



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2015년12월04일

(11) 등록번호 10-1574433

(24) 등록일자 2015년11월27일

(51) 국제특허분류(Int. Cl.)

H04N 5/63 (2006.01)

(21) 출원번호 10-2014-0113862

(22) 출원일자 2014년08월29일

심사청구일자 2014년08월29일

(56) 선행기술조사문헌

KR100772502 B1

JP2004153553 A

KR100479329 B1

KR101042614 B1

(73) 특허권자

인하대학교 산학협력단

인천광역시 남구 인화로 100, 인하대학교 (용현동)

(72) 발명자

송민석

서울 서초구 남부순환로 2311-12, 107동 402호 (방배동, 래미안방배아트힐)

박진한

경기 부천시 원미구 장말로 71, 1504동 101호 (상동, 한아름아파트)

정현미

인천 남구 매소홀로438번길 8, 2동 304호 (학익동, 동진아파트)

(74) 대리인

양성보

전체 청구항 수 : 총 5 항

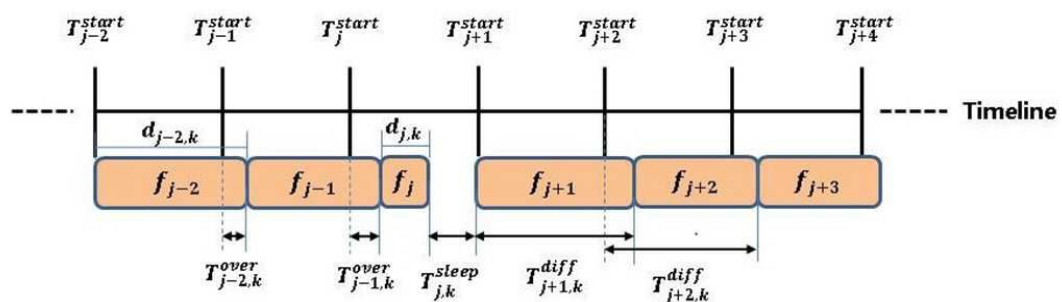
심사관 : 김희주

(54) 발명의 명칭 버퍼링을 이용한 비디오 재생 시 최적의 CPU 주파수를 선택하는 동적 전압 조절 방법 및 장치

(57) 요약

버퍼링을 이용한 비디오 재생 시 최적의 CPU 주파수를 선택하는 동적 전압 조절 방법 및 장치가 개시된다. 버퍼링을 이용한 비디오 재생 시 최적의 CPU 주파수를 선택하는 동적 전압 조절 방법에 있어서, 각 프레임을 디코딩하기 이전에, CPU에 부과되는 상기 프레임의 작업량을 데드라인 전에 처리할 수 있는 주파수를 선택하는 단계; 선택된 상기 주파수의 전력을 측정하는 단계; 및 측정된 상기 주파수의 전력을 이용하여 상기 프레임의 디코딩 시 에너지 소비를 최소화 하는 단계를 포함한다.

대표도 - 도1



이 발명을 지원한 국가연구개발사업

과제고유번호 2012R1A1A2005787

부처명 교육부

연구관리전문기관 한국연구재단

연구사업명 기본연구지원사업

연구과제명 멀티미디어 응용에서 사용자 인지 품질 향상을 고려한 시스템 레벨 전력 최적화 기법에 관한 연구

기 여 율 1/1

주관기관 인하대학교 산학협력단

연구기간 2012.05.01 ~ 2015.04.30

명세서

청구범위

청구항 1

버퍼링을 이용한 비디오 재생 시 최적의 CPU 주파수를 선택하는 동적 전압 조절 방법에 있어서,
 각 프레임을 디코딩하기 이전에, CPU의 동작 전압과 주파수를 동적으로 변화시키는 프레임 레벨(frame-level)의 동적 전압 조절(DVS: dynamic voltage scaling)을 이용하여 상기 CPU에 부과되는 상기 프레임의 작업량을 디코딩 시간의 테드라인 전에 처리하는 주파수를 선택하는 단계; 및
 선택된 상기 주파수를 이용하여 상기 프레임의 디코딩 시 에너지 소비를 최소화 하는 단계를 포함하고,
 상기 프레임의 작업량을 디코딩 시간의 테드라인 전에 처리하는 주파수를 선택하는 단계는,
 버퍼에 미리 디코딩한 상기 프레임을 저장하는 버퍼링을 이용하는 것을 특징으로 하는 버퍼링을 이용한 비디오 재생 시 최적의 CPU 주파수를 선택하는 동적 전압 조절 방법.

청구항 2

제1항에 있어서,
 상기 프레임의 작업량을 디코딩 시간의 테드라인 전에 처리하는 주파수를 선택하는 단계는,
 선택된 상기 주파수를 다음 상기 프레임이 되기 전까지 변경하지 않는 것을 특징으로 하는 버퍼링을 이용한 비디오 재생 시 최적의 CPU 주파수를 선택하는 동적 전압 조절 방법.

청구항 3

제1항에 있어서,
 상기 프레임의 디코딩 시 에너지 소비를 최소화 하는 단계는,
 주파수 선택 알고리즘(FSA: frequency selection algorithm)을 이용하여 에너지 소비를 최소화 하는 상기 주파수를 선택하는 것을 특징으로 하는 버퍼링을 이용한 비디오 재생 시 최적의 CPU 주파수를 선택하는 동적 전압 조절 방법.

청구항 4

제1항에 있어서,
 상기 프레임의 디코딩 시 에너지 소비를 최소화 하는 단계는,
 상기 프레임을 디코딩하는데 필요한 최소의 에너지 소비량, 상기 프레임의 디코딩 초과시간, 및 상기 프레임의 디코딩 시간을 초기화 하여, 전압과 상기 주파수를 동적으로 변화시켜 전력 소비를 최소화 하는 상기 주파수를 선택하는 알고리즘을 형성하는 것을 특징으로 하는 버퍼링을 이용한 비디오 재생 시 최적의 CPU 주파수를 선택하는 동적 전압 조절 방법.

청구항 5

버퍼링을 이용한 비디오 재생 시 최적의 CPU 주파수를 선택하는 동적 전압 조절 장치에 있어서,
 각 프레임을 디코딩하기 이전에, CPU의 동작 전압과 주파수를 동적으로 변화시키는 프레임 레벨(frame-level)의 동적 전압 조절(DVS: dynamic voltage scaling)을 이용하여 상기 CPU에 부과되는 상기 프레임의 작업량을 디코딩 시간의 테드라인 전에 처리하는 주파수를 선택하는 주파수 선택부; 및

선택된 상기 주파수를 이용하여 상기 프레임의 디코딩 시 에너지 소비를 최소화 하는 디코딩부를 포함하고,

상기 주파수 선택부는,

선택된 상기 주파수를 다음 상기 프레임이 되기 전까지 변경하지 않고, 버퍼에 미리 디코딩한 상기 프레임을 저장하는 것

을 특징으로 하는 버퍼링을 이용한 비디오 재생 시 최적의 CPU 주파수를 선택하는 동적 전압 조절 장치.

발명의 설명

기술 분야

- [0001] 본 발명은 버퍼링을 이용한 비디오 재생 시 최적의 CPU 주파수를 선택하는 동적 전압 조절 방법 및 장치에 관한 것이다. 더욱 상세하게는, CPU의 동작 전압과 주파수를 동적으로 변화시키는 동적 전압 조절(DVS)을 사용하여 CPU 전력 소비를 줄이는 버퍼링을 이용한 비디오 재생 시 최적의 CPU 주파수를 선택하는 동적 전압 조절 방법 및 장치에 관한 것이다.

배경 기술

- [0002] 소비자 전자 제품의 발전으로 사람들이 휴대용 장치를 사용하여 언제든지 비디오 서비스를 이용할 수 있게 되었다. 그러나, 비디오 응용은 많은 계산량을 요구하며, 이는 많은 CPU 에너지 소모로 이어진다. 전력 소비를 줄일 수 있는 효과적인 방법은 동적 전압 조절 기법(DVS)을 사용하는 것이다. 프로세서에서 소모되는 에너지는 공급되는 전력과 함께 2차적으로 조절되기 때문에, DVS를 이용하면 높은 에너지 감소효과를 볼 수 있다. 하지만, CPU의 주파수를 낮추는 것은 프로그램의 실행속도를 느려지게 한다.
- [0003] 비디오 품질의 손실이 일어나지 않기 위해서, 각 프레임은 재생시간 안에 디코딩과 렌더링이 되어야 한다. 따라서, 프레임 디코딩 시 데드라인을 만족하는 적절한 주파수를 선택하는 것이 중요하다. 동영상 재생 전에 여러 개의 프레임을 디코딩하여 버퍼에 저장해두는 버퍼링 기술을 이용하면 전력 오버헤드를 분할할 수 있고, 각 프레임의 데드라인이 늘어나기 때문에 CPU를 낮은 주파수에서 동작시킬 수 있다. 이러한 버퍼 비디오를 재생에 있어서, 에너지 소비를 감소시키면서도 버퍼의 한계를 초과하지 않는 주파수를 선택하여야 한다. 미디어 플레이어에서 대부분의 DVS 방법은 소비전력을 줄이면서 디코딩 시간 데드라인을 만족하는 가장 낮은 주파수를 선택한다. 하지만, 이는 최적화되지 않은 방법이고 에너지 소비를 줄일 수 있는 방법이 더 존재한다.
- [0004] 우리는 프레임 디코딩 시 특정 버퍼 요구를 만족하면서 에너지소비를 최소화할 수 있는 주파수를 선택하는 최적화 알고리즘을 제안한다.
- [0005] CPU 전력 관리 기법은 많은 연구의 주제가 되어왔으며 기법들의 결과 대부분은 동적 전력 관리(DPM: dynamic power management) 또는 동적 전압 관리(DVS: dynamic voltage management)로 나눌 수 있다. DPM은 DVS가 활동 상태인 CPU의 전압과 주파수를 줄이는데 반하여 활동 상태에서 높은 주파수를 선택하여 작업을 빨리 끝낸 후 데드라인까지 유휴상태의 CPU를 슬립(sleep) 모드로 변환 시키는 것이다. 일반적으로, DPM은 CPU의 슬립 모드로 진입을 허용하기엔 유휴 간격이 짧은 실시간 응용에 적합하지 않다.
- [0006] 범용 시스템을 위한 DVS 기법은 인터벌(interval) 기반과 태스크(task) 기반의 알고리즘으로 분류될 수 있다. 인터벌 기반의 기법은 CPU 작업량을 인터벌마다 관찰하고 CPU 주파수와 전압의 변화에 의해 반응한다. 대표적인 예로는 리눅스의 온디맨드 정책(on-demand-governor)은 이전 인터벌에서의 CPU 이용량을 토대로 CPU 주파수 단계에 적용시킨다. 인터벌 기반의 또 다른 DVS 기법인 LongRun은 측정된 하드웨어 이용량에 맞도록 시스템 전체 주파수를 바꾼다.
- [0007] 이 방법들은 쉽게 구현할 수 있지만 미래의 CPU 작업량을 자동적으로 최근의 작업량과 유사하다고 가정하기 때문에 예측이 정확하지 않을 수 있다.
- [0008] 태스크 기반의 기법은 서로 다른 주파수 선택 정책을 적용된 타입(type)들로 태스크들을 분류함으로써 어느 정도까지는 극복할 수 있다. Ayoub 연구진에 의하면 최대 시스템 성능의 부분으로서 표현되는 성능 목표를 충족시키기 위해 CPU 주파수와 전압을 처리 할 수 있다. Flautner 연구진은 최근 계산 요건을 기반으로 각 태스크의 CPU 주파수를 선택하는 방법을 제안하였다.

- [0009] Seo 연구진은 태스크들의 평균 응답 시간을 줄이기 위한 CPU 주파수를 할당 방법을 제시하였다. 이러한 모든 방법들은 일반적인 작업량에 대하여 개발되었으므로 실시간 제약성이 있는 멀티미디어 응용에 대해서는 적합하지 않을 수 있다.
- [0010] 실시간 시스템을 위한 DVS 기법으로는 일반적으로 실시간 스케줄링(scheduling)이 통합된 것이다. 최악의 경우에서의 실행 시간 분석에 기반하여 대체적으로 실시간 제약성을 만족하는 최적의 CPU 주파수를 끌어낼 수 있다. 그러나, 태스크는 때때로 최악의 실행 시간 전에 완료될 수도 있으므로 몇몇 알고리즘들은 더 많은 에너지를 절약하기 위해 사용되지 않는 CPU 시간 되찾는 방법을 포함한다. 이러한 기법은 먼저 최악의 경우에서의 수요에 기반한 CPU 주파수를 선택하고 실제 계산 요구량에 대한 반응을 고려하여 프로세서를 감속시킨다.
- [0011] Pillai과 Aydin 연구진은 DVS 스케줄링이 사용될 때 rate-monotonic(RM)과 earliest deadline first(EDF) 스케줄링 방법을 위한 스케줄 가능성 조건들을 제안했다. Shin 연구진은 실시간 태스크의 스케줄 가능성 조건을 제공하기 위해 프로그램 단편들의 실행시간을 찾는 정적 시간조절 분석을 사용했다. Marinoni 연구진은 데드라인 범위를 만족하는 CPU 스케줄링 메카니즘을 소개했다. AlEnawy 연구진은 모든 태스크들의 예측 가능한 성능을 보장하기 위한 weakly-hard 실시간 스케줄링 알고리즘을 소개했다. Zhao 연구진은 주어진 시간과 에너지 제약 내에서 전반적으로 신뢰성을 최대화시키기 위해 실시간 태스크들의 한 집합에게 주파수를 할당하는 방법을 고려했다. 이외에 경성 실시간 제약성을 보장하기 위해 많은 작업이 수행되어왔다. 이러한 기법들은 모두 경성 실시간 응용을 위해 개발되어왔으며 멀티미디어 응용을 위한 적합성에 대해서는 의문의 여지가 있다.
- [0012] 이와 같이, 여러 연구진이 비디오 응용을 위한 DVS 기법들을 연구해왔으며, 주요 쟁점은 연속적인 프레임들의 계산 요구량을 추정하는 것이다. 이러한 기법들 대부분은 이전의 복호화된 프레임들의 작업량을 기반으로 하여 연속적인 프레임들의 작업량을 예측하는 것이다. 프로세서의 기본적인 전압과 주파수는 데드라인 요건들을 만족시키는 것으로 채택된다. 이 기법의 정확성은 이 전의 예측 실패를 고려되는 control-theoretic 피드백(feedback) 메카니즘을 사용하여 향상되어왔다.
- [0013] 여러 연구에서 프레임 크기와 디코딩 시간 사이에 밀접한 관계가 있음을 발견했으며, 특히 MPEG 종류의 비디오들이 디코딩 시간을 예측하는 데 사용될 수 있다. 예를 들어, Liu 연구진은 디코딩 시간을 예측하기 위해 디코딩 시간과 프레임 크기의 선형 관계를 확립한 반면에, Yang 연구진은 대수적인 관계를 사용하여 정확성을 향상시켰다. Bavier 연구진은 디코딩 시간을 예측하기 위해 대수 관계를 사용하여 만들었으며, Lee 연구진은 H.264 코덱(codec)들의 정확성을 향상시키는 입자여과기 기법을 적용하였다.
- [0014] Yuan 연구진은 CPU 속도가 통계적인 CPU 수요에 부합하도록 적용되는 몇몇의 DVS 기법들을 제안했다. Urunuela 연구진은 과거 기반 DVS 기법을 제안했지만 일반적인 비디오 플레이어보다는 광고판에 맞추어 제작되었다. Choi 연구진은 부분 프레임들의 특성에 의존하도록 적용되는 서로 다른 DVS 정책들을 혼합하여 결합하는 접근 방법을 적용했다. Im 연구진은 버퍼(buffer)가 사용되지 않는 CPU 시간을 회수하도록 사용하는 DVS 기법들 Huang 연구진은 비디오 특성에 기반하여 디코딩 시간을 계산하는 방법을 소개했다.
- [0015] 그러나, 이러한 연구 대부분은 특정 부분에서 데드라인 미스와 프레임 스키핑이 적용되어야 하는 비디오 재생의 특성이 고려되지 않았다. Yang 연구진이 에너지 소비에 대해 데드라인 미스 비율에 따른 효과를 검사하였으나 에너지 소비를 최소화하는 적절한 주파수 선택에 대한 체계적인 해결책을 제시하지 못했다. 게다가, 이러한 접근 방법들은 프레임 스키핑(skipping)과 버퍼링 기법이 전력소비에 어떻게 영향을 미치는 지를 검토하지 못했다.

발명의 내용

해결하려는 과제

- [0016] 본 발명이 이루고자 하는 기술적 과제는 CPU의 동작 전압과 주파수를 동적으로 변화시키는 동적 전압 조절(DVS)을 사용하여 CPU 전력 소비를 줄이는 버퍼링을 이용한 비디오 재생 시 최적의 CPU 주파수를 선택하는 동적 전압 조절 방법 및 장치를 제공하는데 있다.
- [0017] 또한, 본 발명이 이루고자 하는 기술적 과제는 프레임을 디코딩 할 때 지정된 버퍼와 데드라인을 만족하면서 전력소모를 최소로 할 수 있는 CPU 주파수를 결정하는 최적화 알고리즘을 제안함으로써, 에너지 소비를 줄일 수 있는 버퍼링을 이용한 비디오 재생 시 최적의 CPU 주파수를 선택하는 동적 전압 조절 방법 및 장치를 제공하는데 있다.

과제의 해결 수단

- [0018] 일 측면에 있어서, 본 발명에서 제안하는 버퍼링을 이용한 비디오 재생 시 최적의 CPU 주파수를 선택하는 동적 전압 조절 방법에 있어서, 각 프레임을 디코딩하기 이전에, CPU에 부과되는 상기 프레임의 작업량을 데드라인 전에 처리할 수 있는 주파수를 선택하는 단계; 선택된 상기 주파수의 전력을 측정하는 단계; 및 측정된 상기 주파수의 전력을 이용하여 상기 프레임의 디코딩 시 에너지 소비를 최소화 하는 단계를 포함한다.
- [0019] 상기 프레임의 작업량을 데드라인 전에 처리할 수 있는 주파수를 선택하는 단계는 선택된 상기 주파수를 다음 상기 프레임이 되기 전까지 변경하지 않는 것을 특징으로 할 수 있다.
- [0020] 상기 프레임의 디코딩 시 에너지 소비를 최소화 하는 단계는 버퍼에 미리 디코딩한 상기 프레임을 저장할 수 있다.
- [0021] 상기 프레임의 디코딩 시 에너지 소비를 최소화 하는 단계는 상기 프레임을 디코딩하는데 필요한 최소의 상기 에너지 소비량, 상기 프레임의 디코딩 초과시간, 및 상기 프레임의 디코딩 시간을 초기화 하여, 전압과 상기 주파수를 동적으로 변화시켜 상기 전력을 최소화 하는 상기 주파수를 선택하는 알고리즘을 형성할 수 있다.
- [0022] 다른 측면에 있어서, 본 발명에서 제안하는 버퍼링을 이용한 비디오 재생 시 최적의 CPU 주파수를 선택하는 동적 전압 조절 장치에 있어서, 각 프레임을 디코딩하기 이전에, CPU에 부과되는 상기 프레임의 작업량을 데드라인 전에 처리할 수 있는 주파수를 선택하는 주파수 선택부; 선택된 상기 주파수의 전력을 측정하는 주파수 측정부; 및 측정된 상기 주파수의 전력을 이용하여 상기 프레임의 디코딩 시 에너지 소비를 최소화 하는 디코딩부를 포함하고, 선택된 상기 주파수를 다음 상기 프레임이 되기 전까지 변경하지 않고, 버퍼에 미리 디코딩한 상기 프레임을 저장할 수 있다.

발명의 효과

- [0023] 본 발명의 실시예들에 따르면 CPU의 동작 전압과 주파수를 동적으로 변화시키는 동적 전압 조절(DVS)을 사용하여 CPU 전력 소비를 줄이는 버퍼링을 이용한 비디오 재생 시 최적의 CPU 주파수를 선택하는 동적 전압 조절 방법 및 장치를 제공할 수 있다.
- [0024] 또한, 본 발명의 실시예들에 따르면 프레임을 디코딩 할 때 지정된 버퍼와 데드라인을 만족하면서 전력소모를 최소화 할 수 있는 CPU 주파수를 결정하는 최적화 알고리즘을 제안함으로써, 에너지 소비를 줄일 수 있는 버퍼링을 이용한 비디오 재생 시 최적의 CPU 주파수를 선택하는 동적 전압 조절 방법 및 장치를 제공할 수 있다.

도면의 간단한 설명

- [0025] 도 1은 본 발명의 일 실시예에 따른 변수들의 개념을 구체적으로 설명하기 위한 도이다.
- 도 2는 본 발명의 일 실시예에 따른 최적의 CPU 주파수를 선택하는 동적 전압 조절 방법을 나타내는 순서도이다.
- 도 3은 본 발명의 일 실시예에 따른 최적의 CPU 주파수를 선택하는 동적 전압 조절 장치를 나타내는 블록도이다.

발명을 실시하기 위한 구체적인 내용

- [0026] 이하, 본 발명의 실시 예를 첨부된 도면을 참조하여 상세하게 설명한다.
- [0027] 비디오 플레이어는 r fps(frame per second)의 속도로 디코딩한다고 가정할 수 있다. 한 프레임의 디코딩 주기가 T^d 일 때, 이는 $\frac{1}{r}$ 로 계산될 수 있다. 그리고, CPU는 N^f 단계의 주파수를 가지고, 주파수 단계 k 의 주파수는 f_k 라고 가정할 수 있다. ($k = 1, \dots, N_f$). 만약, $k < m$ 이면 $f_k < f_m$ 이다. 즉, f_{N_f} 가 CPU가 지원하는 가장 높은 주파수가 될 수 있다.

[0028] 또한, 비디오 파일의 총 프레임 수를 N^d 라 하고, P_k^{active} P_k^{idle} 을 각각 주파수 k 단계에서 유효 전력, 유티 전력이라고 할 수 있다.

[0029] 각 프레임의 디코딩 시간은 알고 있다고 가정할 수 있다. 디코딩 시간은 비디오 클립의 오프라인(offline) 비트 스트림(bitstream) 분석하거나 프레임 사이즈와 디코딩 시간의 관계를 분석하여 예측 가능하다. 디코딩 시간 정보는 비디오 클립의 헤더에 삽입 가능하고, 이 정보들은 제안하는 주파수 선택 알고리즘을 위해 사용될 수 있다.

[0030] $d_{j,k}$ 를 주파수 단계 k 에서 프레임 f_j 를 디코딩 하는데 걸리는 시간이고, 버퍼는 N^b 개의 프레임을 저장할 수 있다고 가정할 수 있다. 만약, 버퍼에 N^b 개의 프레임이 저장되어 있다면, 새로운 프레임은 들어갈 공간이 없기 때문에 하나의 프레임이 화면에 출력될 때까지 디코딩이 지연되어야 한다.

[0031] 여기서, 미디어 플레이어에 적합한 프레임 레벨(frame-level) DVS 모델을 사용하여 설명할 수 있다. 이 모델은 각 프레임을 디코딩하기 전에 CPU에 부과되는 프레임의 작업량을 데드라인 전에 처리할 수 있는 가장 적절한 주파수를 선택할 수 있다. 플레이어는 선택된 주파수를 다음 프레임이 되기 전까지 바꾸지 않는다.

[0032] 각 주파수의 전력을 측정하고, 주파수 k 단계에서의 전력을 P_k 라고 할 수 있다.

[0033] 버퍼 한계에 따른 전체 에너지 소비를 최소화하는 최적화 문제를 공식화 할 수 있다. 이때, 프레임 $j, (j = 1, \dots, N^d)$ 는 반드시 해당 프레임의 데드라인 (T_j^{end}) 전에 디코딩 되어야 한다. S_j 를 j 프레임을 디코딩할 때 선택한 주파수라고 할 수 있다. 또한, T_j^{start} 는 j 프레임의 디코딩을 시작할 수 있는 가장 빠른 시간이라고 할 수 있다. 버퍼에 N^b 개의 프레임을 저장할 수 있기 때문에 j 프레임은 $T_{j-N^b}^{\text{end}}$ 부터 디코딩이 가능하고, 다음 식과 같이 계산할 수 있다.

수학식 1

$$T_j^{\text{start}} = \begin{cases} 0 & j = 1, \dots, N^b \\ (j - N^b) T^d & j = N^b + 1, \dots, N^d \end{cases}$$

[0034]

[0035] $T_{j,k}^{\text{over}} (j = 1, \dots, N^d - 1)$ 을 주파수 k 단계에서 T_{j+1}^{start} 과 비교했을 때 j 프레임의 초과시간이라 하며, 다음과 같이 계산할 수 있다.

수학식 2

$$T_{j,k}^{\text{over}} = \max(0, d_{j,k} + T_{j-1, S_{j-1}}^{\text{over}} + T_j^{\text{start}} - T_{j+1}^{\text{start}})$$

[0036]

[0037] 만약 j 프레임이 T_{j+1}^{start} 후에 디코딩 된다면, $T_{j,k}^{\text{over}}$ 는 주파수 k 단계에서, j 프레임이 디코딩을 마친 실제시간 ($d_{j,k} + T_{j-1, S_{j-1}}^{\text{over}} + T_j^{\text{start}}$) 과 T_{j+1}^{start} 의 차이를 나타낼 수 있다. 그렇지 않은 경우, CPU는 T_{j+1}^{start} 까지 유티 상태로 있을 수 있고, $T_{j,k}^{\text{over}}$ 는 0이 될 수 있다. 기본적으로 모든 $T_{j,k}^{\text{over}}$ 는 0으로 초기화 하는 것이 바람직하다. $T_{j,k}^{\text{sleep}}$ 는 다음 식과 같이 정의할 수 있다.

수학식 3

$$T_{j,k}^{sleep} = \begin{cases} T_{j+1}^{start} - T_j^{start} - d_{j,k} - T_{j-1,S_{j-1}}^{over} & \text{if } T_{j,k}^{over} = 0 \\ 0 & \text{otherwise.} \end{cases}$$

주파수 k 단계에서 j 프레임을 디코딩할 때의 에너지 소비량을 나타내는 $E_{j,k}$ 는 다음 식과 같이 표현할 수 있다.

수학식 4

$$E_{j,k} = d_{j,k} P_k^{active} + T_{j,k}^{sleep} P_k^{idle}$$

또한, 새로운 변수 $T_{j,k}^{diff}$ 를 도입하고, 이는 j 프레임의 실제 디코딩이 끝난 시간 $(d_{j,k} + T_{j-1,S_{j-1}}^{over} + T_j^{start})$ 과 T_{j+1}^{start} 의 차이를 나타내며, 다음 식과 같이 표현할 수 있다.

수학식 5

$$T_{j,k}^{diff} = d_{j,k} + T_{j-1,S_{j-1}}^{over}$$

도 1은 본 발명의 일 실시예에 따른 변수들의 개념을 구체적으로 설명하기 위한 도이다.

도 1을 참조하면, 위에서 설명한 변수들의 개념을 구체적으로 설명할 수 있다.

여기서, j 프레임의 디코딩은 반드시 T_j^{start} 이후에 시작되고, T_j^{end} 이전에 끝나야 한다. 이때의 주기를 T_j^{round} 라 하고 다음 식과 같이 표현할 수 있다.

수학식 6

$$T_j^{round} = T_j^{end} - T_j^{start}$$

각 프레임은 반드시 테드라인 T_j^{end} 전에 디코딩이 완료되어야 한다. 이는 곧 $T_{j,k}^{diff}$ 는 반드시 T_j^{round} 보다 작거나 같아야 한다는 것을 의미할 수 있다. 본 발명에서 주파수 선택은 전체 에너지 소비량 $(\sum_{j=1}^{N^d} E_{j,S_j})$ 을 최소화 하기 위한 것이다. 따라서, 주파수 선택 문제(FSP: frequency selection problem)를 공식화하여 표현할 수 있다.

수학식 7

$$\begin{aligned} & \text{Minimize} \quad \sum_{j=1}^{N^d} E_{j,S_j} \\ & \text{Subject to} \quad T_{j,S_j}^{\text{diff}} \leq T_j^{\text{round}}, \forall j, j=1, \dots, N^d \end{aligned}$$

도 2는 본 발명의 일 실시예에 따른 최적의 CPU 주파수를 선택하는 동적 전압 조절 방법을 나타내는 순서도이다.

단계(210)에서, 각 프레임을 디코딩하기 이전에, CPU에 부과되는 상기 프레임의 작업량을 테드라인 전에 처리할 수 있는 주파수를 선택할 수 있다. 그리고, 상기 프레임의 작업량을 테드라인 전에 처리할 수 있는 주파수를 선택하는 단계는 선택된 상기 주파수를 다음 상기 프레임이 되기 전까지 변경하지 않는 것이 바람직하다.

단계(220)에서, 선택된 상기 주파수의 전력을 측정할 수 있다.

단계(230)에서, 측정된 상기 주파수의 전력을 이용하여 상기 프레임의 디코딩 시 에너지 소비를 최소화 할 수 있다. 여기서, 프레임을 디코딩할 때의 유효 전력과 유휴 상태의 유휴 전력을 이용하여 에너지 소비량을 계산할 수 있다. 그리고, 상기 프레임의 디코딩 시 에너지 소비를 최소화 하는 단계는 버퍼에 미리 디코딩한 상기 프레임을 저장할 수 있다.

또한, 상기 프레임의 디코딩 시 에너지 소비를 최소화 하는 단계는 상기 프레임을 디코딩하는데 필요한 최소의 상기 에너지 소비량, 상기 프레임의 디코딩 초과시간, 및 상기 프레임의 디코딩 시간을 초기화 하여, 전압과 상기 주파수를 동적으로 변화시켜 상기 전력을 최소화 하는 상기 주파수를 선택하는 알고리즘을 형성할 수 있다. 이에 대한 구체적인 설명은 아래에서 계속하기로 한다.

이와 같이, 주파수 선택 문제(FSP)를 풀기 위해 다이나믹 프로그래밍 기법을 이용하여 최적화 알고리즘을 제안할 수 있다. $T_{j,k}^{\text{diff}}$ 의 최솟값은 1ms 라고 가정할 수 있다. 여기서, $E_{j,u}^{\text{min}}$ 는 T_{j,S_j}^{diff} 가 u ms 일 때 f_j 를 디코딩하는데 필요한 최소 에너지 소비량이라고 할 수 있다. 그리고, $F_{j,u}$ 는 T_{j,S_j}^{diff} 가 u ms 일 때 f_j 를 디코딩하는데 필요한 $E_{j,u}^{\text{min}}$ 를 최소로 할 수 있는 주파수 단계일 수 있다. 또한, $V_{j,u}^{\text{over}}$ 는 T_{j,S_j}^{diff} 가 u ms 일 때 $E_{j,u}^{\text{min}}$ 를 최소로 할 수 있는 T_{j,S_j}^{over} 의 값이고, $V_{j,u}^{\text{over}}$ 는 T_{j,S_j}^{diff} 가 u ms 일 때 $E_{j,u}^{\text{min}}$ 를 최소로 할 수 있는 $T_{j-1,S_i}^{\text{diff}}$ 의 값이 될 수 있다.

이러한, 제안하는 다이나믹 프로그래밍 알고리즘은 총 세 단계로 구성될 수 있다.

(1) 초기화 단계

초기화 단계에서는 $E_{j,u}^{\text{min}}$, $V_{j,u}^{\text{over}}$ 와 $V_{j,u}^{\text{diff}}$ 를 초기화 할 수 있다.

모든 $E_{j,u}^{\text{min}}$ ($\forall j, u, j=1, \dots, N^d$ and $u=1, \dots, T_j^{\text{round}}$)는 무한대로 초기화 할 수 있다.

$d_{1,k} \leq T_1^{\text{end}}$ 인 모든 k 에 대하여, $T_{1,k}^{\text{over}}$, $T_{1,k}^{\text{sleep}}$ 와 $E_{1,k}$ 값을 각각 [수학식 2], [수학식 3], 및 [수학식 4]를 이용하여 계산할 수 있다. 이 후, $V_{1,d_{1,k}}^{\text{over}}$ 는 $T_{1,k}^{\text{over}}$ 값으로 대체되고, $E_{1,d_{1,k}}^{\text{min}}$ 는 $E_{1,d_{1,k}}^{\text{min}}$ 값으로 갱신될 수 있다. 그리고, $F_{1,d_{1,k}}$ 값은 주파수 단계 k 로 갱신될 수 있다.

(2) 재귀 구성단계

[0061] 재귀 구성단계에서는 $E_{j,u}^{\min}$, $F_{j,u}$, $V_{j,u}^{\text{diff}}$ 와 $V_{j,u}^{\text{over}}$ ($j = 2, \dots, N^d$ and $u = 1, \dots, T_j^{\text{round}}$) 값을 갱신할 수 있다.

[0062] 먼저, 모든 j 와 u 값에 대한 2차 배열 $A_{j,u}(i, k)$ ($i = 1, \dots, T_{j-1}^{\text{round}}$ and $k = 1, \dots, N^f$) 을 유지할 수 있다. 모든 i , k 에 대하여, $E_{j-1,i}^{\min}$ 가 무한대가 아니라면 다음과 같은 과정의 반복을 통해 $A_{j,u}(i, k)$ 의 값을 찾아낼 수 있다.

[0063] [수학식 2]에 $T_{j-1, S_{j-1}}^{\text{over}}$ 에 $V_{j-1,i}^{\text{over}}$ 값을 대입하여 $T_{j,k}^{\text{over}}$ 를 계산할 수 있다.

[0064] 그리고, 위에서 계산한 $T_{j,k}^{\text{over}}$ 을 [수학식 3]에 대입하여 $T_{j,k}^{\text{sleep}}$ 을 계산할 수 있다.

[0065] 이후, 위에서 계산한 $T_{j,k}^{\text{sleep}}$ 을 [수학식 4]에 대입하여 $E_{j,k}$ 을 계산하고, $A_{j, d_{j,k} + V_{j-1,i}^{\text{over}}}(i, k)$ 값을 $E_{j,k}$ 값으로 갱신할 수 있다.

[0066] 두 번째로, $E_{j,u}^{\min}$, $F_{j,u}$, $V_{j,u}^{\text{diff}}$ 값을 다음과 같이 갱신할 수 있다.

수학식 8

$$\begin{aligned} E_{j,u}^{\min} &= \min_{i=1, \dots, T_{j-1}^{\text{round}} \text{ and } k=1, \dots, N^f} (E_{j-1,i}^{\min} + A_{j,u}(i, k)), \\ F_{j,u} &= \arg \min_{k=1, \dots, N^f} E_{j,u}^{\min}, \\ V_{j,u}^{\text{diff}} &= \arg \min_{i=1, \dots, N^f} E_{j,u}^{\min}, \\ V_{j,u}^{\text{over}} &= \max(0, V_{j,u}^{\text{diff}} + d_{j, F_{j,u}} - T_{j+1}^{\text{start}}). \end{aligned}$$

[0068] (3) 역추적 단계

[0069] 역추적 단계에서는 최적의 S_j 값을 찾을 수 있다. 먼저, j 를 N^d 값으로 초기화 하고, u 는 $\arg \min_{u=1, \dots, T_{N^d}^{\text{round}}} E_{N^d,u}^{\min}$ 로 설정할 수 있다. 그리고, j 가 0보다 큰 값을 가진 동안, 다음 과정을 반복할 수 있다.

[0070] 첫 번째로, S_j 를 $F_{j,u}$ 로 설정할 수 있다.

[0071] 두 번째로, u 를 $V_{j,u}^{\text{diff}}$ 로 대체할 수 있다.

[0072] 세 번째로, j 값 1 감소시킬 수 있다.

[0073] 주파수 선택 알고리즘(FSA: frequency selection algorithm)은 위의 세 단계로 구성될 수 있으며, [표 1]에 자세한 내용이 기술되어 있다.

[0074] [표 1]은 본 발명의 일 실시예에 따른 주파수 선택 알고리즘을 나타내는 표이다. 이때, T^{max} 를 최대 순환 횟수라 하고, $T^{\text{max}} = \max_{j=2, \dots, N^d} T_j^{\text{round}}$ 가 될 수 있다.

㉠ 1

```

1: Temporary variables :  $j, u, i$  and  $k$ ;
2: for  $j = 1$  to  $N^d$  do
3:   for  $u = 1$  to  $T_j^{\text{round}}$  do
4:      $E_{j,u}^{\text{min}} \leftarrow \infty$ ;
5:   end for
6: end for
7: for  $u = 1$  to  $T_j^{\text{round}}$  do
8:   for  $k = 1$  to  $N^f$  do
9:     if  $d_{1,k} \leq T_1^{\text{end}}$  and  $u = d_{1,k}$  then
10:      Calculate  $T_{1,k}^{\text{over}}$ ,  $T_{1,k}^{\text{sleep}}$  and  $E_{1,k}$  using Equeations (2),(3), and
      (4), respectively;
11:       $E_{1,u}^{\text{min}} \leftarrow E_{1,k}$ ;
12:       $F_{1,u} \leftarrow k$ ;
13:       $V_{1,u}^{\text{over}} \leftarrow T_{1,k}^{\text{over}}$ ;
14:    end if
15:  end for
16: end for
17: for  $j = 2$  to  $N^d$  do
18:   for  $i = 1$  to  $T_{j-1}^{\text{round}}$  do
19:     for  $k = 1$  to  $N^f$  do
20:       if  $E_{j-1,i}^{\text{min}} \neq \infty$  then
21:         Calculate the value of  $T_{j,k}^{\text{over}}$  using Equation (2), in which
          $T_{j-1,S_{j-1}}^{\text{over}}$  is replaced by  $V_{j-1,i}^{\text{over}}$ ;
22:          $T_{j,k}^{\text{sleep}}$  is cacuated using Equation (3) in which  $T_{j-1,S_{j-1}}^{\text{over}}$  is
         replaced by  $V_{j-1,i}^{\text{over}}$ ;
23:          $E_{j,k}$  is calculated using Equation (4) and  $A_{j,d_{jk}+V_{j-1,i}^{\text{over}}}(i,k)$  is
         updated by this  $E_{j,k}$  value;
24:       end if

```

[0075]

```

25:     end for
26:   end for
27:    $E_{j,u}^{\min} \leftarrow \min_{i=1, \dots, T_{j-1}^{\text{end}} \text{ and } k=1, \dots, N^d} (E_{j-1,i}^{\min} + A_{j,u}(i,k));$ 
28:    $F_{j,u} \leftarrow \operatorname{argmin}_{k=1, \dots, N^d} E_{j,u}^{\min};$ 
29:    $V_{j,u}^{\text{diff}} \leftarrow \operatorname{argmin}_{i=1, \dots, T_{j-1}^{\text{end}}} E_{j,u}^{\min};$ 
30:    $V_{j,u}^{\text{over}} \leftarrow \max(0, V_{j,u}^{\text{diff}} + d_{j,F_{j,u}} - T_{j+1}^{\text{start}});$ 
31:   end for
32:    $j \leftarrow N^d;$ 
33:    $u \leftarrow \operatorname{argmin}_{i=1, \dots, T_{j-1}^{\text{end}}} E_{N^d,i}^{\min};$ 
34:   while  $j > 0$  do
35:      $S_j \leftarrow F_{j,u};$ 
36:      $u \leftarrow V_{j,u}^{\text{diff}};$ 
37:      $j \leftarrow j - 1;$ 
38:   end while

```

[0076]

[0077]

도 3은 본 발명의 일 실시예에 따른 최적의 CPU 주파수를 선택하는 동적 전압 조절 장치를 나타내는 블록도이다.

[0078]

주파수 선택부(310)는 각 프레임을 디코딩하기 이전에, CPU에 부과되는 상기 프레임의 작업량을 테드라인 전에 처리할 수 있는 주파수를 선택할 수 있다. 여기서, 선택된 상기 주파수를 다음 상기 프레임이 되기 전까지 변경하지 않는 것이 바람직하다.

[0079]

주파수 측정부(320)는 선택된 상기 주파수의 전력을 측정할 수 있다.

[0080]

디코딩부(330)는 측정된 상기 주파수의 전력을 이용하여 상기 프레임의 디코딩 시 에너지 소비를 최소화 할 수 있다. 이때, 버퍼에 미리 디코딩한 상기 프레임을 저장하는 것이 바람직하다.

[0081]

[표 2]는 본 발명의 일 실시예에 따른 시뮬레이션 결과를 나타내는 것이다.

표 2

주파수 (MHz)	1000	800	400	200	100
유효 전력	1	0.712	0.306	0.111	0.031
유허 전력	1	0.968	0.923	0.864	0.770

[0082]

[0083]

위에서 설명한 바를 시뮬레이션을 위해서, Monsoon 사의 파워 모니터(Power Monitor) 기기를 이용하여 넥서스(Nexus) S 스마트 폰의 시스템 전체 전력을 측정할 수 있다. 그 결과는 다음의 [표 2]와 같이 나타낼 수 있다.

[0084]

또한, 두 가지 영상에 대해 디코딩 시간을 측정할 수 있다. [표 3]은 본 발명의 일 실시예에 따른 측정에 사용된 영상에 대한 정보를 나타낸 것이다.

표 3

영상 종류	영상 제목	해상도	평균 비트율	재생 시간
애니메이션	IceAge4	352 x 288	736 Kb/s	5 분
스포츠	soccer	352 x 288	199 Kb/s	5 분

[0085]

[0086]

이와 같이, 제안한 알고리즘을 다음의 두 가지 방법과 비교할 수 있다.

[0087]

첫 번째 방법은 HF 방법으로, 항상 가장 높은 주파수가 선택되는 방법이다. 이 방법은 DVS가 적용되지 않은 방법과 동일하다.

[0088]

두 번째 방법은 LF 방법으로, 디코딩 시간이 데드라인을 넘지 않는 가장 낮은 주파수를 선택하는 방법이다. CPU 주파수와 에너지 소비는 보통 1차적인 관계를 가지기 때문에 이 방법이 에너지 소비를 최소화하는 경험적 알고리즘으로 생각될 수 있다.

[0089]

[표 4]는 본 발명의 일 실시예에 따른 버퍼의 수에 따른 에너지 소비량의 관계를 나타낸다.

표 4

영상 종류	애니메이션				스포츠			
버퍼 수	1	2	3	4	1	2	3	4
HF에 대한 상대적인 에너지 소비량	80%	76%	76%	75%	80%	77%	76%	76%
LF에 대한 상대적인 에너지 소비량	90%	86%	85%	85%	92%	88%	88%	88%

[0090]

[0091]

[표 4]를 참조하면, FSA가 에너지 소비 효율이 가장 좋음을 확인할 수 있고, 버퍼 수가 증가함에 따라 에너지 소비를 더 줄일 수 있다는 것을 알 수 있다. 하지만, 버퍼 수가 일정 수준이 되면 에너지 소비가 일정해진다. 여기서, FSA가 좋은 성능을 보이는 이유는 버퍼에 미리 디코딩한 프레임을 저장함으로써 생기는 여유 시간을 효과적으로 사용하기 때문이다. 이 여유 시간 덕분에 CPU는 디코딩할 때 더 낮은 주파수를 선택할 수 있게 되는 것이다.

[0092]

[표 5]는 본 발명의 일 실시예에 따른 프레임이 디코딩 될 때 선택되는 주파수 단계의 평균 비율을 나타낸다.

표 5

영상 종류	애니메이션				
주파수(MHz)	100	200	400	800	1000
LF	6.54%	1.11%	46.21%	45.57%	0.66%
$FSA(N^b=1)$	30.16%	0.83%	29.74%	34.07%	5.20%
$FSA(N^b=2)$	38.69%	0.01%	1.29%	44.93%	15.09%
$FSA(N^b=3)$	35.91%	0.00%	0.21%	46.17%	17.70%
$FSA(N^b=4)$	35.03%	0.00%	0.17%	46.67%	18.13%
	스포츠				
LF	1.40%	15.01%	82.24%	1.34%	0%
$FSA(N^b=1)$	11.19%	56.30%	20.54%	10.47%	1.50%
$FSA(N^b=2)$	32.84%	10.21%	12.81%	35.51%	8.61%
$FSA(N^b=3)$	36.84%	2.34%	8.61%	41.26%	10.94%
$FSA(N^b=4)$	38.09%	0.44%	6.14%	43.11%	12.21%

[0093]

[0094]

단, LF 방법은 여유 시간을 고려하지 않고, 디코딩 시간이 데드라인을 만족할 수 있는 최저 주파수 단계를 선택할 수 있다.

[0095]

이상과 같이, 본 발명에 따르면 CPU의 동작 전압과 주파수를 동적으로 변화시키는 동적 전압 조절(DVS)을 사용하여 CPU 전력 소비를 줄일 수 있고, 프레임을 디코딩 할 때 지정된 버퍼와 데드라인을 만족하면서 전력소모를 최소로 할 수 있는 CPU 주파수를 결정하는 최적화 알고리즘을 제안할 수 있다. 실제 측정된 비디오 재생 시 스마트폰의 에너지 소비를 바탕으로 시뮬레이션 한 결과, 제안한 알고리즘이 DVS를 사용하지 않을 때와 비교하여 시스템 전체적으로 25%의 에너지 소비를 줄일 수 있다.

[0096]

이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 컨트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPA(field programmable array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0097]

소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치, 또는 전송되는 신호 파(signal wave)에 영구적으로, 또는 일시적으로 구체화(embodiment)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서,

분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0098]

실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 실시예의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0099]

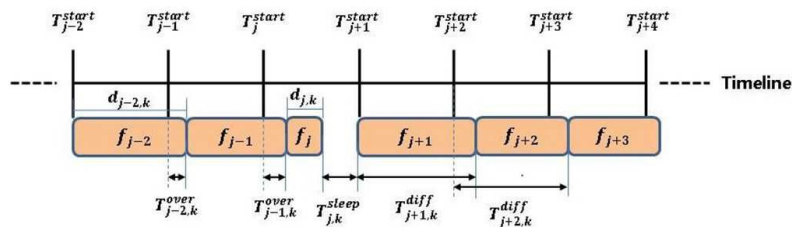
이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

[0100]

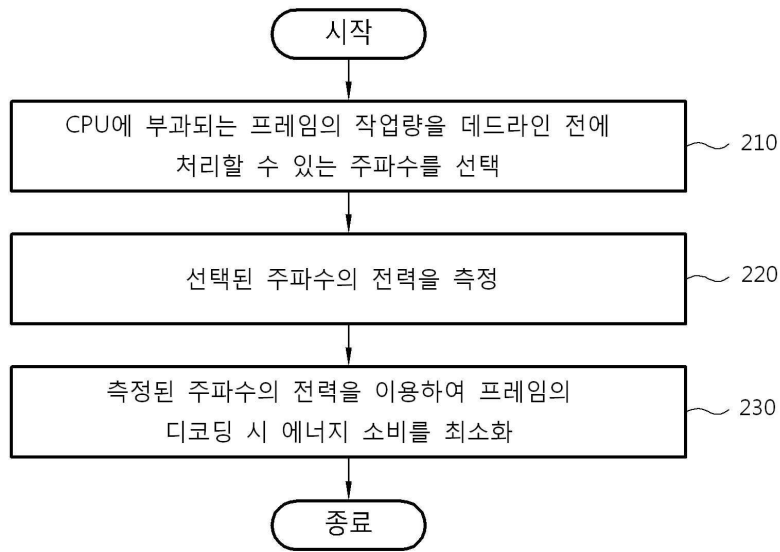
그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

도면1



도면2



도면3

