



**(19) 대한민국특허청(KR)**  
**(12) 등록특허공보(B1)**

(45) 공고일자 2018년04월03일  
(11) 등록번호 10-1844809  
(24) 등록일자 2018년03월28일

(51) 국제특허분류(Int. Cl.)  
G06F 13/42 (2006.01) G06F 13/00 (2006.01)  
G06F 9/46 (2006.01)  
(52) CPC특허분류  
G06F 13/4273 (2013.01)  
G06F 13/00 (2013.01)  
(21) 출원번호 10-2016-7005183  
(22) 출원일자(국제) 2013년09월30일  
심사청구일자 2016년02월26일  
(85) 번역문제출일자 2016년02월26일  
(65) 공개번호 10-2016-0038003  
(43) 공개일자 2016년04월06일  
(86) 국제출원번호 PCT/US2013/062751  
(87) 국제공개번호 WO 2015/047422  
국제공개일자 2015년04월02일  
(56) 선행기술조사문헌  
KR1020060046226 A  
WO2013027951 A1  
US20100235672 A1  
US6232820 A

(73) 특허권자  
인텔 코포레이션  
미합중국 캘리포니아 95054 산타클라라 미션 칼리지 블러바드 2200  
(72) 발명자  
수구마르 수레쉬  
인도 방갈로르 560037 에어포트 바르수르 로드 푸르바 리비에라 알에프-비-912  
쿠마쉬카르 마헤쉬 케이  
인도 방갈로르 560066 화이트필드 바르수르 로드 라마콘다나할리 스카일라크 그린스 #38에이  
(74) 대리인  
제일특허법인

전체 청구항 수 : 총 25 항

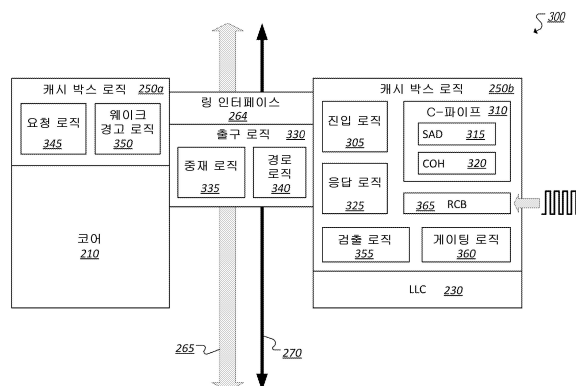
심사관 : 김병균

(54) 발명의 명칭 **클록 게이팅 제어를 위한 조기의 웨이크-경고**

**(57) 요약**

특정 캐시 기록과 연관된 요청은 상호접속부를 통해 캐시 बैं크와 연관된 시스템 구성요소로 전송되도록 생성된다. 요청이 도착한다는 것을 시스템 구성요소에 표시하기 위한 웨이크-경고 신호는 전용 웨이크-경고 채널을 통해 전송된다. 웨이크-경고 신호는 디스에이블 클록이 인에이블 상태로 연게이팅되게 한다. 그후, 요청은 시스템 구성요소로 전송된다.

**대표도**



(52) CPC특허분류

*G06F 9/46* (2013.01)

(72) 발명자

**펠 라홀**

인도 방갈로르 560037 바르수르 메인 로드 푸르바  
리비에라 알지201

**무스라사날루르 스리드하르**

인도 방갈로르 560034 카르나타카 8번 메인 코라망  
갈라 3번 블럭 아파트먼트 # 204 엠바스시 트랜쿼  
일 #22

## 명세서

### 청구범위

#### 청구항 1

로직을 포함하되, 상기 로직은,

특정 캐시 기록(cache record)과 연관된 요청을 생성하고 - 상기 요청은 상호접속부(interconnect)를 통해 캐시 बैं크와 연관된 시스템 구성요소로 전송됨 - ,

상기 요청이 도착한다는 것을 상기 시스템 구성요소에 표시하기 위한 웨이크-경고 신호(wake-warn signal)를 전용 웨이크-경고 채널을 통해 전송하고 - 상기 웨이크-경고 신호는 디스에이블 클록(disabled clock)이 인에이블 상태(enabled state)로 언게이팅(ungated)되게 함 - ,

상기 요청을 상기 시스템 구성요소로 전송하는 장치.

#### 청구항 2

제 1 항에 있어서,

상기 요청은 상기 전용 웨이크-경고 채널과 별개인 상호접속부를 사용하여 상기 시스템 구성요소로 전송되는 장치.

#### 청구항 3

제 2 항에 있어서,

상기 상호접속부는 링 상호접속부를 포함하는 장치.

#### 청구항 4

제 2 항에 있어서,

상기 웨이크-경고 채널은 상기 장치를 적어도 상기 시스템 구성요소에 접속하는 링으로서 배향되는 장치.

#### 청구항 5

제 1 항에 있어서,

상기 시스템 구성요소는 상기 캐시 बैं크에 대응하는 캐시 박스 로직을 포함하고, 상기 클록은 상기 캐시 박스 로직에 의해 활용되는 장치.

#### 청구항 6

제 1 항에 있어서,

상기 로직은 또한 상기 요청의 전송 전에 상기 요청의 버퍼링(buffering)을 식별하고, 상기 요청의 버퍼링에 기초하여 상기 웨이크-경고 신호의 전송 전에 상기 웨이크-경고 신호를 버퍼링하는 장치.

#### 청구항 7

제 1 항에 있어서,

상기 웨이크-경고 신호는 상기 시스템 구성요소에서의 상기 요청의 도착 전에 상기 시스템 구성요소에 도착하는 장치.

#### 청구항 8

제 7 항에 있어서,

상기 웨이크-경고 신호는 상기 요청의 전송과 실질적으로 동시에 전송되는 장치.

#### 청구항 9

제 1 항에 있어서,

상기 웨이크-경고 채널은 양방향 채널이고,  
상기 웨이크-경고 신호는 상기 웨이크-경고 채널 상에서 양방향으로 전송되는 장치.

#### 청구항 10

제 1 항에 있어서,

상기 웨이크-경고 신호는 상기 캐시 뱅크에 대응하는 영역 클록 버퍼의 게이팅에 영향을 주는 장치.

#### 청구항 11

제 1 항에 있어서,

상기 웨이크-경고 신호는 인에이블 클록의 디스에이블링이 연기되게 하는 장치.

#### 청구항 12

로직을 포함하되, 상기 로직은,

다른 시스템 구성요소로부터 시스템의 전용 웨이크-경고 채널을 통해 전송되는 웨이크-경고 신호를 수신하고 - 상기 웨이크-경고 신호는 상기 웨이크-경고 신호 다음에 인입하는 요청을 표시함 - ,

상기 웨이크-경고 신호에 적어도 부분적으로 기초하여, 상기 인입하는 요청의 수신 전에 클록을 인에이블하도록 클록을 언게이팅하고,

상호접속부를 통해 상기 인입하는 요청을 수신하고,

상기 인입하는 요청이 프로세싱되게 하는

장치.

### 청구항 13

제 12 항에 있어서,

상기 클록은, 상기 웨이크-경고 신호가 수신될 때 디스에이블되는

장치.

### 청구항 14

제 12 항에 있어서,

상기 로직은 또한 상기 클록에 대응하는 특정 시스템 구성요소의 비활동 조건을 검출하고, 상기 비활동 조건에 기초하여 상기 클록을 디스에이블하도록 상기 클록을 게이팅하는

장치.

### 청구항 15

제 14 항에 있어서,

상기 비활동 조건은 비활동 타이머에 기초하는

장치.

### 청구항 16

제 15 항에 있어서,

상기 클록이 인에이블될 때 수신되는 웨이크-경고 신호는 상기 비활동 타이머가 재시작되게 하는

장치.

### 청구항 17

제 12 항에 있어서,

상기 요청은 캐시 기록에 대한 것이고,

상기 요청은, 상기 캐시 기록이 상기 클록에 대응하는 특정 캐시 बैं크에 저장되는지를 결정하도록 프로세싱되는

장치.

### 청구항 18

제 12 항에 있어서,

상기 요청은 제 1 요청을 포함하고,

상기 로직은 또한,

상기 제 1 요청의 프로세싱에 기초하여 제 2 요청을 생성하고 - 상기 제 2 요청은 특정 시스템 구성요소로 전송됨 - ,

상기 제 2 요청이 도착한다는 것을 상기 특정 시스템 구성요소에 표시하기 위한 다른 웨이크-경고 신호를 상기 전용 웨이크-경고 채널을 통해 전송하고 - 상기 다른 웨이크-경고 신호는 상기 특정 시스템 구성요소에서의 디스에이블 클럭이 인에이블 상태로 언के이팅되게 함 - ,

상기 제 2 요청을 상기 특정 시스템 구성요소로 전송하는 장치.

### 청구항 19

제 18 항에 있어서,

상기 제 2 요청은 상기 제 1 요청의 프로세싱으로부터의 캐시 미스(cache miss)의 결정에 기초하는 장치.

### 청구항 20

제 12 항에 있어서,

상기 로직은 또한 복수의 시스템 구성요소 중 특정 시스템 구성요소를 어드레싱하는 웨이크-경고 신호에 대해 상기 웨이크-경고 채널을 모니터링하는

장치.

### 청구항 21

제 20 항에 있어서,

상기 웨이크-경고 신호는 상기 특정 시스템 구성요소를 식별하는 장치.

### 청구항 22

제 1 하드웨어 구성요소와,

상호접속부 및 전용 웨이크-경고 채널에 의해 상기 제 1 하드웨어 구성요소에 접속된 제 2 하드웨어 구성요소를 포함하되,

상기 제 2 하드웨어 구성요소는 상기 상호접속부 및 상기 전용 웨이크-경고 채널에 대한 인터페이스, 및 로직을 포함하되, 상기 로직은,

상기 전용 웨이크-경고 채널을 통해 전송된 웨이크-경고 신호를 상기 제 1 하드웨어 구성요소로부터 수신하고 - 상기 웨이크-경고 신호는 상기 웨이크-경고 신호 다음의 인입하는 요청을 표시함 - ,

상기 인입하는 요청의 수신 전에 상기 제 2 하드웨어 구성요소와 연관된 클럭을 인에이블하기 위해 상기 클럭을 언के이팅하고,

상기 상호접속부를 통해 상기 인입하는 요청을 수신하고,

상기 인입하는 요청을 프로세싱하는

시스템.

**청구항 23**

제 22 항에 있어서,

상기 제 1 하드웨어 구성요소는 로직을 포함하되, 상기 로직은,

특정 캐시 기록과 연관된 요청을 생성하고 - 상기 요청은 상호접속부를 통해 캐시 बैं크와 연관된 시스템 구성요소로 전송됨 - ,

상기 웨이크-경고 신호를 생성하고,

상기 전용 웨이크-경고 채널을 통해 상기 웨이크-경고 신호를 상기 제 2 하드웨어 구성요소로 전송하고,

상기 상호접속부를 통해 상기 요청을 상기 제 2 하드웨어 구성요소로 전송하는 시스템.

**청구항 24**

제 22 항에 있어서,

상기 제 2 하드웨어 구성요소는 또한,

특정 하드웨어 구성요소로 전송될 제 2 요청을 생성하고,

상기 제 2 요청이 도착한다는 것을 상기 특정 하드웨어 구성요소에 표시하기 위한 제 2 웨이크-경고 신호를 상기 전용 웨이크-경고 채널을 통해 전송하고 - 상기 제 2 웨이크-경고 신호는 상기 특정 하드웨어 구성요소의 디스에이블 클럭이 인에이블 상태로 언계이팅되게 함 - ,

상기 상호접속부를 통해 상기 제 2 요청을 상기 특정 하드웨어 구성요소로 전송하는 시스템.

**청구항 25**

제 22 항에 있어서,

상기 제 1 하드웨어 구성요소는, 제 1 프로세서 코어 및 제 1 캐시 बैं크를 포함하는 제 1 타일(tile)을 포함하고,

상기 제 2 하드웨어 구성요소는, 제 2 프로세서 코어 및 제 2 캐시 बैं크를 포함하는 제 2 타일을 포함하는 시스템.

**청구항 26**

삭제

**청구항 27**

삭제

**청구항 28**

삭제

**청구항 29**

삭제

청구항 30

삭제

청구항 31

삭제

청구항 32

삭제

청구항 33

삭제

청구항 34

삭제

**발명의 설명**

**기술 분야**

[0001] 본 개시물은 컴퓨팅 시스템에 관한 것이며, 특히(그러나 비배타적으로) 컴퓨팅 시스템에서의 클록 게이팅(clock gating)에 관한 것이다.

**배경 기술**

[0002] 프로세서 칩은 최근 수십년 동안에 상당히 진화하였다. 멀티-코어 칩의 출현은 개인용 컴퓨터 및 서버를 포함하는 컴퓨팅 디바이스 내에서 병렬의 컴퓨팅 및 다른 기능을 가능하게 하였다. 프로세서는 원래 단지 하나의 코어로 개발되었다. 각각의 코어는 실행 프로그램 명령을 관독할 수 있는 독립적인 CPU(central processing unit)일 수 있다. 개인용 컴퓨팅 디바이스에 대한 듀얼-, 쿼드- 및 심지어 헥사-코어 프로세서가 개발되었고, 한편 고성능 서버 칩은 일 십, 이십 개 및 그보다 더 많은 코어들로 개발되었다. 코어는 배선 도전체 또는 다른 송신 매체의 온-칩 상호접속부를 사용하여 다른 온-칩 구성요소와 함께 상호접속될 수 있다. 칩 상의 코어들의 수를 스케일링하는 것은 코어의 고속 상호 접속을 가능하게 하고자 하는 칩 설계자에게 도전과제일 수 있다. 다른 예 중에서도 링 버스 상호접속 아키텍처를 포함하여 다양한 상호접속 아키텍처가 개발되었다.

[0003] 고속 컴퓨팅 시스템에 대한 클록 설계는 계속해서 발전하고 있다. 현대의 클록은 8 GHz를 초과하는 속도로 실현되고 있다. 집적 회로(IC)는 회로 또는 칩의 상이한 구성요소를 동기화하기 위해 클록 신호를 사용할 수 있다. 다수의 구성요소를 포함하는 더 복잡한 시스템은 구성요소의 상호작용을 구동 및 동기화하기 위해 다수의 동기화된 클록 신호를 활용할 수 있다. 클록 신호는 또한 회로의 특정 부분에 대한 클록 신호를 선택적으로 디스에이블 및/또는 인에이블함으로써 게이팅/언게이팅될 수 있다. 클록 게이팅은, 예를 들면, 대응하는 구성요소가 사용중이 아닐 때, 클록에 의해 구동되는 디지털 회로의 부분을 효과적으로 셧 다운함으로써 전력을 절약하는데 사용될 수 있다.

**도면의 간단한 설명**

[0004] 도 1은 멀티코어 프로세서를 포함하는 컴퓨팅 시스템에 대한 블록도의 실시예를 도시한다.  
 도 2는 상호접속부 및 예시적인 웨이크-경고 채널을 포함하는 멀티-코어 시스템의 블록도를 도시한다.  
 도 3은 조기의 웨이크-경고 기능의 특징부를 인에이블하는 시스템 구성요소 로직의 블록도를 도시한다.  
 도 4는 적어도 일부 구현과 관련하여 요청 경로 및 조기의 경고 경로의 블록도를 도시한다.  
 도 5a 내지 도 5c는 시스템 구성요소의 요청에 앞서 시스템 구성요소의 클록을 게이팅/언게이팅하기 위한 웨이크-경고 신호를 수신하는 예시적인 흐름을 도시한다.



도 6a 및 도 6b는 조기의 웨이크-경고 기능을 지원하는 시스템의 예시적인 기술을 도시한 흐름도를 도시한다.

도 7은 컴퓨팅 시스템에 대한 블록도의 다른 실시예를 도시한다.

다양한 도면들에서 유사한 참조 부호들 및 지정사항들은 유사한 요소들을 표시한다.

**발명을 실시하기 위한 구체적인 내용**

[0005] 다음의 설명에서, 특정 타입의 프로세서들 및 시스템 구성들, 특정 하드웨어 구조들, 특정 아키텍처 및 마이크로 아키텍처 세부사항들, 특정 레지스터 구성들, 특정 명령 타입들, 특정 시스템 구성요소들, 특정 측정/높이, 특정 프로세서 파이프라인 스테이지들 및 동작들 등과 같은, 다수의 특정 세부사항들이 본 발명의 완전한 이해를 제공하기 위해 제시된다. 그러나, 이러한 특정 세부사항들은 본 발명의 논의 대상을 실시하는데 있어서 반드시 채용되어야만 하는 것은 아님이 본 기술 분야의 당업자에게 명백할 것이다. 다른 예들에서, 특정 및 선택적 프로세서 아키텍처들, 기술된 알고리즘을 위한 특정 로직 회로들/코드, 특정 펌웨어 코드, 저레벨 상호접속 동작, 특정 로직 구성들, 특정 제조 기법들 및 재료들, 특정 컴파일러 구현사항들, 코드로의 알고리즘들의 특정 표현, 특정 파워 다운(power down) 및 게이팅 기법들/로직 및 컴퓨터 시스템의 다른 특정 동작 상의 세부사항들과 같은, 알려진 구성요소들 또는 방법들은 본 발명을 불필요하게 모호하게 알지 않도록 하기 위해 상세하게 설명되지는 않는다.

[0006] 다음의 실시예들이 컴퓨팅 플랫폼들 또는 마이크로프로세서들에서와 같이, 특정 집적 회로들에서 에너지 보존, 에너지 효율, 프로세싱 효율 등을 참조하여서 기술될 수 있지만, 다른 실시예들이 다른 타입들의 집적 회로들 및 로직 디바이스들에 적용가능하다. 본 명세서에서 기술된 실시예들의 유사한 기법들 및 교시사항들은 또한 더 양호한 에너지 효율 및 에너지 보존으로부터 이점들을 취할 수 있는 다른 타입들의 회로들 또는 반도체 디바이스들에 적용될 수도 있다. 예를 들어, 개시된 실시예들은 데스크탑 컴퓨터 시스템들 또는 Ultrabooks™에 한정되지 않는다. 그리고 이것은 또한 핸드헬드 디바이스들, 태블릿들, 다른 얇은 노트북들, SOC(systems on chip) 디바이스들, 및 내장형 애플리케이션들과 같은 다른 디바이스들에서 사용될 수 있다. 핸드헬드 디바이스들의 몇몇 예들은 셀룰러 전화, 인터넷 프로토콜 디바이스들, 디지털 카메라들, PDA들(personal digital assistants), 핸드헬드 PC들을 포함한다. 내장형 애플리케이션들은 통상적으로 마이크로제어기, 디지털 신호 프로세서(DSP), 시스템 온 칩, 네트워크 컴퓨터들(NetPC), 셋탑 박스들, 네트워크 허브들, WAN(wide area network) 스위치들, 또는 이하에서 교시되는 기능들 및 동작들을 수행할 수 있는 임의의 다른 시스템을 포함한다. 또한, 본 명세서에서 기술된 장치들, 방법들, 및 시스템들은 물리적 컴퓨팅 디바이스들로 한정되지 않지만, 또한 에너지 보존 및 효율을 위한 소프트웨어 최적화와 관련될 수 있다. 이하의 설명에서 용이하게 자명할 바와 같이, (하드웨어, 펌웨어, 소프트웨어, 또는 이들의 조합 중 어느 것을 참조하든 상관없이) 본 명세서에서 기술된 장치들, 방법들, 및 시스템들의 실시예들은 성능 고려사항들과 균형을 이루는 "녹색 기술" 미래에 있어서 매우 중요하다고 사료될 수 있다.

[0007] 컴퓨팅 시스템들이 진보함에 따라서, 그 내부의 구성요소들은 보다 복잡해지고 있다. 결과적으로, 이러한 구성요소들 간을 연결 및 통신시키는 상호접속 아키텍처도 또한 최적의 구성요소 동작을 위해서 대역폭 요구수준이 만족되도록 보장하기 위해서 그 복잡성이 증가하였다. 또한, 상이한 시장 영역들도 각각의 시장에 맞도록 상이한 양태들의 상호접속 아키텍처들을 요구한다. 예를 들어, 서버들은 보다 높은 성능을 요구하는 한편, 이동 예코시스템은 때로 전력 절감을 위해서 전체적인 성능을 희생할 수도 있다. 그러나, 최대의 전력 절감으로 최고의 가능한 성능을 제공하는 것이 대부분의 구조들의 단일한 목적이다. 하기에, 다양한 상이한 상호접속부들은 잠재적으로 본 명세서에서 기술된 본 발명의 양태로부터 이점을 얻을 수 있다.

[0008] 도 1을 참조하면, 멀티코어 프로세서를 포함하는 컴퓨팅 시스템에 대한 블록도의 실시예가 도시되어 있다. 프로세서(100)는 마이크로프로세서, 내장형 프로세서, 디지털 신호 프로세서(DSP), 네트워크 프로세서, 핸드헬드 프로세서, 애플리케이션 프로세서, 코-프로세서, 시스템 온 칩(SOC), 또는 코드를 실행하기 위한 다른 디바이스들과 같은 임의의 프로세서 또는 프로세싱 디바이스를 포함한다. 일 실시예에서, 프로세서(100)는 비대칭 코어들 또는 대칭 코어들(예시된 실시예)을 포함할 수 있는 적어도 2개의 코어들(코어(101 및 102))을 포함한다. 그러나, 프로세서(100)는 대칭 또는 비대칭일 수 있는 임의의 수의 프로세싱 요소들을 포함할 수 있다.

[0009] 일 실시예에서, 프로세싱 요소는 소프트웨어 스레드를 지원하기 위한 하드웨어 또는 로직을 지칭한다. 하드웨어 프로세싱 요소들의 예들은: 스레드 유닛, 스레드 슬롯, 스레드, 프로세스 유닛, 컨텍스트, 컨텍스트 유닛, 논리 프로세서, 하드웨어 스레드, 코어, 및/또는 임의의 다른 요소를 포함하고, 이것은 실행 상태 또는 구조적 상태와 같은 프로세서에 대한 상태를 홀딩할 수 있다. 다시 말해, 일 실시예에서, 프로세싱 요소는 소프트웨어 스레

드, 운영 시스템, 애플리케이션, 또는 다른 코드와 같은 코드와 독립적으로 연관될 수 있는 임의의 하드웨어를 지칭한다. 물리적 프로세서(또는 프로세서 소켓)는 코어들 또는 하드웨어 스레드들과 같은 임의의 수의 다른 프로세싱 요소들을 잠재적으로 포함하는 집적 회로를 통상적으로 지칭한다.

[0010] 코어는 독립적인 구조적 상태를 유지할 수 있는 집적 회로상에 위치한 로직을 종종 지칭하고, 여기서, 각 독립적으로 유지된 구조적 상태는 적어도 일부 전용 실행 자원들과 연관된다. 코어들과 반대로, 하드웨어 스레드는 독립적인 구조적 상태를 유지할 수 있는 집적 회로상에 위치한 임의의 로직을 통상적으로 지칭하고, 여기서, 독립적으로 유지된 구조적 상태들은 실행 자원에 대한 액세스를 공유한다. 알 수 있는 바와 같이, 특정한 자원들이 공유되고 다른 자원들이 구조적 상태에 전용될 때, 하드웨어 스레드와 코어의 명칭 사이의 라인이 오버랩한다. 하지만 종종, 코어와 하드웨어 스레드는 운영 시스템에 의해 개별 논리 프로세서들로서 보여지고, 여기서, 운영 시스템은 각 논리 프로세서에 대한 동작들을 개별적으로 스케줄링할 수 있다.

[0011] 도 1에 예시된 바와 같이, 물리적 프로세서(100)는 2개의 코어들(코어(101 및 102))을 포함한다. 여기서, 코어(101 및 102)는 대칭 코어들, 즉, 동일한 구성들, 기능적 유닛들, 및/또는 로직을 갖는 코어들로 고려된다. 다른 실시예에서, 코어(101)는 비순차적(out-of-order) 프로세서 코어를 포함하지만, 코어(102)는 순차적(in-order) 프로세서 코어를 포함한다. 그러나, 코어들(101 및 102)은 네이티브 코어, 소프트웨어 관리된 코어, 네이티브 명령 세트 아키텍처(ISA)를 실행하도록 적응된 코어, 변환된(translated) 명령 세트 아키텍처(ISA)를 실행하도록 적응된 코어, 병행 설계된 코어, 또는 다른 공지된 코어와 같은 임의의 타입의 코어로부터 개별적으로 선택될 수 있다. 이종의 코어 환경(즉, 비대칭 코어들)에서, 이진 변환과 같은 변환의 일부 형태가 코어들 중 하나 또는 양자에 대한 코드를 스케줄링하거나 실행하도록 이용될 수 있다. 또 다른 논의로, 코어(101)에 예시된 기능적 유닛들이, 코어(102)에서의 유닛들이 도시된 실시예에서 유사한 방식으로 동작하기 때문에, 더 상세히 후술된다.

[0012] 도시된 바와 같이, 코어(101)는 하드웨어 스레드 슬롯들(101a 및 101b)로서 또한 지칭될 수 있는 2개의 하드웨어 스레드들(101a 및 101b)을 포함한다. 따라서, 일 실시예에서, 운영 시스템과 같은 소프트웨어 엔티티들은 프로세서(100)를 4개의 개별 프로세서들, 즉, 4개의 소프트웨어 스레드들을 동시에 실행할 수 있는 4개의 논리 프로세서들 또는 프로세싱 요소들로서 잠재적으로 본다. 위에서 언급한 바와 같이, 제 1 스레드는 아키텍처 상태 레지스터들(101a)과 연관되고, 제 2 스레드는 아키텍처 상태 레지스터들(101b)과 연관되고, 제 3 스레드는 아키텍처 상태 레지스터들(102a)과 연관될 수 있으며, 제 4 스레드는 아키텍처 상태 레지스터들(102b)과 연관될 수 있다. 여기서, 아키텍처 상태 레지스터들(101a, 101b, 102a, 및 102b) 각각은 상술한 바와 같이, 프로세싱 요소들, 스레드 슬롯들, 또는 스레드 유닛들로서 지칭될 수 있다. 예시된 바와 같이, 아키텍처 상태 레지스터들(101a)은 아키텍처 상태 레지스터들(101b)에서 복제되어서, 개별 아키텍처 상태들/컨텍스트들이 논리 프로세서(101a) 및 논리 프로세서(101b)용으로 저장될 수 있다. 코어(101, 102)에서, 할당기 및 리네이머(renamer) 블록(130, 131)에서의 명령 포인터들 및 리네이밍(renaming) 로직과 같은 다른 더 작은 자원들이 스레드들(101a 및 101b 및 102a 및 102)에 대해 각각 또한 복제될 수 있다. 재정렬/리타이어먼트 유닛(135, 136)에서의 재정렬 버퍼들, ILTB(120, 121), 로딩/저장 버퍼들, 및 큐우들과 같은 일부 자원들이 파티셔닝을 통해 공유될 수 있다. 범용 내부 레지스터들, 페이지-테이블 베이스 레지스터(들), 저레벨 데이터-캐시 및 데이터-TLB(150, 151), 실행 유닛(들)(140, 141), 및 비순차적 유닛의 부분들과 같은 다른 자원들이 잠재적으로 전체 공유된다.

[0013] 프로세서(100)는 전체 공유되고, 파티셔닝을 통해 공유되거나, 프로세싱 요소들에/프로세싱 요소들에 의해 전용될 수 있는 다른 자원들을 종종 포함한다. 도 1에서, 프로세서의 예시적인 논리 유닛들/자원들을 갖는 순수하게 예시적인 프로세서의 실시예가 예시되어 있다. 프로세서가 임의의 이들 기능적 유닛들을 포함하거나 생략할 뿐만 아니라 도시되지 않은 임의의 다른 공지된 기능적 유닛들, 로직, 또는 펌웨어를 포함할 수 있다는 것에 유의한다. 예시된 바와 같이, 코어(101)는 단순화된 대표적인 비순차적(000) 프로세서 코어를 포함한다. 그러나, 순차적 프로세서가 상이한 실시예들에서 이용될 수 있다. 000 코어는 실행되고/취해질 브랜치들을 예측하기 위한 브랜치 타겟 버퍼(120) 및 명령들에 대한 어드레스 변환 엔트리들을 저장하기 위한 명령-변환 버퍼(I-TLB)(120)를 포함한다.

[0014] 코어(101)는 페치된 요소들을 디코딩하기 위해 페치 유닛에 연결된 디코딩 모듈(125)을 더 포함한다. 일 실시예에서, 페치 로직은 스레드 슬롯들(101a, 101b) 각각과 연관된 개별 시퀀서들을 포함한다. 일반적으로, 코어(101)는 프로세서(100)상에서 실행가능한 명령들을 정의/특정하는 제 1 ISA와 연관된다. 종종, 제 1 ISA의 일부인 머신 코드 명령들은 수행될 명령 또는 동작을 참조/특정하는 (연산코드(opcode)로서 지칭되는) 명령의 일부를 포함한다. 디코딩 로직(125)은 그들의 연산코드들로부터의 이들 명령들을 인식하고 제 1 ISA에 의해 정의된 바와 같이 프로세싱을 위해 파이프라인에서 디코딩된 명령들을 패스하는 회로를 포함한다. 예를 들어, 아래에서

더 상세히 논의하는 바와 같이, 일 실시예에서, 디코더(125)는 트랜잭션 명령과 같은 특정한 명령들을 인식하도록 설계되거나 적용된 로직을 포함한다. 디코더들(125)에 의한 인식의 결과로서, 아키텍처 또는 코어(101)는 적절한 명령과 연관된 작업들을 수행하기 위해 특정하고, 사전정의된 액션들을 취한다. 여기에 설명하는 임의의 작업들, 블록들, 동작들, 및 방법들이 단일 또는 다중의 명령들에 응답하여 수행될 수 있고, 이 명령들 중 일부는 새롭거나 오래된 명령들일 수 있다는 것에 유의하는 것이 중요하다. 일 실시예에서, 디코더들(126)은 동일한 ISA(또는 그것의 서브세트)를 인식한다는 것에 유의한다. 대안으로서, 이종의 코어 환경에서, 디코더들(126)은 제 2 ISA(제 1 ISA의 서브세트 또는 별개의 ISA 중 어느 하나)를 인식한다.

[0015] 일 실시예에서, 할당기 및 리네이머 블록(130)은 명령 프로세싱 결과들을 저장하기 위한 레지스터 파일들과 같은 자원들을 예약하기 위한 할당기를 포함한다. 그러나, 스레드들(101a 및 101b)은 잠재적으로 비순차적 실행이 가능할 수 있고, 여기서, 할당기 및 리네이머 블록(130)은 명령 결과들을 트래킹하기 위해 재정렬 버퍼들과 같은 다른 자원들을 또한 예약한다. 유닛(130)은 프로그램/명령 레퍼런스 레지스터들을 프로세서(100) 내부의 다른 레지스터들로 리네이밍하기 위한 레지스터 리네이머를 또한 포함할 수 있다. 재정렬/리타이어먼트 유닛(135)은 비순차적 실행 및 비순차적으로 실행된 명령들의 나중의 순차적 리타이어먼트를 지원하기 위해 상기 언급한 재정렬 버퍼들, 로드 버퍼 및 저장 버퍼들과 같은 구성요소들을 포함한다.

[0016] 일 실시예에서, 스케줄러 및 실행 유닛(들) 블록(140)은 실행 유닛들에 대한 명령들/동작을 스케줄링하기 위한 스케줄러 유닛을 포함한다. 예를 들어, 부동 소수점 명령(floating point instruction)이 가용 부동 소수점 실행 유닛을 갖는 실행 유닛의 포트에 대해 스케줄링된다. 실행 유닛들과 연관된 레지스터 파일들이 정보 명령 프로세싱 결과들을 저장하기 위해 또한 포함된다. 예시적인 실행 유닛들은 부동 소수점 실행 유닛, 정수 실행 유닛, 점프 실행 유닛, 부하 실행 유닛, 저장 실행 유닛, 및 다른 공지된 실행 유닛들을 포함한다.

[0017] 하위 레벨 데이터 캐시 및 데이터 변환 버퍼(D-TLB)(150)가 실행 유닛(들)(140)에 연결된다. 데이터 캐시는 메모리 코히어런스 상태들에서 잠재적으로 홀딩된 데이터 피연산 함수들과 같은 최근에 사용되고/동작된 요소들을 저장하기 위한 것이다. D-TLB는 최근의 가상/선형-물리적 어드레스 변환들을 저장하기 위한 것이다. 특정한 예로서, 프로세서는 물리적 메모리를 복수의 가상 페이지들로 분할하기 위한 페이지 테이블 구조를 포함할 수 있다.

[0018] 여기서, 코어들(101 및 102)은 온-칩 인터페이스(110)와 연관된 제 2 레벨 캐시와 같은 상위 레벨 또는 더 먼(further-out) 레벨 캐시에 대한 액세스를 공유한다. 상위 레벨 또는 더 먼 레벨은 실행 유닛(들)으로부터 증가하는 또는 더 멀어지는 캐시 레벨들을 지칭한다에 유의한다. 일 실시예에서, 상위 레벨 캐시는 제 2 또는 제 3 레벨 데이터 캐시와 같은 프로세서(100)에 대한 메모리 계층에서 최종 레벨 데이터 캐시-최종 캐시이다. 그러나, 상위 레벨 캐시는 명령 캐시와 연관될 수 있거나 명령 캐시를 포함할 수 있기 때문에, 이에 제한되지 않는다. 대신에, 트레이스 캐시 - 명령 캐시의 타입 - 가 최근에 디코딩된 트레이스들을 저장하기 위해 디코더(125) 이후에 연결될 수 있다. 여기서, 명령은 다수의 마이크로명령들(마이크로-동작들)로 디코딩할 수 있는 매크로-명령(즉, 디코더들에 의해 인식된 일반 명령)을 잠재적으로 지칭한다.

[0019] 도시된 구성에서, 프로세서(100)는 온-칩 인터페이스 모듈(110)을 또한 포함한다. 이력적으로, 더 상세히 후술하는 메모리 제어가 프로세서(100) 외부의 컴퓨팅 시스템에 포함된다. 이러한 시나리오에서, 온-칩 인터페이스(110)는 시스템 메모리(175), 칩셋(메모리(175)에 접속시키기 위한 메모리 제어기 허브 및 주변 디바이스들에 접속시키기 위한 I/O 제어기 허브를 종종 포함함), 메모리 제어기 허브, 노스브리지(northbridge), 또는 다른 집적 회로와 같은 프로세서(100) 외부의 디바이스들과 통신하기 위한 것이다. 그리고, 이러한 시나리오에서, 버스(105)가 멀티-드롭(multi-drop) 버스, 점 대 점 상호접속부, 직렬 상호접속부, 병렬 버스, 코히어런트(예를 들어, 캐시 코히어런트) 버스, 계층화된 프로토콜 아키텍처, 차동 버스, 및 GTL 버스와 같은 임의의 공지된 상호접속부를 포함할 수 있다.

[0020] 메모리(175)는 프로세서(100)에 전용될 수 있거나 시스템에서의 다른 디바이스들과 공유될 수 있다. 메모리(175)의 타입들의 공통적인 예들이 DRAM, SRAM, 비휘발성 메모리(NV 메모리), 및 다른 공지된 저장 디바이스들을 포함한다. 디바이스(180)가 메모리 제어기 허브에 연결된 그래픽 가속기, 프로세서 또는 카드, I/O 제어기 허브에 연결된 데이터 스토리지, 무선 트랜시버, 플래시 디바이스, 오디오 제어기, 네트워크 제어기, 또는 다른 공지된 디바이스를 포함할 수 있다는 것에 유의한다.

[0021] 그러나, 최근에, 더 많은 로직 및 디바이스들이 SOC와 같은 단일 다이상에 집적됨에 따라, 이들 디바이스들 각각은 프로세서(100)에 통합될 수 있다. 예를 들어, 일 실시예에서, 메모리 제어기 허브가 프로세서(100)와 동일한 패키지 및/또는 다이상에 있다. 여기서, 코어의 부분(온-코어 부분)(110)이 메모리(175) 또는 그래픽 디바이스



스(180)와 같은 다른 디바이스들과 인터페이스하기 위해 하나 이상의 제어기(들)를 포함한다. 상호접속부 및 이러한 디바이스들과 인터페이스하는 제어기들을 포함하는 구성을 종종 온-코어(또는 언-코어 구성)로서 지칭한다. 일례로서, 온-칩 인터페이스(110)는 온-칩 통신을 위한 링 상호접속부 및 오프-칩 통신을 위한 고속 직렬 점 대 점 링크(105)를 포함한다. 그러나, SOC 환경에서, 네트워크 인터페이스들, 코-프로세서들, 메모리(175), 그래픽 프로세서(180), 및 임의의 다른 공지된 컴퓨터 디바이스들/인터페이스와 같은 더욱더 많은 디바이스들이 높은 기능성 및 낮은 전력 소모를 갖는 작은 폼 팩터를 제공하기 위해 단일 다이 또는 집적 회로상에 집적될 수 있다.

[0022] 일 실시예에서, 프로세서(100)는 여기에 설명한 장치 및 방법들을 지원하거나 이들과 인터페이스하기 위해 애플리케이션 코드(176)를 컴파일하고, 변환하고, 그리고/또는 최적화하도록 컴파일러, 최적화, 및/또는 변환기 코드(177)를 실행할 수 있다. 컴파일러는 소스 텍스트/코드를 타겟 텍스트/코드로 변환하기 위해 프로그램 또는 프로그램들의 세트를 종종 포함한다. 일반적으로, 컴파일러를 이용한 프로그램/애플리케이션 코드의 컴필레이션(compilation)은 하이 레벨 프로그래밍 언어 코드를 로우 레벨 기계 또는 어셈블리 언어 코드로 변환하기 위해 다중의 단계들 및 경로들에서 행해진다. 그러나, 단일 패스 컴파일러들이 단순한 컴필레이션을 위해 여전히 이용될 수 있다. 컴파일러는 임의의 공지된 컴필레이션 기법들을 이용할 수 있고, 어휘 분석, 사전 프로세싱, 파싱, 의미 분석, 코드 생성, 코드 변환, 및 코드 최적화와 같은 임의의 공지된 컴파일러 동작들을 수행할 수 있다.

[0023] 더 큰 컴파일러들은 다중의 단계들을 종종 포함하지만, 이들 단계들은 대부분 종종 2개의 일반적인 단계들: (1) 프론트-엔드, 즉, 일반적으로, 구문 프로세싱, 의미 프로세싱, 및 일부 변환/최적화가 발생할 수 있는 경우, 및 (2) 백-엔드, 즉, 일반적으로, 분석, 변환들, 최적화들, 및 코드 생성이 발생하는 경우에 포함된다. 일부 컴파일러들은 컴파일러의 프론트-엔드와 백-엔드 사이의 서술(delineation)의 블러링(blurring)을 예시하는 중간을 지칭한다. 그 결과, 삽입, 연관, 생성, 또는 컴파일러의 다른 동작에 대한 레퍼런스는 임의의 상기 언급한 단계들 또는 경로들뿐만 아니라 컴파일러의 임의의 다른 공지된 단계들 또는 경로들에서 발생할 수 있다. 예시적인 예로서, 컴파일러는 컴필레이션의 프론트-엔드 단계에서의 호들/동작들의 삽입과 같은 동작들, 호들, 함수들 등을 컴필레이션의 하나 이상의 단계들에 잠재적으로 삽입한 후, 호들/동작들을 변환 단계 동안 하위 레벨 코드로 변환한다. 동적 컴필레이션 동안, 컴파일러 코드 또는 동적 최적화 코드는 이러한 동작들/호들을 삽입할 수 있을 뿐만 아니라 런타임 동안 실행을 위해 코드를 최적화할 수 있다는 것에 유의한다. 특정한 예시적인 예로서, 이진 코드(이미 컴파일링된 코드)가 런타임 동안 동적으로 최적화될 수 있다. 여기서, 프로그램 코드는 동적 최적화 코드, 이진 코드, 또는 이들 조합을 포함할 수 있다.

[0024] 컴파일러와 유사하게, 이진 변환기와 같은 변환기는 코드를 최적화하고 그리고/또는 변환하기 위해 코드를 정적으로 또는 동적으로 변환한다. 따라서, 코드, 애플리케이션 코드, 프로그램 코드, 또는 다른 소프트웨어 환경의 실행에 대한 레퍼런스는: (1) 프로그램 코드를 컴파일링하고, 소프트웨어 구조들을 유지하고, 다른 동작들을 수행하고, 코드를 최적화하거나, 코드를 변환하기 위해 컴파일러 프로그램(들), 최적화 코드 최적화기, 또는 변환기를 동적으로 또는 정적으로 실행하는 것; (2) 최적화되고/컴파일링된 애플리케이션 코드와 같은 동작들/호들을 포함하는 메인 프로그램 코드를 실행하는 것; (3) 소프트웨어 구조들을 유지하고, 다른 소프트웨어 관련 동작들을 수행하거나, 코드를 최적화하기 위해 메인 프로그램 코드와 연관된 라이브러리들과 같은 다른 프로그램 코드를 실행하는 것; 또는 (4) 이들의 조합을 지칭할 수 있다.

[0025] 예시적인 상호접속 패브릭 및 프로토콜은 다른 것들 중에서 PCIe(PCI(Peripheral Component Interconnect) Express) 아키텍처, 인텔 QPI(QuickPath Interconnect) 아키텍처, MIPI(Mobile Industry Processor Interface)와 같은 예를 포함할 수 있다. 다양한 지원되는 프로세서는 다수의 도메인 또는 노드 제어기 사이의 다른 상호접속부의 사용을 통해 도달될 수 있다.

[0026] 상호접속 패브릭 아키텍처는 계층 프로토콜 아키텍처의 정의를 포함할 수 있다. 일 실시예에서, 프로토콜 계층(코히어런트, 비-코히어런트, 및 선택적으로 다른 메모리 기반 프로토콜), 라우팅 계층, 링크 계층 및 물리 계층이 제공될 수 있다. 게다가, 상호접속부는 전력 관리기, DFT(design for test and debug), 결함 처리, 레지스터, 보안 등에 관련된 개선책을 포함할 수 있다.

[0027] 일 실시예에서, 상호접속 패브릭의 물리 계층은 물리적 매체(전기 또는 광학 등) 상에서 정보의 고속 전송을 담당할 수 있다. 물리적 링크는 2 개의 링크 계층 엔티티 사이의 점 대 점이다. 링크 계층은 상위 계층으로부터 물리 계층을 추상화하고, 신뢰할 수 있게 데이터(뿐만 아니라 요청)를 전송하고 2 개의 직접적으로 접속된 엔티티 사이의 흐름 제어를 관리하기 위한 능력을 제공할 수 있다. 이것은 또한 물리 채널을 다수의 가상 채널 및

메시지 클래스로 가상화하는 것을 담당한다. 프로토콜 계층은, 물리적 링크를 통한 전송을 위해 메시지를 물리 계층에 넘겨주기 전에 프로토콜 메시지를 적절한 메시지 클래스 및 가상 채널로 맵핑하기 위해 링크 계층을 필요로 할 수 있다. 링크 계층은 요청, 스누프(snoop), 응답, 답장(writeback), 비-코히어런트 데이터 등과 같은 다수의 메시지를 지원할 수 있다.

[0028] 일부 구현에서, 링크 계층은 흐름 제어를 위한 크레딧 방식을 활용할 수 있다. 비-크레딧 흐름이 또한 지원될 수 있다. 크레딧 흐름에 관련하여, 초기화 동안에, 패킷 또는 플릿(flit)을 수신기로 전송하기 위해 다수의 크레딧의 세트가 전송자에게 제공된다. 패킷 또는 플릿이 수신기로 전송될 때마다, 전송자는 사용되는 가상 네트워크의 타입에 의존하여 패킷 또는 플릿 중 어느 하나를 나타내는 하나의 크레딧만큼 자신의 크레딧 카운터를 감분한다. 버퍼가 수신기에서 해방될 때마다, 그 버퍼 타입에 대해 크레딧은 다시 전송자로 반환된다. 일 실시예에서, 주어진 채널에 대한 전송자의 크레딧이 소멸될 때, 전송자는 그 채널에서 임의의 플릿을 전송하는 것을 중지한다. 근본적으로, 수신기가 정보를 소비하고 적절한 버퍼를 해방한 후에, 크레딧이 반환된다.

[0029] 일 실시예에서, 라우팅 계층은 스스로부터 목적지로 패킷을 라우팅하기 위한 유연하고 분산형 방식을 제공할 수 있다. 일부 플랫폼 타입(예를 들면, 유니프로세서 및 듀얼 프로세서 시스템)에서, 이러한 계층은 명시적이지 않을 수 있지만, 링크 계층의 부분일 수 있고, 그러한 경우에, 이러한 계층은 선택적이다. 이것은 패킷을 라우팅하는 방법을 결정하기 위한 기능의 부분으로서 링크 계층에 의해 제공되는 가상 네트워크 및 메시지 클래스 추상화를 필요로 한다. 일 구현에서, 라우팅 기능은 특정 라우팅 테이블의 구현을 통해 정의된다. 그러한 정의는 다양한 사용 모델을 허용한다.

[0030] 일 구현에서, 프로토콜 계층은 통신 프로토콜, 정렬 규칙, 코히어런스 유지, I/O, 인터럽트 및 다른 고레벨 통신을 구현할 수 있다. 일 구현에서, 프로토콜 계층이 구성요소 및 시스템에 대한 전력 상태를 협상하기 위한 메시지를 제공할 수 있다는 것을 유의하라. 잠재적인 부가로서, 물리 계층은 또한 별개의 링크의 전력 상태를 독립적으로 또는 이에 관련하여 설정할 수 있다.

[0031] 홈 에이전트(메모리에 대한 요청을 명령), 캐싱(코히어런트 메모리에 대한 요청을 발행하고 스누프에 응답), 구성(구성 트랜잭션을 처리), 인터럽트(인터럽트를 프로세싱), 레거시(레거시 트랜잭션을 처리), 비-코히어런트(비-코히어런트 트랜잭션을 처리) 및 다른 것과 같은 다수의 에이전트가 상호접속 아키텍처에 접속될 수 있다.

[0032] 일부 구현에서, 클록 게이팅 기능은 시스템의 전력 소비를 관리하는 것을 보조하기 위해 시스템 상에 존재하는 다른 기능과 관련하여 제공될 수 있다. 클록 게이팅/언게이팅(본 명세서에서 때때로 총괄적으로 "게이팅"으로 지칭됨)은, 클록에 의해 구동되는 구성요소에 의해 수행될 작업이 존재할 때 클록이 인에이블되고 어떠한 작업도 없을 때 디스에이블되도록, 시스템 내의 특정 구성요소, 유닛, 서브-구성요소 및 서브-유닛 등의 클록이 선택적으로 활성화 또는 인에이블 및 디스에이블되도록 허용할 수 있다. 일부 시스템에서, 클록 전력은 총 칩 전력의 상당한 부분을 소비할 수 있고, 클록 속도가 증가함에 따라, 시스템 클록에 의해 잠재적으로 활용되는 전력의 공유는 상승할 것으로 예상될 수 있다. 실제로, 일부 경우에서, 클록 전력은 총 칩 전력의 다수 또는 그 이상을 소비할 수 있다. 시스템 클로킹의 효과적인 전력 관리를 통해 상당한 전력 절약 기회가 실현될 수 있다.

[0033] 일부 경우에서, 시스템 클록은, 하나 이상의 루트 클록 신호가 더 로컬적인 클록 싱크, "영역", 또는 다른 서브-네트워크의 하나 이상의 레벨 또는 계층으로 분산되는 트리, 스파인(spine) 또는 메시 네트워크를 포함하는 분산형 클록일 수 있다. 게이팅은 이론상 시스템의 분산 클록 네트워크의 임의의 레벨에서 수행될 수 있다. 예를 들면, 일부 구현에서, 고레벨 클록 스파인은 더 로컬적인, 또는 하위 레벨 클록 영역을 "피딩(feed)"할 수 있다. 클록 영역은 다른 구현들 중에서도 심지어 하위 레벨 로컬 클록에 대한 패런트(parent) 신호로서 기능할 수 있다. 상위 레벨 또는 패런트 클록 신호를 게이팅함으로써, 하위 레벨 칠드런(children) 클록 신호가 또한 게이팅될 수 있다. 예를 들면, RCB(regional clock buffer) 레벨에서의 게이팅은, 다른 예 중에서도, 동일한 게이팅이 LCB(local clock buffer) 레벨에서 영역의 하위 레벨 로컬 클록에 적용하게 할 수 있다.

[0034] 일부 구현에서, 클록 게이팅 레지스터는 대응하는 클록을 선택 가능하게 인에이블, 또는 게이팅하는데 사용하기 위한 인에이블 조건을 수용하는 로직을 포함할 수 있다. 일부 예에서, 게이팅된 클록에 의해 구동되는 시스템 구성요소에 의해 수행될 일부 작업에 참여 시에 디스에이블된 클록이 인에이블되게 하는 메시지가 클록 게이팅 로직으로 전송될 수 있다. 클록이 이미 인에이블되면, 메시지 또는 신호가 더 이상 유효하지 않을 수 있다. 클록이 디스에이블되는 경우에, 신호가 게이팅 로직의 인에이블 조건에 따라 수신되면, 클록은 인에이블되도록 게이팅되고, 클록 신호를 소비하는 구성요소(들)의 기능을 인에이블할 수 있다.

[0035] 일부 현대 칩 및 시스템에서 클록 게이팅이 채용되었지만, 그러한 해결책은 통상적으로 충분히 활용되지 않았다. 전통적인 클록 게이팅은 부가적인 시스템 레이턴시, 위험-드롭 패킷, 및 다른 예시적인 이슈를 도입할 수 있다. 예를 들면, 클록 게이팅에 의해 실현되는 전력 절약은 클록을 디스에이블 상태에 배치함으로써 달성된다. 그러나, 게이팅된 클록에 의존하는 구성요소로 예정된 요청 또는 데이터가 클록의 언게이팅(인에이블링)의 완료 전에 목적지 구성요소에 도달하면, 패킷은 큐잉 또는 드롭될 수 있다. 추가로, 일부 전통적인 시스템은 요청을 목적지 구성요소로 전송하기 전에 목적지 구성요소의 클록의 언게이팅을 보장하려고 시도할 수 있다. 그러나, 그러한 접근법은 다른 이슈 중에서도 목적지 구성요소의 클록의 재-인에이블링을 펜딩하는 요청을 효과적으로 큐잉함으로써 부가적인 레이턴시를 도입할 수 있다. 많은 현대 디바이스가 일부 형태의 클록 게이팅 기능을 포함하지만, 많은 경우에서, 현재 클록 게이팅 해결책의 성능 페널티 및 단점이 주어지면, 시스템은 이러한 기능을 대체로 무시 또는 디스에이블하도록 설계된다.

[0036] 일부 구현에서, 요청이 목적지 구성요소에 도달할 때까지, 목적지 구성요소(및 더 나아가, 목적지 구성요소 자체)의 클록이 인에이블되는 것을 보장하기 위해 동일한 목적지 구성요소에 전송되는 요청과 관련하여 목적지 구성요소의 클록을 웨이크 또는 언게이팅하기 위해 목적지 구성요소에서의 클록 게이팅 로직에 의해 사용하기 위한 조기의 웨이크-경고 신호를 목적지 구성요소로 전송하기 위한 전용 채널을 포함하는 클록 게이팅 해결책이 제공될 수 있다. 요청은 상호접속부를 통해 목적지 구성요소로 전송될 조기의 웨이크-경고 신호와 동시에 별개의 경로 상에서 효과적으로 전송될 수 있고, 한편 웨이크-경고 신호는 웨이크-경고 통신 채널(또는 링크)의 전용 배선을 통해 전송되고, 이것은 록-업 또는 중재 특징부가 더 적거나 아예 없는 간략화된, 낮은-레이턴시 전송을 허용할 수 있어서, 더 강인한 프로토콜, 완전한 메시지 중재, 경로 록 업, 및 다른 특징 및 정책을 포함하여, 목적지로 타겟팅된 대응하는 요청이 상호접속 링크를 통해 전송되기 하나 이상의 사이클 전에 웨이크-경고 신호가 목적지에 도달하는 것을 가능하게 한다. 상호접속부는 다른 예 중에서 PCIe-준수, QPI-준수, MIPI-준수 또는 다른 상호접속 아키텍처를 포함할 수 있다. 그러한 해결책은 사실상 시스템 내의 구성요소 사이에서 전송되는 모든 요청과 관련하여 웨이크-경고 신호가 전송되도록 허용할 수 있어서, 각각의 구성요소에서의 클록 게이팅의 미세-튜닝 및 적극적인 제어가 시스템 상에 존재하는 클록 게이팅 기회의 더 강인한 활용을 실현하도록 허용한다. 명시적으로 열거되지 않은 다른 예시적인 이점 및 장점 중에서도, 다른 전통적인 해결책을 초과하여 대응하는 전력 절약이 실현될 수 있다.

[0037] 앞서 설명된 것과 같은 원리를 통합하고, 아래의 예에 도시 및 설명되는 것과 유사한 시스템이 제공될 수 있다. 예를 들면, 도 2의 간략한 블록도에 예시된 예로 넘어가면, 복수의 타일을 포함하는 시스템(200)이 도시되고, 각각의 타일은 복수의 프로세서 코어(예를 들면, 202, 204, 206, 208, 210, 212, 214, 216, 218, 220) 및 (예를 들면, LLC(last level cache)의) 대응하는 캐시 뱅크(예를 들면, 222, 224, 226, 228, 230, 232, 234, 236, 238, 240) 중 하나를 포함한다. 일부 경우에서, 시스템(200)의 구성요소는 마이크로서버 또는 다른 서버 시스템에서 구현될 수 있다. 각각의 타일은 캐시 뱅크(예를 들면, 222, 224, 226, 228, 230, 232, 234, 236, 238, 240) 중 하나 이상과 코어들(예를 들면, 202, 204, 206, 208, 210, 212, 214, 216, 218, 220)의 상호작용을 관리하기 위한 로직을 더 포함할 수 있다. 예를 들면, 각각의 타일은 시스템 캐시(예를 들면, LLC)를 통해 코어와 인터페이스하고 캐시 기록을 수반하여 코어(또는 다른 구성요소)로부터의 요청과 관련하여 사용하기 위한 기능을 제공하기 위해 캐시 박스 로직, 코어 박스 로직, 또는 결합된 코어-캐시 박스 로직(본 명세서에서 총괄적으로 "Cbo" 로직으로 지칭됨)(242, 244, 246, 248, 250, 252, 254, 256, 258, 260)을 포함할 수 있다. 그러한 기능은, 다른 예시적인 기능 중에서도, 요청된 캐시 기록에 대응하는 캐시 뱅크를 결정하고, 요청을 (예를 들면, 대응하는 캐시 박스 상의) 캐시 에이전트로 전송하고, 요청을 전송하기 위한 경로를 결정하고, 캐시 코히어런시를 수행하기 위한 로직을 포함할 수 있다. 각각의 타일 또는 유닛은 상호접속부(265)를 통해 유닛(및 구성요소 코어, 캐시 뱅크 등)을 인터페이스하는 각각의 상호접속 인터페이스(예를 들면, 264, 266)를 더 포함할 수 있다. Cbo 로직은 대응하는 상호접속 인터페이스(예를 들면, 264, 266)와 인터페이스하거나, 일부 경우에서, 이를 통합할 수 있다. 부가적으로, 도 2의 특정 예에서, 전용 웨이크-경고 링크 또는 채널(270)이 제공될 수 있고, 각각의 유닛은 (예를 들면, 인터페이스(예를 들면, 264, 266)를 통해) 상호접속부(265) 및 웨이크-경고 채널(270) 양자에 접속될 수 있다.

[0038] 도 2의 예에서, 링 상호접속부(265)가 도시되고, 대응하는 웨이크-경고 채널(270)이 링-스타일 방향으로 제공된다. 메시, 멀티-링, 하이브리드 링-메시, 및 다른 상호접속부를 포함하여 다른 상호접속 아키텍처가 제공될 수 있다는 것이 인지되어야 한다. 상호접속부의 일반적인 물리적 레이아웃 또는 방향을 미러링(mirror)하는 대응하는 웨이크-경고 채널이 제공될 수 있다. 예를 들면, 링-스타일 웨이크 경고 채널이 링-스타일 상호접속부와 관련하여 제공될 수 있고, 메시-스타일 웨이크 경고 채널이 메시-스타일 상호접속부와 관련하여 제공될 수 있고,



기타 등등이다. 다른 경우에서, 웨이크-경고 채널은 상호접속부의 것과 독립적인 또는 상이한 설계 또는 레이아웃을 채택할 수 있다. 웨이크-경고 채널은 간략한 통신 채널일 수 있고, 상호접속부(예를 들면, 265)의 레이아웃을 구동시키는 것의 고려사항은 웨이크-경고 채널에 적용되지 않을 수 있다. 일부 구현에서, 웨이크-경고 채널의 레이아웃은, 다른 예시적인 예 및 고려사항 중에서도, 대응하는 실질적인 요청 또는 메시지를 전송하기 위해 상호접속부에 의해 사용되는 것보다 더 적은 사이클로 웨이크-경고 신호를 목적지로 전송하는 웨이크-경고 채널의 능력을 추가로 개선하기 위해 채택될 수 있다.

[0039] 일부 구현에서, 유닛 상의 Cbo 로직 또는 다른 로직(가령, 대응하는 코어 및/또는 상호접속 인터페이스의 로직)은 조기의 웨이크-경고 신호를 생성하고 이를 웨이크-경고 채널(270) 상에서 전송하기 위한 로직을 포함할 수 있다. 웨이크-경고 신호는 특정 목적지 유닛을 타겟팅할 수 있고, 특정 캐시 बैं크와 같은 유닛 및 유닛의 서브구성요소를 식별할 수 있다. 일부 구현에서, 전용 웨이크-경고 채널은 다수의 배선을 포함할 수 있고, 웨이크-경고 신호는 목적지 유닛에 대응하는 그러한 배선 상에서 전송될 수 있다. 일부 예에서, 웨이크-경고 신호는 다수의 유닛을 타겟팅하거나 웨이크할 수 있다. 실제로, 그 설계는, 그러한 유닛 중 단지 하나 또는 서브세트가 궁극적인 요청을 수신하는 것을 통해, 유닛의 그룹이 웨이크-경고 신호에 의해 어웨이브되도록 허용할 수 있다. 그러한 경우에, 요청을 수신하지 않는 유닛은 짧은 듀레이션 동안에(예를 들면, 대응하는, 인입하는 요청을 검출하는데 실패할 때 타임아웃까지) 웨이크되고, 다시 디스에이블 상태로 게이팅될 수 있고, 한편 요청을 수신한 유닛(들)은 다른 예 중에서도 인에이블 상태에 있다.

[0040] 앞서 언급된 바와 같이, 일부 구현에서, 웨이크-경고 신호는, 중재, 일부 흐름 제어, 및 경로 룩업 로직을 포함하여 상호접속부(265)의 로직 중 일부를 제거한 전용 웨이크-경고 채널 상에서 전송될 수 있어서, 웨이크-경고 신호가 그와 연관된 요청보다 더 빠르게 목적지로 진행하도록 허용한다. 일부 경우에서, 웨이크-경고 신호는 목적지 유닛에 의해 처리되는 대응하는 요청의 전송과 관련하여 전송될 수 있다. 웨이크-경고 신호의 전송은, 웨이크-경고 신호가 요청의 전송 근처에서 타임 윈도우 내에 전송되도록 타이밍될 수 있다. 예를 들면, 일부 예에서, 대역폭이 타임 윈도우 내에서 이용 가능하면 언제라도, 웨이크-경고 신호는 웨이크-경고 채널(270)에 주입될 수 있고, 웨이크-경고 신호는 웨이크-경고 채널(270) 상의 양방향에서 진행될 수 있다. 인터페이스 로직(예를 들면, 인터페이스(예를 들면, 264, 266)에 있음)은 웨이크-경고 신호를 식별하기 위해 웨이크-경고 채널(270)을 모니터링하여 인터페이스와 연관된 대응하는 구성요소를 식별할 수 있다. 웨이크-경고 채널(270)의 프로토콜 및 정책은, 목적지 구성요소의 대응하는 요청이 웨이크-경고 신호와 동시에(또는 실질적으로 동시에) 전송되기 전에, 웨이크-경고 신호가 목적지 구성요소에 항상 도착하도록 웨이크-경고 신호의 전송을 유도하도록 구성될 수 있다. 이것은, 목적지 구성요소에서 디스에이블 상태로 게이팅된 클럭이 인입하는 요청의 도착 전에 인입하는 요청이 경고되어, 클럭(들)이 인에이블 상태로 게이팅되어 목적지 구성요소가 잠재적으로 요청이 도착한 순간에 요청을 처리하도록 허용하는 것을 보장할 수 있다.

[0041] 전용 웨이크-경고 채널(예를 들면, 270)은 잠재적으로 무제한적인 다양한 시스템 레이아웃, 아키텍처 또는 평면도 중 어느 하나로 상호접속부(예를 들면, 265)와 함께 제공될 수 있고, 예를 들면, SOC(system on chip)에 포함될 수 있는 잠재적으로 구성요소의 임의의 조합을 접속하는데 사용될 수 있다. 도 2의 예는 그러한 시스템의 하나의 간략한 예로서 인지되어야 한다. 코어-캐시 타일 이외의 구성요소 또는 유사한 구성요소는 또한 웨이크-경고 채널(예를 들면, 270) 및 상호접속부(예를 들면, 265)와 인터페이스하고, 조기의 웨이크-경고 기능의 이점을 누릴 수 있다. 실제로, 도 2의 예에 도시된 바와 같이, 다른 구성요소(예를 들면, 271, 272, 274), 메모리 제어기 및 PCIe, QPI 또는 다른 상호접속 구성요소와 같은 외부 인터페이스 구성요소는 웨이크-경고 채널(예를 들면, 270) 및 상호접속부(예를 들면, 265)와 인터페이스할 수 있다. 따라서, 조기의 웨이크-경고 신호는 그러한 구성요소로 어드레싱되고 그에 의해 소비될 수 있어서, 다른 구성요소 및 채널(예를 들면, 274, 276, 278, 280, 282, 284, 286, 288, 290, 292)이 또한 조기의 웨이크-경고 기능을 사용하여 잠재적으로 전달되는 전력 절약 및 다른 이점을 잠재적으로 누리도록 허용한다.

[0042] 도 3의 예로 넘어가면, 프로세서 코어(210) 및 LLC बैं크(230) 및 지원 로직(예를 들면, 대응하는 코어 박스(예를 들면, 250a), 캐시 박스(250b) 등(본 명세서에서 때때로 총괄적으로 "Cbo"(예를 들면, 250)로 지칭됨))을 포함하는 유닛과 같은 예시적인 시스템 유닛의 서브구성요소를 예시한 간략한 블록도(300)가 도시된다. 도 3의 특정 예에서, 상호접속 링크(265) 및 전용 웨이크-경고 채널 링크(270) 양자와 인터페이스할 수 있는 링 인터페이스(264)가 제공된다. 이러한 특정 예에서, 특정 캐시 बैं크(230)를 수반하여 요청을 처리하기 위한 캐시 박스 로직(250a)이 제공될 수 있다. 코어(210), 상호접속부(265)를 통해 요청을 통신하는 시스템의 다른 코어, 또는 캐시로부터 기록을 찾는 외부 구성요소를 포함하여 다른 구성요소로부터 요청이 직접적으로 수신될 수 있다. 진입 로직(330)은 캐시 बैं크(230)에 대한 인입하는 요청을 처리할 수 있다. C-파이프(310)는 요청을 프로세싱하기 위

한 로직을 포함할 수 있다. 예를 들면, 예컨대, 다른 예 중에서도, 모든 캐시 뱅크에 걸쳐 캐시 미스(cache miss)를 결정할 때, 또는 요청된 기록이 실제로 시스템 메모리 내에 있을 것 같다는 다른 결정 시에, 요청된 기록에 대응하는 시스템 메모리 내의 위치(예를 들면, 캐시 외부)를 결정하기 위한 SAD(system address decoder)(315)가 제공될 수 있다. 캐시 뱅크를 수반하여 캐시 코히어런시 관리를 제공하기 위한 코히어런시 로직(320)이 또한 제공될 수 있다. 응답 로직(325)은 수신된 요청에 대한 응답을 생성하기 위해 C-파이프 로직과 인터페이스할 수 있다. 그러한 응답은, 다른 예 중에서도, 캐시 뱅크(230)로부터 반환된 데이터(예를 들면, 요청된 기록이 캐시 뱅크에 존재하는 경우에), 코히어런시 메시지, "미스"를 나타내는 메시지(즉, 요청된 기록이 캐시 뱅크 상에 존재하지 않을 때), 기록이 발견될 수 있는 캐시 또는 메모리 내의 다른 위치를 지시하는 메시지를 포함할 수 있다.

[0043] 캐시 박스 로직(250b)에 의해 생성되는 응답은 출구 로직(330)의 사용을 통해 상호접속부(265) 상에서 전송하기 위해 마련될 수 있다. 출구 로직(330)은 응답의 목적지를 식별하고, 응답을 전송하기 위한 상호접속부 상의 경로(340)를 (예를 들면, 경로 로직(340)을 사용하여) 결정할 수 있다. 룩-업 테이블 또는 다른 데이터 및 로직은 응답에 대한 가장 적절한 경로를 결정하기 위해 경로 로직(340)에 의해 사용될 수 있다. 출구 로직(330)은 상호접속부(265) 상의 데이터의 전송을 중재하기 위한 중재 로직(335)을 더 포함할 수 있다. 캐시 박스 로직 또는 코어 박스 로직에 의해 전송될 메시지는, 일부 경우에서, 예를 들면, 상호접속부(265) 상에 존재하거나 막 주입된 다른 더 높은 우선순위 트래픽이 존재할 때, 큐잉될 수 있다. 중재 로직(335)은 출구 로직(330)을 사용하여 전송될 다수의 경합하는 메시지(예를 들면, 요청 및 응답)를 포함하여 상호접속 대역폭에 대해 경합하는 메시지의 주입을 중재할 수 있다. 출구 로직(330)은 데이터를 상호접속부(265)상에서 전송하기 위해 코어 박스 로직(250a) 및 캐시 박스 로직(250b) 양자에 의해 사용될 수 있다. 일부 구현에서, 코어 박스 로직(250a) 및 캐시 박스 로직(250b) 양자에 의해 사용하기 위한 단일 공유 출구 로직 모듈(예를 들면, 330)이 제공될 수 있지만, 한편 다른 예에서, 다른 예 중에서도, 코어 박스 로직(250a) 및 캐시 박스 로직(250b) 각각에 대한 전용 출구 로직이 제공될 수 있다.

[0044] 코어 박스 로직(250a)은 메모리, 캐시 또는 다른 시스템 자원에 대한 코어(예를 들면, 210)로부터 전송된 요청을 프로세싱하기 위한 로직을 포함할 수 있다. 요청의 생성을 위한 요청 생성 로직(345)이 제공될 수 있다. 요청 생성 로직(345)은 요청이 전송되는 목적지 구성요소의 어드레스를 식별할 수 있다. 출구 로직(330)은 요청을 전송하기 위해 상호접속부(265) 상의 최상의 경로를 식별하기 위해 어드레스 정보를 활용할 수 있다. 일 예에서, 코어 박스 로직(250a)은 요청 로직(345)을 사용하여 생성된 요청과 연관된 웨이크-경고 신호를 생성하기 위한 로직(350)을 더 포함할 수 있다. 웨이크-경고 로직(350)은 대응하는 요청 전에 목적지에 도달하도록 보장되는 웨이크-경고 신호를 생성할 수 있다. 일부 구현에서, 웨이크-경고 신호는 웨이크-경고 로직(350)에 의해 선택적으로 생성될 수 있다. 예를 들면, 로컬 캐시 뱅크(예를 들면, 230)를 타겟팅하는 요청에 대해, 어떠한 웨이크-경고 신호도 필요도 되지 않을 수 있다. 다른 예에서, 일부 구성요소는 조기의 웨이크-경고가 지원 또는 디스에이블되지 않는다는 것을 지정할 수 있고, 웨이크-경고 로직(350)은, 다른 예 중에서도, 그러한 구성요소로 예정된 요청과 관련하여 웨이크-경고 신호를 생성하는 것에 선행할 수 있다. 상호접속 인터페이스(264)는 코어(210)를 전용 웨이크-경고 채널에 접속하고, 전용 웨이크-경고 채널 상에서 생성된 웨이크-경고 신호를 웨이크-경고 신호의 목적지로 전송하는데 사용될 수 있다.

[0045] 도 3의 예에 도시된 유닛은 유닛(예를 들면, 캐시 뱅크(230))의 요청과 관련하여 웨이크-경고 채널(270)을 통해 다른 구성요소로부터 웨이크-경고 신호를 수신할 수 있다. 검출 로직에 대응하는 Cbo, 캐시 에이전트, 코어 등과 같은 구성요소로 예정된 웨이크-경고 채널(270) 상의 웨이크-경고 신호를 식별하기 위한 웨이크-경고 신호 검출 로직(예를 들면, 355)이 제공될 수 있다. 구성요소 또는 심지어 구성요소의 개별적인 서브-구성요소 또는 로컬 클록 버퍼(LCB)에 대한 웨이크-경고 신호를 검출할 때, 검출 로직(355)은 구성요소의 하나 이상의 클록 신호(및 버퍼)를 게이팅하도록 구성된 게이팅 로직(360)에 대한 웨이크-경고 이벤트를 식별할 수 있다. 예를 들면, 구성요소 또는 서브-구성요소와 연관된 클록 네트워크의 하나 이상의 RCB(regional clock buffer) 또는 다른 부분은 게이팅되고, 이로써 웨이크-경고 신호의 수신과 관련하여 인에이블될 수 있다. 클록이 (예를 들면, 전력-절약 모드에서) 디스에이블되도록 게이팅되면, 웨이크-경고 신호의 수신은 클록이 구성요소로 의도된 대응하는 요청의 수신에 앞서 게이팅 및 인에이블되게 할 수 있다. 클록이 이미 인에이블되면, 웨이크-경고 신호의 수신은 일부 경우에서 무시될 수 있다. 다른 구현에서, 검출 로직(355)은 클록을 인에이블 상태에서부터 디스에이블 상태로 게이팅하는 것과 관련하여 웨이크-경고 신호를 추가로 활용할 수 있다. 예를 들면, 검출 로직(355)은 이들 클록의 디스에이블 상태로의 게이팅을 트리거링하기 위해 클록 네트워크의 특정 부분에 의해 구동되는 구성요소 또는 서브-구성요소의 비활동을 검출하기 위한 기능을 더 포함할 수 있다. 일 예에서 타이머는 클록을 디스에이블 상태로 게이팅하도록 적용될 수 있다. 클록(또는 연관된 구성요소)으로 지향된 웨이크-경고 신호의



수신은 일부 구현에서 타이머로 하여금 재설정되게 할 수 있다. 그렇지 않다면, 웨이크-경고 신호의 주파수 및 이력은 게이팅과 연관된 전력 절약 노력을 최적화하기 위해 인에이블 상태에서 디스에이블 상태로의 클록의 게이팅을 타이밍 및 결정하는데 활용될 수 있다. 예를 들면, 구성요소 또는 서브-구성요소를 수반하여 검출된 유향 또는 앰프티 상태에 기초하여 비활동 상태의 검출을 카운팅 및 트리거링하기 위해 검출 로직(355)과 관련하여 카운터가 구현될 수 있다.

[0046] 일부 구현에서, 웨이크-경고 신호가 특정 Cbo(및 연관된 코어(예를 들면, 210) 및 캐시 뱅크(예를 들면, 230))와 같은 특정 구성요소를 타겟팅하게 함으로써 덜 입상(granular)의 클록 게이팅이 구현될 수 있다. 그러한 웨이크-경고 신호는 (가령, 구성요소의 하나 이상의 RCB(예를 들면, 365)의 레벨에서) 구성요소에 적용 가능한 복수의 클록 신호를 균일하게 게이팅하도록 프로세싱될 수 있다. 이것은, 코스(coarse), 그렇지만 효율적인 게이팅(예를 들면, Cbo-레벨에서)을 실현하면서, 웨이크-경고 시그널링의 실행을 간략화할 수 있다. 다른 구현은, 다른 예 중에서도, 더 입상의 레벨에서의 게이팅을 위해 더 많은 로컬 클록을 식별하고 구체적으로 이를 타겟팅하는 웨이크-경고 신호를 지원할 수 있다.

[0047] 단지 예시적인 목적으로 도 3의 예가 제공되고, 도 3의 예가 대응하는 클록 게이팅과 함께 조기의 웨이크-경고 시그널링 및 검출을 수행할 수 있는 로직의 다양한 구현을 제한하지 않는다는 것이 인지되어야 한다. 예를 들면, 코어 박스 로직(250a) 및 캐시 박스 로직(250b) 중 어느 하나 또는 양자와 관련하여 검출 로직(355), 게이팅 로직(360) 및 웨이크-경고 로직(350)이 제공될 수 있거나, 대안적으로, 예를 들면, 인터페이스 로직, 결합된 코어 캐시 Cbo 로직 또는 다른 예를 포함하여 다른 모듈과 관련하여 제공될 수 있다. 실제로, 검출 로직(355), 게이팅 로직(360) 및 웨이크-경고 로직(350)은 메모리 제어기, 허브 제어기와 같은 시스템의 코어 또는 캐시, 또는 다른 예시적인 구성요소 이외의 구성요소를 포함하는 다른 구성요소와 관련하여 제공될 수 있다.

[0048] 도 4로 넘어가면, (예를 들면, 415에서) 상호접속부 상에서 요청을 전송하고 (예를 들면, 420a, 420b에서) 별개의 웨이크-경고 채널 상에서 별개의 웨이크-경고 신호를 전송하기 위한 조기의 웨이크-경고 기능을 제공할 수 있는 구성요소의 Cbo 로직(425)에 의해 지원되는 예시적인 메시지 경로(405, 410)를 예시하는 간략한 블록도(400)가 도시된다. 도 4의 특정 예에서, 요청 경로(405)가 제공될 수 있고, 요청은 (예를 들면, 415에서) 상호접속부 상에서 요청을 전송하기 위해 제공되는 프로토콜과 관련하여 LUT(look-up table) 로직(430) 및 그후 중재 로직(435, 440)으로 포워딩되도록 Cbo 로직(425)에 의해 생성된다.

[0049] 상호접속부로 의도된 요청들의 더 집약적인 프로세싱은 (예를 들면, 420a, 420b에서) (경로(410)를 따른) 웨이크-경고 신호의 웨이크-경고 채널로의 전송으로 기인한 것을 초과하는 레이턴시를 발생시킬 수 있다. 예를 들면, 이러한 특정 예에서, 요청을 I/O 로직(425)으로부터 LUT 로직(430) 및 중재 로직(435)의 제 1 블록에 제공하기 위해 완전한 사이클이 활용될 수 있다. 레이턴시(455)의 부가적인 절반의 사이클은 제 1 중재 로직 블록(435)으로부터 제 2 중재 로직 블록(440)으로 실현될 수 있고, 부가적인 절반의 사이클(460)은 상호접속부 상의 요청에 대해 결정된 경로에 따라 요청을 상호접속부에 적절히 주입(415)하는데 사용될 수 있다.

[0050] 이와 대조적으로, 도 4의 예에서, 유사한 LUT(430) 또는 중재 로직(435, 440)에 선행하는 조기의 경고 경로(410)가 제공될 수 있어서, 웨이크-경고 신호가 생성되고, 웨이크-경고 채널에 주입되고, 상호접속부를 사용하여 요청을 (경로(405)를 통해) 동일한 목적지로 전송하는데 사용되는 것보다 더 적은 사이클로 목적지에 도달하도록 허용한다. 예를 들면, 이러한 간략한 예에서, 경로(405)를 통해 링 상의 요청을 획득하기 위해 2 개의 완전한 사이클이 활용되고, 한편 단일 사이클(절반 사이클(465) 및 절반 사이클(470a, 470b)을 포함함)은 대응하는 웨이크-경고 신호를 웨이크-경고 채널에 주입하도록 활용된다. 웨이크-경고 신호의 전송은 적어도 동일한 목적지로의 상호접속부 상의 요청의 전송만큼 빠를 수 있다. 실제로, 이러한 예에서, 특정 요청에 대응하는 웨이크-경고 신호가 요청의 생성에서 절반 사이클 뒤에 생성될지라도, 이러한 예에서, 웨이크-경고 신호는 여전히 요청의 상호접속부로의 주입 전에 자신의 채널로 주입될 수 있어서, 웨이크-경고 신호가 요청 전에 목적지에 도달하고 인입하는 요청을 예상하고 목적지에서 클록의 임의의 원하는 게이팅을 발생시키도록 허용한다. 일부 구현에서, 웨이크-경고 신호가 생성되고, 경로(405)를 따른 대응하는 요청의 생성 및 진행과 동시에 경로(410)를 따라 진행될 수 있다. 실제로, 대응하는 조기의 웨이크-경고 신호의 부재 시에 경로(410)에서 관찰되는 것에 대해 어떠한 부가적인 레이턴시도 요청 경로(405)에 부가하지 않는 조기의 웨이크-경고 기능이 제공될 수 있다. 따라서, 조기의 웨이크-경고 기능의 제공을 통해 요청 레이턴시가 절충될 필요가 없다.

[0051] 대응하는 요청의 정상 진행에 따른 웨이크-경고 신호들의 전송을 타이밍하는 것과 관련하여, 일부 구현에서, 요청 경로(405)에서 관련 요청의 대응하는 버퍼링을 처리하기 위해 웨이크-경고 신호를 버퍼링하기 위한 부가적인 로직이 제공될 수 있다. 중재 및 다른 흐름 제어를 활용하는 상호접속 정책이 주어지면, 일부 경우에서, 요청에

대해 결정된 상호접속부 또는 경로의 이용 가능성을 펜딩하여 요청이 큐잉될 수 있다(예를 들면, 445). 웨이크-경고 신호가 또한 큐잉되어야 하는지를 결정하기 위해 요청 경로의 큐 상태(445)를 모니터링하는 큐잉 제어부(448)가 조기의 경고 경로(410) 상에 제공될 수 있다. 일부 구현에서, 웨이크-경고 채널의 간략화가 주어지면, 웨이크-경고 신호가 그들의 대응하는 요청보다 훨씬 더 일찍 전송되지 않는 것을 보장하는 것이 중요할 수 있다. 일부 구현에서, 웨이크-경고 신호를 수신하는 구성요소는, 다른 예 중에서도, 인입하는 요청이 웨이크-경고 신호 다음에 특정 윈도우 내에 도착할 것이라는 전제로, 웨이크-경고 신호를 대응하는 인입하는 요청으로 맵핑할 수 있다. 일 예에서, 웨이크-경고 신호는 대응하는 요청이 수신 유닛에 의해 버퍼링되는 기간에 대응하는 기간 동안에 수신 유닛의 클럭을 언계이팅하도록 허용될 수 있고, 이로써, 다른 예 중에서도, 대응하는 요청이 프로세싱될 예상 기간에 대응하는 기간 동안에 웨이크-경고 신호의 효과를 유지한다. 따라서, 일부 경우에서, 웨이크-경고 신호는, 다른 예 중에서도, 요청 경로(405) 상의 중재 또는 다른 로직에 의해 요청 상에 부여된 지연에 따라 웨이크-경고 신호의 전송을 지연시키도록(예를 들면, 448에서) 부가적으로 버퍼링될 수 있다.

[0052] 또한, 도 4의 특정 예에 도시된 바와 같이, 요청이 상호접속부 상의 특정 정의된 경로에 따라(예를 들면, 링 상호접속부의 상향 또는 하향 방향으로) 전송될 수 있지만, 웨이크-경고 신호는 그렇게 제약되지 않을 수 있고, 웨이크-경고 채널에 접속된 모든 다른 구성요소가 웨이크-경고 신호를 보도록 웨이크-경고 채널의 설계에 의해 제공된 바와 같이 다수의 방향으로 효과적으로 방송될 수 있다. 다른, 더 많이 제어되는 프로토콜이 웨이크-경고 신호의 구현에 적용될 수 있지만, 도 4의 특정 예에서, 웨이크-경고 채널은(상호접속부와 같이) 링-스타일 상호접속에서 배향된다. 또한, (예를 들면, 가장 짧은 또는 가장 빠른 경로에 따라)(양방향) 웨이크-경고 채널 상에서 특정 방향으로 웨이크-경고 신호를 전송하는 것보다는, 다른 예 중에서도, (예를 들면, PCIe, QPI 또는 다른 상호접속 아키텍처 정책 및 프로토콜을 사용하여) 링 상호접속부 상에서 대응하는 요청을 전송(415)하기 위한 레이턴시에 대해, 웨이크-경고 신호의 전송과 연관된 레이턴시를 추가로 간략화 및 감소시키는 것을 보조하기 위해, 웨이크-경고 신호는 웨이크-경고 링 상에서 양방향으로 동시에(420a, 420b) 전송될 수 있다.

[0053] 도 5a 내지 도 5c의 간략한 블록도(500a-c)는 본 명세서 다른 곳에서 설명되는 것과 같은 원리를 활용하는 시스템에서 하나의 구성요소로부터 다른 것으로 전송되는 요청 및 웨이크-경고 신호의 특정 예를 예시한다. 도 5a의 예에서, 코어(206)는 코어(206)가 결정한 캐시 뱅크(230) 상에 위치한 캐시 기록에 대한 요청을 생성한다. 요청(515)은 웨이크-경고 신호(520)와 함께 생성되고, 예를 들면, Cbox(246), 상호접속 인터페이스(505) 또는 다른 모듈에서 I/O 로직을 사용하여 상호접속부(265) 상에서 전송된다. 웨이크-경고는 마찬가지로 전송되지만 웨이크-경고 채널(270) 상에서 전송된다. 도 5a에 표현된 바와 같이, 웨이크-경고 채널은, 요청(515)을 주입하고 요청(515)을 상호접속부(270) 상에서 전송하는데 걸리는 것보다 더 적은 사이클에서 웨이크-경고 신호가 주입되고 목적지(예를 들면, Cbox(250))로 전송되는 것을 가능하게 할 수 있다. 이것은, 요청(515)을 전송하는데 사용될 수 있는 최소의 레이턴시를 보존하면서, 대응하는 요청 모두의 도착을 지연에 방지할 웨이크-경고 신호가 지속적으로 그리고 신뢰할 수 있게 전송되도록 허용한다. 다시 말해서, 요청(예를 들면, 515)의 도착 또는 프로세싱을 연기하는 어떠한 레이턴시도 요청 경로에 부가되지 않아서, 요청의 목적지(예를 들면, 250)에 경고하고, 목적지를 게이팅하도록 허용한다.

[0054] 도 5a의 예에 도시된 바와 같이, 웨이크-경고 신호(520)는 복수의 Cbo(예를 들면, 246, 248, 250) 중 특정 하나 또는 시스템의 유닛을 제어하는 다른 구성요소를 타겟팅하도록 전송될 수 있다. 웨이크-경고 신호(520)는 웨이크-경고 신호가 Cbo(250)(및 더 나아가, 코어(210) 및 캐시 뱅크(230))로 의도된다고 결정하기 위해 인터페이스(264)(또는 Cbo 로직(250))에 의해 식별될 수 있다. 또한, 이러한 예에서, 웨이크-경고 신호(520)의 도착 전에, Cbo(250)의 클럭 신호는 클럭을 디스에이블하도록 게이팅된다. 웨이크-경고 신호(520)를 수신하는 것에 응답하여, Cbo(250)의 게이팅 로직은 클럭(525)을 인에이블하고, 이로써 대응하는 요청(515)의 도착을 위해 Cbox(250)를 준비시킬 수 있다. 실제로, 도 5b의 예로 넘어가면, 웨이크-경고 신호(520)가 검출되고, 프로세싱되고, 클럭(525)이 인에이블되는 경우에, Cbox(250)는(도 5a에서) 웨이크-경고 신호(520)의 도착 후의 시간에서 어웨이크되고 요청(515)의 도착을 위해 준비될 수 있다. 예를 들면, 요청(515)은 웨이크-경고 신호(520)가 도착하고 하나 이상의 사이클 후에 도착할 수 있다.

[0055] 이전의 예에서 계속하여 그리고 도 5c의 간략한 블록도(500c)로 넘어가면, Cbox(250)는 요청(515)을 프로세싱하고, 예를 들면, 캐시 미스를 결정하고, 또한 요청(515)에 대응하는 기록이 대신에 시스템 메모리에 위치된다고 결정할 수 있다. Cbox(250)는 기록이 저장되는 것으로 이해되는 시스템 메모리를 관리하는 메모리 제어기(530)를 식별하기 위한 로직을 포함하고, 기록에 대한 요청을 메모리 제어기(530)로 포워딩하기 위해 원래 요청자(예를 들면, 코어(206)) 대신에 부가적인 요청(540)을 생성할 수 있다. 또한, Cbox(250)는 마찬가지로 대응하는 웨이크-경고 신호(545)가 인입하는 요청(540)을 메모리 제어기(530)에 경고하도록 허용하는 로직을 포함하거나 이

에 대한 액세스를 가질 수 있다. 따라서, 웨이크-경고 신호(545)는 또한 웨이크-경고 채널(270)에 주입되고, 요청(540)의 도착 전에 메모리 제어기(530)에 도착할 수 있다. 메모리 제어기(530)가 클록 게이팅을 지원하고, 메모리 제어기(530)의 관련 클록(550)이 디스에이블 상태로 게이팅되면, (상호접속 인터페이스(535)에서) 웨이크-경고 신호의 수신은, 다른 잠재적인 예 및 상호작용 중에서도, 메모리 제어기(530)의 게이팅 로직이 클록(550)을 인에이블 상태로 게이팅하고, 요청(540)의 도착 및 프로세싱을 위해 메모리 제어기(530)(또는 메모리 제어기의 특정한, 관련 서브-구성요소)를 준비시키게 할 수 있다.

[0056] 이제 도 6a 및 도 6b의 간략한 흐름도(600a-b)로 넘어가면, 시스템에서 조기의 웨이크-경고 기능의 제공과 관련하여 예시적인 기술이 예시된다. 예를 들면, 도 6a의 예에서, 상호접속부를 통해 시스템 내의 다른 구성요소, 가령, 시스템-온-칩 상의 캐시 뱅크로 전송되는 요청이 생성될 수 있다(605). 클록 게이팅은 그 구성요소에서 적용될 수 있다. 인입하는 요청을 시스템 구성요소에 경고하고, (예를 들면, 클록의 전력 절약 게이팅과 관련하여) 클록이 디스에이블된 경우에 클록을 인에이블하기 위해 시스템 구성요소에서의 하나 이상의 클록의 게이팅 (또는 더 정확하게는, 언게이팅)을 트리거링하기 위한 웨이크-경고 신호가 생성되어, 상호접속부와 별개인 전용 채널을 통해 시스템 구성요소로 전송될 수 있다(610). 요청은 상호접속부를 통해 전송될 수 있다(615). 웨이크-경고 신호가 (요청의 버퍼링 없이) 요청 전에 시스템 구성요소에 도착하여 언게이팅이 요청의 도착에 앞서 발생하도록 허용하는 것을 요청과 동시에 또는 거의 동시에 웨이크-경고 신호를 생성하고 전송하는 것이 여전히 허용하고, 일부 경우에, 보장할 수 있도록 웨이크-경고 채널이 구성될 수 있다.

[0057] 도 6b의 예에서, 채널 상의 적용 가능한 웨이크-경고 신호에 대해 전용 웨이크-경고 채널이 모니터링될 수 있다. 웨이크-경고 신호는 다른 시스템 구성요소로부터 웨이크-경고 채널을 통해 수신될 수 있다(620). 클록, 예를 들면, 로컬 또는 영역의 클록 버퍼 또는 심지어 클록 스파인은 웨이크-경고 신호의 수신(620)에 기초하여 언게이팅될 수 있다(625). 예를 들면, 웨이크-경고 신호가 수신/검출될 때, 클록이 디스에이블되면, 웨이크-경고 신호의 수신은 디스에이블 상태에서부터 인에이블 상태로의 클록의 언게이팅을 트리거링할 수 있다. 웨이크-경고 신호의 수신(620) 및 대응하는 클록 게이팅(625)은 상호접속부를 통해 다른 시스템 구성요소로부터 요청을 수신하기(630) 전에 완료될 수 있고, 요청은 전용 웨이크-경고 채널을 통해 수신(620)되는 이전에 수신된 웨이크-경고 신호에 대응한다. 클록이 인에이블된 경우에, 요청이 프로세싱될 수 있다(635). 예를 들면, 요청은 캐시 뱅크의 캐시 기록을 요청할 수 있고, 요청은 캐시 기록이 캐시에 존재하는지 및 그렇지 않고, 수신하는 시스템 구성요소가 연관될 수 있는 상호접속부를 사용하여 접속된 구성요소 사이의 잠재적으로 많은 다른 형태의 요청 및 트랜잭션 중에서, 요청 기록이 위치할 가능성이 있는 곳(예를 들면, 다른 캐시 뱅크 또는 메모리)을 결정하기 위해 프로세싱될 수 있다(635).

[0058] 앞서 설명된 장치, 방법 및 시스템이 전술된 임의의 전자 디바이스 또는 시스템에서 구현될 수 있다는 것에 유의한다. 특정 예시로서, 아래의 예는 본 명세서에 설명된 본 발명을 활용하기 위한 예시적인 시스템을 제공한다. 아래의 시스템이 더 상세히 설명되기 때문에, 다수의 상이한 상호접속부가 개시, 설명되고, 앞선 논의로부터 개정된다. 그리고 용이하게 명백한 바와 같이, 앞서 설명된 발전은 그러한 상호접속부, 패브릭 또는 아키텍처 중 임의의 것에 적용될 수 있다.

[0059] 이제 도 7을 참조하면, 본 발명의 실시예에 따른 제 2 시스템(700)의 블록도가 도시된다. 도 7에 도시된 바와 같이, 멀티프로세서 시스템(700)은 점 대 점 상호접속 시스템이며, 점 대 점 상호접속부(750)를 통해 결합되는 제 1 프로세서(770) 및 제 2 프로세서(780)를 포함한다. 프로세서들(770 및 780) 각각은 일정 버전의 프로세서일 수 있다. 일 실시예에서, (752 및 754)는 인텔의 퀵 패스 상호접속(QPI) 아키텍처와 같은 직렬 점 대 점 코히어런트 상호접속 패브릭의 일부이다. 결과적으로, 본 발명은 QPI 아키텍처 내에서 구현될 수 있다.

[0060] 단 두 개의 프로세서들(770, 780)을 가진 것으로 도시되지만, 본 발명의 범위가 그렇게 국한되는 것은 아니라는 것을 알아야 한다. 다른 실시예들에서, 하나 이상의 부가적인 프로세서들이 주어진 프로세서 내에 존재할 수 있다.

[0061] 프로세서들(770 및 780)은 각각 통합된 메모리 제어기 유닛들(772 및 782)을 포함하는 것으로 보여진다. 프로세서(770)는 자신의 버스 제어기 유닛들의 일부로서 점 대 점(P-P) 인터페이스들(776 및 778)을 또한 포함한다; 마찬가지로 제 2 프로세서(780)는 P-P 인터페이스들(786 및 788)을 포함한다. 프로세서들(770, 780)은 점 대 점(P-P) 인터페이스 회로들(778, 788)을 이용하여 P-P 인터페이스(750)를 통해 정보를 교환할 수 있다. 도 7에 도시된 바와 같이, IMC들(772 및 782)은 프로세서들을 각자의 메모리들, 즉 메모리(732) 및 메모리(734)에 결합시키며, 이 메모리들은 각자의 프로세서들에 내부적으로 부착된 메인 메모리의 일부들일 수 있다.

[0062] 프로세서들(770, 780)은 각각 점 대 점 인터페이스 회로들(776, 794, 786, 798)을 이용하여 개별 P-P 인터페이스



스들(752, 754)을 통해 칩셋(790)과 정보를 교환한다. 칩셋(790)은 역시 고성능 그래픽 상호접속부(739)를 따라 인터페이스 회로(792)를 통해 고성능 그래픽 회로(738)와 정보를 교환한다.

- [0063] 공유된 캐시(미도시)가 어느 한 프로세서 안이나 두 프로세서들 외부에 포함될 수 있지만, P-P 상호접속을 통해 프로세서들과 연결되어, 어느 한 프로세서나 두 프로세서들 모두의 로컬 캐시 정보가 어느 프로세서가 저전력 모드에 놓이게 되면 공유 캐시 안에 저장될 수 있도록 할 수 있다.
- [0064] 칩셋(790)은 인터페이스(796)를 통해 제 1 버스(716)에 연결될 수 있다. 일 실시예에서 제 1 버스(716)는 PCI(Peripheral Component Interconnect) 버스이거나, PCI 익스프레스 버스나 다른 제 3 세대 I/O 상호접속 버스와 같은 버스일 수 있지만, 본 발명의 범위가 그렇게 한정되는 것은 아니다.
- [0065] 도 7에 도시된 바와 같이, 다양한 I/O 디바이스들(714)은 제 1 버스(716)를 제 2 버스(720)에 결합시키는 버스 브리지(718)와 함께 제 1 버스(716)에 연결된다. 일 실시예에서 제 2 버스(720)는 낮은 핀 카운트(LPC) 버스를 포함한다. 다양한 디바이스들이 예컨대, 일 실시예에서 키보드 및/또는 마우스(722), 통신 디바이스들(727) 및 종종 명령들/코드 및 데이터(730)를 포함하는 디스크 드라이브 또는 다른 대용량 저장 디바이스와 같은 저장 유닛(728)을 포함하는 제 2 버스(720)에 연결된다. 또한 오디오 I/O(724)가 제 2 버스(720)에 연결되는 것으로 보여진다. 포함된 구성요소들과 상호접속 구조들이 가변되는 다른 구조들이 가능하다는 것을 알아야 한다. 예를 들어 도 7의 점 대 점 구조 대신, 시스템은 멀티-드롭 버스나 그러한 다른 구조를 구현할 수 있다.
- [0066] 본 발명이 한정된 수의 실시예들에 관하여 설명되었지만, 당업자는 다수의 변형들 및 그로부터의 변경들을 이해할 것이다. 첨부한 청구항들이 본 발명의 실제 사상 및 범위 내에 있는 한은 모든 이러한 변형들 및 변경들을 커버하는 것으로 의도된다.
- [0067] 설계는 작성으로부터 시뮬레이션으로 그리고 제조로의 다양한 스테이지들을 통과할 수 있다. 설계를 나타내는 데이터는 다수의 방식으로 설계를 나타낼 수 있다. 먼저, 시뮬레이션들에서 유용하기 때문에, 하드웨어는 하드웨어 기술 언어 또는 다른 기능적 기술 언어를 사용하여 표현될 수 있다. 추가로, 로직 및/또는 트랜지스터 게이트들을 갖는 회로 레벨 모델이 설계 프로세스의 일부 스테이지들에서 생성될 수 있다. 또한, 일부 스테이지에서, 대부분의 설계들은 하드웨어 모델에서의 다양한 디바이스들의 물리적 배치를 나타내는 데이터의 레벨에 도달한다. 종래의 반도체 제조 기법들이 사용되는 경우에, 하드웨어 모델을 나타내는 데이터는 집적 회로를 생성하기 위해 사용된 마스크들에 대해 상이한 마스크 층들 상에 다양한 특징들의 존재 또는 부재를 특징하는 데이터일 수 있다. 설계의 임의의 표현에서, 데이터는 임의의 형태의 머신 판독가능 매체에 저장될 수 있다. 디스크와 같은 메모리 또는 자기 또는 광학 스토리지가 정보를 송신하기 위해 변조되거나 그렇지 않으면 생성된 광학 또는 전기파를 통해 송신된 정보를 저장하기 위한 머신 판독가능 매체일 수 있다. 코드 또는 설계를 나타내거나 반송하는 전기 반송파가 전기 신호의 카피, 버퍼링, 또는 재송신이 수행된 결과 송신될 때, 새로운 카피가 이루어진다. 따라서, 통신 제공자 또는 네트워크 제공자는 본 발명의 실시예들이 기법들을 실시하는, 반송파로 인코딩된 정보와 같은 물건을 유형의 머신 판독가능 매체상에 적어도 일시적으로 저장할 수 있다.
- [0068] 본원에 사용된 바와 같은 모듈은 하드웨어, 소프트웨어, 및/또는 펌웨어의 임의의 조합을 나타낸다. 일례로, 모듈은 마이크로제어기에 의해 실행되도록 적용된 코드를 저장하는 비-일시적 매체와 연관되는 마이크로컨트롤러와 같은 하드웨어를 포함한다. 그러므로, 일 실시예에서, 모듈에 대한 언급은 비-일시적 매체 상에 유지될 코드를 인식 및/또는 실행하도록 특별하게 구성되는 하드웨어를 나타낸다. 또한, 다른 실시예에서, 모듈의 사용은 소정 동작들을 수행하는 마이크로컨트롤러에 의해 실행되도록 특별하게 적용되는 코드를 포함하는 비-일시적 매체를 나타낸다. 그리고, 추론될 수 있는 바와 같이, 또 다른 실시예에서, (이 예에서의) 용어 모듈은 마이크로컨트롤러 및 비-일시적 매체의 조합을 나타낼 수 있다. 종종, 개별적으로 도시되는 모듈 경계들은 공통적으로 변화하거나 잠재적으로 중복된다. 예를 들어, 제 1 및 제 2 모듈은 일부 독립적 하드웨어, 소프트웨어, 펌웨어를 잠재적으로 유지하면서, 하드웨어, 소프트웨어, 펌웨어, 또는 이의 조합을 공유할 수 있다. 일 실시예에서, 용어 로직의 사용은 트랜지스터들, 레지스터들과 같은 하드웨어, 또는 프로그래밍 가능한 로직 디바이스들과 같은 다른 하드웨어를 포함한다.
- [0069] 일 실시예에서, 구 '하기 위한' 또는 '하도록 구성된'의 사용은 지정되거나 결정된 태스크를 수행하는 장치, 하드웨어, 로직, 또는 요소를 배열, 조립, 제조, 구매 제안, 수입 및/또는 설계하는 것을 나타낸다. 이 예에서, 동작하고 있지 않은 장치 또는 이의 요소는 지정된 태스크를 수행하기 위하여 설계, 연결, 및/또는 상호접속되는 경우에 여전히 지정된 태스크를 수행하도록 구성된다. 순전히 설명적인 예에서, 로직 게이트는 동작 동안 0 또는 1을 제공할 수 있다. 그러나, 클록에 인에이블 신호를 '제공하도록 구성된' 로직 게이트는 1 또는 0을 제공할 수 있는 모든 잠재적인 로직 게이트를 포함하지는 않는다. 그 대신에, 로직 게이트는 동작 동안 1 또는 0

출력이 클록을 인에이블하도록 하는 어떤 방식으로 연결된 것이다. 용어 '하도록 구성된'의 사용이 동작을 필요로 하는 것이 아니라, 장치, 하드웨어, 및/또는 요소의 잠재적인 상태에 초점을 맞춘다는 것을 다시 한번 주의하고, 잠재적인 상태에서, 장치, 하드웨어, 및/또는 요소는 이들이 동작하고 있을 때 특정 태스크를 수행하도록 설계된다.

[0070] 또한, 일 실시예에서, 구들 '할 수 있는/하기 위한' 및/또는 '하도록 동작 가능한'의 사용은 지정된 방식으로 장치, 로직, 하드웨어, 및/또는 요소의 사용을 가능하게 하는 방식으로 설계된 어떤 장치, 로직, 하드웨어, 및/또는 요소를 나타낸다. 일 실시예에서, '하기 위한', '할 수 있는' 또는 '하도록 동작 가능한'의 사용이 장치, 로직, 하드웨어, 및/또는 요소가 동작하고 있지 않지만, 지정된 방식으로 장치의 사용을 가능하게 하는 방식으로 설계된 경우에 장치, 로직, 하드웨어, 및/또는 요소의 잠재적인 상태를 나타낸다는 것을 상술된 바와 같이 주의하라.

[0071] 본원에 사용된 바와 같은 값은 수, 상태, 논리 상태, 또는 이진 논리 상태의 임의의 공지된 표현을 포함한다. 종종, 로직 레벨들, 로직 값들, 또는 논리 값들의 사용은 이진 로직 상태들을 간단하게 표현하는 1들 및 0들이라 칭해진다. 예를 들어, 1은 하이 로직 레벨을 나타내고, 0은 로우 로직 레벨을 나타낸다. 일 실시예에서, 트랜지스터 또는 플래시 셀과 같은 스토리지 셀은 단일 논리 값 또는 다중 논리 값들을 유지할 수 있다. 그러나, 컴퓨터 시스템들에서의 값들을 다른 표현들이 사용되었다. 예를 들어, 십진수 10은 1110의 이진값 및 16진수 문자 A로서 표현될 수 있다. 그러므로, 값은 컴퓨터 시스템에 유지될 수 있는 정보의 임의의 표현을 포함한다.

[0072] 더구나, 상태들은 값들 또는 값들의 부분들에 의해 표현될 수 있다. 일례로서, 논리 1과 같은 제 1 값은 디폴트 또는 초기 상태를 나타낼 수 있는 반면, 논리 0과 같은 제 2 값은 논-디폴트 상태(non-default state)를 나타낼 수 있다. 또한, 일 실시예에서, 용어들 리셋 및 세트는 각각 디폴트 및 갱신 값 또는 상태를 나타낸다. 예를 들어, 디폴트 값은 잠재적으로 하이 논리 값, 즉, 리셋을 포함하는 반면, 갱신 값은 잠재적으로 로우 논리 값, 즉, 세트를 포함한다. 값들의 임의의 조합이 임의의 수의 상태들을 나타내는데 사용될 수 있다는 것을 주의하라.

[0073] 상술된 방법들, 하드웨어, 소프트웨어, 펌웨어, 또는 코드 세트의 실시예들은 프로세싱 요소에 의해 실행 가능한 머신-엑세스 가능하거나, 머신 판독 가능하거나, 컴퓨터 액세스 가능하거나, 컴퓨터 판독 가능한 매체 상에 저장된 명령들 또는 코드를 통하여 구현될 수 있다. 비-일시적 머신-엑세스 가능한/판독 가능한 매체는 컴퓨터 또는 전자 시스템과 같은 머신에 의해 판독 가능한 형태로 정보를 제공하는 (즉, 저장 및/또는 송신하는) 임의의 메커니즘을 포함한다. 예를 들어, 비-일시적 머신-엑세스 가능한 매체는 정보가 수신될 수 있는 비-일시적 매체들과 구별되도록 의도된 정적 램(SRAM) 또는 동적 램(DRAM)과 같은 RAM(random access memory; ROM(read only memory); 자기 또는 광 저장 매체; 플래시 메모리 디바이스들; 전기 저장 디바이스들; 광 저장 디바이스들; 음향 저장 디바이스들; 일시적 (전파) 신호들(예를 들어, 반송파들, 적외선 신호들, 디지털 신호들)로부터 수신된 정보를 유지하기 위한 다른 형태의 저장 디바이스들; 등을 포함한다.

[0074] 본 발명의 실시예들을 수행하는 로직을 프로그래밍하는데 사용되는 명령들은 DRAM, 캐시, 플래시 메모리, 또는 다른 스토리지와 같은, 시스템 내의 메모리 내에 저장될 수 있다. 또한, 명령들은 네트워크를 통하여 또는 다른 컴퓨터 판독 가능한 매체들을 경유하여 분배될 수 있다. 따라서, 머신-판독 가능한 매체는 플로피 디스크들, 광 디스크들, CD-ROM (compact disc read only memory), 자기-광 디스크들, ROM, RAM, 소거 가능하고 프로그래밍 가능한 ROM(EPROM), 전기적으로 소거 가능하고 프로그래밍 가능한 ROM(EEPROM), 자기 또는 광 카드들, 플래시 메모리, 또는 전기, 광, 음향, 또는 다른 형태의 전파 신호들(예를 들어, 반송파들, 적외선 신호들, 디지털 신호들, 등)을 통한 인터넷을 통한 정보의 전송에서 사용되는 유형 머신-판독 가능한 스토리지를 포함하지만, 이에 제한되지 않는, 머신(예를 들어, 컴퓨터)에 의해 판독 가능한 형태로 정보를 저장 또는 송신하기 위한 임의의 메커니즘을 포함할 수 있다. 따라서, 컴퓨터-판독 가능한 매체는 머신(예를 들어, 컴퓨터)에 의해 판독 가능한 형태로 전자 명령들 또는 정보를 저장 또는 송신하는데 적합한 임의의 타입의 유형 머신-판독 가능한 매체를 포함한다.

[0075] 다음의 예는 본 명세서에 따른 실시예에 관한 것이다. 하나 이상의 실시예는 특정 캐시 기록과 연관된 요청을 생성하기 위한 장치, 시스템, 머신 판독 가능 저장소, 머신 판독 가능 매체 및 방법을 제공할 수 있고, 여기서 요청은 상호접속부를 통해 캐시 뱅크와 연관된 시스템 구성요소로 전송된다. 요청이 도착한다는 것을 시스템 구성요소에 표시하는 웨이크-경고 신호는 전용 웨이크-경고 채널을 통해 전송될 수 있다. 웨이크-경고 신호는 디스에이블 클록이 인에이블 상태로 언계이팅되게 할 수 있다. 그후, 요청은 시스템 구성요소로 전송된다.

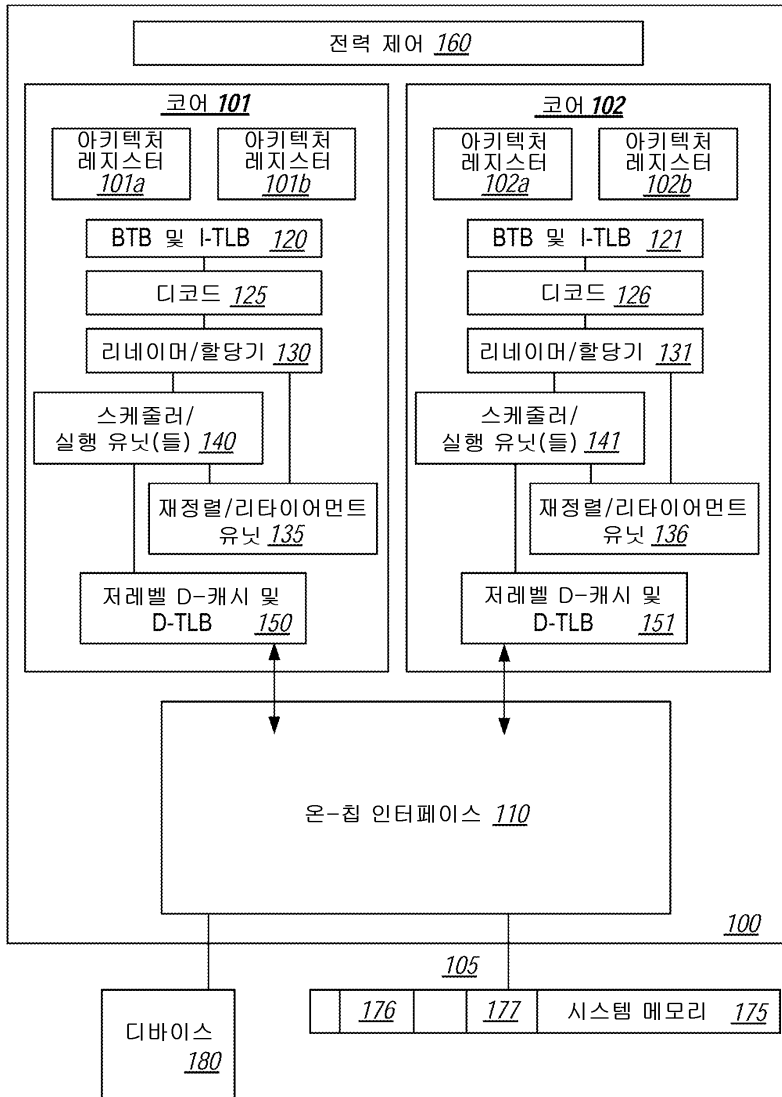
- [0076] 적어도 하나의 예에서, 요청은 전용 웨이크-경고 채널로부터 별개인 상호접속부를 사용하여 시스템 구성요소로 전송된다.
- [0077] 적어도 하나의 예에서, 상호접속부는 링 상호접속부를 포함한다.
- [0078] 적어도 하나의 예에서, 웨이크-경고 채널은 장치를 적어도 시스템 구성요소에 접속하는 링으로서 배향된다.
- [0079] 적어도 하나의 예에서, 시스템 구성요소는 캐시 뱅크에 대응하는 캐시 박스 로직을 포함하고, 클록은 캐시 박스 로직에 의해 활용된다.
- [0080] 적어도 하나의 예에서, 로직은 또한 요청의 전송 전에 요청의 버퍼링(buffering)을 식별하고, 요청의 버퍼링에 기초하여 웨이크-경고 신호의 전송 전에 웨이크-경고 신호를 버퍼링한다.
- [0081] 적어도 하나의 예에서, 웨이크-경고 신호는 시스템 구성요소에서의 요청의 도착 전에 시스템 구성요소에 도착한다.
- [0082] 적어도 하나의 예에서, 웨이크-경고 신호는 요청의 전송과 실질적으로 동시에 전송된다.
- [0083] 적어도 하나의 예에서, 웨이크-경고 채널은 양방향 채널이고, 웨이크-경고 신호는 웨이크-경고 채널 상에서 양방향으로 전송된다.
- [0084] 적어도 하나의 예에서, 웨이크-경고 신호는 캐시 뱅크에 대응하는 영역의 클록 버퍼의 게이팅에 영향을 준다.
- [0085] 적어도 하나의 예에서, 웨이크-경고 신호는 인에이블 클록의 디스에이블링이 연기되게 한다.
- [0086] 하나 이상의 실시예들은 다른 시스템 구성요소로부터 시스템의 전용 웨이크-경고 채널을 통해 전송되는 웨이크-경고 신호를 수신하기 위한 장치, 시스템, 머신 판독 가능 저장소, 머신 판독 가능 매체, 및 방법을 제공할 수 있고, 웨이크-경고 신호는 웨이크-경고 신호 다음에 인입하는 요청을 표시한다. 클록은, 웨이크-경고 신호에 적어도 부분적으로 기초하여, 인입하는 요청의 수신 전에 클록을 인에이블하도록 연게이팅될 수 있다. 인입하는 요청은 상호접속부를 통해 수신될 수 있고, 인입하는 요청이 프로세싱되게 할 수 있다.
- [0087] 적어도 하나의 예에서, 클록은, 웨이크-경고 신호가 수신될 때 디스에이블된다.
- [0088] 적어도 하나의 예에서, 로직은 또한 클록에 대응하는 특정 시스템 구성요소의 비활동 조건을 검출하고, 비활동 조건에 기초하여 클록을 디스에이블하도록 클록을 게이팅한다.
- [0089] 적어도 하나의 예에서, 비활동 조건은 비활동 타이머에 기초한다.
- [0090] 적어도 하나의 예에서, 클록이 인에이블될 때 수신되는 웨이크-경고 신호는 비활동 타이머가 재시작되게 한다.
- [0091] 적어도 하나의 예에서, 요청은 캐시 기록에 대한 것이고, 요청은, 캐시 기록이 클록에 대응하는 특정 캐시 뱅크에 저장되는지를 결정하도록 프로세싱된다.
- [0092] 적어도 하나의 예에서, 요청은 제 1 요청을 포함하고, 로직은 또한, 제 1 요청의 프로세싱에 기초하여 특정 시스템 구성요소로 전송될 제 2 요청을 생성하고, 제 2 요청이 도착한다는 것을 특정 시스템 구성요소에 표시하기 위한 다른 웨이크-경고 신호를 전용 웨이크-경고 채널을 통해 전송한다. 다른 웨이크-경고 신호는 특정 시스템 구성요소에서의 디스에이블 클록이 인에이블 상태로 연게이팅되게 하고, 로직은 또한 제 2 요청이 특정 시스템 구성요소로 전송되게 한다.
- [0093] 적어도 하나의 예에서, 제 2 요청은 제 1 요청의 프로세싱으로부터의 캐시 미스(cache miss)의 결정에 기초한다.
- [0094] 적어도 하나의 예에서, 로직은 또한 복수의 시스템 구성요소 중 특정 시스템 구성요소를 어드레싱하는 웨이크-경고 신호에 대해 웨이크-경고 채널을 모니터링한다.
- [0095] 적어도 하나의 예에서, 웨이크-경고 신호는 특정 시스템 구성요소를 식별한다.
- [0096] 하나 이상의 실시예는 제 1 하드웨어 구성요소와, 상호접속부 및 웨이크-경고 채널에 의해 제 1 하드웨어 구성요소에 접속된 제 2 하드웨어 구성요소를 포함하는 시스템을 제공할 수 있고, 제 2 하드웨어 구성요소는 상호접속부 및 웨이크-경고 채널에 대한 인터페이스, 및 로직을 포함하고, 로직은 웨이크-경고 채널을 통해 전송된 웨이크-경고 신호를 제 1 하드웨어 구성요소로부터 수신하고, 웨이크-경고 신호는 웨이크-경고 신호 다음의 인입하는 요청을 표시한다. 제 2 하드웨어 구성요소와 연관된 클록은 인입하는 요청의 수신 전에 클록을 인에이블하

도록 언케이팅될 수 있다. 인입하는 요청은 상호접속부를 통해 수신되어 프로세싱될 수 있다.

- [0097] 적어도 하나의 예에서, 제 1 하드웨어 구성요소는 특정 캐시 기록과 연관된 요청을 생성하기 위한 로직을 포함하고, 요청은 상호접속부를 통해 캐시 뱅크와 연관된 시스템 구성요소로 전송된다. 제 1 하드웨어 구성요소의 로직은 또한 웨이크-경고 신호를 생성하고, 전용 웨이크-경고 채널을 통해 웨이크-경고 신호를 제 2 하드웨어 구성요소로 전송하고, 상호접속부를 통해 요청을 제 2 하드웨어 구성요소로 전송할 수 있다.
- [0098] 적어도 하나의 예에서, 제 2 하드웨어 구성요소는 또한 특정 하드웨어 구성요소로 전송될 제 2 요청을 생성하고, 제 2 요청이 도착한다는 것을 특정 하드웨어 구성요소에 표시하기 위한 제 2 웨이크-경고 신호를 전용 웨이크-경고 채널을 통해 전송하고, 상호접속부를 통해 제 2 요청을 특정 하드웨어 구성요소로 전송한다. 제 2 웨이크-경고 신호는 특정 하드웨어 구성요소의 디스에이블 클럭이 인에이블 상태로 언케이팅되게 한다.
- [0099] 적어도 하나의 예에서, 제 1 하드웨어 구성요소는 제 1 프로세서 코어 및 제 1 캐시 뱅크를 포함하는 제 1 타일(tile)을 포함하고, 제 2 하드웨어 구성요소는 제 2 프로세서 코어 및 제 2 캐시 뱅크를 포함하는 제 2 타일을 포함한다.
- [0100] 적어도 하나의 예에서, 제 1 및 제 2 하드웨어 구성요소는 마이크로서버의 구성요소이다.
- [0101] 하나 이상의 실시예는 복수의 타일을 연결하기 위한 온-다이 상호접속부(on-die interconnect)를 포함하는 집적 회로를 포함하는 장치를 제공할 수 있고, 각각의 타일은 코어 및 캐시 에이전트를 포함한다. 복수의 타일 중 특정 타일은, 특정 타일과 연관된 비활동 표시자에 기초하여 특정 타일 중 적어도 일부에 대한 클럭을 게이팅하기로 결정하기 위한 검출 로직과, 검출 로직이 클럭을 게이팅하기로 결정하는 것에 응답하여 특정 타일의 적어도 일부에 대한 클럭을 게이팅하고, 수신된 조기의 웨이크 신호에 기초하여 적어도 요청의 도착 시간에 특정 타일의 적어도 일부에 대한 클럭을 언케이팅하기 위해, 검출 로직에 연결된 제어 로직을 포함한다.
- [0102] 적어도 하나의 예에서, 특정 타일 중 일부는 로컬 클럭 버퍼, 영역의 클럭 버퍼, 및 클럭 스파인(clock spine) 중 하나를 포함한다.
- [0103] 적어도 하나의 예에서, 온-다이 상호접속부는 링, 메시(mesh), 하이브리드 링-메시 상호접속 아키텍처 중 하나이다.
- [0104] 적어도 하나의 예에서, 비활동 표시자는 유향 및 앰프티 조건 카운터(idle and empty conditions counter) 중 적어도 하나를 포함한다.
- [0105] 적어도 하나의 예에서, 조기의 웨이크 신호는 웨이크 신호 링 상에서 수신된다.
- [0106] 본 명세서 전반에 걸쳐 "일 실시예" 또는 "실시예"에 대한 언급은 실시예와 관련하여 설명된 특정 특성, 구조, 또는 특징이 본 발명의 적어도 하나의 실시예에 포함된다는 것을 의미한다. 따라서, 본 명세서 전반에 걸쳐 다양한 장소들에서의 구들 "일 실시예에서" 또는 "실시예에서"의 출현들이 반드시 모두 동일한 실시예를 나타내는 것은 아니다. 또한, 특정 피쳐들, 구조들, 또는 특성들은 하나 이상의 실시예들에서 임의의 적절한 방식으로 조합될 수 있다.
- [0107] 상기의 명세서에서, 상세한 설명이 특정 예시적 실시예들을 참조하여 제공되었다. 그러나, 첨부된 청구항들에 설명된 바와 같은 본 발명의 넓은 사상 및 범위로부터 벗어남이 없이 다양한 변경들 및 변화들이 행해질 수 있다는 것이 분명할 것이다. 따라서, 명세서 및 도면들은 제한적인 의미라기보다는 설명적인 의미로서 간주되어야 한다. 더구나, 상기의 실시예 및 다른 예시적 언어의 사용이 반드시 동일한 실시예 또는 동일한 예를 나타내는 것이 아니라, 잠재적으로 동일한 실시예 뿐만 아니라, 상이하고 별개의 실시예들을 나타낼 수 있다.

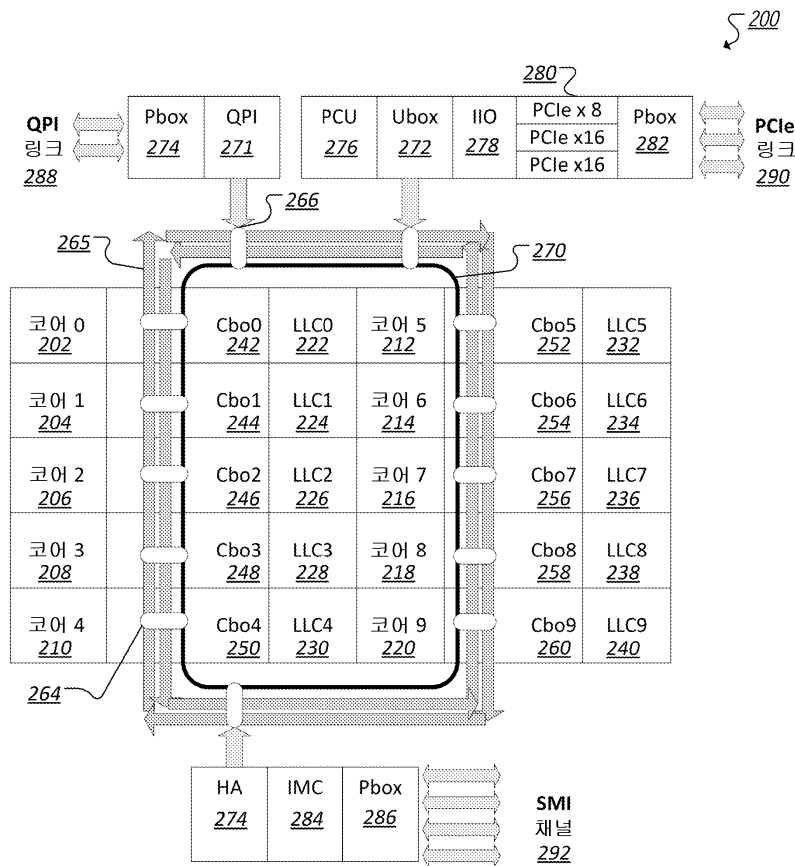
도면

도면1

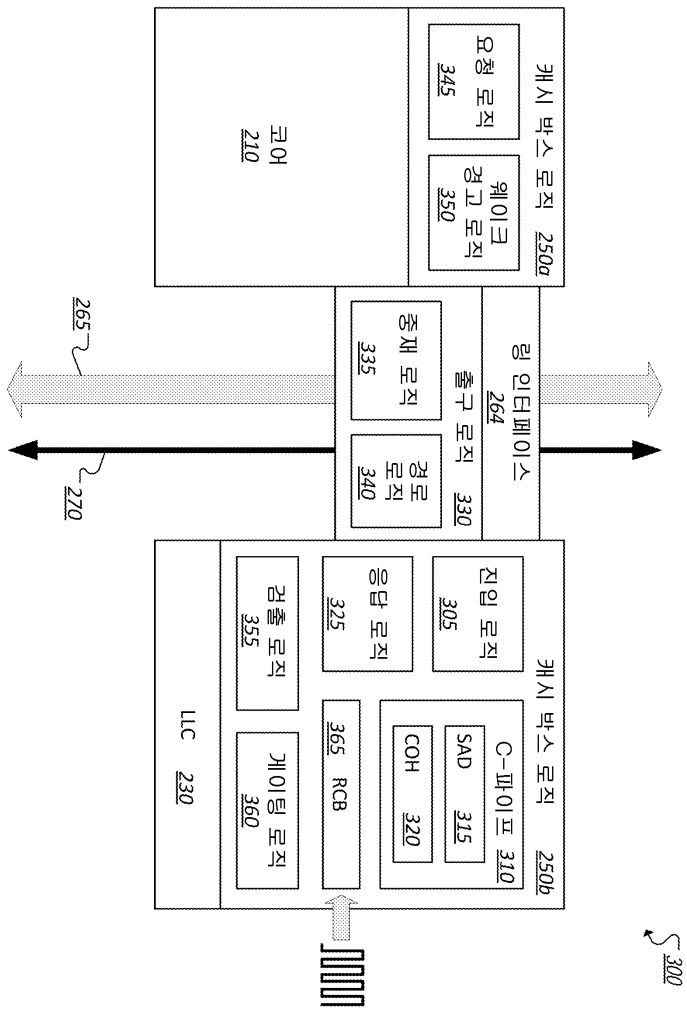




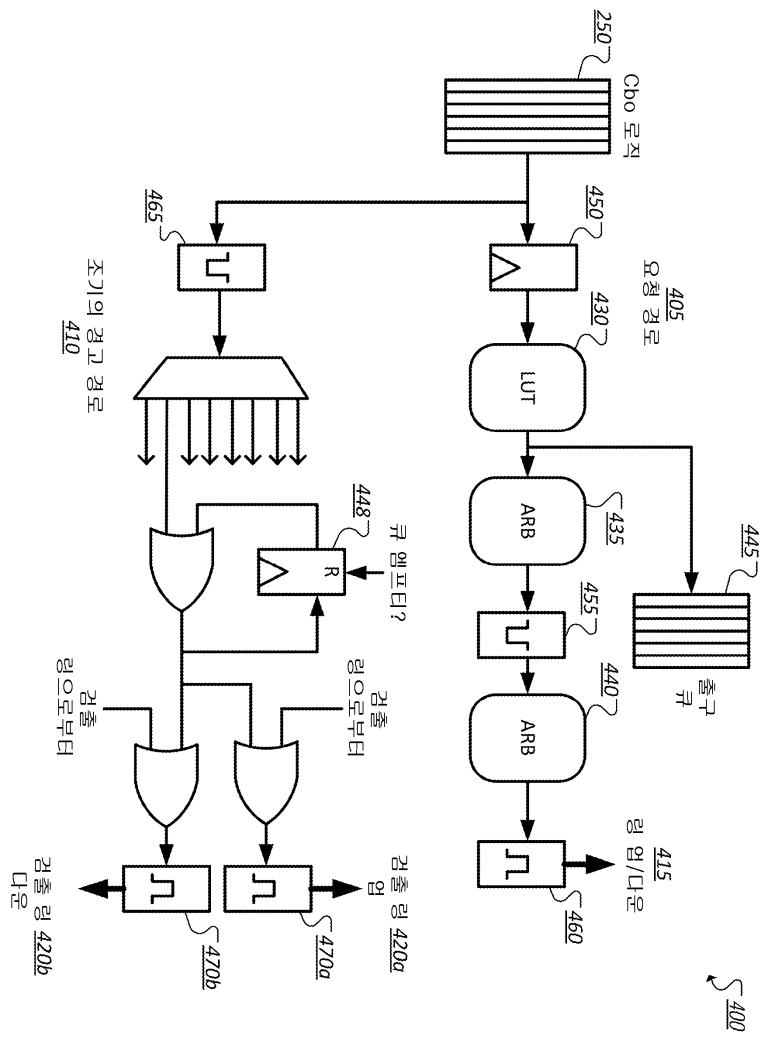
도면2



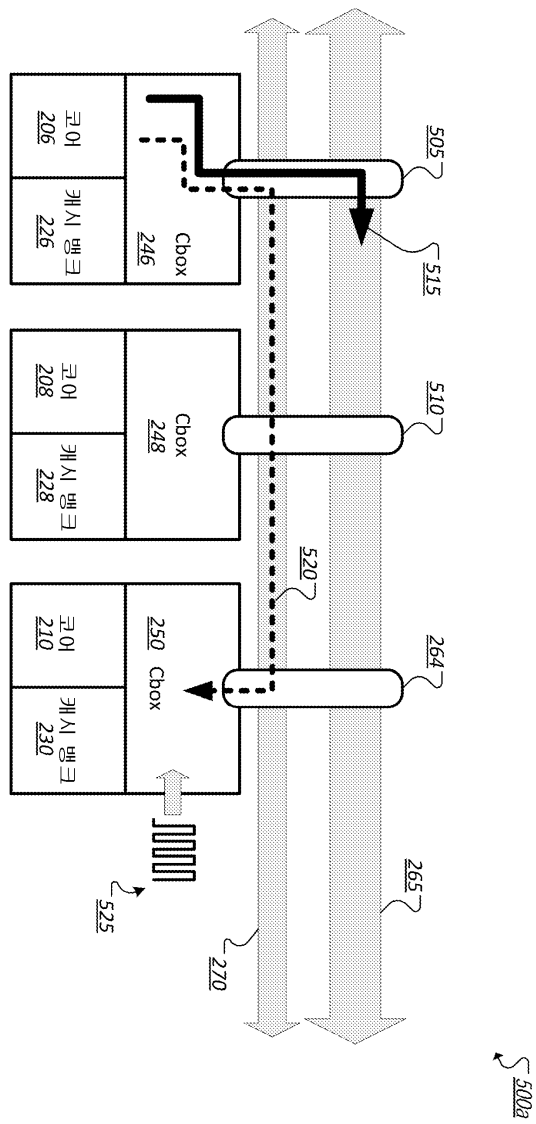
도면3



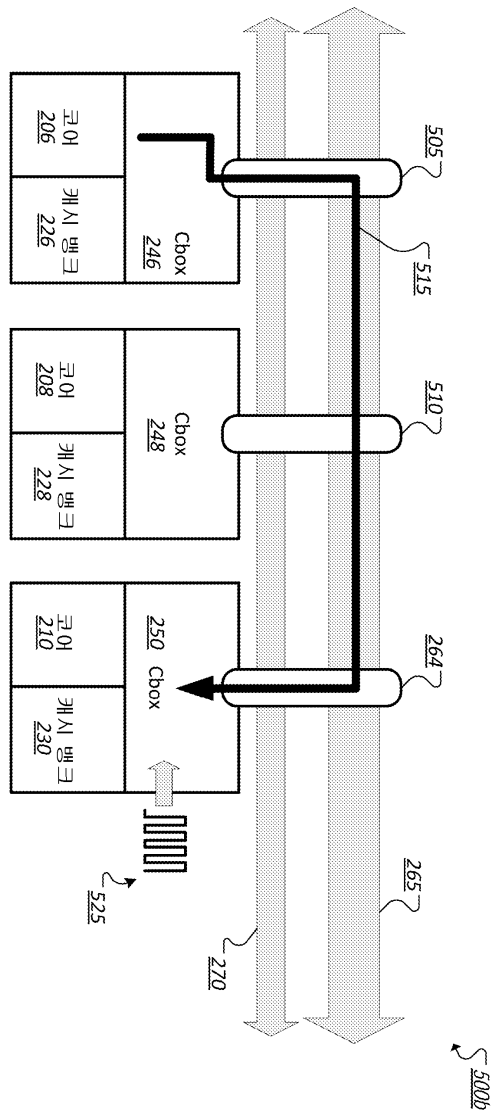
도면4



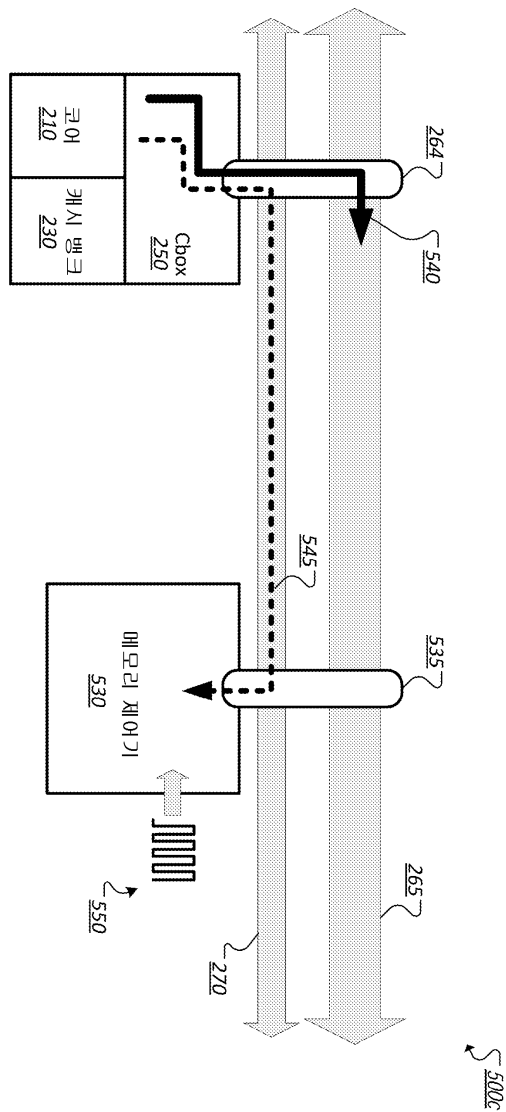
도면5a



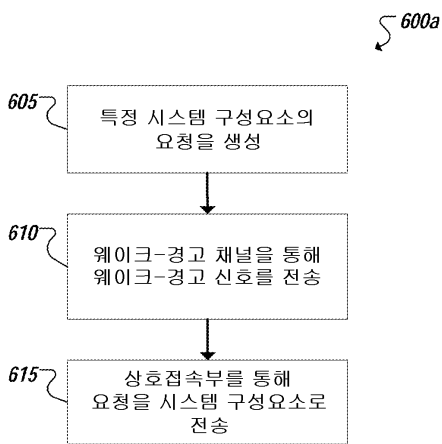
도면5b



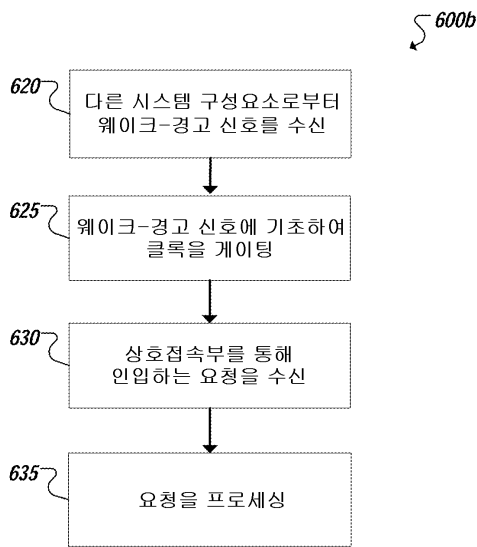
도면5c



도면6a



도면6b



도면7

