(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2012-0102629
(43) 공개일자 2012년09월18일

(51) 국제특허분류(Int. Cl.)
G06F 1/32 (2006.01) *G06F 12/08* (2006.01)
(21) 출원번호 10-2012-7011940
(22) 출원일자(국제) 2010년11월05일
심사청구일자 없음
(85) 번역문제출일자 2012년05월08일
(86) 국제출원번호 PCT/US2010/055598
(87) 국제공개번호 WO 2011/057059
국제공개일자 2011년05월12일
(30) 우선권주장
12/623,997 2009년11월23일 미국(US)
61/258,798 2009년11월06일 미국(US)

(71) 출원인
어드밴스드 마이크로 디바이시즈, 인코포레이티드
미국 캘리포니아 94088-3453 서니베일 원 에이엠디 플레이스 메일 스톱68
(72) 발명자
브래노버 알렉산더
미국 메사추세츠 02467 체스트넛 힐 뉴턴 스트리트 783
슈타인만 모리스 비.
미국 메사추세츠 01752 말보로 디논코트 스트리트 78
(뒷면에 계속)
(74) 대리인
박장원

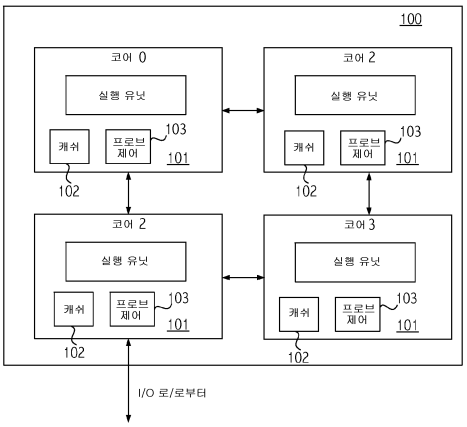
전체 청구항 수 : 총 14 항

(54) 발명의 명칭 **프로브 활동 레벨의 추적에 의한 성능 상태의 제어**

(57) 요 약

프로세싱 노드가 그것의 내부 캐쉬 또는 메모리 시스템과 관련된 프로브 활동 레벨을 추적한다. 만일 프로브 활동 레벨이 프로브 활동 레벨의 임계값보다 위로 증가하는 경우, 프로브 요청들에 응답하는 데 있어서 향상된 성능 능력을 제공하기 위해 프로세싱 노드의 성능 상태는 현재 성능 상태보다 위로 증가된다. 프로브 활동 레벨이 프로브 활동 레벨의 임계값보다 위에 있는 것에 응답하여 더 높은 성능 상태로 진입한 후에, 프로브 활동의 감소에 응답하여 프로세싱 노드는 더 낮은 성능 상태로 되돌아간다. 복수의 프로브 활동 레벨의 임계값들 및 관련 성능 상태들이 존재할 수 있다.

대 표 도 - 도1



(72) 발명자

하우케 조나단 디.

미국 메사추세츠 02420 렉싱턴 베드포드 스트리트 443

오웬 조나단 엠.

미국 메사추세츠 01532 노스보로 스쿨 스트리트 155

특허청구의 범위

청구항 1

프로세싱 노드의 프로브 활동 레벨(probe activity level)을 추적하는 단계와;

상기 프로브 활동 레벨을 프로브 활동 레벨의 제1 임계값과 비교하는 단계와; 그리고

만일 상기 프로브 활동 레벨이 상기 프로브 활동 레벨의 제1 임계값보다 위에 있는 경우 상기 프로세싱 노드의 성능 상태를 현재 성능 상태보다 더 높은 제1 성능 상태로 증가시키는 단계를 포함하는

방법.

청구항 2

제1항에 있어서,

상기 프로브 활동 레벨이 상기 프로브 활동 레벨의 제1 임계값보다 위에 있는 것에 응답하여 상기 제1 성능 상태로 진입한 후에, 상기 프로브 활동 레벨이 상기 프로브 활동 레벨의 제1 임계값보다 아래의 소정의 레벨 아래로 떨어질 때 상기 제1 성능 상태보다 더 낮은 제2 성능 상태로 진입하는 단계를 더 포함하는

방법.

청구항 3

제2항에 있어서,

상기 프로브 활동 레벨이 상기 제1 임계값에서 히스테리시스 계수(hysteresis factor) 만큼 작은 것보다 더 낮을 때 상기 제2 성능 상태로 진입하는 단계를 더 포함하는

방법.

청구항 4

제3항에 있어서,

상기 제2 성능 상태는 상기 프로세싱 노드가 상기 제1 성능 상태로 진입하기 전의 성능 상태인

방법.

청구항 5

제1항 내지 제4항 중 어느 한 항에 있어서,

상기 제1 성능 상태와 상기 제2 성능 상태는 전압 및 주파수 중에서 적어도 하나에 의해 정의되는

방법.

청구항 6

제1항 내지 제4항 중 어느 한 항에 있어서,

상기 프로브 활동 레벨이 프로브 활동 레벨의 제2 임계값보다 위로 증가하는 것에 응답하여 상기 프로세싱 노드의 성능 상태를 제3 성능 상태로 증가시키는 단계와, 상기 프로브 활동 레벨의 제2 임계값은 상기 프로브 활동 레벨의 제1 임계값보다 더 높고, 상기 제3 성능 상태는 상기 제1 성능 상태보다 더 높으며; 그리고

상기 프로브 활동 레벨이 프로브 활동 레벨의 제2 임계값보다 위로 증가하는 것에 응답하여 상기 프로세싱 노드의 성능 상태를 제3 성능 상태로 증가시키는 단계 후에, 상기 성능 상태를 감소시키는 단계를 더 포함하는

방법.

청구항 7

제1항 내지 제4항 중 어느 한 항에 있어서,

상기 프로세싱 노드가 상기 제1 성능 상태보다 더 낮은 성능 상태에 있을 때 상기 프로브 활동 레벨의 추적을 시작하는 단계를 더 포함하는

방법.

청구항 8

제1항 내지 제4항 중 어느 한 항에 있어서,

상기 프로브 활동을 추적하는 단계는

각 프로브 요청을 큐(queue) 안으로 넣고, 상기 프로세싱 노드가 상기 프로브 요청에 대해 데이터 이동 및 응답 중에서 적어도 하나로 응답한 후에 상기 큐로부터 프로브 요청을 퇴거시키는 것과; 그리고

상기 프로브 활동 레벨이 상기 프로브 활동 레벨의 제1 임계값보다 위에 있는지를 판별하기 위해 상기 큐의 엔트리들의 개수를 상기 프로브 활동 레벨의 제1 임계값과 비교하는 것을 더 포함하는

방법.

청구항 9

제1항 내지 제4항 중 어느 한 항에 있어서,

상기 프로브 활동을 추적하는 단계는 프로브 활동의 발생에 응답하여 프로브 활동 레벨을 표시하는 카운트 값을 증가시키고, 소정의 양의 시간이 경과한 것에 의거하여 상기 카운트 값을 감소시키는 것을 더 포함하는

방법.

청구항 10

프로세싱 노드의 프로브 활동 레벨을 추적하는 프로브 추적기(probe tracker)를 포함하는 장치로서,

상기 장치는 만일 상기 프로브 활동 레벨이 프로브 활동 레벨의 제1 임계값보다 위로 증가하는 경우 상기 프로세싱 노드의 성능 상태를 현재 성능 상태로부터 제1 성능 상태로 증가시키도록 응답하며; 그리고

상기 장치는 상기 프로브 활동 레벨이 상기 프로브 활동 레벨의 제1 임계값보다 아래의 소정의 레벨로 떨어지는 것에 응답하여 상기 프로세싱 노드를 상기 제1 성능 상태보다 더 낮은 제2 성능 상태로 진입시키며; 그리고

상기 제1 성능 상태와 상기 제2 성능 상태는 전압 및 주파수 중에서 적어도 하나에 의해 정해지는

장치.

청구항 11

제10항에 있어서,

상기 장치는 또한 상기 프로브 활동 레벨이 프로브 활동 레벨의 제2 임계값보다 위로 증가하는 것에 응답하여 상기 프로세싱 노드의 성능 상태를 제3 성능 상태로 증가시키도록 동작가능하며, 상기 프로브 활동 레벨의 제2 임계값은 상기 프로브 활동 레벨의 제1 임계값보다 더 높고, 상기 제3 성능 상태는 상기 제1 성능 상태보다 더 높은

장치.

청구항 12

제10항에 있어서,

상기 프로브 추적기는 상기 노드가 상기 제1 성능 상태보다 아래의 성능 상태에 있는 것에 응답하여 상기 프로브 활동 레벨의 추적을 시작하는

장치.

청구항 13

제10항 내지 제12항 중 어느 한 항에 있어서,

상기 프로브 추적기는 큐를 더 포함하며, 프로브 요청이 상기 큐로 입력되고, 상기 프로세싱 노드가 상기 프로브 요청에 대해 데이터 이동 및 응답 중에서 적어도 하나로 응답한 후에 상기 프로브 요청은 상기 큐에서 퇴거되며; 그리고

상기 장치는 상기 프로브 활동 레벨이 프로브 활동 레벨의 제1 임계값보다 위에 있는지를 판별하기 위해 상기 큐의 엔트리들의 개수를 상기 프로브 활동 레벨의 제1 임계값과 비교하도록 동작가능한

장치.

청구항 14

제10항 내지 제12항 중 어느 한 항에 있어서,

상기 프로브 추적기는 카운터를 포함하며, 상기 카운터는 프로브 활동에 응답하여 프로브 활동 레벨을 표시하는 카운트 값을 증가시키고, 소정의 시간 간격이 경과한 것에 응답하여 상기 카운트 값을 감소시키는

장치.

명 세 서

기 술 분 야

[0001] 본 발명은 컴퓨터 시스템의 성능에 관한 것이며, 더욱 상세하게는 캐쉬 프로브와 관련된 성능에 관한 것이다.

배 경 기 술

[0002] 컴퓨터 시스템에서 프로세싱 노드는 복수의 성능 상태들(또는 동작 상태들) Pn 중 하나에 놓여질 수 있으며, 특정한 성능 상태(또는 P-상태)는 관련된 전압 및 주파수에 의해 특징지어진다. 노드의 적절한 성능 상태를 결정하는 하나의 인자는 그것의 활용도(utilization)이다. 활용도는 활성(실행) 상태에서 프로세싱 노드에 의해 소비된 시간 대 실행 시간을 추적하거나 측정한 전체 시간 구간의 비이다. 예를 들어, 만일 전체 시간 구간이 10 밀리초(millisecond, ms)이었고 프로세서 노드가 활성(C0) 상태에서 6 ms를 소비하였다면, 상기 프로세서 노드의 활용도는 60%(= 6/10)이다. 프로세서 노드는 코드 실행이 중단(suspend)된 유휴(idle) 상태(비-C0(non-C0) 상태)에서 나머지 4 ms를 소비한다. 더 높은 노드 활용도는 더 높은 성능 상태 P의 선택을 트리거하며, 더 높은 성능 상태 P는 와트당 성능비(performance/watt) 요구조건들에 더욱 잘 대처하도록 더 높은 전압 및/또는 주파수를 가진다. 일반적으로 프로세싱 노드를 성능 상태들 간에 전환시키는 결정은 운영체제(OS)나 상위 레벨의 소프트웨어, 드라이버, 또는 일부 하드웨어 제어기에 의해 이루어진다. 예를 들어, 만일 프로세싱 노드가 낮은 성능 상태에서 실행되어 더 긴 코드 실행 시간이 걸리는 경우, 시스템은 더 높은 활용도에 대한 필요성을 인지하고, 코드 실행을 더 빨리 완료하고 유휴 상태에서 더 많은 시간을 보낼 수 있는 더 높은 성능 상태로 프로세싱 노드를 전환시키기 위해 하드웨어 또는 소프트웨어를 트리거한다. 이로써 전체적으로 더 나은 와트당 성능비로부터 전력 절감이 증가될 수 있다. 활용도를 트리거로서 사용하는 것은 일부 상황들에서 와트당 성능비의 증가를 제공할 수 있지만, 더 나은 와트당 성능비 또는 와트당 성능비의 저하를 방지하는 것과 관련된 일부 문제들을 해결하지는 못한다.

발명의 내용

[0003] 이에 따라, 한 실시예에서, 프로세싱 노드의 프로브 활동 레벨을 추적하는 것을 포함하는 방법이 제공된다. 프로브 활동 레벨은 프로브 활동 레벨의 임계값과 비교된다. 한 실시예에서, 만일 프로브 활동 레벨이 프로브 활동 레벨의 임계값보다 위에 있는 경우, 상기 프로세싱 노드의 성능 상태는 현재 성능 상태보다 위로 증가된다. 한 실시예에서, 만일 프로브 활동 레벨이 프로브 활동 레벨의 제1 임계값보다 위에 있고 프로세싱 노드의 예측된 유휴 기간이 유휴 임계값보다 더 큰 경우, 프로세싱 노드의 캐쉬 메모리는 플러쉬된다. 한 실시예에서, 프로브 활동 레벨이 프로브 활동 레벨의 임계값보다 위에 있는 것에 응답하여 제1 성능 상태로 진입한 후에, 프로브 활동 레벨의 충분한 감소에 응답하여 프로세싱 노드는 시작하였던 더 낮은 성능 상태로 되돌아간다. 한 실시예에서, 상기 충분한 감소는 제1 임계값에서 히스테리시스 계수만큼 작은 레벨까지의 감소이다. 실시예들에서, 복

수의 프로브 활동 레벨의 임계값들 및 관련 성능 상태들이 존재할 수 있다.

[0004] 다른 실시예에서, 장치는 프로세싱 노드의 프로브 활동 레벨을 추적하는 프로브 추적기(probe tracker)를 포함한다. 상기 장치는 프로브 활동 레벨이 프로브 활동 레벨의 제1 임계값보다 위로 증가하는 것에 응답하여 상기 프로세싱 노드의 성능 상태를 현재 성능 상태로부터 제1 성능 상태로 증가시킨다. 한 실시예에서, 상기 장치는 프로브 활동 레벨이 프로브 활동 레벨의 제1 임계값보다 아래의 소정의 레벨로 떨어지는 것에 응답하여 상기 프로세싱 노드를 제1 성능 상태보다 더 낮은 제2 성능 상태로 진입시킨다.

[0005] 한 실시예에서, 프로브 추적기는 큐를 더 포함하며, 프로브 요청이 상기 큐로 입력되고, 상기 프로세싱 노드가 상기 프로브 요청에 대해 데이터 이동 및 응답 중에서 적어도 하나로 응답한 후에 상기 프로브 요청은 상기 큐에서 퇴거된다. 다른 실시예에서, 프로브 추적기는 프로브 활동 레벨을 나타내는 카운터 값을 가지는 카운터를 포함한다. 카운터는 프로브 활동에 응답하여 소정의 양만큼 카운트 값을 증가시키고, 소정의 시간 간격이 경과한 것에 응답하여 또 다른 소정의 양만큼 카운트 값을 감소시킨다.

도면의 간단한 설명

[0006] 본 발명은 첨부된 도면들을 참조함으로써 더욱 잘 이해될 수 있으며, 본 발명의 목적, 특징, 및 장점들도 당해 기술분야의 통상의 기술자들에게 명백해질 수 있다.

도 1은 본 발명의 한 실시예에 따른 다중-코어 프로세서를 예시한 것이다.

도 2는 단일 임계값을 가지는 본 발명의 한 실시예의 흐름도를 예시한 것이다.

도 3a는 다중 임계값들을 가지는 본 발명의 한 실시예의 상태도를 예시한 것이다.

도 3b는 다중 임계값들을 가지는 본 발명의 한 실시예의 상태도를 예시한 것이다.

도 4는 전력 절감을 위해 노드의 캐쉬들이 플러쉬되는 본 발명의 한 실시예를 예시한 것이다.

도 5는 단일 임계값을 가지는 인-플라이트 큐(In-Flight Queue, IFQ)를 사용하여 프로브 활동을 추적하는 한 실시예를 예시한 것이다.

도 6은 복수의 임계값들을 가지는 IFQ를 사용하여 프로브 활동을 추적하는 한 실시예를 예시한 것이다.

도 7은 서로 다른 증분 및 감소분 기준을 가지는 카운터를 사용하여 프로브 활동을 추적하는 또 다른 실시예를 예시한 것이다.

서로 다른 도면들에서 동일한 참조 부호를 사용하는 것은 유사하거나 동일한 항목을 가리킨다는 것을 유의해야 한다.

발명을 실시하기 위한 구체적인 내용

[0007] 도 1을 보면, 각 코어 또는 노드가 캐쉬 메모리(102)와 프로브 제어(103)를 포함하는 다중-코어 프로세서 실시예가 상위 레벨 블록도로 예시되어 있으며, 본 명세서에서 상세히 서술된다. 도 1의 캐쉬 시스템에서, 시스템의 각 프로세싱 노드는 비록 프로세싱 노드들이 낮은 성능 상태나 유휴 상태에 있더라도 다른 노드들이나 입력/출력(I/O) 도메인으로부터 들어오는 프로브 요청들에 응답함으로써(캐쉬로부터 더티(dirty) 데이터를 제공하거나, 캐쉬 라인 무효화(cache line invalidation) 등) 메모리의 일관성(coherency)을 유지할 필요가 있다. 따라서, 다양한 캐쉬들에서 메모리 위치들의 로컬 카피(local copy)가 유지될 수 있겠지만, 메모리 시스템의 일관성이 유지된다. 하지만, 프로브 동작의 요청 노드들의 성능 상태는 활용도를 평가함으로써 효과적으로 제어될 수 있지만, 그 접근법은 직접적인 방법으로 응답 노드들의 성능 상태 P를 증가시키지 않는다. 응답 노드가 병목인 경우들에 있어서 요청 노드들에 적용가능한 활용도 기반의 성능 제어는 전체 시스템 성능을 취약하게 만든다.

[0008] 노드가 유휴 상태에 있더라도 여전히 프로브 요청들에 응답할 수 있기 때문에 응답 노드에서 일관성 활동(coherent activity)은 노드 그 자체의 활용도 증가(노드의 실행 스트림에 근거한)에 기여하지 않는다. 게다가, 노드의 실행 스트림은 프로브 응답들과 완전히 독립적일 수 있으므로, 응답 노드에서 일관성 활동은 일반적으로 성능 상태의 증가를 트리거하는 더 높은 실행 활용도(execution utilization)로 이어지지 않는다. 만일 응답 노드가 낮은 성능 상태에 있고 많은 수의 요청 노드들에 의해 프로브되는 경우, 응답 노드의 클록 주파수에 의존적인 응답 노드의 프로브 응답 능력(프로브 대역폭(probing bandwidth))은 성능 병목이 되고 요청 프로세싱 노드들 상에서 실행중인 애플리케이션 쓰레드들에 대하여 성능 손실을 유발하기 시작할 수 있다. 따라서, 응답 프

로세싱 노드의 프로브 대역폭이 불충분한 경우의 시나리오들을 파악하고 응답 노드를 더 높은 성능 상태로 빠르고 제어가능하게 전환함으로써 대역폭 결여 문제를 해결하는 것이 유용하다. 일단 버스트(burst) 프로브 활동이 종료되고 여분의 대역폭이 더 이상 필요치 않다면, 응답 노드는 그것의 실행 활용도에 의해 지시되는 이전의 성능 상태로 다시 전환할 수 있다.

[0009] 잠재적인 프로브 응답 병목을 해결하는 하나의 접근법은 시스템 디바이스들을 핸들링(handle)하는 운영체제(OS)나 상위 레벨 소프트웨어가 적절히 프로세서 P-상태를 튜닝(tune)할 수 있는 시스템들에서 소프트웨어 기반의 솔루션이다. 하나의 소프트웨어 기반의 솔루션은 OS나 상위 레벨 소프트웨어가 더욱 빈번히 프로세서 P-상태를 재평가할 것을 요구하며(버스트 활동에 적절히 응답하기 위해) 따라서 임의의 애플리케이션을 이용한 이러한 재평가를 위해 프로세서를 더욱 빈번히 깨운다. 이 접근법은 이러한 빈번한 재평가가 불필요한 경우에도 애플리케이션을 이용하여 더 높은 전력 소비로 이어질 가능성이 있다. OS나 상위 레벨 거동을 더욱 정교하고 애플리케이션-불변(application-invariant)이지 않도록 만드는 것은 유휴 상태 핸들러(idle handler) 또는 루틴에서 추가적인 오버헤드로 이어져서(P-상태 재평가가 규칙으로서 발생) 역시 더 높은 전력 소비로 이어진다. 일반적으로 말하자면, 소프트웨어 기반의 솔루션의 세부 단위(granularity)는 하드웨어 기반의 접근법과의 매칭을 제공하지 못하고 프로브 활동의 시작과 프로브 활동의 종료를 모두 빠르게 파악할 수 없다. 프로세서가 여분의 시간 동안 더 높은 성능 상태에 남겨지면 이는 또한 여분의 전력 소비로 이어져 와트당 성능비를 저하시키기 때문에 전력 절감을 위해 후자(프로브 활동의 종료)를 파악하는 것도 똑같이 중요하다.

[0010] 또 다른 솔루션은 모든 요청 노드들과 응답 노드들에게 공유 전압/클록 층(shared voltage/clock plane)을 제공하는 하드웨어 기반의 솔루션이다. 이러한 하드웨어 구성은 요청 노드(코어)가 그 주파수를 증가시킬 때 응답 노드(코어)의 주파수를 증가시킨다. 응답 노드의 느린 응답들은 요청 노드(코어)의 활용도 증가에 기여할 것이다. 따라서, 요청 노드의 성능 상태를 제어하는 소프트웨어는 요청 노드의 성능 상태를 증가시킬 것이며, 응답 노드 성능 상태도 역시 증가될 것이어서(공유 주파수 전압 층으로 인해), 최종적으로 응답 코어의 프로브 대역폭이 증가된다. 하지만, 이 접근법은 모바일 또는 울트라 모바일 시장 부문에서 가장 전형적인 작업 부하 유형인 애플리케이션들이 오로지 단일 노드(코어)나 소수의 노드들(코어들) 상에서 실행중인 경우에 다중-코어 프로세서들에서 추가 전력을 소비한다. 또한, 소프트웨어는 보통 전형적으로 수백 마이크로초(microsecond) 내지 밀리초(millisecond) 범위의 시간 구간에서 요청 노드(코어)의 활용도 증가로 인한 더 높은 클록 주파수에 대한 요구에 즉시 응답하지 못하며, 이는 이 구간 동안 성능 손실로 이어질 수 있다.

[0011] 따라서, 본 발명의 한 실시예에서, 각 프로세싱 노드는 그것의 프로브 활동을 추적한다. 만일 프로브 활동의 레벨이 임계값을 초과하는 경우, 프로브 활동 대역폭에 대한 증가 요구를 해결하기 위해 프로세싱 노드의 성능 상태는 최소 성능 상태 MinPstateLimit으로 올려진다. 프로브 활동이 임계값에서 관련 히스테리시스(hysteresis)를 뺀 것의 아래로 내려간 후, 프로세싱 노드는 그것의 이전 P-상태가 MinPstateLimit보다 낮은(성능 관점에서) 상황일 때 그것의 이전 성능 상태(P-상태)로 다시 전환한다. 일부 실시예들에서, 히스테리시스 값은 0(zero)일 수 있으며, 다른 실시예에서 고정 값이거나 프로그램 가능한 값일 수 있다는 것을 유의해야 한다.

[0012] 도 2의 흐름도는 본 발명의 한 실시예에 따라 프로브 제어 로직(103)(도 1 참조)에서 동작할 수 있는 예시적인 판단 프로세스를 예시한 것이다. 201에서, 노드는 프로세싱 유닛이 MinPstateLimit보다 더 낮은 성능 상태에 있는지를 판별한다. 만일 더 낮은 상태에 있지 않으면, 현재 성능 상태는 프로브 활동을 핸들링하기에 충분하며 흐름은 201에 머물러 있는다. 만일 현재 성능 상태가 더 낮으면, 203에서 노드는 프로브 활동을 추적한다. 205에서, 만일 프로브 활동이 임계값보다 더 크면, 207에서 노드는 성능 상태를 MinPstateLimit으로 상승시키고 208에서 계속 프로브 활동을 추적한다. 예시의 편의를 위해 성능 상태를 조절하는 제어 로직이 프로브 제어 로직(103)의 일부인 것으로 가정한다는 것을 유의해야 한다. 일부 실시예들에서, 그것은 프로브 제어 로직과 분리될 수 있다. 전압 및 주파수를 사용하여 프로세싱 노드들의 성능 상태들을 제어하는 것은 당해 기술분야에서 잘 알려져 있으므로 본 명세서에서 상세히 서술되지 않을 것이다. 만일 프로브 활동이 임계값에서 히스테리시스 인자를 뺀 것보다 위에 있는 경우, 노드는 프로브 활동에 대처하기 위해 MinPstateLimit에 머무른다. 하지만, 209에서 만일 프로브 활동이 임계값에서 히스테리시스 인자를 뺀 것보다 아래의 레벨로 다시 내려가는 경우, 211에서 노드는 이전 성능 상태(단계 201 및 203의)가 MinPstateLimit보다 낮았었는지 여부를 판별한다. 만일 그렇다면, 213에서 노드는 이전의 더 낮은 성능 상태로 전환한 다음, 현재의 성능 상태가 임계 레벨 위로의 프로브 활동 증가에 대처하기에 적정한지 여부를 판별하기 위해 201로 되돌아간다. 유의할 점은 만일 소프트웨어(또는 하드웨어)에 의해 관리되는 정상 흐름에 의해 프로세싱 노드 활용도 인자에 의거하여 프로세싱 노드의 현재 성능 상태가 MinPstateLimit 이상으로 증가되었다면 211에서 더 낮은 성능 상태로의 전환은 일어나지 않는다는 것이다.

[0013] 도 2에 예시된 실시예는 MinPstateLimit 성능 상태에 의해 해결되는 오직 하나의 프로브 성능 임계값을 포함한다. MinPstateLimit보다 더 높은 임의의 성능 상태(P-상태)는 최악의 프로브 대역폭 요구조건들을 만족시킨다고 가정한다. 하지만, 다른 실시예들은 프로브 대역폭과 관련된 하나 이상의 임계값을 가질 수 있다. 더 높은 프로브 대역폭 요구조건은 더 높은 동작 P-상태가 프로브 대역폭 제약을 해결할 것을 요구한다. 표 1은 서로 다른 프로브 대역폭에 대한 요구조건들에 해당하는 3개의 성능 상태(P-상태)들을 가지는 한 실시예를 예시한 것이다.

표 1

[0014]

P-상태	프로브 활동 임계값	히스테리시스
Pm	ProbActM	HystM
Pn	ProbActN	HystN
Pk	ProbActK	HystK

[0015] P-상태들의 경우, Pm > Pn > Pk 이다. 성능 관점에서, PrbActM > PrbActN > PrbActK 이다. 히스테리시스 값들 HystM, HystN, HystK은 동일할 수도 있고, 각 임계값에 대해 서로 다를 수도 있다. 히스테리시스 값들은 임계값들과 마찬가지로 구성가능할 수 있다.

[0016] 프로브 활동이 (ProbeActivityM - HysteresisM)보다 위에 머무르는 한 프로세싱 노드도 P-상태 Pm에 머무른다. 일단 프로브 활동이 (ProbeActivityM - HysteresisM)보다 아래로 떨어지고 만일 이전 성능 상태(프로브 활동의 증가가 있기 전의 성능 상태)가 Pm보다 더 낮은 경우, 프로세싱 노드는 더 낮은 성능 상태로 전환한다. 유의할 점은 만일 소프트웨어(또는 하드웨어)에 의해 관리되는 정상 흐름에 의해 프로세싱 노드 활용도 인자에 의거하여 프로세싱 노드의 현재 성능 상태가 Pm 이상으로 증가되었다면 더 낮은 성능 상태로의 전환은 일어나지 않는다는 것이다.

[0017] 도 3a와 도 3b는 하나 이상의 프로브 성능 임계값을 갖는 실시예들에 대해 상태간 전환을 예시한 것이며, 각 성능 임계값은 프로브 활동의 서로 다른 레벨에 해당한다. 상태 전환은 프로브 제어 로직(103)(도 1)에 구현될 수 있다. 일단 프로브 활동이 임계값들 중 하나를 초과하면, 응답 노드는 그 프로브 활동의 레벨에 해당하는 P-상태로 전환한다. 이는 더 높은 성능 상태(P-상태)가 요구되는 증가된 프로브 활동 기간을 제외한 모든 시간 동안 유휴 상태의 응답 노드가 최소 성능 상태(또는 심지어 보유 상태(retention state))에 있도록 보장하는 데 도움이 된다. 도 3a를 보면, 주파수 관점에서 Pm(301) > Pn(303) > Pk(305) > 현재 P-상태(307) 이라고 가정한다. 그러면, 만일 P-상태(307)에 있는 동안 더 높은 프로브 활동(Prob_Act) 레벨로 프로브 활동의 증가가 발생한다면, 노드는 후술되는 바와 같이 프로브 활동의 레벨에 따라 P-상태들(Pk, Pm, 또는 Pn) 중 하나로 진입할 수 있다. 다음은 노드가 현재 낮은 전력 상태(307)에 있다고 가정할 때 프로세싱 노드의 전환을 서술한 것이다.

If (Prob_Act > PrbActM), then P-state = Pm

Else If (Prob_Act > PrbActN), then P-state = Pn

[0018] *Else If (Prob_Act > PrbActK), then P-state = Pk*

[0019] 이외에도, 한 실시예에서, 노드는 도 3b에 도시된 바와 같이 P-상태 Pn(303) 또는 Pk(305)에 있을 때 그 다음 상위 레벨의 P-상태로 전환할 수 있다. 만일 노드가 P-상태 Pn(303)에 있는 동안 프로브 활동의 증가를 검출하는 경우(Prob_Act > PrbActM), 노드는 306을 통해 P-상태 Pm(301)으로 전환한다. 만일 노드가 P-상태 Pk(305)에 있는 동안 프로브 활동의 증가를 검출하는 경우(PrbActM > Prob_Act > PrbActN), 노드는 전환(308)을 통해 P-상태 Pn(303)으로 전환한다. 만일 노드가 P-상태 Pk(303)에 있는 동안 프로브 활동의 증가를 검출하는 경우(Prob_Act > PrbActM), 노드는 310을 통해 P-상태 Pm(301)으로 전환한다.

[0020] 실시예의 추가적인 양상은 만일 프로브 활동이 임계값 아래에 있는 경우 유휴 노드의 P-상태를 최소 P-상태로 낮추는 것이다. 만일 프로세싱 노드의 P-상태의 활용도 기반 설정을 담당하는 소프트웨어 또는 하드웨어가 프로세싱 노드를 차선으로 높은 P-상태(MinPstateLimit보다 높은)에 남겨 두었다면, P-상태 제어를 프로브하는 기능은 노드가 전력을 절감하는 동안 버스트가 아닌 프로브 활동이나 더욱 낮은 레벨의 프로브 활동에 여전히 응답할 수 있도록 노드 P-상태를 Pmin(최소 동작 P-상태)로 낮추거나 심지어 보유 전력 상태로 낮출 수 있다. 다음은 감소된 레벨의 프로브 활동(Prob_Act)에 의거하여 도 3a에 도시된 전환을 서술한 것이다.

If (Prob_Act < (PrbActM-HystM) AND Prob_Act > PrbActN AND Current P-state <Pm),

then P-state = Pn

Else If (Prob_Act < (PrbActN-HystN) AND Prob_Act > PrbActK AND Current P-state <Pn),

then P-state = Pk

[0021] *Else If (Prob_Act < PrbActK-HystK AND Current P-state <Pk), then P-state = Current P-state*

[0022] 마찬가지로, 도 3b에 도시된 바와 같이, 한 실시예에서, 노드는 프로브 활동의 감소를 반영하기 위해 하나의 P-상태(303 또는 305)에서 적절한 P-상태로 아래로 전환할 수 있다. 예를 들어, P-상태 Pn(303)에 있는 동안, 노드는 프로브 활동에 따라 P-상태 Pk(305) 또는 현재 P-상태(307)로 전환할 수 있다. 만일 프로브 활동이 (Prob_Act < PrbActN-HystN AND Prob_Act > PrbActK)이도록 감소하는 경우, 노드는 P-상태 Pk(305)로 전환한다. 만일 프로브 활동이 P-상태 Pn(303)에 있는 동안 Prob_Act < PrbActK-HystK 이도록 감소한다면, 노드는 현재 P-상태(307)로 전환한다. 유사하게, 만일 프로브 활동이 P-상태 Pk(305)에 있는 동안 Prob_Act < PrbActK-HystK 이도록 감소한다면, 노드는 현재 P-상태(307)로 전환한다.

[0023] 따라서, 제어 로직은 프로브 활동이 필요로 하는 현재 전력 상태에 맞도록 하기 위해 현재 프로브 활동에 의거하여 전력 상태를 위나 아래로 전환할 것이다. 이는 응답 노드들에서 병목 현상을 방지하면서도 가능한 여전히 전력 절감을 달성하려는 데 도움이 될 수 있다.

[0024] 또 다른 실시예에서, 프로브 활동은 노드가 유휴이고 그것의 프로브 활동이 임계값을 초과할 때 노드의 캐쉬 시스템을 플러쉬(후-쓰기 무효화(write-back invalidate) 및 디스에이블)하는 것을 트리거할 수 있다. 그 접근법은 다중-노드 시스템에 유용하거나 상대적으로 짧은 캐쉬 플러쉬 시간을 갖는 노드들에 유용할 수 있다. 플러쉬할 것인지의 판단은 프로브 활동이 임계값을 초과하는 것(캐쉬 프로브시 응답 노드에 의해 소비되는 전력이 캐쉬 시스템을 플러쉬하는 것과 관련된 전력보다 더 높게 되는 것을 의미)과 노드가 충분한 시간 동안 유휴 상태에 있을 것으로 예측되는 것과 같은 인자들에 의거할 수 있다. 유휴 상태를 예측하는 접근법들은 전형적으로 노스-브리지(North-Bridge)(또는 더욱 일반적으로 프로세서 집적 회로 중에서 프로세서 코어들이 아닌 부분(언코어(Uncore))으로서 전형적으로 메모리 제어기 및 전력 관리와 같은 기능들을 포함하는 부분들)에 있는 활동 추적기(activity tracker) 및 내부 추적기들을 기반으로 예측하는 것을 포함한다. 게다가, 인터럽트, 들어오거나 나가는 전송들, 타이머-틱(timer-tick)들과 같은 I/O 서브시스템 활동 예측들이 또한 유휴 상태 예측에 이용될 수 있으며 별도의 집적 회로(예컨대 사우스-브리지(South-Bridge))를 기반으로 할 수 있다.

[0025] 도 4는 프로브 활동과 노드 유휴 상태 예측에 의거한 캐쉬 플러쉬(cache flushing)의 한 실시예의 예시적인 흐름도를 예시한 것이다. 401에서, 만일 프로세싱 노드가 유휴 상태에 있다면, 402에서 프로세싱 노드는 프로브 활동을 추적하고, 403에서 노드는 프로브 활동이 프로브 임계값보다 큰 지를 검사한다. 만일 크다면, 405에서 흐름은 프로세싱 노드 유휴 상태가 유휴 임계값(idle threshold)보다 큰 것으로 예측되는지를 검사한다. 만일 그렇다면, 407에서 프로세싱 노드는 그것의 캐쉬를 플러쉬하고, 캐쉬 시스템을 디스에이블하고, 보유 전압이나 다른 적절한 전력 절감 전압을 인가하고, 시스템은 노드를 프로브하는 것을 정지한다. 하지만, 노드 유휴 상태의 예측된 기간이 임계값 아래에 있어서 캐쉬 플러쉬가 전력을 절감하지 못하거나 충분히 절감하지 못하기 때문에 매력적이지 못한 경우에, 409에서 P-상태 제어 알고리즘(전술됨)이 적용될 수 있으며, 노드는 계속 프로브 활동을 추적하고 필요하다면 프로브 활동의 레벨에 따라 P-상태를 조정한다.

[0026] 프로브 활동 추적의 한 실시예는 도 5에 도시된 바와 같이 본 명세서에서 인-플라이트 큐(In-Flight Queue, IFQ)로서 지칭되는 큐 구조를 이용한다. IFQ 구조(500)는 프로브 활동 레벨을 논리적으로 반영하는 다중-엔트리 어레이(array)이다. 임의의 트랜잭션(일관성(coherent) 트랜잭션이나 비-일관성(non-coherent) 트랜잭션)(501)이 IFQ의 사용가능한 엔트리에 놓여지고 축출시(eviction point)까지 거기에 존재한다. 응답 노드에 의해 응답된 후에 503에서 트랜잭션은 IFQ로부터 할당 해제(축출)된다. 응답은 데이터 이동을 수반하는 트랜잭션들에 대한 데이터 단계(data phase)(즉, 프로세싱 노드로부터 공유 메모리로 또는 공유 메모리로부터 프로세싱 노드로의 데이터 이동)이거나 또는 데이터 이동이 없는 트랜잭션(즉, 프로세싱 노드의 메모리나 로컬 캐쉬의 캐쉬 엔트리를 무효화하라는 요청)들에 대한 응답 단계(response phase)일 수 있다. IFQ 구조는 프로세싱 노드들 간에 공유될 수 있으며, 또는 프로세싱 노드마다 인스턴스화(instantiate)될 수 있다. 프로브 활동 레벨은 활성 IFQ 엔트리들(완료를 기다리는 미해결 일관성 요청들을 가지는 엔트리들)의 개수에 의해 표현된다.

[0027] 한 실시예에서, 노드(또는 제어 기능들이 존재하는 임의의 부분)는 활성 IFQ 엔트리들의 개수를 단일 임계값(502)과 비교한다. 유의할 점은 제어 기능들은 노드의 내부에 존재하거나 외부에 존재할 수 있다는 것이다. 만

일 노드의 외부에 있는 경우에도, 전술된 바와 같이 그것은 여전히 다이의 언코어 부분과 동일한 다이 상에 존재할 수 있다. 만일 엔트리들의 개수가 임계값을 초과하는 경우, 상위 P-상태(MinPstateLimit)로의 전환이 일어난다. 활성 IFQ 엔트리들의 개수가 임계값에서 히스테리시스를 뺀 것보다 더 낮은 레벨로 떨어진 후에, MinPstateLimit 성능 상태는 취소되고 프로세싱 노드는 더욱 낮은 전력에서 실행되면서 더욱 낮은 프로브 대역폭에 대처할 수 있는 현재 P-상태로 다시 전환한다.

[0028] 다른 실시예들은 도 6에 도시된 다중-레벨 IFQ 기반의 접근법을 이용할 수 있으며, 각 레벨은 서로 다른 프로브 대역폭과 관련된 최소 성능 레벨(P-상태 임계값)을 가진다. 예를 들어, 16-엔트리 IFQ 구조(600)는 P-상태 Pm과 P-상태 Pk에 각각 해당하는 2개의 임계값들(602, 604)을 가질 수 있으며, 이들은 프로브 대역폭의 증가 필요성을 나타낸다. 상태 간 전환은 도 3a 및 도 3b에 도시된 바와 같이 이루어질 수 있다.

[0029] 다른 실시예들에서, 프로브 활동 추적에 대한 다른 접근법들이 사용될 수 있다. 예를 들어, 프로브 요청의 완료 단계(completion phase)가 은닉되거나, 이용불가능하거나, 또는 추적하기 곤란한 시스템들에 있어서, 추적하기 위한 접근법은 서로 다른 증분(increment) 값 및 감소분(decrement) 값을 가지는 프로브 카운트 메커니즘에 의거하여 예측될 수 있다. 예를 들어, 도 7을 보면, 프로세싱 노드로 디스패치되는 새로운 프로브 요청(703)이 식별될 때마다 매번 카운터(701)가 증가된다(CNT = CNT + w_inc). 프로세싱 노드의 특정 P-상태와 관련된 프로브 속도(probing rate)(대역폭)와 매칭되는 구성가능한 시간 구간(IntervalTolerated) 마다 카운터 값이 감소된다(CNT = CNT - w_dec). 한 실시예에서, 구성가능한 시간 구간은 특정 P-상태와 관련된 최대 프로브 대역폭과 매칭된다. 따라서, 실제 응답(데이터 이동 단계, 데이터 이동이 없는 트랜잭션들에 대한 응답 단계)이 추적되지 않더라도 프로브 요청들은 특정한 속도로 서비스된다고 가정할 수 있다.

[0030] 임의의 새로운 프로브 요청들은 카운터를 증가시키며(CNT = CNT + w_inc), w_inc는 카운터의 현재 값에 더해지는 구성가능한 가중치(configurable weight)이다. 일부 실시예들에서, 증분/감소분 값들은 구성가능할 수 있으며 그들의 설정은 고객이나 상위 레벨 소프트웨어 설정(성능 편중 설정, 균형 잡힌 설정, 또는 전력 편중 설정)에 따라 달라질 수 있다. 성능 편중 설정의 경우, w_inc(증분 가중치)는 더 높은 값으로 설정되고 w_dec(감소분 가중치)는 더 낮은 값으로 설정된다. 전력 절감 편중 설정의 경우, 이들 파라미터들은 그와 반대의 방식으로 설정될 수 있다. 또한, IntervalTolerated 값도 고객이나 상위 레벨 소프트웨어의 성능/전력 설정에 따라 구성가능할 수 있다. 카운터 값은 프로브 활동의 레벨을 나타내며 최적 P-상태를 알아내기 위해 ProbeActivity 임계값들과 비교된다. 더 높은 카운터 값은 차례로 현재 P-상태가 만족시킬 수 없는 증가된 프로브 대역폭과 매칭되도록 더 높은 동작 P-상태일 것을 요한다.

[0031] 저역 통과 필터(low pass filter, LPF)(705)는 버스트 프로브 활동을 필터링하여 걸러내는 데 사용될 수 있는데, 버스트 프로브 활동의 경우 작업 부하의 균일성을 적절히 나타내지 못하고 와트당 성능비의 관점에서 차선일 수 있는 성능 상태(P-상태)를 선택하게 하고 카운터의 과도한 증가로 이어질 수 있다. 특정한 실시예에 따라, 구성가능한 개수(1 내지 N)의 프로브 요청들이 구성가능한 구간 T 동안 추적된다. 저역 통과 필터는 프로브 요청들의 출현 빈도가 시간 구간 동안 일정한 구성가능한 한계치를 초과하는 경우 프로브 요청들의 과도한 카운트를 방지하는 서로 다른 방식들로 설계될 수 있다. 예를 들어, 저역 통과 필터는 구간 T 동안 n($1 \leq n \leq N$)개 이하의 프로브 이벤트들을 추적하도록 구현될 수 있다. 따라서, 프로브 이벤트들의 개수가 n을 초과하는 경우, 카운터는 오로지 n을 카운트한다. 저역 통과 필터는 필터링된 프로브 요청들을 카운터로 제공한다.

[0032] 대체가능한 것으로, 저역 통과 필터(705)는 복수의 구간들에 대한 프로브 이벤트들의 개수를 평균하여 특정한 구간 T가 높은 버스트 활동을 가지는 경우 그 높은 버스트는 복수의 구간들에 대한 평균으로 제한되도록 구현될 수 있다. 평균은 예컨대 이동 평균(moving average)으로서 구현될 수 있다. 한 구현예에서, 이동 평균보다 더 높은 속도의 프로브 요청들은 카운터로 제공되지 않는다.

[0033] 저역 통과 필터의 구현은 물론 가중치 w_inc가 어떻게 결정될지에 영향을 줄 수 있다. 따라서, 예를 들어, 만일 다수의 시간 구간들에 대한 평균이 이용되는 경우, 가중치는 그 시간 구간을 반영하도록 스케일링될 수 있다. 다른 실시예들에서, 프로브 요청들은 필터링 없이 직접 카운터로 제공될 수 있다.

[0034] 본 명세서의 실시예들의 양상들은 부분적으로 소프트웨어로 구현될 수 있으며, 상기 소프트웨어는 도 1에 도시된 프로세서와 연계된 휘발성 또는 비휘발성 메모리에 저장된다. 소프트웨어는 컴퓨터 시스템의 비휘발성 부분들에 저장되고, 휘발성 메모리로 로드되어 실행될 수 있다. 따라서, 본 발명의 실시예들은 비휘발성 메모리와 같은 기계 판독가능한 매체(machine-readable medium) 내에 제공되는 기계 실행가능한 명령어들(machine-executable instructions)에 의해 실시되는 특징들 또는 프로세스들을 포함할 수 있다. 이러한 매체는 마이크로프로세서 또는 더욱 일반적으로 컴퓨터 시스템과 같은 기계에 의해 액세스가능한 형태로 데이터를 저장하는 임

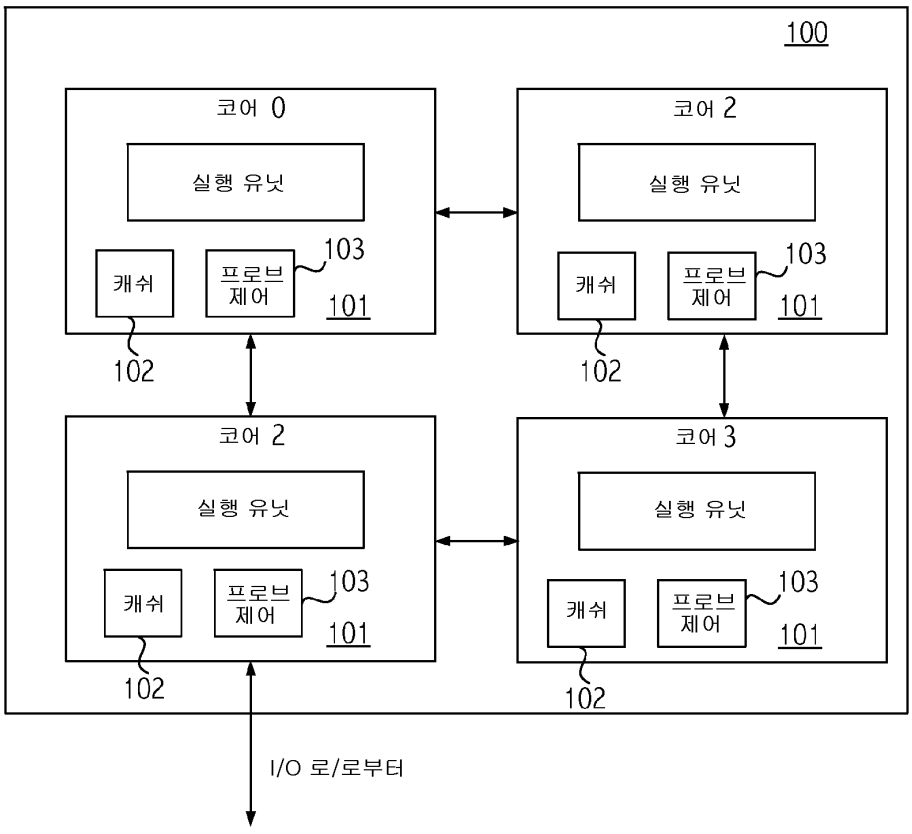
의의 메커니즘을 포함할 수 있다. 기계 판독가능한 매체는 ROM(read only memory), RAM(random access memory), 자기 디스크 저장 매체, 광학 저장 매체, 플래쉬 메모리 디바이스, 테이프, 또는 다른 자기적, 광학적, 또는 전자적 저장 매체와 같은 휘발성 및/또는 비휘발성 메모리를 포함할 수 있다. 이러한 저장된 명령어들은 상기 명령어들로 프로그램되는 범용(general purpose) 또는 전용(special purpose) 프로세서가 본 발명의 프로세스들을 수행하게 하도록 사용될 수 있다.

[0035] 유의할 점은 본 발명의 일부 프로세스들은 하드웨어가 프로그램된 명령어들에 응답하여 동작하는 것을 포함할 수 있다. 대체가능한 것으로, 본 발명의 프로세스들은 상태 머신(state machine)과 같은 하드 와이어드 로직(hard-wired logic)을 포함하는 특정 하드웨어 부분들에 의해 수행되거나 또는 프로그램되는 데이터 프로세싱 부분들과 하드웨어 부분들의 임의의 조합에 의해 수행될 수 있다. 따라서, 본 발명의 실시예들은 본 명세서에서 서술된 바와 같은 소프트웨어, 데이터 프로세싱 하드웨어, 데이터 프로세싱 시스템으로 구현되는 방법들 및 다양한 프로세싱 동작들을 포함할 수 있다.

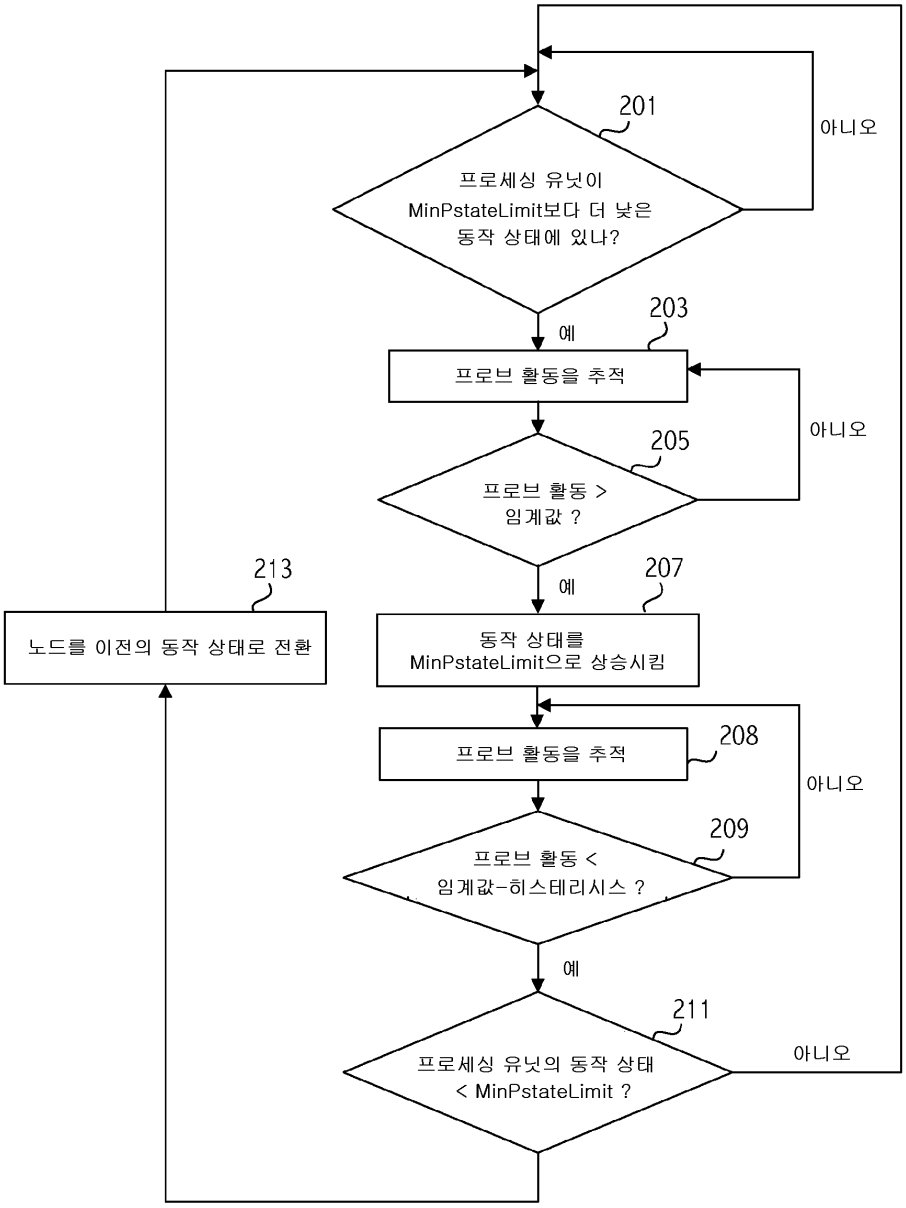
[0036] 따라서, 다양한 실시예들이 서술되었다. 유의할 점은 본 명세서에서 제시된 본 발명의 상세한 설명은 예시적인 것이며, 첨부된 특허청구범위에 의해 제시되는 본 발명의 범위를 제한하고자 의도된 것이 아니라는 것이다. 첨부된 특허청구범위에 의해 제시되는 본 발명의 범위를 벗어남이 없이, 본 명세서에서 제시된 설명을 기반으로 본 명세서에서 개시된 실시예들에 대해 변형 및 수정들이 이루어질 수 있다.

도면

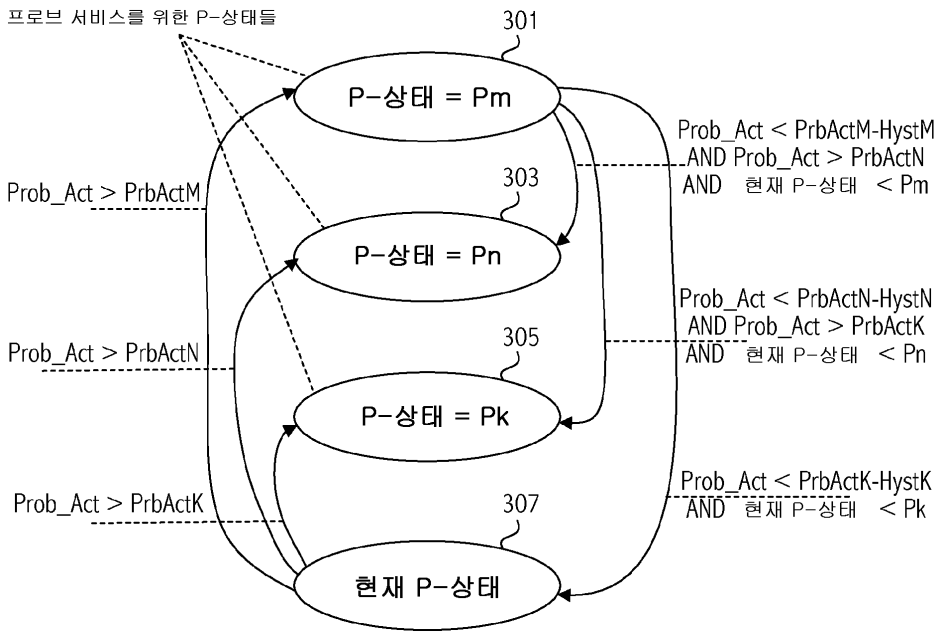
도면1



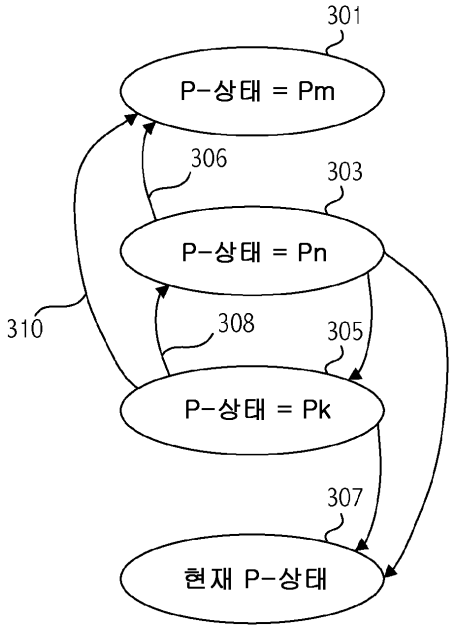
도면2


201
프로세싱 유닛이
MinPstateLimit보다 더 낮은
동작 상태에 있나?
아니오
예
203
프로브 활동을 추적
아니오
205
프로브 활동 >
임계값 ?
213
노드를 이전의 동작 상태로 전환
예
207
동작 상태를
MinPstateLimit으로 상승시킴
208
프로브 활동을 추적
아니오
209
프로브 활동 <
임계값-히스테리시스 ?
예
211
프로세싱 유닛의 동작 상태
< MinPstateLimit ?
아니오

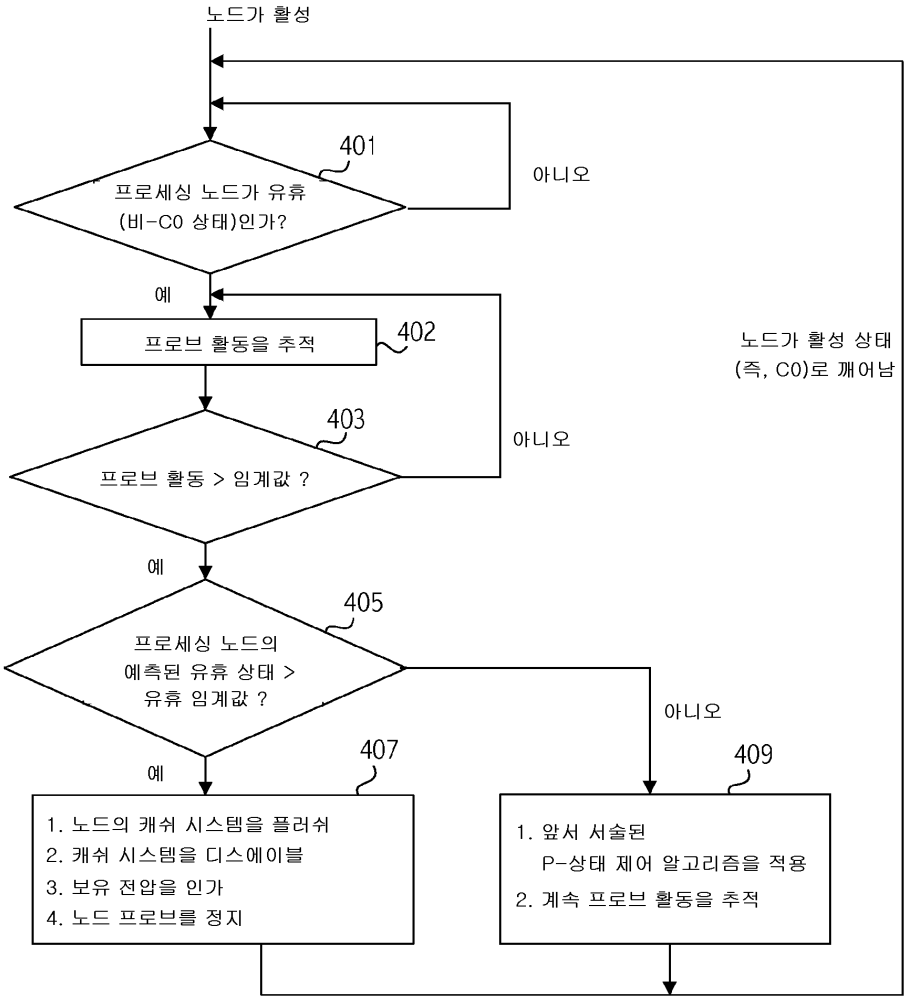
도면3a


프로브 서비스를 위한 P-상태들
301
P-상태 = Pm
303
P-상태 = Pn
305
P-상태 = Pk
307
현재 P-상태
Prob_Act > PrbActM
Prob_Act > PrbActN
Prob_Act > PrbActK
Prob_Act < PrbActM-HystM
AND Prob_Act > PrbActN
AND 현재 P-상태 < Pm
Prob_Act < PrbActN-HystN
AND Prob_Act > PrbActK
AND 현재 P-상태 < Pn
Prob_Act < PrbActK-HystK
AND 현재 P-상태 < Pk

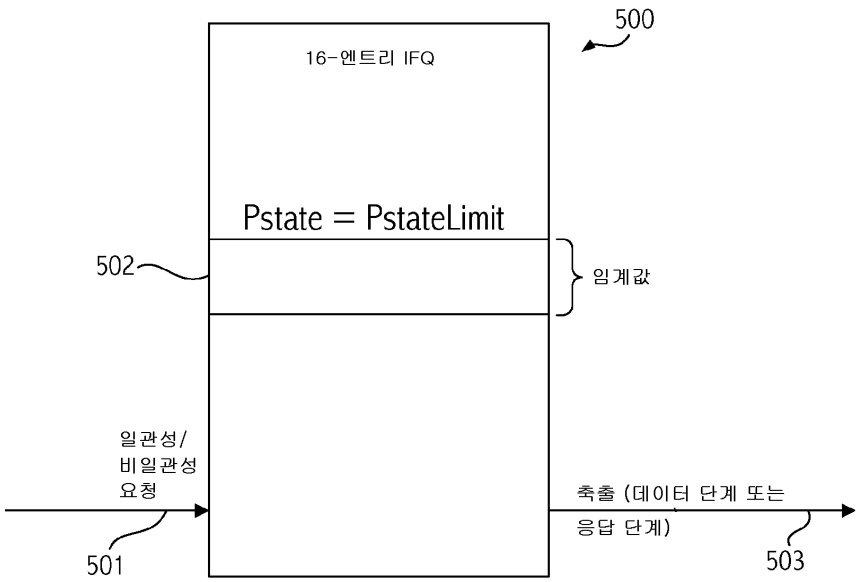
도면3b


301
P-상태 = Pm
306
303
P-상태 = Pn
310
308
305
P-상태 = Pk
307
현재 P-상태

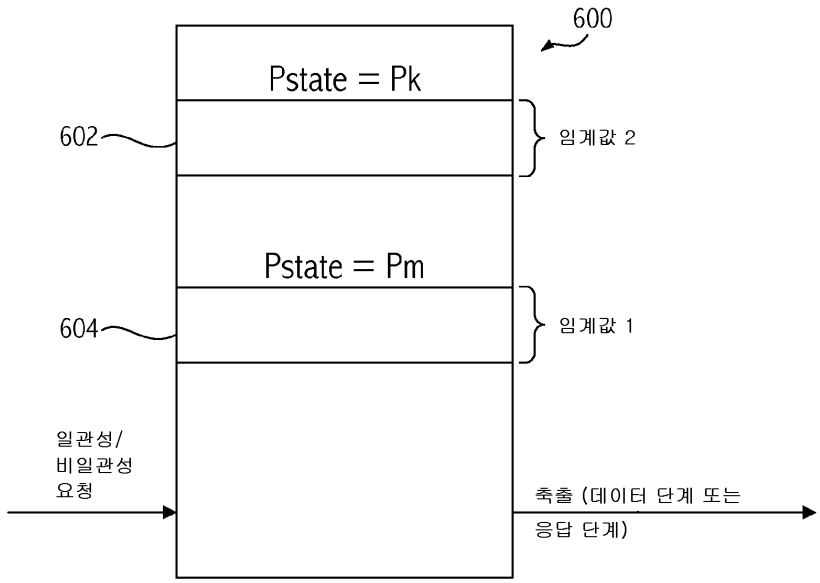
도면4


노드가 활성
401
프로세싱 노드가 유휴
(비-C0 상태)인가?
아니오
예
프로브 활동을 추적
402
노드가 활성 상태
(즉, C0)로 깨어남
403
프로브 활동 > 임계값 ?
아니오
예
405
프로세싱 노드의
예측된 유휴 상태 >
유휴 임계값 ?
아니오
예
407
409
1. 노드의 캐쉬 시스템을 플러쉬
2. 캐쉬 시스템을 디스에이블
3. 보유 전압을 인가
4. 노드 프로브를 정지
1. 앞서 서술된
P-상태 제어 알고리즘을 적용
2. 계속 프로브 활동을 추적

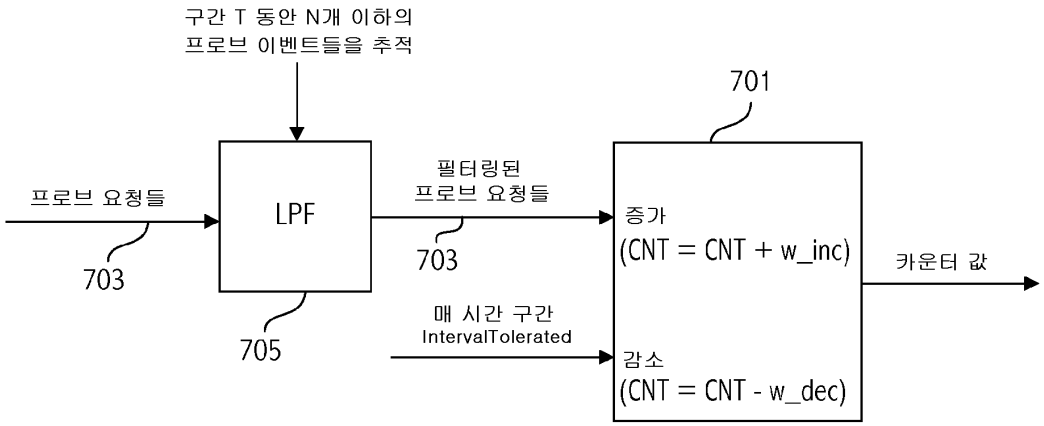
도면5


500
16-엔트리 IFQ
Pstate = PstateLimit
502
임계값
일관성/
비일관성
요청
501
축출 (데이터 단계 또는
응답 단계)
503

도면6


600
Pstate = Pk
602
임계값 2
Pstate = Pm
604
임계값 1
일관성/
비일관성
요청
축출 (데이터 단계 또는
응답 단계)

도면7


구간 T 동안 N개 이하의
프로브 이벤트들을 추적
701
프로브 요청들
703
LPF
705
필터링된
프로브 요청들
703
매 시간 구간
IntervalTolerated
증가
(CNT = CNT + w_inc)
감소
(CNT = CNT - w_dec)
카운터 값