



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2017년07월03일
(11) 등록번호 10-1750669
(24) 등록일자 2017년06월19일

(51) 국제특허분류(Int. Cl.)
G06F 12/10 (2016.01)
(52) CPC특허분류
G06F 12/1027 (2013.01)
G06F 12/1009 (2013.01)
(21) 출원번호 10-2016-7009817
(22) 출원일자(국제) 2014년09월19일
심사청구일자 2016년09월23일
(85) 번역문제출일자 2016년04월14일
(65) 공개번호 10-2016-0065873
(43) 공개일자 2016년06월09일
(86) 국제출원번호 PCT/US2014/056664
(87) 국제공개번호 WO 2015/050726
국제공개일자 2015년04월09일
(30) 우선권주장
14/046,341 2013년10월04일 미국(US)
(56) 선행기술조사문헌
US20100228944 A1
US20120137079 A1

(73) 특허권자
켈컴 인코퍼레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(72) 발명자
선 지안
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
추아-에오안 류 고
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
(74) 대리인
특허법인코리아나

전체 청구항 수 : 총 28 항

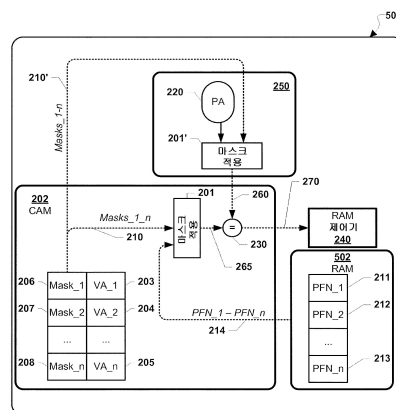
심사관 : 임정복

(54) 발명의 명칭 멀티-코어 이중 시스템 변환 색인 버퍼 일관성

(57) 요약

물리적 주소 입력을 이용하여 역 변환 색인 버퍼 (TLB) 록-업을 수행하는 방법들, 디바이스들, 명령들은 물리적 주소 입력을 제 1 프로세서로 획득하고 (여기서 물리적 주소 입력은 공유 메모리에 대응하는 물리적 주소를 나타낸다), 제 1 프로세서와 연관된 TLB 내의 제 1 TLB 엔트리로부터 제 1 가상 주소와 연관된 제 1 마스크를 획득하고 (여기서 획득된 제 1 마스크는 비트 패턴이다), 공유 메모리와 연관된 제 1 페이지 프레임 번호를 제 1 TLB 엔트리로부터 획득하고, 획득된 제 1 마스크를 획득된 제 1 페이지 프레임 번호에 적용하여 제 1 값을 생성하고, 획득된 제 1 마스크를 획득된 물리적 주소 입력에 적용하여 제 2 값을 생성하고, 제 1 값과 제 2 값을 비교하여 제 1 값과 제 2 값이 매칭하는지 여부를 결정하는 것을 포함한다.

대표도



(52) CPC특허분류

G06F 12/1036 (2013.01)

G06F 2212/656 (2013.01)

명세서

청구범위

청구항 1

멀티-코어 이중 컴퓨팅 디바이스에 의해 수행되는, 물리적 주소 입력을 이용하여 역 변환 색인 버퍼 (translation lookaside buffer; TLB) 룩-업을 수행하는 방법으로서,

상기 물리적 주소 입력을 제 1 프로세서로 획득하는 단계로서, 상기 물리적 주소 입력은 공유 메모리에 대응하는 물리적 주소를 나타내는, 상기 물리적 주소 입력을 제 1 프로세서로 획득하는 단계;

상기 제 1 프로세서와 연관된 TLB 내의 제 1 TLB 엔트리로부터 제 1 가상 주소와 연관된 제 1 마스크를 획득하는 단계로서, 획득된 상기 제 1 마스크는 비트 패턴인, 상기 제 1 마스크를 획득하는 단계;

상기 공유 메모리와 연관된 제 1 페이지 프레임 번호를 상기 제 1 TLB 엔트리로부터 획득하는 단계;

상기 획득된 제 1 마스크를 획득된 상기 제 1 페이지 프레임 번호에 적용하여 제 1 값을 생성하는 단계;

상기 획득된 제 1 마스크를 획득된 상기 물리적 주소 입력에 적용하여 제 2 값을 생성하는 단계; 및

상기 제 1 값과 상기 제 2 값을 비교하여 상기 제 1 값과 상기 제 2 값이 매칭하는지 여부를 결정하는 단계를 포함하는, 역 변환 색인 버퍼 (TLB) 룩-업을 수행하는 방법.

청구항 2

제 1 항에 있어서,

상기 제 1 값과 상기 제 2 값이 매칭하지 않는다고 결정하는 것에 응답하여 상기 TLB 내의 다른 TLB 엔트리들로부터 제 2 마스크 및 제 2 페이지 프레임 번호를 반복적으로 획득하는 단계;

획득된 상기 제 2 마스크를 획득된 상기 제 2 페이지 프레임 번호에 반복적으로 적용하여 상기 제 1 값을 생성하는 단계; 및

상기 획득된 제 2 마스크를 상기 획득된 물리적 주소 입력에 반복적으로 적용하여 상기 제 2 값을 생성하는 단계

를 더 포함하고,

상기 제 1 값과 상기 제 2 값을 비교하여 상기 제 1 값과 상기 제 2 값이 매칭하는지 여부를 결정하는 단계는 상기 제 1 값과 상기 제 2 값을 반복적으로 비교하여 상기 제 1 값과 상기 제 2 값이 매칭하는지 여부를 결정하는 단계를 포함하는, 역 변환 색인 버퍼 (TLB) 룩-업을 수행하는 방법.

청구항 3

제 2 항에 있어서,

상기 획득된 제 2 마스크를 상기 획득된 제 2 페이지 프레임 번호에 반복적으로 적용하여 상기 제 1 값을 생성하는 단계는 복사 없이 시스템 페이지 테이블에 있는 엔트리들로부터의 페이지 프레임 번호들을 재이용하는 단계를 포함하는, 역 변환 색인 버퍼 (TLB) 룩-업을 수행하는 방법.

청구항 4

제 1 항에 있어서,

상기 제 1 값과 상기 제 2 값이 매칭한다고 결정하는 것에 응답하여 상기 제 1 TLB 엔트리를 플러시하는 단계를 더 포함하는, 역 변환 색인 버퍼 (TLB) 룩-업을 수행하는 방법.

청구항 5

제 1 항에 있어서,

TLB 액션들 및 상기 물리적 주소를 나타내는 메시지를 제 2 프로세서로 전송하는 단계를 더 포함하고, 상기 TLB 액션들은 TLB 플러시 동작을 포함하는, 역 변환 색인 버퍼 (TLB) 룩-업을 수행하는 방법.

청구항 6

제 1 항에 있어서,

상기 제 1 값과 상기 제 2 값이 매칭한다고 결정하는 것에 응답하여 상기 제 1 TLB 엔트리의 인덱스를 반환하는 단계를 더 포함하는, 역 변환 색인 버퍼 (TLB) 룩-업을 수행하는 방법.

청구항 7

제 6 항에 있어서,

상기 물리적 주소 입력을 제 1 프로세서로 획득하는 단계는 TLB 액션들 및 상기 물리적 주소를 포함하는 메시지를 제 2 프로세서로부터 수신하는 단계를 포함하고,

상기 방법은,

상기 제 2 프로세서에 의해 송신되는 상기 메시지를 수신하는 것에 응답하여 활성 플래그를 클리어하는 단계;

상기 제 1 TLB 엔트리의 반환된 상기 인덱스를 이용하여 수신된 상기 메시지에 나타내어진 상기 TLB 액션들을 수행하는 단계; 및

상기 TLB 액션들을 수행하는 것에 응답하여 상기 활성 플래그를 설정하는 단계를 더 포함하는, 역 변환 색인 버퍼 (TLB) 룩-업을 수행하는 방법.

청구항 8

멀티-코어 이중 컴퓨팅 디바이스로서,

물리적 주소 입력을 제 1 프로세서로 획득하는 수단으로서, 상기 물리적 주소 입력은 공유 메모리에 대응하는 물리적 주소를 나타내는, 상기 물리적 주소 입력을 제 1 프로세서로 획득하는 수단;

상기 제 1 프로세서와 연관된 TLB 내의 제 1 TLB 엔트리로부터 제 1 가상 주소와 연관된 제 1 마스크를 획득하는 수단으로서, 획득된 상기 제 1 마스크는 비트 패턴인, 상기 제 1 마스크를 획득하는 수단;

상기 공유 메모리와 연관된 제 1 페이지 프레임 번호를 상기 제 1 TLB 엔트리로부터 획득하는 수단;

상기 획득된 제 1 마스크를 획득된 상기 제 1 페이지 프레임 번호에 적용하여 제 1 값을 생성하는 수단;

상기 획득된 제 1 마스크를 획득된 상기 물리적 주소 입력에 적용하여 제 2 값을 생성하는 수단; 및

상기 제 1 값과 상기 제 2 값을 비교하여 상기 제 1 값과 상기 제 2 값이 매칭하는지 여부를 결정하는 수단을 포함하는, 멀티-코어 이중 컴퓨팅 디바이스.

청구항 9

제 8 항에 있어서,

상기 제 1 값과 상기 제 2 값이 매칭하지 않는다고 결정하는 것에 응답하여 상기 TLB 내의 다른 TLB 엔트리들로부터 제 2 마스크 및 제 2 페이지 프레임 번호를 반복적으로 획득하는 수단;

획득된 상기 제 2 마스크를 획득된 상기 제 2 페이지 프레임 번호에 반복적으로 적용하여 상기 제 1 값을 생성하는 수단; 및

상기 획득된 제 2 마스크를 상기 획득된 물리적 주소 입력에 반복적으로 적용하여 상기 제 2 값을 생성하는 수단

을 더 포함하고,

상기 제 1 값과 상기 제 2 값을 비교하여 상기 제 1 값과 상기 제 2 값이 매칭하는지 여부를 결정하는 수단은

상기 제 1 값과 상기 제 2 값을 반복적으로 비교하여 상기 제 1 값과 상기 제 2 값이 매칭하는지 여부를 결정하는 수단을 포함하는, 멀티-코어 이중 컴퓨팅 디바이스.

청구항 10

제 9 항에 있어서,

상기 획득된 제 2 마스크를 상기 획득된 제 2 페이지 프레임 번호에 반복적으로 적용하여 상기 제 1 값을 생성하는 수단은 복사 없이 시스템 페이지 테이블에 있는 엔트리들로부터의 페이지 프레임 번호들을 재이용하는 수단을 포함하는, 멀티-코어 이중 컴퓨팅 디바이스.

청구항 11

제 8 항에 있어서,

상기 제 1 값과 상기 제 2 값이 매칭한다고 결정하는 것에 응답하여 상기 제 1 TLB 엔트리를 플러시하는 수단을 더 포함하는, 멀티-코어 이중 컴퓨팅 디바이스.

청구항 12

제 8 항에 있어서,

TLB 액션들 및 상기 물리적 주소를 나타내는 메시지를 제 2 프로세서로 전송하는 수단을 더 포함하고, 상기 TLB 액션들은 TLB 플러시 동작을 포함하는, 멀티-코어 이중 컴퓨팅 디바이스.

청구항 13

제 8 항에 있어서,

상기 제 1 값과 상기 제 2 값이 매칭한다고 결정하는 것에 응답하여 상기 제 1 TLB 엔트리의 인덱스를 반환하는 수단을 더 포함하는, 멀티-코어 이중 컴퓨팅 디바이스.

청구항 14

제 13 항에 있어서,

상기 물리적 주소 입력을 제 1 프로세서로 획득하는 수단은 TLB 액션들 및 상기 물리적 주소를 포함하는 메시지를 제 2 프로세서로부터 수신하는 수단을 포함하고,

상기 멀티-코어 이중 컴퓨팅 디바이스는,

상기 제 2 프로세서에 의해 송신되는 상기 메시지를 수신하는 것에 응답하여 활성 플래그를 클리어하는 수단;

상기 제 1 TLB 엔트리의 반환된 상기 인덱스를 이용하여 수신된 상기 메시지에 나타내어진 상기 TLB 액션들을 수행하는 수단; 및

상기 TLB 액션들을 수행하는 것에 응답하여 상기 활성 플래그를 설정하는 수단을 더 포함하는, 멀티-코어 이중 컴퓨팅 디바이스.

청구항 15

멀티-코어 이중 컴퓨팅 디바이스로서,

제 1 프로세서;

제 2 프로세서; 및

공유 메모리를 포함하고,

상기 제 1 프로세서는,

물리적 주소 입력을 상기 제 1 프로세서로 획득하는 것으로서, 상기 물리적 주소 입력은 상기 공유 메

모리에 대응하는 물리적 주소를 나타내는, 상기 물리적 주소 입력을 상기 제 1 프로세서로 획득하고;

상기 제 1 프로세서와 연관된 TLB 내의 제 1 TLB 엔트리로부터 제 1 가상 주소와 연관된 제 1 마스크를 획득하는 것으로서, 획득된 상기 제 1 마스크는 비트 패턴인, 상기 제 1 마스크를 획득하며;

상기 공유 메모리와 연관된 제 1 페이지 프레임 번호를 상기 제 1 TLB 엔트리로부터 획득하고;

상기 획득된 제 1 마스크를 획득된 상기 제 1 페이지 프레임 번호에 적용하여 제 1 값을 생성하며;

상기 획득된 제 1 마스크를 획득된 상기 물리적 주소 입력에 적용하여 제 2 값을 생성하고;

상기 제 1 값과 상기 제 2 값을 비교하여 상기 제 1 값과 상기 제 2 값이 매칭하는지 여부를 결정하는

것을 포함하는 동작들을 수행하는 프로세서-실행가능 명령들로 구성되는, 멀티-코어 이중 컴퓨팅 디바이스.

청구항 16

제 15 항에 있어서,

상기 제 1 프로세서는,

상기 제 1 값과 상기 제 2 값이 매칭하지 않는다고 결정하는 것에 응답하여 상기 TLB 내의 다른 TLB 엔트리들로부터 제 2 마스크 및 제 2 페이지 프레임 번호를 반복적으로 획득하고;

획득된 상기 제 2 마스크를 획득된 상기 제 2 페이지 프레임 번호에 반복적으로 적용하여 상기 제 1 값을 생성하며;

상기 획득된 제 2 마스크를 상기 획득된 물리적 주소 입력에 반복적으로 적용하여 상기 제 2 값을 생성하는

것을 더 포함하는 동작들을 수행하는 프로세서-실행가능 명령들로 구성되고,

상기 제 1 값과 상기 제 2 값을 비교하여 상기 제 1 값과 상기 제 2 값이 매칭하는지 여부를 결정하는 것은 상기 제 1 값과 상기 제 2 값을 반복적으로 비교하여 상기 제 1 값과 상기 제 2 값이 매칭하는지 여부를 결정하는 것을 포함하는, 멀티-코어 이중 컴퓨팅 디바이스.

청구항 17

제 16 항에 있어서,

상기 제 1 프로세서는 상기 획득된 제 2 마스크를 상기 획득된 제 2 페이지 프레임 번호에 반복적으로 적용하여 상기 제 1 값을 생성하는 것이 복사 없이 시스템 페이지 테이블에 있는 엔트리들로부터의 페이지 프레임 번호들을 재이용하는 것을 포함하도록 동작들을 수행하는 프로세서-실행가능 명령들로 구성되는, 멀티-코어 이중 컴퓨팅 디바이스.

청구항 18

제 15 항에 있어서,

상기 제 1 프로세서는 상기 제 1 값과 상기 제 2 값이 매칭한다고 결정하는 것에 응답하여 상기 제 1 TLB 엔트리를 플러시하는 것을 더 포함하는 동작들을 수행하는 프로세서-실행가능 명령들로 구성되는, 멀티-코어 이중 컴퓨팅 디바이스.

청구항 19

제 15 항에 있어서,

상기 제 1 프로세서는 TLB 액션들 및 상기 물리적 주소를 나타내는 메시지를 제 2 프로세서로 전송하는 것을 더 포함하는 동작들을 수행하는 프로세서-실행가능 명령들로 구성되고, 상기 TLB 액션들은 TLB 플러시 동작을 포함하는, 멀티-코어 이중 컴퓨팅 디바이스.

청구항 20

제 15 항에 있어서,

상기 제 1 프로세서는 상기 제 1 값과 상기 제 2 값이 매칭한다고 결정하는 것에 응답하여 상기 제 1 TLB 엔트리의 인덱스를 반환하는 것을 더 포함하는 동작들을 수행하는 프로세서-실행가능 명령들로 구성되는, 멀티-코어 이중 컴퓨팅 디바이스.

청구항 21

제 20 항에 있어서,

상기 제 1 프로세서는 상기 물리적 주소 입력을 상기 제 1 프로세서로 획득하는 것이 TLB 액션들 및 상기 물리적 주소를 포함하는 메시지를 상기 제 2 프로세서로부터 수신하는 것을 포함하도록 동작들을 수행하는 프로세서-실행가능 명령들로 구성되고,

상기 제 1 프로세서는,

상기 제 2 프로세서에 의해 송신되는 상기 메시지를 수신하는 것에 응답하여 활성 플래그를 클리어하고;

상기 제 1 TLB 엔트리의 반환된 상기 인덱스를 이용하여 수신된 상기 메시지에 나타내어진 상기 TLB 액션들을 수행하며;

상기 TLB 액션들을 수행하는 것에 응답하여 상기 활성 플래그를 설정하는 것을 더 포함하는 동작들을 수행하는 프로세서-실행가능 명령들로 구성되는, 멀티-코어 이중 컴퓨팅 디바이스.

청구항 22

프로세서-실행가능 소프트웨어 명령들을 저장한 비밀시적 프로세서-판독가능 저장 매체로서,

상기 프로세서-실행가능 소프트웨어 명령들은, 제 1 프로세서로 하여금,

물리적 주소 입력을 제 1 프로세서로 획득하는 것으로서, 상기 물리적 주소 입력은 공유 메모리에 대응하는 물리적 주소를 나타내는, 상기 물리적 주소 입력을 제 1 프로세서로 획득하고;

상기 제 1 프로세서와 연관된 TLB 내의 제 1 TLB 엔트리로부터 제 1 가상 주소와 연관된 제 1 마스크를 획득하는 것으로서, 획득된 상기 제 1 마스크는 비트 패턴인, 상기 제 1 마스크를 획득하며;

상기 공유 메모리와 연관된 제 1 페이지 프레임 번호를 상기 제 1 TLB 엔트리로부터 획득하고;

상기 획득된 제 1 마스크를 획득된 상기 제 1 페이지 프레임 번호에 적용하여 제 1 값을 생성하며;

상기 획득된 제 1 마스크를 획득된 상기 물리적 주소 입력에 적용하여 제 2 값을 생성하고;

상기 제 1 값과 상기 제 2 값을 비교하여 상기 제 1 값과 상기 제 2 값이 매칭하는지 여부를 결정하는

것을 포함하는 동작들을 수행하게 하도록 구성되는, 비밀시적 프로세서-판독가능 저장 매체.

청구항 23

제 22 항에 있어서,

저장된 상기 프로세서-실행가능 소프트웨어 명령들은, 상기 제 1 프로세서로 하여금,

상기 제 1 값과 상기 제 2 값이 매칭하지 않는다고 결정하는 것에 응답하여 상기 TLB 내의 다른 TLB 엔트리들로부터 제 2 마스크 및 제 2 페이지 프레임 번호를 반복적으로 획득하고;

획득된 상기 제 2 마스크를 획득된 상기 제 2 페이지 프레임 번호에 반복적으로 적용하여 상기 제 1 값을 생성하며;

상기 획득된 제 2 마스크를 상기 획득된 물리적 주소 입력에 반복적으로 적용하여 상기 제 2 값을 생성하는

것을 더 포함하는 동작들을 수행하게 하도록 구성되고,

상기 제 1 값과 상기 제 2 값을 비교하여 상기 제 1 값과 상기 제 2 값이 매칭하는지 여부를 결정하는 것은 상기 제 1 값과 상기 제 2 값을 반복적으로 비교하여 상기 제 1 값과 상기 제 2 값이 매칭하는지 여부를 결정하는 것을 포함하는, 비밀시적 프로세서-판독가능 저장 매체.

청구항 24

제 23 항에 있어서,

상기 저장된 프로세서-실행가능 소프트웨어 명령들은, 상기 제 1 프로세서로 하여금, 상기 획득된 제 2 마스크를 상기 획득된 제 2 페이지 프레임 번호에 반복적으로 적용하여 상기 제 1 값을 생성하는 것이 복사 없이 시스템 페이지 테이블에 있는 엔트리들로부터의 페이지 프레임 번호들을 재이용하는 것을 포함하도록 동작들을 수행하게 하도록 구성되는, 비밀시적 프로세서-판독가능 저장 매체.

청구항 25

제 22 항에 있어서,

저장된 상기 프로세서-실행가능 소프트웨어 명령들은, 상기 제 1 프로세서로 하여금, 상기 제 1 값과 상기 제 2 값이 매칭한다고 결정하는 것에 응답하여 상기 제 1 TLB 엔트리를 플러시하는 것을 더 포함하는 동작들을 수행하게 하도록 구성되는, 비밀시적 프로세서-판독가능 저장 매체.

청구항 26

제 22 항에 있어서,

저장된 상기 프로세서-실행가능 소프트웨어 명령들은, 상기 제 1 프로세서로 하여금, TLB 액션들 및 상기 물리적 주소를 나타내는 메시지를 제 2 프로세서로 전송하는 것을 더 포함하는 동작들을 수행하게 하도록 구성되고, 상기 TLB 액션들은 TLB 플러시 동작을 포함하는, 비밀시적 프로세서-판독가능 저장 매체.

청구항 27

제 22 항에 있어서,

저장된 상기 프로세서-실행가능 소프트웨어 명령들은, 상기 제 1 프로세서로 하여금, 상기 제 1 값과 상기 제 2 값이 매칭한다고 결정하는 것에 응답하여 상기 제 1 TLB 엔트리의 인덱스를 반환하는 것을 더 포함하는 동작들을 수행하게 하도록 구성되는, 비밀시적 프로세서-판독가능 저장 매체.

청구항 28

제 27 항에 있어서,

상기 저장된 프로세서-실행가능 소프트웨어 명령들은, 상기 제 1 프로세서로 하여금, 상기 물리적 주소 입력을 상기 제 1 프로세서로 획득하는 것이 TLB 액션들 및 상기 물리적 주소를 포함하는 메시지를 제 2 프로세서로부터 수신하는 것을 포함하도록 동작들을 수행하게 하도록 구성되고,

상기 저장된 프로세서-실행가능 소프트웨어 명령들은, 상기 제 1 프로세서로 하여금,

상기 제 2 프로세서에 의해 송신되는 상기 메시지를 수신하는 것에 응답하여 활성 플래그를 클리어하고;

상기 제 1 TLB 엔트리의 반환된 상기 인덱스를 이용하여 수신된 상기 메시지에 나타내어진 상기 TLB 액션들을 수행하며;

상기 TLB 액션들을 수행하는 것에 응답하여 상기 활성 플래그를 설정하는 것을 더 포함하는 동작들을 수행하게 하도록 구성되는, 비밀시적 프로세서-판독가능 저장 매체.

발명의 설명**배경 기술**

[0001]

컴퓨팅 디바이스들은 다양한 명령들/시스템들을 실행하는 수많은 프로세서들을 종종 포함한다. 예를 들어, 스마트 모바일 디바이스는 다운로드된 애플리케이션과 관련된 루틴들을 수행하기 위한 애플리케이션 프로세서, 뿐만 아니라 시그널링 프로세싱 동작들을 수행하기 위한 디지털 신호 프로세서를 사용할 수도 있다. 그러한 프로세서들은 그것들의 각각의 프로세스들이 시스템 메모리 내의 다양한 물리적 주소들에 맵핑하는 단순한 메모

리 주소들을 이용하여 데이터에 액세스할 수도 있도록 가상 메모리 주소 기법을 사용하도록 각각 구성될 수도 있다. 예를 들어, 애플리케이션 프로세서는 연속적인 가상 주소들을 활용하여 애플리케이션에 대한 데이터와 연관된 임의적 RAM 위치들에 액세스할 수도 있다.

[0002]

디바이스들은 지금 상이한 아키텍처들의 다양한 프로세서들을 사용하기 시작하고 있다. 그러한 이종 시스템은 소프트웨어가 공유 메모리로부터 혜택을 받을 가능성을 증가시킬 수도 있고, OpenCL, Renderscript 등과 같은 다양한 애플리케이션 프로그래밍 인터페이스(application programming interfaces; API) 들을 사용할 수도 있다. 그러나, 어드레싱 기법들에서의 차이들은 종래의 접근법들을 이용하는 비효율적인 메커니즘들을 요구할 수도 있다. 예를 들어, 현대의 스마트폰들과 같은 이종 시스템 디바이스들은 64-비트 및 32-비트 프로세서들 양자 모두 및/또는 프로세싱 코어들을 사용할 수도 있다. 상이한 아키텍처들로, 다양한 프로세서들은 상이한 가상 메모리 주소 기법들을 활용하여 동일한 (또는 공유) 시스템 메모리에 액세스할 수도 있다. 예를 들어, RAM의 동일한 페이지 (또는 메모리 블록)가 제 1 및 제 2 프로세서 양자 모두에 의해 이용될 수도 있다; 그러나, 그 페이지는 2 개의 프로세서들에 의한 2 개의 상이한 가상 주소들에 의해 참조될 수도 있다.

그러한 이종 시스템들에서의 프로세서들은 가상 주소들을 통해 시스템 메모리에 액세스하기 위한 속도 및 효율을 향상시키는 변환 색인 버퍼(TLB)를 각각 활용할 수도 있고, 따라서 TLB 일관성에 대한 강력한 지원을 요구할 수도 있다. TLB 들은 시스템 또는 공통 메모리에서의 가상 페이지들 대 물리적 페이지들의 다-대-일 맵핑을 이용하여, 프로세싱 시간 및 전력 소비의 면에서 비용이 많이 드는 엔트리들의 일관성 있는 식별을 행할 수도 있다.

[0003]

멀티-코어 시스템에서 일관성을 유지하기 위한 기법들이 존재한다. 예를 들어, Linux ARM TLB 플러시 메소드들 `flush_tlb_all()`, `flush_tlb_mm(memory_block_identifier)`, `flush_tlb_range(memory_block_identifier, start_index, end_index)`, 및 `flush_tlb_page(virtual_address, virtual_mask)` 과 같은, 가상 주소들을 이용하여 TLB 엔트리들을 수정하는 API 커맨드들이 있다. 다른 예로서, 일부 시스템들은 TLB 들 내의 엔트리들에 대한 록-업 및 무효 동작들을 위해 가상 주소들을 활용하는 TLB "슛-다운(shoot-down)" 기법을 수행할 수도 있다. 통상적인 TLB 슛-다운 절차는 제 2 프로세서의 페이지 테이블에서 무효화될 가상 주소를 포함하는 인터럽트를 제 2 프로세서에 전송하는 제 1 프로세서를 포함할 수도 있다.

발명의 내용

과제의 해결 수단

[0004]

일 양상에서, 물리적 주소 입력을 이용하여 역 변환 색인 버퍼(translation lookaside buffer; TLB) 록-업을 수행하는 방법은 물리적 주소 입력을 제 1 프로세서로 획득하는 단계 (여기서 물리적 주소 입력은 공유 메모리에 대응하는 물리적 주소를 나타낸다), 제 1 프로세서와 연관된 TLB 내의 제 1 TLB 엔트리로부터 제 1 가상 주소와 연관된 제 1 마스크를 획득하는 단계 (여기서 획득된 제 1 마스크는 비트 패턴이다), 공유 메모리와 연관된 제 1 페이지 프레임 번호를 제 1 TLB 엔트리로부터 획득하는 단계, 획득된 제 1 마스크를 획득된 제 1 페이지 프레임 번호에 적용하여 제 1 값을 생성하는 단계, 획득된 제 1 마스크를 획득된 물리적 주소 입력에 적용하여 제 2 값을 생성하는 단계, 및 제 1 값과 제 2 값을 비교하여 제 1 값과 제 2 값이 매칭하는지 여부를 결정하는 단계를 포함할 수도 있다. 다른 양상에서, 방법은 제 1 값과 제 2 값이 매칭하지 않는다고 결정하는 것에 응답하여 TLB 내의 다른 TLB 엔트리들로부터 제 2 마스크 및 제 2 페이지 프레임 번호를 반복적으로 획득하는 단계, 획득된 제 2 마스크를 획득된 제 2 페이지 프레임 번호에 반복적으로 적용하여 제 2 값을 생성하는 단계 (여기서 제 1 값과 제 2 값이 매칭하는지 여부를 결정하기 위해 제 1 값과 제 2 값을 비교하는 것은 제 1 값과 제 2 값을 반복적으로 비교하여 제 1 값과 제 2 값이 매칭하는지 여부를 결정하는 것을 포함할 수도 있다) 단계를 더 포함할 수도 있다. 다른 양상에서, 획득된 제 2 마스크를 획득된 제 2 페이지 프레임 번호에 반복적으로 적용하여 제 1 값을 생성하는 단계는 복사 없이 시스템 페이지 테이블에서의 엔트리들로부터의 페이지 프레임 번호들을 재이용하는 단계를 포함할 수도 있다. 다른 양상에서, 방법은 제 1 값과 제 2 값이 매칭한다고 결정하는 것에 응답하여 제 1 TLB 엔트리를 플러시하는(flush) 단계를 더 포함할 수도 있다. 다른 양상에서, 방법은 제 2 프로세서에 TLB 액션들 및 물리적 주소를 나타내는 메시지를 전송하는 단계를 더 포함할 수도 있으며, 여기서 TLB 액션들은 TLB 플러시 동작을 포함할 수도 있다. 다른 양상에서, 방법은 제 1 값과 제 2 값이 매칭한다고 결정하는 것에 응답하여 제 1 TLB 엔트리의 인덱스를 반환하는 단계를 더 포함할 수도 있다. 다른 양상에서, 물리적 주소를 제 1 프로세서로 획득하는 단계는 TLB 액션들 및 물리적 주소를 포함하는 메시지를 제 2 프로세서로부터 수신하는 단계를 포함할 수도 있고, 방법은 제 2 프로세서에 의해 송신된 메시지를 수신하는 것에 응답하여 활성 플래그를 클리어하는(clear) 단계, 제 1 TLB 엔트리의 반환된 인덱스를

이용하여 수신된 메시지에 나타내어진 TLB 액션들을 수행하는 단계, TLB 액션들을 수행하는 것에 응답하여 활성 플래그를 설정하는 단계를 더 포함할 수도 있다.

[0005] 일 양상에서, 멀티-코어 이중 컴퓨팅 디바이스는 물리적 주소 입력을 제 1 프로세서로 획득하는 수단 (여기서 물리적 주소 입력은 공유 메모리에 대응하는 물리적 주소를 나타낸다), 제 1 프로세서와 연관된 TLB 내의 제 1 TLB 엔트리로부터 제 1 가상 주소와 연관된 제 1 마스크를 획득하는 수단 (여기서 획득된 제 1 마스크는 비트 패턴일 수도 있다), 공유 메모리와 연관된 제 1 페이지 프레임 번호를 제 1 TLB 엔트리로부터 획득하는 수단, 획득된 제 1 마스크를 획득된 제 1 페이지 프레임 번호에 적용하여 제 1 값을 생성하는 수단, 획득된 제 1 마스크를 획득된 물리적 주소 입력에 적용하여 제 2 값을 생성하는 수단, 및 제 1 값과 제 2 값을 비교하여 제 1 값과 제 2 값이 매칭하는지 여부를 결정하는 수단을 포함할 수도 있다.

[0006] 일 양상에서, 멀티-코어 이중 컴퓨팅 디바이스는 제 1 프로세서, 제 2 프로세서, 및 공유 메모리를 포함할 수도 있으며, 여기서 제 1 프로세서는 물리적 주소 입력을 제 1 프로세서로 획득하는 것 (여기서 물리적 주소 입력은 공유 메모리에 대응하는 물리적 주소를 나타낸다), 제 1 프로세서와 연관된 TLB 내의 제 1 TLB 엔트리로부터 제 1 가상 주소와 연관된 제 1 마스크를 획득하는 것 (여기서 획득된 제 1 마스크는 비트 패턴일 수도 있다), 공유 메모리와 연관된 제 1 페이지 프레임 번호를 제 1 TLB 엔트리로부터 획득하는 것, 획득된 제 1 마스크를 획득된 제 1 페이지 프레임 번호에 적용하여 제 1 값을 생성하는 것, 획득된 제 1 마스크를 획득된 물리적 주소 입력에 적용하여 제 2 값을 생성하는 것, 및 제 1 값과 제 2 값을 비교하여 제 1 값과 제 2 값이 매칭하는지 여부를 결정하는 것을 포함할 수도 있는 동작들을 수행하기 위한 프로세서-실행가능 명령들로 구성될 수도 있다.

[0007] 일 양상에서, 프로세서-실행가능 소프트웨어 명령들이 저장된 비일시적 프로세서-판독가능 저장 매체는, 제 1 프로세서로 하여금, 물리적 주소 입력을 제 1 프로세서로 획득하는 동작 (여기서 물리적 주소 입력은 공유 메모리에 대응하는 물리적 주소를 나타낸다), 제 1 프로세서와 연관된 TLB 내의 제 1 TLB 엔트리로부터 제 1 가상 주소와 연관된 제 1 마스크를 획득하는 동작 (여기서 획득된 제 1 마스크는 비트 패턴일 수도 있다), 공유 메모리와 연관된 제 1 페이지 프레임 번호를 제 1 TLB 엔트리로부터 획득하는 동작, 획득된 제 1 마스크를 획득된 제 1 페이지 프레임 번호에 적용하여 제 1 값을 생성하는 동작, 획득된 제 1 마스크를 획득된 물리적 주소 입력에 적용하여 제 2 값을 생성하는 동작, 및 제 1 값과 제 2 값을 비교하여 제 1 값과 제 2 값이 매칭하는지 여부를 결정하는 동작을 수행하도록 구성된다.

도면의 간단한 설명

[0008] 본원에 첨부되고 본 명세서의 일부분을 구성하는 첨부 도면들은 본 발명의 예시적인 양상들을 도시하고, 위에서 주어진 일반적인 설명 및 하기에서 주어진 상세한 설명과 함께 본 발명의 특색들을 설명하는 역할을 한다.

도 1 은 제 1 유형의 제 1 프로세서 및 제 2 유형의 제 2 프로세서를 포함하는 멀티-코어 이중 컴퓨팅 디바이스의 컴포넌트 블록도이다.

도 2 는 공유 시스템 메모리와 연관된 물리적 주소를 활용하여 역 TLB 룩-업 동작들을 수행하도록 구성된 양상의 로직 회로의 컴포넌트 블록도이다.

도 3a 내지 도 3b 는 공유 시스템 메모리와 연관된 물리적 주소를 활용하여 역 TLB 룩-업 동작들을 수행하기 위한 멀티-코어 이중 컴퓨팅 디바이스에 대한 양상의 방법들을 도시하는 프로세스 흐름도들이다.

도 4 는 멀티-코어 이중 컴퓨팅 디바이스 내의 제 1 프로세서 및 제 2 프로세서에 의해 수행되는 역 TLB 룩-업 동작들을 이용하는 양상 방법들을 도시하는 프로세스 흐름도이다.

도 5 는 다양한 양상들에서의 이용에 적합한 멀티-코어 이중 컴퓨팅 디바이스의 컴포넌트 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0009] 다양한 양상들은 첨부 도면들을 참조하여 상세히 설명될 것이다. 가능한 경우에는 언제든지, 동일한 참조 부호들은 동일하거나 유사한 부분들을 지칭하기 위해 도면들에 걸쳐 이용될 것이다. 참조된 특정 예들 및 구현예들은 예시용이고, 본 발명 또는 본 청구항들의 범위를 제한하려고 의도되지는 않는다.

[0010] 단어 "예시적인" 은 본원에서 "예, 사례, 또는 실례로서의 역할을 하는" 것을 의미하기 위해 이용된다. 본원에서 "예시적인"으로 설명된 임의의 구현예는 반드시 다른 구현예들에 비해 더 선호되거나 더 유익한 것으로 해석될 필요는 없다.

- [0011] 용어들 "컴퓨팅 디바이스" 및 "멀티-코어 이중 컴퓨팅 디바이스" 는 본원에서 셀룰러 전화기들, 스마트폰들 (예를 들어, iPhone), 웹-패드들, 태블릿 컴퓨터들, 인터넷 가능 셀룰러 전화기들, WiFi 가능 전자 디바이스들, 개인용 휴대정보 단말기 (personal data assistant; PDA) 들, 랩탑 컴퓨터들, 데스크탑 컴퓨터들, 개인용 컴퓨터들, 및 적어도 제 1 프로세서 (또는 제 1 프로세싱 유닛) 및 제 2 프로세서 (또는 제 2 프로세싱 유닛) 를 구비한 유사한 전자 디바이스들 중 임의의 하나 또는 모두를 지칭하기 위해 이용되며, 여기서 제 1 프로세서의 아키텍처 또는 기술은 제 2 프로세서의 아키텍처 또는 기술과 상이하다. 예를 들어, 멀티-코어 이중 컴퓨팅 디바이스는 64-비트 애플리케이션 프로세서 및 32-비트 보조 프로세서를 포함할 수도 있다. 다양한 양상들에서, 그러한 디바이스들은 광역 네트워크 (wide area network; WAN) 또는 근거리 네트워크 (local area network; LAN) 접속 (예를 들어, LTE, 3G 또는 4G 무선 광대역 네트워크 트랜시버, 인터넷에 대한 유선 접속, 또는 WiFi) 을 확립하기 위한 네트워크 트랜시버로 구성될 수도 있다.
- [0012] 현재, 멀티-코어 시스템들에서 일관성을 유지하기 위한 회로, 하드웨어, 명령들, 소프트웨어, 및/또는 커맨드들은 다-대-일 가상 주소 기법들을 이용하는 이중 시스템들에 부적절하다. 예를 들어, 가상 주소들을 활용하는 기존의 API 커맨드들은 이중 시스템들에 대해서는 기능하지 않는데, 이러한 커맨드들이 시스템 내의 프로세서들에 의해 이용되는 가상 주소들 사이의 패리티를 가정하기 때문이다. 다른 예에서, 일관성을 유지하기 위한 공지의 숫-다운 기법들은 가상 주소들 및/또는 매칭하는 물리적 프레임을 식별하기 위해 프로세서의 TLB 내의 각각의 엔트리의 순차적 판독을 수반하여, 많은 사이클들을 요구할 수도 있다 (예를 들어, 프로세서는 각각의 TLB 엔트리를 반복적으로 검사하여 시스템 메모리 내의 소정의 물리적 메모리 주소와 관련되는 엔트리들을 식별할 수도 있다.).
- [0013] 다양한 양상들은 멀티-코어 이중 컴퓨팅 디바이스에서 변환 색인 버퍼 (TLB) 들을 활용하기 위한 디바이스들, 명령들, 및 방법들을 제공한다. 다양한 양상들에서, 멀티-코어 이중 컴퓨팅 디바이스의 프로세서는 연관된 TLB 에 대한 물리적 주소를 이용하여 TLB 룩-업을 수행하도록 구성될 수도 있다 (즉, "역 TLB 룩-업"). 그러한 역 TLB 룩-업 동작들은 TLB 와 연관되는 (그리고 따라서 가상 주소 기법을 활용하는) 멀티-코어 이중 컴퓨팅 디바이스 내의 임의의 프로세서 또는 프로세서 코어들에 의해 수행될 수도 있다. 예를 들어, 멀티-코어 이중 컴퓨팅 디바이스 내의 애플리케이션 프로세서 및 DSP 는 그것들 각각의 TLB 들과 관련된 TLB 룩-업 동작들을 동시에 또는 별도로 수행할 수도 있다. 설명의 간결함을 위해, 양상 동작들은 멀티-코어 이중 컴퓨팅 디바이스 내의 "제 1 프로세서" 및 "제 2 프로세서" 를 참조하여 하기에서 설명된다. 그러나, 그러한 일반적인 참조들은 프로세서의 특정 순서, 우선순위, 또는 유형을 요구하거나 암시하는 의도는 아니다.
- [0014] 역 TLB 룩-업을 위한 일 양상의 방법을 수행함에 있어서, 멀티-코어 이중 컴퓨팅 디바이스 내의 제 1 프로세서는 제 1 프로세서에 의한 문맥 전환에 기초하여 수정될 페이지와 연관된 시스템 메모리 위치와 같은 물리적 주소 (또는 물리적 주소 입력) 를 수신할 수도 있다. 수신된 물리적 주소는 시스템 호출 (예를 들어, API 커맨드) 또는 명령 (예를 들어, 축소 명령 세트 내의 하위-레벨 명령) 에서의 파라미터일 수도 있다. 제 1 프로세서는 제 1 프로세서와 연관된 TLB 의 제 1 엔트리 내의 가상 주소에 대응하는 마스크를 획득할 수도 있다. 제 1 프로세서는 또한 시스템 메모리와 관련되고 TLB 엔트리에 가상 주소와 함께 저장된 페이지 프레임 번호와 같은 가상 주소와 연관된 페이지 프레임 번호를 획득할 수도 있다. 제 1 프로세서는 획득된 마스크를 획득된 페이지 프레임 번호에 적용함으로써 제 1 값을 생성할 수도 있다. 병행하여, 제 1 프로세서는 마스크를 수신된 물리적 주소에 적용함으로써 제 2 값을 생성할 수도 있다. 제 1 프로세서는 그 다음에 제 1 값과 제 2 값을 비교하여 매칭이 있는지 여부를 결정할 수도 있다. 매칭이 발견되면, 제 1 프로세서는 페이지 사이즈 정보를 감안하여, 어떤 TLB 엔트리들이 물리적 주소들을 가지고 있는지를 나타내는 히트 벡터 (hit vector) 로 마스크 (및 그것의 대응하는 가상 주소) 와 연관된 TLB 엔트리의 인덱스를 저장할 수도 있다. 그러한 "히트" TLB 엔트리는 일관성을 증진하기 위해 엔트리를 무효화하거나 플러시하기 위해 제 1 프로세서 디바이스에 의해 이용될 수도 있다. 일 양상에서, 다수의 히트들이 히트 벡터에 저장될 수도 있다 (즉, 하나를 초과하는 TLB 엔트리가 수신된 물리적 주소에 대응할 수도 있다).
- [0015] 제 1 값과 제 2 값을 비교하는 것에 기초하여 매칭이 없다고 결정되는 경우, 제 1 프로세서는 다른 TLB 엔트리들에 가상 주소들과 연관되어 저장된 새로운 마스크들 및 페이지 프레임 번호들을 이용하여 새로운 제 1 및 제 2 값들을 계속 생성할 수도 있다. 다시 말해서, 제 1 프로세서는 TLB 에 표현된 마스크들을 반복적으로 적용해 제 2 값과의 매치가 있는지 여부를 결정할 수도 있다. 제 1 프로세서는 다양한 TLB 엔트리들의 모든 가상 주소들에 대해 이를 반복적으로 계속할 수도 있다. 일 양상에서, TLB 는 콘텐츠-주소지정가능 메모리 (content-addressable memory; CAM) 일 수도 있다.
- [0016] 일 양상에서, 멀티-코어 이중 컴퓨팅 디바이스 내의 제 2 프로세서는 역 TLB-룩-업들을 수행하기 위해 낮은 추

상 명령-레벨 기능을 활용할 수도 있다. 특히, 제 1 프로세서는 물리적 주소를 제어 레지스터 (예를 들어, ARM 프로세서에서의 제어 레지스터들 중 하나의 제어 레지스터) 에 복사하며, TLB 엔트리들의 가상 주소들과 연관된 마스크들을 물리적 주소에 적용하고, 그 마스크들을 갖는 물리적 주소와 그 마스크들을 갖는 페이지 프레임 번호들 (page frame number; PFN) 사이에 매치들이 발견되는 경우 TLB 엔트리 (엔트리들) 인덱스들을 반환하는 역 TLB 프로브 (probe) 명령을 활용할 수도 있다. 이러한 명령-레벨 기능으로, 제 2 프로세서는 빠르게, 비-직렬 방식으로 TLB 록-업 동작들을 수행할 수도 있다. 이러한 명령-레벨 기능의 일 양상의 하드웨어 구현이 하기에서 도 2 에 도시된다.

[0017] 다양한 양상들에서, 멀티-코어 이중 컴퓨팅 디바이스 내의 프로세서들은 어떤 TLB 엔트리들을 무효화시키거나 플러시할지를 나타내는 입력들과 같은, TLB 동작들에 대한 입력 값들로서 물리적 주소들을 이용하는 API 커맨드들을 활용하도록 구성될 수도 있다. API 커맨드들은 멀티-코어 이중 컴퓨팅 디바이스로 하여금 역 TLB 록-업의 소프트웨어 또는 하드웨어 구현들을 활용하게 할 수도 있다. 예를 들어, 멀티-코어 이중 컴퓨팅 디바이스 내의 프로세서들은 소정의 메모리 블록에 대한 범위 내의 물리적 주소들과 일치하는 제 1 프로세서의 TLB 엔트리들을 무효화하는 방법의 실행을 야기하는 API 커맨드 (예를 들어, `flush_tlb_range(memory_block_identifier, start_PA, end_PA)`) 를 이용할 수도 있다. 다른 예로서, 멀티-코어 이중 컴퓨팅 디바이스 내의 프로세서들은 소정의 물리적 주소들과 일치하는 제 1 프로세서의 TLB 엔트리들을 플러시하는 하드웨어 또는 회로의 이용을 야기할 수도 있는 다른 API 커맨드 (예를 들어, `flush_tlb_page(physical_address, memory_block_identifier)`) 를 이용할 수도 있다. 그러한 API 커맨드들은 TLB 액션들을 구현하기 위해 프로세서들에 물리적 주소 공간 프로토콜을 구현하기 위한 편리한 방식을 제공하는 가치가 있을 수도 있다. 일 양상에서, 멀티-코어 이중 컴퓨팅 디바이스 내의 프로세서들은 API 커맨드들을 활용하여 복수의 TLB 엔트리들의 일괄 무효화를 수행하도록 구성될 수도 있다.

[0018] 다른 양상에서, 멀티-코어 이중 컴퓨팅 디바이스 내의 프로세서들은 강화된 TLB 쏏-다운 방법 또는 (또는 쏏-다운 프로토콜) 을 수행하도록 구성될 수도 있으며, 여기서 프로세서는 가상 주소들과 반대되는 물리적 주소들을 입력으로서 활용하여 무효화되어야 하는 TLB 엔트리들을 식별할 수도 있다. 그러한 방법에서, 멀티-코어 이중 컴퓨팅 디바이스는 하기에서 설명되는 바와 같은 역 TLB 록-업 동작들을 수행하기 위해 하드웨어, 명령들, API 호출, 또는 소프트웨어 방법들을 활용할 수도 있다. 예를 들어, 제 1 TLB 내의 페이지의 가상 주소와 관련된 변환 정보가 수정될 필요가 있다고 검출하는 것에 응답하여, 멀티-코어 이중 컴퓨팅 디바이스 내의 제 1 프로세서는 프로세서-간 인터럽트들을 디스에이블하며, 제 1 TLB (및 제 1 프로세서) 와 연관된 활성 플래그를 클리어하며, 시스템 페이지 테이블을 잠그며, 역 TLB 명령 또는 회로를 이용해 물리적 주소를 이용하여 제 1 프로세서의 TLB 에 대한 록업을 수행하며, 물리적 주소와 관련된 TLB 엔트리들을 플러시하고, 멀티-코어 이중 컴퓨팅 디바이스 내의 제 2 프로세서가 제 2 프로세서의 TLB 에 있는 물리적 주소에 대응하는 엔트리들에 대해 무효화 동작을 수행해야 한다는 것을 나타내는 메시지를 전송할 수도 있다. 위에서 설명된 방법을 이용하여, 제 2 프로세서는 또한 예컨대, 역 TLB 록-업 명령 또는 회로를 통해, 물리적 주소를 이용하여 제 2 TLB 에 대해 TLB 록-업 동작을 수행할 수도 있고, 임의의 매칭하는 TLB 엔트리를 플러시할 수도 있다.

[0019] 다른 양상에서, ARM 아키텍처를 활용하는 디바이스들과 같은, 분산 가상 메모리 (Distributed Virtual Memory; DVM) 를 활용하도록 구성된 시스템 또는 디바이스는 본 개시물의 다양한 방법들 및/또는 명령들을 활용하도록 구성될 수도 있다. 메모리에서의 일관성 있는 메모리 관리 유닛 (memory management unit; MMU) 페이지 테이블들의 세트를 공유하는 멀티-클러스터 일관된 CPU 시스템들은 TLB 일관성을 요구할 수도 있다. 예를 들어, TLB 가 메모리에 있는 메모리 관리 유닛 (MMU) 페이지 테이블들의 캐쉬인 경우, 그러한 시스템은 마스터가 페이지 테이블들을 업데이트할 때 MMU 페이지 테이블 엔트리의 오래된 복사본을 포함할 수도 있는 TLB 엔트리들의 무효화들을 수행할 수도 있다. 캐쉬 일관성 프로토콜들에서의 분산 가상 메모리 지원은 물리적 주소들을 이용하여 무효화 메시지들을 브로드캐스팅하는 것으로 구성될 수도 있다. DVM 메시지들은 TLB 무효화, 분기 예측, (프로세서가 메모리에 코드를 기록한 경우에 있어서) 가상 또는 물리적 명령 캐쉬 무효화, 및 모든 이전의 DVM 커맨드들이 완료되기를 기다리는 동기화를 지원할 수도 있다. DVM 메시지들은 스누프 (snoop) 시그널링을 이용하여 관독 채널로 전송될 수도 있다. 시스템 MMU (SMMU) 는 그것의 엔트리들이 최신 상태인 것을 보장하기 위해 물리적 주소들을 이용하는 TLB 무효화 메시지들을 이용할 수도 있다.

[0020] 도 1 은 제 1 유형의 제 1 프로세서 (501) 및 제 2 유형의 제 2 프로세서 (531) 를 포함하는 멀티-코어 이중 컴퓨팅 디바이스 (500) 를 도시한다. 예를 들어, 제 1 프로세서 (501) 는 64-비트 아키텍처를 갖는 애플리케이션 프로세서일 수도 있고, 제 2 프로세서 (531) 는 32-비트 아키텍처를 갖는 코-프로세서 (coprocessor) (예를 들어, 그래픽 프로세싱 유닛, 디지털 신호 프로세서 등) 일 수도 있다. 제 1 프로세서 (501) 및/또는 제

2 프로세서 (531) 는 시스템-온-칩에 포함될 수도 있다. 제 1 프로세서 (501) 는 (도 1 에서 "Page Table_A" 라고 지칭되는) 제 1 페이지 테이블 (120) 과 연관될 수도 있고, 제 2 프로세서 (531) 는 (도 1 에서 "Page Table_B" 라고 지칭되는) 제 2 페이지 테이블 (140) 과 연관될 수도 있으며, 각각의 페이지 테이블 (120, 140) 은 가상 주소들을 공유 시스템 메모리 (502) 의 물리적 주소들에 링크하는 저장된 데이터, 예컨대, 애플리케이션 데이터의 부분들을 저장하는 랜덤 액세스 메모리 (RAM) 내의 인덱싱된 블록들 또는 페이지들을 포함한다. 일 양상에서, 시스템 메모리 (502) 는 시스템 페이지 테이블 (도 1 에 미도시) 과 연관될 수도 있다.

[0021] 예시용으로, 제 1 페이지 테이블 (120) 은 제 1 가상 주소 (즉, 제 1 프로세서 (501) 에 대응하는 주소 '1') 를 제 1 물리적 주소 메모리 블록 (150) (즉, 메모리 인덱스 '2') 에 링크하는 제 1 데이터 레코드 (122), 제 2 가상 주소 (즉, 제 1 프로세서 (501) 에 대응하는 주소 '2') 를 제 1 물리적 주소 메모리 블록 (152) (즉, 메모리 인덱스 '3') 에 링크하는 제 2 데이터 레코드 (124), 및 제 3 가상 주소 (즉, 제 1 프로세서 (501) 에 대응하는 주소 '3') 를 제 1 물리적 주소 메모리 블록 (154) (즉, 메모리 인덱스 'X') 에 링크하는 제 3 데이터 레코드 (126) 를 포함할 수도 있다. 제 2 페이지 테이블 (140) 은 제 4 가상 주소 (즉, 제 2 프로세서 (531) 에 대응하는 주소 '1') 를 제 3 물리적 주소 메모리 블록 (154) (즉, 메모리 인덱스 'X') 에 링크하는 제 4 데이터 레코드 (142), 및 제 5 가상 주소 (즉, 제 2 프로세서 (531) 에 대응하는 주소 '2') 를 제 3 물리적 주소 메모리 블록 (156) (즉, 메모리 인덱스 'X+3') 에 링크하는 제 4 데이터 레코드 (144) 를 포함할 수도 있다.

[0022] 메모리 (502) 가 공유 시스템 메모리이기 때문에, 제 1 페이지 테이블 (120) 및 제 2 페이지 테이블 (140) 은 메모리 (502) 의 동일한 물리적 주소들에 대한 참조들을 포함하는 데이터 레코드들을 포함할 수도 있다. 비제한적인 예로서: 제 3 데이터 레코드 (126) 는 제 3 가상 주소 (즉, 제 1 프로세서 (501) 에 대응하는 주소 '3') 를 제 3 물리적 주소 메모리 블록 (154) (즉, 메모리 인덱스 'X') 에 링크할 수도 있고, 제 4 데이터 레코드 (142) 는 제 4 가상 주소 (즉, 제 2 프로세서 (531) 에 대응하는 주소 '1') 를 제 3 물리적 주소 메모리 블록 (154) 에 링크할 수도 있다. 다시 말해서, 페이지 테이블들 (120, 140) 을 통해, 제 1 프로세서 (501) 및 제 2 프로세서 (531) 양자 모두는 상이한 가상 주소들을 이용하여 공유 시스템 메모리 (502) 에 저장된 동일한 데이터에 링크하고 액세스할 수도 있다.

[0023] 제 1 프로세서 (501) 는 또한 (도 1 에서 "TLB_A" 라고 지칭되는) 제 1 TLB (110) 와 연관될 수도 있고, 제 2 프로세서 (531) 는 (도 1 에서 "TLB_B" 라고 지칭되는) 제 2 TLB (130) 와 연관될 수도 있으며, 각각의 TLB (110, 130) 는 각기 제 1 페이지 테이블 (120) 및 제 2 페이지 테이블 (140) 에 저장된 데이터를 포함한다. 예를 들어, 제 1 TLB (110) 는 제 1 페이지 테이블 (120) 의 데이터 레코드들 (124, 126) 의 서브세트를 포함할 수도 있고, 제 2 TLB (130) 는 제 2 페이지 테이블 (140) 의 데이터 레코드들 (142, 144) 의 서브세트 (또는 모두) 를 포함할 수도 있다.

[0024] 도 2 는 공유 시스템 메모리와 연관된 물리적 주소를 활용하여 역 TLB 룩-업 동작들을 수행하도록 구성된 일 양상의 멀티-코어 이중 컴퓨팅 디바이스 (500) 를 도시한다. 멀티-코어 이중 컴퓨팅 디바이스 (500) 는 TLB 또는 메모리 엔트리들의 복사 없이 그리고 중요한 시간 영향없이 하드웨어 역 TLB 프로브 명령들을 수행하기 위한 회로, 하드웨어, 및/또는 다양한 모듈들로 구성될 수도 있다. 특히, 멀티-코어 이중 컴퓨팅 디바이스 (500) 는 물리적 주소들과 연관되는 가상 주소들 (203-205) 에 대응하는 마스크들 (206-208) 의 세트를 저장할 수도 있는 (도 2 에서 'CAM' 이라고 지칭되는) 콘텐츠-주소지정가능 메모리 (202) 를 포함할 수도 있다. CAM (202) 은 프로세서 (예를 들어, 애플리케이션 프로세서, 제 1 프로세서, 제 2 프로세서 등) 에 대한 TLB 로서 기능할 수도 있다. 마스크들 (206-208) 은 페이지 프레임 번호 또는 가상 주소 값들의 최상위 비트들의 수를 제어하는데 이용될 수도 있고, 통상적으로 대응하는 가상 주소들 (203-205) 과 함께 TLB 엔트리들에 저장된 비트 필드들로서 구현될 수도 있다. 일 양상에서, CAM (202) 은 가상 주소들 (203-205) 및 마스크들 (206-208) 과 연관하여 또한 페이지 프레임 번호들의 세트를 저장할 수도 있다. 예를 들어, CAM (202) 은 관련된 가상 주소들 (203-205) 의 TLB 엔트리들에 저장하기 위해 시스템 페이지 테이블로부터 페이지 프레임 번호들을 수신하여 저장할 수도 있다.

[0025] 멀티-코어 이중 컴퓨팅 디바이스 (500) 는 또한 물리적 주소 입력 (220) 을 수신하여 저장하기 위한 하드웨어/회로부, 예컨대, 레지스터, 시스템 변수, 또는 다른 버퍼 엘리먼트를 포함할 수도 있고, 도 2 의 목적에서는 "입력 모듈" (250) 이라고 지칭될 수도 있다. 예를 들어, 멀티-코어 이중 컴퓨팅 디바이스 (500) 가 물리적 주소를 이용하여 TLB 엔트리들을 무효화하기 위해 API 커맨드의 인스턴스를 검출하는 것에 응답하여, 멀티-코어 이중 컴퓨팅 디바이스 (500) 는 입력 모듈 (250) 과 연관된 레지스터 내에 물리적 주소 입력 (220) 을 저장할 수도 있다. 일 양상에서, 입력 모듈 (250) 은 CAM (202) 의 일부분일 수도 있다. 다른 양상에서, 입력

모듈 (250) 및/또는 CAM (202) 은 멀티-코어 이중 컴퓨팅 디바이스 (500) 내의 개개의 프로세서와 연관될 수도 있다.

[0026] 멀티-코어 이중 컴퓨팅 디바이스 (500) 는 또한 랜덤 액세스 메모리 (또는 RAM (502)) 와 같은 공유 시스템 메모리를 관리하기 위한 RAM 제어기 (240) 를 포함할 수도 있다. RAM 제어기 (240) 는 또한 TLB 록-업 동작들과 연관된 정보를 저장하기 위해 RAM 제어기 (240) 에 의해 이용될 수도 있는 버퍼와 같은 히트 벡터를 활용할 수도 있다. 예를 들어, RAM 제어기 (240) 는 "히트들" (즉, CAM (202) 의 엔트리들에 저장된 데이터 및 물리적 주소와 같은 검색 토큰 사이의 매치들) 을 검출하는 것에 응답하여 CAM (202) 의 인덱스들 (또는 엔트리 인덱스들) 을 저장할 수도 있다. 일 양상에서, 히트 벡터는 CAM (202) 엔트리들의 다수의 인덱스들 (또는 히트들) 을 저장할 수도 있다. RAM 제어기 (240) 는, 예컨대, 엔트리마다 (entry by entry) (또는 사이클마다 (cycle by cycle)) RAM (502) 의 각각의 메모리 블록에 저장된 데이터를 평가함으로써, RAM (502) 을 버스트 판독하도록 (burst read) 구성될 수도 있다. 특히, RAM 제어기 (240) 는 한 번에 하나씩 RAM (502) 의 블록들 (211-213) 을 반복적으로 판독하여 블록들 (211-213) 의 각각과 연관된 페이지 프레임 번호들을 식별하도록 구성될 수도 있다. 일 양상에서, 멀티-코어 이중 컴퓨팅 디바이스 (500) 는 RAM 제어기 (240) 가 판독하는/평가하는 각각의 엔트리에 대해 하나의 클럭 사이클을 이용할 수도 있다. 일 양상에서, 페이지 프레임 번호들은 RAM (502) 과 연관된 시스템 페이지 테이블 내에 나타내어질 수도 있다. 다른 양상에서, 블록들 (211-213) 은 CAM (202) 의 TLB 엔트리들과 연관될 수도 있다. 예를 들어, 제 1 블록 (211) 의 페이지 프레임 번호 (예를 들어, "PFN_1") 는 TLB 엔트리에 저장된 제 1 가상 주소 (203) (예를 들어, "VA_1") 및 제 1 마스크 (206) (예를 들어, "Mask_1") 와 연관될 수도 있다.

[0027] RAM 제어기 (240) 는, RAM (502) 의 블록들 (211-213) 의 각각의 블록을 판독하는 것에 응답하여, 예컨대, 페이지 프레임 번호 (예를 들어, 'PFN_1', 'PFN_2', 'PFN_n' 등) 를 나타내는 신호 (214) 를 송신함으로써, 다양한 페이지 프레임 번호들을 나타내는 신호들 (214) 을 송신할 수도 있다. RAM 제어기 (240) 는 신호들 (214) 을 통해 반복적으로 그리고 한 번에 하나씩 개개의 페이지 프레임 번호들을 송신할 수도 있다. 멀티-코어 이중 컴퓨팅 디바이스 (500) 는 RAM 제어기 (240) 를 통해 송신된 신호들 (214) 을 CAM (202) 에 저장된 데이터와의 비교 또는 마스크 동작들을 위해 나타내어진 페이지 프레임 번호들을 이용하도록 구성될 수도 있다.

[0028] 일 양상에서, 멀티-코어 이중 컴퓨팅 디바이스 (500) 는 CAM (202) 에 저장된 가상 주소들 (203-205) 에 대응하는 다양한 마스크들 (206-208) 을 나타내는 신호들 (210, 210') 을 송신하도록 구성될 수도 있다. 예를 들어, 멀티-코어 이중 컴퓨팅 디바이스 (500) 는 입력 모듈 (250) 에 의한 이용을 위한 신호들 (210') 및/또는 CAM (202) 에 의한 이용을 위한 신호들 (210) 을 송신할 수도 있다. CAM (202) 에 저장된 가상 주소들 (203-205) 에 대응하는 개개의 마스크들 (206-208) 은 신호들 (210, 210') 을 통해 반복적으로 그리고 한 번에 하나씩 송신될 수도 있다.

[0029] 멀티-코어 이중 컴퓨팅 디바이스 (500) 는 신호들 (210, 210') 을 통해 송신하는 마스크들 (206-208) 과 연계하여 마스크 적용 컴포넌트들 (201, 201') 을 활용하도록 구성될 수도 있다. 마스크 적용 컴포넌트들 (201, 201') 은, 다중화기 또는 게이트와 같은, 2 개의 입력 값들을 단일의 고유한 결과 값으로 필터링, 증가, 디코딩, 결합, 또는 그렇지 않으면 변화시키기 위한 회로, 하드웨어, 또는 모듈일 수도 있다. 예를 들어, 마스크 적용 컴포넌트들 (201, 201') 은 수학적 연산, 비트 패턴, 또는 등식을 페이지 프레임 번호 입력 값들에 적용하는 하드웨어 로직일 수도 있다. 다른 예로서, 마스크 적용 컴포넌트들 (201, 201') 은 마스크들 (206-208) 중 하나의 마스크를 주소 (예를 들어, 물리적 메모리 주소) 의 어떤 비트들 또는 프레임 번호가 영-출력될지를 (zero out) 나타내는 비트 패턴으로서 적용할 수도 있다. 다른 양상에서, 마스크 적용 컴포넌트들 (201, 201') 은 상태 머신들일 수도 있다.

[0030] CAM (202) 은 제 1 마스크 적용 컴포넌트 (201) 를 통해 마스크들 (206-208) 을 적용하도록 구성될 수도 있고, 입력 모듈 (250) 은 제 2 마스크 적용 컴포넌트 (201') 를 통해 마스크들 (206-208) 을 적용하도록 구성될 수도 있다. 특히, 제 1 마스크 적용 컴포넌트 (201) 를 이용하여, CAM (202) 은 신호들 (210) 내에 개별적으로 나타내어진 마스크들 (206-208) 을 신호들 (214) 내에 개별적으로 나타내어진 페이지 프레임 번호들에 적용하도록 구성될 수도 있다. 예를 들어, CAM (202) 은 제 1 마스크 적용 컴포넌트 (201) 를 통해 마스크들 (206-208) 을 신호들 (214) 을 통해 송신된 페이지 프레임 번호들 (예를 들어, 'PFN_1' 등) 에 반복적으로 적용하여 출력 값들을 생성할 수도 있다. 일 양상에서, 멀티-코어 이중 컴퓨팅 디바이스 (500) 는 수신된 신호들 (214) 내의 데이터를 재이용하도록 (즉, 데이터의 복사 없이 시스템 페이지 테이블에 있는 엔트리들로부터의 페이지 프레임 번호들을 재이용) 구성될 수도 있다. 다시 말해서, CAM (202) 은, 예컨대, 신호들 (214) 을 통해 수신된 다양한 페이지 프레임 번호들을 버퍼링하거나 그렇지 않으면 저장함으로써, 제 1 마스크 적용 컴포넌

트 (201) 를 통한 마스크들 (206-208) 의 페이지 프레임 번호들에 대한 반복적인 적용들 중에 비교기들이 복사하는 것을 방지하도록 구성될 수도 있다. CAM (202) 은 제 1 마스크 적용 컴포넌트 (201) 를 통해 마스크들 (206-208) 을 다양한 페이지 프레임 번호들에 적용함으로써 출력 값들을 나타내는 신호들 (265) 을 송신할 수도 있다.

[0031] 또한, 제 2 마스크 적용 컴포넌트 (201') 를 이용하여, 입력 모듈 (250) 은 신호들 (210') 내에 개별적으로 나타내어진 마스크들 (206-208) 을 물리적 주소 입력 (220) 에 적용하도록 구성될 수도 있다. 예를 들어, 제 2 마스크 적용 컴포넌트 (201') 를 통해, 입력 모듈 (250) 은 제 1 마스크 (206) 를 물리적 주소 입력 (220) 에 의해 나타내어진 물리적 주소에 적용하여 출력 값을 생성할 수도 있다. 입력 모듈 (250) 은 제 2 마스크 적용 컴포넌트 (201') 를 통해 마스크들 (206-208) 을 적용함으로써 출력 값들을 나타내는 신호들 (260) 을 송신할 수도 있다.

[0032] 각기 제 1 및 제 2 마스크 적용 컴포넌트들 (201, 201') 을 통해 CAM (202) 및 입력 모듈 (250) 에 의해 동일한 마스크들 (206-208) 이 이용될 수도 있기 때문에, 신호들 (260, 265) 에서 송신된 출력 값들은 동일한 마스크를 동일한 값 (즉, 신호들 (214) 에서의 물리적 주소 입력 (220) 또는 페이지 프레임 번호들) 에 적용함으로써 생성되는 경우 동일할 수도 있다. 예를 들어, CAM (202) 와 연관된 제 1 마스크 적용 컴포넌트 (201) 및 입력 모듈 (250) 과 연관된 제 2 마스크 적용 컴포넌트 (201') 로부터의 출력 값들은 물리적 주소 입력 (220) 이 신호들 (214) 을 통해 CAM (202) 에 의해 수신된 페이지 프레임 번호와 동일한 경우 동일할 수도 있다.

[0033] 멀티-코어 이중 컴퓨팅 디바이스 (500) 는 비교 회로 (230) 를 활용하여 마스크 적용 컴포넌트들 (201, 201') 로부터 마스크들 (206-208) 을 적용한 결과이고 신호들 (260, 265) 에 의해 송신되는 출력 값들이 동일한지 여부를 결정할 수도 있다. 다시 말해서, 비교 회로 (230) 는 CAM (202) 및 입력 모듈 (250) 이 매칭하는 출력 값들을 생성했는지 여부, 및 따라서 CAM (202) 및 입력 모듈 (250) 이 동일한 입력 값들로 각기의 마스크 적용 컴포넌트들 (201, 201') 을 이용했는지 여부 (즉, "히트" 가 있는지 여부) 를 결정하는데 이용될 수도 있다. 비교 회로 (230) 를 통해, CAM (202) 은 신호들 (210) 내의 마스크들 (206-208) 을 신호들 (214) 내에서 수신된 페이지 프레임 번호들에 적용한 결과들과 병행하여 신호들 (260) 에 있는 출력 값들을 비교하도록 구성될 수도 있다. 신호들 (260, 265) 로부터의 출력 값들이 동일하다고 (즉, "히트") 비교 회로 (230) 가 결정하는 경우, 출력 값들 (즉, 마스크들 (206-208)) 에 대응하는 CAM (202) (또는 TLB) 엔트리들 또는 엔트리 인덱스들이 마스크 적용 컴포넌트들 (201, 201') 을 이용하여 동일한 출력 값들을 생성했다고 나타내는 신호들 (270) 이 RAM 제어기 (240) 로 송신될 수도 있다. 일 양상에서, CAM (202) 은, 예컨대, CAM (202) 에서의 각각의 엔트리에 대해 하나씩, 복수의 비교 회로들을 활용함으로써 병렬 비교들을 수행할 수도 있다.

[0034] 도 3a 는 공유 시스템 메모리와 연관된 물리적 주소를 활용하여 역 TLB 룩-업 동작들을 수행하는 멀티-코어 이중 컴퓨팅 디바이스의 프로세서에 대한 일 양상 방법 (300) 을 도시한다. 방법 (300) 은 멀티-코어 이중 컴퓨팅 디바이스 내의 다른 프로세서들에 의해 수행된 액션들에 응답하여 수정되거나 플러시될 필요가 있는 엔트리들과 같은, 특정 물리적 주소들에 대응하는 TLB 엔트리들을 식별하도록 수행될 수도 있다. 예를 들어, 제 1 프로세서가 RAM 의 공유 메모리 블록 내의 콘텐츠를 수정하는 것에 응답하여, 멀티-코어 이중 컴퓨팅 디바이스에서의 제 2 프로세서는 TLB 슛-다운 메커니즘의 일부로서 방법 (300) 을 수행할 수도 있다. 다양한 양상들에서, 방법 (300) 은, 위에서 설명된 바와 같이, 명령, 소프트웨어 방법, 또는 하드웨어를 통해 멀티-코어 이중 컴퓨팅 디바이스 내의 프로세서들로 구현될 수도 있다. 설명의 간결함을 위해, 방법 (300) 양상 동작들은 멀티-코어 이중 컴퓨팅 디바이스 내의 "제 1 프로세서" 를 참조하여 하기에서 설명된다.

[0035] 블록 302 에서, 멀티-코어 이중 컴퓨팅 디바이스 내의 제 1 프로세서는, 예컨대, API 커맨드와 관련하여 제 1 프로세서로 물리적 주소 입력을 수신함으로써, 공유 메모리에 대응하는 물리적 주소 입력을 획득할 수도 있다. 물리적 주소 입력은 시스템 메모리 블록 식별자, 페이지 프레임 번호 혹은 식별자, 또는 공유 시스템 메모리와 관련된 다른 인덱스 값일 수도 있다. 멀티-코어 이중 컴퓨팅 디바이스는, 물리적 주소에 대응하는 TLB 엔트리에 대한 커맨드들이라고 불리는 바와 같은, 물리적 주소를 나타내는 명령들, API 커맨드들, 및/또는 방법들의 인스턴스들이 무효화되거나 플러시되는 것을 모니터링할 수도 있다. 예를 들어, 멀티-코어 이중 컴퓨팅 디바이스는 API 커맨드 "무효화(PA)" 가 실행을 위해 호출되었음을 검출할 수도 있으며, 여기서 'PA' 는 무효화될 특정 물리적 주소를 표현하는 입력 변수 또는 파라미터이다.

[0036] 블록 304 에서, 제 1 프로세서는 제 1 프로세서와 연관된 TLB 내의 TLB 엔트리를 선택할 수도 있다. 다양한 양상들에서, 선택된 TLB 엔트리는 가상 주소, 가상 주소와 연관된 마스크, 및 디바이스의 공유 메모리와 연관된 시스템 페이지 테이블로부터의 페이지 프레임 번호를 포함할 수도 있다. 예를 들어, TLB 엔트리는 소정의

가상 주소, 비트 패턴 마스크, 및 시스템 페이지 테이블로부터의 연관된 페이지 프레임 번호를 포함하여, 다양한 데이터를 포함하는 데이터 테이블 레코드일 수도 있다. TLB 엔트리는 TLB 엔트리와 관련하여 저장된 정보를 참조하는데 이용될 수도 있는, 대응하는 인덱스 또는 데이터 테이블 인덱스와 같은 다른 식별자를 가질 수도 있다. 일 양상에서, TLB 는 위에서 설명된 바와 같은 CAM 일 수도 있다.

[0037] 블록 306 에서, 제 1 프로세서는 선택된 TLB 엔트리로부터 가상 주소와 연관된 마스크를 획득할 수도 있다. 각각의 TLB 엔트리는 다양한 마스크 사이즈들 (예를 들어, 4K, 4MB 등) 일 수도 있는 마스크를 포함할 수도 있다. 마스크는 비트 패턴 또는 비트 표현일 수도 있다. 제 1 프로세서는 블록 308 에서 선택된 TLB 엔트리로부터 페이지 프레임 번호를 획득할 수도 있다. 일 양상에서, 중복 사이클들을 방지하고 병목현상들을 방지하기 위해, TLB 파플레이션 (population) 시에, 멀티-코어 이중 컴퓨팅 디바이스는 제 1 프로세서를 통해 페이지 프레임 번호들을 TLB 에 기록하게 할 수도 있고, 따라서 방법 (300) 의 수행 중에 시스템 메모리로부터 어떤 것도 불러올 필요가 없을 수도 있다. 예를 들어, 획득된 페이지 프레임 번호는 실행시간 전에 TLB 에 저장하기 위한 시스템 페이지 테이블로부터 추출된 시스템 메모리에 현재 저장된 페이지의 물리적 주소일 수도 있다. 일 양상에서, 페이지 프레임 번호는 인덱스 또는 메모리 블록 식별자를 나타내는 정보일 수도 있다.

[0038] 블록 310 에서, 제 1 프로세서는 획득된 마스크를 획득된 페이지 프레임 번호에 적용하여 제 1 출력 값을 생성할 수도 있다. 위에서 설명된 바와 같이, 제 1 프로세서는 입력 값들 (예를 들어, 획득된 마스크 및 획득된 페이지 프레임 번호) 을 취하여 출력 값을 생성하고, 동일한 입력 값들을 이용하여 복사하거나 반복될 수 있는 필터, 등식, 방법, 또는 다른 동작을 통해 마스크를 적용할 수도 있다. 예를 들어, 제 1 프로세서는 마스크 비트 패턴을 획득된 페이지 프레임 번호에 적용하여 비트들을 0 으로 출력할 수도 있다. 다시 말해서, 동일한 마스크가 동일한 입력 값과 이용되는 경우, 동일한 출력 값이 생성될 것이다. 블록 310 에서의 동작들과 유사하게, 블록 312 에서, 제 1 프로세서는 획득된 마스크를 획득된 물리적 주소 입력에 적용하여 제 2 출력 값을 생성할 수도 있다.

[0039] 블록 314 에서, 제 1 프로세서는 제 1 출력 값과 제 2 출력 값을 비교할 수도 있다. 예를 들어, 제 1 프로세서는 제 1 및 제 2 출력 값들에 대해 패턴-매칭, 수학적, 또는 다른 비교 동작을 수행하여 등가 (equivalency) 를 검출할 수도 있다. 결정 블록 316 에서, 제 1 프로세서는, 예컨대, 비교 동작들에 기초하여, 제 1 출력 값과 제 2 출력 값이 매칭하는지 여부를 결정할 수도 있다. 제 1 출력 값과 제 2 출력 값이 매칭하지 않는 것으로 결정되면 (즉, 블록 316 = "아니오"), 제 1 프로세서는 방법 (300) 을 종료할 수도 있다. 제 1 출력 값과 제 2 출력 값이 매칭하면 (즉, 결정 블록 316 = "예"), 제 1 프로세서는 블록 318 에서 선택된 TLB 엔트리의 인덱스를 반환할 수도 있다. 예를 들어, 제 1 프로세서는 무효화 또는 플러시 동작들 (또는 플러시하는 동작들) 에서 이용하기 위해 매칭하는 TLB 엔트리의 인덱스를 TLB (예를 들어, CAM) 로 송신할 수도 있다. 반환된 인덱스를 이용하여, 선택적 블록 320 에서, 제 1 프로세서는 선택된 TLB 엔트리를 플러시할 수도 있다. 예를 들어, 제 1 프로세서는 역 TLB 록-업 동작들에 기초하여 선택된 TLB 엔트리와 연계하여 저장된 가상 주소 및 다른 정보를 부정하거나, 제거하거나, 무효화하거나, 그렇지 않으면 무효로 할 수도 있다.

[0040] 도 3b 는 멀티-코어 이중 컴퓨팅 디바이스 내의 프로세서들이 공유 시스템 메모리와 연관된 물리적 주소를 활용하여 역 TLB 록-업 동작들을 수행하는 다른 양상 방법 (350) 을 도시한다. 방법 (350) 은, 방법 (350) 이 멀티-코어 이중 컴퓨팅 디바이스 내의 프로세서들이 시스템 메모리와 관련된 각각의 페이지 프레임 번호 및 TLB 에 나타내어진 가상 주소들과 연관된 각각의 마스크로 출력 값들을 반복적으로 생성하는 것을 가능하게 하는 동작 루프를 가능하게 하는 동작들을 포함한다는 것을 제외하고, 도 3a 를 참조하여 위에서 설명된 방법 (300) 과 유사하다. 다시 말해서, 멀티-코어 이중 컴퓨팅 디바이스는 TLB 엔트리들로부터 마스크들 및 페이지 프레임 번호들을 반복적으로 획득하고, 획득된 마스크들을 획득된 페이지 프레임 번호들 뿐만 아니라 수신된/획득된 물리적 주소 (또는 물리적 주소 입력) 에 반복적으로 적용하고, 결과 출력 값들을 반복적으로 비교하여 어떠한 매치들이 있는지 여부를 결정하도록 구성될 수도 있다. 다양한 양상들에서, 방법 (350) 은, 위에서 설명된 바와 같이, 명령, 소프트웨어 방법, 또는 하드웨어를 통해 멀티-코어 이중 컴퓨팅 디바이스 내의 프로세서들로 구현될 수도 있다. 다시, 설명의 간결함을 위해, 방법 (350) 의 양상 동작들은 멀티-코어 이중 컴퓨팅 디바이스 내의 "제 1 프로세서" 를 참조하여 하기에서 설명된다.

[0041] 블록 302 에서, 멀티-코어 이중 컴퓨팅 디바이스 내의 제 1 프로세서는 공유 메모리에 대응하는 물리적 주소 입력을 획득할 수도 있다. 블록 304' 에서, 제 1 프로세서는 제 1 프로세서와 연관된 TLB 내의 다음 TLB 엔트리를 선택할 수도 있다. 블록 304' 에서의 동작들은 도 3a 를 참조하여 블록 304 에서 위에서 설명된 것과 동일할 수도 있다; 그러나, 제 1 프로세서는 순서대로 TLB 엔트리들을 선택하도록 구성될 수도 있다. 예를

들어, 방법 (350) 이 처음 실행되는 경우 다음 TLB 엔트리는 제 1 TLB 엔트리일 수도 있다. 블록 306 에서, 제 1 프로세서는 선택된 TLB 엔트리로부터 가상 주소와 연관된 마스크를 획득할 수도 있다. 제 1 프로세서는 블록 308 에서 선택된 TLB 엔트리로부터 페이지 프레임 번호를 획득할 수도 있다. 블록 310 에서, 제 1 프로세서는 획득된 마스크를 획득된 페이지 프레임 번호에 적용하여 제 1 출력 값을 생성할 수도 있다. 블록 312 에서, 제 1 프로세서는 획득된 마스크를 획득된 물리적 주소 입력에 적용하여 제 2 출력 값을 생성할 수도 있다.

[0042] 블록 314 에서, 제 1 프로세서는 제 1 출력 값과 제 2 출력 값을 비교할 수도 있다. 결정 블록 316 에서, 제 1 프로세서는, 예컨대, 비교 동작들에 기초하여, 제 1 출력 값과 제 2 출력 값이 매칭하는지 여부를 결정할 수도 있다. 제 1 출력 값과 제 2 출력 값이 매칭하지 않는다고 결정되면 (즉, 결정 블록 316 = "아니오"), 결정 블록 352 에서, 제 1 프로세서는 선택된 TLB 엔트리가 TLB 에서의 마지막 엔트리인지 여부를 결정할 수도 있다. 선택된 TLB 엔트리가 마지막 TLB 엔트리라고 결정되면 (예를 들어, 결정 블록 352 = "예"), 제 1 프로세서는 방법 (350) 을 종료할 수도 있다. 그러나, 선택된 TLB 엔트리가 결정되면 (예를 들어, 결정 블록 352 = "아니오"), 제 1 프로세서는 다음 TLB 엔트리를 선택하고 다양한 출력 값들을 계속 생성하고 평가함으로써 블록 304' 에서 동작들을 계속할 수도 있다. 제 1 출력 값과 제 2 출력 값이 매칭하는 것으로 결정되면 (즉, 결정 블록 316 = "예"), 제 1 프로세서는 블록 318 에서 선택된 TLB 엔트리의 인덱스를 반환할 수도 있다. 반환된 인덱스를 이용하여, 제 1 프로세서는 선택적 블록 320 에서 선택된 TLB 엔트리를 플러시할 수도 있다.

[0043] 멀티-코어 이중 컴퓨팅 디바이스에서의 다양한 프로세서들은 핸드-셰이킹 동작들 또는 프로토콜들을 실행하여 TLB 쉷-다운 메커니즘들을 구현하도록 구성될 수도 있다. 예를 들어, TLB 내에 있는 가상 주소와 관련된 변환 정보가 수정될 필요가 있다고 제 1 프로세서가 결정하는 경우, 제 1 프로세서는 제 1 프로세서의 활성 플래그 (예를 들어, 멀티-코어 이중 컴퓨팅 디바이스의 운영 시스템에 액세스가능한 비트/변수) 를 클리어하고, 가상 주소와 관련된 TLB 엔트리들을 플러시하고, 수행될 TLB 액션들 (예를 들어, TLB 엔트리를 무효화한다) 을 서술하는 메시지를 제 2 프로세서로 전송하고 (또는 인터럽트하고), 제 2 프로세서의 활성 플래그가 클리어될 때까지 비지-대기할 (busy-wait) 수도 있다. 메시지를 수신하는 것에 응답하여, 제 2 프로세서는 제 2 프로세서의 활성 플래그를 클리어하고 시스템 페이지 테이블이 잠길 때까지 비지-대기할 수도 있다 (즉, 제 2 프로세서가 TLB 로컬을 수정하기에 안전할 때까지 대기한다). 제 2 프로세서의 활성 플래그가 클리어되는 경우, 제 1 프로세서는 시스템 메모리의 페이지-테이블을 수정하고, 제 1 프로세서의 활성 플래그를 설정하고, 시스템 페이지 테이블을 잠금해제하고, 제 1 프로세서의 정규 프로세스들을 계속 실행할 수도 있다. 제 2 프로세서는 그 다음에 메시지에 나타내어진 TLB 액션들 (예를 들어, TLB 엔트리들을 무효화한다) 을 실행하고, 제 2 프로세서의 활성 플래그를 설정하고, 제 2 프로세서의 프로세스들의 실행을 계속할 수도 있다. 그러나, 그러한 종래의 핸드셰이킹 메커니즘은 멀티-코어 이중 컴퓨팅 디바이스에서 일관성을 유지하기 위해 프로세서들 사이의 가상 주소 패리티 및/또는 비용이 많이 드는 직렬 동작들에 의존할 수도 있다.

[0044] 도 4 는 제 1 프로세서와 연관된 제 1 TLB 엔트리를 무효화 (또는 플러시) 하기 위해 물리적 주소를 활용하는 멀티-코어 이중 컴퓨팅 디바이스에 대한 일 양상의 방법 (400), 및 제 2 프로세서와 연관된 제 2 TLB 엔트리를 무효화하기 위해 물리적 주소를 활용하는 일 양상의 방법 (450) 을 도시한다. 다시 말해서, 도 4 는 물리적 주소에 매칭하는 물리적 프레임 번호에 대응하는 제 1 및 제 2 프로세서에 대한 TLB 엔트리들을 무효화하기 위한 강화된 TLB 쉷-다운 메커니즘을 가능하게 하도록 동시에 2 개의 프로세서들에 의해 수행될 수도 있는 방법들 (400, 450) 을 도시한다. 다양한 양상들에서, 방법들 (400, 450) 은 소프트웨어, 애플리케이션들, 루틴들, 또는 운영 체제 스레드들을 동시에 실행하는 멀티-코어 이중 컴퓨팅 디바이스의 다양한 프로세서들에 의해 수행될 수도 있다.

[0045] 블록 402 에서, 멀티-코어 이중 컴퓨팅 디바이스 내의 제 1 프로세서는 공유 메모리에 대응하는 TLB 액션들 및 물리적 주소 입력을 수신할 수도 있다. 예를 들어, 멀티-코어 이중 컴퓨팅 디바이스는 제 1 프로세서와 연관된 TLB 의 엔트리를 플러시하기 위한 API 호출을 수신할 수도 있다. TLB 액션 및/또는 물리적 주소 입력이 인터럽트 내에 데이터로서 수신될 수도 있다. 블록 404 에서, 제 1 프로세서는, 예컨대, 제 1 프로세서와 연관된 시스템 비트 또는 변수를 영-출력함으로써 제 1 프로세서와 연관된 활성 플래그를 클리어할 수도 있다. 그러한 활성 플래그들은 멀티-코어 이중 컴퓨팅 디바이스 내의 다른 프로세서들 및/또는 멀티-코어 이중 컴퓨팅 디바이스 상에서 실행하는 운영 체제 루틴들이 액세스가능하거나 그렇지 않으면 볼 수 있을 수도 있다. 제 1 프로세서는 블록 406 에서 TLB 액션들을 나타내는 인터럽트와 같은 메시지 및 수신된 물리적 주소를 제 2 프로세서로 전송할 수도 있다. 블록 407 에서, 제 1 프로세서는 TLB 액션들 및 물리적 주소 입력을

수신하는 것에 응답하여 프로세서간 인터럽트 (inter-processor interrupt) 들 (또는 IPI) 을 디스에이블하도록 구성될 수도 있다. 제 1 프로세서는 그러면, 예컨대, 일 시간의 기간 동안 슬립 또는 대기 모드로 진입함으로써, 블록 408 에서 비지-대기 동작들을 수행할 수도 있고, 결정 블록 410 에서 멀티-코어 이중 컴퓨팅 디바이스 내의 다른 프로세서들의 활성 플래그들이 클리어되었는지 여부를 결정할 수도 있다. 일 양상에서, 제 1 프로세서는 모든 프로세서의 활성 플래그들을 나타내는 저장된 정보를 주기적으로 평가할 수도 있거나, 대안으로 다양한 프로세서들이 다양한 프로세서들의 각기의 활성 플래그들을 클리어하거나 설정하는 경우 신호 또는 메시지를 수신할 수도 있다. 활성 플래그들의 모두가 클리어되지 않았다고 제 1 프로세서가 결정하면 (즉, 결정 블록 410 = "아니오"), 제 1 프로세서는 블록 408 에서 비지-대기 동작들을 계속할 수도 있다.

[0046] 방법 (450) 을 참조하면, 제 1 프로세서에 의한 블록 406 에서의 동작들의 실행에 응답하여, 블록 452 에서, 멀티-코어 이중 컴퓨팅 디바이스 내의 제 2 프로세서는 TLB 액션들 및 물리적 주소를 나타내는 메시지를 제 1 프로세서로부터 수신할 수도 있다. 블록 454 에서, 제 2 프로세서는 수신된 메시지에 기초하여 제 2 프로세서의 활성 플래그를 클리어할 수도 있다. 제 2 프로세서는 그러면, 예컨대, 일 시간의 기간 동안 슬립 또는 대기 모드로 진입함으로써, 블록 456 에서 비지-대기 동작들을 수행할 수도 있고, 결정 블록 458 에서, 시스템 페이지 테이블이 제 1 프로세서에 의해 잠겼는지 여부를 결정할 수도 있다. 일 양상에서, 제 2 프로세서는 시스템 페이지 테이블의 잠긴 상태를 나타내는 저장된 정보를 주기적으로 평가할 수도 있거나, 대안으로 시스템 페이지 테이블이 잠긴 경우, 다중-코어 이중 컴퓨팅 디바이스의 운영 체제를 통한 신호와 같은 신호 또는 메시지를 수신할 수도 있다. 시스템 페이지 테이블이 제 1 프로세서에 의해 잠기지 않았다고 제 2 프로세서가 결정하는 경우 (즉, 결정 블록 458 = "아니오"), 제 2 프로세서는 블록 456 에서 비지-대기 동작들을 계속할 수도 있다.

[0047] 방법 (400) 을 다시 참조하면, 활성 플래그들의 모두가 클리어되었다고 제 1 프로세서가 결정하면 (즉, 결정 블록 410 = "예"), 블록 412 에서, 제 1 프로세서는 공유 메모리와 연관된 페이지 테이블과 같은, 시스템 페이지 테이블을 잠글 수도 있다. 이는 제 1 프로세서가 공유 메모리의 페이지 테이블을 잠그기 위해, 운영 시스템 비트, 변수, 또는 표시자를 설정함으로써, 또는 대안으로 시스템 API 호출을 실행함으로써 달성될 수도 있다.

[0048] 블록 414 에서, 제 1 프로세서는 역 TLB 록-업 동작들을 수행하여 수신된 물리적 주소를 이용해 제 1 프로세서와 연관된 제 1 TLB 의 엔트리들을 식별할 수도 있다. 다양한 양상들에서, 제 1 프로세서는 도 3b 를 참조하여 위에서 설명된 방법 (350) 을 수행함으로써 역 TLB 록-업 동작들을 수행하여, 물리적 주소와 연관된 TLB 엔트리들을 찾기 위해 API 커맨드를 실행하고/하거나, 위에서 설명된 하드웨어-구현 역 TLB 록-업 명령들을 실행할 수도 있다. 예를 들어, 블록 414 의 동작들은 위에서 설명된 도 3b 의 방법 (350) 의 동작들 또는 도 2 를 참조하여 위에서 설명된 바와 같은 역 TLB 록-업 하드웨어의 이용을 포함할 수도 있다.

[0049] 블록 416 에서, 제 1 프로세서는 제 1 TLB 의 식별된 엔트리들에 대해 TLB 액션들을 수행할 수도 있다. 예를 들어, 제 1 프로세서는 물리적 주소에 대응하는 식별된 엔트리들을 플러시하거나 그렇지 않으면 무효화할 수도 있다. 제 1 프로세서는, 예컨대, 공유 메모리에서의 페이지들의 블록 또는 범위를 클리어함으로써, 블록 418 에서 TLB 액션들 및 물리적 주소에 기초하여 시스템 페이지 테이블을 수정할 수도 있다. 블록 420 에서, 제 1 프로세서는 시스템 페이지 테이블을 잠금해제하고 블록 422 에서 제 1 프로세서의 활성 플래그를 설정할 수도 있다. 제 1 프로세서는 그 다음에 블록 424 에서 프로세서간 인터럽트 (IPI) 들을 다시 인에이블하고, TLB 액션들 및 물리적 주소를 수신하기에 앞서 진행 중인 소프트웨어 루틴들을 실행하는 것과 같은 정규 동작들을 계속할 수도 있다. 일 실시형태에서, 제 1 프로세서는 제 2 프로세서가 제 2 프로세서의 활성 플래그를 설정하는 것에 응답하여 블록 464 를 참조하여 하기에서 설명된 동작들에 대해 프로세서간 인터럽트들을 다시 인에이블할 수도 있다.

[0050] 방법 (450) 을 다시 참조하면, 시스템 페이지 테이블이 제 1 프로세서에 의해 잠겼다고 제 2 프로세서가 결정하면 (즉, 결정 블록 458 = "예"), 제 2 프로세서는 역 TLB 록-업 동작들을 수행하여 수신된 물리적 주소를 이용해 제 2 프로세서와 연관된 제 2 TLB 의 엔트리들을 식별할 수도 있다. 블록 460 에서의 동작들은, 블록 460 에서의 동작들이 제 2 프로세서에 의해 수행되어 제 2 프로세서와 연관된 TLB 엔트리들을 식별할 수도 있다는 것을 제외하고, 블록 414 를 참조하여 위에서 설명된 것들과 유사할 수도 있다. 예를 들어, 제 2 프로세서는 위에서 설명된 도 3b 에서의 방법 (350) 의 동작들을 수행할 수도 있다. 블록 462 에서, 제 2 프로세서는, 도 2 를 참조하여 위에서 설명된 바와 같은 역 TLB 록-업 방법, API 커맨드, 또는 하드웨어를 이용하여 식별된 TLB 엔트리들을 플러시하거나 무효화하는 것과 같은, 제 2 TLB 의 식별된 엔트리들에 대한 TLB 액션들을 수행할 수도 있다. 블록 464 에서, 제 2 프로세서는 제 2 프로세서의 활성 플래그를 설정하고, 계류 중이거

나 이전에 착수된 애플리케이션들 또는 루틴들과 같은, 다른 루틴들을 계속 실행할 수도 있다.

[0051]

도 5 는 다양한 양상들에서의 이용에 적합한 멀티-코어 이중 컴퓨팅 디바이스 (500) 의 시스템 블록도이다. 다양한 실시형태들에서, 멀티-코어 이중 컴퓨팅 디바이스 (500) 는 터치스크린 제어기 (504) 및 내부 메모리 (502) 에 커플링된 프로세서 (501) 를 포함할 수도 있다. 프로세서 (501) 는 일반적인 또는 특수한 프로세싱 태스크들을 위해 설계된 하나 이상의 멀티-코어 IC 들일 수도 있다. 일 양상에서, 멀티-코어 이중 컴퓨팅 디바이스 (500) 는 내부 메모리 (502) 에 커플링된 제 2 프로세서 (531) 를 포함할 수도 있다. 일 양상에서, 제 1 프로세서 (501) 는 애플리케이션 프로세서일 수도 있고, 제 2 프로세서 (531) 는 보조 프로세서 (예를 들어, 디지털 신호 프로세서, 그래픽 프로세서 (또는 GPU) 등) 일 수도 있다. 내부 메모리 (502) 는 휘발성 또는 비휘발성 메모리일 수도 있고, 또한 안전하고/하거나 암호화된 메모리, 또는 안전하지 않고/않거나 암호화되지 않은 메모리, 또는 이들의 임의의 조합일 수도 있다. 터치스크린 제어기 (504) 및 프로세서 (501) 는 또한 터치스크린 패널 (512), 예컨대, 저항-감지 터치스크린, 정전용량-감지 터치스크린, 적외선 감지 터치스크린등에 커플링될 수도 있다. 멀티-코어 이중 컴퓨팅 디바이스 (500) 는 서로 및/또는 프로세서 (501) 에 커플링된, 전송하고 수신하기 위한, 하나 이상의 무선 신호 트랜시버들 (508) (예를 들어, Peanut®, Bluetooth®, Zigbee®, Wi-Fi, RF 무선) 및 안테나들 (510) 을 구비할 수도 있다. 트랜시버들 (508) 및 안테나들 (510) 은 위에서 언급된 회로와 함께 이용되어 다양한 무선 송신 프로토콜 스택들 및 인터페이스들을 구현할 수도 있다. 멀티-코어 이중 컴퓨팅 디바이스 (500) 는 셀룰러 네트워크를 통한 통신을 인에이블하고 프로세서에 커플링되는 셀룰러 네트워크 무선 모듈 칩 (516) 을 포함할 수도 있다. 멀티-코어 이중 컴퓨팅 디바이스 (500) 는 프로세서 (501) 에 커플링된 주변 디바이스 접속 인터페이스 (518) 를 포함할 수도 있다. 주변 디바이스 접속 인터페이스 (518) 는 하나의 유형의 접속을 받아들이도록 단독으로 구성될 수도 있거나, USB, Fire Wire, Thunderbolt, 또는 PCIe 와 같은, 일반 또는 사유의, 다양한 유형의 물리적 통신 접속들을 받아들이도록 다중으로 구성될 수도 있다. 주변 디바이스 접속 인터페이스 (518) 는 또한 유사하게 구성된 주변 디바이스 접속 포트 (미도시) 에 커플링될 수도 있다. 멀티-코어 이중 컴퓨팅 디바이스 (500) 는 또한 오디오 출력들을 제공하는 스피커들 (514) 을 포함할 수도 있다. 멀티-코어 이중 컴퓨팅 디바이스 (500) 는 또한 본원에서 논의된 컴포넌트들의 모두 또는 일부를 포함하는, 플라스틱, 금속, 또는 재료들의 조합으로 구성되는, 하우징 (520) 을 포함할 수도 있다. 멀티-코어 이중 컴퓨팅 디바이스 (500) 는 일회용 또는 재충전가능한 배터리와 같은, 프로세서 (501) 에 커플링된 전원 (522) 을 포함할 수도 있다. 재충전가능한 배터리는 또한 멀티-코어 이중 컴퓨팅 디바이스 (500) 의 외부의 소스로부터 충전 전류를 수신하도록 주변 디바이스 접속 포트에 커플링될 수도 있다.

[0052]

프로세서들 (501 및 531) 은 임의의 프로그램가능 마이크로프로세서, 마이크로컴퓨터, 또는 본원에서 설명된 다양한 양상들의 기능들을 포함하여 다양한 기능들을 수행하기 위한 소프트웨어 명령들 (애플리케이션들) 에 의해 구성될 수도 있는 다수의 프로세서 칩 또는 칩들일 수도 있다. 다양한 디바이스들에서, 다수의 프로세서들은, 예컨대 전용된 하나의 프로세서가 무선 통신 기능들을 제공하게 그리고 전용된 하나의 프로세서가 다른 애플리케이션들을 구동시키게 제공될 수도 있다. 통상적으로, 소프트웨어 애플리케이션들은 액세스되어 프로세서 (501 및 531) 로 로딩되기 전에 내부 메모리 (502) 에 저장될 수도 있다. 프로세서 (501 및 531) 는 애플리케이션 소프트웨어 명령들을 저장하기에 충분한 내부 메모리를 포함할 수도 있다. 많은 디바이스들에서, 내부 메모리는 휘발성 메모리, 또는 플래시 메모리와 같은 비휘발성 메모리, 또는 이들 양자 모두의 혼합일 수도 있다. 이러한 설명의 목적으로, 메모리 또는 시스템 메모리에 대한 일반적인 언급은 내부 메모리 또는 다양한 디바이스들 내에 플러깅된 제거가능 메모리, 및 프로세서들 (501 및 531) 내의 메모리를 포함하여, 프로세서들 (501 및 531) 에 의해 액세스가능한 메모리를 지칭한다.

[0053]

앞서 언급한 방법 설명들 및 프로세스 흐름도들은 단지 예시적인 예들로서 제공되고, 다양한 양상들의 단계들이 제시된 순서로 수행되어야 함을 요구하거나 암시하려고 의도되지는 않는다. 당업자에 의해 이해될 바와 같이, 앞서 언급한 양상들에서의 단계들의 순서는 임의의 순서로 수행될 수도 있다. "그 후에", "그리고 나서", "다음으로" 등과 같은 단어들은 단계들의 순서를 제한하고자 의도되지 않으며; 이러한 단어들은 단지 방법들의 설명을 통해 독자를 가이드하는데 이용된다. 또한, 예를 들어, 관사 "하나 (a)", "한 (an)", 또는 "그 (the)" 를 이용하여 단수형으로 요소들을 청구하는 임의의 언급은 단수형으로 요소를 제한하는 것으로 해석되지 않는다.

[0054]

본원에서 개시된 양상들과 연계하여 설명된 다양한 예시적인 논리 블록들, 모듈들, 회로들, 및 알고리즘 단계들은 전자 하드웨어, 컴퓨터 소프트웨어, 또는 이들 양자 모두의 조합들로서 구현될 수도 있다. 하드웨어 및 소프트웨어의 이러한 상호교환성을 명확하게 설명하기 위해, 다양한 예시적인 컴포넌트들, 블록들, 모듈들, 회

로들, 및 단계들은 그들의 기능성의 관점에서 일반적으로 위에서 설명되었다. 그러한 기능이 하드웨어 또는 소프트웨어로 구현되는지 여부는 특정 애플리케이션 및 전체 시스템에 부과되는 설계 제약들에 따라 달라진다.

당업자라면, 상기 상술한 기능성을 각각의 특정 애플리케이션에 대해 다양한 방식으로 구현할 수도 있지만, 이러한 구현 결정은 본 발명의 범위를 벗어나게 하는 것으로 이해되어서는 안된다.

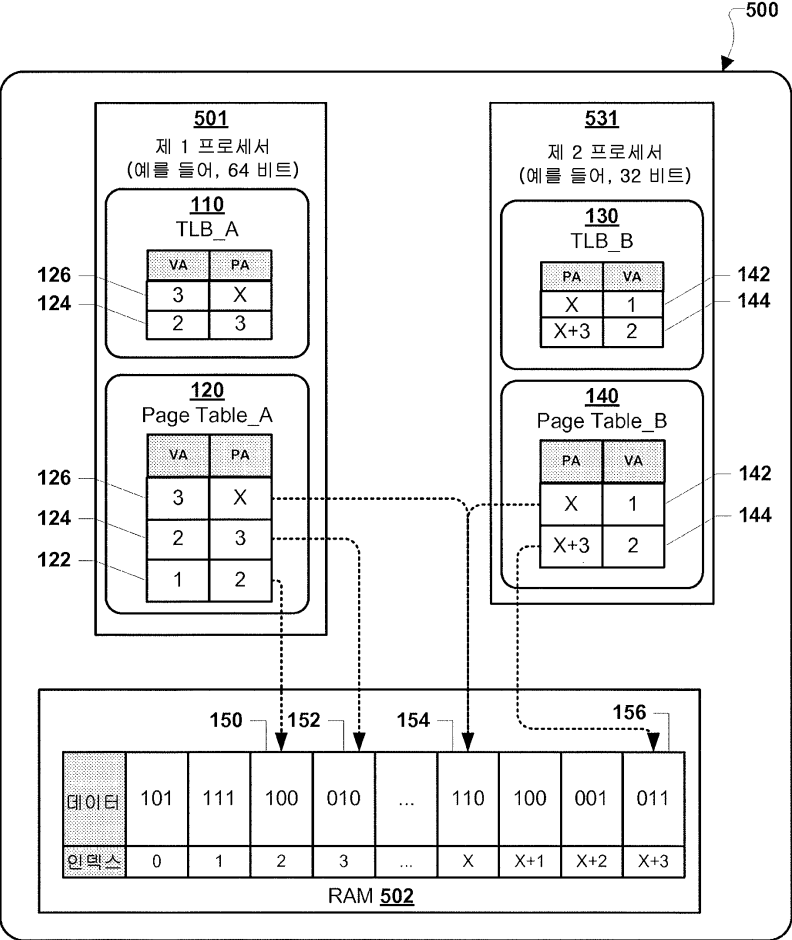
[0055] 본원에서 개시된 실시형태들과 연계하여 설명된 다양한 예시적인 논리 블록들, 모듈들, 및 회로들은 범용 프로세서, 디지털 신호 프로세서 (digital signal processor; DSP), 주문형 반도체 (application specific integrated circuit; ASIC), 필드 프로그래머블 게이트 어레이 (field programmable gate array; FPGA) 또는 다른 프로그래머블 로직 디바이스, 이산 게이트 또는 트랜지스터 로직, 이산 하드웨어 컴포넌트들, 또는 본원에서 개시된 기능들을 수행하도록 설계된 것들의 임의의 조합에 의해 구현되거나 수행될 수도 있다. 범용 프로세서는 마이크로프로세서일 수도 있지만, 대안에서, 프로세서는 임의의 종래의 프로세서, 제어기, 마이크로제어기, 또는 상태 머신일 수도 있다. 프로세서는 또한 컴퓨팅 디바이스들의 조합, 예를 들어, DSP 와 마이크로프로세서의 조합, 복수의 마이크로프로세서들, DSP 코어와 연계한 하나 이상의 마이크로프로세서들, 또는 임의의 다른 그러한 구성으로 구현될 수도 있다. 대안으로, 일부 단계들 또는 방법들은 주어진 기능에 특정한 회로로 수행될 수도 있다.

[0056] 하나 이상의 예시적인 양상들에서, 상술된 기능들은 하드웨어, 소프트웨어, 펌웨어 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현된다면, 기능들은 비일시적 컴퓨터-판독가능 또는 서버-판독가능 매체 또는 비일시적 프로세서-판독가능 저장 매체에 하나 이상의 명령들 또는 코드로 저장되거나 송신될 수도 있다. 본원에서 개시된 방법 또는 알고리즘의 단계들은 유형의 비일시적 컴퓨터-판독가능 저장 매체, 비일시적 서버-판독가능 저장 매체, 및/또는 비일시적 프로세서-판독가능 저장 매체에 있을 수도 있는 프로세서-실행가능 소프트웨어 모듈로 구체화될 수도 있다. 다양한 양상들에서, 그러한 명령들은 저장된 프로세서-실행가능 명령들 또는 저장된 프로세서-실행가능 소프트웨어 명령들일 수도 있다. 유형의 비일시적 컴퓨터-판독가능 저장 매체들은 컴퓨터에 의해 액세스될 수도 있는 임의의 이용가능한 매체들일 수도 있다. 비제한적인 예로서, 이러한 컴퓨터 판독가능 매체는 RAM, ROM, EEPROM, CD-ROM 또는 다른 광학 디스크 스토리지, 자기 디스크 스토리지 또는 다른 자기 스토리지 디바이스들, 또는 요구되는 프로그램 코드를 명령들 또는 데이터 구조들의 형태로 저장하기 위해 사용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 저장 매체를 포함할 수 있다. 본원에서 사용된 디스크 (disk) 와 디스크 (disc) 는, 콤팩트 디스크 (CD), 레이저 디스크, 광학 디스크, 디지털 다기능 디스크 (DVD), 플로피디스크 및 블루레이 디스크를 포함하며, 여기서 디스크 (disk) 는 통상 자기적으로 데이터를 재생하고, 디스크 (disc) 는 레이저를 이용하여 광학적으로 데이터를 재생한다. 위의 조합들도 컴퓨터 판독가능 매체들의 범위 내에 포함되어야 한다. 또한, 방법들 또는 알고리즘의 동작들은 컴퓨터 프로그램 제품에 포함될 수도 있는 유형의 비일시적 프로세서-판독가능 매체 및/또는 컴퓨터-판독가능 매체 상의 코드들 및/또는 명령들의 하나 또는 임의의 조합 또는 세트로서 있을 수도 있다.

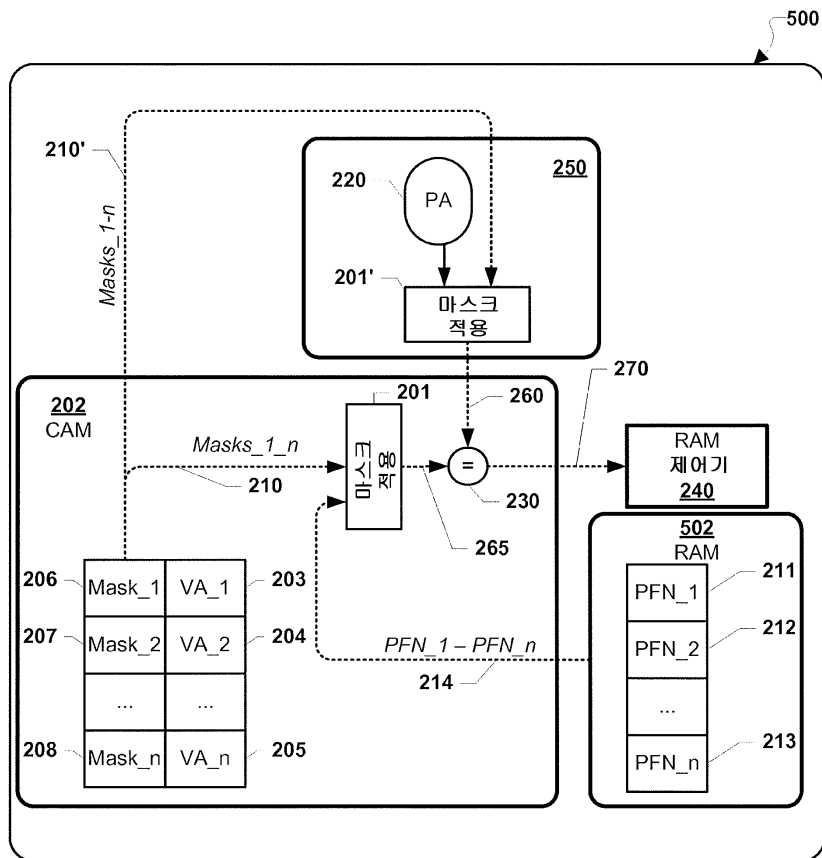
[0057] 개시된 양상들의 상기 설명들은 임의의 당업자가 본 발명을 실시하거나 이용하는 것을 가능하게 하도록 하기 위해 제공된다. 이러한 양상들에 대한 다양한 수정예들이 당업자에게는 자명할 것이고, 본원에서 정의된 일반적인 원칙들은 본 발명의 취지와 범위를 벗어나지 않으면서 다른 양상들에 적용될 수도 있다. 따라서, 본 발명은 본원에서 보여진 양상들로 제한되도록 의도된 것은 아니며, 다음의 청구항들 및 본원에 개시된 원칙들과 신규의 특징들과 일치하는 광의의 범위를 제공하기 위한 것이다.

도면

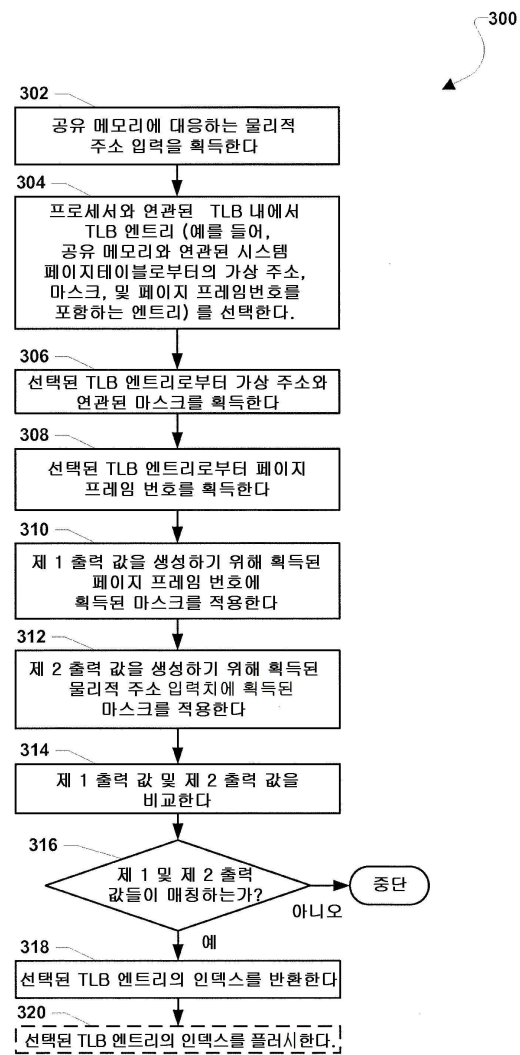
도면1



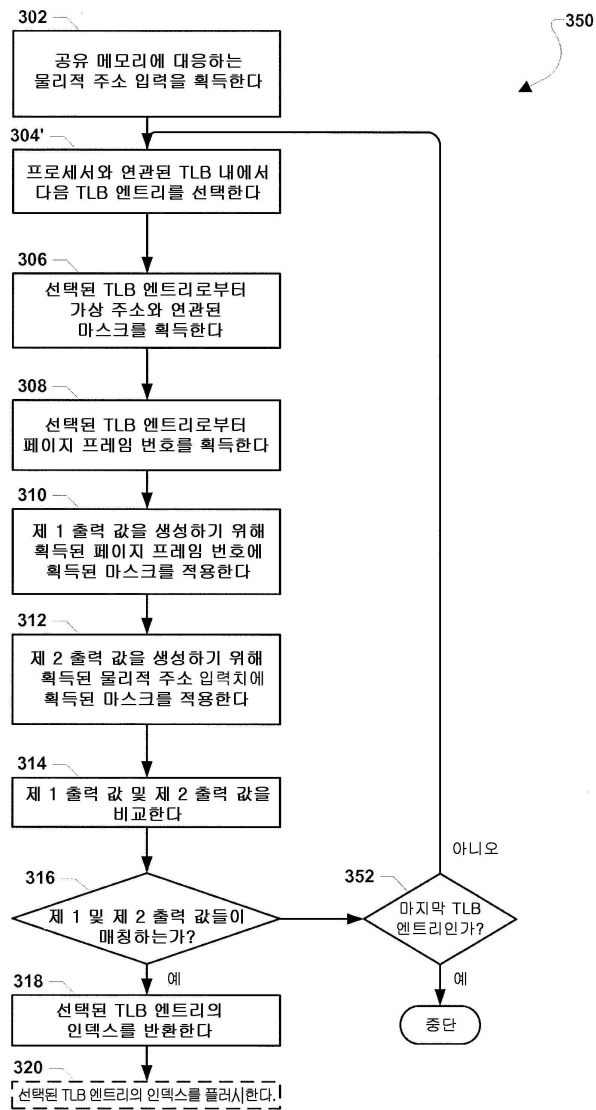
도면2



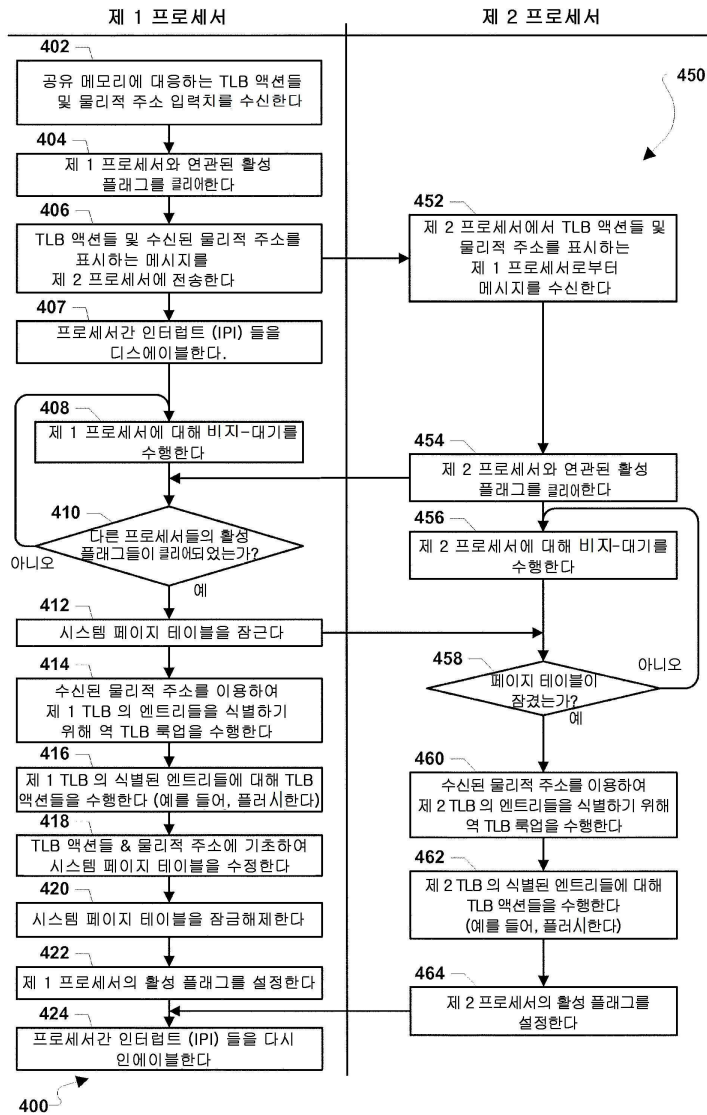
도면3a



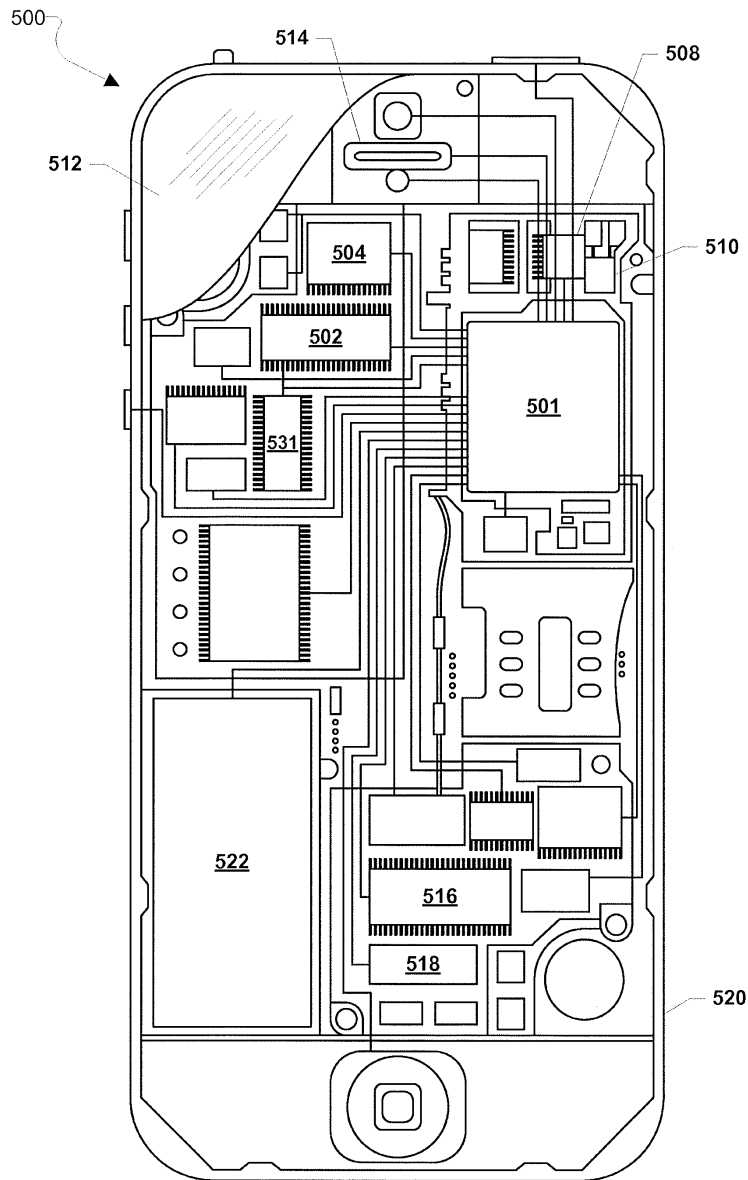
도면3b



도면4



도면5



【심사관 직권보정사항】

【직권보정 1】

【보정항목】 발명(고안)의 설명

【보정세부항목】 식별번호 [0056]

【변경전】

이송 또는 저장하기 위해 사용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다.

【변경후】

저장하기 위해 사용될 수 있고 컴퓨터에 의해 액세스될 수 있는 임의의 다른 저장 매체를 포함할 수 있다.