

US Patent & Trademark Office

Patent Public Search | Text View

United States Patent Application Publication

20250265216

Kind Code

A1

Publication Date

August 21, 2025

Inventor(s)

LEE; Kyunghan et al.

STORAGE DEVICE, MEMORY DEVICE, AND SYSTEM INCLUDING STORAGE DEVICE AND MEMORY DEVICE

Abstract

A storage device includes a nonvolatile memory device that stores user data, and a storage controller that controls the nonvolatile memory device under control of a host device. The storage controller includes a storage interface circuit that communicates with the host device through a compute express link (CXL) interface, a NAND interface circuit that communicates with the nonvolatile memory device, and a processor that loads map data from an external memory device through the storage interface circuit and controls the nonvolatile memory device through the NAND interface circuit based on the map data.

Inventors: LEE; Kyunghan (Suwon-si, KR), LEE; Jae-Gon (Suwon-si, KR), LEE; Chon Yong (Suwon-si, KR)

Applicant: SAMSUNG ELECTRONICS CO., LTD. (Suwon-si, KR)

Family ID: 1000008578040

Assignee: SAMSUNG ELECTRONICS CO., LTD. (Suwon-si, KR)

Appl. No.: 19/199709

Filed: May 06, 2025

Foreign Application Priority Data

KR

10-2022-0054254

May. 02, 2022

Related U.S. Application Data

parent US continuation 18132734 20230410 parent-grant-document US 12321302 child US 19199709

Publication Classification

Int. Cl.: G06F13/42 (20060101); G06F13/16 (20060101); G06F13/40 (20060101)

U.S. Cl.:

CPC G06F13/4221 (20130101); G06F13/1668 (20130101); G06F13/4081 (20130101);

Background/Summary

CROSS-REFERENCE TO RELATED APPLICATIONS [0001] This application is a continuation application of U.S. patent application Ser. No. 18/132,734 filed Apr. 10, 2023, which claims priority under 35 U.S.C. § 119 to Korean Patent Application No. 10-2022-0054254 filed on May 2, 2022, in the Korean Intellectual Property Office, the disclosures of which are incorporated by reference herein in their entirety.

BACKGROUND

[0002] The present disclosure relates to a semiconductor memory device, and more particularly, relate to a storage device, a memory device, and a system including the storage device and the memory device.

[0003] A semiconductor memory device is classified as a volatile memory device, in which stored data disappears when a power is turned off, such as a static random access memory (SRAM) or a dynamic random access memory (DRAM), or a nonvolatile memory device, in which stored data is retained even when a power is turned off, such as a flash memory device, a phase-change RAM (PRAM), a magnetic RAM (MRAM), a resistive RAM (RRAM), or a ferroelectric RAM (FRAM).

[0004] A storage device such as a solid state drive may include a flash memory. A system of logical block addresses used in a host is different from a system of physical block addresses of the flash memory of the storage device. For this reason, the storage device may perform negotiation between a logical block address of the host and a physical block address of the flash memory by using map data in which the logical block address and the physical block address are mapped. A solid state drive may conform to a form factor.

[0005] As the capacity of the flash memory increases, the capacity of the map data may also increase. As such, a high-capacity buffer memory may be required to be used in the storage device, thereby causing costs for new research and development.

SUMMARY

[0006] Embodiments of the present disclosure provide a storage device capable of managing a large amount of map data without a high-capacity buffer memory dedicated for the storage device, a memory device, and a system including the storage device and the memory device.

[0007] According to an embodiment, a storage device includes a nonvolatile memory device that stores user data, and a storage controller that controls the nonvolatile memory device based on control of a host device. The storage controller includes a storage interface circuit that communicates with the host device through a compute express link (CXL) interface, a NAND interface circuit that communicates with the nonvolatile memory device, and a processor that loads map data from an external memory device placed outside the storage device through the storage interface circuit and controls the nonvolatile memory device through the NAND interface circuit based on the map data.

[0008] According to an embodiment, a memory device includes a buffer memory that stores map data of an external storage device, and a memory controller that communicates with the external storage device and a host device through a compute express link (CXL) interface and sends at least

a portion of the map data present in the buffer memory to the external storage device through the CXL interface.

[0009] According to an embodiment, a system includes a storage device, a memory device, and a compute express link (CXL) switch that is connected with the storage device and the memory device through a CXL interface and is configured to arbitrate communications between the storage device and the memory device. The memory device stores map data of the storage device, and the storage device exchanges, through the CXL switch, at least a portion of the map data with the memory device.

Description

BRIEF DESCRIPTION OF THE FIGURES

[0010] The above and other aspects and features of the present disclosure will become apparent by describing in detail example embodiments thereof with reference to the accompanying drawings, in which:

[0011] FIG. 1 is a block diagram illustrating a computing system including a storage device;

[0012] FIG. 2 is a block diagram illustrating a computing system to which a storage system according to an embodiment of the present disclosure is applied;

[0013] FIG. 3 is a block diagram illustrating components of a computing system of FIG. 2 in detail, according to an embodiment;

[0014] FIG. 4 is a flowchart illustrating an initialization operation or a power-up operation of a computing system of FIG. 3, according to an embodiment;

[0015] FIG. 5 is a diagram for describing an operation in which a computing system stores map data, according to an embodiment;

[0016] FIGS. 6 and 7 are diagrams for describing an operation in which map data are stored in a CXL memory, according to an embodiment;

[0017] FIG. 8 is a flowchart illustrating a read operation for CXL storage of FIG. 2, according to an embodiment;

[0018] FIG. 9 is a flowchart illustrating a write operation for CXL storage of FIG. 2, according to an embodiment;

[0019] FIG. 10 is a flowchart illustrating a power-off operation of a computing system of FIG. 2, according to an embodiment;

[0020] FIG. 11 is a block diagram illustrating a computing system according to an embodiment;

[0021] FIG. 12 is a block diagram illustrating a computing system according to an embodiment;

[0022] FIG. 13 is a block diagram illustrating a computing system according to an embodiment;

[0023] FIG. 14 is a block diagram illustrating a computing system according to an embodiment;

[0024] FIG. 15 is a block diagram illustrating a computing system according to an embodiment; and

[0025] FIG. 16 is a block diagram illustrating a data center to which a computing system according to an embodiment of the present disclosure is applied.

DETAILED DESCRIPTION

[0026] Below, embodiments of the present disclosure will be described in detail and clearly to such an extent that an ordinary one in the art easily implements them.

[0027] FIG. 1 is a block diagram illustrating a computing system including a storage device.

Referring to FIG. 1, a computing system 10 may include a host 11, a plurality of memory devices 12a and 12b, and a storage device 13. The host 11 may control an overall operation of the computing system 10. The plurality of memory devices 12a and 12b may be used as a working memory or a system memory of the host 11.

[0028] The storage device 13 may include a storage controller 13a, a buffer memory 13b, and a

nonvolatile memory **13c**. Under control of the host **11**, the storage controller **13a** may store data in the nonvolatile memory **13c** or may send data stored in the nonvolatile memory **13c** to the host **11**. [0029] The buffer memory **13b** may store a variety of information necessary for the storage device **13** to operate. For example, the storage controller **13a** may manage data stored in the nonvolatile memory **13c** by using map data. The map data may include information about relationship between a logical block address managed by the host **11** and a physical block address of the nonvolatile memory **13c**.

[0030] In an embodiment, the buffer memory **13b** may be a high-speed memory such as a DRAM. As the capacity of the nonvolatile memory **13c** increases, the size of necessary map data may increase. However, because the capacity of the buffer memory **13b** included in the single storage device **13** is limited, it is impossible to cope with the increase in the size of the map data due to the increase in the capacity of the nonvolatile memory **13c**.

[0031] FIG. 2 is a block diagram illustrating a computing system to which a storage system according to an embodiment of the present disclosure is applied. Referring to FIG. 2, a computing system **100** may include a host **101**, a plurality of memory devices **102a** and **102b**, CXL (Compute express Link) storage **110**, and a CXL memory **120**. In an embodiment, the computing system **100** may be included in user devices such as a personal computer, a laptop computer, a server, a media player, and a digital camera or automotive devices such as a navigation system, a black box, and an automotive electronic device/part. Alternatively, the computing system **100** may be a mobile system such as a mobile phone, a smartphone, a tablet personal computer (PC), a wearable device, a health care device, or an Internet of things (IoT) device.

[0032] The host **101** may control an overall operation of the computing system **100**. In an embodiment, the host **101** may be one of various processors such as a central processing unit (CPU), a graphics processing unit (GPU), a neural processing unit (NPU), and a data processing unit (DPU). In an embodiment, the host **101** may include a single core processor or a multi-core processor.

[0033] The plurality of memory devices **102a** and **102b** may be used as a main memory or a system memory of the computing system **100**. In an embodiment, each of the plurality of memory devices **102a** and **102b** may be a dynamic random access memory (DRAM) device and may have the form factor of the dual in-line memory module (DIMM). However, the present disclosure is not limited thereto. For example, the plurality of memory devices **102a** and **102b** may include a nonvolatile memory such as a flash memory, a phase change RAM (PRAM), a resistive RAM (RRAM), or a magnetic RAM (MRAM).

[0034] The plurality of memory devices **102a** and **102b** may directly communicate with the host **101** through the DDR interface. In an embodiment, the host **101** may include a memory controller configured to control the plurality of memory devices **102a** and **102b**. However, the present disclosure is not limited thereto. For example, the plurality of memory devices **102a** and **102b** may communicate with the host **101** through various interfaces.

[0035] The CXL storage **110** may include a CXL storage controller **111** and a nonvolatile memory NVM. Under control of the host **101**, the CXL storage controller **111** may store data in the nonvolatile memory NVM or may send data stored in the nonvolatile memory NVM to the host **101**. In an embodiment, the nonvolatile memory NVM may be a NAND flash memory, but the present disclosure is not limited thereto.

[0036] The CXL memory **120** may include a CXL memory controller **121** and a buffer memory BFM. Under control of the host **101**, the CXL memory controller **121** may store data in the buffer memory BFM or may send data stored in the buffer memory BFM to the host **101**. In an embodiment, the buffer memory BFM may be a volatile memory such as a DRAM, but the present disclosure is not limited thereto.

[0037] In an embodiment, the host **101**, the CXL storage **110**, and the CXL memory **120** may be configured to share the same interface. For example, the host **101**, the CXL storage **110**, and the

CXL memory **120** may communicate with each other through a CXL interface IF_CXL. In an embodiment, the CXL interface IF_CXL may indicate a low-latency and high-bandwidth link that supports coherency, memory access, and dynamic protocol muxing of 10 protocols such that various connections between accelerators, memory devices, or various electronic devices are possible.

[0038] In an embodiment, unlike the storage device **13** of FIG. **1**, the CXL storage **110** may not include a separate buffer memory for storing or managing map data. In this case, the CXL storage **110** may require a buffer memory for storing or managing the map data. In an embodiment, at least a partial area of the CXL memory **120** may be used as a buffer memory of the CXL storage **110**. In this case, a mapping table that is managed by the CXL storage controller **111** of the CXL storage **110** may be stored in the CXL memory **120**. For example, at least a partial area of the CXL memory **120** may be allocated for a buffer memory of the CXL storage **110** (i.e., for an area dedicated for the CXL storage **110**) by the host **101**.

[0039] In an embodiment, the CXL storage **110** may access the CXL memory **120** through the CXL interface IF_CXL. For example, the CXL storage **110** may store the mapping table in the allocated area of the CXL memory **120** or may read the mapping table from the allocated area of the CXL memory **120**. Under control of the CXL storage **110**, the CXL memory **120** may store data (e.g., the map data) in the buffer memory BFM or may send the data (e.g., the map data) stored in the buffer memory BFM to the CXL storage **110**.

[0040] As described with reference to FIG. **1**, the conventional storage device **13** stores and manages the map data by using the buffer memory **13b** included therein. As the capacity of the storage device **13** increases, the size of the map data increase, thereby causing an increase in a capacity requirement of the buffer memory **13b** included in the storage device **13**. However, there is a limitation on an increase in capacity due to the structure and physical characteristics of the buffer memory **13b** included in the storage device **13**; in this case, a design change or additional integration of the buffer memory **13b** is required.

[0041] In contrast, according to an embodiment of the present disclosure, the CXL storage **110** may use at least a partial area of the CXL memory **120** placed outside the CXL storage **110** as a buffer memory. In this case, because the CXL memory **120** is implemented independently of the CXL storage **110**, the CXL memory **120** may be implemented with a high-capacity memory. As such, even though the size of the map data increases due to an increase in the capacity of the CXL storage **110**, the map data may be normally managed by the CXL memory **120**.

[0042] In an embodiment, the storage controller **13a** of the conventional storage device **13** communicates with the host **11** through the host interface such as PCIe or NVMe, and communicates with the buffer memory **13b** through the memory interface such as a DDR interface or an LPDDR interface. That is, the storage controller **13a** of the conventional storage device **13** communicates with the host **11** placed outside and the buffer memory **13b** included therein, through different interfaces (i.e., heterogeneous interfaces).

[0043] In contrast, according to an embodiment of the present disclosure, the CXL storage controller **111** of the CXL storage **110** may communicate with the host **101** and the CXL memory **120** (i.e., a buffer memory) through the CXL interface IF_CXL. In other words, the CXL storage controller **111** of the CXL storage **110** may communicate with the host **101** and the CXL memory **120** through a homogeneous interface or a common interface and may use a partial area of the CXL memory **120** as a buffer memory.

[0044] Below, for convenience of description, it is assumed that the host **101**, the CXL storage **110**, and the CXL memory **120** communicate with each other through the CXL interface IF_CXL. However, the present disclosure is not limited thereto. For example, the host **101**, the CXL storage **110**, and the CXL memory **120** may communicate with each other based on various computing interfaces complying with the following: GEN-Z protocol, NV Link protocol, CCIX protocol, and Open CAPI protocol.

[0045] FIG. 3 is a block diagram illustrating components of a computing system of FIG. 2 in detail. Referring to FIGS. 2 and 3, the computing system **100** may include a CXL switch SW_CXL, the host **101**, the CXL storage **110**, and the CXL memory **120**.

[0046] The CXL switch SW_CXL may be a component included in the CXL interface IF_CXL. The CXL switch SW_CXL may be configured to arbitrate the communication between the host **101**, the CXL storage **110**, and the CXL memory **120**. For example, when the host **101** and the CXL storage **110** communicate with each other, the CXL switch SW_CXL may be configured to send information, which is provided from the host **101** or the CXL storage **110**, such as a request, data, a response, or a signal to the CXL storage **110** or the host **101**. When the host **101** and the CXL memory **120** communicate with each other, the CXL switch SW_CXL may be configured to send information, which is provided from the host **101** or the CXL memory **120**, such as a request, data, a response, or a signal to the CXL memory **120** or the host **101**. When the CXL storage **110** and the CXL memory **120** communicate with each other, the CXL switch SW_CXL may be configured to send information, which is provided from the CXL storage **110** or the CXL memory **120**, such as a request, data, a response, or a signal to the CXL memory **120** or the CXL storage **110**.

[0047] The host **101** may include a CXL host interface circuit **101a**. The CXL host interface circuit **101a** may communicate with the CXL storage **110** or the CXL memory **120** through the CXL switch SW_CXL.

[0048] In an embodiment, a configuration of the CXL storage **110** may be different from configurations of conventional storages. For example, a conventional storage device (e.g., an SSD including a DRAM buffer) stores and manages map data in the DRAM buffer included in the conventional storage device. In this case, a high-capacity DRAM buffer for storing the map data should be included in the conventional storage device. Alternatively, another type of conventional storage device (e.g., a DRAM-less SSD or a DRAM-less memory card) stores the entire map data in a nonvolatile memory (e.g., a NAND flash memory) included in the conventional storage device and loads and uses a portion of the map data onto an SRAM buffer. In this case, to load the map data, the access to the nonvolatile memory whose operating speed is lower than that of the DRAM buffer is frequently performed, thereby reducing the performance of operation.

[0049] In contrast, the CXL storage **110** according to the present disclosure may not include a separate DRAM buffer configured to store the map data. In this case, map data MD may be stored and managed in the CXL memory **120** placed outside the CXL storage **110**. As will be described below, because the CXL memory **120** supports a fast operating speed, the CXL storage **110** may have the same performance as the conventional storage device (e.g., a storage device including a DRAM). In addition, because the CXL memory **120** is placed outside the CXL storage **110**, it may be easy to cope with a large amount of map data of the CXL storage **110**. For example, it is assumed that a storage capacity of the CXL storage **110** is 128 TB and the CXL storage uses 4 KB mapping structure. In this case, about 128 GB of DRAM is required to store and manage the map data of the CXL storage **110**. It is difficult to embed a large capacity of DRAM into a storage device. However, when the DRAM for storing and managing the map data of the CXL storage **110** is placed outside the CXL storage **110**, it is easy to increase a capacity of the DRAM. Thus, according to the present disclosure, the large amount of map data of the CXL storage **110** is managed by using the CXL memory **120** (or, an external memory) without a performance degradation.

[0050] The CXL storage **110** may include the CXL storage controller **111** and the nonvolatile memory NVM. The CXL storage controller **111** may include a CXL storage interface circuit **111a**, a processor **111b**, a RAM **111c**, a flash translation layer (FTL) **111d**, an error correction code (ECC) engine **111e**, and a NAND interface circuit **111f**.

[0051] The CXL storage interface circuit **111a** may be connected with the CXL switch SW_CXL. The CXL storage interface circuit **111a** may communicate with the host **101** or the CXL memory

120 through the CXL switch SW_CXL. The storage interface circuit **111a** may conform to various form factors such as an E1.S form factor, an E1.L form factor, an E3.S form factor, an E3.L form factor, a PCIe AIC (CEM) form factor, an U.2 form factor, or an M.2 form factor.

[0052] The processor **111b** may be configured to control an overall operation of the CXL storage controller **111**. The RAM **111c** may be used as a working memory or a buffer memory of the CXL storage controller **111**. In an embodiment, the RAM **111c** may be an SRAM and may be used as a read buffer and a write buffer for the CXL storage **110**. In an embodiment, as will be described below, the RAM **111c** may be configured to temporarily store the map data MD read from the CXL memory **120** or a portion of the map data M D.

[0053] The FTL **111d** may perform various management operations for efficiently using the nonvolatile memory NVM. For example, the FTL **111d** may perform address translation between a logical block address managed by the host **101** and a physical block address used in the nonvolatile memory NVM, based on map data (or a mapping table). The FTL **111d** may perform a bad block management operation for the nonvolatile memory NVM. The FTL **111d** may perform a wear leveling operation for the nonvolatile memory NVM. The FTL **111d** may perform a garbage collection operation for the nonvolatile memory NVM.

[0054] In an embodiment, the FTL **111d** may be implemented in hardware, firmware, or software, or a combination thereof. In the case where the FTL **111d** is implemented in the form of firmware or software, program codes associated with the FTL **111d** may be stored in the RAM **111c** and may be driven by the processor **111b**. In the case where the FTL **111d** is implemented by hardware, hardware components configured to perform the above management operations may be implemented in the CXL storage controller **111**.

[0055] The ECC engine **111e** may perform error detection and correction on data read from the nonvolatile memory NVM. For example, the ECC engine **111** may generate parity bits for user data UD to be stored in the nonvolatile memory NVM, and the parity bits thus generated may be stored in the nonvolatile memory NVM together with the user data UD. When the user data UD are read from the nonvolatile memory NVM, the ECC engine **111** may detect and correct an error of the user data UD by using the parity bits read from the nonvolatile memory NVM together with the user data UD.

[0056] The NAND interface circuit **111f** may control the nonvolatile memory NVM such that data are stored in the nonvolatile memory NVM or data are read from the nonvolatile memory NVM. In an embodiment, the NAND interface circuit **111f** may be implemented to comply with the standard protocol such as a toggle interface or ON FI. For example, the nonvolatile memory NVM may include a plurality of NAND flash devices; in the case where the NAND interface circuit **111f** is implemented based on the toggle interface, the NAND interface circuit **111f** communicates with the plurality of NAND flash devices through a plurality of channels. The plurality of NAND flash devices may be connected with the plurality of channels through a multi-channel, multi-way structure.

[0057] The NAND interface circuit **111f** may send a chip enable signal/CE, a command latch enable signal CLE, an address latch enable signal ALE, a read enable signal/RE and a write enable signal/WE to the plurality of NAND flash devices through the plurality of channels. The NAND interface circuit **111f** and each NAND flash device may exchange a data signal DQ and a data strobe signal DQS through each channel.

TABLE-US-00001

| TABLE 1 | /CE | CLE | ALE | /WE | /RE | DQS | DQx | MODE | L | H | L | ↑ | H | X | CMD |
|---------------|-----|-----|-----|-----|-----|----------|-------------|---------------|---|---|---|----|---|----|---------|
| Command Input | L | L | H | ↑ | H | X | ADDR | Address Input | L | L | L | H | H | ↑↓ | DATA_in |
| Data Input | L | L | L | H | ↑↓ | DATA_out | Data Output | L | L | L | H | ↑↓ | | | |

[0058] Table 1 shows operating modes of a NAND flash device according to a state of each signal. Referring to Table 1, while the NAND flash device receives a command CM D or an address ADDR or receives/outputs data “DATA”, the chip enable signal/CE maintains a low state “L”. During a command input mode, the NAND interface circuit **111f** may control signal lines such that

the command latch enable signal CLE has a high level “H”, the address latch enable signal ALE has the low level “L”, the write enable signal/WE toggles between the high level “H” and the low level “L” and the read enable signal/RE has the high level “H”. During the command input mode, the NAND interface circuit **111f** may send the command CM D to the NAND flash device through data signals DQx in synchronization with the rising edge ↑ of the write enable signal/WE. The NAND flash device may identify the command CM D from the data signals DQx in response to the rising edge ↑ of the write enable signal/WE. During an address input mode, the NAND interface circuit **111f** may control signal lines such that the command latch enable signal CLE has the low level “L”, the address latch enable signal ALE has the high level “H”, the write enable signal/WE toggles between the high level “H” and the low level “L”, and the read enable signal/RE has the high level “H”. During the address input mode, the NAND interface circuit **111f** may send the address ADDR to the NAND flash device through the data signals DQx in synchronization with the rising edge ↑ of the write enable signal/WE. The NAND flash device may identify the address ADDR from the data signals DQx in response to the rising edge ↑ of the write enable signal/WE. In an embodiment, the address ADDR may be a value corresponding to a physical block address of the NAND flash device.

[0059] During a data input mode, the NAND interface circuit **111f** may control signal lines such that the command latch enable signal CLE has the low level “L”, the address latch enable signal ALE has the low level “L”, the write enable signal/WE has the high level “H”, the read enable signal/RE has the high level “H”, and the data strobe signal DQS toggles between the high level “H” and the low level “L”. During the data input mode, the NAND interface circuit **111f** may send the data “DATA” to the NAND flash device through the data signals DQx in synchronization with the rising edge ↑ and the falling edge ↓ of the data strobe signal DQS. The NAND flash device may identify the data “DATA” from the data signals DQx in response to the rising edge ↑ and the falling edge ↓ of the data strobe signal DQS.

[0060] During a data output mode, the NAND interface circuit **111f** may control signal lines that the command latch enable signal CLE has the low level “L”, the address latch enable signal ALE has the low level “L”, the write enable signal/WE has the high level “H”, and the read enable signal/RE toggles between the high level “H” and the low level “L”. During the data output mode, the NAND flash device may generate the data strobe signal DQS toggling between the high level “H” and the low level “L” in response to the read enable signal/RE. The NAND flash device may send the data “DATA” to the NAND interface circuit **111f** through the data signals DQx in synchronization with the rising edge ↑ and the falling edge ↓ of the data strobe signal DQS. The NAND interface circuit **111f** may identify the data “DATA” from the data signals DQx in response to the rising edge ↑ and the falling edge ↓ of the data strobe signal DQS.

[0061] The toggle interface described above is an example, and the present disclosure is not limited thereto.

[0062] The nonvolatile memory NVM may store or output the user data UD under control of the CXL storage controller **111**. The nonvolatile memory NVM may store or output the map data MD under control of the CXL storage controller **111**. In an embodiment, the map data M D stored in the nonvolatile memory NVM may include mapping information corresponding to the entire user data UD stored in the nonvolatile memory NVM. The map data MD stored in the nonvolatile memory NVM may be stored in the CXL memory **120** in the initialization operation of the CXL storage **110**.

[0063] The CXL memory **120** may include the CX L memory controller **121** and the buffer memory BFM. The CXL memory controller **121** may include a CXL memory interface circuit **121a**, a processor **121b**, a memory manager **121c**, and a buffer memory interface circuit **121d**.

[0064] The CXL memory interface circuit **121a** may be connected with the CXL switch SW_CXL. The CXL memory interface circuit **121a** may communicate with the host **101** or the CXL storage **110** through the CXL switch SW_CXL.

[0065] The processor **121b** may be configured to control an overall operation of the CXL memory

controller **121**. The memory manager **121c** may be configured to manage the buffer memory BFM. For example, the memory manager **121c** may be configured to translate a memory address (e.g., a logical address or a virtual address) from the host **101** or the CXL storage **110** into a physical address for the buffer memory BFM. In an embodiment, the memory address that is an address for managing a storage area of the CXL memory **120** may be a logical address or a virtual address that is designated and managed by the host **101**.

[0066] The buffer memory interface circuit **121d** may control the buffer memory BFM such that data are stored in the buffer memory BFM or data are read from the buffer memory BFM. In an embodiment, the buffer memory interface circuit **121d** may be implemented to comply with the standard protocol such as a DDR interface or an LPDDR interface.

[0067] Under control of the CXL memory controller **121**, the buffer memory BFM may store data or may output the stored data. In an embodiment, the buffer memory BFM may be configured to store the map data MD that are used in the CXL storage **110**. The map data MD may be transferred from the CXL storage **110** to the CXL memory **120** when the computing system **100** is initialized or the CXL storage **110** is initialized.

[0068] As described above, the CXL storage **110** according to an embodiment of the present disclosure may store the map data MD, which are necessary to manage the nonvolatile memory NVM, in the CXL memory **120** connected through the CXL switch SW_CXL (or the CXL interface IF_CXL). Afterwards, when the CXL storage **110** performs the read operation depending on a request of the host **101**, the CXL storage **110** may read at least a portion of the map data MD from the CXL memory **120** through the CXL switch SW_CXL (or the CXL interface IF_CXL) and may perform the read operation based on the map data MD thus read. Alternatively, when the CXL storage **110** performs the write operation depending on a request of the host **101**, the CXL storage **110** may perform the write operation on the nonvolatile memory NVM and may update the map data MD. In this case, the updated map data MD may be first stored in the RAM **111c** of the CXL storage controller **111**, and the map data MD stored in the RAM **111c** may be transferred to the buffer memory BFM of the CXL memory **120** through the CXL switch SW_CXL (or the CXL interface IF_CXL), so as to be updated in the buffer memory BFM.

[0069] In an embodiment, at least a partial area of the buffer memory BFM of the CXL memory **120** may be allocated for a dedicated area for the CXL storage **110**, and the remaining area may be used as an area that is capable of being accessed by the host **101**.

[0070] In an embodiment, the host **101** and the CXL storage **110** may communicate with each other by using CXL.io being an input/output protocol. The CXL.io may have a PCIe-based non-coherency input/output protocol. The host **101** and the CXL storage **110** may exchange user data or variety of information with each other by using the CXL.io.

[0071] In an embodiment, the CXL storage **110** and the CXL memory **120** may communicate with each other by using CXL.mem being a memory access protocol. The CXL.mem may be a memory access protocol that supports memory access. The CXL storage **110** may access a partial area (e.g., an area where the map data MD are stored or a CXL storage-dedicated area) of the CXL memory **120** by using the CXL.mem.

[0072] In an embodiment, the host **101** and the CXL memory **120** may communicate with each other by using the CXL.mem being a memory access protocol. The host **101** may access, as a system memory, the remaining area (e.g., the remaining area other than the area where the map data MD are stored or the remaining area other than the CXL storage-dedicated area) of the CXL memory **120** by using the CXL.mem.

[0073] The above access types including CXL.io and CXL.mem are provided as an example, and the present disclosure is not limited thereto.

[0074] In an embodiment, the CXL storage **110** and the CXL memory **120** may be installed in a CXL interface-based physical port (e.g., a PCIe physical port). In an embodiment, the CXL storage **110** and the CXL memory **120** may be implemented based on the E1.S, E1.L, E3.S, E3.L, or PCIe

AIC (CEM) form factor. Alternatively, the CXL storage **110** and the CXL memory **120** may be implemented based on the U.2 form factor, the M.2 form factor, various different types of PCIe-based form factors, or various different types of small form factors. As will be described with reference to FIG. **14**, the CXL storage **110** and the CXL memory **120** may be implemented with various types of form factors, and may support a function of a hot-plug capable of being installed in or removed from the physical port.

[0075] FIG. **4** is a flowchart illustrating an initialization operation or a power-up operation of a computing system of FIG. **3**. Referring to FIGS. **3** and **4**, in operation PUP-S10, the computing system **100** may be powered up. When the computing system **100** is powered up, the host **101** may send information about power-up or initialization start to the CXL storage **110**, the CXL switch SW_CXL, and the CXL memory **120**. In response to the information about power-up or - initialization start, each of the CXL storage **110**, the CXL switch SW_CXL, and the CXL memory **120** may perform an individual initialization operation.

[0076] In operation PUP-S11, the CXL storage **110** may check a storage capacity (i.e., a capacity of the nonvolatile memory NVM). For example, the CXL storage **110** may check the storage capacity of the nonvolatile memory NVM in response to the information about power-up or initialization start.

[0077] In operation PUP-S12, the CXL memory **120** may check a memory capacity (i.e., a capacity of the buffer memory BFM). For example, the CXL memory **120** may check the capacity of the buffer memory BFM in response to the information about power-up or initialization start.

[0078] The host **101** may recognize information of the CXL storage **110** through operation PUP-S21 and operation PUP-S22. For example, in operation PUP-S21, the host **101** may issue a first device information request REQ_id1 for recognizing device information of the CXL storage **110** through the CXL host interface circuit **101a**. The first device information request REQ_id1 may be transferred to the CXL switch SW_CXL. The CXL switch SW_CXL may transfer the first device information request REQ_id1 to the CXL storage **110** targeted for the first device information request REQ_id1.

[0079] In operation PUP-S22, the CXL storage **110** may output a first device information response REP_id1 through the CXL storage interface circuit **111a** in response to the first device information request REQ_id1 received from the CXL switch SW_CXL. The first device information response REP_id1 may be transferred to the CXL switch SW_CXL. The CXL switch SW_CXL may transfer the first device information response REP_id1 to the host **101** targeted for the first device information response REP_id1.

[0080] The host **101** may identify the device information of the CXL storage **110** in response to the first device information response REP_id1 received from the CXL switch SW_CXL. In an embodiment, the first device information response REP_id1 may include information about a device type and a storage capacity of the CXL storage **110**.

[0081] The host **101** may recognize information of the CXL memory **120** through operation PUP-S31 and operation PUP-S32. For example, in operation PUP-S31, the host **101** may issue a second device information request REQ_id2 for recognizing device information of the CXL memory **120** through the CXL host interface circuit **101a**. The second device information request REQ_id2 may be transferred to the CXL switch SW_CXL. The CXL switch SW_CXL may transfer the second device information request REQ_id2 to the CXL memory **120** targeted for the second device information request REQ_id2.

[0082] In operation PUP-S32, the CXL memory **120** may output a second device information response REP_id2 through the CXL memory interface circuit **121a** in response to the second device information request REQ_id2 received from the CXL switch SW_CXL. The second device information response REP_id2 may be transferred to the CXL switch SW_CXL. The CXL switch SW_CXL may transfer the second device information response REP_id2 to the host **101** targeted for the second device information response REP_id2.

[0083] The host **101** may identify the device information of the CXL memory **120** in response to the second device information response REP_id2 received from the CXL switch SW_CXL. In an embodiment, the second device information response REP_id2 may include information about a device type and a storage capacity of the CXL memory **120**.

[0084] As described above, the host **101** may identify the information about the device types (e.g., a storage type and a memory type) and capacities of the CXL storage **110** and the CXL memory **120** through operation PUP-S21 to operation PUP-S32.

[0085] The host **101** may allocate at least a partial area of the CXL memory **120** for an area dedicated for the CXL storage **110** through operation PUP-S41 to operation PUP-S46. For example, in operation PUP-S41, the CXL storage **110** may output a memory allocation request REQ_mem_alc through the CXL storage interface circuit **111a**. The memory allocation request REQ_mem_alc may be transferred to the CXL switch SW_CXL. The CXL switch SW_CXL may transfer the memory allocation request REQ_mem_alc to the host **101**. In an embodiment, the memory allocation request REQ_mem_alc may refer to an allocation request for an area, which is to be used as a dedicated area of the CXL storage **110**, from among areas of the CXL memory **120**.

[0086] In operation PUP-S42, the host **101** may allocate at least a partial area of the CXL memory **120** for the dedicated area of the CXL storage **110** in response to the memory allocation request REQ_mem_alc. For example, the host **101** may determine a buffer capacity required by the CXL storage **110** based on the storage capacity of the CXL storage **110**. The host **101** may allocate the area of the CXL memory **120**, which corresponds to the determined buffer capacity, for the dedicated area of the CXL storage **110**.

[0087] In operation PUP-S43, the host **101** may output a memory allocation response REP_mem_alc through the CXL host interface circuit **101a**. The memory allocation response REP_mem_alc may be transferred to the CXL switch SW_CXL. The CXL switch SW_CXL may transfer the memory allocation response REP_mem_alc to the CXL storage **110** targeted for the memory allocation response REP_mem_alc. In an embodiment, the memory allocation response REP_mem_alc may include information about a device identifier of the CXL memory **120** and a memory address (e.g., a logical address range or a virtual address range) of an area of the CXL memory **120**, which is allocated for a dedicated area of the CXL memory **120**.

[0088] The CXL storage **110** may identify the area of the CXL memory **120**, which is dedicated for the CXL storage **110**, based on the memory allocation response REP_mem_alc.

[0089] In operation PUP-S44, the CXL storage **110** may output a write request REQ_WR through the CXL storage interface circuit **111a**. The write request REQ_WR may be transferred to the CXL switch SW_CXL. The CXL switch SW_CXL may transfer the write request REQ_WR to the CXL memory **120** targeted for the write request REQ_WR. The CXL memory **120** may perform the write operation in response to the write request REQ_WR.

[0090] In operation PUP-S45, the CXL memory **120** may output, through the CXL memory interface circuit **121a**, a write response REP_WR providing notification that the write request is completed. The CXL switch SW_CXL may transfer the write response REP_WR to the CXL storage **110** targeted for the write response REP_WR. The CXL storage **110** may recognize that the write operation is completely performed on the CXL memory **120**, in response to the write response REP_WR.

[0091] In an embodiment, the write request REQ_WR may refer to a request for storing the map data MD present in the nonvolatile memory NVM of the CXL storage **110** in the dedicated area of the CXL memory **120**. That is, the write request REQ_WR may include address information about the map data MD and the dedicated area. Through operation PUP-S44 and operation PUP-S45, the map data MD present in the CXL storage **110** may be stored in the dedicated area of the CXL memory **120**.

[0092] In operation PUP-S46, the CXL storage **110** may output acknowledge information ACK_md through the CXL storage interface circuit **111a**. The CXL switch SW_CXL may transfer the

acknowledge information ACK_md to the host **101**. In response to the acknowledge information ACK_md, the host **101** may recognize that the CXL storage **110** stores the map data MD in the CXL memory **120**. Afterwards, the host **101**, the CXL storage **110**, and the CXL memory **120** may perform a normal operation (e.g., a read operation or a write operation).

[0093] FIG. 5 is a diagram for describing an operation in which a computing system stores map data. For convenience of description and for brevity of drawing, components of the host **101**, the CXL storage **110**, and the CXL memory **120** are conceptually illustrated, and some unnecessary components are omitted.

[0094] Referring to FIGS. 3 to 5, the host **101** may allocate a partial area of the CXL memory **120** for a dedicated area of the CXL storage **110**. In this case, the dedicated area of the CXL memory **120** may be accessed by the CXL storage **110** and may be used to store map data of the CXL storage **110**.

[0095] For example, as illustrated in FIG. 5, the nonvolatile memory NVM of the CXL storage **110** may store the user data UD and the map data MD. As described above, because the CXL storage **110** does not include a separate buffer memory, the CXL storage **110** may require a buffer area in which the map data MD are to be stored. According to an embodiment of the present disclosure, the map data MD of the CXL storage **110** may be stored in a partial area (e.g., a dedicated area allocated by the host **101**) of the CXL memory **120**. In this case, the dedicated area of the CXL memory **120** may be accessed by the CXL storage **110** through the CXL switch SW_CXL.

[0096] In an embodiment, the remaining area of the CXL memory **120**, which is not allocated, other than the dedicated area may be an area that is accessible by the host **101** or is managed by the host **101**. In this case, the host **101** may access the remaining area of the CXL memory **120** through the CXL switch SW_CXL. In an embodiment, the remaining area of the CXL memory **120**, which is not allocated for the dedicated area, may be used as a memory expander.

[0097] As described above, depending on the request of the CXL storage **110**, the host **101** may allocate at least a partial area of the CXL memory **120** for the dedicated area of the CXL storage **110**. In this case, the CXL storage **110** may access a portion of the CXL memory **120**, which is allocated for the dedicated area, and the host **101** may access the remaining area of the CXL memory **120** (i.e., the remaining area other than the dedicated area thus allocated). In an embodiment, both the access of the CXL storage **110** to the CXL memory **120** and the access of the host **101** to the CXL memory **120** may be performed through the same interface (e.g., a CXL interface or a CXL switch).

[0098] In an embodiment, when the system memory of the host **101** is insufficient, the host **101** may retrieve (or recover) the dedicated area of the CXL memory **120**, which allocated to the CXL storage **110** (i.e., an area for storing map data). The retrieved (or recovered) dedicated area may be used as the system memory by the host **101**.

[0099] FIGS. 6 and 7 are diagrams for describing an operation in which map data are stored in a CXL memory. In an embodiment, the map data MD present in the CXL storage **110** may be transferred and stored to the CXL memory **120** from the CXL storage **110** through various manners.

[0100] As an example, the CXL storage **110** and the CXL memory **120** may exchange the map data MD based on a peer-to-peer (P2P) manner. For example, as illustrated in FIG. 6, the CXL storage controller **111** of the CXL storage **110** may include a direct memory access (DMA) engine. The DMA engine included in the CXL storage controller **111** may transfer the map data MD present in the nonvolatile memory NVM to the CXL memory **120** without the interference or control of the host **101**. That is, the map data MD may be transferred from the CXL storage **110** to the CXL memory **120** based on the P2P manner.

[0101] As an example, under control of the host **101**, the CXL storage **110** and the CXL memory **120** may exchange the map data MD based on the DMA manner. For example, as illustrated in FIG. 7, the host **101** may include a direct memory access (DMA) engine. The DMA engine of the host **101** may read the map data MD from the CXL storage **110** and may transfer the map data MD

thus read to the CXL memory **120**. In an embodiment, the DM A engine of the host **101** may read the map data MD from the CXL storage **110** based on the CXL.io and may transfer the map data MD to the CXL memory **120** based on the CXL.mem.

[0102] The above manners in which map data are transferred from the CXL storage **110** to the CXL memory **120** are provided as an example, and the present disclosure is not limited thereto. It may be understood that the transfer of map data from the CXL storage **110** to the CXL memory **120** is implemented in various manners using the CXL interface or the CXL switch. In an embodiment, the transfer (i.e., the backup or flush) of map data from the CXL memory **120** to the CXL storage **110** may also be implemented in a manner(s) similar to the above manners.

[0103] FIG. **8** is a flowchart illustrating a read operation for CXL storage of FIG. **2**. In an embodiment, the read operation for the CXL storage **110** according to the flowchart of FIG. **8** may be performed after the initialization operation of FIG. **4** is performed (i.e., after the map data MD of the CXL storage **110** are stored in the dedicated area of the CXL memory **120**).

[0104] Referring to FIGS. **2**, **3**, and **8**, in operation RD-S10, the host **101** may output a first read request REQ_RD1 through the CXL host interface circuit **101a**. The CXL switch SW_CXL may transfer the first read request REQ_RD1 to the CXL storage **110** targeted for the first read request REQ_RD1. In an embodiment, the first read request REQ_RD1 may refer to a request for reading first user data UD1 stored in the CXL storage **110** and may include a first logical block address LBA1 corresponding to the first user data UD1.

[0105] In operation RD-S21, the CXL storage **110** may output a second read request REQ_RD2 through the CXL storage interface circuit **111a** in response to the first read request REQ_RD1. The CXL switch SW_CXL may transfer the second read request REQ_RD2 to the CXL memory **120**. In an embodiment, the second read request REQ_RD2 may refer to a request for reading first map data MD1 corresponding to the first logical block address LBA1. That is, the second read request REQ_RD2 may refer to a request for reading the first map data MD1 from the CXL memory **120**. The second read request REQ_RD2 may include information about a memory address (e.g., a logical address or a virtual address) of the CXL memory **120**, which indicates an area where the first map data MD1 are stored.

[0106] In operation RD-S22, the CXL memory **120** may read the first map data MD1 in response to the second read request REQ_RD2. For example, the CXL memory controller **121** of the CXL memory **120** may read the first map data MD1 from an area corresponding to a memory address (e.g., a logical address or a virtual address) included in the second read request REQ_RD2. In an embodiment, the CXL memory controller **121** may read the first map data MD1 from the buffer memory BFM by using the buffer memory interface circuit **121d**.

[0107] In an embodiment, the first map data MD1 read in operation RD-S22 may be a portion of the entire map data MD and may be map data corresponding to the first logical block address LBA1. That is, the first map data MD1 may include information about a first physical block address PBA1 corresponding to the first logical block address LBA1.

[0108] In operation RD-S23, the CXL memory **120** may output a second read response REP_RD2 including the first map data MD1 through the CXL memory interface circuit **121a**. The CXL switch SW_CXL may transfer the second read response REP_RD2 to the CXL storage **110**. In an embodiment, the first map data MD1 included in the second read response REP_RD2 received through the CXL switch SW_CXL may be stored or temporarily stored in the RAM **111c** of the CXL storage controller **111**.

[0109] In an embodiment, when the first map data MD1 corresponding to the first logical block address LBA1 is already present in the RAM **111c** of the CXL storage controller **111**, operation RD-S21 to operation RD-S23 (i.e., operations for loading the first map data MD1 from the CXL memory **120**) may be omitted.

[0110] In operation RD-S31, the CXL storage **110** may search for the first physical block address PBA1 corresponding the first logical block address LBA1 based on the first map data MD1. For

example, the FTL **111d** of the CXL storage controller **111** may search for the first physical block address PBA1 corresponding to the first logical block address LBA1 based on the first map data MD1. Based on the search, the CXL storage **110** finds the first physical block address PBA1. [0111] In operation RD-S32, the CXL storage **110** may read the first user data UD1 present in an area corresponding to the first physical block address PBA1 from the nonvolatile memory NVM. For example, the CXL storage controller **111** may read the first user data UD1 from the area of the nonvolatile memory NVM, which corresponds to the first physical block address PBA1. In an embodiment, the CXL storage controller **111** may read the first user data UD1 from the nonvolatile memory NVM by using the NAND interface circuit **111f**.

[0112] In operation RD-S33, the CXL storage **110** may output a first read response REP_RD1 to the first read request REQ_RD1 through the CXL storage interface circuit **111a**. The CXL switch SW_CXL may transfer the first read response REP_RD1 to the host **101**. In an embodiment, the first read response REP_RD1 may include the first user data UD1 requested through the first read request REQ_RD1. The host **101** may obtain the first user data UD1 through the first read response REP_RD1.

[0113] In an embodiment, operation RD-S10 to operation RD-S33 corresponding to the communications between the host **101** and the CXL storage **110** may be performed based on the CXL.io, and operation RD-S21 to operation RD-S23 corresponding to the communications between the CXL storage **110** and the CXL memory **120** may be performed based on the CXL.mem. However, the present disclosure is not limited thereto. For example, the communications between the host **101**, the CXL storage **110**, and the CXL memory **120** may be performed through the CXL switch SW_CXL (i.e., a common interface, a common link, or a common switch).

[0114] FIG. **9** is a flowchart illustrating a write operation for CXL storage of FIG. **2**. In an embodiment, the write operation for the CXL storage **110** according to the flowchart of FIG. **9** may be performed after the initialization operation of FIG. **4** is performed (i.e., after the map data MD of the CXL storage **110** are stored in the dedicated area of the CXL memory **120**).

[0115] Referring to FIGS. **2**, **3**, and **9**, in operation WR-S10, the host **101** may output a first write request REQ_WR1 through the CXL host interface circuit **101a**. The CXL switch SW_CXL may transfer the first write request REQ_WR1 to the CXL storage **110**. In an embodiment, the first write request REQ_WR1 may refer to a request for writing the first user data UD1 in the CXL storage **110**.

[0116] In operation WR-S21, the CXL storage **110** may determine a memory block in which the first user data UD1 are to be written, in response to the first write request REQ_WR1. For example, the FTL **111d** of the CXL storage controller **111** may manage block information about a memory block, which is free, capable of being written, or capable of being allocated, from among memory blocks included in the nonvolatile memory NVM. The FTL **111d** may select a memory block, in which the first user data UD1 are to be written, based on the block information.

[0117] In operation WR-S22, the CXL storage **110** may write the first user data UD1 in the selected memory block. For example, the CXL storage controller **111** may control the nonvolatile memory NVM such that the first user data UD1 are written in the selected memory block. In an embodiment, the CXL storage controller **111** may write the first user data UD1 in the nonvolatile memory NVM by using the NAND interface circuit **111f**.

[0118] When the first user data UD1 are completely written in the nonvolatile memory NVM (i.e., when a program operation for the nonvolatile memory NVM is passed), in operation WR-S23, the CXL storage **110** may update the first map data MD1 or may generate the first map data MD1. For example, the first map data MD1 may include information indicating that the first user data UD1 corresponding to the first logical block address LBA1 are stored in an area of the nonvolatile memory NVM, which corresponds to the first physical block address PBA1. That is, the CXL storage **110** may generate the first map data MD1 indicating that the first user data UD1

corresponding to the first logical block address LBA1 are stored in the area corresponding to the first physical block address PBA1.

[0119] In operation WR-S24, the CXL storage **110** may output a first write response REP_WR1 to the first write request REQ_WR1 through the CXL storage interface circuit **111a**. The CXL switch SW_CXL may transfer the first write response REP_WR1 to the host **101**. In response to the first write response REP_WR1, the host **101** may determine that the first user data UD1 corresponding to the first write request REQ_WR1 are normally stored in the CXL storage **110**.

[0120] After the write operation for the CXL storage **110** requested by the host **101** is completed, the CXL storage **110** may perform a map data update operation. For example, in operation WR-S31, the CXL storage **110** may output a second write request REQ_WR2 through the CXL storage interface circuit **111a**. The CXL switch SW_CXL may transfer the second write request REQ_WR2 to the CXL memory **120**.

[0121] In an embodiment, the second write request REQ_WR2 may refer to a request for writing the first map data MD1, which are updated or generated as the first user data UD1 are stored, in the CXL memory **120**. The second write request REQ_WR2 may include a memory address at which the first map data MD1 are to be stored. The memory address included in the second write request REQ_WR2 may indicate the area of the CXL memory **120**, which is dedicated for the CXL storage **110**.

[0122] In operation WR-S32, the CXL memory **120** may store the first map data MD1 in the corresponding area in response to the second write request REQ_WR2. For example, the CXL memory **120** may write the first map data MD1 in the area corresponding to the memory address included in the second write request REQ_WR2.

[0123] In operation WR-S33, the CXL memory **120** may output a second write response REP_WR2 to the second write request REQ_WR2 through the CXL memory interface circuit **121a**. The CXL switch SW_CXL may transfer the second write response REP_WR2 to the CXL storage **110**.

[0124] In an embodiment, operation WR-S31 to operation WR-S33 (i.e., an operation of storing map data in the CXL memory **120** or an operation of updating map data) may be performed whenever the write operation for the CXL storage **110** is completed. Alternatively, operation WR-S31 to operation WR-S33 may be performed when the size of map data updated or newly generated reaches a given value. Alternatively, operation WR-S31 to operation WR-S33 may be performed periodically. However, the present disclosure is not limited thereto. For example, map data that are generated or updated during the operation of the CXL storage **110** may be stored in the CXL memory **120** depending on various operation policies.

[0125] FIG. **10** is a flowchart illustrating a power-off operation of a computing system of FIG. **2**. In an embodiment, a power-off operation of a computing system will be described with reference to FIG. **10**, but the present disclosure is not limited thereto. For example, it may be understood that the operating method of FIG. **10** is applicable to the power-off operation or reset operation of each of various components (e.g., a host, CXL storage, a CXL memory, and a CXL switch) included in the computing system.

[0126] Referring to FIGS. **2** and **10**, in operation POF-S10, the host **101** may output power-off information IFM_off through the CXL host interface circuit **101a**. The CXL switch SW_CXL may transfer the power-off information IFM_off to the CXL storage **110**. For example, the host **101** may recognize or detect information about power-off of the computing system **100**. The host **101** may send the power-off information IFM_off to the CXL storage **110** through the CXL switch SW_CXL such that the CXL storage **110** performs a power-off operation.

[0127] In operation POF-S21, the CXL storage **110** may output a read request REQ_RD through the CXL storage interface circuit **111a** in response to the power-off information IFM_off. The CXL switch SW_CXL may transfer the read request REQ_RD to the CXL memory **120**. In an embodiment, the read request REQ_RD in operation POF-S21 may refer to a request for reading

the entire map data MD stored in the CXL memory **120**. The read request REQ_RD may include a memory address of an area where the map data MD are stored.

[0128] In operation POF-S22, the CXL memory **120** may read the map data MD in response to the read request REQ_RD. For example, the CXL memory **120** may read the map data MD from the buffer memory BFM based on the memory address included in the read request REQ_RD.

[0129] In operation POF-S23, the CXL memory **120** may output a read response REP_RD to the read request REQ_RD through the CXL memory interface circuit **121a**. The CXL switch SW_CXL may transfer the read response REP_RD to the CXL storage **110**.

[0130] In operation POF-S24, the CXL storage **110** may write the map data MD included in the read response REP_RD in the nonvolatile memory NVM. In an embodiment, the CXL storage **110** may store the map data MD in a given area of the nonvolatile memory NV M.

[0131] After the entire map data MD associated with the CXL storage **110** are stored in the nonvolatile memory NVM, in operation POF-S31, the CXL storage **110** may output a response ACK_off to the power-off information IFM_off. The CXL switch SW_CXL may send the response ACK_off to the host **101**. The host **101** may recognize that the map data MD present in the CXL memory **120** are normally stored in the CXL storage **110**, based on the response ACK_off.

[0132] Afterwards, in operation POF-S32, the host **101**, the CXL storage **110**, the CXL memory **120**, and the CXL switch SW_CXL may be powered off. For example, the host **101** may interrupt a power that is provided to the CXL storage **110**, the CXL memory **120**, and the CXL switch SW_CXL.

[0133] The power-off operation described with reference to FIG. **10** is provided as an example, and the present disclosure is not limited thereto. For example, in the embodiment of FIG. **10**, after the CXL storage **110** stores the map data MD present in the CXL memory **120** in the nonvolatile memory NVM, the CXL storage **110** may provide notification that the map data MD are completely backed up, by sending acknowledge ACK_off to the host **101** (i.e., an interrupt manner). Alternatively, the CXL storage **110** may set a value of a specific register to a given value after storing the map data MD present in the CXL memory **120** in the nonvolatile memory NVM. The host **101** may check whether the map data MD are completed backed up, by periodically checking the value of the specific register of the CXL storage **110** (a polling manner). Alternatively, the CXL storage **110** may be configured to complete a backup operation for the map data MD within a given time from a point in time when a power-off information INF_off is received from the host **101** (a time-out manner). As described above, the CXL storage **110** may transfer information about backup completion of the map data MD to the host **101** through at least one of various manners.

[0134] In an embodiment, the power-off operation may be changed depending on an operation manner of the CXL storage **110**. For example, when the CXL storage **110** performs the write operation, the CXL storage **110** may perform the program operation on the nonvolatile memory NVM and thus may update the map data MD.

[0135] In an embodiment, the operation of updating the map data MD may be performed only on the CXL memory **120**. In this case, the map data MD stored in the CXL memory **120** may be up-to-date information, and the map data MD stored in the nonvolatile memory NVM may not be up-to-date information. That is, when the map data update operation is performed only on the CXL memory **120**, the up-to-date map data MD may be maintained only in the CXL memory **120**, and thus, an operation of flushing, backing up, or dumping the map data MD from the CXL memory **120** is required when the CXL storage **110**, the CXL memory **120**, or the computing system **100** is powered off. In an embodiment, the map data update operation may be first performed with respect to the map data MD stored in the nonvolatile memory NVM and may then be performed with respect to the map data MD stored in the CXL memory **120** through a background operation. In this case, because the map data MD stored in the nonvolatile memory NVM are guaranteed to be up-to-date, the operation of flushing, dumping, or backing up the map data MD from the CXL memory

120 may not be required when the CXL storage **110**, the CXL memory **120**, or the computing system **100** is powered off.

[0136] In an embodiment, the map data update operation may be first performed with respect to the map data MD stored in the CXL memory **120** and may then be performed with respect to the map data MD stored in the nonvolatile memory NVM through the background operation. In this case, the map data MD stored in the CXL memory **120** may be up-to-date information, and the map data MD stored in the nonvolatile memory NVM may not be up-to-date information. As such, when the CXL storage **110**, the CXL memory **120**, or the computing system **100** is powered off, at least a portion of the map data MD of the CXL memory **120** has to be backed up to the nonvolatile memory NVM of the CXL storage **110**. In an embodiment, at least a portion of the map data MD to be backed up to the nonvolatile memory NVM may be the up-to-date map data MD that are not stored in the nonvolatile memory NVM. In an embodiment, the CXL storage **110** may manage or store flag information or table information indicating that the map data MD stored in the nonvolatile memory NVM are up-to-date information.

[0137] As described above, when the CXL storage **110**, the CXL memory **120**, or the computing system **100** is powered off, depending on a way to manage the map data MD (i.e., depending on a place where up-to-date information is managed), the map data MD may be selectively flushed, backed up, or dumped to the CXL storage **110** from the CXL memory **120**.

[0138] FIG. **11** is a block diagram illustrating a computing system according to an embodiment of the present disclosure. Below, for convenience of description, additional description associated with the components described above will be omitted to avoid redundancy. Referring to FIG. **11**, a computing system **200** may include a host **201**, a plurality of memory devices **202a** and **202b**, the CXL switch SW_CXL, CXL storage **210**, and a plurality of CXL memories **220_1** to **220_n**.

[0139] The host **201** may be directly connected with the plurality of memory devices **202a** and **202b**. The host **201**, the CXL storage **210**, and the plurality of CXL memories **220_1** to **220_n** may be connected with the CXL switch SW_CXL and may communicate with each other through the CXL switch SW_CXL.

[0140] In an embodiment, each of the plurality of CXL memories **220_1** to **220_n** may have a structure similar to that of the CXL memory **120** described with reference to FIGS. **2** to **10**. That is, each of the plurality of CXL memories **220_1** to **220_n** may be implemented with an individual memory device or memory module and may be connected with the CXL switch SW_CXL through different physical ports. That is, as the plurality of CXL memories **220_1** to **220_n** are connected with the CXL switch SW_CXL, a memory area (or capacity) that is managed by the host **201** may increase.

[0141] In an embodiment, the host **201** may manage the plurality of CXL memories **220_1** to **220_n** as one memory cluster. In an embodiment, the host **201** may allocate at least some of the plurality of CXL memories **220_1** to **220_n** for a memory dedicated for the CXL storage **210**. Alternatively, the host **201** may allocate at least a partial area of each of the plurality of CXL memories **220_1** to **220_n** for a memory dedicated for the CXL storage **210**.

[0142] FIG. **12** is a block diagram illustrating a computing system according to an embodiment of the present disclosure. Below, for convenience of description, additional description associated with the components described above will be omitted to avoid redundancy. Referring to FIG. **12**, a computing system **300** may include a host **301**, a plurality of memory devices **302a** and **302b**, the CXL switch SW_CXL, a plurality of CXL storages **310_1** to **310_m**, and a CXL memory **320**.

[0143] The host **301** may be directly connected with the plurality of memory devices **302a** and **302b**. The host **301**, the plurality of CXL storages **310_1** to **310_m**, and the CXL memory **320** may be connected with the CXL switch SW_CXL and may communicate with each other through the CXL switch SW_CXL.

[0144] In an embodiment, each of the plurality of CXL storages **310_1** to **310_m** may have a structure similar to that of the CXL storage **110** described with reference to FIGS. **2** to **10**. That is,

each of the plurality of CXL storages **310_1** to **310_m** may be implemented with an individual storage device or storage module and may be connected with the CXL switch SW_CXL through different physical ports. That is, as the plurality of CXL storages **310_1** to **310_m** are connected with the CXL switch SW_CXL, a storage area (or capacity) that is available by the host **301** may increase.

[0145] In an embodiment, at least a partial area of the CXL memory **320** may be allocated for an area dedicated for the plurality of CXL storages **310_1** to **310_m**. For example, the host **301** may manage the plurality of CXL storages **310_1** to **310_m** as one storage cluster and may allocate a partial area of the CXL memory **320** for a dedicated area of one storage cluster. Alternatively, the host **201** may allocate partial areas of the CXL memory **320** for dedicated areas of the respective CXL storages **310_1** to **310_m**.

[0146] FIG. **13** is a block diagram illustrating a computing system according to an embodiment of the present disclosure. Below, for convenience of description, additional description associated with the components described above will be omitted to avoid redundancy. Referring to FIG. **13**, a computing system **400** may include a host **401**, a plurality of memory devices **402a** and **402b**, the CXL switch SW_CXL, a plurality of CXL storages **410_1** to **410_m**, and a plurality of CXL memories **420_1** to **420_n**.

[0147] The host **401** may be directly connected with the plurality of memory devices **402a** and **402b**. The host **401**, the plurality of CXL storages **410_1** to **410_m**, and the plurality of CXL memories **420_1** to **420_n** may be connected with the CXL switch SW_CXL and may communicate with each other through the CXL switch SW_CXL.

[0148] In an embodiment, the host **401** may manage the plurality of CXL storages **410_1** to **410_m** as one storage cluster, and may manage the plurality of CXL memories **420_1** to **420_n** as one memory cluster. The host **401** may allocate a partial area of the memory cluster for a dedicated area (i.e., an area for storing map data of the storage cluster) of the storage cluster. Alternatively, the host **201** may allocate areas of the CXL memories **420_1** to **420_n** for dedicated areas of the respective CXL storages **410_1** to **410_m**.

[0149] FIG. **14** is a block diagram illustrating a computing system according to an embodiment of the present disclosure. Below, for convenience of description, additional description associated with the components described above will be omitted to avoid redundancy. Referring to FIG. **14**, a computing system **500** may include a host **501**, a plurality of memory devices **502a** and **502b**, the CXL switch SW_CXL, a plurality of CXL storages **510_1**, **510_2**, and **510_3**, and a plurality of CXL memories **520_1**, **520_2**, and **502_3**.

[0150] The host **501** may be directly connected with the plurality of memory devices **502a** and **502b**. The host **501**, the plurality of CXL storages **510_1** and **510_2**, and the plurality of CXL memories **520_1** and **520_2** may be connected with the CXL switch SW_CXL and may communicate with each other through the CXL switch SW_CXL. As in the above description, a partial area of the CXL memories **520_1** and **520_2** may be allocated for a dedicated area of the CXL storages **510_1** and **510_2**.

[0151] In an embodiment, while the computing system **500** is being driven, some of the CXL storages **510_1** and **510_2** or some of the CXL memories **520_1** and **520_2** may be hot-removed from the CXL switch SW_CXL. Alternatively, while the computing system **500** is being driven, the CXL storage **510_3** or the CXL memory **520_3** may be hot-added to the CXL switch SW_CXL. In this case, the host **501** may again perform memory allocation by again performing the initialization operation on devices connected with the CXL switch SW_CXL through the reset operation or the hot-plug operation. That is, CXL storage and a CXL memory according to an embodiment of the present disclosure may support the hot-plug function, and it may be possible to expand a storage capacity and a memory capacity of a computing system through various connections.

[0152] FIG. **15** is a block diagram illustrating a computing system according to an embodiment of the present disclosure. For convenience of description, additional description associated with the

components described above will be omitted to avoid redundancy. Referring to FIG. 15, a computing system **1000** may include a first CPU **1110**, a second CPU **1120**, a GPU **1130**, an NPU **1140**, the CXL switch SW_CXL, CXL storage **1210**, a CXL memory **1220**, a PCIe device **1310**, and an accelerator (CXL device) **1320**.

[0153] The first CPU **1110**, the second CPU **1120**, the GPU **1130**, the NPU **1140**, the CXL storage **1210**, the CXL memory **1220**, the PCIe device **1310**, and the accelerator (CXL device) **1320** may be connected in common with the CXL switch SW_CXL and may communicate with each other through the CXL switch SW_CXL.

[0154] In an embodiment, each of the first CPU **1110**, the second CPU **1120**, the GPU **1130**, and the NPU **1140** may be the host described with reference to FIGS. 1 to 14 and may be directly connected with individual memory devices.

[0155] In an embodiment, the CXL storage **1210** and the CXL memory **1220** may be the CXL storage and the CXL memory described with reference to FIGS. 2 to 14, and at least a partial area of the CXL memory **1220** may be allocated for an area dedicated for the CXL storage **1210** by one or more of the first CPU **1110**, the second CPU **1120**, the GPU **1130**, and the NPU **1140**. That is, the CXL storage **1210** and the CXL memory **1220** may be used as a storage space STR of the computing system **1000**.

[0156] In an embodiment, the CXL switch SW_CXL may be connected with the PCIe device **1310** or the accelerator **1320** configured to support various functions, and the PCIe device **1310** or the accelerator **1320** may communicate with each of the first CPU **1110**, the second CPU **1120**, the GPU **1130**, and the NPU **1140** through the CXL switch SW_CXL or may access the storage space STR including the CXL storage **1210** and the CXL memory **1220** through the CXL switch SW_CXL.

[0157] In an embodiment, the CXL switch SW_CXL may be connected with an external network or Fabric and may be configured to communicate with an external server through the external network or Fabric.

[0158] FIG. 16 is a block diagram illustrating a data center to which a computing system according to an embodiment of the present disclosure is applied. Referring to FIG. 16, a data center **2000** that is a facility collecting various data and providing services may be referred to as a “data Storage center”. The data center **2000** may be a system for operating a search engine and a database, and may be a computing system used in a business such as a bank or in a government institution. The data center **2000** may include application servers **2110**, **2120**, . . . **21m0** and storage servers **2210** to **22n0**. The number of application servers and the number of storage servers may be variously selected depending on an embodiment, and the number of application servers and the number of storage servers may be different from each other.

[0159] Below, a configuration of the first storage server **2210** will be mainly described. The application servers **2110** to **21m0** may have similar structures, the storage servers **2210** to **22n0** may have similar structures, and the application servers **2110** to **21m0** and the storage servers **2210** to **22n0** may communicate with each other over a network NT.

[0160] The first storage server **2210** may include a processor **2211**, a memory **2212**, a switch **2213**, a storage device **2215**, a CXL memory **2214**, and a network interface card (NIC) **2216**. The processor **2211** may control an overall operation of the first storage server **2210** and may access the memory **2212** to execute an instruction loaded onto the memory **2212** or to process data. The memory **2212** may be implemented with a DDR SDRAM (Double Data Rate Synchronous DRAM), an HBM (High Bandwidth Memory), an HMC (Hybrid Memory Cube), a DIMM (Dual In-line Memory Module), an Optane DIMM, and/or an NVM DIMM (NonVolatile DIMM). The processor **2211** and the memory **2212** may be directly connected, and the numbers of processors and memories included in one storage server **2210** may be variously selected.

[0161] In an embodiment, the processor **2211** and the memory **2212** may provide a processor-memory pair. In an embodiment, the number of processors **2211** and the number of memories **2212**

may be different from each other. The processor **2211** may include a single core processor or a multi-core processor. The detailed description of the storage server **2210** may be similarly applied to the application servers **2110** to **21m0**.

[0162] The switch **2213** may be configured to arbitrate or route the communications between various components included in the first storage server **2210**. In an embodiment, the switch **2213** may be implemented with the CXL switch SW_CXL described with reference to FIGS. **1** to **15**. That is, the switch **2213** may be a switch implemented based on the CXL protocol.

[0163] The CXL memory **2214** may be connected with the switch **2213**. In an embodiment, the CXL memory **2214** may be used as a memory expander for the processor **2211**. Alternatively, as described with reference to FIGS. **1** to **15**, the CXL memory **2214** may be allocated for a dedicated memory or a buffer memory of the storage device **2215**.

[0164] The storage device **2215** may include a CXL interface circuit CXL_IF, a controller CTRL, and a NAND flash NAND. Depending on a request of the processor **2211**, the storage device **2215** may store data or may output the stored data. In an embodiment, the storage device **2215** may be implemented with the CXL storage described with reference to FIGS. **1** to **15**. In an embodiment, as in the description given with reference to FIGS. **1** to **15**, at least a partial area of the CXL memory **2214** may be allocated for a dedicated area of the storage device **2215**, and the dedicated area may be used as a buffer memory (i.e., may be used to store map data in the CXL memory **2214**).

[0165] According to an embodiment, the application servers **2110** to **21m0** may not include the storage device **2215**. The storage server **2210** may include at least one or more storage devices **2215**. The number of storage devices **2215** included in the storage server **2210** may be variously selected depending on an embodiment.

[0166] The NIC **2216** may be connected with the CXL switch SW_CXL. The NIC **2216** may communicate with the remaining storage servers **2220** to **22n0** or the application servers **2210** to **21m0** over the network NT.

[0167] In an embodiment, the NIC **2216** may include a network interface card, a network adapter, etc. The NIC **2216** may be connected with the network NT by a wired interface, a wireless interface, a Bluetooth interface, an optical interface, etc. The NIC **2216** may include an internal memory, a digital signal processor (DSP), a host bus interface, etc. and may be connected with the processor **2211** and/or the switch **2213** through the host bus interface. In an embodiment, the NIC **2216** may be integrated with at least one of the processor **2211**, the switch **2213**, and the storage device **2215**.

[0168] In an embodiment, the network NT may be implemented by using a Fibre channel (FC) or an Ethernet. In this case, the FC may be a medium that is used in high-speed data transmission and may use an optical switch that provides high performance/high availability. Storage servers may be provided as file storage, block storage, or object storage depending on an access manner of the network NT.

[0169] In an embodiment, the network NT may be a storage-dedicated network such as a storage area network (SAN). For example, the SAN may be a FC-SAN that uses a FC network and is implemented depending on a FC protocol (FCP). For another example, the SAN may be an IP-SAN that uses a TCP/IP network and is implemented depending on an iSCSI (SCSI over TCP/IP or Internet SCSI). In an embodiment, the network NT may be a legacy network such as a TCP/IP network. For example, the network NT may be implemented depending on the following protocol: FCOE (FC over Ethernet), NAS (Network Attached Storage), or NV Me-oF (NV Me over Fabrics).

[0170] In an embodiment, at least one of the application servers **2110** to **21m0** may store data, which are store-requested by a user or a client, in one of the storage servers **2210** to **22n0** over the network NT. At least one of the application servers **2110** to **21m0** may obtain data, which are read-requested by the user or the client, from one of the storage servers **2210** to **22n0** over the network NT. For example, at least one of the application servers **2110** to **21m0** may be implemented with a

web server, a database management system (DBM S), etc.

[0171] In an embodiment, at least one of the application servers **2110** to **21m0** may access a memory, a CXL memory, or a storage device included in any other application server over the network NT or may access memories, CXL memories, or storage devices included in the storage servers **2210** to **22n0** over the network NT. As such, at least one of the application servers **2110** to **21m0** may perform various operations on data stored in the remaining application servers and/or storage servers. For example, at least one of the application servers **2110** to **21m0** may execute an instruction for moving or copying data between the remaining application servers and/or storage servers. In this case, the data may be moved from storage devices of storage servers to memories or CXL memories of application servers through memories or CXL memories of the storage servers or directly. The data that are transferred over a network may be data that are encrypted for security or privacy.

[0172] In an embodiment, a CXL memory included in at least one of the application servers **2110** to **21m0** and the storage servers **2210** to **22n0** may be allocated for a dedicated area of a storage device included in at least one of the application servers **2110** to **21m0** and the storage servers **2210** to **22n0**, and the storage device may use the dedicated area thus allocated as a buffer memory (i.e., may store map data in the dedicated area). For example, a CXL memory included in a storage server (e.g., **22n0**) may be allocated to the storage device **2215** included in the storage server **2210**, and the storage device **2215** included in the storage server **2210** may access the CXL memory included in the storage server (e.g., **22n0**) over the switch **2213** and the NIC **2216**. In this case, the map data associated with the storage device **2215** of the first storage server **2210** may be stored in the CXL memory of the storage server **22n0**. That is, storage devices and CXL memories of a data center according to the present disclosure may be connected and implemented in various manners.

[0173] According to the present disclosure, map data of a storage device may be stored in a memory device separated from the storage device. Accordingly, a large amount of map data may be used without costs for separate research and development.

[0174] While the present disclosure has been described with reference to embodiments thereof, it will be apparent to those of ordinary skill in the art that various changes and modifications may be made thereto without departing from the spirit and scope of the present disclosure as set forth in the following claims.

Claims

1. A storage device comprising: a nonvolatile memory configured to store user data; a buffer memory configured to temporarily store at least a portion of a mapping table stored in an external memory device; and a storage controller configured to control the nonvolatile memory and the buffer memory, wherein the storage controller is further configured to update the user data in the nonvolatile memory under a control of a host and update a first portion of the mapping table stored in the external memory device, wherein the mapping table indicates a relationship between a logical address of the user data managed by the host and a physical address of the nonvolatile memory at which the user data are stored, and wherein the storage controller is further configured to receive power-off information from the host, read the updated first portion of the mapping table from the external memory device through a compute express link (CXL) switch in response to the power-off information, and store the updated first portion of the mapping table in the nonvolatile memory.
2. The storage device of claim 1, wherein the storage controller communicates with the host through the CXL switch.
3. The storage device of claim 2, wherein the storage controller is further configured to: communicate with the host based on a CXL.io protocol that is a Peripheral Component Interconnect Express (PCIe)-based non-coherency input/output protocol, and communicate with the

external memory device based on a CXL.mem protocol that is a memory access protocol.

4. The storage device of claim 1, wherein the storage controller is further configured to transmit a response to the power-off information to the host through the CXL switch after storing the updated first portion of the mapping table in the nonvolatile memory.
5. The storage device of claim 4, wherein the storage device is powered-off after transmitting the response.
6. The storage device of claim 1, wherein when the storage device is powered-up, the storage controller is further configured to send an entirety of the mapping table stored in the nonvolatile memory to the external memory device through the CXL switch.
7. The storage device of claim 1, wherein the storage controller is further configured to receive the updated first portion of the mapping table from the external memory device through the CXL switch based on a peer-to-peer (P2P) manner or a direct memory access (DMA) manner.
8. The storage device of claim 1, wherein the storage controller is further configured to: receive a read request for first user data corresponding to a first logical address from the host; load a second portion of the mapping table from the external memory device; find a first physical address corresponding to the first logical address based on the second portion of the mapping table; read the first user data from the nonvolatile memory based on the first physical address; and transmit the first user data to the host through the CXL switch.
9. The storage device of claim 1, wherein the storage controller is further configured to: receive a write request for second user data corresponding to a second logical address from the host; store the second user data in the nonvolatile memory in response to the write request; generate first map data based on a second physical address corresponding to an area where the second user data are stored, and the second logical address; and update the mapping table stored in the external memory device based on the first map data through the CXL switch.
10. The storage device of claim 9, wherein a first part of the mapping table is not stored in the nonvolatile memory before a power-off of the storage device.
11. A computing system comprising: a storage device including a nonvolatile memory configured to store user data; a memory device including a buffer memory configured to store a mapping table, wherein the mapping table indicating a relationship between a logical address of the user data managed by a host and a physical address of the nonvolatile memory at which the user data are stored; and a compute express link (CXL) switch connected to the storage device, the memory device, and the host, the CXL switch configured to arbitrate communications between the storage device and the memory device, wherein the storage device is configured to receive power-off information from the host and read a first portion of the mapping table from the memory device through the CXL switch in response to the power-off information.
12. The computing system of claim 11, wherein the storage device is further configured to store the first portion of the mapping table in the nonvolatile memory, and to transmit a response to the power-off information to the host through the CXL switch after storing the first portion of the mapping table in the nonvolatile memory.
13. The computing system of claim 12, wherein the storage device is powered-off after transmitting the response.
14. The computing system of claim 11, wherein the storage device is further configured to: receive a read request for first user data corresponding to a first logical address from the host; load a second portion of the mapping table from the memory device; find a first physical address corresponding to the first logical address based on the second portion of the mapping table; read the first user data from the nonvolatile memory based on the first physical address; and transmit the first user data to the host through the CXL switch.
15. The computing system of claim 11, wherein the storage device is further configured to: receive a write request for second user data corresponding to a second logical address from the host; store the second user data in the nonvolatile memory in response to the write request; generate first map

data based on a second physical address corresponding to an area where the second user data are stored, and the second logical address; and update the mapping table stored in the memory device based on the first map data through the CXL switch.

16. The computing system of claim 11, wherein a first memory area of the memory device is allocated to the storage device in an initialization of the computing system, and wherein the storage device is further configured to write the mapping table into the first memory area of the memory device in the initialization of the computing system.

17. An operation method of a computing system including a storage device, a memory device, and a compute express link (CXL) switch, the operation method comprising: receiving, by the storage device, power-off information from a host through the CXL switch; transmitting, by the storage device, a read request to the memory device through the CXL switch in response to the power-off information; reading, by the memory device, a first portion of a mapping table stored in a buffer memory of the memory device in response to the read request, the mapping table indicating a relationship between a logical address of user data managed by the host and a physical address of a first nonvolatile memory at which user data are stored; transmitting, by the memory device, the first portion of the mapping table to the storage device through the CXL switch; and storing, by the storage device, the first portion of the mapping table in a second nonvolatile memory of the storage device.

18. The operation method of claim 17, further comprising: transmitting, by the storage device, a response to the host after storing the first portion of the mapping table in the second nonvolatile memory; and interrupting a power that is provided to the storage device and the memory device.

19. The operation method of claim 17, further comprising: receiving, by the storage device, a read request for first user data corresponding to a first logical address from the host; loading, by the storage device, a second portion of the mapping table from the memory device; finding, by the storage device, a first physical address corresponding to the first logical address based on the second portion of the mapping table; reading, by the storage device, the first user data from the second nonvolatile memory based on the first physical address; and transmit, by the storage device, the first user data to the host through the CXL switch.

20. The operation method of claim 17, further comprising: receiving, by the storage device, a write request for second user data corresponding to a second logical address from the host; storing, by the storage device, the second user data in the second nonvolatile memory in response write request; generating, by the storage device, first map data based on a second physical address corresponding to an area where the second user data are stored, and the second logical address; and updating, by the storage device, the mapping table stored in the memory device based on the first map data through the CXL switch.
