



(51) International Patent Classification:

G06F 3/06 (2006.01)

G06F 12/02 (2006.01)

(21) International Application Number:

PCT/US2022/030439

(22) International Filing Date:

22 May 2022 (22.05.2022)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

17/653,364

03 March 2022 (03.03.2022)

US

(71) Applicant: **WESTERN DIGITAL TECHNOLOGIES, INC.** [US/US]; 5601 Great Oaks Parkway, San Jose, California 95119 (US).

(72) Inventors: **KELNER, Vered**; c/o Western Digital Technologies, Inc., 5601 Great Oaks Parkway, San Jose, California 95119 (US). **FRID, Marina**; c/o Western Digital Technologies, Inc., 5601 Great Oaks Parkway, San Jose, California 95119 (US). **GENSHAFT, Igor**; c/o Western Digital Technologies, Inc., 5601 Great Oaks Parkway, San Jose, California 95119 (US).

(74) Agent: **VERSTEEG, Steven H.**; Patterson & Sheridan LLP, (Western Digital), 24 Greenway Plaza, Suite 1600, Houston, TX 77046 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA,

(54) Title: DATA RELOCATION WITH PROTECTION FOR OPEN RELOCATION DESTINATION BLOCKS

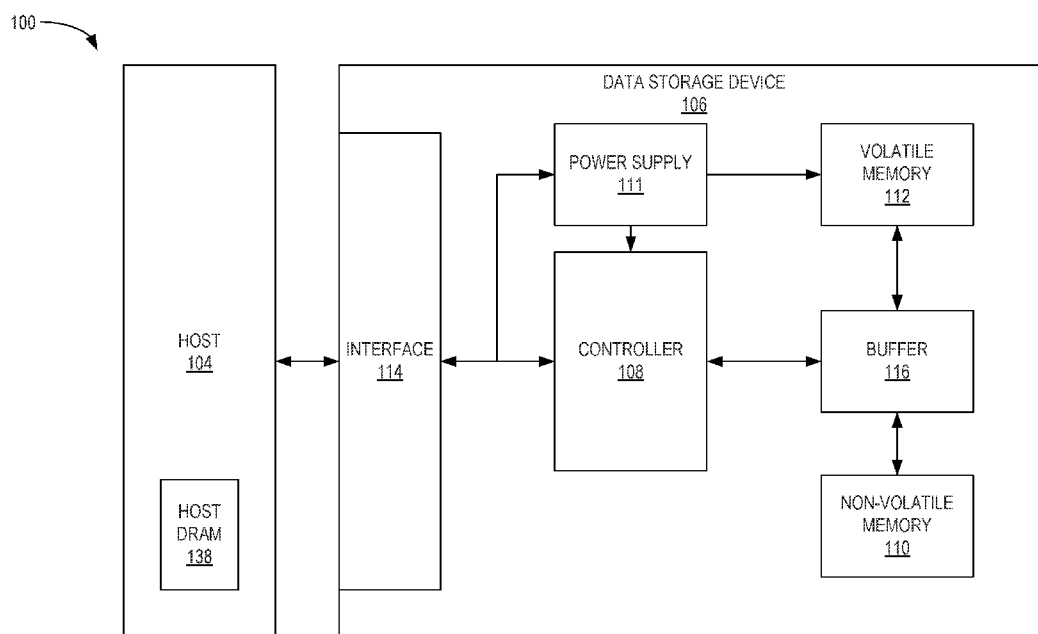


FIG. 1

(57) Abstract: A data storage device includes a memory device and a controller coupled to the memory device. The controller is configured to relocate first valid data from a first source block to a destination block, relocate second valid data from a second source block to the destination block, determine that the destination block is closed, re-mark the first and second source block with a second indication, and erase the source blocks that have the second indication. The first source block and the second source block are marked with a first indication after each respective data is relocated. The first indication indicates that the source block cannot be freed. The second indication indicates that the destination block is closed and the associated source blocks can be erased. Prior to closing the destination block, parity data may be generated for the data of the destination block and programmed to the destination block.

MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI,
NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU,
RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM,
TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM,
ZW.

- (84) Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

Data Relocation With Protection For Open Relocation Destination Blocks**CROSS-REFERENCE TO RELATED APPLICATION(S)**

[0001] This application claims the benefit of and hereby incorporates by reference, for all purposes, the entirety of the contents of U.S. Nonprovisional Application No. 17/653,364, filed March 3, 2022, and entitled “Data Relocation With Protection For Open Relocation Destination Blocks.”

BACKGROUND OF THE DISCLOSURE**Field of the Disclosure**

[0002] Embodiments of the present disclosure generally relate to data storage devices, such as solid state drives (SSDs), and, more specifically, protecting data located in open relocation destination blocks.

Description of the Related Art

[0003] Data storage devices may include one or more memory devices that may be used to store data, such as user data for one or more hosts, metadata, control information, and the like. During operation of a data storage device, the data stored in the one or more memory devices may be moved as part of a data management operation. For example, valid data may be consolidated in a destination block in order to recover memory space as part of a garbage collection operation. When data is programmed to the memory device, the data is also programmed with protection against errors. For example, the data may be encoded with one or more of error correction code (ECC), cyclic redundancy code (CRC), parity data, exclusive or (XOR) data, and the like.

[0004] Furthermore, blocks of the memory device that are currently being programmed to and/or opened may accumulate errors over time. In order to avoid accumulating errors in open blocks, the data storage device may close an open block by programming pad data to the open blocks until the open block is filled. When the open block is filled, the data storage device may close the open block in order to avoid errors from accumulating in the filled open block. However, filling an open block with pad data may decrease the available storage space of the memory device, which may lead to less than advertised storage capabilities. Likewise, by programming additional data protection data to the open blocks, less user data may be stored in the open blocks as the additional data protection data is stored in the open blocks.

[0005] Therefore, there is a need in the art for an improved data relocation operation to protect open relocation destination blocks.

SUMMARY OF THE DISCLOSURE

[0006] The present disclosure generally relates to data storage devices, such as solid state drives (SSDs), and, more specifically, protecting data located in open relocation destination blocks. A data storage device includes a memory device and a controller coupled to the memory device. The controller is configured to relocate first valid data from a first source block to a destination block, relocate second valid data from a second source block to the destination block, determine that the destination block is closed, re-mark the first and second source block with a second indication, and erase the source blocks that have the second indication. The first source block and the second source block are marked with a first indication after each respective data is relocated. The first indication indicates that the source block cannot be freed. The second indication indicates that the destination block is closed and the associated source blocks can be erased. Prior to closing the destination block, parity data may be generated for the data of the destination block and programmed to the destination block.

[0007] In one embodiment, a data storage device includes a memory device and a controller coupled to the memory device. The controller is configured to relocate first valid data from a first source block to a destination block, where the first source block is marked with a first indication to indicate that that first source block cannot be freed, relocate second valid data from a second source block to the destination block, where the second source block is marked with the first indication to indicate that that second source block cannot be freed, determine that the destination block is closed, and erase the first source block and the second source block.

[0008] In another embodiment, a data storage device includes a memory device and a controller coupled to the memory device. The controller is configured to relocate data from a source block to a destination block without erasing the relocated data from the source block and erase the source block when the destination block is closed.

[0009] In another embodiment, a data storage device includes memory means and a controller coupled to the memory means. The controller is configured to erase a source block during a control sync (CS) operation having an indication of can be freed. The erasing occurs after a destination block including valid data of the source block is closed.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] So that the manner in which the above recited features of the present disclosure can be understood in detail, a more particular description of the disclosure, briefly summarized above, may be had by reference to embodiments, some of which are illustrated in the appended drawings. It is to be noted, however, that the appended drawings illustrate only typical

embodiments of this disclosure and are therefore not to be considered limiting of its scope, for the disclosure may admit to other equally effective embodiments.

[0011] Figure 1 is a schematic block diagram illustrating a storage system in which a data storage device may function as a storage device for a host device, according to certain embodiments.

[0012] Figure 2 is an illustration of a superblock of a memory device, according to certain embodiments.

[0013] Figure 3 is a block diagram illustrating moving valid data from source block to a destination block, according to certain embodiments.

[0014] Figure 4 is a block diagram illustrating mapping valid data to invalid data in a plurality of blocks, according to certain embodiments.

[0015] Figure 5 is a block diagram illustrating a data management operation, according to certain embodiments.

[0016] Figure 6 is a flow diagram illustrating a method of reliable relocation of valid source block data to a destination block, according to certain embodiments.

[0017] To facilitate understanding, identical reference numerals have been used, where possible, to designate identical elements that are common to the figures. It is contemplated that elements disclosed in one embodiment may be beneficially utilized on other embodiments without specific recitation.

DETAILED DESCRIPTION

[0018] In the following, reference is made to embodiments of the disclosure. However, it should be understood that the disclosure is not limited to specifically described embodiments. Instead, any combination of the following features and elements, whether related to different embodiments or not, is contemplated to implement and practice the disclosure. Furthermore, although embodiments of the disclosure may achieve advantages over other possible solutions and/or over the prior art, whether or not a particular advantage is achieved by a given embodiment is not limiting of the disclosure. Thus, the following aspects, features, embodiments, and advantages are merely illustrative and are not considered elements or limitations of the appended claims except where explicitly recited in a claim(s). Likewise, reference to “the disclosure” shall not be construed as a generalization of any inventive subject matter disclosed herein and shall not be considered to be an element or limitation of the appended claims except where explicitly recited in a claim(s).

[0019] The present disclosure generally relates to data storage devices, such as solid state drives (SSDs), and, more specifically, protecting data located in open relocation destination blocks. A

data storage device includes a memory device and a controller coupled to the memory device. The controller is configured to relocate first valid data from a first source block to a destination block, relocate second valid data from a second source block to the destination block, determine that the destination block is closed, re-mark the first and second source block with a second indication, and erase the source blocks that have the second indication. The first source block and the second source block are marked with a first indication after each respective data is relocated. The first indication indicates that the source block cannot be freed. The second indication indicates that the destination block is closed and the associated source blocks can be erased. Prior to closing the destination block, parity data may be generated for the data of the destination block and programmed to the destination block.

[0020] Figure 1 is a schematic block diagram illustrating a storage system 100 in which a host device 104 is in communication with a data storage device 106, according to certain embodiments. For instance, the host device 104 may utilize a non-volatile memory (NVM) 110 included in data storage device 106 to store and retrieve data. The host device 104 comprises a host DRAM 138. In some examples, the storage system 100 may include a plurality of storage devices, such as the data storage device 106, which may operate as a storage array. For instance, the storage system 100 may include a plurality of data storage devices 106 configured as a redundant array of inexpensive/independent disks (RAID) that collectively function as a mass storage device for the host device 104.

[0021] The host device 104 may store and/or retrieve data to and/or from one or more storage devices, such as the data storage device 106. As illustrated in Figure 1, the host device 104 may communicate with the data storage device 106 via an interface 114. The host device 104 may comprise any of a wide range of devices, including computer servers, network-attached storage (NAS) units, desktop computers, notebook (i.e., laptop) computers, tablet computers, set-top boxes, telephone handsets such as so-called “smart” phones, so-called “smart” pads, televisions, cameras, display devices, digital media players, video gaming consoles, video streaming device, or other devices capable of sending or receiving data from a data storage device.

[0022] The data storage device 106 includes a controller 108, NVM 110, a power supply 111, volatile memory 112, the interface 114, and a write buffer 116. In some examples, the data storage device 106 may include additional components not shown in Figure 1 for the sake of clarity. For example, the data storage device 106 may include a printed circuit board (PCB) to which components of the data storage device 106 are mechanically attached and which includes electrically conductive traces that electrically interconnect components of the data storage device 106 or the like. In some examples, the physical dimensions and connector configurations of the data storage device 106 may conform to one or more standard form factors. Some example

standard form factors include, but are not limited to, 3.5" data storage device (e.g., an HDD or SSD), 2.5" data storage device, 1.8" data storage device, peripheral component interconnect (PCI), PCI-extended (PCI-X), PCI Express (PCIe) (e.g., PCIe x1, x4, x8, x16, PCIe Mini Card, MiniPCI, etc.). In some examples, the data storage device 106 may be directly coupled (e.g., directly soldered or plugged into a connector) to a motherboard of the host device 104.

[0023] Interface 114 may include one or both of a data bus for exchanging data with the host device 104 and a control bus for exchanging commands with the host device 104. Interface 114 may operate in accordance with any suitable protocol. For example, the interface 114 may operate in accordance with one or more of the following protocols: advanced technology attachment (ATA) (e.g., serial-ATA (SATA) and parallel-ATA (PATA)), Fibre Channel Protocol (FCP), small computer system interface (SCSI), serially attached SCSI (SAS), PCI, and PCIe, non-volatile memory express (NVMe), OpenCAPI, GenZ, Cache Coherent Interface Accelerator (CCIX), Open Channel SSD (OCSSD), or the like. Interface 114 (e.g., the data bus, the control bus, or both) is electrically connected to the controller 108, providing an electrical connection between the host device 104 and the controller 108, allowing data to be exchanged between the host device 104 and the controller 108. In some examples, the electrical connection of interface 114 may also permit the data storage device 106 to receive power from the host device 104. For example, as illustrated in Figure 1, the power supply 111 may receive power from the host device 104 via interface 114.

[0024] The NVM 110 may include a plurality of memory devices or memory units. NVM 110 may be configured to store and/or retrieve data. For instance, a memory unit of NVM 110 may receive data and a message from controller 108 that instructs the memory unit to store the data. Similarly, the memory unit may receive a message from controller 108 that instructs the memory unit to retrieve data. In some examples, each of the memory units may be referred to as a die. In some examples, the NVM 110 may include a plurality of dies (i.e., a plurality of memory units). In some examples, each memory unit may be configured to store relatively large amounts of data (e.g., 128MB, 256MB, 512MB, 1GB, 2GB, 4GB, 8GB, 16GB, 32GB, 64GB, 128GB, 256GB, 512GB, 1TB, etc.).

[0025] In some examples, each memory unit may include any type of non-volatile memory devices, such as flash memory devices, phase-change memory (PCM) devices, resistive random-access memory (ReRAM) devices, magneto-resistive random-access memory (MRAM) devices, ferroelectric random-access memory (F-RAM), holographic memory devices, and any other type of non-volatile memory devices.

[0026] The NVM 110 may comprise a plurality of flash memory devices or memory units. NVM Flash memory devices may include NAND or NOR-based flash memory devices and may

store data based on a charge contained in a floating gate of a transistor for each flash memory cell. In NVM flash memory devices, the flash memory device may be divided into a plurality of dies, where each die of the plurality of dies includes a plurality of physical or logical blocks, which may be further divided into a plurality of pages. Each block of the plurality of blocks within a particular memory device may include a plurality of NVM cells. Rows of NVM cells may be electrically connected using a word line to define a page of a plurality of pages. Respective cells in each of the plurality of pages may be electrically connected to respective bit lines. Furthermore, NVM flash memory devices may be 2D or 3D devices and may be single level cell (SLC), multi-level cell (MLC), triple level cell (TLC), or quad level cell (QLC). The controller 108 may write data to and read data from NVM flash memory devices at the page level and erase data from NVM flash memory devices at the block level.

[0027] The power supply 111 may provide power to one or more components of the data storage device 106. When operating in a standard mode, the power supply 111 may provide power to one or more components using power provided by an external device, such as the host device 104. For instance, the power supply 111 may provide power to the one or more components using power received from the host device 104 via interface 114. In some examples, the power supply 111 may include one or more power storage components configured to provide power to the one or more components when operating in a shutdown mode, such as where power ceases to be received from the external device. In this way, the power supply 111 may function as an onboard backup power source. Some examples of the one or more power storage components include, but are not limited to, capacitors, super-capacitors, batteries, and the like. In some examples, the amount of power that may be stored by the one or more power storage components may be a function of the cost and/or the size (e.g., area/volume) of the one or more power storage components. In other words, as the amount of power stored by the one or more power storage components increases, the cost and/or the size of the one or more power storage components also increases.

[0028] The volatile memory 112 may be used by controller 108 to store information. Volatile memory 112 may include one or more volatile memory devices. In some examples, controller 108 may use volatile memory 112 as a cache. For instance, controller 108 may store cached information in volatile memory 112 until the cached information is written to the NVM 110. As illustrated in Figure 1, volatile memory 112 may consume power received from the power supply 111. Examples of volatile memory 112 include, but are not limited to, random-access memory (RAM), dynamic random access memory (DRAM), static RAM (SRAM), and synchronous dynamic RAM (SDRAM (e.g., DDR1, DDR2, DDR3, DDR3L, LPDDR3, DDR4, LPDDR4, and the like)).

[0029] Controller 108 may manage one or more operations of the data storage device 106. For instance, controller 108 may manage the reading of data from and/or the writing of data to the NVM 110. In some embodiments, when the data storage device 106 receives a write command from the host device 104, the controller 108 may initiate a data storage command to store data to the NVM 110 and monitor the progress of the data storage command. Controller 108 may determine at least one operational characteristic of the storage system 100 and store at least one operational characteristic in the NVM 110. In some embodiments, when the data storage device 106 receives a write command from the host device 104, the controller 108 temporarily stores the data associated with the write command in the internal memory or write buffer 116 before sending the data to the NVM 110.

[0030] Figure 2 is an illustration of a superblock of a memory device 200, according to certain embodiments. The memory device 200 includes a plurality of dies 202a-202n, collectively referred to as dies 202, where each die of the plurality of dies 202a-202n includes a first plane 204a and a second plane 204b, collectively referred to as planes 204. It is to be understood that each die may include more than two planes (e.g., 4 planes, 8 planes, etc.). It is to be understood that the embodiments herein may be applicable to any die architecture having one or more planes. Each of the planes 204 includes a plurality of blocks 206a-206n, collectively referred to as block 206. While 32 dies 202 are shown in the memory device 200, any number of dies may be included. Furthermore, data may be written sequentially in a per block and per plane basis so that data is written to B0 206a before data is written to B1 206b.

[0031] Figure 3 is a block diagram 300 illustrating moving valid data from source block 302 to a destination block 306, according to certain embodiments. For example, the source block 302 may be B0 206a of the first plane 204a and the destination block 306 may be B1 206b of the second plane 204b. The source block 302 includes a plurality of flash management units (FMUs) 304a-304n. Each FMU of the plurality of FMUs 304a-304n may be equal to a maximum storage unit of each block. Furthermore, each FMU may be mapped in a logical block address (LBA) to physical block address (PBA) (L2P) table, which may be stored in volatile memory, such as the volatile memory 112 of Figure 1, a host memory buffer (HMB) of a host DRAM, such as the host DRAM 138 of Figure 1, and/or a controller memory buffer (CMB) of a controller, such as the controller 108 of Figure 1, and/or an NVM, such as the NVM 110 of Figure 1. For exemplary purposes, aspects of Figure 1 may be referenced herein.

[0032] When a data management operation occurs, such as garbage collection, the controller 108 may scan a closed block that includes data. A closed block may refer to a block that is not able to be programmed to. An open block may refer to a block that is able to be programmed to. For example, the destination block 306 may be an open block. Open blocks may be closed due to a

time threshold being met or exceeded or due to the open block being filled to capacity or past a threshold size. For example, the source block 302 may be a closed block. The plurality of FMUs 304a-304n may either be valid FMUs or invalid FMUs. For example, FMU 0 304a, FMU 3 304d, and FMU 5 304f are invalid FMUs and FMU 1 304b, FMU 2 304c, FMU 4 304e, and FMU N 304n are valid FMUs. Invalid FMUs may be FMUs whose data is updated in a different location of the memory device, which may be a different block or the same block. Valid FMUs may be FMUs whose data has not been updated in a different location of the memory device. The controller 108 may store indicators corresponding to valid data and invalid data as well as a mapping of valid data to invalid data. For example, the indicators and/or mappings may be part of a L2P table.

[0033] During a data management operation, such as garbage collection, valid FMUs are moved from a selected source block, such as the source block 302, to a destination block, such as the destination block 306. The destination block 306 includes at least one free FMU, where data may be programmed to the at least one free FMU. If the destination block 306 is filled to capacity, then the controller 108 may choose another destination block that has available space to program data to for the data management operation. In the current example, FMU 1 304b, FMU 2 304c, FMU 4 304e, and FMU N 304n are programmed to the destination block 306, where FMU 1 is now denoted as FMU 1 308a, FMU 2 304c is now denoted as FMU 2 308b, FMU 4 304e is now denoted as FMU 4 308c, and FMU N 304n is now denoted as FMU N 308n. After the valid FMUs are moved to the destination block 306, the controller 108 may erase the source block 302 in order to recover memory space for future write operations.

[0034] The valid FMUs programmed to the destination block 306 may still include data protection mechanisms, such as ECC data, XOR data, parity data, CRC data, and the like. However, in other embodiments, the source block 302 may not be erased until the destination block 306 is closed. If the source block 302 is not erased until the destination block 306 is closed, the data protection mechanisms for data relocated to the destination block 306 may not be needed. Thus, the destination block 306 may store additional data not found in the source block 302. It is to be understood that although only the source block 302 is exemplified, the embodiments described herein may also be applicable to valid data of two or more source blocks being relocated to the destination block 306.

[0035] If data in the destination block 306 is found to be corrupted or have errors, such as through a post-write read operation, a read operation, or the like, then the data may be recovered using the relevant source block data. For example, if FMU 1 308a becomes corrupted, such as due to aggregating one or more flipped bits or errors, then the data of FMU 1 308a may be recovered using FMU 1 304b of the source block 304b. Thus, by not erasing the source block

302 until the destination block 306 is closed, additional data may be programmed to the destination block 306 while still retaining data protection mechanisms for the data relocated to the destination block 306 from the source block 302.

[0036] Furthermore, when the destination block 306 is closed or about to be closed, such as when the controller 108 determines that the destination block 306 is either filled to a minimum threshold limit for closing a block or has been open for at least a predetermined threshold period of time, the controller 108 may then generate parity data for the closed or about to be closed destination block 306. In one example, if the destination block 306 has available memory space and is not yet closed, then the controller 108 may program the generated parity data to the destination block 306. The generated parity data may be a minimal amount of parity data to protect the data stored in the destination block 306. In other examples, even if the destination block 306 has available memory space and is not yet closed, the controller 108 may program the generated parity data to a selected block for parity data. In the previously described example, the controller 108 may program dummy data to the destination block 306 to fill and close the destination block. In yet another example, if the destination block 306 is filled and/or closed, the controller 108 may program the generated parity data to a selected block for parity data.

[0037] Figure 4 is a block diagram 400 illustrating mapping valid data to invalid data in a plurality of blocks 402a-402n, according to certain embodiments. The plurality of blocks 402a-402n may be part of an NVM, such as the NVM 110 of Figure 1. FMUs including invalid data are denoted by a crosshatched pattern and FMUs including valid data do not have a pattern. For example, in block 0 402a, FMUs 000, 003, 005, and 006 are invalid FMUs and FMUs 001, 002, 004, and 007 are valid FMUs. Invalid FMUs may be FMUs whose data is updated in a different block or in the same block. For example, block 1 402b includes FMU 003, which may be updated data corresponding to FMU 003 of block 0 402a, and FMU 006, which may be updated data corresponding to FMU 006 of block 402a. However, FMU 006 of block 1 402b is not the most recent update to the data. Block n 402n includes FMU 006 which may be the most updated version of the data corresponding to FMU 006. Likewise, in block n 402n, FMU 900 is updated within the same block, as indicated by the line with an x-head.

[0038] A controller, such as the controller 108 of Figure 1, may track the updates to the FMUs, where the updates may be stored in a table, such as a L2P table. The stored entries may have a pointer indicating where the updated data is. For example, if the controller 108 searches for FMU 000 in block 0 402a, the controller 108 may first parse through the table storing the table entries corresponding to FMU updates for the relevant entry. Based on the relevant entry, the controller 108 determines that FMU 000 in block 0 402a is not the most recent update to the

FMU 000 data. Rather, the controller 108 may determine that the most recent update to the FMU 000 data is in block n 402n.

[0039] Figure 5 is a block diagram 500 illustrating a data management operation, according to certain embodiments. Relocation (RLC) source blocks 502 include a first source RLC block 504a, a second source RLC block 504b, and a third source RLC block 504c. In one example, a controller, such as the controller 108 of Figure 1, may initiate a data management operation, such as garbage collection or wear-leveling. Valid data or FMUs from the first source RLC block 504a, the second source RLC block 504b, and the third RLC block 504c are relocated to an open RLC block 506. The open RLC block 506 is a destination block. Data of each relocated block may be programmed sequentially to the open RLC block 506. For example, the valid data or FMUs of the first source RLC block 504a is first programmed to the open RLC block 506, followed by the valid data or FMUs of the second RLC block 504b, and finally the valid data or FMUs of the third RLC block 504c. For simplification purposes, the valid data or FMUs of the first source RLC block 504a, the second source RLC block 504b, and the third RLC block 504c in the open RLC block 506 are noted with the same reference numerals corresponding to the same the first source RLC block 504a, the second source RLC block 504b, and the third RLC block 504c in the RLC source blocks 502.

[0040] The controller 108 may maintain a database of source jumbo block addresses (JBAs) 508, where the database of source JBAs 508 may store a plurality of entries, where each entry corresponds to a JBA, a FMU, or the like. For example, the database of source JBAs 508 includes a first entry 510a that corresponds to the valid data of the first RLC block 504a, a second entry 510b that corresponds to the valid data of the second RLC block 504b, and a third entry 510c that corresponds to the valid data of the third RLC block 504c. It is to be understood that each entry may comprise a plurality of sub-entries, where each sub-entry is a data mapping for a relevant FMU or data of the entry.

[0041] Furthermore, each entry and/or sub-entry may also include an indication that indicates if the source data may be erased. For example, when the open RLC block 506 remains open, then the relevant entries corresponding to data stored in the open RLC block 506 are marked with a first indication that indicates that the source data (i.e., the data in the first source RLC block 504a, the second source RLC block 504b, and the third RLC block 504c of the source RLC blocks 502) cannot be erased or freed. In some examples, the first indication may also indicate that the source data cannot be erased or freed during a control sync operation. For example, if a garbage collection operation occurs or a control sync operation occurs, the blocks marked with the first indication are skipped and/or ignored. In some examples, the source blocks may be marked with the first indication when the data of the source blocks is relocated to a destination

block. However, when the open RLC block 506 is closed, then the relevant entries corresponding to data stored in the open RLC block 506 are re-marked with a second indication that indicates that the source data (i.e., the data in the first source RLC block 504a, the second source RLC block 504b, and the third RLC block 504c of the source RLC blocks 502) can be erased or freed.

[0042] A list of source blocks 512 may store information corresponding to the database of source JBAs 508. In some examples, the list of source blocks 512 stores each entry of the database of source JBAs 508. The list of source blocks 512 may be maintained in volatile memory, such as the volatile memory 112 of Figure 1, which may be SRAM, DRAM, or controller RAM. In other examples, each entry of the database of source JBAs may be stored in a block header of a relevant JBA. For example, the first entry 510a may be stored in a block header of the first source RLC block 504a stored in the open RLC block 506. In some examples, the controller 108 may maintain a table of source blocks that have the first indication indicating that the source block cannot be freed or erased in random access memory, such as SRAM, DRAM, and/or controller RAM.

[0043] Figure 6 is a flow diagram illustrating a method 600 of reliable relocation of valid source block data, such as the valid FMUs of the source block 302 of Figure 3, to a destination block, such as the destination block 306 of Figure 3, according to certain embodiments. For exemplary purposes, aspects of the storage system 100 of Figure 1 and the block diagram 500 of Figure 5 may be referenced herein. For example, method 600 may be implemented by the controller 108 of the data storage device 106.

[0044] At block 602, the controller 108 relocates valid data from a first source block, such as the first RLC block 504a of the RLC source blocks 502, to a destination block, such as the open RLC block 506. The relocated valid data from the first source block does not include parity data. Although parity data is exemplified, the embodiments herein may be applicable to other data protection mechanisms that may involve adding additional data to the data programmed to the NVM 110. At block 604, the controller 108 marks the first source block with a first indication, where the first indication indicates that the source block cannot be erased or freed. In some examples, the first indication indicates that the source block cannot be erased or freed during a control sync operation. At block 606, the controller 108 relocates valid data from a second source block, such as the second RLC block 504b of the RLC source blocks 502, to the destination block (e.g., the open RLC block 506). Likewise, the relocated valid data from the second source block does not include parity data. At block 608, the controller 108 marks the second source block with the first indication. If the data stored in the destination block (e.g., the

open RLC block 506) becomes corrupted or needs to be rebuilt, then the controller 108 may read the relevant source block to restore the relevant data.

[0045] At block 610, the controller 108 determines that the destination block (e.g., the open RLC block 506) is full and closed. It is to be understood that the controller 108 may program parity data to the destination block (e.g., the open RLC block 506) when the controller 108 determines that the destination block (e.g., the open RLC block 506) is about to be closed. When destination block (e.g., the open RLC block 506) is closed or about to be closed, such as when the controller 108 determines that the destination block (e.g., the open RLC block 506) is either filled to a minimum threshold limit for closing a block or has been open for at least a predetermined threshold period of time, the controller 108 may then generate parity data for the closed or about to be closed destination block (e.g., the open RLC block 506). The generated parity data may be a minimal amount of parity data to protect the data stored in the destination block 306. In one example, if the destination block (e.g., the open RLC block 506) has available memory space and is not yet closed, then the controller 108 may program the generated parity data to the destination block (e.g., the open RLC block 506). In other examples, even if the destination block (e.g., the open RLC block 506) has available memory space and is not yet closed, the controller 108 may program the generated parity data to a selected block for parity data. In the previously described example, the controller 108 may program dummy data to the destination block (e.g., the open RLC block 506) to fill and close the destination block. In yet another example, if the destination block (e.g., the open RLC block 506) is filled and/or closed, the controller 108 may program the generated parity data to a selected block for parity data.

[0046] At block 612, the controller 108 re-marks the first source block and the second source block with a second indication, where the second indication indicates that the relevant source block can be freed or erased. At block 614, the controller 108 is configured to erase the first source block and the second source block when the first source block and the second source block are the target of a data management operation. In some examples, the controller 108 may erase the first source block and the second source block immediately after the destination block (e.g., the open RLC block 506) is closed. In other examples, the controller 108 may erase the first source block and the second source block after a threshold period of time has elapsed. In yet other examples, the controller 108 may erase the first source block and the second source block when a control sync operation occurs. It is to be understood that the controller 108 may still program user data with the necessary data protection for data that was not relocated to the destination block (e.g., the open RLC block 506). In other words, data that is programmed to the source block that is not associated with any source blocks may still include the necessary data protection mechanisms.

[0047] By not freeing or erasing source blocks whose data is located in an open destination block, parity data and the like may not be necessary for relocated data in the open destination block, which may lead to increased storage space and decreased write amplification of the data storage device.

[0048] In one embodiment, a data storage device includes a memory device and a controller coupled to the memory device. The controller is configured to relocate first valid data from a first source block to a destination block, where the first source block is marked with a first indication to indicate that that first source block cannot be freed, relocate second valid data from a second source block to the destination block, where the second source block is marked with the first indication to indicate that that second source block cannot be freed, determine that the destination block is closed, and erase the first source block and the second source block.

[0049] The controller is further configured to re-mark the first source block and the second source block with a second indication upon determining that the destination block is closed. The second indication indicates that the first source block and the second source block can be freed. The relocated first valid data and the relocated second valid data does not include parity data. The first valid data and the second valid data includes parity data. The controller is further configured to maintain a list of source blocks that have the first indication. The list of source blocks are maintained in random access memory (RAM). When a block is closed, the controller is further configured to check the list of source blocks to determine if there are any source blocks associated with the block that is closed. The controller is further configured to erase one or more source blocks associated with the block that is closed. The controller is further configured to store source block information associated with relocated source data in a destination block header of the destination block. The first indication indicates that a block cannot be freed during a control sync (CS) operation.

[0050] In another embodiment, a data storage device includes a memory device and a controller coupled to the memory device. The controller is configured to relocate data from a source block to a destination block without erasing the relocated data from the source block and erase the source block when the destination block is closed.

[0051] The source block is closed when the data is relocated from the source block to the destination block. The controller is further configured to recover data from the source block after the data is relocated from the source block to the destination block. The controller is further configured to execute a control sync (CS) operation. The source block is not erased during the CS operation. The controller is further configured to mark the source block as can be freed when the destination block is closed. The can be freed indication indicates that the source block can be freed during a control sync (CS) operation. The controller is further configured to

maintain a mapping associating one or more source blocks associated with the relocated data of the destination block to the destination block, and wherein the mapping is stored in random access memory. The destination block includes a destination block header, and wherein the destination block header includes a mapping associating one or more source blocks associated with the relocated data of the destination block to the destination block.

[0052] In another embodiment, a data storage device includes memory means and a controller coupled to the memory means. The controller is configured to erase a source block during a control sync (CS) operation having an indication of can be freed. The erasing occurs after a destination block including valid data of the source block is closed.

[0053] The source block includes parity data and the destination block does not include parity data. The source block is used for data recovery of the destination block when the destination block is open.

[0054] While the foregoing is directed to embodiments of the present disclosure, other and further embodiments of the disclosure may be devised without departing from the basic scope thereof, and the scope thereof is determined by the claims that follow.

WHAT IS CLAIMED IS:

1. A data storage device, comprising:
a memory device; and
a controller coupled to the memory device, the controller configured to:
relocate first valid data from a first source block to a destination block, wherein the first source block is marked with a first indication to indicate that that first source block cannot be freed;
relocate second valid data from a second source block to the destination block, wherein the second source block is marked with the first indication to indicate that that second source block cannot be freed;
determine that the destination block is closed; and
erase the first source block and the second source block.
2. The data storage device of claim 1, wherein the controller is further configured to re-mark the first source block and the second source block with a second indication upon determining that the destination block is closed, and wherein the second indication indicates that the first source block and the second source block can be freed.
3. The data storage device of claim 1, wherein the relocated first valid data and the relocated second valid data does not include parity data.
4. The data storage device of claim 3, wherein the first valid data and the second valid data includes parity data.
5. The data storage device of claim 1, wherein the controller is further configured to maintain a list of source blocks that have the first indication.
6. The data storage device of claim 5, wherein the list of source blocks are maintained in random access memory (RAM).
7. The data storage device of claim 5, wherein, when a block is closed, the controller is further configured to check the list of source blocks to determine if there are any source blocks associated with the block that is closed.

8. The data storage device of claim 7, wherein the controller is further configured to erase one or more source blocks associated with the block that is closed.
9. The data storage device of claim 1, wherein the controller is further configured to store source block information associated with relocated source data in a destination block header of the destination block.
10. The data storage device of claim 1, wherein the first indication indicates that a block cannot be freed during a control sync (CS) operation.
11. A data storage device, comprising:
 - a memory device; and
 - a controller coupled to the memory device, the controller configured to:
 - relocate data from a source block to a destination block without erasing the relocated data from the source block; and
 - erase the source block when the destination block is closed.
12. The data storage device of claim 11, wherein the source block is closed when the data is relocated from the source block to the destination block.
13. The data storage device of claim 11, wherein the controller is further configured to recover data from the source block after the data is relocated from the source block to the destination block.
14. The data storage device of claim 11, wherein the controller is further configured to execute a control sync (CS) operation, and wherein the source block is not erased during the CS operation.
15. The data storage device of claim 11, wherein the controller is further configured to mark the source block as can be freed when the destination block is closed.
16. The data storage device of claim 15, wherein the can be freed indication indicates that the source block can be freed during a control sync (CS) operation.

17. The data storage device of claim 11, wherein the controller is further configured to maintain a mapping associating one or more source blocks associated with the relocated data of the destination block to the destination block, and wherein the mapping is stored in random access memory.
18. The data storage device of claim 11, wherein the destination block includes a destination block header, and wherein the destination block header includes a mapping associating one or more source blocks associated with the relocated data of the destination block to the destination block.
19. A data storage device, comprising:
memory means; and
a controller coupled to the memory means, the controller configured to:
erase a source block during a control sync (CS) operation having an indication of can be freed, wherein the erasing occurs after a destination block including valid data of the source block is closed.
20. The data storage device of claim 19, wherein the source block includes parity data and the destination block does not include parity data, and wherein the source block is used for data recovery of the destination block when the destination block is open.

100

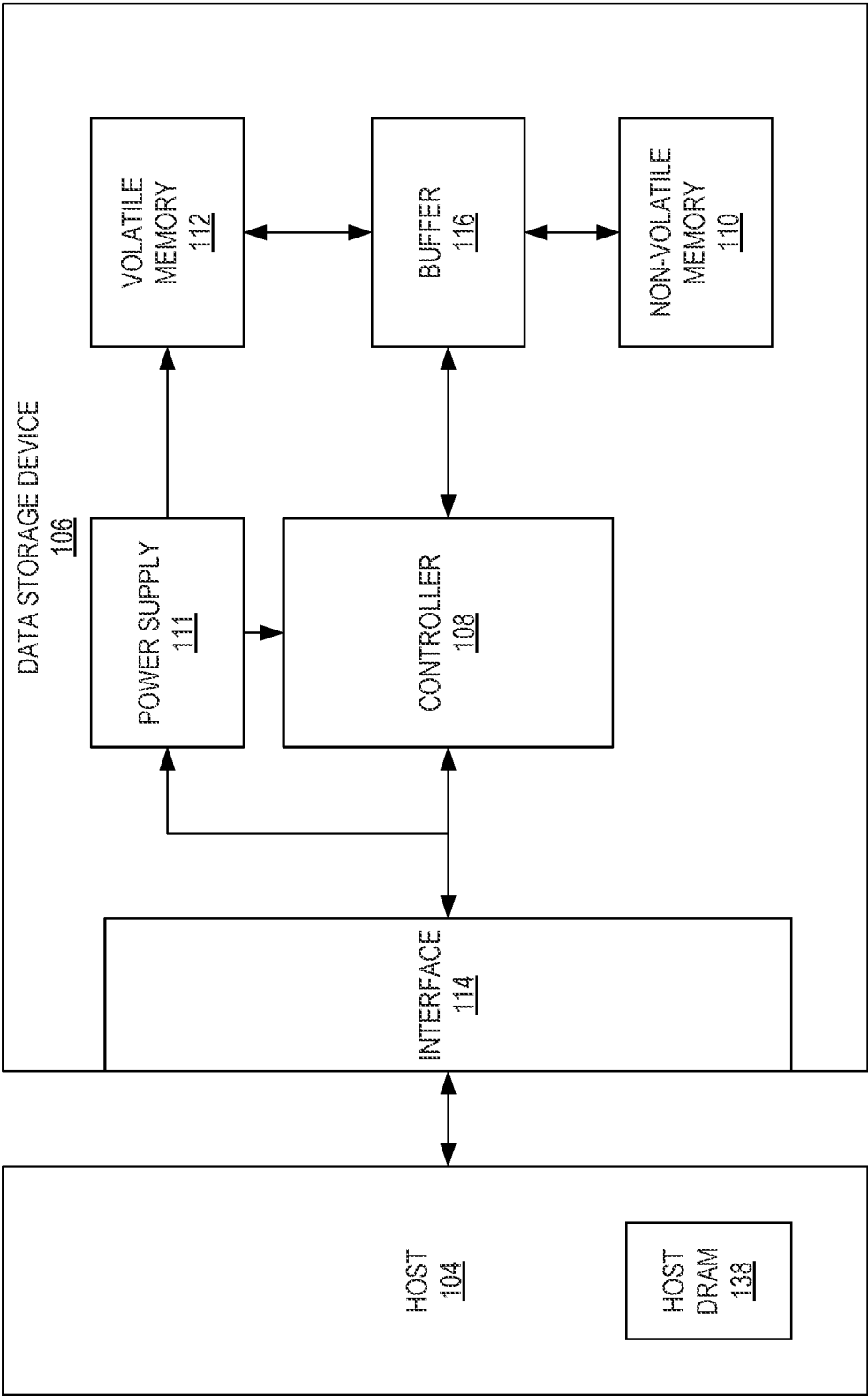
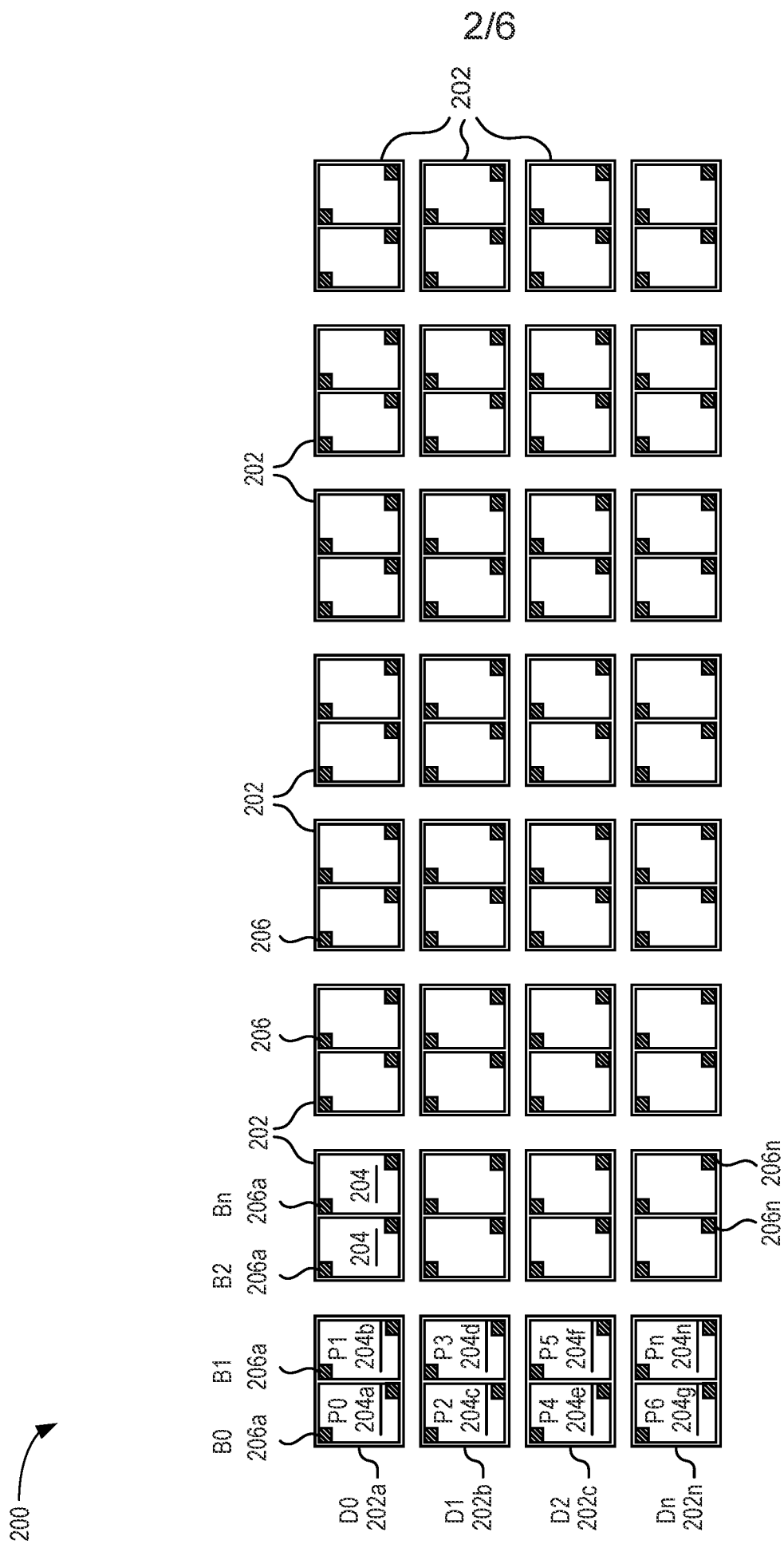


FIG. 1



2GLL

300

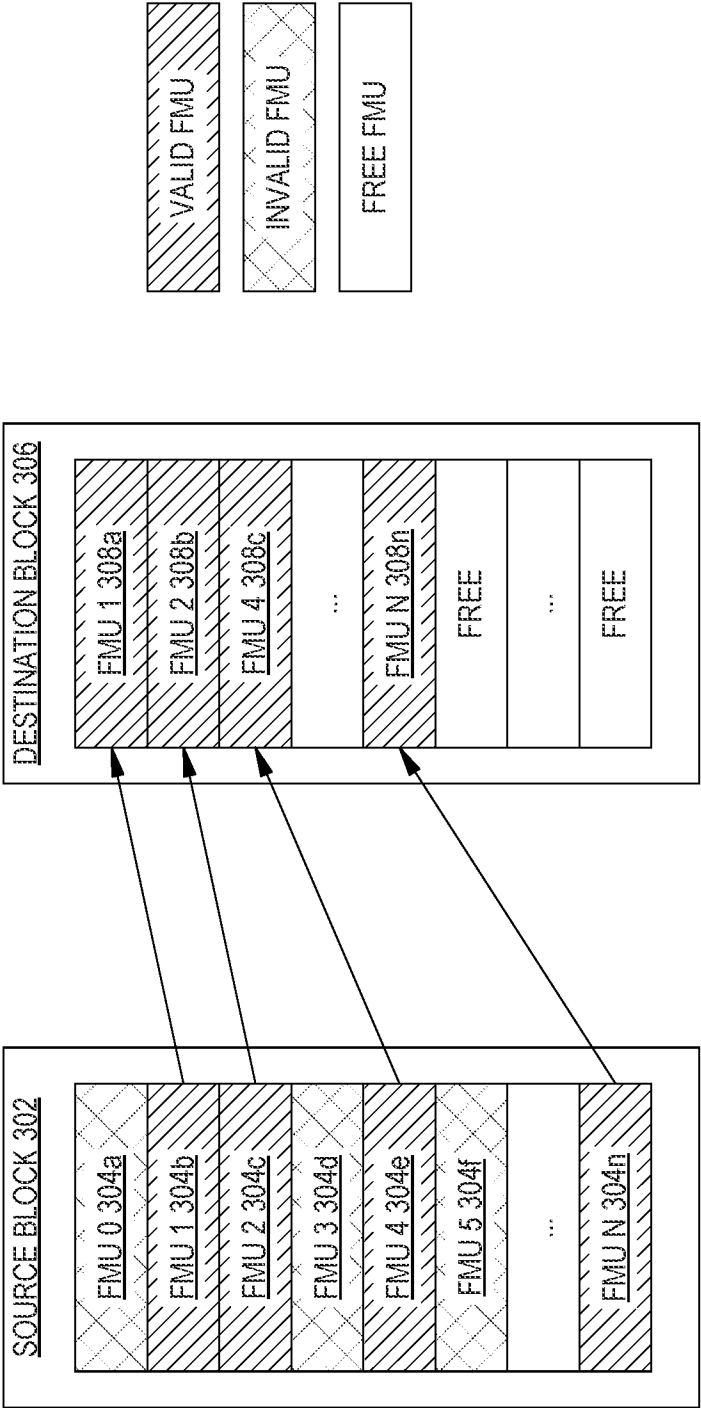


FIG. 3

400

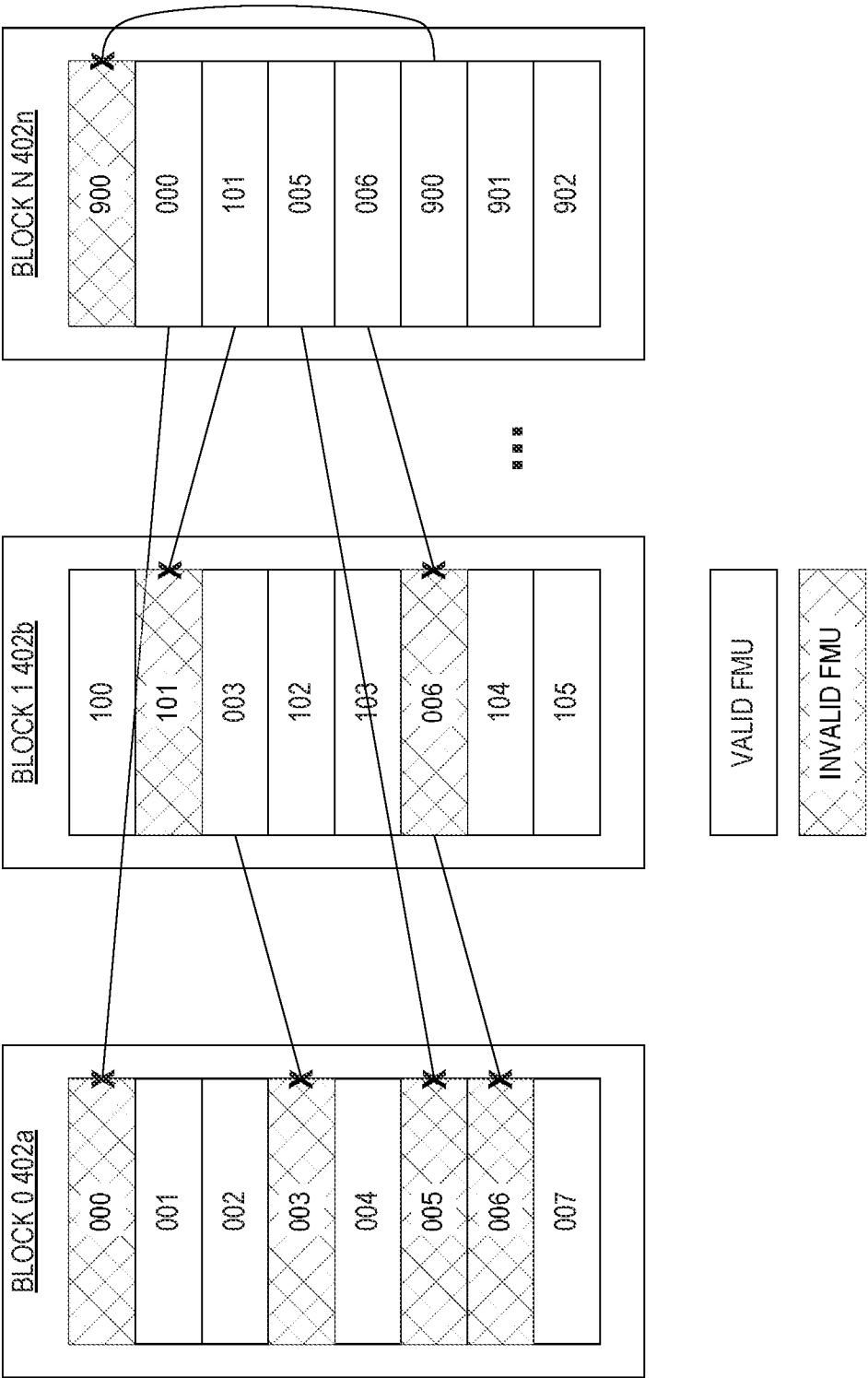


FIG. 4

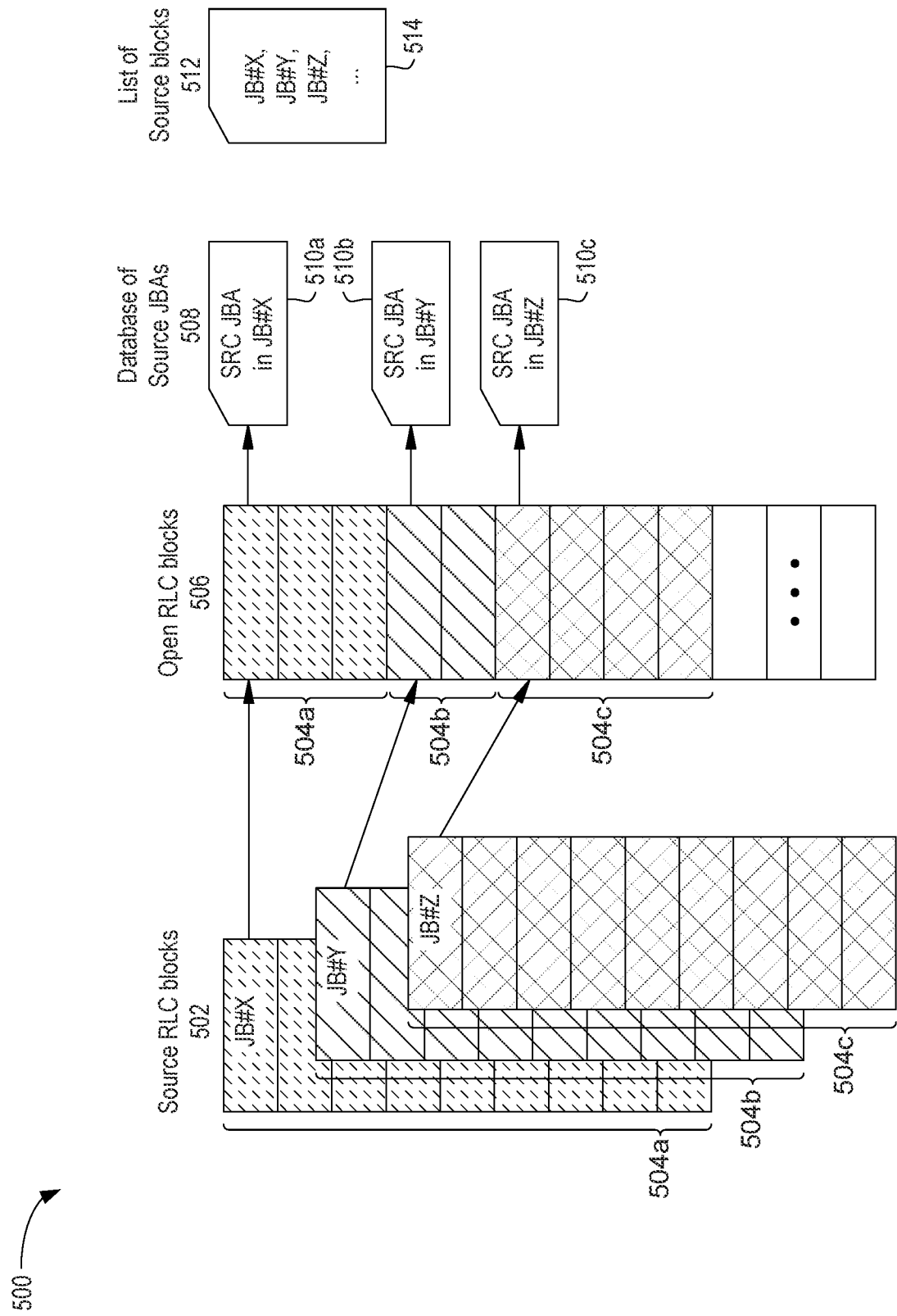


FIG. 5

600 

6/6

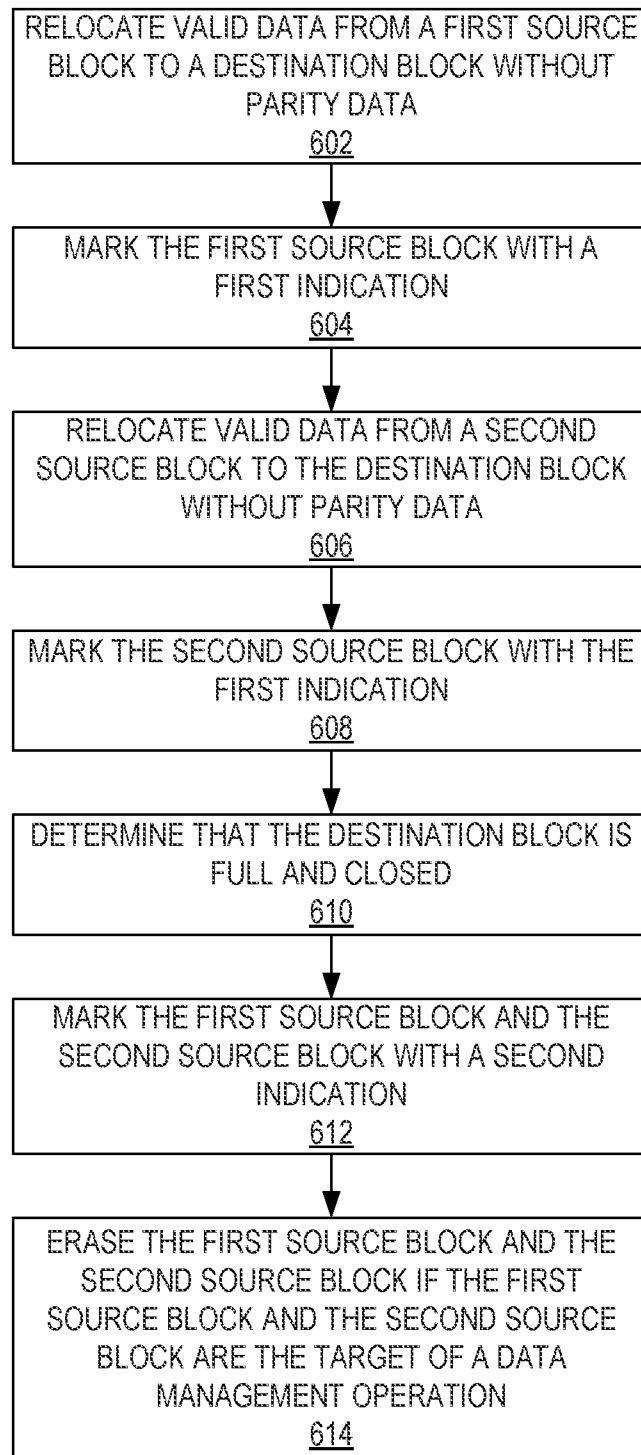


FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US2022/030439

A. CLASSIFICATION OF SUBJECT MATTER**G06F 3/06**(2006.01)i; **G06F 12/02**(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 3/06(2006.01); G06F 12/00(2006.01); G06F 12/02(2006.01); G06F 12/10(2006.01); G11C 16/26(2006.01)

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: memory, block, relocate, destination, erase, close, valid, control sync (CS)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2017-0351439 A1 (SANDISK TECHNOLOGIES LLC) 07 December 2017 (2017-12-07) paragraphs [0068], [0075], [0131]; claims 1, 9-10; and figures 1A, 3	1-20
Y	US 2006-0184720 A1 (ALAN W. SINCLAIR et al.) 17 August 2006 (2006-08-17) paragraphs [0086]-[0087]; and figure 12A	1-20
Y	US 2018-0357010 A1 (WESTERN DIGITAL TECHNOLOGIES, INC.) 13 December 2018 (2018-12-13) paragraph [0006]	10,14-17,19-20
A	US 2018-0024920 A1 (SANDISK TECHNOLOGIES LLC) 25 January 2018 (2018-01-25) paragraphs [0017]-[0027]; claims 1-8; and figures 1A-1C	1-20
A	US 2021-0318810 A1 (WESTERN DIGITAL TECHNOLOGIES, INC.) 14 October 2021 (2021-10-14) paragraphs [0010]-[0023]; claims 1-10; and figures 1-2	1-20



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“D” document cited by the applicant in the international application

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

21 November 2022

Date of mailing of the international search report

21 November 2022

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon
35208, Republic of Korea

Facsimile No. +82-42-481-8578

Authorized officer

YANG, Jeong Rok

Telephone No. +82-42-481-5709

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/US2022/030439

Patent document cited in search report				Publication date (day/month/year)		Patent family member(s)	Publication date (day/month/year)
US	2017-0351439	A1	07 December 2017	US	10268396	B2	23 April 2019
				US	2016-0188221	A1	30 June 2016
				US	9766819	B2	19 September 2017
US	2006-0184720	A1	17 August 2006	CN	101238431	A	06 August 2008
				EP	1849079	A2	31 October 2007
				IL	185174	A	03 December 2007
				JP	2008-530710	A	07 August 2008
				JP	2009-503731	A	29 January 2009
				JP	5236469	B2	17 July 2013
				KR	10-1042588	B1	20 June 2011
				KR	10-1344688	B1	26 December 2013
				KR	10-1449543	B1	13 October 2014
				KR	10-2007-0116792	A	11 December 2007
				KR	10-2008-0000557	A	02 January 2008
				KR	10-2008-0046648	A	27 May 2008
				TW	200641602	A	01 December 2006
				TW	I400608	B	01 July 2013
				US	10055147	B2	21 August 2018
				US	10126959	B2	13 November 2018
				US	2006-0184719	A1	17 August 2006
				US	2007-0033362	A1	08 February 2007
				US	2010-0223423	A1	02 September 2010
				US	2015-0363135	A1	17 December 2015
				US	7877539	B2	25 January 2011
				US	7984233	B2	19 July 2011
				US	8214583	B2	03 July 2012
				US	9104315	B2	11 August 2015
				WO	2006-088719	A2	24 August 2006
				WO	2006-088727	A2	24 August 2006
				WO	2007-019076	A2	15 February 2007
US	2018-0357010	A1	13 December 2018	CN	109032958	A	18 December 2018
				DE	102018110722	A1	13 December 2018
				US	10732896	B2	04 August 2020
US	2018-0024920	A1	25 January 2018	US	9921956	B2	20 March 2018
US	2021-0318810	A1	14 October 2021	CN	110955385	A	03 April 2020
				DE	102019116520	A1	26 March 2020
				US	11086529	B2	10 August 2021
				US	2020-0097189	A1	26 March 2020