



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0064230
(43) 공개일자 2022년05월18일

<p>(51) 국제특허분류(Int. Cl.) G06F 13/38 (2006.01) G06F 12/0815 (2016.01) G06F 13/42 (2006.01) G06F 3/06 (2006.01)</p> <p>(52) CPC특허분류 G06F 13/387 (2013.01) G06F 12/0815 (2013.01)</p> <p>(21) 출원번호 10-2020-0150510 (22) 출원일자 2020년11월11일 심사청구일자 없음</p>	<p>(71) 출원인 삼성전자주식회사 경기도 수원시 영통구 삼성로 129 (매탄동)</p> <p>(72) 발명자 이정호 경기도 과천시 별양로 12, 343동 803호 (원문동, 래미안 슈르) 전윤호 경상남도 김해시 김해대로2517번길 6-17, 201호 (삼정동, 삼성맥스빌10차) (뒀면에 계속)</p> <p>(74) 대리인 리엔목특허법인</p>
---	--

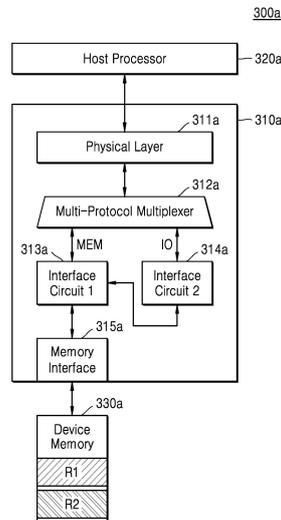
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 **다중 프로토콜에 기초하여 메모리에 액세스하기 위한 시스템, 장치 및 방법**

(57) 요약

버스를 통해 통신하도록 구성된 장치는, 제1 프로토콜에 기초하여 제1 메모리에 대한 제1 액세스를 버스를 통해 제공하도록 구성된 제1 인터페이스 회로, 및 제2 프로토콜에 기초하여 비일관적(non-coherent) 입출력 인터페이스를 버스를 통해 제공하도록 구성된 제2 인터페이스 회로를 포함할 수 있고, 제2 인터페이스 회로는, 제2 프로토콜에 기초하여 버스를 통해 수신된 제1 메시지에 응답하여 제1 메모리에 액세스함으로써 제1 메모리에 대한 제2 액세스를 버스를 통해 제공할 수 있다.

대표도 - 도3a



(52) CPC특허분류

G06F 13/4234 (2013.01)

G06F 3/0604 (2013.01)

G06F 3/0611 (2013.01)

G06F 3/0661 (2013.01)

G06F 2213/0064 (2013.01)

(72) 발명자

김대회

경상북도 구미시 선산읍 선주로 18

남희현

서울특별시 영등포구 영등포로 112, 101동 403호
(당산동2가)

명세서

청구범위

청구항 1

버스를 통해 통신하도록 구성된 장치로서,

제1 프로토콜에 기초하여 제1 메모리에 대한 제1 액세스를 상기 버스를 통해 제공하도록 구성된 제1 인터페이스 회로; 및

제2 프로토콜에 기초하여 비일관적(non-coherent) 입출력 인터페이스를 상기 버스를 통해 제공하도록 구성된 제2 인터페이스 회로를 포함하고,

상기 제2 인터페이스 회로는, 상기 제2 프로토콜에 기초하여 상기 버스를 통해 수신된 제1 메시지에 응답하여 상기 제1 메모리에 액세스함으로써 상기 제1 메모리에 대한 제2 액세스를 상기 버스를 통해 제공하도록 구성된 것을 특징으로 하는 장치.

청구항 2

청구항 1에 있어서,

상기 제1 액세스는, 고정된 제1 단위의 데이터를 포함하고,

상기 제2 액세스는, 가변적인 제2 단위의 데이터를 포함하는 것을 특징으로 하는 장치.

청구항 3

청구항 2에 있어서,

상기 제2 단위의 데이터는, 상기 제1 단위의 데이터보다 큰 것을 특징으로 하는 장치.

청구항 4

청구항 1에 있어서,

상기 제1 액세스 및 상기 제2 액세스를 모니터링하고, 상기 제1 메모리의 일관성(coherency)을 해소하도록 구성된 일관성 엔진을 더 포함하는 장치.

청구항 5

청구항 4에 있어서,

제3 프로토콜에 기초하여 상기 일관성 엔진의 메시지를 상기 버스를 통해 제공하도록 구성된 제3 인터페이스 회로를 더 포함하는 장치.

청구항 6

청구항 1에 있어서,

상기 제2 인터페이스 회로는, 상기 버스로부터 상기 제2 인터페이스 회로를 통해서 액세스되도록 구성된 제2 메모리에 저장된 데이터의 상기 제1 메모리로의 복사를 지시하는 제2 메시지를 상기 제2 프로토콜에 기초하여 상기 버스로부터 수신하도록 구성된 것을 특징으로 하는 장치.

청구항 7

청구항 6에 있어서,

상기 제2 인터페이스 회로는, 상기 제2 메시지에 응답하여 상기 제2 프로토콜에 기초하여 상기 데이터를 상기 버스로 송신하고, 상기 제2 프로토콜에 기초하여 상기 버스로부터 수신된 상기 데이터를 상기 제1 메모리에 제공하도록 구성된 것을 특징으로 하는 장치.

청구항 8

청구항 6에 있어서,

상기 제2 인터페이스 회로는, 상기 제2 메시지에 응답하여 상기 제2 메모리로부터 상기 데이터를 수신하고, 수신된 상기 데이터를 상기 제1 메모리에 제공하도록 구성된 것을 특징으로 하는 장치.

청구항 9

버스에 연결된 장치의 동작 방법으로서,

제1 프로토콜에 기초하여 제1 메모리에 대한 제1 액세스를 상기 버스를 통해 제공하는 단계; 및

제2 프로토콜에 기초하여 비일관적(non-coherent) 입출력 인터페이스를 상기 버스를 통해 제공하는 단계를 포함하고,

상기 비일관적 입출력 인터페이스를 상기 버스를 통해 제공하는 단계는, 상기 제2 프로토콜에 기초하여 상기 버스로부터 수신된 제1 메시지에 응답하여 상기 제1 메모리를 액세스함으로써 상기 제1 메모리에 대한 제2 액세스를 상기 버스를 통해 제공하는 단계를 포함하는 것을 특징으로 하는 장치의 동작 방법.

청구항 10

버스에 연결되고, 제1 메모리와 통신하도록 구성된 장치; 및

상기 제1 메모리에 기입하거나 상기 제1 메모리로부터 독출할 제1 데이터의 크기에 기초하여, 제1 프로토콜 또는 제2 프로토콜을 선택하고, 선택된 프로토콜에 기초하여 상기 버스를 통해 상기 제1 메모리에 액세스하도록 구성된 호스트 프로세서를 포함하고,

상기 제1 프로토콜은, 메모리 액세스를 위한 전용의 프로토콜이고,

상기 제2 프로토콜은, 비일관적(non-coherent) 입출력을 위한 프로토콜인 것을 특징으로 하는 시스템.

발명의 설명

기술 분야

[0001] 본 개시의 기술적 사상은 메모리 액세스에 관한 것으로서, 자세하게는 다중 프로토콜에 기초하여 버스를 통해 메모리에 액세스하기 위한 시스템, 장치 및 방법에 관한 것이다.

배경 기술

[0002] 데이터를 처리하도록 구성된 기기(apparatus)는 메모리에 액세스함으로써 다양한 동작을 수행할 수 있다. 예를 들면, 기기는 메모리로부터 독출한 데이터를 처리할 수도 있고, 처리된 데이터를 메모리에 기입할 수도 있다. 시스템에 요구되는 성능 및 기능들에 기인하여, 높은 대역폭 및 낮은 레이턴시를 제공하는 링크를 통해서 상호 통신하는 다양한 기기들이 시스템에 포함될 수 있다. 시스템에 포함된 메모리는 2이상의 기기들에 의해서 공유될 수 있고 액세스될 수 있다. 이에 따라, 시스템의 성능은, 기기들 각각의 동작 속도뿐만 아니라, 기기들 사이 통신 효율성 및 메모리 액세스에 소요되는 시간에 의존할 수 있다.

발명의 내용

해결하려는 과제

[0003] 본 개시의 기술적 사상은, 메모리 액세스의 감소된 레이턴시를 위한 시스템, 장치 및 방법을 제공한다.

과제의 해결 수단

[0004] 상기와 같은 목적을 달성하기 위하여, 본 개시의 기술적 사상의 일측면에 따라 버스를 통해 통신하도록 구성된 장치는, 제1 프로토콜에 기초하여 제1 메모리에 대한 제1 액세스를 버스를 통해 제공하도록 구성된 제1 인터페이스 회로, 및 제2 프로토콜에 기초하여 비일관적(non-coherent) 입출력 인터페이스를 버스를 통해 제공하도록 구성된 제2 인터페이스 회로를 포함할 수 있고, 제2 인터페이스 회로는, 제2 프로토콜에 기초하여 버스를 통해

수신된 제1 메시지에 응답하여 제1 메모리에 액세스함으로써 제1 메모리에 대한 제2 액세스를 버스를 통해 제공할 수 있다.

[0005] 본 개시의 기술적 사상의 일측면에 따라, 버스에 연결된 장치의 동작 방법은, 제1 프로토콜에 기초하여 제1 메모리에 대한 제1 액세스를 버스를 통해 제공하는 단계, 및 제2 프로토콜에 기초하여 비일관적 입출력 인터페이스를 버스를 통해 제공하는 단계를 포함할 수 있고, 비일관적 입출력 인터페이스를 버스를 통해 제공하는 단계는, 제2 프로토콜에 기초하여 버스로부터 수신된 제1 메시지에 응답하여 제1 메모리를 액세스함으로써 제1 메모리에 대한 제2 액세스를 버스를 통해 제공하는 단계를 포함할 수 있다.

[0006] 본 개시의 기술적 사상의 일측면에 따라, 버스에 연결되고 제1 메모리와 통신하도록 구성된 장치, 및 제1 메모리에 기입하거나 제1 메모리로부터 독출할 제1 데이터의 크기에 기초하여, 제1 프로토콜 또는 제2 프로토콜을 선택하고, 선택된 프로토콜에 기초하여 버스를 통해 제1 메모리에 액세스하도록 구성된 호스트 프로세서를 포함할 수 있고, 제1 프로토콜은, 메모리 액세스를 위한 전용의 프로토콜일 수 있고, 제2 프로토콜은, 비일관적 입출력을 위한 프로토콜일 수 있다.

발명의 효과

[0007] 본 개시의 예시적 실시예에 따른 시스템, 장치 및 방법에 의하면, 데이터의 크기에 따라 최적의 레이턴시를 제공하는 프로토콜이 선택될 수 있다.

[0008] 또한, 본 개시의 예시적 실시예에 따른 시스템, 장치 및 방법에 의하면, 최적의 프로토콜에 기초하여 메모리가 액세스됨으로써 메모 액세스의 감소된 레이턴시가 달성될 수 있고, 메모리를 포함하는 시스템의 성능이 향상될 수 있다.

[0009] 본 개시의 예시적 실시예들에서 얻을 수 있는 효과는 이상에서 언급한 효과들로 제한되지 아니하며, 언급되지 아니한 다른 효과들은 이하의 기재로부터 본 개시의 예시적 실시예들이 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 도출되고 이해될 수 있다. 즉, 본 개시의 예시적 실시예들을 실시함에 따른 의도하지 아니한 효과들 역시 본 개시의 예시적 실시예들로부터 당해 기술분야의 통상의 지식을 가진 자에 의해 도출될 수 있다.

도면의 간단한 설명

- [0010] 도 1은 본 개시의 예시적 실시예에 따른 시스템을 나타내는 블록도이다.
- 도 2a 및 도 2b는 본 개시의 예시적 실시예들에 따른 시스템의 예시들을 나타내는 블록도들이다.
- 도 3a 및 도 3b는 본 개시의 예시적 실시예들에 따른 시스템의 예시들을 나타내는 블록도들이다.
- 도 4는 본 개시의 예시적 실시예에 따라 메모리를 액세스하는 동작의 예시를 나타내는 메시지도이다.
- 도 5는 본 개시의 예시적 실시예에 따른 어드레스 맵을 나타내는 도면이다.
- 도 6은 본 개시의 예시적 실시예에 따른 시스템을 나타내는 블록도이다.
- 도 7a 및 도 7b는 본 개시의 예시적 실시예들에 따른 시스템의 예시들을 나타내는 블록도들이다.
- 도 8은 본 개시의 예시적 실시예에 따른 시스템을 나타내는 블록도이다.
- 도 9는 본 개시의 예시적 실시예에 따라 다중 프로토콜에 기초하여 메모리에 액세스하기 위한 방법을 나타내는 순서도이다.
- 도 10a 및 도 10b는 본 개시의 예시적 실시예들에 따라 다중 프로토콜에 기초하여 메모리에 액세스하기 위한 방법의 예시들을 나타내는 순서도들이다.
- 도 11a 및 도 11b는 본 개시의 예시적 실시예들에 따른 시스템의 예시들을 나타내는 도면이다.
- 도 12는 본 개시의 예시적 실시예에 따른 시스템을 포함하는 데이터 센터를 나타내는 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0011] 도 1은 본 개시의 예시적 실시예에 따른 시스템(100)을 나타내는 블록도이다. 시스템(100)은 상호 통신하는 장치(110) 및 호스트 프로세서(120)를 포함하는 임의의 컴퓨팅 시스템(또는 컴퓨팅 시스템에 포함된 구성요소)일

수 있다. 예를 들면, 시스템(100)은, 데스크탑 컴퓨터, 서버, 키오스크(kiosk) 등과 같은 고정형(stationary) 컴퓨팅 시스템에 포함될 수도 있고, 랩탑 컴퓨터, 모바일 폰, 웨어러블 장치 등과 같은 휴대형(portable) 컴퓨팅 시스템에 포함될 수도 있다. 또한, 일부 실시예들에서, 시스템(100)은, 장치(110) 및 호스트 프로세서(120)가 하나의 칩 또는 패키지에 구현된, 시스템-온-칩(system-on-chip; SoC) 또는 시스템-인-패키지(system-in-package; SiP)에 포함될 수도 있다. 도 1에 도시된 바와 같이, 시스템(100)은, 장치(110), 호스트 프로세서(120), 장치 메모리(130) 및 호스트 메모리(140)를 포함할 수 있다.

[0012] 도 1을 참조하면, 장치(110) 및 호스트 프로세서(120)는 링크(150)를 통해서 상호 통신할 수 있고, 링크(150) 상으로 메시지 및/또는 데이터를 상호 송신하거나 수신할 수 있다. 본 개시의 예시적 실시예들은, CXL(compute express link) 프로토콜들을 지원하는 CXL 사양에 기초한 링크(150)를 참조하여 설명될 것이나, 장치(110) 및 호스트 프로세서(120)는, 비제한적인 예시로서 XBus 프로토콜, NVLink 프로토콜, Infinity Fabric 프로토콜, CCIX(cache coherent interconnect for accelerators) 프로토콜, CAPI(coherent accelerator processor interface) 등과 같은 일관적 상호연결(coherent interconnect) 기술들에 기초하여 상호 통신할 수도 있다.

[0013] 일부 실시예들에서, 링크(150)는 다수의(multiple) 프로토콜들을 지원할 수 있고, 다수의 프로토콜들을 통해서 메시지 및/또는 데이터가 전달될 수 있다. 예를 들면, 링크(150)는, 비일관적(non-coherent) 프로토콜(예컨대, CXL.io), 일관적 프로토콜(예컨대, CXL.cache) 및 메모리 액세스 프로토콜(또는 메모리 프로토콜)(예컨대, CXL.mem)을 포함하는 CXL 프로토콜들을 지원할 수 있다. 일부 실시예들에서, 링크(150)는, 비제한적인 예시로서 PCI(peripheral component interconnect), PCIe(PCI express), USB(universal serial bus), SATA(serial advanced technology attachment) 등과 같은 프로토콜을 지원할 수도 있다. 본 명세서에서, 링크(150)에 의해서 지원되는 프로토콜은 상호연결 프로토콜로서 지칭될 수도 있다.

[0014] 장치(110)는 호스트 프로세서(120)에 유용한 기능을 제공하는 임의의 장치를 지칭할 수 있고, 일부 실시예들에서 CXL 사양의 가속기(accelerator)에 대응할 수 있다. 예를 들면, 호스트 프로세서(120) 상에서 실행되는 소프트웨어는 연산(computing) 및/또는 입출력(I/O) 작업의 적어도 일부를 장치(110)에 일임(offload)할 수 있다. 일부 실시예들에서, 장치(110)는, GPU(graphic processing unit), NPU(neural processing unit) 등과 같은 프로그래밍가능한(programmable) 구성요소(component), IP(intellectual property) 코어 등과 같은 고정된 기능을 제공하는 구성요소 및 FPGA(field programmable gate array) 등과 같은 재구성가능한(reconfigurable) 구성요소 중 적어도 하나를 포함할 수 있다. 도 1에 도시된 바와 같이, 장치(110)는 물리 계층(111), 다중-프로토콜 멀티플렉서(112), 인터페이스 회로들(113) 및 가속기 회로(114)를 포함할 수 있고, 장치 메모리(130)와 통신할 수 있다.

[0015] 가속기 회로(114)는, 장치(110)가 호스트 프로세서(120)에 제공하는 유용한 기능을 수행할 수 있고, 가속기 로직으로 지칭될 수도 있다. 도 1에 도시된 바와 같이, 가속기 회로(114)는 다수의 프로토콜들을 사용하여 인터페이스 회로들(113)을 통해서 호스트 프로세서(120)와 통신할 수 있다. 일부 실시예들에서, 가속기 회로(114)는 장치 메모리(130)의 일관성(coherency)을 해소하기 위한 구성요소(예컨대, 도 6의 617)를 포함할 수도 있다. 또한, 일부 실시예들에서, 도 2a를 참조하여 후술되는 바와 같이, 장치(110)에서 가속기 회로(114)는 생략될 수도 있다.

[0016] 인터페이스 회로들(113)은, 다수의 프로토콜들을 지원할 수 있다. 예를 들면, 인터페이스 회로들(113)은 비일관적 프로토콜, 일관적 프로토콜 및 메모리 액세스 프로토콜 중 적어도 2개의 프로토콜들을 위한 2개 이상의 회로들을 포함할 수 있다. 2개 이상의 회로들 각각은 대응하는 프로토콜에 기초하여 호스트 프로세서(120)로부터 수신된 메시지를 가속기 회로(114)에 제공하거나, 가속기 회로(114)로부터 수신된 메시지를 호스트 프로세서(120)에 제공할 수 있다. 도 3a 및 도 3b 등을 참조하여 후술되는 바와 같이, 인터페이스 회로들(113)은, 2이상의 프로토콜들에 기초한, 장치 메모리(130)에 대한 액세스를 호스트 프로세서(120)에 제공할 수 있고, 호스트 프로세서(120)는 필요에 따라 장치 메모리(130)에 액세스하기 위한 최적의 프로토콜을 선택할 수 있다. 이에 따라, 장치 메모리(130)에 대한 액세스의 감소된 레이턴시를 제공하는 최적의 프로토콜에 기인하여, 호스트 프로세서(120)의 성능 및 시스템(100)의 성능이 향상될 수 있다.

[0017] 다중-프로토콜 멀티플렉서(112)는, 가속기 회로(114) 및 호스트 프로세서(120) 사이 통신을 위한 메시지 및/또는 데이터에 기초하여 다수의 프로토콜들 중 하나의 프로토콜을 판정할 수 있다. 다중-프로토콜 멀티플렉서(112)는, 인터페이스 회로들(113)이 연결된 적어도 하나의 프로토콜 큐를 포함할 수 있고, 인터페이스 회로들(113)은 적어도 하나의 프로토콜 큐를 통해서 호스트 프로세서(120)와 메시지 및/또는 데이터를 주고 받을 수 있다. 일부 실시예들에서, 인터페이스 회로들(113) 및 다중-프로토콜 멀티플렉서(112)는 하나의 구성요소로 통

합될 수도 있다. 일부 실시예들에서, 다중-프로토콜 멀티플렉서(112)는 링크(150)에 의해서 지원되는 다수의 프로토콜들 각각에 대응하는 다수의 프로토콜 큐들을 포함할 수 있다. 또한, 일부 실시예들에서, 다중-프로토콜 멀티플렉서(112)는 상이한 프로토콜들에 의한 통신들 사이를 중재(arbitration)할 수 있고, 선택된 통신들을 물리 계층(111)에 제공할 수 있다. 일부 실시예들에서, 물리 계층(111)은, 호스트 프로세서(120)의 물리 계층(121)과 단일 상호연결, 버스, 트레이스(trace) 등을 통해서 연결될 수 있다.

[0018] 장치 메모리(130)는, 도 1에 도시된 바와 같이, 장치(110)에 연결될 수 있고, 장치-부착 메모리(device-attached memory)로서 지칭될 수 있다. 도 1에 도시된 바와 같이 장치 메모리(130)가 시스템(100)에 포함되는 경우, 가속기 회로(114)는 장치 메모리(130)와 통신할 수 있고, 링크(150)에 독립적인 프로토콜, 즉 장치-고유의(device-specific) 프로토콜에 기초하여 장치 메모리(130)와 통신할 수 있다. 일부 실시예들에서, 도 3a 및 도 3b 등을 참조하여 후술되는 바와 같이, 장치(110)는 장치 메모리(130)에 액세스하기 위한 구성요소로서 메모리 인터페이스(예컨대, 도 3a의 315a)를 포함할 수 있고, 가속기 회로(114) 및/또는 인터페이스 회로들(113)은 메모리 인터페이스를 통해서 장치 메모리(130)에 액세스할 수 있다. 메모리 인터페이스는, 장치(110), 즉 가속기 회로(114)의 장치 메모리(130)에 대한 액세스뿐만 아니라 링크(150) 및 인터페이스 회로들(113)을 통한 호스트 프로세서(120)의 장치 메모리(130)에 대한 액세스를 제공할 수 있다. 일부 실시예들에서, 장치 메모리(130)는 CXL 사양의 장치 부착 메모리에 대응할 수 있다. 본 명세서에서, 장치 메모리(130)는 장치(110)에 포함되는 것으로 지칭될 수 있고, 메모리로서 단순하게 지칭될 수 있다.

[0019] 호스트 프로세서(120)는 시스템(100)의 메인 프로세서, 예컨대 CPU(central processing unit)일 수 있고, 일부 실시예들에서 CXL 사양의 호스트 프로세서(또는 호스트)에 대응할 수 있다. 도 1에 도시된 바와 같이, 호스트 프로세서(120)는 호스트 메모리(140)에 연결될 수 있고, 물리 계층(121), 다중-프로토콜 멀티플렉서(122), 인터페이스 회로들(123), 일관성/캐시 회로(124), 버스 회로(125), 적어도 하나의 코어(126) 및 입출력 장치(127)를 포함할 수 있다.

[0020] 적어도 하나의 코어(126)는 명령어(instruction)를 실행할 수 있고, 일관성/캐시 회로(124)와 연결될 수 있다. 일관성/캐시 회로(124)는 캐시 계층(cache hierarchy)을 포함할 수 있고, 일관성/캐시 로직으로 지칭될 수도 있다. 도 1에 도시된 바와 같이, 일관성/캐시 회로(124)는 적어도 하나의 코어(126) 및 인터페이스 회로들(123)과 통신할 수 있다. 예를 들면, 일관성/캐시 회로(124)는 일관적 프로토콜 및 메모리 액세스 프로토콜을 포함하는 2이상의 프로토콜들을 통한 통신을 가능하게 할 수 있다. 일부 실시예들에서, 일관성/캐시 회로(124)는 DMA(direct memory access) 회로를 포함할 수도 있다. 입출력 장치(127)는 버스 회로(125)와 통신하기 위하여 사용될 수 있다. 예를 들면, 버스 회로(125)는 PCIe 로직일 수 있고, 입출력 장치(127)는 PCIe 입출력 장치일 수 있다.

[0021] 인터페이스 회로들(123)은, 호스트 프로세서(120)의 구성요소들, 예컨대 일관성/캐시 회로(124) 및 버스 회로(125)와 장치(110) 사이 통신을 가능하게 할 수 있다. 일부 실시예들에서, 인터페이스 회로들(123)은 다수의 프로토콜들, 예컨대 비일관적 프로토콜, 일관적 프로토콜 및 메모리 프로토콜에 따라 호스트 프로세서(120)의 구성요소들 및 장치(110) 사이 메시지 및/또는 데이터의 통신을 가능하게 할 수 있다.

[0022] 다중-프로토콜 멀티플렉서(122)는 적어도 하나의 프로토콜 큐를 포함할 수 있다. 인터페이스 회로들(123)은 다중-프로토콜 멀티플렉서(122)에 포함된 적어도 하나의 프로토콜 큐에 연결될 수 있고, 적어도 하나의 프로토콜 큐를 통해서 장치(110)와 메시지 및/데이터를 주고 받을 수 있다. 일부 실시예들에서, 다중-프로토콜 멀티플렉서(122)는, 호스트 프로세서(120)의 구성요소들 및 장치(110) 사이 통신을 위한 메시지 및/또는 데이터에 기초하여 다수의 프로토콜들 중 하나의 프로토콜을 판정할 수 있다. 일부 실시예들에서, 인터페이스 회로들(123) 및 다중-프로토콜 멀티플렉서(122)는 하나의 구성요소로 통합될 수도 있다. 일부 실시예들에서, 다중-프로토콜 멀티플렉서(122)는 링크(150)에 의해서 지원되는 다수의 프로토콜들 각각에 대응하는 다수의 프로토콜 큐들을 포함할 수 있다. 또한, 일부 실시예들에서, 다중-프로토콜 멀티플렉서(122)는 상이한 프로토콜들에 의한 통신들 사이를 중재할 수 있고, 선택된 통신들을 물리 계층(121)에 제공할 수 있다.

[0023] 도 2a 및 도 2b는 본 개시의 예시적 실시예들에 따른 시스템의 예시들을 나타내는 블록도들이다. 구체적으로, 도 2a의 블록도는 CXL 사양에서 정의된 유형 3 CXL 장치(type 3 CXL device)를 포함하는 시스템(200a)을 나타내고, 도 2b의 블록도는 CXL 사양에서 정의된 유형 2 CXL 장치(type 2 CXL device)를 포함하는 시스템(200b)을 나타낸다. 이하에서, 도 2a 및 도 2b에 대한 설명 중 상호 중복되는 내용은 생략될 것이다.

[0024] 도 2a를 참조하면, 시스템(200a)은 호스트 프로세서(220a), 장치(210a), 장치 메모리(230a) 및 호스트 메모리(240a)를 포함할 수 있다. 일부 실시예들에서, 호스트 프로세서(220a)는 루트 복합기(root complex)로서 지칭될

수 있고, 장치(210a)는 메모리 확장기(memory expander)로서 지칭될 수 있으며, 장치(210a) 및 장치 메모리(230a)는 유형 3 CXL 장치로서 총괄적으로(collectively) 지칭될 수 있다.

[0025] 호스트 프로세서(200a)는 메모리 컨트롤러(221a), 홈 에이전트(222a) 및 입출력 브릿지(223a)를 포함할 수 있다. 메모리 컨트롤러(221a)는 호스트 메모리(240a)의 인터페이스에 기초하여 호스트 메모리(240a)에 대한 액세스를 제공할 수 있다. 홈 에이전트(222a)는 일관성 엔진으로 지칭될 수 있고, 메모리 프로토콜(MEM)에 기초하여 장치(210a)와 통신할 수 있고, 정해진(given) 어드레스를 위한 일관성을 해소할 수 있다. 메모리 프로토콜(MEM)은 장치 메모리(230a)에 대한 액세스를 위한 전용의 프로토콜일 수 있고, 본 명세서에서 제1 프로토콜로서 지칭될 수 있다. 또한, 입출력 브릿지(223a)는 비일관적 프로토콜(IO)에 기초하여 장치(210a)와 통신할 수 있고, 일부 실시예들에서, IOMMU(input/output memory management unit)을 포함할 수 있다. 호스트 프로세서(220a)는 입출력 브릿지(223a)를 통해서, 장치 발견(device discovery), 열거(enumeration), 오류 보고 및 관리(error reporting and management) 등을 위한 메시지를 장치(210a)와 교환할 수 있다. 본 명세서에서, 비일관적 프로토콜(IO)은 제2 프로토콜로서 지칭될 수 있다.

[0026] 장치(210a)는 메모리 인터페이스(211a)를 포함할 수 있다. 메모리 인터페이스(211a)는 메모리 인터페이스 회로, 메모리 컨트롤러로 지칭될 수 있고, 장치 메모리(230a)의 인터페이스에 기초하여 장치 메모리(230a)에 대한 액세스를 제공할 수 있다. 일부 실시예들에서, 장치(210a)는, 도 1의 장치(110)와 상이하게, 능동적인 컴퓨팅 엔진을 포함하지 아니할 수 있고, 메모리 확장기로서 기능할 수 있다. 이에 따라, 장치(210a)는 일관적 프로토콜(CACHE)에 기초한 호스트 프로세서(220a)에 대한 요청을 발행하지 아니할 수 있고, 주로 메모리 프로토콜(MEM)에 기초하여 호스트 프로세서(220a)의 요청을 처리할 수 있다.

[0027] 도 2b를 참조하면, 시스템(200b)은 호스트 프로세서(220b), 장치(210b), 장치 메모리(230b) 및 호스트 메모리(240b)를 포함할 수 있다. 일부 실시예들에서, 장치(210b)는 메모리를 갖는 장치(device with memory)로서 지칭될 수 있고, 장치(210b) 및 장치 메모리(230b)는 유형 2 CXL 장치로서 총괄적으로 지칭될 수 있다.

[0028] 호스트 프로세서(200b)는 메모리 컨트롤러(221b), 홈 에이전트(222b), 입출력 브릿지(223b) 및 일관성 브릿지(224b)를 포함할 수 있다. 일관성 브릿지(224b)는 장치(210b) 및 호스트 프로세서(220b) 사이 인터랙션들(interactions)을 정의하는 일관적 프로토콜(CACHE)에 기초하여, 장치(210b)와 통신할 수 있다. 예를 들면, 일관성 브릿지(224b)는, 요청(예컨대, D2H request), 응답(예컨대, D2H response) 및 데이터(예컨대, D2H data)를 장치(210b)로부터 수신할 수 있고, 요청(H2D request), 응답(예컨대, H2D response) 및 데이터(예컨대, H2D data)를 장치(210b)에 제공할 수 있다.

[0029] 장치(210b)는 메모리 인터페이스(211b), 캐시(212b) 및 일관성 엔진(213b)을 포함할 수 있다. 장치(210b)는, 도 1의 장치(110)와 유사하게, 캐시(212b)는 가속기 회로에 의해서 사용될 수 있다. 장치(210b)는 캐시(212b)의 일관성 트랜잭션들을 위하여 일관적 프로토콜(CACHE)을 사용할 수 있다. 일관성 엔진(213b)은 DCOH(device coherency engine)으로 지칭될 수 있고, 캐시(212b)의 일관성을 해소할 수 있고, 바이어스 상태들(예컨대, 호스트 바이어스 모드, 장치 바이어스 모드)을 관리할 수 있다.

[0030] 도 3a 및 도 3b는 본 개시의 예시적 실시예들에 따른 시스템의 예시들을 나타내는 블록도들이다. 구체적으로, 도 3a 및 도 3b의 블록도들은 메모리 프로토콜(MEM) 및 비일관적 프로토콜(IO)에 기초하여 장치 메모리들(330a, 330b)에 대한 액세스를 제공하는 장치들(310a, 310b)을 포함하는 시스템들(300a, 300b)을 나타낸다. 일부 실시예들에서, 도 3a 및 도 3b의 시스템들(300a, 300b) 각각은 CXL 사양에서 정의된 유형 3 CLX 장치를 포함하는 시스템에 대응할 수 있다. 이하에서, 도 3a 및 도 3b에 대한 설명 중 상호 중복되는 내용은 생략될 것이다.

[0031] 도 3a를 참조하면, 시스템(300a)은 호스트 프로세서(320a), 장치(310a) 및 장치 메모리(330a)를 포함할 수 있다. 일부 실시예들에서, 시스템(300a)은 호스트 메모리를 더 포함할 수 있고, 호스트 프로세서(320a)는 호스트 메모리에 액세스하기 위한 메모리 컨트롤러를 포함할 수 있다. 장치(310a)는, 물리 계층(311a), 다중-프로토콜 멀티플렉서(312a), 제1 인터페이스 회로(313a), 제2 인터페이스 회로(314a) 및 메모리 인터페이스(315a)를 포함할 수 있다.

[0032] 제1 인터페이스 회로(313a)는 메모리 프로토콜(MEM)에 기초하여 호스트 프로세서(320a)와 통신할 수 있고, 메모리 프로토콜(MEM)에 기초하여 호스트 프로세서(320a)로부터 수신된 메시지에 기초하여 메모리 인터페이스(315a)를 통해 장치 메모리(330a)에 액세스할 수 있다. 예를 들면, 제1 인터페이스 회로(313a)는 메모리 프로토콜(MEM)에 기초하여 수신된 메시지에 응답하여, 메모리 인터페이스(315a)에 데이터의 독출 및/또는 데이터의 기입을 요청할 수 있다. 메모리 인터페이스(315a)는 제1 인터페이스 회로(313a)의 요청에 응답하여, 장치 메모리

(330a)로부터 데이터를 독출하여 제1 인터페이스 회로(313a)에 제공하거나, 제1 인터페이스 회로(313a)의 요청에 포함된 데이터를 장치 메모리(330a)에 기입할 수 있다. 본 명세서에서, 메모리 프로토콜(MEM)에 기초하여 제1 인터페이스 회로(313a)를 통한 장치 메모리(330a)에 대한 액세스는 제1 액세스로서 지칭될 수 있다.

[0033] 제2 인터페이스 회로(314a)는 비일관적 프로토콜(IO)에 기초하여 호스트 프로세서(320a)와 통신할 수 있다. 제2 인터페이스 회로(314a)는 비일관적 프로토콜(IO)에 기초하여, 도 2a를 참조하여 전송된 메시지뿐만 아니라 장치 메모리(330a)에 대한 액세스를 위한 메시지를 수신할 수 있고, 장치 메모리(330a)에 대한 액세스를 호스트 프로세서(320a)에 제공할 수 있다. 예를 들면, 도 3a에 도시된 바와 같이, 제2 인터페이스 회로(314a)는 제1 인터페이스 회로(313a)와 통신할 수 있고, 비일관적 프로토콜(IO) 및 메모리 프로토콜(MEM) 사이에서 메시지 및/또는 데이터를 변환할 수 있다. 본 명세서에서, 비일관적 프로토콜(IO)에 기초하여 제2 인터페이스 회로(314a)를 통한 장치 메모리(330a)에 대한 액세스는 제2 액세스로서 지칭될 수 있다. 이에 따라, 장치(310a)는 장치 메모리(330a)에 대하여, 메모리 프로토콜(MEM)에 기초한 제1 액세스 및 비일관적 프로토콜(IO)에 기초한 제2 액세스를 호스트 프로세서에 제공할 수 있다.

[0034] 메모리 인터페이스(315a)는 제1 인터페이스 회로(313a) 및 제2 인터페이스 회로(314a)와 통신할 수 있고, 장치 메모리(330a)의 인터페이스에 기초하여 장치 메모리(330a)와 통신할 수 있다. 장치 메모리(330a)는 데이터를 저장하기 위한 임의의 매체를 지칭할 수 있다. 일부 실시예들에서, 장치 메모리(330a)는 반도체 메모리 장치를 포함할 수 있고, 예를 들면 플래시 메모리, RRAM(resistive random access memory) 등과 같은 비휘발성(non-volatile) 메모리 장치 및/또는 DRAM(dynamic random access memory), SRAM(static random access memory) 등과 같은 휘발성 메모리 장치를 포함할 수 있다.

[0035] 장치 메모리(330a)는 복수의 영역들로 분할될 수 있고, 분할된 영역들 중 적어도 하나의 영역은, 메모리 인터페이스(315a)를 통해 메모리 프로토콜(MEM)에 기초한 액세스(즉, 제1 액세스) 및 비일관적 프로토콜(IO)에 기초한 액세스(즉, 제2 액세스)의 대상이 될 수 있다. 예를 들면, 도 3a에 도시된 바와 같이, 장치 메모리(330a)는 제1 영역(R1) 및 제2 영역(R2)을 포함할 수 있고, 제1 영역(R1)은 제1 액세스의 대상이 될 수 있는 한편, 제2 영역(R2)은 제1 액세스 및 제2 액세스 양자의 대상이 될 수 있다. 이에 따라, 제2 영역(R2)은, 비일관적 프로토콜(IO)에 의해서 지원되는 CMB(controller memory buffer), PMR(persistent memory region), 도어벨(doorbell) 등으로 사용될 수 있고, 도 5를 참조하여 후술되는 바와 같이, 호스트 프로세서는 BAR(base address register)를 통해서 제2 영역(R2)에 액세스할 수 있다. 또한, 일부 실시예들에서, 제1 영역(R1) 및 제2 영역(R2) 모두 제2 액세스의 대상이 될 수 있고, 이 경우 제1 영역(R1) 및 제2 영역(R2)은 상이한 BAR들에 각각 대응할 수 있다. 상이한 프로토콜들, 즉 메모리 프로토콜(MEM) 및 비일관적 프로토콜(IO)에 기초하여 호스트 프로세서(320a)가 장치 메모리(330a)에 액세스하는 동작의 예시가 도 4을 참조하여 후술될 것이다.

[0036] 도 3b를 참조하면, 시스템(300b)은 호스트 프로세서(320b), 장치(310b) 및 장치 메모리(330b)를 포함할 수 있다. 장치(310b)는, 도 3a의 장치(310a)와 유사하게, 물리 계층(311b), 다중-프로토콜 멀티플렉서(312b), 제1 인터페이스 회로(313b), 제2 인터페이스 회로(314b) 및 메모리 인터페이스(315b)를 포함할 수 있다. 도 3a의 예시와 비교할 때, 도 3b의 제2 인터페이스 회로(314b)는 메모리 인터페이스(315b)와 통신할 수 있다. 예를 들면, 제2 인터페이스 회로(314b)는 비일관적 프로토콜(IO)에 기초하여 호스트 프로세서(320b)와 통신할 수 있고, 메모리 인터페이스(315b)를 통해서 장치 메모리(330b)에 액세스할 수 있다. 이에 따라, 제2 액세스는 제2 인터페이스 회로(314b) 및 메모리 인터페이스(315b)를 통해서 가능할 수 있다.

[0037] 이하에서, 도 3a를 참조하여 전송된 바와 같이, 제2 액세스가 제2 인터페이스 회로(314a), 제1 인터페이스 회로(313a) 및 메모리 인터페이스(315a)를 통해서 수행되는 예시들이 주로 설명될 것이나, 본 개시의 예시적 실시예들이 이에 제한되지 아니하는 점이 유의된다.

[0038] 도 4은 본 개시의 예시적 실시예에 따라 메모리를 액세스하는 동작의 예시를 나타내는 메시지도이다. 도면들을 참조하여 전송된 바와 같이, 호스트 프로세서(420)는 다수의 프로토콜들에 기초하여 장치(410)에 포함된 메모리를 액세스할 수 있다. 도 4에서, 제1 메시지(MSG1)는 제1 프로토콜, 즉 도 3의 메모리 프로토콜(MEM)에 기초할 수 있고, 제2 메시지(MSG2)는 제2 프로토콜, 즉 도 3의 비일관적 프로토콜(IO)에 기초할 수 있다.

[0039] 도 4을 참조하면, 제1 메시지(MSG1)는 제1 헤더(H1), 제1 데이터(D1) 및 제1 트레일러(trailer)(T1)를 포함할 수 있다. 일부 실시예들에서, 메모리 프로토콜(MEM)은 고정된 단위의 제1 데이터(D1)를 포함하는 제1 메시지(MSG1)를 지원할 수 있다. 예를 들면, CXL.mem은 64B(byte)의 데이터를 포함하는 메시지를 지원할 수 있다. 이에 따라, 호스트 프로세서(420)는 메모리 프로토콜(MEM)에 기초하여 메모리에 데이터를 저장하기 위하여, 고정된 단위의 제1 데이터(D1)를 포함하는 제1 메시지(MSG1)를 장치(410)에 제공할 수 있다. 또한, 장치(410)는 메

모리 프로토콜(MEM)에 기초하여 메모리에 저장된 데이터를 제공하기 위하여, 고정된 단위의 제1 데이터(D1)를 포함하는 제1 메시지(MSG1)를 호스트 프로세서(420)에 제공할 수 있다. 고정된 단위의 제1 데이터(D1)에 기인하여, 제1 헤더(H1) 및/또는 제1 트레일러(T1)는 후술되는 제2 헤더(H2) 및/또는 제2 트레일러(T2)보다 작은 크기를 가질 수 있고 단순할 수 있다.

[0040] 제2 메시지(MSG2)는 제2 헤더(H2), 제2 데이터(D2) 및 제2 트레일러(T2)를 포함할 수 있다. 일부 실시예들에서, 비일관적 프로토콜(IO)은 가변적인 단위의 제2 데이터(D2)를 포함하는 제2 메시지(MSG2)를 지원할 수 있다. 예를 들면, CXL.io는 PCIe에 기초할 수 있고, PCIe 버스를 통해 액세스되는 메모리에 관한 NVMe(non-volatile memory express)는 최대 4KB(kilobyte)의 데이터를 포함하는 메시지를 지원할 수 있다. 도 3a 및 도 3b를 참조하여 전술된 바와 같이, 장치(410)에 포함된 메모리는 메모리 프로토콜(MEM)에 기초하여 액세스될 수 있을 뿐만 아니라, 비일관적 프로토콜(IO)에 기초하여 액세스될 수 있고, 이에 따라 호스트 프로세서(420)는 가변적인 단위의 제2 데이터(D2)를 포함하는 제2 메시지(MSG2)를 장치(410)에 송신하거나 장치로부터 수신할 수 있다. 예를 들면, 호스트 프로세서(420)는 비일관적 프로토콜(IO)에 기초하여 메모리에 데이터를 저장하기 위하여, 가변적인 단위의 제2 데이터(D2)를 포함하는 제2 메시지(MSG2)를 장치(410)에 제공할 수 있다. 또한, 장치(410)는 비일관적 프로토콜(IO)에 기초하여 메모리에 저장된 데이터를 제공하기 위하여, 가변적인 단위의 제2 데이터(D2)를 포함하는 제2 메시지(MSG2)를 호스트 프로세서(420)에 제공할 수 있다. 메모리 프로토콜(MEM)에서 지원되는 단위(예컨대, 64B)보다 큰 가변적인 크기를 가지는 제2 데이터(D2)에 기인하여, 제2 헤더(H2) 및/또는 제2 트레일러(T2)는 전술된 제1 헤더(H1) 및/또는 제1 트레일러(T1)보다 큰 크기를 가질 수 있고 보다 복잡할 수 있다.

[0041] 호스트 프로세서(420)는 필요에 따라 메모리 프로토콜(MEM) 및 비일관적 프로토콜(IO) 중 하나의 프로토콜을 선택할 수 있고, 선택된 프로토콜에 기초하여 장치(410)에 액세스할 수 있다. 일부 실시예들에서, 호스트 프로세서(420)는 문턱값을 초과하는 크기의 데이터를 위하여 비일관적 프로토콜(IO)을 선택할 수 있는 한편, 문턱값 이하의 크기를 데이터를 위하여 메모리 프로토콜(MEM)을 선택할 수 있다. 예를 들면, 호스트 프로세서(420)는 작은 크기(예컨대, 64B)의 데이터를 기입하거나 독출하기 위하여, 작은 오버헤드, 즉 제1 헤더(H1) 및 제1 트레일러(T1)를 포함하는 제1 메시지(MSG1)를 장치(410)와 송수신할 수 있다. 또한, 호스트 프로세서(420)는 큰 크기(예컨대, 4KB)의 데이터를 기입하거나 독출하기 위하여, 제1 메시지(MSG1)를 반복하여 송신하거나 수신하는 대신, 제2 메시지(MSG2)를 장치(410)와 송수신할 수 있다.

[0042] 장치(410)에 포함된 메모리가 호스트 프로세서(420)의 SCM(storage class memory)으로 사용되는 경우, 페이지 크기(page size), 즉 4KB의 데이터의 스왑-인(swap-in) 및 스왑-아웃(swap-out)이 빈번하게 발생할 수 있다. 메모리 프로토콜(MEM)에 기초한 액세스만이 지원되는 경우, 4KB 데이터의 스왑-인 및 스왑-아웃은 과도한 제1 메시지들을 유발할 수 있고, 이에 따라 메모리 액세스, 예컨대 DMA(direct memory access)는 매우 비효율적일 수 있다. 그러나, 전술된 바와 같이, 비일관적 프로토콜(IO)에 기초한 메모리 액세스가 지원되는 경우, 메모리 프로토콜(MEM) 대신 비일관적 프로토콜(IO)이 선택될 수 있고, 비일관적 프로토콜(IO)에 기초한 데이터의 스왑-인 및 스왑-아웃에 기인하여 효율적인 DMA가 달성될 수 있다.

[0043] 도 5는 본 개시의 예시적 실시예에 따른 어드레스 맵(500)을 나타내는 도면이다. 일부 실시예들에서, 도 4의 호스트 프로세서(420)는 어드레스 맵(500)에 기초하여 장치(410)에 포함된 메모리에 액세스할 수 있고, 이하에서 도 5는 도 4를 참조하여 설명될 것이다.

[0044] 도 5를 참조하면, 호스트 프로세서(420)는 ACPI(advanced configuration and power interface), 즉 메모리 프로토콜(MEM)에 기초하여 어드레스 0x2000에 대응하는 메모리의 영역에 액세스할 수 있다. 예를 들면, 어드레스 0x2000에 대응하는 메모리의 영역에 저장된 데이터를 독출하기 위하여, 호스트 프로세서(420)는 메모리 프로토콜(MEM)에 기초하여 어드레스 0x2000을 포함하는 메시지 MemRD를 장치(410)에 제공할 수 있다.

[0045] 호스트 프로세서(420)는 BAR, 즉 비일관적 프로토콜(IO)에 기초하여 어드레스 0x2000에 대응하는 메모리의 영역에 액세스할 수 있다. 예를 들면, 어드레스 0x2000에 대응하는 메모리의 영역은 BAR1에 대응할 수 있고, 어드레스 0x2000에 대응하는 메모리 영역에 저장된 데이터를 독출하기 위하여, 호스트 프로세서(420)는 비일관적 프로토콜(IO)에 기초하여, BAR1에 대응하는 어드레스 0x4000를 포함하는 메시지 MemRD를 장치(410)에 제공할 수 있다. 결과적으로, 호스트 프로세서(420)는 비일관적 프로토콜(IO)의 BAR에 대응하는 어드레스 및 메모리 프로토콜(MEM)의 어드레스의 2개 어드레스를 통해서 메모리의 동일한 영역에 액세스할 수 있다.

[0046] 도 6는 본 개시의 예시적 실시예에 따른 시스템(600)을 나타내는 블록도이다. 도 6에 도시된 바와 같이, 시스템(600)은 호스트 프로세서(620), 장치(610) 및 장치 메모리(630)를 포함할 수 있고, 장치(610)는, 도 3a의 장치(310a)와 유사하게 물리 계층(611), 다중-프로토콜 멀티플렉서(612), 제1 인터페이스 회로(613), 제2 인터페이스

스 회로(614), 메모리 인터페이스(615)를 포함할 수 있고, 제3 인터페이스 회로(616) 및 일관성 엔진(617)을 더 포함할 수 있다. 이하에서, 도 6에 대한 설명 중 도 1, 도 3a 및 도 3b에 대한 설명과 중복되는 내용은 생략될 것이다.

[0047] 도 3a를 참조하여 전송된 바와 같이, 제1 인터페이스 회로(613)는 메모리 프로토콜(MEM)에 기초하여 호스트 프로세서(620)에 장치 메모리(630)에 대한 액세스(즉, 제1 액세스)를 제공할 수 있고, 제2 인터페이스 회로(614)는 비일관적 프로토콜(IO)에 기초하여 호스트 프로세서(620)에 장치 메모리(630)에 대한 액세스(즉, 제2 액세스)를 제공할 수 있다. 이에 따라, 장치 메모리(630)는 호스트 프로세서(620) 및/또는 버스에 연결된 다른 장치들(예컨대, 가속기, 스토리지 등)에 의해서 상이한 프로토콜들에 기초하여 액세스될 수 있고, 장치(610)는 장치 메모리(630)의 일관성을 해소하기 위하여 제3 인터페이스 회로(616) 및 일관성 엔진(617)을 더 포함할 수 있다.

[0048] 일관성 엔진(617)은, 도 6에 도시된 바와 같이, 장치 메모리(630)에 대한 제1 인터페이스 회로(613)를 통한 액세스(즉, 제1 액세스) 및 장치 메모리(630)에 대한 제2 인터페이스 회로(614)를 통한 액세스(즉, 제2 액세스)를 모니터링할 수 있다. 일관성 엔진(617)은 제1 액세스 및 제2 액세스의 모니터링 결과에 기초하여 장치 메모리(630)의 일관성을 검증할 수 있고, 검증 결과에 기초하여 일관성을 해소할 수 있다. 예를 들면, 일관성 엔진(617)은 일관성 해소를 위하여 제1 액세스 및 제2 액세스의 순서를 조정할 수도 있고, 호스트 프로세서(620)에 일관성의 해소를 요청할 수도 있다. 제3 인터페이스 회로(616)는 일관적 프로토콜(CACHE)에 기초하여 호스트 프로세서와 통신할 수 있고, 일관적 프로토콜(CACHE)에 기초하여 일관성 엔진(617)으로부터 수신된 메시지, 예컨대 일관성 해소의 요청을 호스트 프로세서에 제공할 수 있다.

[0049] 도 7a 및 도 7b는 본 개시의 예시적 실시예들에 다른 시스템의 예시들을 나타내는 블록도들이다. 구체적으로, 도 7a 및 도 7b는, 데이터의 스왑-인 및 스왑-아웃을 수행하는 시스템들(700a, 700b)을 각각 나타낸다. 도 7a 및 도 7b에서 해칭된 넓은 화살표가 데이터의 스왑-인 및/또는 스왑-아웃에 의한 데이터 흐름을 나타낸다. 도 4을 참조하여 전송된 바와 같이, 데이터의 스왑-인 및 스왑-아웃은 페이지 크기의 단위로 수행될 수 있다. 이하에서, 도 7a 및 도 7b에 대한 설명 중 상호 중복되는 내용은 생략될 것이다.

[0050] 도 7a를 참조하면, 시스템(700a)은 호스트 프로세서(720a), 스위치(750a), 장치(710a), 제1 장치 메모리(731a) 및 제2 장치 메모리(732a)를 포함할 수 있다. 일부 실시예들에서, 시스템(700a)은 호스트 메모리를 더 포함할 수 있고, 호스트 프로세서(720a)는 호스트 메모리에 액세스하기 위한 메모리 컨트롤러를 포함할 수 있다. 일부 실시예들에서, 장치(710a), 제1 장치 메모리(731a) 및 제2 장치 메모리(732a)는 유형 3 CXL 장치로서 총괄적으로 지칭될 수 있다. 스위치(750a)는 메시지 라우터로서 기능할 수 있고, 호스트 프로세서(720a) 및 장치(710a) 뿐만 아니라 추가적인 장치들에 연결될 수 있다. 장치(710a)는, 도 3a의 장치(310a)와 유사하게 물리 계층(711a), 다중-프로토콜 멀티플렉서(712a), 제1 인터페이스 회로(713a), 제2 인터페이스 회로(714a), 메모리 인터페이스(715a)를 포함할 수 있고, 제2 인터페이스 회로(714a)는 제2 장치 메모리(732a)와 통신할 수 있다. 제1 장치 메모리(731a)는 도 3a의 장치 메모리(330a)에 대응할 수 있고, 도면들을 참조하여 전송된 바와 같이, 제2 인터페이스 회로(714a)는 비일관적 프로토콜(IO)에 기초하여 수신된 메시지에 응답하여 제1 장치 메모리(731a)에 대한 액세스(즉, 제2 액세스)를 제공할 수 있다.

[0051] 제2 장치 메모리(732a)는 비일관적 프로토콜(IO)에 기초하여 호스트 프로세서(720a)에 의해서 액세스될 수 있다. 예를 들면, 제2 장치 메모리(732a)는, CXL.io에 기초하여 액세스될 수 있는 NVMe 스토리지를 포함할 수 있다. 호스트 프로세서(720a)는 제2 장치 메모리(732a)에 저장된 데이터를 제1 장치 메모리(731a)로 복사하는 것을 지시하는 메시지를 비일관적 프로토콜(IO)에 기초하여 장치(710a)에 제공할 수 있다. 제2 인터페이스 회로(714a)는, 호스트 프로세서(720a)로부터 비일관적 프로토콜(IO)에 기초하여 수신된 메시지에 응답하여, 비일관적 프로토콜(IO)에 기초하여 제2 장치 메모리(732a)에 저장된 데이터를 스위치(750a)에 제공할 수 있다. 스위치(750a)는 장치(710a)로부터 수신된 데이터를 다시 장치(710a)로 라우팅할 수 있고, 제2 인터페이스 회로(714a)는 비일관적 프로토콜(IO)에 기초하여 스위치(750a)로부터 수신된 데이터를 제2 인터페이스 회로(714a), 제1 인터페이스 회로(713a) 및 메모리 인터페이스(715a)를 통해서 제1 장치 메모리(731a)에 기입할 수 있다.

[0052] 도 7b를 참조하면, 시스템(700b)은 호스트 프로세서(720b) 및 장치(710b)를 포함할 수 있다. 일부 실시예들에서, 도 7a의 시스템(700a)과 같이, 시스템(700b)은 호스트 프로세서(720b) 및 장치(710b) 사이에 스위치를 더 포함할 수도 있다. 장치(710b)는 도 7a의 장치(710a)와 유사하게, 물리 계층(711b), 다중-프로토콜 멀티플렉서(712b), 제1 인터페이스 회로(713b), 제2 인터페이스 회로(714b), 메모리 인터페이스(715b), 제1 장치 메모리(731b) 및 제2 장치 메모리(732b)를 포함할 수 있다.

- [0053] 호스트 프로세서(720b)는 제2 장치 메모리(732b)에 저장된 데이터를 제1 장치 메모리(731b)에 복사하는 것을 지시하는 메시지를 비일관적 프로토콜(IO)에 기초하여 장치(710b)에 제공할 수 있다. 제2 인터페이스 회로(714b)는, 호스트 프로세서(720b)로부터 비일관적 프로토콜(IO)에 기초하여 수신된 메시지에 응답하여, 제2 장치 메모리(732b)로부터 수신된 데이터를 제1 인터페이스 회로(713b)에 제공할 수 있고, 제1 인터페이스 회로(713b)는 메모리 인터페이스(715b)를 통해서 데이터를 제1 장치 메모리(731b)에 저장할 수 있다. 이에 따라, 도 7a의 장치(710a)와 상이하게, 데이터의 스왑-인 및 스왑-아웃은 도 7b의 장치(710b) 내에서 수행될 수 있다.
- [0054] 도 8은 본 개시의 예시적 실시예에 따른 시스템(800)을 나타내는 블록도이다. 도 8에 도시된 바와 같이, 시스템(800)은 장치(810), 스위치(820), 장치 메모리(830), 스토리지(840) 및 가속기(850)를 포함할 수 있고, 장치(810), 스토리지(840) 및 가속기(850)는 스위치(820)를 통해서 상호 통신할 수 있다. 장치(810)는, 도 3a의 장치(310a)와 유사하게, 물리 계층(811), 다중-프로토콜 멀티플렉서(812), 제1 인터페이스 회로(813), 제2 인터페이스 회로(814) 및 메모리 인터페이스(815)를 포함할 수 있다.
- [0055] 장치 메모리(830)는 호스트 프로세서뿐만 아니라 버스에 연결된 다른 구성요소들, 예컨대 스토리지(840) 및/또는 가속기(850)에 의해서 다중 프로토콜에 기초하여 액세스될 수 있다. 예를 들면, 도 8에 도시된 바와 같이, 호스트 프로세서 또는 DMA(direct memory access) 등에 의해서, 제1 장치 메모리(830)에 저장된 데이터를 스토리지(840)에 복사하거나 가속기(850)에 제공하는 메시지가 비일관적 프로토콜(IO)에 기초하여 장치(810)에 전달될 수 있다. 제2 인터페이스 회로(814)는 비일관적 프로토콜(IO)에 기초하여 수신된 메시지에 응답하여, 제1 인터페이스 회로(813) 및 메모리 인터페이스(815)를 통해서 장치 메모리(830)로부터 데이터를 수신할 수 있고, 수신된 데이터를 스위치(820)에 제공할 수 있다. 스위치(820)는 장치(810)로부터 수신된 데이터를 스토리지(840) 또는 가속기(850)로 라우팅할 수 있고, 스위치(820)로부터 제공된 데이터는 스토리지(840)에 저장되거나 가속기(850)에 의해서 사용(예컨대, 처리)될 수 있다.
- [0056] 도 9는 본 개시의 예시적 실시예에 따라 다중 프로토콜에 기초하여 메모리에 액세스하기 위한 방법을 나타내는 순서도이다. 도 9에 도시된 바와 같이, 다중 프로토콜에 기초하여 메모리에 액세스하기 위한 방법은 단계 S10 및 단계 S20을 포함할 수 있다. 일부 실시예들에서, 도 9의 방법은 도 3a의 장치(310a)에서 수행될 수 있고, 이하에서 도 9는 도 3a를 참조하여 설명될 것이다.
- [0057] 도 9를 참조하면, 단계 S10에서 메모리에 대한 액세스를 요청하는 메시지가 제2 프로토콜에 기초하여 수신될 수 있다. 예를 들면, 장치(310a)는 제1 프로토콜, 예컨대 메모리 프로토콜(MEM)에 기초하여 액세스되는 장치 메모리(330a)를 포함할 수 있고, 제2 인터페이스 회로(314a)는 장치 메모리(330a)에 대한 액세스를 요청하는 메시지를 제2 프로토콜, 예컨대 비일관적 프로토콜(IO)에 기초하여 버스로부터 수신할 수 있다. 제2 인터페이스 회로(314a)는 비일관적 프로토콜(IO)에 기초하여 수신된 메시지가 장치 메모리(330a)에 대한 액세스를 요청하는 것을 식별할 수 있다.
- [0058] 단계 S20에서, 메모리가 액세스될 수 있다. 예를 들면, 제2 인터페이스 회로(314a)는 제1 인터페이스 회로(313a)와 통신할 수 있고, 비일관적 프로토콜(IO)에 기초하여 수신된 메시지에 응답하여, 제1 인터페이스 회로(313a) 및 메모리 인터페이스(315a)를 통해서 장치 메모리(330a)에 액세스할 수 있다. 버스로부터 수신된 메시지가 데이터의 기입을 지시하는 경우, 제2 인터페이스 회로(314a)는 비일관적 프로토콜(IO)에 기초하여 버스로부터 수신된 데이터를 제1 인터페이스 회로(313a)를 통해서 메모리 인터페이스(315a)에 제공할 수 있고, 메모리 인터페이스(315a)는 데이터를 w장치 메모리(330a)에 기입할 수 있다. 또한, 버스로부터 수신된 메시지가 데이터의 독출을 지시하는 경우, 제2 인터페이스 회로(314a)는 장치 메모리(330a)로부터 독출된 데이터를 메모리 인터페이스(315a)로부터 제1 인터페이스 회로(313a)를 통해서 수신할 수 있고, 비일관적 프로토콜(IO)에 기초하여 버스에 데이터를 송신할 수 있다.
- [0059] 도 10a 및 도 10b는 본 개시의 예시적 실시예들에 따라 다중 프로토콜에 기초하여 메모리에 액세스하기 위한 방법의 예시들을 나타내는 순서도들이다. 구체적으로, 도 10a 및 도 10b는, 제1 프로토콜 및 제2 프로토콜에 의해서 액세스가능한 제1 장치 메모리 및 제2 프로토콜에 의해서 액세스가능한 제2 장치 메모리를 포함하는 장치에서 메모리에 액세스하기 위한 방법들을 나타낸다. 일부 실시예들에서, 도 10a의 방법은 도 7a의 장치(710a)에 의해서 수행될 수 있고, 도 10b의 방법은 도 7b의 장치(710b)에 의해서 수행될 수 있다. 이하에서, 도 10a 및 도 10b는 도 7a 및 도 7b를 참조하여 설명될 것이다.
- [0060] 도 10a를 참조하면, 다중 프로토콜에 기초하여 메모리에 액세스하기 위한 방법은 복수의 단계들(S30, S40, S50, S60)을 포함할 수 있다. 단계 S30에서, 데이터의 복사를 요청하는 메시지가 제2 프로토콜에 기초하여 수신될 수 있다. 예를 들면, 도 7a의 제2 인터페이스 회로(714a)는 제2 장치 메모리(732a)에 저장된 데이터를 제1 장치 메

모리(731a)에 복사하는 것을 요청하는 메시지를 제2 프로토콜, 즉 비일관적 프로토콜(IO)에 기초하여 호스트 프로세서(720a)로부터 수신할 수 있다.

- [0061] 단계 S40에서, 제2 프로토콜에 기초하여 데이터가 버스로 송신될 수 있다. 예를 들면, 제2 인터페이스 회로(714a)는 비일관적 프로토콜(IO)에 기초하여 수신된 메시지에 응답하여, 제2 장치 메모리(732a)로부터 수신된 데이터를 비일관적 프로토콜(IO)에 기초하여 버스로 송신할 수 있다.
- [0062] 단계 S50에서, 제2 프로토콜에 기초하여 버스로부터 데이터가 수신될 수 있다. 예를 들면, 스위치(750a)는 단계 S40에서 장치(710a)로부터 수신된 데이터를 다시 장치(710a)로 라우팅할 수 있고, 제2 인터페이스 회로(714a)는 비일관적 프로토콜(IO)에 기초하여 데이터를 수신할 수 있다.
- [0063] 단계 S60에서, 데이터가 제1 장치 메모리(731a)에 저장될 수 있다. 전술된 바와 같이, 제1 장치 메모리(731a)는 제1 프로토콜, 즉 메모리 프로토콜(MEM)을 지원하는 제1 인터페이스 회로(713a)뿐만 아니라 제2 프로토콜, 즉 비일관적 프로토콜(IO)을 지원하는 제2 인터페이스 회로(714a)에 의해서 액세스될 수 있다. 제2 인터페이스 회로(714a)는 단계 S50에서 수신된 데이터를 제1 인터페이스 회로(713a)를 통해서 메모리 인터페이스(715a)에 제공할 수 있고, 메모리 인터페이스(715a)는 데이터를 제1 장치 메모리(731a)에 저장할 수 있다.
- [0064] 도 9b를 참조하면, 다중 프로토콜에 기초하여 메모리에 액세스하기 위한 방법은 복수의 단계들(S70, S80, S90)을 포함할 수 있다. 단계 S70에서, 데이터의 복사를 요청하는 메시지가 제2 프로토콜에 기초하여 수신될 수 있다. 예를 들면, 도 7b의 제2 인터페이스 회로(714b)는 제2 장치 메모리(732b)에 저장된 데이터를 제1 장치 메모리(731b)에 복사하는 것을 요청하는 메시지를 제2 프로토콜, 즉 비일관적 프로토콜(IO)에 기초하여 호스트 프로세서(720b)로부터 수신할 수 있다.
- [0065] 단계 S80에서 제2 장치 메모리(732b)로부터 데이터가 독출될 수 있고, 단계 S90에서 독출된 데이터가 제1 장치 메모리(731b)에 저장될 수 있다. 예를 들면, 제2 인터페이스 회로(714b)는 단계 S70에서 수신된 메시지에 응답하여, 제2 장치 메모리(732b)로부터 독출된 데이터를 수신할 수 있다. 전술된 바와 같이, 제2 인터페이스 회로(714b)는 제2 프로토콜, 즉 비일관적 프로토콜(IO)에 기초하여 제1 장치 메모리(731b)에 대한 액세스를 제공하기 위하여, 제1 인터페이스 회로(713b)를 통해서 메모리 인터페이스(715b)와 통신할 수 있고, 이에 따라 단계 S80에서 제2 장치 메모리(732b)로부터 수신된 데이터를 메모리 인터페이스(715b)에 제공할 수 있다. 메모리 인터페이스(715b)는 제1 인터페이스 회로(713b)를 통해서 제2 인터페이스 회로(714b)로부터 수신된 데이터를 제1 장치 메모리(731b)에 저장할 수 있다.
- [0066] 도 11a 및 도 11b는 본 개시의 예시적 실시예들에 따른 시스템의 예시들을 나타내는 도면이다. 구체적으로, 도 11a 및 도 11b의 블록도들은 다수의 CPU들을 포함하는 시스템들(5a, 5b)을 나타낸다.
- [0067] 도 11a를 참조하면, 시스템(5a)은, 제1 CPU(11a) 및 제2 CPU(21a)를 포함할 수 있고, 제1 CPU(11a) 및 제2 CPU(21a)에 각각 연결된 제1 DDR(double data rate) 메모리(12a) 및 제2 DDR 메모리(22a)를 포함할 수 있다. 제1 CPU(11a) 및 제2 CPU(21a)는 프로세서 상호 연결 기술에 기초한 상호연결 시스템(30a)을 통해서 연결될 수 있다. 도 11a에 도시된 바와 같이, 상호연결 시스템(30a)은, 적어도 하나의 CPU간(CPU-to-CPU) 일관적 링크를 제공할 수 있다.
- [0068] 시스템(5a)은, 제1 CPU(11a)와 통신하는 제1 입출력 장치(13a) 및 제1 가속기(14a)를 포함할 수 있고, 제1 가속기(14a)에 연결된 제1 장치 메모리(15a)를 포함할 수 있다. 제1 CPU(11a) 및 제1 입출력 장치(13a)는 버스(16a)를 통해서 통신할 수 있고, 제1 CPU(11a) 및 제1 가속기(14a)는 버스(17a)를 통해서 통신할 수 있다. 또한, 시스템(5a)은, 제2 CPU(21a)와 통신하는 제2 입출력 장치(23a) 및 제2 가속기(24a)를 포함할 수 있고, 제2 가속기(24a)에 연결된 제2 장치 메모리(25a)를 포함할 수 있다. 제2 CPU(21a) 및 제2 입출력 장치(23a)는 버스(26a)를 통해서 통신할 수 있고, 제2 CPU(21a) 및 제2 가속기(24a)는 버스(27a)를 통해서 통신할 수 있다. 일부 실시예들에서, 제1 장치 메모리(15a) 및 제2 장치 메모리(25a) 중 적어도 하나는 시스템(5a)에서 생략될 수도 있다.
- [0069] 버스들(16a, 17a, 18a, 26a, 27a, 28a)을 통해서 다수의 프로토콜들 중 적어도 일부에 에 기초한 통신이 수행될 수 있다. 예를 들면, 버스들(16a, 17a, 18a, 26a, 27a, 28a) 각각을 통해서, 비일관적 프로토콜에 기초하여 초기 설정 등의 정보가 전달될 수 있다. 또한, 버스들(17a, 27a)을 통해서, 일관적 프로토콜 및/또는 메모리 프로토콜에 기초하여 메시지 및/또는 데이터가 전달될 수 있다. 또한, 버스들(18a, 28a)을 통해서, 메모리 프로토콜에 기초하여 메시지 및/또는 데이터가 전달될 수 있다.
- [0070] 제1 CPU(1a)는 데이터의 크기에 기초하여, 다수의 프로토콜들, 예컨대 메모리 프로토콜 및 비일관적 프로토콜

중 하나를 선택할 수 있고, 선택된 프로토콜에 기초하여 제1 장치 메모리(15a)에 액세스할 수 있다. 이에 따라, 최적의 프로토콜이 선택될 수 있고, 장치 메모리(15a)의 액세스와 관련된 레이턴시가 감소할 수 있다. 또한, 제2 CPU(21a)는 데이터의 크기에 기초하여, 다수의 프로토콜들, 예컨대 메모리 프로토콜 및 비일관적 프로토콜 중 하나를 선택할 수 있고, 선택된 프로토콜에 기초하여 제2 장치 메모리(25a)에 액세스할 수 있다. 이에 따라, 최적의 프로토콜이 선택될 수 있고, 장치 메모리(25a)의 액세스와 관련된 레이턴시가 감소할 수 있다.

[0071] 도 11b를 참조하면, 시스템(5b)은, 도 11a의 시스템(5a)과 유사하게, 제1 및 제2 CPU(11b, 21b), 제1 및 제2 DDR 메모리(12b, 22b), 제1 및 제2 입출력 장치(13b, 23b) 및 제1 및 제2 가속기(14b, 24b)를 포함할 수 있는 한편, 원격 원거리 메모리(40)를 더 포함할 수 있다. 제1 CPU(11b) 및 제2 CPU(21b)는 버스들(18b, 28b)을 통해서 원격 원거리 메모리(40)에 각각 연결될 수 있다. 원격 원거리 메모리(40)는, 시스템(5b)에서 메모리의 확장을 위하여 사용될 수 있고, 버스들(18b, 28b)은 메모리 확장 포트로서 사용될 수 있다. 일부 실시예들에서, 원격 원거리 메모리(40)는 시스템(5b)에서 생략될 수도 있다.

[0072] 도 12는 본 개시의 예시적 실시예에 따른 시스템을 포함하는 데이터 센터(7)를 나타내는 블록도이다. 일부 실시예들에서, 도면들을 참조하여 전술된 시스템은 어플리케이션 서버 및/또는 스토리지 서버로서 데이터 센터(7)에 포함될 수 있다.

[0073] 도 12를 참조하면, 데이터 센터(7)는 다양한 데이터를 수집하고 서비스를 제공할 수 있고, 데이터 스토리지 센터로 지칭될 수도 있다. 예를 들면, 데이터 센터(7)는 검색 엔진 및 데이터 베이스 운용을 위한 시스템일 수 있고, 은행 등의 기업 또는 정부기관에서 사용되는 컴퓨팅 시스템일 수도 있다. 도 12에 도시된 바와 같이, 데이터 센터(7)는 어플리케이션 서버들(50_1 내지 50_n) 및 스토리지 서버들(60_1 내지 60_m)을 포함할 수 있다(m 및 n은 1보다 큰 정수). 어플리케이션 서버들(50_1 내지 50_n)의 개수 n 및 스토리지 서버들(60_1 내지 60_m)의 개수 m은 실시예에 따라 다양하게 선택될 수 있고, 어플리케이션 서버들(50_1 내지 50_n)의 개수 n 및 스토리지 서버들(60_1 내지 60_m)의 개수 m은 상이할 수 있다(m≠n).

[0074] 어플리케이션 서버(50_1 또는 50_n)는 프로세서(51_1 또는 51_n), 메모리(52_1 또는 52_n), 스위치(53_1 또는 53_n), NIC(network interface controller)(54_1 또는 54_n) 및 스토리지 장치(55_1 또는 55_n) 중 적어도 하나를 포함할 수 있다. 프로세서(51_1 또는 51_n)는 어플리케이션 서버(50_1 또는 50_n)의 전반적인 동작을 제어할 수 있고, 메모리(52_1 또는 52_n)에 액세스하여 메모리(52_1 또는 52_n)에 로딩된 명령어들(instructions) 및/또는 데이터를 실행할 수 있다. 메모리(52_1 또는 52_n)는 비제한적인 예시로서, DDR SDRAM(Double Data Rate Synchronous DRAM), HBM(High Bandwidth Memory), HMC(Hybrid Memory Cube), DIMM(Dual In-line Memory Module), Optane DIMM 또는 NVMDIMM(Non-Volatile DIMM)를 포함할 수 있다. 실시예에 따라, 어플리케이션 서버(50_1 또는 50_n)에 포함되는 프로세서들의 개수 및 메모리들의 개수는 다양하게 선택될 수 있다. 일부 실시예들에서, 프로세서(51_1 또는 51_n)와 메모리(52_1 또는 52_n)는 프로세서-메모리 페어를 제공할 수 있다. 일부 실시예들에서, 프로세서(51_1 또는 51_n)와 메모리(52_1 또는 52_n)의 개수는 상이할 수 있다. 프로세서(51_1 또는 51_n)는 단일 코어 프로세서 또는 다중 코어 프로세서를 포함할 수 있다. 일부 실시예들에서, 도 12에서 점선으로 도시된 바와 같이, 어플리케이션 서버(50_1 또는 50_n)에서 스토리지 장치(55_1 또는 55_n)는 생략될 수도 있다. 스토리지 서버(60_1 또는 60_m)에 포함되는 스토리지 장치(55_1 또는 55_n)의 개수는 실시예에 따라 다양하게 선택될 수 있다. 프로세서(51_1 또는 51_n), 메모리(52_1 또는 52_n), 스위치(53_1 또는 53_n), NIC(network interface controller)(54_1 또는 54_n) 및/또는 스토리지 장치(55_1 또는 55_n)는, 도면들을 참조하여 전술된 링크를 통해서 상호 통신할 수 있다.

[0075] 스토리지 서버(60_1 또는 60_m)는 프로세서(61_1 또는 61_m), 메모리(62_1 또는 62_m), 스위치(63_1 또는 63_m), NIC(64_1 또는 64_n) 및 스토리지 장치(65_1 또는 65_m) 중 적어도 하나를 포함할 수 있다. 프로세서(61_1 또는 61_m) 및 메모리(62_1 또는 62_m)는, 전술된 어플리케이션 서버(50_1 또는 50_n)의 프로세서(51_1 또는 51_n) 및 메모리(52_1 또는 52_n)와 유사하게 동작할 수 있다.

[0076] 어플리케이션 서버들(50_1 내지 50_n) 및 스토리지 서버들(60_1 내지 60_m)은 네트워크(70)를 통해 상호 통신할 수 있다. 일부 실시예들에서, 네트워크(70)는 FC(Fibre Channel) 또는 이더넷(Ethernet) 등을 이용하여 구현될 수 있다. FC는 상대적으로 고속의 데이터 전송에 사용되는 매체일 수 있고, 고성능/고가용성을 제공하는 광 스위치가 사용될 수 있다. 네트워크(70)의 액세스 방식에 따라 스토리지 서버들(60_1 내지 60_m)은 파일 스토리지, 블록 스토리지, 또는 오브젝트 스토리지로서 제공될 수 있다.

[0077] 일부 실시예들에서, 네트워크(70)는 SAN(Storage Area Network)와 같은 스토리지 전용 네트워크일 수 있다. 예를 들어, SAN은 FC 네트워크를 이용할 수 있고 FCP(FC Protocol)에 따라 구현된 FC-SAN일 수 있다. 다르게는,

SAN은 TCP/IP 네트워크를 이용하고 iSCSI(SCSI over TCP/IP 또는 Internet SCSI) 프로토콜에 따라 구현된 IP-SAN일 수 있다. 일부 실시예들에서, 네트워크(70)는 TCP/IP 네트워크와 같은 일반 네트워크일 수 있다. 예를 들면, 네트워크(70)는 FCoE(FC over Ethernet), NAS(Network Attached Storage), NVMe-oF(NVMe over Fabrics) 등의 프로토콜에 따라 구현될 수 있다.

- [0078] 이하에서, 어플리케이션 서버(50_1) 및 스토리지 서버(60_1)가 주로 설명되나, 어플리케이션 서버(50_1)에 대한 설명은 다른 어플리케이션 서버(예컨대, 20_n)에도 적용될 수 있고, 스토리지 서버(60_1)에 대한 설명은 다른 스토리지 서버(예컨대, 60_m)에도 적용될 수 있는 점이 유의된다.
- [0079] 어플리케이션 서버(50_1)는 사용자 또는 클라이언트가 저장을 요청한 데이터를 네트워크(70)를 통해 스토리지 서버들(60_1 내지 60_m) 중 하나에 저장할 수 있다. 또한, 어플리케이션 서버(50_1)는 사용자 또는 클라이언트가 독출을 요청한 데이터를 스토리지 서버들(60_1 내지 60_m) 중 하나로부터 네트워크(70)를 통해 획득할 수 있다. 예를 들어, 어플리케이션 서버(50_1)는 웹 서버 또는 DBMS(Database Management System) 등으로 구현될 수 있다.
- [0080] 어플리케이션 서버(50_1)는 네트워크(70)를 통해 다른 어플리케이션 서버(50_n)에 포함된 메모리(52_n) 및/또는 스토리지 장치(55_n)에 액세스할 수 있고, 그리고/또는 네트워크(70)를 통해 스토리지 서버들(60_1 내지 60_m)에 포함된 메모리들(62_1 내지 62_m) 및/또는 스토리지 장치들(65_1 내지 65_m)에 액세스할 수 있다. 이에 따라, 어플리케이션 서버(50_1)는 어플리케이션 서버들(50_1 내지 50_n) 및/또는 스토리지 서버들(60_1 내지 60_m)에 저장된 데이터에 대해 다양한 동작들을 수행할 수 있다. 예를 들어, 어플리케이션 서버(50_1)는 어플리케이션 서버들(50_1 내지 50_n) 및/또는 스토리지 서버들(60_1 내지 60_m) 사이에서 데이터를 이동시키거나 복사(copy)하기 위한 명령어를 실행할 수 있다. 이 때 데이터는 스토리지 서버들(60_1 내지 60_m)의 스토리지 장치로(65_1 내지 65_m)부터 스토리지 서버들(60_1 내지 60_m)의 메모리들(62_1 내지 62_m)을 통해서 또는 직접적으로 어플리케이션 서버들(50_1 내지 50_n)의 메모리(52_1 내지 52_n)로 이동될 수 있다. 일부 실시예들에서, 네트워크(70)를 통해 이동하는 데이터는 보안 또는 프라이버시를 위해 암호화된 데이터일 수 있다.
- [0081] 스토리지 서버(60_1)에서, 인터페이스(IF)는 프로세서(61_1)와 컨트롤러(CTRL)의 물리적 연결 및 NIC(64_1)와 컨트롤러(CTRL)의 물리적 연결을 제공할 수 있다. 예를 들어, 인터페이스(IF)는 스토리지 장치(65_1)를 전용 케이블로 직접 접속하는 DAS(Direct Attached Storage) 방식으로 구현될 수 있다. 또한, 예를 들어, 인터페이스(IF)는 ATA(Advanced Technology Attachment), SATA(Serial ATA), e-SATA(external SATA), SCSI(Small Computer Small Interface), SAS(Serial Attached SCSI), PCI(Peripheral Component Interconnection), PCIe(PCI express), NVMe(NVM express), IEEE 1394, USB(universal serial bus), SD(secure digital) 카드, MMC(multi-media card), eMMC(embedded multi-media card), UFS(Universal Flash Storage), eUFS(embedded Universal Flash Storage), CF(compact flash) 카드 인터페이스 등과 같은 다양한 인터페이스 방식으로 구현될 수 있다.
- [0082] 스토리지 서버(60_1)에서, 스위치(63_1)는 프로세서(61_1)의 제어에 따라 프로세서(61_1)와 스토리지 장치(65_1)를 선택적으로 접속시키거나, NIC(64_1)과 스토리지 장치(65_1)를 선택적으로 접속시킬 수 있다.
- [0083] 일부 실시예들에서, NIC(64_1)는 네트워크 인터페이스 카드, 네트워크 어댑터 등을 포함할 수 있다. NIC(54_1)는 유선 인터페이스, 무선 인터페이스, 블루투스 인터페이스, 광학 인터페이스 등에 의해 네트워크(70)에 연결될 수 있다. NIC(54_1)는 내부 메모리, DSP, 호스트 버스 인터페이스 등을 포함할 수 있으며, 호스트 버스 인터페이스를 통해 프로세서(61_1) 및/또는 스위치(63_1) 등과 연결될 수 있다. 일부 실시예들에서, NIC(64_1)는 프로세서(61_1), 스위치(63_1), 스토리지 장치(65_1) 중 적어도 하나와 통합될 수도 있다.
- [0084] 어플리케이션 서버(50_1 내지 50_n) 또는 스토리지 서버(60_1 내지 60_m)에서 프로세서(51_1 내지 51_m, 61_1 내지 61_n)는 스토리지 장치들(55_1 내지 55_n, 65_1 내지 65_m) 또는 메모리(52_1 내지 52_n, 62_1 내지 62_m)로 커맨드를 전송하여 데이터를 프로그램하거나 리드할 수 있다. 이 때 데이터는 ECC(Error Correction Code) 엔진을 통해 에러 정정된 데이터일 수 있다. 데이터는 데이터 버스 변환(Data Bus Inversion: DBI) 또는 데이터 마스킹(Data Masking: DM) 처리된 데이터로서, CRC(Cyclic Redundancy Code) 정보를 포함할 수 있다. 데이터는 보안 또는 프라이버시를 위해 암호화된 데이터일 수 있다.
- [0085] 스토리지 장치(55_1 내지 55_n, 65_1 내지 65_m)는 프로세서(51_1 내지 51_m, 61_1 내지 61_n)로부터 수신된 리드 커맨드에 응답하여, 제어 신호 및 커맨드/어드레스 신호를 비휘발성 메모리 장치(예컨대 NAND 플래시 메모리 장치)(NVM)로 전송할 수 있다. 이에 따라 비휘발성 메모리 장치(NVM)로부터 데이터를 독출하는 경우, 독출

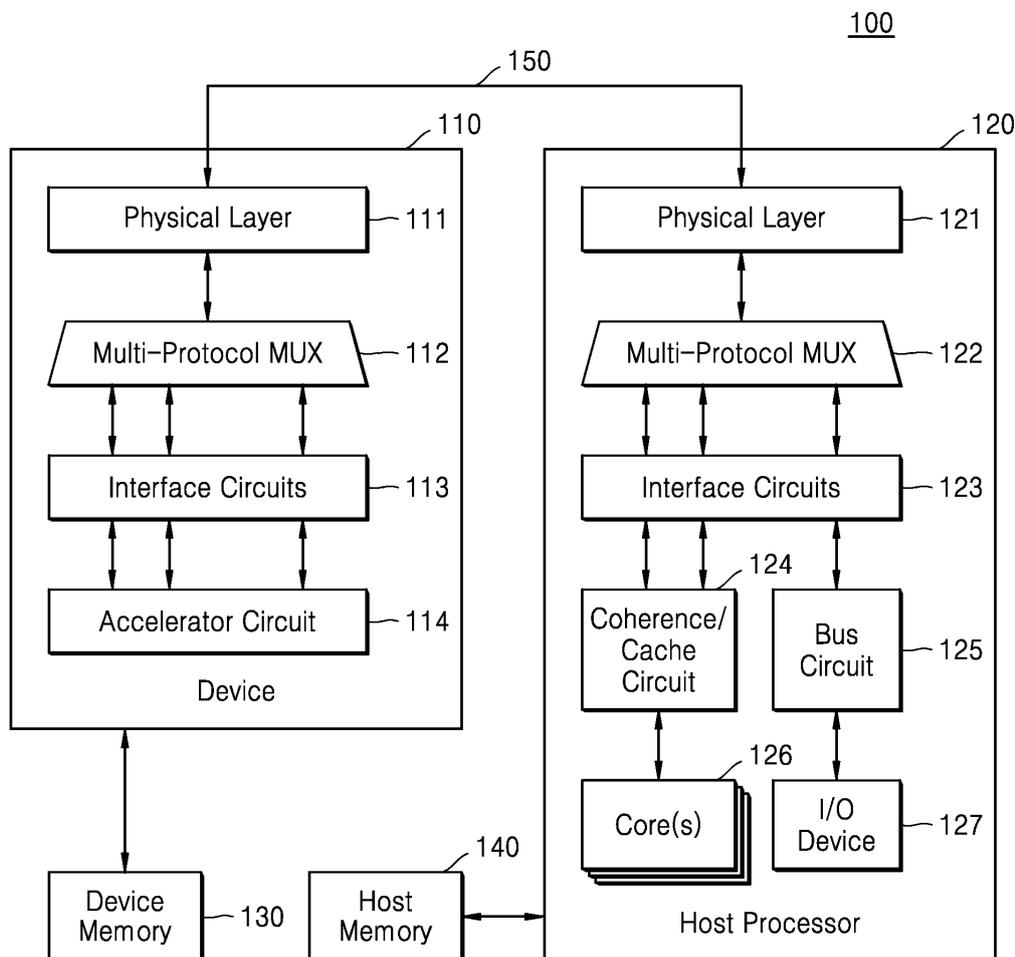
인에이블 신호는 데이터 출력 제어 신호로 입력되어, 데이터를 DQ 버스로 출력하는 역할을 할 수 있다. 독출 인에이블 신호를 이용하여 데이터 스트로브 신호를 생성할 수 있다. 커맨드와 어드레스 신호는 기입 인에이블 신호의 상승 엣지 또는 하강 엣지에 따라 래치될 수 있다.

[0086] 컨트롤러(CTRL)는 스토리지 장치(65_1)의 동작을 전반적으로 제어할 수 있다. 일 실시예에서, 컨트롤러(CTRL)는 SRAM(Static Random Access Memory)을 포함할 수 있다. 컨트롤러(CTRL)는 기입 커맨드에 응답하여 비휘발성 메모리 장치(NVM)에 데이터를 기입할 수 있고, 또는 독출 커맨드에 응답하여 비휘발성 메모리 장치(NVM)로부터 데이터를 독출할 수 있다. 예를 들어, 기입 커맨드 및/또는 독출 커맨드는 호스트, 예컨대 스토리지 서버(60_1) 내의 프로세서(61_1), 다른 스토리지 서버(60_m) 내의 프로세서(61_m) 또는 어플리케이션 서버(50_1 또는 50_n) 내의 프로세서(51_1 또는 51_n)로부터 제공된 요청에 기초하여 생성될 수 있다. 버퍼(BUF)는 비휘발성 메모리 장치(NVM)에 기입될 데이터 또는 비휘발성 메모리 장치(NVM)로부터 독출된 데이터를 임시 저장(버퍼링)할 수 있다. 일부 실시예들에서 버퍼(BUF)는 DRAM을 포함할 수 있다. 또한, 버퍼(BUF)는 메타 데이터를 저장할 수 있고, 메타 데이터는 사용자 데이터 또는 비휘발성 메모리 장치(NVM)를 관리하기 위해 컨트롤러(CTRL)에서 생성된 데이터를 지칭할 수 있다. 스토리지 장치(65_1)는 보안 또는 프라이버시를 위해 SE(Secure Element)를 포함할 수 있다.

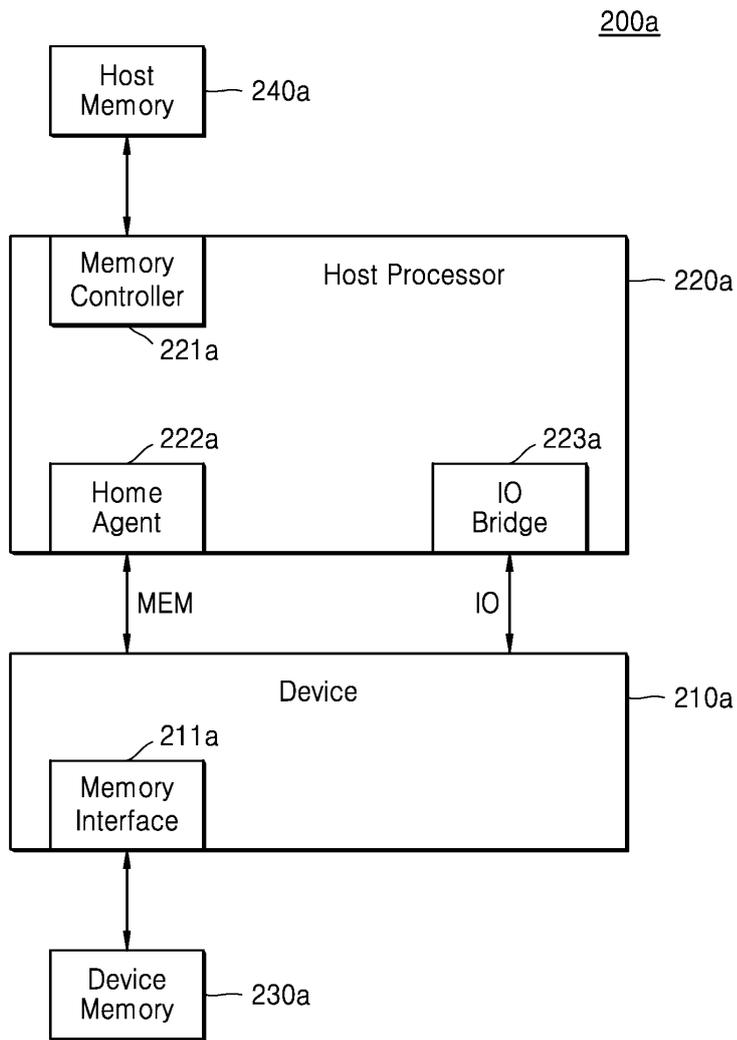
[0087] 이상에서와 같이 도면과 명세서에서 예시적인 실시예들이 개시되었다. 본 명세서에서 특정한 용어를 사용하여 실시예들이 설명되었으나, 이는 단지 본 개시의 기술적 사상을 설명하기 위한 목적에서 사용된 것이지 의미 한정이나 특허청구범위에 기재된 본 개시의 범위를 제한하기 위하여 사용된 것은 아니다. 그러므로 본 기술분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다. 따라서, 본 개시의 진정한 기술적 보호범위는 첨부된 특허청구범위의 기술적 사상에 의해 정해져야 할 것이다.

도면

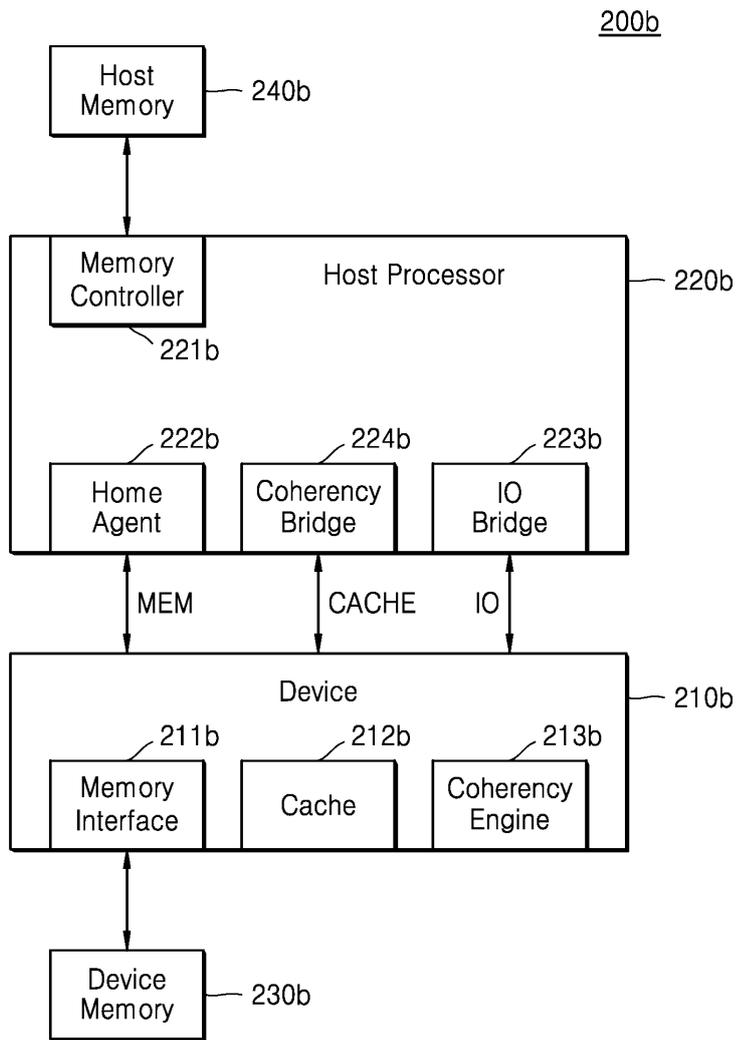
도면1



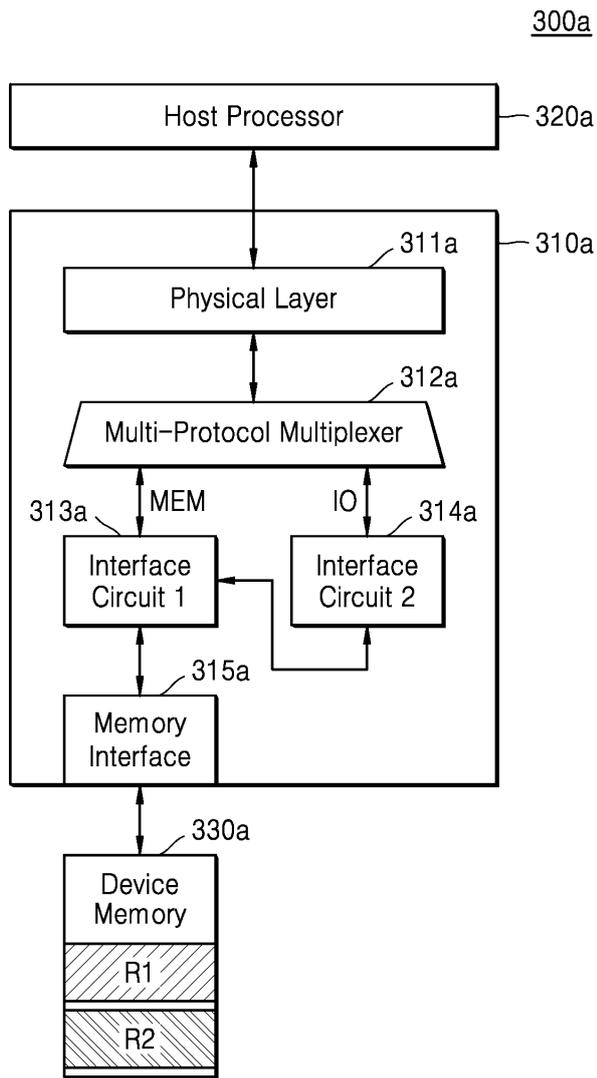
도면2a



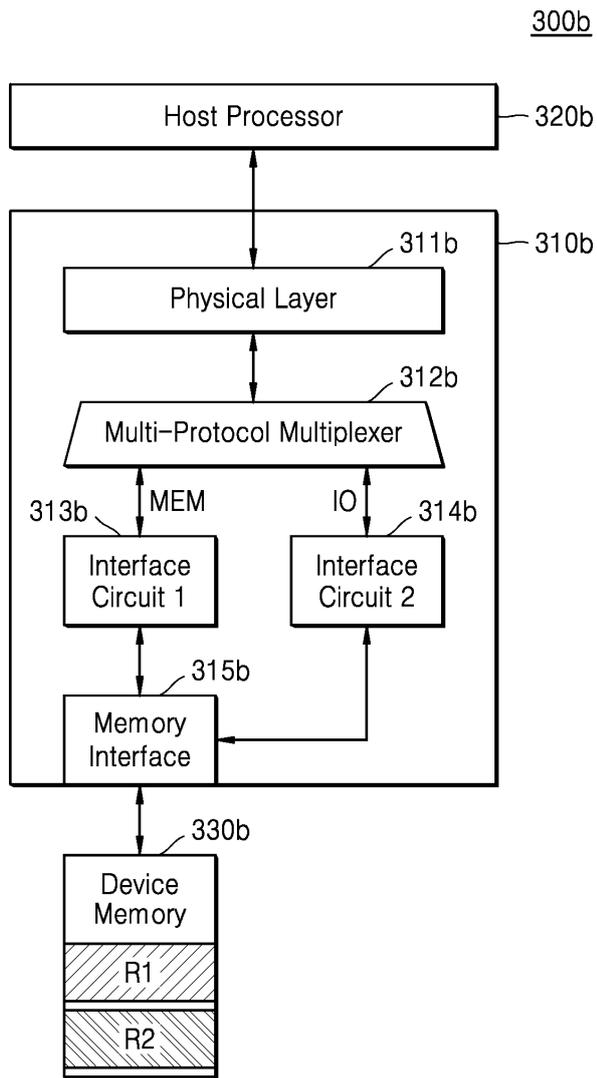
도면2b



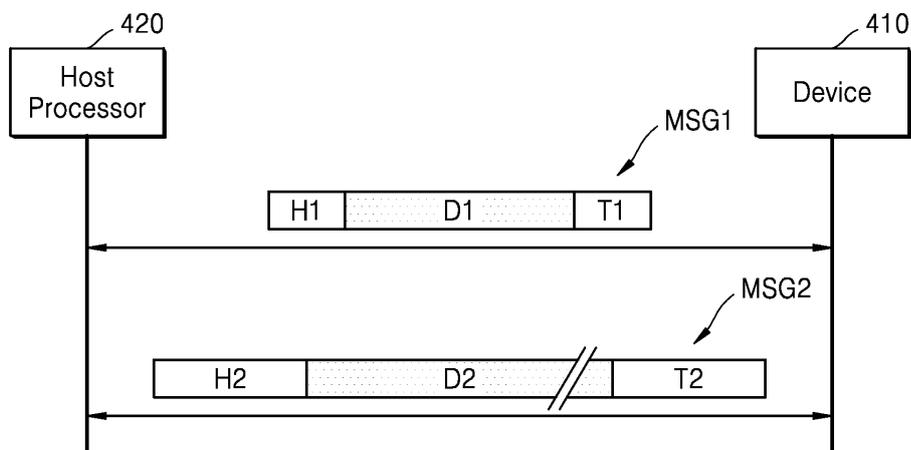
도면3a



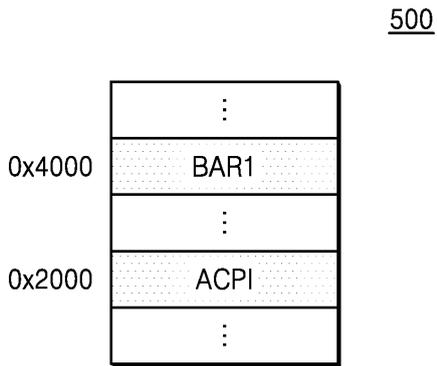
도면3b



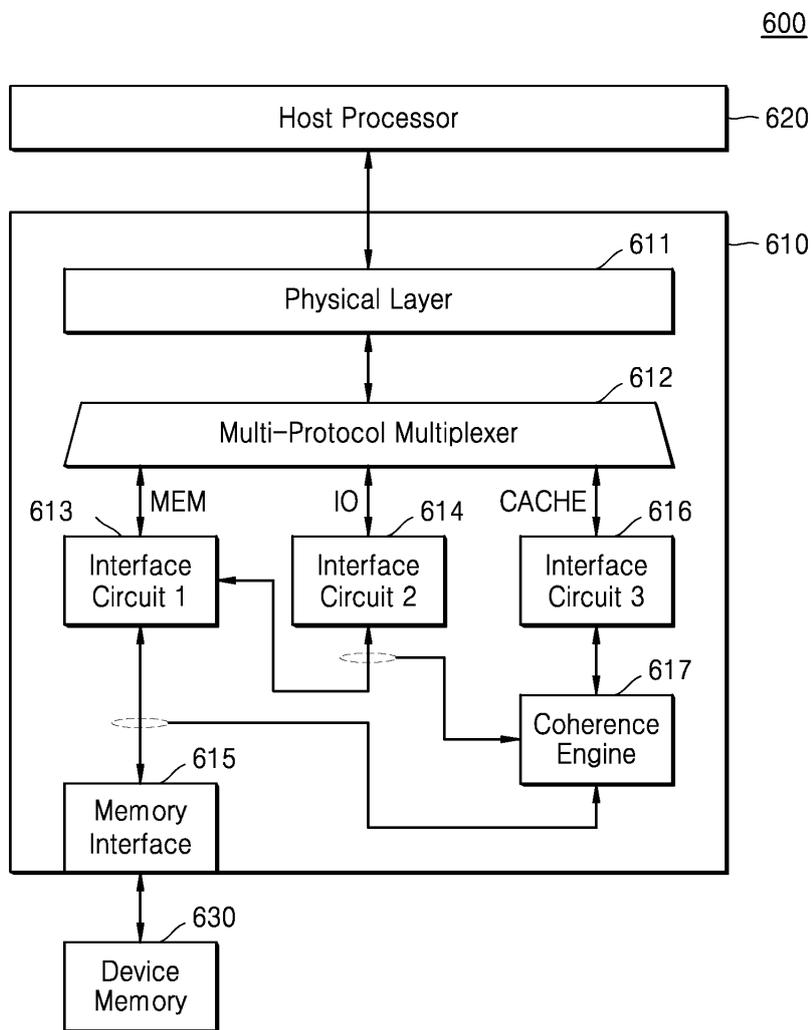
도면4



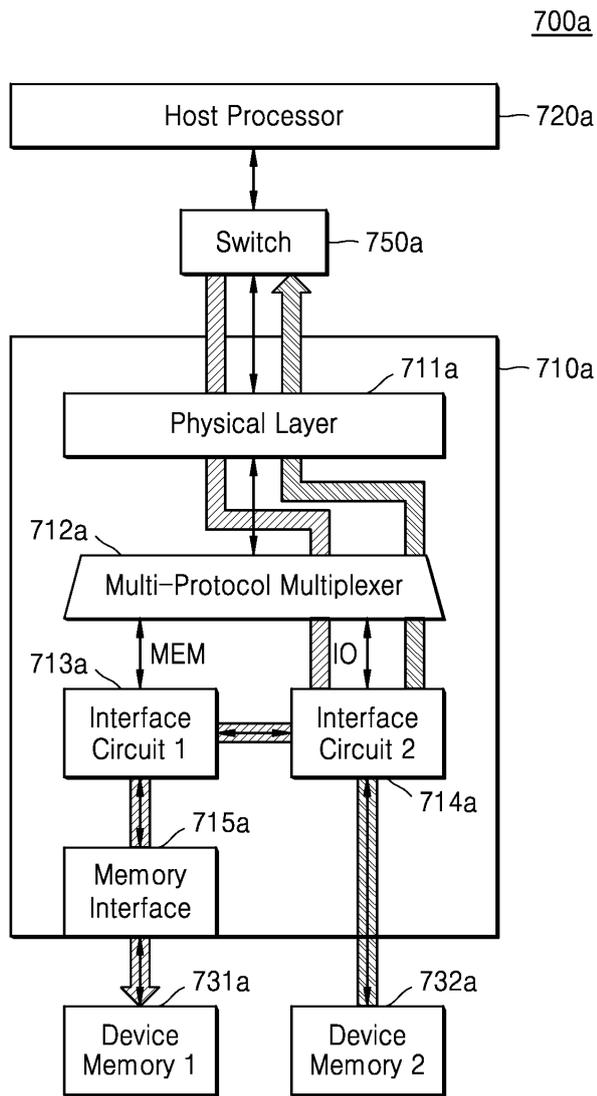
도면5



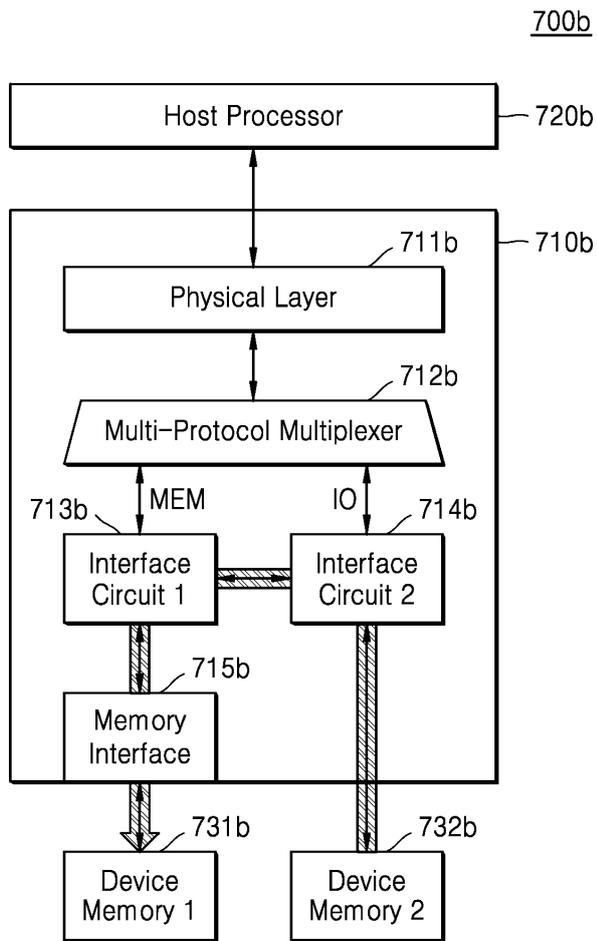
도면6



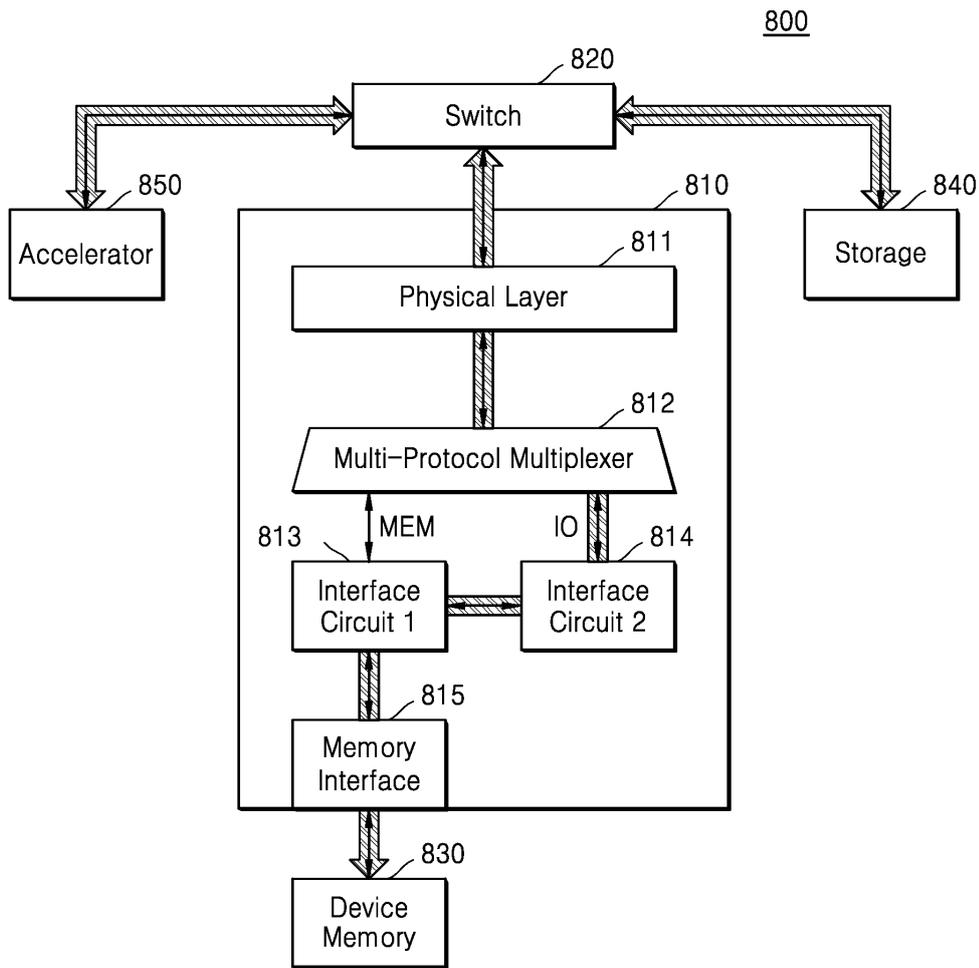
도면7a



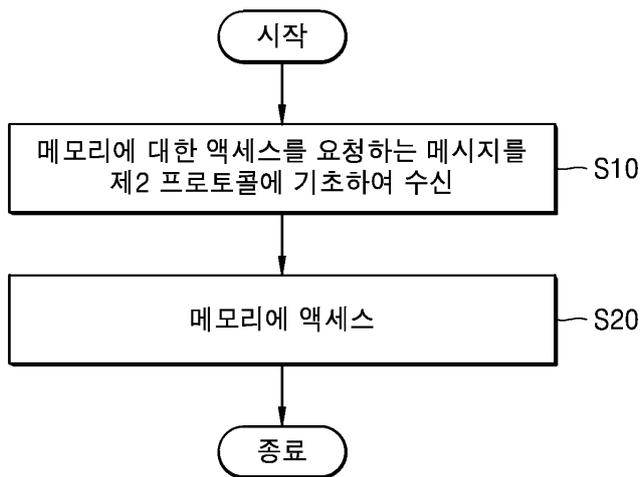
도면7b



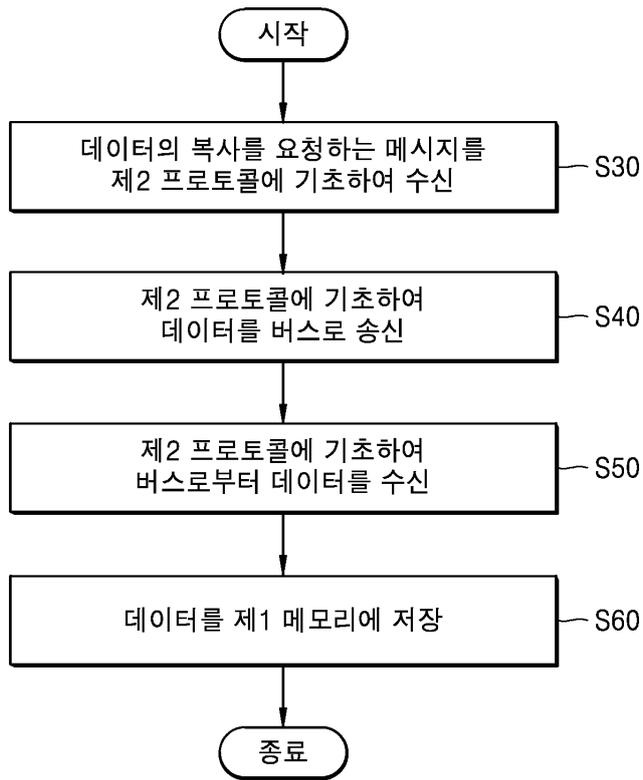
도면8



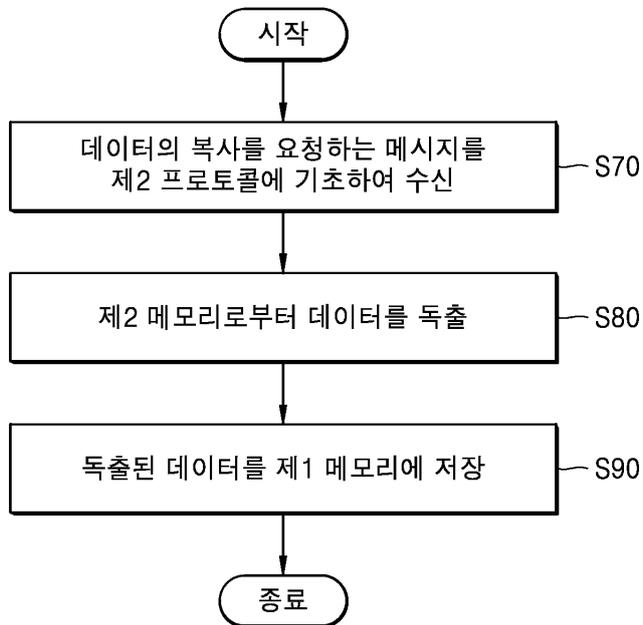
도면9



도면10a

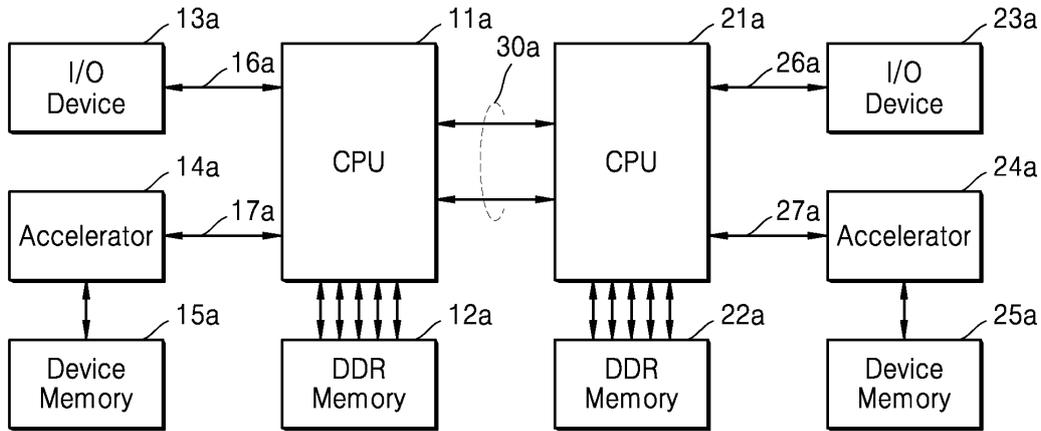


도면10b



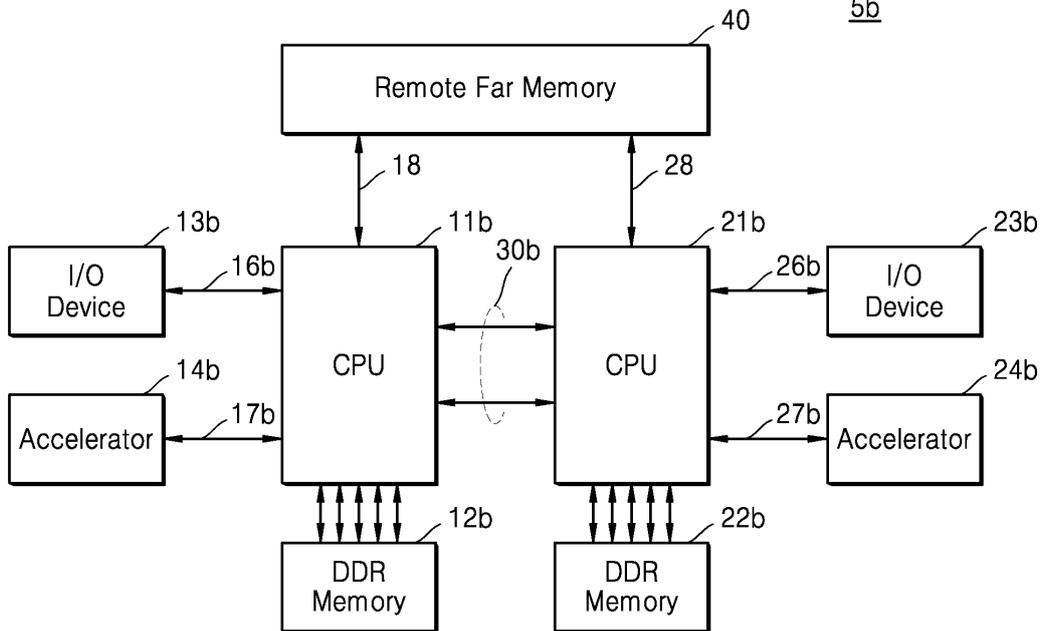
도면11a

5a



도면11b

5b



도면12

