



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2019-0050988
(43) 공개일자 2019년05월14일

- | | |
|---|--|
| <p>(51) 국제특허분류(Int. Cl.)
G06F 12/0815 (2016.01) G06F 12/084 (2016.01)
G06F 12/0877 (2016.01) G06F 12/128 (2016.01)
G06F 9/50 (2018.01)</p> <p>(52) CPC특허분류
G06F 12/0815 (2013.01)
G06F 12/084 (2013.01)</p> <p>(21) 출원번호 10-2019-7007689</p> <p>(22) 출원일자(국제) 2017년08월22일
심사청구일자 없음</p> <p>(85) 번역문제출일자 2019년03월15일</p> <p>(86) 국제출원번호 PCT/US2017/047975</p> <p>(87) 국제공개번호 WO 2018/052669
국제공개일자 2018년03월22일</p> <p>(30) 우선권주장
15/268,768 2016년09월19일 미국(US)</p> | <p>(71) 출원인
헬컴 인코포레이티드
미국 92121-1714 캘리포니아주 샌 디에고 모어하우스 드라이브 5775</p> <p>(72) 발명자
터너 앤드류 에드먼드
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775
리칠릭 보후슬라프
미국 92121-1714 캘리포니아주 샌디에고 모어하우스 드라이브 5775</p> <p>(74) 대리인
특허법인코리아나</p> |
|---|--|

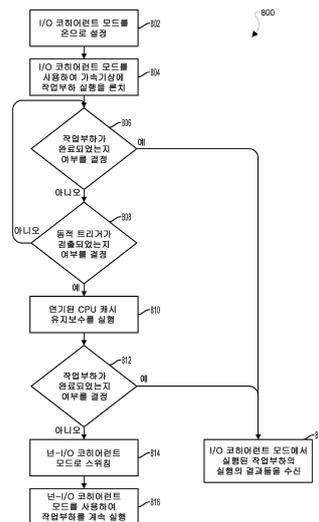
전체 청구항 수 : 총 26 항

(54) 발명의 명칭 동적 입력/출력 코히어런스

(57) 요약

양태들은 컴퓨팅 디바이스상에서 동적 입력/출력 (I/O) 코히어런트 작업부하 프로세싱을 구현하기 위한 디바이스들, 장치들 및 방법들을 포함한다. 양태 방법들은, 프로세싱 디바이스에 의해, I/O 코히어런트 모드를 사용하여 실행을 위해 하드웨어 가속기에 작업 부하를 오프로딩하는 단계, 하드웨어 가속기에 의해 작업 부하가 실행되는 동안 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 스위칭하기 위한 동적 트리거를 검출하는 단계, 및 하드웨어 가속기에 의해 작업 부하가 실행되는 동안 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 스위칭하는 단계를 포함한다.

대표도 - 도8



(52) CPC특허분류

G06F 12/0877 (2013.01)

G06F 12/128 (2013.01)

G06F 9/505 (2013.01)

G06F 9/5083 (2013.01)

G06F 2209/509 (2013.01)

G06F 2212/1041 (2013.01)

G06F 2212/621 (2013.01)

명세서

청구범위

청구항 1

컴퓨팅 디바이스에서 동적 입력/출력 (I/O) 코히어런트 작업부하 프로세싱을 구현하는 방법으로서, 프로세싱 디바이스에 의해, I/O 코히어런트 모드를 사용하여 실행을 위해 하드웨어 가속기로 작업부하를 오프로딩하는 단계;

상기 프로세싱 디바이스에 의해, 상기 하드웨어 가속기에 의해 상기 작업부하가 실행되는 동안 상기 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 스위칭하기 위한 동적 트리거를 검출하는 단계; 및

상기 동적 트리거를 검출하는 것에 응답하여 상기 하드웨어 가속기에 의해 상기 작업부하가 실행되는 동안 상기 I/O 코히어런트 모드에서 상기 비 I/O 코히어런트 모드로 스위칭하는 단계를 포함하는, 동적 I/O 코히어런트 작업부하 프로세싱을 구현하는 방법.

청구항 2

제 1 항에 있어서,

상기 하드웨어 가속기에 의한 상기 작업부하의 실행이 완료되었는지를 결정하는 단계를 더 포함하고,

상기 하드웨어 가속기에 의해 상기 작업부하가 실행되는 동안 상기 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 스위칭하기 위한 동적 트리거를 검출하는 단계는 상기 하드웨어 가속기에 의한 상기 작업부하의 실행이 완료되지 않았다고 결정하는 것에 응답하여 상기 I/O 코히어런트 모드에서 상기 비 I/O 코히어런트 모드로 스위칭하기 위한 상기 동적 트리거를 검출하는 단계를 포함하는, 동적 I/O 코히어런트 작업부하 프로세싱을 구현하는 방법.

청구항 3

제 1 항에 있어서,

상기 I/O 코히어런트 모드에서 상기 비 I/O 코히어런트 모드로 스위칭을 위한 상기 동적 트리거를 검출하는 것에 응답하여 상기 하드웨어 가속기에 의해 상기 작업부하가 실행되는 동안 상기 프로세싱 디바이스의 캐시에 대한 캐시 플러시 유지보수 동작을 실행하는 단계;

상기 하드웨어 가속기로부터, 상기 오프로드된 작업부하의 결과를 상기 I/O 코히어런트 모드에서 공유 메모리에 의해 수신하는 단계;

상기 하드웨어 가속기에 의한 상기 작업부하의 실행이 완료되었는지를 결정하는 단계; 및

상기 프로세싱 디바이스에 의해, 상기 하드웨어 가속기에 의한 상기 작업부하의 실행이 완료되었다는 결정에 응답하여 상기 프로세싱 디바이스의 캐시에 대한 캐시 무효화 유지보수 동작을 구현하지 않고 상기 오프로드된 작업부하의 결과를 수신하는 단계를 더 포함하는, 동적 I/O 코히어런트 작업부하 프로세싱을 구현하는 방법.

청구항 4

제 1 항에 있어서,

상기 하드웨어 가속기에 의한 상기 작업부하의 실행이 완료되었는지를 결정하는 단계로서, 상기 하드웨어 가속기에 의해 상기 작업부하가 실행되는 동안 상기 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 스위칭하는 단계는 상기 하드웨어 가속기에 의한 상기 작업부하의 실행이 완료되지 않았다고 결정하는 것에 응답하여 상기 I/O 코히어런트 모드에서 상기 비 I/O 코히어런트 모드로 스위칭하는 단계를 포함하는, 상기 결정하는 단계;

상기 하드웨어 가속기로부터, 상기 오프로드된 작업부하의 결과를 상기 비 I/O 코히어런트 모드에서 공유 메모리에 의해 수신하는 단계;

상기 하드웨어 가속기에 의한 상기 작업부하의 실행이 완료되었다고 결정하는 것에 응답하여 상기 프로세싱 디

바이스의 캐시에 대한 캐시 무효화 유지보수 동작을 실행하는 단계; 및

상기 프로세싱 디바이스에 의해, 상기 오프로드된 작업부하의 결과를 수신하는 단계를 더 포함하는, 동적 I/O 코히어런트 작업부하 프로세싱을 구현하는 방법.

청구항 5

제 4 항에 있어서,

상기 하드웨어 가속기에 의한 상기 작업부하의 실행이 완료되지 않았다고 결정하는 것에 응답하여 상기 하드웨어 가속기에 의해 상기 오프로드된 작업부하의 나머지 부분을 실행하는 단계를 더 포함하는, 동적 I/O 코히어런트 작업부하 프로세싱을 구현하는 방법.

청구항 6

제 1 항에 있어서,

상기 I/O 코히어런트 모드에서 비 I/O 코히어런트모드로 스위칭하기 위한 동적 트리거를 검출하는 단계는 경과 시간을 검출하는 단계, 유휴 상태에서 상기 프로세싱 디바이스를 검출하는 단계, 및 상기 오프로드된 작업부하의 실행의 서퍼링(suffering)을 검출하는 단계 중 적어도 하나를 포함하는, 동적 I/O 코히어런트 작업부하 프로세싱을 구현하는 방법.

청구항 7

제 6 항에 있어서,

상기 오프로드된 작업부하의 실행의 서퍼링을 검출하는 단계는 증대된 작업부하로부터 성능의 드라이버의 통지를 수신하는 단계, 작업부하 분류에 기초하여 정적 할당 또는 방법을 사용하는 단계, 서퍼링을 나타내는 데이터에 대해 하드웨어 성능 레지스터들을 폴링하는 단계, 최대값 근처의 상호연결 동작 레벨들을 측정하는 단계, 가속기 메모리 레이턴시를 측정하는 단계, 가속기 메모리 백프레서(backpressure)를 측정하는 단계, 및 하드웨어 가속기 유휴 상태를 검출하는 단계 중 적어도 하나를 포함하는, 동적 I/O 코히어런트 작업부하 프로세싱을 구현하는 방법.

청구항 8

컴퓨팅 디바이스로서,

공유 메모리;

하드웨어 가속기; 및

상기 공유 메모리 및 상기 하드웨어 가속기에 커플링된 프로세서 디바이스를 포함하고, 상기 프로세서 디바이스는,

I/O 코히어런트 모드를 사용하여 실행을 위해 상기 하드웨어 가속기로 작업부하를 오프로드하는 것;

상기 하드웨어 가속기에 의해 상기 작업부하가 실행되는 동안 상기 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 스위칭을 위한 동적 트리거를 검출하는 것; 및

상기 동적 트리거를 검출하는 것에 응답하여 상기 하드웨어 가속기에 의해 상기 작업부하가 실행되는 동안 상기 I/O 코히어런트 모드에서 상기 비 I/O 코히어런트 모드로 스위칭하는 것을 포함하는 동작들을 수행하도록 구성된, 컴퓨팅 디바이스.

청구항 9

제 8 항에 있어서,

상기 프로세싱 디바이스는 상기 하드웨어 가속기에 의한 상기 작업 부하의 실행이 완료되었는지 여부를 결정하는 것을 더 포함하는 동작들을 수행하도록 구성되고,

상기 프로세싱 디바이스는, 상기 하드웨어 가속기에 의해 상기 작업 부하가 실행되고 있는 동안 상기 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 스위칭하기 위한 동적 트리거를 검출하는 것이 상기 하드웨어 가속

기에 의한 상기 작업 부하의 실행이 완료되지 않았다고 결정하는 것에 응답하여 상기 I/O 코히어런트 모드에서 상기 비 I/O 코히어런트 모드로 스위칭하기 위한 상기 동적 트리거를 검출하는 것을 포함하도록 하는 동작들을 수행하도록 구성되는, 컴퓨팅 디바이스.

청구항 10

제 8 항에 있어서,

상기 프로세싱 디바이스는,

상기 I/O 코히어런트 모드에서 상기 비 I/O 코히어런트 모드로 스위칭을 위한 상기 동적 트리거를 검출하는 것에 응답하여 상기 하드웨어 가속기에 의해 상기 작업 부하가 실행되는 동안 상기 프로세싱 디바이스의 캐시에 대한 캐시 플러시 유지보수 동작을 실행하는 것;

상기 하드웨어 가속기로부터, 상기 오프로드된 작업 부하의 결과를 상기 I/O 코히어런트 모드에서 상기 공유 메모리에 의해 수신하는 것;

상기 하드웨어 가속기에 의한 상기 작업 부하의 실행이 완료되었는지를 결정하는 것; 및

상기 하드웨어 가속기에 의한 상기 작업 부하의 실행이 완료되었다고 결정하는 것에 응답하여 상기 프로세싱 디바이스의 캐시에 대한 캐시 무효화 유지보수 동작을 구현하지 않고 상기 오프로드된 작업 부하의 결과를 수신하는 것을 더 포함하는 동작들을 수행하도록 구성되는, 컴퓨팅 디바이스.

청구항 11

제 8 항에 있어서,

상기 프로세싱 디바이스는,

상기 하드웨어 가속기에 의한 상기 작업 부하의 실행이 완료되었는지를 결정하는 것으로서, 상기 하드웨어 가속기에 의해 상기 작업부하가 실행되는 동안 상기 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 스위칭하는 것은 상기 하드웨어 가속기에 의한 상기 작업 부하의 실행이 완료되지 않았다고 결정하는 것에 응답하여 상기 I/O 코히어런트 모드에서 상기 비 I/O 코히어런트 모드로 스위칭하는 것을 포함하는, 상기 결정하는 것;

상기 하드웨어 가속기로부터, 상기 오프로드된 작업 부하의 결과를 상기 비 I/O 코히어런트 모드에서 상기 공유 메모리에 의해 수신하는 것;

상기 하드웨어 가속기에 의한 상기 작업 부하의 실행이 완료되었다고 결정하는 것에 응답하여 상기 프로세싱 디바이스의 캐시에 대한 캐시 무효화 유지보수 동작을 실행하는 것; 및

상기 오프로드된 작업 부하의 결과를 수신하는 것을 더 포함하는 동작들을 수행하도록 구성되는, 컴퓨팅 디바이스.

청구항 12

제 8 항에 있어서,

상기 프로세싱 디바이스는,

상기 I/O 코히어런트 모드에서 비 I/O 코히어런트모드로 스위칭하기 위한 동적 트리거를 검출하는 것이 경과 시간을 검출하는 것, 유희 상태에 있는 상기 프로세싱 디바이스를 검출하는 것, 및 상기 오프로드된 작업부하의 실행의 서퍼링(suffering)을 검출하는 것 중 적어도 하나를 포함하도록 하는 동작들을 수행하도록 구성되는, 컴퓨팅 디바이스.

청구항 13

제 12 항에 있어서,

상기 프로세싱 디바이스는,

상기 오프로드된 작업 부하의 실행의 서퍼링을 검출하는 것이 증대된 작업 부하로부터 성능의 드라이버의 통지를 수신하는 것, 작업부하 분류에 기초하여 정적 할당 또는 방법을 사용하는 것, 서퍼링을 나타내는 데이터에

대해 하드웨어 성능 레지스터들을 폴링하는 것, 최대값 근처의 상호연결 동작 레벨들을 측정하는 것, 가속기 메모리 레이턴시를 측정하는 것, 가속기 메모리 백프레서 (backpressure) 를 측정하는 것, 및 하드웨어 가속기 유휴 상태를 검출하는 것 중 적어도 하나를 포함하도록 하는 동작들을 수행하도록 구성되는, 컴퓨팅 디바이스.

청구항 14

컴퓨팅 디바이스로서,

하드웨어 가속기;

I/O 코히어런트 모드를 사용하여 실행을 위해 상기 하드웨어 가속기로 작업 부하를 오프로딩하는 수단;

상기 하드웨어 가속기에 의해 상기 작업 부하가 실행되는 동안 상기 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 스위칭을 위한 동적 트리거를 검출하는 수단; 및

상기 동적 트리거를 검출하는 것에 응답하여 상기 하드웨어 가속기에 의해 상기 작업 부하가 실행되는 동안 상기 I/O 코히어런트 모드에서 상기 비 I/O 코히어런트 모드로 스위칭하는 수단을 포함하는, 컴퓨팅 디바이스.

청구항 15

제 14 항에 있어서,

상기 하드웨어 가속기에 의한 상기 작업 부하의 실행이 완료되었는지 여부를 결정하는 수단을 더 포함하고,

상기 하드웨어 가속기에 의해 상기 작업 부하가 실행되는 동안 상기 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 스위칭하기 위한 동적 트리거를 검출하는 수단은 상기 하드웨어 가속기에 의한 상기 작업 부하의 실행이 완료되지 않았다고 결정하는 것에 응답하여 상기 I/O 코히어런트 모드에서 상기 비 I/O 코히어런트 모드로 스위칭하기 위한 상기 동적 트리거를 검출하는 수단을 포함하는, 컴퓨팅 디바이스.

청구항 16

제 14 항에 있어서,

상기 I/O 코히어런트 모드에서 상기 비 I/O 코히어런트 모드로 스위칭을 위한 상기 동적 트리거를 검출하는 것에 응답하여 상기 하드웨어 가속기에 의해 상기 작업 부하가 실행되는 동안 캐시에 대한 캐시 플러시 유지보수 동작을 실행하는 수단;

상기 하드웨어 가속기로부터, 상기 오프로드된 작업 부하의 결과를 상기 I/O 코히어런트 모드에서 공유 메모리에 의해 수신하는 수단;

상기 하드웨어 가속기에 의한 상기 작업 부하의 실행이 완료되었는지를 결정하는 수단; 및

상기 하드웨어 가속기에 의한 상기 작업 부하의 실행이 완료되었다고 결정하는 것에 응답하여 상기 캐시에 대한 캐시 무효화 유지보수 동작을 구현하지 않고 상기 오프로드된 작업 부하의 결과를 수신하는 수단을 더 포함하는, 컴퓨팅 디바이스.

청구항 17

제 14 항에 있어서,

상기 하드웨어 가속기에 의한 상기 작업 부하의 실행이 완료되었는지를 결정하는 수단으로서, 상기 하드웨어 가속기에 의해 상기 작업부하가 실행되는 동안 상기 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 스위칭하는 것은 상기 하드웨어 가속기에 의한 상기 작업 부하의 실행이 완료되지 않았다고 결정하는 것에 응답하여 상기 I/O 코히어런트 모드에서 상기 비 I/O 코히어런트 모드로 스위칭하는 것을 포함하는, 상기 결정하는 수단;

상기 하드웨어 가속기로부터, 상기 오프로드된 작업 부하의 결과를 상기 비 I/O 코히어런트 모드에서 공유 메모리에 의해 수신하는 수단;

상기 하드웨어 가속기에 의한 상기 작업 부하의 실행이 완료되었다고 결정하는 것에 응답하여 캐시에 대한 캐시 무효화 유지보수 동작을 실행하는 수단; 및

상기 오프로드된 작업 부하의 결과를 수신하는 수단을 더 포함하는, 컴퓨팅 디바이스.

청구항 18

제 17 항에 있어서,

상기 하드웨어 가속기에 의한 상기 작업 부하의 실행이 완료되지 않았다고 결정하는 것에 응답하여 상기 하드웨어 가속기에 의해 상기 오프로드된 작업 부하의 나머지 부분을 실행하는 수단을 더 포함하는, 컴퓨팅 디바이스.

청구항 19

제 14 항에 있어서,

상기 I/O 코히어런트 모드에서 비 I/O 코히어런트모드로 스위칭하기 위한 동적 트리거를 검출하는 수단은 경과 시간을 검출하는 수단, 프로세싱 디바이스가 유희 상태에 있는 것을 검출하는 수단, 및 상기 오프로드된 작업부하의 실행의 서퍼링 (suffering) 을 검출하는 수단 중 적어도 하나를 포함하는, 컴퓨팅 디바이스.

청구항 20

제 19 항에 있어서,

상기 오프로드된 작업 부하의 실행의 서퍼링을 검출하는 수단은 증대된 작업 부하로부터 성능의 드라이버의 통지를 수신하는 수단, 작업부하 분류에 기초하여 정적 할당 또는 방법을 사용하는 수단, 서퍼링을 나타내는 데이터에 대해 하드웨어 성능 레지스터들을 폴링하는 수단, 최대값 근처의 상호연결 동작 레벨들을 측정하는 수단, 가속기 메모리 레이턴시를 측정하는 수단, 가속기 메모리 백프레서 (backpressure) 를 측정하는 수단, 및 하드웨어 가속기 유희 상태를 검출하는 수단 중 적어도 하나를 포함하는, 컴퓨팅 디바이스.

청구항 21

프로세서 실행 가능 명령들을 저장한 비밀시적 프로세서 관독 가능 저장 매체로서,

상기 명령들은, 컴퓨팅 디바이스의 프로세싱 디바이스로 하여금 동작들을 수행하게 하도록 구성되고,

상기 동작들은,

I/O 코히어런트 모드를 사용하여 실행을 위해 하드웨어 가속기로 작업 부하를 오프로딩하는 것;

상기 하드웨어 가속기에 의해 상기 작업 부하가 실행되는 동안 상기 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 스위칭하기 위한 동적 트리거를 검출하는 것; 및

상기 동적 트리거를 검출하는 것에 응답하여 상기 하드웨어 가속기에 의해 상기 작업 부하가 실행되는 동안 상기 I/O 코히어런트 모드에서 상기 비 I/O 코히어런트 모드로 스위칭하는 것을 포함하는, 비밀시적 프로세서 관독 가능 저장 매체.

청구항 22

제 21 항에 있어서,

저장된 상기 실행 가능 명령들은, 상기 컴퓨팅 디바이스의 상기 프로세싱 디바이스로 하여금, 상기 하드웨어 가속기에 의한 상기 작업부하의 실행이 완료되었는지 여부를 결정하는 것을 더 포함하는 동작들을 수행하게 하도록 구성되고,

상기 저장된 실행가능 명령들은, 상기 컴퓨팅 디바이스의 상기 프로세싱 디바이스로 하여금, 상기 하드웨어 가속기에 의해 상기 작업 부하가 실행되는 동안 상기 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 스위칭하기 위한 동적 트리거를 검출하는 것이 상기 하드웨어 가속기에 의한 상기 작업 부하의 실행이 완료되지 않았다고 결정하는 것에 응답하여 상기 I/O 코히어런트 모드에서 상기 비 I/O 코히어런트 모드로 스위칭하기 위한 상기 동적 트리거를 검출하는 것을 포함하도록 하는 동작들을 수행하게 하도록 구성되는, 비밀시적 프로세서 관독 가능 저장 매체.

청구항 23

제 21 항에 있어서,

저장된 상기 실행 가능 명령들은, 상기 컴퓨팅 디바이스의 상기 프로세싱 디바이스로 하여금, 동작들을 수행하

게 하도록 구성되고, 상기 동작들은,

상기 I/O 코히어런트 모드에서 상기 비 I/O 코히어런트 모드로 스위칭 위한 상기 동적 트리거를 검출하는 것에 응답하여 상기 하드웨어 가속기에 의해 상기 작업 부하가 실행되는 동안 상기 프로세싱 디바이스의 캐시에 대한 캐시 플러시 유지보수 동작을 실행하는 것;

상기 하드웨어 가속기로부터, 상기 오프로드된 작업 부하의 결과를 상기 I/O 코히어런트 모드에서 공유 메모리에 의해 수신하는 것;

상기 하드웨어 가속기에 의한 상기 작업 부하의 실행이 완료되었는지를 결정하는 것; 및

상기 하드웨어 가속기에 의한 상기 작업 부하의 실행이 완료되었다고 결정하는 것에 응답하여 상기 프로세싱 디바이스의 캐시에 대한 캐시 무효화 유지보수 동작을 구현하지 않고 상기 오프로드된 작업 부하의 결과를 수신하는 것을 더 포함하는, 비일시적 프로세서 판독 가능 저장 매체.

청구항 24

제 21 항에 있어서,

저장된 상기 실행 가능 명령들은, 상기 컴퓨팅 디바이스의 상기 프로세싱 디바이스로 하여금, 동작들을 수행하게 하도록 구성되고, 상기 동작들은,

상기 하드웨어 가속기에 의한 상기 작업 부하의 실행이 완료되었는지를 결정하는 것으로서, 상기 하드웨어 가속기에 의해 상기 작업부하가 실행되는 동안 상기 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 스위칭하는 것은 상기 하드웨어 가속기에 의한 상기 작업 부하의 실행이 완료되지 않았다고 결정하는 것에 응답하여 상기 I/O 코히어런트 모드에서 상기 비 I/O 코히어런트 모드로 스위칭하는 것을 포함하는, 상기 결정하는 것;

상기 하드웨어 가속기로부터, 상기 오프로드된 작업 부하의 결과를 상기 비 I/O 코히어런트 모드에서 공유 메모리에 의해 수신하는 것;

상기 하드웨어 가속기에 의한 상기 작업 부하의 실행이 완료되었다고 결정하는 것에 응답하여 상기 프로세싱 디바이스의 캐시에 대한 캐시 무효화 유지보수 동작을 실행하는 것; 및

상기 오프로드된 작업 부하의 결과를 수신하는 것을 더 포함하는, 비일시적 프로세서 판독 가능 저장 매체.

청구항 25

제 21 항에 있어서,

저장된 상기 실행가능 명령들은, 상기 컴퓨팅 디바이스의 상기 프로세싱 디바이스로 하여금, 상기 I/O 코히어런트 모드에서 비 I/O 코히어런트모드로 스위칭하기 위한 동적 트리거를 검출하는 것이 경과 시간을 검출하는 것, 유틸 상태에 있는 상기 프로세싱 디바이스를 검출하는 것, 및 상기 오프로드된 작업부하의 실행의 서퍼링(suffering) 을 검출하는 것 중 적어도 하나를 포함하도록 하는 동작들을 수행하게 하도록 구성되는, 비일시적 프로세서 판독 가능 저장 매체.

청구항 26

제 25 항에 있어서,

저장된 상기 실행가능 명령들은, 상기 컴퓨팅 디바이스의 상기 프로세싱 디바이스로 하여금, 상기 오프로드된 작업 부하의 실행의 서퍼링을 검출하는 것이 증대된 작업 부하로부터 성능의 드라이버의 통지를 수신하는 것, 작업부하 분류에 기초하여 정적 할당 또는 방법을 사용하는 것, 서퍼링을 나타내는 데이터에 대해 하드웨어 성능 레지스터들을 폴링하는 것, 최대값 근처의 상호연결 동작 레벨들을 측정하는 것, 가속기 메모리 레이턴시를 측정하는 것, 가속기 메모리 백프레셔(backpressure) 를 측정하는 것, 및 하드웨어 가속기 유틸 상태를 검출하는 것 중 적어도 하나를 포함하도록 하는 동작들을 수행하게 하도록 구성되는, 비일시적 프로세서 판독 가능 저장 매체.

발명의 설명

기술분야

배경기술

- [0001] 하드웨어 가속기는 중앙 처리 장치 (CPU) 가 작업 부하를 프로세싱하는 것을 돕는 데 사용될 수 있다. 작업 부하는 종종 CPU 캐시로부터 데이터를 사용하는 것을 요구한다. 가속기에 CPU 캐시로부터 데이터를 제공하기 위해, 캐시 유지보수 동작이 구현될 필요가 있다. 그러나 캐시 유지보수 동작은 가속기에 오프로드된 작업 부하를 프로세싱하는 성능에 부정적인 영향을 미칠 수 있다. 수동 캐시 유지보수 동작 실행은 일반적으로 작업을 가속기에 오프로딩하는 데 너무 오래 걸린다. 가속기 실행 전에 캐시 플러시 유지보수 동작이 구현되어야 하며 가속기 실행 후 캐시 무효화 유지보수 동작이 구현되어야 한다. 가속기에 대한 작은 작업부하 오프로드는 프로세싱 성능을 향상시키는 데 점점 더 중요하다. 이러한 작은 작업 부하 오프로드는 캐시 유지보수 페널티로부터 가장 많은 성능 저하를 겪을 수 있다.
- [0002] 입력/출력 (I/O) 코히어런시는 값 비싼 캐시 유지보수 절차를 구현하지 않고도 작업을 가속기에 오프로드하는 데 사용될 수 있다. 그러나 작업 부하를 오프로드하기 위한 I/O 코히어런트 경로는 낮은 성능의 신호 전송 (예, 낮은 대역폭)으로 인해 오버 헤드를 유발한다. I/O 코히어런스 페널티는 I/O 코히어런트 경로의 다양한 팩터로 인해 오프로드된 작업 부하 프로세싱에 부정적인 영향을 줄 수 있다.

발명의 내용

해결하려는 과제

과제의 해결 수단

- [0003] 다양한 양태는 컴퓨팅 디바이스상에서 동적 입력/출력 (I/O) 코히어런트 작업부하 프로세싱을 구현하기 위한 장치 및 방법을 포함할 수도 있다. 다양한 양태는 프로세싱 디바이스가 I/O 코히어런트 모드를 사용하여 실행을 위해 하드웨어 가속기에 작업 부하를 오프로딩하는 것, 하드웨어 가속기에 의해 작업 부하가 실행되는 동안 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 스위칭하기 위한 동적 트리거를 검출하는 것, 동적 트리거를 검출하는 것에 응답하여 하드웨어 가속기에 의해 작업 부하가 실행되는 동안 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 스위칭하는 것을 포함한다.
- [0004] 일부 양태는 하드웨어 가속기에 의한 작업 부하의 실행이 완료되었는지를 결정하는 것을 더 포함할 수도 있고, 여기서 하드웨어 가속기에 의해 작업부하가 실행되는 동안 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 스위칭하기 위한 동적 트리거를 검출하는 것은 하드웨어 가속기에 의한 작업 부하의 실행이 완료되지 않았다고 결정하는 것에 응답하여 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 스위칭하기 위한 동적 트리거를 검출하는 것을 포함할 수도 있다.
- [0005] 일부 양태는, I/O 코히어런트 모드로부터 비 I/O 코히어런트 모드로 스위칭하기 위한 동적 트리거의 검출에 응답하여 하드웨어 가속기에 의해 작업 부하가 실행되는 동안 프로세싱 디바이스의 캐시에 대한 캐시 플러시 유지보수 동작을 실행하는 것, 하드웨어 가속기로부터 I/O 코히어런트 모드에서 상기 오프로드된 작업 부하의 결과를 공유 메모리에 의해 수신하는 것, 하드웨어 가속기에 의한 작업 부하의 실행이 완료되었는지를 결정하는 것, 하드웨어 가속기에 의한 상기 작업 부하의 실행이 완료되었다는 결정에 응답하여 프로세싱 디바이스의 캐시에 대한 캐시 무효화 유지보수 동작을 구현하지 않고 오프로드된 작업부하의 결과를 수신하는 것을 더 포함할 수도 있다.
- [0006] 일부 양태는 하드웨어 가속기에 의한 작업 부하의 실행이 완료되었는지 여부를 결정하는 것을 더 포함하며, 여기서 하드웨어 가속기에 의해 작업 부하가 실행되는 동안 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 스위칭하는 것은 하드웨어 가속기에 의한 작업 부하의 실행이 완료되지 않은 것으로 결정하는 것에 응답하여 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 스위칭하는 것, 오프로드된 작업 부하의 결과를 하드웨어 가속기로부터 공유 메모리에 의해 비 I/O 코히어런트 모드에서 수신하는 것, 하드웨어 가속기에 의한 작업 부하의 실행이 완료되었다는 결정에 응답하여 프로세싱 디바이스의 캐시에 대한 캐시 무효화 유지보수 동작을 실행하는 것, 및 오프로드 된 작업 부하의 결과를 수신하는 것을 더 포함한다. 이러한 양태들은 하드웨어 가속기에

의한 작업 부하의 실행이 완료되지 않았다고 결정한 것에 응답하여 하드웨어 가속기에 의해 오프로드된 작업 부하의 나머지 부분을 실행하는 것을 더 포함할 수도 있다.

[0007] 일부 양태들에서, I/O 코히어런트 모드에서 비 I/O 코히어런트모드로 스위칭하기 위한 동적 트리거를 검출하는 것은 경과 시간을 검출하는 것, 유틸리티 상태에 있는 프로세싱 디바이스를 검출하는 것, 오프로드된 작업부하의 실행의 서퍼링 (suffering) 을 검출하는 것 중 적어도 하나를 포함할 수도 있다. 몇몇 양태들에서, 오프로드된 작업 부하의 실행의 서퍼링을 검출하는 것은 증대된 작업 부하로부터 성능의 드라이버의 통지를 수신하는 것, 작업부하 분류에 기초하여 정적 할당 또는 방법을 사용하는 것, 서퍼링을 나타내는 데이터에 대해 하드웨어 성능 레지스터들을 폴링하는 것, 최대값 근처의 상호연결 동작 레벨들을 측정하는 것, 가속기 메모리 레이턴시를 측정하는 것, 가속기 메모리 백프레서 (backpressure) 를 측정하는 것, 및 하드웨어 가속기 유틸리티 상태를 검출하는 것 중 적어도 하나를 포함할 수도 있다.

[0008] 다른 양태는 메모리, 하드웨어 가속기 및 상기 요약된 방법의 동작을 수행하도록 구성된 프로세서를 갖는 컴퓨팅 디바이스를 포함한다. 다른 양태는 상기 요약된 방법의 기능을 수행하기 위한 수단을 갖는 컴퓨팅 디바이스를 포함한다. 또 다른 양태는 컴퓨팅 디바이스의 프로세싱 디바이스로 하여금 상기 요약된 방법의 동작을 수행하게 하도록 구성된 프로세서 실행 가능 명령들이 저장된 비일시적 프로세서 판독 가능 저장 매체를 포함한다.

도면의 간단한 설명

[0009] 본 명세서에 포함되고 본 명세서의 일부를 구성하는 첨부 도면은 다양한 양태의 예시적인 양태를 도시하고, 상기에 제공된 일반적인 설명 및 이하의 상세한 설명과 함께 청구 범위의 특징을 설명하는 역할을 한다.

도 1 은 일 양태를 구현하는데 적합한 컴퓨팅 디바이스를 나타내는 컴포넌트 블록도이다.

도 2 는 일 양태를 구현하는데 적합한 예시의 멀티코어 프로세서를 나타내는 컴포넌트 블록도이다.

도 3 은 일 양태를 구현하기에 적합한 코히어런트 상호연결을 갖는 예시적인 이종 컴퓨팅 디바이스를 나타내는 블록도이다.

도 4는 일 양태를 구현하기 위한 비 I/O 코히어런트 (coherent) 작업부하 프로세싱의 일례를 나타내는 신호 및 프로세스 다이어그램이다.

도 5는 일 양태를 구현하기 위한 I/O 코히어런트 작업부하 프로세싱의 일례를 나타내는 신호 및 프로세스 다이어그램이다.

도 6는 일 양태를 구현하기 위한 동적 I/O 코히어런트 작업부하 프로세싱의 일례를 나타내는 신호 및 프로세스 다이어그램이다.

도 7 은 일 양태를 구현하기 위한 동적 I/O 코히어런트 작업부하 프로세싱의 일례를 나타내는 신호 및 프로세스 다이어그램이다.

도 8은 일 양태에 따라 동적 I/O 코히어런트 작업부하 프로세싱을 구현하는 방법을 나타내는 프로세스 흐름도이다.

도 9은 일 양태에 따라 동적 I/O 코히어런트 작업부하 프로세싱을 구현하는 방법을 나타내는 프로세스 흐름도이다.

도 10 은 여러 양태와 함께 사용하는 데 적합한 예시의 모바일 컴퓨팅 디바이스를 나타내는 컴포넌트 블록도이다.

도 11 은 여러 양태와 함께 사용하는 데 적합한 예시의 모바일 컴퓨팅 디바이스를 나타내는 컴포넌트 블록도이다.

도 12 은 여러 양태와 함께 사용하는 데 적합한 예시의 서버를 나타내는 컴포넌트 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0010] 여러 양태들이 첨부된 도면을 참조하여 상세히 설명될 것이다. 가능하다면, 도면 전체에 걸쳐 동일한 참조 번호가 동일하거나 유사한 부분을 나타내기 위해 사용될 것이다. 특정 예 및 구현 예에 대한 참조는 설명의 목적

을 위한 것이며 청구항의 범위를 제한하려는 것이 아니다.

- [0011] 다양한 양태는 하드웨어 가속기에 의한 실행을 위해 작업 부하를 오프 로딩하기 위한 오버 헤드 패널티를 감소시키기 위해 입력/출력 (I/O) 코히어런트 및 비 I/O 코히어런트 모드들 간의 동적 스위칭을 구현하는 방법, 시스템 및 그러한 방법을 구현하는 디바이스를 포함 할 수도 있다. 다중 프로세서들은 멀티 코어 프로세서들일 수도 있다. 다양한 양태의 장치 및 방법은 작업 부하를 I/O 코히어런트 모드에서 가속기로 오프로딩하는 것, 긴 작업 부하의 조건을 검출하는 것, 캐시 유지 보수 동작을 구현하는 것, 및 비 I/O 코히어런트 모드로 스위칭하는 것을 포함할 수도 있다.
- [0012] "컴퓨팅 디바이스" 및 "모바일 컴퓨팅 디바이스"라는 용어는 셀룰러 전화, 스마트 폰, 개인 또는 모바일 멀티미디어 플레이어, 개인 휴대 정보 단말기 (PDA), 랩톱 컴퓨터, 태블릿 컴퓨터, 변환가능 랩톱/태블릿 (2-in-1 컴퓨터), 스마트북, 울트라 북, 넷북, 팜톱 컴퓨터, 무선 전자 메일 수신기, 멀티미디어 인터넷 가능 휴대 전화, 모바일 게임 콘솔, 무선 게임 컨트롤러 및 메모리, 및 프로그램 가능 프로세서를 포함하는 유사한 개인용 전자 디바이스 중 임의의 것 또는 전부를 지칭하기 위해 여기서 교환가능하게 사용된다. "컴퓨팅 디바이스"라는 용어는 개인용 컴퓨터, 데스크탑 컴퓨터, 울인원 컴퓨터, 워크 스테이션, 슈퍼 컴퓨터, 메인 프레임 컴퓨터, 임베디드 컴퓨터, 서버, 홈 시어터 컴퓨터 및 게임 콘솔을 포함하는 고정식 컴퓨팅 디바이스를 추가로 지칭 할 수 있다.
- [0013] 동적 I/O 코히어런시는 작업부하를 가속기로 오프로드하기 위한 캐시 유지보수 동작의 회피 또는 연기를 허용할 수도 있다. 이러한 캐시 유지보수 동작의 회피 또는 연기는 I/O 코히어런트 및 비 I/O 코히어런트 모드에서 작업부하를 오프로딩하는 것과 관련된 오버 헤드를 감소시킬 수도 있다. 중앙 처리 장치 (CPU)와 같은 프로세싱 디바이스의 작업부하를 프로세싱하는 것을 돕기 위해 가속기에서 오프로드된 작업부하가 론칭될 수도 있다. 프로세싱 디바이스는 프로세싱을 위해 가속기에 그의 작업부하의 일부를 오프로드할 수도 있다. 가속기는 오프로드된 작업부하를 실행할 때 프로세싱 디바이스보다 효율적이고 및/또는 더 빠를 수도 있다. 가속기는 프로세싱 디바이스에 의한 나머지 작업부하의 실행과 병렬로 오프로드된 작업부하를 실행할 수 있을 수도 있다. 코히어런트 상호연결에 통신 가능하게 연결된 프로세싱 디바이스 용 캐시 및/또는 공유 메모리는 초기에 I/O 코히어런트 모드로 설정될 수도 있다.
- [0014] I/O 코히어런트 모드에서, 코히어런시 메커니즘은 프로세싱 디바이스 캐시 및/또는 공유 메모리에서의 데이터의 정확성을 보장하기 위해 연속적으로 구현될 수도 있다. 이들 코히어런시 메커니즘으로 인해, 캐시 유지보수 동작들은 프로세싱 디바이스 캐시 및/또는 공유 메모리에서의 데이터의 정확성을 보장하기 위해 필요로 되지 않을 수도 있다. 그러나, I/O 코히어런트 모드는 I/O 코히어런트 경로의 변수에 의해 영향을 받을 수 있는 오버 헤드 패널티를 초래할 수 있다. 예를 들어 I/O 코히어런트 모드 오버 헤드 패널티는 작업부하에 따라 증가할 수도 있다.
- [0015] 작업부하가 I/O 코히어런트 모드의 증가하는 오버 헤드 패널티를 겪고 있는 것을 검출하기 전에 오프로드된 작업부하가 완료되면 I/O 코히어런트 모드 오버헤드 패널티가 낮게 유지 될 수도 있고 캐시 유지보수 동작 오버헤드가 발생되지 않을 수도 있다. 작업부하가 I/O 코히어런트 모드 오버 헤드 패널티가 허용 한계를 초과하고 있을 수도 있는 크기까지 증가하고 있는 시기를 결정하기 위해 특정 조건이 사용될 수 있다. 그러한 조건은 계산된 또는 미리 설정된 경과 시간, 유희 상태에 있는 CPU, 및/또는 I/O 코히어런트 경로를 사용하는 데 어려움이 있다는 표시를 나타내는 작업부하를 포함할 수도 있다. 작업부하 서퍼링의 검출은 다양한 메트릭을 기반으로 할 수 있다. 증대된 작업부하는 드라이버에게 성능을 알릴 수도 있다. 테스트 또는 무작위 테스트를 통한 작업부하 분류에 기초한 정적 할당 또는 방법의 사용은 표준 시간 초과나 작업부하 서퍼링을 나타내는 신호를 야기할 수도 있다. 하드웨어 성능 레지스터를 폴링하는 것은 작업부하 서퍼링을 나타내는 데이터를 제공할 수도 있다. 대역폭 및/또는 미해결 트랜잭션과 같이 최대값 근처의 지속적으로 높은 상호연결 동작 레벨은 스로틀링을 나타낼 수도 있다. 높은 가속기 메모리 레이턴시는 레이턴시 제한을 나타낼 수도 있다. 높은 가속기 메모리 백프레셔는 대역폭 스로틀링을 나타낼 수도 있다. 높은 점유 메모리 요청 대기열과 결합한 메모리로 인한 높은 가속기 유희 상태는 대역폭이 제한되는 것을 나타낼 수도 있다.
- [0016] 프로세서 또는 하드웨어가 오프로드된 작업부하가 I/O 코히어런트 모드의 증가하는 오버 헤드 패널티를 겪고 있을 수도 있다는 것을 검출하면, 프로세싱 디바이스 캐시 및/또는 공유 메모리는 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 동적으로 변화할 수도 있다. 비 I/O 코히어런트 모드로 동적으로 스위칭하기 전에, 그리고 여전히 I/O 코히어런트 모드에 있는 동안, 프로세싱 디바이스 캐시에 대해 캐시 플러시 유지보수 동작이 구현될 수도 있다. 여전히 I/O 코히어런트 모드에 있는 동안 캐시 플러시 동작을 구현하는 것은 가속기가 캐시

플러시 유지보수 동작에 대한 비 I/O 코히어런트 오버헤드 패널티를 발생시키지 않고 오프로드된 작업부하를 계속 실행하는 것을 허용할 수도 있다. 프로세싱 디바이스 캐시 및/또는 공유 메모리는 캐시 플러시 동작 후에 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 동적으로 스위칭될 수도 있다.

[0017] 서퍼링 검출 및 캐시 유지보수 동작 관리는 프로세싱 디바이스가 성능 정보를 위해 가속기를 폴링하고, 캐시 유지보수 동작을 트리거하고 프로세싱 디바이스 캐시의 I/O 코히어런시 모드를 동적으로 스위칭하도록 구성된 소프트웨어로 구현될 수도 있다. 하드웨어는 서퍼링을 검출하고 소프트웨어에게 캐시 유지보수 동작을 트리거하고 프로세싱 디바이스 캐시 및/또는 공유 메모리의 I/O 코히어런시 모드를 동적으로 스위칭하도록 알리기 위해 사용될 수도 있다. 하드웨어는 서퍼링을 검출하고 캐시 유지보수 동작을 트리거하고 프로세싱 디바이스 캐시 및/또는 공유 메모리의 I/O 코히어런시 모드를 동적으로 스위칭하는 데 사용될 수도 있다.

[0018] 프로세싱 디바이스 캐시 및/또는 공유 메모리에 대한 비 I/O 코히어런시 모드에서, 가속기는 오프로드된 작업부하의 나머지를 계속 실행할 수도 있다. 오프로드된 작업부하의 실행을 완료한 후에, 캐시 무효화 유지보수 동작이 구현될 수도 있다. 캐시 무효화 유지보수 동작이 비 I/O 코히어런트 오프로드 작업부하 실행 중에 구현된 유일한 캐시 유지보수 동작일 수도 있기 때문에, 오프로드된 작업부하 실행은 캐시 무효화 유지보수 동작에 대한 비 I/O 코히어런트 오버헤드 패널티만을 초래할 수도 있다.

[0019] 가속기에 의한 전체 오프로드된 작업부하 실행의 경우, 서퍼링이 검출되는 경우 I/O 코히어런트 모드를 사용하여 실행의 지속기간을 최소화하여 I/O 코히어런트 모드를 사용하여 전체 오프로드된 작업부하를 실행하는 것보다 적은 양으로 I/O 코히어런트 오버헤드 패널티가 유지될 수도 있다. 비 I/O 코히어런트 오버헤드 패널티는 여전히 I/O 코히어런트 모드에 있는 동안 오프로드된 작업부하 실행과 병렬로 캐시 플러시 유지보수 동작을 실행하여 비 I/O 코히어런트 모드를 사용하여 전체 오프로드된 작업부하를 실행하는 것보다 더 낮은 양으로 유지될 수도 있다.

[0020] 도 1은 다양한 양태와 함께 사용하기에 적합한 컴퓨팅 디바이스 (10) 를 포함하는 시스템을 도시한다. 컴퓨팅 디바이스 (10) 는 프로세서 (14), 메모리 (16), 통신 인터페이스 (18) 및 저장 메모리 인터페이스 (20) 를 갖는 시스템 온 칩 (SoC) (12) 을 포함할 수도 있다. 컴퓨팅 디바이스 (10) 는 유선 또는 무선 모델과 같은 통신 컴포넌트 (22), 저장 메모리 (24) 및 무선 통신 링크를 확립하기 위한 안테나 (26) 를 더 포함할 수도 있다. 프로세서 (14)는 예를 들어 다수의 프로세서 코어와 같은 다양한 프로세싱 디바이스 중 임의의 것을 포함할 수도 있다.

[0021] "시스템-온-칩" (SoC) 이라는 용어는, 배타적이지 않고 통상적으로, 프로세싱 디바이스, 메모리, 및 통신 인터페이스를 포함하는 상호연결된 전자 회로의 세트를 지칭하는데 여기서 사용된다. 프로세싱 디바이스는 범용 프로세서, 중앙 처리 유닛 (CPU), 디지털 신호 프로세서 (DSP), 그래픽 처리 유닛 (GPU), 가속화된 (accelerated) 프로세싱 유닛 (APU), 보조 프로세서, 단일-코어 프로세서, 및 멀티코어 프로세서 등과 같은 다양한 상이한 유형의 프로세서 (14) 및 프로세서 코어를 포함 할 수 있다. 프로세싱 디바이스는 필드 프로그래머블 게이트 어레이 (FPGA), 주문형 집적 회로 (ASIC), 다른 프로그래머블 로직 디바이스, 이산 게이트 로직, 트랜지스터 로직, 성능 모니터링 하드웨어, 워치독 (watchdog) 하드웨어, 및 시간 참조 등과 같은 다른 하드웨어 및 하드웨어 조합을 더 구현할 수도 있다. 집적 회로는 집적 회로의 컴포넌트가 실리콘과 같은 단일 반도체 재료 상에 상주하도록 구성될 수도 있다.

[0022] SoC (12)는 하나 이상의 프로세서 (14)를 포함할 수도 있다. 컴퓨팅 디바이스 (10) 는 2 개 이상의 SoC (12) 를 포함할 수도 있으며, 이에 따라 프로세서 (14) 및 프로세서 코어의 수를 증가시킨다. 컴퓨팅 디바이스 (10) 는 또한 SoC (12) 와 관련되지 않은 프로세서 (14) 를 포함할 수도 있다. 개개의 프로세서 (14) 는 도 2 를 참조하여 이하에서 설명되는 바와 같이 멀티코어 프로세서일 수도 있다. 프로세서 (14) 는 컴퓨팅 디바이스 (10) 의 다른 프로세서 (14) 와 동일하거나 다를 수 있는 특정 목적을 위해 각각 구성될 수도 있다. 동일한 또는 다른 구성의 하나 이상의 프로세서 (14) 및 프로세서 코어는 함께 그룹화 될 수도 있다. 프로세서 (14) 또는 프로세서 코어의 그룹은 멀티프로세서 클러스터로 지칭될 수도 있다.

[0023] SoC (12) 의 메모리 (16) 는 프로세서 (14) 에 의한 액세스를 위해 데이터 및 프로세서 실행 가능 코드를 저장하도록 구성된 휘발성 또는 비 휘발성 메모리 일 수도 있다. 컴퓨팅 디바이스 (10) 및/또는 SoC (12) 는 다양한 목적으로 구성된 하나 이상의 메모리 (16) 를 포함할 수도 있다. 하나 이상의 메모리 (16) 는 RAM (random access memory) 또는 주 메모리 또는 캐시 메모리와 같은 휘발성 메모리를 포함할 수도 있다. 이들 메모리 (16) 는 데이터 센서 또는 서버 시스템으로부터 수신된 제한된 양의 데이터, 비 휘발성 메모리로부터 요청되고 다양한 팩터에 기초하여 장래의 액세스를 예상하여 비 휘발성 메모리로부터 메모리 (16) 로 로딩되는 데이터 및

/또는 프로세서 실행 가능 코드 명령, 및/또는 프로세서 (14) 에 의해 생성되고 비 휘발성 메모리에 저장되지 않고 미래의 빠른 액세스를 위해 일시적으로 저장되는 중간 프로세싱 데이터 및/또는 프로세서 실행 가능 코드 명령을 일시적으로 유지하도록 구성될 수도 있다.

[0024] 메모리 (16) 는, 적어도 일시적으로, 하나 이상의 프로세서 (14) 에 의한 액세스를 위해 다른 메모리 (16) 또는 저장 메모리 (24) 와 같은 다른 메모리 디바이스로부터 메모리 (16) 에 로딩되는 데이터 및 프로세서 실행 가능 코드를 저장하도록 구성될 수도 있다. 메모리 (16) 에 로딩된 데이터 또는 프로세서 실행 가능 코드는 프로세서 (14)에 의한 기능의 실행에 응답하여 로딩될 수도 있다. 기능의 실행에 응답하여 데이터 또는 프로세서 실행 가능 코드를 메모리 (16) 에 로딩하는 것은 실패한 또는 "미스(miss)" 인 메모리 (16) 에 대한 메모리 액세스 요청으로부터 야기될 수 있는데, 이는 요청된 데이터 또는 프로세서 실행 가능 코드가 메모리 (16) 에 위치되지 않기 때문이다. 미스에 응답하여, 다른 메모리 (16) 또는 저장 메모리 (24) 로의 메모리 액세스 요청은 요청된 데이터 또는 프로세서 실행 가능 코드를 다른 메모리 (16) 또는 저장 메모리 (24)로부터 메모리 디바이스 (16) 로 로딩하도록 이루어질 수도 있다. 기능의 실행에 응답하여 데이터 또는 프로세서 실행 가능 코드를 메모리 (16) 에 로딩하는 것은 다른 메모리 (16) 또는 저장 메모리 (24) 에 대한 메모리 액세스 요청으로 인해 발생할 수도 있고, 데이터 또는 프로세서 실행 가능 코드는 나중의 액세스를 위해 메모리 (16) 에 로딩될 수도 있다.

[0025] 저장 메모리 인터페이스 (20) 및 저장 메모리 (24) 는 컴퓨팅 디바이스 (10) 가 비 휘발성 저장 매체 상에 데이터 및 프로세서 실행 가능 코드를 저장할 수 있게하기 위해 일제히 작동할 수도 있다. 저장 메모리 (24) 는 저장 메모리 (24) 가 하나 이상의 프로세서 (14) 에 의한 액세스를 위해 데이터 또는 프로세서 실행 가능 코드를 저장할 수도 있는 메모리 (16) 의 양태와 매우 유사하게 구성될 수도 있다. 비 휘발성인 저장 메모리 (24) 는 컴퓨팅 디바이스 (10)의 전력이 차단된 후에 정보를 보유할 수 있다. 전원이 다시 턴온되고 컴퓨팅 디바이스 (10) 가 재부팅 될 때, 저장 메모리 (24) 에 저장된 정보는 컴퓨팅 디바이스 (10) 에 이용 가능할 수도 있다. 저장 메모리 인터페이스 (20) 는 저장 메모리 (24) 에 대한 액세스를 제어하고 프로세서 (14) 가 저장 메모리 (24) 로부터 데이터를 판독하고 저장 메모리 (24) 에 데이터를 기록하게 할 수도 있다.

[0026] 컴퓨팅 디바이스 (10)의 일부 또는 모든 컴포넌트는 다양한 양태의 기능을 여전히 제공하면서 상이하게 배열되고 및/또는 결합될 수도 있다. 컴퓨팅 디바이스 (10) 는 각 컴포넌트 중 하나에 한정되지 않을 수도 있고, 각 컴포넌트의 다수의 인스턴스가 컴퓨팅 디바이스 (10) 의 다양한 구성에 포함될 수도 있다.

[0027] 도 2 는 일 양태를 구현하기에 적합한 멀티코어 프로세서를 도시한다. 멀티 코어 프로세서 (14) 는 예를 들어, 중앙 처리 장치, 그래픽 처리 장치 및/또는 디지털 처리 장치를 포함하는 다수의 프로세서 유형을 포함할 수도 있다. 멀티코어 프로세서 (14) 는 또한 커스텀 (custom) 하드웨어 가속기를 포함할 수도 있으며, 커스텀 하드웨어 가속기는 커스텀 프로세싱 하드웨어 및/또는 특수 기능 세트를 구현하도록 구성된 범용 하드웨어를 포함할 수도 있다.

[0028] 멀티코어 프로세서는 복수의 동종 또는 이종 프로세서 코어 (200, 201, 202, 203) 를 가질 수도 있다. 동종 멀티코어 프로세서는 복수의 동종 프로세서 코어를 포함할 수도 있다. 멀티코어 프로세서 (14) 의 프로세서 코어들 (200, 201, 202, 203)은 동일한 목적을 위해 구성될 수도 있고 동일하거나 유사한 성능 특성을 가진다는 점에서 프로세서 코어들 (200, 201, 202, 203) 은 동종일 수도 있다. 예를 들어, 멀티코어 프로세서 (14) 는 범용 프로세서일 수 있고, 프로세서 코어 (200, 201, 202, 203) 는 동종의 범용 프로세서 코어일 수도 있다. 멀티코어 프로세서 (14) 는 그래픽 처리 장치 또는 디지털 신호 프로세서일 수 있고, 프로세서 코어 (200, 201, 202, 203) 는 각각 동종 그래픽 프로세서 코어 또는 디지털 신호 프로세서 코어일 수도 있다. 멀티코어 프로세서 (14) 는 동종 프로세서 코어 (200, 201, 202, 203) 를 갖는 커스텀 하드웨어 가속기일 수도 있다. 참조를 쉽게하기 위해, "커스텀 하드웨어 가속기", "프로세서" 및 "프로세서 코어"라는 용어는 여기서 서로 교환가능하게 사용될 수도 있다.

[0029] 이종 멀티코어 프로세서는 복수의 이종 프로세서 코어를 포함할 수도 있다. 멀티코어 프로세서 (14) 의 프로세서 코어들 (200, 201, 202, 203) 은 상이한 목적을 위해 구성될 수도 있고 상이한 성능 특성을 가질 수도 있다는 점에서 프로세서 코어들 (200, 201, 202, 203) 은 이종일 수도 있다. 이러한 이종 프로세서 코어의 이질성은 상이한 명령 세트 아키텍처, 파이프 라인, 동작 주파수 등을 포함할 수 있다. 이러한 이종 프로세서 코어의 예는 보다 느리고 저전력의 프로세서 코어가 보다 강력하고 전력 소모가 많은 프로세서 코어와 커플링될 수도 있는 "big.LITTLE" 아키텍처로 알려진 것을 포함할 수 있다. 유사한 양태들에서, SoC (예를 들어, 도 1 의 SoC (12)) 는 임의의 수의 동종 또는 이종 멀티코어 프로세서 (14) 를 포함할 수 있다. 다양한 양태에서, 이종

의 멀티코어 프로세서는 적어도 하나의 이중 프로세서 코어를 포함하는 프로세서 코어 (200, 201, 202, 203) 의 임의의 조합을 포함할 수 있기 때문에, 모든 프로세서 코어 (200, 201, 202, 203) 가 이중 프로세서 코어일 필요는 없다.

[0030] 멀티코어 프로세서 (14) 의 프로세서 코어들 (200, 201, 202, 203) 각각은 지정된 프로세서 코어 (200, 201, 202, 203) 에 의한 관독 및/또는 기입 액세스에 전용될 수도 있는 전용 캐시 (210, 212, 214, 216) 가 지정될 수도 있다. 전용 캐시 (210, 212, 214, 216) 는 데이터 및/또는 명령을 저장하고, 저장된 데이터 및/또는 명령을, 프로세서 코어 (200, 201, 202, 203) 에 의한 실행에 사용하기 위해, 전용 캐시 (210, 212, 214, 216) 가 전용되는 프로세서 코어 (200, 201, 202, 203) 에 이용가능하게 할 수도 있다. 전용 캐시 (210, 212, 214, 216) 는 도 1 의 메모리 (16) 를 참조하여 여기에 설명된 휘발성 메모리를 포함할 수 있다.

[0031] 멀티 코어 프로세서 (14) 는 프로세서 코어 (200, 201, 202, 203) 에 의한 관독 및/또는 기입 액세스를 위해 구성될 수 있는 공유 캐시 (230) 를 더 포함할 수도 있다. 전용 캐시 (210, 212, 214, 216) 는 데이터 및/또는 명령을 저장하고, 저장된 데이터 및/또는 명령을, 프로세서 코어 (200, 201, 202, 203) 에 의한 실행에 사용하기 위해, 프로세서 코어 (200, 201, 202, 203) 에 이용가능하게 할 수도 있다. 공유 캐시 (230) 는 또한 멀티 코어 프로세서 (14) 로 입력되는 및/또는 멀티 코어 프로세서 (14) 로부터 출력되는 데이터 및/또는 명령에 대한 버퍼로서 기능할 수 있다. 공유 캐시 (230) 는 도 1 의 메모리 (16) 를 참조하여 여기에 설명된 휘발성 메모리를 포함할 수 있다.

[0032] 도 2에 도시된 예에서, 멀티 코어 프로세서 (14) 는 4 개의 프로세서 코어 (200, 201, 202, 203) (즉, 프로세서 코어 0, 프로세서 코어 1, 프로세서 코어 2 및 프로세서 코어 3) 를 포함한다. 이 예에서, 각각의 프로세서 코어 (200, 201, 202, 203) 는 개별적인 전용 캐시 (210, 212, 214, 216) (즉, 프로세서 코어 0 및 전용 캐시 0, 프로세서 코어 1 및 전용 캐시 1, 프로세서 코어 2 및 전용 캐시 2, 및 프로세서 코어 3 및 전용 캐시 3) 가 지정된다. 설명을 쉽게하기 위해, 본 명세서의 예들은 도 2 에 도시된 4 개의 프로세서 코어들 (200, 201, 202, 203) 및 4 개의 전용 캐시들 (210, 212, 214, 216) 을 참조할 수도 있다. 그러나, 본 명세서에서 설명되기도 2 에 도시된 4 개의 프로세서 코어들 (200, 201, 202, 203) 및 4 개의 전용 캐시들 (210, 212, 214, 216) 은 단지 예로서 제공되며, 4 개의 지정된 전용 캐시를 갖는 4-코어 프로세서 시스템에 다양한 양태를 제한하는 것이 결코 아니다. 컴퓨팅 디바이스 (10), SoC (12) 또는 멀티 코어 프로세서 (14) 는 여기에 도시되고 설명된 4 개의 프로세서 코어 (200, 201, 202, 203) 및 전용 캐시 (210, 212, 214, 216) 보다 더 적거나 더 많은 것을 개별적으로 또는 조합하여 포함할 수 있다.

[0033] 도 3 은 코히어런트 상호연결을 갖는 이중 컴퓨팅 디바이스의 예시적인 양태를 도시한다. 동적 I/O 코히어런트 시 시스템을 구현하도록 구성된 이중 컴퓨팅 디바이스 (예를 들어, 도 1 의 컴퓨팅 디바이스 (10)) 는 적어도 2 개의, 그러나 최대 임의의 정수 "N" 개의 프로세싱 디바이스 (예를 들어, 도 1 및 도 2 의 프로세서 (14)); 예를 들어, 프로세싱 디바이스 (예를 들어, CPU) (302), 하드웨어 가속기 (예를 들어, GPU) (306a), 하드웨어 가속기 (예를 들어, DSP) (306b) 및/또는 커스텀 하드웨어 가속기 (306c) 를 포함할 수 있다. 각각의 프로세싱 디바이스 (302, 306a, 306b, 306c) 는 프로세싱 디바이스 캐시 (예를 들어, 도 2 의 전용 캐시 (210, 212, 214, 216) 및/또는 공유 캐시 (230)) 와 연관될 수 있다. 예를 들어, 프로세싱 디바이스 (302) 는 프로세싱 디바이스 캐시 (예를 들어, CPU 캐시) (308) 를 포함할 수도 있고, 하드웨어 가속기 (306a) 는 하드웨어 가속기 캐시 (예컨대, GPU 캐시) (310a) 를 포함할 수도 있고, 하드웨어 가속기 (306b) 는 하드웨어 가속기 캐시 (예를 들어, DSP 캐시) (310b) 를 포함할 수도 있고 및/또는 커스텀 하드웨어 가속기 (306c) 는 커스텀 하드웨어 가속기 캐시 (310c) 를 포함할 수도 있다. 본 명세서에서 논의 된 바와 같이, 각각의 프로세싱 디바이스 (302, 306a, 306b, 306c) 는 서로에 대해 및/또는 그들 각각의 프로세서 코어 (예를 들어, 도 2 의 프로세서 코어 (200, 201, 202, 203)) 사이에서 동종이고 및/또는 이중일 수도 있다.

[0034] 코히어런트 상호연결 (312) 은 프로세싱 디바이스 (302, 306a, 306b, 306c) 및 공유 메모리 (304) (예를 들어, 도 1 의 메모리 (16,24)) 와 같은 임의의 수의 주 메모리 또는 랜덤 액세스 메모리 컴포넌트에 통신 가능하게 접속될 수 있다. 코히어런트 상호연결 (312) 은 다양한 접속된 컴포넌트들 간의 데이터 전송을 가능하게하고 제어하도록 구성될 수 있다. 코히어런트 모드로 동작하는 프로세싱 디바이스 캐시 (308, 310a, 310b, 310c) 및 /또는 공유 메모리 (304) 에 대해, 코히어런트 메카니즘은 코히어런트 상호 연결 (312) 에 의해 연속적으로 구현되어 프로세싱 디바이스 캐시 (308, 310a, 310b, 310c) 및/또는 공유 메모리 (304) 에서 데이터의 정확성을 보장할 수 있다. 상호연결 (312) 은 프로세싱 디바이스들 (302, 306a, 306b, 306c) 사이에서 데이터를 전송하도록 구성되어, 프로세싱 디바이스들 (302, 306a, 306b, 306c) 이 작업부하의 프로세싱을 공유하고 작업부하 프로세싱의 결과를 출력할 수 있도록 한다. 결과가 정확하기 위해서는, 프로세싱 디바이스들 (302, 306a, 306b,

306c) 에 입력되고 그로부터 출력된 데이터는 I/O 코히어런트 모드가 활성화된 동안 코히어런트 상호연결 (312) 의 코히어런시 메카니즘에 의해 구현될 수도 있는 코히어런시 요건에 종속될 수도 있다. 다양한 양태들에서, 코히어런트 도메인 (314) 또는 스누프 필터 (미도시) 와 같은 코히어런트 상호연결 (312) 의 다른 코히어런시 메카니즘들은 프로세싱 디바이스 캐시 (308, 310a, 310b, 310c) 및/또는 공유 메모리 (304) 에 대한 메모리 액세스 요청들의 수신에 기초하여 코히어런시 검사 및 무효화를 수행하도록 구성될 수 있다. 또한, 코히어런시 도메인 (314) 은 또한 프로세싱 디바이스들 (302, 306a, 306b, 306c) 및/또는 공유 메모리 (304) 사이의 트랜잭션들의 확인 응답의 수신에 기초하여 트랜잭션들을 클리어하도록 구성될 수 있다.

[0035] 코히어런트 모드가 비활성이거나, 프로세싱 디바이스 캐시 (308, 310a, 310b, 310c) 및/또는 공유 메모리 (304) 가 비 I/O 코히어런트 모드에 있는 동안, 프로세싱 디바이스 (302, 306a, 306b, 306c) 로 입력되고 그것으로부터 출력되는 데이터의 정확성은 프로세싱 디바이스 (302, 306a, 306b, 306c) 자체에 의해 관리될 수 있다. 다양한 양태들에서, 캐시 유지 보수 동작들은 프로세싱 디바이스 캐시 (308, 310a, 310b, 310c) 에 저장된 데이터에 대해 프로세싱 디바이스들 (302, 306a, 306b, 306c) 에 의해 구현될 수 있다. 다양한 양태들에서, 캐시 관리 디바이스들 (미도시) 및/또는 프로세서 코어들을 포함하는 프로세싱 디바이스들 (302, 306a, 306b, 306c) 의 컴포넌트들은 캐시 유지보수 동작을 구현하도록 구성될 수 있다. 코히어런트 상호연결 (312) 는 여전히 코히어런시 동작을 수행하지 않고 프로세싱 디바이스들 (302, 306a, 306b) 및/또는 공유 메모리 (304) 사이에서 데이터를 전송할 수 있다.

[0036] 도 3 은 동적 I/O 코히어런시 시스템을 구현하도록 구성된 이중 컴퓨팅 디바이스의 비 제한적인 예를 도시한다. 특히 도 3 의 및 도 3 과 관련된 예들을 참조하여, 여기에 도시되고 설명된 예들은 비제한적이다. 동적 I/O 코히어런시 시스템을 구현하도록 구성된 이중 컴퓨팅 디바이스는 임의의 개수의 및/또는 조합의 프로세싱 디바이스, 프로세서 코어, 프로세싱 디바이스 캐시 (예를 들어, 전용 캐시 및/또는 공유 캐시), 코히어런트 상호연결, 공유 메모리, 및 그러한 컴포넌트 간의 연결을 포함할 수도 있다.

[0037] 도 4 내지 도 7 은 작업부하 오프로딩의 예시의 양태들을 도시한다. 이들 예는 프로세싱 디바이스 (예를 들어, CPU) (302) 로부터 하드웨어 가속기 (예를 들어, GPU, DSP, 커스텀 하드웨어 가속기 등) (306) 로의 작업부하 오프로딩을 도시한다. 이러한 예는 프로세싱 디바이스 (302) 와 단일 하드웨어 가속기 (306) 사이의 작업부하 오프로딩을 도시한다. 다양한 양태들에서, 프로세싱 디바이스 (302) 와 하드웨어 가속기 (306) 사이에서 전송되는 데이터는 공유 메모리 (예를 들어, 도 3 의 공유 메모리 (304)) 에 대한 판독/기입을 통해 전송될 수도 있고, 캐시 미스 메카니즘을 통해 트리거될 수 있다. 다양한 양태들에서, 공유 메모리를 통한 것을 포함하는 데이터 전송은 오프로드된 작업부하의 실행 전체에 걸쳐 발생할 수 있다. 일대일 작업부하 오프로딩의 이러한 예는 비 제한적이며, 프로세싱 디바이스 (302) 는 임의의 수의 공유 메모리를 통하는 것을 포함하여 임의의 수의 하드웨어 가속기 (306) 및/또는 다른 프로세싱 디바이스 (302) 로 작업부하의 부분을 오프로딩할 수 있다.

[0038] 도 4 내지 도 7 에 도시된 예들 각각에서, 프로세싱 디바이스 (302) 는 작업부하를 실행하는 임무를 맡게 될 수 있다. 프로세싱 디바이스 (302) 는 작업부하의 일부를 하드웨어 가속기 (306) 로 오프로드하도록 구성될 수 있다. 오프로드 된 작업부하에 대한 데이터는 프로세싱 디바이스 캐시 (예를 들어, 도 6 의 프로세싱 디바이스 캐시 (308)) 에 저장될 수 있다. 데이터는 코히어런트 상호연결 (예를 들어, 도 3 의 코히어런트 상호연결 (312))를 통해 하드웨어 가속기 (306) 에 전송될 수 있다. 다양한 양태들에서, 데이터는 코히어런트 상호연결에 통신 가능하게 연결되거나 통합된 공유 메모리를 통해 전송될 수 있다. 하드웨어 가속기 (306) 는 전송된 데이터를 사용하여 오프로드 된 작업부하를 실행하고, 프로세싱 디바이스 캐시에 기입되도록 프로세싱 디바이스 (302) 에 결과를 다시 전송할 수 있다.

[0039] 도 4 는 비 I/O 코히어런트 작업부하 프로세싱의 예시의 양태를 도시한다. 오프로드 된 작업부하에 대한 데이터를 하드웨어 가속기 (306) 에 전송하고, 하드웨어 가속기 (306) 로부터 수신된 결과를 출력하기 위해, 프로세싱 디바이스 (302) 는 데이터를 공유 메모리에 및 그 후 하드웨어 가속기 (306) 에 또는 코히어런트 상호연결을 통해 하드웨어 가속기 (306) 에 직접 기입하기 위해 캐시 유지보수 동작을 구현할 수 있다. 본 명세서에 언급된 바와 같이, 비 I/O 코히어런트 데이터 전송 동안, 코히어런트 상호연결은 코히어런시 동작을 수행하지 않고 프로세싱 디바이스 (302) 와 공유 메모리 및/또는 하드웨어 가속기 (306) 사이의 데이터간에 데이터를 전송할 수 있다.

[0040] 오프로드 된 작업부하에 대한 데이터를 하드웨어 가속기 (306) 에 전송하기 위해, 프로세싱 디바이스 (302) 는 데이터를 코히어런트 상호연결을 통해 공유 메모리 및/또는 하드웨어 가속기 (306) 에 기록하기 위해 캐시 플러시 유지 보수 동작 (400) 을 구현할 수 있다. 데이터를 다중 목적지에 기입하는 캐시 플러시 유지 보수 동작

(400) 은 오프로드 된 작업부하에 대해 기록되는 프로세싱 디바이스 캐시 내의 데이터의 양에 따라 변할 수 있는 오버 헤드 (404a) 를 생성한다. 프로세싱 디바이스 캐시에 저장된 데이터는 파티션 또는 버퍼에 저장될 수 있으며, 캐시 플러시 유지 보수 동작 (400) 은 프로세싱 디바이스 캐시의 이러한 특정 부분에 대해 구현될 수 있다. 그렇지 않으면, 캐시 플러시 유지 보수 동작 (400) 은 전체 프로세싱 디바이스 캐시에 대해 구현될 수 있다. 오버헤드 (404a) 는 하드웨어 가속기 (306) 에 의한 오프로드된 작업부하 실행의 시작을 지연시킬 수 있다.

[0041] 캐시 플러시 유지 보수 동작 (400) 의 오버 헤드 (404a) 에 상관되는 시간 동안, 프로세싱 디바이스 (302) 는 하드웨어 가속기 (306) 에 대한 작업부하의 일부를 오프로드하는 것을 완료할 수 있다. 다양한 양태들에서, 작업부하의 일부분을 하드웨어 가속기 (306) 로 오프로딩하는 것은 프로세싱 디바이스 캐시 및/또는 공유 메모리에 액세스하는 하드웨어 가속기 (306) 에 의한 데이터 판독 및 기입을 포함할 수 있다. 공유 메모리에 대한 액세스는 프로세싱 디바이스 캐시의 캐시 미스에 대한 캐시 미스 (cache miss) 메커니즘을 통해 구현될 수 있다. 하드웨어 가속기 (306) 는 데이터를 국부적으로 캐시할 필요없이 프로세싱 디바이스 캐시 및/또는 공유 메모리로부터 추출된 데이터를 사용하여 오프로드 된 작업부하를 실행할 수 있다. 다양한 양태들에서, 하드웨어 가속기 (306) 는 오프로드 된 작업부하에 대한 임의의 전송된 데이터를 그 하드웨어 가속기 캐시 (예를 들어, 도 3 의 프로세싱 디바이스 캐시 (310a, 310b, 310c)) 에 저장할 수 있다. 하드웨어 가속기 (306) 는 오프로드 된 작업부하 (408) 의 실행을 론치하고, 오프로드 된 작업부하 (406a) 를 실행할 수 있다. 오프로드 된 작업부하를 실행하는 데 소요되는 시간의 길이는 오프로드 된 작업부하의 크기 및/또는 복잡성, 전력, 컴퓨팅 전력, 메모리 공간을 포함하는 하드웨어 가속기 (306) 의 및/또는 그것에 대한 리소스 사용 및/또는 이용가능성, 통신 대역폭, 및/또는 온도를 포함하는 환경 인자를 포함하는 여러 인자들에 의존할 수 있다. 하드웨어 가속기 (306) 는 오프로드 된 작업부하 실행을 완료하여 (410) 오프로드 된 작업부하 실행 (406a)의 결과를 코히어런트 상호 연결을 통해 프로세싱 디바이스 (302) 로 복귀시킬 수 있다.

[0042] 프로세싱 디바이스 (302) 가 오프로드 된 작업부하 실행의 결과를 이용할 수 있게 하기 위해, 프로세싱 디바이스 (302) 는 결과가 하드웨어 가속기 (306) 에 의해 출력된 후에 결과를 미스로 현재 저장하고 있는 공유 메모리로 프로세싱 디바이스 캐시 공간의 장래 판독을 지시하기 위해 캐시 무효화 유지보수 동작 (402) 을 구현할 수 있다. 캐시 무효화 유지보수 동작 (402) 은 또한 저장된 오프로드 된 작업부하 결과의 데이터량 또는 오프로드 된 작업부하 결과를 저장하기 위해 프로세싱 디바이스 캐시에서 사용되는 공간에 따라 변할 수 있는 오버 헤드 (404b) 를 생성 할 수 있다. 오버 헤드 (404b) 는 프로세싱 디바이스 (302) 에 의한 액세스를 위한 오프로드 된 작업부하 결과의 이용가능성을 지연시킬 수 있다. 캐시 무효화 유지보수 동작 (402) 의 완료 시, 오프로드 된 작업부하 결과는 프로세싱 디바이스 (302) 에 의한 오프로드 된 작업부하 결과의 액세스, 사용 및/또는 출력을 위해 이용 가능하게 될 수 있다. 프로세싱 디바이스 (302) 는 오프로드 된 작업부하 실행 (406a) 의 결과를 수신하고 결과를 그 프로세싱 디바이스 캐시에 저장할 수 있다.

[0043] 도 5 는 I/O 코히어런트 작업부하 프로세싱의 예시의 양태를 도시한다. 오프로드 된 작업부하에 대한 데이터를 하드웨어 가속기 (306) 에 전송하고, 하드웨어 가속기 (306) 로부터 수신된 결과를 출력하기 위해, 코히어런시 메커니즘 (coherency mechanisms) (예를 들어, 도 3 의 코히어런시 도메인 (314) 및/또는 스누프 필터) 은 프로세싱 디바이스 캐시 및/또는 공유 메모리에서 데이터의 정확성을 보장하기 위해 코히어런트 상호 연결 (312) 에 의해 연속적으로 구현될 수 있다. 따라서, 프로세싱 디바이스 캐시 및/또는 공유 메모리가 I/O 코히어런트 모드에 있는 동안, 프로세싱 디바이스 (302) 는 그것이 비 I/O 코히어런트 모드에 대해 구현할 수 있는 캐시 유지 보수 동작을 구현하지 않을 수 있다.

[0044] 프로세싱 디바이스 (302) 는 작업부하의 일부를 하드웨어 가속기 (306) 로 오프로드할 수 있다. 다양한 양태들에서, 작업부하의 일부분을 하드웨어 가속기 (306) 로 오프로딩하는 것은 프로세싱 디바이스 캐시 및/또는 공유 메모리에 액세스하는 하드웨어 가속기 (306) 에 의한 데이터 판독 및 기입을 포함할 수 있다. 공유 메모리에 대한 액세스는 프로세싱 디바이스 캐시의 캐시 미스에 대한 캐시 미스 메커니즘을 통해 구현될 수 있다. 하드웨어 가속기 (306) 는 데이터를 국부적으로 캐시할 필요없이 프로세싱 디바이스 캐시 및/또는 공유 메모리로부터 추출된 데이터를 사용하여 오프로드 된 작업부하를 실행할 수 있다. 다양한 양태들에서, 하드웨어 가속기 (306) 는 오프로드 된 작업부하에 대한 임의의 전송된 데이터를 그 하드웨어 가속기 캐시 (예를 들어, 도 3 의 프로세싱 디바이스 캐시 (310a, 310b, 310c)) 에 저장할 수 있다. 하드웨어 가속기 (306) 는 오프로드 된 작업부하 (408) 의 실행을 론치하고, 오프로드 된 작업부하 (406b) 를 실행할 수 있다. 비 I/O 코히어런트 모드에서 프로세싱 디바이스 캐시 및/또는 공유 메모리에 대한 오프로드 된 작업부하 데이터를 전송하기위한 오버 헤드를 발생시키기보다는, I/O 코히어런트 모드에서 데이터가 전송되는 동안 코히어런시가 유지될 수 있어서, 오프로드

된 작업부하의 론치 실행 (408) 이전에 오버 헤드(404)가 발생되지 않도록 한다.

[0045] 오버헤드 (404c)의 증가는 여전히 오프로드 된 작업부하의 실행 (406b) 중에 발생할 수 있다. 이 오버 헤드 (404c)는 오프로드 된 작업부하 결과의 데이터가 이동하는 코히어런트 경로의 성능에 의해 야기될 수 있다. 코히어런트 상호연결은 그의 코히어런시 메커니즘을 구현할 수 없고 하드웨어 가속기 (306)가 오프로드 된 작업부하 결과를 생성하는 것만큼 빠르게 오프로드 된 작업부하 결과를 전달할 수 없을 수도 있다. 상호연결은 코히어런시 메커니즘을 구현하는 동안 오프로드 된 작업부하 결과에 병목 현상을 야기하여 추가된 오버 헤드 (404c)를 야기할 수도 있다. 다양한 양태들에서, 오버 헤드 (404c)는 오프로드 된 작업부하를 실행 (406b)하기 위한 하드웨어 가속기 (306)의 실행 시간에 비례하거나 대략 비례할 수 있다.

[0046] 상호연결의 코히어런시 메커니즘들이 그들의 동작을 완료하여 프로세싱 디바이스 캐시 및/또는 공유 메모리 내의 오프로드 된 작업부하 결과의 정확성을 보장하면, 오프로드 된 작업부하 결과는 프로세싱 디바이스 (302)에 의한 오프로드 된 작업부하 결과의 액세스, 사용 및/또는 출력에 이용가능하게 될 수 있다.

[0047] 도 6 및 도 7은 동적 I/O 코히어런트 작업부하 프로세싱의 예시의 양태를 도시한다. 도 6 및 도 7 각각에 도시된 예에서, 프로세싱 디바이스 (302)에 의한 작업부하 오프로딩은 I/O 코히어런트 모드에서 프로세싱 디바이스 캐시 및/또는 공유 메모리로 시작될 수 있다. 따라서, 오프로드 된 작업부하에 대한 데이터를 하드웨어 가속기 (306)에 전송하고, 하드웨어 가속기 (306)에 의해 생성된 결과를 수신 및/또는 출력하기 위해, 코히어런시 메커니즘 (예를 들어, 도 3의 코히어런시 도메인 (314) 및/또는 스누프 필터)은 프로세싱 디바이스 캐시 및/또는 공유 메모리에서 데이터의 정확성을 보장하기 위해 코히어런트 상호 연결 (312)에 의해 연속적으로 구현될 수 있다. 따라서, 프로세싱 디바이스 캐시 및/또는 공유 메모리가 I/O 코히어런트 모드에 있는 동안, 프로세싱 디바이스 (302)는 비 I/O 코히어런트 모드를 위해 구현할 수 있는 캐시 유지 보수 동작을 구현하지 않을 수 있다.

[0048] 프로세싱 디바이스 (302)는 작업부하의 일부를 하드웨어 가속기 (306)로 오프로드할 수 있다. 다양한 양태들에서, 작업부하의 일부분을 하드웨어 가속기 (306)로 오프로딩하는 것은 프로세싱 디바이스 캐시 및/또는 공유 메모리에 액세스하는 하드웨어 가속기 (306)에 의한 데이터 관독 및 기입을 포함할 수 있다. 공유 메모리에 대한 액세스는 프로세싱 디바이스 캐시의 캐시 미스에 대한 캐시 미스 메커니즘을 통해 구현될 수 있다. 하드웨어 가속기 (306)는 데이터를 국부적으로 캐시할 필요없이 프로세싱 디바이스 캐시 및/또는 공유 메모리로부터 추출된 데이터를 사용하여 오프로드 된 작업부하를 실행할 수 있다. 다양한 양태들에서, 하드웨어 가속기 (306)는 오프로드 된 작업부하에 대한 임의의 전송된 데이터를 그 하드웨어 가속기 캐시 (예를 들어, 도 3의 프로세싱 디바이스 캐시 (310a, 310b, 310c))에 저장할 수 있다. 하드웨어 가속기 (306)는 오프로드된 작업부하의 실행을 론치하고 (408) 오프로드 된 작업부하를 실행 (406c, 406d)할 수 있다. 비 코히어런트 모드에 프로세싱 디바이스 캐시 및/또는 공유 메모리에 대한 오프로드 된 작업부하 데이터를 전송하기 위한 오버 헤드를 발생시키기보다는, I/O 코히어런트 모드에서 데이터가 전송되는 동안 코히어런시가 유지될 수 있어서, 오프로드 된 작업부하의 론치 실행 (408) 이전에 오버 헤드가 발생되지 않도록 한다.

[0049] 도 6에 도시된 예는 하드웨어 가속기 (306)에 의한 단기간 오프로드 된 작업부하 실행을 위한 동적 I/O 코히어런트 작업부하 프로세싱의 구현이다. 여기서 기술된 바와 같이, 오버헤드 (404d)는 여전히 오프로드 된 작업부하의 실행 (406c)에 대해 발생할 수 있다. 동적 I/O 코히어런트 작업부하 프로세싱에서, 오버 헤드 (404d)가 오버 헤드 임계치 아래로 유지되는 한, 오프로드 된 작업부하의 실행 (406c)은 I/O 코히어런트 모드에서 프로세싱 디바이스 캐시 및/또는 공유 메모리로 계속할 수 있다. 따라서, 오버 헤드 (404d)는 본 명세서에 설명된 바와 같이 오프로드 된 작업부하 결과를 전송하는 코히어런트 상호연결의 성능에 의해 야기되는 오버 헤드로 제한된다. 상호연결의 코히어런시 메커니즘들이 그들의 동작을 완료하여 프로세싱 디바이스 캐시 및/또는 공유 메모리 내의 오프로드된 작업부하 결과의 정확성을 보장하면, 오프로드된 작업부하 결과는 프로세싱 디바이스 (302)에 의한 오프로드된 작업부하 결과의 액세스, 사용 및/또는 출력에 이용가능하게 될 수 있다 (412).

[0050] 도 7에 도시된 예는 하드웨어 가속기 (306)에 의한 장기간 오프로드 된 작업부하 실행을 위한 동적 I/O 코히어런트 작업부하 프로세싱의 구현이다. 여기서 기술된 바와 같이, 오버헤드 (404e)는 여전히 오프로드 된 작업부하의 실행 (406c)에 대해 발생할 수 있다. 동적 I/O 코히어런트 작업부하 프로세싱에서, 장기간 오프로드 된 작업부하는 오버 헤드 임계값을 초과하는 오버 헤드 (404e)를 생성할 수 있다. 여기에 더 설명되는 바와 같이, 유희 상태에 진입하는 프로세싱 디바이스 (302), 경과 시간, 및/또는 오프로드 된 작업부하의 실행의 서퍼팅을 포함하는, 오버 헤드 임계값을 초과하는 장기간 작업부하를 검출하기 위해 다양한 방법이 사용될 수 있다.

- [0051] 장기간 오프로드 된 작업부하를 검출하는 것에 응답하여, 프로세싱 디바이스 (302) 는 프로세싱 디바이스 캐시 및/또는 공유 메모리를 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 동적으로 천이시킬 수 있다. I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 천이시키기 위해, 프로세싱 디바이스 (302) 는 프로세싱 디바이스 캐시 및/또는 공유 메모리가 I/O 코히어런트 모드로 유지되는 동안 캐시 유지 보수 동작을 구현하기 시작할 수 있다. 예를 들어, 프로세싱 디바이스 (302) 는 하드웨어 가속기가 오프로드 된 작업부하 실행 (404d) 을 계속하는 동안 캐시 플러시 유지 보수 동작 (400) 을 구현할 수 있다. 캐시 플러시 유지 보수 동작 (400) 동안, 오프로드 된 작업부하 실행 (404d) 은 코히어런트 모드 오버 헤드 (404e) 를 계속 초래할 수 있지만, 캐시 플러시 유지보수 동작 (400) 에 대한 오버 헤드가 발생하지 않을 수 있다. 따라서, 오버 헤드 (404d) 는 본 명세서에 설명된 바와 같이 오프로드 된 작업부하 결과를 전송하는 코히어런트 상호연결의 성능에 의해 야기되는 오버 헤드로 제한될 수 있다.
- [0052] 캐시 플러시 유지보수 동작 (400) 의 완료 시에, 프로세싱 디바이스 캐시 및/또는 공유 메모리는 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 동적으로 스위칭할 수도 있다. I/O 코히어런트 모드와 비 I/O 코히어런트 모드 간의 스위칭 (414) 은 오프로드 된 작업부하 실행 (404d) 을 인터럽트하지 않을 수 있다. 오프로드 된 작업부하 실행 (404d) 이 계속되는 동안, 더 이상의 I/O 코히어런트 모드 오버 헤드 (404e) 가 발생할 수 없다. 하드웨어 가속기 (306) 는 오프로드된 작업부하 실행을 완료 (410) 하여 오프로드된 작업부하 실행 (406d) 의 결과를 코히어런트 상호연결 및/또는 공유 메모리를 통해 프로세싱 디바이스 (302) 로 복귀시킬 수 있다.
- [0053] 오프로드 된 작업부하 실행의 결과를 프로세싱 디바이스 (302) 에 이용 가능하게하기 위해, 프로세싱 디바이스 (302) 는 여기에서 더 논의되는 바와 같이 캐시 무효화 유지보수 동작 (402) 을 구현할 수 있다. 캐시 무효화 유지보수 동작 (402) 은 저장된 오프로드 된 작업부하 결과의 데이터량 또는 오프로드 된 작업부하 결과를 저장하기 위해 프로세싱 디바이스 캐시에서 사용되는 공간에 따라 변할 수 있는 오버 헤드 (404b) 를 생성할 수 있다. 오버 헤드 (404b) 는 프로세싱 디바이스 (302) 에 의한 액세스를 위한 오프로드 된 작업부하 결과의 이용가능성을 지연시킬 수 있다. 캐시 무효화 유지보수 동작 (402) 의 완료 시, 오프로드 된 작업부하 결과는 프로세싱 디바이스 (302) 에 의한 오프로드 된 작업부하 결과의 액세스, 사용 및/또는 출력을 위해 이용 가능하게 될 수 있다 (412). 프로세싱 디바이스 (302) 는 오프로드 된 작업부하 실행 (406a) 의 결과를 수신하고 결과를 그 프로세싱 디바이스 캐시에 저장할 수 있다.
- [0054] 도 8 은 일 양태에 따라 동적 I/O 코히어런트 작업부하 프로세싱을 구현하는 방법 (800) 을 도시한다. 방법 (800) 은 프로세서 (예를 들어, 도 1 및 도 2 의 프로세서 (14) 및 도 3 내지 도 7 의 프로세싱 디바이스 (302)) 에서, 범용 하드웨어에서, 전용 하드웨어에서, 또는 다른 개개의 컴포넌트들을 포함하는 동적 I/O 코히어런시 시스템 내에서 소프트웨어를 실행하는 프로세서 (예를 들어, 도 3) 와 같은 소프트웨어 구성 프로세서와 전용 하드웨어의 결합에서 실행되는 소프트웨어로 컴퓨팅 디바이스에서 구현될 수 있다. 다양한 양태들에서 인에이블된 대안적인 구성들을 포함하기 위해, 방법 (800) 을 구현하는 하드웨어는 본 명세서에서 "프로세싱 디바이스" 로 지칭된다.
- [0055] 블록 (802) 에서, 프로세싱 디바이스는 프로세싱 디바이스 캐시 (예를 들어, 도 3 의 프로세싱 디바이스 캐시 (308)) 및/또는 공유 메모리 (예를 들어, 도 3 의 공유 메모리 (304)) 에 대해 I/O 코히어런트 모드를 "온" 으로 설정할 수 있다. 대안적으로, 프로세싱 디바이스는 프로세싱 디바이스 캐시 및/또는 공유 메모리에 대해 비 I/O 코히어런트 모드를 "오프" 로 설정할 수 있다. 다양한 양태들에서, I/O 코히어런트 모드는 프로세싱 디바이스 캐시 및/또는 공유 메모리에 대한 디폴트 상태로 "온"으로 설정 될 수 있으므로, I/O 코히어런트 모드를 "온"으로 설정하는 것은 불필요할 수 있다. I/O 코히어런트 모드를 "온"으로 설정하는 것은 "온" 이 디폴트 I/O 코히어런트 모드 상태 일지라도 사전 예방 조치일 수 있다.
- [0056] 블록 (804) 에서, 프로세싱 디바이스는 I/O 코히어런트 모드를 사용하여 하드웨어 가속기 (예를 들어, 도 3 의 GPU (306a), DSP (306b), 커스텀 하드웨어 가속기 (306c) 및 도 4 내지 도 7 의 하드웨어 가속기 (306)) 상에서 오프로드 된 작업부하 실행을 론치할 수 있다. 본 명세서에서 논의된 바와 같이, I/O 코히어런트 모드를 사용할 때 프로세싱 디바이스로부터 전송하는 데에 오버헤드가 발생되지 않을 수 있기 때문에, 그 실행을 론치하는 것은 하드웨어 가속기로의 작업부하의 오프로드에 즉각적일 수 있다. 프로세싱 디바이스와 하드웨어 가속기 사이에서 데이터를 전송하기 위한 메모리 액세스 요청은 코히어런트 상호연결 (예를 들어, 도 3 의 코히어런트 도메인 (314)) 의 I/O 코히어런트 경로를 통해 라우팅될 수 있다.
- [0057] 프로세싱 디바이스와 하드웨어 가속기 사이에서 전송된 데이터는 오프로드 된 작업부하를 실행하기 위한 프로세싱 디바이스로부터 하드웨어 가속기로의 데이터 및 그 실행된 오프로드 된 작업부하의 결과로서 하드웨어 가

속기로부터 프로세싱 디바이스로의 데이터를 포함할 수 있다. 프로세싱 디바이스와 하드웨어 가속기 사이에서 전송된 데이터는 공유 메모리를 통해 전송될 수 있다.

- [0058] 결정 블록 (806) 에서, 프로세싱 디바이스는 오프로드 된 작업부하가 하드웨어 가속기에 의해 완료되었는지 여부를 결정할 수 있다.
- [0059] 오프로드된 작업부하가 하드웨어 가속기에 의해 완료되었다는 결정에 응답하여 (즉, 결정 블록 (806) = "예"), 프로세싱 디바이스는 블록 (818)에서 하드웨어 가속기로부터 오프로드 된 작업부하의 실행 결과를 수신할 수 있다. 오프로드 된 작업부하의 실행 결과는 하드웨어 가속기에 의해 공유 메모리에 출력되고 프로세싱 디바이스에 의해 공유 메모리로부터 취출될 수 있다. 코히어런트 모드에서 전체적으로 오프로드 된 작업부하의 실행의 완료는 데이터의 코히어런시를 관리하는 코히어런트 상호연결의 I/O 코히어런트 메커니즘의 결과로 캐시 유지 보수 동작이 불필요하다는 것을 의미할 수 있다.
- [0060] 오프로드 된 작업부하가 하드웨어 가속기에 의해 완료되지 않았다고 결정한 것에 응답하여 (즉, 결정 블록 (806) = "아니오"), 프로세싱 디바이스는 I/O 코히어런트 모드와 비 I/O 코히어런트 모드 사이를 스위칭하기 위한 동적 트리거가 결정 블록 (808) 에서 검출되는지 여부를 결정할 수 있다. 동적 트리거는 오프로드 된 작업부하가 I/O 코히어런트 모드 오버 헤드 패널티가 바람직하지 않거나 수용할 수 없는 지점까지 증가할 때를 결정하는 데 사용될 수 있다. 동적 트리거가 검출되는지 여부를 결정하는 것은 다양한 방식의 임의의 조합으로 성취될 수 있다.
- [0061] 일부 양태들에서, 동적 트리거는 미리 설정되거나 계산된 경과 시간을 포함할 수 있으며, 동적 트리거를 검출하는 것은 그 시간이 경과한 것을 검출하는 것을 수반할 수 있다. 다양한 양태들에서, 그 시간은 하드웨어 가속기에 의한 오프로드 된 작업부하의 실행에 대한 예상 시간량에 기초하여 미리 설정되거나 계산될 수 있다. 오프로드 된 작업부하의 예상 실행 시간은 오프로드 된 작업부하의 크기 및/또는 복잡성, 전력, 컴퓨팅 전력, 메모리 공간 및/또는 통신 대역폭을 포함하는 하드웨어 가속기의 및/또는 그것에 대한 리소스 사용 및/또는 이용 가능성, 및/또는 온도를 포함하는 환경 인자를 포함하는 여러 인자들에 기초할 수 있다. 다양한 양태들에서, 그 시간은 I/O 코히어런트 모드와 비 I/O 코히어런트 모드 간의 즉각적인 스위칭을 트리거하기 위해 제로 (0) 와 같은 사소한 값으로서 미리 설정되거나 계산될 수 있다.
- [0062] 몇몇 양태들에서, 동적 트리거는 유휴 상태로 천이하는 및/또는 유휴 상태에 있는 프로세싱 디바이스를 감지하는 것을 포함할 수 있다. 유휴 상태는 프로세싱 디바이스가 작동하지 않고 하드웨어 가속기가 오프로드 된 작업부하를 완료하기를 기다리고 있기 때문에 하드웨어 가속기가 오프로드 된 작업부하를 예상보다 느리게 실행 중일 수 있다는 것을 나타낼 수 있다. 작업부하를 오프로딩하는 목표는 컴퓨팅 리소스를 효율적으로 사용하여 작업부하를 실행하는 것이다. 따라서, 유휴 상태로 천이하는 및/또는 유휴 상태에 있는 프로세싱 디바이스는 컴퓨팅 자원의 비효율적인 사용을 나타낼 수 있다.
- [0063] 일부 양태들에서, 동적 트리거는 I/O 코히어런트 경로를 사용하는 것을 겪는 오프로드 된 작업부하의 실행의 표시자를 포함할 수 있다. 오프로드된 작업부하 서퍼링은 다양한 메트릭을 기반으로 검출될 수 있다. 증대된 작업부하는 드라이버에게 성능을 알릴 수도 있다. 테스트링 또는 무작위 테스트를 통한 작업부하 분류에 기초한 정적 할당 또는 방법의 사용은 표준 시간 초과나 작업부하 서퍼링을 나타내는 신호를 야기할 수도 있다. 하드웨어 성능 레지스터를 폴링하는 것은 작업부하 서퍼링을 나타내는 데이터를 제공할 수도 있다. 최대값 근처의, 대역폭 및/또는 미해결 트랜잭션과 같은 지속적으로 높은 상호연결 동작 레벨은 스토틀링을 나타낼 수도 있다. 높은 가속기 메모리 레이턴시는 레이턴시 제한을 나타낼 수도 있다. 높은 가속기 메모리 백프레셔는 대역폭 스토틀링을 나타낼 수도 있다. 높은 점유 메모리 요청 대기열과 결합한 메모리로 인한 높은 가속기 유휴 상태는 대역폭이 제한되는 것을 나타낼 수도 있다.
- [0064] I/O 코히어런트 모드와 비 I/O 코히어런트 모드 간의 스위칭을 위한 동적 트리거가 검출되지 않는다는 결정에 응답하여 (즉, 결정 블록 (808) = "아니오"), 프로세싱 디바이스는 결정 블록 (806) 에서 하드웨어 가속기에 의해 오프로드 된 작업부하가 완료되었는지를 결정할 수 있다.
- [0065] I/O 코히어런트 모드와 비 I/O 코히어런트 모드 간의 스위칭을 위한 동적 트리거가 검출된다는 결정에 응답하여 (즉, 결정 블록 (808) = "예"), 프로세싱 디바이스는 블록 (810) 에서 연기된 프로세싱 디바이스 캐시 유지보수를 실행할 수 있다. 본 명세서에서 논의된 바와 같이, I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 스위칭하기위한 준비에서, 프로세싱 디바이스는 프로세싱 디바이스 캐시에 대한 캐시 플러시 유지 보수 동작을 실행할 수 있다. 하드웨어 가속기는 프로세싱 디바이스가 블록 (810) 에서 캐시 플러시 유지보수 동작을 실행하는

동안 오프로드 된 작업부하를 계속해서 실행할 수 있다.

- [0066] 결정 블록 (812) 에서, 프로세싱 디바이스는 오프로드 된 작업부하가 하드웨어 가속기에 의해 완료되었는지 여부를 결정할 수 있다.
- [0067] 오프로드 된 작업부하가 하드웨어 가속기에 의해 완료되었다는 결정에 응답하여 (즉, 결정 블록 (812) = "예"), 프로세싱 디바이스는 블록 (818)에서 하드웨어 가속기로부터 수신된 오프로드 된 작업부하의 실행 결과를 수신할 수 있다. I/O 코히어런트 모드에서 전체적으로 오프로드 된 작업부하의 실행의 완료는 데이터의 코히어런시를 관리하는 코히어런트 상호연결의 I/O 코히어런트 메커니즘의 결과로 캐시 유지 보수 동작이 불필요하다는 것을 의미할 수 있다. 캐시 플러시 유지 보수 동작이 이미 실행되었을 수도 있지만, I/O 코히어런트 모드에 있는 동안 오프로드 된 작업부하의 실행과 병행하는 캐시 플러시 유지 보수 동작의 실행으로 인해, 캐시 플러시 유지 보수 동작의 실행에 의해 추가 오버 헤드가 발생하지 않을 수 있다.
- [0068] 오프로드 된 작업부하가 하드웨어 가속기에 의해 완료되지 않았다고 결정한 것에 응답하여 (즉, 결정 블록 (812) = "아니오"), 프로세싱 디바이스는 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로 프로세싱 디바이스 캐시 및/또는 공유 메모리를 스위칭할 수 있다. 블록 (814) 에서의 I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로의 스위칭은 I/O 코히어런트 모드의 상태를 "오프"로 설정하거나 비 I/O 코히어런트 모드의 상태를 "온" 으로 설정하는 것을 포함할 수 있다. I/O 코히어런트 모드에서 비 I/O 코히어런트 모드로의 스위칭은 코히어런트 상호연결을 통한 데이터 및 메모리 액세스 요청의 라우팅을 변경하여 데이터 및 요청이 코히어런트 도메인을 통해 더 이상 라우팅되지 않고 코히어런시 동작이 데이터 및 메모리 액세스 요청에 대해 더 이상 구현되지 않는다. 오히려, 프로세싱 디바이스는 프로세싱 디바이스 캐시 및/또는 공유 메모리가 비 I/O 코히어런트 모드로 설정되는 것에 응답하여 캐시 유지 보수 동작을 구현하기 시작할 수 있다.
- [0069] 블록 (816) 에서, 하드웨어 가속기는, 여기서 더 기술되는 바와 같이, 프로세싱 디바이스 캐시 및/또는 공유 메모리가 비 I/O 코히어런트 모드에 있는 동안 오프로드 된 작업부하를 계속 실행할 수 있다.
- [0070] 도 9 는 일 양태에 따라 동적 I/O 코히어런트 작업부하 프로세싱을 구현하는 방법 (900) 을 도시한다. 방법 (900) 은 프로세서 (예를 들어, 도 1 및 도 2 의 프로세서 (14) 및 도 3 내지 도 7 의 프로세싱 디바이스 (302)) 에서, 범용 하드웨어에서, 전용 하드웨어에서, 또는 다른 개개의 컴포넌트들을 포함하는 동적 I/O 코히어런시 시스템 내에서 소프트웨어를 실행하는 프로세서 (예를 들어, 도 3) 와 같은 소프트웨어 구성 프로세서와 전용 하드웨어의 결합에서 실행되는 소프트웨어로 컴퓨팅 디바이스에서 구현될 수 있다.
- [0071] 다양한 양태들에서 가능하게 되는 대안적인 구성들을 포함하기 위해, 방법 (900) 을 구현하는 하드웨어는 본 명세서에서 "프로세싱 디바이스" 로 지칭된다. 다양한 양태들에서, 방법 (900) 은 방법 (800) 의 블록들 (814 및 816)을 더 기술 할 수 있다.
- [0072] 블록 (902) 에서, 프로세싱 디바이스는 프로세싱 디바이스 캐시 (예를 들어, 도 3 의 프로세싱 디바이스 캐시 (308)) 및/또는 공유 메모리 (예를 들어, 도 3 의 공유 메모리 (304)) 에 대해 I/O 코히어런트 모드를 "오프" 로 설정할 수 있다.
- [0073] 대안적으로, 프로세싱 디바이스는 프로세싱 디바이스 캐시 및/또는 공유 메모리에 대해 비 I/O 코히어런트 모드를 "온" 으로 설정할 수 있다.
- [0074] 블록 (904) 에서, 하드웨어 가속기는 프로세싱 디바이스 캐시 및/또는 공유 메모리가 비 I/O 코히어런트 모드에 있는 동안 나머지 오프로드 된 작업부하를 실행할 수 있다.
- [0075] 결정 블록 (906) 에서, 프로세싱 디바이스는 나머지 오프로드 된 작업부하가 하드웨어 가속기에 의해 완료되었는지 여부를 결정할 수 있다.
- [0076] 오프로드 된 작업부하가 하드웨어 가속기에 의해 완료되지 않았다는 결정에 응답하여 (즉, 결정 블록 (906) = "아니오"), 프로세싱 디바이스는 결정 블록 (906) 에서 하드웨어 가속기에 의해 나머지 오프로드 된 작업부하가 완료되었는지 여부를 계속해서 결정할 수 있다. 오프로드 된 작업부하가 하드웨어 가속기에 의해 완료되지 않는 동안, 하드웨어 가속기는 블록 (904) 에서 나머지 오프로드 된 작업부하를 계속 실행할 수 있다. 하드웨어 가속기에 의한 나머지 오프로드된 작업부하의 실행 동안, 프로세싱 디바이스 캐시 및/또는 공유 메모리가 비 I/O 코히어런트 모드로 스위칭되었기 때문에, 나머지 작업부하는 프로세싱 디바이스와 하드웨어 가속기 간에 전송된 데이터 및 메모리 액세스 요청에 대한 코히어런시 동작을 구현하는 코히어런트 상호연결 (즉, 도 3 의 코히어런트 상호연결 (312)) 의 코히어런시 메커니즘 (예를 들어, 도 3 의 코히어런시 도메인 (314) 및/또는 스누

프 필터) 에 기인한 오버헤드를 더 이상 발생시키지 않을 수 있다.

- [0077] 오프로드 된 작업부하가 하드웨어 가속기에 의해 완료되었다는 결정에 응답하여 (즉, 결정 블록 (906) = "예"), 프로세싱 디바이스는 블록 (908) 에서 비 I/O 코어어런트 모드에 대한 임의의 나머지 캐시 유지보수 동작을 실행할 수 있다. 본 명세서에서 논의된 바와 같이, 오프로드 된 작업부하의 실행 결과를 프로세싱 디바이스에 이용 가능하게 하기 위해, 프로세싱 디바이스는 결과가 하드웨어 가속기에 의해 출력된 후 결과를 미스로서 현재 저장하고 있는 공유 메모리 (예를 들어, 도 3 의 공유 메모리 (304)) 로 프로세싱 디바이스 캐시 공간의 장래의 관독을 지시하기 위해 캐시 무효화 유지보수 동작을 구현할 수 있다.
- [0078] 캐시 무효화 유지보수 동작은 또한 오프로드 된 작업부하 결과의 데이터량 또는 오프로드 된 작업부하 결과를 저장하기 위해 프로세싱 디바이스 캐시에서 사용되는 공간에 따라 변할 수 있는 오버 헤드를 생성할 수 있다. 오버 헤드는 프로세싱 디바이스에 의한 오프로드 된 작업부하의 액세스, 사용 및/또는 출력을 위해 오프로드 된 작업부하 결과의 이용가능성을 지연시킬 수 있다.
- [0079] 블록 (910) 에서, 프로세싱 디바이스는 오프로드 된 작업부하 결과를 수신할 수 있다. 프로세싱 디바이스는 오프로드 된 작업부하의 실행 결과를 수신하여 프로세싱 디바이스 캐시에 저장할 수 있다.
- [0080] (도 1 내지 도 9 를 참조하여 상술된 양태들을 포함하지만, 이들에 제한되지 않는) 다양한 양태들은 모바일 컴퓨팅 디바이스를 포함하는 다양한 컴퓨팅 시스템에서 구현될 수 있으며, 다양한 양태와 함께 사용하기에 적합한 그 예가 도 10 에 도시된다.
- [0081] 모바일 컴퓨팅 디바이스 (1000) 는 터치 스크린 제어기 (1004) 및 내부 메모리 (1006) 에 커플링된 프로세서 (1002) 를 포함할 수 있다. 프로세서 (1002) 는 일반적인 또는 특정의 프로세싱 작업을 위해 지정된 하나 이상의 멀티 코어 집적 회로일 수 있다. 내부 메모리 (1006) 는 휘발성 또는 비 휘발성 메모리일 수 있고, 보안 및/또는 암호화된 메모리, 또는 비보안 및/또는 비 암호화된 메모리, 또는 이들의 임의의 조합일 수도 있다. 레버리징될 수 있는 메모리 유형의 예로는 DDR, LPDDR, GDDR, WIDEIO, RAM, SRAM, DRAM, P-RAM, R-RAM, M-RAM, STT-RAM 및 내장 DRAM 을 포함하지만 이들에 제한되지 않는다. 터치 스크린 제어기 (1004) 및 프로세서 (1002) 는 또한 저항성 감지 터치 스크린, 용량성 감지 터치 스크린, 적외선 감지 터치 스크린 등과 같은 터치 스크린 패널 (1012) 에 커플링될 수 있다. 또한, 컴퓨팅 디바이스 (1000) 의 디스플레이는 터치 스크린 능력을 가질 필요가 없다.
- [0082] 모바일 컴퓨팅 디바이스 (1000) 는 하나 이상의 무선 신호 송수신기 (1008) (예컨대, 피넷, 블루투스, 지그비, Wi-Fi, RF 라디오) 및 서로에 및/또는 프로세서 (1002) 에 커플링된, 통신을 송신 및 수신하기 위한 안테나 (1010) 를 가질 수 있다. 송수신기들 (1008) 및 안테나 (1010) 는 다양한 무선 전송 프로토콜 스택들 및 인터페이스들을 구현하기 위해 상술한 회로와 함께 사용될 수 있다. 모바일 컴퓨팅 디바이스 (1000) 는 셀룰러 네트워크를 통한 통신을 가능하게하고 프로세서에 커플링되는 셀룰러 네트워크 무선 모듈 칩 (1016) 을 포함할 수 있다.
- [0083] 모바일 컴퓨팅 디바이스 (1000) 는 프로세서 (1002) 에 커플링된 주변 디바이스 연결 인터페이스 (1018) 를 포함 할 수 있다. 주변 디바이스 연결 인터페이스 (1018) 는 하나의 유형의 연결을 수용하도록 단독으로 구성될 수 있거나, 범용 직렬 버스 (USB), 파이어와이어 (FireWire), 썬더 볼트 (Thunderbolt) 또는 PCIe 와 같은 공통 또는 독점적인 다양한 유형의 물리적 및 통신 연결을 수용하도록 구성될 수 있다. 또한, 주변 디바이스 연결 인터페이스 (1018) 는 유사하게 구성된 주변 장치 연결 포트 (미도시) 에 커플링될 수 있다.
- [0084] 모바일 컴퓨팅 디바이스 (1000) 는 또한 오디오 출력을 제공하기위한 스피커 (1014) 를 포함 할 수 있다. 모바일 컴퓨팅 디바이스 (1000) 는 또한 여기에 설명된 구성 요소 전부 또는 일부를 포함하기 위해 플라스틱, 금속 또는 재료의 조합으로 구성된 하우징 (1020) 을 포함 할 수 있다. 모바일 컴퓨팅 디바이스 (1000) 는 일회용 또는 재충전 가능한 배터리와 같은, 프로세서 (1002) 에 커플링된 전원 (1022) 을 포함할 수 있다. 재충전 가능한 배터리는 또한 주변 장치 연결 포트에 커플링되어 모바일 컴퓨팅 디바이스 (1000) 외부의 소스로부터 충전 전류를 수신할 수 있다. 모바일 컴퓨팅 디바이스 (1000) 는 또한 사용자 입력을 수신하기위한 물리적 버튼 (1024) 를 포함할 수 있다. 모바일 컴퓨팅 디바이스 (1000) 는 또한 모바일 컴퓨팅 디바이스 (1000) 를 턴 온 및 턴 오프하기 위한 전원 버튼 (1026) 을 포함할 수 있다.
- [0085] (도 1 내지 도 9 를 참조하여 상술된 양태들을 포함하지만, 이들에 제한되지 않는) 다양한 양태들은 랩탑 컴퓨터 (1100) 를 포함하는 다양한 컴퓨팅 시스템에서 구현될 수 있으며, 그 예가 도 11 에 도시된다.
- [0086] 많은 랩탑 컴퓨터는 컴퓨터의 포인팅 디바이스로서 기능하는 터치패드 터치 표면 (1117) 을 포함하고, 따라서

터치 스크린 디스플레이가 구비되고 위에서 설명된 컴퓨팅 디바이스 상에 구현된 것과 유사한 드래그, 스크롤링 및 플릭 제스처를 수신 할 수 있다. 랩탑 컴퓨터 (1100) 는 전형적으로 휘발성 메모리 (1112) 및 플래시 메모리의 디스크 드라이브 (1113) 와 같은 대용량 비 휘발성 메모리에 커플링된 프로세서 (1111) 를 포함할 것이다. 또한, 컴퓨터 (1100) 는 무선 데이터 링크에 연결될 수 있는 전자기 방사를 전송 및 수신하기 위한 하나 이상의 안테나 (1108) 및/또는 프로세서 (1111) 에 커플링된 셀룰러 전화 송수신기 (1116) 를 가질 수 있다. 컴퓨터 (1100) 는 또한 프로세서 (1111) 에 커플링된 플로피 디스크 드라이브 (1114) 및 콤팩트 디스크 (CD) 드라이브 (1115) 를 포함할 수 있다. 노트북 구성에서, 컴퓨터 하우징은 모두 프로세서 (1111) 에 커플링된 터치패드 (1117), 키보드 (1118) 및 디스플레이 (1119) 를 포함한다. 의 다른 구성은 잘 알려져 있는 바와 같이 (예를 들어, USB 입력을 통해) 프로세서에 커플링된 컴퓨터 마우스 또는 트랙볼을 포함할 수 있으며, 이것은 또한 다양한 양태와 관련하여 사용될 수 있다.

[0087] (도 1 내지 도 9 를 참조하여 상술된 양태들을 포함하지만, 이들에 제한되지 않는) 다양한 양태들은 다양한 상업적으로 이용 가능한 서버들 중 임의의 것과 같은 고정 컴퓨팅 시스템에서 구현될 수 있다. 예시적인 서버 (1200) 가 도 12 에 도시되어 있다.

[0088] 이러한 서버 (1200) 는 일반적으로 휘발성 메모리 (1202) 및 디스크 드라이브 (1204) 와 같은 대용량 비 휘발성 메모리에 커플링된 하나 이상의 멀티코어 프로세서 어셈블리 (1201) 를 포함한다. 도 12 에 도시 된 바와 같이, 도 12에 도시 된 바와 같이, 멀티 코어 프로세서 어셈블리 (1201) 는 어셈블리의 랙에 이들을 삽입함으로써 서버 (1200) 에 추가될 수 있다. 서버 (1200) 는 또한 프로세서 (1201) 에 커플링된 플로피 디스크 드라이브, 콤팩트 디스크 (CD) 또는 디지털 다용도 디스크 (DVD) 디스크 드라이브 (1206) 를 포함할 수 있다. 서버 (1200) 는 또한 다른 브로드캐스트 시스템 컴퓨터 및 서버에 커플링된 로컬 영역 네트워크, 인터넷, 공중 교환 전화 네트워크 및/또는 셀룰러 데이터 네트워크 (예를 들어, CDMA, TDMA, GSM, PCS, 3G, 4G, LTE, 또는 임의의 다른 유형의 셀룰러 데이터 네트워크) 와 같은 네트워크 (1205) 와의 네트워크 인터페이스 연결을 설정하기 위해 멀티 코어 프로세서 어셈블리 (1201) 에 커플링된 네트워크 액세스 포트 (1203) 를 포함할 수 있다.

[0089] 다양한 양태의 동작을 수행하기 위한 프로그램 가능한 프로세서상에서의 실행을 위한 컴퓨터 프로그램 코드 또는 "프로그램 코드" 는 C, C++, C#, 스몰토크 (Smalltalk), 자바, 자바 스크립트, 비주얼 베이직, 구조화된 (Structured) 쿼리 언어 (예, Transact-SQL), Perl 과 같은 고수준 프로그래밍 언어로, 또는 다양한 다른 프로그래밍 언어로 기록될 수 있다. 본 출원에서 사용된 바와 같은 컴퓨터 판독 가능 저장 매체 상에 저장된 프로그램 코드 또는 프로그램들은 그 포맷이 프로세서에 의해 이해 가능한 기계 언어 코드 (예를 들어, 오브젝트 코드) 를 지칭 할 수 있다.

[0090] 전문한 방법 설명 및 프로세스 흐름도들은 단지 예시적인 예로서 제공되며, 다양한 양태들의 동작들이 제시된 순서대로 수행되어야 한다는 것을 요구하거나 암시하려는 것은 아니다. 당업자에 의해 인식되는 바와 같이, 전문한 양태들에서의 동작들의 순서는 임의의 순서로 수행될 수 있다. "이후", "그 후", "다음" 등과 같은 단어는 동작 순서를 제한하려는 것이 아니고; 이 단어들은 단순히 방법의 설명을 통해 독자를 안내하는 데 사용될 뿐이다. 또한, 예를 들어, 관사 "a", "an" 또는 "the" 을 사용하여 단수로의 청구항 엘리먼트들에 대한 임의의 참조는 그 엘리먼트를 단수로 제한하는 것으로 해석되어서는 안된다.

[0091] 다양한 양태들과 관련하여 설명된 다양한 예시적인 논리 블록, 모듈, 회로, 및 알고리즘 동작들이 전자 하드웨어, 컴퓨터 소프트웨어, 또는 이 양자의 조합으로 구현될 수도 있다. 이러한 하드웨어 및 소프트웨어의 상호 교환 가능성을 명확히 설명하기 위해, 다양한 예시적인 컴포넌트, 블록, 모듈, 회로 및 동작이 일반적으로 그 기능의 관점에서 위에서 설명되었다. 그러한 기능이 하드웨어 또는 소프트웨어로 구현될지 여부는, 전체 시스템에 부과된 특정 응용 및 설계 제약에 달려 있다. 당업자는 설명된 가능성을 각각의 특정 응용에 대해 다른 방식으로 구현할 수도 있지만, 이러한 구현 결정이 본 청구범위의 범위를 벗어나게 하는 것으로 해석되지 않아야 한다.

[0092] 여기에 개시된 양태들과 관련하여 설명된 다양한 예시적인 로직들, 로지컬 블록들, 모듈, 및 회로를 구현하기 위해 사용되는 하드웨어는 범용 프로세서, 디지털 신호 프로세서 (DSP), 주문형 집적 회로 (ASIC), 필드 프로그램가능 게이트 어레이 (FPGA) 또는 다른 프로그램가능 로직 디바이스, 이산 게이트 또는 트랜지스터 로직, 이산 하드웨어 컴포넌트, 또는 여기에 설명된 기능을 수행하도록 설계된 이들의 임의의 조합으로 구현 또는 수행될 수도 있다. 범용 프로세서는 마이크로프로세서일 수도 있지만, 다르게는, 프로세서는 임의의 종래의 프로세서, 제어기, 마이크로제어기, 또는 상태 머신일 수도 있다. 또한, 프로세서는 컴퓨팅 디바이스들의 조합, 예를 들어, DSP 와 마이크로프로세서의 조합, 복수의 마이크로프로세서, DSP 코어와 결합한 하나 이상의 마이크로프로

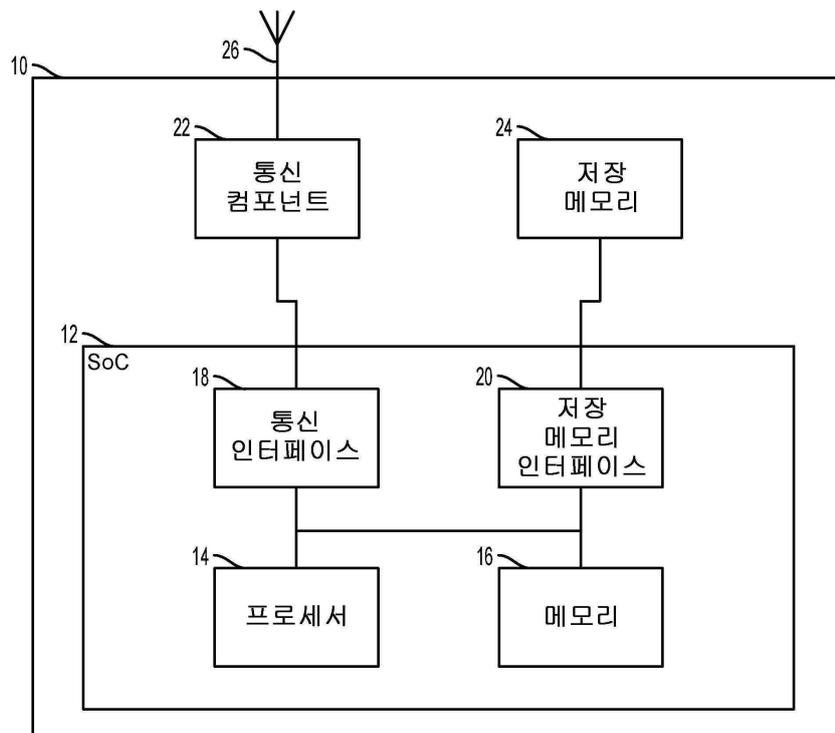
세서, 또는 임의의 다른 이러한 구성으로서 구현될 수도 있다. 대안으로, 소정의 동작 또는 방법은 주어진 기능에 특정한 회로에 의해 수행될 수 있다.

[0093] 하나 이상의 양태들에서, 설명된 기능은 하드웨어, 소프트웨어, 펌웨어 또는 이들의 임의의 조합으로 구현될 수도 있다. 소프트웨어로 구현되는 경우, 기능은 하나 이상의 명령 또는 코드로서 비 일시적인 컴퓨터 판독 가능 매체 또는 비 일시적 프로세서 판독 가능 매체 상에 저장될 수 있다. 여기에 개시된 방법 또는 알고리즘의 동작은 비 일시적인 컴퓨터 판독 가능 또는 프로세서 판독 가능 저장 매체 상에 상주할 수 있는 프로세서 실행 가능 소프트웨어 모듈로 구현될 수 있다. 비 일시적 컴퓨터 판독 가능 또는 프로세서 판독 가능 저장 매체는 컴퓨터 또는 프로세서에 의해 액세스될 수 있는 임의의 저장 매체일 수 있다. 제한이 아닌 예로서, 이러한 비 일시적 컴퓨터 판독 가능 또는 프로세서 판독가능 매체들은 RAM, ROM, EEPROM, FLASH 메모리, CD-ROM 또는 다른 광학 디스크 스토리지, 자기 디스크 스토리지, 또는 다른 자기 저장 디바이스들, 또는 명령들 또는 데이터 구조들의 형태로 회망하는 프로그램 코드를 저장하기 위해 이용될 수 있으며 컴퓨터에 의해 액세스될 수 있는 임의의 다른 매체를 포함할 수 있다. 여기에 사용된 바와 같이, 디스크 (disk) 및 디스크 (disc) 는 콤팩트 디스크 (compact disc; CD), 레이저 디스크 (laser disc), 광 디스크 (optical disc), DVD (digital versatile disc), 플로피 디스크 (floppy disk) 및 블루레이 디스크 (Blu-ray disc) 를 포함하며, 여기서, 디스크 (disk) 는 보통 데이터를 자기적으로 재생하지만, 디스크 (disc) 는 레이저를 이용하여 광학적으로 데이터를 재생한다. 또한, 상기의 조합은 비일시적 컴퓨터 판독 가능 및 프로세서 판독가능 매체의 범위 내에 포함된다. 또한, 방법 또는 알고리즘의 동작은 컴퓨터 프로그램 제품에 포함될 수도 있는 비 일시적인 프로세서 판독 가능 매체 및/또는 컴퓨터 판독 가능 매체상의 코드 및/또는 명령의 하나 또는 임의의 조합 또는 세트로서 상주할 수 있다.

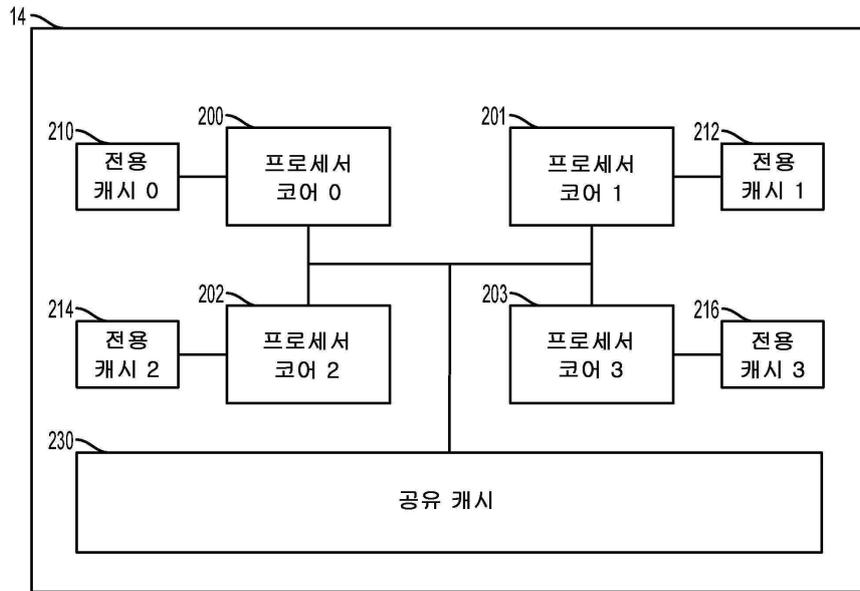
[0094] 개시된 양태의 이전의 설명은 당업자가 청구범위를 실시하거나 사용하는 것을 가능하게 하기 위해 제공된다. 이들 양태들에 대한 다양한 변경은 당업자에게는 용이하게 명백할 것이며, 여기에 정의된 일반적 원리는 청구범위의 범위를 벗어남이 없이 다른 양태들 및 구현들에 적용될 수도 있다. 따라서, 본 개시는 여기에 설명된 양태들 및 구현들에 한정되도록 의도되는 것이 아니라, 다음의 청구범위 및 여기에 개시된 원리 및 신규한 특징들에 부합하는 최광의 범위가 허여되어야 한다.

도면

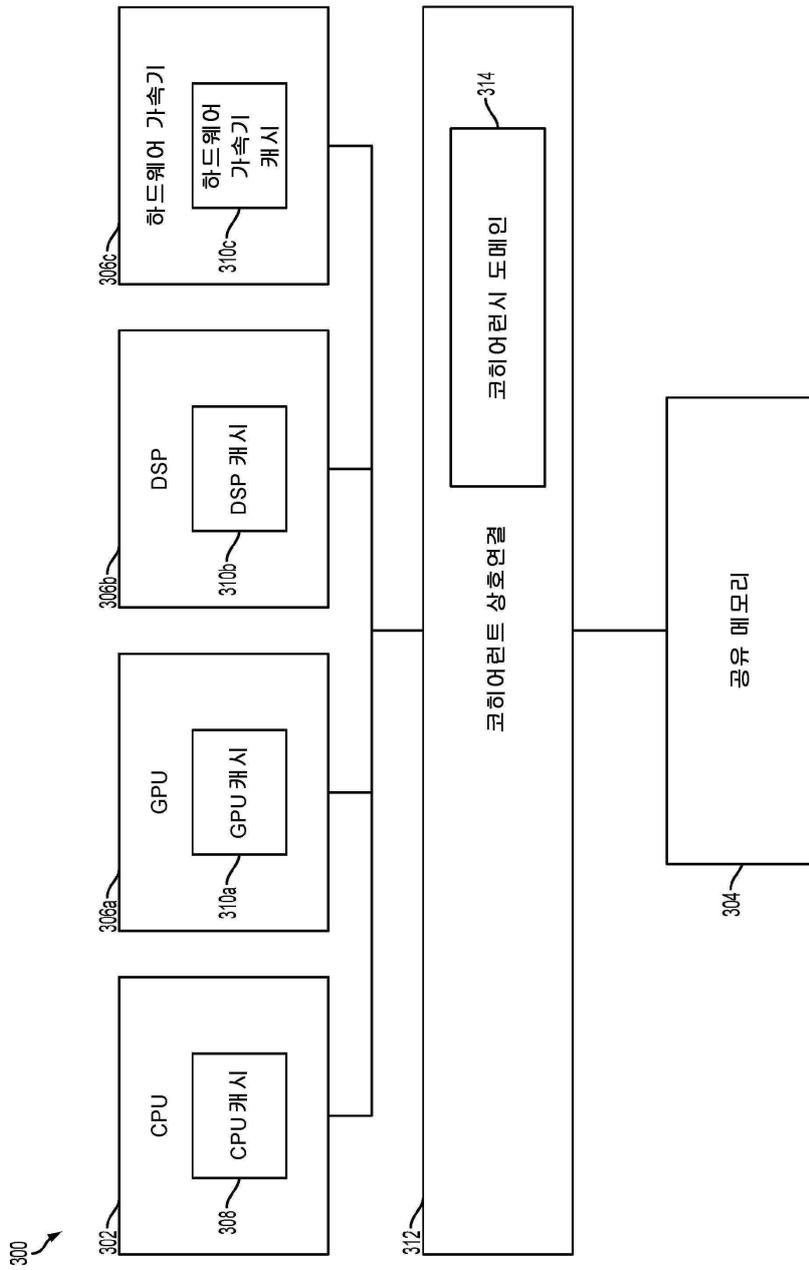
도면1



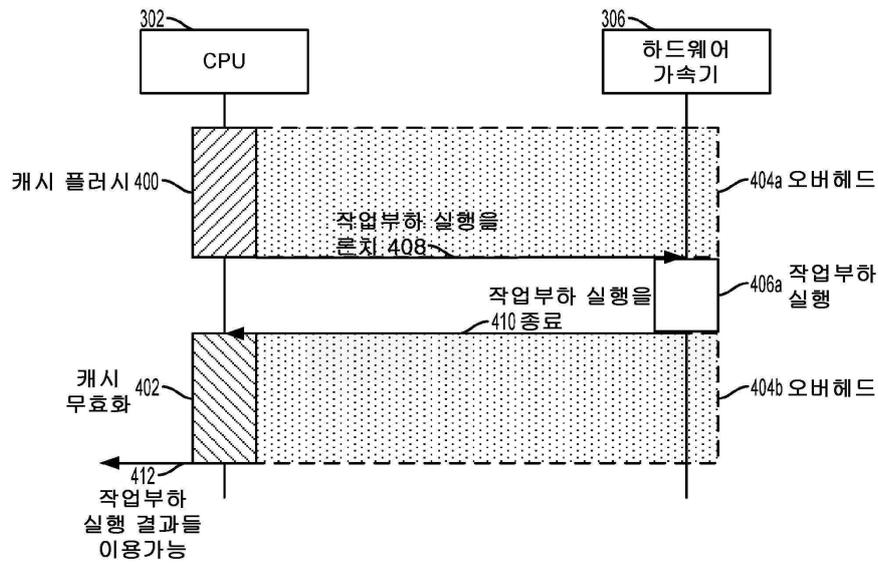
도면2



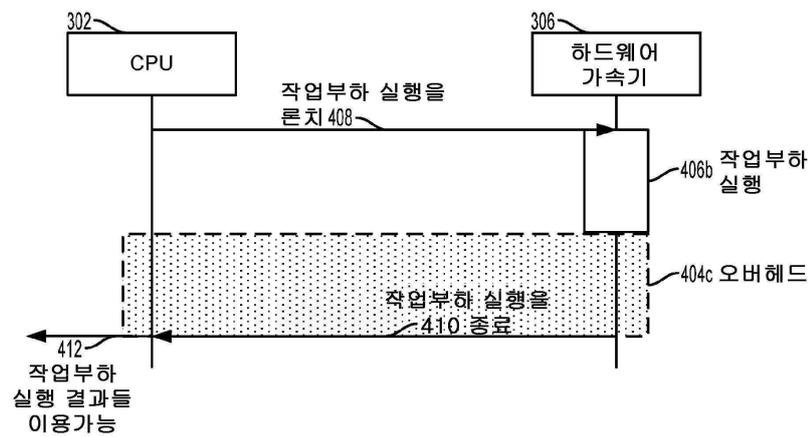
도면3



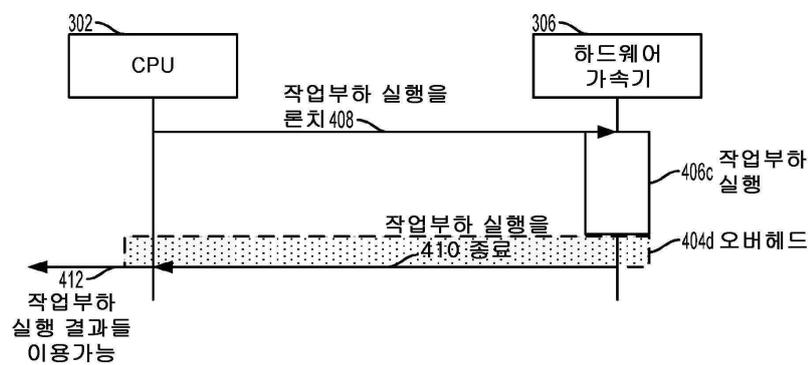
도면4



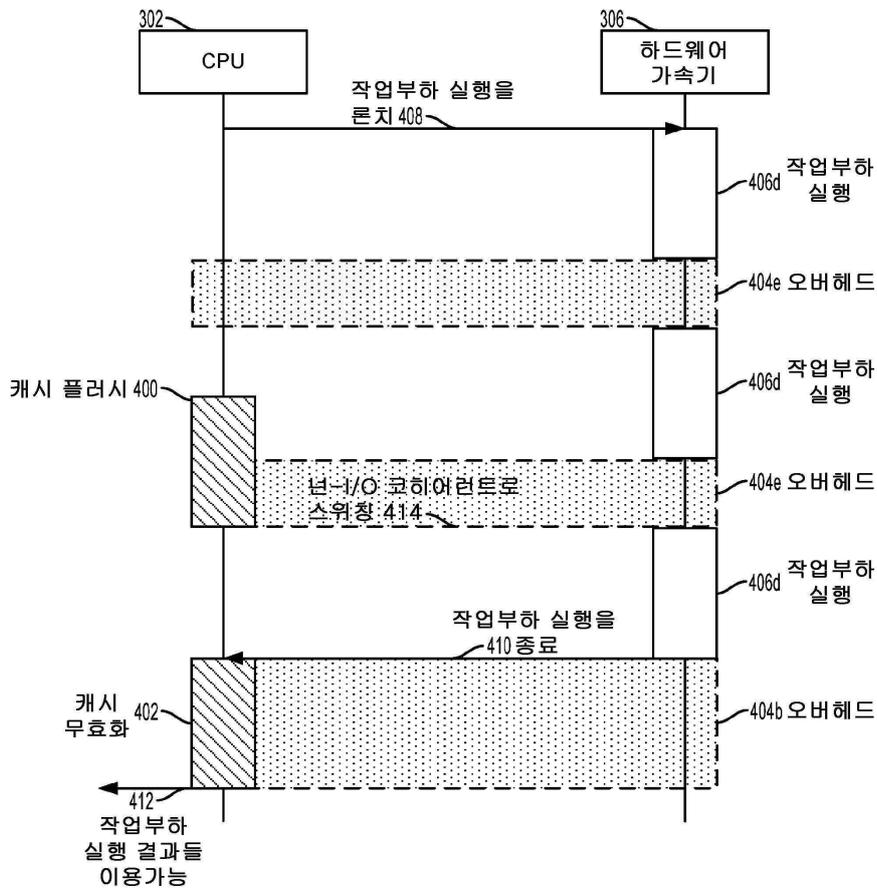
도면5



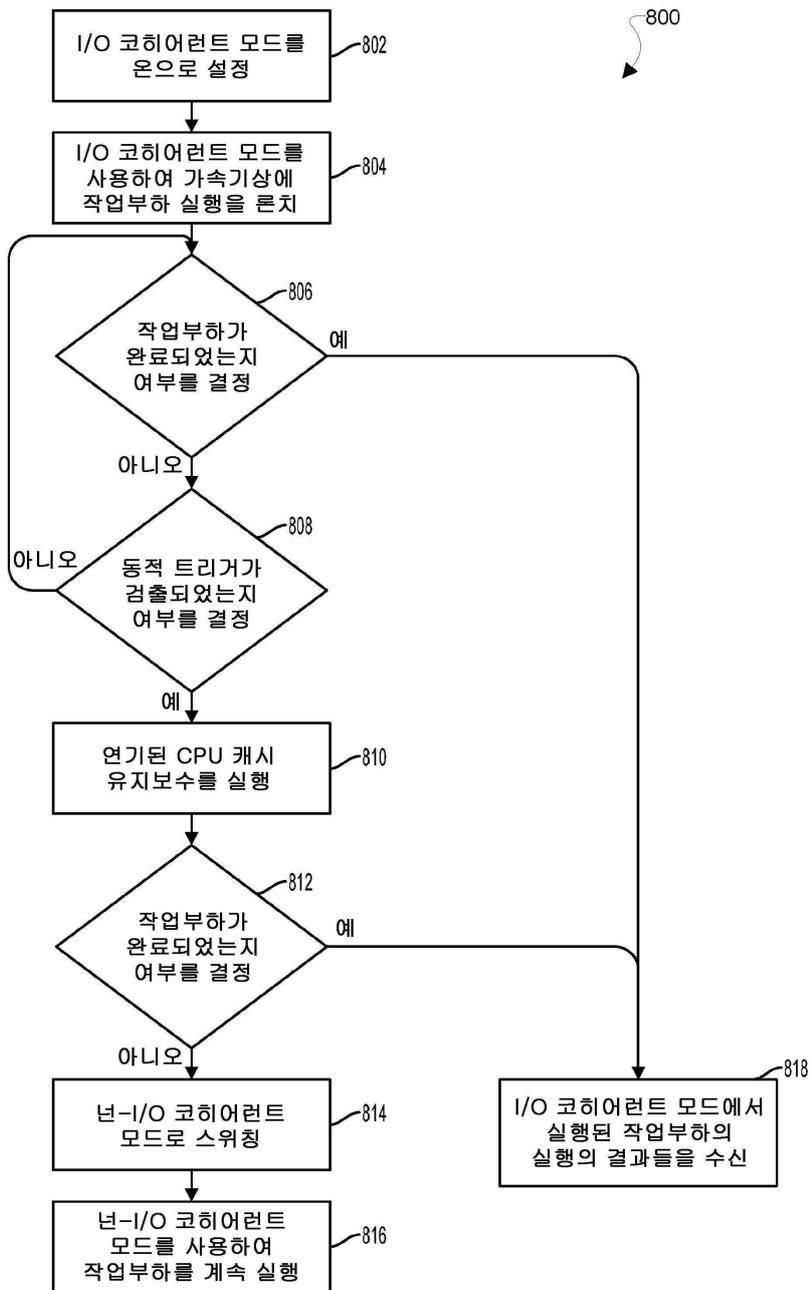
도면6



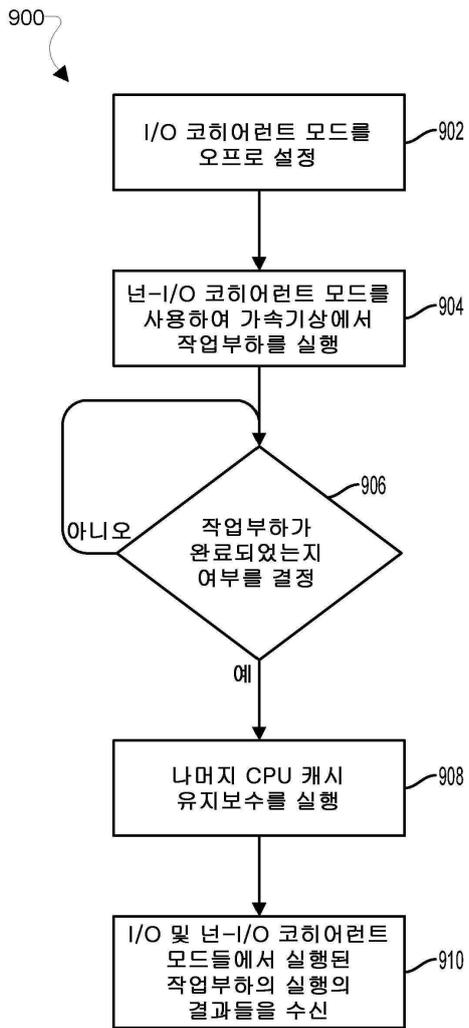
도면7



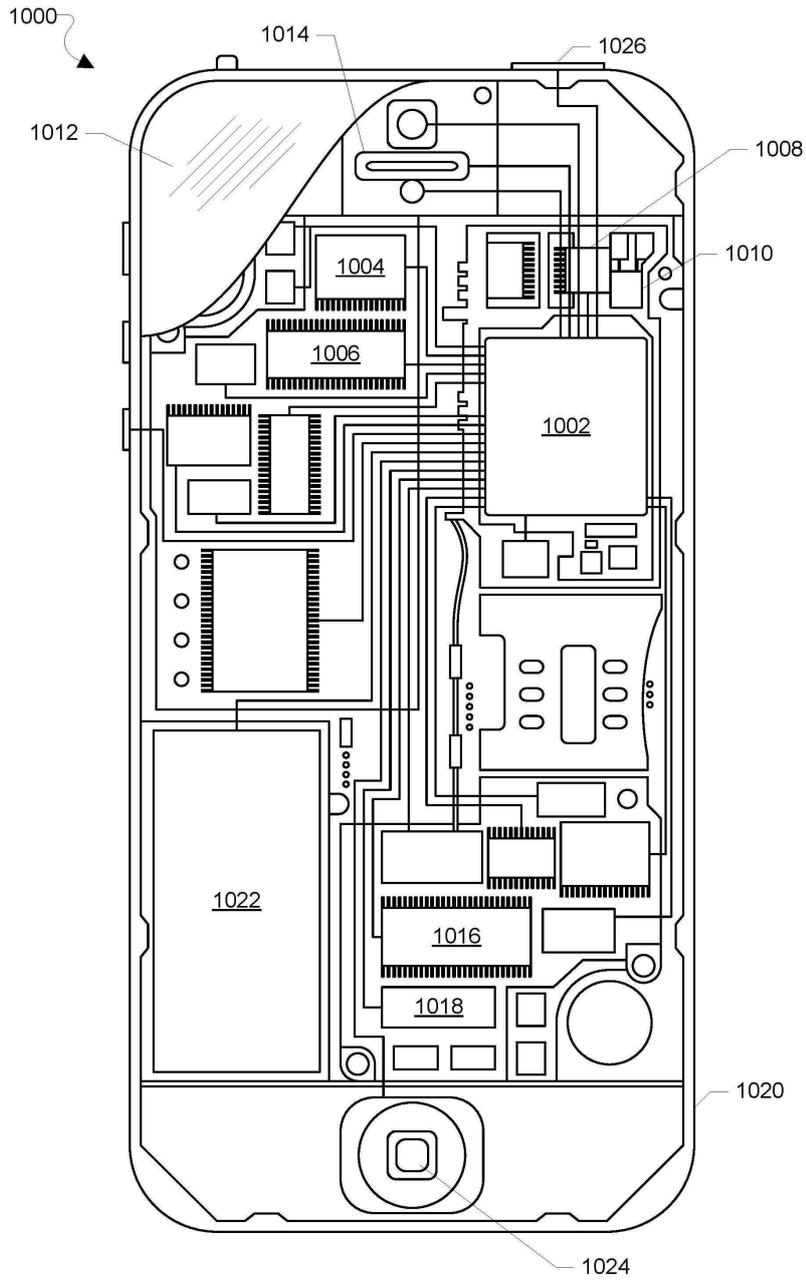
도면8



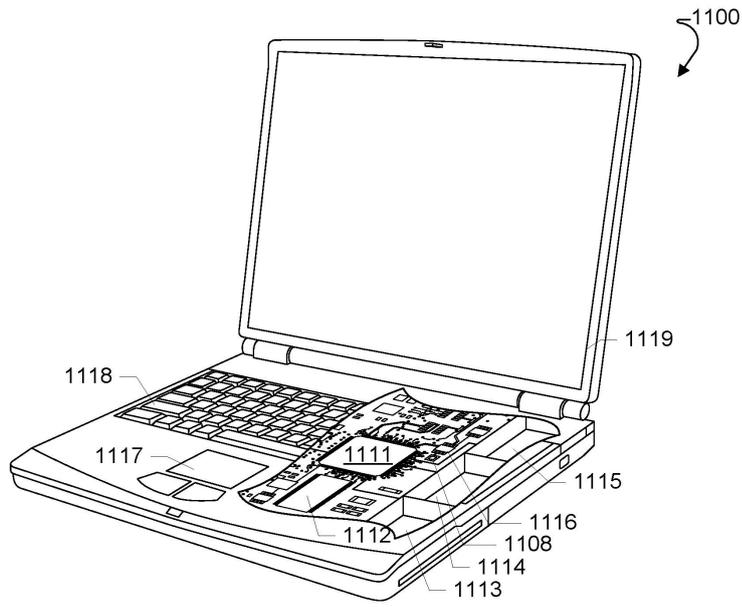
도면9



도면10



도면11



도면12

