



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0004361
(43) 공개일자 2024년01월11일

(51) 국제특허분류(Int. Cl.)
G06F 15/78 (2006.01) G06F 7/57 (2006.01)
G06F 9/38 (2024.01)
(52) CPC특허분류
G06F 15/7821 (2013.01)
G06F 7/57 (2021.08)
(21) 출원번호 10-2023-7036802
(22) 출원일자(국제) 2022년03월29일
심사청구일자 없음
(85) 번역문제출일자 2023년10월25일
(86) 국제출원번호 PCT/US2022/022357
(87) 국제공개번호 WO 2022/212383
국제공개일자 2022년10월06일
(30) 우선권주장
17/217,792 2021년03월30일 미국(US)

(71) 출원인
어드밴스드 마이크로 디바이시즈, 인코포레이티드
미국 캘리포니아 95054 산타 클라라 어거스틴 드
라이브 2485
(72) 발명자
하사안 무하마드 앰버
미국 캘리포니아 95054 산타 클라라 어거스틴 드
라이브 2485
추 마이클 엘.
미국 캘리포니아 95054 산타 클라라 어거스틴 드
라이브 2485
아지 애쉬윈
미국 캘리포니아 95054 산타 클라라 어거스틴 드
라이브 2485
(74) 대리인
박장원

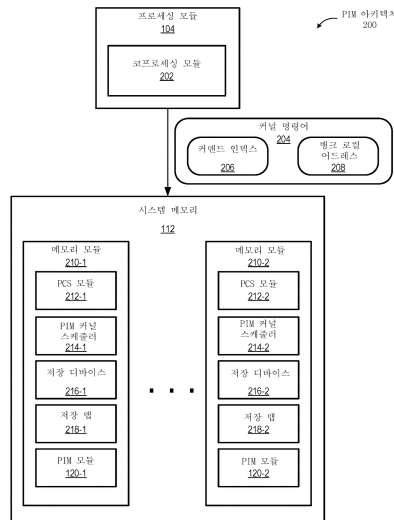
전체 청구항 수 : 총 16 항

(54) 발명의 명칭 프로세싱-인-메모리 동시적 프로세싱 시스템 및 방법

(57) 요약

프로세싱 시스템(100)은 프로세싱 모듈(104, 106) 및 메모리 디바이스(112)를 포함한다. 메모리 디바이스는 프로세싱 모듈을 대신하여 프로세싱 동작을 수행하는 프로세싱-인-메모리(PIM) 모듈(120)을 포함한다. PIM 모듈의 명령어 세트 아키텍처(instruction set architecture; ISA)는 프로세싱 모듈의 ISA보다 더 적은 수의 명령어를 갖는다. 프로세싱 모듈로부터 수신된 명령어는, PIM 모듈의 프로세싱 자원이 가상화되도록 변환된다. 결과, PIM 모듈은 프로세싱 모듈의 다수의 스레드 또는 애플리케이션에 대한 프로세싱 동작을 동시에 수행한다.

대표도 - 도2



(52) CPC특허분류

G06F 9/3877 (2013.01)

Y02D 10/00 (2020.08)

명세서

청구범위

청구항 1

메모리 디바이스로서,

프로세싱 유닛에서 실행되는 프로그램을 대신하여 데이터를 저장하도록 구성된 복수의 저장 디바이스 - 상기 프로세싱 유닛은 상기 메모리 디바이스의 외부에 있음 -; 및

상기 프로세싱 유닛으로부터의 상기 프로그램에 대응하는 커널 명령어에 응답하여 프로세싱-인-메모리 (processing-in-memory; PIM) 커맨드를 프로세싱하도록 구성된 PIM 모듈을 포함하며, 상기 PIM 커맨드를 포함하고 상기 PIM 모듈에 의해 구현되는 명령어 세트 아키텍처(instruction set architecture; ISA)는 상기 프로세싱 유닛에 의해 구현되는 ISA보다 더 적은 수의 명령어를 가지며, 상기 PIM 모듈의 프로세싱 자원은, 상기 PIM 모듈이 다수의 PIM 커널로부터의 PIM 커맨드를 동시에 프로세싱하게끔 가상화되도록 구성되는, 메모리 디바이스.

청구항 2

제1항에 있어서,

상기 PIM 커맨드를 저장하고 상기 PIM 커맨드를 상기 PIM 모듈에 제공하도록 구성된 PCS 커맨드 저장(PIM command store; PCS) 모듈을 추가로 포함하고, 상기 PIM 모듈의 상기 프로세싱 자원은 상기 PCS 모듈을 포함하는, 메모리 디바이스.

청구항 3

제2항에 있어서,

적어도 2개의 PIM 커널에 대응하는 적어도 2개의 커널 명령어의 부분들을 조합하고 상기 PCS 모듈에서 상기 적어도 2개의 커널 명령어에 대응하는 단일 PIM 커맨드를 저장하도록 구성된 PIM 커널 스케줄러를 추가로 포함하는, 메모리 디바이스.

청구항 4

제3항에 있어서,

상기 PIM 커널 스케줄러는 조합 메트릭(combination metric)에 기초하여 상기 적어도 2개의 커널 명령어를 조합하도록 추가로 구성되는, 메모리 디바이스.

청구항 5

제4항에 있어서,

상기 조합 메트릭은, 상기 PIM 모듈에서 실행을 기다리는 PIM 커널들 간의 각각의 커널 명령어의 공통성; 대응하는 PIM 커널의 예상되는 자원 사용량에 기초한 커널 명령어의 가중화된 조합; 또는 상기 프로세싱 유닛으로부터 수신된 상기 PIM 커널의 종속성 그래프 중 적어도 하나에 기초하는, 메모리 디바이스.

청구항 6

제3항에 있어서,

상기 복수의 저장 디바이스는 레지스터 뱅크로서 구성된 레지스터 파일을 추가로 포함하고;

상기 PIM 커널 스케줄러는 상기 커널 명령어에 의해 표시된 가상 레지스터 어드레스를 상기 레지스터 파일의 물리적 레지스터 어드레스로 변환하도록 추가로 구성되는, 메모리 디바이스.

청구항 7

제3항에 있어서,

상기 복수의 저장 디바이스는 레지스터 뱅크로서 구성된 레지스터 파일을 추가로 포함하고;

상기 PIM 모듈의 상기 프로세싱 자원은 상기 레지스터 파일을 포함하고, 상기 레지스터 파일의 적어도 일부 레지스터는 상기 적어도 2개의 PIM 커널 중의 제1 PIM 커널에 할당되도록 구성되는 반면, 상기 레지스터 파일의 상기 레지스터 중 적어도 일부는 상기 적어도 2개의 PIM 커널 중의 제2 PIM 커널에 할당되는, 메모리 디바이스.

청구항 8

제3항에 있어서,

상기 PIM 커널 스케줄러는 상기 프로세싱 유닛으로부터의 상기 커널 명령어에 기초하여, 상기 PIM 모듈에서의 실행을 위해 상기 PIM 커맨드를 스케줄링하도록 추가로 구성되는, 메모리 디바이스.

청구항 9

방법으로서,

프로세싱 유닛으로부터 프로세싱-인-메모리 (PIM) 산술 논리 유닛(arithmetic logic unit; ALU)에서, 상기 PIM ALU에 의해 동시에 프로세싱될 복수의 커널에 대한 커널 명령어를 수신하는 단계 - 상기 PIM ALU에 의해 구현되는 명령어 세트 아키텍처(ISA)는 상기 프로세싱 유닛에 의해 구현되는 ISA보다 더 적은 수의 명령어를 가짐 -;

PIM 커맨드 저장(PCS) 모듈로부터 상기 커널 명령어에 대응하는 PIM ALU 커맨드를 검색(retrieve)하는 단계 - 적어도 하나의 PIM 명령어는 적어도 2개의 커널에 대응함 -; 및

상기 PIM ALU를 포함하는 메모리의 레지스터 뱅크 내에 상기 커널 중 적어도 2개에 대한 데이터를 저장하는 것을 포함하는, 상기 PIM ALU 커맨드를 수행하는 단계를 포함하는, 방법.

청구항 10

제9항에 있어서,

상기 커널 명령어는 각각의 PIM ALU 커맨드 및 적어도 하나의 각각의 뱅크 로컬 어드레스를 나타내는 각각의 커맨드 인덱스를 포함하는, 방법.

청구항 11

제9항에 있어서,

상기 PCS 모듈로부터 상기 PIM ALU 커맨드를 검색하는 단계 전에, 상기 커널 명령어에 의해 표시된 상기 PIM ALU 커맨드 중 적어도 2개가 동일한 동작을 요청한다고 검출하는 것에 응답하여, 상기 적어도 2개의 PIM ALU 커맨드의 표시를 상기 PCS 모듈의 단일 엔트리에 저장하는 단계를 추가로 포함하는, 방법.

청구항 12

제9항에 있어서,

상기 PIM ALU 커맨드를 검색하는 단계는 대응하는 PIM 커널의 종속성에 기초하여 상기 PIM ALU 커맨드 중 적어도 하나의 실행을 지연시키는 단계를 포함하는, 방법.

청구항 13

시스템으로서,

복수의 프로그램을 실행하도록 구성된 프로세싱 유닛; 및

상기 프로세싱 유닛으로부터 커널 명령어를 수신하도록 구성된 메모리 디바이스를 포함하고, 상기 커널 명령어는 상기 메모리 디바이스의 프로세싱-인-메모리(PIM) 모듈로 하여금 복수의 PIM 커널에 대한 PIM 커맨드를 동시에 프로세싱하게 하고, PIM 커맨드를 동시에 프로세싱하는 것은 가상 레지스터 뱅크로서 작동하도록 구성된 상기 메모리 디바이스의 복수의 저장 디바이스에 상기 PIM 커널 중 적어도 일부에 대한 데이터를 저장하는 것을 포함하며, 상기 PIM 커맨드를 포함하고 상기 PIM 모듈에 의해 구현되는 명령어 세트 아키텍처(ISA)는 상기 프로

세싱 유닛에 의해 구현되는 ISA보다 더 적은 수의 명령어를 갖는, 시스템.

청구항 14

제13항에 있어서,

상기 메모리 디바이스는 상기 PIM 커맨드를 저장하고 상기 PIM 커맨드를 상기 PIM 모듈에 제공하도록 구성된 PIM 커맨드 저장(PCS) 모듈을 포함하고,

상기 프로세싱 유닛은, 애플리케이션 어드레스를 PCS 어드레스로 변환하고 상기 PCS 어드레스를 상기 커널 명령어의 일부로서 전송하는 PIM 커널 스케줄러의 일부분을 포함하는, 시스템.

청구항 15

제13항에 있어서,

상기 메모리 디바이스는 제2 복수의 PIM 커널에 대한 제2 PIM 커맨드를 동시에 프로세싱하도록 구성된 제2 PIM 모듈을 포함하는, 시스템.

청구항 16

제13항에 있어서,

상기 프로세싱 유닛은 상기 커널 명령어를 상기 메모리 디바이스에 전송하도록 구성된 코프로세서를 포함하는, 시스템.

발명의 설명

기술 분야

배경 기술

[0001]

프로세싱 시스템은 종종, 프로세서(예를 들어, 중앙처리장치(central processing unit; CPU) 또는 그래픽 처리 장치(graphics processing unit; GPU))에 의해 비교적 신속하게 액세스될 수 있는 로컬 메모리 자원 및 액세스 하기에 비교적 더 긴 시간량을 취하는 외부 메모리 자원을 포함한다. 일부 경우에서, 외부 메모리 자원은 프로세서로부터 상이한 칩 상에 위치된다. 그러나, 일부 경우에서, 더 긴 액세스 시간으로 인해, 외부 메모리 자원에 저장된 데이터를 수반하는 명령어는 프로세싱하기에 바람직하지 않게 긴 시간을 취한다. 추가적으로, 일부 경우에서, 외부 메모리 자원과 프로세서 사이의 데이터 통신은 바람직하지 않은 양의 상호접속 용량을 소비한다. 일부 경우에, 외부 메모리 자원에 대한 액세스의 수를 감소시키기 위해, 프로세싱-인-메모리(processing-in-memory; PIM) 회로가, 데이터를 프로세서로 가져오기보다는, 외부 메모리 자원에서 데이터를 프로세싱하는 데 사용된다. PIM 회로는 더 일반적인 프로세서의 전력 소비 또는 영역 사용량 없이 외부 메모리 자원에 저장된 데이터에 대해 제한된 프로세싱 애플리케이션을 수행한다. 그 결과, 일부 경우에서, 프로세서와 외부 메모리 자원 사이의 통신의 수가 감소되고, 외부 메모리 자원에 저장된 데이터를 수반하는 일부 명령어의 프로세싱 시간이 감소된다. 그러나, 이러한 제한된 아키텍처로 인해, 다수의 커널을 동시에 프로세싱하기 위한 하드웨어 병렬성의 이점을 취하는 것이 어렵다. 예를 들어, 가상화의 결여로 인해, PIM 회로는 한 번에 하나의 커널의 커맨드만을 로드(load)하고, 일부 경우에서 데이터 종속성으로 인해 직렬화 병목을 생성한다.

발명의 내용

[0002]

하나의 일반적인 양태에서, 메모리 디바이스는, 프로세싱 유닛에서 실행되는 프로그램을 대신하여 데이터를 저장하도록 구성된 복수의 저장 디바이스 - 프로세싱 유닛은 메모리 디바이스의 외부에 있음 -; 및 프로세싱 유닛으로부터의 프로그램에 대응하는 커널 명령어에 응답하여 프로세싱-인-메모리(PIM) 커맨드를 프로세싱하도록 구성된 PIM 모듈을 포함하며, PIM 커맨드를 포함하고 PIM 모듈에 의해 구현되는 명령어 세트 아키텍처(instruction set architecture; ISA)는 프로세싱 유닛에 의해 구현되는 ISA보다 더 적은 수의 명령어를 가지며, PIM 모듈의 프로세싱 자원은, PIM 모듈이 다수의 PIM 커널로부터의 PIM 커맨드를 동시에 프로세싱하게끔 가상화되도록 구성된다.

[0003] 또 다른 일반적인 양태에 따라, 방법은 프로세싱 유닛으로부터 프로세싱-인-메모리(PIM) 산술 논리 유닛(arithmetic logic unit; ALU)에서, PIM ALU에 의해 동시에 프로세싱될 복수의 커널에 대한 커널 명령어를 수신하는 단계를 포함하고, PIM ALU에 의해 구현되는 명령어 세트 아키텍처(ISA)는 프로세싱 유닛에 의해 구현되는 ISA보다 더 적은 수의 명령어를 가진다. 본 방법은 PIM 커맨드 저장(PIM command store; PCS) 모듈로부터 커널 명령어에 대응하는 PIM ALU 커맨드를 검색(retrieve)하는 단계를 추가로 포함하고, 적어도 하나의 PIM 명령어는 적어도 2개의 커널에 대응한다. 본 방법은 또한 PIM ALU를 포함하는 메모리의 레지스터 뱅크 내에 커널 중 적어도 2개에 대한 데이터를 저장하는 것을 포함하는, PIM ALU 커맨드를 수행하는 단계를 포함한다.

[0004] 또 다른 일반적인 양태에 따르면, 시스템은 복수의 프로그램을 실행하도록 구성된 프로세싱 유닛을 포함한다. 본 시스템은 프로세싱 유닛으로부터 커널 명령어를 수신하도록 구성된 메모리 디바이스를 추가로 포함하고, 커널 명령어는 메모리 디바이스의 프로세싱-인-메모리(PIM) 모듈로 하여금 복수의 PIM 커널에 대한 PIM 커맨드를 동시에 프로세싱하게 하고, PIM 커맨드를 동시에 프로세싱하는 것은 가상 레지스터 뱅크로서 작동하도록 구성된 메모리 디바이스의 복수의 저장 디바이스에 PIM 커널 중 적어도 일부에 대한 데이터를 저장하는 것을 포함하며, PIM 커맨드를 포함하고 PIM 모듈에 의해 구현되는 명령어 세트 아키텍처(ISA)는 프로세싱 유닛에 의해 구현되는 ISA보다 더 적은 수의 명령어를 갖는다.

도면의 간단한 설명

[0005] 본 개시내용은 첨부 도면을 참조하여 더 잘 이해될 수 있고, 그것의 많은 특징 및 이점이 당업자에게 분명해진다. 상이한 도면에서의 동일한 참조 부호의 사용은 유사하거나 동일한 항목을 나타낸다.

도 1은 일부 실시예에 따른 프로세싱 시스템의 PIM 동시적 프로세싱(concurrent processing) 시스템의 블록도이다.

도 2는 일부 실시예에 따른 PIM 아키텍처의 블록도이다.

도 3은 일부 실시예에 따른 예시적인 가상-대-물리적 커널 레지스터 매핑(virtual-to-physical kernel register mapping)의 블록도이다.

도 4는 일부 실시예에 따른 예시적인 커널 명령어 병합 및 매핑의 블록도이다.

도 5는 일부 실시예에 따른 PIM 동시적 프로세싱 시스템을 사용하여 다수의 커널에 대한 커널 명령어를 동시에 프로세싱하는 방법의 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0006] 프로세싱 시스템은 프로세싱 유닛 및 프로세싱 유닛 외부의 메모리 디바이스를 포함한다. 메모리 디바이스는 프로세싱 시스템을 대신하여 데이터를 프로세싱하는 프로세싱-인-메모리(PIM) 모듈을 포함한다. PIM 모듈의 명령어 세트 아키텍처(ISA)는 프로세싱 유닛의 ISA보다 작아서, 일부 경우에서, PIM 커맨드는 프로세싱 유닛의 ISA의 명령어의 서브세트가 된다. 예를 들어, 일부 경우에, PIM 모듈은 산술 논리 유닛(ALU)이다. 그 결과, GPU와 같은 또 다른 프로세서로 명령어를 전송하는 CPU와 같은 프로세서와 비교하여, 추가 명령어가 PIM 모듈로 전송된다. 일부 실시예에서, 추가 명령어는 수행될 PIM 커맨드를 나타내는 커널 명령어로서 전송된다. 커널 명령어에 응답하여, PIM 모듈은 메모리 디바이스에 저장된 데이터를 사용하여 PIM 커맨드를 프로세싱한다.

[0007] 일부 경우에, PIM 모듈은 다수의 애플리케이션의 PIM 커맨드, 또는 동일한 애플리케이션의 상이한 스레드를 동시에 프로세싱한다. 어드레스 가상화는 일반적으로 다수의 애플리케이션이 단일 어드레스를 사용하는 것으로 인한 에러를 방지하고, 애플리케이션이 물리적 어드레스의 사용으로 인해 다른 애플리케이션에 관한 정보를 수신하는 것으로 인한 잠재적인 보안 문제를 감소시키는 데 사용된다. 그러나, 어드레스 가상화는 프로세싱 유닛과 비교하여 PIM 모듈의 상대적인 단순성으로 인해 일부 PIM 모듈에서 어렵다. 그 결과, 그러한 PIM 모듈은 직렬로 자원(예를 들어, PIM 모듈 및 레지스터 파일과 같은 연관된 저장 디바이스)에 대한 액세스를 제공한다. 일부 실시예에서, 본원의 메모리 디바이스는 PIM 모듈을 대신하여 가상-대-물리적 레지스터 매핑을 수행하는 PIM 커널 스케줄러를 추가로 포함하여, 메모리 디바이스에서 레지스터 어드레스 가상화를 가능하게 하고, PIM 모듈이 다수의 커널을 동시에 프로세싱할 수 있게 한다.

[0008] PIM 커맨드는 PIM 커맨드 저장(PCS) 모듈에 저장된다. 그러나, 일부 실시예에서, PCS 모듈 내의 저장 공간이 제한된다. 그 결과, 일부 경우에서, PIM 커널 스케줄러는 유입 커널 명령어를 추가로 분석하여 명령어의 다수의 세트가 동일한 동작을 요청하는지 여부를 식별한다. 동일한 동작이 요청된다고 검출하는 것에 응답하여, PIM 커

널 스케줄러는 커널 명령어를 조합하여, PCS 모듈이 단일 PIM 커맨드만을 저장하게 한다. 그 결과, PCS 모듈에 의한 자원 활용이 증가된다. 또한, 일부 실시예에서, PCS 어드레스는 다수의 커널의 명령어가 동시에 PCS 모듈에 저장되도록 가상화된다. 따라서, 일부 경우에서, 다수의 커널에 의해 동일한 동작이 요청된다고 검출하는 것에 응답하여, PIM 커널 스케줄러는 다수의 커널로부터의 커널 명령어를 조합한다.

[0009] 본원에 사용되는 바와 같이, "동시에 프로세싱하는 것"은 시작 후에 하지만 또 다른 애플리케이션 또는 커널의 일부를 완료하기 전에 애플리케이션 또는 커널의 적어도 일부를 실행하는 것을 지칭한다. 예를 들어, 2개의 커널의 일부가 인터리빙된 방식으로 실행되어 제2 커널의 일부가 제1 커널의 제1 부분에 후속하여 그러나 제1 커널의 제2 부분 이전에 실행되는 경우, 제1 커널 및 제2 커널은 "동시에 프로세싱되는 것"으로 간주된다. 또 다른 예로서, 제1 커널의 일부가 파이프라인의 제1 부분에 있고 제2 커널의 일부가 파이프라인의 제2 부분에 있는 파이프라인된 프로세서는, 2개의 커널을 "동시에 프로세싱하는 것"으로 간주된다. 2개의 애플리케이션 또는 커널이 "동시에 프로세싱되는 것"으로 간주되기 위해 정확히 동일한 하드웨어 디바이스에 의해 정확히 동일한 시간에 프로세싱될 필요는 없다.

[0010] 본 명세서에 설명된 기술은, 상이한 실시예에서, 다양한 병렬 프로세서(예를 들어, 벡터 프로세서, 그래픽 처리 장치(graphics processing unit; GPU), 범용 GPU(general-purpose GPU; GPGPU), 논-스칼라 프로세서, 고정밀 프로세서, 인공 지능(artificial intelligence; AI) 프로세서, 추론 엔진, 머신 러닝 프로세서, 다른 멀티스레드 프로세싱 유닛 등) 중 임의의 것을 사용하여 이용된다. 예시의 용이함을 위해, 프로세싱 모듈이 이용되는 예시적인 시스템 및 방법에 대한 참조가 본원에서 이루어진다. 그러나, 본원에 설명된 시스템 및 기술은 달리 언급되지 않는 한 다른 유형의 병렬 프로세서의 사용에 동등하게 적용된다는 것이 이해될 것이다.

[0011] 도 1은 적어도 일부 실시예에 따라 PIM 모듈에서 애플리케이션을 동시에 프로세싱하는 프로세싱 시스템(100)을 예시한다. 프로세싱 시스템(100)은 프로세싱 모듈(104 내지 106), 하나 이상의 메모리 제어기(108), 및 하나 이상의 I/O 허브(110)와 같은 복수의 프로세싱 모듈을 포함하여, 프로세싱 시스템(100)의 다양한 구성 요소를 상호접속하기 위해 사용되는 데이터 패브릭(102)을 포함한다. 각 메모리 제어기(108)는 차례로 시스템 메모리(112)에 결합되고, 각 I/O 허브(110)는 차례로 하나 이상의 I/O 디바이스, 이를테면 I/O 디바이스(114, 115)에 결합된다. 적어도 하나의 시스템 메모리(112)는 PIM 모듈(120)을 포함한다.

[0012] 프로세싱 모듈(104 내지 106)은 하나 이상의 프로세서 코어 및 로컬 캐시 계층구조를 포함한다. 프로세서 코어는 예를 들어, CPU 코어, GPU 코어, DSP 코어, 병렬 프로세서 코어, 또는 이들의 조합을 포함한다. 프로세싱 클라이언트의 로컬 캐시 계층구조는 하나 이상의 레벨의 캐시를 포함한다. 일부 실시예에서, 프로세싱 모듈(104 내지 106) 중 적어도 하나는 프로세싱 모듈(104 내지 106) 중 적어도 다른 하나와 다르다(즉, 프로세싱 모듈(104 내지 106)은 이종이다). 다른 실시예에서, 프로세싱 모듈(104 내지 106)은 동종이다. 일부 경우에서, 도 2를 참조하여 아래에서 추가로 논의되는 바와 같이, 프로세싱 모듈(104 내지 106) 중 하나 이상은 프로세싱 모듈(104 내지 106)을 대신하여 작업(task)을 수행하는 하나 이상의 코프로세서를 포함한다.

[0013] 메모리 제어기(108)는 대응하는 시스템 메모리(112)와 프로세싱 시스템(100)의 다른 구성 요소 사이의 인터페이스로서 동작한다. 이에 따라, 프로세싱 클라이언트의 로컬 캐시 계층구조에 캐싱될 데이터는 통상적으로 "캐시 라인"이라고 지칭되는 데이터의 블록으로서 조작되고, 이 캐시 라인은 시스템 메모리(112)의 물리적 어드레스를 사용하여 메모리 계층구조에서 어드레싱되거나 달리 위치된다. 캐시 라인은 프로세싱 클라이언트로부터의 액세스 요청에 응답하여 메모리 제어기(108)에 의해 시스템 메모리(112)로부터 액세스되고, 캐시 라인은 프로세싱 클라이언트의 하나 이상의 캐시에 설치, 또는 캐싱된다. 마찬가지로, 수정된 데이터를 포함하는 캐시 라인이 프로세싱 클라이언트의 로컬 캐시 계층구조로부터 축출(evict)되고, 이에 따라 시스템 메모리(112)에서 갱신될 필요가 있을 때, 메모리 제어기(108)는 이러한 후기록(write-back) 프로세스를 관리한다. 또한, 도 2 내지 도 5를 참조하여 아래에서 논의되는 바와 같이, 일부 경우에, PIM 모듈(120)은 프로세싱 모듈(104 내지 106) 중 하나 이상을 대신하여 데이터를 프로세싱한다. 일부 실시예에서, 프로세싱 시스템(100)은 다수의 시스템 메모리(112)를 포함한다. 추가적으로, 일부 실시예에서, 시스템 메모리(112)는 다수의 PIM 모듈(120)을 포함한다.

[0014] I/O 디바이스(114, 115)는 DMA 액세스 동작을 사용하여 프로세싱 시스템(100) 내외로 데이터를 전송하도록 동작한다. 예를 들어, I/O 디바이스(114, 115) 중 하나는 데이터를 수신 및 송신하기 위해 노드를 네트워크에 접속하기 위한 네트워크 인터페이스 카드(network interface card; NIC), 또는 하드 디스크 드라이브(hard disk drive; HDD) 또는 프로세싱 모듈(104 내지 106)에 의한 사용을 위해 비교적 많은 양의 데이터의 비휘발성 저장을 위한 다른 대용량 저장 디바이스 등을 포함할 수 있다. 적어도 일 실시예에서, I/O 허브(110)는 I/O 디바이스(114, 115)를 관리하고, 데이터 패브릭(102)과 I/O 디바이스(114, 115) 사이의 인터페이스로서 기능한다. 예

시하자면, 일부 실시예에서, I/O 허브(110)는 I/O 디바이스(114, 115)와 데이터 패브릭(102) 사이의 PCIe(Peripheral Component Interconnect Express) 상호접속부로서 동작하도록 PCIe 루트 콤플렉스(root complex)를 포함한다.

[0015] 데이터 패브릭(102)은 일반적으로 커맨드, 데이터, 요청, 상황 통신 정보, 및 다른 시그널링을 프로세싱 시스템(100)의 다른 구성 요소들 사이에서, 그리고 프로세싱 시스템(100)과 다른 노드(141) 사이에서 전송한다. 이들 전송 동작의 하나의 이러한 서브세트는 하나 이상의 프로세싱 모듈(104 내지 106)에 의한 사용을 위해 I/O 디바이스(114, 115)에 의해 시스템 메모리(112)에서 제공되는 데이터의 저장이다. 도 2 내지 도 5와 관련하여 추가로 논의되는 바와 같이, 일부 경우에, 시스템 메모리(112)에 저장된 데이터는 데이터 패브릭(102)을 통해 프로세싱 모듈(104 내지 106) 중 하나 이상으로부터 전송된 커널 명령어에 응답하여 PIM 모듈(120)을 사용하여 프로세싱된다. I/O 에이전트(140)는 I/O 허브(110) 및 I/O 디바이스(114, 115)에 대한 코히어런트 에이전트(coherent agent)로서 동작한다. 또한, 일부 실시예에서, 전송 계층(130)은 하나 이상의 브리지 구성 요소 또는 코히어런트 에이전트(도시되지 않음)를 통해 하나 이상의 다른 노드(141)의 대응 전송 계층에 또는 프로세싱 모듈(104 내지 106)에 결합된다. 다양한 실시예에서, 데이터 패브릭(102)은 HyperTransport(TM) 사양 또는 Infinity Fabric(TM) 사양과 같은 하나 이상의 표준화된 상호접속부 사양과 호환 가능하다.

[0016] 도 2는 일부 실시예에 따른 프로세싱 시스템(100)의 PIM 아키텍처(200)의 블록도를 예시하는 블록도이다. 다른 실시예에서, PIM 아키텍처(200)는 프로세싱 시스템(100)보다 더 많거나 더 적은 구성 요소를 포함하는 상이한 프로세싱 시스템에 대응한다. 예시된 실시예에서, PIM 아키텍처는 적어도 프로세싱 모듈(104) 및 시스템 메모리(112)를 포함한다. 프로세싱 모듈(104)은 코프로세싱 모듈(202)을 포함한다. 시스템 메모리(112)는 메모리 모듈(210)을 포함한다. 메모리 모듈(210)은 PCS 모듈(212), PIM 커널 스케줄러(214), 저장 디바이스(216), 저장 맵(218), 및 PIM 모듈(120)을 포함한다. 예시된 실시예가 구성 요소의 특정 구성을 도시하지만, 다양한 실시예에서, 구성 요소의 다른 조합이 고려된다. 예를 들어, 일부 실시예에서, 시스템 메모리(112)는 단일 메모리 모듈(210)만을 포함한다. 또 다른 예로서, 일부 실시예에서, 프로세싱 모듈(104)은 코프로세싱 모듈(202)을 포함하지 않는다. 또한, 일부 실시예에서, 버퍼 또는 데이터 패브릭과(102) 같은 추가적인 구성 요소가 고려되지만, 논의의 용이함을 위해 예시되지 않는다.

[0017] 프로세싱 모듈(104)은 다양한 프로그램을 실행한다. 이러한 프로그램 중 적어도 일부에 대한 데이터는 시스템 메모리(112)(예를 들어, 저장 디바이스(216)에서)에 저장된다. 일부 경우에서, 프로세싱 모듈(104)과 시스템 메모리(112) 사이에서 데이터 요청과 응답을 반복적으로 전송하는 대신 시스템 메모리(112)에서(예를 들어, 하나 이상의 PIM 모듈(120)을 사용하여) 적어도 일부 프로세싱 기능을 수행하기로 결정이 내려진다. 따라서, 프로세싱 모듈(104), 또는 일부 경우에서, 코프로세싱 모듈(202)은 커널 명령어(204)를 시스템 메모리(112)에 전송한다. 예시된 실시예에서, 커널 명령어(204)는 다수의 애플리케이션, 단일 애플리케이션 내의 상이한 스레드, 또는 둘 모두에 대응하고, 시스템 메모리(112)에서 별개의 PIM 커널로서 실행된다. 일부 실시예에서, 프로세싱 모듈(104)은 추가적으로 일부 커널 명령어(예를 들어, PIM 모듈(120)의 ISA의 일부가 아닌 커널 명령어 또는 PIM 모듈(120)이 수행할 대역폭을 갖지 않는 커널 명령어)를 실행한다.

[0018] 예시된 실시예에서, 커널 명령어(204)는 하나 이상의 커맨드 인덱스(206) 및 하나 이상의 뱅크 로컬 어드레스(208)를 포함한다. 커맨드 인덱스(206)는 PIM 모듈(예를 들어, PIM 모듈(120-1))에 의해 수행될 PIM 커맨드를 나타낸다. 일부 경우에, 커맨드 인덱스(206)는 각각의 명령어 식별자를 통해 수행될 각각의 PIM 커맨드를 나타낸다. 또한, 도 3을 참조하여 후술되는 바와 같이, 일부 경우에, 커맨드 인덱스(206) 중 적어도 일부는 PIM 커맨드를 수행하는 것의 일부로서 사용되도록 시스템 메모리(112) 내의 (예를 들어, 저장 디바이스(216-2) 내의) 레지스터를 나타내는 가상 레지스터 어드레스를 포함한다. 뱅크 로컬 어드레스(208)는, PIM 커맨드가 메모리 동작(예를 들어, 로드 동작)을 수행하는 경우 액세스할 시스템 메모리(112)의 저장 디바이스(예를 들어, 저장 디바이스(216-1)) 내의 하나 이상의 메모리 어드레스를 나타낸다. 일부 실시예에서, 뱅크 로컬 어드레스(208)는 완전한(full) 메모리 어드레스이다. 다른 실시예에서, 뱅크 로컬 어드레스(208)는 (예를 들어, 메모리 어레이의 베이스 어드레스를 데이터 요소의 인덱스에 추가함으로써) 시스템 메모리(112)에 의해 수정되거나 생성된다. PIM 모듈(120)이 예를 들어, 프로세싱 모듈(104 내지 106)과 비교하여 더 작은 ISA 및 더 적은 하드웨어 자원을 갖기 때문에, 일부 실시예에서, 별개의 커널 명령어(204)가 각각의 PIM 커맨드에 대해 전송된다(예를 들어, 하나의 커널 명령어가 로드 동작을 수행하기 위해 전송되고, 제2 커널 명령어가 추가 동작을 수행하기 위해 전송됨). 일부 실시예에서, 하나의 애플리케이션 또는 다수의 애플리케이션에 대한 커널 명령어(204)의 다수의 세트(예를 들어, 5개의 커널 명령어 또는 50개의 커널 명령어)가 함께 전송되어, 일부 경우에서, 다수의 PIM 커맨드로 하여금 PIM 모듈(120) 중 하나 이상에 의한 프로세싱을 위해 PCS 모듈(212)의 각각의 엔트리에 저장되게 한

다. 그러나, 일부 실시예에서, 프로세싱 자원(예를 들어, PCS 모듈(212) 내의 저장 공간)은 제한되고, 따라서 커널 명령어(204)의 더 작은 그룹이 전송된다. 추가적으로, 위에서 언급된 바와 같이, 일부 경우에서, 일부 커널 명령어는 프로세싱 모듈(104)에서 실행된다. 또한, 일부 경우에, 적어도 일부 PIM 커맨드는 유사한 작용을 수행하기 위해 프로세싱 모듈(104)에서 실행될 대응하는 명령어와는 상이하다. 그 결과, 일부 경우에서, 커맨드 인덱스(206)는 프로세싱 모듈(104)에서 대응하는 PIM 커맨드로 변환된다. 다른 실시예에서, PIM 커널 스케줄러(214)는 커널 명령어(204)에 의해 표시된 커맨드를 대응하는 PIM 커맨드로 변환한다.

[0019] 시스템 메모리(112)는 하나 이상의 메모리 모듈(210)을 포함한다. 다양한 실시예에서, 메모리 모듈(210)은 동일한 애플리케이션에 대해 또는 상이한 애플리케이션에 대해 동작을 수행한다. 일부 실시예에서, 메모리 모듈(210)은 동적 랜덤 액세스 메모리(dynamic random-access memory; DRAM) 디바이스이다. 그러나, 다른 실시예에서, 메모리 모듈(210)은 다른 유형의 메모리 디바이스(예를 들어, 정적 랜덤 액세스 메모리(static random-access memory; SRAM) 디바이스, 동기식 DRAM(synchronous DRAM; SDRAM) 메모리 디바이스, 플래시 메모리 디바이스 등)이다. 일부 실시예에서, 메모리 모듈(210)은 동일한 유형의 메모리 디바이스이다. 다른 실시예에서, 메모리 모듈(210)은 상이한 메모리 디바이스이다. 일부 실시예에서, 메모리 모듈(210) 중 하나 이상은 더 적거나 추가적인 구성 요소를 포함한다. 예를 들어, 일부 실시예에서, PIM 커널 스케줄러(214-1)는 메모리 모듈(210-1 및 210-2) 둘 모두에 대응하고 메모리 모듈(210-2)은 PIM 커널 스케줄러(214-2)를 포함하지 않는다. 일부 실시예에서, 메모리 모듈(210)은 저장 디바이스(216)에서 레지스터 뱅크로서 조직화된 (예를 들어, 물리적으로 조직화되거나 가상으로 조직화된) 각각의 레지스터를 포함한다. 다른 실시예에서, 레지스터는 메모리 모듈(210) 중 하나 이상과 연관된 하나 이상의 별개의 디바이스의 일부이다. 다양한 실시예에서, PIM 모듈(120)은

[0020] PCS 모듈(212)은 PIM 커맨드를 저장하고, 실행을 위해 PIM 커맨드를 하나 이상의 PIM 모듈(120)에 전송한다. 도 4의 예에서 추가로 설명되는 바와 같이, 일부 실시예에서, PCS 모듈(212) 중 하나 이상은 가상화된다. 예를 들어, 일부 경우에 PCS 모듈(212-2)은 PIM 모듈(120-2)에 의해 수행될 PIM 커맨드의 시퀀스를 저장하며, 여기서 PIM 커맨드는 다수의 각각의 애플리케이션에 대응하는 다수의 PIM 커널의 부분이다.

[0021] PIM 커널 스케줄러(214)는 하나 이상의 PCS 모듈(212)의 가상화를 관리한다. 예를 들어, 전송한 바와 같이, 커맨드 인덱스(206)는 수행될 각각의 PIM 커맨드를 나타낸다. 일부 실시예에서, PIM 커널 스케줄러(214-2)는 PCS 모듈(212-2)을 채우고(populate) PIM 모듈(120-2)에 어드레싱된 커맨드 인덱스(206)를 PCS 모듈(212-2) 내의 위치로 변환하며, 여기서 각각의 위치는 대응하는 PIM 커맨드를 저장한다. 추가적으로, 일부 실시예에서, PIM 커널 스케줄러(214) 중 하나 이상은 PIM 커맨드에 의해 사용되는 데이터가 저장되는 (예를 들어, 저장 디바이스(216-2)의) 물리적 레지스터를 나타내는 가상-대-물리적 매핑 테이블(virtual-to-physical mapping table)로 저장 맵(218-2)을 유지하고 채운다. 일부 실시예에서, 저장 맵(218) 중 하나 이상은 각각의 PIM 모듈(120)에 물리적으로 가까운 (예를 들어, 동일한 칩 상에 있는) 저장 디바이스에 저장된다. 일부 실시예에서, 저장 맵(218) 중 하나 이상은 메모리 모듈(210)과는 별개이지만 하나 이상의 메모리 모듈(210)과 연관된다.

[0022] 일부 실시예에서, PCS 모듈(212)을 채우는 동안 PCS 자원 사용량을 감소시키기 위해, PIM 커널 스케줄러(214)는 적어도 2개의 PIM 커널에 대응하는 적어도 2개의 커널 명령어(204)의 부분들을 조합한다. 예를 들어, 도 4를 참조하여 아래에서 추가로 설명되는 바와 같이, 두 개의 커널 명령어가 PIM 모듈(120-1)로부터 "LOAD R0 ← [ADDR]"을 요청하면, PIM 커널 스케줄러(214-1)는 "LOAD R0 ← [ADDR]"의 단일 표시만으로 PCS 모듈(212-1)을 채우므로, PCS 모듈(212-1)의 저장 공간을 절약하고, PCS 모듈(212-1)이 프로그래밍되는 횟수를 감소시킨다. 추가적으로, 일부 실시예에서, PCS 모듈(212), PIM 커널 스케줄러(214) 중 하나 이상, 또는 둘 모두가 다수의 PIM 모듈(120) 간에 공유된다. 따라서, 일부 실시예에서, PIM 커널 스케줄러는, PIM 모듈(120-1 및 120-2)이 각각 상이한 PIM 커널에 대해 "LOAD R0 ← [ADDR]"을 수행하고, "LOAD R0 ← [ADDR]"의 단일 표시만을 PCS 모듈(212-1)에 저장하는 것을 검출한다.

[0023] 다양한 실시예에서, PIM 커널 스케줄러(214)는 PIM 모듈(120)에서 스케줄링하기 위한 PIM 커널의 조합을 추가로 결정하고, 일부 경우에는 적어도 일부 PIM 커맨드의 실행을 지연시킨다. 이러한 조합은 PIM 커널들 간의 각각의 커널 명령어의 공통성, PIM 커널의 각각의 우선순위, PIM 커널의 예상 자원 사용량(예를 들어, PIM 모듈 사용량, 메모리 대역폭 사용량, 통신 대역폭 등), (예를 들어, 프로세싱 모듈(104)로부터 수신된 종속성 그래프에 표시되는) PIM 커널의 종속성과 같은 다양한 요인을 나타내는 조합 메트릭(combination metric)에 기초해 결정된다. 일부 실시예에서, 조합은 (예를 들어, 컴파일러로부터) 프로세싱 모듈(104)에 의해 표시된다. 다른 실시예에서, 조합은, 예컨대 PIM 커널의 종속성 그래프를 수신함으로써, 런타임에 PIM 커널 스케줄러(214)에 의해 결정된다. PIM 커맨드의 순서는 또한 PIM 커널들 사이의 각각의 커널 명령어의 공통성, PIM 커널의 각각의 우선순위, PIM 커널의 예상 자원 사용량, PIM 커널의 종속성(예를 들어, 프로세싱 모듈(104)로부터 수신된 종속성

그래프에 표시됨), 또는 대응하는 커널 명령어가 수신된 시간과 같은 다양한 요인에 기초하여 결정된다.

- [0024] PIM 모듈(120)은 PCS 모듈(212)로부터 PIM 커맨드를 수신하고, 일부 경우에서, 저장 디바이스(216) 및 저장 디바이스(216)의 레지스터에 저장된 데이터를 사용해 그러한 PIM 커맨드를 프로세싱한다. 일부 실시예에서, PIM 모듈(120) 중 하나 이상은 산술 논리 유닛(ALU)이다. 추가적으로, 일부 실시예에서, 저장 디바이스(216)는 PIM 모듈(120)에 대한 물리적 레지스터 뱅크 또는 가상 레지스터 뱅크로서 조직화된 레지스터 파일이다. 일부 경우에서, 레지스터 뱅크의 적어도 일부 레지스터는 제1 커널에 할당되는 반면, 레지스터 뱅크의 적어도 일부 레지스터는 제2 커널에 할당된다. (예를 들어, 커널 명령어(204)를 통해) 프로세싱 모듈(104)로부터의 요청에 응답하여, PIM 모듈(120)은 데이터를 프로세싱 모듈(104)에 반환한다. 따라서, 일부 경우에 PIM 모듈(120)이 다수의 PIM 커널을 동시에 프로세싱하도록 메모리 모듈(210)의 프로세싱 자원이 가상화된다.
- [0025] 일부 실시예에서, 하나 이상의 저장 맵(218) 및 하나 이상의 저장 맵(218)에 대응하는 PIM 커널 스케줄러(214)의 일부가 프로세싱 모듈(104)의 스크래치패드(scratchpad)에 저장된다. 그 결과, 그러한 실시예에서, 커널 명령어(204)가 시스템 메모리(112)로 전송되기 전에 뱅크 로컬 어드레스(208)에 대해 변환이 발생한다.
- [0026] 도 3은 적어도 일부 실시예에 따른 예시(300)의 가상-대-물리적 커널 레지스터 매핑(310)을 예시한다. 전술된 바와 같이, 일부 실시예에서, 가상-대-물리적 커널 레지스터 매핑(310)은 도 2의 PIM 커널 스케줄러(214-1)와 같은 PIM 커널 스케줄러(214)에 의해 수행된다. 예시된 예시(300)에서, 커널 1 가상 레지스터 맵(302)과 커널 2 가상 레지스터 맵(304)은 PIM 커널 스케줄러(214)에서 수신된다. 커널 1 가상 레지스터 맵(302)의 명령어에 의해 표시된 레지스터는 R0, R1, 및 R2이다. PIM 커널 스케줄러(214)는 수신된 가상 레지스터 어드레스를 물리적 레지스터 어드레스 [0], [2], 및 [7]로 각각 변환하고, 레지스터 어드레스는 저장 디바이스(216-1)의 레지스터와 같은 PIM 커널 스케줄러(214)와 연관된 저장 디바이스의 어드레스에 대응한다. 유사하게, 커널 2 가상 레지스터 맵(304)의 명령어에 의해 표시된 레지스터는 R0, R1, R2, 및 R3이다. PIM 커널 스케줄러는 그러한 수신된 가상 레지스터 어드레스를 물리적 레지스터 어드레스 [3], [4], [8], 및 [9]로 각각 변환한다. 따라서, PIM 모듈은 가상 인덱스를 나타내는 커널 명령어를 프로세싱할 수 있다. 추가적으로, 일부 경우에서, 적어도 일부 레지스터는 제1 커널에 할당되는 반면, 적어도 일부 레지스터는 제2 커널에 할당된다.
- [0027] 도 4는 적어도 일부 실시예에 따른 예시(400)의 커널 명령어 병합 및 매핑(420)을 예시한다. 전술된 바와 같이, 일부 실시예에서, 병합 및 매핑(420)은 도 2의 PIM 커널 스케줄러(214-2)와 같은 PIM 커널 스케줄러에 의해 수행된다. 예시된 예시(400)에서, 커널 1 명령어(402)와 커널 2 명령어(404) 각각은 각각의 인덱스에 대응한다. PCS 모듈(212-2)과 같은 PCS 모듈에 저장하기 전에, 명령어는 PCS 모듈 내의 저장 활용을 감소시키도록 병합된다. 따라서, 커널 1 명령어(402)와 커널 2 명령어(404) 둘 모두에 대해 "LOAD R0 ← [ADDR]"은 커널 1 명령어 인덱스(406)에서와 커널 2 명령어 인덱스(408)에서 물리적 인덱스 0으로 병합된다. 추가적으로, PIM 커널 스케줄러는 명령어의 순서를 결정하여, PCS 병합된 커널 콘텐츠(410)를 생성한다. 결과적으로, 커널 1 명령어(402)와 커널 2 명령어(404)는 표시된 순서로 PIM 모듈(120-2)과 같은 PIM 모듈에 의해 동시에 프로세싱된다. 개별 PIM 커널에 대한 뱅크 로컬 어드레스가 호스트 명령어에 의해 생성되고 각각의 PCS 모듈 인덱스와 함께 전송되기 때문에, 일부 경우에서, 동일한 PCS 모듈 엔트리가 다수의 커널에 의해 사용되더라도 상이한 물리적 레지스터가 사용된다. 따라서, PIM 모듈은 다수의 커널에 대한 커널 명령어를 동시에 프로세싱할 수 있고, PCS 모듈의 자원 소비가 감소된다.
- [0028] 도 5는 일부 실시예에 따른, PIM 동시적 프로세싱 시스템을 사용하여 다수의 커널에 대한 커널 명령어를 동시에 프로세싱하는 방법(500)을 예시하는 흐름도이다. 일부 실시예에서, 방법(500)은 컴퓨터 판독 가능 저장 매체에 의해 저장된 하나 이상의 명령어에 응답하여 하나 이상의 프로세서에 의해 개시된다.
- [0029] 블록(502)에서, PIM ALU에 의해 동시에 프로세싱될 복수의 커널에 대한 커널 명령어가 수신된다. 예를 들어, PIM 모듈(120-1)에 의해 프로세싱될 다수의 커널에 대응하는 커널 명령어(204)가 프로세싱 모듈(104)로부터 수신되며, 여기서 PIM 모듈(120-1)에 의해 구현된 ISA는 프로세싱 모듈(104)에 의해 구현된 ISA보다 더 적은 수의 명령어를 갖는다. 블록(504)에서, 커널 명령어에 의해 표시된 커맨드의 가상 레지스터 어드레스는 물리적 레지스터 어드레스로 변환된다. 예를 들어, PIM 커널 스케줄러(214-1)는 커널 명령어(204)에 의해 표시된 가상 레지스터 어드레스를 저장 디바이스(216-1)의 레지스터의 물리적 레지스터 어드레스로 변환한다.
- [0030] 블록(506)에서, 커널 명령어에 의해 표시된 PIM 커맨드 중 적어도 일부가 병합된다. 예를 들어, PIM 커널 스케줄러(214-1)는 커널 명령어(204)에 의해 표시된 PIM 커맨드를 분석하고 조합 메트릭에 기초하여 적어도 일부 PIM 커맨드를 병합한다. 블록(508)에서, 변환되고 병합된 커맨드가 PCS 모듈에 저장된다. 예를 들어, 변환되고 병합된 커맨드는 PCS 모듈(212-1)에 저장된다. 블록(510)에서, 적어도 하나의 병합된 커맨드가 검색되고 수행된

다. 예를 들어, 적어도 2개의 PIM 커널에 대응하는 적어도 하나의 PIM 커맨드가 PCS 모듈(212-1)로부터 검색되고 실행을 위해 PIM 모듈(120-1)로 전송된다. 따라서, 다수의 커널에 대한 커널 명령어를 동시에 프로세싱하는 방법이 묘사된다.

[0031] 일부 실시예에서, 컴퓨터 판독 가능 저장 매체는 컴퓨터 시스템에 명령어 및/또는 데이터를 제공하기 위해 사용하는 동안 컴퓨터 시스템에 의해 액세스 가능한 임의의 비일시적 저장 매체 또는 비일시적 저장 매체의 조합을 포함한다. 이러한 저장 매체는 광학 매체(예를 들어, CD(compact disc), DVD(digital versatile disc), 블루레이 디스크), 자기 매체(예를 들어, 플로피 디스크, 자기 테이프 또는 자기 하드 드라이브), 휘발성 메모리(예를 들어, 랜덤 액세스 메모리(RAM) 또는 캐시), 비휘발성 메모리(예를 들어, 판독 전용 메모리(ROM) 또는 플래시 메모리), 또는 미세 전자 기계 시스템(microelectromechanical system; MEMS) 기반 저장 매체를 포함할 수 있으나, 이에 제한되는 것은 아니다. 일부 실시예에서, 컴퓨터 판독 가능 저장 매체는 컴퓨팅 시스템(예를 들어, 시스템 RAM 또는 ROM)에 내장되거나, 컴퓨팅 시스템(예를 들어, 자기 하드 드라이브)에 고정적으로 부착되거나, 컴퓨팅 시스템(예를 들어, 광학 디스크 또는 범용 직렬 버스(Universal Serial Bus; USB) 기반 플래시 메모리)에 제거 가능하게 부착되거나, 또는 유선 또는 무선 네트워크(예를 들어, 네트워크 액세스 가능 저장소(network accessible storage; NAS))를 통해 컴퓨터 시스템에 결합된다.

[0032] 일부 실시예에서, 위에서 설명된 기술의 특정 양태는 소프트웨어를 실행하는 프로세싱 시스템의 하나 이상의 프로세서에 의해 구현된다. 소프트웨어는 비일시적 컴퓨터 판독 가능 저장 매체에 저장되거나 달리 유형적으로 구현된 실행 가능 명령어의 하나 이상의 세트를 포함한다. 소프트웨어는 하나 이상의 프로세서에 의해 실행될 때, 전술된 기법의 하나 이상의 양태를 수행하도록 하나 이상의 프로세서를 조작하는 명령어 및 특정 데이터를 포함할 수 있다. 비일시적 컴퓨터 판독 가능 저장 매체는, 예를 들어, 자기 또는 광학 디스크 저장 디바이스, 예를 들어, 플래시 메모리, 캐시, 랜덤 액세스 메모리(RAM) 또는 다른 비휘발성 메모리 디바이스 또는 디바이스들 등과 같은 솔리드 스테이트 저장 디바이스를 포함할 수 있다. 일부 실시예에서, 비일시적 컴퓨터 판독 가능 저장 매체 상에 저장된 실행 가능한 명령어는 소스 코드, 어셈블리 언어 코드, 객체 코드, 또는 하나 이상의 프로세서에 의해 해석되거나 다른 방식으로 실행 가능한 다른 명령어 형식이다.

[0033] 일반적인 설명에서 위에서 설명된 모든 활동 또는 요소가 요구되지는 않으며, 특정 활동 또는 디바이스의 일부가 요구되지 않으며, 일부 경우에서, 위에서 설명된 것에 더하여 하나 이상의 추가적인 활동이 수행되거나 요소가 포함된다. 또한, 추가적으로, 활동이 열거된 순서는 반드시 그들이 수행되는 순서는 아니다. 또한, 특정 실시예를 참조하여 개념이 설명되었다. 그러나, 당업자는 아래의 청구항에 기재된 것과 같은 본 개시내용의 범위로부터 벗어나지 않으면서 다양한 수정 및 변경이 이루어질 수 있음을 이해한다. 따라서, 명세서 및 도면은 제한적인 의미보다는 예시적인 의미로 간주되어야 하며, 모든 그러한 수정은 본 개시내용의 범위 내에 포함되도록 의도된다.

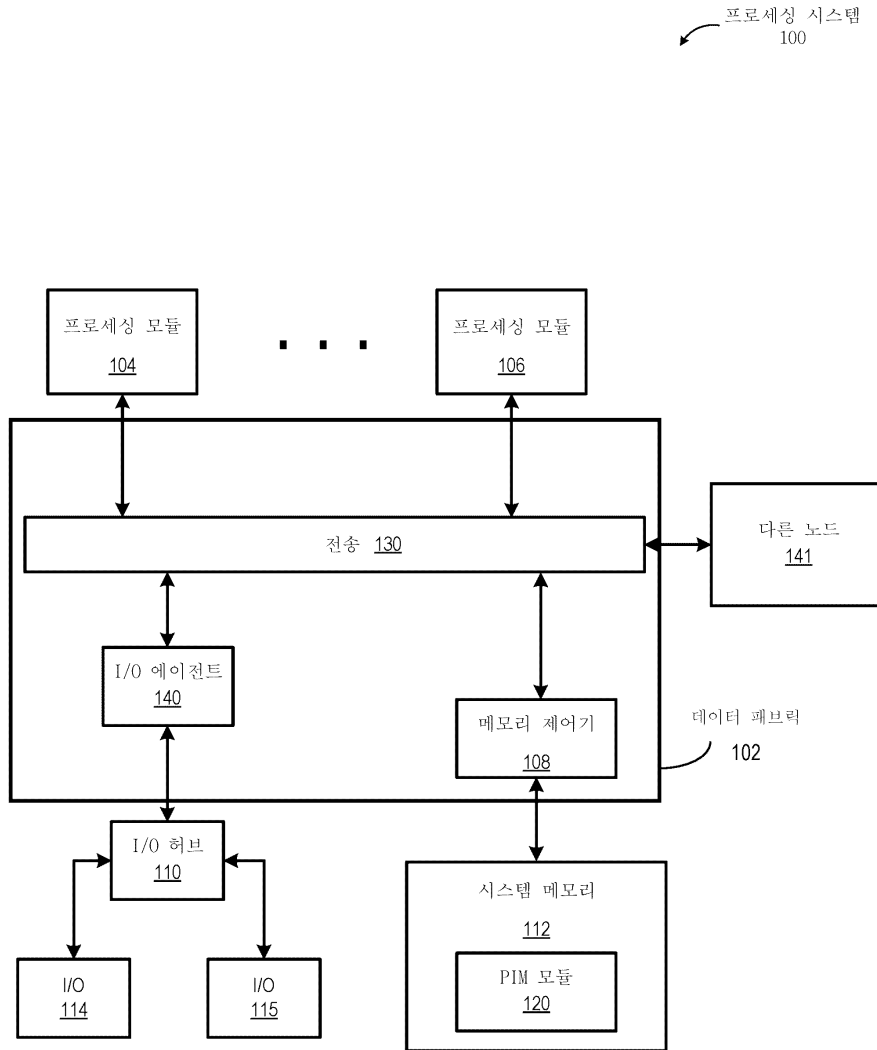
[0034] 이익, 다른 이점, 및 문제에 대한 해결책이 특정 실시예와 관련하여 전술되었다. 그러나, 이익, 이점, 문제에 대한 해결책, 및 어떤 이익, 이점, 또는 해결책이 발생하게 하거나 더 두드러지게 하는 어떤 특징(들)은 청구항 중 어느 하나 또는 모두의 중요한, 요구되는, 또는 필수적인 특징으로서 해석되지 않아야 한다. 더욱이, 위에서 개시된 특정 실시예는 단지 예시적이며, 개시된 요지가 본원에서의 교시의 이익을 갖는 당업자에게 명백한 상이 하지만 동등한 방식으로 수정되고 실시될 수 있다. 아래의 청구항에 설명된 것 이외에, 본원에서 보여진 구성 또는 설계의 상세 사항에 대한 제한이 의도되지 않는다. 따라서, 위에서 개시된 특정 실시예는 변경 또는 수정될 수 있고, 이러한 모든 변형은 개시된 요지의 범위 내에서 고려된다는 것이 명백하다. 따라서, 본원에서 추구되는 보호는 아래의 청구항에 기재된 것과 같다.

[0035] 본 개시내용 내에서, 일부 경우에서, 상이한 엔티티("구성 요소", "유닛", "디바이스" 등으로 다양하게 지칭됨)는 하나 이상의 작업 또는 동작을 수행하도록 "구성된" 것으로 설명되거나 청구된다. 이 포물레이션(formulation)-[하나 이상의 작업을 수행하도록] 구성된 [엔티티]-은 본원에서는 구조(즉, 전자 회로와 같은 어떤 물리적 것)를 지칭하기 위해 사용된다. 보다 구체적으로, 이 포물레이션은 이 구조가 동작 동안 하나 이상의 작업을 수행하도록 배열된다는 것을 나타내기 위해 사용된다. 구조는, 구조가 현재 동작하지 않더라도 일부 작업을 수행하도록 "구성"된다고 할 수 있다. "데이터를 저장하도록 구성된 메모리 디바이스"는, 예를 들어, 해당 집적 회로가 현재 사용되고 있지 않더라도(예를 들어, 전력 공급부가 그것에 접속되어 있지 않더라도), 동작 동안 데이터를 저장하는 회로를 갖는 집적 회로를 포괄하도록 의도된다. 따라서, 일부 작업을 수행하도록 "구성된" 것으로서 설명되거나 인용된 엔티티는 디바이스, 회로, 작업을 구현하도록 실행 가능한 프로그램 명령어를 저장하는 메모리 등과 같은 어떤 물리적인 것을 지칭한다. 이 문구는 무형의 것을 지칭하기 위해 본원에서 사용되지 않는다. 또한, "구성됨"이라는 용어는 "구성 가능"을 의미하는 것으로 의도되지 않는다. 프로그래밍되

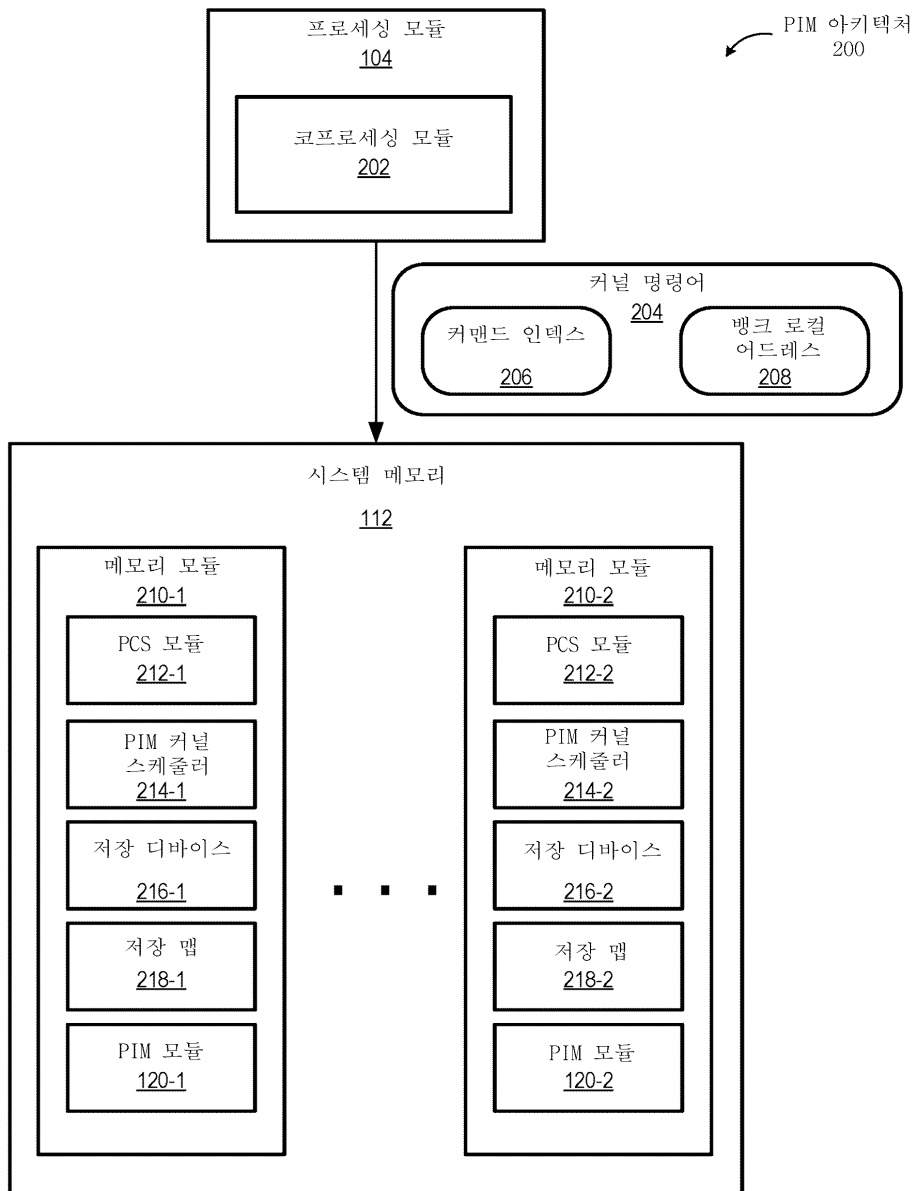
지 않은 필드 프로그래밍 가능 게이트 어레이는, 예를 들어, 일부 특정 기능을 수행하도록 "구성된" 것으로 간주되지 않을 것이지만, 프로그래밍 후에 그 기능을 수행하도록 "구성 가능할" 수 있다. 추가적으로, 구조가 하나 이상의 작업을 수행하도록 "구성"된다는 첨부된 청구항에서의 인용은 수단-플러스-기능 요소를 갖는 것으로 해석되지 않도록 명시적으로 의도된다.

도면

도면1

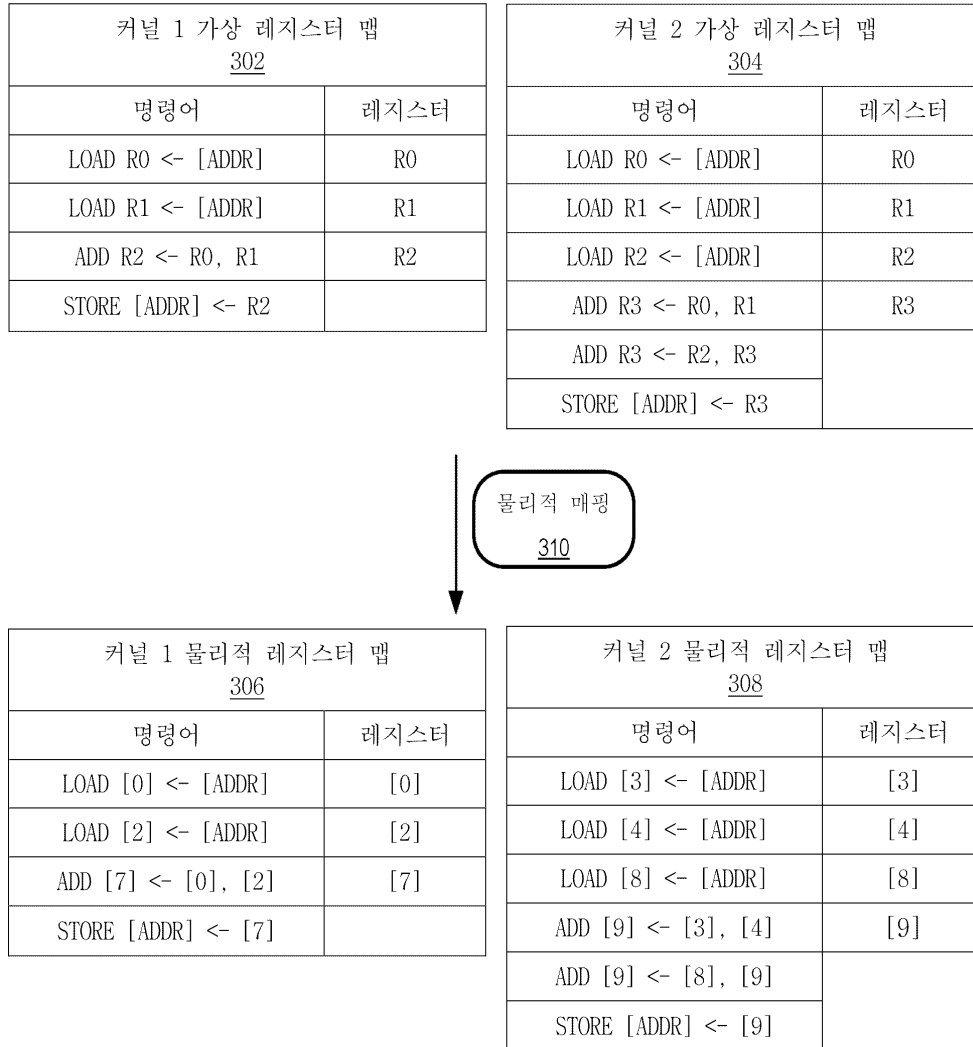


도면2

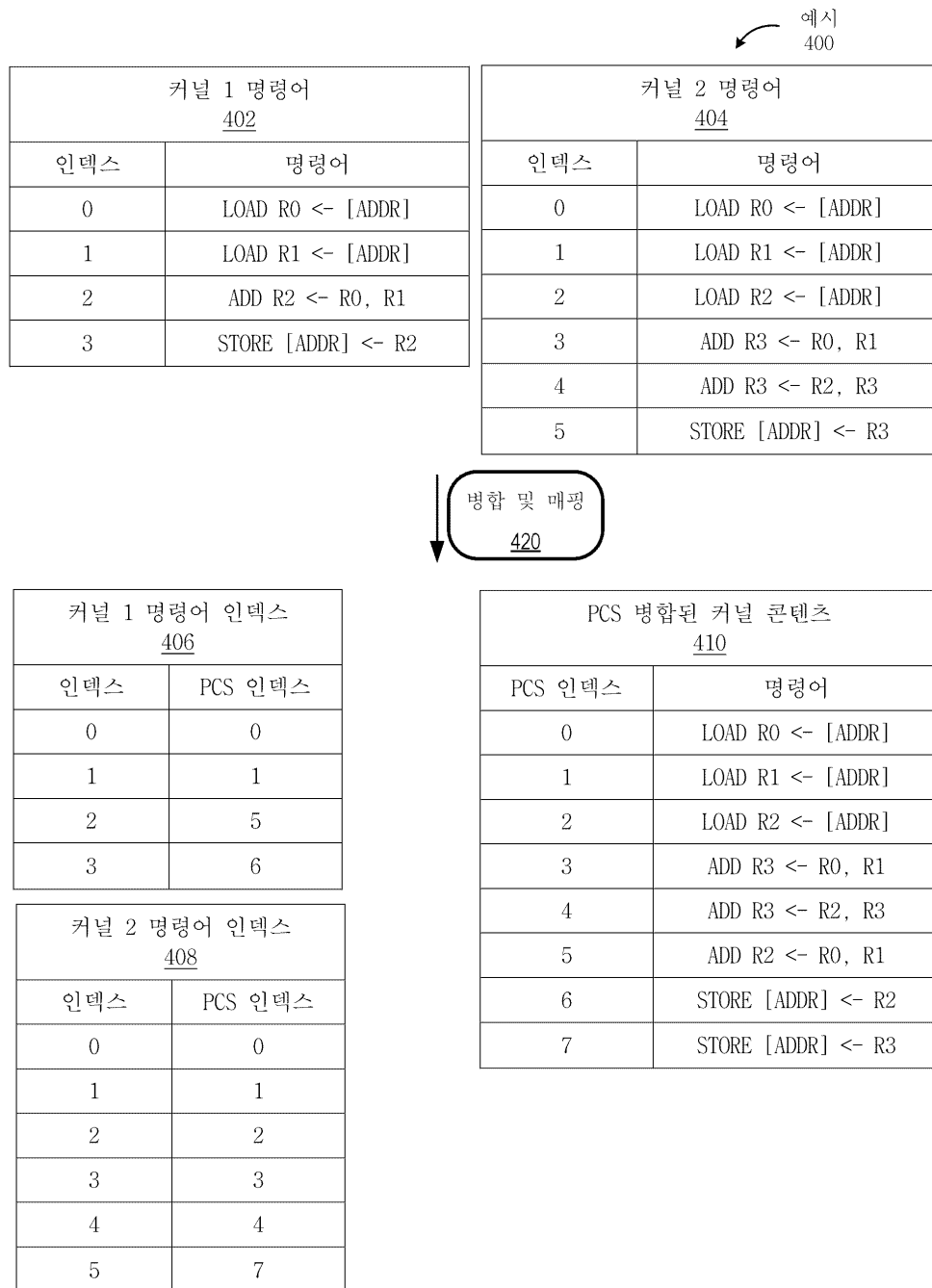


도면3

예시
300



도면4



도면5

