



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2017-0131366  
(43) 공개일자 2017년11월29일

- |  |   |
|--|---|
| <p>(51) 국제특허분류(Int. Cl.)<br/>G06F 9/50 (2006.01)</p> <p>(52) CPC특허분류<br/>G06F 9/5016 (2013.01)<br/>G06F 2209/504 (2013.01)</p> <p>(21) 출원번호 10-2017-7023392</p> <p>(22) 출원일자(국제) 2016년02월18일<br/>심사청구일자 없음</p> <p>(85) 번역문제출일자 2017년08월22일</p> <p>(86) 국제출원번호 PCT/US2016/018460</p> <p>(87) 국제공개번호 WO 2016/153646<br/>국제공개일자 2016년09월29일</p> <p>(30) 우선권주장<br/>14/668,044 2015년03월25일 미국(US)</p> | <p>(71) 출원인<br/>인텔 코퍼레이션<br/>미합중국 캘리포니아 95054 산타클라라 미션 칼리지 블러바드 2200</p> <p>(72) 발명자<br/>콜맨, 제임스 에이.<br/>미국 85203 아리조나주 메사 노스 마운트뷰 2620</p> <p>(74) 대리인<br/>양영준, 김연송, 백만기</p> |
|--|---|

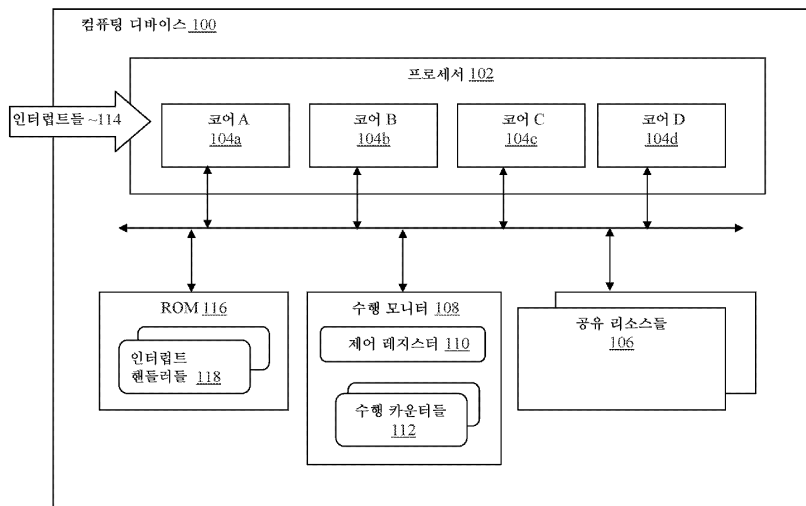
전체 청구항 수 : 총 25 항

(54) 발명의 명칭 공유 리소스 액세스 제어 방법 및 장치

(57) 요약

공유 리소스의 코어 액세스를 모니터링 및 제어하는 것과 관련된 장치들, 방법들 및 저장 매체들이 본 명세서에 개시된다. 실시예들에서, 장치는 복수의 코어를 갖는 프로세서; 상기 복수의 코어 간에 공유되도록 상기 프로세서와 결합된 리소스; 및 상기 복수의 코어의 상기 공유 리소스의 액세스 예산들을 저장하기 위해 상기 복수의 코어와 대응되게 관련된 복수의 수행 카운터를 포함할 수 있다. 상기 장치는 상기 수행 카운터들에 저장된 그 각자의 액세스 예산들에 따라 상기 복수의 코어에 의한 상기 공유 리소스의 액세스를 관리하기 위한 수행 모니터를 추가로 포함할 수 있다. 다른 실시예들이 설명 및/또는 청구될 수도 있다.

대표도



(52) CPC특허분류  
Y02B 60/142 (2013.01)

---

## 명세서

### 청구범위

#### 청구항 1

컴퓨팅 디바이스로서,

복수의 코어를 갖는 프로세서;

상기 복수의 코어 간에 공유되도록 상기 프로세서와 결합된 리소스;

상기 복수의 코어의 상기 공유 리소스의 액세스 예산들을 저장하기 위해 상기 복수의 코어와 대응되게 관련된 복수의 수행 카운터; 및

상기 프로세서, 상기 리소스 및 상기 수행 카운터들과 결합되어, 상기 수행 카운터들에 저장된 그 각자의 액세스 예산들에 따라 상기 복수의 코어에 의한 상기 공유 리소스의 액세스를 관리하기 위한 수행 모니터를 포함하는, 컴퓨팅 디바이스.

#### 청구항 2

제1항에 있어서,

제어 레지스터를 추가로 포함하고; 상기 수행 모니터는 상기 제어 레지스터와 추가로 결합되고, 상기 수행 카운터들과 함께 상기 제어 레지스터를 사용하여 그 각자의 액세스 예산들에 따라 상기 복수의 코어에 의한 상기 공유 리소스의 액세스를 관리하는, 컴퓨팅 디바이스.

#### 청구항 3

제2항에 있어서,

상기 수행 모니터는 상기 복수의 코어 중 어느 것이 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되어야 하는지를 나타내도록 상기 제어 레지스터를 구성하는 것인, 컴퓨팅 디바이스.

#### 청구항 4

제3항에 있어서,

상기 수행 모니터는 상기 복수의 코어 중 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 각각의 코어에 대해, 예산 시간 쿼텀에 기초하여, 다음 예산 체크 시간을 나타내도록 상기 제어 레지스터를 추가로 구성하는 것인, 컴퓨팅 디바이스.

#### 청구항 5

제1항에 있어서,

상기 수행 모니터는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 코어에 대응하는 각각의 수행 카운터를, 예산 시간 쿼텀에 대한 액세스 예산으로 구성하는 것인, 컴퓨팅 디바이스.

#### 청구항 6

제5항에 있어서,

예산 시간 쿼텀에 대한 상기 액세스 예산은 상기 공유 리소스의 액세스 이벤트들의 유형과 관련되는, 컴퓨팅 디바이스.

#### 청구항 7

제5항에 있어서, 상기 수행 모니터는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 코어에 대응하는 각각의 수행 카운터를, 오버플로 값에서 상기 예산 시간 쿼텀에 대한 상기 액세스 예산을 뺀 값으로 구성하는

것인, 컴퓨팅 디바이스.

**청구항 8**

제7항에 있어서,

상기 수행 모니터는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 코어에 대응하는 각각의 수행 카운터를, 오버플로시 인터럽트를 생성하도록 구성하는 것인, 컴퓨팅 디바이스.

**청구항 9**

제1항 내지 제8항 중 어느 한 항에 있어서,

상기 수행 모니터는 상기 복수의 코어에 의한 상기 공유 리소스의 액세스에 대해 모니터링하고, 코어에 의한 상기 공유 리소스의 액세스의 검출시, 액세스하는 코어가 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되었다면 대응하는 수행 카운터를 업데이트하는 것인, 컴퓨팅 디바이스.

**청구항 10**

제9항에 있어서,

상기 수행 모니터는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화된 코어에 대해, 상기 코어가 예산 시간 쿼텀에 대한 그의 액세스 예산에 도달한 것임을 나타내는 상기 대응하는 수행 카운터로부터의 표시의 검출시, 상기 공유 리소스의 추가 액세스를 거부하는 것인, 컴퓨팅 디바이스.

**청구항 11**

제10항에 있어서,

상기 공유 리소스의 예산 기반 액세스 제어가 활성화된 상기 코어에 대해, 대응하는 수행 카운터가 상기 코어가 상기 예산 시간 쿼텀에 대한 그의 액세스 예산에 도달한 것임을 나타내는 조건에 도달하는 결과로서 생성된 인터럽트에 응답하여, 상기 공유 리소스의 추가 액세스를 거부하기 위해 실행 제어를 넘겨받을 인터럽트 핸들러를 추가로 포함하는, 컴퓨팅 디바이스.

**청구항 12**

제11항에 있어서,

실행 제어를 넘겨받을 때, 상기 인터럽트 핸들러는:

현재 코어를 결정하고;

상기 현재 코어에 대해 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되어 있는지를 결정하고;

상기 현재 코어에 대해 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되어 있다는 결정시, 상기 현재 코어의 현재 틱들이 상기 현재 코어의 다음 예산 체크 시간보다 크지를 추가로 결정하는 것인, 컴퓨팅 디바이스.

**청구항 13**

제12항에 있어서,

상기 현재 코어의 현재 틱들이 상기 현재 코어의 다음 예산 체크 시간보다 크지 않다는 결정시, 상기 인터럽트 핸들러는 상기 현재 코어의 현재 틱들이 상기 현재 코어의 상기 다음 예산 체크 시간과 같을 때까지 상기 현재 코어를 회전시키는 것인, 컴퓨팅 디바이스.

**청구항 14**

제12항에 있어서,

상기 현재 코어의 현재 틱들이 상기 현재 코어의 다음 예산 체크 시간보다 크다는 결정시, 상기 인터럽트 핸들러는:

상기 현재 코어의 상기 다음 예산 체크 시간을 상기 현재 코어의 상기 현재 틱들과 상기 현재 코어의 예산 시간

퀵텀의 함으로 설정하고;

그 조건이 상기 인터럽트 핸들러가 실행 제어를 넘겨받는 것으로 이어진 상기 인터럽트를 야기하는 상기 현재 코어의 대응하는 수행 카운터를, 예산 시간 퀵텀에 대한 액세스 예산으로 리셋하는 것인, 컴퓨팅 디바이스.

#### 청구항 15

제1항에 있어서,

상기 수행 모니터는 상기 컴퓨팅 디바이스의 운영 체제 또는 하이퍼바이저의 일부인, 컴퓨팅 디바이스.

#### 청구항 16

컴퓨팅 디바이스상의 공유 리소스에 대한 코어 액세스들을 제어하는 방법으로서,

상기 컴퓨팅 디바이스의 수행 모니터에 의해, 상기 컴퓨팅 디바이스의 프로세서의 복수의 코어의 복수의 대응하는 수행 카운터 각각을 예산 시간 퀵텀에 대한 상기 공유 리소스에 대한 대응하는 코어의 액세스 예산으로 구성하는 단계; 및

상기 수행 모니터에 의해, 상기 수행 카운터들을 이용하여, 상기 코어들의 상기 액세스 예산들에 따라, 상기 코어들에 의한 상기 공유 리소스의 액세스를 모니터링 및 제어하는 단계를 포함하는, 방법.

#### 청구항 17

제16항에 있어서,

구성하는 단계는 상기 액세스 예산과 관련된 제어 데이터로 제어 레지스터를 구성하는 단계를 추가로 포함하고, 모니터링 및 제어하는 단계는 상기 제어 레지스터를 이용하는 단계를 추가로 포함하고;

구성하는 단계는 상기 복수의 코어 중 어느 것이 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되어야 하는지를 나타내고, 상기 복수의 코어 중 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 각각의 코어에 대해, 예산 시간 퀵텀에 기초하여, 다음 예산 체크 시간을 나타내도록 상기 제어 레지스터를 구성하는 단계를 포함하는, 방법.

#### 청구항 18

제16항에 있어서,

구성하는 단계는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 코어에 대응하는 각각의 수행 카운터를, 예산 시간 퀵텀에 대한 액세스 예산으로 구성하는 단계를 포함하고;

예산 시간 퀵텀에 대한 상기 액세스 예산은 상기 공유 리소스의 액세스 이벤트들의 유형과 관련되고;

구성하는 단계는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 코어에 대응하는 각각의 수행 카운터를, 오버플로 값에서 상기 예산 시간 퀵텀에 대한 상기 액세스 예산을 뺀 값으로 구성하는 단계를 포함하고;

구성하는 단계는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 코어에 대응하는 각각의 수행 카운터를, 오버플로시 인터럽트를 생성하도록 구성하는 단계를 포함하는, 방법.

#### 청구항 19

제16항에 있어서,

모니터링 및 제어하는 단계는 상기 복수의 코어에 의한 상기 공유 리소스의 액세스에 대해 모니터링하는 단계, 및 코어에 의한 상기 공유 리소스의 액세스의 검출시, 액세스하는 코어가 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되었다면 대응하는 수행 카운터를 업데이트하는 단계를 포함하는, 방법.

#### 청구항 20

제19항에 있어서,

모니터링 및 제어하는 단계는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화된 코어에 대해, 상기 코어가

예산 시간 퀀텀에 대한 그의 액세스 예산에 도달한 것임을 나타내는 상기 대응하는 수행 카운터로부터의 표시의 검출시, 상기 공유 리소스의 추가 액세스를 거부하는 단계를 포함하는, 방법.

**청구항 21**

제20항에 있어서,

모니터링 및 제어하는 단계는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화된 상기 코어에 대해, 대응하는 수행 카운터가 상기 코어가 상기 예산 시간 퀀텀에 대한 그의 액세스 예산에 도달한 것임을 나타내는 조건에 도달하는 결과로서 생성된 인터럽트에 응답하여, 상기 공유 리소스의 추가 액세스를 거부하기 위해 실행 제어를 인터럽트 핸들러에 넘겨주는 단계를 추가로 포함하는, 방법.

**청구항 22**

제21항에 있어서,

상기 방법은 상기 인터럽트 핸들러가, 실행 제어를 넘겨받을 때:

현재 코어를 결정하는 단계;

상기 현재 코어에 대해 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되어 있는지를 결정하는 단계; 및

상기 현재 코어에 대해 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되어 있다는 결정시, 상기 현재 코어의 현재 톱들이 상기 현재 코어의 다음 예산 체크 시간보다 크지를 추가로 결정하는 단계를 추가로 포함하는, 방법.

**청구항 23**

제22항에 있어서,

상기 방법은 상기 인터럽트 핸들러가, 상기 현재 코어의 현재 톱들이 상기 현재 코어의 다음 예산 체크 시간보다 크지 않다는 결정시, 상기 현재 코어의 현재 톱들이 상기 현재 코어의 상기 다음 예산 체크 시간과 같을 때까지 상기 현재 코어를 회전시키는 단계; 및

상기 현재 코어의 현재 톱들이 상기 현재 코어의 다음 예산 체크 시간보다 크다는 결정시,

상기 현재 코어의 상기 다음 예산 체크 시간을 상기 현재 코어의 상기 현재 톱들과 상기 현재 코어의 예산 시간 퀀텀의 합으로 설정하는 단계; 및

그 조건이 상기 인터럽트 핸들러가 실행 제어를 넘겨받는 것으로 이어진 상기 인터럽트를 야기하는 상기 현재 코어의 대응하는 수행 카운터를, 예산 시간 퀀텀에 대한 액세스 예산으로 리셋하는 단계를 추가로 포함하는, 방법.

**청구항 24**

저장된 명령어들을 갖는 하나 이상의 컴퓨터 판독 가능 매체로서, 상기 명령어들은 컴퓨팅 디바이스에 의한 실행에 응답하여 상기 컴퓨팅 디바이스로 하여금 제16항 내지 제23항의 방법들 중 어느 하나를 실시하게 하는, 하나 이상의 컴퓨터 판독 가능 매체.

**청구항 25**

컴퓨팅을 위한 장치로서,

복수의 코어를 갖는 프로세서;

상기 복수의 코어 간에 공유되도록 상기 프로세서와 결합된 리소스;

상기 복수의 코어와 대응되게 관련된 복수의 수행 카운터;

상기 복수의 수행 카운터 각각을 예산 시간 퀀텀에 대한 상기 공유 리소스에 대한 대응하는 코어의 액세스 예산으로 구성하는 수단; 및

상기 수행 카운터들을 이용하여, 상기 코어들의 상기 액세스 예산들에 따라, 상기 코어들에 의한 상기 공유 리

소스의 액세스를 모니터링 및 제어하는 수단을 포함하는, 장치.

**발명의 설명**

**기술 분야**

- [0001] 관련 출원
- [0002] 본 출원은 2015년 3월 25일자로 출원된, 발명의 명칭이 "공유 리소스 액세스 제어 방법 및 장치(SHARED RESOURCE ACCESS CONTROL METHOD AND APPARATUS)"인 미국 출원 제14/668,044호에 대한 우선권을 주장한다.
- [0003] 기술 분야
- [0004] 본 개시는 컴퓨팅 분야에 관한 것이다. 더 상세하게는, 본 개시는 멀티코어 프로세서의 다양한 코어들에 의한 공유 리소스의 액세스를 모니터링하고 제어하기 위한 장치 및 방법에 관한 것이다.

**배경 기술**

- [0005] 본 명세서에 제공된 배경 설명은 본 개시의 상황을 일반적으로 제시하기 위한 것이다. 본 명세서에 달리 나타내지 않는 한, 본 섹션에 설명된 자료들은 본 출원에서의 청구항들에 대한 종래 기술이 아니며, 본 섹션에 포함되어 있다고 해서 종래 기술인 것으로 인정되지는 않는다.
- [0006] 실시간 애플리케이션이 다수의 코어 및 공유 마지막 레벨 캐시(LLC)를 갖는 프로세서에서 다른 애플리케이션들과 동시에 실행될 때, 시스템의 공유 LLC 및 메모리는 다른 코어들에서 실행되는 다른 애플리케이션들이 단기간에 더 많은 수의 LLC 또는 메모리 참조를 발행할 때 혼잡해질 수 있다. 이러한 공유 리소스 혼잡은 더 높은 LLC 대기 시간 및 메모리 대기 시간으로 나타나며, 이는 실시간 애플리케이션의 실패를 야기할 수 있다.

**도면의 간단한 설명**

- [0007] 실시예들은 첨부 도면들과 함께 다음의 상세한 설명에 의해 용이하게 이해될 것이다. 이러한 설명을 용이하게 하기 위해서, 유사한 참조 번호들은 유사한 구조적 요소들을 지시한다. 실시예들은 첨부 도면들의 도들에서 제한으로서가 아니라 예로서 도시되어 있다.
  - 도 1은 다양한 실시예들에 따른, 본 개시의 공유 리소스 액세스 제어 기술을 갖는 컴퓨팅 배열을 도시한다.
  - 도 2는 다양한 실시예들에 따른, 그 각자의 액세스 예산들에 따라 공유 리소스의 코어 액세스를 모니터링하고 제어하기 위한 예시적인 프로세스를 도시한다.
  - 도 3은 다양한 실시예들에 따른, 그 각자의 액세스 예산들에 따라 공유 리소스의 코어 액세스를 모니터링하고 제어하기 위한 제어 레지스터 및 다수의 수행 카운터를 구성하기 위한 예시적인 프로세스를 도시한다.
  - 도 4는 다양한 실시예들에 따른, 그의 액세스 예산들에 도달하는 코어를 처리하기 위한 예시적인 프로세스를 도시한다.
  - 도 5는 다양한 실시예들에 따른, 본 개시의 양태들을 실행하기에 적합한 예시적인 컴퓨터 시스템을 도시한다.
  - 도 6은 다양한 실시예들에 따른, 장치가 본 개시의 양태들을 실시할 수 있게 하는 명령어들을 갖는 저장 매체를 도시한다.

**발명을 실시하기 위한 구체적인 내용**

- [0008] 공유 리소스의 코어 액세스를 모니터링 및 제어하는 것과 관련된 장치들, 방법들 및 저장 매체들이 본 명세서에 개시된다. 실시예들에서, 장치는 복수의 코어를 갖는 프로세서; 상기 복수의 코어 간에 공유되도록 상기 프로세서와 결합된 리소스(예를 들어, LLC 또는 메모리); 및 상기 복수의 코어의 상기 공유 리소스의 액세스 예산들을 저장하기 위해 상기 복수의 코어와 대응되게 관련된 복수의 수행 카운터를 포함한다. 상기 장치는 상기 수행 카운터들에 저장된 그 각자의 액세스 예산들에 따라 상기 복수의 코어에 의한 상기 공유 리소스의 액세스를 관리하기 위한 수행 모니터를 추가로 포함할 수 있다. 본 개시의 공유 리소스 액세스 제어 기술의 이들 및 다른 양태들이 더 상세히 설명될 것이다.
- [0009] 다음의 상세한 설명에서는, 공유 리소스 액세스 제어 기술은 본 명세서의 일부를 형성하는 첨부 도면들을 참조하여 설명될 것이며, 도면들에서 유사한 번호들은 전체에 걸쳐 유사한 부분들을 지시하고, 도면들에는 본 개시

의 내용이 실시될 수 있는 실시예들이 예시로서 도시되어 있다. 다른 실시예들이 이용될 수 있고, 본 개시의 범위 내에서 구조적 또는 논리적 변경들이 이루어질 수 있다는 점이 이해되어야 한다. 그러므로, 다음의 상세한 설명은 제한적인 의미로 간주되어서는 안 되며, 실시예들의 범위는 첨부된 청구항들 및 그것의 등가물들에 의해 규정된다.

- [0010] 본 개시의 양태들은 첨부 설명에서 개시된다. 본 개시의 대안 실시예들 및 그의 균등물들이 본 개시의 사상 또는 범위로부터 벗어나지 않고서 안출될 수 있다. 아래에 개시되는 동일한 요소들은 도면들에서 동일한 참조 번호들에 의해 지시된다는 점에 유의해야 한다.
- [0011] 다양한 동작들이 청구된 내용을 이해하는 데 가장 유용한 방식으로 복수의 개별적인 액션들 또는 동작들로서 차례로 설명될 수 있다. 그러나, 설명의 순서는 이러한 동작들이 반드시 순서 종속적이라고 암시하는 것으로 해석되어서는 안 된다. 특히, 이러한 동작들은 제시된 순서로 수행되지 않을 수 있다. 설명된 동작들은 설명된 실시예와 상이한 순서로 수행될 수 있다. 다양한 부가적인 동작들이 수행될 수 있고, 그리고/또는 설명된 동작들이 부가적인 실시예들에서 생략될 수 있다.
- [0012] 본 개시의 목적을 위해, "A 및/또는 B"라는 문구는 (A), (B), 또는 (A 및 B)를 의미한다. 본 개시의 목적을 위해, 문구 "A, B, 및/또는 C"는 (A), (B), (C), (A 및 B), (A 및 C), (B 및 C), 또는 (A, B 및 C)를 의미한다.
- [0013] 본 설명은 "실시예에서" 또는 "실시예들에서"라는 문구들을 이용할 수 있고, 이들 각각은 동일한 또는 상이한 실시예들 중 하나 이상을 언급할 수 있다. 더욱이, 본 개시의 실시예들과 관련하여 사용되는 바와 같은 "구비하는", "포함하는", "갖는", 및 그와 유사한 용어들은 동의어들이다.
- [0014] 본 명세서에서 사용되는, 용어 "모듈"은 ASIC(Application Specific Integrated Circuit), 전자 회로, 하나 이상의 소프트웨어 또는 펌웨어 프로그램을 실행하는 프로세서(공유, 전용, 또는 그룹) 및/또는 메모리(공유, 전용, 또는 그룹), 조합 로직 회로, 및/또는 설명된 기능을 제공하는 다른 적합한 컴포넌트(component)들을 언급할 수도 있거나, 그 일부일 수도 있거나, 이를 포함할 수 있다.
- [0015] 이제 도 1을 참조하면, 다양한 실시예들에 따른, 본 개시의 공유 리소스 액세스 제어 기술을 갖는 컴퓨팅 배열이 도시되어 있다. 도시된 바와 같이, 컴퓨팅 디바이스(100)는 다수의 코어들(104a-104d)을 갖는 프로세서(102) 및 코어들(104a-104d)과 결합되고 이들 사이에 공유되는 LLC, 메모리 등과 같은 하나 이상의 리소스(106)을 포함할 수 있다. 또한, 컴퓨팅 디바이스(100)는 코어들(104a-104d)의 각자의 액세스 예산들에 따라 코어들(104a-104d)에 의한 하나 이상의 리소스(106)의 액세스를 모니터링하고 제어하도록 구성된 수행 모니터(108)를 추가로 포함할 수 있다. 실시예들에서, 액세스 예산들은 예산 쿼텀마다, 예를 들어, x 밀리초에 대해, 및/또는 예를 들어 LLC 액세스와 메모리 액세스 간에 구별되는 액세스 이벤트 유형에 의해 설정될 수 있다.
- [0016] 실시예들에서, 컴퓨팅 디바이스(100)는 제어 레지스터(110) 및 수행 카운터들(112)을 추가로 포함할 수 있다. 제어 레지스터(110)는 코어들(104a-104d) 중 어느 것(있다면)이 공유 리소스(106)의 예산 기반 액세스 제어에 활성화되어야 하는지를 나타내는 제어 데이터를 저장하도록 구성될 수 있다. 추가로, 제어 레지스터(110)는 또한 다양한 액세스 예산들의 다음 예산 체크 시간을 나타내는 제어 데이터를 저장하도록 구성될 수 있다. 수행 카운터들(112)은 각자의 액세스 예산들을 저장하도록 구성될 수 있다. 실시예들에서, 각각의 수행 카운터(112)는 모든 또는 하나의 액세스 이벤트 유형에 대한 예산 시간 쿼텀의 액세스 예산을 저장하도록 구성될 수 있다. 또한, 각각의 수행 카운터(112)는 오버플로 값에서 예산 시간 쿼텀의 액세스 예산을 뺀 값을 저장함으로써, 예산 시간 쿼텀의 액세스 예산을 암시적으로 저장하도록 구성될 수 있고, 따라서 예산 시간 쿼텀 내에서 액세스 예산에 도달할 때 수행 카운터(112)의 오버플로가 발생할 것이다. 또한, 컴퓨팅 디바이스(100)는 프로세서(102)에 대한 인터럽트(114), 예를 들어 마스킹 가능하지 않은 인터럽트를 트리거/생성하기 위한 회로(도시되지 않음)를 추가로 포함할 수 있고, 각각의 수행 카운터(112)는 회로가 오버플로시 인터럽트를 트리거/생성하게 하도록 구성될 수 있다.
- [0017] 도 1에 도시된 실시예들에서, 제어 레지스터(110) 및 수행 카운터들(112)은 수행 모니터(108)의 일부로 도시되어 있다. 이러한 실시예들에서, 수행 모니터(108)는 하드웨어 컴포넌트의 펌웨어로서 구현된 모니터링 및 제어 로직을 갖는 하드웨어 컴포넌트일 수 있다. 대안의 실시예들에서, 제어 레지스터(110)는 프로세서(102)의 일부로서 구현될 수 있는 반면, 수행 모니터(112)는 코어들(104a-104d)의 일부로서 각각 구현될 수 있다. 이러한 대안의 실시예들에서, 모니터링 및 제어 로직은 컴퓨팅 디바이스(100)의 운영 체제(OS) 또는 하이퍼바이저(도시되지 않음)의 일부로서 구현될 수 있다. 또 다른 실시예들에서, 다수의 제어 레지스터가 대신 사용될 수 있다.
- [0018] 계속해서 도 1을 참조하면, 예산 시간 쿼텀에서 코어에 대한 공유 리소스에 대한 액세스 예산에 도달할 때 인터

럽트(114)가 트리거/생성되게 하도록 구성된 수행 카운터들(112)을 갖는 도 1의 실시예들에서, 컴퓨팅 디바이스(100)는 도시된 바와 같이, 프로세서(102), 수행 모니터(108), 및 공유 리소스들(106)과 결합된 판독 전용 메모리(ROM)(116)를 추가로 포함할 수 있다. ROM(116)은 트리거/생성 될 때 인터럽트들(114)을 서비스하기 위한 인터럽트 핸들러들(118)을 포함할 수 있다. 다른 실시예들에서, 인터럽트 핸들러들(118)은 다른 휘발성 또는 비휘발성 메모리에 상주할 수 있다.

- [0019] 이해의 편의상, 프로세서(102)는 4개의 코어(104a-104d)를 갖는 것으로 도시되었지만, 본 개시는 그렇게 제한되지 않는다. 본 개시의 공유 액세스 제어 기술은 2개 이상의 코어를 갖는 다수의 멀티코어 프로세서 중 임의의 하나와 함께 실시될 수 있다.
- [0020] 이제 도 2를 참조하면, 다양한 실시예들에 따른, 그 각자의 액세스 예산들에 따라 공유 리소스의 코어 액세스를 모니터링하고 제어하기 위한 예시적인 프로세스가 도시되어 있다. 도시된 바와 같이, 그 각자의 액세스 예산들에 따라 공유 리소스의 코어 액세스를 모니터링하고 제어하기 위한 프로세스(200)는 블록 202 내지 블록 208에서 수행되는 동작들을 포함할 수 있다. 동작들은 예를 들어, 전술한 도 1의 수행 모니터(108)에 의해 수행될 수 있다.
- [0021] 프로세스(200)는 블록 202에서 시작될 수 있다. 블록 202에서, 컴퓨팅 디바이스의 제어 레지스터 및 다양한 수행 카운터들이 구성될 수 있다. 전술한 바와 같이, 제어 레지스터는 컴퓨팅 디바이스의 프로세서의 프로세서 코어들 중 어느 것(있다면)이 활성화된 공유 리소스의 예산 기반 액세스 제어를 가져야 하는지를 나타내는 제어 데이터를 저장하도록 구성될 수 있다. 또한, 제어 레지스터는 공유 리소스의 예산 기반 액세스 제어를 가질 프로세서 코어 각각에 대해, (액세스 예산의 예산 시간 퀀텀에 기초하여) 다음 예산 체크 시간을 나타내는 제어 데이터를 저장하도록 구성될 수 있다. 또한 전술한 바와 같이, 각각의 수행 카운터는 공유 리소스의 예산 기반 액세스 제어가 활성화되어야 할 프로세서 코어에 대응하고, 프로세서 코어에 대한 모든 또는 하나의 액세스 이벤트 유형에 대한 예산 시간 퀀텀의 액세스 예산을 저장하도록 구성될 수 있다. 실시예들에서, 각각의 수행 카운터는 오버플로 값에서 예산 시간 퀀텀의 액세스 예산을 뺀 값을 저장함으로써, 예산 시간 퀀텀의 액세스 예산을 암시적으로 저장하도록 구성될 수 있고, 따라서 예산 시간 퀀텀 내에서 액세스 예산에 도달할 때 수행 카운터의 오버플로가 발생할 것이다. 또한, 각각의 수행 카운터(112)는 인터럽트 회로가 오버플로시 인터럽트를 트리거/생성하게 하도록 구성될 수 있다.
- [0022] 블록 204에서, 공유 리소스들에 대한 액세스가 모니터링될 수 있다. 프로세스(200)는 공유 리소스들에 대한 어떠한 액세스도 검출되지 않을 때 블록 204에 머무를 수 있다. 프로세서 코어에 의한 공유 리소스의 액세스의 검출시, 프로세스(200)는 블록 206으로 진행할 수 있다. 블록 206에서, 대응하는 수행 카운터는 액세스의 수행을 반영하도록 업데이트될 수 있다. 수행 카운터가 액세스 예산에 도달할 때 오버플로하도록 구성되는 실시예들에서, 수행 카운터는 액세스 수행을 반영하도록 증가될 수 있다. 오버플로를 야기하지 않는 수행 카운터의 업데이트시에, 프로세스(200)는 블록 204로 되돌아가고 그로부터 전술한 바와 같이 계속할 수 있다.
- [0023] 그러나, 오버플로를 야기하는 수행 카운터의 업데이트시, 프로세스(200)는 블록 208로 진행할 수 있다. 블록 208에서, 인터럽트가 서비스될 수 있다. 인터럽트를 서비스할 때, 프로세스(200)는 블록 204로 되돌아가고 그로부터 전술한 바와 같이 계속할 수 있다.
- [0024] 이제 도 3을 참조하면, 다양한 실시예들에 따른, 그 각자의 액세스 예산들에 따라 공유 리소스의 코어 액세스를 모니터링하고 제어하기 위한 제어 레지스터 및 다수의 수행 카운터를 구성하기 위한 예시적인 프로세스가 도시되어 있다. 도시된 바와 같이, 그 각자의 액세스 예산들에 따라 공유 리소스의 코어 액세스를 모니터링하고 제어하기 위한 제어 레지스터 및 다수의 수행 카운터를 구성하기 위한 프로세스(300)는 블록 302 내지 블록 320에서 수행되는 동작들을 포함할 수 있다. 실시예들에서, 동작들은 예를 들어, 도 1의 수행 모니터(108)에 의해 수행될 수 있다.
- [0025] 프로세스(300)는 블록 302에서 시작될 수 있다. 블록 302에서, 코어에 대한 (액세스 이벤트 유형에 대한) 공유 리소스에 대한 예산 기반 액세스 제어를 활성화하는 것과 관련된 스위치가 토글될 수 있다. 블록 304에서, (액세스 이벤트 유형에 대한) 공유 리소스에 대한 예산 기반 액세스 제어를 가질 코어가 결정될 수 있다. 블록 306에서, (액세스 이벤트 유형에 대한) 공유 리소스에 대한 예산 기반 액세스 제어가 코어에 대해 이미 활성화되어 있는지에 대한 결정이 이루어질 수 있다. (액세스 이벤트 유형에 대한) 공유 리소스에 대한 예산 기반 액세스 제어가 코어에 대해 이미 활성화되어 있다고 결정되면, 프로세스(300)는 블록 320으로 진행할 수 있다. 블록 320에서, 코어에 대한 (액세스 이벤트 유형에 대한) 공유 리소스에 대한 예산 기반 액세스 제어가 비활성화될 수 있다. 그 후, 프로세스(300)가 종료될 수 있다.

- [0026] 한편, (액세스 이벤트 유형에 대한) 공유 리소스에 대한 예산 기반 액세스 제어가 코어에 대해 이미 활성화되어 있지 않으면, 프로세스(300)는 블록 308로 진행할 수 있다. 블록 308에서, 현재 코어 틱들(core ticks)이 획득될 수 있다. 다음으로, 블록 310에서, (액세스 이벤트 유형에 대한) 코어에 대한 다음 예산 체크 시간이 설정될 수 있다. 실시예들에서, (액세스 이벤트 유형에 대한) 코어에 대한 다음 예산 체크 시간은 현재 틱들(액세스 이벤트 유형에 대한) 코어에 대한 예산 시간 퀀텀을 더한 합과 같도록 설정될 수 있다. 블록 312에서, (액세스 이벤트 유형에 대한) 공유 리소스에 대한 예산 기반 액세스 제어가 코어에 대해 활성화될 수 있다.
- [0027] 블록 314에서, 대응하는 수행 카운터는 전술한 바와 같이 코어에 대한 (액세스 이벤트 유형에 대한) 공유 리소스에 액세스하기 위한 예산 시간 퀀텀에 대한 액세스 예산을 저장하도록 설정될 수 있다. 실시예들에서, 액세스 예산 및 예산 시간 퀀텀은 디폴트로 되고/되거나 컴퓨팅 디바이스의 조작 경험으로부터 시간이 흐르면서 학습될 수 있다. 다른 실시예들에서, 관리자는 액세스 예산 및 예산 시간 퀀텀에 대해 프롬프트 받을 수 있다.
- [0028] 블록 316에서, 도달된 액세스 예산들을 다루기 위해 인터럽트 메커니즘을 사용하는 실시예들에서, 수행 카운터는 액세스 예산에 도달할 때(예를 들어, 수행 카운터가 오버플로할 때) 인터럽트(예를 들어, NMI)가 트리거되게 하도록 구성될 수 있다. 그 후, 블록 318에서, 수행 모니터는 액세스 예산들에 따라 (액세스 이벤트 유형에 대한) 코어에 의한 공유 리소스의 액세스의 모니터링 및 제어를 시작하도록 통지받을 수 있다. 그 후, 프로세스(300)가 종료될 수 있다.
- [0029] 이제 도 4를 참조하면, 다양한 실시예들에 따라 (액세스 이벤트 유형에 대한) 공유 리소스에 액세스하기 위해 그의 액세스 예산에 도달하는 코어를 처리하기 위한 예시적인 프로세스가 도시되어 있다. 도시된 바와 같이, (액세스 이벤트 유형에 대한) 공유 리소스에 액세스하기 위해 그의 액세스 예산에 도달하는 코어를 처리하기 위한 프로세스(400)는 블록 402 내지 블록 416에서 수행되는 동작들을 포함할 수 있다. 실시예들에서, 동작들은 예를 들어, 도 1의 인터럽트 핸들러(118)에 의해 수행될 수 있다.
- [0030] 프로세스(400)는 블록 402에서 시작될 수 있다. 블록 402에서, 실행 제어의 수신시, 현재 코어가 결정될 수 있다. 다음으로, 블록 404에서, 현재 코어에 대한 (액세스 이벤트 유형에 대한) 공유 리소스에 대한 예산 기반 액세스 제어가 활성화되어 있는지에 대한 결정이 이루어질 수 있다. 현재 코어에 대한 (액세스 이벤트 유형에 대한) 공유 리소스에 대한 예산 기반 액세스 제어가 활성화되지 않은 경우, 프로세스(400)는 블록 416으로 진행할 수 있고, 여기서 프로세스(400)는 종료될 수 있다. 인터럽트 핸들러 실시예들에서, 인터럽트 핸들러는 종료할 수 있다.
- [0031] 한편, 블록 404에서 현재 코어에 대한 (액세스 이벤트 유형에 대한) 공유 리소스에 대한 예산 기반 액세스 제어가 활성화되어 있는 것으로 결정되면, 프로세스(400)는 블록 406으로 진행할 수 있다. 블록 406에서, 예를 들어, 현재 코어의 특정 제어 레지스터들을 판독함으로써, 현재 코어에 대한 코어 틱들이 획득될 수 있다. 블록 408에서, 현재 틱들이 다음 예산 체크 시간보다 크지에 대한 결정이 이루어질 수 있다. 결정의 결과가 현재 틱들이 다음 예산 체크 시간보다 크다는 것을 나타내면, 프로세스(400)는 블록 412로 진행할 수 있다. 그러나, 결정의 결과가 현재 틱들이 다음 예산 체크 시간보다 크지 않다는 것을 나타내면, 프로세스(400)는 먼저 블록 410으로 진행할 수 있다. 블록 410에서, 현재 코어는 현재 코어의 현재 틱들이 다음 예산 체크 시간과 같을 때까지 회전될 수 있다.
- [0032] 블록 412에서, 블록 408으로부터 직접 진행되는 또는 블록 410에서 현재 코어를 회전시킨 후이든, 다음 예산 체크 시간은 현재 코어의 현재 틱들과 현재 코어의 (액세스 이벤트 유형에 대한) 예산 시간 퀀텀의 합과 같도록 업데이트될 수 있다. 그 후, 블록 414에서, 오버플로는 클리어될 수 있고, 수행 카운터는 다시 액세스 예산으로 리셋될 수 있다. 블록 414로부터, 프로세스(400)는 블록 416으로 진행할 수 있고, 여기서 프로세스(400)는 종료될 수 있다. 인터럽트 핸들러 실시예들에서, 인터럽트 핸들러는 종료될 수 있다.
- [0033] 도 5는 본 개시의 공유 액세스 제어 기술의 다양한 양태들을 실시하기에 적합한 예시적인 컴퓨터 시스템을 도시한다. 도시된 바와 같이, 컴퓨터(500)는 각각이 복수의 코어 및 이 코어들에 의해 공유되는 LLC(503)를 갖는 하나 이상의 멀티코어 프로세서(502)를 포함할 수 있다. 또한, 각각의 멀티코어 프로세서(502)는 전술한 도 1의 제어 레지스터(110)를 포함할 수 있고, 각각의 코어는 도 1의 대응하는 수행 카운터들(112)을 포함할 수 있다.
- [0034] 컴퓨터(500)는 ROM(505), 시스템 메모리(504) 및 대용량 저장 디바이스들(506)을 추가로 포함할 수 있다. 실시예들에서, ROM(505)은 다수의 인터럽트 핸들러, 특히 도 1의 인터럽트 핸들러(118)를 포함할 수 있고, 시스템 메모리(504)는 집합적으로 계산 로직(522)이라고 언급되는, 하이퍼바이저/운영 체제 및 다양한 애플리케이션들

을 구현하는 프로그래밍 명령어들의 작업 복사본을 저장하기 위해 사용될 수 있다. 실시예들에서, 하이퍼바이저/운영 체제는 도 1의 수행 모니터(108)의 모니터링 및 제어 로직을 포함할 수 있다. 대용량 저장 디바이스들(506)은 계산 로직(522)을 구현하는 프로그래밍 명령어들의 영구 복사본을 저장하기 위해 사용될 수 있다. 실시예들에서, 계산 로직(522)은 프로세서(들)(502)에 의해 지원되는 어셈블러 명령어들 또는 그러한 명령어들로 컴파일될 수 있는 예를 들어 C와 같은 고급 언어들에 의해 구현될 수 있다.

[0035] 또한, 컴퓨터(500)는 (디스플레이, 키보드, 커서 제어 등과 같은 I/O 디바이스들과 인터페이싱하기 위한) 입력/출력 디바이스 인터페이스들(508) 및 (네트워크 인터페이스 카드, 모뎀 등과 같은) 통신 디바이스들을 위한 통신 인터페이스들(510)을 포함할 수 있다. 요소들은 하나 이상의 버스를 나타낼 수 있는 시스템 버스(512)를 통해 서로 결합될 수 있다. 다수의 버스의 경우, 이들은 하나 이상의 버스 브리지(도시되지 않음)에 의해 브리징될 수 있다. 또한, 컴퓨터(500)는 (디스켓, 하드 드라이브, 콤팩트 디스크 판독 전용 메모리(CD-ROM) 등과 같은) 대용량 저장 디바이스들(506)을 포함할 수 있다.

[0036] 이들 요소(510 내지 512)의 수, 능력, 및/또는 용량은 컴퓨터(500)가 클라이언트 디바이스로 사용되는지 또는 서버 디바이스로 사용되는지에 따라 달라질 수 있다. 특히, 클라이언트 디바이스로서 사용되는 경우, 이들 요소(510 내지 512)의 능력 및/또는 용량은 클라이언트 디바이스가 거치형 디바이스인지 스마트폰, 컴퓨팅 태블릿, 울트라북 또는 랩톱과 같은 모바일 디바이스인지에 따라 달라질 수 있다. 본 개시의 공유 리소스 액세스 제어 기술을 제외하고, 요소들(510-512)의 구성들은 공지되어 있고, 따라서 더 설명하지 않는다.

[0037] 도 6은 장치에 의한 명령어들의 실행에 응답하여 장치로 하여금 본 개시의 선택된 양태들을 실행하게 하는 명령어들을 저장하는 데 사용하기에 적합한 예시적인 컴퓨터 판독 가능 비일시적 저장 매체를 도시한다. 도시된 바와 같이, 비일시적 컴퓨터 판독 가능 저장 매체(602)는 다수의 프로그래밍 명령어들(604)을 포함할 수 있다. 프로그래밍 명령어들(604)은 프로그래밍 명령어들의 실행에 응답하여 디바이스, 예를 들어 컴퓨터(500)가 도 1의 수행 모니터(108) 및/또는 인터럽트 핸들러(118)와 관련된 다양한 동작들을 수행할 수 있게 하도록 구성될 수 있다. 대안의 실시예들에서, 프로그래밍 명령어들(604)은 대신에 다수의 컴퓨터 판독 가능 비일시적 저장 매체(602) 상에 배치될 수 있다. 대안의 실시예들에서, 프로그래밍 명령어들(604)은 신호와 같은 컴퓨터 판독 가능 일시적 저장 매체(602) 상에 배치될 수 있다.

[0038] 도 5를 다시 참조하면, 일 실시예에서, 프로세서들(502) 중 적어도 하나는 수행 모니터(108) 및/또는 인터럽트 핸들러(118)의 모니터링 및 제어 로직을 갖는 메모리와 함께 패키징될 수 있다. 일 실시예에서, 프로세서들(502) 중 적어도 하나는 수행 모니터(108) 및/또는 인터럽트 핸들러(118)의 모니터링 및 제어 로직을 갖는 메모리와 함께 패키징되어 SiP(System in Package)를 형성할 수 있다. 일 실시예에서, 프로세서들(502) 중 적어도 하나는 수행 모니터(108) 및/또는 인터럽트 핸들러(118)의 모니터링 및 제어 로직을 갖는 메모리와 함께 동일한 다이 상에 집적될 수 있다. 일 실시예에서, 프로세서들(502) 중 적어도 하나는 수행 모니터(108) 및/또는 인터럽트 핸들러(118)의 모니터링 및 제어 로직을 갖는 메모리와 함께 패키징되어 SoC(System on Chip)를 형성할 수 있다. 적어도 하나의 실시예에서, SoC는 예를 들어, 웨어러블 디바이스, 스마트폰 또는 컴퓨팅 태블릿에서 이용될 수 있지만, 이에 제한되는 것은 아니다.

[0039] 따라서, 다음을 포함하지만 이에 제한되지 않는 본 개시의 다양한 예시적인 실시예들이 설명되었다.

[0040] 예 1은 복수의 코어를 갖는 프로세서; 및 상기 복수의 코어 간에 공유되도록 상기 프로세서와 결합된 리소스를 포함하는 컴퓨팅 디바이스일 수 있다. 이 컴퓨팅 디바이스는 상기 복수의 코어의 상기 공유 리소스의 액세스 예산들을 저장하기 위해 상기 복수의 코어와 대응되게 관련된 복수의 수행 카운터; 및 상기 프로세서, 상기 리소스 및 상기 수행 카운터들과 결합되어, 상기 수행 카운터들에 저장된 그 각자의 액세스 예산들에 따라 상기 복수의 코어에 의한 상기 공유 리소스의 액세스를 관리하기 위한 수행 모니터를 추가로 포함할 수 있다.

[0041] 예 2는 제어 레지스터를 추가로 포함하는, 예 1일 수 있고; 상기 수행 모니터는 상기 제어 레지스터와 추가로 결합되고, 상기 수행 카운터들과 함께 상기 제어 레지스터를 사용하여 그 각자의 액세스 예산들에 따라 상기 복수의 코어에 의한 상기 공유 리소스의 액세스를 관리할 수 있다.

[0042] 예 3은 예 2일 수 있고, 상기 수행 모니터는 상기 복수의 코어 중 어느 것이 상기 공유 리소스의 예산 기반 액세스 제어기가 활성화되어야 하는지를 나타내도록 상기 제어 레지스터를 추가로 구성할 수 있다.

[0043] 예 4는 예 3일 수 있고, 상기 수행 모니터는 상기 복수의 코어 중 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 각각의 코어에 대해, 예산 시간 퀀텀에 기초하여, 다음 예산 체크 시간을 나타내도록 상기 제어 레지스터를 추가로 구성할 수 있다.

- [0044] 예 5는 예 2일 수 있고, 상기 제어 레지스터는 상기 수행 모니터의 일부일 수 있다.
- [0045] 예 6은 예 2일 수 있고, 상기 제어 레지스터는 상기 프로세서의 일부일 수 있다.
- [0046] 예 7은 예 1일 수 있고, 상기 수행 모니터는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 코어에 대응하는 각각의 수행 카운터를, 예산 시간 퀀텀에 대한 액세스 예산으로 구성할 수 있다.
- [0047] 예 8은 예 7일 수 있고, 예산 시간 퀀텀에 대한 상기 액세스 예산은 상기 공유 리소스의 액세스 이벤트들의 유형과 관련될 수 있다.
- [0048] 예 9는 예 7일 수 있고, 상기 수행 모니터는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 코어에 대응하는 각각의 수행 카운터를, 오버플로 값에서 상기 예산 시간 퀀텀에 대한 상기 액세스 예산을 뺀 값으로 구성할 수 있다.
- [0049] 예 10은 예 9일 수 있고, 상기 수행 모니터는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 코어에 대응하는 각각의 수행 카운터를, 오버플로시 인터럽트를 생성하도록 구성할 수 있다.
- [0050] 예 11은 예 7일 수 있고, 상기 수행 카운터는 상기 수행 모니터의 일부일 수 있다.
- [0051] 예 12는 예 7일 수 있고, 상기 수행 카운터는 상기 프로세서의 일부일 수 있다.
- [0052] 예 13은 예 1-12일 수 있고, 상기 수행 모니터는 상기 복수의 코어에 의한 상기 공유 리소스의 액세스에 대해 모니터링할 수 있고, 코어에 의한 상기 공유 리소스의 액세스의 검출시, 액세스하는 코어가 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되었다면 대응하는 수행 카운터를 업데이트한다.
- [0053] 예 14는 예 13일 수 있고, 상기 수행 모니터는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화된 코어에 대해, 상기 코어가 예산 시간 퀀텀에 대한 그의 액세스 예산에 도달한 것임을 나타내는 상기 대응하는 수행 카운터로부터의 표시의 검출시, 상기 공유 리소스의 추가 액세스를 거부할 수 있다.
- [0054] 예 15는 예 14일 수 있고, 상기 공유 리소스의 예산 기반 액세스 제어가 활성화된 상기 코어에 대해, 대응하는 수행 카운터가 상기 코어가 상기 예산 시간 퀀텀에 대한 그의 액세스 예산에 도달한 것임을 나타내는 조건에 도달하는 결과로서 생성된 인터럽트에 응답하여, 상기 공유 리소스의 추가 액세스를 거부하기 위해 실행 제어를 넘겨받을 인터럽트 핸들러를 추가로 포함한다.
- [0055] 예 16은 예 15일 수 있고, 실행 제어를 넘겨받을 때, 상기 인터럽트 핸들러는: 현재 코어를 결정하고; 상기 현재 코어에 대해 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되어 있는지를 결정하고; 상기 현재 코어에 대해 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되어 있다는 결정시, 상기 현재 코어의 현재 틱들이 상기 현재 코어의 다음 예산 체크 시간보다 큰지를 추가로 결정할 수 있다.
- [0056] 예 17은 예 16일 수 있고, 상기 현재 코어의 현재 틱들이 상기 현재 코어의 다음 예산 체크 시간보다 크지 않다는 결정시, 상기 인터럽트 핸들러는 상기 현재 코어의 현재 틱들이 상기 현재 코어의 상기 다음 예산 체크 시간과 같을 때까지 상기 현재 코어를 회전시킨다.
- [0057] 예 18은 예 16일 수 있고, 상기 현재 코어의 현재 틱들이 상기 현재 코어의 다음 예산 체크 시간보다 크다는 결정시, 상기 인터럽트 핸들러는: 상기 현재 코어의 상기 다음 예산 체크 시간을 상기 현재 코어의 상기 현재 틱들과 상기 현재 코어의 예산 시간 퀀텀의 합으로 설정하고; 그 조건이 상기 인터럽트 핸들러가 실행 제어를 넘겨받는 것으로 이어진 상기 인터럽트를 야기하는 상기 현재 코어의 대응하는 수행 카운터를, 예산 시간 퀀텀에 대한 액세스 예산으로 리셋한다.
- [0058] 예 19는 예 1일 수 있고, 상기 수행 모니터는 상기 컴퓨팅 디바이스의 운영 체제 또는 하이퍼바이저의 일부일 수 있다.
- [0059] 예 20은 컴퓨팅 디바이스상의 공유 리소스에 대한 코어 액세스들을 제어하는 방법일 수 있고, 이 방법은: 상기 컴퓨팅 디바이스의 수행 모니터에 의해, 상기 컴퓨팅 디바이스의 프로세서의 복수의 코어의 복수의 대응하는 수행 카운터 각각을 예산 시간 퀀텀에 대한 상기 공유 리소스에 대한 대응하는 코어의 액세스 예산으로 구성하는 단계; 및 상기 수행 모니터에 의해, 상기 수행 카운터를 이용하여, 상기 코어들의 상기 액세스 예산들에 따라, 상기 코어들에 의한 상기 공유 리소스의 액세스를 모니터링 및 제어하는 단계를 포함한다.
- [0060] 예 21은 예 20일 수 있고, 구성하는 단계는 상기 액세스 예산과 관련된 제어 데이터로 제어 레지스터를 구성하는 단계를 추가로 포함할 수 있고, 모니터링 및 제어하는 단계는 상기 제어 레지스터를 이용하는 단계를 추가로

포함한다.

- [0061] 예 22는 예 21일 수 있고, 구성하는 단계는 상기 복수의 코어 중 어느 것이 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되어야 하는지를 나타내도록 상기 제어 레지스터를 구성하는 단계를 포함할 수 있다.
- [0062] 예 23은 예 22일 수 있고, 구성하는 단계는 상기 복수의 코어 중 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 각각의 코어에 대해, 예산 시간 퀀텀에 기초하여, 다음 예산 체크 시간을 나타내도록 상기 제어 레지스터를 구성하는 단계를 포함할 수 있다.
- [0063] 예 24는 예 20일 수 있고, 구성하는 단계는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 코어에 대응하는 각각의 수행 카운터를, 예산 시간 퀀텀에 대한 액세스 예산으로 구성하는 단계를 포함할 수 있다.
- [0064] 예 25는 예 24일 수 있고, 예산 시간 퀀텀에 대한 상기 액세스 예산은 상기 공유 리소스의 액세스 이벤트들의 유형과 관련된다.
- [0065] 예 26은 예 24일 수 있고, 구성하는 단계는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 코어에 대응하는 각각의 수행 카운터를, 오버플로 값에서 상기 예산 시간 퀀텀에 대한 상기 액세스 예산을 뺀 값으로 구성하는 단계를 포함할 수 있다.
- [0066] 예 27은 예 26일 수 있고, 구성하는 단계는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 코어에 대응하는 각각의 수행 카운터를, 오버플로시 인터럽트를 생성하도록 구성하는 단계를 포함할 수 있다.
- [0067] 예 28은 예 20-27일 수 있고, 모니터링 및 제어하는 단계는 상기 복수의 코어에 의한 상기 공유 리소스의 액세스에 대해 모니터링하는 단계, 및 코어에 의한 상기 공유 리소스의 액세스의 검출시, 액세스하는 코어가 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되었다면 대응하는 수행 카운터를 업데이트하는 단계를 포함할 수 있다.
- [0068] 예 29는 예 28일 수 있고, 모니터링 및 제어하는 단계는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화된 코어에 대해, 상기 코어가 예산 시간 퀀텀에 대한 그의 액세스 예산에 도달한 것임을 나타내는 상기 대응하는 수행 카운터로부터의 표시의 검출시, 상기 공유 리소스의 추가 액세스를 거부하는 단계를 포함할 수 있다.
- [0069] 예 30은 예 29일 수 있고, 모니터링 및 제어하는 단계는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화된 상기 코어에 대해, 대응하는 수행 카운터가 상기 코어가 상기 예산 시간 퀀텀에 대한 그의 액세스 예산에 도달한 것임을 나타내는 조건에 도달하는 결과로서 생성된 인터럽트에 응답하여, 상기 공유 리소스의 추가 액세스를 거부하기 위해 실행 제어를 인터럽트 핸들러에 넘겨주는 단계를 추가로 포함할 수 있다.
- [0070] 예 31은 예 30일 수 있고, 상기 방법은 상기 인터럽트 핸들러가, 실행 제어를 넘겨받을 때: 현재 코어를 결정하는 단계; 상기 현재 코어에 대해 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되어 있는지를 결정하는 단계; 및 상기 현재 코어에 대해 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되어 있다는 결정시, 상기 현재 코어의 현재 틱들이 상기 현재 코어의 다음 예산 체크 시간보다 크지를 추가로 결정하는 단계를 추가로 포함할 수 있다.
- [0071] 예 32는 예 31일 수 있고, 상기 방법은 상기 인터럽트 핸들러가, 상기 현재 코어의 현재 틱들이 상기 현재 코어의 다음 예산 체크 시간보다 크지 않다는 결정시, 상기 현재 코어의 현재 틱들이 상기 현재 코어의 상기 다음 예산 체크 시간과 같을 때까지 상기 현재 코어를 회전시키는 단계를 추가로 포함할 수 있다.
- [0072] 예 33은 예 31일 수 있고, 상기 방법은 상기 인터럽트 핸들러가, 상기 현재 코어의 현재 틱들이 상기 현재 코어의 다음 예산 체크 시간보다 크다는 결정시, 상기 현재 코어의 상기 다음 예산 체크 시간을 상기 현재 코어의 상기 현재 틱들과 상기 현재 코어의 예산 시간 퀀텀의 합으로 설정하는 단계; 및 그 조건이 상기 인터럽트 핸들러가 실행 제어를 넘겨받는 것으로 이어진 상기 인터럽트를 야기하는 상기 현재 코어의 대응하는 수행 카운터를, 예산 시간 퀀텀에 대한 액세스 예산으로 리셋하는 단계를 추가로 포함할 수 있다.
- [0073] 예 34는 저장된 명령어들을 갖는 하나 이상의 컴퓨터 판독 가능 매체일 수 있고, 상기 명령어들은 컴퓨팅 디바이스에 의한 실행에 응답하여 상기 컴퓨팅 디바이스로 하여금: 상기 컴퓨팅 디바이스의 프로세서의 복수의 코어의 복수의 대응하는 수행 카운터 각각을 예산 시간 퀀텀에 대한 상기 컴퓨팅 디바이스의 공유 리소스에 대한 대응하는 코어의 액세스 예산으로 구성하고; 상기 수행 카운터들을 이용하여, 상기 코어들의 상기 액세스 예산들에 따라, 상기 코어들에 의한 상기 공유 리소스의 액세스를 모니터링 및 제어하도록 야기한다.
- [0074] 예 35는 예 34일 수 있고, 상기 컴퓨팅 디바이스는 제어 레지스터를 구성하고, 상기 수행 카운터들과 함께 상기

제어 레지스터를 이용하여, 상기 복수의 코어에 의한 상기 공유 리소스의 액세스를 그 각자의 액세스 예산들에 따라 관리하도록 추가로 야기될 수 있다.

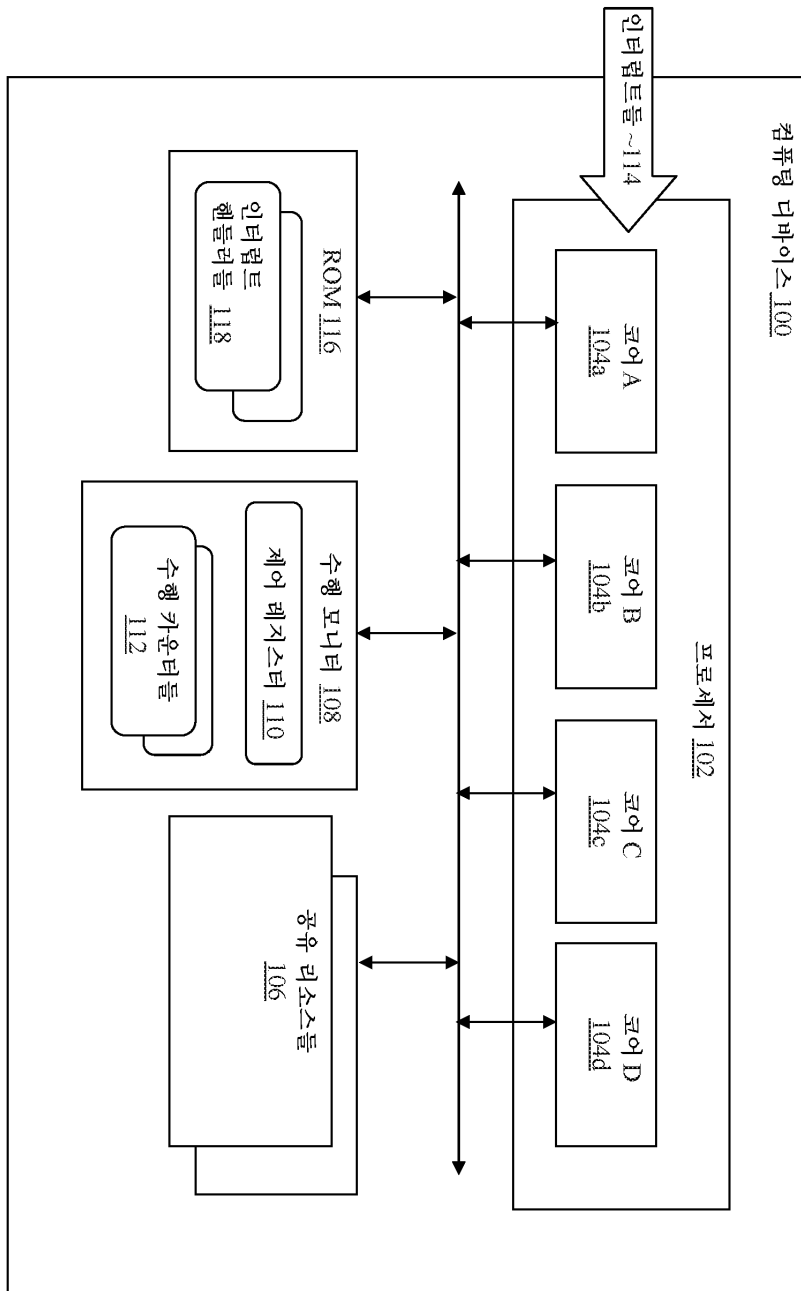
- [0075] 예 36은 예 35일 수 있고, 상기 컴퓨팅 디바이스는 상기 복수의 코어 중 어느 것이 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되어야 하는지를 나타내도록 상기 제어 레지스터를 구성하도록 추가로 야기될 수 있다.
- [0076] 예 37은 예 36일 수 있고, 상기 컴퓨팅 디바이스는 상기 복수의 코어 중 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 각각의 코어에 대해, 예산 시간 퀀텀에 기초하여, 다음 예산 체크 시간을 나타내도록 상기 제어 레지스터를 구성하도록 추가로 야기될 수 있다.
- [0077] 예 38은 예 34일 수 있고, 상기 컴퓨팅 디바이스는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 코어에 대응하는 각각의 수행 카운터를, 예산 시간 퀀텀에 대한 액세스 예산으로 구성하도록 추가로 야기될 수 있다.
- [0078] 예 39는 예 38일 수 있고, 예산 시간 퀀텀에 대한 상기 액세스 예산은 상기 공유 리소스의 액세스 이벤트들의 유형과 관련될 수 있다.
- [0079] 예 40은 예 38일 수 있고, 상기 컴퓨팅 디바이스는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 코어에 대응하는 각각의 수행 카운터를, 오버플로 값에서 상기 예산 시간 퀀텀에 대한 상기 액세스 예산을 뺀 값으로 구성하도록 추가로 야기될 수 있다.
- [0080] 예 41은 예 40일 수 있고, 상기 컴퓨팅 디바이스는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 코어에 대응하는 각각의 수행 카운터를, 오버플로시 인터럽트를 생성하도록 구성하도록 추가로 야기될 수 있다.
- [0081] 예 42는 예 34-41일 수 있고, 상기 컴퓨팅 디바이스는 상기 복수의 코어에 의한 상기 공유 리소스의 액세스에 대해 모니터링하고, 및 코어에 의한 상기 공유 리소스의 액세스의 검출시, 액세스하는 코어가 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되었다면 대응하는 수행 카운터를 업데이트하도록 추가로 야기될 수 있다.
- [0082] 예 43은 예 42일 수 있고, 상기 컴퓨팅 디바이스는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화된 코어에 대해, 상기 코어가 예산 시간 퀀텀에 대한 그의 액세스 예산에 도달한 것임을 나타내는 상기 대응하는 수행 카운터로부터의 표시의 검출시, 상기 공유 리소스의 추가 액세스를 거부하도록 추가로 야기될 수 있다.
- [0083] 예 44는 예 43일 수 있고, 상기 컴퓨팅 디바이스는 상기 공유 리소스의 예산 기반 액세스 제어가 활성화된 상기 코어에 대해, 대응하는 수행 카운터가 상기 코어가 상기 예산 시간 퀀텀에 대한 그의 액세스 예산에 도달한 것임을 나타내는 조건에 도달하는 결과로서 생성된 인터럽트에 응답하여, 상기 공유 리소스의 추가 액세스를 거부하기 위해 실행 제어를 인터럽트 핸들러에 넘겨주도록 추가로 야기될 수 있다.
- [0084] 예 45는 예 44일 수 있고, 실행 제어를 넘겨받을 때, 상기 인터럽트 핸들러는: 현재 코어를 결정하고; 상기 현재 코어에 대해 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되어 있는지를 결정하고; 상기 현재 코어에 대해 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되어 있다는 결정시, 상기 현재 코어의 현재 틱들이 상기 현재 코어의 다음 예산 체크 시간보다 큰지를 추가로 결정할 수 있다.
- [0085] 예 46은 예 45일 수 있고, 상기 현재 코어의 현재 틱들이 상기 현재 코어의 다음 예산 체크 시간보다 크지 않다는 결정시, 상기 인터럽트 핸들러는 상기 현재 코어의 현재 틱들이 상기 현재 코어의 상기 다음 예산 체크 시간과 같을 때까지 상기 현재 코어를 회전시킬 수 있다.
- [0086] 예 47은 예 46일 수 있고, 상기 현재 코어의 현재 틱들이 상기 현재 코어의 다음 예산 체크 시간보다 크다는 결정시, 상기 인터럽트 핸들러는: 상기 현재 코어의 상기 다음 예산 체크 시간을 상기 현재 코어의 상기 현재 틱들과 상기 현재 코어의 예산 시간 퀀텀의 합으로 설정하고; 그 조건이 상기 인터럽트 핸들러가 실행 제어를 넘겨받는 것으로 이어진 상기 인터럽트를 야기하는 상기 현재 코어의 대응하는 수행 카운터를, 예산 시간 퀀텀에 대한 액세스 예산으로 리셋할 수 있다.
- [0087] 예 48은 컴퓨팅을 위한 장치일 수 있고, 이 장치는: 복수의 코어를 갖는 프로세서; 상기 복수의 코어 간에 공유되도록 상기 프로세서와 결합된 리소스; 및 상기 복수의 코어와 대응되게 관련된 복수의 수행 카운터를 포함한다. 이 장치는 복수의 수행 카운터 각각을 예산 시간 퀀텀에 대한 상기 공유 리소스에 대한 대응하는 코어의 액세스 예산으로 구성하는 수단; 및 상기 수행 카운터를 이용하여, 상기 코어들의 상기 액세스 예산들에 따라, 상기 코어들에 의한 상기 공유 리소스의 액세스를 모니터링 및 제어하는 수단을 추가로 포함할 수 있다.
- [0088] 예 49는 예 48일 수 있고, 구성하는 수단은 상기 액세스 예산과 관련된 제어 데이터로 제어 레지스터를 구성하

는 수단을 추가로 포함할 수 있고, 모니터링 및 제어하는 것은 상기 제어 레지스터를 이용하는 것을 추가로 포함한다.

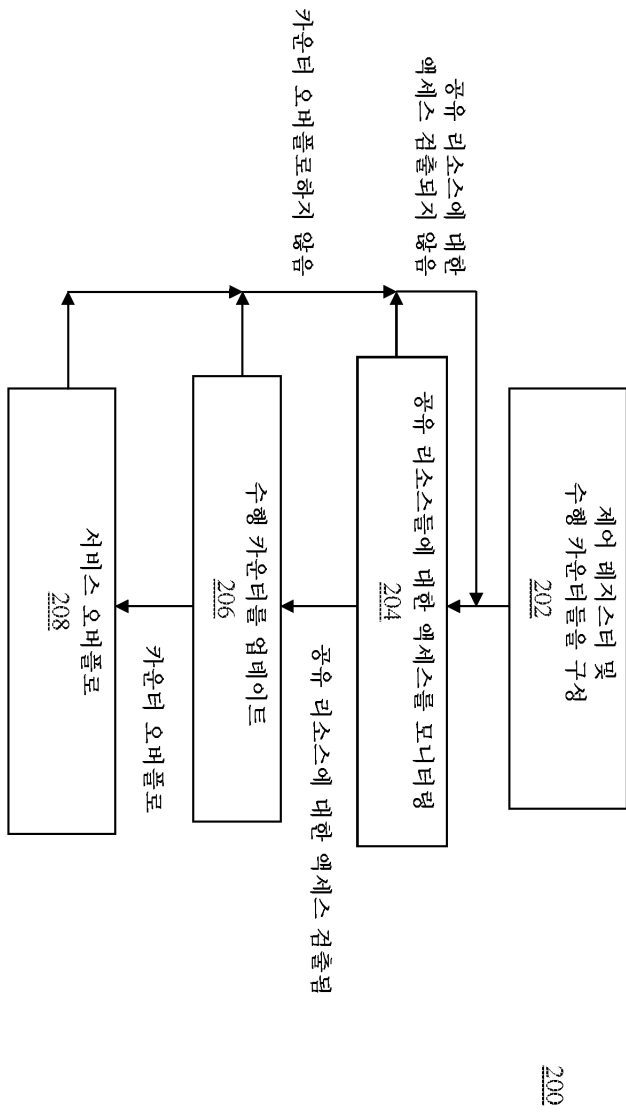
- [0089] 예 50은 예 49일 수 있고, 구성하는 수단은 상기 복수의 코어 중 어느 것이 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되어야 하는지를 나타내도록 상기 제어 레지스터를 구성하는 수단을 포함할 수 있다.
- [0090] 예 51은 예 50일 수 있고, 구성하는 수단은 상기 복수의 코어 중 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 각각의 코어에 대해, 예산 시간 퀀텀에 기초하여, 다음 예산 체크 시간을 나타내도록 상기 제어 레지스터를 구성하는 수단을 포함할 수 있다.
- [0091] 예 52는 예 48일 수 있고, 구성하는 수단은 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 코어에 대응하는 각각의 수행 카운터를, 예산 시간 퀀텀에 대한 액세스 예산으로 구성하는 수단을 포함할 수 있다.
- [0092] 예 53은 예 52일 수 있고, 예산 시간 퀀텀에 대한 상기 액세스 예산은 상기 공유 리소스의 액세스 이벤트들의 유형과 관련될 수 있다.
- [0093] 예 54는 예 52일 수 있고, 구성하는 수단은 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 코어에 대응하는 각각의 수행 카운터를, 오버플로 값에서 상기 예산 시간 퀀텀에 대한 상기 액세스 예산을 뺀 값으로 구성하는 수단을 포함할 수 있다.
- [0094] 예 55는 예 54일 수 있고, 구성하는 수단은 상기 공유 리소스의 예산 기반 액세스 제어가 활성화될 코어에 대응하는 각각의 수행 카운터를, 오버플로시 인터럽트를 생성하도록 구성하는 수단을 포함할 수 있다.
- [0095] 예 56은 예 48-55일 수 있고, 모니터링 및 제어하는 수단은 상기 복수의 코어에 의한 상기 공유 리소스의 액세스에 대해 모니터링하는 수단, 및 코어에 의한 상기 공유 리소스의 액세스의 검출시, 액세스하는 코어가 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되었다면 대응하는 수행 카운터를 업데이트하는 수단을 포함할 수 있다.
- [0096] 예 57은 예 56일 수 있고, 모니터링 및 제어하는 수단은 상기 공유 리소스의 예산 기반 액세스 제어가 활성화된 코어에 대해, 상기 코어가 예산 시간 퀀텀에 대한 그의 액세스 예산에 도달한 것임을 나타내는 상기 대응하는 수행 카운터로부터의 표시의 검출시, 상기 공유 리소스의 추가 액세스를 거부하는 수단을 포함할 수 있다.
- [0097] 예 58은 예 57일 수 있고, 모니터링 및 제어하는 수단은 상기 공유 리소스의 예산 기반 액세스 제어가 활성화된 상기 코어에 대해, 대응하는 수행 카운터가 상기 코어가 상기 예산 시간 퀀텀에 대한 그의 액세스 예산에 도달한 것임을 나타내는 조건에 도달하는 결과로서 생성된 인터럽트에 응답하여, 상기 공유 리소스의 추가 액세스를 거부하기 위해 실행 제어를 인터럽트 핸들러에 넘겨주는 수단을 추가로 포함할 수 있다.
- [0098] 예 59는 예 58일 수 있고, 상기 인터럽트 핸들러는 실행 제어를 넘겨받을 때, 현재 코어를 결정하는 수단; 상기 현재 코어에 대해 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되어 있는지를 결정하는 수단; 및 상기 현재 코어에 대해 상기 공유 리소스의 예산 기반 액세스 제어가 활성화되어 있다는 결정시, 상기 현재 코어의 현재 틱들이 상기 현재 코어의 다음 예산 체크 시간보다 큰지를 추가로 결정하는 수단을 포함할 수 있다.
- [0099] 예 60은 예 59일 수 있고, 상기 인터럽트 핸들러는, 상기 현재 코어의 현재 틱들이 상기 현재 코어의 다음 예산 체크 시간보다 크지 않다는 결정시, 상기 현재 코어의 현재 틱들이 상기 현재 코어의 상기 다음 예산 체크 시간과 같을 때까지 상기 현재 코어를 회전시키는 수단을 추가로 포함할 수 있다.
- [0100] 예 61은 예 59일 수 있고, 상기 인터럽트 핸들러는, 상기 현재 코어의 현재 틱들이 상기 현재 코어의 다음 예산 체크 시간보다 크다는 결정시, 상기 현재 코어의 상기 다음 예산 체크 시간을 상기 현재 코어의 상기 현재 틱들과 상기 현재 코어의 예산 시간 퀀텀의 합으로 설정하는 수단; 및 그 조건이 상기 인터럽트 핸들러가 실행 제어를 넘겨받는 것으로 이어진 상기 인터럽트를 야기하는 상기 현재 코어의 대응하는 수행 카운터를, 예산 시간 퀀텀에 대한 액세스 예산으로 리셋하는 수단을 추가로 포함할 수 있다.
- [0101] 본 개시의 사상 또는 범위를 벗어나지 않고 개시된 디바이스 및 관련 방법의 개시된 실시예들에서 다양한 수정 및 변경이 이루어질 수 있음은 이 분야의 기술자에게 명백할 것이다. 따라서, 본 개시는, 이러한 수정들 및 변경들이 임의의 청구항들 및 그들의 등가물들 내에 속한다면, 본 개시가 위에 개시된 실시예들의 수정들 및 변경들을 커버하는 것으로 의도된다.

도면

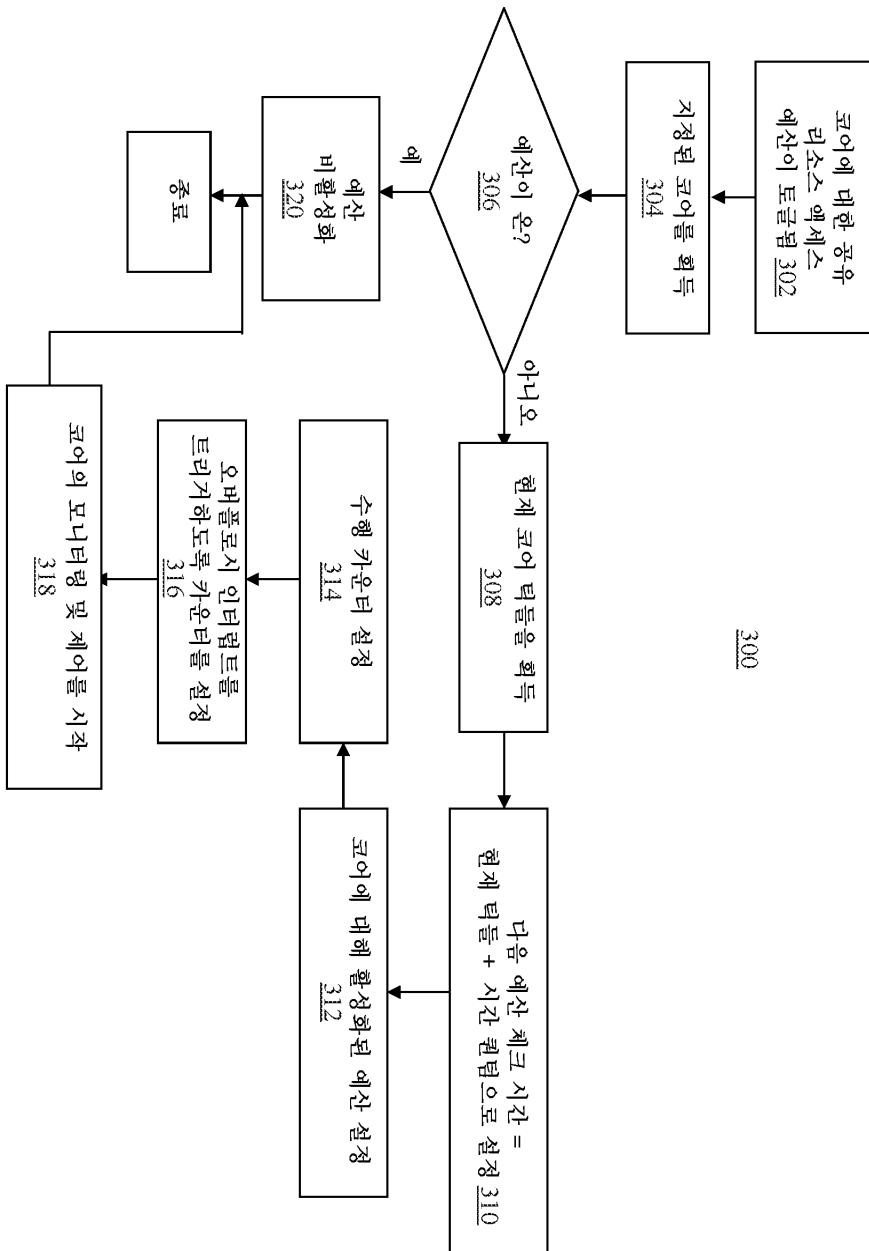
도면1



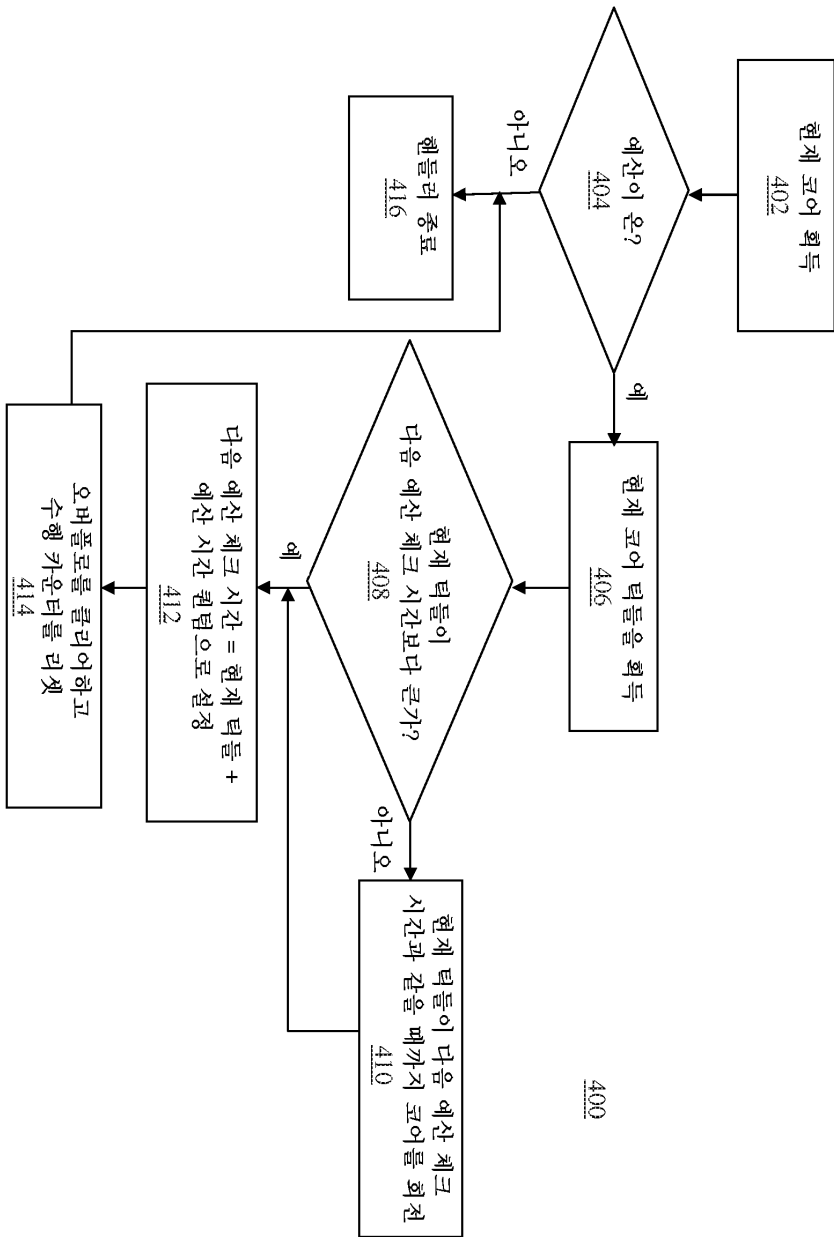
도면2



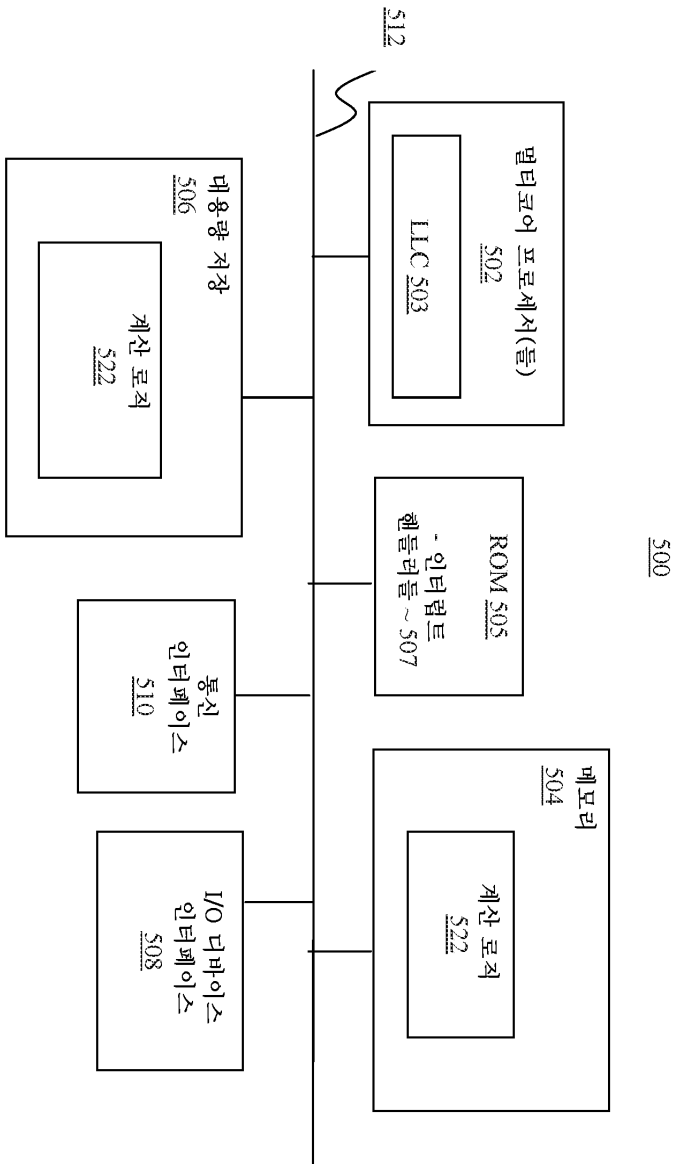
도면3



도면4



도면5



도면6

