



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2014-0026308
(43) 공개일자 2014년03월05일

(51) 국제특허분류(Int. Cl.)
G06F 1/32 (2006.01) G06F 9/06 (2006.01)
(21) 출원번호 10-2013-0100987
(22) 출원일자 2013년08월26일
심사청구일자 없음
(30) 우선권주장
3504/CHE/2012 2012년08월24일 인도(IN)

(71) 출원인
삼성전자주식회사
경기도 수원시 영통구 삼성로 129 (매탄동)
(72) 발명자
균투르, 라비 상카르
인도, 방갈로르 560093, 비라산드라, 씨 브이 라만 나자르, 반마네 테크 파크, 넘버 66/1, 블록 'B', 반마네 레이크뷰
암바스타, 니티시
인도, 방갈로르 560093, 비라산드라, 씨 브이 라만 나자르, 반마네 테크 파크, 넘버 66/1, 블록 'B', 반마네 레이크뷰
(74) 대리인
이전주, 김정훈

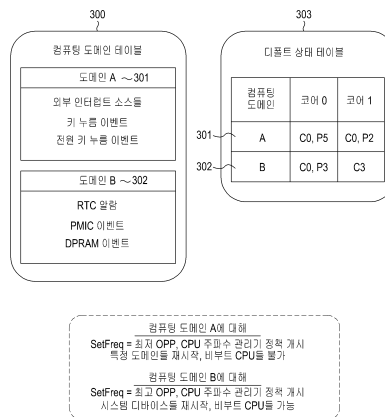
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 **멀티 코어 시스템의 전력 관리 장치 및 방법**

(57) 요약

본 발명의 실시 예는 전자 장치의 부분인 멀티-코어 프로세서의 하나 이상의 코어들을 재시작하는 방법으로서, 웨이크업 소스들을 복수의 컴퓨팅 도메인들로 그룹화하는 과정; 웨이크업 소스와 관련된 인터럽트를 수신하는 과정; 상기 복수의 컴퓨팅 도메인들로부터 상기 웨이크업 소스가 속한 제1 컴퓨팅 도메인을 확인하는 과정; 상기 제1 컴퓨팅 도메인을 상기 프로세서의 제1 코어의 하나 이상의 상태 중 제1 식별자에 매핑하는 과정; 상기 제1 코어가 상기 제1 식별자에 의해 지시된 하나 이상의 상태로 진입하도록 설정하는 과정; 및 상기 제1 코어가 설정된 이후 상기 제1 코어를 재시작하는 과정;을 포함할 수 있다.

대표도 - 도3



특허청구의 범위

청구항 1

전자 장치의 부분인 멀티-코어 프로세서의 하나 이상의 코어들을 재시작하는 방법으로서,
 웨이크업 소스들을 복수의 컴퓨팅 도메인들로 그룹화하는 과정;
 웨이크업 소스와 관련된 인터럽트를 수신하는 과정;
 상기 복수의 컴퓨팅 도메인들로부터 상기 웨이크업 소스가 속한 제1 컴퓨팅 도메인을 확인하는 과정;
 상기 제1 컴퓨팅 도메인을 상기 프로세서의 제1 코어의 하나 이상의 상태 중 제1 식별자에 매핑하는 과정;
 상기 제1 코어가 상기 제1 식별자에 의해 지시된 하나 이상의 상태로 진입하도록 설정하는 과정; 및
 상기 제1 코어가 설정된 이후 상기 제1 코어를 재시작하는 과정;을 포함하는, 멀티 코어 시스템의 전력 관리 방법.

청구항 2

제1항에 있어서, 상기 방법은,
 상기 디바이스가 슬립 모드, 깊은 슬립 모드, 및 저전력 모드 중 적어도 하나에 있을 때, 상기 인터럽트가 수신되는, 멀티 코어 시스템의 전력 관리 방법.

청구항 3

제1항에 있어서, 상기 그룹화하는 과정은,
 상기 복수의 컴퓨팅 도메인들에 대해 컴퓨팅 도메인 테이블을 생성하는 과정을 포함하는, 멀티 코어 시스템의 전력 관리 방법.

청구항 4

제3항에 있어서, 상기 컴퓨팅 도메인 테이블은,
 상기 웨이크업 소스들에서의 인터럽트들을 상기 복수의 컴퓨팅 도메인들로의 맵핑을 포함하는, 멀티 코어 시스템의 전력 관리 방법.

청구항 5

제3항에 있어서, 상기 제1 컴퓨팅 도메인은 상기 컴퓨팅 도메인 테이블에 기반하여 식별되는, 멀티 코어 시스템의 전력 관리 방법.

청구항 6

제1항에 있어서, 상기 방법은,
 상기 복수의 컴퓨팅 도메인들 중 각각을 상기 제1 코어의 하나 이상의 상태들의 상이한 세트와 관련시키는 디폴트 상태 테이블을 생성하는, 멀티 코어 시스템의 전력 관리 방법.

청구항 7

제6항에 있어서, 상기 맵핑하는 과정은,
상기 디폴트 상태 테이블에 기초하여 수행되는, 멀티 코어 시스템의 전력 관리 방법.

청구항 8

제6항에 있어서, 상기 디폴트 상태 테이블은 상기 제1 컴퓨팅 도메인을 상기 프로세서의 상이한 코어들에 대해 상이한 P-상태로 매핑하는, 멀티 코어 시스템의 전력 관리 방법.

청구항 9

제6항에 있어서, 상기 디폴트 상태 테이블은 상기 제1 컴퓨팅 도메인을 상기 제1 코어에 대해 P-상태 및 C-상태의 조합으로 매핑하는, 멀티 코어 시스템의 전력 관리 방법.

청구항 10

제6항에 있어서,
상기 디폴트 상태 테이블은 상기 제1 컴퓨팅 도메인을 상기 제1 코어의 제1 상태로 매핑하며;
상기 디폴트 상태 테이블은 상기 복수의 컴퓨팅 도메인들 중 제2 컴퓨팅 도메인을 상기 제1 상태보다 더 높은 전력 소비에 의해 특성화된 상기 제1 코어의 제2 상태로 매핑하는, 멀티 코어 시스템의 전력 관리 방법.

청구항 11

적어도 하나의 프로세서를 포함하는 집적 회로를 포함하며,
상기 집적 회로는,
웨이크업 소스들을 복수의 컴퓨팅 도메인들로 그룹화하고,
웨이크업 소스와 관련된 인터럽트를 수신하고,
상기 복수의 컴퓨팅 도메인들로부터 상기 웨이크업 소스가 속한 제1 컴퓨팅 도메인을 확인하며,
상기 제1 컴퓨팅 도메인을 상기 프로세서의 제1 코어의 하나 이상의 상태 중 제1 식별자에 매핑하고,
상기 제1 코어가 상기 제1 식별자에 의해 지시된 하나 이상의 상태로 진입하도록 설정하며,
상기 제1 코어가 설정된 이후 상기 제1 코어를 재시작도록 구성되는, 멀티 코어 시스템의 전력 관리 장치.

청구항 12

제11항에 있어서, 상기 장치는,
상기 장치가 슬립 모드, 깊은 슬립 모드, 및 저전력 모드 중 적어도 하나에 있을 때, 상기 인터럽트가 수신되는, 멀티 코어 시스템의 전력 관리 장치.

청구항 13

제11항에 있어서, 상기 그룹화하는 것은,
상기 복수의 컴퓨팅 도메인들에 대해 컴퓨팅 도메인 테이블을 생성하는 것을 포함하는, 멀티 코어 시스템의 전

력 관리 장치.

청구항 14

제13항에 있어서, 상기 컴퓨팅 도메인 테이블은,

상기 웨이크업 소스들에서의 인터럽트들을 상기 복수의 컴퓨팅 도메인들로의 맵핑을 포함하는, 멀티 코어 시스템의 전력 관리 장치.

청구항 15

제13항에 있어서, 상기 제1 컴퓨팅 도메인은 상기 컴퓨팅 도메인 테이블에 기반하여 식별되는, 멀티 코어 시스템의 전력 관리 장치.

청구항 16

제11항에 있어서, 상기 집적 회로는,

상기 복수의 컴퓨팅 도메인들 중 각각을 상기 제1 코어의 하나 이상의 상태들의 상이한 세트와 관련시키는 디폴트 상태 테이블을 생성하도록 더 구성되는, 멀티 코어 시스템의 전력 관리 장치.

청구항 17

제16항에 있어서, 상기 맵핑하는 것은,

상기 디폴트 상태 테이블에 기초하여 수행되는, 멀티 코어 시스템의 전력 관리 장치.

청구항 18

제16항에 있어서, 상기 디폴트 상태 테이블은 상기 제1 컴퓨팅 도메인을 상기 프로세서의 상이한 코어들에 대해 상이한 P-상태로 매핑하는, 멀티 코어 시스템의 전력 관리 장치.

청구항 19

제16항에 있어서, 상기 디폴트 상태 테이블은 상기 제1 컴퓨팅 도메인을 상기 제1 코어에 대해 P-상태 및 C-상태의 조합으로 매핑하는, 멀티 코어 시스템의 전력 관리 장치.

청구항 20

제16항에 있어서,

상기 디폴트 상태 테이블은 상기 제1 컴퓨팅 도메인을 상기 제1 코어의 제1 상태로 매핑하며;

상기 디폴트 상태 테이블은 상기 복수의 컴퓨팅 도메인들 중 제2 컴퓨팅 도메인을 상기 제1 상태보다 더 높은 전력 소비에 의해 특성화된 상기 제1 코어의 제2 상태로 매핑하는, 멀티 코어 시스템의 전력 관리 장치.

명세서

기술분야

본 발명은 멀티 코어 시스템에서의 전력 관리 장치 및 방법에 관한 것이다.

[0001]

배경 기술

- [0002] 통신 및 엔터테인먼트 애플리케이션들의 융합으로, 이동 장치 제조업체들은 더 나은 성능을 제공하면서 전력 소비를 줄여야 하는 과제, 즉 더 적은 전력으로 더 나은 성능을 제공하는 과제를 안고 있다. 전력 관리는 시스템이 비활성 상태일 때 전력을 절약하는 데 도움이 되므로, 배터리 구동식 시스템에서 중요하다.
- [0003] 시스템은 사용되지 않을 경우에 저전력 소비 상태인 '슬립 상태(sleep state)'에 놓인다. 이러한 '슬립 상태'에서 시스템은 사용자를 위한 어떠한 유용한 작업들(tasks)도 수행하지 않는다. 예를 들어, 시스템이 잠시 동안 비활성(inactive)일 때 디스플레이를 오프 상태로 스위칭함으로써 전력을 절약할 수 있다.
- [0004] 기존 시스템들은 C-상태 및 P-상태라고 하는 다양한 프로세서 상태들을 규정함으로써, 효과적인 전력 소비를 다루고 있다. C-상태는 명령을 실행하거나 아이들(idle) 상태인 프로세서에 대응한다. P-상태는 전압 및 주파수 상태 쌍의 세트에 대응한다. 운영 체제는 전력을 절약하기 위해, CPUIDLE 및 CPUFREQ 서브시스템들을 배치한다. 작업 부하에 의거하여, CPUFREQ 서브시스템 구현에서는 시스템에 존재하는 각 코어의 전압 및 주파수를 낮추거나 높인다. 어떤 상황에서는, 코어마다 상이한 C-상태를 적용하도록 선택함으로써 시스템의 전력 수요를 더 줄일 수 있다. 인터럽트에 의한 웨이크업(wakeup) 시, CPUFREQ 서브 시스템은 시스템을 보다 높은 전압 및 주파수 상태에 놓이는 코어들을 갖는 적합한 C-상태(C0)로 천이하도록 한다. 그런 다음, 상기 시스템의 현재 작업 부하에 의거하여 상기 코어들의 전압과 주파수가 더 낮아질 수 있다.
- [0005] 배터리 구동식 휴대 시스템들에서, 종종 대부분의 인터럽트들 및 그 애플리케이션들은 전력을 절약하기 위해 낮은 전압 및 주파수 상태로 서비스될 수 있다. 이러한 시스템에서, 상기 기술들은 코어를 먼저 더 높은 P-상태가 되도록 하고, 이어서 시스템의 현재 작업 로드를 위해 더 적합한 더 낮은 P-상태로 내려가게 한다. 예를 들면, 웨이크업 이벤트의 수신 시, 디바이스는 '슬립 상태'에서 벗어나, 상기 장치에 놓여진 요구에 관계없이 높은 성능 포인트에서 동작할 수 있다. 이후 상기 요구에 의거하여 디바이스는 최적의 성능 포인트에 놓일 수 있다. 그러나, 높은 성능 포인트에서 최적의 성능 포인트로의 천이는 일정 상황에서 비효율적일 수 있다.
- [0006] 마찬가지로, 웨이크업 이벤트의 수신 시, 상기 장치의 2차 코어는 슬립 상태에서 벗어나며, 상기 장치에 놓여진 요구에 기초하여 최적의 성능 포인트 또는 슬립 상태에 놓일 수 있다. 마찬가지로, 이러한 천이는 또한 때때로 비효율적일 수 있다. 따라서, 새롭고 더 효율적인 전력 관리 기술들이 요구된다.

발명의 내용

해결하려는 과제

- [0007] 본 발명의 실시 예들은 웨이크업 이벤트를 식별하고 초래되는 추가 소비 전력을 절약하기 위해 지능적으로 상이한 코어를 사용하는 장치 및 방법을 제공할 수 있다.
- [0008] 또한, 본 발명의 다른 실시 예들은 컴퓨팅 요구에 의거하여 웨이크업 소스를 도메인으로 그룹화하는 장치 및 방법을 제공할 수 있다.

과제의 해결 수단

- [0009] 따라서, 본 발명의 실시 예는 멀티 코어 디바이스의 재시작 시, 멀티 코어 디바이스의 코어의 불필요한 웨이크업을 방지함으로써 전력을 절약하는 방법을 제공하며, 이 방법은 컴퓨팅 요구에 의거하여 웨이크업 소스들을 적어도 하나의 컴퓨팅 도메인으로 그룹화하는 것, 코어 ID들 및 P-상태 정보에 의거하여 적어도 하나의 컴퓨팅 도메인을 구성하는 것, 웨이크업 소스들의 적어도 하나의 인터럽트의 수신 시 정보에 의거하여 적어도 하나의 컴퓨팅 도메인을 획득하는 것, 및 적어도 하나의 인터럽트를 처리하기 위해 코어를 재시작하는 것을 포함한다.
- [0010] 또한, 본 발명의 실시 예는 멀티 코어 디바이스의 재시작 시 멀티 코어 디바이스의 코어의 불필요한 웨이크업을 방지함으로써 전력을 절약하는 디바이스를 제공하며, 디바이스는 적어도 하나의 프로세서를 더 포함하는 집적 회로, 및 집적 회로 내의 컴퓨터 프로그램 코드를 갖는 적어도 하나의 메모리를 포함하고, 적어도 하나의 메모리 및 컴퓨터 프로그램 코드는 적어도 하나의 프로세서로 하여금 디바이스가, 컴퓨팅 요구에 의거하여 웨이크업 소스들을 적어도 하나의 컴퓨팅 도메인으로 그룹화하도록 하고, 코어 ID들 및 P-상태 정보에 의거하여 적어도 하나의 컴퓨팅 도메인을 구성하도록 하고, 웨이크업 소스들의 적어도 하나의 인터럽트의 수신 시 정보에 의거하여 적어도 하나의 컴퓨팅 도메인을 획득하도록 하고, 적어도 하나의 인터럽트를 처리하기 위해 코어를 재시작하게 하도록 구성된다.

[0011] 또한, 본 발명의 실시 예들의 이들 양태 및 다른 양태들은 다음의 설명 및 첨부 도면과 함께 고려되는 경우 보다 용이하게 인식 및 이해될 것이다. 그러나, 다음의 설명은 바람직한 실시 예들 및 그 다수의 특정 상세를 제시하지만, 예시이며 그에 한정되는 것이 아니라는 것을 알아야 한다. 다수의 변형들 및 변경들이 본 발명의 실시 예들의 범위 내에서 그 사상을 일탈하지 않고 이루어질 수 있으며, 본 발명의 실시 예들은 이러한 모든 변경을 포함한다.

발명의 효과

[0012] 본 발명의 실시 예들에 따르면, 지능적으로 상이한 코어를 사용하여 웨이크업 이벤트를 식별하고 초래되는 추가 소비 전력을 절약할 수 있다.

도면의 간단한 설명

[0013] 본 발명은 첨부 도면에서 예시되며, 도면 전반에 걸쳐 동일한 부호는 여러 도면들에서 대응하는 부분을 지시한다. 본 발명의 실시 예들은 도면을 참조하여 다음의 설명에서 보다 잘 이해될 것이다.

- 도 1은 본 발명에 개시된 실시 예들에 따른 디바이스 전력 상태 및 전환을 나타내는 도면이다.
- 도 2는 본 발명에 개시된 실시 예들에 따른 디바이스의 슬립 상태 및 대응하는 웨이크업 소스를 나타내는 도면이다.
- 도 3은 본 발명에 개시된 실시 예들에 따른 컴퓨팅 도메인 테이블 및 디폴트 상태 테이블을 나타내는 도면이다.
- 도 4는 본 발명에 개시된 실시 예들에 따른 일시 중지(suspend) 플로우를 설명하는 흐름도이다.
- 도 5는 본 발명에 개시된 실시 예들에 따른 웨이크업 시퀀스를 설명하는 흐름도이다.
- 도 6은 본 발명에 개시된 실시 예들에 따른 애플리케이션을 구현하는 컴퓨팅 환경을 나타내는 도면이다.

발명을 실시하기 위한 구체적인 내용

[0014] 본 발명의 실시 예들과 그 다양한 특징들 및 유리한 상세들을, 첨부 도면들에 도시되고 이후의 설명에 상세히 기술된 비한정적 실시 예들을 참조하여 더욱 충분하게 설명하기로 한다. 본 발명의 실시 예들을 불필요하게 모호하게 만들지 않도록 공지된 구성 요소들 및 처리 기법들의 설명을 생략하기로 한다. 본 명세서에서 사용되는 예들은 본 발명의 실시 예들을 실시할 수 있는 방안들의 이해를 용이하게 하고 나아가 당업자들로 하여금 본 발명의 실시 예들을 실시할 수 있게 하려고 의도된 것에 지나지 않는다. 따라서 그 예들을 본 발명의 실시 예들의 범위를 한정하는 것으로 해석하여서는 안 된다.

[0015] 본 발명의 실시 예들은 웨이크업 이벤트를 식별하고 전력을 절약하기 위해 지능적으로 디바이스의 다른 코어들 및 그 동작 상태를 이용하는 방법 및 시스템을 제공할 수 있다. 웨이크업 이벤트들은 컴퓨팅 요구에 기반하여 도메인들로 그룹화될 수 있다. 각 컴퓨팅 도메인은 다음으로 컴퓨팅 도메인 테이블 및/또는 디폴트 상태 테이블을 사용하여 하나 이상의 코어 ID, P-상태(Processor performance states) 및 C-상태(Processor operating states)로 매핑될 수 있다. 인터럽트가 디바이스에서 발생될 때, 상기 컴퓨팅 도메인 및 디폴트 상태 테이블들은 상기 코어들의 특정 전압 및 주파수 레벨뿐만 아니라 어느 코어들이 전력 공급될 지를 결정하기 위해 사용될 수 있다.

[0016] 이하 도면들, 보다 구체적으로는 도 1 내지 도 6을 참조하면, 실시 예들이 도시되어 있으며, 여기서 동일한 참조 부호는 도면 전반에 걸쳐 일관되도록 대응하는 특징을 나타낸다.

[0017] 도 1은 배터리 전력 공급 디바이스의 상태도이다. 상기 배터리 전력 공급 디바이스는 이동 전화, PDA(Personal Digital Assistant), 및/또는 임의의 다른 적합한 유형의 디바이스일 수 있다. 도 1은 상기 디바이스의 다양한 전력 상태들을 나타낸다. 이러한 예에서, 상기 디바이스는 노멀(normal) 모드, 저전력(low power) 모드, 슬립(sleep) 모드 및 깊은 슬립(deep sleep) 모드 등의 여러 모드들을 갖는다. 상기 디바이스는 타임아웃 이벤트 시 전력 모드를 낮추도록 전환된다. 상기 디바이스는 임의의 웨이크업 이벤트 시 노멀 상태로 돌아온다. 슬립 모드에서, 디바이스의 모든 코어들이 전력 오프된다. 깊은 슬립 모드에서는, RAM만 전원 온 되지만, CPU 및 다른 주변 기기는 셧다운(shut down)된다. 슬립 모드에서 웨이크업 이벤트가 발생되면, 디바이스는 노멀 모드로 이동된다. 상기 디바이스에 이용 가능한 다른 저전력 상태들의 개수는 상기 디바이스의 종류, 및 하드웨어가 제공할 수 저전력 옵션의 종류에 의존한다. 예를 들면, 상기 디바이스는 비사용 시 슬립 모드에 놓일 것이며, 그런 다

음, 디바이스상에서 키가 눌러질 경우, 디바이스는 노멀 모드로 이동한다.

- [0018] 도 2는 본 발명에 개시된 실시 예들에 따른 디바이스의 슬립 상태 및 대응하는 웨이크업 소스들(sources)을 나타낸다. 상이한 웨이크업 소스들은 복수의 저전력 상태들 중 각각에 대해 보여진다. 도 2에서는 깊은 슬립 모드에서의 웨이크업 소스들은 슬립 모드에 대한 웨이크업 소스의 서브세트이며, 슬립 모드의 웨이크업 소스들은 저전력 모드에 대한 웨이크업 소스들의 서브세트이다. 예를 들면, 디바이스가 깊은 슬립 모드에 있을 때의 웨이크업 소스는 RTC(Real-time clock) 알람, 외부 인터럽트 소스, 키 누름 이벤트, 인터럽트 요청 핸들러를 처리하고 어느 소스로부터 인터럽트가 오는 지를 알아내는 PMIC(Power Management Interrupt Controller) 및 착신 콜 등의 모뎀 인터럽트이다.
- [0019] 도 3은 본 발명에 개시된 실시 예들에 따른 컴퓨팅 도메인 테이블 및 디폴트 상태 테이블의 예를 나타낸다. 상기 도면은 상이한 웨이크업 이벤트들을 도메인 A(301) 및 도메인 B(302)와 같은 각각 분리된 도메인들로 나누는 컴퓨팅 도메인 테이블(300)을 나타낸다. 일부 실시 예들에서, 상기 웨이크업 이벤트들은 상기 이벤트들과 연관된 컴퓨팅 전력 요구들에 기초하여 상이한 도메인들로 그룹될 수 있다. 디폴트 상태 테이블(303)은 상기 도메인 테이블에 정의된 각 도메인을 상기 코어들과 관련된 코어 ID들 및 P-상태로 매핑하도록 구성될 수 있다. 일부 구현들에서, 컴퓨팅 도메인 테이블(300)에서의 도메인 A(301)는 외부 인터럽트 소스들, 키 누름 이벤트 및 전원 키 입력 이벤트 등과 같은 웨이크업 이벤트들을 그룹화할 수 있다. 도메인 B(302)는 RTC 알람, PMIC 이벤트 및 DPRAM 이벤트 등과 같은 웨이크업 이벤트들을 그룹화할 수 있다.
- [0020] 일부 구현들에서, 상기 디폴트 상태 테이블(303)에 따라, 상기 도메인 A에서 임의의 인터럽트들이 발생할 때, 상기 디바이스의 코어 0은 C0 동작 상태 및 P5 프로세서 성능 상태로 놓여지며, 반면 상기 디바이스의 코어 1은 C0 동작 상태 및 P2 프로세서 상태로 놓여진다. 그러한 점에서, 여기에 설명된 기술은 그 이벤트의 유형에 기반하여 특정 웨이크업 이벤트가 발생한 때, CPU들의 C-상태 및 P-상태를 설정한다. 일부 예들에서, 상기 코어 0 및 코어 1에 대한 P-상태는 일부 칩셋들의 설계 제한들 때문에 서로 다를 수도 있고 다르지 않을 수도 있다.
- [0021] 프로세서 성능 상태(P-상태) 및 프로세서의 동작 상태(C-상태)는 상이한 지원되는 동작 주파수와 전압간을 전환하여 전력 소비를 조절하는 프로세서의 능력을 초래한다. ACPI(Advanced Configuration and Power Interface) 규격은 CPU P-상태 전력 관리 상태를 정의한다. P-상태의 수는 프로세서 특성이다. 시스템의 작업 부하에 따라 적절하게 구성될 경우, 이 특징은 전력 절약을 제공한다. 더 높은 P-상태 번호들은 더 느린 프로세서 속도들을 나타낸다. 더 높은 P-상태에서는 전력 소비가 더 낮다. 예를 들면, P3 상태는 P1 상태보다 더 높다. P3 상태의 프로세서는 P1 상태에서 실행되는 프로세서보다 실행 속도가 더 느리고 더 적은 전력을 사용할 것이다. 임의의 P-상태에서 동작하기 위해서, 상기 프로세서는 프로세서가 작동하고 아이들이 아닌 C0 동작 상태에 있어야 한다. 이러한 상태는 구현에 의존하지만, P0은 항상 최고 성능 상태이고, P1 내지 Pn은 순차적으로 더 낮은 성능 상태가 되고, n의 구현 특정 한계는 16 이하이다. P0은 전력 및 주파수가 최대이고, P1은 P0보다 더 낮게 전압/주파수가 스케일링되고, Pn은 P(n-1)보다 더 낮게 전압/주파수가 스케일링된다. 또한, ACPI 규격은 CPU C-상태 전력 관리 상태를 정의한다. CPU의 동작 상태(C-상태)는 전력을 절약하기 위해 비사용 컴포넌트들을 턴오프하는 아이들 프로세서의 능력이다. 프로세서가 C0 상태에서 실행될 때, 그것이 동작하고 있다. 임의의 다른 C-상태에서 실행하는 프로세서는 아이들이다. 더 높은 C-상태 번호들은 더 깊은 CPU 슬립 상태를 나타낸다. 더 높은 C-상태에서, 더 많은 컴포넌트가 전력을 절약하기 위해 셧다운된다. 셧다운되는 일부 컴포넌트들은 프로세서의 클럭을 정지하고 인터럽트를 정지하는 것을 포함한다. CPU 전력 상태(C0 ~ C3)로서, C0는 동작 상태이고, C1(중중 홀트(Halt)로 알려짐)은 프로세서가 명령을 실행하지 않은 상태이지만 본질적으로 실행 상태로 즉시 돌아갈 수 있는 것으로서 정의된다. 모든 ACPI-컴포넌트 프로세서들은 이러한 전력 상태를 지원해야 한다. 펜티엄 4와 같은 일부 프로세서들은 또한 저전력 소비를 위해 강화된 C1 상태(C1E 또는 강화된 홀트 상태)를 지원하고, C2(중중 스톱-클럭(Stop-Clock)으로 알려짐)은 프로세서가 모든 소프트웨어 보기 가능한 상태를 유지하는 상태이지만, 웨이크업하는 데 시간이 더 걸릴 수 있다. 이 프로세서 상태는 선택적이며, C3(중중 슬립으로 알려짐)은 프로세서가 캐시 일관성(cache coherent)을 유지할 필요가 없는 상태이지만, 다른 상태를 유지한다. 일부 프로세서들은 프로세서를 웨이크업하는 데 걸리는 시간이 상이한 C3 상태에 대한 변화(깊은 슬립, 더 깊은 슬립 등)를 갖는다. 이 프로세서 상태는 선택적이다.
- [0022] 일부 구현들에서, 상기 디바이스는 컴퓨팅 도메인 A(301)에서 상기 디바이스가 CPU0의 최고 OPP(Operating Performance Point)를 가지고, 모든 코어가 동작할 수 있게 하는 방식으로 구성될 수 있다. 컴퓨팅 도메인 B(302)에서, 상기 디바이스는 CPU0의 최저 OPP를 가지며, 다른 코어들 중 일부가 동작하지 않게 할 수 있다.
- [0023] 언급한 바와 같이, 일부 구현들에서, 상기 디바이스가 특정 웨이크업 이벤트들을 수신했을 때, 슬립 모드에서

웨이크업 하도록 설정될 수 있다. 일부 구현들에서, 웨이크업 이벤트들은 외부 인터럽트 소스들, RTC 알람, 키 누름 이벤트, PMIC 이벤트, DPRAM 이벤트 등에 의해 생성될 수 있다. 예를 들면, 상기 디바이스는 RTC 알람 이벤트가 발생한 경우에만 슬립 모드에서 웨이크업하도록 구성될 수 있다.

[0024] 도 4는 본 발명에 개시된 실시 예들에 따른 방법(400)의 흐름도이다. 도시된 바와 같이, 상기 방법은 CPU의 현재 주파수를 저장하고 CPU는 슬립 주파수(SLEEP FREQ)를 디폴트 주파수로서 설정된다(401). 또한, 상기 방법은 CPU에 대해 전압을 설정한다. 이어서, 상기 방법은 하나 이상의 디바이스들의 상태를 저장하고, 그 디바이스들은 일시 중지된다(402). 이어서, 상기 방법은 비부팅(non-booting) CPU들 및 시스템 디바이스들을 중단시킨다(403). 이어서 상기 방법은 웨이크업 소스들을 설정한다(404). 그런 다음, 상기 방법은 마스터 CPU를 일시 중지 로 설정하거나 내부적으로 CPU_일시 중지를 호출할 수 있다(405). 상기 방법 400에서의 다양한 동작들이 제시된 순서, 다른 순서 또는 동시에 수행될 수 있다. 또한 일부 실시 예에서, 도 4에 열거된 몇 가지 동작은 생략될 수 있다.

[0025] 도 5는 본 발명에 개시된 실시 예들에 따른 방법(500)의 흐름도이다. 도시된 바와 같이, 상기 방법은 새로운 웨이크업 이벤트를 수신(501)함으로써 시작한다. 이어서 상기 방법은 마스터 CPU(CPU 0-부트 코어)를 웨이크업한다(502). 그런 다음, 상기 방법은 컴퓨팅 도메인 테이블(300)을 이용함으로써 상기 이벤트가 속한 컴퓨팅 도메인(예컨대, 도메인(301) 또는 도메인 B(302))을 획득한다(503). 또한, 상기 방법은 획득된 컴퓨팅 도메인에 대한 디폴트 상태를 디폴트 상태 테이블(303)에 의해 판독한다(504). 이어서 상기 방법은 디폴트 상태 테이블(303)에서 확인된 상기 프로세서의 코어들을 상기 디폴트 상태 테이블(303)에서 확인된 그 코어들을 위한 각 상태로 가져온다(505). 상기 방법은 다음으로 이벤트에 대해 필요한 코어를 재시작하고 다른 디바이스는 일시 중지 상태로 유지될 것이다(506). 마지막으로 상기 방법은 상기 이벤트를 처리한다(507). 방법(500)의 다양한 동작들은 제시된 순서, 다른 순서 또는 동시에 수행될 수 있다. 또한 일부 구현들에서, 도 5에 열거된 일부 동작이 생략될 수 있다.

[0026] 제안된 방법을 이용하여 절약된 추가 전력은 다음과 같이 계산될 수 있다.

[0027]
$$\text{에너지} = N_i * N_c * P_c * T$$

[0028] 여기에서,

[0029] N_i = 선택된 인터럽트가 발생하는 횟수,

[0030] N_c = 디바이스에서 이용 가능한 코어들의 수,

[0031] P_c = 슬립 주파수 (성능) 값에서 실행하는 각 코어에 의해 소비되는 전력,

[0032] T = 올바른 주파수 또는 온라인 CPU 수를 식별하도록 반응하는데 관리기가 걸린 시간이다.

[0033] 도 6은 본 발명에 개시된 실시 예들에 따른 기술들을 구현하는 컴퓨팅 환경을 나타낸다. 도시되어 있는 바와 같이, 컴퓨팅 환경(600)에서는 제어 유닛(611) 및 ALU(Arithmetic Logic Unit)(612), 메모리(640), 저장소(650), 복수의 네트워킹 디바이스들(620), 및 복수의 입출력(I/O) 디바이스들(630)을 구비하는 적어도 하나의 처리 유닛(610)을 포함한다. 처리 유닛(610)은 알고리즘 명령을 처리하는 것을 담당한다. 처리 유닛(610)은 처리를 수행하기 위해, 제어 유닛(611)으로부터 명령을 수신한다. 또한 명령의 실행에 관련된 모든 논리 및 산술 연산은 ALU(612)의 도움으로 계산된다.

[0034] 전체 컴퓨팅 환경(600)은 다수의 동종 및/또는 이종의 코어, 상이한 종류의 다수의 CPU, 특수 미디어 및 다른 액셀레이터로 구성될 수 있다. 처리 유닛(610)은 알고리즘의 명령의 처리를 담당한다. 처리 유닛(610)은 처리를 행하기 위해, 제어 유닛(611)으로부터 명령을 수신한다. 또한, 명령의 실행에 관련된 모든 논리 및 산술 연산은 ALU(612)의 도움으로 계산된다. 또한, 복수의 처리 유닛(610)들은 단일 칩에 또는 다수의 칩에 걸쳐 위치될 수 있다.

[0035] 구현에 필요한 명령 및 코드로 이루어진 알고리즘은 메모리(640) 또는 저장소(650) 중 어느 하나 또는 모두에 저장된다. 실행 시, 명령은 대응하는 메모리(640) 및/또는 저장소(650)로부터 페치(fetch)되어 처리 유닛(610)에 의해 실행될 수 있다.

[0036] 임의의 하드웨어 구현의 경우에, 다양한 네트워킹 디바이스(620) 또는 외부 I/O 디바이스(630)가 컴퓨팅 환경에 연결되어 네트워킹 유닛 및 I/O 디바이스 유닛을 통해 구현을 지원할 수 있다.

[0037] 본 발명에 개시된 실시 예들은 적어도 하나의 하드웨어 디바이스에서 실행되고 요소를 제어하는 네트워크 관리

기능을 수행하는 적어도 하나의 소프트웨어 프로그램을 통해 구현될 수 있다. 도 6에 나타난 요소는 하드웨어 디바이스 및 소프트웨어 모듈 중 적어도 하나 또는 그 조합일 수 있는 블록을 포함한다.

[0038] 전술한 특정 실시 예들의 설명은 다른 사람들이 현재의 지식을 적용하여 일반적 사상으로부터 벗어남이 없이 그러한 특정 실시 예들을 다양한 적용들을 위해 손쉽게 변경 및/또는 개조할 수 있을 정도로 본 발명의 실시 예들의 일반적 특성을 매우 충분히 제시하고 있고, 그에 따라 그러한 변경들 및 개조들을 개시된 실시 예들의 균등물의 의미 및 범위 내에 포함하여야 하고 또한 포함하고자 한다. 본 명세서에 채용된 표현들 및 용어들은 설명을 위한 것이지 한정을 위한 것이 아님을 알아야 할 것이다. 따라서 본 발명의 실시 예들을 바람직한 실시 예들의 측면에서 설명하였으나, 당업자라면 본 발명의 실시 예들이 본 명세서에 설명된 실시 예들의 사상 및 범위 내에서 변경되어 실시될 수 있는 것임을 인지할 것이다.

[0039] 본 개시의 장치들 및 방법들은 하드웨어 또는 펌웨어에서 실행되거나 또는 하드웨어 또는 펌웨어 또는 그 조합들에 의해 실행되는 소프트웨어 또는 컴퓨터 코드로 구현될 수 있다. 컨트롤러, 중앙 처리 유닛(CPU), 프로세서, 및 본 개시의 임의의 유닛 또는 디바이스는 적어도 하드웨어 및/또는 다른 물리적 구조 및 엘리먼트들을 포함한다. 추가로, 상기 소프트웨어 또는 컴퓨터 코드는 또한 삭제할 수 있거나 재기록할 수 있는 또는 그렇지 않은 CD ROM, RAM, 판독 전용 메모리(Read Only Memory; ROM)과 같은 비-일시적 저장매체의 형태로, 플로피 디스크, CD, DVD, 메모리 칩들, 하드 디스크, 자기 스토리지 매체, 광학 기록 매체, 또는 자기-광학 디스크 또는 원격 기록 매체, 컴퓨터 판독 가능 기록 매체 또는 비-일시적 기계 판독 가능 매체에 기본적으로 저장된 네트워크를 통해 다운로드되어 로컬 기록 매체에 저장될 수 있으며, 그에 의해, 본 개시의 방법들이 범용 컴퓨터, 디지털 컴퓨터 또는 특정 프로세서 또는 ASIC 또는 FPGA와 같은 프로그램 가능한 또는 전용의 하드웨어에서 저장되는 소프트웨어, 컴퓨터 코드, 소프트웨어 모듈들, 소프트웨어 오브젝트들, 명령들, 애플리케이션들, 애플릿(applet)들, 앱들(apps)등에 렌더링될 수 있다. 당 업계에 이해될 수 있는 바와 같이, 컴퓨터, 프로세서, 마이크로프로세서 컨트롤러 또는 프로그램 가능한 하드웨어는 일시적 및/또는 비-일시적 스토리지 및 메모리 컴포넌트들, 예컨대, 그에 의해 상기 컴퓨터, 프로세서 또는 하드웨어에 의해 액세스되고 실행될 때, 본 개시의 프로세싱 방법들을 구현하는 소프트웨어 또는 컴퓨터 코드를 저장 또는 수신할 수 있는 일시적 및/또는 비-일시적 스토리지 및 메모리 컴포넌트들, 예컨대, RAM, ROM, 플래시 등을 포함한다. 추가로, 본 개시의 프로세싱을 구현하기 위한 코드를 액세스할 때, 상기 코드의 실행은 범용 목적 컴퓨터를 본 개시의 프로세싱을 실행하기 위한 특정 목적 컴퓨터로 변환시킨다. 또한, 상기 프로그램은 무선/유선 접속들에 의해 전송된 통신 신호들 및 그 등가물들과 같은 임의의 매체를 통해 전기적으로 전달된다. 상기 프로그램 및 컴퓨터 판독 가능 기록 매체는 또한 상기 컴퓨터 판독 가능 코드가 분산된 방법으로 저장 및 실행되기 위해 네트워크-커플링된 컴퓨터 시스템들에 분산될 수 있다.

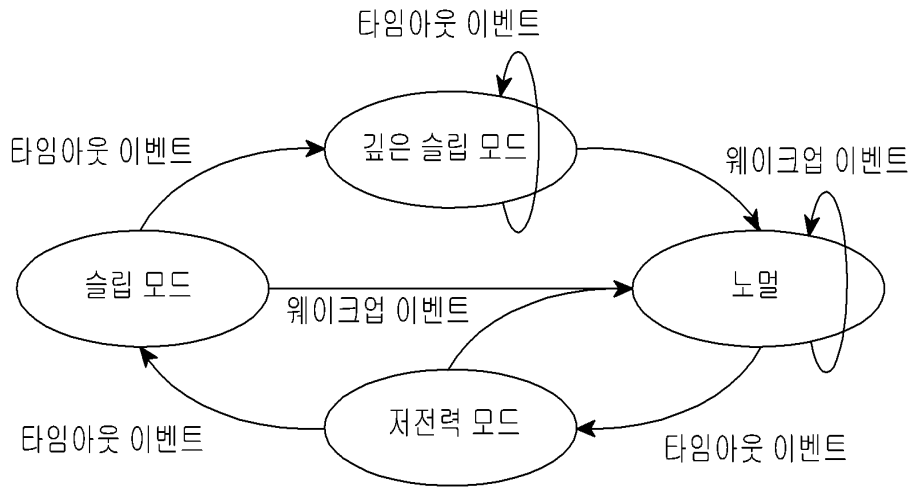
[0040] 비록 본 개시가 논의되었으나, 다양한 변경들이 본 개시로부터 벗어나지 않고 본 개시에 포함될 수 있다. 그러므로, 상기 개시는 상기 실시 예들로 제한되지 않으며 첨부된 청구항들 및 그 등가물들에 의해 정의된다.

부호의 설명

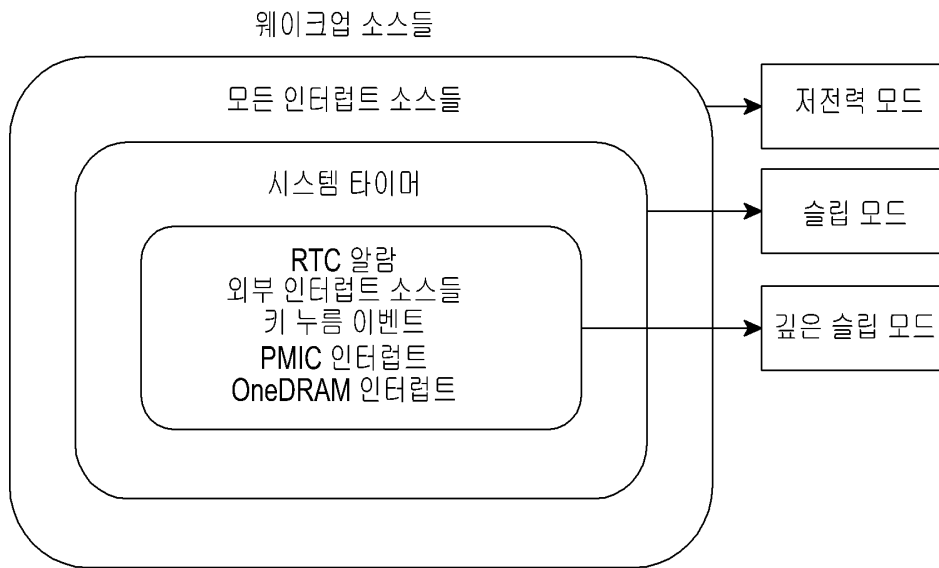
- [0041]
- | | |
|-------------------|------------------|
| 300 : 컴퓨팅 도메인 테이블 | 303 : 디폴트 상태 테이블 |
| 600 : 컴퓨팅 환경 | 610 : 처리 유닛 |
| 611 : 제어 유닛 | 612 : ALU |
| 620 : 네트워크 디바이스들 | 630 : I/O 디바이스들 |
| 640 : 메모리 | 650 : 저장소 |

도면

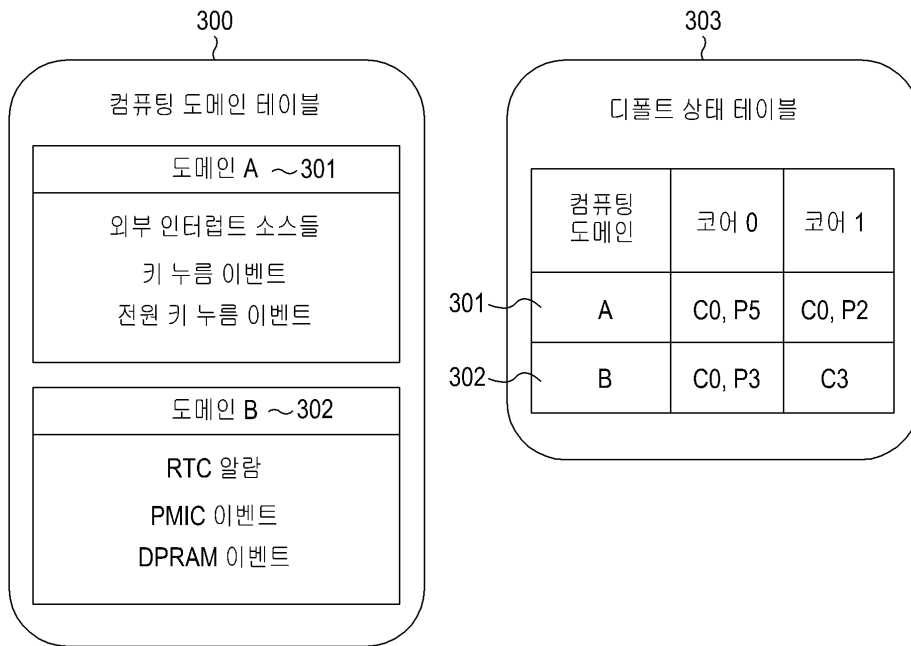
도면1



도면2



도면3



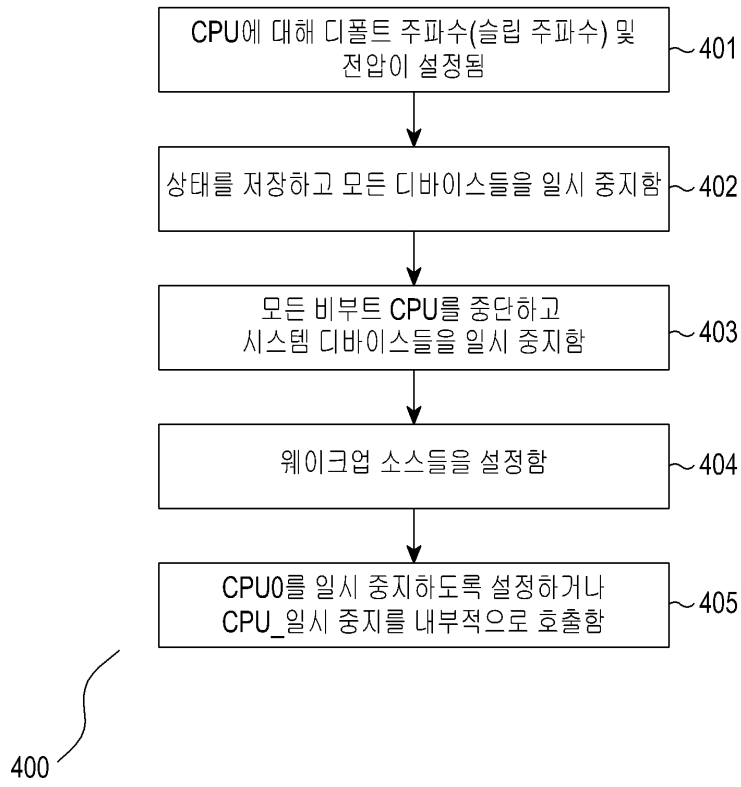
컴퓨팅 도메인 A에 대해

SetFreq = 최저 OPP, CPU 주파수 관리기 정책 개시
특정 도메인들 재시작, 비부트 CPU들 불가

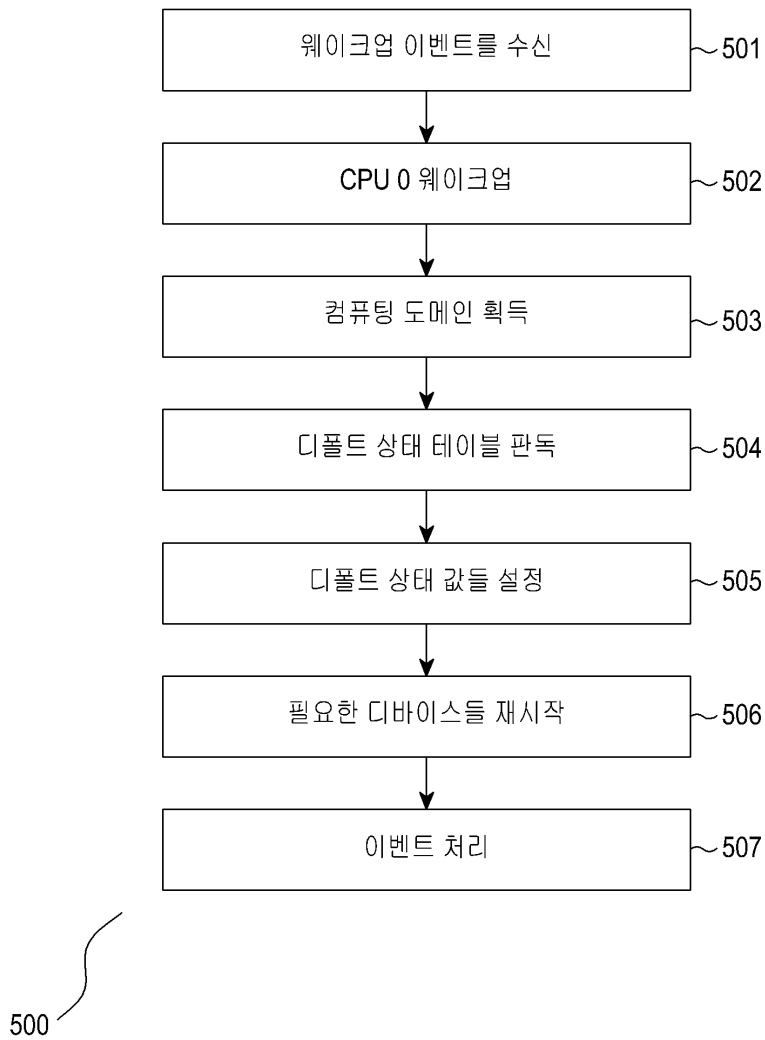
컴퓨팅 도메인 B에 대해

SetFreq = 최고 OPP, CPU 주파수 관리기 정책 개시
시스템 디바이스들 재시작, 비부트 CPU들 가능

도면4



도면5



도면6

