



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0053966
(43) 공개일자 2016년05월13일

- (51) 국제특허분류(Int. Cl.)
G06F 12/08 (2016.01)
- (52) CPC특허분류
G06F 12/0831 (2013.01)
G06F 12/0824 (2013.01)
- (21) 출원번호 10-2016-7008837
- (22) 출원일자(국제) 2014년08월19일
심사청구일자 없음
- (85) 번역문제출일자 2016년04월04일
- (86) 국제출원번호 PCT/US2014/051712
- (87) 국제공개번호 WO 2015/034667
국제공개일자 2015년03월12일
- (30) 우선권주장
61/875,436 2013년09월09일 미국(US)
14/195,792 2014년03월03일 미국(US)

- (71) 출원인
헬컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
- (72) 발명자
맥도날드, 조셉, 지.
미국 92121 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
가나산, 자야, 프라카쉬, 수브라마니암
미국 92121 캘리포니아주 샌 디에고 모어하우스 드라이브 5775
(뒷면에 계속)
- (74) 대리인
특허법인 남앤드남

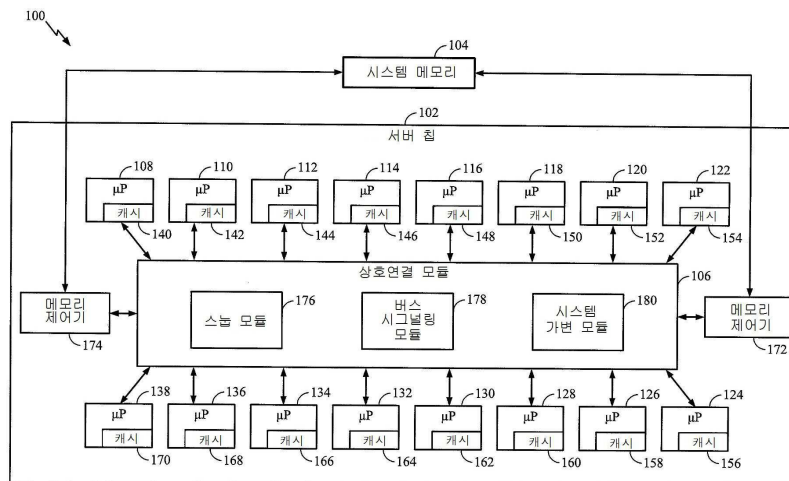
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 **다이렉트 스냅 개재**

(57) 요약

낮은 레이턴시 캐시 개재 메커니즘은 컴퓨터 시스템의 멀티프로세서 아키텍처에서 캐시 "히트"에 대한 개재자 캐시를 동적으로 선택하도록 스냅 필터를 구현한다. 개재자의 선택은 컴퓨터 시스템의 레이턴시, 토폴로지, 주파수, 활용, 로드, 웨어 밸런스 및/또는 전력 상태와 같은 변수들에 기초한다.

대표도 - 도1



(52) CPC특허분류

G06F 2212/1024 (2013.01)

G06F 2212/1041 (2013.01)

(72) 발명자

스피어, 토마스, 필립

미국 92121 캘리포니아주 샌 디에고 모어하우스 드
라이브 5775

로빈슨, 에릭, 에프.

미국 92121 캘리포니아주 샌 디에고 모어하우스 드
라이브 5775

파나비치, 제이슨, 로렌스

미국 92121 캘리포니아주 샌 디에고 모어하우스 드
라이브 5775

트루옹, 투옹, 쉐.

미국 92121 캘리포니아주 샌 디에고 모어하우스 드
라이브 5775

명세서

청구범위

청구항 1

방법으로서,

요청된 캐시 라인을 판독하기 위한 요청을 컴퓨터 시스템 내의 요청 프로세서로부터 획득하는 단계;

하나 또는 그 초과와 소유 프로세서들과 연관된 하나 또는 그 초과와 캐시들이 상기 요청된 캐시 라인을 포함함을 결정하는 단계;

상기 요청된 캐시 라인을 상기 요청 프로세서에 제공하기 위해 상기 하나 또는 그 초과와 소유 프로세서들 중 소유 프로세서를 선택하는 단계 - 상기 소유 프로세서를 선택하는 단계는 하나 또는 그 초과와 변수들에 기초함 - ; 및

상기 요청된 캐시 라인을 상기 요청 프로세서에 제공하도록, 선택된 소유 프로세서에 통지하는 단계를 포함하는,

방법.

청구항 2

제 1 항에 있어서,

상기 하나 또는 그 초과와 소유 프로세서들과 연관된 캐시 라인들에 대한 엔트리들의 디렉토리를 유지하는 단계를 더 포함하는,

방법.

청구항 3

제 1 항에 있어서,

하나의 소유 프로세서와 연관된 캐시를 선택하는 단계는 하나의 소유 프로세서와 연관된 변수를 적어도 하나의 다른 소유 프로세서와 연관된 변수와 비교하는 단계를 포함하는,

방법.

청구항 4

제 3 항에 있어서,

상기 하나의 소유 프로세서와 연관된 변수를 적어도 하나의 다른 소유 프로세서와 연관된 변수와 비교하는 단계는 동등한 변수를 비교하는 단계를 포함하는,

방법.

청구항 5

제 1 항에 있어서,

상기 하나 또는 그 초과와 변수들은 상기 컴퓨터 시스템에 대한 토폴로지를 포함하는,

방법.

청구항 6

제 1 항에 있어서,

상기 하나 또는 그 초과와 변수들은 상기 컴퓨터 시스템에 대한 전력 상태를 포함하는,

방법.

청구항 7

제 1 항에 있어서,

상기 하나 또는 그 초과 변수들은 상기 컴퓨터 시스템에 대한 주파수를 포함하는,

방법.

청구항 8

제 1 항에 있어서,

상기 하나 또는 그 초과 변수들은 상기 컴퓨터 시스템에 대한 레이턴시를 포함하는,

방법.

청구항 9

제 1 항에 있어서,

상기 하나 또는 그 초과 변수들은 상기 컴퓨터 시스템에 대한 활용을 포함하는,

방법.

청구항 10

제 1 항에 있어서,

상기 하나 또는 그 초과 변수들은 상기 컴퓨터 시스템에 대한 웨어 밸런스(wear balance)를 포함하는,

방법.

청구항 11

제 1 항에 있어서,

상기 하나 또는 그 초과 변수들은 상기 컴퓨터 시스템에 대한 로드(load)를 포함하는,

방법.

청구항 12

컴퓨터 시스템에서 캐시 개재(intervention)를 수행하기 위한 장치로서, 상기 컴퓨터 시스템은, 다수의 프로세서들 및 연관 캐시들 - 상기 연관 캐시들은 하나 또는 그 초과 캐시 라인들을 포함함 - 을 가지며,

상기 장치는,

요청된 캐시 라인을 판독하기 위한 요청을 요청 프로세서로부터 획득하고; 그리고

하나 또는 그 초과 소유 프로세서들과 연관된 하나 또는 그 초과 캐시들이 상기 요청된 캐시 라인을 포함함을 결정하도록 구성되는,

스냅 모듈;

상기 컴퓨터 시스템과 연관된 하나 또는 그 초과 변수들을 추적하도록 구성되는 가변 모듈 - 상기 스냅 모듈은 상기 하나 또는 그 초과 변수들에 기초하여 상기 요청된 캐시 라인을 상기 요청 프로세서에 제공하기 위해 소유 프로세서를 선택하도록 추가로 구성됨 - ; 및

상기 요청된 캐시 라인을 상기 요청 프로세서에 제공하도록, 선택된 소유 프로세서에 시그널링하도록 구성되는 시그널링 모듈을 포함하는,

컴퓨터 시스템에서 캐시 개재를 수행하기 위한 장치.

청구항 13

제 12 항에 있어서,

상기 하나 또는 그 초과 변수들은 상기 컴퓨터 시스템에 대한 토폴로지를 포함하는,
컴퓨터 시스템에서 캐시 개재를 수행하기 위한 장치.

청구항 14

제 12 항에 있어서,

상기 다수의 프로세서들 및 연관 캐시들과 연관된 하나 또는 그 초과 변수들은 상기 컴퓨터 시스템에 대한 전력 상태를 포함하는,
컴퓨터 시스템에서 캐시 개재를 수행하기 위한 장치.

청구항 15

제 12 항에 있어서,

상기 하나 또는 그 초과 변수들은 상기 컴퓨터 시스템에 대한 주파수를 포함하는,
컴퓨터 시스템에서 캐시 개재를 수행하기 위한 장치.

청구항 16

제 12 항에 있어서,

상기 하나 또는 그 초과 변수들은 멀티프로세서 아키텍처에 대한 레이턴시를 포함하는,
컴퓨터 시스템에서 캐시 개재를 수행하기 위한 장치.

청구항 17

제 12 항에 있어서,

상기 하나 또는 그 초과 변수들은 상기 컴퓨터 시스템에 대한 활용을 포함하는,
컴퓨터 시스템에서 캐시 개재를 수행하기 위한 장치.

청구항 18

제 12 항에 있어서,

상기 하나 또는 그 초과 변수들은 상기 컴퓨터 시스템에 대한 웨어 밸런싱(wear balancing)을 포함하는,
컴퓨터 시스템에서 캐시 개재를 수행하기 위한 장치.

청구항 19

제 12 항에 있어서,

상기 하나 또는 그 초과 변수들은 상기 컴퓨터 시스템에 대한 로드를 포함하는,
컴퓨터 시스템에서 캐시 개재를 수행하기 위한 장치.

청구항 20

비-일시적 컴퓨터 판독가능한 저장 매체로서, 기계에 의해 액세스될 때, 상기 기계로 하여금 동작들을 수행하게 하는 데이터를 포함하며,

상기 동작들은,

요청된 캐시 라인을 판독하기 위한 요청을 컴퓨터 시스템 내의 요청 프로세서로부터 획득하는 것;

하나 또는 그 초과 소유 프로세서들과 연관된 하나 또는 그 초과 캐시들이 상기 요청된 캐시 라인을 포함함을 결정하는 것;

상기 요청된 캐시 라인을 상기 요청 프로세서에 제공하기 위해 상기 하나 또는 그 초과와 소유 프로세서들 중 소유 프로세서를 선택하는 것 - 상기 소유 프로세서를 선택하는 것은 하나 또는 그 초과와 변수들에 기초함 - ; 및

상기 요청된 캐시 라인을 상기 요청 프로세서에 제공하도록, 선택된 소유 프로세서에 통지하는 것을 포함하는, 비-일시적 컴퓨터 판독가능한 저장 매체.

발명의 설명

기술 분야

[0001] 본 개시의 양상들은 일반적으로 프로세서들에 관한 것으로, 보다 특히, 멀티프로세서들에서의 다이렉트 스누프 개재(direct snoop intervention)에 관한 것이다.

배경 기술

[0002] 전형적인 종래의 멀티프로세서 집적 회로(즉, 칩)는 상호연결 버스를 사용하여 상호연결된 다수의 프로세서 코어들을 활용한다. 각각의 프로세서 코어는 하나 또는 그 초과와 캐시들에 의해 지원된다. 각각의 캐시는 데이터 파일들을 저장하며, 데이터 파일들은 전형적으로 고정 크기의 블록들로 시스템 메모리와 캐시들 사이에서 전달된다. 데이터의 블록들은 "캐시 라인들"이라 칭해진다. 각각의 캐시는 그가 캐싱한 데이터 파일들과 연관된 어드레스들 전부의 디렉토리를 포함한다.

[0003] 각각의 프로세서 코어의 캐싱된 데이터는 상호연결 버스 상에서 모든 다른 프로세서 코어들에 의해 공유될 수 있다. 따라서, 시스템에서 데이터의 많은 카피들: 온-칩 또는 오프-칩일 수 있는 메인 메모리 내의 하나의 카피, 및 각각의 프로세서 코어 캐시 내의 하나의 카피를 갖는 것이 가능하다. 더욱이, 각각의 프로세서 코어는 그것의 캐시 내에 있는 데이터를 상호연결 버스 상에서 임의의 다른 프로세서 코어와 공유할 수 있다. 따라서, 공유되는 데이터와의 일치성 또는 일관성을 유지하기 위한 요건이 존재한다. 상호연결 버스는 일관성이 유지됨을 보장하기 위해 다양한 프로세서 코어들 및 캐시들 사이의 모든 일관성 트래픽을 핸들링한다.

[0004] 멀티프로세서에서 일관성을 유지하기 위한 하나의 메커니즘은 "스누핑(snooping)"이라 칭해지는 것을 활용한다. 프로세서 코어가 특정 캐시 라인을 필요로 하는 경우, 프로세서 코어는 먼저, 자기 자신의 캐시를 조사한다. 프로세서 코어가 자기 자신의 캐시 내의 캐시 라인을 발견하면, 캐시 "히트"가 발생한다. 그러나, 프로세서 코어가 자기 자신의 캐시 내의 캐시 라인을 발견하지 못하면, 캐시 "미스"가 발생한다. 캐시 "미스"가 발생하는 경우, 다른 프로세서들의 캐시들은 다른 캐시들 중 임의의 것이, 요청된 캐시 라인을 갖는지 여부를 결정하도록 스누핑된다. 요청된 데이터가 또 다른 프로세서 코어의 캐시에 로케이팅되면, 다른 프로세서 코어의 캐시는 요청 프로세서 코어가 메인 메모리로부터의 데이터에 액세스할 필요가 없도록 캐시 라인을 요청 프로세서 코어에 제공하기 위해 캐시 라인을 "개재"할 수 있다.

[0005] 상호연결 버스 상에 단지 2개의 프로세서 코어들 및 연관 캐시들이 존재하는 경우, 이러한 스누핑 기법이 잘 동작한다. 예를 들어, 제 1 프로세서 코어가 캐시 라인을 요청하고, 제 2 프로세서 코어의 캐시가 요청된 캐시 라인을 포함하면, 제 2 프로세서 코어의 캐시는 요청된 캐시 라인을 제 1 프로세서 코어에 제공할 것이다. 제 2 프로세서 코어의 캐시가 요청된 캐시 라인을 포함하지 않으면, 제 1 프로세서 코어의 캐시는 오프-칩 메인 메모리로부터 요청된 캐시 라인에 액세스할 것이다. 그러나, 상호연결 버스가 더 많은 프로세서 코어들을 지원하고, 이들 중 임의의 것이 자신의 캐시 내에 요청된 데이터를 가질 수 있음에 따라, 어떤 프로세서 코어의 캐시가 요청된 캐시 라인을 요청 프로세서 코어에 제공할 것인지를 판정하기 위해 더 복잡한 중재 메커니즘이 필요하다.

[0006] 상호연결 버스에 의해 지원되는 2개 초과와 프로세서 코어들 및 연관 캐시들이 존재하는 경우에 대한 하나의 중재 메커니즘은, 스누프 요청 시 데이터의 제공에 대한 책임(responsibility)을 표시하는 캐시 내의 저장 상태 정보(즉, "개재자(intervener)" 내의 저장 상태 정보)를 포함한다. 프로세서 코어가 캐시 라인을 요청하는 경우, 상호연결 버스는 (예를 들어, 상호연결 버스 상의 모든 프로세서 캐시들에 스누프 요청을 브로드캐스팅함으로써) 모든 연결된 캐시들을 "스누핑"한다. 상호연결 버스에 의해 지원되는 각각의 프로세서 코어는, 개재자가 요청된 캐시 라인을 요청 프로세서 코어에 제공할 것으로 마킹된 캐시 및 자신의 캐시 라인들을 체크한다.

[0007] [0007] 더 복잡한 상호연결 버스들은 상호연결 버스 상의 모든 프로세서 코어 캐시들에 의해 소유되는 캐시 라인들을 표현하는 엔트리들을 유지하는 스누프 필터를 구현한다. 상호연결 버스 상의 모든 프로세서 캐시들에 스누프 요청을 브로드캐스팅하는 대신에, 스누프 필터는 가능하게는 데이터의 카피를 가질 수 있는 프로세서 캐시들만을 스누프링하도록 상호연결 버스에 지시한다.

[0008] [0008] 이력적으로, 개재 캐시를 결정하기 위한 판정-수행 프로세스가 고정 방식에 기초하여 수행된다. 예를 들어, 개재 캐시는 캐시 라인을 요청한 마지막 프로세서 코어 또는 캐시 라인을 요청한 제 1 프로세서 코어에 기초하여 결정된다. 불행하게도, 제 1 프로세서 코어 또는 마지막 프로세서 코어는 캐시 라인을 제공할 가장 최적의 프로세서 코어가 아닐 수 있다.

[0009] [0009] 따라서, 상호연결 버스를 중재하기 위한 개선된 장치들 및 방법들이 필요하다.

발명의 내용

[0010] [0010] 본 발명의 예시적 구현들은 멀티프로세서 아키텍처에서 상호연결 모듈 버스에 걸친 다이렉트 스누프 개재 (directed snoop intervention)를 위한 장치들, 방법들, 시스템들 및 비-일시적 기계 판독가능한 매체들에 관련된다. 하나 또는 그 초과와 구현들은 멀티프로세서 아키텍처에서 캐시 "히트"에 대한 개재자 캐시를 동적으로 선택하도록 스누프 필터를 구현하는 낮은 레이턴시 캐시 개재 메커니즘을 포함한다.

[0011] [0011] 메커니즘은, 요청된 캐시 라인을 판독하기 위한 요청을 요청 프로세서로부터 획득하고, 그리고 하나 또는 그 초과와 소유 프로세서들과 연관된 하나 또는 그 초과와 캐시들이 요청된 캐시 라인을 포함함을 결정하도록 구성되는 스누프 모듈을 포함하는 장치를 포함한다. 장치는 컴퓨터 시스템과 연관된 하나 또는 그 초과와 변수들을 추적하도록 구성되는 가변 모듈을 더 포함한다. 스누프 모듈은 하나 또는 그 초과와 변수들에 기초하여, 요청된 캐시 라인을 요청 프로세서에 제공하기 위해 소유 프로세서를 선택하도록 추가로 구성된다. 장치는 요청된 캐시 라인을 요청 프로세서에 제공하도록, 선택된 소유 프로세서에 시그널링하도록 구성되는 시그널링 모듈을 더 포함한다.

[0012] [0012] 메커니즘은, 요청된 캐시 라인을 판독하기 위한 요청을 컴퓨터 시스템 내의 요청 프로세서로부터 획득하는 단계, 하나 또는 그 초과와 소유 프로세서들과 연관된 하나 또는 그 초과와 캐시들이 요청된 캐시 라인을 포함함을 결정하는 단계, 요청된 캐시 라인을 요청 프로세서에 제공하기 위해 하나 또는 그 초과와 소유 프로세서들 중 소유 프로세서를 선택하는 단계 - 소유 프로세서를 선택하는 단계는 하나 또는 그 초과와 변수들에 기초함 -, 및 요청된 캐시 라인을 요청 프로세서에 제공하도록, 선택된 소유 프로세서에 통지하는 단계를 포함하는 방법을 수행한다. 비-일시적 컴퓨터 프로그램 물건은 이 방법 및 본원에 설명되는 다른 방법들을 구현할 수 있다.

[0013] [0013] 본 발명의 내용은, 그것이 청구항들의 범위 또는 의미를 해석 또는 제한하는데 이용되지 않을 것이라는 점을 포함해서 제시된다. 본 발명의 내용은 청구되는 대상의 핵심 특징들 또는 본질적 특징들을 식별하는 것으로도, 청구되는 대상의 범위의 결정을 돕는 것으로서 사용되는 것으로도 의도되지 않는다.

도면의 간단한 설명

[0014] [0014] 첨부한 도면들은 본원에 설명되는 기술의 구현들의 설명을 돕기 위해 제시되며, 구현들의 제한이 아닌 단지 구현들의 예시를 위해 제공된다.

[0015] 도 1은 하나 또는 그 초과와 구현들에 따른, 멀티프로세서 아키텍처에서 상호연결 모듈 버스에 걸친 다이렉트 스누프 개재를 구현하는데 적합한 예시적 환경의 블록도이다.

[0016] 도 2는 하나 또는 그 초과와 구현들에 따른, 캐시 "미스"에 대한 응답으로 다이렉트 스누프 개재를 예시하는 블록도이다.

[0017] 도 3은 하나 또는 그 초과와 구현들에 따른 컴퓨터 시스템을 예시하는 블록도이다.

[0018] 도 4는 하나 또는 그 초과와 구현들에 따른, 멀티프로세서 아키텍처에서 상호연결 모듈 버스에 걸친 다이렉트 스누프 개재를 구현하기 위한 방법의 예시적 흐름도이다.

[0019] 상세한 설명은 첨부한 도면들을 참조한다. 도면들에서, 참조 번호의 가장-왼쪽 숫자(들)는 참조 번호가 처음 나타나는 도면을 식별한다. 동일한 번호들은 동일한 특징들 및 컴포넌트들을 참조하기 위해 도면들 전반에 걸쳐 사용된다.

발명을 실시하기 위한 구체적인 내용

- [0015] [0020] 일반적으로, 본원에 개시되는 대상은 멀티프로세서 아키텍처에서 낮은 레이턴시 캐시 개재 메커니즘을 위한 시스템들, 장치들, 비-일시적 컴퓨터 판독가능한 매체들 및 방법들에 관한 것이다. 하나 또는 그 초과 of 구현들에서, 상호연결 모듈은 멀티프로세서 아키텍처에서 캐시 라인들의 위치를 추적한다.
- [0016] [0021] 상호연결 모듈에 의해 지원되는 프로세서가 하나 또는 그 초과 of 다른 프로세서 코어 캐시들에서 캐시 라인을 판독하기 위한 요청을 발행하는 경우, 상호연결 모듈은 어떤 캐시들이 요청된 캐시 라인을 포함하거나 또는 소유하는지를 결정한다. 상호연결 모듈은 요청된 캐시 라인을 포함하는 프로세서 코어 캐시들과 연관된 변수들을 비교한다. 그 다음, 상호연결 모듈은, 변수들을 비교함으로써 결정되는, 최저 레이턴시, 최저 전력, 최고 속도 등을 표현하는 요청된 캐시 라인을 포함하는 캐시를 선택한다. 선택된 캐시는 (즉, 요청된 데이터를 제공하기 위한) 요청 프로세서 코어에 대한 개재자가 된다.
- [0017] [0022] 그 다음, 상호연결 모듈은 요청된 캐시 라인을 요청 프로세서에 제공하도록, 선택된 개재자 캐시에게 통지한다. 그 다음, 선택된 개재자 캐시는 요청된 캐시 라인을 요청 프로세서 코어에 제공한다. 따라서, 요청된 캐시 라인을 제공하도록 어떤 캐시를 개재할 것인지를 결정하기 위한 고정 방식을 갖기 보다는, 상호연결 모듈은 변화하는 시스템 변수들에 기초하여 개재 캐시를 동적으로 선택한다.
- [0018] [0023] 어떤 캐시가 개재자일 것인지를 결정하기 위해 최소 하나의 시스템 변수가 고려될 수 있다는 점이 주목될 것이다. 하나 초과 of 시스템 변수의 고려는 요구되지 않는다.
- [0019] [0024] 상호연결 모듈에 의해 고려될 수 있는 하나의 시스템 변수는 멀티프로세서 아키텍처의 토폴로지를 포함할 수 있다. 토폴로지 변수는 캐시 라인이 온-칩인지 여부, 캐시 라인이 오프-칩인지 여부, 캐시 라인이 메인 메모리에 있는지 여부, 캐시 라인이 또 다른 멀티프로세서 칩 상에 있는지 여부 등을 고려할 수 있다.
- [0020] [0025] 상호연결 모듈에 의해 고려될 수 있는 또 다른 시스템 변수는 프로세서 코어 및/또는 캐시의 전력 상태를 포함할 수 있다. 예를 들어, 상호연결 모듈은 코어/캐시가 동작 모드에 있는지 아니면 전력 절약 모드에 있는지를 고려할 수 있다. 모드들은 "슬립" 모드, "전력 붕괴" 모드, "유휴" 모드 등을 포함할 수 있다.
- [0021] [0026] 상호연결 모듈에 의해 고려될 수 있는 또 다른 시스템 변수는 프로세서 코어의 주파수 및/또는 캐시의 주파수를 포함할 수 있다.
- [0022] [0027] 상호연결 모듈에 의해 고려될 수 있는 또 다른 시스템 변수는 이중 시스템에서의 레이턴시를 포함할 수 있다. 예를 들어, 상호연결 모듈은 하나 또는 그 초과 of GPU(Graphic Processing Unit), 하나 또는 그 초과 of DSP(digital signal processors) 및/또는 32 비트 및 64 비트의 혼합 범용 마이크로프로세서 코어들과 같은 상이한 아키텍처들을 갖는 프로세서 코어들을 지원할 수 있다. 이러한 시나리오에서, 상호연결 모듈은 개별 프로세서 코어들 또는 프로세서 코어들의 결합의 레이턴시를 고려할 수 있다.
- [0023] [0028] 상호연결 모듈에 의해 고려될 수 있는 또 다른 시스템 변수는 프로세서 코어 및/또는 캐시의 현재 활용을 포함할 수 있다. 예를 들어, 상호연결 모듈은 명령들을 프로세싱하기 위해 프로세서 코어 및/또는 캐시가 사용하는 시간의 양을 고려할 수 있다.
- [0024] [0029] 상호연결 모듈에 의해 고려될 수 있는 또 다른 시스템 변수는 요청된 캐시 라인을 제공할 소유 프로세서 코어 및/또는 캐시를 선택하기 전에, 마이크로프로세서 아키텍처에서 상호연결 모듈 세그먼트들의 현재 활용을 포함할 수 있다.
- [0025] [0030] 상호연결 모듈에 의해 고려될 수 있는 또 다른 시스템 변수는 프로세서 코어 및/또는 캐시 요청들의 웨어 밸런싱(wear balancing) 등을 포함할 수 있다. 예를 들어, 특정 반도체 기술들(예를 들어, FinFET와 같은 멀티-게이트 디바이스들)은, 회로의 실패율이 회로가 얼마나 빈번하게 "사용", 즉, 스위칭 온 및 스위치 오프되는지와 관련된다는 특성을 갖는다. 하나 또는 그 초과 of 구현들에서, 상호연결 모듈은 반도체(들)의 수명을 최대화하기 위해 "동등한 경로들" 사이에서 균등하게 작업을 분배하려고 시도하는 것에 기초하여 개재자가 될 캐시를 선택할 수 있다.
- [0026] [0031] 물론, 상호연결 모듈이 고려할 수 있는 시스템 변수들의 이러한 리스트는 전면적(exhaustive)이지 않고, 훨씬 더 많은 시스템 변수들이 고려될 수 있다. 예를 들어, 시스템 변수들의 리스트는 효율성 인자들을 포함할 수 있다.
- [0027] [0032] 예시하기 위해, 하나의 프로세서 코어의 캐시가 자기 자신의 동작들에 의해 과도하게(heavily)

로딩되고, 또 다른 프로세서 코어의 캐시가 유티 상태이면, 유티 프로세서 코어의 캐시가 요청 프로세서 코어로부터 더 멀리 떨어져 있음에도 불구하고 요청 프로세서 코어로부터 더 멀리 떨어져 있는 유티 프로세서 코어의 캐시로부터 요청된 캐시 라인을 획득하는 것이 더 효율적일 수 있다. 대안적으로, 2개의 상이한 타입들의 프로세서 코어들이 존재하고, 하나의 타입의 프로세서 코어가 다른 타입의 프로세서 코어보다 고유하게 더 긴 레이턴시를 포함한다면, 이러한 변수도 역시 상호연결 모듈에 의해 고려될 수 있다. 본원에서의 개시를 읽은 이후, 가장 효율적 개재자의 선택을 결정하기 위해 멀티프로세서 아키텍처에 대한 더 정교한(sophisticated) 또는 덜 정교한 낮은 레이턴시 캐시 개재 메커니즘들을 설계하는 방법이 명백해질 것이다.

- [0028] [0033] 도 1은 본원에 설명되는 하나 또는 그 초과와 구현들에 따른, 상호연결 버스가 요청된 캐시 라인을 요청 프로세서 코어에 제공할 개재자 캐시를 결정하는 아키텍처(100)의 하이-레벨 블록도를 예시한다. 예시되는 아키텍처(100)는 칩(102)을 포함한다.
- [0029] [0034] "서버" 칩으로서 도시되지만, 칩(102)이 그렇게 제한되는 것은 아니다. 예를 들어, 칩(102)은 다수의 프로세서 코어들을 지원할 수 있는 임의의 적합한 집적 회로일 수 있다.
- [0030] [0035] 예시되는 아키텍처(100)는 시스템 메모리(104)를 포함한다. 하나 또는 그 초과와 구현들에서, 시스템 메모리(104)는 RAM(random access memory), 이블테면, DRAM(dynamic RAM) 및/또는 이의 변형들을 포함할 수 있다. 예시되는 바와 같이, 시스템 메모리(104)는 칩(102)의 외부에 로케이팅되거나, 또는 칩(102)으로부터의 오프-칩이다.
- [0031] [0036] 예시되는 아키텍처(100)는 상호연결 모듈(106)을 포함한다. 하나 또는 그 초과와 구현들에서, 상호연결 모듈(106)은 환경(100)에서 컴포넌트들 사이의 데이터 전달들을 관리한다.
- [0032] [0037] 예시되는 상호연결 모듈(106)은 다수의 프로세서 코어들, 이블테면, 프로세서 코어들(108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136 및 138)을 지원한다. 각각의 프로세서 코어(108, 110, 112, 114, 116, 118, 120, 122, 124, 126, 128, 130, 132, 134, 136 및 138)는 하나 또는 그 초과와 연관 캐시들(140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 166, 168 및 170)을 포함한다. 캐시들은 전형적으로, 시스템 메모리(104)에 또한 저장된 데이터 파일들의 카피들을 저장하는 소형의 빠른 메모리 디바이스들이다. 캐시들은 또한, 데이터 파일들을 서로 공유할 수 있다.
- [0033] [0038] 예시되는 아키텍처(100)는 메모리 제어기(172) 및 메모리 제어기(174)를 포함한다. 메모리 제어기들(172 및 174)은 시스템 메모리(104)로부터의 그리고 시스템 메모리(104)로의 데이터의 흐름을 관리한다. 예시되는 구현에서, 메모리 제어기들(172 및 174)은 칩(102) 상에 통합된다. 그러나, 메모리 제어기들(172 및 174)은 개별 칩들일 수 있거나, 하나 또는 그 초과와 다른 칩들로 통합될 수 있다.
- [0034] [0039] 예시되는 상호연결 모듈(106)은 스눅 모듈(176)을 포함한다. 하나 또는 그 초과와 구현들에서, 스눅 모듈(176)은 요청된 캐시 라인을 관독하기 위한 요청을 요청 프로세서로부터 획득한다. 스눅 모듈(176)은 하나 또는 그 초과와 소유 프로세서들과 연관된 하나 또는 그 초과와 캐시들이 요청된 캐시 라인을 포함하는지 여부를 결정한다. 스눅 모듈(176)은 멀티프로세서 아키텍처(100)에서 캐시 파일들의 위치를 추적하고, 각각의 캐시에 저장되는 캐시 라인들을 표현하는 엔트리들을 유지함으로써 이것을 달성할 수 있다. 스눅 모듈(176)은 하나 또는 그 초과와 변수들에 기초하여, 요청된 캐시 라인을 요청 프로세서 코어에 제공하기 위해 소유 프로세서를 선택할 수 있다.
- [0035] [0040] 예시되는 상호연결 모듈(106)은 또한 버스 시그널링 모듈(178)을 포함한다. 하나 또는 그 초과와 구현들에서, 버스 시그널링 모듈(178)은 요청된 캐시 라인을 요청 프로세서에 제공하도록, 선택된 프로세서 코어의 캐시에 통지하는 하나 또는 그 초과와 신호들을 포함한다. 즉, 버스 시그널링 모듈(178)은 요청된 캐시 라인을 요청 프로세서 코어에 제공하도록, 선택된 소유 프로세서 코어에 시그널링한다.
- [0036] [0041] 예시되는 상호연결 모듈(106)은 또한, 시스템 가변 모듈(180)을 포함한다. 예시되는 시스템 가변 모듈(180)은 멀티프로세서 아키텍처(100)인 컴퓨터 시스템과 연관된 하나 또는 그 초과와 변수들을 추적할 수 있다. 시스템 가변 모듈(180)은 프로세서 코어들 및 이들의 캐시들과 연관된 변수들을 포함한다.
- [0037] [0042] 시스템 변수들은 이블테면, 캐시 라인이 온-칩이든, 오프-칩이든(예를 들어, 시스템 메모리 내에 있든, 또 다른 멀티프로세서 칩 상에 있든 등) 간에 멀티프로세서 아키텍처의 토폴로지를 포함할 수 있다.
- [0038] [0043] 시스템 변수들은 (예를 들어, 코어/캐시가 동작 모드에 있든 아니면 전력 절약 모드(예를 들어, "슬립" 모드, "전력 붕괴" 모드, "유티" 모드)에 있든 간에) 프로세서 코어 및/또는 캐시의 전력 상태를 포함할 수 있

다.

- [0039] [0044] 상호연결 모듈에 의해 고려될 수 있는 또 다른 시스템 변수는 프로세서 코어의 주파수 및/또는 캐시의 주파수를 포함할 수 있다.
- [0040] [0045] 시스템 변수는 또한, 컴퓨터 시스템이 이중 시스템인 경우 시스템 레이턴시를 포함할 수 있다. 예를 들어, 상호연결 모듈은 하나 또는 그 초과 GPU(Graphic Processing Unit), 하나 또는 그 초과 DSP(digital signal processors) 및/또는 32 비트 및 64 비트의 혼합 범용 마이크로프로세서 코어들과 같은 상이한 아키텍처들을 갖는 프로세서 코어들을 지원할 수 있다. 이러한 시나리오에서, 상호연결 모듈은 개별 프로세서 코어들 또는 프로세서 코어들의 결합의 레이턴시를 고려할 수 있다.
- [0041] [0046] 상호연결 모듈에 의해 고려될 수 있는 또 다른 시스템 변수는 프로세서 코어 및/또는 캐시의 현재 활용을 포함할 수 있다.
- [0042] [0047] 상호연결 모듈에 의해 고려될 수 있는 또 다른 시스템 변수는 요청된 캐시 라인을 제공할 소유 프로세서 코어 및/또는 캐시를 선택하기 전에, 마이크로프로세서 아키텍처에서 상호연결 모듈 세그먼트들의 현재 활용을 포함할 수 있다.
- [0043] [0048] 상호연결 모듈에 의해 고려될 수 있는 또 다른 시스템 변수는 프로세서 코어 및/또는 캐시 요청들의 웨어 밸런싱 등을 포함할 수 있다. 예를 들어, 특정 반도체 기술들(예를 들어, FinFET와 같은 멀티-게이트 디바이스들)은, 회로의 실패율이 회로가 얼마나 빈번하게 "사용", 즉, 스위칭 온 및 스위치 오프되는지와 관련된다는 특성을 갖는다. 하나 또는 그 초과 구현들에서, 상호연결 모듈은 반도체(들)의 수명을 최대화하기 위해 "동등한 경로들" 사이에서 균등하게 작업을 분배하려고 시도하는 것에 기초하여 개재자가 될 캐시를 선택할 수 있다.
- [0044] [0049] 물론, 시스템 변수들의 이러한 리스트는 전면적이지 않고, 시스템 가변 모듈(180)은 훨씬 더 많은 시스템 변수들을 포함할 수 있다.
- [0045] [0050] 캐시들(140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164, 166, 168 및 170) 각각은 캐시 라인들을 포함한다. 데이터 파일들은 전형적으로, 고정 크기의 블록들로 시스템 메모리(104)와 캐시들 사이에서 전달된다. 본원에서 사용되는 바와 같이, 데이터의 블록들은 "캐시 라인들"이라 칭해진다. 각각의 캐시는 그가 캐싱한 캐시 라인들과 연관된 어드레스들 전부의 디렉토리를 포함한다.
- [0046] [0051] 캐시 라인이 시스템 메모리(104)로부터 캐시로 카피되는 경우, 캐시 엔트리가 생성된다. 캐시 엔트리는 카피된 캐시 라인뿐만 아니라, 요청된 시스템 메모리(104) 위치(전형적으로, "태그"라 칭해짐)를 포함할 것이다. 프로세서 코어가 시스템 메모리(104) 내의 위치를 판독 또는 기록할 필요가 있는 경우, 프로세서 코어는 먼저, 캐시 내의 대응하는 엔트리에 대해 체크한다. 캐시는 해당 어드레스를 포함할 수 있는 자신의 캐시 라인들 중 임의의 캐시 라인 내의 요청된 메모리 위치의 콘텐츠들에 대해 체크한다. 프로세서 코어가 메모리 위치가 자신의 캐시 내에 있음을 발견하면, 캐시 "히트"가 발생한다. 그러나, 프로세서 코어가 자신의 캐시 내의 메모리 위치를 발견하지 못하면, 캐시 "미스"가 발생한다.
- [0047] [0052] 도 2는 하나 또는 그 초과 구현들에 따른, 캐시 "미스"에 대한 응답으로 다이렉트 스눅 개재를 예시하는 블록도이다. 본원에 설명되는 기술의 하나 또는 그 초과 구현들에 따라, 프로세서 코어 자신의 캐시에서 "미스"인 경우, 프로세서 코어는 하나 또는 그 초과 다른 프로세서들과 연관된 캐시 내의 캐시 라인을 판독하기 위한 요청을 발행한다.
- [0048] [0053] 설명을 목적으로 그리고 도 2를 참조하면, 프로세서 코어(134)가 시스템 메모리(104) 내의 캐시 라인 0을 판독할 필요가 있지만 자신의 캐시 내의 메모리 위치를 발견하지 못한다고, 즉, 캐시 "미스"가 발생한다고 가정하기로 한다. 프로세서 코어(134)는 캐시 라인 0을 판독하기 위한 요청을 상호연결 모듈(106)에 발행한다. 스눅 모듈(176)은 각각의 캐시들에서 명명법 "CL0"에 의해 표시되는 바와 같이, 캐시들(152, 164 및 168)이 요청된 캐시 라인을 포함함을 결정한다. 스눅 모듈(176)은 캐시들(152, 164 및 168)에 대한 시스템 가변 모듈(180)에 포함되는 변수들을 비교한다. 그 다음, 스눅 모듈(176)은 최저 레이턴시, 최저 전력, 최고 속도 등을 표현하는 캐시 라인 0을 포함하는 캐시를 선택한다. 예시되는 구현에 따라, 스눅 모듈(176)은 명명법 "CL0:IN"에 의해 표시되는 바와 같이, 캐시(152)를 선택한다.
- [0049] [0054] 그 다음, 버스 시그널링 모듈(178)은 프로세서 코어(120)의 캐시(152)가 캐시 라인 0을 프로세서 코어(134)에 제공하도록 프로세서 코어(120)에 통지한다. 하나 또는 그 초과 구현들에서, 버스 시그널링 모듈

(178)은 프로세서 코어(120)의 캐시(152)가 캐시 라인 0을 프로세서 코어(134)에 제공하도록 프로세서 코어(120)에 통지하기 위해 "InterveneIfValid" 신호(202)를 어서트(assert)한다. 버스 시그널링 모듈(178)로부터의 "InterveneIfValid" 신호에 대한 응답으로, 캐시(152)는 그 다음, 캐시 라인 0을 프로세서 코어(134)에 제공한다.

- [0050] [0055] 도 3은 하나 또는 그 초과와 구현들에 따른, 다이렉트 스눅 개재가 활용될 수 있는 컴퓨터 시스템(300)을 예시하는 블록도이다. 예시되는 컴퓨터 시스템(300)은 서버 칩(102), 시스템 메모리(104) 및 캐시(304)를 갖는 멀티프로세서 칩(302), 캐시(308)를 갖는 GPU(Graphics Processing Unit)(306), 캐시(312)를 갖는 DSP(Digital Signal Processor)(310), 하나 또는 그 초과와 캐시들(316)을 갖는 하나 또는 그 초과와 32-비트 범용 마이크로프로세서 코어들(32-비트 GP 코어(들))(314) 및 하나 또는 그 초과와 캐시들(320)을 갖는 하나 또는 그 초과와 64-비트 범용 마이크로프로세서 코어들(64-비트 GP 코어(들))(318)에 커플링된 상호연결 모듈(106)을 포함한다.
- [0051] [0056] 하나 또는 그 초과와 구현들에서, 멀티프로세서 칩(302)은 다수의 프로세서 코어들을 지원할 수 있는 임의의 적합한 집적 회로일 수 있다.
- [0052] [0057] 하나 또는 그 초과와 구현들에서, 캐시들(304, 308, 312, 316 및 320) 각각은 그것이 캐싱한 캐시 라인들과 연관된 어드레스들 전부의 디렉토리를 포함한다. 하나 또는 그 초과와 구현들에서, GPU(306)는 디스플레이를 위한 스틸 또는 비디오와 같은 이미지들을 프로세싱할 수 있는 임의의 프로세싱 유닛일 수 있다. 하나 또는 그 초과와 구현들에서, DSP(310)는 데이터에 대한 수학적 연산들을 수행할 수 있는 임의의 적합한 종래의 디지털 신호 프로세서일 수 있다. 하나 또는 그 초과와 구현들에서, 32-비트 GP 코어(314)는 32-비트 명령 세트 아키텍처를 사용하여 동작할 수 있는 임의의 적합한 멀티프로세서일 수 있다. 하나 또는 그 초과와 구현들에서, 64-비트 GP 코어(318)는 64-비트 명령 세트 아키텍처를 사용하여 동작할 수 있는 임의의 적합한 멀티프로세서일 수 있다.
- [0053] [0058] 컴퓨터 시스템(300)의 동작은 도 4를 참조하여 설명되고, 도 4는 하나 또는 그 초과와 구현들에 따른, 멀티프로세서 아키텍처에서 상호연결 모듈에 걸친 다이렉트 스눅 개재를 구현하기 위한 방법(400)의 예시적 흐름도이다. 하나 또는 그 초과와 구현들에서, 비-일시적 컴퓨터 관독가능한 저장 매체는, 기계에 의해 액세스될 때 기계로 하여금 방법(400)을 포함하는 동작들을 수행하게 하는 데이터를 포함할 수 있다.
- [0054] [0059] 블록(402)에서, 방법(400)은 요청되는 캐시 라인을 관독하기 위한 요청을 요청 프로세서로부터 획득한다. 하나 또는 그 초과와 구현들에서, 방법(400)은 요청 프로세서 코어에 의한 캐시 "미스" 이후, 캐시 라인에 대한 요청을 프로세서 코어로부터 획득한다. 설명을 목적으로, 프로세서 코어(134)(도 2에 예시됨)가 캐시 라인 0을 관독하기 위한 요청을 상호연결 모듈(106)에 발행한다고 가정하기로 한다.
- [0055] [0060] 블록(404)에서, 방법(400)은 어떤 소유 프로세서 캐시들이 요청된 캐시 라인을 포함하는지를 결정한다. 예에 따라서, 스눅 모듈(176)은 프로세서 코어들(120, 132 및 136)에 대한 캐시들(152, 164 및 168)이 각각, 각각의 캐시들에서의 명명법 "CLO"에 의해 표시되는 바와 같이, 요청된 캐시 라인을 포함함을 결정한다고 가정하기로 한다. 따라서, 프로세서 코어들(120, 132 및 136)은 캐시 라인 0을 소유하며, 따라서, 요청된 캐시 라인 0을 갖는 "소유" 프로세서 코어들로 고려된다.
- [0056] [0061] 블록(406)에서, 방법(400)은 효율적 방식으로 하나 또는 그 초과와 변수들에 기초하여, 요청된 캐시 라인을 요청 프로세서 코어에 제공하기 위해 소유 프로세서 코어를 선택한다. 하나 또는 그 초과와 구현들에서, 상호연결 모듈(106)은 컴퓨터 시스템(300)의 토폴로지에 기초하여, 요청된 캐시 라인을 요청 프로세서 코어에 제공하기 위해 소유 프로세서 코어를 선택할 수 있다.
- [0057] [0062] 예를 들어, 스눅 모듈(176)은 요청된 캐시 라인이 서버 칩(102) 상에 있는지 여부, 요청된 캐시 라인이, 이를테면, 캐시들(304, 308, 312, 316 및 320)에서 오프-칩인지 여부, 요청된 캐시 라인이 시스템 메모리(104) 내에 있는지 여부 그리고/또는 요청된 캐시 라인이, 이를테면, 멀티프로세서 칩(302)의 캐시(304)에서, 또 다른 멀티프로세서 칩 상에 있는지 여부를 고려하기 위해 시스템 가변 모듈(180)과 상호작용할 수 있다.
- [0058] [0063] 일반적으로, 요청된 캐시 라인이 요청 프로세서 코어에 도달하기 위해 칩 바운더리(chip boundary)를 교차하여야 할 때는 언제나, 그것은 훨씬 더 느린 프로세스이다. 상호연결 모듈(106)은 개입 캐시가 될 캐시를 선택할 때 이러한 인자를 고려할 것이다. 따라서, 요청된 캐시 라인이 온-칩 대 오프-칩인 경우, 상호연결 모듈(106)은 캐시 라인의 마지막 카피가 오프-칩일 수 있음에도 불구하고 캐시 라인을 제공하기 위해 온-칩인 소유 프로세서 코어를 선택할 수 있다.

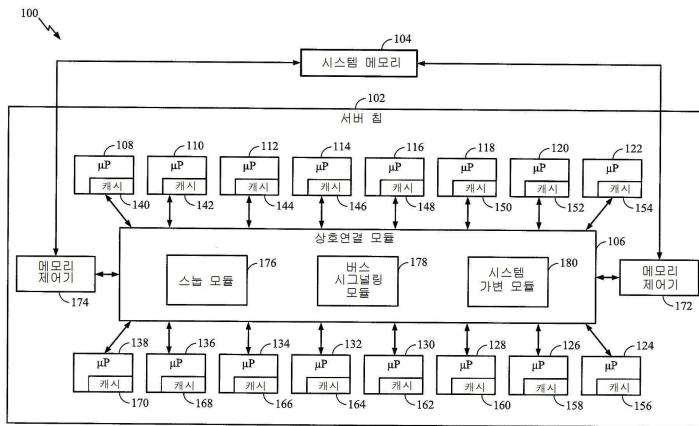
- [0059] [0064] 하나 또는 그 초과와 구현들에서, 스냅 모듈(176)은 요청된 캐시 라인을 제공할 소유 프로세서 코어 및/또는 캐시의 전력 상태를 결정하기 위해 시스템 가변 모듈(180)과 상호작용할 수 있다. 예에 따라서, 상호연결 모듈(106)은 프로세서 코어들(120, 132 및 136)에 대해뿐만 아니라, 캐시들(152, 164 및 168)의 동작 모드 또는 전력 절약 모드를 고려할 수 있다. 예를 들어, 프로세서 코어(136)가 프로세서 코어(132)보다 더 낮은 전력 상태에 있으면, 프로세서 코어(136)는 요청된 캐시 라인을 제공하는데 선택되지 않을 수 있는데, 그 이유는 프로세서 코어(136)가 요청된 캐시 라인을 제공할 수 있도록 하기 위해서는 프로세서(136)를 웨이크업하기 위한 전력 및/또는 시간이 소비될 수 있기 때문이다.
- [0060] [0065] 하나 또는 그 초과와 구현들에서, 스냅 모듈(176)은 요청된 캐시 라인을 제공할 소유 프로세서 코어 및/또는 캐시의 주파수를 결정하기 위해 시스템 가변 모듈(180)과 상호작용할 수 있다. 예에 따라서, 상호연결 모듈(106)은 프로세서 코어들(120, 132 및 136)에 대해뿐만 아니라 캐시들(152, 164 및 168)의 주파수를 고려할 수 있다. 예를 들어, 프로세서 코어(136)가 프로세서 코어(132)보다 더 높은 주파수에서 동작하고 있으면, 프로세서 코어(136)로부터 요청된 캐시 라인을 제공하는 것이 더 효율적일 수 있고, 프로세서 코어(132)는 요청된 캐시 라인을 제공하는데 선택되지 않을 수 있는데, 그 이유는 프로세서(132)가 요청된 캐시 라인을 제공하는데 더 오래 걸릴 수 있기 때문이다.
- [0061] [0066] 하나 또는 그 초과와 구현들에서, 스냅 모듈(176)은 요청된 캐시 라인을 제공할 소유 프로세서 코어 및/또는 캐시를 선택하기 전에 레이턴시를 결정하기 위해 시스템 가변 모듈(180)과 상호작용할 수 있다. 예에 따라서, 상호연결 모듈(106)은 프로세서 코어들(120, 132 및 136)의 레이턴시를 고려할 수 있다. 예를 들어, 프로세서 코어(136)가 프로세서 코어(132)와 상이한 타입의 프로세서이면, 프로세서 코어(132)는 프로세서 코어(136)의 레이턴시보다 고유하게 더 긴 레이턴시를 가질 수 있다. 이로써, 프로세서 코어(132)는 그것이 요청 프로세서 코어(134)에 더 근접할 수 있음에도 불구하고, 프로세서 코어(136)로부터 요청된 캐시 라인을 제공하는 것이 더 효율적일 수 있고, 프로세서 코어(132)는 요청된 캐시 라인을 제공하도록 선택되지 않을 수 있다.
- [0062] [0067] 하나 또는 그 초과와 구현들에서, 스냅 모듈(176)은 요청된 캐시 라인을 제공할 소유 프로세서 코어 및/또는 캐시를 선택하기 전에 로드를 결정하기 위해 시스템 가변 모듈(180)과 상호작용할 수 있다. 예에 따라서, 프로세서 코어(132)에 대한 캐시(164)가 자기 자신의 동작들에 의해 과도하게 로딩되고, 프로세서 코어(136)에 대한 캐시(168)가 유향한 상태이면, 캐시(168)가 요청 프로세서 코어(120)로부터 더 멀리 떨어져 있음에도 불구하고 캐시(168)로부터 요청된 캐시 라인 0을 획득하는 것이 더 효율적일 수 있다.
- [0063] [0068] 하나 또는 그 초과와 구현들에서, 스냅 모듈(176)은 요청된 캐시 라인을 제공할 소유 프로세서 코어 및/또는 캐시를 선택하기 전에 프로세서 코어 및/또는 캐시의 현재 활용을 결정하기 위해 시스템 가변 모듈(180)과 상호작용할 수 있다. 즉, 상호연결 모듈(106)은 명령들을 프로세싱하기 위해 프로세서 코어 및/또는 캐시가 사용하는 시간의 양을 고려할 수 있다. 예에 따라서, 상호연결 모듈(106)은, 프로세서 코어들(120, 132 및 136) 및/또는 캐시들(152, 164 및 168)의 현재 활용이 요청된 캐시 라인을 개재하기 위해 레이턴시를 가질 것이라는 효과를 고려할 수 있다.
- [0064] [0069] 하나 또는 그 초과와 구현들에서, 스냅 모듈(176)은 요청된 캐시 라인을 제공할 소유 프로세서 코어 및/또는 캐시를 선택하기 전에 상호연결 모듈(106) 세그먼트들의 현재 활용을 결정하기 위해 시스템 가변 모듈(180)과 상호작용할 수 있다. 즉, 상호연결 모듈(106)은 프로세서 코어들(120, 132 및 136) 및/또는 캐시들(152, 164 및 168)의 현재 활용이 요청된 캐시 라인을 개재하기 위해 레이턴시를 가질 것이라는 효과를 결정할 수 있다.
- [0065] [0070] 하나 또는 그 초과와 구현들에서, 스냅 모듈(176)은 요청된 캐시 라인을 제공할 소유 프로세서 코어 및/또는 캐시를 선택하기 전에 웨어 밸런스를 결정하기 위해 시스템 가변 모듈(180)과 상호작용할 수 있다. 예에 따라서, 상호연결 모듈(106)은 프로세서 코어들(120, 132 및 136)의 웨어 밸런스를 고려할 수 있다. 예를 들어, 프로세서 코어들(120, 132 및/또는 136)이 자신의 회로들의 실패율이 회로들이 얼마나 빈번하게 "사용", 즉, 스위칭 온 및 스위치 오프되는지와 관련된다는 특성을 갖는 특정 반도체 기술들(예를 들어, FinFET와 같은 멀티-게이트 디바이스들)을 활용할 경우, 상호연결 모듈(106)은 반도체(들)의 수명을 최대화하기 위해 "동등한 경로들" 사이에서 균등하게 작업을 분배하려고 시도하는 것에 기초하여 개재자가 될 캐시를 선택할 수 있다.
- [0066] [0071] 설명을 목적으로, 요청된 캐시 라인 0이 최저 레이턴시, 최저 전력, 최고 속도 등을 표현하기 때문에, 상호연결 모듈(106)이 이 요청된 캐시 라인 0을 요청 프로세서(134)에 제공하기 위한 개재자로서 프로세서 코어(120)의 캐시(152)를 선택한다고 가정하기로 한다. 선택은 캐시(152)에 도시된 명명법 "CLO:IN"에 의해 표시된

다.

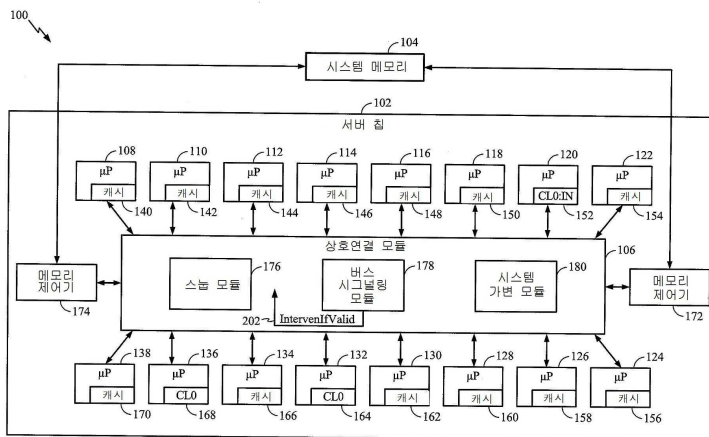
- [0067] [0072] 블록(408)에서, 방법(400)은 요청된 캐시 라인을 요청 프로세서 코어에 제공하도록, 선택된 소유 프로세서에 통지한다. 하나 또는 그 초과된 구현들에서, 스눅 모듈(176)은 버스 시그널링 모듈(178)이 캐시 라인 0을 요청 프로세서(134)에 제공하도록 프로세서 코어(120) 내의 캐시(152)에 통지하기 위해 버스 시그널링 모듈(178)과 상호작용한다. 그 다음, 버스 시그널링 모듈(178)은 프로세서 코어(120)의 캐시(152)가 캐시 라인 0을 프로세서 코어(134)에 제공하도록 프로세서 코어(120)에게 통지한다. 예를 들어, 버스 시그널링 모듈(178)은 프로세서 코어(120)의 캐시(152)가 캐시 라인 0을 프로세서 코어(134)에 제공하도록 프로세서 코어(120)에 통지하기 위해 "InterveneIfValid" 신호(202)를 어서트(assert)할 수 있다.
- [0068] [0073] 블록(410)에서, 선택된 소유 프로세서는 요청된 캐시 라인을 요청 프로세서에 제공한다. 하나 또는 그 초과된 구현들에서, 버스 시그널링 모듈(178)로부터의 "InterveneIfValid" 신호(202)에 대한 응답으로, 프로세서 코어(120)에 대한 캐시(152)는 캐시 라인 0을 프로세서 코어(134)에 제공한다.
- [0069] [0074] 다양한 방법들의 단계들 관정들이 본 개시에 연속하여 설명될 수 있지만, 이러한 단계들 및 관정들 일부는 함께 또는 동시에, 비동기식으로 또는 동기식으로, 파이프라인 방식(pipelined manner)으로 또는 그 외의 방식으로 개별 엘리먼트들에 의해 수행될 수 있다. 명시적으로 그렇게 표시되거나, 그렇지 않으면 문맥으로부터 명백하거나, 또는 고유하게 요구되는 경우를 제외하고, 본 설명이 열거하고 있는 것과 동일한 순서로 단계들 및 관정들이 수행된다는 어떠한 특정 요건이 있는 것은 아니다. 그러나, 선택된 변형들에서, 단계들 및 관정들이 위에서 설명된 순서로 수행된다는 점이 주목되어야 한다. 게다가, 모든 각각의 예시된 단계 및 관정이 본 발명에 따라 모든 각각의 실시예/변형에서 요구되지 않을 수 있지만, 구체적으로 예시되지 않은 일부 단계들 및 관정들이 본 발명에 따른 일부 실시예들/변형들에서 요구되거나 또는 필요하지 않을 수 있다.
- [0070] [0075] 당업자들은 정보 및 신호들이 다양한 상이한 기술들 및 기법들 중 임의의 기술 및 기법을 사용하여 표현될 수 있다는 것을 이해할 것이다. 예를 들어, 상기 설명 전반에 걸쳐 참조될 수 있는 데이터, 명령들, 커맨드들, 정보, 신호들, 비트들, 심볼들 및 칩들은 전압들, 전류들, 전자기파들, 자기장들 또는 자기 입자들, 광 필드들 또는 광 입자들, 또는 이들의 임의의 결합에 의해 표현될 수 있다.
- [0071] [0076] 당업자들은 본원에 개시된 실시예들과 관련하여 설명된 다양한 예시적 논리 블록들, 모듈들, 회로들 및 알고리즘 단계들이 전자 하드웨어, 컴퓨터 소프트웨어 또는 이 둘의 결합들로서 구현될 수 있다는 것을 추가로 인식할 것이다. 하드웨어 및 소프트웨어의 이러한 상호교환가능성을 명백하게 나타내기 위해, 다양한 예시적 컴포넌트들, 블록들, 모듈들, 회로들 및 단계들이 일반적으로 그들의 기능에 관하여 위에서 설명되었다. 이러한 기능이 하드웨어로서 구현되는지, 소프트웨어로서 구현되는지 아니면 하드웨어 및 소프트웨어의 결합으로서 구현되는지는 전체 시스템 상에 부과되는 설계 제약들 및 특정 애플리케이션에 의존한다. 당업자들은 각각의 특정 애플리케이션에 대해 다양한 방식으로 설명된 기능을 구현할 수 있지만, 이러한 구현 관정들은 본 발명의 범위를 벗어나게 하는 것으로 해석되어서는 안 된다.
- [0072] [0077] 본원에 개시된 실시예들과 관련하여 설명된 알고리즘 또는 방법의 단계들은 직접적으로 하드웨어로, 프로세서에 의해 실행되는 소프트웨어 모듈로, 또는 이 둘의 결합으로 구현될 수 있다. 소프트웨어 모듈은 RAM 메모리, 플래시 메모리, ROM 메모리, EPROM 메모리, EEPROM 메모리, 레지스터들, 하드디스크, 이동식 (removable) 디스크, CD-ROM, 또는 당해 기술 분야에 공지된 임의의 다른 형태의 저장 매체에 상주할 수 있다. 예시적 저장 매체는, 프로세서가 저장 매체로부터 정보를 판독하고, 저장 매체에 정보를 기록할 수 있도록, 프로세서에 커플링된다. 대안적으로, 저장 매체는 프로세서에 통합될 수 있다. 프로세서 및 저장 매체는 ASIC에 상주할 수 있다. ASIC는 액세스 단말에 상주할 수 있다. 대안적으로, 프로세서 및 저장 매체는 액세스 단말에서 별개의 컴포넌트들로서 상주할 수 있다.
- [0073] [0078] 개시된 실시예들의 이전 설명은 임의의 당업자가 본 발명을 실시하거나 또는 사용할 수 있도록 제공된다. 이러한 실시예들에 대한 다양한 변형들은 당업자들에게 쉽게 명백할 것이고, 본원에서 정의된 일반적인 원리들은 본 발명의 사상 또는 범위를 벗어나지 않으면서 다른 실시예들에 적용될 수 있다. 따라서, 본 발명은 본원에 도시된 실시예들에 제한되는 것으로 의도된 것이 아니라, 본원에 개시된 원리들 및 신규한 특징들과 일치하는 가장 넓은 범위를 따를 것이다.

도면

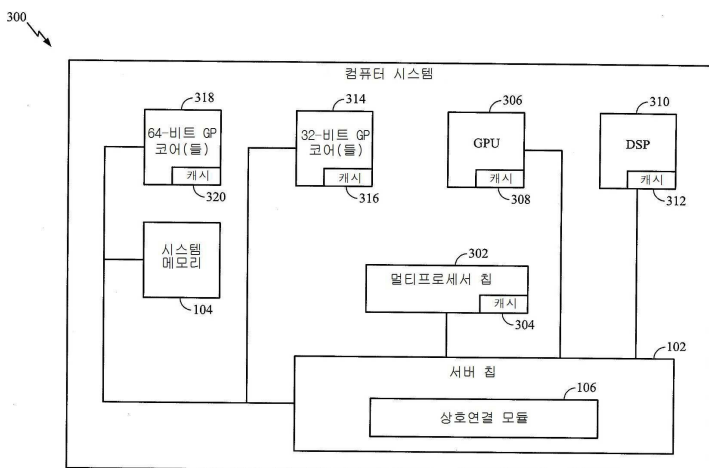
도면1



도면2



도면3



도면4

