



(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0091861  
(43) 공개일자 2023년06월23일

- (51) 국제특허분류(Int. Cl.)  
G06F 13/38 (2006.01) G06F 13/42 (2006.01)  
G06F 15/78 (2006.01)
- (52) CPC특허분류  
G06F 13/385 (2013.01)  
G06F 13/4221 (2013.01)
- (21) 출원번호 10-2023-7009883
- (22) 출원일자(국제) 2021년06월21일  
심사청구일자 없음
- (85) 번역문제출일자 2023년03월22일
- (86) 국제출원번호 PCT/US2021/038273
- (87) 국제공개번호 WO 2022/093334  
국제공개일자 2022년05월05일
- (30) 우선권주장  
17/085,740 2020년10월30일 미국(US)

- (71) 출원인  
자일링크스 인코포레이티드  
미합중국 95124 캘리포니아 산 호세 로직 드라이브 2100
- (72) 발명자  
산탄, 소날  
미국 95124 캘리포니아 새너제이 로직 드라이브 2100  
쿠를라군다, 라비, 엔.  
미국 95124 캘리포니아 새너제이 로직 드라이브 2100  
(뒷면에 계속)
- (74) 대리인  
특허법인 남앤남

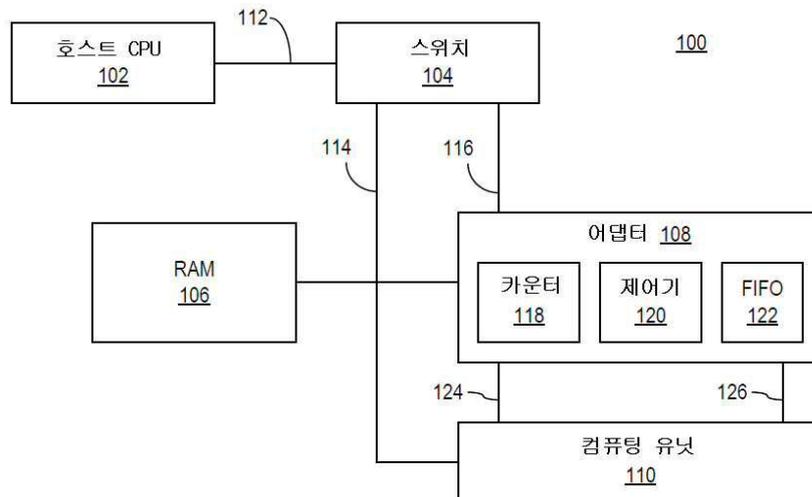
전체 청구항 수 : 총 15 항

(54) 발명의 명칭 하드웨어 가속을 위한 높은 처리량 회로 아키텍처

(57) 요약

하드웨어 가속 디바이스는 호스트 중앙 프로세싱 유닛(CPU)에 통신 가능하게 링크된 스위치, 제어 버스를 통해 스위치에 결합된 어댑터 - 제어 버스는 호스트 중앙 CPU로부터 어댑터로 기술자(descriptor)들의 주소들을 전달하도록 구성됨 -, 및 데이터 버스를 통해 스위치에 결합된 RAM(Random-Access Memory)을 포함할 수 있다. RAM은 데이터 버스를 통해 호스트 CPU로부터 수신된 기술자들을 저장하도록 구성된다. 하드웨어 가속 디바이스는, 어댑터에 결합되고 기술자들에 의해 지정된 조작들을 수행하도록 구성된 컴퓨팅 유닛을 포함할 수 있다. 어댑터는 데이터 버스를 통해 RAM으로부터 기술자들을 검색하고, 기술자들로부터의 인수들을 컴퓨팅 유닛으로 제공하고, 인수들을 사용하여 조작들을 개시하기 위해 컴퓨팅 유닛으로 제어 신호들을 제공하도록 구성될 수 있다.

대표도 - 도1



(52) CPC특허분류

*G06F 15/7889* (2013.01)  
*G06F 2213/0026* (2013.01)  
*Y02D 10/00* (2020.08)

(72) 발명자

**마, 민**

미국 95124 캘리포니아 새너제이 로직 드라이브  
2100

**초드하리, 히만슈**

미국 95124 캘리포니아 새너제이 로직 드라이브  
2100

**채푸리, 만주나스**

미국 95124 캘리포니아 새너제이 로직 드라이브  
2100

**전, 칭**

미국 95124 캘리포니아 새너제이 로직 드라이브  
2100

**조시, 프란잘**

미국 95124 캘리포니아 새너제이 로직 드라이브  
2100

**투를롤스, 세바스티안**

미국 95124 캘리포니아 새너제이 로직 드라이브  
2100

**쿠마르, 아미트**

미국 95124 캘리포니아 새너제이 로직 드라이브  
2100

**만지, 카우스투브**

미국 95124 캘리포니아 새너제이 로직 드라이브  
2100

**샤르마, 라빈데르**

미국 95124 캘리포니아 새너제이 로직 드라이브  
2100

**크리슈나, 츠 밤시**

미국 95124 캘리포니아 새너제이 로직 드라이브  
2100

## 명세서

### 청구범위

#### 청구항 1

하드웨어 가속 디바이스로서,

호스트 중앙 프로세싱 유닛(CPU)에 통신 가능하게 링크된 스위치;

제어 버스를 통해 상기 스위치에 결합된 어댑터 회로 - 상기 제어 버스는 상기 호스트 CPU로부터 상기 어댑터 회로로 기술자(descriptor)들의 주소들을 전달하도록 구성됨 -;

데이터 버스를 통해 상기 스위치에 결합된 랜덤-액세스 메모리 - 상기 랜덤-액세스 메모리는 상기 데이터 버스를 통해 상기 호스트 CPU로부터 수신된 기술자들을 저장하도록 구성됨 -; 및

상기 어댑터 회로에 결합되고, 상기 기술자들에 의해 지정된 조작들을 수행하도록 구성된 컴퓨팅 유닛을 포함하고,

상기 어댑터 회로는 상기 데이터 버스를 통해 상기 랜덤-액세스 메모리로부터 상기 기술자들을 검색하고, 상기 기술자들로부터의 인수(argument)들을 상기 컴퓨팅 유닛으로 제공하고, 상기 인수들을 사용하여 상기 조작들을 개시하기 위해 상기 컴퓨팅 유닛으로 제어 신호들을 제공하도록 구성되는, 하드웨어 가속 디바이스.

#### 청구항 2

제1 항에 있어서,

상기 데이터 버스는 메모리-매핑 버스인, 하드웨어 가속 디바이스.

#### 청구항 3

제1 항에 있어서,

상기 스위치는 PCIe(Peripheral Component Interconnect Express) 스위치이고; 그리고

상기 제어 버스와 상기 데이터 버스는 온-칩 상호연결들로 구현되는, 하드웨어 가속 디바이스.

#### 청구항 4

제3 항에 있어서,

상기 PCIe 스위치는 상기 제어 버스에 결합된 제1 BAR 및 상기 데이터 버스에 결합된 제2 BAR을 포함하는, 하드웨어 가속 디바이스.

#### 청구항 5

제1 항에 있어서,

상기 어댑터 회로는:

상기 주소들을 저장하도록 구성된 선입선출 메모리(first-in-first-out memory)를 포함하는, 하드웨어 가속 디바이스.

#### 청구항 6

제5 항에 있어서,

상기 어댑터 회로는:

상기 컴퓨팅 유닛에 의해 프로세싱을 완료한 다수의 조작들을 저장하도록 구성된 카운터를 포함하는, 하드웨어 가속 디바이스.

**청구항 7**

제1 항에 있어서,

상기 스위치, 상기 어댑터 회로 및 상기 컴퓨팅 유닛을 포함하는 프로그램 가능 집적 회로를 포함하는, 하드웨어 가속 디바이스.

**청구항 8**

제7 항에 있어서,

상기 프로그램 가능 집적 회로는 상기 랜덤-액세스 메모리를 포함하는, 하드웨어 가속 디바이스.

**청구항 9**

제8 항에 있어서,

상기 랜덤-액세스 메모리는 상기 프로그램 가능 집적 회로 내에서 구현되고, 상기 프로그램 가능 집적 회로 외부에 있고 상기 프로그램 가능 집적 회로를 갖는 회로 기판에 배치되거나, 또는 고대역폭 메모리로 구현되는, 하드웨어 가속 디바이스.

**청구항 10**

제7 항에 있어서,

상기 프로그램 가능 집적 회로는:

임베디드 프로세서; 및

네트워크 인터페이스를 포함하고;

상기 임베디드 프로세서는 상기 네트워크 인터페이스를 통해 네트워크로부터 수신된 추가 기술자들을 상기 랜덤-액세스 메모리에 저장하고 상기 추가 기술자들의 주소들을 상기 어댑터 회로에 제공하도록 구성되는, 하드웨어 가속 디바이스.

**청구항 11**

제7 항에 있어서,

상기 프로그램 가능 집적 회로는 내부에 내장된 상기 호스트 CPU를 포함하고; 그리고

상기 스위치는 상기 제어 버스에 결합된 제1 포트 및 상기 데이터 버스에 결합된 제2 포트를 포함하는, 하드웨어 가속 디바이스.

**청구항 12**

하드웨어 가속 방법으로서,

하드웨어 가속 디바이스의 스위치를 통해 복수의 기술자들 및 상기 복수의 기술자들에 대한 복수의 주소들을 수신하는 단계 - 상기 스위치는 호스트 중앙 프로세싱 유닛(CPU)에 통신 가능하게 링크되고, 상기 기술자들은 상기 하드웨어 가속 디바이스의 컴퓨팅 유닛에 의해 수행될 조작들을 지정함 -;

데이터 버스를 통해 저장하기 위해 랜덤-액세스 메모리에 상기 복수의 기술자들을 전달하는 단계;

제어 버스를 통해 어댑터 회로로 상기 복수의 주소들을 전달하는 단계; 및

상기 어댑터 회로를 사용하고, 상기 데이터 버스를 통해 상기 랜덤-액세스 메모리로부터 상기 기술자들을 검색하고, 상기 기술자들로부터의 인수들을 상기 컴퓨팅 유닛으로 제공하고, 상기 인수들을 사용하여 상기 조작들을 개시하기 위해 상기 컴퓨팅 유닛에 제어 신호들을 제공하는 단계를 포함하는, 하드웨어 가속 방법.

**청구항 13**

제12 항에 있어서,

상기 스위치는 PCIe(Peripheral Component Interconnect Express) 스위치인, 하드웨어 가속 방법.

**청구항 14**

제13 항에 있어서,

상기 PCIe 스위치는 상기 제어 버스에 결합된 제1 BAR 및 상기 데이터 버스에 결합된 제2 BAR을 포함하는, 하드웨어 가속 방법.

**청구항 15**

제12 항에 있어서,

상기 하드웨어 가속 디바이스는 프로그램 가능 집적 회로를 포함하고; 그리고

상기 프로그램 가능 집적 회로는 상기 스위치, 상기 어댑터 회로 및 상기 컴퓨팅 유닛을 포함하는, 하드웨어 가속 방법.

**발명의 설명**

**기술 분야**

[0001] [0001] 본 개시내용은 집적 회로(IC)들 및 하드웨어 가속에 관한 것이다. 특히, 본 개시내용은 하드웨어 가속 기들에 대한 높은 데이터 처리량 달성에 관한 것이다.

**배경 기술**

[0002] [0002] 하드웨어 가속은 서버와 같은 컴퓨터 시스템의 범용 중앙 프로세싱 유닛(CPU)에서 기능을 수행하는 것보다 더 효율적으로 기능을 수행하기 위해서 하드웨어 가속 디바이스라고 지칭되는 특수 하드웨어를 사용하는 것을 말한다. 하드웨어 가속 디바이스는 일반적으로 CPU에 의한 기능 실행을 통해 달성할 수 없는 하나 이상의 이점들을 제공한다. 이점(들)은 하나 이상의 더 빠른 조작, 전력 소비 감소 및 중복성을 포함할 수 있으나, 이에 제한되는 것은 아니다. 하드웨어 가속 디바이스를 사용하는 것의 이점들이 더 빠른 조작을 포함할 수 있거나 그렇지 않을 수 있지만, CPU에서 특수 하드웨어로 기능을 오프로딩하는 프로세스는 "하드웨어 가속"으로 지칭된다.

[0003] [0003] 최신 컴퓨터 시스템들과 주변 디바이스들은 계속 증가하는 데이터 처리량들을 처리할 수 있다. 예를 들어, NVMe(Non-Volatile Memory Express) 저장 디바이스들 및/또는 이더넷(Ethernet), 예를 들어, 네트워킹 디바이스들과 같은 소정 PCIe(Peripheral Component Interconnect Express) 디바이스들은 초당 수백만 조작 정도의 데이터 속도로 조작할 수 있다. 실행 가능한 상태를 유지하기 위해, 종종 CPU들 및/또는 호스트 컴퓨터들에 대한 주변 디바이스들로 조작하는 하드웨어 가속 디바이스들이 필요하다.

**발명의 내용**

[0004] [0004] 일 양상에서, 하드웨어 가속 디바이스는 호스트 중앙 프로세싱 유닛(CPU)에 통신 가능하게 링크된 스위치, 제어 버스를 통해 스위치에 결합된 어댑터 - 제어 버스는 호스트 CPU로부터 어댑터로 기술자(descriptor)들의 주소들을 전달하도록 구성됨 -, 및 데이터 버스를 통해 스위치에 결합된 RAM(Random-Access Memory)을 포함할 수 있다. RAM은 데이터 버스를 통해 호스트 CPU로부터 수신된 기술자들을 저장하도록 구성된다. 하드웨어 가속 디바이스는, 어댑터에 결합되고 기술자들에 의해 지정된 조작들을 수행하도록 구성된 컴퓨팅 유닛을 포함할 수 있다. 어댑터는 데이터 버스를 통해 RAM으로부터 기술자들을 검색하고, 기술자들들로부터의 인수(argument)들을 컴퓨팅 유닛으로 제공하고, 인수들을 사용하여 조작들을 개시하기 위해 컴퓨팅 유닛으로 제어 신호들을 제공하도록 구성될 수 있다.

[0005] [0005] 다른 양상에서, 하드웨어 가속 방법은 하드웨어 가속 디바이스의 스위치를 통해 복수의 기술자들 및 복수의 기술자들에 대한 복수의 주소들을 수신하는 단계를 포함할 수 있고, 스위치는 호스트 CPU에 통신 가능하게 링크된다. 기술자들은 하드웨어 가속 디바이스의 컴퓨팅 유닛에 의해 수행될 조작들을 지정한다. 방법은 데이터 버스를 통해 저장하기 위해 RAM에 복수의 기술자들을 전달하는 단계와 제어 버스를 통해 어댑터 회로에 복수의 주소들을 전달하는 단계를 포함할 수 있다. 방법은 어댑터를 사용하여, 데이터 버스를 통해 RAM로부터 기술자들을 검색하는 단계, 기술자들로부터의 인수들을 컴퓨팅 유닛으로 제공하는 단계, 인수들을 사용하여 조작들

을 개시하기 위해 컴퓨팅 유닛으로 제어 신호들을 제공하는 단계를 포함할 수 있다.

[0006] 이 요약 섹션은 단지 소정 개념들을 소개하기 위해 제공되고 청구된 주제의 임의의 핵심 또는 필수 특징들을 식별하기 위해 제공되지 않는다. 본 발명의 어레이먼트(arrangement)들의 다른 특징들은 첨부된 도면들 및 다음의 상세한 설명으로부터 명백할 것이다.

**도면의 간단한 설명**

[0007] 본 발명의 어레이먼트들은 첨부 도면들에서 예로서 예시된다. 그러나, 도면들은 본 발명의 어레이먼트들을 도시된 특정 구현들로만 제한하는 것으로 해석되어서는 안 된다. 다양한 양상들 및 장점들은 다음의 상세한 설명을 검토하고 도면들을 참조하면 명백해질 것이다.

[0008] 도 1은 하드웨어 가속을 위한 예시적인 회로 아키텍처를 예시한다.

[0009] 도 2는 도 1의 아키텍처로부터의 스위치의 예시적인 구현을 예시한다.

[0010] 도 3은 하드웨어 가속을 위한 다른 예시적인 회로 아키텍처를 예시한다.

[0011] 도 4는 도 1 또는 도 3의 예시적인 아키텍처를 사용하는 하드웨어 가속의 예시적인 방법을 예시한다.

[0012] 도 5는 도 1 또는 도 3의 예시적인 아키텍처를 사용하는 하드웨어 가속의 다른 예시적인 방법을 예시한다.

[0013] 도 6은 하드웨어 가속을 위한 예시적인 컴퓨팅 환경을 예시한다.

[0014] 도 7은 하드웨어 가속에 사용될 수 있는 집적 회로(IC)에 대한 예시적인 아키텍처를 예시한다.

[0015] 도 8은 하드웨어 가속을 위해 사용될 수 있는 IC에 대한 다른 예시적인 아키텍처를 예시한다.

**발명을 실시하기 위한 구체적인 내용**

[0008] 본 개시내용이 새로운 특징들을 정의하는 청구범위들로 결론을 내리지만, 본 개시내용 내에서 설명된 다양한 특징들이 도면들과 함께 설명을 고려함으로써 더 잘 이해될 것이라고 믿어진다. 본원에 설명된 프로세스(들), 기계(들), 제조(들) 및 이들의 임의의 변형은 예시의 목적들로 제공된다. 본 개시내용 내에서 설명된 특정한 구조 및 기능적 세부사항들은 제한하는 것으로 해석되어서는 안 되며, 청구범위들에 대한 기초로서 그리고 통상의 기술자가 사실상 임의의 적절하게 상세한 구조로 설명된 특징들을 다양하게 이용하도록 교시하기 위한 대표적인 기초로서만 해석되어야 한다. 또한, 본 개시내용 내에서 사용된 용어들 및 구문들은 제한하려는 것이 아니라, 설명된 특징들의 이해 가능한 설명을 제공하기 위한 것이다.

[0009] 본 개시내용은 집적 회로(IC)들 및 하드웨어 가속에 관한 것이다. 보다 특히, 본 개시내용은 하드웨어 가속기들에 대한 높은 데이터 처리량 달성에 관한 것이다. 일반적으로, 하드웨어 가속기들은 초당 수만 조작들(OPS: Operations Per Second) 내지 약 십만 OPS 정도의 속도로 데이터를 프로세싱한다. 이 범위의 데이터 레이트들을 달성하기 위해, 입력 배치(batching)이 자주 사용된다. 이 범위의 데이터 레이트들은 대략 백만 OPS의 데이터 레이트들로 조작할 수 있는 보다 최신 주변 디바이스들과 보조를 맞출 수 없다.

[0010] 본 개시내용에 기술된 본 발명의 어레이먼트들에 따르면, 하드웨어 가속기들에 대한 고속 데이터 처리량을 제공할 수 있는 회로 아키텍처가 제공된다. 회로 아키텍처는 저장 디바이스들에서 흔히 볼 수 있는 NVMe(Non-Volatile Memory Express)와 같은 고속 입/출력(I/O) 인터페이스들의 회선 레이트(line rate)와 매칭하는 데이터 처리량들을 제공할 수 있다. 예를 들어, 본원에 설명된 예시적인 회로 아키텍처들을 사용하여, 대략 백만 OPS의 데이터 처리 레이트들이 달성될 수 있다. 본 개시내용에 기술된 본 발명의 어레이먼트가 데이터 저장과 관련된 것들을 넘어 다양한 하드웨어 가속 디바이스들 및/또는 애플리케이션들 중 임의의 것에서 사용될 수 있음이 이해되어야 한다.

[0011] 예시적인 구현에서, 본원에 기술된 회로 아키텍처는 하드웨어 가속기들 및 그 안에 포함된 가속기 엔진들, 예를 들어 컴퓨팅 유닛들과 함께 사용될 수 있다. 하나 이상의 가속기 엔진들이 내부에 구현되어 있는 하드웨어 가속기는 PCIe(Peripheral Component Interconnect Express) 버스와 같은 통신 버스를 통해 호스트 중앙 프로세싱 유닛(CPU)에 통신 가능하게 링크될 수 있다. 호스트 CPU는 운영 소프트웨어를 실행할 수 있으므로 호스트 CPU에 의해 실행되는 호스트 애플리케이션들은 하드웨어 가속기에 포함된 가속기 엔진(들)의 조작을 통해 미세 제어를 제공할 수 있다. 또한, 회로 아키텍처는 운영 소프트웨어를 실행할 때, 호스트 CPU가 가속기

엔진(들)을 계속 비지 상태(busy)로 유지되게 하는 가속기 엔진(들)에 대한 하드웨어 인터페이스를 제공하여, 지속적인 데이터 처리량을 달성하는 동시에 또한 호스트 CPU와 가속기 엔진(들) 사이에서 트랜잭션 오버헤드(transactional overhead)를 감소시킨다.

- [0012] [0020] 본 발명의 어레인지먼트들의 추가 양상들은 도면들을 참조하여 아래에서 더 상세히 설명된다. 예시의 단순성과 명확성을 위해, 도면들에 도시된 요소들은 반드시 실체대로 도시되지 않았다. 예를 들어, 요소들의 일부의 치수는 명확성을 위해 다른 요소들에 비해 과장될 수 있다. 또한, 적절한 것으로 간주되는 경우, 참조 번호들은 대응하거나, 유사하거나, 비슷한 피처(feature)들을 나타내기 위해 도면들 사이에서 반복된다.
- [0013] [0021] 도 1은 하드웨어 가속을 위한 예시적인 회로 아키텍처(100)를 예시한다. 회로 아키텍처(100)는 컴퓨팅 유닛(110)에 의해 처리될 수 있는 데이터 처리량을 증가시킬 수 있다. 일 양상에서, 회로 아키텍처(100)는 하드웨어 가속을 위해 호스트 CPU가 조작을 컴퓨팅 유닛으로 오프로드하기 위한 트랜잭션 오버헤드를 적어도 부분적으로 줄임으로써 데이터 처리량을 증가시킨다.
- [0014] [0022] 도 1의 예에서, 회로 아키텍처(100)는 호스트 CPU(102), 스위치(104), RAM(random-access memory)(106), 어댑터(108) 및 컴퓨팅 유닛(110)을 포함한다. 호스트 CPU(102)는 통신 채널(112)에 의해 스위치(104)에 결합된다. 예시적인 구현에서, 호스트 CPU(102)는 호스트 컴퓨터(도시되지 않음)에 포함되는 반면, 스위치(104), 어댑터(108), 컴퓨팅 유닛(110) 및 선택적으로 RAM(106)은 하드웨어 가속기에 위치된다. 예시적인 호스트 컴퓨터(602)가 도 6에 도시되어 있다. 하드웨어 가속기는 IC로 구현될 수 있다. 그 경우에, 통신 채널(112)은 통신 버스로서 구현된다. 통신 버스의 예는 PCIe(Peripheral Component Interconnect Express) 버스이다. 그러한 구현에서, 스위치(104)는 PCIe 엔드포인트로서 구현될 수 있다.
- [0015] [0023] 다른 예시적인 구현에서, 호스트 CPU(102)는 하드웨어 가속기에 내장된다. 예를 들어, 호스트 CPU(102)는 스위치(104), 어댑터(108), 컴퓨팅 유닛(110) 및 선택적으로 RAM(106)과 동일한 IC에 내장된다. 그 경우에, 호스트 CPU(102)는 하나 이상의 온-칩 와이어들 또는 상호연결들을 통해 스위치(104)에 연결될 수 있다. 그러한 구현에서, 스위치(104)는 온-칩 상호연결로서 구현될 수 있다. 예를 들어, 스위치(104)는 크로스-바 스위치로 구현될 수 있다. 프로세서들/CPU들을 포함하는 예시적인 IC 아키텍처들은 도 7 및 도 8에 예시된다. 호스트 CPU(102)는 호스트 컴퓨터에서 구현되든지 설명된 다른 구성요소들과 함께 IC에 내장되든지 간에 아래에 설명된 바와 같이 조작할 수 있다.
- [0016] [0024] 스위치(104)는 데이터 버스(114)를 통해 RAM(106), 어댑터(108) 및 선택적으로 컴퓨팅 유닛(110)에 결합된다. 하나의 예시적인 구현에서, 데이터 버스(114)는 고속 데이터 버스로 구현될 수 있다. 데이터 버스(114)는 메모리-매핑 버스로 구현될 수 있다. 메모리-매핑 버스의 예는 AMBA(Advanced Microcontroller Bus Architecture) AXI(eXtensible Interface) 버스가 있다. 메모리-매핑 AXI 버스는 스위치(104)가 온-칩 연결들을 확립하게 하고 버스에 결합된 하나 이상의 슬레이브 회로 블록들과 통신하게 한다. 본 개시내용 내에서, AXI는 버스의 예로서 제공되고 제한하려는 의도가 아니다. 다른 유형들의 버스들은 AXI 버스 대신 사용될 수 있다.
- [0017] [0025] 스위치(104)는 또한 제어 버스(116)를 통해 어댑터(108)에 결합된다. 제어 버스(116)는 데이터 버스(114)보다 더 느린 버스일 수 있다. 또한, 제어 버스(116)는 포인트-투-포인트 인터페이스로서 구현될 수 있는 반면 데이터 버스(114)는 그렇지 않다. 예시적인 구현에서, 제어 버스(116)는 AXI-Lite(예를 들어, AXI4-Lite) 준수 버스로서 구현될 수 있다. 본원에 기술된 예시적인 구현들에서, 데이터 버스(114)는 제어 버스(116)보다 더 높은 클럭 레이트에서 조작하고 더 높은 데이터 처리량 또는 대역폭을 제공할 수 있다. 본 개시내용 내에서, 데이터 버스(114) 및 제어 버스(116)는 온-칩 상호연결들의 예들로서 간주될 수 있다.
- [0018] [0026] 어댑터(108)는 카운터(118), 제어기(120) 및 선입선출(FIFO) 메모리(122)를 포함할 수 있다. 예시적인 구현에서, 제어기(120)는 카운터(118) 및 FIFO 메모리(122)의 조작을 조정할 수 있는 상태 머신 회로로서 구현될 수 있다. 어댑터(108)는 컴퓨팅 유닛(110)에 결합되어 통신 채널(124)을 통해 인수들을 제공하고 제어 채널(126)을 통해 제어 신호들을 제공한다.
- [0019] [0027] 일 양상에서, RAM(106)은 스위치(104), 어댑터(108) 및 컴퓨팅 유닛(110)을 포함하는 동일한 IC의 프로그램 가능 회로 또는 프로그램 가능 로직 내에 포함된 하나 이상의 블록 RAM(BRAM)들 또는 다른 RAM 회로 블록들로 구현될 수 있다. 예를 들어, 하드웨어 가속기가 FPGA(field programmable gate array) 또는 일부 프로그램 가능 회로 및/또는 로직을 포함하는 다른 IC와 같은 프로그램 가능 IC인 경우, RAM(106)은 IC의 로직 및/또는 프로그램 가능 회로를 사용하여 구현될 수 있다.

- [0020] [0028] 다른 양상에서, RAM(106)은 오프-칩 메모리로서 구현될 수 있다. 예를 들어, RAM(106)은 하드웨어 가속기와 동일한 회로 기판 또는 카드에 배치되는 DDR(Double Data Rate) RAM으로 구현될 수 있다. 그 경우에, 하드웨어 가속기는 데이터 버스(114)를 통해 스위치(104) 및 어댑터(108)에 결합된 메모리 제어기(도시되지 않음)를 포함할 수 있다. 메모리 제어기는 판독 및 기록 조작들을 실행하기 위해 오프-칩 메모리에 결합될 수 있다.
- [0021] [0029] 다른 양상에서, RAM(106)은 높은 대역폭 메모리(HBM)로서 구현될 수 있다. 예에서, 하드웨어 가속기는 HBM이 스위치(104), 어댑터(108) 및 제어기(120)와는 상이한 하드웨어 가속기의 다이 상에 구현되는 멀티-다이 IC일 수 있다. HBM은 여전히 동일한 패키지로 구현될 수 있다. 다른 예에서, HBM은 IC와 동일한 회로 기판 또는 카드에 배치되는 오프-칩 메모리일 수 있다.
- [0022] [0030] 도 1의 예에서, 호스트 CPU(102)가 하드웨어 가속기와 별개의 그리고 별도의 호스트 컴퓨터 시스템에서 구현되는 경우, RAM(106)은 하드웨어 가속기에 로컬인 것으로 간주된다. 즉, RAM(106)은 호스트 CPU(102)를 포함할 수 있는 호스트 컴퓨터 시스템에 포함될 수 있는 임의의 RAM 또는 메모리와 구별된다.
- [0023] [0031] 아키텍처(100)를 사용하여, 호스트 CPU(102)는 아키텍처(100)가 사용되지 않았을 때보다 더 큰 데이터 처리량을 달성하기 위해 컴퓨팅 유닛(110)으로 조작들을 오프로딩할 수 있다. 예를 들어, 호스트 CPU(102)는 컴퓨팅 유닛(110)에 제공된 인수들을 기술자로 조합할 수 있다. 기술자의 인수들은 컴퓨팅 유닛(110)이 오프로드된 조작(들)을 수행하기 위해 조작할 데이터이다. 본원에 기술된 본 발명의 어레이먼트들에 따르면, 호스트 CPU(102)에 의해 제공된 기술자는 RAM(106)에 저장된다. 기술자는 데이터 버스(114)를 사용하여 RAM(106)에 제공된다.
- [0024] [0032] 기술자가 메모리에 제공되기 때문에, 호스트 CPU(102)는 레지스터 기록 조작들과 반대로 메모리 기록 트랜잭션들을 사용하여 기술자를 전송할 수 있다. 예를 들어, CPU가 컴퓨팅 유닛에 인수들을 제공하는 기존 하드웨어 가속 아키텍처들에서, CPU는 인수들(예를 들어, 데이터)을 컴퓨팅 유닛의 레지스터-기반 인터페이스에 기록함으로써 이를 수행한다. CPU는 CPU의 기본 레지스터 기록 명령들을 실행하여 컴퓨팅 유닛의 레지스터 인터페이스에 기록한다. 일반적으로, 각각의 레지스터 기록 명령은 작거나 제한된 양의 데이터, 예를 들어 대략 4 바이트들의 데이터를 전달한다. 레지스터 기록 조작들의 예들은 "readb()", "readl()", "readq()" 또는 메모리 매핑 레지스터 액세스에 사용되는 다른 동등한 기록 조작들을 포함하지만, 이에 제한되는 것은 아니다. 나열된 예들과 같은 조작들은 예를 들어 1 바이트들, 4 바이트들 또는 8 바이트들의 양들로 데이터를 전달한다. 따라서, 큰 인수들을 컴퓨팅 유닛으로 전송하는 것은 더 많은 시간을 요구하는 더 많은 레지스터 기록 명령들이 실행되어야 한다는 점에서 더 큰 트랜잭션 오버헤드를 요구한다.
- [0025] [0033] 비교에 의해, 메모리 기록들을 지원하는 호스트 CPU(102)의 기본 명령들은 종종 버스트 데이터 전송들을 구현할 수 있다. 디바이스들의 제어 레지스터들에 액세스하는 데 사용되는 레지스터 기록 조작들과 대조적으로, "memcpy()" 조작은 더 많은 양의 데이터를 전송할 수 있는 메모리 기록 조작의 예이다. 예시적이고 비제한적인 예로서, 호스트 CPU(102)에 의해 실행되는 하나의 레지스터 기록 명령이 4 바이트들의 데이터를 전달할 수 있는 반면, 호스트 CPU(102)에 의해 개시된 메모리 기록 명령은 32 바이트들의 데이터를 버스팅할 수 있다. 컴퓨팅 유닛(110)의 레지스터들에 기록하는 대신 설명된 바와 같이 메모리 전송들을 사용하여 호스트 CPU(102)에서 RAM(106)으로 기술자들을 전송함으로써, 하드웨어 가속 디바이스 및 컴퓨팅 유닛(110)으로 데이터를 전송하는 데 필요한 트랜잭션 오버헤드 및 시간이 감소될 수 있어서, 데이터 처리량이 증가한다.
- [0026] [0034] 호스트 CPU(102)는 데이터 버스(114)를 통해 기술자를 RAM(106)에 직접 기록할 수 있다. 또한, 호스트 CPU(102)는 기술자가 RAM(106)에 기록된 주소를 어댑터(108)에 제공할 수 있다. 어드레스를 제공할 때, 호스트 CPU(102)는 어드레스 값의 레지스터 기록을 수행한다. 호스트 CPU(102)에 의해 제공되는 기술자의 주소는 제어 버스(116)를 통해 어댑터(108)에 제공된다. 어드레스는 제어기(120)의 제어 하에 FIFO(122)에 저장될 수 있다.
- [0027] [0035] 일부 경우들에서, 호스트 CPU(102)는 레지스터 판독 조작이 수행될 때까지 레지스터 기록 조작들이 내부적으로 큐잉되는 휴리스틱들을 구현했을 수 있다. 도 1의 예시적인 아키텍처는 레지스터 기록 조작들 대신 인수들을 전송하기 위해 메모리 기록 조작들을 활용하여 이러한 휴리스틱들을 극복한다. 이것은 호스트 CPU(102)가 큐잉되거나 그렇지 않으면 하드웨어 가속기에 데이터 제공을 지연시키는 것을 방지하는데, 이는 메모리 기록들이 임의의 레지스터 판독/기록 휴리스틱에 의해 영향을 받지 않고 수행될 수 있기 때문이다.
- [0028] [0036] FIFO(122)는 RAM(106)에 저장된 기술자들의 주소들의 목록을 저장할 수 있다. 어댑터(108)는 컴퓨팅 유닛(110)이 지연 없이 조작들을 계속 수행할 수 있도록 컴퓨팅 유닛(110)에 인수들을 계속해서 제공하도록 컴

퓨팅 유닛(110)의 조작을 제어할 수 있다. 예를 들어, 제어기(120)는 FIFO(122)로부터 주소를 추출하고 RAM(106)으로부터 주소에 저장된 기술자를 검색할 수 있다. 제어기(120)는 통신 채널(124)을 통해 검색된 기술자로부터 컴퓨팅 유닛(110)으로 인수들을 제공할 수 있다. 일단 인수들이 컴퓨팅 유닛(110)에 제공되면, 제어기(120)는 제어 버스(126)를 통해 컴퓨팅 유닛(110)을 시작하여 인수들에 대한 조작을 시작할 수 있다.

[0029] [0037] 일 양상에서, 어댑터(108)는 컴퓨팅 유닛(110)에 대한 고속 인터페이스 역할을 한다. 호스트 CPU(102)는 인수들을 제공하기 위해 컴퓨팅 유닛(110)의 레지스터들에 기록하는 데 시간을 소비할 필요가 없다. 대신에, 호스트 CPU(102)는 보다 효율적인 메모리 기록 명령들을 실행함으로써 데이터 버스(114)를 사용하여 기술자들을 RAM(106)에 계속 제공할 수 있다. 호스트 CPU(102)에 의해 수행된 레지스터 기록 조작들의 개수는 호스트 CPU(102)가 RAM(106)에 저장된 각각의 기술자의 주소를 어댑터(108)에 기록만 하면 되므로 감소된다. 어댑터(108)는 데이터 버스(114)를 통한 주소를 사용하여 기술자를 검색할 수 있다. 이어서 어댑터(108)는 호스트 CPU(102)로부터 오프로드된 조작을 수행하는 데 필요한 인수들을 컴퓨팅 유닛(110)에 제공하기 위해 레지스터 기록 조작들을 수행한다.

[0030] [0038] 도 1의 예에서, 호스트 CPU(102)는 컴퓨팅 유닛(110)의 특정 레지스터 인터페이스를 알 필요가 없다. 호스트 CPU(102)는 인수들을 포함하는 기술자를 메모리에 기록할 수 있다. 어댑터(108)는 기술자를 파싱하여 그 안에 포함된 인수들을 추출하고 컴퓨팅 유닛(110)의 레지스터 인터페이스의 적절한 레지스터들에 인수들을 제공하도록 구성된다. 이것은 호스트 CPU(102)가 어댑터(108)가 컴퓨팅 유닛(110)의 조작을 관리하는 동안 컴퓨팅 유닛(110)이 수행할 추가 오프로드된 조작들을 큐잉하기 위해 RAM(106)에 기술자들을 계속 기록할 수 있음을 의미한다. 이는 컴퓨팅 유닛(110)이 호스트 CPU(102)가 추가 인수들을 전송하기를 기다리지 않고 조작을 완료한 후 인수들의 다음 세트를 계속 프로세싱하게 하도록 한다. 컴퓨팅 유닛(110)은 기술자들이 FIFO(122)로부터 취한 주소들에 기반하여 RAM(106)에서 프로세싱을 기다리기 때문에 인수들에 대해 계속해서 조작할 수 있다.

[0031] [0039] 카운터(118)는 호스트 CPU(102)에서 하드웨어 가속기로 전송되고 완료된 작업들, 예를 들어 기술자들의 수를 추적할 수 있다. 완료된 작업 또는 기술자는 컴퓨팅 유닛(110)이 프로세싱을 완료한 것이다. 일 양상에서, 제어기(120)는 컴퓨팅 유닛(110)이 기술자의 실행을 완료하는 시기를 결정하기 위해 컴퓨팅 유닛(110)의 특정 레지스터를 폴링(polling)할 수 있다. 다른 양상에서, 컴퓨팅 유닛(110)은 어댑터(108), 예를 들어 제어기(120)에 대한 인터럽트를 생성할 수 있고, 이는 컴퓨팅 유닛(110)이 기술자의 실행을 완료했음을 표시한다. 어쨌든, 제어기(120)는 컴퓨팅 유닛(110)이 기술자의 실행을 완료했다는 결정에 응답하여, 카운터(118)를 증분시킨다. 또한, 이어서 제어기(120)는 통신 채널(124)을 통해 컴퓨팅 유닛(110)에 다른 기술자를 전송할 수 있고 제어 채널(126)을 통해 컴퓨팅 유닛(110)을 시작할 수 있다.

[0032] [0040] 호스트 CPU(102)는 하드웨어 가속기로 오프로드된 작업들의 내부 카운트를 유지할 수 있다. 호스트 CPU(102)는 또한 작업들이 하드웨어 가속기로 전송된 순서를 추적할 수 있다. 일 양상에서, 호스트 CPU(102)는 제어 버스(116)를 통해 카운터(118)의 내용들을 관독할 수 있다. 호스트 CPU(102)에 의해 유지되는 내부 카운트를 카운터(118)로부터 관독된 카운트와 비교함으로써, 호스트 CPU(102)는 하드웨어 가속기에 대한 미처리 작업들의 수를 결정할 수 있다. 즉, 호스트 CPU(102)는 내부 카운트와 카운터(118)로부터 관독된 카운트 간의 차이를 결정할 수 있다. 차이는 아직 완료되지 않은 하드웨어 가속기로 오프로드된 작업들의 수를 나타낸다. 추가로, 기술자들이 FIFO(122)의 주소들 저장에 기반한 FIFO 순서로 하드웨어 가속기에 의해 프로세싱되고 호스트 CPU(102)는 작업들이 하드웨어 가속기로 오프로드되는 순서를 추적하기 때문에, 호스트 CPU(102)는 하드웨어 가속기로 오프로드된 작업들 중 어느 것이 완료되고 완료되지 않았는지를 결정할 수 있다. 따라서, 임의의 주어진 시간에, 호스트 CPU(102)는 하드웨어 가속기로 오프로드된 작업들(예를 들어, 기술자들 및 대응 주소들)이 얼마나 많이 프로세싱되었는지(예를 들어, 완료되지 않았는지), 얼마나 많이 프로세싱되었는지, 및 특정 작업들이 프로세싱되도록 남아 있는지 결정할 수 있다.

[0033] [0041] FIFO(122)는 저장된 데이터의 양 및/또는 이용 가능한 공간의 양을 나타내는 하나 이상의 레지스터들을 포함할 수 있다. 예를 들어, FIFO(122)는 FIFO 공석을 카운트하는 빈 레지스터를 포함할 수 있다. 데이터가 FIFO(122)로부터 관독되고/관독되거나 FIFO(122)에서 이용 가능한 공간의 양을 나타내기 위해 FIFO(122)에 기록됨에 따라 빈 레지스터가 조정(예를 들어, 증분 및/또는 감분)될 수 있다. 일 양상에서, 이러한 레지스터(들)는 FIFO(122)에서 이용 가능한 공간의 양을 결정하기 위해 제어기(120) 및/또는 호스트 CPU(102)에 의해 관독될 수 있다.

[0034] [0042] 도 2는 도 1의 스위치(104)의 예시적인 구현을 예시한다. 도 2의 예에서, 스위치(104)는 PCIe 통신 링크를 통해 호스트 CPU(102)와 통신할 수 있는 PCIe 엔드포인트로서 구현된다. 스위치(104)는 바이패스될 수 있

는 DMA 회로(202)를 포함할 수 있다. 스위치(104)는 메모리 BAR(Base Address Register)(204) 및 제어 BAR(206)을 더 포함한다. 메모리 BAR(204)은 메모리 매핑 인터페이스(208) 및 데이터 버스(114)에 결합된다. 제어 BAR(206)은 제어 버스(116)를 통해 통신하기 위해 제어 인터페이스(210)에 결합된다. 하나는 데이터 버스(114)용이고 다른 하나는 제어 버스(116)용인 2 개의 상이한 BAR들을 사용하는 것은 호스트 CPU(102)가 제어 버스(116)를 통해 어댑터(108)의 FIFO(122)에 제공되는 기술자들의 주소들과 별도로 데이터 버스(114)를 통해 기술자들과 같은 데이터를 전송할 수 있다. 예를 들어, 메모리 기록 명령들은 데이터를 메모리 BAR(204)로 지향시키는 반면 레지스터 기록 조작들은 데이터를 제어 BAR(206)로 지향시킨다.

[0035] [0043] 호스트 CPU(102)가 하드웨어 가속기에 내장된 예시적인 구현들에서, 스위치(104)는 온-칩 상호연결로 구현될 수 있다. 예시적이고 비제한적인 예로서, 스위치(104)는 AXI 버스로 구현될 수 있다. 그 경우, 도 2에 예시된 바와 같이 BAR들을 사용하는 대신에, 스위치(104)의 BAR들은 각각 데이터 버스(114) 및 제어 버스(116)에 연결되는 범용 포트들을 사용하는 애퍼처(aperture)들(예를 들어, 슬레이브/엔드포인트들의 전체 어드레싱 범위의 서브세트들)로 대체될 수 있다. 이 예에서, 슬레이브는 어댑터(108) 및/또는 RAM(106)일 수 있다.

[0036] [0044] 도 3은 하드웨어 가속을 위한 아키텍처(100)의 다른 예를 예시한다. 도 3의 예에서, 임베디드 프로세서(302) 및 네트워크 인터페이스(304)는 하드웨어 가속기에 포함된다. RAM(106)은 도 1과 관련하여 설명된 상이한 유형들의 메모리들 중 임의의 것으로 구현될 수 있다.

[0037] [0045] 일 양상에서, 임베디드 프로세서(302)는 하드와이어드 프로세서로서 구현될 수 있다. 다른 양상에서, 프로세서(302)는 소프트웨어-프로세서로서 구현될 수 있다. 소프트웨어 프로세서는 프로그램 가능 회로 또는 IC의 프로그램 가능 로직을 사용하여 구현되는 프로세서이다. 어느 경우든, 임베디드 프로세서(302)는 스위치(104), 어댑터(108), 컴퓨팅 유닛(110), 사용된 메모리 유형에 따라 선택적으로 RAM(106), 및 아키텍처(100) 구현에 사용되는 IC의 유형에 따라 선택적으로 호스트 CPU(102)와 동일한 하드웨어 가속기 또는 IC에 위치된다.

[0038] [0046] 네트워크 인터페이스(304)는 이더넷 네트워크와 같은 네트워크(306)를 통해 다른 오프-칩 디바이스들 및/또는 노드들과 통신할 수 있는 네트워크 인터페이스와 같은 입력/출력(I/O) 인터페이스일 수 있다. 도 1의 예에서, 아키텍처(100)는 NIC(Network-Interface-Controller) 구현일 수 있다.

[0039] [0047] 도 3의 예에서, 데이터, 예를 들어 기술자들은 네트워크(306) 상의 노드들 및/또는 디바이스들로부터 네트워크 인터페이스(304)를 통해 수신될 수 있다. 그 경우에, 컴퓨팅 유닛(110)에 의해 수행될 조작들은 네트워크(306) 및/또는 호스트 CPU(102)로부터 발생할 수 있다. 임베디드 프로세서(302)는 네트워크 인터페이스(304)로부터 네트워크(306)를 통해 수신된 기술자들을 수신하고 기술자들을 데이터 버스(114)를 통해 RAM(106)에 기록할 수 있다. 추가로, 임베디드 프로세서(302)는 제어 버스(308)를 통해 RAM(106)에 저장된 기술자들의 주소들을 FIFO(122)에 기록할 수 있다. 제어 버스(308)는 임베디드 프로세서(302)와 제어 버스(116)와 분리되고 독립적인 어댑터(108) 사이의 점대점 통신 링크로 구현될 수 있다. 예를 들어, 제어 버스(308)는 또한 AXI-Lite 호환 버스로 구현될 수 있다. 어댑터(108) 및 컴퓨팅 유닛(110)은 컴퓨팅 유닛(110)이 호스트 CPU(102) 및/또는 네트워크(306)로부터 발생하는 하드웨어 가속 조작들을 수행할 수 있도록 설명된 바와 같이 조작할 수 있다. 컴퓨팅 유닛(110)은 기술자들의 주소들이 FIFO(122)에 저장되는 순서로 둘 모두의 소스들에서 발생하는 조작들을 수행할 수 있다.

[0040] [0048] 도 4는, 도 1 또는 도 3의 예시적인 아키텍처를 사용하는 하드웨어 가속의 예시적인 방법(400)을 예시한다. 블록(402)에서, 호스트 CPU는 컴퓨팅 유닛에 대한 하나 이상의 인수들을 포함하는 기술자를 생성한다. 호스트 CPU는 컴퓨팅 유닛을 포함하는 하드웨어 가속기와 별도의 호스트 컴퓨터에 포함될 수 있다. 예를 들어, 하드웨어 가속기는 호스트 CPU와 하드웨어 가속기가 PCIe 통신 링크를 통해 통신하는 호스트 컴퓨터의 주변 디바이스인 하드웨어 가속 카드 또는 보드에 배치될 수 있다. 다른 예에서 호스트 CPU는 컴퓨팅 디바이스와 동일한 하드웨어 가속기에 내장된 프로세서이다.

[0041] [0049] 블록(404)에서, 호스트 CPU는 기술자를 하드웨어 가속기에 로컬인 RAM에 기록하기 위해 메모리 기록 조작을 개시한다. 예를 들어, 호스트 CPU는 하나 이상의 기본 메모리 기록 명령들을 실행하며, 이러한 각각의 메모리 기록 명령은 버스트 데이터 전송을 수행할 수 있다. 기술자는 하드웨어 가속기에 로컬인 RAM의 지정된 주소에 기록된다. 호스트 CPU가 호스트 컴퓨터에 배치된 예에서, 호스트 CPU는 하드웨어 가속기의 메모리 BAR에 메모리 기록 조작을 지시한다. 호스트 CPU가 호스트 컴퓨터에 배치되든 하드웨어 가속기에 내장되든, 호스트 CPU는 기술자가 데이터 버스를 통해 RAM으로 전달되고 RAM에 기록되도록 기술자를 데이터 버스로 지향시킨다.

[0042] [0050] 블록(406)에서, 호스트 CPU는 FIFO에 기술자의 주소를 저장하기 위해 어댑터에 대한 레지스터 기록 조

작을 개시한다. 예를 들어, 호스트 CPU는 하나 이상의 기본 레지스터 기록 명령들을 실행한다. 호스트 CPU가 호스트 컴퓨터에 배치된 예에서, 호스트 CPU는 하드웨어 가속기의 제어 BAR에 레지스터 기록 조작을 지시한다. 호스트 CPU가 호스트 컴퓨터에 배치되든 하드웨어 가속기에 내장되든, 호스트 CPU는 주소가 제어 버스를 통해 어댑터로 전달되고 FIFO에 기록되도록 주소를 제어 버스로 지향시킨다.

[0043] [0051] 블록들(408-412)은 하드웨어 가속기가 네트워크 인터페이스 및 임베디드 프로세서를 포함하는 구현들에서 수행될 수 있는 예시적인 조작들을 예시한다. 하드웨어 가속기가 도 3과 관련하여 설명된 바와 같이 네트워크 인터페이스 및 임베디드 프로세서를 포함하는 구현들에서, 블록들(408-412)에 기술된 조작들은 블록들(402-406) 대신, 블록들(402-406)과 교대로, 또는 블록들(402-406)과 동시에 수행될 수 있다.

[0044] [0052] 블록(408)에서, 임베디드 프로세서는 네트워크 인터페이스를 통해 네트워크로부터 컴퓨팅 유닛에 대한 하나 이상의 인수들을 포함하는 기술자를 수신한다. 블록(410)에서, 임베디드 프로세서는 기술자를 RAM에 기록하기 위해 메모리 기록 조작을 개시한다. 예를 들어, 임베디드 프로세서는 하나 이상의 기본 메모리 기록 명령들을 실행하며, 이러한 각각의 메모리 기록 명령은 버스트 데이터 전송을 수행할 수 있다. 임베디드 프로세서는 데이터 버스를 통해 기술자를 RAM에 기록할 수 있다. 기술자는 하드웨어 가속기에 로컬인 RAM의 지정된 주소에 기록된다. 블록(412)에서, 임베디드 프로세서는 FIFO에 기술자의 주소를 저장하기 위해 어댑터에 대한 레지스터 기록 조작을 개시한다. 예를 들어, 임베디드 프로세서는 어댑터의 FIFO에 기술자의 주소를 기록하기 위해 하나 이상의 기본 레지스터 기록 명령들을 실행한다. 일 양상에서, 임베디드 프로세서는 도 3에 예시된 버스 연결을 통해 주소를 기록할 수 있다. 다른 양상에서, 임베디드 프로세서는 임베디드 프로세서가 기술자의 주소를 전달하는 어댑터로 확립되는 별도의 제어 인터페이스(예를 들어, 추가 포인트-투-포인트 연결)를 가질 수 있다.

[0045] [0053] 블록(414)에서, FIFO는 이용 가능한 공간의 양을 나타내기 위해 제어 레지스터(들)를 조정할 수 있다. 블록(416)에서, 어댑터의 제어기는 FIFO에서 주소를 판독하고 주소를 사용하여 RAM으로부터 기술자를 검색한다. 블록(418)에서, 제어기는 기술자를 파싱하여 그 안에 포함된 인수들을 추출한다. 제어기는 기술자로부터의 인수(들)를 컴퓨팅 유닛에 제공한다. 예를 들어, 제어기는 통신 채널(124)을 통해 컴퓨팅 유닛의 레지스터 인터페이스에 인수들을 기록한다.

[0046] [0054] 블록(420)에서, 제어기는 컴퓨팅 유닛의 실행을 시작한다. 예를 들어, 제어기는 제어 채널(126)을 통해 컴퓨팅 유닛에 시작 신호를 제공한다. 블록(422)에서, 제어기는 컴퓨팅 유닛의 실행 완료를 검출한다. 언급된 바와 같이, 제어기는 컴퓨팅 유닛이 조작을 완료하는 시기를 검출하기 위해 컴퓨팅 유닛의 레지스터를 폴링할 수 있거나 컴퓨팅 유닛은 컴퓨팅 유닛이 조작을 완료했음을 나타내는 제어기에 대한 인터럽트를 생성할 수 있다. 블록(424)에서, 제어기는 컴퓨팅 유닛이 조작을 완료했다는 결정에 응답하여 어댑터 내의 카운터에 저장된 카운트를 증분시킨다.

[0047] [0055] 도 4에 설명된 조작들은 제한이 아니라 예시의 목적으로 제공된다. 설명된 방법이 추가 기술자들을 프로세싱하기 위해 계속 반복될 수 있음이 인식되어야 한다. 방법(400)의 부분들은 또한 다른 것들과 동시에 그리고 병렬로 조작할 수 있다. 예를 들어, 블록(402-406)은 블록들(408-412) 및 블록들(414-424)과 동시에 실행될 수 있다.

[0048] [0056] 도 5는 도 1 또는 도 3의 예시적인 아키텍처를 사용하는 하드웨어 가속의 다른 예시적인 방법(500)을 예시한다.

[0049] [0057] 블록(502)에서, 하드웨어 가속기 내의 스위치, 예를 들어 하드웨어 가속 디바이스 상에 배치된 IC는 복수의 기술자들 및 복수의 기술자들에 대한 복수의 주소들을 수신한다. 스위치는 복수의 기술자들과 복수의 주소들을 전송하는 호스트 CPU에 통신 가능하게 결합된다. 기술자들은 IC 내의 컴퓨팅 유닛에 의해 수행될 조작들을 지정한다.

[0050] [0058] 블록(504)에서, 복수의 기술자들은 데이터 버스를 통해 그 안에 저장하기 위해 스위치로부터 RAM으로 전달된다. 블록(506)에서, 복수의 주소들은 제어 버스를 통해 어댑터 회로로 전달된다. 예를 들어, 어댑터 회로는 주소들을 저장하도록 구성된 FIFO를 포함할 수 있다. 데이터 버스는 제어 버스보다 높은 주파수 또는 데이터 레이트로 조작하도록 구성된다.

[0051] [0059] 블록(508)에서, 어댑터는 데이터 버스를 통해 RAM으로부터 기술자들을 검색하고, 기술자들로부터의 인수들을 컴퓨팅 유닛으로 제공하고, 인수들을 사용하여 조작을 개시하기 위해 컴퓨팅 유닛으로 제어 신호들을 제공할 수 있다.

- [0052] [0060] 전술한 구현 및 다른 구현은 각각 선택적으로 다음 특징들 중 하나 이상을 단독으로 또는 조합하여 포함할 수 있다. 일부 예시적인 구현들은 다음 특징들 모두를 조합하여 포함한다.
- [0053] [0061] 일 양상에서, 스위치는 PCIe 스위치이다. PCIe 스위치는 제어 버스에 결합된 제1 BAR, 예를 들어 제어 BAR 및 데이터 버스에 결합된 제2 bar, 예를 들어 메모리 BAR를 포함할 수 있다.
- [0054] [0062] 다른 양상에서, 하드웨어 가속 디바이스의 하드웨어 가속기는 스위치, 어댑터 및 컴퓨팅 유닛을 포함하는 프로그램 가능 IC이다. 하드웨어 가속기는 또한 기술자들을 저장하는 데 사용되는 RAM을 포함할 수 있다. 예시적인 구현에서, RAM은 프로그램 가능 IC의 프로그램 가능 회로를 사용하여 구현된다.
- [0055] [0063] 다른 양상에서, 하드웨어 가속기는 임베디드 프로세서 및 네트워크 인터페이스를 포함한다. 임베디드 프로세서는 네트워크 인터페이스를 통해 네트워크로부터 수신된 추가 기술자들을 RAM에 저장하도록 구성될 수 있다. 임베디드 프로세서는 또한 어댑터의 FIFO 내에 추가 기술자들의 주소들을 기록하도록 구성될 수 있다.
- [0056] [0064] 도 6은 하드웨어 가속을 위한 예시적인 컴퓨팅 환경(600)을 예시한다. 컴퓨팅 환경(600)은 하드웨어 가속 디바이스(604)에 결합된 호스트 컴퓨터(602)를 포함한다.
- [0057] [0065] 호스트 컴퓨터(602)의 구성요소들은 호스트 CPU(606), 메모리(608), 및 메모리(608)를 포함하는 다양한 시스템 구성요소들을 호스트 CPU(606)에 결합하는 버스(610)를 포함할 수 있으나, 이에 제한되는 것은 아니다. 호스트 CPU(606)는 프로그램 코드를 실행할 수 있는 다양한 프로세서들 중 임의의 프로세서로 구현될 수 있다. 예시적인 프로세서 유형들은 x86 유형의 아키텍처(Intel Architecture 32-비트, Intel Architecture 64-비트 등), Power Architecture, ARM 프로세서들 등을 갖는 프로세서들을 포함하지만, 이에 제한되는 것은 아니다. 추가로, 호스트 컴퓨터(602)가 단일 호스트 CPU(606)를 포함하는 것으로 도시되어 있지만, 다른 예들에서, 다수의 호스트 CPU들이 포함될 수 있다.
- [0058] [0066] 버스(610)는 임의의 여러 유형들의 통신 버스들 중 하나 이상을 나타낸다. 통신 버스들의 예들은 ISA(Industry Standard Architecture) 버스, MCA(Micro Channel Architecture) 버스, EISA(Enhanced ISA) 버스, VESA(Video Electronics Standards Association) 로컬 버스, PCI(Peripheral Component Interconnect) 버스 및 PCI Express(PCIe) 버스를 포함하지만, 이에 제한되는 것은 아니다.
- [0059] [0067] 호스트 컴퓨터(602)는 일반적으로 다양한 컴퓨터 판독가능 매체를 포함한다. 그러한 매체는 호스트 컴퓨터(602)에 의해 액세스 가능한 임의의 이용 가능한 매체일 수 있고 휘발성 매체, 비휘발성 매체, 이동식 매체 및/또는 비이동식 매체의 임의의 조합을 포함할 수 있다.
- [0060] [0068] 메모리(608)는 RAM(612) 및/또는 캐시 메모리(614)와 같은 휘발성 메모리 형태의 컴퓨터 판독가능 매체를 포함할 수 있다. 호스트 컴퓨터(602)는 또한 다른 이동식/비이동식, 휘발성/비휘발성 컴퓨터 시스템 저장 매체를 포함할 수 있다. 예로서, 저장 시스템(616)은 그 안에 포함된 비이동식, 비휘발성 자기 및/또는 고체-상태 매체(예를 들어, "하드 드라이브")를 판독 및 기록하기 위해 제공될 수 있다. 도시되지는 않았지만, 이동식, 비휘발성, 자기 및/또는 광학 및/또는 고체-상태 매체를 판독 및 기록하기 위한 하나 이상의 디스크 드라이브들이 제공될 수 있다. 그러한 경우들에서, 각각은 하나 이상의 데이터 매체 인터페이스들에 의해 버스(610)에 연결될 수 있다. 메모리(608)는 본 개시내용에서 설명된 기능들 및/또는 동작들을 수행하도록 구성된 프로그램 모듈들(예를 들어, 프로그램 코드)의 세트(예를 들어, 하나 이상)를 갖는 적어도 하나의 컴퓨터 프로그램 제품의 예이다.
- [0061] [0069] 예를 들어, 프로그램/유틸리티(618)는 프로그램 모듈들(620)의 세트(예를 들어, 하나 이상)를 포함한다. 메모리(608)에 저장되는 프로그램 모듈들(620)은 운영 체제, 하나 이상의 애플리케이션 프로그램들(예를 들어, 사용자 애플리케이션들), 다른 프로그램 모듈들 및/또는 프로그램 데이터를 포함할 수 있으나, 이에 제한되는 것은 아니다. 프로그램 모듈들(620)은 일반적으로 적어도 호스트 컴퓨터(602)에 의해 수행되는 동작들과 관련하여 본원에 기술된 바와 같은 기능들 및/또는 방법론들을 수행한다. 예를 들어, 프로그램 모듈들(620)은 사용자 애플리케이션들 및/또는 소프트웨어 스택을 포함할 수 있다. 소프트웨어 스택은 하드웨어 가속 디바이스(604)와 관련하여 본원에 기술된 호스트 컴퓨터(602) 동작들을 수행할 수 있는 런타임 환경을 구현할 수 있다. 일 양상에서, 프로그램 모듈들(620)은 IC(632)와 통신할 수 있는 드라이버 또는 데몬을 포함한다.
- [0062] [0070] 프로그램/유틸리티(618)는 호스트 CPU(606)에 의해 실행 가능하다. 프로그램/유틸리티(618) 및 호스트 CPU(606)에 의해 사용, 생성 및/또는 조작되는 임의의 데이터 항목들은 호스트 CPU(606)에 의해 이용될 때 기능을 부여하는 기능적 데이터 구조들이다. 본 개시내용 내에서 정의된 바와 같이, "데이터 구조"는 물리적 메모

리 내의 데이터 모델의 데이터 구성의 물리적 구현이다. 이와 같이, 데이터 구조는 메모리의 특정 전기 또는 자기 구조 요소들로 형성된다. 데이터 구조는 프로세서를 사용하여 실행되는 애플리케이션 프로그램에 의해 사용되는 메모리에 저장된 데이터에 물리적 구성을 부여한다.

- [0063] [0071] 호스트 컴퓨터(602)는 버스(610)에 통신 가능하게 결합된 하나 이상의 입력/출력(I/O) 인터페이스들(628)을 포함할 수 있다. I/O 인터페이스(들)(628)는 호스트 컴퓨터(602)가 외부 디바이스들과 통신하게 하고, 사용자(들)가 호스트 컴퓨터(602)와 상호 작용하게 하는 외부 디바이스에 결합하게 하고, 호스트 컴퓨터(602)가 다른 컴퓨팅 디바이스들과 통신하게 하는 외부 디바이스들에 결합하게 하는 등을 한다. 예를 들어, 호스트 컴퓨터(602)는 I/O 인터페이스(들)(628)를 통해 디스플레이(630) 및 하드웨어 가속 디바이스(604)에 통신 가능하게 링크될 수 있다. 호스트 컴퓨터(602)는 I/O 인터페이스(들)(628)를 통해 키보드(도시되지 않음)와 같은 다른 외부 디바이스들에 결합될 수 있다. I/O 인터페이스(들)(628)의 예들은 네트워크 카드들, 모듈들, 네트워크 어댑터들, 하드웨어 제어기들 등을 포함할 수 있으나, 이에 제한되는 것은 아니다.
- [0064] [0072] 예시적인 구현에서, 호스트 컴퓨터(602)가 하드웨어 가속 디바이스(604)와 통신하는 I/O 인터페이스(628)는 PCIe 어댑터이다. 하드웨어 가속 디바이스(604)는 호스트 컴퓨터(602)에 결합되는 회로 기판, 예를 들어 카드에 구현될 수 있다. 예를 들어, 하드웨어 가속 디바이스(604)는 호스트 컴퓨터(602)의 이용 가능한 카드 슬롯, 예를 들어 이용 가능한 버스 및/또는 PCIe 슬롯에 삽입될 수 있다.
- [0065] [0073] 하드웨어 가속 디바이스(604)는 IC(예를 들어, 하드웨어 가속기)(632)를 포함한다. 일 양상에서, IC(632)는 프로그램 가능 IC이다. 하드웨어 가속 디바이스(604)는 또한 IC(632)에 결합된 휘발성 메모리(634) 및 또한 IC(632)에 결합된 비휘발성 메모리(636)를 포함한다. 휘발성 메모리(634)는 IC(632) 외부에 있는 RAM으로 구현될 수 있다. 휘발성 메모리(634)는 프로그램 가능 IC(632)의 "로컬 메모리"로 간주되는 반면, 호스트 컴퓨터(602) 내에 있는 메모리(608)는 프로그램 가능 IC(632)에 로컬인 것으로 간주되지 않는다. 비휘발성 메모리(636)는 플래시 메모리로 구현될 수 있다. 비휘발성 메모리(636)는 또한 IC(632) 외부에 있고 IC(632)에 로컬인 것으로 간주될 수 있다.
- [0066] [0074] 예시적인 구현에서, 도 1 또는 도 3의 RAM(106)은 휘발성 메모리(634)로 구현될 수 있다. 다른 예시적인 구현에서, 도 1 또는 도 3의 RAM(106)은 하드와이어드 회로 블록으로서 또는 프로그램 가능 회로로 IC(132) 내에서 구현될 수 있다.
- [0067] [0075] 예시적인 구현에서, IC(632)는 프로그램 가능 회로(예를 들어, 프로그램 가능 로직)를 포함한다. 예를 들어, IC(632)는 FPGA 또는 적어도 어느 정도의 프로그램 가능 회로를 포함하는 다른 유형의 IC로 구현될 수 있다. 명시적으로 도시되지는 않았지만, IC(632)가 프로그램 가능 회로와 협력하여 사용될 수 있는 하나 이상의 하드와이어드 회로 블록들을 포함할 수 있음을 인식되어야 한다.
- [0068] [0076] 하나 이상의 예시적인 구현들에서, IC(632)는 네트워크(638)에 연결된다. 네트워크(638)는 이더넷 유형의 네트워크로서 구현될 수 있다. 네트워크(638)는 임의의 다양한 상이한 속도들로 조작할 수 있다. 예를 들어, 네트워크(638)는 10G, 25G, 50G, 100G, 200G, 400G 또는 다른 속도 네트워크일 수 있다. 특정 구현들에서, 네트워크(638)는 5G 네트워크이거나, 포함하거나, 5G 네트워크에 결합될 수 있다. 따라서, IC(632)는 IC(632)를 네트워크(638)에 연결, 예를 들어 통신가능하게 링크하는 데 사용되는 도 3의 네트워크 인터페이스(304)와 같은 이더넷 인터페이스(도 6의 예에는 도시되지 않음)를 포함할 수 있다. 예를 들어, IC(632)는 네트워크(638)를 통해 이더넷 스위치 또는 하나 이상의 다른 네트워크 연결 디바이스들에 연결될 수 있다. 예시를 위해, "네트워크"라는 용어는 본원에서 네트워크(638), 예를 들어 이더넷 네트워크를 지칭한다.
- [0069] [0077] 도 6은 본원에 설명된 예들의 사용 또는 기능의 범위에 대한 임의의 제한을 제안하려는 의도가 아니다. 호스트 컴퓨터(602)는 호스트 컴퓨터 및/또는 호스트 CPU에 기인하여 본 개시내용 내에서 설명된 다양한 조작들을 수행할 수 있는 컴퓨터 하드웨어(예를 들어, 시스템)의 예이다.
- [0070] [0078] 호스트 컴퓨터(602)는 하드웨어 가속 디바이스(604)와 함께 사용될 수 있는 컴퓨터의 단지 하나의 예시적인 구현이다. 호스트 컴퓨터(602)는 예를 들어 컴퓨터 또는 서버와 같은 컴퓨팅 디바이스의 형태로 도시된다. 한 양상에서, 호스트 컴퓨터(602)는 데이터센터 내에서 실시될 수 있다. 예를 들어, 호스트 컴퓨터(602)는 독립형 디바이스, 베어 메탈 서버, 클러스터 또는 분산 클라우드 컴퓨팅 환경에서 실시될 수 있다. 분산 클라우드 컴퓨팅 환경에서, 태스크들은 통신 네트워크를 통해 링크된 원격 프로세싱 디바이스들에 의해 수행된다. 분산 클라우드 컴퓨팅 환경에서, 프로그램 모듈들은 메모리 저장 디바이스들을 포함하는 로컬 및 원격 컴퓨터 시스템 저장 매체 둘 모두에 위치될 수 있다.

- [0071] [0079] 본원에서 사용되는 바와 같이, "클라우드 컴퓨팅"이라는 용어는 네트워크들, 서버들, 저장 장치, 애플리케이션들, IC(예를 들어, 프로그램 가능 IC)들 및 /또는 서비스들 같은 구성가능 컴퓨팅 리소스들의 공유 풀에 편리한 주문형 네트워크 액세스를 가능하게 하는 컴퓨팅 모델을 지칭한다. 이러한 컴퓨팅 리소스들은 최소한의 관리 노력이나 서비스 제공자 상호작용으로 신속하게 프로비저닝 및 릴리스될 수 있다. 클라우드 컴퓨팅은 가용성을 촉진하고 주문형 셀프 서비스, 광범위한 네트워크 액세스, 리소스 풀링, 빠른 탄력성 및 측정된 서비스를 특징으로 할 수 있다.
- [0072] [0080] 클라우드 컴퓨팅 환경들, 예지 컴퓨팅 환경들 및/또는 데이터센터들과 같은 일부 컴퓨팅 환경들은 일반적으로 FaaS(FPGA-as-a-Service) 모델을 지원한다. FaaS 모델에서 사용자 기능들 및/또는 네트워크 기능들은 호스트 컴퓨터의 제어 하에 조작하는 프로그램 가능 IC들 내에서 구현되는 회로 설계들로 하드웨어 가속된다. 클라우드 컴퓨팅 모델들의 다른 예들은 NIST(National Institute of Standards and Technology), 특히 NIST의 ITL(Information Technology Laboratory)에 설명된다.
- [0073] [0081] 호스트 컴퓨터(602)는 다수의 다른 범용 또는 특수 목적 컴퓨팅 시스템 환경들 또는 구성들과 함께 조작한다. 호스트 컴퓨터(602)와 함께 사용하기에 적합할 수 있는 컴퓨팅 시스템들, 환경들 및/또는 구성들의 예들은 개인용 컴퓨터 시스템들, 서버 컴퓨터 시스템들, 썬 클라이언트(thin client)들, 씩 클라이언트(thick client)들, 핸드헬드 또는 랩톱 디바이스들, 멀티프로세서 시스템들, 마이크로프로세서-기반 시스템들, 셋톱 박스들, 프로그램 가능 가전 제품들, 네트워크 PC들, 미니 컴퓨터 시스템들, 메인프레임 컴퓨터 시스템들 및 위의 시스템들 또는 디바이스들 중 임의의 것을 포함하는 분산 클라우드 컴퓨팅 환경들 등을 포함하지만, 이에 제한되는 것은 아니다.
- [0074] [0082] 도 7은 하드웨어 가속에 사용될 수 있는 IC에 대한 예시적인 아키텍처(700)를 예시한다. 아키텍처(700)는 도 1 및/또는 도 3과 관련하여 설명된 바와 같이 하드웨어 가속기를 구현하는 데 사용될 수 있다. 예를 들어, 도 1의 예에서, 아키텍처(700)는 스위치(104), 어댑터(108), 컴퓨팅 유닛(110) 및 선택적으로 RAM(106)을 구현하는 데 사용될 수 있다. 도 3의 예에서, 아키텍처(700)는 스위치(104), 어댑터(108), 컴퓨팅 유닛(110), 선택적으로 RAM(106), 임베디드 프로세서(302) 및 네트워크 인터페이스(304)를 구현하는 데 사용될 수 있다.
- [0075] [0083] 일 양상에서, 아키텍처(700)는 내부에 내장된 호스트 CPU를 포함하는 하드웨어 가속기를 구현하는 데 사용될 수 있다. 다른 양상에서, 아키텍처(700)는 도 6과 관련하여 설명된 바와 같이 하드웨어 가속기를 구현하는 데 사용될 수 있고, 여기서 호스트 CPU는 호스트 컴퓨터에 포함된다.
- [0076] [0084] 아키텍처(700)는 프로그램 가능 IC 내에서 구현될 수 있다. 예를 들어, 아키텍처(700)는 FPGA를 구현하는 데 사용될 수 있다. 아키텍처(700)는 또한 SoC(System-on-Chip) 유형의 IC를 나타낼 수 있다. SoC는 프로그램 코드와 하나 이상의 다른 회로들을 실행하는 프로세서를 포함하는 IC이다. 다른 회로들은 하드와이어드 회로, 프로그램 가능 회로 및/또는 로직, 및/또는 이들의 조합으로 구현될 수 있다. 회로들은 서로 및/또는 프로세서와 협력하여 조작할 수 있다.
- [0077] [0085] 도시된 바와 같이, 아키텍처(700)는 여러 상이한 유형들의 프로그램 가능 회로, 예를 들어 로직, 블록들을 포함한다. 예를 들어, 아키텍처(700)는 멀티-기가비트 트랜스시버(MGT)들(701), 구성 가능 로직 블록(CLB)들(702), 랜덤 액세스 메모리 블록(BRAM)들(703), 입력/출력 블록(IOB)들(704), 구성 및 클럭킹 로직(CONFIG/CLOCKS)들(705), 디지털 신호 프로세싱 블록(DSP)들(706), 특수 I/O 블록(707)들(예를 들어, 구성 포트들 및 클럭 포트들) 및 디지털 클럭 관리자들과 같은 다른 프로그램 가능 로직(708), 아날로그-투-디지털 컨버터들, 시스템 모니터링 로직 등을 포함하는 다수의 상이한 프로그램 가능 타일들을 포함할 수 있다.
- [0078] [0086] 일부 IC들에서, 각각의 프로그램 가능 타일은 각각의 인접 타일의 대응하는 INT(711)와의 표준화된 연결들을 갖는 프로그램 가능 상호연결 요소(INT)(711)를 포함한다. 함께 취해진 INT(711)는 예시된 IC를 위한 프로그램 가능 상호연결 구조를 구현한다. 각각의 INT(711)는 또한 도 7의 우측에 포함된 예들에 의해 도시된 바와 같이 동일한 타일 내의 프로그램 가능 로직 요소와의 연결들을 포함한다.
- [0079] [0087] 예를 들어, CLB(702)는 사용자 로직과 단일 INT(711)를 구현하도록 프로그래밍될 수 있는 구성 가능 로직 요소(CLE)(712)를 포함할 수 있다. BRAM(703)은 하나 이상의 INT들(711) 외에 BRL(BRAM logic element)(713)을 포함할 수 있다. 일반적으로, 타일에 포함된 INT들(711)의 수는 타일의 높이에 의존한다. 도시된 바와 같이, BRAM 타일은 5 개의 CLB들과 동일한 높이를 갖지만, 다른 수들(예를 들어, 4 개)이 또한 사용될 수 있다. DSP 타일(706)은 적절한 수의 INT들(711)에 더하여 DSP 로직 요소(DSPL)(714)를 포함할 수 있다.

I/OB(704)는 예를 들어 INT(711)의 하나의 인스턴스에 추가하여 I/O 로직 요소(IOL)(715)의 2 개의 인스턴스를 포함할 수 있다. IOL(715)에 연결된 실제 I/O 패드들은 IOL(715) 영역에 국한되지 않을 수 있다.

- [0080] [0088] 도 7에 도시된 예에서, 예를 들어 구역들(705, 707, 및 708)으로 형성된 다이의 중심 근처의 수평 영역은 구성, 클록 및 다른 제어 로직을 위해 사용될 수 있다. 이 수평 영역에서 연장되는 수직 영역(709)은 프로그램 가능 IC의 폭에 걸쳐 클록들 및 구성 신호들을 분배하는 데 사용될 수 있다.
- [0081] [0089] 도 7에 예시된 아키텍처를 활용하는 일부 IC들은 IC의 많은 부분을 구성하는 일반 기동 구조를 방해하는 추가 로직 블록들을 포함한다. 추가 로직 블록들은 프로그램 가능 블록들 및/또는 전용 회로일 수 있다. 예를 들어 PROC(710)로 묘사된 프로세서 블록은 CLB들 및 BRAM들의 여러 열에 걸쳐 있다.
- [0082] [0090] 일 양상에서, PROC(710)는 IC의 프로그램 가능 회로를 구현하는 다이의 일부로서 제조되는 하드와이어드 프로세서와 같은 전용 회로로서 구현될 수 있다. PROC(710)는 프로그램 코드를 실행할 수 있는 단일 코어와 같은 개별 프로세서에서 하나 이상의 코어들, 모듈들, 코-프로세서들, 인터페이스들 등을 갖는 전체 프로세서 시스템까지 복잡도 범위의 다양한 상이한 프로세서 유형들 및/또는 시스템들 중 임의의 것을 나타낼 수 있다.
- [0083] [0091] 다른 양상에서, PROC(710)는 아키텍처(700)에서 생략될 수 있고 다른 종류들의 설명된 프로그램 가능 블록들 중 하나 이상으로 대체될 수 있다. 추가로, 이러한 블록들은 PROC(710)의 경우와 같이 프로그램 가능 회로의 다양한 블록들이 프로그램 코드를 실행할 수 있는 프로세서를 형성하기 위해 사용될 수 있다는 점에서 "소프트 프로세서"를 형성하는 데 활용될 수 있다.
- [0084] [0092] 예시적인 구현에서, PROC(710)는 호스트 CPU를 구현하는 데 사용될 수 있다. 따라서, IC가 임베디드 프로세서를 더 포함하는 경우들에서, 임베디드 프로세서는 소프트-프로세서로 구현될 수 있다. 호스트 CPU가 호스트 컴퓨터에 포함되는 다른 예에서, PROC(710)는 임베디드 프로세서를 구현할 수 있다. 아키텍처(700)가 하드와이어드 프로세서를 포함하지 않는 경우, 포함된다면, 임베디드 프로세서는 소프트-프로세서로서 구현될 수 있다.
- [0085] [0093] "프로그램 가능 회로"라는 어구는 IC 내의 프로그램 가능 회로 요소들, 예를 들어 본원에 기술된 다양한 프로그램 가능하거나 구성 가능한 회로 블록들 또는 타일들뿐만 아니라, IC에 로드된 구성 데이터에 따라 다양한 회로 블록들, 타일들 및/또는 요소들을 선택적으로 결합하는 상호연결 회로를 지칭한다. 예를 들어, CLB들(702) 및 BRAM들(703)과 같이 PROC(710) 외부에 있는 도 7에 도시된 회로 블록들은 IC의 프로그램 가능 회로로 간주된다.
- [0086] [0094] 일반적으로, 프로그램 가능 회로의 기능은 구성 데이터가 IC에 로드될 때까지 확립되지 않는다. 구성 비트들의 세트는 FPGA와 같은 IC의 프로그램 가능 회로를 프로그래밍하는 데 사용될 수 있다. 구성 비트(들)는 일반적으로 "구성 비트스트림"으로 지칭된다. 일반적으로, 프로그램 가능 회로는 IC에 구성 비트스트림을 먼저 로드하지 않고 조작하거나 기능하지 않는다. 구성 비트스트림은 프로그램 가능 회로 내에서 특정 회로 설계를 효과적으로 구현한다. 회로 설계는 예를 들어 프로그램 가능 회로 블록들의 기능적 양상들과 다양한 프로그램 가능 회로 블록들 간의 물리적 연결을 지칭한다.
- [0087] [0095] 하나 이상의 예시적인 구현들에서, 스위치(104), RAM(106), 어댑터(108), 컴퓨팅 유닛(110), 및/또는 네트워크 인터페이스(304)와 같은 회로들은 IC의 프로그램 가능 회로를 사용하여 구현될 수 있다. 예를 들어, RAM(106)은 BRAM들(703) 또는 프로그램 가능 회로에 포함된 다른 RAM 회로 블록들 중 하나 이상을 사용하여 구현될 수 있다. 다른 예시적인 구현에서, 스위치(104) 및/또는 네트워크 인터페이스(304)와 같이 언급된 회로들 중 하나 이상은 하드와이어드 회로로 구현될 수 있다.
- [0088] [0096] 다른 예시적인 구현에서, 아키텍처(700)는 다중-다이 IC의 다이를 구현하는 데 사용될 수 있고, 동일한 패키지 내의 다른 다이는 HBM을 구현한다. HBM은 RAM(106)으로서 조작할 수 있다.
- [0089] [0097] "하드와이어드" 또는 "경화(hardened)", 즉 프로그래밍이 불가능한 회로는 IC의 일부로 제조된다. 프로그램 가능 회로와 달리, 하드와이어드 회로 또는 회로 블록들은 구성 비트스트림의 로딩을 통해 IC 제조 후에 구현되지 않는다. 하드와이어드 회로는 일반적으로 예를 들어 구성 비트스트림을 IC, 예를 들어 PROC(710)에 먼저 로드하지 않고 기능하는 전용 회로 블록들 및 상호연결들을 갖는 것으로 간주된다.
- [0090] [0098] 일부 인스턴스들에서, 하드와이어드 회로는 레지스터 설정들 또는 IC 내의 하나 이상의 메모리 요소들에 저장된 값들에 따라 설정되거나 선택될 수 있는 하나 이상의 조작 모드들을 가질 수 있다. 조작 모드들은 예를 들어 구성 비트스트림을 IC로 로딩함으로써 설정될 수 있다. 이러한 능력에도 불구하고, 하드와이어드 회

로는 조작 가능하고 IC의 일부로 제조될 때 특정 기능을 갖기 때문에 프로그램 가능 회로로 간주되지 않는다.

- [0091] [0099] SoC의 경우, 구성 비트스트림은 프로그램 가능 회로 내에서 구현될 회로와 PROC(710) 또는 소프트웨어-프로세서에 의해 실행될 프로그램 코드를 지정할 수 있다. 일부 경우들에서, 아키텍처(700)는 구성 비트스트림을 적절한 구성 메모리 및/또는 프로세서 메모리에 로드하는 전용 구성 프로세서를 포함한다. 전용 구성 프로세서는 사용자-지정 프로그램 코드를 실행하지 않는다. 다른 경우들에서, 아키텍처(700)는 PROC(710)를 활용하여 구성 비트스트림을 수신하고, 구성 비트스트림을 적절한 구성 메모리에 로드하고/하거나, 실행을 위한 프로그램 코드를 추출할 수 있다.
- [0092] [00100] 도 7은 프로그램 가능 회로, 예를 들어 프로그램 가능 패브릭을 포함하는 IC를 구현하는 데 사용될 수 있는 예시적인 아키텍처를 예시하기 위한 것이다. 예를 들어, 열에 있는 로직 블록들의 수, 열들의 상대적 폭, 열들의 수와 순서, 열들에 포함된 로직 블록들의 유형, 로직 블록들의 상대적 크기들, 및 도 7의 우측에 포함된 상호연결/로직 구현들은 순전히 예시이다. 예를 들어 실제 IC에서, 사용자 회로 설계의 효율적인 구현을 용이하게 하기 위해, 일반적으로 CLB들이 나타나는 곳마다 하나 초과인 인접한 CLB들의 열이 포함된다. 그러나, 인접한 CLB 열들의 수는 IC의 전체 크기에 따라 다를 수 있다. 추가로, IC 내의 PROC(710)와 같은 블록들의 크기 및/또는 포지셔닝은 단지 예시를 위한 것이고 제한들을 위한 것이 아니다.
- [0093] [00101] 도 8은 하드웨어 가속을 위한 IC에 대한 또 다른 예시적인 아키텍처(800)를 예시한다. 아키텍처(800)는 하드웨어 가속기를 구현하는 데 사용될 수 있고 IC의 SoC 유형을 나타낸다. 이 예에서, 아키텍처(800)는 프로세서 시스템(PS)(802)을 포함한다. "시스템"으로 설명되지만, PS(802)는 아키텍처(800)의 서브시스템의 예이다. 일반적으로, PS(802)는 아키텍처(800)의 일부로 제조되는 하드와이어드 회로로 구현된다. PS(802)는 각각 프로그램 코드를 실행할 수 있는 다양한 상이한 프로세서 유형들 중 임의의 프로세서로 구현되거나, 이를 포함할 수 있다. 프로세서들은 서로 다른 아키텍처들 및/또는 명령 세트들을 가질 수 있다.
- [0094] [00102] 이 예에서, PS(802)는 실시간 프로세싱 유닛(RPU)(806), 애플리케이션 프로세싱 유닛(APU)(808), 그래픽 프로세싱 유닛(GPU)(810), 구성 및 보안 유닛(CSU)(812), 등 같은 다양한 프로세서들을 포함한다. 프로세서들 각각은 프로그램 코드를 실행할 수 있다.
- [0095] [00103] PS(802)는 또한 온-칩 메모리(OCM)(814), 트랜스시버들(816), 주변장치들(818), 상호연결(820), DMA 회로(822), 메모리 제어기(824), 주변장치들(826) 및 다중 입력/출력(MIO) 회로(828)와 같은 다양한 지원 회로들을 포함한다. 프로세서들과 지원 회로들은 상호연결(820)에 의해 상호연결된다.
- [0096] [00104] 트랜스시버(816)는 외부 핀들(830)에 결합될 수 있다. 아래에서 더 상세히 설명될 프로그램 가능 로직(PL)(850)은 외부 핀들(832)에 결합될 수 있다. 메모리 제어기(820)는 외부 핀들(834)에 결합될 수 있다. MIO(828)는 외부 핀들(836)에 결합될 수 있다. PS(802)는 외부 핀들(838)에 결합될 수 있다. APU(808)는 CPU(840), 메모리(842) 및 지원 회로들(844)을 포함할 수 있다.
- [0097] [00105] 도 8의 예에서, 프로세서들(806, 808 및 810) 각각은 하나 이상의 중앙 프로세싱 유닛(CPU)들 및 연관된 회로들, 이를테면 메모리들, 인터럽트 제어기들, DMA(direct memory access) 제어기들, MMU(memory management unit)들, 부동 소수점 유닛(FPU)들 등을 포함할 수 있다. 상호연결(820)은 프로세서들(806, 808, 810)을 상호연결하고 PS(802)의 다른 구성요소들을 프로세서들(806, 808, 및 810)에 상호연결하도록 구성된 다양한 스위치들, 버스들, 통신 링크들 등을 포함한다.
- [0098] [00106] 예시적인 구현에서, CPU(840) 또는 APU(808)의 다른 프로세서는 호스트 CPU를 구현하는 데 사용될 수 있다. 따라서, IC가 임베디드 프로세서를 더 포함하는 경우, 임베디드 프로세서는 소프트웨어-프로세서, RPU(806), 또는 APU(808)에 포함될 수 있는 다른 하드와이어드 프로세서로서 구현될 수 있다. 호스트 CPU가 호스트 컴퓨터에 포함되는 다른 예에서, 임베디드 프로세서는 CPU(840), APU(808)의 다른 하드와이어드 프로세서, RPU(806) 또는 소프트웨어-프로세서로 구현될 수 있다.
- [0099] [00107] OCM(814)은 PS(802) 전체에 분포될 수 있는 하나 이상의 RAM 모듈들을 포함한다. 예를 들어, OCM(814)은 BBRAM(battery backed RAM), TCM(tightly coupled memory) 등을 포함할 수 있다. 메모리 제어기(824)는 외부 DRAM에 액세스하기 위한 동적 DRAM 인터페이스를 포함할 수 있다. 주변장치들(818, 826)은 PS(802)에 대한 인터페이스를 제공하는 하나 이상의 구성요소들을 포함할 수 있다. 예를 들어, 주변장치들(822)은 그래픽 프로세싱 유닛(GPU), 디스플레이 인터페이스(예를 들어, DisplayPort, 고화질 멀티미디어 인터페이스(HDMI) 포트 등), 범용 직렬 버스(USB) 포트들, 이더넷 포트들, 범용 비동기식 트랜스시버(UART) 포트들, SPI(Serial Peripheral Interface) 포트들, GPIO(General Purpose IO) 포트들, SATA(Serial Advanced

Technology Attachment) 포트들, PCIe 포트들 등을 포함할 수 있다. 주변장치들(826)은 MIO(828)에 결합될 수 있다. 주변장치들(818)은 트랜스시버들(816)에 결합될 수 있다. 트랜스시버들(816)은 SERDES(serializer/deserializer) 회로들, MGT들 등을 포함할 수 있다.

- [0100] [00108] 도 8의 예에서, 디바이스(800)는 선택적으로 프로그램 가능 로직(850) 서브시스템을 포함한다. 프로그램 가능 로직(850)은 특정 기능을 수행하도록 프로그래밍될 수 있는 회로이다. 예로서, 프로그램 가능 로직(850)은 필드 프로그램 가능 게이트 어레이 유형의 회로로 구현될 수 있다. 프로그램 가능 로직(850)은 프로그램 가능 회로 블록들의 어레이를 포함할 수 있다.
- [0101] [00109] 도 8의 예에서, 프로그램 가능 로직(850)은 상호연결(820)에 결합될 수 있다. 일 양상에서, 프로그램 가능 로직(850)은 AXI 상호연결을 통해 PS(802)에 결합될 수 있다. 이와 같이, 프로그램 가능 로직(850)에서 구현되는 임의의 컴퓨팅 유닛들은 AXI 상호연결로 구현될 수 있는 상호연결(820)을 통해 PS(802)에 의해 액세스될 수 있다. 도 7 관련하여 논의된 바와 같이, 프로그램 가능 로직(850)의 토폴로지는 하드와이어드 회로와 달리 고도로 구성가능하다.
- [0102] [00110] 디바이스(800)는 프로세서 어레이(예를 들어, 데이터 프로세싱 엔진(DPE) 어레이), 네트워크-온-칩(NoC), 플랫폼 관리 제어기(PMC) 및 하나 이상의 하드와이어드 회로 블록들을 포함하는 하나 이상의 다른 서브시스템들을 포함할 수 있다. 하드와이어드 회로 블록은 아키텍처(800)의 일부로 제작된 특수 목적 회로 블록을 지칭한다.
- [0103] [00111] 하드와이어드이지만, 하드와이어드 회로 블록들은 구성 데이터를 제어 레지스터들에 로드하여 하나 이상의 상이한 조작 모드들을 구현함으로써 구성될 수 있다. 하드와이어드 회로 블록들의 예들은 입력/출력(I/O) 블록들, 회로들 및/또는 아키텍처(800) 외부의 시스템들에 신호들을 전송 및 수신하기 위한 트랜스시버들, 메모리 제어기들 등을 포함할 수 있다. 상이한 I/O 블록들의 예들은 단일-종단 및 의사 차동 I/O들을 포함할 수 있다. 트랜스시버들의 예들은 고속 차동 클록 트랜스시버들을 포함할 수 있다. 하드와이어드 회로 블록들의 다른 예들은 암호화 엔진들, 디지털-투-아날로그 컨버터(DAC)들, 아날로그-투-디지털 컨버터(ADC)들 등을 포함하지만, 이에 제한되는 것은 아니다. 일반적으로, 하드와이어드 회로 블록들은 애플리케이션별 회로 블록들이다.
- [0104] [00112] 설명의 목적을 위해, 특정 명명법은 본원에 개시된 다양한 발명적 개념들의 완전한 이해를 제공하기 위해 제시된다. 그러나, 본원에서 사용된 용어는 본 어레인지먼트들의 특정 양상들을 설명하기 위한 목적일 뿐 제한하려는 의도는 아니다.
- [0105] [00113] 본원에서 정의된 바와 같이, 단수형("a", "an" 및 "the")은 문맥상 명백하게 달리 나타내지 않는 한 복수형을 또한 포함하는 것으로 의도된다.
- [0106] [00114] 본원에서 정의된 바와 같이, "대략"이라는 용어는 거의 정확하거나 정확하고, 값이나 양이 비슷하지만 정확하지는 않음을 의미한다. 예를 들어, "대략"이라는 용어는 인용된 특성, 파라미터 또는 값이 정확한 특성, 파라미터 또는 값의 미리 결정된 양 내에 있음을 의미할 수 있다.
- [0107] [00115] 본원에서 정의된 바와 같이, "적어도 하나", "하나 이상" 및 "및/또는"이라는 용어는 달리 명시적으로 언급되지 않는 조작에서 접속사 및 이접사 둘 모두인 개방형 표현들이다. 예를 들어, "A, B 및 C 중 적어도 하나", "A, B 또는 C 중 적어도 하나", "A, B 및 C 중 하나 이상", "B 또는 C 중 하나 이상" 및 "A, B 및/또는 C"의 표현들 각각은 A 단독, B 단독, C 단독, A 및 B 함께, A 및 C 함께, B 및 C 함께 또는 A, B 및 C 함께를 의미한다.
- [0108] [00116] 본원에서 정의된 바와 같이, "자동으로"라는 용어는 사람의 개입이 없는 것을 의미한다.
- [0109] [00117] 본원에서 정의된 바와 같이, "컴퓨터 판독가능 저장 매체"라는 용어는 명령 실행 시스템, 장치 또는 디바이스에 의해 또는 이와 관련하여 사용하기 위한 프로그램 코드를 포함하거나 저장하는 저장 매체를 의미한다. 본원에 정의된 바와 같이, "컴퓨터 판독가능 저장 매체"는 그 자체로 일시적인 전파 신호가 아니다. 컴퓨터 판독가능 저장 매체는 전자 저장 디바이스, 자기 저장 디바이스, 광학 저장 디바이스, 전자기 저장 디바이스, 반도체 저장 디바이스, 또는 전술한 바의 임의의 적합한 조합일 수 있으나, 이에 제한되는 것은 아니다. 본원에 기술된 다양한 형태들의 메모리는 컴퓨터 판독가능 저장 매체의 예들이다. 컴퓨터 판독가능 저장 매체의 보다 구체적인 예들의 비제한적인 목록은: 휴대용 컴퓨터 디스켓, 하드 디스크, RAM, 판독-전용 메모리(ROM), 소거가능 프로그램 가능 판독-전용 메모리(EPROM 또는 플래시 메모리), EEPROM(Electrically Erasable Programmable Read-Only Memory), SRAM(Static Random Access Memory), 휴대용 CD-ROM(Compact Disc Read-

Only Memory), DVD(Digital Versatile Disk), 메모리 스틱, 플로피 디스크 등을 포함할 수 있다.

- [0110] [00118] 본원에서 정의된 바와 같이, 용어 "만약"이라는 용어는 문맥에 따라 "~할 때" 또는 "~에 따라" 또는 "~에 응답하여" 또는 "~에 응답하는"을 의미한다. 따라서, "결정된 경우" 또는 "[언급된 조건 또는 이벤트]가 검출된 경우"라는 문구는 "결정 시" 또는 "결정에 대한 응답으로" 또는 "[언급된 조건 또는 이벤트] 검출 시" 또는 문맥에 따라 "[언급된 조건 또는 이벤트] 검출에 응답하여" 또는 "[언급된 조건 또는 이벤트] 검출에 응답하는"을 의미하는 것으로 해석될 수 있다.
- [0111] [00119] 본원에 정의된 바와 같이, "~에 응답하는"이라는 용어 및 위에서 설명된 바와 같은 유사 언어, 예를 들어 "만약", "~할 때" 또는 "시"는 행동 또는 이벤트에 쉽게 응답하거나 반응하는 것을 의미한다. 응답 또는 반응은 자동으로 수행된다. 따라서, 제2 행동이 제1 행동에 "응답하여" 수행된다면, 제1 행동의 발생과 제2 행동의 발생 사이에는 인과관계가 존재한다. "에 반응하는"이라는 용어는 인과 관계를 나타낸다.
- [0112] [00120] 본원에서 정의된 바와 같이, "실질적으로"라는 용어는 인용된 특성, 파라미터 또는 값이 정확히 달성될 필요는 없지만, 예를 들어 공차들, 측정 에러, 측정 정확도 제한들 및 통상의 기술자들에게 알려진 다른 요인들을 포함하는 편차들 또는 변동들이 특성이 제공하고자 하는 효과를 배제하지 않는 양도로 발생할 수 있음을 의미한다.
- [0113] [00121] 제1, 제2 등의 용어들은 본원에서 다양한 요소들을 설명하기 위해 사용될 수 있다. 이들 요소들은 이러한 용어들에 의해 제한되어서는 안 되는 데, 그 이유는 이러한 용어들이 달리 언급되지 않거나 문맥에서 달리 명시하지 않는 한 한 요소를 다른 요소와 구별하기 위해서만 사용되기 때문이다.
- [0114] [00122] 컴퓨터 프로그램 제품은 프로세서로 하여금 본원에 기술된 본 발명의 어레인지먼트들의 양상들을 수행하게 하기 위한 컴퓨터 판독가능 프로그램 명령들을 갖는 컴퓨터 판독가능 저장 매체(또는 매체)를 포함할 수 있다. 본 개시내용서 내에서, "프로그램 코드"라는 용어는 "컴퓨터 판독가능 프로그램 명령들"이라는 용어와 상호교환적으로 사용된다. 본원에 설명된 컴퓨터 판독가능 프로그램 명령들은 컴퓨터 판독가능 저장 매체로부터 각각의 컴퓨팅/프로세싱 디바이스들로 또는 네트워크, 예를 들어, 인터넷, 근거리 통신망(LAN), 광역 네트워크(WAN), 및/또는 무선 네트워크를 통해 외부 컴퓨터 또는 외부 저장 디바이스로 다운로드될 수 있다. 네트워크는 구리 송신 케이블들, 광 송신 섬유들, 무선 송신, 라우터들, 방화벽들, 스위치들, 게이트웨이 컴퓨터들 및/또는 에지 서버들을 포함하는 에지 디바이스들을 포함할 수 있다. 각 컴퓨팅/프로세싱 디바이스의 네트워크 어댑터 카드 또는 네트워크 인터페이스는 네트워크로부터 컴퓨터 판독가능 프로그램 명령들을 수신하고 각각의 컴퓨팅/프로세싱 디바이스 내의 컴퓨터 판독가능 저장 매체에 저장하기 위해 컴퓨터 판독가능 프로그램 명령들을 포워딩한다.
- [0115] [00123] 본원에 설명된 본 발명의 어레인지먼트들에 대한 조작들을 수행하기 위한 컴퓨터 판독가능 프로그램 명령들은 어셈블러 명령들, 명령-세트-아키텍처(ISA) 명령들, 기계 명령들, 기계 종속 명령들, 마이크로코드, 펌웨어 명령들, 객체 지향 프로그래밍 언어 및/또는 절차적 프로그래밍 언어들을 포함하는 하나 이상의 프로그래밍 언어들의 임의의 조합으로 작성된 소스 코드 또는 객체 코드일 수 있다. 컴퓨터 판독가능 프로그램 명령들은 상태-설정 데이터를 포함할 수 있다. 컴퓨터 판독가능 프로그램 명령들은 전체적으로 사용자의 컴퓨터에서, 부분적으로 사용자의 컴퓨터에서, 독립형 소프트웨어 패키지로, 부분적으로 사용자의 컴퓨터에서 부분적으로 원격 컴퓨터에서 또는 전체적으로 원격 컴퓨터 또는 서버에서 실행될 수 있다. 후자의 시나리오에서, 원격 컴퓨터는 LAN 또는 WAN을 포함한 모든 유형의 네트워크를 통해 사용자의 컴퓨터에 연결될 수 있거나, (예를 들어, 인터넷 서비스 제공자를 사용하여 인터넷을 통해) 외부 컴퓨터에 연결이 이루어질 수 있다. 일부 경우들에서, 예를 들어 프로그램가능 로직 회로, FPGA 또는 PLA를 포함하는 전자 회로는 본원에 설명된 본 발명의 어레인지먼트들의 양상들을 수행하기 위해, 전자 회로를 개인화하기 위한 컴퓨터 판독가능 프로그램 명령들의 상태 정보를 활용함으로써 컴퓨터 판독가능 프로그램 명령들을 실행할 수 있다.
- [0116] [00124] 본 발명의 어레인지먼트들의 소정 양상들은 본원에서 방법들, 장치(시스템들), 및 컴퓨터 프로그램 제품들의 흐름도 예시들 및/또는 블록도들을 참조하여 설명된다. 흐름도 예시들 및/또는 블록도들의 각각의 블록, 및 흐름도 예시들 및/또는 블록도들의 블록들 조합들이 컴퓨터 판독가능 프로그램 명령들, 예를 들어 프로그램 코드에 의해 구현될 수 있음이 이해될 것이다.
- [0117] [00125] 이러한 컴퓨터 판독가능 프로그램 명령들은 기계를 생성하기 위해 범용 컴퓨터, 특수 목적 컴퓨터 또는 다른 프로그램가능 데이터 프로세싱 유닛의 프로세서에 제공되어, 컴퓨터의 프로세서 또는 다른 프로그램가능 데이터 프로세싱 유닛을 통해 실행되는 명령들은 흐름도 및/또는 블록도 블록 또는 블록들에 지정된 기능/조

작들을 구현하기 위한 수단을 생성한다. 이러한 컴퓨터 관독가능 프로그램 명령들은 또한 특정 방식으로 가능하도록 컴퓨터, 프로그램가능 데이터 프로세싱 유닛, 및/또는 다른 디바이스들에 지시할 수 있는 컴퓨터 관독가능 저장 매체에 저장될 수 있어서, 명령들이 내부에 저장된 컴퓨터 관독가능 저장 매체는 흐름도 및/또는 블록도 블록 또는 블록들에 지정된 조작들의 양상들을 구현하는 명령들을 포함하는 제조 물품을 포함한다.

[0118] [00126] 컴퓨터 관독가능 프로그램 명령들은 또한 컴퓨터, 다른 프로그램가능 데이터 프로세싱 장치 또는 다른 디바이스에 로드되어 일련의 조작들이 컴퓨터 구현 프로세스를 생성하기 위해 컴퓨터, 다른 프로그램가능 장치 또는 다른 디바이스에서 수행될 수 있게 하여, 컴퓨터, 다른 프로그램가능 장치 또는 다른 디바이스에서 실행되는 명령들은 흐름도 및/또는 블록도 블록 또는 블록들에 지정된 기능들/조작들을 구현한다.

[0119] [00127] 도면들의 흐름도 및 블록도들은 본 발명의 어레인지먼트들의 다양한 양상들에 따라 시스템들, 방법들, 및 컴퓨터 프로그램 제품들의 가능한 구현들의 아키텍처, 기능 및 조작을 예시한다. 이와 관련하여, 흐름도 또는 블록도의 각 블록은 지정된 조작들을 구현하기 위한 하나 이상의 실행가능 명령들을 포함하는 모듈, 세그먼트 또는 명령들의 일부를 나타낼 수 있다.

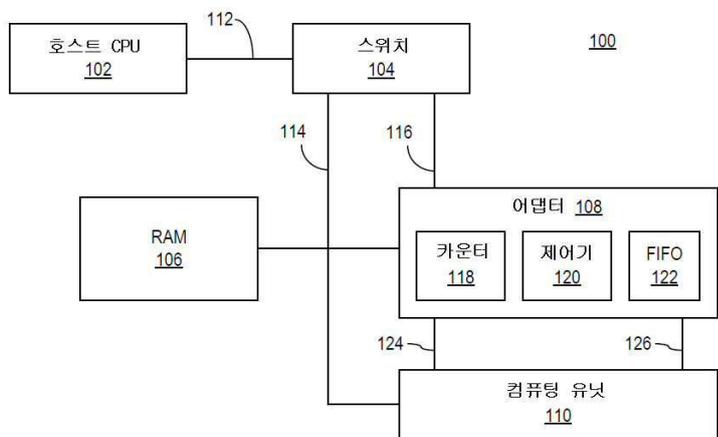
[0120] [00128] 일부 대안적인 구현들에서, 블록들에 언급된 조작들은 도면들에 언급된 순서와 다르게 발생할 수 있다. 예를 들어, 연속적으로 도시된 2 개의 블록들은 실제로 실질적으로 동시에 실행될 수 있거나, 블록들은 때때로 관련된 기능에 따라 역순으로 실행될 수 있다. 다른 예들에서, 블록들은 일반적으로 증가하는 숫자 순서로 수행될 수 있는 반면, 다른 예들에서 하나 이상의 블록들은 다양한 순서로 수행될 수 있고 결과들은 저장되고 바로 뒤따르지 않는 후속 또는 다른 블록들에서 활용된다. 또한 블록도들 및/또는 흐름도 예시의 각 블록과, 블록도들 및/또는 흐름도 예시의 블록들의 조합들이 지정된 기능 또는 조작들을 수행하거나 특수 목적 하드웨어 및 컴퓨터 명령들의 조합들을 수행하는 특수 목적 하드웨어-기반 시스템에 의해 구현될 수 있다는 것이 또한 유의될 것이다.

[0121] [00129] 아래 청구범위에서 찾을 수 있는 모든 수단 또는 단계 플러스 기능 요소들의 대응 구조들, 재료들, 조작들 및 등가물들은 특정하게 청구된 다른 청구된 요소와 조합하여 기능을 수행하기 위한 모든 구조, 재료 또는 조작을 포함하도록 의도된다.

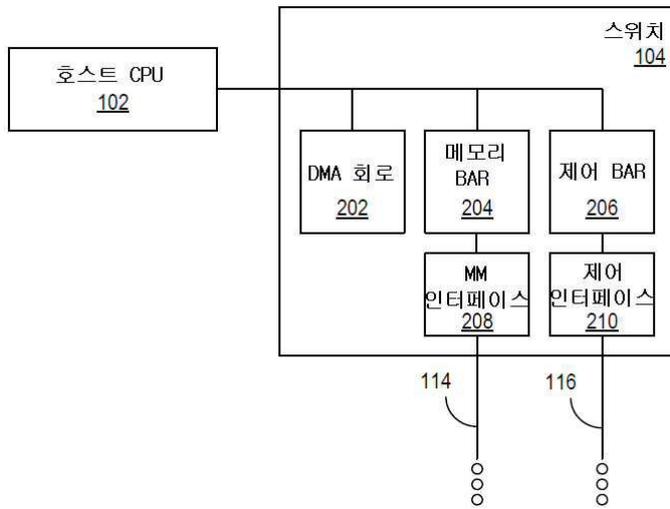
[0122] [00130] 본원에 제공된 본 발명의 어레인지먼트들의 설명은 예시를 위한 것이며 개시된 형태 및 예들로 제한되거나 배타적이지 않도록 의도된다. 본원에서 사용된 용어는 본 발명의 어레인지먼트들의 원리들, 시장에서 발견되는 기술들에 대한 실제 적용 또는 기술적 개선을 설명하고/하거나, 통상의 기술자들이 본원에 개시된 본 발명의 어레인지먼트들을 이해할 수 있도록 하기 위해 선택되었다. 수정들 및 변동들은 기술된 본 발명의 어레인지먼트들의 범위 및 사상을 벗어나지 않고 통상의 기술자들에게 명백할 수 있다. 따라서, 그러한 특징들 및 구현들의 범위를 나타내는 것으로서 전술한 개시내용보다는 다음의 청구범위들이 참조되어야 한다.

**도면**

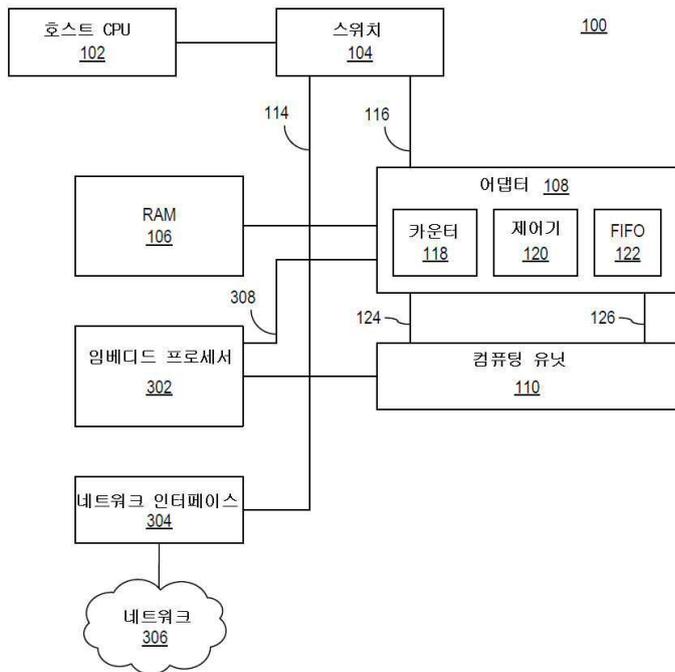
**도면1**



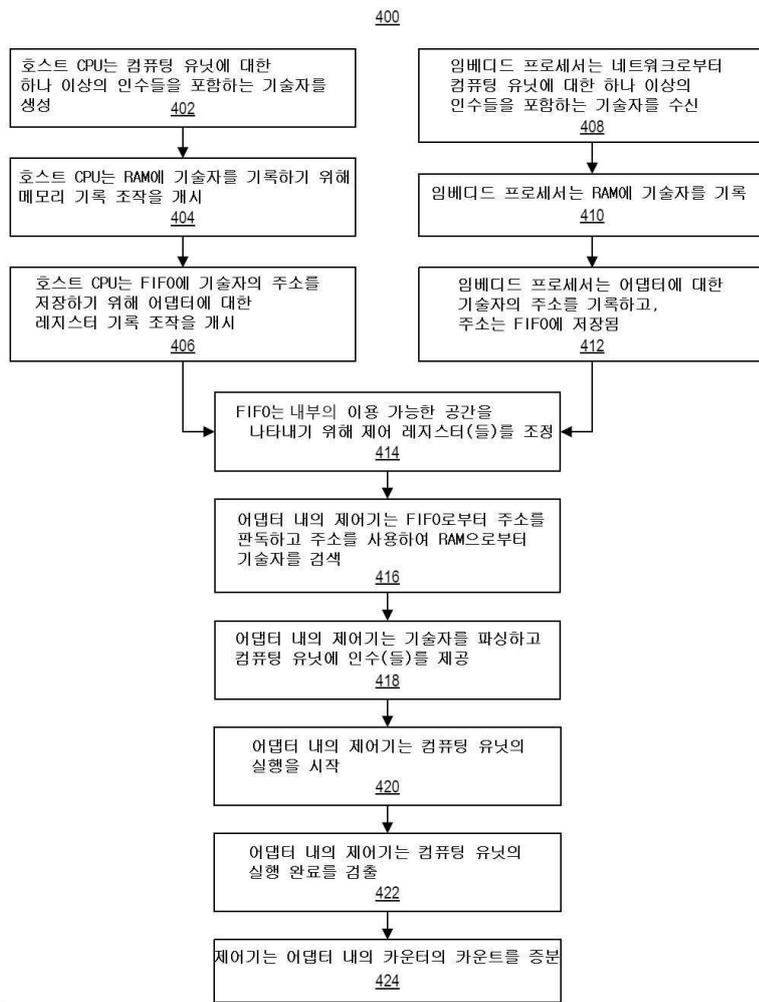
도면2



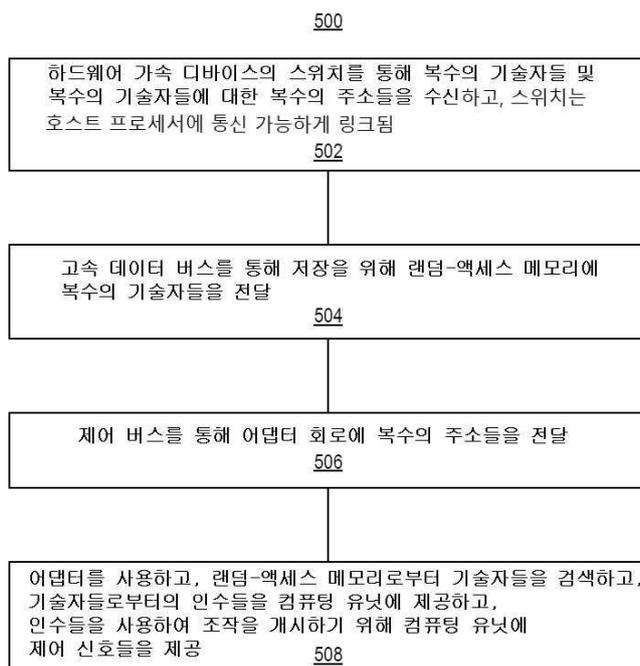
도면3



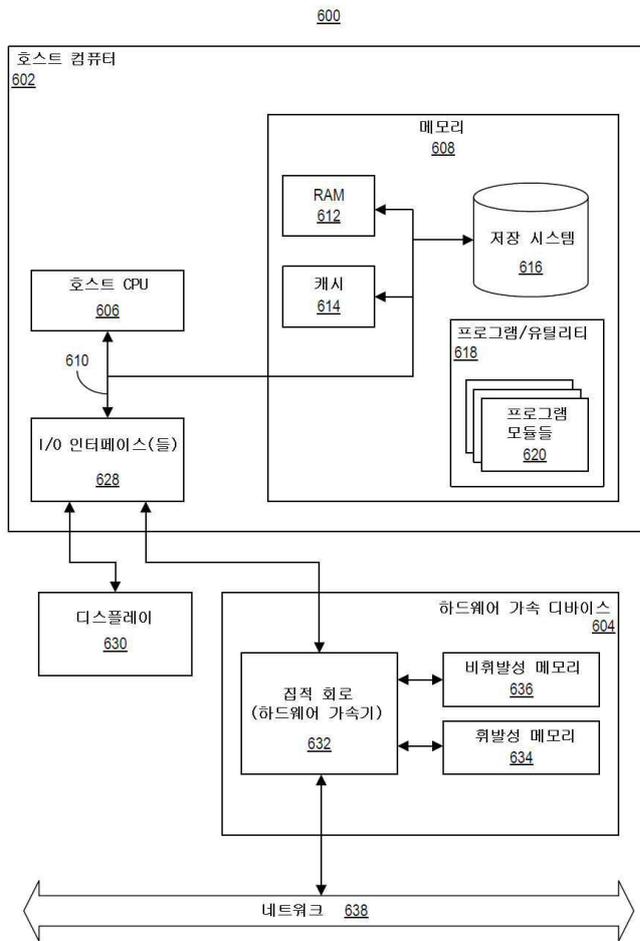
도면4



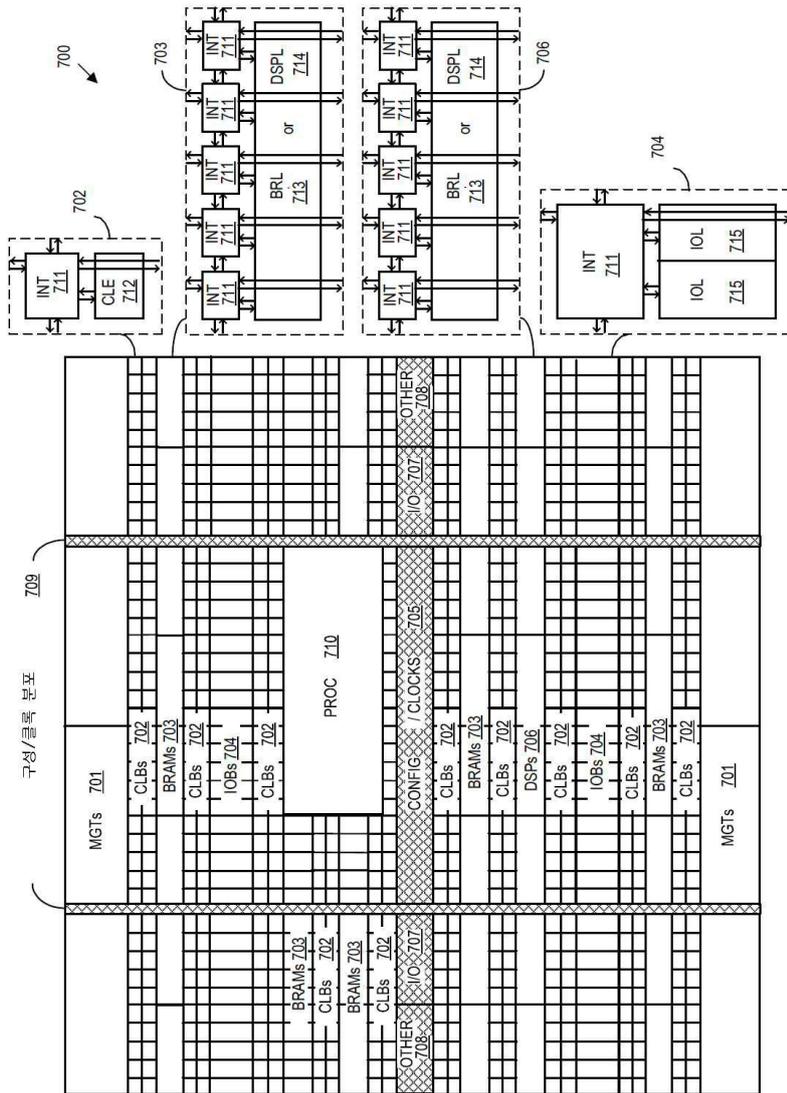
도면5



도면6



도면7



도면8

