리와 일치하는 값을 갖는다는 것을 결정하는 것에 응답하여, 상기 보호 엔트리 체크 회로부는 상기 주어진 메모리 시스템 위치의 비트의 무효 엔트리 업데이트 제한 서브세트가 상기 판독 체크 기록 명령어에 응답하여 업데이트되는 것을 방지하도록 구성되고; 그리고

상기 주어진 메모리 시스템으로부터 판독되는 상기 데이터가 보호된 엔트리로서 명시되는 유효한 스테이지 1 변환 테이블 엔트리와 일치하는 값을 갖는다는 것을 결정하는 것에 응답하여, 상기 보호 엔트리 체크 회로부는 상기 주어진 메모리 시스템 위치의 비트의 유효 엔트리 업데이트 제한 서브세트가 상기 판독 체크 기록 명령어에 응답하여 업데이트되는 것을 방지하도록 구성되는, 장치.

청구항 13

제12항에 있어서,

유효한 스테이지 1 변환 테이블 엔트리는 그 유효한 스테이지 1 변환 테이블 엔트리가 보호된 엔트리인지의 여부를 나타내는 제1 비트필드를 포함하고; 그리고

무효한 스테이지 1 변환 테이블 엔트리는 그 무효한 스테이지 1 변환 테이블 엔트리가 보호된 엔트리인지의 여부를 나타내는 제2 비트필드를 포함하고; 그리고

상기 제1 비트필드 및 상기 제2 비트필드는 스테이지 1 변환 테이블 엔트리 인코딩 내에서 상이한 비트 포지션에 있는, 장치.

청구항 14

제6항 내지 제13항 중 어느 한 항에 있어서, 상기 보호 엔트리 체크 회로부는:

상기 요청된 업데이트가 상기 비트의 업데이트 제한 서브세트 중 적어도 하나에 대한 업데이트를 야기할 때 상기 주어진 메모리 시스템 위치의 상기 업데이트를 거부하는 것; 및

상기 비트의 업데이트 제한 서브세트가 상기 주어진 메모리 시스템 위치의 비트의 적절한 서브세트인 경우, 상기 판독 체크 기록 명령어에 응답하여 상기 주어진 메모리 시스템 위치의 적어도 하나의 다른 비트의 업데이트를 허용하면서 상기 비트의 업데이트 제한 서브세트의 상기 업데이트를 억제하는 것

중 하나에 의해 상기 비트의 업데이트 제한 서브세트의 상기 업데이트를 억제하도록 구성되는, 장치.

청구항 15

제6항 내지 제14항 중 어느 한 항에 있어서, 상기 보호 엔트리 체크 타입의 판독 체크 기록 명령어에 응답하여,

상기 프로세싱 회로부는, 상기 주어진 메모리 시스템으로부터 판독되는 상기 데이터가 보호된 엔트리로서 명시되는 스테이지 1 변환 테이블 엔트리인지의 여부를 상기 체크에 기초하여 상기 보호 엔트리 체크 타입의 판독 체크 기록 명령어에 의해 요청되는 상기 업데이트의 적어도 일부가 억제되었는지의 여부를 나타내기 위해, 적어도 하나의 조건 상태 값을 설정하도록 구성되는, 장치.

청구항 16

제6항 내지 제15항 중 어느 한 항에 있어서, 상기 보호 엔트리 체크 타입의 판독 체크 기록 명령어는:

상기 보호 엔트리 체크 타입의 판독 체크 기록 명령어의 스왑 변형(swap variant) - 이 스왑 변형에 응답하여 상기 프로세싱 회로부는 상기 주어진 메모리 시스템 위치로부터 판독되는 상기 데이터를, 후속하는 명령어에 의한 피연산자로서의 사용을 위해, 목적지 레지스터에 기록하도록 구성됨 -;

상기 프로세싱 회로부가 상기 주어진 메모리 시스템 위치로부터 판독되는 상기 데이터를 목적지 레지스터에 기록하지 않고도 상기 저장 변형을 프로세싱하도록 구성되는, 상기 보호 엔트리 체크 타입의 판독 체크 기록 명령어의 저장 변형(store variant);

상기 보호 엔트리 체크 회로부가 상기 주어진 메모리 시스템 위치로부터 판독되는 상기 데이터 및 상기 비교 피연산자의 비교가 비교 조건을 충족하는지의 여부를 결정하도록, 상기 비교 조건이 충족되지 않는 경우 상기 주어진 메모리 시스템 위치에 대한 상기 업데이트를 억제하도록, 그리고 후속하는 명령어에 의한 피연산자로서의 사용을 위해 상기 주어진 메모리 시스템 위치로부터 판독되는 상기 데이터를 목적지 레지스터에 기록하도록 구성되는, 비교 피연산자를 명시하는 상기 보호 엔트리 체크 타입의 판독 체크 기록 명령어의 비교 및 스왑 변형(compare-and-swap variant); 및

상기 주어진 메모리 시스템 위치의 상기 업데이트가 상기 주어진 메모리 시스템 위치에서 상기 데이터의 적어도 하나의 선택된 비트를 설정하는 것 또는 클리어하는 것을 포함하는 비트 설정/클리어 명령어

중 하나인, 장치.

청구항 17

제5항 내지 제16항 중 어느 한 항에 있어서, 주어진 중간 어드레스에 대응하는 주어진 스테이지 2 액세스 변환 테이블 엔트리는, 상기 주어진 중간 어드레스에 대응하는 상기 메모리 영역에 대한 액세스가 허용되기 위해서는, 주어진 가상 어드레스로부터 상기 주어진 중간 어드레스로의 상기 스테이지 1 어드레스 매핑을 명시하는 상기 대응 스테이지 1 변환 테이블 엔트리를 찾기 위해 상기 스테이지 1 변환 구조의 변환 테이블 워크에서 액세스될 하나 이상의 워킹된(walked) 스테이지 1 변환 테이블 엔트리 각각이 안티 앨리어싱 조건을 충족할 것을 요구받는다는 요건을 나타내는 안티 앨리어싱 특성(property)을, 상기 주어진 중간 어드레스에 대응하는 상기 메모리 영역이 갖는지의 여부를 명시하는 인코딩을 가지며,

주어진 워킹된 스테이지 1 변환 테이블 엔트리에 대한 상기 안티 앨리어싱 조건은:

상기 주어진 워킹된 스테이지 1 변환 테이블 엔트리가 보호된 엔트리로서 명시되고 상기 주어진 워킹된 스테이지 1 변환 테이블 엔트리를 저장하는 상기 메모리 시스템 위치의 물리적 어드레스를 유도하기 위해 사용되는 상기 스테이지 2 어드레스 매핑을 제공하는 관련 스테이지 2 변환 테이블 엔트리에 대한 상기 스테이지 2 액세스 퍼미션 정보에 의해 상기 부분 판독 전용 퍼미션이 명시되는 경우; 또는

판독 전용 퍼미션이 상기 관련 스테이지 2 변환 테이블 엔트리에 대한 상기 스테이지 2 액세스 퍼미션 정보에 의해 명시되는 경우

중 어느 하나의 경우에 충족되고;

그리고

상기 메모리 관리 회로부는, 상기 타겟 중간 어드레스에 대응하는 상기 메모리 영역이 상기 안티 앨리어싱 특성을 갖는다는 것 및 상기 하나 이상의 워킹된 스테이지 1 변환 테이블 엔트리 중 임의의 것이 상기 안티 앨리어싱 조건을 충족하지 못한다는 것을 상기 대응 스테이지 2 변환 테이블 엔트리가 명시한다는 것을 결정하는 것에 응답하여, 상기 메모리 액세스 요청을 거부하도록 구성되는, 장치.

청구항 18

제1항 내지 제17항 중 어느 한 항에 있어서, 상기 메모리 관리 회로부는 스테이지 1 변환 테이블 기본 어드레스 레지스터의 값에 기초하여 상기 스테이지 1 변환 테이블 구조의 스테이지 1 최상위 레벨 변환 테이블의 스테이지 1 최상위 레벨 기본 어드레스를 결정하도록 구성되는, 장치.

청구항 19

제18항에 있어서, 스테이지 1 최상위 레벨 테이블 부분 판독 전용 체크가 인에이블되는 경우, 상기 메모리 관리 회로부는, 관련된 스테이지 2 변환 테이블 엔트리가 상기 부분 판독 전용 퍼미션을 명시하지 않는 중간 어드레스에 상기 스테이지 1 최상위 레벨 기본 어드레스가 대응한다는 것을 결정하는 것에 응답하여, 장애(fault)를 시그널링하도록 구성되는, 장치.

청구항 20

제19항에 있어서, 상기 메모리 관리 회로부는, 제어 레지스터에 저장되는 스테이지 1 최상위 레벨 부분 판독 전용 체크 인에이블 제어 값에 기초하여 상기 스테이지 1 최상위 레벨 테이블 부분 판독 전용 체크가 인에이블되는지의 여부를 결정하도록 구성되는, 장치.

청구항 21

제18항 내지 제20항 중 어느 한 항에 있어서,

스테이지 1 최상위 레벨 테이블 존재 퍼미션 체크가 인에이블되는 경우, 상기 메모리 관리 회로부는, 상기 스테이지 1 최상위 레벨 기본 어드레스에 대응하는 주어진 메모리 영역이, 상기 스테이지 1 변환 테이블 구조의 상기 스테이지 1 최상위 레벨 테이블을 저장하기 위해 상기 주어진 메모리 영역이 할당되는 것이 허용된다는 것을 나타내는 스테이지 1 최상위 레벨 테이블 존재 퍼미션과 관련되는지의 여부를 결정하도록, 그리고 상기 스테이지 1 최상위 레벨 기본 어드레스에 대응하는 상기 주어진 메모리 영역이 상기 스테이지 1 최상위 레벨 테이블 존재 퍼미션을 갖지 않는 경우 장애를 트리거하도록 구성되는, 장치.

청구항 22

제21항에 있어서, 상기 메모리 관리 회로부는, 상기 스테이지 1 최상위 레벨 기본 어드레스에 대응하는 관련된 스테이지 2 변환 테이블 엔트리의 인코딩에 기초하여, 상기 주어진 메모리 영역이 상기 스테이지 1 최상위 레벨 테이블 존재 퍼미션과 관련되는지의 여부를 결정하도록 구성되는, 장치.

청구항 23

제21항 또는 제22항에 있어서, 상기 메모리 관리 회로부는, 제어 레지스터에 저장되는 스테이지 1 최상위 레벨 테이블 존재 퍼미션 체크 인에이블 제어 값에 기초하여, 상기 스테이지 1 최상위 레벨 테이블 존재 퍼미션 체크가 인에이블되는지의 여부를 결정하도록 구성되는, 장치.

청구항 24

방법으로서,

상이한 레벨의 특권과 관련되는 복수의 실행 상태 중 하나에서 명령어를 프로세싱하는 단계; 및

메모리 액세스 요청에 응답하여 액세스될 메모리 시스템 위치를 나타내는 타겟 물리적 어드레스로의 상기 메모리 액세스 요청에 의해 명시되는 타겟 가상 어드레스의 2 스테이지 어드레스 변환을 수행하는 단계 - 상기 2 스테이지 어드레스 변환은 스테이지 1 변환 테이블 엔트리를 포함하는 스테이지 1 변환 테이블 구조 및 스테이지 2 변환 테이블 엔트리를 포함하는 스테이지 2 변환 테이블 구조에 기초하고, 상기 타겟 가상 어드레스로부터 상기 타겟 물리적 어드레스로의 매핑은 스테이지 1 어드레스 매핑 및 스테이지 2 어드레스 매핑에 의존하고, 상기 스테이지 1 어드레스 매핑은 상기 타겟 가상 어드레스로부터 상기 타겟 가상 어드레스에 대응하는 대응 스테이지 1 변환 테이블 엔트리에 의해 명시되는 타겟 중간 어드레스로의 매핑을 포함하고, 그리고 상기 스테이지 2 어드레스 매핑은 상기 타겟 중간 어드레스로부터 상기 타겟 중간 어드레스에 대응하는 대응 스테이지 2 변환 테이블 엔트리에 의해 명시되는 상기 타겟 물리적 어드레스로의 매핑을 포함함 -;

상기 대응 스테이지 2 변환 테이블 엔트리에 의해 명시되는 스테이지 2 액세스 퍼미션 정보에 적어도 기초하여 상기 메모리 액세스 요청에 대한 퍼미션 체크를 수행하는 단계 - 상기 스테이지 2 액세스 퍼미션 정보는, 상기

프로세싱 회로부가 상기 복수의 실행 상태 중 사전 결정된 실행 상태에 있을 때 발행되는, 상기 타겟 중간 어드레스에 대응하는 상기 메모리 영역에 대한 기록 요청이 기록 요청 타입의 제한된 서브세트에 대해 허락되지만 그러나 상기 제한된 서브세트 이외의 기록 요청 타입에 대해서는 금지된다는 것을 나타내는 부분 판독 전용 퍼미션을 상기 타겟 중간 어드레스에 대응하는 메모리 영역이 갖는지의 여부를 명시하는 인코딩을 가지며, 상기 기록 요청 타입의 제한된 서브세트는 변환 테이블 엔트리의 액세스 추적 메타데이터를 업데이트하기 위해 메모리 관리 회로부에 의해 생성되는 메타데이터 업데이트 기록 요청을 포함함 -; 및

상기 메모리 액세스 요청이 상기 사전 결정된 실행 상태에서 발행되는 기록 요청이라는 것을 결정하는 것에 응답하여 상기 메모리 액세스 요청을 거부하는 단계 - 상기 스테이지 2 액세스 퍼미션 정보는 상기 타겟 중간 어드레스에 대응하는 상기 메모리 영역이 상기 부분 판독 전용 퍼미션을 갖는다는 것을 명시하고, 상기 메모리 액세스 요청은 상기 기록 요청 타입의 제한된 서브세트 이외의 기록 요청 타입임 - 를 포함하는, 방법.

청구항 25

타겟 코드를 실행하기 위한 명령어 실행 환경을 제공하기 위해 호스트 데이터 프로세싱 장치를 제어하기 위한 명령어를 포함하는 컴퓨터 프로그램으로서,

상기한 레벨의 특권과 관련되는 복수의 실행 상태 중 하나에서 상기 타겟 코드의 명령어의 프로세싱을 시뮬레이션하는 프로세싱 프로그램 로직; 및

상기 타겟 코드에 의해 트리거되는 시뮬레이팅된 메모리 액세스 요청에 의해 명시되는 타겟 가상 어드레스를, 상기 메모리 액세스 요청에 응답하여 액세스될 시뮬레이팅된 메모리 시스템 위치를 나타내는 타겟 물리적 어드레스로 변환하는 메모리 관리 프로그램 로직을 포함하고; 상기 메모리 관리 프로그램 로직은 스테이지 1 변환 테이블 엔트리를 포함하는 스테이지 1 변환 테이블 구조 및 스테이지 2 변환 테이블 엔트리를 포함하는 스테이지 2 변환 테이블 구조에 기초하여 2 스테이지 어드레스 변환을 지원하도록 구성되고, 여기서 상기 타겟 가상 어드레스로부터 상기 타겟 물리적 어드레스로의 매핑은 스테이지 1 어드레스 매핑 및 스테이지 2 어드레스 매핑에 의존하고, 상기 스테이지 1 어드레스 매핑은 상기 타겟 가상 어드레스로부터 상기 타겟 가상 어드레스에 대응하는 대응 스테이지 1 변환 테이블 엔트리에 의해 명시되는 타겟 중간 어드레스로의 매핑을 포함하고, 그리고 상기 스테이지 2 어드레스 매핑은 상기 타겟 중간 어드레스로부터 상기 타겟 중간 어드레스에 대응하는 대응 스테이지 2 변환 테이블 엔트리에 의해 명시되는 상기 타겟 물리적 어드레스로의 매핑을 포함하고;

상기 메모리 관리 프로그램 로직은 상기 대응 스테이지 2 변환 테이블 엔트리에 의해 명시되는 스테이지 2 액세스 퍼미션 정보에 적어도 기초하여 상기 시뮬레이팅된 메모리 액세스 요청에 대한 퍼미션 체크를 수행하도록 구성되고, 상기 스테이지 2 액세스 퍼미션 정보는, 상기 프로세싱 프로그램 로직이 상기 복수의 실행 상태 중 사전 결정된 실행 상태에 있을 때 발행되는, 상기 타겟 중간 어드레스에 대응하는 상기 메모리 영역에 대한 기록 요청이 기록 요청 타입의 제한된 서브세트에 대해 허락되지만 그러나 상기 제한된 서브세트 이외의 기록 요청 타입에 대해서는 금지된다는 것을 나타내는 부분 판독 전용 퍼미션을 상기 타겟 중간 어드레스에 대응하는 메모리 영역이 갖는지의 여부를 명시하는 인코딩을 가지며, 상기 기록 요청 타입의 제한된 서브세트는 변환 테이블 엔트리의 액세스 추적 메타데이터를 업데이트하기 위해 상기 메모리 관리 프로그램 로직에 의해 생성되는 메타데이터 업데이트 기록 요청을 포함하고; 그리고

상기 메모리 관리 프로그램 로직은 상기 시뮬레이팅된 메모리 액세스 요청이 상기 사전 결정된 실행 상태에서 발행되는 기록 요청이라는 것을 결정하는 것에 응답하여 상기 시뮬레이팅된 메모리 액세스 요청을 거부하도록 구성되고, 상기 스테이지 2 액세스 퍼미션 정보는 상기 타겟 중간 어드레스에 대응하는 상기 메모리 영역이 상기 부분 판독 전용 퍼미션을 갖는다는 것을 명시하고, 상기 시뮬레이팅된 메모리 액세스 요청은 상기 기록 요청 타입의 제한된 서브세트 이외의 기록 요청 타입인, 명령어를 포함하는 컴퓨터 프로그램.

발명의 설명

기술 분야

[0001] 본 출원은 데이터 프로세싱의 분야에 관한 것이다. 더욱 상세하게는, 그것은 메모리 관리에 관한 것이다.

발명의 내용

[0002] 데이터 프로세싱 장치는 어드레스 변환을 수행하고, 변환 테이블에서 정의되는 어드레스 매핑 및 액세스 퍼미션

정보에 기초하여, 특정한 어드레스에 대한 액세스가 허락되는지의 여부를 제어하는 메모리 관리 회로부(circuitry)를 가질 수 있다. 메모리 관리 회로부는 2 스테이지 어드레스 변환(two-stage address translation)을 지원할 수 있는데, 여기서 가상 어드레스로부터 물리적 어드레스로의 변환은 가상 어드레스로부터 중간 어드레스로의 스테이지 1 매핑 및 중간 어드레스로부터 물리적 어드레스로의 스테이지 2 매핑에 의존한다. 이것은 다수의 게스트 오퍼레이팅 시스템이 동일한 하드웨어 플랫폼 상에서 공존할 수 있는 가상화를 지원하는 데 유용할 수 있는데, 스테이지 1 매핑은 주로 게스트 오퍼레이팅 시스템에 의해 제어되고 스테이지 2 매핑은 주로 하이퍼바이저에 의해 제어되며, 그 결과, 중간 어드레스(이것은 게스트 오퍼레이팅 시스템의 관점에서 물리적 메모리를 참조하는 것처럼 보임)는 상이한 물리적 어드레스로 다시 매핑될 수 있다. 2 스테이지 어드레스 변환은 상이한 게스트 오퍼레이팅 시스템에 의해 설정되는 충돌하는 중간 어드레스 사이의 충돌을 방지하는 데, 그리고 게스트 오퍼레이팅 시스템에 의해 보이는 하드웨어 리소스의 가상화된 모습(virtualised view)과 프로세싱 장치에서 제공되는 실제 하드웨어 리소스 사이를 매핑하는 데 유용할 수 있다.

[0003]

적어도 몇몇 예는 장치를 제공하는데, 그 장치는: 상이한 레벨의 특권과 관련되는 복수의 실행 상태 중 하나에서 명령어를 프로세싱하는 프로세싱 회로부; 및 메모리 액세스 요청에 의해 명시되는 타겟 가상 어드레스를 메모리 액세스 요청에 응답하여 액세스될 메모리 시스템 위치를 나타내는 타겟 물리적 어드레스로 변환하는 메모리 관리 회로부를 포함하고; 여기서: 메모리 관리 회로부는 스테이지 1 변환 테이블 엔트리를 포함하는 스테이지 1 변환 테이블 구조 및 스테이지 2 변환 테이블 엔트리를 포함하는 스테이지 2 변환 테이블 구조에 기초하여 2 스테이지 어드레스 변환을 지원하도록 구성되고, 여기서 타겟 가상 어드레스로부터 타겟 물리적 어드레스로의 매핑은 스테이지 1 어드레스 매핑 및 스테이지 2 어드레스 매핑에 의존하고, 스테이지 1 어드레스 매핑은 타겟 가상 어드레스로부터 타겟 가상 어드레스에 대응하는 대응 스테이지 1 변환 테이블 엔트리에 의해 명시되는 타겟 중간 어드레스로의 매핑을 포함하고, 그리고 스테이지 2 어드레스 매핑은 타겟 중간 어드레스로부터 타겟 중간 어드레스에 대응하는 대응 스테이지 2 변환; 메모리 관리 회로부는 대응하는 스테이지 2 변환 테이블 엔트리에 의해 명시되는 스테이지 2 액세스 퍼미션 정보에 적어도 기초하여 메모리 액세스 요청에 대한 퍼미션 체크를 수행하도록 구성되고, 스테이지 2 액세스 퍼미션 정보는, 프로세싱 회로부가 복수의 실행 상태 중 사전 결정된 실행 상태에 있을 때 발행되는, 타겟 중간 어드레스에 대응하는 메모리 영역에 대한 기록 요청이 기록 요청 타입의 제한된 서브세트에 대해 허락되지만 그러나 제한된 서브세트 이외의 기록 요청 타입에 대해서는 금지된다는 것을 나타내는 부분 판독 전용 퍼미션(partially-read-only permission)을 타겟 중간 어드레스에 대응하는 메모리 영역이 갖는지의 여부를 명시하는 인코딩을 가지며, 기록 요청 타입의 제한된 서브세트는 변환 테이블 엔트리의 액세스 추적 메타데이터를 업데이트하기 위해 메모리 관리 회로부에 의해 생성되는 메타데이터 업데이트 기록 요청을 포함하고; 그리고 메모리 관리 회로부는 메모리 액세스 요청이 사전 결정된 실행 상태에서 발행되는 기록 요청이라는 것을 결정하는 것에 응답하여 메모리 액세스 요청을 거부하도록 구성되고, 스테이지 2 액세스 퍼미션 정보는 타겟 중간 어드레스에 대응하는 메모리 영역이 부분 판독 전용 퍼미션을 갖는다는 것을 명시하고, 메모리 액세스 요청은 기록 요청 타입의 제한된 서브세트 이외의 기록 요청 타입이다.

[0004]

적어도 몇몇 예는 방법을 제공하는데, 그 방법은: 상이한 레벨의 특권과 관련되는 복수의 실행 상태 중 하나에서 명령어를 프로세싱하는 것; 및 메모리 액세스 요청에 응답하여 액세스될 메모리 시스템 위치를 나타내는 타겟 물리적 어드레스로의 메모리 액세스 요청에 의해 명시되는 타겟 가상 어드레스의 2 스테이지 어드레스 변환을 수행하는 것 - 2 스테이지 어드레스 변환은 스테이지 1 변환 테이블 엔트리를 포함하는 스테이지 1 변환 테이블 구조 및 스테이지 2 변환 테이블 엔트리를 포함하는 스테이지 2 변환 테이블 구조에 기초하고, 여기서 타겟 가상 어드레스로부터 타겟 물리적 어드레스로의 매핑은 스테이지 1 어드레스 매핑 및 스테이지 2 어드레스 매핑에 의존하고, 스테이지 1 어드레스 매핑은 타겟 가상 어드레스로부터 타겟 가상 어드레스에 대응하는 대응 스테이지 1 변환 테이블 엔트리에 의해 명시되는 타겟 중간 어드레스로의 매핑을 포함하고, 그리고 스테이지 2 어드레스 매핑은 타겟 중간 어드레스로부터 타겟 중간 어드레스에 대응하는 대응 스테이지 2 변환 테이블 엔트리에 의해 명시되는 타겟 물리적 어드레스로의 매핑을 포함함 -; 대응하는 스테이지 2 변환 테이블 엔트리에 의해 명시되는 스테이지 2 액세스 퍼미션 정보에 적어도 기초하여 메모리 액세스 요청에 대한 퍼미션 체크를 수행하는 것 - 스테이지 2 액세스 퍼미션 정보는, 프로세싱 회로부가 복수의 실행 상태 중 사전 결정된 실행 상태에 있을 때 발행되는, 타겟 중간 어드레스에 대응하는 메모리 영역에 대한 기록 요청이 기록 요청 타입의 제한된 서브세트에 대해 허락되지만 그러나 제한된 서브세트 이외의 기록 요청 타입에 대해서는 금지된다는 것을 나타내는 부분 판독 전용 퍼미션을 타겟 중간 어드레스에 대응하는 메모리 영역이 갖는지의 여부를 명시하는 인코딩을 가지며, 기록 요청 타입의 제한된 서브세트는 변환 테이블 엔트리의 액세스 추적 메타데이터를 업데이트하기 위해 메모리 관리 회로부에 의해 생성되는 메타데이터 업데이트 기록 요청을 포함함 -; 및 메모리 액세스 요청이 사전 결정된 실행 상태에서 발행되는 기록 요청이라는 것을 결정하는 것에 응답하여 메모리 액세스 요청을

거부하는 것 - 스테이지 2 액세스 퍼미션 정보는 타겟 중간 어드레스에 대응하는 메모리 영역이 부분 판독 전용 퍼미션을 갖는다는 것을 명시하고, 메모리 액세스 요청은 기록 요청 타입의 제한된 서브세트 이외의 기록 요청 타입임 - 을 포함한다.

[0005] 적어도 몇몇 예는 타겟 코드를 실행하기 위한 명령어 실행 환경을 제공하기 위해 호스트 데이터 프로세싱 장치를 제어하기 위한 명령어를 포함하는 컴퓨터 프로그램을 제공하는데, 그 컴퓨터 프로그램은: 상이한 레벨의 특권과 관련되는 복수의 실행 상태 중 하나에서 타겟 코드의 명령어의 프로세싱을 시뮬레이팅하는 프로세싱 프로그램 로직; 및 타겟 코드에 의해 트리거되는 시뮬레이팅된 메모리 액세스 요청에 의해 명시되는 타겟 가상 어드레스를, 메모리 액세스 요청에 응답하여 액세스될 시뮬레이팅된 메모리 시스템 위치를 나타내는 타겟 물리적 어드레스로 변환하는 메모리 관리 프로그램 로직을 포함하고; 여기서: 메모리 관리 프로그램 로직은 스테이지 1 변환 테이블 엔트리를 포함하는 스테이지 1 변환 테이블 구조 및 스테이지 2 변환 테이블 엔트리를 포함하는 스테이지 2 변환 테이블 구조에 기초하여 2 스테이지 어드레스 변환을 지원하도록 구성되고, 여기서 타겟 가상 어드레스로부터 타겟 물리적 어드레스로의 매핑은 스테이지 1 어드레스 매핑 및 스테이지 2 어드레스 매핑에 의존하고, 스테이지 1 어드레스 매핑은 타겟 가상 어드레스로부터 타겟 가상 어드레스에 대응하는 대응 스테이지 1 변환 테이블 엔트리에 의해 명시되는 타겟 중간 어드레스로의 매핑을 포함하고, 그리고 스테이지 2 어드레스 매핑은 타겟 중간 어드레스로부터 타겟 중간 어드레스에 대응하는 대응 스테이지 2 변환 테이블 엔트리에 의해 명시되는 타겟 물리적 어드레스로의 매핑을 포함하고; 메모리 관리 프로그램 로직은 대응하는 스테이지 2 변환 테이블 엔트리에 의해 명시되는 스테이지 2 액세스 퍼미션 정보에 적어도 기초하여 시뮬레이팅된 메모리 액세스 요청에 대한 퍼미션 체크를 수행하도록 구성되고, 스테이지 2 액세스 퍼미션 정보는, 프로세싱 프로그램 로직이 복수의 실행 상태 중 사전 결정된 실행 상태에 있을 때 발행되는, 타겟 중간 어드레스에 대응하는 메모리 영역에 대한 기록 요청이 기록 요청 타입의 제한된 서브세트에 대해 허락되지만 그러나 제한된 서브세트 이외의 기록 요청 타입에 대해서는 금지된다는 것을 나타내는 부분 판독 전용 퍼미션을 타겟 중간 어드레스에 대응하는 메모리 영역이 갖는지의 여부를 명시하는 인코딩을 가지며, 기록 요청 타입의 제한된 서브세트는 변환 테이블 엔트리의 액세스 추적 메타데이터를 업데이트하기 위해 메모리 관리 프로그램 로직에 의해 생성되는 메타데이터 업데이트 기록 요청을 포함하고; 그리고 메모리 관리 프로그램 로직은 시뮬레이팅된 메모리 액세스 요청이 사전 결정된 실행 상태에서 발행되는 기록 요청이라는 것을 결정하는 것에 응답하여 시뮬레이팅된 메모리 액세스 요청을 거부하도록 구성되고, 스테이지 2 액세스 퍼미션 정보는 타겟 중간 어드레스에 대응하는 메모리 영역이 부분 판독 전용 퍼미션을 갖는다는 것을 명시하고, 시뮬레이팅된 메모리 액세스 요청은 기록 요청 타입의 제한된 서브세트 이외의 기록 요청 타입이다.

[0006] 컴퓨터 프로그램은 컴퓨터 판독 가능 저장 매체 상에 저장될 수 있다. 기록 매체는 비일시적 저장 매체일 수도 있다.

도면의 간단한 설명

[0007] 본 기술의 추가적인 양태, 피쳐 및 이점은 첨부 도면과 연계하여 판독될 예의 다음의 설명으로부터 명백해질 것인데, 첨부 도면에서:

도 1은 프로세싱 회로부 및 메모리 관리 회로부를 갖는 데이터 프로세싱 장치의 예를 예시한다;

도 2는 프로세싱 회로부의 실행 상태의 예를 예시한다;

도 3은 2 스테이지 어드레스 변환을 예시한다;

도 4 및 도 5는, 각각, 스테이지 1 및 스테이지 2 어드레스 변환을 위한 변환 테이블 워크(translation table walk)의 예를 예시한다;

도 6은 스테이지 1에서 결정되는 각각의 테이블 기본 어드레스 및 최종 중간 어드레스가 각각의 스테이지 2 변환을 사용하여 물리적 어드레스로 변환되어야 한다는 것을 고려할 때, 스테이지 1 및 스테이지 2 변환 둘 모두의 전체 변환 테이블 워크에 대해 수행되는 테이블 룩업을 예시한다;

도 7은 메모리 관리를 제어하기 위한 메모리 관리 회로부 및 제어 레지스터의 예를 예시한다;

도 8은 스테이지 1 및 스테이지 2 어드레스 변환을 위한 변환 테이블 엔트리의 예를 예시한다;

도 9는 스테이지 2 변환 테이블 엔트리에 의해 정의되는 부분 판독 전용 메모리 영역에 대한 기록이 허락되고 금지되는 기록 요청의 타입을 예시한다;

도 10은 어드레스 변환 및 퍼미션 체크를 수행하기 위한 방법을 예시한다;

도 11은 부분 판독 전용 퍼미션에 기초한 체크를 예시하는 흐름도이다;

도 12, 도 13 및 도 14는 보호 엔트리 체크 타입의 판독 체크 기록 명령어(read-check-write instruction)의 상이한 변형(variant)의 프로세싱을 예시한다;

도 15는 동일한 중간 어드레스 상으로의 가상 어드레스의 앨리어싱을 예시한다;

도 16은 안티 앨리어싱 체크를 예시한다;

도 17은 스테이지 1 최상위 레벨 테이블 존재 체크 및 스테이지 1 최상위 레벨 부분 판독 전용 체크를 예시한다; 그리고

도 18은 사용될 수도 있는 시뮬레이터 구현예를 예시한다.

발명을 실시하기 위한 구체적인 내용

[0008] 부분 판독 전용 스테이지 2 액세스 퍼미션

[0009] 장치는 상이한 레벨의 특권과 관련되는 복수의 실행 상태 중 하나에서 명령어를 프로세싱하는 프로세싱 회로부, 및 메모리 액세스 요청에 의해 명시되는 타겟 가상 어드레스를 메모리 액세스 요청에 응답하여 액세스될 메모리 시스템 위치를 나타내는 타겟 물리적 어드레스로 변환하는 메모리 관리 회로부를 갖는다. 메모리 관리 회로부는 스테이지 1 변환 테이블 엔트리를 포함하는 스테이지 1 변환 테이블 구조 및 스테이지 2 변환 테이블 엔트리를 포함하는 스테이지 2 변환 테이블 구조에 기초하여 2 스테이지 어드레스 변환을 지원하는데, 여기서 타겟 가상 어드레스로부터 타겟 물리적 어드레스로의 매핑은 스테이지 1 어드레스 매핑 및 스테이지 2 어드레스 매핑에 의존한다. 스테이지 1 어드레스 매핑은 타겟 가상 어드레스로부터 타겟 가상 어드레스에 대응하는 대응 스테이지 1 변환 테이블 엔트리에 의해 명시되는 타겟 중간 어드레스로의 매핑을 포함한다. 스테이지 2 어드레스 매핑은 타겟 중간 어드레스로부터 타겟 중간 어드레스에 대응하는 대응 스테이지 2 변환 테이블 엔트리에 의해 명시되는 타겟 물리적 어드레스로의 매핑을 포함한다.

[0010] 2 스테이지 어드레스 변환을 지원하는 시스템에서, 스테이지 1 어드레스 매핑은 통상적으로 오퍼레이팅 시스템에 의해 제어되고, 한편, 스테이지 2 어드레스 매핑은 통상적으로 하이퍼바이저에 의해 제어된다. 그러므로, 스테이지 1 어드레스 매핑을 저장하기 위해 사용되는 메모리 시스템 위치는 통상적으로 오퍼레이팅 시스템에 의해 기록 가능하고, 그 결과, 오퍼레이팅 시스템은 스테이지 1 변환 테이블 엔트리에 의해 표현되는 어드레스 매핑 및 퍼미션을 유지할 수 있다. 최신 오퍼레이팅 시스템 소프트웨어는 상대적으로 클 수도 있고 따라서 메모리에 대한 기록을 수반하는 많은 라인의 코드를 가질 수도 있는데, 이것은, 만약 저장 명령어의 어드레스 피연산자가 스테이지 1 변환 테이블 엔트리를 유지하기 위해 사용되는 메모리 영역에 대응하도록 설정되는 경우, 스테이지 1 변환 테이블 엔트리에 대한 업데이트를 야기할 수 있을 수 있다. 따라서, 오퍼레이팅 시스템은, 액세스 퍼미션 또는 어드레스 매핑을 변경하고 스테이지 1 변환 테이블 엔트리를 사용하여 구현되도록 의도되는 메모리 관리 제어를 우회하기 위해 오퍼레이팅 시스템 코드로 하여금 스테이지 1 변환 테이블 엔트리 중 일부를 손상시키게 하는 것에 의해, 민감한 데이터 또는 코드에 대한 액세스를 얻으려고 또는 장치 상에서 실행되는 소프트웨어의 기능을 손상시키려고 시도하는 공격자에 의해 사용될 수 있는 큰 공격 표면을 제공할 수도 있다. 따라서, 일부 스테이지 1 변환 테이블 엔트리의 손상을 방지하기 위한 대응 대책을 제공하는 것이 바람직할 수도 있다(일부 스테이지 1 변환 테이블 엔트리가 민감하지 않은 정보를 유지하기 위해 사용되는 메모리 영역에 대응할 수도 있기 때문에, 이 대응 대책을 모든 스테이지 1 변환 테이블 엔트리에 적용하는 것은 필수는 아닐 수도 있다).

[0011] 공격에 대비하여 스테이지 1 변환 테이블을 강화하기 위해 제안된 한 가지 접근법은, 일반 스테이지 1 변환 테이블을 가리키기 위해 사용되는 기본 어드레스와는 별개로, 대안적인 기본 어드레스를 제공하는 것이다. 대안적인 기본 어드레스는 스테이지 1 변환 테이블의 대안적인 세트를 가리킨다. 대안적인 변환 테이블 기본 어드레스 및 스테이지 1 변환 테이블의 대안적인 세트는 오퍼레이팅 시스템이 아닌 하이퍼바이저에 의해 제어될 수도 있으며 어드레스 공간의 민감한 영역의 소정의 서브세트에 대한 액세스를 제어하기 위해 사용될 수도 있다. 대안적인 변환 테이블 구조의 각각의 레벨에 있는 변환 테이블 엔트리는 스테이지 1 변환 테이블의 대안적인 세트의 변환 테이블 워크가 중지되어야 하고 일반 스테이지 1 변환 테이블을 대신 사용하여 재시작되어야 하는지의 여부를 나타내는 재시작 지시자(restart indicator)를 명시할 수도 있다. 대안적인 변환 테이블 구조의 워크 동안 변환 테이블 워크를 재시작하라는 어떠한 표시도 조우되지 않는 경우, 그러면, 결국에는 어드레스 매핑

(대안적인 변환 테이블의 일부로서 하이퍼바이저에 의해 제어됨)이 도달되고 이것은 민감한 정보를 유지하는 메모리 영역의 중간 어드레스를 식별하기 위해 사용될 수 있다. 그러나, 변환 테이블 구조의 주어진 레벨에서 변환 테이블 워크가 중지되고 재시작되어야 한다는 것을 재시작 지시자가 명시하는 경우, 그러면 오퍼레이팅 시스템에 의해 제어되는 일반 스테이지 1 변환 테이블을 위해 사용되는 기본 어드레스로부터 시작하는 추가적인 변환 테이블 워크가 개시된다. 이 접근법은 소정의 민감한 메모리 영역에 대한 액세스가 오퍼레이팅 시스템에 의해 무단 변경될 수 없는 대안적인 변환 테이블 구조에 기초하여 제어되는 것을 허용하고, 한편 공격에 대비하여 보호될 필요가 없는 다른 메모리 영역은 그들의 액세스가 오퍼레이팅 시스템에 의해 설정되는 스테이지 1 변환 테이블의 일반 세트에 기초하여 제어되게 할 수 있다. 그러나, 이 접근법을 갖는 문제점은, 민감하지 않은 메모리 영역에 대한 액세스의 경우, 두 개의 별개의 변환 테이블 워크 동작을 수행하기 때문에 이미 느린 변환 테이블 워크 동작이 더욱더 느려지게(최대 2배 길이) 만들어진다는 것이다: 제1 동작은 대안적인 변환 테이블 구조를 워크하는 것이고 제2 동작은 대안적인 구조의 워크를 중지한 이후 일반 변환 테이블 구조를 워크하는 것이다. 이것은, 특히 대부분의 어드레스 공간이 민감하도록 의도되지 않고 따라서 일반 스테이지 1 변환 테이블을 사용하는 경우, 성능에 상당한 영향을 끼칠 수도 있다.

[0012] 다른 접근법은, 스테이지 1 변환 테이블 엔트리를 저장하기 위해 사용되는 메모리 영역에 대응하는 스테이지 2 변환 테이블 엔트리에 대해 설정되는 스테이지 2 액세스 퍼미션을 사용하여, 그 메모리 영역을 관독 전용으로 마킹하는 것일 수 있는데, 이것은 오퍼레이팅 시스템이 스테이지 1 변환 테이블 엔트리를 업데이트할 수 있는 것을 방지할 것이다. 그러나, 이 접근법의 문제점은, 오퍼레이팅 시스템이 스테이지 1 변환 테이블 엔트리의 액세스 추적 메타데이터를 업데이트하여, 주어진 메모리 영역이 얼마나 자주 액세스되거나 또는 기록되는지에 대한 정보를 기록할 수 있는 것이 종종 유용하다는 것이다. 예를 들면, 메타데이터는 액세스 플래그가 이전에 클리어된 이후 대응하는 메모리 영역이 적어도 한 번 액세스되었다는 것을 추적하도록 설정되는 액세스 플래그, 및/또는 더티 플래그(dirty flag)가 클리어된 이후 대응하는 메모리 영역에 대한 적어도 한 번의 기록이 발생하였다는 것을 나타내도록 설정되는 더티 플래그를 포함할 수 있다. 액세스 추적 메타데이터의 다른 예는 대응하는 메모리 영역에 대한 액세스의 횟수를 카운트하는 액세스 카운터일 수 있다. 그러한 액세스 추적 메타데이터는, 오퍼레이팅 시스템이 외부 스토리지로 메모리 영역의 페이징 아웃을 더욱 효율적으로 관리하는 것을 허용하는 데(예를 들면, 오퍼레이팅 시스템이 외부 스토리지로 페이징 아웃하기 위한 최상의 후보인 가장 덜 빈번하게 액세스된 페이지를 식별하는 것을 허용함) 유용할 수 있다. 그러므로, 스테이지 1 변환 테이블 엔트리를 저장하기 위해 사용되는 메모리 영역이 스테이지 2 액세스 퍼미션 정보를 사용하여 관독 전용으로 마킹된 경우, 이것은 액세스 추적 메타데이터에 대한 업데이트를 방지할 것이고 따라서 페이징 및 오퍼레이팅 시스템에 의해 관리되는 다른 동작의 덜 효율적인 관리로 이어질 수도 있는데, 이것은 액세스 빈도에 대한 정보의 손실에 기인하여 빈번하게 액세스된 데이터가 외부 스토리지로 페이징 아웃되는 것에 기인하는 성능의 손실의 위험이 있다.

[0013] 하기에서 논의되는 예에서, 메모리 관리 회로부는 메모리 액세스의 타겟 중간 어드레스에 대응하는 대응 스테이지 2 변환 테이블 엔트리에 의해 명시되는 스테이지 2 액세스 퍼미션 정보에 적어도 기초하여 메모리 액세스 요청에 대한 퍼미션 체크를 수행한다. 스테이지 2 액세스 퍼미션 정보는 타겟 중간 어드레스에 대응하는 메모리 영역이 부분 관독 전용 퍼미션을 가지는지의 여부를 명시하는 인코딩을 갖는다. 부분 관독 전용 퍼미션은, 관독 전용, 관독/기록, 실행, 관독/실행, 등등과 같은 다른 타입의 퍼미션에 대한 대안으로서, 주어진 메모리 영역에 대한 선택에 이용 가능한 다른 퍼미션 옵션일 수도 있다. 관독 전용 퍼미션이 대응하는 메모리 영역에 대한 모든 기록을 금지할 수도 있지만, 대조적으로 부분 관독 전용 퍼미션은, 프로세싱 회로부가 복수의 실행 상태 중 사전 결정된 실행 상태에 있는 경우 발행되는, 타겟 중간 어드레스에 대응하는 메모리 영역에 대한 기록 요청 중에서도, 기록 요청은 기록 요청 타입의 제한된 서브세트에 대해서는 허락되지만 그러나 제한된 서브세트 이외의 기록 요청 타입에 대해서는 금지된다. 기록 요청 타입의 제한된 서브세트는 변환 테이블 엔트리의 액세스 추적 메타데이터를 업데이트하기 위해 메모리 관리 회로부에 의해 생성되는 메타데이터 업데이트 기록 요청을 적어도 포함한다. 메모리 관리 회로부는 메모리 액세스 요청이 사전 결정된 실행 상태에서 발행되는 기록 요청이라는 것을 결정하는 것에 응답하여 메모리 액세스 요청을 거부하고, 스테이지 2 액세스 퍼미션 정보는 타겟 중간 어드레스에 대응하는 메모리 영역이 부분 관독 전용 퍼미션을 갖는다는 것을 명시하고, 메모리 액세스 요청은 기록 요청 타입의 제한된 서브세트 이외의 기록 요청 타입이다.

[0014] 그러므로, 이 접근법을 통해, 스테이지 2 액세스 퍼미션을 제어하는 하이퍼바이저는 스테이지 1 변환 테이블 엔트리를 저장하기 위해 사용되는 메모리 영역을 부분 관독 전용 퍼미션을 갖는 것으로 설정할 수 있는데, 이것은 액세스 추적 메타데이터에 대한 업데이트가 발생하는 것을 여전히 허용하지만(이것은 외부 스토리지로 페이징과 같은 동작의 제어의 효율성을 증가시키는 것에 의해 성능을 향상시키는 데 도움이 되는 경향이 있을 것임) 그러나 적어도 어떤 다른 요청 타입이 대응하는 메모리 영역에 기록하는 것을 방지한다. 부분 관독 전용 퍼미

선과 관련되는 영역에 기록하는 능력을 기록 요청 타입의 제한된 서브세트로 제한하는 것에 의해, 이것은 공격자가 악용하는 데 이용 가능한 공격 표면을 감소시켜, 그러한 공격에 대비하여 보호를 강화한다.

[0015] 몇몇 예에서, 메타데이터 업데이트 기록 요청은 제한된 서브세트 내에서 기록 요청의 유일한 타입일 수 있다. 이 경우, 오퍼레이팅 시스템이 스테이지 2 액세스 퍼미션 정보에 의해 부분 판독 전용으로 마킹되는 메모리 영역에 저장되는 스테이지 1 변환 테이블 엔트리를 업데이트할 필요가 있는 경우, 그러면 업데이트는 실패하여 하이퍼바이저에 트랩될(trap) 수도 있는데, 하이퍼바이저는, 그 다음, 오퍼레이팅 시스템에 의해 요청되는 업데이트가 적절한지의 여부를 결정할 수 있고, 그 다음, 오퍼레이팅 시스템에 제어를 반환하기 이전에 업데이트를 에뮬레이션할 수 있다.

[0016] 그러나, 실제로는, 스테이지 1 변환 테이블 엔트리에 대한 업데이트는 상대적으로 빈번할 수도 있고 따라서 각각의 업데이트에 대한 하이퍼바이저에 트랩핑하는 것은 성능에 영향을 끼칠 수도 있다. 따라서, 기록 요청 타입의 제한된 서브세트가 사전 결정된 실행 상태에서 프로세싱 회로부에 의해 실행되는 적어도 하나의 사전 결정된 타입의 변환 테이블 엔트리 업데이트 명령어에 응답하여 발행되는 기록 요청을 또한 포함하는 것이 유용할 수 있다. 다른 한편으로, 사전 결정된 실행 상태에서 프로세싱 회로부에 의해 실행되는 적어도 하나의 타입의 저장 명령어에 응답하여 발행되는 기록 요청이 기록 요청 타입의 제한된 서브세트로부터 배제될 수도 있다. 그러므로, 부분 판독 전용 영역에 기록하도록 허락되는 기록 요청의 타입을 제한하는 것에 의해, 이것은 공격자가 저장 명령어의 거동을 수정하여 스테이지 1 어드레스 변환 테이블 엔트리의 손상을 야기할 기회를 감소시킨다. 부분 판독 전용 영역에 기록하는 능력이 변환 테이블 엔트리를 업데이트하도록 의도되는 전용 명령어 타입의 제한된 서브세트로 제한될 때, 이것은 (변환 테이블 엔트리를 업데이트하도록 코드 작성자에 의해 의도되지 않는) 오퍼레이팅 시스템 코드의 일부에서 사용되는 다른 타입의 범용 저장 명령어가 그들로 하여금 메모리의 부분 판독 전용 영역에 저장되는 변환 테이블 엔트리를 손상시키게 하도록 해킹될 수 없다는 것을 의미한다. 변환 테이블 엔트리 업데이트 명령어는, 보안성을 강화하기 위해, 프로세싱 회로부로 하여금, 메모리에 대한 기록을 트리거하기 위해 사용되는 다른 타입의 저장 명령어에 응답하여 수행되지 않을 적어도 하나의 추가적인 체크를 수행하게 할 수 있다.

[0017] 사전 결정된 실행 상태는 오퍼레이팅 시스템 레벨의 특권과 관련되는 실행 상태일 수도 있다. 예를 들면, 하기의 예에서, 사전 결정된 실행 상태는 예외 레벨(exception level)(EL1)일 수도 있다. 몇몇 예에서, 사전 결정된 실행 상태에 의해 부과되는 기록 제한은, 프로세싱 회로부가 그 사전 결정된 실행 상태 또는 더 적은 특권이 있는 실행 상태에 있는 경우에만 적용될 수도 있다. 대안적으로, 부분 판독 전용으로 마킹되는 영역에 대한 기록 제한은 현재의 실행 상태와 관계없이 적용될 수 있고, 그 결과, 그들은 사전 결정된 실행 상태를 포함한 모든 실행 상태에서 적용된다. 그럼에도 불구하고, 적어도 사전 결정된 실행 상태에서 발행되는 메모리 액세스 요청에 대해서는, 그 사전 결정된 실행 상태에서 발행되는 일부 기록 요청 타입이 허용될 수도 있고 다른 기록 요청 타입이 허용되지 않을 수도 있도록 기록 능력에 대한 제한이 부분 판독 전용 퍼미션에 기초하여 부과되기 때문에, 이것은 또한 구현되고 있는 임의의 특권 기반의 제어에 직교하는 추가적인 제어를 제공하는데, 이것은, 변환 테이블 엔트리의 액세스 추적 메타데이터를 업데이트하기 위한 기록을 발행하는 메모리 관리 회로부의 능력을 손상시키지 않으면서, 오퍼레이팅 시스템의 능력에 대한 추가적인 제한이 스테이지 1 변환 테이블 엔트리를 저장하기 위해 사용되는 소정의 메모리 영역의 데이터를 업데이트하는 것을 허용할 수 있다.

[0018] 몇몇 구현예에서, 메모리 관리 회로부는, 부분 판독 전용 퍼미션을 갖는 메모리 영역에 기록하기 위해, 사전 결정된 실행 상태보다 더 많은 특권이 있는 실행 상태에서 프로세싱 회로부에 의해 실행되는 상기 적어도 하나의 타입의 저장 명령어에 응답하여 발행되는 기록 요청을 허락하도록 구성된다. 그러나, 실제로는, 통상적으로 더 많은 특권이 있는 실행 상태에서 동작하는 하이퍼바이저가 임의의 경우에 사전 결정된 실행 상태에서 오퍼레이팅 시스템에 의해 사용되는 스테이지 2 변환 테이블과는 상이한 변환 테이블의 세트를 사용할 수도 있기 때문에, 이것은 필수적인 것은 아니며, 그 결과, 심지어 스테이지 1 변환 테이블 엔트리를 저장하기 위해 사용되는 메모리 영역이 오퍼레이팅 시스템에 의해 사용되는 스테이지 2 변환 테이블에서 부분 판독 전용으로 마킹되더라도, 이것은 동일한 메모리 영역이 하이퍼바이저에 의해 사용되는 변환 테이블에서 부분 판독 전용으로 마킹되어야 한다는 것을 의미하는 것은 아니다. 그 메모리 영역은 하이퍼바이저에 의해 사용되는 변환 테이블에서 기록 가능한 것으로서 정의될 수 있고, 따라서 하이퍼바이저는, 심지어 오퍼레이팅 시스템에 의해 사용되는 스테이지 2 변환 테이블이 그 영역을 부분 판독 전용으로 마킹하는 경우에도, 대응하는 스테이지 1 변환 테이블 엔트리에서 정보(예컨대 스테이지 1 어드레스 매핑 그 자체 및/또는 하기의 예에서 추가로 논의되는 바와 같이 보호된 엔트리를 정의하는 비트필드)를 여전히 설정할 수 있을 수도 있다.

[0019] 스테이지 2 액세스 퍼미션 정보는 대응하는 스테이지 2 변환 테이블 엔트리에 의해 직접적으로 또는 간접적으로

(또는 양자의 조합으로) 명시될 수 있다. 직접적인 명시를 통해, 스테이지 2 변환 테이블 엔트리 그 자체의 인코딩은 스테이지 2 액세스 퍼미션 정보를 명시한다. 간접적인 명시를 통해, 스테이지 2 변환 테이블 엔트리의 인코딩은 스테이지 2 변환 테이블 엔트리와는 별개의 구조에 저장되는 스테이지 2 액세스 퍼미션 정보의 복수의 세트 중 하나를 참조하는 값을 명시할 수도 있다. 예를 들면, 대응하는 퍼미션 타입을 나타내는 인코딩으로 각각 설정될 수 있는 다수의 퍼미션 필드를 포함하는 퍼미션 간접 표현 레지스터(permission indirection register)가 제공될 수도 있다. 스테이지 2 변환 테이블 엔트리의 일부는 퍼미션 간접 표현 레지스터의 퍼미션 필드 중 어떤 것이 그 스테이지 2 변환 테이블 엔트리에 대한 스테이지 2 액세스 퍼미션 정보를 제공하는지를 선택하기 위해 사용되는 인덱스 값으로서 해석될 수도 있다. 그러므로, 몇몇 예에서, 부분 판독 전용 퍼미션을 명시하는 정보는 스테이지 2 변환 테이블 엔트리에서 직접적으로 명시될 수 있다. 다른 예에서, 부분 판독 전용 퍼미션을 명시하는 정보는, 부분 판독 전용 퍼미션을 나타내기 위해 인코딩되는 스테이지 2 퍼미션 간접 표현 레지스터(또는 다른 데이터 구조)의 필드에 대한 인덱스를 명시하는 스테이지 2 변환 테이블 엔트리에 의해 간접적으로 명시될 수 있다. 액세스 퍼미션의 간접 명시를 사용하는 것은 유용할 수 있는데, 그 이유는, 각각의 변환 테이블 엔트리를 개별적으로 업데이트하는 것을 필요로 하는 것이 아니라, 각각이 간접 표현 레지스터의 동일한 필드를 간접 표현 레지스터의 그 필드에 대한 단일의 기록에 의해 참조하는 다수의 상이한 변환 테이블 엔트리에 대한 퍼미션을 소프트웨어가 업데이트하는 것을 그것이 허용하기 때문이다. 몇몇 구현예에서, 간접 표현 레지스터를 사용하는 것은 변환 엔트리 그 자체에서 인코딩될 수 있는 것보다 더 많은 수의 퍼미션 타입에 대한 옵션을 지원하는 데 또한 도움이 될 수 있다.

[0020] 보호된 스테이지 1 변환 테이블 엔트리

[0021] 몇몇 구현예는 스테이지 2 액세스 퍼미션에 의해 부분 판독 전용으로 마킹되는 메모리 영역 내에 저장되는 모든 변환 테이블 엔트리가, 사전 결정된 실행 상태에서 발행되는 요청 타입의 제한된 서브세트 중 하나가 아닌 기록 요청에 대해 판독 전용으로 간주되어야 한다는 것을 가정할 수도 있다. 그러나, 실제로는, 스테이지 2 변환 엔트리에 대응하는 주어진 메모리 영역은 상이한 메모리 영역에 대응하는 많은 수의 상이한 스테이지 1 변환 테이블 엔트리를 포함하는 변환 테이블을 저장할 수도 있으며, 이들 메모리 영역 모두가 오퍼레이팅 시스템으로부터의 부적절한 기록 액세스 요청에 의한 손상에 대비하여 보호되는 대응하는 스테이지 1 변환 테이블 엔트리를 필요로 하지는 않을 수도 있다. 예를 들면, 상이한 스테이지 1 변환 테이블 엔트리에 의해 표현되는 메모리 영역은 민감한 그리고 민감하지 않은 정보의 혼합에 대응할 수도 있다. 대응하는 메모리 영역이 스테이지 2 액세스 퍼미션에서 부분 판독 전용 퍼미션을 가지도록 설정되는 것에도 불구하고, 오퍼레이팅 시스템이 민감하지 않은 정보와 관련되는 스테이지 1 변환 테이블 엔트리를 업데이트하는 것을 허용하는 것이 여전히 소망되는 경우, 그러면, 한 가지 접근법은 업데이트가 허락되는지의 여부를 결정할 수 있는 하이퍼바이저에 각각의 업데이트를 트랩하는 것일 수 있다. 그러나, 그러한 업데이트 요청은 상대적으로 빈번할 수도 있고 따라서 이것은 성능의 손실을 야기할 수도 있다.

[0022] 따라서, 하기에서 논의되는 예에서, 각각의 스테이지 1 변환 테이블 엔트리는, 그 엔트리가 보호된 엔트리로서 인코딩되지 않는 스테이지 1 변환 테이블 엔트리와 비교하여 업데이트가 제한되는 보호된 엔트리인지의 여부를 명시하는 인코딩을 가질 수도 있다. 이것은 민감한 및 민감하지 않은 정보에 대응하는 엔트리가 서로 구별되는 것을 가능하게 하여 보호되지 않은 엔트리와 비교하여 이들 보호된 엔트리에 대한 업데이트에 대한 더욱 엄격한 체크를 허용한다. 예를 들면, 부분 판독 전용 메모리 영역에 기록하도록 허락되는, 상기에서 논의되는 사전 결정된 타입의 변환 테이블 엔트리 업데이트 명령어는 기록 요청에 의해 타겟으로 되는 메모리 영역에 저장되는 데이터가 보호된 엔트리로서 인코딩되는지의 여부를 체크를 포함할 수 있고, 메모리 영역의 콘텐츠가 보호된 엔트리로서 인코딩되는 것으로 확인되는 경우 메모리 영역의 적어도 일부 비트에 대한 기록을 억제할 수 있다. 다른 한편으로, 보호되지 않은 엔트리의 경우, 기록은 변환 테이블 엔트리 업데이트 명령어에 응답하여 허락될 수도 있다(임의의 다른 액세스 퍼미션 체크가 충족된다는 것을 가정함). 그러므로, 스테이지 1 변환 테이블 엔트리를, 다른 보호되지 않은 엔트리와 구별하여, 보호된 엔트리로서 인코딩하기 위한 지원을 제공하는 것에 의해, 이것은 보호되지 않은 엔트리가 하이퍼바이저 개입 없이 업데이트되는 것을 가능하게 할 수 있는데, 이것은 성능 향상에 도움이 될 수 있다.

[0023] 장치는, 주어진 스테이지 1 변환 테이블 엔트리가 보호된 엔트리로서 마킹되는지의 여부에 기초하여, 주어진 스테이지 1 변환 테이블 엔트리에 의해 명시되는 액세스 퍼미션 또는 어드레스 매핑에 대한 업데이트를 제한할지의 여부를 체크하는 보호 엔트리 체크 회로부(protected-entry-checking circuitry)를 가질 수도 있다. 변환 테이블 엔트리가 액세스 추적 메타데이터를 명시하는 경우, 액세스 추적 메타데이터에 대한 업데이트는 엔트리가 보호되는지의 여부에 기초하여 제한되지 않을 수도 있고, 따라서 (예를 들면, 페이지 테이블 워크 동작

동안) 하드웨어의 메모리 관리 회로부에 의해 생성되는 메타데이터 업데이트 기록 요청은, 심지어 업데이트되고 있는 스테이지 1 변환 테이블 엔트리가 보호된 엔트리인 경우에도, 허락될 수도 있다.

[0024] 스테이지 2 액세스 퍼미션 정보에서 정의되는 부분 판독 전용 퍼미션 및 스테이지 1 변환 테이블 엔트리를 위해 사용되는 보호된 엔트리 인코딩의 조합은, 어느 하나의 피처의 단독의 사용과 비교하여, 특히 효과적일 수 있다. 보호된 스테이지 1 변환 테이블 엔트리의 인코딩을 지원하지 않고도 스테이지 2에서 부분 판독 전용 퍼미션을 지원하는 것이 가능할 것이지만, 상기에서 언급되는 바와 같이, 이것은 하이퍼바이저에 대한 더욱 빈번한 트랩을 필요로 할 경향이 있을 것이다. 그러므로, 이들 피처의 조합은 특히 효과적일 수 있다.

[0025] 몇몇 예에서, 엔트리가 보호된 엔트리인지의 여부를 인코딩하는 스테이지 1 변환 테이블 엔트리의 정보는, 엔트리가 프로세싱 회로부의 동작 상태 또는 모든 모드에서 보호된 엔트리인지의 여부를 나타내는 것으로 처리되는 전용 비트 또는 필드일 수도 있다. 그러나, 엔트리가 보호된 엔트리인지의 여부를 인코딩하기 위해 사용되는 스테이지 1 변환 테이블 엔트리의 정보가 제어 레지스터에 저장되는 제어 상태에 따라 또는 프로세싱 회로부의 동작 모드에 따라 상이한 방식으로 해석되는 것이 또한 가능하다. 예를 들면, 제어 레지스터의 제어 값은 스테이지 1 변환 테이블 엔트리의 소정의 비트 또는 비트필드가 엔트리가 보호된 엔트리인지의 여부를 나타내는 것으로 해석되어야 하는지, 또는 (보호된 엔트리 인코딩을 지원하지 않는 시스템에서 사용되는 레거시 인코딩에 따라) 다른 방식으로 해석되어야 하는지의 여부를 나타낼 수 있다. 따라서, 장치가 적어도 하나의 동작 상태에서 스테이지 1 변환 테이블 엔트리를 보호된 엔트리로서 인코딩하는 것을 지원할 수도 있지만, 스테이지 1 변환 테이블 엔트리가 프로세싱 회로부의 모든 동작 상태에서 그것이 보호되는지의 여부를 인코딩하는 것은 필수적인 것은 아니다. 몇몇 예에서, 스테이지 1 변환 테이블 엔트리가 보호되는지의 여부는 부분적으로는 엔트리 그 자체의 인코딩에, 그리고 부분적으로는 제어 레지스터에 저장되는 제어 값에 의존할 수도 있다.

[0026] 보호 엔트리 체크 타입의 판독 체크 기록 명령어

[0027] 프로세싱 회로부는 주어진 메모리 시스템 위치에 대한 업데이트를 요청하기 위한 보호 엔트리 체크 타입의 판독 체크 기록(read-check-write; RCW) 명령어를 지원할 수도 있다. 보호 엔트리 체크 회로부는 보호 엔트리 체크 타입의 RCW 명령어를 실행하는 프로세싱 회로부에 응답하여 다음의 것을 행할 수도 있다: 주어진 메모리 시스템 위치로부터 데이터를 판독하는 것, 주어진 메모리 시스템 위치로부터 판독되는 데이터가 보호된 엔트리로서 명시되는 스테이지 1 변환 테이블 엔트리와 일치하는 값을 갖는지의 여부를 체크하는 것, 및 주어진 메모리 시스템 위치로부터 판독되는 데이터가 보호된 엔트리로서 명시되는 스테이지 1 변환 테이블 엔트리와 일치하는 값을 갖는다는 것을 결정하는 것에 응답하여, 스테이지 1 변환 테이블 엔트리의 비트의 업데이트 제한 서브세트가 판독 체크 기록 명령어에 응답하여 업데이트되는 것을 방지하는 것. 그러므로, 프로세싱 회로부에 의해 지원되는 명령어 세트 아키텍처에서, 업데이트될 데이터가 보호된 스테이지 1 변환 테이블 엔트리의 인코딩과 일치하는 값을 갖는지의 여부를 체크를 트리거하는 명령어를 지원하는 것에 의해, 이것은 각각의 업데이트를 하이퍼바이저에 트랩할 필요성을 방지하고, 그러므로, 오퍼레이팅 시스템이 하이퍼바이저 개입 없이 보호되지 않은 스테이지 1 변환 테이블 엔트리를 업데이트할 메커니즘을 제공하여, 성능을 향상시킨다.

[0028] 판독된 데이터의 체크는, 데이터가 실제로 스테이지 1 변환 테이블 엔트리가 되도록 의도되는지 또는 단지 다른 목적을 위해 설정되는 값을 갖는 일부 데이터인지의 여부를 확실하게 구별하지 못할 수도 있다는 것을 유의한다. 체크는 데이터의 값이 보호된 엔트리로서 명시되는 스테이지 1 변환 테이블 엔트리의 인코딩과 일치하는지의 여부를 결정할 수도 있다(예를 들면, 보호된 스테이지 1 변환 테이블 엔트리가 주어진 메모리 시스템 위치에서 저장된 경우 그들 비트가 가질 값과 데이터의 소정의 비트가 매치하는지의 여부를 체크함).

[0029] 보호 엔트리 체크 타입의 RCW 명령어는 메모리의 부분 판독 전용 영역에 기록하도록 허락되는 기록 요청 타입의 제한된 서브세트 중 하나일 수도 있고, 그 결과, 메모리의 영역이 부분 판독 전용으로 마킹되는 경우 그것은 스테이지 2 액세스 퍼미션 체크를 통과할 수도 있다.

[0030] 주어진 스테이지 1 변환 테이블 엔트리가 보호된 엔트리인지의 여부를 명시하는 정보의 설정은 사전 결정된 실행 상태보다 더 높은 특권 레벨에서 실행되는 소프트웨어, 예컨대 하이퍼바이저에 의해 제어될 수도 있다. 예를 들면, 하이퍼바이저는 대응하는 오퍼레이팅 시스템을 초기화할 때 스테이지 1 변환 테이블 엔트리의 세트를 메모리에 처음 기록할 때 소정의 보호된 스테이지 1 변환 테이블 엔트리를 명시할 수 있다. 이것은 대응하는 메모리 구조에 대한 기록이 하이퍼바이저에 의해 기록되는 것을 하이퍼바이저에 의해 사용되는 변환 테이블 구조의 스테이지 2 변환 테이블 엔트리가 제한하지 않을 수도 있기 때문에 가능하다. 그 다음 후속하여 오퍼레이팅 시스템이 보호되지 않은 스테이지 1 변환 테이블 엔트리를 업데이트하는 것이 허용될 수도 있지만, 보호된 엔트리는 오퍼레이팅 시스템에 의해 실행되는 명령어에 응답하여 불변인 적어도 일부 비트를 가질 수도 있다.

- [0031] 보호 엔트리 체크 타입의 RCW 명령어의 판독, 체크 및 기록은 보호 엔트리 체크 회로부에 의해 원자성 동작(atomic operation)으로서 수행될 수도 있다. 그러므로, 판독, 체크 및 기록은 그 결과가 완전히 관찰되는 또는 전혀 관찰되지 않는 분할할 수 없는 동작의 세트로서 처리될 수도 있고, 그 결과, 데이터 프로세싱 장치 상에서 실행되는 다른 소프트웨어 프로세스가 보호 엔트리 체크 타입의 RCW 명령어를 실행하는 것의 부분적인 결과를 보는 것, 또는 RCW 명령어에 대해 수행되고 있는 판독과 기록 사이의 기간에 RCW 명령어에 의해 판독/기록되고 있는 기억 위치에 기록하는 것이 불가능하다. 이것은 보호 엔트리 체크 타입의 RCW 명령어에 응답하여 수행되는 판독과 기록 사이의 기간에 다른 소프트웨어 프로세스가 주어진 메모리 시스템 위치에 기록하는 것이 가능한 경우에 발생할 수 있는 경쟁 조건 발생을 방지한다.
- [0032] 보호된 엔트리 인코딩 체크에 후속하여 메모리 시스템 위치에 대한 업데이트가 허락되고, 따라서 보호 엔트리 체크 타입의 RCW 명령어에 응답하여 주어진 메모리 시스템 위치 중 적어도 하나의 비트가 업데이트되는 경우, 보호 엔트리 체크 회로부는 주어진 메모리 시스템 위치에 대한 업데이트가 주어진 메모리 시스템 위치에 저장되는 스테이지 1 변환 테이블 엔트리의 보호된 명시를 변경하는 것을 금지할 수도 있다. 그러므로, 보호되지 않은 엔트리의 경우, 오퍼레이팅 시스템은 그 엔트리의 일부를 업데이트하는 것이 허용될 수도 있지만, 오퍼레이팅 시스템은 RCW 명령어를 사용하여 그 엔트리로 하여금 보호된 엔트리가 되게 할 수 없다. 또한, 보호 엔트리 체크 타입의 RCW 명령어를 실행하는 것에 의해, 보호된 엔트리는 보호되지 않은 엔트리로 변환될 수 없다. 이것은, 하이퍼바이저에 의해 소정의 스테이지 1 변환 테이블 엔트리에 대해 시행되는 보호가 RCW 명령어를 실행하는 오퍼레이팅 시스템에 의해 우회될 수 없다는 것, 및 공격자에 의해 생성되는 임의적인 변환 테이블 엔트리로 하여금 보호된 것으로 마킹되게 하여 - 이것은 보호된 엔트리가 하이퍼바이저에 의해 승인된다는 가정에 의존하는 다른 보안성 체크를 야기할 수도 있음 - 우회되게 할 수 없다는 것을 보장한다.
- [0033] 상이한 구현에는 보호된 스테이지 1 변환 테이블 엔트리에 대한 업데이트가 제한되는 정도를 제어하기 위한 상이한 메커니즘을 가질 수도 있다.
- [0034] 몇몇 예에서, 비트의 업데이트 제한 서브세트는 스테이지 1 변환 테이블 엔트리의 모든 비트를 포함할 수 있다. 그러므로, 보호 엔트리 체크 타입의 RCW 명령어가 실행되고 주어진 메모리 시스템 위치로부터 판독되는 데이터가 보호된 스테이지 1 변환 테이블 엔트리와 일치하는 값을 갖는다는 것이 확인되는 경우, 그러면 주어진 메모리 시스템 위치의 데이터에 대한 요청된 업데이트는 완전히 억제될 수도 있다. 그럼에도 불구하고, 이것은 변환 테이블 워크 동안 메모리 관리 회로부에 의해 하드웨어에서 생성되는 메타데이터 업데이트 기록 요청에 응답하여 스테이지 1 변환 테이블 엔트리의 액세스 추적 메타데이터가 업데이트되는 것을 방지하지 못한다(RCW 명령어에 응답하여 업데이트를 제한하기 위해 사용되는 보호 엔트리 체크는 메타데이터 업데이트 기록 요청에 적용되지 않는다).
- [0035] 다른 예에서, 비트의 업데이트 제한 서브세트는 스테이지 1 변환 테이블 엔트리의 비트의 적절한 서브세트를 포함할 수 있고, 그 결과, 심지어 보호 엔트리의 경우에도, 일부 비트는 다른 비트가 업데이트되는 것이 금지되는 동안 보호 엔트리 체크 타입의 RCW 명령어에 의해 여전히 업데이트될 수 있다. 비트의 업데이트 제한 서브세트는, 상기에서 논의되는 바와 같이 오퍼레이팅 시스템이 보호된 엔트리를 보호되지 않은 엔트리로 변환할 수 있는 것 또는 그 반대로 할 수 있는 것을 방지하기 위해, 스테이지 1 변환 테이블 엔트리가 보호된 엔트리인지의 여부를 인코딩하기 위해 사용되는 적어도 임의의 하나 이상의 비트를 포함할 수도 있다. 그러나, 하이퍼바이저에 의해 의도되는 보호를 우회할 위험 없이 오퍼레이팅 시스템에 의해 업데이트될 수 있는 스테이지 1 변환 테이블 엔트리의 다른 비트가 존재할 수도 있고, 따라서 심지어 엔트리가 보호된 엔트리인 경우에도 일부 비트가 업데이트되는 것을 허용하는 것이 유용할 수도 있다는 것이 인식된다.
- [0036] 비트의 업데이트 제한 서브세트가 비트의 적절한 서브세트(보호 엔트리 체크 타입의 RCW 명령어에 응답하여 업데이트가 제한되게 하지 않는 적어도 하나의 다른 비트를 가짐)를 포함하는 예에서, 몇몇 구현에는 비트의 어떤 서브세트가 비트의 업데이트 제한 서브세트인지의 선택을 고정할 수도 있다. 그러므로, 어떤 비트가 비트의 적절한 서브세트에 있는지의 선택은 프로그래밍 불가능할 수도 있다. 예를 들면, 업데이트 제한 비트의 선택은 프로세싱 회로부에 의해 지원되는 명령어 세트 아키텍처에 의해 규정되는 하드웨어 고정식 아키텍처 선택(hardwired architectural choice)일 수 있다.
- [0037] 대안적으로, 어떤 비트가 비트의 업데이트 제한 서브세트에 있는지의 선택은, 제어 레지스터에 저장되는 소프트웨어 프로그래밍 가능 선택 값에 따라, 프로그래밍 가능할 수도 있다. 이것은 소정의 소프트웨어가, 그 소프트웨어의 필요에 따라, 어떤 특정한 비트가 그들의 업데이트를 제한되게 해야하는지를 프로그래밍하는 것을 허용할 수 있다. 예를 들면, 제어 레지스터에 저장되는 소프트웨어 프로그래밍 가능 선택 값을 업데이트하는 능력

은, 예를 들면 하이퍼바이저 레벨의 특권일 수 있는, 적어도 소정의 임계치 레벨의 특권을 갖는 실행 상태에서 실행되는 소프트웨어로 제한될 수도 있다. 상기에서 설명되는 사전 결정된 실행 상태에서 실행되는 소프트웨어는 소프트웨어 프로그래밍 가능 선택 값에 기록하는 것이 허용되지 않을 수도 있다.

[0038]

몇몇 구현예에서, 각각의 유효한 스테이지 1 변환 테이블 엔트리는 그 엔트리가 보호된 엔트리인지의 여부를 명시하는 인코딩을 가질 수도 있다. 그러나, 무효한 스테이지 1 변환 테이블 엔트리는 보호된 것으로 또는 보호되지 않은 것으로 인코딩될 수 없을 수도 있고, 디폴트로 보호되지 않은 엔트리인 것으로 가정될 수 있다. 이 경우, 무효한 엔트리는 RCW 명령어에 의해 업데이트되는 것이 허용될 수도 있는데, 업데이트가 엔트리로 하여금 유효하게 되게 하고 보호되게 하지 않는 한, 엔트리의 임의의 부분은 업데이트되는 것이 허용된다. 그러므로, 주어진 메모리 시스템 위치로부터 판독되는 데이터가 무효한 스테이지 1 변환 테이블 엔트리와 일치하는 값을 갖는다는 것을 결정하는 것에 응답하여, 보호 엔트리 체크 회로부는, 주어진 메모리 시스템 위치의 데이터로 하여금 보호된 엔트리로서 명시되는 유효한 스테이지 1 변환 테이블 엔트리가 되게 하는 업데이트가 아니라, 무효한 스테이지 1 변환 테이블 엔트리의 업데이트를 허락할 수도 있다.

[0039]

그러나, 다른 구현예에서, 유효한 그리고 무효한 스테이지 1 변환 테이블 엔트리 둘 모두는 그 엔트리가 보호된 엔트리인지의 여부를 명시하는 인코딩을 가질 수도 있다. 이 경우, 주어진 메모리 시스템으로부터 판독되는 데이터가 보호된 엔트리로서 명시되는 무효한 스테이지 1 변환 테이블 엔트리와 일치하는 값을 갖는다는 것을 결정하는 것에 응답하여, 보호 엔트리 체크 회로부는 주어진 메모리 시스템 위치의 비트의 무효 엔트리 업데이트 제한 서브세트가 판독 체크 기록 명령어에 응답하여 업데이트되는 것을 방지할 수도 있고; 주어진 메모리 시스템으로부터 판독되는 데이터가 보호된 엔트리로서 명시되는 유효한 스테이지 1 변환 테이블 엔트리와 일치하는 값을 갖는다는 것을 결정하는 것에 응답하여, 보호 엔트리 체크 회로부는 주어진 메모리 시스템 위치의 비트의 유효 엔트리 업데이트 제한 서브세트가 판독 체크 기록 명령어에 응답하여 업데이트되는 것을 방지할 수도 있다. 이 접근법은, 변환 테이블 엔트리가 아직 유효하지 않은 기간에 승인되지 않은 업데이트에 대비하여 여전히 보호되면서, 메모리에 대한 액세스를 허락하기 위해 현재 사용될 수 없는, 그러나 나중에 유효하게 만들어질 수도 있는 일부 무효한 엔트리를 비롯하여, 일부 엔트리가 보호되는 스테이지 1 변환 테이블 엔트리의 세트를 하이퍼바이저가 초기화하는 것을 허용하기 때문에 유용할 수 있다.

[0040]

몇몇 구현예에서, 비트의 유효 엔트리 업데이트 제한 서브세트는 비트의 무효 엔트리 업데이트 제한 서브세트와 동일한 비트 포지션에 있을 수 있다. 대안적으로, 비트의 유효 엔트리 업데이트 제한 서브세트는 비트의 무효 엔트리 업데이트 제한 서브세트와 비교하여 변환 테이블 엔트리 인코딩의 비트의 상이한 서브세트일 수 있다. 비트의 유효 엔트리 업데이트 제한 서브세트는, 제어 레지스터에 저장되는 소프트웨어 프로그래밍 가능 유효 엔트리 업데이트 제한 비트 선택 값에 따라, 스테이지 1 변환 테이블 엔트리의 모든 비트, 스테이지 1 변환 테이블 엔트리의 비트의 비프로그래밍 방식으로 선택된 적절한 서브세트, 또는 스테이지 1 변환 테이블 엔트리의 비트의 프로그래밍 방식으로 선택된 적절한 서브세트 중 어느 하나일 수 있다. 마찬가지로, 비트의 무효 엔트리 업데이트 제한 서브세트는, 제어 레지스터에 저장되는 소프트웨어 프로그래밍 가능 무효 엔트리 업데이트 제한 비트 선택 값에 따라, 스테이지 1 변환 테이블 엔트리의 모든 비트, 스테이지 1 변환 테이블 엔트리의 비트의 비프로그래밍 방식으로 선택된 적절한 서브세트, 또는 스테이지 1 변환 테이블 엔트리의 비트의 프로그래밍 방식으로 선택된 적절한 서브세트 중 어느 하나일 수 있다. 어떤 비트가 비트의 유효 엔트리 업데이트 제한 서브세트인지를 선택하기 위해 취해지는 접근법은 어떤 비트가 비트의 무효 엔트리 업데이트 제한 서브세트인지를 선택하기 위해 취해지는 접근법과 동일할 수 있거나 또는 상이할 수 있다. 예를 들면, 유효한 보호된 엔트리의 모든 비트는 그들의 업데이트가 제한되게 할 수 있고, 한편 무효한 보호 엔트리의 비트의 서브세트만이 그들의 업데이트가 제한되게 할 수 있다. 대안적으로, 업데이트 제한 비트는 무효한/유효한 엔트리 중 하나에 대해 프로그래밍 방식으로 선택될 수 있지만 그러나 다른 하나에 대해서는 선택될 수 없다. 예를 들면, 하나의 예시적인 구현예는 어떤 비트가 비트의 유효 엔트리 업데이트 제한 서브세트인지를 선택하기 위해 마스크 레지스터를 제공할 수 있지만, 그러나 어떤 비트가 비트의 무효 엔트리 업데이트 제한 서브세트인지의 선택은 명령어 세트 아키텍처에 의해 규정되는 고정된 선택일 수도 있다. 그러므로, 유효한 및 무효한 스테이지 1 변환 테이블 엔트리 둘 모두에서 어떤 비트가 업데이트 제한되는지를 선택하기 위한 아주 다양한 옵션이 있다는 것이 인식될 것이다.

[0041]

몇몇 예에서, 유효한 스테이지 1 변환 테이블 엔트리는 그 유효한 스테이지 1 변환 테이블 엔트리가 보호된 엔트리인지의 여부를 나타내는 제1 비트필드를 포함할 수도 있고; 무효한 스테이지 1 변환 테이블 엔트리는 그 무효한 스테이지 1 변환 테이블 엔트리가 보호된 엔트리인지의 여부를 나타내는 제2 비트필드를 포함할 수도 있다. 제1 및 제2 비트필드는 스테이지 1 변환 테이블 엔트리 인코딩 내에서 동일한 비트 포지션에, 또는 스테

이지 1 변환 테이블 엔트리 인코딩 내에서 상이한 비트 포지션에 있을 수 있다. 몇몇 경우에, 제1 및 제2 비트 필드를 상이한 비트 포지션에 위치시키는 것은, 소프트웨어가 자신의 데이터를 배치하는 방식에서 소프트웨어에 대해 더 많은 유연성을 제공할 수도 있다. 무효한 변환 테이블 엔트리 인코딩은 어떤 임의적인 비변환 테이블 데이터에 대해 소프트웨어에 의해 사용된다. 유효한 변환 테이블 엔트리 인코딩에서는, "보호된 상태"를 인코딩하는 데 이용 가능한 제한된 비트만이 있을 수도 있다. 따라서, 보호된 상태를 인코딩하기 위해 사용되는 비트필드가 유효한 및 무효한 엔트리 둘 모두에서 동일한 비트 포지션에 있는 경우, 이것은 소프트웨어가 변환 테이블 엔트리로서의 사용을 위해 의도되지 않는 아닌 기억 위치에서 자신의 데이터를 사용할 수 있는 방식을 제한할 수도 있다. 제1 및 제2 비트필드의 비트필드 포지션이 상이하게 되는 것을 허용하는 것에 의해, 무효한 스테이지 1 변환 테이블 엔트리 인코딩 내에서의 제2 비트필드는, 정상 데이터의 소프트웨어의 사용에 불편함을 야기할 가능성이 더 적은 포지션에서 위치될 수도 있는데, 심지어 그 포지션이, 유효한 스테이지 1 변환 테이블 엔트리 내의 다른 정보에 대해 이미 할당된 비트 포지션에 기인하여, 유효한 스테이지 1 변환 테이블 엔트리에 대해 적합하지 않더라도, 위치될 수도 있다.

[0042] (유효한 또는 무효한) 보호된 스테이지 1 변환 테이블 엔트리의 비트의 업데이트 제한 서브세트에 대한 업데이트의 억제는 상이한 방식으로 허락될 수 있다. 몇몇 예에서, 비트의 업데이트 제한 서브세트 중 적어도 하나가 업데이트되도록 요청받고 있고 주어진 메모리 시스템 위치로부터 판독되는 데이터가 보호된 엔트리를 나타내는 데이터와 일치하는 인코딩을 갖는다는 것이 결정되는 경우, 그러면, 보호 엔트리 체크 회로부는, 비트의 업데이트 제한 서브세트 이외의 비트에 대한 임의의 업데이트를 비롯하여, 주어진 메모리 시스템 위치 전체의 업데이트를 간단하게 억제할 수 있다. 이 경우, 기록은, 임의의 업데이트 제한 비트에 기록하려고 시도하는 경우, 실패한다.

[0043] 다른 예에서, 비트의 업데이트 제한 서브세트의 업데이트만이 억제될 수도 있고, 한편 주어진 메모리 시스템 위치의 적어도 하나의 다른 (업데이트 비제한(non-update-restricted)) 비트의 업데이트는 여전히 허락될 수도 있다. 이 경우, 보호된 엔트리의 경우, 기록이 업데이트하는 것이 허용되는 비트에 대한 업데이트에 대해 기록이 성공하지만, 그러나 심지어 RCW 명령어의 피연산자에 기초하여 그렇게 하도록 요청되는 경우에도, 업데이트 제한 비트를 변경하지 않는다.

[0044] 보호 엔트리 체크 타입의 판독 체크 기록 명령어에 응답하여, 프로세싱 회로부는, 주어진 메모리 시스템으로부터 판독되는 데이터가 보호된 엔트리로서 명시되는 스테이지 1 변환 테이블 엔트리의 여부의 체크에 기초하여 보호 엔트리 체크 타입의 판독 체크 기록 명령어에 의해 요청되는 업데이트의 적어도 일부가 억제되었는지의 여부를 나타내기 위해, 적어도 하나의 조건 상태 값을 설정할 수도 있다. 적어도 하나의 조건 상태 값은 후속하는 조건부 명령어(예컨대 조건부 비교 명령어 또는 조건부 분기 명령어)에 의해 테스트되어 RCW 명령어에 의해 요청되는 업데이트가 성공적이었는지 또는 성공적이지 않았는지의 여부에 따라 조건부로 후속하는 동작이 이루어지는 것을 허용할 수 있다. RCW 명령어가 성공하였는지 또는 실패하였는지 여부를 아는 것이 유용한 경우, 이것은 성능을 향상시키는 데 도움이 될 수 있는데, 그 이유는 기록이 성공적이었는지의 여부를 결정하기 위해 명령어의 추가적인 세트를 실행할 필요가 없다는 것을 의미하기 때문이다.

[0045] 보호 엔트리 체크 타입의 RCW 명령어는 다양한 방식으로 구현될 수 있다. 예를 들면, 프로세싱 회로부는 보호 엔트리 체크 타입의 RCW 명령어의 다음의 변형 중 임의의 하나 이상을 지원할 수 있다:

[0046]  보호 엔트리 체크 타입의 판독 체크 기록 명령어의 스왑 변형(swap variant) - 이것에 응답하여 프로세싱 회로부는 주어진 메모리 시스템 위치로부터 판독되는 데이터를, 후속하는 명령어에 의한 피연산자로서의 사용을 위해, 목적지 레지스터에 기록하도록 구성됨 - . 이 변형은, 후속하는 명령어가 업데이트 이전에 변환 테이블 엔트리의 이전 상태에 따라 조건부로 동작을 수행할 수 있도록, 유용할 수 있다.

[0047]  프로세싱 회로부가 주어진 메모리 시스템 위치로부터 판독되는 데이터를 목적지 레지스터에 기록하지 않고도 저장 변형을 프로세싱하도록 구성되는, 보호 엔트리 체크 타입의 판독 체크 기록 명령어의 저장 변형(store variant); 이 변형은, 목적지 레지스터를 불필요하게 소비하는 것을 방지하기 위해, 후속하는 동작이 주어진 메모리 시스템 위치의 이전 내용에 의존할 필요성이 없는 경우, 유용할 수 있다. 몇몇 예에서, 스왑 및 저장 변형은 공통 인코딩을 가질 수 있고, 스왑 변형의 목적지 레지스터 명시자(destination register specifier)가 사전 결정된 값을 명시하는 경우, 이것은, 명령어로 하여금, 주어진 메모리 시스템 위치로부터 판독되는 데이터를 반환할 필요가 없는 저장 변형으로서 취급되게 할 수 있다. 다른 아키텍처는 스왑 변형과는 별개로 저장 변형을 독립형 명령어로서 제공할 수 있다.

- [0048]  보호 엔트리 체크 회로부가 주어진 메모리 시스템 위치로부터 판독되는 데이터 및 비교 피연산자의 비교가 비교 조건을 충족하는지의 여부를 결정하도록, 비교 조건이 충족되지 않는 경우 주어진 메모리 시스템 위치에 대한 업데이트를 억제하도록, 그리고 후속하는 명령어에 의한 피연산자로서의 사용을 위해 주어진 메모리 시스템 위치로부터 판독되는 데이터를 목적지 레지스터에 기록하도록 구성되는, 비교 피연산자를 명시하는 보호 엔트리 체크 타입의 판독 체크 기록 명령어의 비교 및 스왑 변형(compare-and-swap variant). 비교 및 스왑 변형은 또한 비교 조건이 충족되는지의 여부를 나타내기 위해 상태 정보로 하여금 설정되게 할 수도 있다(예를 들면, 후속하는 조건부 명령어에 의해 테스트될 수 있는 하나 이상의 조건 플래그를 설정함). 비교 및 스왑 변형에 응답하여, 상태 정보는, 비교 조건이 충족되지만 그러나 업데이트 제한 비트에 대한 업데이트를 기록이 요청하고 있었을 때 판독된 데이터가 보호된 스테이지 1 변환 테이블 엔트리로서 인코딩되는 것에 기인하여 요청된 기록 동작의 적어도 일부가 억제되는 경우와 비교하여, 비교 조건이 충족되지 않는 경우 상이한 상태로 설정될 수도 있다.
- [0049]  주어진 메모리 시스템 위치의 업데이트가 주어진 메모리 시스템 위치에서 데이터의 적어도 하나의 선택된 비트를 설정하는 것 또는 클리어하는 것을 포함하는 비트 설정/클리어 명령어. 예를 들면, 비트 설정/클리어 명령어의 피연산자는 주어진 메모리 시스템 위치의 데이터에서 설정될 또는 클리어될 비트의 포지션을 나타내는 비트 마스크를 명시할 수도 있다. 옵션 사항으로(optionally), 비트 설정/클리어 명령어는 또한, 스왑 변형 및 비교 및 스왑 변형과 유사하게, 후속하는 명령어에 의한 피연산자로서의 사용을 위해, 주어진 메모리 시스템 위치로부터 판독되는 데이터로 하여금 목적지 레지스터로 반환되게 할 수 있다.
- [0050] 데이터 프로세싱 장치의 임의의 주어진 구현예에서 이들 변형 모두가 지원될 필요는 없다는 것이 인식될 것이다.
- [0051] 안티 앨리어싱
- [0052] 상기에서 논의되는 기술은 악의적인 당사자에 의한 공격 하에 있는 오퍼레이팅 시스템에 의한 손상에 대비하여 스테이지 1 변환 테이블 엔트리의 소정의 세트가 보호되는 것을 허용하는 데 유용할 수 있다. 이것은 특정한 가상 어드레스를 특정한 중간 어드레스에 매핑하는 엔트리를 손상에 대비하여 보호할 수도 있다. 그러나, 상이한 가상 어드레스를 동일한 중간 어드레스에 매핑하는 두 개 이상의 상이한 스테이지 1 변환 테이블 엔트리를 제공하는 것이 가능하다. 따라서, 심지어 특정한 스테이지 1 변환 테이블 엔트리가 (부분 판독 전용 퍼미션 및 옵션 사항으로 보호된 엔트리 인코딩의 사용에 의해) 손상에 대비하여 보호되더라도, 공격자가, 보호된 엔트리에 적용되는 보호를 우회하기 위해, 다른 가상 어드레스를 보호된 스테이지 1 변환 테이블 엔트리에 의해 명시되는 동일한 중간 어드레스로 매핑하는 다른 스테이지 1 변환 테이블 엔트리를 설정하려고 시도할 수도 있는 위험이 있을 수 있다. 이 문제를 해결하기 위한 한 가지 접근법은, 스테이지 1 변환 테이블 엔트리를 설정하거나 또는 사용할 때, 그들이 보호된 스테이지 1 변환 테이블 엔트리를 사용하여 이미 매핑된 중간 어드레스를 매핑하지 않는지를 체크하는 체크를 도입하는 것일 수 있지만, 그러나 이것은 성능 및 실제로 구현함에 있어서의 어려움의 관점에서 극도로 비용이 많을 들 것이다.
- [0053] 하기에서 논의되는 예에서, 주어진 중간 어드레스에 대응하는 주어진 스테이지 2 액세스 변환 테이블 엔트리는, 주어진 중간 어드레스에 대응하는 메모리 영역에 대한 액세스가 허용되기 위해서는, 주어진 가상 어드레스로부터 주어진 중간 어드레스로의 스테이지 1 어드레스 매핑을 명시하는 대응하는 스테이지 1 변환 테이블 엔트리를 찾기 위해 스테이지 1 변환 구조의 변환 테이블 워크에서 액세스될 하나 이상의 워킹된(walked) 스테이지 1 변환 테이블 엔트리 각각이 안티 앨리어싱 조건을 충족할 것을 요구받는다는 요건을 나타내는 안티 앨리어싱 특성(property)을, 주어진 중간 어드레스에 대응하는 메모리 영역이 갖는지의 여부를 명시하는 인코딩을 갖는다. 주어진 워킹된 스테이지 1 변환 테이블 엔트리에 대한 안티 앨리어싱 조건은 다음의 경우 중 어느 하나의 경우에 충족되는 것으로 간주된다: (a) 주어진 워킹된 스테이지 1 변환 테이블 엔트리가 보호된 엔트리로서 명시되고 주어진 워킹된 스테이지 1 변환 테이블 엔트리를 저장하는 메모리 시스템 위치의 물리적 어드레스를 유도하기 위해 사용되는 스테이지 2 어드레스 매핑을 제공하는 관련 스테이지 2 변환 테이블 엔트리에 대한 스테이지 2 액세스 퍼미션 정보에 의해 부분 판독 전용 퍼미션이 명시되는 경우, 또는 (b) 판독 전용 퍼미션이 관련 스테이지 2 변환 테이블 엔트리에 대한 스테이지 2 액세스 퍼미션 정보에 의해 명시되는 경우. 메모리 관리 회로부는, 타겟 중간 어드레스에 대응하는 메모리 영역이 안티 앨리어싱 특성을 갖는다는 것 및 하나 이상의 워킹된 스테이지 1 변환 테이블 엔트리 중 임의의 것이 안티 앨리어싱 조건을 충족하지 못한다는 것을 대응하는 스테이지 2 변환 테이블 엔트리가 명시한다는 것을 결정하는 것에 응답하여, 메모리 액세스 요청을 거부한다.

- [0054] 이 접근법은 스테이지 1 변환 테이블 엔트리의 앨리어싱에 기초하여 상기에서 논의되는 공격을 방지한다. 공격에 대비하여 보호될 필요가 있고, 이를 위해 가상 어드레스로부터 그 중간 어드레스로의 매핑을 식별하기 위해 사용되는 스테이지 1 변환 테이블 엔트리가 그 엔트리를 부분 판독 전용 메모리에 저장하는 것 및 그 스테이지 1 변환 테이블 엔트리를 보호된 엔트리로서 인코딩하는 것에 의해 보호되는 민감한 정보에 대응하는 주어진 중간 어드레스가 있는 경우, 스테이지 2 변환 테이블 엔트리를 제어하는 하이퍼바이저는 주어진 중간 어드레스에 대응하는 주어진 스테이지 2 변환 테이블 엔트리를 안티 앨리어싱 특성을 갖는 것으로 또한 마킹할 수 있다. 이것은, 심지어 공격자가 오퍼레이팅 시스템 코드로 하여금 동일한 중간 어드레스에 매핑되는 새로운 스테이지 1 변환 테이블 엔트리를 생성하게 하거나, 또는 상이한 가상 어드레스가 동일한 중간 어드레스로 매핑되게 하도록 변환 테이블 워크를 통한 순회의 경로 상에서 현존하는 스테이지 1 변환 테이블 엔트리를 수정하게 하더라도, 공격자는: (공격자 스테이지 2 액세스 퍼미션을 제어할 수 없기 때문에) 새로운/수정된 스테이지 1 변환 테이블 엔트리로 하여금 메모리의 부분 판독 전용 영역에서 저장되게 하는 것, 및 새로운 엔트리로 하여금 보호된 엔트리로서 인코딩되게 하는 것 둘 모두를 할 수 없을 것이라는 것을 의미한다(공격자에 의해 기록되는 영역이 이미 부분 판독 전용 영역으로서 나타내어진 경우, 공격자는, 예를 들면, 상기에서 언급되는 RCW 명령어를 사용하여 기록을 실행하기 위해 기록 요청의 제한된 서브세트를 사용할 필요가 있을 것이고, 그러한 기록 요청 타입은, 기록되고 있는 값으로 하여금 보호된 스테이지 1 변환 테이블 엔트리가 되게 할 수 없다는 제약이 부과되게 할 수도 있음). 또한, 공격자는 판독 전용 영역에 새로운 앨리어싱 엔트리를 기록할 수 없다. 그러므로, 대응하는 스테이지 2 변환 테이블 엔트리에서 안티 앨리어싱 특성이 정의되는 메모리 영역에 대한 액세스에 대해, (주어진 가상 어드레스로부터 주어진 중간 어드레스로의 스테이지 1 어드레스 매핑에 도달하기 위해 스테이지 1 변환 테이블 구조의 순회에서 워킹될) 워킹된 스테이지 1 변환 테이블 엔트리 각각이 (PRO 또는 판독 전용 퍼미션 중 어느 하나가 스테이지 2에서 설정되는 것, 그리고 PRO가 명시되는 경우 스테이지 1 엔트리가 또한 보호된 엔트리이다는 것에 기초하여) 안티 앨리어싱 조건을 충족하는지의 여부를 체크하는 것에 의해, 이것은 부분 판독 전용 퍼미션 및 보호된 스테이지 1 엔트리 피처를 사용하여 상기에서 논의되는 보호를 우회하는 수단으로서 공격자가 앨리어싱을 사용할 수 있는 것을 방지한다.
- [0055] 상기에서 논의되는 부분 판독 전용 퍼미션 및 보호된 엔트리 인코딩과 유사하게, 안티 앨리어싱 특성은 스테이지 2 액세스 변환 엔트리에 의해 직접적으로 또는 간접적으로 명시될 수 있고, 전용 비트 또는 비트필드 중 어느 하나에 의해 또는 자신의 해석이 프로세싱 회로부의 동작 상태에 의존하는 또는 제어 레지스터에서 명시되는 제어 정보에 의존하는 용도 변경된 비트 또는 비트필드에 의해 나타내어질 수 있다.
- [0056] 스테이지 1 최상위 레벨 기본 어드레스 보호 체크
- [0057] 메모리 관리 회로부는 스테이지 1 변환 테이블 기본 어드레스 레지스터의 값에 기초하여 스테이지 1 변환 테이블 구조의 스테이지 1 최상위 레벨 변환 테이블의 스테이지 1 최상위 레벨 기본 어드레스를 결정할 수 있다. 상기에서 설명되는 피처가 소정의 스테이지 1 변환 테이블 엔트리의 손상을 방지할 수 있지만, 공격자가 이들 보호를 우회하려고 시도할 수도 있는 다른 방식은, 공격자에 의해 정의되는 스테이지 1 변환 테이블의 대안적인 세트 대신 스테이지 1 변환 테이블의 보호된 세트를 대응하기 위해, 승인된 사용을 위해 의도되는 어드레스로부터 상이한 어드레스를 가리키도록 스테이지 1 최상위 레벨 기본 어드레스를 수정하는 것일 수도 있다.
- [0058] 그러한 공격을 단속하기 위한 한 가지 접근법은, 오퍼레이팅 시스템에 의해 요청되는 스테이지 1 변환 테이블 기본 어드레스 레지스터에 대한 임의의 업데이트를 트랩하여, 하이퍼바이저로 하여금 오퍼레이팅 시스템에 의해 요청되는 업데이트가 적절한지의 여부를 체크하게 하는 것일 수 있다. 그러므로, 스테이지 1 변환 테이블 기본 어드레스 레지스터에 대한 업데이트에 대해 임의의 하드웨어 강제 체크를 제공하는 것이 필수적인 것은 아닐 수도 있다.
- [0059] 그러나, 실제로는, 오퍼레이팅 시스템에 의한 스테이지 1 변환 테이블 기본 어드레스 레지스터에 대한 업데이트가 상대적으로 빈번할 수도 있고(예를 들면, 오퍼레이팅 시스템은 각각의 컨텍스트 스위치에서 이것을 행할 수도 있음) 따라서 각각의 업데이트를 하이퍼바이저에 트랩하는 것은 성능 비용을 초래할 수도 있다. 따라서, 일부 추가적인 체크는 성능을 향상시키는 데 유용할 수 있다.
- [0060] 그러한 추가적인 체크의 하나의 예는 스테이지 1 최상위 레벨 테이블 부분 판독 전용 체크일 수도 있는데, 이것은 관련된 스테이지 2 변환 테이블 엔트리가 상기에서 논의되는 부분 판독 전용 퍼미션을 명시하는 메모리 영역에 스테이지 1 최상위 레벨 기본 어드레스가 대응해야 한다는 요건을 강제한다. 그러므로, 스테이지 1 최상위 레벨 테이블 부분 판독 전용 체크가 인에이블되는 경우, 메모리 관리 회로부는, 관련된 스테이지 2 변환 테이블 엔트리가 부분 판독 전용 퍼미션을 명시하지 않는 중간 어드레스에 스테이지 1 최상위 레벨 기본 어드레스가 대

응한다는 것을 결정하는 것에 응답하여, 장애(fault)를 시그널링한다. 예를 들면, 이러한 체크는 스테이지 1 최상위 레벨 기본 어드레스 레지스터에 저장되는 스테이지 1 최상위 레벨 기본 어드레스에 기초하여 변환 테이블 워크를 수행할 때 수행될 수도 있다. 이 체크는, 하이퍼바이저가 오퍼레이팅 시스템에 의해 사용되도록 허락되는 스테이지 1 최상위 레벨 변환 테이블을 유지하는 메모리의 영역(이것은 하이퍼바이저에 의해 이미 안전한 것으로 검증되었을 수도 있음)을 부분 판독 전용으로서 정의할 수 있다는 것을 의미한다. 공격자는 메모리의 일부 다른 영역에서 임의적인 스테이지 1 최상위 레벨 변환 테이블을 정의할 수 없고 그 임의적인 스테이지 1 최상위 레벨 변환 테이블로 하여금 페이지 테이블 워크를 위해 사용되게 할 수 없을 수 있고, 따라서 대응하는 스테이지 2 변환 테이블 엔트리에서 하이퍼바이저에 의해 정의되는 부분 판독 전용 영역에 대한 올바른 기본 어드레스 매핑을 통해 액세스 가능한 올바른 스테이지 1 변환 테이블에 대해 정의되는 보호된 매핑 및 부분 판독 전용 보호를 우회하기 위한 수단으로서 기본 어드레스의 재프로그래밍을 사용할 수 없다.

[0061] 몇몇 구현예에서, 스테이지 1 최상위 레벨 부분 판독 전용 체크는 영구적으로 인에이블되는 것으로 간주될 수도 있고, 그 결과, 이 체크가 인에이블되는지 또는 디스에이블되는지의 여부의 임의의 프로그래밍 가능 제어가 없을 수도 있다.

[0062] 그러나, 다른 구현예에서, 메모리 관리 회로부는, 제어 레지스터에 저장되는 스테이지 1 최상위 레벨 부분 판독 전용 체크 인에이블 제어 값에 기초하여 스테이지 1 최상위 레벨 테이블 부분 판독 전용 체크가 인에이블되는지의 여부를 결정할 수도 있다. 제어 레지스터의 스테이지 1 최상위 레벨 부분 판독 전용 체크 인에이블 제어 값에 기록하는 능력은 임계 레벨의 특권 또는 그 이상에서 실행되는 소프트웨어로 제한될 수도 있다(예를 들면, 임계 레벨의 특권은 앞서 언급되는 사전 결정된 실행 상태와 관련되는 특권보다 더 많은 특권이 있는 하이퍼바이저 레벨의 특권일 수도 있음). 스테이지 1 최상위 레벨 테이블 부분 판독 전용 체크를 인에이블 또는 디스에이블하는 능력을 제공하는 것에 의해, 이것은 부분 판독 전용 퍼미션을 사용하도록 설계되지 않는 레거시 소프트웨어와의 하위 호환성을 허용한다.

[0063] 스테이지 1 변환 테이블 기본 레지스터에 대한 부적절한 업데이트를 단속하기 위한 체크의 다른 예는, 스테이지 1 또는 스테이지 2 변환 테이블 구조 중 어느 하나를 사용하여 메모리 영역에 대해 설정되는 스테이지 1 최상위 레벨 테이블 존재 퍼미션에 기초한 스테이지 1 최상위 레벨 테이블 존재 퍼미션 체크일 수도 있다. 스테이지 1 최상위 레벨 테이블 존재 퍼미션은, 스테이지 1 최상위 레벨 기본 어드레스 레지스터에서의 어드레스에 기초하여 스테이지 1 변환 테이블 워크에서 액세스되는 최상위 레벨의 변환 테이블로서 유효하게 사용될 수 있는 스테이지 1 최상위 레벨 변환 테이블을 저장하기 위해 대응하는 메모리 영역이 할당되는 것이 허용된다는 것을 나타낸다. 그러므로, 스테이지 1 최상위 레벨 테이블 존재 퍼미션 체크가 인에이블되는 경우, 메모리 관리 회로부는, 스테이지 1 최상위 레벨 기본 어드레스에 대응하는 주어진 메모리 영역이, 스테이지 1 변환 테이블 구조의 스테이지 1 최상위 레벨 테이블을 저장하기 위해 주어진 메모리 영역이 할당되는 것이 허용된다는 것을 나타내는 스테이지 1 최상위 레벨 테이블 존재 퍼미션과 관련되는지의 여부를 결정하고, 스테이지 1 최상위 레벨 기본 어드레스에 대응하는 주어진 메모리 영역이 스테이지 1 최상위 레벨 테이블 존재 퍼미션을 갖지 않는 경우 장애를 트리거한다.

[0064] 스테이지 1 최상위 레벨 테이블 존재 퍼미션 체크는, 스테이지 1 변환 테이블 구조의 최상위 레벨 이외의 또 다른 레벨에서 사용되도록 의도되는 추가 레벨 스테이지 1 변환 테이블 엔트리(further-level stage-1 translation table entry) 중 하나를 저장하기 위해 사용되는 메모리 영역을 가리키도록 스테이지 1 최상위 레벨 기본 어드레스 레지스터의 어드레스를 공격자가 업데이트할 수도 있는 또 다른 타입의 공격을 단속하는 데 도움이 된다. 다시 말하면, 최신 변환 테이블 엔트리를 생성하려고 시도하는 대신, 변환 테이블 구조의 나중의 레벨에 있는 엔트리로 하여금 변환 테이블 구조의 이전 레벨에 있는 엔트리로서 취급되게 하는 것에 의해, 공격자는 메모리에 대한 부적절한 액세스를 또한 야기할 수 있고, 그 결과, 그 변환 테이블 엔트리를 선택하기 위해 사용되는 타겟 어드레스의 비트는 비트의 의도된 서브세트가 아니어서(테이블의 상이한 레벨이 입력 어드레스의 비트의 상이한 서브세트에 의해 변환 테이블 워크로 인덱싱됨), 변환 테이블 구조의 나중의 레벨에 있는 부적절하게 액세스된 엔트리에 의해 명시되는 어드레스 매핑 또는 액세스 퍼미션으로 하여금, 변환 테이블 구조를 확립한 당사자에 의해 의도되는 것과는 상이한 가상 어드레스에 적용되게 한다. 최상위 레벨 변환 테이블 엔트리 이외의 스테이지 1 변환 테이블 엔트리를 유지하기 위해 사용되는 메모리 영역이 스테이지 1 최상위 레벨 테이블 존재 퍼미션을 가지지 않도록 정의될 수 있기 때문에, 그러한 공격은 스테이지 1 최상위 레벨 테이블 존재 퍼미션을 사용하여 보호될 수 있고, 따라서 그러한 엔트리를 가리키도록 공격자가 스테이지 1 최상위 레벨 기본 어드레스를 업데이트하는 경우, 그러면, 최상위 레벨 엔트리를 저장하기 위한 퍼미션의 결여가 검출되고, 그러면, 이것은 장애로 하여금 시그널링되게 할 수도 있다.

- [0065] 상이한 메모리 액세스 시나리오에서의 사용을 위해 적어도 두 개의 스테이지 1 최상위 레벨 기본 어드레스 레지스터가 제공되어 스테이지 1 최상위 레벨 기본 어드레스를 저장하는 몇몇 구현예에서, 스테이지 1 최상위 레벨 테이블 존재 퍼미션의 적어도 두 가지 변형이 지원될 수 있는데, 각각의 변형은 복수의 스테이지 1 최상위 레벨 기본 어드레스 레지스터의 각각의 서브세트에 대응한다. 주어진 메모리 영역에 대해 스테이지 1 최상위 레벨 테이블 존재 퍼미션의 주어진 변형이 명시되는 경우, 대응하는 메모리 영역은 주어진 변형에 대한 스테이지 1 최상위 레벨 기본 어드레스 레지스터의 대응하는 서브세트 중 하나를 사용하여 액세스되는 스테이지 1 최상위 레벨 변환 테이블을 저장하기 위해 할당되는 것이 허용되는 것으로 나타내어지지만, 그러나 주어진 변형에 대한 대응하는 서브세트의 멤버가 아닌 스테이지 1 최상위 레벨 기본 어드레스 레지스터를 사용하여 액세스되는 스테이지 1 최상위 레벨 변환 테이블을 저장하기 위해 할당되는 것이 허용되지는 않는다. 그러므로, 스테이지 1 최상위 레벨 기본 어드레스 레지스터의 대응하는 서브세트가 특정한 스테이지 1 최상위 레벨 기본 어드레스 레지스터를 포함하지 않는 스테이지 1 최상위 레벨 테이블 존재 퍼미션을 할당받는 메모리 영역의 어드레스를 명시하는 특정한 스테이지 1 최상위 레벨 기본 어드레스 레지스터를 사용하여 페이지 테이블 워크를 수행하려는 시도가 있는 경우, 장애가 트리거될 수도 있다. 기본 어드레스 레지스터의 각각의 특정한 서브세트에 대해 스테이지 1 최상위 레벨 변환 테이블에 대한 대응하는 메모리 영역을 사용할 퍼미션을 나타내는 상이한 스테이지 1 최상위 레벨 테이블 존재 퍼미션을 지원하는 것에 의해, 이것은 더욱 정밀한 제어를 허용하는데, 예를 들면, 그 결과, 기본 어드레스 레지스터 중 하나에서의 사용을 위해 의도되는 기본 어드레스가 다른 기본 어드레스 레지스터에 대해 사용되는 것이 방지될 수 있다.
- [0066] 예를 들면, 제1 스테이지 1 최상위 레벨 기본 어드레스 레지스터 및 제2 스테이지 1 최상위 레벨 기본 어드레스 레지스터를 지원하는 구현예에서, 스테이지 1 최상위 레벨 테이블 존재 퍼미션의 변형은 다음의 것 중 적어도 두 개를 포함할 수 있다:
- 대응하는 서브세트가 제1 스테이지 1 최상위 레벨 기본 어드레스 레지스터를 포함하지만 그러나 제2 스테이지 1 최상위 레벨 기본 어드레스 레지스터를 포함하지 않는 제1 변형;
 - 대응하는 서브세트가 제2 스테이지 1 최상위 레벨 기본 어드레스 레지스터를 포함하지만 그러나 제1 스테이지 1 최상위 레벨 기본 어드레스 레지스터를 포함하지 않는 제2 변형; 및
 - 대응하는 서브세트가 제1 및 제2 스테이지 1 최상위 레벨 기본 어드레스 레지스터 둘 모두를 포함하는 제3 변형.
- [0070] 하나의 예에서, 주어진 가상 어드레스에 대해 스테이지 1 페이지 테이블 워크를 수행하는 경우, 메모리 관리 회로부는, 주어진 가상 어드레스의 최상위 비트의 부분이 모두 제로인지 또는 아닌지의 여부에 기초하여, 스테이지 1 최상위 레벨 변환 테이블에 액세스하기 위해 제1 및 제2 스테이지 1 최상위 레벨 기본 어드레스 레지스터 중 어떤 것을 사용할지를 선택할 수 있다. 이 선택을 위해 고려되는 최상위 비트의 수는 고정될 수 있거나, 또는 소프트웨어에 의해 구성되는 제어 값을 사용하여 프로그래밍 가능할 수 있다. 이 접근법은, 예를 들면, 변환 테이블의 상이한 세트를 사용하여 가상 어드레스 공간의 상위 및 하위 범위가 변환되는 것을 허용하는 데 유용할 수 있다. 예를 들면, 어드레스의 최상위 비트가 모두 제로로 설정되는 하위 어드레스 범위는 유저 어드레스 공간을 위해 사용될 수 있고 어드레스의 최상위 부분에서 적어도 하나의 нену로 비트를 갖는 상위 어드레스 범위는 커널 어드레스 공간을 위해 사용될 수 있다.
- [0071] 주어진 메모리 영역에 대한 스테이지 1 최상위 레벨 테이블 존재 퍼미션이 그 주어진 메모리 영역에 대응하는 스테이지 1 변환 테이블 엔트리의 스테이지 1 액세스 퍼미션 내에서 인코딩되는 것이 가능할 수도 있지만, 그러나 그 경우, 스테이지 1 최상위 레벨 테이블 존재 퍼미션을 인코딩하기 위해 사용되는 비트가 오퍼레이팅 시스템 레벨의 특권을 갖는 실행 상태에서 실행되는 명령어에 의해 수정되는 것을 방지하기 위해 적어도 하나의 추가적인 보호 메커니즘을 구현하는 것이 바람직할 수도 있다.
- [0072] 대신, 스테이지 1 최상위 레벨 기본 어드레스에 대응하는 관련된 스테이지 2 변환 테이블 엔트리의 인코딩에 기초하여 주어진 메모리 영역이 스테이지 1 최상위 레벨 테이블 존재 퍼미션과 관련되는지의 여부를 메모리 관리 회로부가 결정하는 것이 더욱 편리할 수도 있다. 관련된 스테이지 2 변환 테이블 엔트리를 사용하여 주어진 메모리 영역이 스테이지 1 최상위 레벨 변환 테이블을 유지하는 것이 허용되는지의 여부를 나타내는 것에 의해, 이것은 일반적으로 스테이지 2 변환 테이블 엔트리를 업데이트할 수 없을 오퍼레이팅 시스템 코드에 의해 트리거되는 스테이지 1 변환 테이블 엔트리에 대한 업데이트에 대한 제어를 단순화한다.

- [0073] 다시 말하지만, 몇몇 구현예에서, 스테이지 1 최상위 레벨 테이블 존재 퍼미션 체크는 영구적으로 인에이블되는 것으로 간주될 수 있다.
- [0074] 대안적으로, 메모리 관리 회로부는, 제어 레지스터에 저장되는 스테이지 1 최상위 레벨 테이블 존재 퍼미션 체크 인에이블 제어 값에 기초하여 스테이지 1 최상위 레벨 테이블 존재 퍼미션 체크가 인에이블되는지의 여부를 결정할 수도 있다. 제어 레지스터의 스테이지 1 최상위 레벨 테이블 존재 퍼미션 체크 인에이블 제어 값에 기록하는 능력은 임계 레벨의 특권 또는 그 이상(예를 들면, 임계 레벨의 특권은 하이퍼바이저 레벨의 특권일 수도 있음)에서 실행되는 소프트웨어로 제한될 수도 있다. 스테이지 1 최상위 레벨 테이블 부분 판독 전용 체크를 인에이블 또는 디스에이블하는 능력을 제공하는 것에 의해, 이것은 스테이지 1 최상위 레벨 테이블 존재 퍼미션을 사용하지 않는 레거시 소프트웨어와의 하위 호환성을 제공한다.
- [0075] 또한, 상기에서 논의되는 바와 같이 기본 어드레스 레지스터의 상이한 서브세트에 대응하여 지원되는 스테이지 1 최상위 레벨 테이블 존재 퍼미션의 두 개 이상의 변형이 있는 구현예에서는, 제어 레지스터에 저장되는 적어도 하나의 제어 값에 의해 정의되는 적어도 하나의 추가적인 제어가 있을 수 있는데, 제어 레지스터에 저장되는 적어도 하나의 제어 값은, 스테이지 1 최상위 레벨 테이블 존재 퍼미션의 상이한 변형이 스테이지 1 최상위 레벨 기본 어드레스 레지스터의 대응하는 서브세트에 대한 대응하는 메모리 영역의 사용을 인에이블하는 것으로만 처리되어야 하는지의 여부, 또는 대응하는 메모리 영역이 스테이지 1 최상위 레벨 기본 어드레스 레지스터 중 임의의 것에 대해 사용되는 것을 가능하게 하기 위해 각각의 변형이 사용될 수 있는지의 여부를 나타낸다. 그러므로, 스테이지 1 최상위 레벨 기본 어드레스를 제공하기 위해 사용되는 기본 어드레스 레지스터가 스테이지 1 최상위 레벨 기본 어드레스를 포함하는 메모리 영역에 대해 명시되는 스테이지 1 최상위 레벨 테이블 존재 퍼미션의 변형에 의해 인에이블되는 기본 어드레스 레지스터의 서브세트 중 하나가 아닌 경우 트리거되는 상기에서 설명되는 장애는, 적어도 하나의 추가적인 제어에 기초하여 인에이블될 수 있거나 또는 디스에이블될 수 있다. 몇몇 예에서, 사용되는 스테이지 1 최상위 레벨 기본 어드레스 레지스터 및 스테이지 1 최상위 레벨 테이블 존재 퍼미션의 변형의 특정한 조합에 대해 장애가 생성되는지의 여부를 구성하기 위해 별개의 제어가 제공될 수 있다. 예를 들면, 기본 어드레스를 포함하는 메모리 영역이 상기에서 설명되는 스테이지 1 최상위 레벨 테이블 존재 퍼미션의 제1 변형을 갖는 경우 제2 스테이지 1 최상위 레벨 기본 어드레스 레지스터의 기본 어드레스를 사용하는 페이지 테이블 워크에 대해 장애가 생성되는지의 여부를 제어하기 위해 하나의 제어 값이 제공될 수도 있고, 기본 어드레스를 포함하는 메모리 영역이 스테이지 1 최상위 레벨 테이블 존재 퍼미션의 제2 변형을 갖는 경우 제1 스테이지 1 최상위 레벨 기본 어드레스 레지스터의 기본 어드레스를 사용하는 페이지 테이블 워크에 대해 장애가 생성되는지의 여부를 제어하기 위해 제2 제어 값이 제공될 수도 있다..
- [0076] 스테이지 1 최상위 레벨 테이블 부분 판독 전용 체크 및 스테이지 1 최상위 레벨 테이블 존재 퍼미션 체크 둘 모두를 지원하는 몇몇 예에서, 별개의 인에이블 제어 값을 사용하여, 이들 체크를 개별적으로 인에이블 또는 디스에이블하는 것이 가능할 수도 있다. 대안적으로, 단일의 제어 값이 체크 둘 모두를 함께 인에이블/디스에이블할 수 있고, 그 결과, 지원되는 옵션은, 체크 둘 모두가 인에이블된다는 것 또는 체크 둘 모두가 디스에이블된다는 것 중 어느 하나일 수도 있지만, 그러나 하나의 체크를 인에이블하고 다른 것을 인에이블하지 않는 것은 가능하지 않을 수도 있다.
- [0077] 다른 구현예는 두 가지 타입 모두의 체크를 지원하지 않을 수도 있다. 예를 들면, 몇몇 구현예에서, 이들 체크 중 하나 또는 다른 것은 충분한 것으로 간주될 수도 있다.
- [0078] 부분 판독 전용 퍼미션에 관해서는, 스테이지 1 최상위 레벨 테이블 존재 퍼미션은, 예를 들면, 변환 테이블 엔트리의 인코딩에서 직접적으로 명시되는 정보 및/또는 퍼미션 레지스터를 참조하여 변환 테이블 엔트리에 의해 간접적으로 명시되는 정보를 사용하여, 변환 테이블 엔트리에 의해 직접적으로 또는 간접적으로(또는 둘 모두의 조합으로) 명시될 수 있다.
- [0079] 아키텍처 시뮬레이션
- [0080] 상기에서 논의되는 기술은 상기에서 논의되는 명령어 디코더 및 프로세싱 회로부를 구현하기 위해 제공되는 하드웨어 회로부를 구비하는 데이터 프로세싱 장치 내에서 구현될 수도 있다.
- [0081] 그러나, 동일한 기술은 타겟 코드의 실행을 위한 명령어 실행 환경을 제공하기 위해 호스트 데이터 프로세싱 장치 상에서 실행되는 컴퓨터 프로그램 내에서도 또한 구현될 수 있다. 그러한 컴퓨터 프로그램은, 소정의 명령어 세트 아키텍처에 따라 타겟 코드를 실제로 지원하는 하드웨어 장치 상에서 제공될 아키텍처 환경을, 심지어 호스트 데이터 프로세싱 장치 그 자체가 그 아키텍처를 지원하지 않더라도, 시뮬레이팅하도록 호스트 데이터 프

로세싱 장치를 제어할 수도 있다. 컴퓨터 프로그램은, 상기에서 논의되는 부분 관독 전용 퍼미션 및 다양한 체크에 대한 지원을 비롯하여, 상기에서 논의되는 프로세싱 회로부 및 메모리 관리 회로부의 기능을 에뮬레이팅하는 메모리 관리 프로그램 로직 및 프로세싱 프로그램 로직을 가질 수도 있다. 그러한 시뮬레이션 프로그램은, 예를 들면, 하나의 명령어 세트 아키텍처에 대해 작성되는 레거시 코드가 상이한 명령어 세트 아키텍처를 지원하는 호스트 프로세서 상에서 실행되고 있을 때, 유용할 수 있다. 또한, 시뮬레이팅된 실행 환경 상에서의 소프트웨어의 실행이 새로운 아키텍처를 지원하는 하드웨어 디바이스의 지속적인 개발과 병행하여 소프트웨어의 테스트를 가능하게 할 수 있기 때문에, 시뮬레이션은 새로운 아키텍처 버전을 지원하는 프로세싱 하드웨어가 준비되기 이전에 새로운 버전의 명령어 세트 아키텍처에 대한 소프트웨어 개발이 시작되는 것을 허용할 수 있다. 시뮬레이션 프로그램은 비밀시적 저장 매체일 수도 있는 저장 매체 상에서 저장될 수도 있다.

[0082] 데이터 프로세싱 장치의 구체적인 예

[0083] 도 1은 데이터 프로세싱 장치(2)의 예를 개략적으로 예시한다. 데이터 프로세싱 장치는 다수의 파이프라인 스테이지를 포함하는 프로세싱 파이프라인(4)을 갖는다. 이 예에서, 파이프라인 스테이지는 명령어 캐시(8)로부터 명령어를 페치하기(fetching) 위한 페치 스테이지(fetch stage; 6); 파이프라인의 나머지 스테이지에 의해 프로세싱될 마이크로 동작(micro-operation)(디코딩된 명령어)을 생성하기 위해 페치된 프로그램 명령어를 디코딩하기 위한 디코드 스테이지(10); 마이크로 동작에 필요한 피연산자가 레지스터 파일(14)에서 이용 가능한지의 여부를 체크하기 위한 그리고 일단 주어진 마이크로 동작을 위한 필요한 피연산자가 이용 가능하면 실행을 위해 마이크로 동작을 발행하기 위한 발행 스테이지(12); 레지스터 파일(14)로부터 관독되는 피연산자를 프로세싱하여 결과 값을 생성하는 것에 의해, 마이크로 동작에 대응하는 데이터 프로세싱 동작을 실행하기 위한 실행 스테이지(16); 및 프로세싱의 결과를 레지스터 파일(14)에 다시 기록하기 위한 라이트백 스테이지(writeback stage; 18)를 포함한다. 이것은 가능한 파이프라인 아키텍처의 단지 하나의 예에 불과하며, 다른 시스템은 추가적인 스테이지 또는 상이한 구성의 스테이지를 가질 수도 있다는 것이 인식될 것이다. 예를 들면, 비순차적 프로세서에서, 프로그램 명령어 또는 마이크로 동작에 의해 명시되는 아키텍처 레지스터를 레지스터 파일(14)의 물리적 레지스터를 식별하는 물리적 레지스터 명시자에 매핑하기 위해 레지스터 리네이밍 스테이지(register renaming stage)가 포함될 수 있다. 몇몇 예에서, 디코드 스테이지(10)에 의해 디코딩되는 프로그램 명령어와 실행 스테이지에 의해 프로세싱되는 대응하는 마이크로 동작 사이에는 일 대 일 관계가 있을 수도 있다. 프로그램 명령어와 마이크로 동작 사이에는 일 대 다 또는 다 대 일 관계가 있는 것이 또한 가능하며, 그 결과, 예를 들면, 단일의 프로그램 명령어가 두 개 이상의 마이크로 동작으로 분할될 수도 있거나, 또는 두 개 이상의 프로그램 명령어가 융합되어 단일의 마이크로 동작으로서 프로세싱될 수도 있다.

[0084] 실행 스테이지(16)는 상이한 클래스의 프로세싱 동작을 실행하기 위한 다수의 프로세싱 유닛을 포함한다. 예를 들면, 실행 유닛은 레지스터(14)로부터 관독되는 스칼라 피연산자에 대한 산술 또는 논리 연산을 수행하기 위한 스칼라 산술/로직 유닛(arithmetic/logic unit; ALU)(20); 부동 소수점 값에 대한 연산을 수행하기 위한 부동 소수점 유닛(22); 분기 동작의 결과를 평가하고 그에 따라 현재의 실행 지점을 나타내는 프로그램 카운터를 조정하기 위한 분기 유닛(24); 및 메모리 시스템(8, 30, 32, 34)의 데이터에 액세스하기 위해 로드/저장 동작을 수행하기 위한 로드/저장 유닛(26)을 포함할 수도 있다. 메모리 관리 회로부의 한 예인 메모리 관리 유닛(memory management unit; MMU)(28)은 데이터 액세스 명령어의 피연산자 및 메모리 시스템에서의 데이터의 보관 위치를 식별하는 물리적 어드레스에 기초하여 로드/저장 유닛(26)에 의해 명시되는 가상 어드레스 사이의 어드레스 변환을 수행하기 위해 제공된다. MMU는 메모리 시스템에 저장되는 페이지 테이블로부터 어드레스 변환 데이터를 캐싱하기 위한 변환 색인 버퍼(translation lookaside buffer; TLB)(29)를 구비하는데, 여기서 페이지 테이블의 페이지 테이블 엔트리는, 예를 들면, 파이프라인 상에서 실행되는 주어진 프로세스가 주어진 메모리 영역으로부터 명령어를 관독, 기록 또는 실행하는 것이 허용되는지의 여부를 지배하는 액세스 퍼미션 및 어드레스 변환 매핑을 정의한다.

[0085] 이 예에서, 메모리 시스템은 레벨 1 데이터 캐시(30), 레벨 1 명령어 캐시(8), 공유 레벨 2 캐시(32) 및 메인 시스템 메모리(34)를 포함한다. 이것은 가능한 메모리 계층 구조의 하나의 예에 불과하며 다른 배열의 캐시가 제공될 수 있다는 것이 인식될 것이다. 실행 스테이지(16)에서 도시되는 특정한 타입의 프로세싱 유닛(20 내지 26)은 단지 하나의 예에 불과하며, 다른 구현에는, 동일한 타입의 다수의 마이크로 동작이 병렬로 핸들링될 수 있도록, 프로세싱 유닛의 상이한 세트를 구비할 수도 있거나 또는 동일한 타입의 프로세싱 유닛의 다수의 인스턴스를 포함할 수 있다. 도 1은 단지 가능한 프로세서 파이프라인 구현예의 일부 컴포넌트의 단순화된 표현에 불과하며, 프로세서는 간결성을 위해 예시되지 않은 많은 다른 엘리먼트를 포함할 수도 있다는 것이 인식될 것이다. 도 1이 메모리(34)에 대한 액세스를 갖는 단일의 프로세서 코어를 도시하지만, 장치(2)는, 각각의 코어

가 각각의 캐시(8, 30, 32)를 가지면서, 메모리(34)에 대한 액세스를 공유하는 하나 이상의 추가적인 프로세서 코어를 또한 가질 수 있다.

[0086] 도 2는, 명령어를 실행할 때 프로세싱 회로부(4)가 동작할 수 있는 상이한 실행 상태(예외 레벨로서 또한 지칭됨)를 예시하는 다이어그램이다. 이 예에서는, 네 개의 예외 레벨(EL0, EL1, EL2, EL3)이 있는데, 여기서 예외 레벨(EL0)은 최소의 특권이 있는 예외 레벨이고 예외 레벨(EL3)은 가장 많은 특권이 있는 예외 레벨이다. 일반적으로, 더 많은 특권이 있는 예외 레벨에서 실행될 때, 프로세싱 회로부는 더 낮은, 더 적은 특권이 있는 예외 레벨에 액세스할 수 없는 일부 기억 위치 또는 레지스터(14)에 대한 액세스를 가질 수도 있다.

[0087] 이 예에서, 예외 레벨(EL0)은 예외 레벨(EL1)에서 실행되는 대응하는 오퍼레이팅 시스템 또는 가상 머신에 의해 관리되는 애플리케이션을 실행하기 위한 것이다. 다수의 가상 머신이 동일한 물리적 플랫폼 상에서 공존하는 경우, 그러면, EL2에서 동작되는 하이퍼바이저가 제공되어, 각각의 가상 머신을 관리할 수도 있다. 하이퍼바이저가 가상 머신을 관리하고 가상 머신이 애플리케이션을 관리하는 예를 도 2가 도시하지만, 하이퍼바이저가 EL0에서 애플리케이션을 직접적으로 관리하는 것도 또한 가능하다.

[0088] 필수적인 것은 아니지만, 몇몇 구현에는 프로세싱 회로부에 대해 별개의 하드웨어로 분할된 보안 및 비보안 동작 도메인을 구현할 수도 있다. 데이터 프로세싱 시스템(2)은, 보안 도메인에서 동작하는 소프트웨어 프로세스와 관련되는 데이터 및 코드가 비보안 도메인에서 동작하는 프로세스에 의한 액세스로부터 격리되는 것을 보장하기 위해, 프로세서 및 메모리 시스템 내에 구현되는 하드웨어 피처를 가질 수도 있다. 예를 들면, 영국(UK) 케임브리지 소재의 Arm® Limited(암 리미티드)에 의해 제공되는 TrustZone®(트러스트존) 아키텍처와 같은 하드웨어 아키텍처가 사용될 수도 있다. 대안적으로, 다른 하드웨어 강제 보안 구현화 아키텍처가 사용될 수 있다. 보안 애플리케이션(실행되는 서비스)은 보안 도메인의 예외 레벨(EL0)에서 동작될 수도 있고 보안(실행되는) 오퍼레이팅 시스템 또는 가상 머신은 보안 도메인의 예외 레벨(EL1)에서 동작될 수도 있다. 몇몇 구현예에서, 보안 상태에서 EL2에 대한 지원이 없으며 하이퍼바이저는 오로지 비보안 EL2에서만 실행될 수도 있다. 다른 구현예에서, 도 2의 별표에 의해 나타내어지는 바와 같이 보안 EL2에서 실행되는 보안 하이퍼바이저에 대한 지원이 있을 수도 있다. 몇몇 예에서, 예외 레벨(EL3)에서 실행되는, 비보안 도메인과 보안 도메인 사이의 전이(transition)를 관리하기 위한 보안 모니터 프로그램이 제공될 수도 있다. 다른 구현예는, 보안 모니터 프로그램이 필요로 되지 않을 수도 있도록, 보안 도메인 사이의 전이를 하드웨어에서 단속할 수 있다.

[0089] 어드레스 변환

[0090] MMU(28)에 의해 수행되는 하나의 태스크는 가상 어드레스(virtual address; VA)와 물리적 어드레스(physical address; PA) 사이의 어드레스 변환이다. 프로세싱 회로부(4) 상에서 실행되는 소프트웨어는 가상 어드레스를 사용하여 기억 위치를 명시하지만, 그러나 이들 가상 어드레스는 MMU(28)에 의해 액세스할 메모리 시스템 위치를 식별하는 물리적 어드레스로 변환될 수 있다. 가상 어드레스를 사용하는 것의 이점은, 관리 소프트웨어, 예컨대 오퍼레이팅 시스템(Operating System; OS)이 소프트웨어에 제공되는 메모리의 모습을 제어하는 것을 허용한다는 것이다. OS는, 어떤 메모리가 보이는지, 그 메모리가 보이는 가상 어드레스, 및 그 메모리에 어떤 액세스가 허락되는지를 제어할 수 있다. 이것은, OS가 애플리케이션을 샌드박스화하는(sandbox) 것(하나의 애플리케이션의 리소스를 다른 애플리케이션에게 숨김) 및 기저의 하드웨어로부터 추상화를 제공하는 것을 허용한다. 가상 어드레스를 사용하는 것의 다른 이점은, OS가 메모리의 다수의 조각의 물리적 영역을 단일의 연속적인 가상 어드레스 공간으로서 애플리케이션에 제공할 수 있다는 것이다. 가상 어드레스는, 그들의 애플리케이션을 작성할 때 시스템의 정확한 메모리 어드레스를 알지 못할 소프트웨어 개발자에게도 또한 도움이 된다. 가상 어드레스를 통해, 소프트웨어 개발자는 그들 자신이 물리적 메모리에 관여할 필요가 없다. 애플리케이션은, 어드레스 변환을 수행하는 것이 함께 작동할 OS 및 하드웨어에 달려 있다는 것을 알고 있다.

[0091] 실제로, 각각의 애플리케이션은 물리적 시스템의 상이한 위치로 매핑될 그 자신의 가상 어드레스의 세트를 사용할 수 있다. 오퍼레이팅 시스템이 상이한 애플리케이션 사이를 전환할 때, 그것은 맵을 다시 프로그래밍한다. 이것은 본 출원의 가상 어드레스가 메모리의 올바른 물리적 위치로 매핑될 것이라는 것을 의미한다.

[0092] 가상 어드레스는 매핑을 통해 물리적 어드레스로 변환된다. 가상 어드레스와 물리적 어드레스 사이의 매핑은 변환 테이블(페이지 테이블로서 종종 지칭됨)에 저장된다. 변환 테이블은 메모리에 저장되며 소프트웨어, 통상적으로 OS 또는 하이퍼바이저에 의해 관리된다. 변환 테이블은 정적이지 않으며, 소프트웨어의 필요성이 변경됨에 따라 테이블은 업데이트될 수 있다. 이것은 가상 어드레스와 물리적 어드레스 사이의 매핑을 변경한다.

[0093] 프로세싱 회로부(4)가 실행 상태의 소정의 서브세트에 있을 때(특히, 프로세싱 회로부(4)가 비보안 EL0 또는 비

보안 EL1에 있을 때) 수행되는 메모리 액세스의 경우, 도 3에서 도시되는 바와 같이 2 스테이지 어드레스 변환이 사용된다(다른 실행 상태의 경우, 스테이지 1 페이지 테이블을 사용하는 어드레스 변환의 하나의 스테이지가 충분함). 그러므로, 비보안 EL0 및 비보안 EL1로부터의 가상 어드레스는 테이블의 두 세트를 사용하여 변환된다. 이들 테이블은 가상화를 지원하며, 주어진 가상 머신(virtual machine; VM)(게스트 오퍼레이팅 시스템에 대응하는 가상 머신 및 그 게스트 오퍼레이팅 시스템에 의해 제어되는 애플리케이션)에 의해 보이는 물리적 메모리의 모습을 하이퍼바이저가 가상화하는 것을 허용한다. OS에 의해 제어되는 변환의 세트를 스테이지 1로 칭한다. 스테이지 1 테이블은 가상 어드레스를 중간 물리적 어드레스(intermediate physical address; IPA - 앞서 언급된 중간 어드레스의 한 예)로 변환한다. 스테이지 1에서, OS는 IPA가 물리적 어드레스 공간이라고 생각한다. 그러나, 하이퍼바이저는 스테이지 2로 칭해지는 변환의 제2 세트를 제어한다. 이 변환의 제2 세트는 IPA를 물리적 어드레스로 변환한다.

[0094] 스테이지 1 및 스테이지 2 변환 테이블은, 스테이지 1 및 스테이지 2에 대해 도 4 및 도 5에서 각각 도시되는 바와 같이, 다수의 레벨의 변환 테이블을 포함하는 계층적 테이블 구조로서 구현된다. 이 예에서, 스테이지 1 및 스테이지 2 테이블 둘 모두는 최대 4 개 레벨의 페이지 테이블, 즉, 레벨 0(L0), 레벨 1(L1), 레벨 2(L2) 및 레벨 3(L3)을 가질 수 있다.

[0095] 주어진 어드레스에 대한 물리적 어드레스 매핑을 찾아내기 위해, 하나 이상의 변환 테이블 lookups을 포함하는 변환 테이블 워크가 수행된다. 변환 테이블 워크는 가상 어드레스를 물리적 어드레스로 변환하는 데 필요한 lookups의 세트이다. 비보안 EL1&0 변환 체제(regime)의 경우, 이 세트는 스테이지 1 변환 및 스테이지 2 변환 둘 모두에 대한 lookups을 포함한다(하기에서 도시되는 도 6 참조). 스테이지 1 및 스테이지 2 lookups을 사용하여 성공적인 변환 테이블 워크에 의해 반환되는 정보는 다음의 것이다:

[0096]  필요한 물리적 어드레스(중간 어드레스로의 스테이지 1 매핑 및 물리적 어드레스로의 스테이지 2 매핑에 기초하여 변환됨).

[0097]  타겟 메모리 영역에 대한 액세스 퍼미션 및/또는 메모리 속성(attribute), 이것은 그 메모리 영역에 대한 액세스를 제어하는 방법에 관한 정보를 제공함. 이들은 스테이지 1 테이블 구조에서 정의되는 스테이지 1 액세스 퍼미션 및/또는 속성 및 스테이지 2 테이블 구조에서 정의되는 스테이지 2 액세스 퍼미션 및/또는 속성을 포함할 수도 있다.

[0098] 스테이지 1 및 스테이지 2 구조 중 주어진 구조를 순회하기 위해, 워크는, 변환 테이블 기본 어드레스 레지스터에서 명시되는 어드레스(스테이지 1의 경우 TTBR, 스테이지 2의 경우 VTTBR_EL2)에 기초하여, 초기 lookups을 위한 최상위 레벨(L0) 변환 테이블의 판독으로 시작된다. 각각의 변환 테이블 lookup은 다음의 것 중 하나를 나타내는 디스크립터를 반환한다:

[0099]  엔트리는, 추궁되고 있는 어드레스 매핑을 제공하는 스테이지 1 또는 스테이지 2 구조의 순회의 최종 엔트리이다. 엔트리가 최종 L3에 있는 경우, 이 엔트리는 페이지 디스크립터(D_Page)로 칭해지며, 한편 워크의 최종 엔트리를 제공하는 엔트리가 더 높은 레벨 중 하나에 있는 경우, 그것은 블록 디스크립터(D_Block)로 칭해진다. 순회의 최종 엔트리는 출력 어드레스(output address; OA - 즉, 스테이지 1의 경우 IPA 또는 스테이지 2의 경우 PA), 및 액세스에 대한 퍼미션 및 속성을 포함한다. 블록 디스크립터가 변환 테이블 구조의 더 높은 레벨에서 발견되는 경우, 이것은 블록 디스크립터가 L3에서 단일의 엔트리에 의해 표현되는 4kB 메모리 페이지보다 더 큰 사이즈의 메모리 영역을 나타낸다는 것을 의미한다(L1 및 L2에서 블록 디스크립터에 의해 표현되는 특정한 사이즈는 L1 또는 L2 테이블로 인덱싱하기 위해 사용되는 인덱스 비트의 수에 의존함 - 이 예에서 L1 및 L2 블록 디스크립터는 1GB 및 2MB 영역을 각각 나타냄).

[0100]  추가적인 레벨의 lookups이 필요로 된다. 이 경우, 엔트리는 테이블 디스크립터(D_Table)로 칭해지는데, 그 이유는, 그것이 테이블의 추가적인 레벨에서 그 lookups을 위한 변환 테이블 기본 어드레스를 제공하기 때문이다. 테이블 디스크립터는 옵션 사항으로 최종 변환에 적용될 수 있는 다른 계층적 속성을 또한 제공할 수 있다. 레벨 1 및 2에서의 변환 테이블 엔트리의 인코딩은 블록 디스크립터를 테이블 디스크립터와 구별한다.

[0101]  디스크립터는 무효하다. 이 경우, 메모리 액세스는 변환 장애를 생성한다.

[0102] 도 4는 테이블 lookups을 위한 입력 어드레스로서 제공되는 가상 어드레스의 각각의 비트를 사용한 스테이지 1 변환 테이블의 인덱싱을 예시한다. 최상위 레벨 테이블 L0의 기본 어드레스는 TTBR로부터 판독되고, L1, L2, L3

테이블의 기본 어드레스는 L0, L1, L2 테이블의 인덱싱된 테이블 디스크립터에 저장되는 어드레스에 의해 각각 표시된다(L1 또는 L2 테이블에서 어떠한 블록 디스크립터도 식별되지 않는 경우 - 블록 디스크립터가 L1 또는 L2의 인덱싱된 엔트리에서 발견되는 경우 - 그러면, 출력 어드레스 매핑이 이미 발견되었기 때문에 순회는 그 레벨에서 중지된다). 스테이지 1 변환 테이블의 주어진 레벨 내에서 선택할 특정한 엔트리는 룩업을 위한 입력 어드레스로서 제공되는 가상 어드레스의 비트의 소정의 서브세트에 대응하는 인덱스 값(a, b, c, d)에 기초하여 결정된다. 도 4는 특정한 예에서 각각의 인덱스 값(a, b, c, d)에 대해 입력 어드레스의 어떤 비트가 사용되는지를 예시한다. 주어진 테이블의 관련 엔트리의 어드레스는 이전 레벨에서 테이블 디스크립터에서 명시되는 어드레스 또는 TTBR에 기초하여 결정되는 바와 같은 그 주어진 테이블의 기본 어드레스에 인덱스 비트(a, b, c 또는 d)의 배수를 추가하는 것에 의해 획득된다(인덱스 값에 적용되는 승수는 하나의 변환 테이블 엔트리의 사이즈에 대응함).

[0103] 유사하게, 도 5는 스테이지 2 테이블 룩업을 위한 입력 어드레스로서 제공되는 중간 어드레스의 각각의 비트를 사용한 스테이지 2 변환 테이블의 인덱싱을 예시한다. 인덱싱은 스테이지 1에 대한 도 4에서 도시되는 것과 유사하지만, 그러나 상이한 기본 어드레스 레지스터(VTTBR_EL2)를 사용하여 L0 테이블의 기본 어드레스를 제공한다. 도 5의 예에서 도시되는 바와 같이, 스테이지 2 룩업의 경우, 룩업이 L0 또는 L1 중 어느 하나에서 시작되어야 한다는 것을 명시할 수 있는 제어 레지스터(VTCR_EL2.SL0)에 저장되는 값에 기초하여, 스테이지 2 변환 테이블의 워크가 시작되는 시작 레벨을 변경하는 것이 가능하다. 스테이지 2 룩업이 L0에서 시작되는 경우, 그러면, 레벨 0, 1, 2, 3에 대한 인덱싱은 스테이지 1에 대한 도 4와 각각 유사한 인덱스 값(a, b1, c, d)을 사용한다. 스테이지 2 룩업이 L1에서 시작되는 경우, 그러면, 인덱싱은 유사한 방식으로 수행되지만, 그러나 이번에는 도 5에서 도시되는 바와 같이 룩업의 최상위 레벨(L1)에서 더 많은 수의 인덱스 비트(b2)가 사용된다. 가변 시작 레벨을 제공하는 것은 필수적인 피처가 아니며 소망되는 경우 생략될 수 있다. 도 4에서 도시되지 않지만, 스테이지 1에서 룩업을 위한 가변 시작 레벨을 제공하는 것도 또한 가능할 것이다.

[0104] 도 6에서 도시되는 바와 같이, 실제로, 스테이지 1 및 스테이지 2 변환 둘 모두를 포함하는 전체 변환 테이블 워크가 수행되는 경우, 그러면, TTBR로부터 획득되는 각각의 스테이지 1 테이블 기본 어드레스 및 스테이지 1 L0, L1, L2 변환 테이블에서 액세스되는 테이블 디스크립터는, 그 자체가 스테이지 2 변환 테이블을 사용하여 변환하는 것을 필요로 하는 중간 어드레스일 것이다. 그러므로, 변환 테이블 워크가 어떠한 블록 디스크립터도 조우하지 않지만, 그러나 페이지 디스크립터가 발견되는 L3까지 내내 진행되는 경우, 그러면, 전체 페이지 테이블 워크 프로세스는 다수의 레벨의 페이지 테이블에 다음의 순서로 액세스하는 것을 포함할 수도 있다:

[0105]  물리적 어드레스로의 스테이지 1 L0 페이지 테이블의 기본 어드레스의 스테이지 2 변환(스테이지 1 변환이 오퍼레이팅 시스템에 의해 구성되기 때문에, 스테이지 1 L0 기본 어드레스는 통상적으로 중간 물리적 어드레스이다). 스테이지 2 변환은 4 개의 룩업(스테이지 2 L0; 스테이지 2 L1; 스테이지 2 L2; 스테이지 2 L3)을 포함한다.

[0106]  타겟 가상 어드레스의 L0 인덱스 부분 "a" 및 변환된 스테이지 1 L0 기본 어드레스에 기초하여 획득되는 어드레스에서의 엔트리의 스테이지 1 L0 룩업, 스테이지 1 L1 기본 어드레스(중간의 물리적 어드레스)를 획득함

[0107]  물리적 어드레스로의 스테이지 1 L1 기본 어드레스의 스테이지 2 변환(역시, 4 개의 룩업을 포함함).

[0108]  타겟 가상 어드레스의 L1 인덱스 부분 "b" 및 변환된 스테이지 1 L1 기본 어드레스에 기초하여 획득되는 어드레스에서의 엔트리의 스테이지 1 L1 룩업, 스테이지 1 L2 기본 어드레스(중간의 물리적 어드레스)를 획득함

[0109]  물리적 어드레스로의 스테이지 1 L2 기본 어드레스의 스테이지 2 변환(역시, 4 개의 룩업을 포함함).

[0110]  타겟 가상 어드레스의 L2 인덱스 부분 "c" 및 변환된 스테이지 1 L2 기본 어드레스에 기초하여 획득되는 어드레스에서의 엔트리의 스테이지 1 L2 룩업, 스테이지 1 L3 기본 어드레스(중간의 물리적 어드레스)를 획득함

[0111]  물리적 어드레스로의 스테이지 1 L3 기본 어드레스의 스테이지 2 변환(역시, 4 개의 룩업을 포함함).

[0112]  타겟 가상 어드레스의 L3 인덱스 부분 "d" 및 변환된 스테이지 1 L3 기본 어드레스에 기초하여 획득되는 어드레스에서의 엔트리의 스테이지 1 L3 룩업, 타겟 가상 어드레스에 대응하는 타겟 중간 물리적 어드레스를 식별함.

- [0113]  원래의 타겟 가상 어드레스에 대응하는 액세스할 메모리 내 위치를 나타내는 타겟 물리적 어드레스로의 타겟 중간 물리적 어드레스의 스테이지 2 변환(역시, 4 개의 룩업을 포함함).
- [0114] 그러므로, 어떠한 캐싱도 없이, 그리고 스테이지 2에 대한 시작 레벨이 L0이라는 것을 가정하면, 변환은 총 24 개의 룩업을 포함할 것이다. 스테이지 2에 대한 시작 레벨이 L1인 경우, 이것은 룩업 수를 19 개로 감소시킬 수 있다(수행되는 5 개의 스테이지 2 변환 각각에 대한 하나 더 적은 룩업). 그럼에도 불구하고, 상기의 순서로부터 확인될 수 있는 바와 같이, 전체 페이지 테이블 워크 프로세스를 수행하는 것은, 어드레스 변환의 스테이지 각각에 대해 페이지 테이블의 레벨 각각을 차례로 옮겨 가기 위해 메모리에 대한 많은 수의 액세스를 필요로 할 수도 있기 때문에, 매우 느릴 수 있다. 이것은 MMU(28)의 TLB(29)에 있는 변환 테이블 워크로부터 유도되는 정보를 캐싱하는 것이 종종 바람직한 이유이다. 캐싱된 정보는 VA로부터 IPA로의 최종 스테이지 1 어드레스 매핑, IPA로부터 PA로의 최종 스테이지 2 매핑, 또는 VA로부터 바로 PA로의 조합된 스테이지 1 및 스테이지 2 매핑(스테이지 1 및 스테이지 2 구조의 이전 룩업으로부터 유도됨)을 포함할 수 있을 뿐만 아니라, 또한, 스테이지 1 및 스테이지 2 테이블의 더 높은 레벨 페이지 테이블로부터의 엔트리도 MMU(28)의 TLB(29) 내에 캐싱될 수 있다. 이것은, 심지어 주어진 타겟 어드레스에 대한 최종 레벨 어드레스 매핑이 현재 어드레스 변환 캐시에 없는 경우에도, 전체 페이지 테이블 워크의 적어도 일부 단계가 우회되는 것을 허용할 수 있다.
- [0115] 도 6은 2 스테이지 어드레스 변환에서 수반되는 변환 테이블 엔트리 중 일부에 대해 정의될 수 있는 특정한 퍼미션 또는 특성을 명시하는 정보를 사용하여 주석 처리된다. 이것은, 일단 이들 퍼미션 및 특성이 하기에서 설명되면, 나중에 더욱 상세하게 설명될 것이다.
- [0116] 메모리 관리 유닛 예
- [0117] 도 7은 메모리 관리 유닛(28)을 더욱 상세하게 예시한다. 도 1 및 도 7이 간결성을 위해 단일의 MMU(28)을 도시하지만, 페치 스테이지(6)에 의해 개시되는 명령어 페치 액세스 및 로드/저장 유닛(26)에 의해 개시되는 데이터 액세스에 대한 메모리 관리 동작을 각각 핸들링하기 위해 별개의 명령어측 및 데이터측 MMU가 제공되는 것도 또한 가능하다 - 이 경우 명령어측 및 데이터측 MMU 둘 모두는 도 7에서 도시되는 컴포넌트를 가질 수도 있다. TLB(29)뿐만 아니라, MMU(28)는 룩업된 변환 테이블 엔트리에서 명시되는 액세스 퍼미션을 체크하고 그들 액세스 퍼미션에 따라 메모리 액세스 요청의 핸들링을 제어하기 위한 액세스 제어 회로부(50)를 포함한다. 예를 들면, 액세스 제어 회로부(50)는, 특정한 타입의 메모리 액세스 요청이 (예를 들면, 요청이 판독, 기록 또는 명령어 페치 요청인지의 여부에 기초하여, 또는 요청으로 하여금 발행되게 한 실행된 명령어의 타입에 기초하여) 액세스되고 있는 메모리의 주어진 영역에 대한 대응하는 변환 테이블 엔트리에서 명시되는 퍼미션을 충족하는지의 여부를 결정하기 위한 체크를 구현할 수도 있다. MMU는, 변환 테이블로부터의 필요한 정보가 TLB(29)에서 이미 이용 가능하지 않은 경우, 도 4 내지 도 6에서 상기에서 도시되는 바와 같이, 변환 테이블 워크 프로세스 동안 변환 테이블 엔트리의 판독을 요청하기 위해 메모리 액세스 요청을 메모리에 발행하는 변환 테이블 워크 제어 회로부(52)를 또한 포함한다. 변환 테이블 워크 제어 회로부(52)는 스테이지 1 또는 스테이지 2 변환 테이블 구조에서 관련 변환 테이블 엔트리를 획득하기 위해 판독될 메모리 시스템 위치의 어드레스를 생성하는 것을 담당할 수 있다. 또한, 변환 테이블 워크를 수행하는 동안, 변환 테이블 워크 제어 회로부(52)는, 변환 테이블 워크에서 순회되고 있는 소정의 변환 테이블 엔트리 내의 액세스 추적 메타데이터에 대한 업데이트를 요청하는 메타데이터 업데이트 기록 요청을 생성할 수도 있다. 이들 메타데이터 업데이트 기록 요청은 액세스 추적 메타데이터를 업데이트하여 특정한 메모리 영역이 액세스되었다는 것을 반영할 수도 있는데, 이것은 페이징과 같은 동작을 관리하는 데 유용할 수 있다. 메타데이터 업데이트 기록 요청의 어드레스가, 액세스되고 있는 데이터 또는 명령어의 어드레스가 아닌, 변환 테이블 워크에서 액세스되는 변환 테이블 엔트리의 어드레스이기 때문에, 메타데이터 업데이트 기록 요청은, 명령어 페치 어드레스로서 직접적으로 명시되지 않은 어드레스, 또는 프로세싱 회로부(4)에 의해 실행되는 로드/저장 명령어의 피연산자로부터 유도되는 로드/저장 타겟 어드레스를 명시하기 위해, 하드웨어에서 생성된다.
- [0118] MMU(28)는 변환 테이블 워크 및 메모리 관리 동작의 다른 양태를 제어하기 위한 다양한 제어 레지스터에 액세스할 수 있다. 예를 들면, 제어 레지스터는 도 4 및 도 5와 관련하여 앞서 언급되는 바와 같이 스테이지 1 및 스테이지 2 기본 어드레스 레지스터(TTBR_EL1, VTTBR_EL2)를 포함할 수도 있다. 레지스터(TTBR_EL1)의 스테이지 1 기본 어드레스는 예외 레벨(EL1, EL2 또는 EL3)에서 실행되는 명령어에 응답하여 기록 가능하다. 레지스터(TTBR_EL2)의 스테이지 2 기본 어드레스는 예외 레벨(EL2 또는 EL3)에서 실행되는 명령어에 응답하여 기록 가능하다. 몇몇 경우에, EL2에서 동작되는 하이퍼바이저가 오퍼레이팅 시스템에 의해 요청되고 있는 업데이트를 체크할 수 있고 필요하다면 업데이트를 방지할 수 있도록, EL1에서 실행되는 명령어에 의해 트리거되는 TTBR_EL1

에 대한 기록이 EL2에 트랩되어야 한다는 것을 나타내도록 제어 값을 설정하는 것이 가능하다. 제어 레지스터는, 스테이지 1 및 스테이지 2 변환 및 액세스 퍼미션 체크가 수행되는 방식에 영향을 끼치는 다양한 제어 상태 값을 명시하기 위해 사용되는 레지스터(TCR_EL1, VTCR_EL2)를 또한 포함된다. 예를 들면, 스테이지 2 워크의 시작 레벨을 명시하는 제어 값은 VTCR_EL2에서 명시될 수도 있다. 또한, TCR_EL1 또는 VTCR_EL2 제어 레지스터는 어떤 타입의 보안 체크가 인에이블/디스에이블되는지 명시하는 제어 값을 명시할 수도 있다. 하기에서 더욱 상세하게 논의되는 바와 같이, 제어 레지스터는 유효한 및 무효한 엔트리 각각에 대한 보호된 스테이지 1 변환 테이블 엔트리에서 업데이트가 제한되는 비트의 업데이트 제한 서브세트를 명시하기 위한 마스크 레지스터(RCWMask_Valid_EL2 및 RCWMask_Invalid_EL2로서 라벨링됨)를 또한 포함할 수 있다. 추가적인 제어 레지스터가 또한 제공될 수 있다는 것, 및 제어 레지스터를 위해 사용되는 라벨은 변환될 수 있다는 것이 인식될 것이다. 예를 들면, 주어진 변환 테이블 엔트리에 대한 액세스 퍼미션 정보를 명시하기 위해 주어진 변환 테이블 엔트리에 의해 참조될 수 있는 간접 액세스 퍼미션 정보를 명시하기 위해, 하나 이상의 퍼미션 간접 표현 레지스터가 제공될 수 있다. 또한, 동일한 정보가 제어 레지스터 내에서 상이한 포맷으로 배열되는 것이 가능하며, 따라서 하나의 제어 레지스터 또는 다른 제어 레지스터에 대한 특정한 정보 할당은 필수적인 피처는 아니다.

[0119] 도 7에서 도시되는 바와 같이, 장치는 보호된 엔트리로서 인코딩되는 스테이지 1 변환 테이블 엔트리에 대한 업데이트가 요청될 때 보호 엔트리 체크를 수행하기 위한 보호 엔트리 체크 회로부(54)를 구비할 수도 있다(이것은 하기에서 더욱 상세하게 설명될 것임). 도 7에서 점선으로 도시되는 바와 같이, 프로세싱 시스템 내의 상이한 위치에서 보호 엔트리 체크 회로부(54)를 구현하는 것이 가능하다. 예를 들면, 보호 엔트리 체크 회로부(54)는 MMU(28)의 액세스 제어 회로부(50)의 일부일 수 있다. 대안적으로, 보호 엔트리 체크 회로부는 보호 엔트리 체크 회로부(54)를 사용하여 체크될 주어진 메모리 액세스 요청에 의해 액세스되고 있는 물리적 메모리 시스템 위치에 더 가까운 메모리 시스템 컴포넌트(56) 내에서 구현될 수 있다. 예를 들면, 메모리 시스템 컴포넌트(56)는 캐시(30, 32), 다중 프로세서 코어에 의한 공유 메모리에 대한 액세스를 관리하기 위해 사용되는 인터커넥트, 또는 특정한 메모리 스토리지 유닛과 관련되는 메모리 컨트롤러일 수 있다. 또한, 보호 엔트리 체크 회로부(54)는 시스템의 다수의 위치(예를 들면, MMU(28), 로드/저장 유닛(26) 및/또는 메모리 시스템 컴포넌트(56))에서 분산 로직을 포함할 수 있다.

[0120] 변환 테이블 액세스 퍼미션 및 속성

[0121] 도 8은 스테이지 1 및 스테이지 2 변환 테이블 구조에서의 변환 테이블 엔트리에 대한 예시적인 포맷을 예시한다. 동일한 정보가 상이한 배열로 표현될 수 있고 따라서 디스크립터 내에서의 필드의 특정한 순서 및 레이아웃이 도 8에서 도시되는 것과는 변환될 수 있다는 것이 인식될 것이다.

[0122] 이 예에서, 유효한 변환 테이블 디스크립터는 최하위 비트가 1로 설정되게 하고 무효한 변환 테이블 디스크립터는 최하위 비트가 0으로 설정되게 한다. 변환 테이블 구조의 레벨 0, 1, 2 중 하나에서 판독되는 유효한 디스크립터의 경우, 제2 최하위 비트는 디스크립터가 테이블 디스크립터(제2 최하위 비트가 1로 설정됨)인지 또는 블록 디스크립터(제2 최하위 비트가 0으로 설정됨)인지를 여부를 구별한다. 스테이지 1 또는 스테이지 2 구조의 L3에 있는 페이지 디스크립터는, 페이지 디스크립터가 블록 디스크립터와 구별되는 것을 허용하기 위해, 제2 최하위 비트가 1로 설정되게 할 수도 있다. 물론, 유효한/무효한 엔트리, 및 테이블/블록/페이지 디스크립터는 다른 인코딩 방법에 의해 또한 구별될 수 있다.

[0123] 스테이지 1 및 스테이지 2 엔트리 둘 모두에 대해, 유효한 테이블 디스크립터는 스테이지 1 또는 스테이지 2 변환 테이블 구조의 다음 레벨에 있는 변환 테이블의 기본 어드레스를 나타내는 다음 레벨 테이블 어드레스(60)를 제공한다. 유효한 스테이지 1 블록 또는 페이지 디스크립터는 스테이지 1 변환 테이블 구조를 인덱싱하기 위해 사용되는 가상 어드레스에 대응하는 중간 어드레스 매핑(62)을 제공한다. 유효한 스테이지 2 블록 또는 페이지 디스크립터는 스테이지 2 변환 테이블 구조를 인덱싱하기 위해 사용되는 중간 어드레스에 대응하는 물리적 어드레스 매핑(64)을 제공한다.

[0124] 스테이지 1 블록 및 페이지 디스크립터는 대응하는 메모리 영역에 대한 액세스를 제어하기 위해 사용되는 스테이지 1 액세스 퍼미션(66)을 또한 제공한다. 예를 들면, 스테이지 1 액세스 퍼미션(66)(통상적으로 EL1에서 OS에 의해 설정됨)은, 영역이 실행 가능 명령어의 명령어 페치를 위해 판독되도록, 기록되도록 및/또는 사용되는 것이 허용되는지의 여부를 명시할 수도 있다.

[0125] 마찬가지로, 스테이지 2 블록 및 페이지 디스크립터는 대응하는 메모리 영역에 대한 액세스를 제어하기 위해 사용되는 스테이지 2 액세스 퍼미션(68)을 제공한다. 다시 말하지만, 스테이지 2 액세스 퍼미션(68)(통상적으로 EL2에서 하이퍼바이저에 의해 설정됨)은, 영역이 실행 가능 명령어의 명령어 페치를 위해 판독, 기록 및/또는

사용되는 것이 허용되는지의 여부를 명시할 수도 있다. 스테이지 1 액세스 퍼미션(66)과 스테이지 2 액세스 퍼미션(68) 사이에서 충돌이 있는 경우, 그러면, 속성의 더욱 제한적인 세트가 우선될 수도 있다.

[0126] 스테이지 1 또는 스테이지 2 블록/페이지 디스크립터는, 도 8에서 도시되지 않는, 메모리 영역과 관련되는 다른 속성을 또한 명시할 수 있다. 예를 들면, 이들 속성은, 대응하는 메모리 영역으로부터 데이터를 캐싱하는 것이 허용되는지의 여부, 디바이스 메모리에 대한 상이한 메모리 액세스의 재정렬 또는 병합이 허용되지 않도록 영역이 디바이스 메모리로서 정의되는지의 여부, 등등과 같은 특성을 명시할 수 있다.

[0127] 스테이지 1 블록/페이지 디스크립터는 주어진 메모리 영역에 대한 액세스의 빈도를 추적하기 위해 오퍼레이팅 시스템에 의해 사용될 수 있는 액세스 추적 메타데이터를 또한 명시할 수 있다. 예를 들면, 메타데이터는 액세스 플래그(access flag; AF)(70) 및 더티 비트 수정자(dirty bit modifier; DBM)(72)를 포함할 수도 있다.

[0128] 주기적으로, 오퍼레이팅 시스템은 모니터링될 메모리 영역의 세트에 대응하는 엔트리의 액세스 플래그를 클리어할 수 있다. 이들 메모리 영역 중 하나에 대해 관독 액세스가 이루어지는 경우, 액세스 플래그(70)는 (이전 액세스에 후속하여 아직 설정되지 않은 경우) 대응하는 스테이지 1 블록 또는 페이지 디스크립터에서 설정될 수도 있다(액세스 플래그(70)로 하여금 설정되게 하는 메모리 액세스 요청은 앞서 언급되는 변환 테이블 워크 제어 회로부(52)에 의해 생성되는 메타데이터 업데이트 기록 요청 중 하나일 수도 있다). 일정 기간의 모니터링 이후, 그 다음, 오퍼레이팅 시스템은 소정의 페이지가 얼마나 자주 액세스되는지에 관한 정보로부터 이익을 얻을 수도 있는 동작을 지원하기 위해 액세스 플래그(70)를 체크할 수 있다. 예를 들면, 오퍼레이팅 시스템은 메모리 영역이 얼마나 많은 횟수 액세스되었는지를 추적하는 메모리 영역당 엔트리를 사용하여 추가적인 추적 데이터 구조를 메모리에서 유지할 수 있고, 따라서 모니터링의 각각의 기간의 끝에서 액세스 플래그(70)가 설정된 메모리 영역에 대응하는 그 추가적인 추적 구조의 엔트리는 충분될 수 있다. 따라서, 다수의 모니터링의 기간 이후, 그 추가적인 추적 구조는 대응하는 메모리 영역에 대한 액세스의 상대적 빈도의 표시를 제공할 것이다. 이것은 페이지징과 같은 동작을 제어하기 위한 유용한 정보를 제공할 수 있는데, 여기서는, 다른 더욱 빈번하게 액세스된 페이지와 비교하여, 대응하는 데이터가 외부 스토리지로 페이지징 아웃하기 위해 우선될 수도 있는 메모리의 가장 덜 빈번하게 액세스된 페이지를 아는 것이 유용할 수 있다.

[0129] 마찬가지로, DBM(72)은 어떤 페이지가 기록 대상이 되었는지를 추적하는 것을 돕는다. 주어진 페이지가 기록되었는지의 여부를 추적하기를 오퍼레이팅 시스템이 소망하는 경우, 페이지가 매핑될 때 또는 모니터링의 기간의 시작에서, 오퍼레이팅 시스템은 그 페이지에 대한 액세스 퍼미션을 (심지어 페이지가 기록되는 것이 허용되도록 의도되는 경우에도) "관독 전용"으로 설정할 수 있고 DBM 비트(72)를 설정할 수 있다. DBM 비트(72)가 설정될 때 관독 전용 페이지에 대한 기록에 의해 야기되는 액세스 퍼미션 장애의 경우, 오퍼레이팅 시스템은, DBM 비트(72)가 설정되는 것으로부터, 이것이 관독 전용 퍼미션의 "실제" 위반이 아니라는 것을 결정할 수도 있고, 대신, 오퍼레이팅 시스템으로 하여금, 기록 요청을 받는 페이지를 추적하는 메모리에 저장되는 데이터 구조를 업데이트하게 할 수도 있고, 또한, 이제 장애를 트리거하지 않고도 페이지가 기록될 수 있다는 것을 나타내도록 페이지에 대한 기록 액세스 퍼미션을 업데이트하게 할 수도 있다. 일정 기간의 모니터링 이후, 메모리의 추적 데이터 구조는, 특정한 영역의 페이지징 아웃시, 그 영역으로부터의 수정된 데이터를 다시 외부 스토리에 기록하는 것이 필요한지의 여부, 또는 (어떠한 기록도 발생하지 않은 경우) 온 칩 메모리에 저장되는 데이터가 영역의 페이지징 아웃시 간단히 폐기될 수 있는지의 여부를 결정하기 위해 소프트웨어에 의해 사용될 수 있는데, 데이터가 깨끗하면 외부 메모리의 대응하는 데이터가 여전히 동일한 것으로 가정될 수 있기 때문이다.

[0130] 액세스 플래그(70) 및 더티 비트 수정자(72)는 변환 테이블 엔트리 내에 저장될 수 있는 가능한 액세스 추적 데이터의 몇몇 예에 불과하며 다른 예는 다른 타입의 액세스 추적 메타데이터를 제공할 수 있다. 예를 들면, 다른 예에서, 다중 비트 액세스 카운터가 액세스 추적 메타데이터로서 제공될 수 있다. 또한, 다른 예에서는, 기록 요청시 관독 전용 액세스 퍼미션 위반의 핸들링을 수정하기 위해 그리고 기록 퍼미션 업데이트를 트리거하기 위해 DBM 비트(72)가 사용되는 대신, 앞서 설명되는 바와 같이 페이지에 대한 제1 액세스시 액세스 플래그(70)가 업데이트되는 방식과 유사한 방식으로, DBM 플래그(72)는, DBM 플래그(72)를 클리어한 이후 페이지에 대한 제1 기록에 응답하여, 대신 직접적으로 업데이트될 수 있다.

[0131] 도 8에서 도시되는 바와 같이, 다양한 변환 테이블 디스크립터는 공격자에 의한 공격 하에 있는 오퍼레이팅 시스템 코드에 의한 손상에 대비하여 스테이지 1 변환 테이블 엔트리를 보호하는 데 유용할 수 있는 다수의 다른 조각의 정보를 또한 포함할 수 있다. 통상적인 오퍼레이팅 시스템 코드는 공격에 대비하여 안전한 것으로 완전히 검증하기 어려울 수도 있는 많은 라인의 코드를 포함할 수도 있고, 따라서, 메모리의 영역에 액세스할 권한을 승인되지 않은 방식으로 공격자에게 제공하기 위해 그들 엔트리에서의 퍼미션 또는 어드레스 매핑을 수정하

려는 시도에서, 변환 테이블 엔트리를 저장하기 위해 사용되는 메모리의 영역에 대한 기록을 야기하도록 오퍼레이팅 시스템 코드의 거동을 수정할 수 있을 수도 있는 가능성에 대비하여 보호하기 위한 아키텍처 메커니즘을 제공하는 것이 바람직하다.

[0132] 도 8에서 도시되는 바와 같이, 공격에 대비하여 스테이지 1 변환 테이블을 강화하기 위한 하기에서 설명되는 추가적인 속성 및 퍼미션은 다음을 포함한다:

[0133]  스테이지 2 블록/페이지 디스크립터는 대응하는 메모리 영역이 부분 판독 전용(partially-read-only; PRO) 퍼미션을 갖는지의 여부를 나타내는 인코딩을 갖는 스테이지 2 액세스 퍼미션(68)을 명시한다(부분 판독 전용 퍼미션은 "대부분 판독 전용" 퍼미션으로서 또한 지칭될 수 있음);

[0134]  (테이블 디스크립터 및 블록/페이지 디스크립터를 포함하는) 스테이지 1 변환 테이블 구조의 모든 레벨에서의 유효한 스테이지 1 디스크립터는, 그 변환 테이블 엔트리가, 보호되지 않은 엔트리와 비교하여, 승인되지 않은 업데이트에 대비한 추가적인 보호가 제공되는 "보호된" 엔트리인지의 여부를 명시하는 인코딩을 갖는다. 예를 들면, 유효한 스테이지 1 디스크립터 내의 비트필드(76)는 엔트리가 보호된 엔트리인지의 여부를 나타낼 수도 있다.

[0135]  스테이지 1 변환 테이블 구조의 모든 레벨에 있는 무효한 스테이지 1 디스크립터는 그 변환 테이블 엔트리가 "보호된" 엔트리인지의 여부를 명시하는 인코딩을 또한 가질 수도 있다. 이것은, 유효한 스테이지 1 변환 테이블 엔트리에 대한 보호된 엔트리를 나타내기 위해 사용되는 비트필드(76)와 비교하여, 변환 테이블 엔트리 인코딩 내에서 상이한 위치에 있을 수도 있는 비트필드(78)를 사용하여 나타내어질 수도 있다.

[0136]  스테이지 2 블록/페이지 디스크립터는 대응하는 메모리 영역이 안티 앨리어싱 특성을 갖는지의 여부를 명시하는 지시자(79)를 가질 수도 있다. 몇몇 예에서, 안티 앨리어싱 특성의 존재 또는 부재는 다른 타입의 특성을 또한 시그널링할 수 있는 결합된 비트필드에서 인코딩될 수 있거나, 또는 대안적으로, 안티 앨리어싱 비트필드는 어떠한 다른 정보도 시그널링하지 않는 독립형 비트필드일 수 있다. 명시되는 경우, 안티 앨리어싱 특성은, 대응하는 물리적 어드레스에 대한 액세스가 허용되기 위해서는, 스테이지 2 디스크립터에 액세스하기 위해 사용되는 중간 어드레스를 획득하기 위해 변환 테이블 워크에서 액세스되는 모든 스테이지 1 변환 테이블 엔트리가 비트필드(76)를 사용하여 보호된 엔트리로서 명시될 필요가 있을 것이고 그리고 대응하는 스테이지 2 블록/페이지 디스크립터가 부분 판독 전용 퍼미션(74)을 나타내는 메모리의 영역에 저장될 필요가 있을 것이라는 것을 나타낸다.

[0137]  스테이지 2 블록/페이지 디스크립터는 또한, 스테이지 2 액세스 퍼미션(68)에서, 대응하는 메모리 영역이 스테이지 1 변환을 위한 최상위 레벨 변환 테이블(즉, 도 4 및 도 6의 예에서 스테이지 1 L0 테이블)을 유지하는 것이 허용되는지의 여부를 나타내는 스테이지 1 최상위 레벨 엔트리 존재 퍼미션(80)을 명시할 수도 있다. 하나의 예에서, 스테이지 2 액세스 퍼미션(68)의 인코딩은 스테이지 1 최상위 레벨 엔트리 존재 퍼미션이 PRO 퍼미션의 추가적인 속성으로서 인코딩되도록 하는 그러한 것일 수도 있고, 그 결과, 스테이지 2 액세스 퍼미션(68)을 위해 인코딩되는 것이 허용되는 값은 다음을 나타내는 값을 포함할 수도 있다:

[0138] o 스테이지 1 최상위 레벨 엔트리 존재 퍼미션이 없는 PRO 퍼미션;

[0139] o 스테이지 1 최상위 레벨 엔트리 존재 퍼미션이 있는 PRO 퍼미션; 또는

[0140] o PRO 퍼미션 또는 스테이지 1 최상위 레벨 엔트리 존재 퍼미션(예를 들면, 판독 전용 퍼미션, 판독/기록 퍼미션, 및/또는 실행 퍼미션) 중 어느 하나를 갖지 않는 하나 이상의 다른 타입의 퍼미션.

[0141] 이 경우, PRO 퍼미션과는 관계없이 스테이지 1 최상위 레벨 엔트리 존재 퍼미션을 명시하는 것이 가능하지 않을 수도 있다. 그러나, 다른 구현에는, 심지어 PRO 퍼미션이 제공되지 않는 경우에도 스테이지 1 최상위 레벨 엔트리 존재 퍼미션이 명시될 수 있도록, PRO 퍼미션과는 독립적으로 스테이지 1 최상위 레벨 엔트리 존재 퍼미션을 인코딩할 수 있다.

[0142] 이들 속성 및 퍼미션의 사용은 하기에서 더욱 상세하게 논의될 것이다. 모든 가능한 구현예에서 이들 속성 및 퍼미션 모두가 지원될 필요는 없다는 것이 인식될 것이다. S1 및 S2 액세스 퍼미션이 변환 테이블 엔트리의 인코딩에서 직접적으로 명시되는 예를 도 8이 예시하지만, 퍼미션 레지스터에서 명시되는 퍼미션을 사용하여 동일한 정보를 간접적으로 명시하는 것도 또한 가능하다. 예를 들면, S1 또는 S2 퍼미션 필드(66, 68)는, 퍼미션

레지스터의 다수의 퍼미션 필드 중 어떤 것이 대응하는 변환 테이블 엔트리에 대한 액세스 퍼미션을 명시하는지를 선택하는 인덱스 값을 명시할 수 있다. 제어 레지스터에 저장되는 제어 상태에 따라 변환 테이블 엔트리의 일부 필드가 상이한 방식으로 해석되는 것도 또한 가능하다. 예를 들면, 스테이지 1 변환 테이블 엔트리의 보호된 엔트리 필드(76)는, 제어 레지스터에 저장되는 제어 값이 제1 값을 갖는 경우 스테이지 1 변환 테이블 엔트리가 보호되는지의 여부를 나타내는 것으로 해석될 수도 있고, 제어 값에 제2 값을 갖는 경우 상이한 방식으로 해석될 수도 있다. 마찬가지로, 스테이지 2 변환 테이블 엔트리의 안티 엘리머싱 특성 필드(79)는, 제어 레지스터에 저장되는 제어 값이 제1 값을 갖는 경우 스테이지 2 변환 테이블 엔트리가 안티 엘리머싱 특성을 갖는지의 여부를 나타내는 것으로 해석될 수도 있고, 제어 값이 제2 값을 갖는 경우 상이한 방식으로 해석될 수도 있다.

[0143] 부분 판독 전용 퍼미션

[0144] 도 9는 부분 판독 전용(PRO) 퍼미션을 갖는 것으로 스테이지 2 액세스 퍼미션(68)에 의해 마킹되는 메모리 영역에 대한 기록 액세스 제어 거동을 요약한다. PRO 퍼미션을 갖는 메모리 영역은, 적어도 현재의 실행 상태가 EL1(오퍼레이팅 시스템 코드가 실행될 것으로 예상되는 상태)일 때 발행되는 기록 요청에 대해, 기록 요청 타입의 제한된 서브세트에 대해 기록 가능하지만, 그러나 기록 요청 타입에 대해서는 판독 전용으로서 처리된다. 단지 일부 기록 요청 타입만이 메모리 영역을 판독 전용으로서 간주하기 때문에, 따라서, 이 퍼미션은 부분 판독 전용 퍼미션으로 설명된다.

[0145] 도 9는 대응하는 스테이지 2 변환 테이블 엔트리가 영역을 PRO 퍼미션을 갖는 것으로 정의하는 메모리 영역에 기록하도록 허락되고 금지되는 기록 요청의 타입을 요약한다. EL1에서 기록이 발행될 때 PRO 영역에 기록하도록 허락되는 기록 요청 타입의 제한된 서브세트는, 상기에서 논의되는 바와 같이, 더티 비트 수정자(72)가 설정되는 경우 기록 퍼미션 또는 액세스 플래그(70)와 같은, 변환 테이블 엔트리의 액세스 추적 메타데이터를 업데이트하기 위해 MMU(28)에 의해 생성되는 메타데이터 업데이트 기록 요청을 포함한다. 또한, 제한된 서브세트는, 변환 테이블 엔트리를 업데이트하기 위한 그러한 기록을 변환 테이블 엔트리 이외의 데이터를 업데이트하기 위한 기록과 구별하기 위한 프로그램 코드의 의도를 표현하기 위해, 변환 테이블 엔트리를 업데이트함에 있어서의 사용을 위해 의도되는 전용 타입의 명령어일 수 있는, 예외 상태(EL1)에서 실행되는 적어도 하나의 사전 결정된 타입의 변환 테이블 엔트리 업데이트 명령어를 포함할 수 있다. 예를 들면, 사전 결정된 타입의 변환 테이블 엔트리 업데이트 명령어는 하기에서 도 12 내지 도 14와 관련하여 논의되는 보호 엔트리 체크 판독 체크 기록(read-check-write; RCW) 명령어의 변형을 포함할 수 있다. 도 9에서 도시되는 바와 같이, 예외 상태(EL0 및 EL1) 중 하나에서 실행되는 적어도 하나의 다른 타입의 저장 명령어에 응답하여 발행되는 기록 요청은 메모리의 PRO 영역에 대한 기록이 금지된다. 예를 들면, 금지된 기록 요청 타입은 프로세싱 회로부(4)에 의해 지원되는 명령어 세트 아키텍처의 저장 명령어의 대부분을 포함할 수도 있다. 영역이 PRO 퍼미션을 사용하여 마킹되는 경우 변환 테이블 엔트리를 명령어 타입의 제한된 세트로 업데이트하는 능력을 제한하는 것에 의해, 이것은 스테이지 1 변환 테이블 엔트리를 손상시키려고 시도할 때 공격자가 악용하는 데 이용 가능한 공격 표면을 감소시키는데, 예를 들면, 기록 요청의 제한된 서브세트 중 하나가 아닌 범용 저장을 위한 어드레스 피연산자를 무단 변경하는 것은 메모리의 PRO 영역에 대한 액세스를 허용하지 않을 것이다.

[0146] 옵션 사항으로, 실행 상태(EL2 또는 EL3) 중 하나에서 실행되는 저장 명령어에 응답하여 발행되는 기록 요청은, 심지어 실행 상태(EL0 또는 EL1)에서 실행되는 경우 그 영역에 기록하는 것이 금지되었을 저장 명령어의 타입을 그들이 갖는 경우에도, 부분 판독 전용 영역에 기록하도록 허락될 수도 있다. 그러나, 이것은 앞서 언급된 바와 같이 필수적인 것은 아니다.

[0147] 도 10은 명령어를 프로세싱하고 메모리 관리 동작을 제어하는 방법을 예시하는 흐름도이다. 단계(500)에서, 프로세싱 회로부(4)는 상이한 레벨의 특권과 관련되는 다수의 실행 상태(EL0-EL3) 중 하나에서 명령어를 프로세싱한다. 단계(502)에서, 메모리 액세스 요청에 응답하여, MMU(28)는 메모리 액세스 요청에 의해 명시되는 타겟 가상 어드레스(VA)의 타겟 물리적 어드레스(PA)로의 2 스테이지 어드레스 변환을 수행하는데, 타겟 VA 대 타겟 PA의 매핑은 타겟 VA로부터 타겟 중간 어드레스(IPA)로의 스테이지 1 어드레스 매핑 및 타겟 IPA로부터 타겟 PA로의 스테이지 2 어드레스 매핑에 의존한다. 단계(504)에서, MMU(28)의 액세스 제어 회로부(50)는, 대응하는 스테이지 2 변환 테이블 엔트리에서 명시되는 스테이지 2 액세스 퍼미션 정보(68)에 적어도 기초하여, 메모리 액세스 요청에 대한 퍼미션 체크를 수행한다. 퍼미션 체크는 스테이지 1 액세스 퍼미션 정보(66)에 또한 기초할 수 있다. 스테이지 2 액세스 퍼미션 정보(66)는 타겟 IPA에 대응하는 메모리 영역이 PRO 퍼미션을 갖는지의 여부를 명시하는 인코딩을 갖는다. 단계(506)에서, MMU(28)는, 메모리 액세스 요청이 사전 결정된 실행 상태(예를 들면, EL1)에서 발행되는 기록 요청이고, 스테이지 2 액세스 퍼미션 정보가 PRO 퍼미션을 명시하고, 기록

요청이 메모리의 PRO 영역에 기록하는 것이 허용되는 기록 요청 타입의 제한된 서브세트 이외의 기록 요청 타입을 갖는 경우, 메모리 액세스 요청을 거부한다. 옵션 사항으로, MMU(28)는, 메모리 액세스 요청이 사전 결정된 실행 상태가 아닌 실행 상태(예를 들면, EL0)에서 발행되는 기록 요청인 경우, PRO 퍼미션이 스테이지 2 액세스 퍼미션 정보에 의해 명시되고 기록 요청 타입이 기록 요청 타입의 제한된 서브세트 중 하나가 아닌 경우, 메모리 액세스 요청을 또한 거부할 수 있다.

[0148] 도 11은 MMU(28)의 액세스 제어 회로부(50)에 의해 수행되는 부분 판독 전용 체크를 더욱 상세하게 예시한다. 단계(600)에서, 프로세싱 회로부(4)가 사전 결정된 실행 상태(EL1)에 있을 때 메모리 기록 액세스 요청(write memory access request)이 발행된다. 기록 요청은 타겟 VA를 명시한다. 기록 요청은, 프로세싱 회로부에 의해 실행되는 로드/저장 명령어에 응답하여 프로세싱 회로부(4)의 로드/저장 유닛(26)에 의해, 또는 명령어 페치를 요청할 때 페치 스테이지(6)에 의해 발행되었을 수 있다. 기록 요청은 또한, 변환 테이블 워크 프로세스 동안 변환 테이블 엔트리의 액세스 추적 메타데이터를 업데이트하기 위해 MMU(28)에 의해 발행되는 메타데이터 업데이트 기록 요청일 수 있다.

[0149] 단계(602)에서, 타겟 VA는 액세스될 메모리 시스템 위치를 식별하는 타겟 PA로 변환된다. 변환을 수행하기 위해, MMU(28)는 TLB(29) 내에서 타겟 VA를 찾을 수 있고 타겟 VA로부터 타겟 PA로의 매핑이 TLB(29)에서 이미 이용 가능한 경우, 그러면, 어떠한 변환 테이블 워크도 필요하지 않다. 그렇지 않으면, 도 6에서 도시되는 변환 테이블 워크 프로세스의 적어도 일부가 필요로 될 수도 있지만, 일부 부분은, 변환 테이블 워크의 그 부분에 대한 대응하는 변환 테이블 엔트리로부터 유도되는 정보가 TLB(29)에 이미 캐싱되어 있는 경우, 스킵될 수 있을 수도 있다. 비보안 EL1로부터 발행되는 메모리 기록 액세스 요청의 경우, 변환은 스테이지 1 및 스테이지 2 변환 테이블 둘 모두에서 정의되는 매핑에 기초하여 2 스테이지 어드레스 변환으로서 수행되지만, 타겟 IPA가 명시적으로 결정될 필요가 있는 것을 방지하기 위해, 대응하는 스테이지 1 및 스테이지 2 퍼미션 정보와 함께, (스테이지 1 및 스테이지 2 매핑 둘 모두에 따라) 타겟 VA로부터 타겟 PA로의 직접적인 매핑을 캐싱하는 결합된 스테이지 1/스테이지 2 TLB가 제공되는 것이 가능하기 때문에, 이것은 두 개의 별개의 변환이 수행될 필요가 있다는 것을 반드시 의미하는 것은 아니다.

[0150] 단계(604)에서, 액세스 제어 회로부는 대응하는 스테이지 2(S2) 변환 테이블 엔트리(타겟 IPA로부터 타겟 PA로의 매핑을 명시하는 스테이지 2 엔트리)가 PRO 퍼미션을 명시하는지의 여부를 체크한다. 만약 그렇다면, 그러면 단계(606)에서, 액세스 제어 회로부는 프로세싱되고 있는 기록 요청이 기록 요청 타입의 제한된 서브세트 중 하나인지의 여부를 체크한다. 현재 기록 요청이 기록 요청 타입의 제한된 서브세트(적어도 메타데이터 업데이트 기록 요청을 포함함) 중 하나가 아닌 경우, 그러면 단계(608)에서, 메모리 액세스 요청은 거부되고 장애가 시그널링된다. 기록 요청이 기록 요청 타입의 제한된 서브세트 중 하나인 경우, 그러면, 부분 판독 전용 체크가 통과되고 따라서 방법은 단계(610)로 진행된다. 또한, 단계(604)에서, 대응하는 스테이지 2 변환 테이블 엔트리가 PRO 퍼미션을 명시하지 않는 경우, 그러면 단계(606)에서의 부분 판독 전용 체크는 생략될 수 있고 따라서 방법은 단계(604)로부터 단계(610)로 바로 진행된다.

[0151] 단계(610)에서, 액세스 제어 회로부는 메모리 기록 액세스 요청이 진행되는 것을 허용하는 데 필요한 임의의 다른 체크가 충족되는지의 여부를 결정한다. 예를 들면, 이들 체크는 스테이지 1 액세스 퍼미션(66)을 사용하는 체크, PRO 퍼미션이 명시되지 않는 경우 다른 스테이지 2 액세스 퍼미션(68)을 사용하는 체크, (보안 상태를 지원하는 구현예에서) 요청이 보안 상태에서부터 발행되는지 또는 비보안 상태에서부터 발행되는지의 여부에 기초하는 보안 체크뿐만 아니라 특정한 명령어 타입에 고유한 다른 체크, 예컨대 타겟 기억 위치의 데이터가 보호된 변환 테이블 엔트리인지의 여부에 관해 도 12 내지 도 14와 관련하여 하기에서 논의되는 체크를 포함할 수 있다. 이들 다른 체크 중 임의의 것이 충족되지 않는 경우, 그러면, 다시 단계(608)에서, 메모리 액세스 요청은 거부되고 장애가 시그널링된다. 장애의 원인을 구별하기 위해, 장애 신드롬 정보(fault syndrome information)가 설정될 수도 있다. 단계(610)에서 수행되는 체크가 충족되는 경우, 그러면 단계(612)에서, 메모리 액세스 요청은 진행되도록 허락될 수 있고, 따라서, 타겟 PA와 관련되는 메모리 시스템 위치는 기록 요청에 의해 명시되는 기록 데이터를 사용하여 업데이트될 수 있다.

[0152] 그러므로, 앞서 논의되는 도 6에서 도시되는 바와 같이, PRO 퍼미션은 민감한 정보를 포함하는 메모리 영역의 중간 어드레스를 획득하기 위해 스테이지 1 테이블 구조의 순회에서 사용되는 다양한 레벨의 스테이지 1 변환 테이블을 유지하는 메모리 시스템 위치의 물리적 어드레스를 제공하기 위해 사용되는 스테이지 2 변환 테이블 엔트리의 스테이지 2 액세스 퍼미션(68)에서 명시될 수도 있다. 이것은 악의적인 당사자에 의한 공격 하에 있는 오퍼레이팅 시스템 코드 또는 잘못 기능하는 코드에 의한 손상에 대비하여 그들 스테이지 1 변환 엔트리를 보호하는데, 그 이유는 PRO 퍼미션이 대응하는 메모리 영역에 기록하는 능력을 명령어 타입의 제한된 서브세트

로 제한하여, 스테이지 1 변환 테이블의 손상에 취약할 수 있는 코드의 라인의 수를 크게 감소시키기 때문이다. 그럼에도 불구하고, 오퍼레이팅 시스템에 의한 페이지 동작의 효율성의 손상을 방지하기 위해, 변환 테이블 엔트리의 액세스 추적 메타데이터를 업데이트하기 위한 하드웨어 생성 기록 요청은 그러한 영역에 대해 계속 허락될 수 있다.

[0153] 도 11이 사전 결정된 실행 상태(EL1)에 있을 때 발행되는 메모리 기록 액세스 요청에 대해 수행되고 있는 부분 판독 전용 체크를 도시하지만, 그것은 다른 실행 상태에 대해서도 또한 수행될 수 있다. 그러나, 최소의 특권이 있는 실행 상태(EL0)에서 발행되는 기록 요청의 경우, 스테이지 1 변환 테이블을 유지하는 메모리 영역이 임의의 경우에, 스테이지 1 액세스 퍼미션(66)에 의해, EL0에서 동작하는 애플리케이션 레벨 코드에 액세스할 수 없는 것으로 마킹될 것이고, 따라서 그러한 기록 요청이 스테이지 1 변환 테이블을 손상시키는 것을 스테이지 1 퍼미션 체크가 이미 방지할 수도 있을 가능성이 있기 때문에, 이것은 필수적인 것은 아닐 수도 있다. 또한, EL2 또는 EL3에서 실행되는 코드의 경우, 변환 테이블은 2 스테이지 어드레스 변환을 사용하지 않을 것이고 스테이지 1 변환 테이블 엔트리를 저장하는 영역에 대한 기록을 제한할 가능성이 없는데, 그 이유는, 하이퍼바이저 또는 모니터 코드가, 스테이지 1 변환 테이블에 대한 어떤 업데이트가 허용되는지에 대한 제한을 제어하고 있는 코드일 수도 있기 때문이다. 따라서, 몇몇 구현에는 부분 판독 전용 체크를 현재의 실행 상태가 EL1인 경우에만 수행되도록 제한할 수도 있다. 다른 구현에는 어떤 실행 상태가 현재의 실행 상태인지에 관계없이 부분 판독 전용 체크를 적용할 수도 있다.

[0154] 스테이지 1 변환 테이블 엔트리에 대한 보호 엔트리 인코딩

[0155] 기록 타입의 제한된 서브세트가 아닌 기록 요청에 대해 소정의 메모리 영역을 판독 전용으로 마킹하기 위한 PRO 퍼미션의 사용은 손상에 대비하여 스테이지 1 테이블을 보호하는 데 도움이 된다. 그러나, 도 6에서 도시되는 바와 같이, 통상적으로 변환 테이블 구조의 주어진 레벨에서의 주어진 변환 테이블은 메모리의 전체 페이지를 차지하며, 따라서 PRO 퍼미션은, 승인되지 않은 액세스에 대비하여 보호될 필요가 있는 민감한 정보를 위해 사용되는 일부 메모리 영역, 및 이러한 보호를 필요로 하지 않는 민감하지 않은 정보만을 포함하는 다른 메모리 영역을 포함할 수도 있는 다양한 메모리 영역에 대한 변환 테이블 엔트리를 포함할 수도 있는, 스테이지 1 변환 테이블 구조의 주어진 레벨에 있는 전체 변환 테이블에 적용될 것이다.

[0156] 하기에서 추가로 논의되는 바와 같이 스테이지 1 변환 테이블 엔트리에 대한 보호 엔트리 인코딩의 사용을 지원하지 않는 구현예를 제공하는 것이 가능하다. 이 경우, 오퍼레이팅 시스템이 민감하지 않은 정보를 포함하는 보호되지 않은 영역과 관련되는 소정의 스테이지 1 변환 테이블 엔트리를 업데이트하는 것을 허용하기 위해, PRO 영역에 기록하는 것이 허용되는 기록 요청 타입 중 하나가 아닌 PRO 영역에 대한 기록 액세스시 생성되는 장애는 EL2에서 동작하는 하이퍼바이저에 대한 트랩을 야기할 수 있으며, 하이퍼바이저는, 그 다음, 오퍼레이팅 시스템이 스테이지 1 변환 테이블 엔트리를 업데이트하기를 소망하는 영역이 오퍼레이팅 시스템에 의해 업데이트되는 것에 대비하여 보호되어야 하는지의 여부를 소프트웨어로 결정할 수 있고, 영역이 보호될 필요가 없는 경우, 오퍼레이팅 시스템에 의해 요청되는 업데이트는 진행하는 것이 허용될 수 있다. 그러나, 실제로, 오퍼레이팅 시스템은 메모리의 보호되지 않은 영역에 대한 변환 테이블 엔트리를 상대적으로 빈번하게 변경하는 것을 필요로 할 수도 있고, 따라서 업데이트가 요구될 때마다 하이퍼바이저에 대한 트랩핑은 성능의 관점에서 느려질 수도 있다.

[0157] 스테이지 1 변환 테이블 엔트리에 대한 보호된 엔트리 인코딩(예를 들면, 비트(76, 78)를 사용함)은 하이퍼바이저 개입 없이 오퍼레이팅 시스템이 보호되지 않은 엔트리에 대한 업데이트를 행하는 것(그리고, 옵션 사항으로, 몇몇 구현예의 경우, 하이퍼바이저 개입 없이 보호된 엔트리의 업데이트 비제한 비트에 대한 업데이트를 행하는 것)을 허용하는 데 도움이 된다. 이것은 하이퍼바이저 개입이 더욱 자주 회피될 수 있기 때문에 성능 향상에 도움이 되고, 심지어 주어진 스테이지 1 변환 테이블이 메모리의 PRO 영역에서 저장되는 경우에도 그것이 보호된 엔트리 및 보호되지 않은 엔트리의 혼합물을 포함하기 때문에, 성능 향상에 도움이 된다. 명령어 디코더(10) 및 프로세싱 회로부(4)의 실행 스테이지(16)는, 보호 엔트리 체크 타입의 판독 체크 기록(RCW) 명령어로 칭해지는 적어도 하나의 타입의 명령어를 지원하는데, 이것은 액세스된 메모리 시스템 위치에 저장되는 데이터를 체크하여 그것이 보호된 스테이지 1 변환 테이블 엔트리의 인코딩과 일치하는 값을 갖는지의 여부를 체크하는 보호된 엔트리 체크에 따라 조건부로 업데이트가 수행되는 주어진 메모리 시스템 위치에 대한 업데이트를 요청하기 위해 사용될 수 있다. 액세스된 메모리 시스템 위치에 있는 데이터가 보호된 스테이지 1 변환 테이블 엔트리인 데이터와 일치하는 값을 갖는다는 것이 확인되면, 그러면, 주어진 메모리 시스템 위치의 비트의 적어도 업데이트 제한 서브세트에 대한 업데이트는 억제된다. RCW 명령어는 EL1에서 실행되는 오퍼레이팅 시스템에 의해 실행되는 것이 허용된다. RCW 명령어는 대응하는 스테이지 2 변환 테이블 엔트리에 의해 PRO 퍼미션을 사

용하여 마킹되는 메모리 영역의 데이터를 업데이트하는 것이 허용되는 명령어의 제한된 서브세트 중 하나이다.

[0158]

보호된 엔트리 체크는 보호 엔트리 체크 회로부(54)에 의해 하드웨어에서 수행되는데, 이것은 도 7에서 도시되는 바와 같이 MMU(28) 내에서, 또는 데이터를 저장하는 메모리 시스템 위치에 더 가깝게, 예를 들면, 캐시, 인터커넥트 또는 메모리 컨트롤러와 같은 메모리 시스템 컴포넌트(56) 내에서 구현될 수 있다. MMU(28)(또는 파이프라인의 실행 스테이지(16))에서 구현되는 보호 엔트리 체크 회로부(54)의 일부 부분 및 메모리 스토리지에 더 가깝게 구현되는 다른 부분과 함께 로직의 분산된 세트를 사용하는 것도 또한 가능하다. 보호된 엔트리 체크에서 수행되는 판독, 체크 및 기록 동작은 원자성(분할할 수 없음) 동작으로서 수행되며, 그 결과, RCW 명령어를 실행하는 스레드와 동일한 프로세서 회로부 상에서 실행되는 다른 스레드에 의해 또는 다른 프로세서 코어에 의해 주어진 메모리 시스템 위치에 대한 액세스가 요청되는 경우, 다른 액세스는 RCW 명령어를 프로세싱하기 이전의 주어진 메모리 시스템 위치의 값, 또는 RCW 명령어를 프로세싱한 이후의 주어진 메모리 시스템 위치의 값 중 어느 하나를 볼 수 있고, RCW 명령어의 임의의 부분적인 효과를 볼 수 없으며, 또한 주어진 메모리 시스템 위치에 있는 데이터를 판독하는 보호된 엔트리 체크와 RCW 명령어에 응답하는 주어진 메모리 시스템 위치에 있는 데이터의 후속하는 업데이트 사이의 기간에서는 주어진 메모리 시스템 위치에 대한 기록이 주어진 메모리 시스템 위치를 업데이트하는 것이 가능하지 않다. 원자성(예를 들면, 판독과 기록 사이의 기간에 다른 동작에 의한 액세스에 대비하여 기억 위치를 잠그거나, 또는 그 기간에 충돌하는 액세스를 허용하지만 그러나 판독과 기록 사이의 기간에 충돌하는 액세스가 검출되는 경우 RCW 동작을 재시작함)을 강제하기 위해, 메모리 시스템 위치에 원자성 판독/기록 동작을 강제하기 위한 임의의 공지된 기술이 사용될 수 있다. 판독, 체크 및 기록의 원자성을 강제하는 것에 의해, 이것은 잘못된 프로세싱 결과로 이어질 수 있는 경쟁 조건을 방지한다. 판독, 체크 및 기록 동작이 원자적으로(atomically) 수행되기 때문에, 업데이트되고 있는 보관 위치에 더 가까운 메모리 시스템 컴포넌트(56) 내에서 보호 엔트리 체크 회로부(54)가 구현되어, 기록이 진행되고 데이터를 저장하는 메모리 시스템 위치에 기록 데이터를 반환할 수 있기 이전에 체크가 수행될 프로세서 파이프라인(4)의 다른 엘리먼트 또는 MMU(28)까지 내내 판독 데이터가 반환되어야 하는 경우의 지속 기간과 비교하여, 판독과 기록 사이의 기간의 지속 기간을 감소시키는 것이 유용할 수 있다.

[0159]

도 8에서 도시되는 예에서, 유효한 및 무효한 스테이지 1 변환 테이블 엔트리 둘 모두는 엔트리가 보호된 엔트리로서 명시되는 것을 허용하는 인코딩을 갖는다. 그러나, 무효한 엔트리가 보호되는 것으로 명시될 수 있는 것이 필수적인 것은 아니며, 다른 예는 무효한 엔트리 포맷으로부터 보호된 엔트리 인코딩 비트필드(78)를 생략할 수 있다. 또한, 상이한 구현에는 RCW 명령어에 응답하여 보호된 엔트리의 얼마나 많은 비트가 그들의 업데이트가 억제되게 하는지에 대한 상이한 옵션을 선택할 수도 있다. 몇몇 구현에는 RCW 명령어에 응답하여 보호된 엔트리에 대한 임의의 업데이트를 방지할 수 있는데, 업데이트는 주어진 메모리 시스템 위치의 데이터가 보호된 엔트리로서 인코딩되지 않는 경우에만 RCW 명령어에 응답하여 허락된다. 다른 구현에는 비트의 업데이트 제한 서브세트 및 비트의 업데이트 비제한 서브세트를 명시할 수 있고, 따라서 비트의 업데이트 비제한 서브세트에 대한 업데이트는 심지어 보호된 엔트리에 대해서도 허용 가능할 수도 있다. 예를 들면, 액세스 플래그(70) 및 더티 비트 수정자(72)와 같은, 보안성이 중요하지 않은 일부 필드가 있을 수도 있는데, 이들은 심지어 보호된 엔트리에서도 업데이트되는 것이 허용될 수 있다. 어떤 비트가 업데이트 제한되는지 또는 업데이트 제한되지 않는지는 주어진 아키텍처에 대해 프로그래밍 가능하지 않은 방식으로 고정될 수 있거나, 또는 제어 레지스터에 저장되는 선택 값을 사용하여 소프트웨어에 의해 프로그래밍 가능할 수 있다. 예를 들면, 선택 값은 여러 가지 상이한 모드 중 하나로부터 선택될 수 있는데, 각각의 모드는 비트가 업데이트 제한되는 상이한 설정과 관련된다. 그러나, 도 7의 예에서, 선택 값은 업데이트 제한 비트의 포지션을 식별하는 비트 마스크로서 인코딩되는데, 선택 값은 마스크 레지스터(예를 들면, 유효한 및 무효한 엔트리에 대한 업데이트 제한 비트를 각각 선택하기 위한 도 7에서 도시되는 RCWMask_Valid_EL2 및 RCWMask_Invalid_EL2 레지스터 - RCW_Mask_Invalid_EL2 레지스터는 무효한 엔트리가 보호된 엔트리로서 인코딩될 수 없는 구현예에서 생략될 수 있음)에 저장된다. 선택 값에 대한 업데이트는 임계 특권 레벨 또는 그 이상에서 실행되는 프로그램 명령어로 제한될 수도 있는데, 예를 들면, EL2 또는 EL3에서 실행되는 명령어만이 선택 값을 업데이트하는 것이 허용될 수도 있다. 선택 값의 상태와 관계없이, 업데이트 제한되도록 강제되는 일부 비트가 있을 수도 있다. 예를 들면, 엔트리가 보호된 엔트리인지의 여부를 인코딩하기 위해 사용되는 비트필드(76, 78)는 어떤 다른 비트가 업데이트될 수 있는지에 관계없이 업데이트되는 것이 허용될 수 없을 수도 있다. 일반적으로, RCW 명령어는 특정한 엔트리가 보호되는지 또는 보호되지 않는지의 여부를 상태를 변경하는 것이 금지될 수도 있다. 비트필드(76, 78)의 설정은, 대신, 보호된 엔트리를 저장하는 메모리 영역을 부분 판독 전용으로서 마킹하지 않는 스테이지 2 변환 테이블의 상이한 세트를 사용하여 메모리에 액세스하고 있을 수도 있는, EL2 또는 EL3과 같은, 더 높은 레벨의 특권에서 실행되는 프로그램 코드에 의해 제어될 것이고, 따라서 보호된 스테이지 1 엔트리의 비트

필드(76, 78)를 표준 범용 저장 명령어를 사용하여 설정할 수 있다.

- [0160] 보호 엔트리 체크가 구현될 수 있는 다양한 방식이 존재한다. 하나의 예에서, 액세스된 메모리 시스템 위치에 저장되는 엔트리가 유효하고 보호되는 경우, 그러면, 그것은 변경될 수 없으며, 그것이 무효한 경우, 그러면, (엔트리가 또한 보호되게 만들어지지 않는 한) 그것을 유효하게 만드는 것을 비롯하여, 그것은 수정될 수 있다. 대안적으로, 유효한 엔트리에 보호된 비트필드(76)가, 그리고 무효한 엔트리에 보호된 비트필드(78)가 있을 수 있으며, 보호된 엔트리에 대한 어떠한 변경도 허용되지 않을 수도 있고, 그렇지 않으면 (엔트리가 보호되는지의 여부의 상태를 변경하는 것) 외에 보호되지 않은(유효한 또는 무효한) 엔트리에 대한 업데이트는 허용된다. 다른 예는 유효한 엔트리에서 보호된 비트필드(76)를, 그리고 무효한 엔트리에서 보호된 비트필드(78)를 제공할 수 있고, RCW 명령어는 보호되지 않는 경우 엔트리에 (그것을 보호되게 만드는 것 외에) 임의의 변경을 행하는 것이 허용될 수도 있지만, 만약 그것이 보호되는 경우, 그러면, RCW 명령어는 유효한 엔트리에 대해 설명되는 바와 같이 일부 비트에, 그리고 무효한 엔트리의 일부 비트에 변경을 행할 수 있다. 몇몇 예에서, 어떤 비트가 보호되는지를 설명하기 위한 메커니즘은 보호 또는 유효한 필드를 제외한 임의의 필드의 하드웨어 고정식 아키텍처 선택(hardwired architectural choice)이다. 대안적으로, 유효한 엔트리에서 어떤 비트가 보호되는지를 설명하기 위한 메커니즘은 EL2에 의해 소유되는 마스크 레지스터를 사용할 수 있고, 및/또는 무효한 엔트리에서 어떤 비트가 보호되는지를 설명하기 위한 메커니즘은 EL2에 의해 소유되는 마스크 레지스터이다. 유효한 및 무효한 엔트리 각각에 대해 업데이트 제한 비트를 정의하기 위한 상이한 선택이 가능하다 - 예를 들면, 한 가지 접근법은 유효한 엔트리의 업데이트 제한 비트를 정의하기 위한 마스크 레지스터를 정의하는 것일 수 있지만, 그러나 무효한 엔트리에 대해서는 업데이트 제한 비트를 하드웨어 고정하는(hardwire) 것일 수 있다.
- [0161] 보호된 엔트리의 일부 필드를 업데이트하는 것이 허용되는 경우 체크를 적용하는 방법에서 변동이 또한 있을 수 있는데, 두 가지 예시적인 옵션은 다음과 같다:
- [0162] A) 보호 엔트리 체크 회로부(54)는 엔트리가 유효한지 및/또는 보호되는지를 체크하기 위해 현재의 엔트리를 판독하고, 그 다음, (기록되는 것이 허용되지 않는 업데이트 제한 비트를 RCW 명령어가 변경하려고 시도하고 있는지를 체크하지 않고) 기록되는 것이 허용되는 비트만을 저장한다.
- [0163] B) 보호 엔트리 체크 회로부(54)는 현재의 엔트리를 판독하여 그 엔트리가 유효한지 및/또는 보호되는지를 체크한다. 보호되어 있고, RCW 명령어가 기록되는 것이 허용되지 않는 임의의 비트를 변경하려고 시도하고 있진 않는 경우, 기록은 발생할 수 있지만, 그러나 RCW 명령어가 임의의 업데이트 제한 비트를 업데이트하려고 시도하는 경우, 기록은 실패하고 아무것도 기록되지 않는다.
- [0164] A 및 B 둘 모두는 실행 가능한 대안예이다.
- [0165] RCW 명령어에 응답하여, 기록이 성공적이었는지 또는 실패하였는지의 여부의 표시가 프로세싱 회로부(4)의 제어 레지스터에 저장되는 조건 상태 플래그에 기록될 수 있다. 예를 들면, 조건 상태 레지스터는, 프로세싱 동작의 결과에 따라 설정될 수 있고 조건부 명령어에 의해 테스트될 수 있는 다수의 조건 플래그(예를 들면, 네거티브 플래그(N), 제로 플래그(Z), 캐리 플래그(C) 및 오버플로우 플래그(V))를 포함할 수도 있는데, 조건부 명령어는, 그 다음, 조건 상태 플래그의 상태에 따라 조건부적으로 동작(예컨대 분기 동작 또는 산술/논리 연산)을 트리거할 수도 있다. 그러므로, RCW 명령어에 의해 요청되는 기록의 임의의 부분이 성공하지 못하였는지의 여부에 기초하여 조건 상태 플래그를 업데이트하여, 기록이 성공하였는지의 여부를 체크하기 위해 메모리 시스템 위치를 재판독할 필요 없이 기록의 결과에 따라 후속하는 명령어가 액션을 취하는 것을 허용하는 것이 유용할 수 있다.
- [0166] 도 12, 도 13 및 도 14는 보호 엔트리 체크 RCW 명령어의 다수의 변형의 프로세싱을 예시한다. 도 12는 명령어의 저장 변형 또는 스왑 변형 중 어느 하나를 프로세싱하는 예를 예시한다(대부분의 프로세싱 단계는, 변형에 의존하는 최종 단계(726 또는 728) 외에는, 명령어 둘 모두에 대해 동일하다). 저장 및 스왑 변형 둘 모두는 업데이트될 주어진 메모리 시스템 위치의 어드레스를 생성하기 위해 사용되는 어드레스 피연산자를 제공하는 레지스터(Xd), 및 주어진 메모리 시스템 위치에 기록될 새로운 데이터를 제공하는 소스 레지스터(Xs)를 명시한다. 적어도 스왑 변형은 주어진 메모리 시스템 위치로부터 판독되는 이전 데이터가 기록될 목적지 레지스터(Xt)를 또한 명시한다. 이전 데이터가 어떠한 목적지 레지스터에도 기록되지 않는다는 점에서, 저장 변형이 스왑 변형과는 상이하기 때문에, 저장 변형은 어떠한 목적지 레지스터도 명시할 필요가 없다. 그러나, 하나의 예에서, 저장 변형 및 스왑 변형은, 목적지 레지스터(Xt)를 명시하기 위해 사용되는 필드가 사전 결정된 값을 갖는 경우 명령어가 저장 변형으로서 처리될 수도 있다는 점을 제외하면, 동일한 인코딩을 사용하여 인코딩될 수 있다. 예를 들면, 일부 명령어 세트 아키텍처는, 특정한 레지스터 명시자(예를 들면, 레지스터(31))를, 명령어에 대한

피연산자로서 명시되는 경우, 하드웨어에서 제공되는 레지스터로부터 피연산자가 실제로 판독될 필요 없이, 피연산자로 하여금 마치 제로 값을 갖는 것처럼 처리되게 하는 "제로" 레지스터에 대응하는 것으로 정의할 수도 있다. 그러므로, 몇몇 예에서, 목적지 레지스터(Xt)가 "제로" 레지스터(X31)로서 명시될 때, 명령어는 저장 변형으로서 처리될 수 있다. 다른 예에서, 저장 명령어는 완전히 상이한 명령어 타입으로서 스왑 명령어에 인코딩될 수 있다.

[0167] 도 12의 단계(700)에서, 기록될 어드레스 피연산자 및 새로운 데이터("NewData")는 레지스터 명시자(Xd 및 Xs)에 의해 식별되는 레지스터로부터 각각 판독된다. 단계(702)에서, 메모리 액세스를 위한 타겟 어드레스는 어드레스 피연산자에 기초하여 결정된다. 예를 들면, 어드레스 피연산자는 타겟 어드레스를 직접적으로 명시하는 절대 어드레스로서 처리될 수 있거나, 또는 추가적인 기본 레지스터의 값 또는 RCW 명령어 그 자체의 어드레스를 식별하는 프로그램 카운터 값과 같은 참조 어드레스에 대한 오프셋을 정의할 수 있다. 어드레스 피연산자에 기초하여 계산되는 타겟 어드레스는 가상 어드레스이며 MMU(28)에 의해 물리적 어드레스로 변환될 수도 있다. 단계(704)에서, 타겟 어드레스가 현재의 명령어에 액세스 가능한지의 여부에 대한 (보호된 엔트리 체크 이외의) 임의의 MMU 체크가 통과된다는 것을 가정하면, (변환된) 타겟 어드레스에 대응하는 메모리 시스템 위치로부터 판독되는 데이터("ReadData")를 획득하기 위해 판독 요청이 메모리 시스템에 발행된다. 이것은 RCW 명령어의 "판독"에 대응한다.

[0168] 단계(706)에서, 보호 엔트리 체크 회로부(54)는 ReadData가 보호된 스테이지 1(S1) 변환 테이블 엔트리의 인코딩과 일치하는 값을 갖는지의 여부를 결정한다. 이것은 RCW 명령어의 "Check(체크)" 부분에 대응한다. ReadData가 보호된 스테이지 1 변환 테이블 엔트리와 일치하는 값을 갖지 않는 경우, 그러면 단계(708)에서, 보호 엔트리 체크 회로부는 NewData가 보호된 스테이지 1 변환 테이블 엔트리로서 인코딩되는지의 여부를 체크한다. 만약 그렇다면, 단계(710)에서, RCW 명령어에 의해 요청되는 기록은 거부되거나, 또는 대안적으로, RCW 명령어에 의해 NewData로서 명시되는 스테이지 1 변환 테이블 엔트리의 업데이트 비제한 비트는 메모리 시스템 위치의 대응하는 비트에 기록될 수 있는데, 적어도 엔트리가 보호되는지의 여부를 인코딩하기 위해 사용되는 비트는 기록되지 않는 업데이트 제한 비트로서 간주된다. 어느 쪽이든, 단계(712)에서, 조건 상태 플래그는 RCW 명령어에 의해 요청되는 기록 수행의 완전한 실패를 보고하도록 설정된다.

[0169] 단계(706)에서, ReadData가 보호된 스테이지 1 변환 테이블 엔트리로서 인코딩되는 것으로 결정되는 경우, 그러면 단계(716)에서, 보호 엔트리 체크 회로부(54)는, 판독된 데이터가 유효한 또는 무효한 스테이지 1 변환 테이블 엔트리와 일치하는 값을 갖는지의 여부를 결정한다. 판독 데이터가 무효한 보호된 스테이지 1 변환 테이블 엔트리와 일치하는 값을 갖는 경우, 그러면 단계(718)에서, 보호 엔트리 체크 회로부(54)는, NewData가 스테이지 1 변환 테이블 엔트리의 임의의 무효 엔트리 업데이트 레지스터 제한 비트(적어도 엔트리가 보호되는지의 여부를 인코딩하기 위해 사용되는 임의의 비트 포함함)를 변경하려고 시도하는지의 여부를 결정하고, 만약 그렇다면, 그러면 또 다시, 방법은 단계(710 및 712)로 진행하여, 적어도 무효 엔트리 업데이트 레지스터 제한 비트의 업데이트를 억제하고 조건 상태 플래그를 설정한다(역시, 이것은 기록을 완전히 거부하는 것에 의해 또는 업데이트 비제한 비트에만 기록하는 것에 의해 행해질 수 있음).

[0170] 다른 한편으로, 단계(716)에서, ReadData가 보호된 엔트리로서 인코딩되는 유효한 스테이지 1 변환 테이블 엔트리와 일치하는 값을 갖는다는 것이 결정되는 경우, 그러면 단계(720)에서, 보호 엔트리 체크 회로부(54)는, 타겟 어드레스에 대응하는 메모리 시스템 위치로부터 판독되는 스테이지 1 변환 테이블 엔트리의 임의의 유효 엔트리 업데이트 제한 비트(적어도 엔트리가 보호되는지의 여부를 인코딩하기 위해 사용되는 임의의 비트를 포함함)를 변경하려고 NewData가 시도하는지의 여부를 결정한다. 만약 그렇다면, 그러면 또 다시, 단계(710)에서, 기록은 거부되거나 또는 업데이트 비제한 비트만을 기록하도록 허락되고, 단계(712)에서, 조건 상태 플래그는 기록의 실패를 보고하도록 설정된다.

[0171] 무효한 변환 테이블 엔트리가 보호된 엔트리로서 인코딩되는 것을 지원하지 않는 구현예에서, 단계(716 및 718)는 생략될 수 있고 ReadData가 보호된 엔트리로서 인코딩되는 경우 방법은 단계(706)로부터 단계(720)로 바로 진행될 수 있다.

[0172] 단계(708)에서, ReadData가 보호된 엔트리가 아니었던 경우에 기록될 NewData, 또는 단계(718 또는 720)에서, (ReadData가 보호된 엔트리가었던 경우에 기록될) NewData가 임의의 업데이트 제한 비트를 변경하려고 시도하지 않는 경우, 그러면 단계(714)에서, RCW 명령어의 기록은 허락되고, 따라서 NewData는 타겟 어드레스에 대응하는 메모리 시스템 위치에 기록된다.

[0173] (적어도 부분적으로) 기록이 성공하였는지 또는 거부되었는지의 여부에 관계없이, 단계(724)에서, 동작은 실행

되고 있는 현재의 RCW 명령어가 저장 변형인지 또는 스왑 변형인지의 여부에 따라 변한다. 명령어가 저장 변형인 경우, 그러면 단계(726)에서, 어떠한 추가적인 액션도 필요로 되지 않으며 ReadData는 어떠한 목적지 레지스터에도 기록되지 않는다. 명령어가 스왑 변형인 경우, 그러면 단계(728)에서, ReadData는 목적지 레지스터(Xt)에 기록된다. 몇몇 구현에는 저장 및 중지 변형 둘 모두를 지원하지 않을 수도 있는데, 이 경우, 단계(724)는 생략될 수 있고 방법은 단계(714)로부터 단계(726 및 728) 중 관련된 단계로 바로 진행될 수 있다.

[0174] 단계(704)에서의 관독, 단계(714)에서의 기록 및 단계(704와 714) 사이에서 수행되는 다양한 체크 동작은, 메모리 시스템 위치에 대한 개입하는 기록이 단계(704)에서의 관독과 단계(710 또는 714)에서의 기록 사이에서 발생하는 것, 또는 실행의 다른 스레드가 동작의 세트를 수행하는 것의 부분적인 결과를 확인하는 것이 가능하지 않은 분할할 수 없는 동작의 세트로서, 원자적으로 수행된다.

[0175] RCW 명령어의 저장/스왑 변형의 기능성(functionality)을 나타내기 위한 몇몇 예시적인 의사코드가 하기에서 나타내어진다. 이것은 단지 하나의 예에 불과하며 도 12에서 도시되는 더욱 일반적인 표현보다 더 구체적이라는 것이 인식될 것이다. 예를 들면, 의사코드는, 유효하고 보호된 엔트리와 무효하고 보호된 엔트리 둘 모두에 대한 업데이트 제한 비트의 선택이 대응하는 마스크 레지스터에서 정의된다는 것을 가정하지만, 그러나 상기에서 논의되는 바와 같이, 이것은 필수적인 것은 아니며 다른 예는 어떤 비트가 업데이트 제한되는지의 선택을 하드웨어 고정할 수 있거나, 또는 보호된 엔트리의 모든 비트를 업데이트 제한되는 것으로 간주할 수 있다. 이 예에서, 조건 상태 플래그의 "제로" 플래그(Z)를 1로 설정하는 것에 의해 기록의 실패가 보고된다. 다른 예는 기록의 실패를 인코딩하기 위해 상이한 조건 상태를 선택할 수 있다. 하기에서 의사코드로 나타내어지는 기능성은 상기에서 설명되는 보호 엔트리 체크 회로부(54)의 하드웨어 회로부 로직을 사용하여 구현될 수 있고, 따라서 의사코드는, 소프트웨어가 이들 기능을 실행한다는 것을 암시하는 것이 아니라, 회로 로직의 논리적 기능성을 나타내도록 의도된다는 것이 인식될 것이다.

[0176] bits(4) pstate_bits = "0000";

[0177] NewData = X[s];

[0178] Address = X[d];

[0179] ReadData<63:0> = Mem[Address];

[0180] protected = ReadData<0> == '1' ? ReadData<Valid_Protected_Bitpos>: ReadData<Invalid_Protected_Bitpos>;

[0181] if protected == '0' then

[0182] new_protected = NewData<0> == '1' ? NewData<Valid_Protected_Bitpos>:
NewData<Invalid_Protected_Bitpos>;

[0183] if new_protected == '0' then // can't change the protected field

[0184] Mem[Address] = NewData;

[0185] else

[0186] pstate_bits="0100"; // Z bit set on failure to write

[0187] else

[0188] if ReadData<0> == '0' then // invalid entry

[0189] if NewData<Invalid_Protected_Bitpos> == '0' && NewData<0> == '0'

[0190] then // can't change valid or protected

[0191] Mem[Address] = NewData;

[0192] else

[0193] pstate_bits="0100"; // Z bit set on failure to write

[0194] else

[0195] if (NewData^ReadData) & ~RCWMask_EL2 == Zeros(64) then

```
[0196]     Mem[Address] = NewData;
[0197]     else
[0198]         pstate_bits = "0100"; // Z bit set on failure to write
[0199]     PSTATE.NZCV = pstate_bits;
[0200]     X[t] = ReadData; // return the old value
```

[0201] 도 13은 보호 엔트리 체크 RCW 명령어의 비교 및 스왑(compare-and-swap; CAS) 변형의 프로세싱을 예시하는 흐름도이다. 도 12에서와 동일한 참조 번호를 사용하여 예시되는 도 13의 단계는 상기에서 논의되는 바와 동일하다. ReadData를 저장하고 NewData를 제공하기 위해 사용되는 레지스터에 대해 사용되는 표기법은, 도 12와 비교하여, 도 13에서 교체되었다는 것을 유의한다 - Xt는 이제 NewData를 제공하는 레지스터이고 Xs는 기록 이전에 ReadData의 이전 값으로 기록되는 목적지 레지스터이다. 도 13은, CAS 변형이 비교 데이터("CompareData")로 칭해지는 추가적인 소스 피연산자를 명시하기 때문에 단계(700)가 단계(750)로 대체된다는 점에서, 도 12와는 상이하다. 이 예에서, 명령어 인코딩에서 필요로 되는 레지스터 필드의 수를 감소시키기 위해, 단계(728)에서 ReadData를 기록하기 위한 목적지 레지스터로서 사용되는 동일한 레지스터(Xs)에서 CompareData가 명시되지만, 그러나 이것은 필수적인 것은 아니며 다른 예는 Xs와는 별개의 추가적인 레지스터에서 비교 데이터를 명시할 수 있다.

[0202] 단계(702 및 704)는 도 12에서와 동일하다. 단계(704) 이후, 타겟 어드레스에 대응하는 주어진 메모리 시스템 위치로부터 판독되는 ReadData와 레지스터(Xs)의 RCW 명령어에 의해 명시되는 CompareData를 비교하기 위해 추가적인 단계(752)가 수행된다. 보호 엔트리 체크 회로부(54)는, 비교에 기초하여, 비교 조건이 충족되는지의 여부를 결정한다. 예를 들면, 비교 조건은 CompareData와 ReadData가 동일한지의 여부일 수 있거나, 또는 보다 더 작음, 보다 더 큼, 동일하지 않음, 등등과 같은 다른 비교 조건에 기초할 수 있다. 몇몇 구현에는 비교 조건을 동일한 조건인 것으로 고정할 수도 있고, 한편, 다른 구현에는 명령어의 파라미터가 어떤 타입의 비교를 수행할지를 선택하는 것을 허용할 수 있다. 비교 조건이 충족되지 않는 경우, 그러면 단계(754)에서, RCW 명령어에 의해 요청되는 기록은 거부되고 조건 상태 플래그는 기록 실패가 비교 조건 실패에 의해 야기된다는 것을 보고하도록 설정된다(예를 들면, 비교 조건 실패에 대해 네거티브 플래그(N)를 설정함). 비교 조건이 충족되는 경우, 그러면 단계(706, 708, 710, 714, 716, 718, 720)는 도 12에서와 동일한 방식으로 수행되어 타겟 어드레스에 대응하는 메모리 시스템 위치로부터 판독되는 데이터의 보호된 엔트리 상태를 체크하고 체크의 결과에 기초하여 기록이 허락되는지의 여부를 제어한다. 기록이 허락되는 경우, 그러면 단계(714)는 NewData를 메모리 시스템 위치에 기록하도록 기록이 진행되는 것을 허용하기 위해 앞서 논의되는 바와 동일하다. 단계(710)에서 기록이 거부되거나 또는 부분적으로만 수행되는 경우, 그러면 도 12의 후속하는 단계(712)는 도 13의 단계(762)로 대체되는데, 이것은 기록 실패를 보고하도록 조건 상태 플래그를 설정한다는 점에서 동일하지만, 그러나 비교 및 스왑 변형의 경우, 이것은 보호 엔트리 체크 실패에 의해 실패가 야기되었다는 것을 구별할 수도 있으며, 조건 상태 플래그를 상이한 값으로 비교 조건 실패의 경우에는 단계(754)에서 사용되는 값과는 상이한 값으로 설정한다.

[0203] 비교 조건이 충족되는지 또는 아닌지 또는 보호 엔트리 체크가 성공적인지 또는 아닌지의 여부에 관계없이, 단계(724, 726 및 728)는, ReadData를 목적지 레지스터(Xs)에 옵션 사항으로 기록하기 위해 도 12에서와 동일하다(역시, 목적지 레지스터 명시자가 사전 결정된 값, 예를 들면, 31을 명시할 때, 목적지 레지스터(Xs)의 업데이트를 억제하는 것이 가능함).

[0204] RCW 명령어의 CAS 변형의 기능성에 대한 하나의 가능한 구현예를 나타내는 의사코드 예가 하기에서 나타내어진다:

```
[0205] bits(4) pstate_bits = "0000";
[0206] NewData = X[t];
[0207] CompData =X[s];
[0208] Address = X[d];
[0209] ReadData<63:0> = Mem[Address];
```

```

[0210] if ReadData == CompData then
[0211] protected = ReadData<0> == '1' ? ReadData<Valid_Protected_Bitpos>:
[0212] ReadData<Invalid_Protected_Bitpos>;
[0213] if protected == '0' then
[0214] new_protected = NewData<0> == '1' ? NewData<Valid_Protected_Bitpos>:
[0215] NewData<Invalid_Protected_Bitpos>;
[0216] if new_protected == '1' then // can't change the protected field
[0217] pstate_bits="0100"; // Z bit set on failure to write
[0218] else
[0219] if ReadData<0> == '0' then // invalid entry
[0220] if NewData<Invalid_Protected_Bitpos> == '1' ||
[0221] NewData<0> == '1' then // can't change valid or protected
[0222] pstate_bits="0100"; // Z bit set on failure to write
[0223] else
[0224] if (NewData^ReadData) & ~RCWMask_EL2 == Zeros(64) then
[0225] Mem[Address] = NewData;
[0226] else
[0227] pstate_bits = "0100"; // Z bit set on failure to write
[0228] else
[0229] pstate_bits = "1000"; // new return value on N bit of a CAS failure
[0230] PSTATE.NZCV = pstate_bits;
[0231] X[s] = ReadData; // return the old value.
[0232] 이 의사 코드 예는 비교 조건 평가가 보호된 엔트리 체크보다 우선되는 경우를 도시하며, 따라서 비교 조건이
실패하지만 그러나 또한 엔트리가 보호되고 단계(708, 718, 720)에서의 체크가 실패하는 경우, 그러면, 조건 상
태 코드는 N 비트를 사용하여 CAS 실패를 나타낼 것이다. 엔트리가 보호되고 단계(708, 718 또는 720)에서의
체크가 실패하는 경우, 그러면 심지어 비교 조건이 실패되었더라도, 그러면 조건 상태 코드가, Z 비트를 사용하
여, 보호된 엔트리 체크에 기인하는 기록 실패를 나타내도록, 보호된 엔트리 체크가 우선되는 것이 또한 가능할
것이다. 예를 들면, 도 13은, 단계(706) 이전 대신 단계(714) 이후에, 단계(752 및 754)를 구현하도록 수정될
수 있다.
[0233] 도 14는 보호 엔트리 체크 RCW 명령어의 비트 설정/클리어 변형의 프로세싱을 예시하는 흐름도이다. 도
14에서, 단계(700-728)는, 도 14에서 NewData가, 소스 레지스터에서 직접적으로 명시되는 것이 아니라, 주소 지
정된(addressed) 메모리 시스템 위치 내에서 설정되도록 또는 클리어되도록 비트의 위치션을 인코딩하는 소스
레지스터(Xs)에서 제공되는 비트 마스크("BitMask")에 기초하여 계산된다는 점을 제외하면, 도 12에서와 동일하
다. 그러므로, 도 12와 비교하여 유일한 차이점은, 단계(700)에서 레지스터(Xs)가 NewData를 직접적으로 명시
하는 것이 아니라 BitMask를 정의한다는 것, 및 단계(704와 706) 사이에서 추가적인 단계(770, 772, 774)가 수
행된다는 것이다. 단계(770)에서, 방법은 비트 설정 또는 비트 클리어 명령어가 실행되고 있는지의 여부에 따
라 상이하다(명령어의 타입 둘 모두를 지원하는 것은 필요하지 않으며, 따라서 이들 명령어 중 단지 하나만이
지원되는 경우, 그러면, 단계(770)는 생략될 수 있고 단계(704) 이후 단계(772, 774) 중 관련된 단계가 수행될
수 있다). 비트 세트 명령어가 실행되고 있는 경우, 그러면 단계(774)에서, NewData는 주소 지정된 메모리 시
스템 위치로부터 판독되는 ReadData와 RCW 명령어에 의해 명시되는 BitMask에 대해 비트 단위 OR 연산을 수행하
는 것의 결과로 설정된다(그러므로, 비트 설정 명령어의 경우, BitMask에서 '1'을 가지고 마킹되는 비트 위치선

```

은 주소 지정된 메모리 시스템 위치 내에서 1로 설정될 것을 요청받는다). 대조적으로, 단계(772)에서, 비트 클리어 명령어가 실행되고 있는 경우, 그러면, NewData는 ReadData와 BitMask의 역에 대한 비트 단위 AND 연산을 수행하는 것의 결과로 설정된다(그러므로, 비트 클리어 명령어의 경우, BitMask에서 '1'을 가지고 마킹되는 비트 포지션은 주소 지정된 메모리 시스템 위치 내에서 0으로 클리어될 것을 요청받는다). 비트 세트 및 비트 클리어 변형 둘 모두의 경우, BitMask의 '0' 비트에 대응하는 ReadData의 비트는 NewData에서 그들의 원래의 값을 유지하며, 따라서 RCW 명령어에 의해 변경되지 않는다. 단계(772 또는 774)에서 NewData를 계산한 이후, 도 14의 후속하는 단계는 모두 도 12에서와 동일하다.

[0234] RCW 명령어의 비트 설정/클리어 변형의 기능성에 대한 하나의 가능한 구현예를 나타내는 의사코드 예가 하기에
서 나타내어진다:

[0235] bits(4) pstate_bits = "0000";

[0236] NewData = X[s];

[0237] Address = X[d];

[0238] ReadData<63:0> = Mem[Address];

[0239] protected = ReadData<0> == '1' ? ReadData<Valid_Protected_Bitpos>: ReadData<Invalid_Protected_Bitpos>;

[0240] NewData = ReadData & ~NewData;//CLR or NewData = ReadData | NewData;//SET

[0241] if protected == '0' then

[0242] new_protected = NewData<0> == '1' ? NewData<Valid_Protected_Bitpos>:
NewData<Invalid_Protected_Bitpos>;

[0243] if new_protected == '0' then // can' t change the protected field

[0244] Mem[Address] = NewData;

[0245] else

[0246] pstate_bits="0100"; // Z bit set on failure to write

[0247] else

[0248] if ReadData<0> == '0' then // invalid entry

[0249] if NewData<Invalid_Protected_Bitpos> == '0' && NewData<0> == '0'

[0250] then // can' t change valid or protected

[0251] Mem[Address] = NewData;

[0252] else

[0253] pstate_bits="0100"; // Z bit set on failure to write

[0254] else

[0255] if (NewData^ReadData) & ~RCWMask_EL2 == Zeros(64) then

[0256] Mem[Address] = NewData;

[0257] else

[0258] pstate_bits = "0100"; // Z bit set on failure to write

[0259] PSTATE.NZCV = pstate_bits;

[0260] X[t] = ReadData; // return the old value

[0261] 본 출원의 모든 흐름도의 경우, 동일한 기능성이 상이한 순서로 또한 구현될 수 있고, 그 결과, 흐름도가 단계
의 순차적인 순서를 도시하지만, 서로 의존하지 않는 임의의 단계의 경우, 단계를 재정렬하는 것 또는 그들을
적어도 부분적으로 병렬로 수행하는 것이 가능하다는 것이 인식될 것이다.

[0262] 그러므로, 도 6에서 도시되는 바와 같이, 민감한 메모리 페이지(예를 들면, 민감한 데이터 또는 명령어에 대해 사용될 것으로 예상되는 페이지)에 대응하는 중간 어드레스 매핑에 도달하도록 변환 테이블 트리 구조를 순회하기 위해 사용되는 다양한 스테이지 1 변환 테이블 엔트리는 보호된 엔트리 인코딩을 가지도록 설정될 수 있다. 이것은, 심지어 (예를 들면, 페이징 동작의 결과로서) 페이지에 대한 액세스를 금지하기 위해 이들 엔트리 중 하나가 일시적으로 무효한 것으로 설정되는 경우에도 해당될 수도 있다. 한편, 보호된 엔트리와 동일한 페이지 테이블의 다른 엔트리가 업데이트는 업데이트가 덜 제한되는 보호되지 않은 엔트리일 수 있다. 보호된 엔트리 인코딩을 지원하는 것에 의해, 이것은 메모리의 PRO 영역에서 저장되는 주어진 페이지 테이블이 보호된 엔트리 및 보호되지 않은 엔트리의 혼합물을 포함하는 것을 가능하게 만드는데, 보호되지 않은 엔트리는, 상기에서 논의되는 변형 중 임의의 것에 따라 RCW 명령어를 실행하는 것에 의해, 하이퍼바이저 개입 없이 EL1에서 실행되는 오퍼레이팅 시스템 코드 실행에 의해 업데이트될 수 있고, 한편 보호된 엔트리는 더 엄격한 제한이 그들에게 부과되게 할 수 있다. 보호 엔트리 체크를 트리거하기 위해 전용 타입의 명령어(예를 들면, 상기에서 언급된 RCW 명령어의 변형 중 하나 이상)가 제공되고, 다른 타입의 명령어(예컨대 대부분의 저장 명령어)가 메모리의 PRO 영역을 업데이트할 수 있는 것을 PRO 퍼미션이 방지하기 때문에, 이것은 메모리에 대한 모든 기록에 대해 보호된 엔트리 체크를 실행할 필요성 - 이것은 성능의 관점에서 매우 비용이 많이 들 것임 - 을 방지한다.

[0263] 안티 앨리어싱 체크

[0264] 상기에서 논의되는 메커니즘은 (악의적인 당사자에 의해 해킹되는 코드에 의해 고의적인 또는 잘못 작성된 코드에 의해 우발적인) 손상에 대비하여 특정한 스테이지 1 변환 엔트리를 보호하는 데 유용하다. 그러나, 도 15에서 도시되는 바와 같이, 동일한 중간 어드레스 매핑(62)을 명시하는 스테이지 1 변환 테이블 엔트리를 정의하는 것에 의해, 여러 가지 상이한 가상 어드레스가 동일한 중간 어드레스에 앨리어싱하는 것이 가능하다. VA1을 IPA로 매핑하는 스테이지 1 변환 엔트리의 제1 세트에 대한 상기에서 논의되는 보호 엔트리 인코딩 및 PRO 퍼미션에 의해 제공되는 보호를 공격자가 우회할 수 있는 것을 방지하기 위해, 상이한 가상 어드레스(VA2)를 동일한 중간 어드레스(IPA)로 매핑하는 앨리어싱 스테이지 1 엔트리를 정의하는 것에 의해, 도 8에서 도시되는 바와 같이, 스테이지 2 변환 테이블 엔트리는 메모리의 대응하는 영역이 안티 앨리어싱 특성을 갖는지의 여부를 명시하는 비트필드(79)를 가질 수도 있다. 영역이 안티 앨리어싱 특성을 갖는 것으로 표시되는 경우, 그러면, (가상 어드레스로부터 그 영역에 대한 물리적 어드레스로의 스테이지 1 어드레스 매핑을 명시하는 대응하는 스테이지 1 변환 테이블 엔트리를 찾기 위해 변환 테이블 워크에서 액세스될) 모든 워킹된 스테이지 1 변환 테이블 엔트리가: (a) 메모리의 PRO 영역에 저장되고 보호된 엔트리로서 인코딩되는지, 또는 (b) 메모리의 판독 전용 영역에서 저장되는지의 여부를 체크하기 위해 그 영역에 대한 액세스에 대해 추가적인 안티 앨리어싱 체크가 수행된다. 이것은 그들 워킹된 스테이지 1 변환 테이블 엔트리를 통한 페이지 테이블 워크가 안티 앨리어싱 특성을 갖는 스테이지 2 변환 테이블 엔트리와 관련되는 메모리 영역에 대한 모든 액세스에 대해 실제로 수행되는 것을 반드시 요구하지는 않는다는 것을 유의한다. 매번 전체 변환 테이블 워크를 수행할 필요성을 방지하기 위해, 정보가 TLB(29)에 캐싱되어 안티 앨리어싱 체크가 통과되는지의 여부를 나타낼 수 있는 것(또는 소정의 워킹된 스테이지 1 변환 테이블 엔트리에 대해 그들이 PRO 영역에 저장되는 것으로 및/또는 보호된 엔트리에 저장되는 것으로 이미 결정되었는지의 여부를 나타낼 수 있는 것)이 가능하다.

[0265] 도 16은 안티 앨리어싱 체크를 수행하는 방법을 도시하는 흐름도이다. 단계(800)에서, 소정의 타겟 가상 어드레스(VAx)를 명시하는 메모리 액세스 요청이 발행된다. 단계(802)에서, MMU(28)는, VAx로부터 중간 어드레스(IPAy)로의 스테이지 1 매핑 및 중간 어드레스(IPAy)로부터 타겟 물리적 어드레스(PAz)로의 스테이지 2 매핑에 따라, 타겟 가상 어드레스(VAx)로부터 타겟 물리적 어드레스(PAz)로의 어드레스 변환을 개시한다(스테이지 1 및 스테이지 2 변환을 위해 두 가지 별개의 액션을 수행하는 것이 필수적인 것은 아니다 - 한편 그것은 하나의 옵션이며, 다른 예는, 이미 캐싱된 변환을 위해 VAx로부터 PAz로 직접적으로 매핑할 수 있는 결합된 스테이지 1 및 2 TLB를 구비할 수 있지만, 그러나, TLB에서의 누락시 도 6에서 도시되는 바와 같이 스테이지 1 및 스테이지 2 워크를 수행하는 것에 의해 새로운 VA 대 PA 매핑이 식별된다).

[0266] 단계(804)에서, MMU(28)는 IPAY에 대응하는 스테이지 2 변환 테이블 엔트리(블록 또는 페이지 디스크립터)가 안티 앨리어싱 특성을 나타내는지의 여부를 체크한다. 만약 그렇다면, 그러면 단계(806)에서, MMU는 모든 워킹된 스테이지 1 변환 테이블 엔트리가 안티 앨리어싱 조건을 충족하는지의 여부를 체크한다. 워킹된 스테이지 1 변환 테이블 엔트리는, 가상 어드레스(VAx)에 대응하는 전체 변환 테이블 워크가 수행된 경우, VAx로부터 IPAY로의 S1 어드레스 매핑을 명시하는 대응하는 S1 변환 테이블 엔트리를 찾기 위해 변환 테이블 워크에서 액세스될 엔트리이다(워크는, TLB(29)에서 이미 캐싱된 것에 따라, 현재의 메모리 액세스를 위해 실제로 필요로 되지 않을 수도 있음). 예를 들면, 도 6에서, 워킹된 스테이지 1 변환 테이블 엔트리는 L0, L1 및 L2 스테이지 1 페이

지 테이블에서 D_Table로 마킹되는 엔트리, 및 L3 스테이지 1 페이지 테이블에서 D_Page로 마킹되는 엔트리일 것이다. 안티 앨리어싱 조건은, 다음의 조건 중 어느 하나가 충족되는 경우, 주어진 워킹된 스테이지 1 변환 테이블 엔트리에 대해 충족된다:

- [0267] (a) 주어진 워킹된 스테이지 1 변환 테이블 엔트리는 보호된 엔트리로서 명시되고, 주어진 워킹된 스테이지 1 변환 테이블 엔트리는, PAj를 유도하기 위해 사용되는 스테이지 2 어드레스 매핑을 제공하는 관련 스테이지 2 변환 테이블 엔트리에 의해 PRO 퍼미션이 명시되는 물리적 어드레스(PAj)에서 저장됨; 또는
- [0268] (b) 주어진 워킹된 스테이지 1 변환 테이블 엔트리의 물리적 어드레스(PAj)를 유도하기 위해 사용되는 스테이지 2 어드레스 매핑을 제공하는 관련 스테이지 2 변환 테이블 엔트리에 의해 판독 전용 퍼미션이 명시됨.
- [0269] 도 6의 특정한 예의 경우, 액세스된 페이지의 PA를 유도하기 위해 사용되는 L1 스테이지 1 엔트리에 대한 관련 스테이지 2 변환 테이블 엔트리는 도 6에서 "D_Page"로서 표시되는 스테이지 2 L3 엔트리일 것이다. 도 6에서 도시되는 다른 워킹된 스테이지 1 엔트리의 경우, 안티 앨리어싱 조건이 충족되기 위해 PRO 또는 판독 전용 퍼미션으로 마킹되어야 하는 관련 스테이지 2 L3 엔트리(또는 IPAy에 대응하여 스테이지 2 블록 디스크립터가 발견되는 경우 스테이지 2 L1 또는 L2 엔트리)가 유사하게 있을 것이다(이들은 각각의 워킹된 스테이지 1 변환 테이블의 물리적 어드레스를 획득하기 위해 도시되는 다양한 스테이지 2 변환 워크의 끝에 있는 엔트리이다).
- [0270] 도 16의 단계(806)에서, MMU(28)가 워킹된 스테이지 1 변환 테이블 엔트리 중 임의의 하나 이상이 안티 앨리어싱 조건을 충족하지 않는다는 것을 결정하는 경우, 그러면 단계(808)에서, 메모리 액세스는 거부된다. 예러 핸들링 액션을 수행하기 위해 예외 핸들러(예를 들면, EL2에서 실행됨)를 트리거할 수 있는 장애가 시그널링될 수 있다.
- [0271] 모든 워킹된 스테이지 1 변환 테이블 엔트리가 안티 앨리어싱 조건을 충족하는 경우, 단계(808)는 생략된다. 이 경우, 단계(810)에서, MMU(28)는 임의의 다른 체크가 충족되는지의 여부를 또한 체크한다(예컨대 메모리 영역에 대한 액세스를 위해 설정되고 있는 PRO 퍼미션에 기초한 임의의 체크, RCW 명령어에 응답하여 메모리 액세스 요청이 발행되는지의 보호 엔트리 체크, 또는 다른 부적절한 액세스에 대비하여 보호하기 위한 임의의 다른 특정한 체크, 예컨대 이것을 지원하는 구현에서 보안/비보안 상태에 기초한 체크). 다른 체크 중 임의의 것이 충족되지 않는 경우, 그러면 다시 단계(808)에서, 메모리 액세스는 거부된다(생성되는 장애 타입은 액세스 체크 실패의 원인에 의존할 수도 있음). 그렇지 않으면, 단계(812)에서, 메모리 액세스 요청은 진행되도록 허락된다.
- [0272] 그러므로, 도 6에서 도시되는 바와 같이, 안티 앨리어싱 특성은 액세스되고 있는 메모리 영역에 대한 물리적 어드레스 매핑을 제공하는 스테이지 2 변환 테이블 엔트리에서 설정될 수 있다(이것은 페이지 디스크립터 또는 블록 디스크립터 중 어느 하나일 수 있음). 그 다음, 대응하는 기억 위치에 대한 유효한 액세스에 대해 안티 앨리어싱 특성이 설정되는 경우, 이것은, 그 메모리 영역에 대한 중간 어드레스 매핑을 제공하는 엔트리까지의 경로 상의 모든 워킹된 스테이지 1 변환 테이블 엔트리가, (a) 보호되는 것으로 마킹되고 그들 워킹된 스테이지 1 변환 테이블 엔트리를 저장하는 영역의 물리적 어드레스 매핑을 제공하는 스테이지 2 엔트리가 PRO 퍼미션을 명시하는 것, 또는 (b) PA에 대한 PA 매핑을 제공하는 스테이지 2 엔트리에 의해 판독 전용 퍼미션이 명시되는 그 PA에서 저장되는 것을 요구할 것이다. 이것은, 메모리의 대응하는 영역에 액세스하기 위해 사용될 것으로 예상되는 신뢰되는 어드레스 매핑에 적용되는 보호를 우회하기 위해, 공격자가 새로운 앨리어싱 스테이지 1 어드레스 매핑을 쉽게 셋업할 수 없다는 것을 의미한다. EL1에서 오퍼레이팅 시스템 코드의 거동에 영향을 끼칠 수 있는 공격자는 메모리 액세스를 제어하기 위해 사용되는 스테이지 2 엔트리에 대한 제어를 가지지 않을 것인데, 그 이유는, 이것이 EL1에서 실행되는 코드에 의해 업데이트될 수 없는 VTTBR_EL2 레지스터의 기본 어드레스에 기초할 것이기 때문이다(따라서, PRO 또는 판독 전용 퍼미션으로 하여금, 공격자의 새롭게 생성된 앨리어싱 스테이지 1 어드레스 매핑을 저장하는 영역에 대해 설정되게 할 수 없다). 또한, 스테이지 1 변환 테이블 엔트리를 업데이트하기 위해 EL1에서 실행되는 코드에 대한 메커니즘이 상기에서 논의되는 보호 엔트리 체크 타입의 RCW 명령어(다른 명령어는 메모리의 PRO 영역에 기록하는 것이 금지됨)를 실행하는 것일 수도 있기 때문에 공격자는 스테이지 1 엔트리로 하여금 보호되게 할 수 없고, 그 명령어는 임의의 변환 테이블 엔트리의 보호된 상태를 변경할 수 없다. 또한, 판독 전용 영역은 어떠한 기록 명령어에 의해서도 기록될 수 없다. 따라서, 공격자가 생성할 수 있는 임의의 새로운 엔트리는, VA-IPA 매핑을 획득하기 위해 사용되는 스테이지 1 엔트리가 보호되는지의 여부 및 그들 엔트리가 PRO에 저장되는지 또는 메모리의 판독 전용 영역에 저장되는지의 여부의 안티 앨리어싱 체크 중 적어도 하나를 실패해야 한다.
- [0273] 스테이지 1 최상위 레벨 기본 어드레스에 대한 부적절한 업데이트에 대비한 보호

- [0274] EL1에서 실행되는 오퍼레이팅 시스템 코드의 거동에 영향을 끼칠 수 있는 공격자에 의해 시도될 수도 있는 다른 공격은 기본 어드레스 레지스터(TTBR_EL1)에 의해 나타내어지는 스테이지 1 최상위 레벨 기본 어드레스(L0 스테이지 1 페이지 테이블의 메모리 내 위치를 식별함)를 수정하는 것일 수 있다. 예를 들면, 공격자는, 하이퍼바이저에 의해 검증되는 바와 같이 올바르게 형성되고 올바르게 동작하는 승인된 페이지 테이블 구조의 어드레스를 대체하기 위해, 공격자에 의해 정의되는 스테이지 1 테이블의 새로운 세트의 어드레스를 제공할 수 있다. 또한, 공격자는, 예를 들면 L1, 2 또는 3 페이지 테이블 중 하나의 기본 어드레스를 대입하여 올바른 L0 기본 어드레스를 대체하는 것에 의해, 현존하는 승인된 스테이지 1 페이지 테이블을 예상치 못한 방식으로 사용하도록 MMU(28)를 강제할 수 있고, 그 결과, 액세스되는 테이블이 예상된 레벨로부터 트리의 잘못된 레벨에서 사용되고 그 결과 가상 어드레스의 상이한 비트가 테이블로부터 관련 엔트리를 선택하기 위해 사용되는 것에 기인하여, 상이한 메모리 영역에 대한 다음 레벨 테이블 포인터 또는 어드레스 매핑은 액세스되고 있는 메모리 영역에 대해 잘못 사용될 수도 있다.
- [0275] 도 17은 이들 타입의 공격을 방지하기 위해 수행될 수 있는 스테이지 1 최상위 레벨 기본 어드레스 체크를 예시한다. 단계(1000)에서, 소정의 타겟 가상 어드레스를 명시하는 메모리 액세스 요청에 대해 페이지 테이블 워크가 필요로 된다(필요한 어드레스 매핑이 TLB에서 이미 이용 가능하기 때문에 페이지 테이블 워크를 필요로 하지 않는 메모리 액세스 요청의 경우, 그러면, 임의의 필요한 스테이지 1 최상위 레벨 기본 어드레스 체크는 이들 어드레스 매핑이 앞선 페이지 테이블 워크에 후속하여 캐시에 할당된 때에 이미 행해졌을 것이고, 따라서 다른 체크를 실행하는 것이 필요하지 않다). 단계(1002)에서, 스테이지 1 최상위 레벨 기본 어드레스는 스테이지 1 변환 테이블 기본 어드레스 레지스터(TTBR_EL1)의 값에 기초하여 결정된다. 기본 어드레스는 중간 어드레스인 IPA_base로서 명시된다. 단계(1004)에서, MMU(28)는 IPA_base로부터 PA_base로의 매핑을 제공하는 관련된 스테이지 2 변환 테이블 엔트리에 의해 명시되는 정보를 획득한다. 예를 들면, 이 정보는 TLB에서 이미 캐싱된 경우 TLB(29)로부터 획득될 수 있거나, 또는 스테이지 2 변환 테이블 기본 어드레스 레지스터(VTTBR_EL2)에서 명시되는 스테이지 2 기본 어드레스를 사용하여 스테이지 2 변환 테이블 워크를 수행하는 것에 의해 획득될 수 있다.
- [0276] 단계(1006)에서, MMU(28)는 스테이지 1 최상위 레벨 테이블 부분 판독 전용 체크가 인에이블되는지의 여부를 결정한다. 예를 들면, 이것은 제어 레지스터(예를 들면, 예외 레벨(EL2) 또는 그 이상에서 실행되는 코드로 업데이트가 제한되는 스테이지 2 변환 제어 레지스터(VTCR_EL2))에 저장되는 스테이지 1 최상위 레벨 테이블 부분 판독 전용 체크 인에이블 제어 값에 기초하여 결정될 수도 있다. 대안적으로, 다른 구현에는 스테이지 1 최상위 레벨 테이블 부분 판독 전용 체크를 영구적으로 인에이블되는 것으로 간주할 수도 있다. 스테이지 1 최상위 레벨 테이블 부분 판독 전용 체크가 인에이블되는 경우, 그러면 단계(1008)에서, MMU(28)는 관련된 스테이지 2 변환 테이블 엔트리가 PRO 퍼미션을 명시하는지의 여부를 체크하고, 만약 그렇지 않다면, 그러면 단계(1010)에서, 메모리 액세스는 거부되고 장애가 시그널링된다. 이것은, 공격자가 공격자에 의해 생성되는 새로운 L0 페이지 테이블을 가리키도록 스테이지 1 변환 테이블 기본 어드레스 레지스터(TTBR_EL1)를 수정하려고 시도하는 경우, 공격자가 메모리의 PRO 영역을 수정할 수 없었을 것이고 따라서 비 PRO 영역에 기록해야만 하였기 때문에, 수정된 기본 어드레스에 기초한 액세스는 실패할 것이고, 따라서 스테이지 1 최상위 레벨 테이블 PRO 체크가 실패할 것이라는 것을 의미한다.
- [0277] 스테이지 1 최상위 레벨 테이블 PRO 체크가 단계(1006)에서 디스에이블되거나, 또는 인에이블되지만 단계(1008)에서 통과하는 경우, 그러면 단계(1012)에서, MMU(28)는 스테이지 1 최상위 레벨 테이블 존재 체크가 인에이블되는지의 여부를 결정한다. 다시 말하지만, 이것은 제어 레지스터(예를 들면, 스테이지 2 변환 제어 레지스터(VTCR_EL2))에 저장되는 스테이지 1 최상위 레벨 테이블 존재 체크 인에이블 제어 값에 기초하여 결정될 수 있다. 단계(1012)에서 체크가 인에이블되는지의 여부를 결정하기 위해 사용되는 인에이블 제어 값은 단계(1006)에 대해 상기에서 언급되는 인에이블 제어 값과 동일할 수 있다(예를 들면, 몇몇 구현예에서, 단계(1008 및 1014)에서의 체크는 전체적으로 인에이블되는 또는 전체적으로 디스에이블되는 단일의 체크의 두 개의 단계로서 간주될 수 있다). 스테이지 1 최상위 레벨 테이블 존재 체크가 인에이블되는 경우, 그러면 단계(1014)에서, MMU(28)는 관련된 스테이지 2 변환 테이블 엔트리가 스테이지 1 최상위 레벨 엔트리 존재 퍼미션(80)을 명시하는지의 여부를 결정한다. 도 8과 관련하여 상기에서 언급되는 바와 같이, 스테이지 1 최상위 레벨 엔트리 존재 퍼미션은 PRO 퍼미션의 대안적인 변형으로서 인코딩될 수 있거나(그 결과, 스테이지 1 최상위 레벨 엔트리 존재 퍼미션을 갖는 메모리 영역도 또한 PRO 퍼미션을 갖는 것으로 가정됨), 또는 대안적으로 PRO 퍼미션이 제공되는지의 여부를 표시하는 별개의 독립적인 속성으로서 인코딩될 수 있다. 관련된 스테이지 2 변환 테이블 엔트리가 스테이지 1 최상위 레벨 엔트리 존재 퍼미션을 명시하지 않는 경우, 그러면 또 다시 단계(1010)에서,

메모리 액세스는 거부되고 장애가 시그널링된다. 이것은 L1, L2 또는 L3 테이블의 어드레스를 가리키도록 스테이지 1 최상위 레벨 기본 어드레스를 수정하는 것에 의해 유효하고 인증된 L1, L2 또는 L3 테이블이 L0 테이블 대신 대응되는 것을 방지하기 위해 사용할 수 있는데, L1, L2 또는 L3 테이블을 유지하기 위해 사용되는 메모리 영역에 대응하는 스테이지 2 엔트리가 스테이지 1 최상위 레벨 엔트리 존재 퍼미션을 갖지 않도록 정의될 수도 있기 때문이다. 임의의 유효하고 인증된 스테이지 1 L0 테이블을 유지하기 위해 사용되는 메모리 영역만이 대응하는 스테이지 2 엔트리에서 스테이지 1 최상위 레벨 엔트리 존재 퍼미션을 갖는 것으로 마킹되어, 유효한 메모리 액세스에 대한 스테이지 1 L0 기본 어드레스로서 어떤 어드레스가 사용될 수 있는지를 제한할 수 있다.

[0278] 스테이지 1 최상위 레벨 테이블 존재 체크가 단계(1012)에서 디스에이블되거나 또는 인에이블되지만 단계(1014)에서 통과하는 경우, 그러면 단계(1016)에서, MMU는 PA_base에 의해 식별되는 메모리 영역에 액세스하는 데 필요한 임의의 다른 체크를 실행하고, 이들이 실패하면 다시 단계(1010)에서 메모리 액세스를 거부한다. 임의의 다른 체크가 통과되는 경우, 그러면 단계(1018)에서, 스테이지 1 최상위 레벨 기본 어드레스(IPA_base)는 페이지 테이블 워크를 수행하는 데 유효하게 사용될 수 있다. 그러므로, 스테이지 1 최상위 레벨 변환 테이블 엔트리의 어드레스는 결정된 PA_base에 인덱스 오프셋을 적용하는 것에 의해 획득되는데, 인덱스 오프셋은 변환될 타겟 VA의 비트의 서브세트에 기초하여 결정된다. 그 다음, 페이지 테이블 워크는 앞서 논의된 바와 같이 계속된다.

[0279] 그러므로, 도 6에서 도시되는 바와 같이, 소프트웨어에 의해 정의되는 예상된 사용에서(따라서 하드웨어에 의한 요건으로서 강제되지 않음), 추가된 보호가 소망되는 메모리 영역에 대한 PA 매핑(64)을 획득하기 위해 페이지 테이블 워크 상에서 액세스되는 스테이지 1 엔트리 중에서도, 스테이지 1 L0 페이지 테이블의 보관 위치를 식별하는 PA 매핑(64)을 제공하는 스테이지 2 페이지/블록 디스크립터는 PRO 및 스테이지 1 최상위 레벨 존재 퍼미션 둘 모두를 식별하고, 스테이지 1 L1, L2 또는 L3 페이지 테이블의 보관 위치를 식별하는 PA 매핑(64)을 제공하는 스테이지 2 페이지/블록 디스크립터는 PRO 퍼미션을 식별하지만 그러나 스테이지 1 최상위 레벨 존재 퍼미션을 식별하지 않는다. 이것은 상기에서 설명되는 기본 어드레스 대체 공격을 방지한다. 기본 어드레스 체크를 구현할 필요성을 방지하는 대안에는 스테이지 1 기본 어드레스 레지스터(TTBR_EL1)에 대한 임의의 액세스를 EL2에서 실행되는 하이퍼바이저에 트랩하는 것일 수 있는데, 하이퍼바이저는 그 다음 EL1에서 실행되는 오퍼레이팅 시스템 코드에 의해 요청되는 업데이트가 유효한지의 여부를 검사할 수 있고 있고 필요하다면 업데이트를 거부할 수 있다. 그러나, 실제로, 오퍼레이팅 시스템은 컨텍스트 전환시 기본 어드레스 레지스터(TTBR_EL1)의 어드레스를 빈번하게 전환하는 것을 필요로 할 수도 있고, 따라서 모든 업데이트를 트랩하는 것은 성능의 관점에서 비용이 많이 들 수도 있다. 도 17에서 도시되는 스테이지 1 최상위 레벨 기본 어드레스 체크를 구현하는 것에 의해, 이것은, 성능 향상시키기 위해, TTBR_EL1에 대한 기본 어드레스 레지스터 업데이트에 대한 그러한 트랩에 대한 필요성을 방지할 수 있다.

[0280] 몇몇 예에서, 스테이지 1 최상위 레벨 존재 퍼미션의 단일의 변형이 지원될 수 있는데, 이 경우 도 17의 단계(1014)에서의 체크는, 단순히, 그 스테이지 1 최상위 레벨 존재 퍼미션이 명시되었는지의 여부의 체크일 수 있다.

[0281] 다른 예는, 특정한 스테이지 1 변환 테이블 기본 어드레스 레지스터가 대응하는 메모리 영역의 어드레스를 명시할 수 있는지에 대한 더 세밀한 제어를 허용하기 위해, 스테이지 1 최상위 레벨 존재 퍼미션의 하나보다 더 많은 변형을 지원할 수도 있다. 예를 들면, 몇몇 구현에는 두 개의 스테이지 1 기본 어드레스 레지스터를 지원할 수도 있다:

[0282] ● TTBR0_EL1, 변환될 가상 어드레스의 구성 가능한 수(N)의 최상위 비트가 모두 0으로 설정되는 경우 EL0 또는 EL1에서 스테이지 1 변환을 위해 선택됨(N은 제어 레지스터에서 명시되는 제어 값에서 정의될 수 있음); 및

[0283] ● TTBR1_EL1, 변환될 가상 어드레스의 N 개의 최상위 비트 중 임의의 것이 нену로인 경우 EL0 또는 EL1에서 스테이지 1 변환을 위해 선택됨.

[0284] 그 다음, 다음을 포함하는 스테이지 1 최상위 레벨 존재 퍼미션의 여러 가지 변형이 정의될 수 있다:

[0285] ● Toplevel0: TTBR0_EL1을 통해 액세스되는 최상위 레벨 변환 테이블에 대해 사용되는 것이 허용되지만 그러나 TTBR1_EL1을 통해 액세스되는 최상위 레벨 변환 테이블에 대해서는 허용되지 않는 메모리 영역을 나타냄;

[0286] ● Toplevel1: TTBR1_EL1을 통해 액세스되는 최상위 레벨 변환 테이블에 대해 사용되는 것이 허용되지만 그러나

나 TTBR0_EL1을 통해 액세스되는 최상위 레벨 변환 테이블에 대해서는 허용되지 않는 메모리 영역을 나타냄;

- [0287]  Toplevel01: TTBR0_EL1 또는 TTBR1_EL1 중 어느 하나를 통해 액세스되는 최상위 레벨 변환 테이블에 대해 사용되는 것이 허용되는 메모리 영역을 나타냄.
- [0288] 이 경우, 단계(1014)에서의 체크의 하나의 구현에는, 다음의 조건 중 어느 하나가 충족되는 경우, 장애로 하여금 도 17의 단계(1010)에서 시그널링되게 할 수도 있다:
- [0289]  대응하는 메모리 영역이 Toplevel0 또는 Toplevel01 퍼미션 중 어느 하나를 갖지 않는 기본 어드레스를 제공하기 위해 TTBR0_EL1이 사용됨; 또는
- [0290]  대응하는 메모리 영역이 Toplevel1 또는 Toplevel01 퍼미션 중 어느 하나를 갖지 않는 기본 어드레스를 제공하기 위해 TTBR1_EL1이 사용됨.
- [0291] 몇몇 구현예에서, EL2에서 소프트웨어에 의해 제어되는 제어 레지스터에서 설정되는 추가적인 제어는, TTBR0_EL1이 Toplevel1 페이지에 대한 액세스를 제공하는 것이 금지되는지의 여부, 및 TTBR1_EL1이 Toplevel0 페이지에 대한 액세스를 제공하는 것이 금지되는지의 여부를 제어할 수 있다. 이 경우, 단계(1014)에서의 체크는, 다음의 조건 중 어느 하나가 충족되는 경우, 장애로 하여금 단계(1010)에서 시그널링되게 할 수도 있다:
- [0292]  Toplevel1 페이지에 액세스하기 위해 TTBR0_EL1이 사용될 때 장애가 트리거되는 것을 가능하게 하도록 TTBR0/Toplevel1 장애 인에이블 제어가 설정되는 경우, 대응하는 메모리 영역이 Toplevel01 퍼미션을 갖지 않는 또는 Toplevel1 퍼미션을 갖는 기본 어드레스를 제공하기 위해 TTBR0_EL1이 사용됨; 또는
- [0293]  Toplevel0 페이지에 액세스하기 위해 TTBR1_EL1이 사용될 때 장애가 트리거되는 것을 가능하게 하도록 TTBR1/Toplevel0 장애 인에이블 제어가 설정되는 경우, 대응하는 메모리 영역이 Toplevel01 퍼미션을 갖지 않는 또는 Toplevel0 퍼미션을 갖는 기본 어드레스를 제공하기 위해 TTBR1_EL1이 사용됨.
- [0294] 이들 피쳐는, 최상위 레벨 스테이지 1 변환 테이블을 제공하기 위해 어떤 메모리 어드레스 영역이 사용될 수 있는지에 대한 더욱 미세하게 세분화된 제어를 허용하지만, 그러나 인에이블 제어는, TTBR0 및 TTBR1 둘 모두에 대한 기본 어드레스를 제공하기 위해 임의의 Toplevel 페이지가 사용되는 것이 허용 가능한 것으로 간주되는 경우, 이들 더욱 미세하게 세분화된 체크가 디스에이블되는 것을 허용한다.
- [0295] 몇몇 구현예에서, 다양한 최상위 레벨 존재 퍼미션(Toplevel0, Toplevel1, Toplevel01)의 인코딩은 PRO 퍼미션의 인코딩과 결합될 수 있고, 그 결과, 이들은 Toplevel0 PRO 퍼미션, Toplevel1 PRO 퍼미션 및 Toplevel01 PRO 퍼미션으로서 처리된다. 그러한 구현예에서, Toplevel0, Toplevel1, 또는 Toplevel01 메모리 영역을 PRO 퍼미션을 갖지 않는 것으로 정의하는 것이 가능하지 않을 것이다. 그럼에도 불구하고, 스테이지 1 최상위 레벨 테이블 존재 퍼미션 중 임의의 것을 갖는 것으로 처리되지 않는 별개의 PRO 퍼미션 인코딩이 있기 때문에, PRO 영역을 Toplevel0, Toplevel1, 또는 Toplevel01 퍼미션 중 임의의 것을 갖지 않는 것으로 정의하는 것이 가능하다.
- [0296] 시뮬레이터 예
- [0297] 도 18은 사용될 수도 있는 시뮬레이터 구현예를 예시한다. 앞서 설명된 실시형태가 관련되는 기술을 지원하는 특정한 프로세싱 하드웨어를 동작시키기 위한 장치 및 방법의 관점에서 본 발명을 구현하지만, 컴퓨터 프로그램의 사용을 통해 구현되는 본원에서 설명되는 실시형태에 따른 명령어 실행 환경을 제공하는 것이 또한 가능하다. 그러한 컴퓨터 프로그램은, 하드웨어 아키텍처의 소프트웨어 기반 구현예를 그들이 제공하는 한, 시뮬레이터로서 종종 지칭된다. 다양한 시뮬레이터 컴퓨터 프로그램은 에뮬레이터, 가상 머신, 모델, 및 동적 바이너리 변환기를 비롯한, 바이너리 변환기를 포함한다. 통상적으로, 시뮬레이터 구현예는 호스트 프로세서(1230) 상에서 실행될 수도 있으며, 옵션 사항으로 호스트 오퍼레이팅 시스템(1220)을 실행하여, 시뮬레이터 프로그램(1210)을 지원할 수도 있다. 일부 배열에서는, 하드웨어와 제공된 명령어 실행 환경, 및/또는 동일한 호스트 프로세서 상에서 제공되는 다수의 별개의 명령어 실행 환경 사이에서 다수의 층의 시뮬레이션이 있을 수도 있다. 역사적으로, 합리적인 속도로 실행되는 시뮬레이터 구현예를 제공하기 위해 강력한 프로세서가 필요로 되었지만, 그러나 그러한 접근법은 소정의 상황에서, 예컨대 호환성 또는 재사용 이유 때문에 다른 프로세서에 기본적인(native) 코드를 실행하려는 소망이 있는 경우, 정당화될 수도 있다. 예를 들면, 시뮬레이터 구현예는 호스트 프로세서 하드웨어에 의해 지원되지 않는 추가적인 기능성을 갖춘 명령어 실행 환경을 제공할 수도 있거

나, 또는 상이한 하드웨어 아키텍처와 통상적으로 관련되는 명령어 실행 환경을 제공할 수도 있다. 시뮬레이션의 개요는 ["Some Efficient Architecture Simulation Techniques", Robert Bedichek, Winter 1990 USENIX Conference, Pages 53 - 63]에서 주어진다.

[0298] 실시형태가 특정한 하드웨어 구성 또는 피처를 참조하여 앞서 설명된 정도까지, 시뮬레이팅된 실시형태에서, 동등한 기능성이 적절한 소프트웨어 구성 또는 피처에 의해 제공될 수도 있다. 예를 들면, 특정한 회로부는 시뮬레이팅된 실시형태에서 컴퓨터 프로그램 로직으로서 구현될 수도 있다. 마찬가지로, 메모리 하드웨어, 예컨대 레지스터 또는 캐시는 시뮬레이팅된 실시형태에서 소프트웨어 데이터 구조로서 구현될 수도 있다. 앞서 설명된 실시형태에서 참조되는 하드웨어 엘리먼트 중 하나 이상이 호스트 하드웨어(예를 들면, 호스트 프로세서(1230))에서 존재하는 배열에서, 일부 시뮬레이팅된 실시형태는, 적절한 경우, 호스트 하드웨어를 사용할 수도 있다.

[0299] 시뮬레이터 프로그램(1210)은 컴퓨터 관독 가능 저장 매체(이것은 비일시적 매체일 수도 있음) 상에 저장될 수도 있고, 타겟 코드(1200)(이것은 애플리케이션, 오퍼레이팅 시스템 및 하이퍼바이저를 포함할 수도 있음)에 프로그램 인터페이스(명령어 실행 환경)를 제공할 수 있는데, 이것은 시뮬레이터 프로그램(1210)에 의해 모델링되는 하드웨어 아키텍처의 인터페이스와 동일하다. 따라서, 상기에서 설명되는 보호 엔트리 체크 RCW 명령어를 비롯하여, 타겟 코드(1200)의 프로그램 명령어는 시뮬레이터 프로그램(1210)을 사용하여 명령어 실행 환경 내에서 실행될 수도 있고, 그 결과, 상기에서 논의되는 장치(2)의 하드웨어 피처를 실제로 갖지 않는 호스트 컴퓨터(1230)가 이들 피처를 에뮬레이팅할 수 있다. 유사하게, PRO 영역 타입에 대한 지원을 비롯하여, 상기에서 논의되는 바와 같은 다양한 메모리 관리 체크 기능은 시뮬레이터 프로그램(1210)의 메모리 관리 프로그램 로직(1218)을 사용하여 에뮬레이팅될 수도 있다.

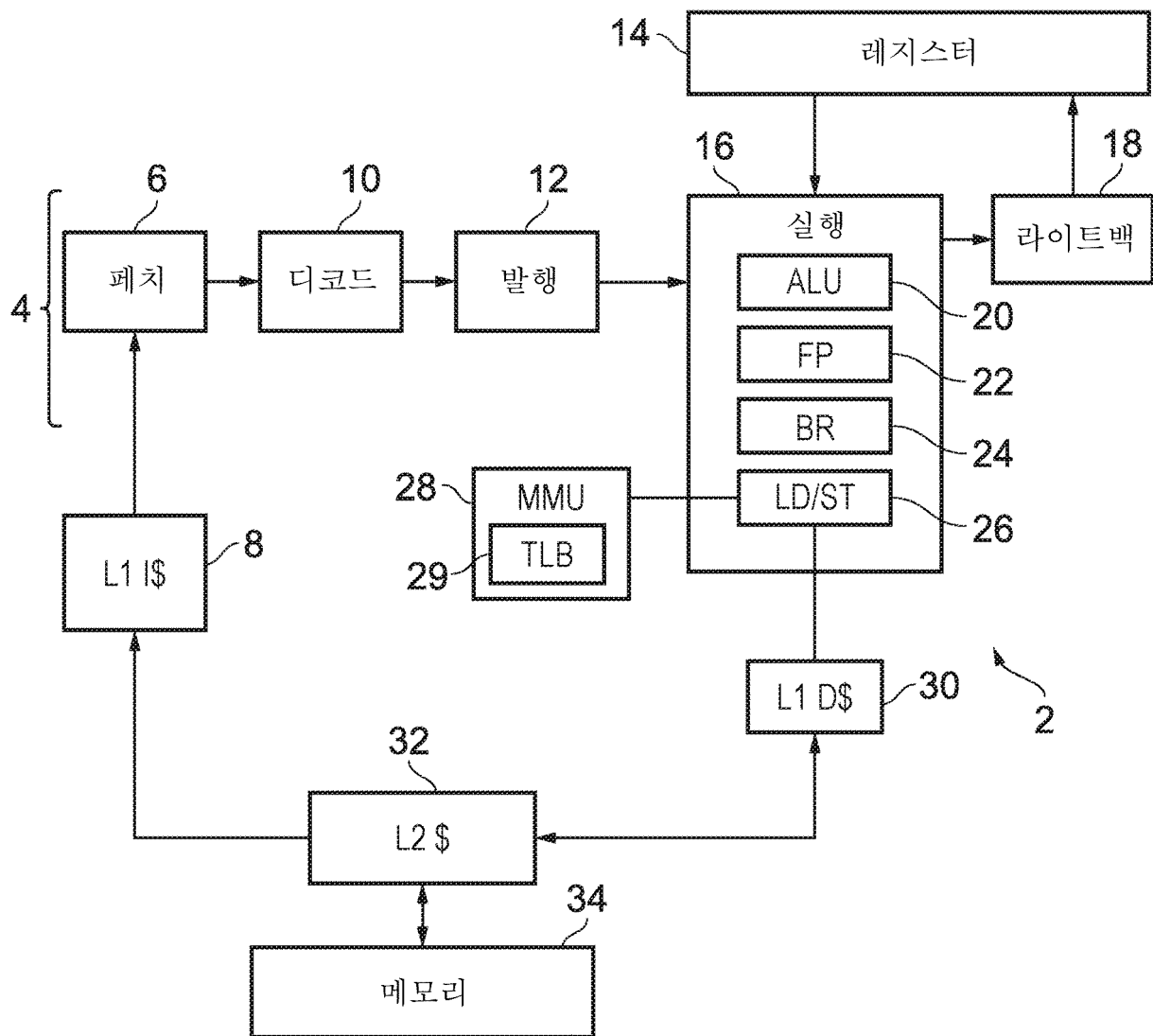
[0300] 그러므로, 시뮬레이터 프로그램(1210)은 상기에서 설명되는 프로세싱 회로부(4)의 상태를 시뮬레이팅하는 프로세싱 프로그램 로직(1212)을 구비할 수도 있다. 예를 들면, 프로세싱 프로그램 로직(1212)은 타겟 코드(1200)의 시뮬레이팅된 실행 동안 발생하는 이벤트에 응답하여 실행 상태(EL0-EL3)의 전이를 제어할 수도 있다. 명령어 디코딩 프로그램 로직(1214)은 타겟 코드(1200)의 명령어를 디코딩하고 이들을 호스트 장치(1230)의 네이티브 명령어 세트의 명령어의 대응하는 세트에 매핑한다. 레지스터 에뮬레이팅 프로그램 로직(1216)은, 예컨대 호스트 장치(1230)의 레지스터 또는 메모리(1232)의 데이터에 액세스하는 것에 의해, 타겟 코드에 의해 요청되는 레지스터 액세스를 호스트 장치(1230)의 호스트 하드웨어 상에서 유지되는 대응하는 데이터 구조에 매핑한다. 메모리 관리 프로그램 로직(1218)은 상기의 하드웨어 구현 실시형태에서 설명되는 MMU(28)에 대응하는 방식으로 어드레스 변환, 페이지 테이블 워크 및 액세스 제어 체크를 구현하지만, 그러나, 타겟 코드(1200)에 대해 정의되는 페이지 테이블에 기초하여 어드레스 변환에서 스테이지 2 매핑에 의해 획득되는 시뮬레이팅된 물리적 어드레스를, 호스트 메모리(1232)에 액세스하기 위해 사용되는 호스트 가상 어드레스로 매핑하는 추가적인 기능을 또한 갖는다. 이들 호스트 가상 어드레스는 그들 자체가 호스트에 의해 지원되는 표준 어드레스 변환 메커니즘을 사용하여 호스트 물리적 어드레스로 변환될 수도 있다(호스트 물리적 어드레스로의 호스트 가상 어드레스의 변환은 시뮬레이터 프로그램(1210)에 의해 제어되는 것의 범위 밖에 있음).

[0301] 본 출원에서, 단어 "...로 구성되는"은 장치의 엘리먼트가 정의된 동작을 실행할 수 있는 구성을 갖는다는 것을 의미하기 위해 사용된다. 이러한 맥락에서, "구성"은 하드웨어 또는 소프트웨어의 상호 접속의 배열 또는 방식을 의미한다. 예를 들면, 장치는 정의된 동작을 제공하는 전용 하드웨어를 가질 수도 있거나, 또는 프로세서 또는 다른 프로세싱 디바이스가 기능을 수행하도록 프로그래밍될 수도 있다. "~하도록 구성되는"은 정의된 동작을 제공하기 위해 장치 엘리먼트가 어떤 방식으로든 변경될 필요가 있다는 것을 암시하는 것은 아니다.

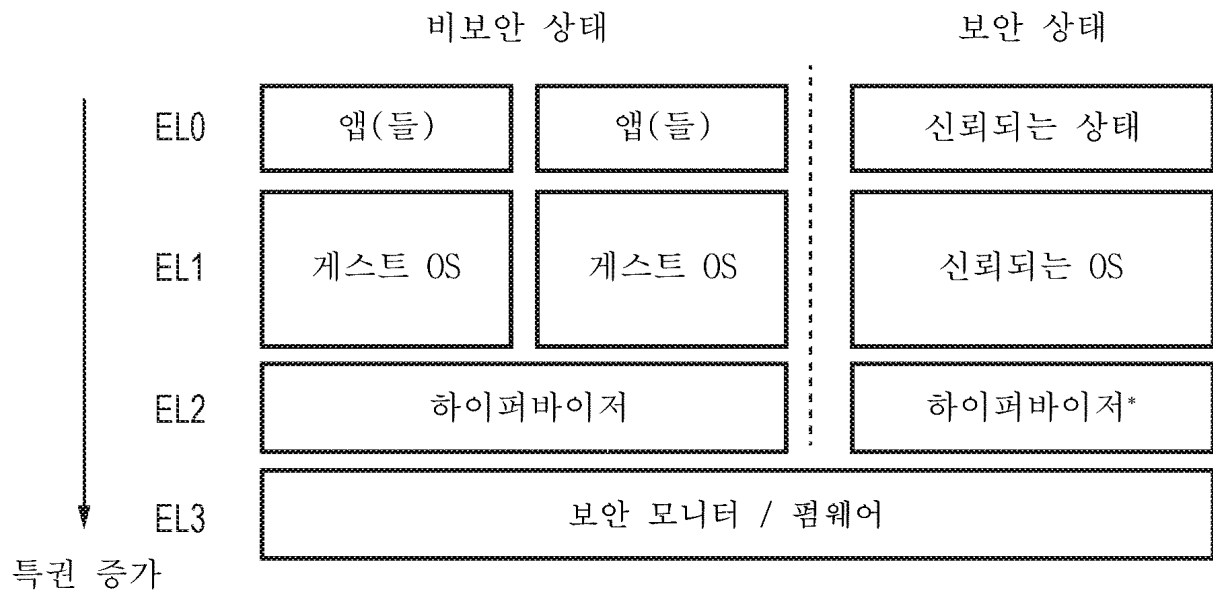
[0302] 본 발명의 예시적인 실시형태가 첨부 도면을 참조하여 본원에서 상세하게 설명되었지만, 본 발명은 그들 정확한 실시형태로 제한되지 않는다는 것, 및 첨부된 청구범위에 의해 정의되는 바와 같은 본 발명의 범위로부터 벗어나지 않으면서 기술 분야에서 숙련된 자에 의해 다양한 변경예 및 수정예가 그 안에서 실행될 수 있다는 것이 이해되어야 한다.

도면

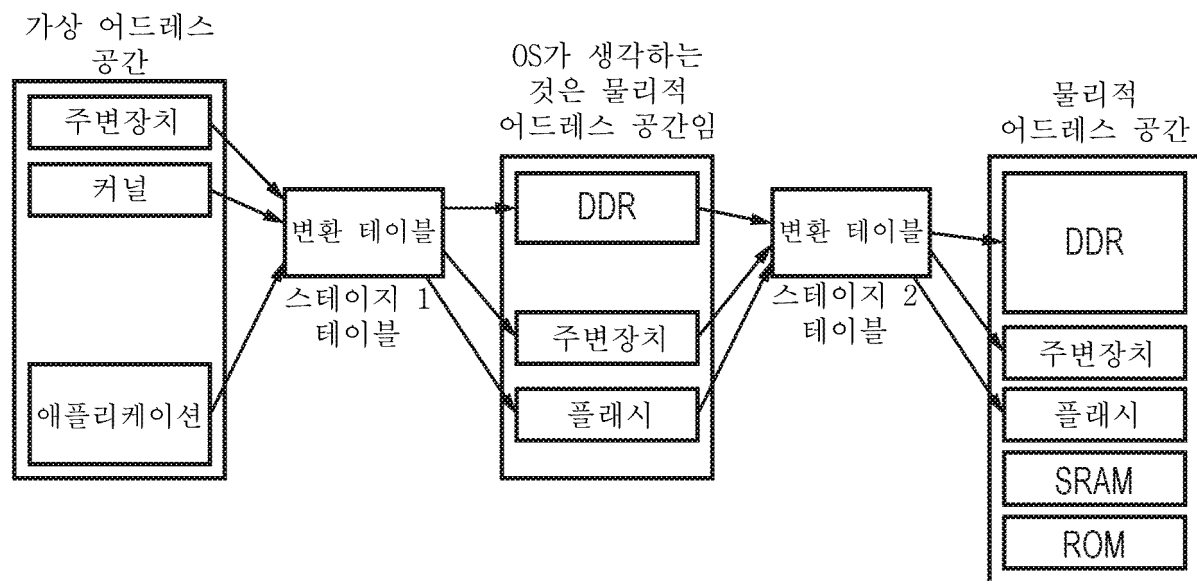
도면1



도면2

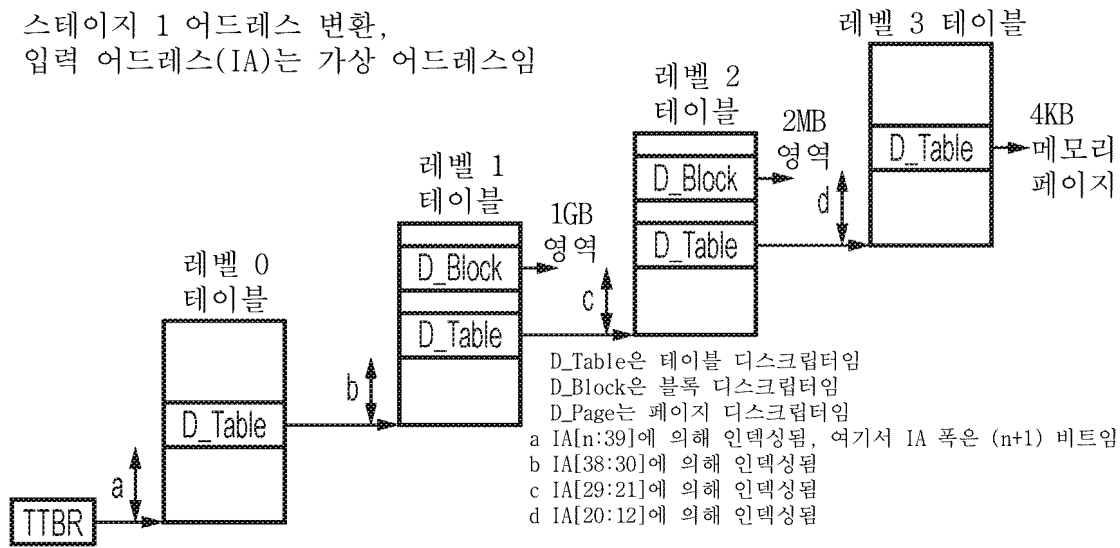


도면3



도면4

스테이지 1 어드레스 변환,
입력 어드레스(IA)는 가상 어드레스임

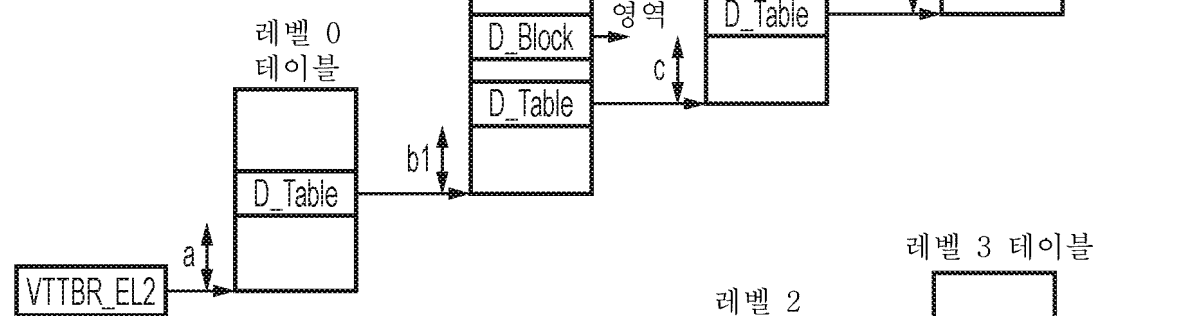


도면5

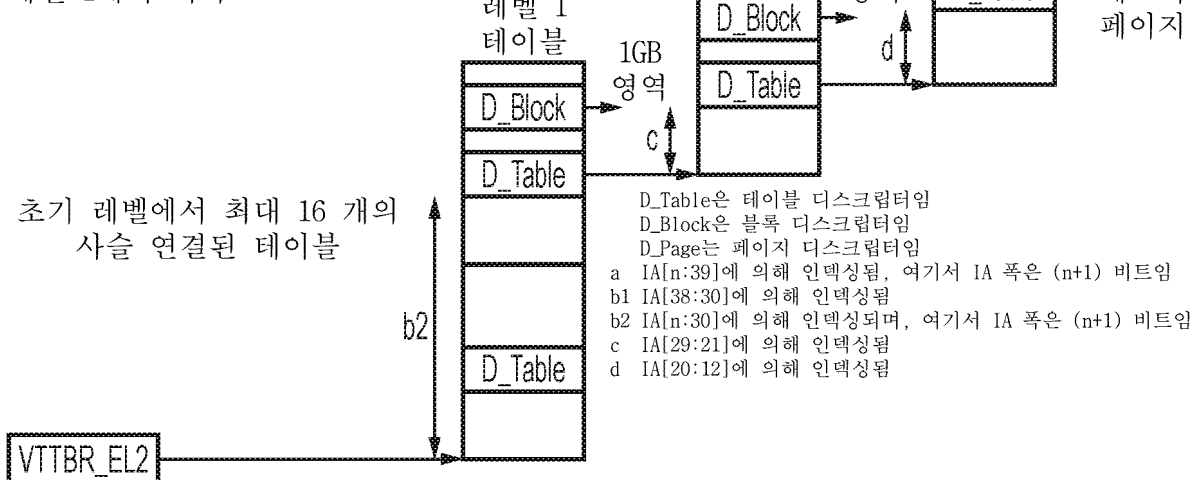
스테이지 2 어드레스 변환,
입력 어드레스(IA)는 중간 어드레스임

VTCR_EL2.SL0은 시작 레벨을 정의함

레벨 0에서 시작

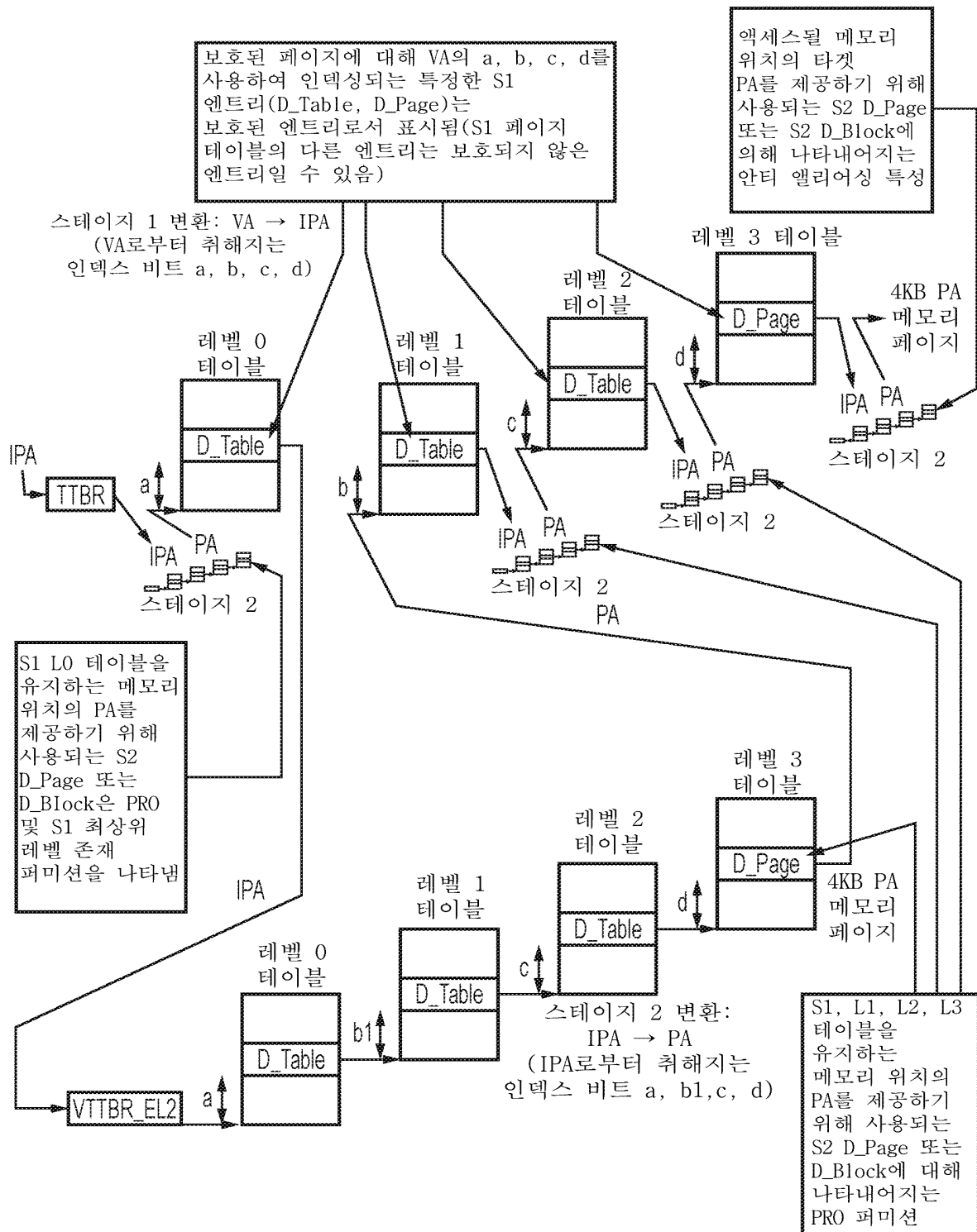


레벨 1에서 시작

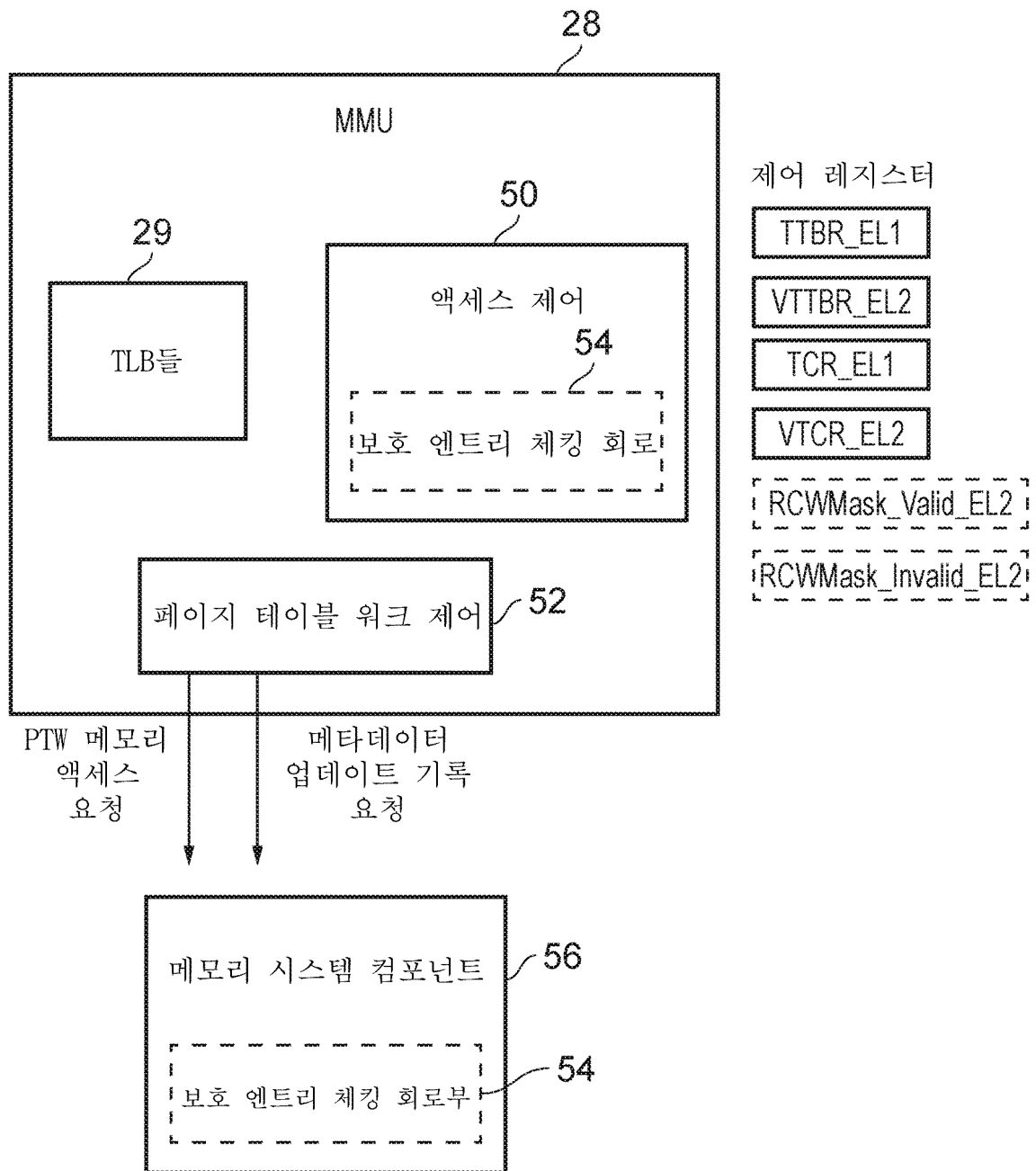


초기 레벨에서 최대 16 개의
사슬 연결된 테이블

도면6

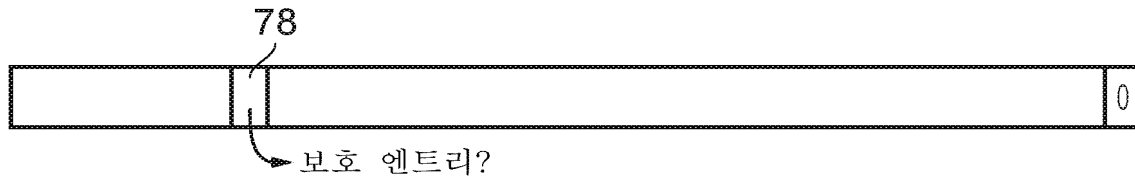


도면7

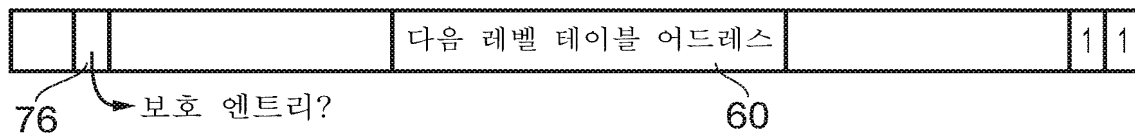


도면8

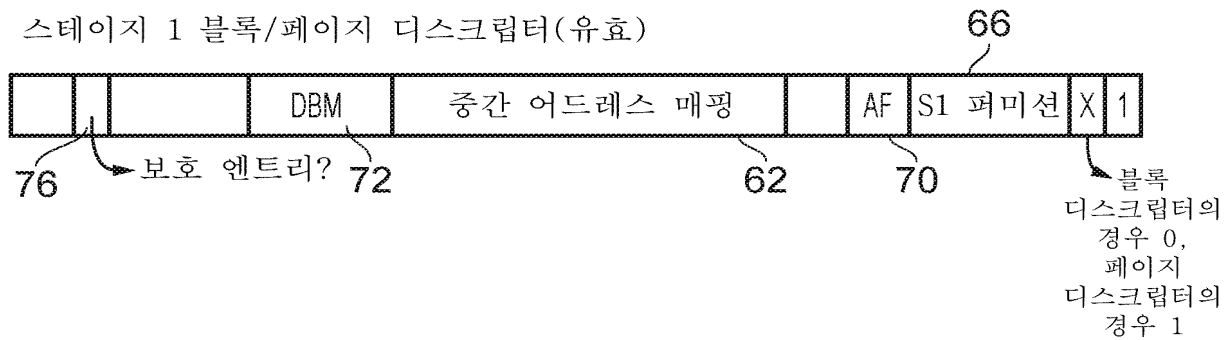
스테이지 1 디스크립터(무효)



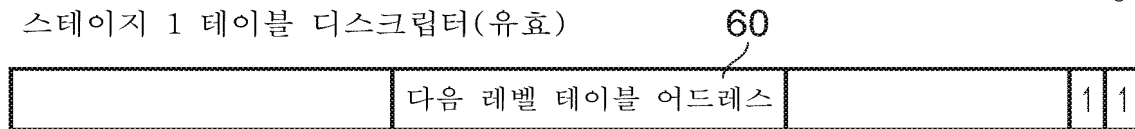
스테이지 1 테이블 디스크립터(유효) L0, L1 또는 L2



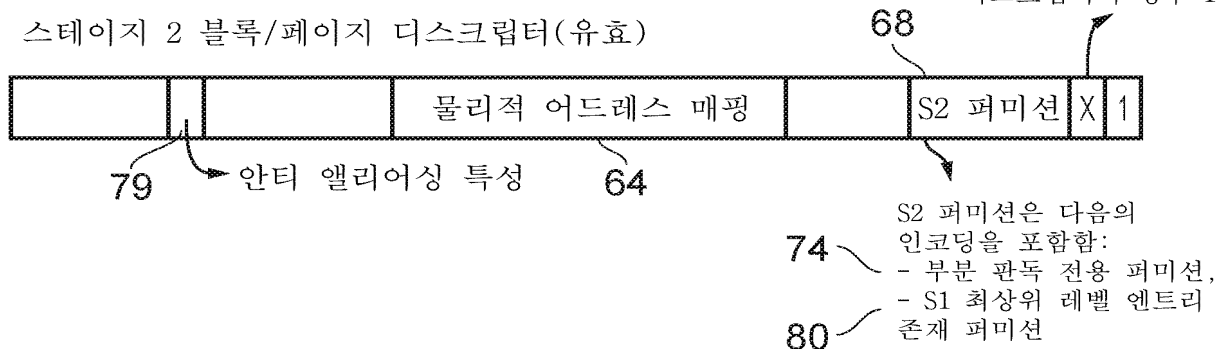
스테이지 1 블록/페이지 디스크립터(유효)



스테이지 1 테이블 디스크립터(유효)



스테이지 2 블록/페이지 디스크립터(유효)



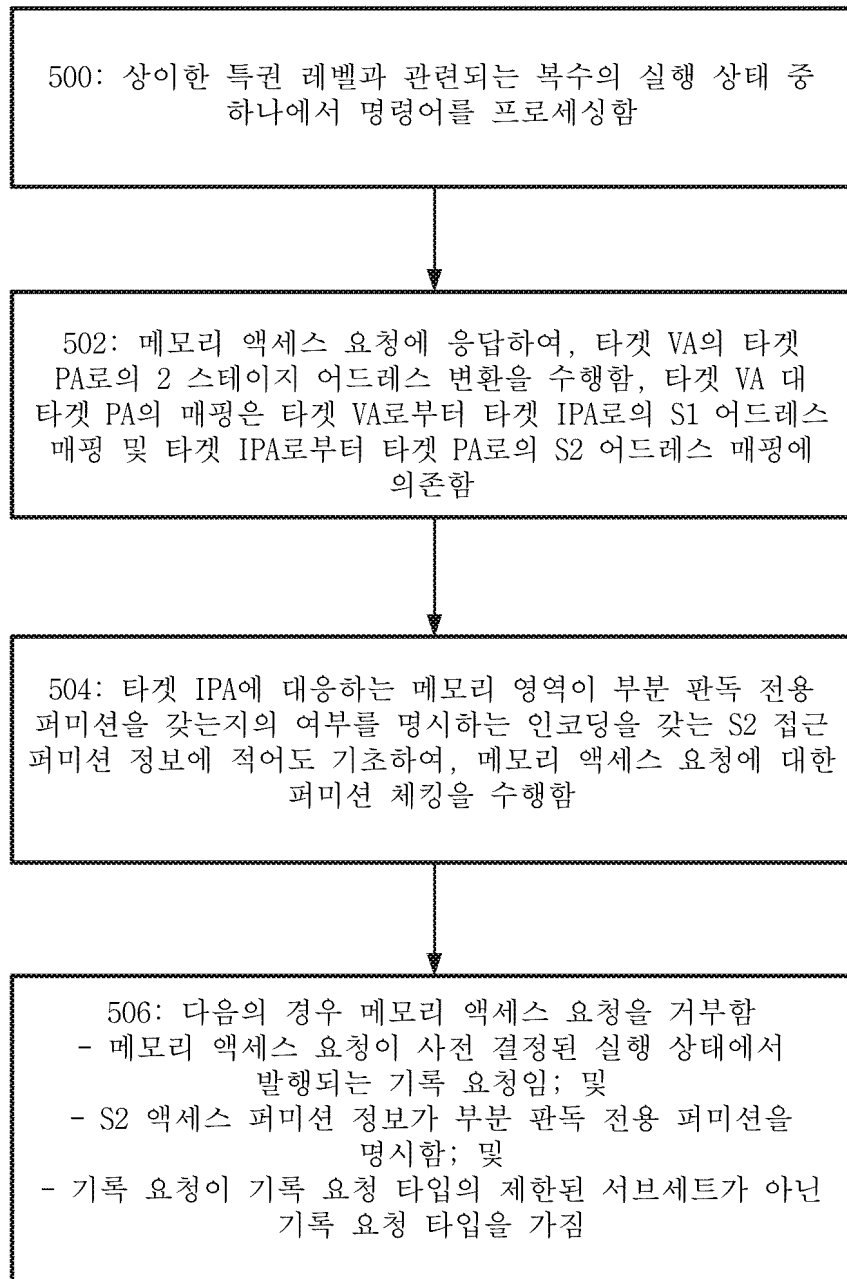
도면9

상이한 기록 요청 타입에 응답하여, S2 퍼미션에 의해 부분 판독 전용으로서 마킹되는 메모리 영역에 대한 기록 액세스 제어:

EL1에서 기록이
발행되는 경우
부분 판독
전용 영역에
기록하도록
허락되는 기록
요청 타입의
제한된 서브세트

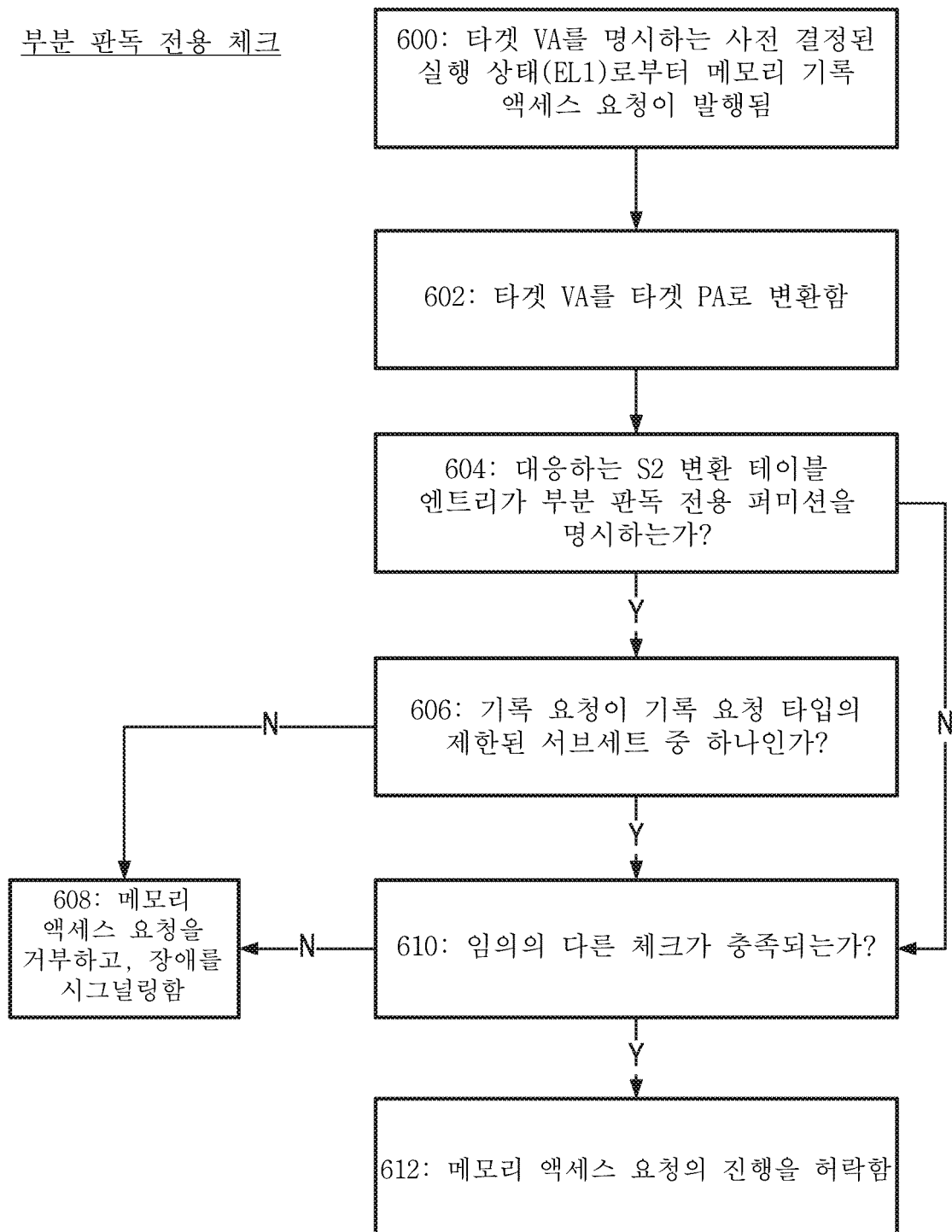
부분 판독 전용 영역에 기록하도록 허락되는 기록 요청 타입	부분 판독 전용 영역에 기록하는 것이 금지되는 기록 요청 타입
(옵션 사항) EL2 또는 EL3에서 실행되는 저장 명령어에 응답하여 발행되는 기록 요청(EL0 또는 EL1에서 실행되는 경우 부분 판독 전용 영역에 기록하는 것이 금지되는 적어도 하나의 타입의 저장을 포함함)	EL0 또는 EL1에서 실행되는 적어도 하나의 타입의 저장 명령어에 응답하여 발행되는 기록 요청(예를 들면, ISA의 저장 명령어의 대부분)
예를 들면 다음의 것인, EL1에서 실행되는 사전 결정된 타입의 변환 테이블 엔트리 업데이트 명령어: • 보호 엔트리 체크가 통과되는 경우, EL1에서 실행되는 보호 엔트리 체크 타입의 RCW 명령어.	
변환 테이블 엔트리의 액세스 추적 메타데이터를 업데이트하기 위해 MMU에 의해 생성되는 메타데이터 업데이트 기록 요청(예를 들면, AF, DBM가 설정되는 경우의 기록 퍼미션)	

도면10

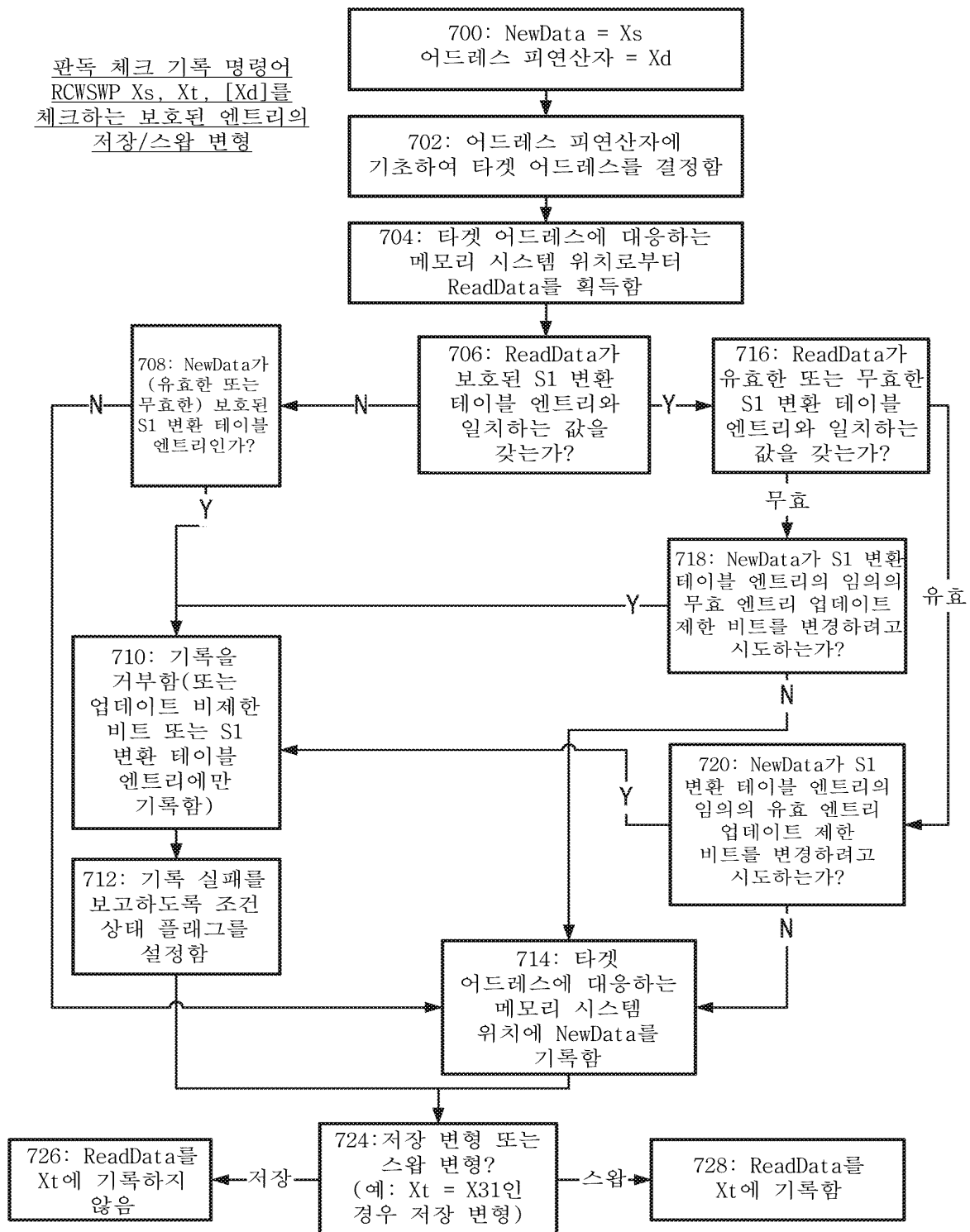


도면11

부분 판독 전용 체크

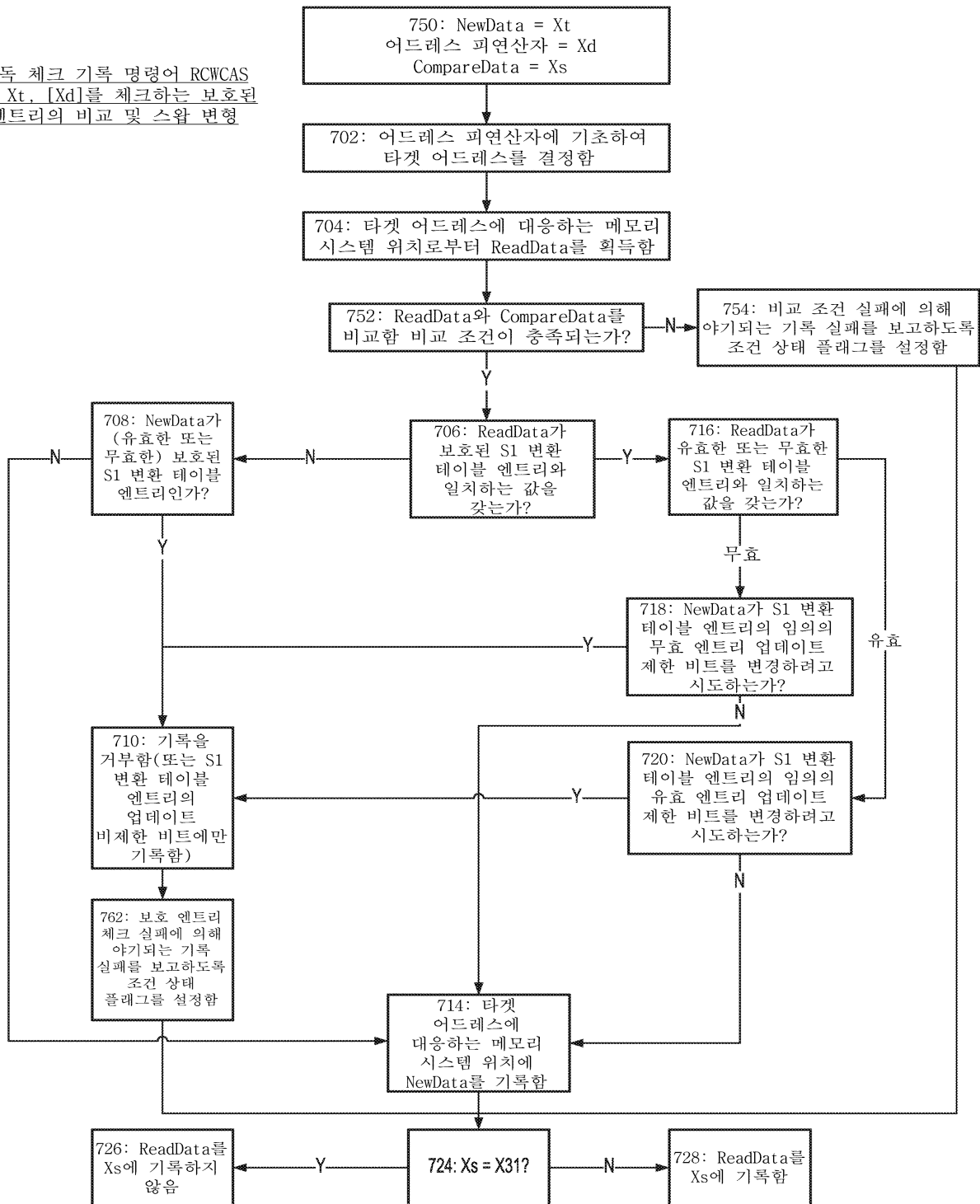


도면12



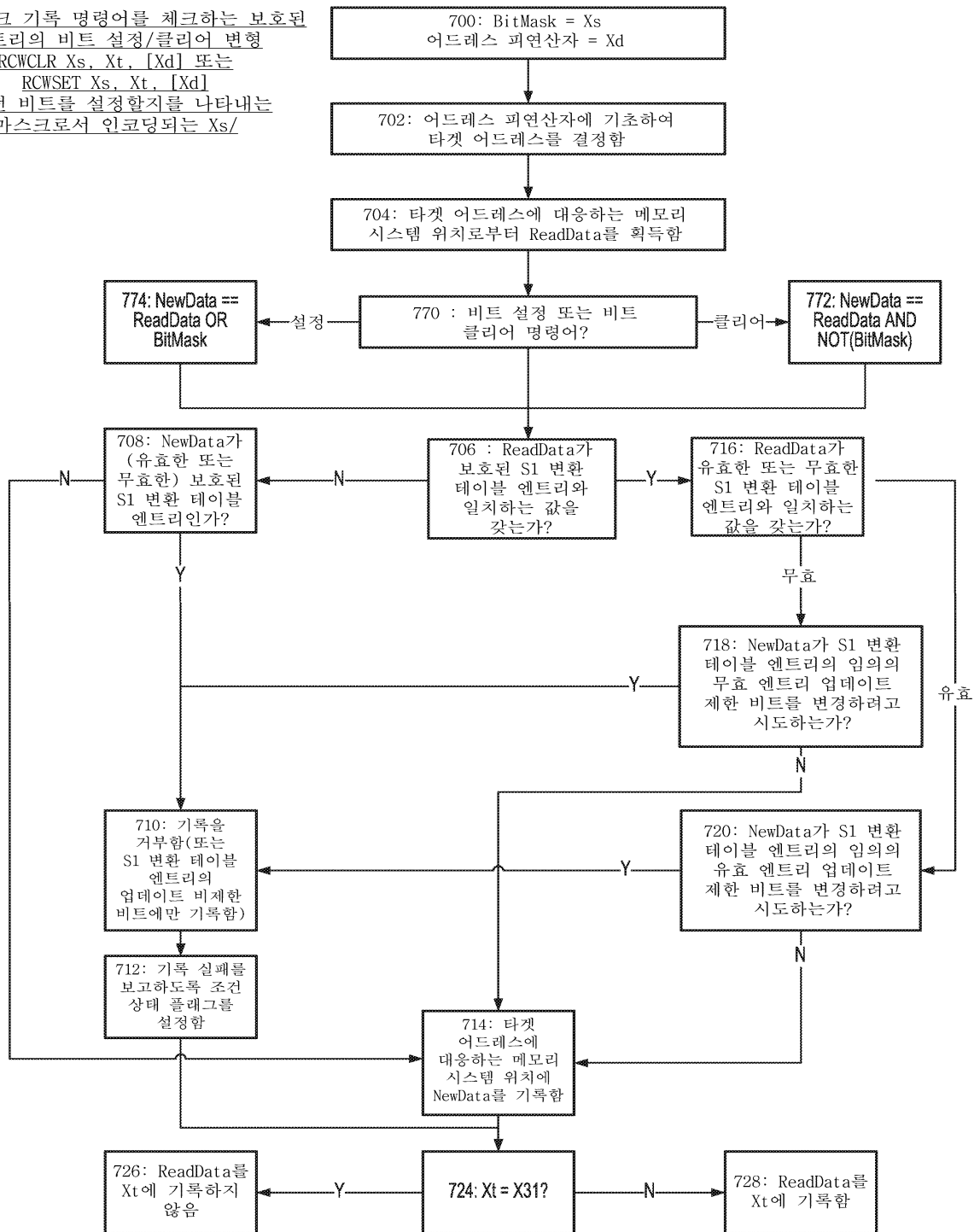
도면13

관독 체크 기록 명령어 RCWCAS
Xs, Xt, [Xd]를 체크하는 보호된
엔트리의 비교 및 스왑 변형



도면14

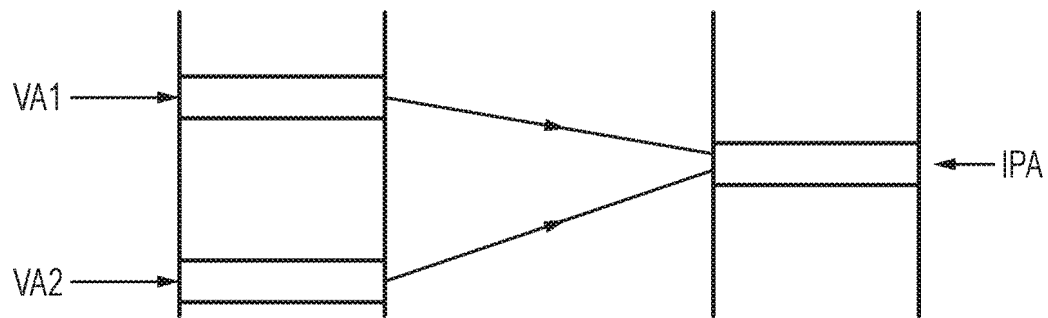
판독 체크 기록 명령어를 체크하는 보호된
엔트리의 비트 설정/클리어 변형
RCWCLR Xs, Xt, [Xd] 또는
RCWSET Xs, Xt, [Xd]
(어떤 비트를 설정할지를 나타내는
마스크로서 인코딩되는 Xs/



도면15

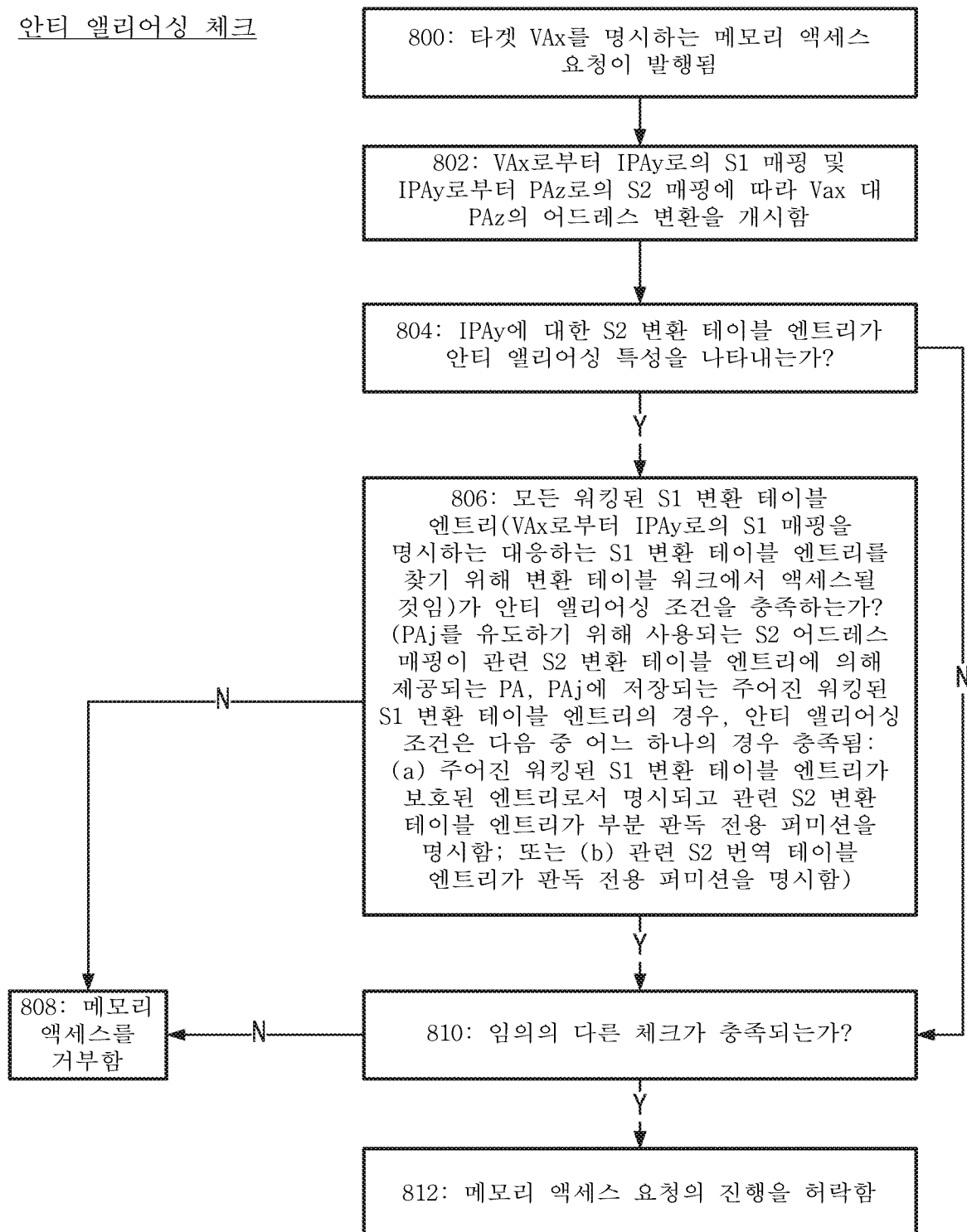
앨리어싱 가상 어드레스 공간

중간 어드레스 공간



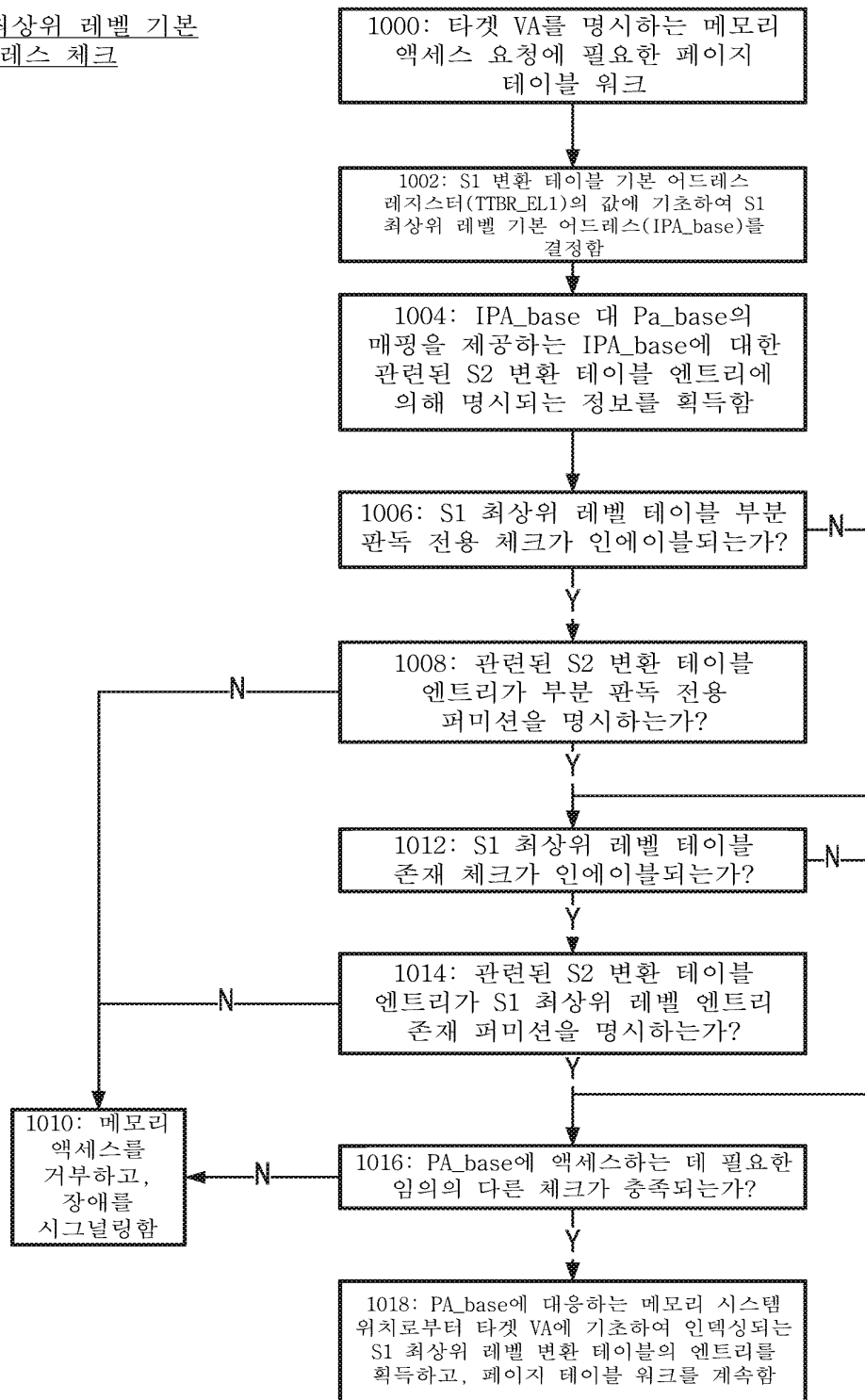
도면16

안티 앨리어싱 체크



도면17

S1 최상위 레벨 기본
어드레스 체크



도면18

