



- (51) **International Patent Classification:**
G06F 3/06 (2006.01) G06F 12/02 (2006.01)
- (21) **International Application Number:**
PCT/US2013/069481
- (22) **International Filing Date:**
11 November 2013 (11.11.2013)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
13/720,024 19 December 2012 (19.12.2012) US
- (71) **Applicant:** APPLE INC. [US/US]; 1 Infinite Loop, Cupertino, California 95014 (US).
- (72) **Inventor:** SCHWARTZ, Yair; 1 Infinite Loop, Cupertino, California 95014 (US).
- (74) **Agent:** KIVLIN, B. Noel; Meyertons, Hood, Kivlin, Kowert &, Goetzel, P.C., P.O. Box 398, Austin, Texas 78767-0398 (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, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, 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, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, 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, SD, SL, 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))

- (54) **Title:** JOINT LOGICAL AND PHYSICAL ADDRESS REMAPPING IN NON-VOLATILE MEMORY

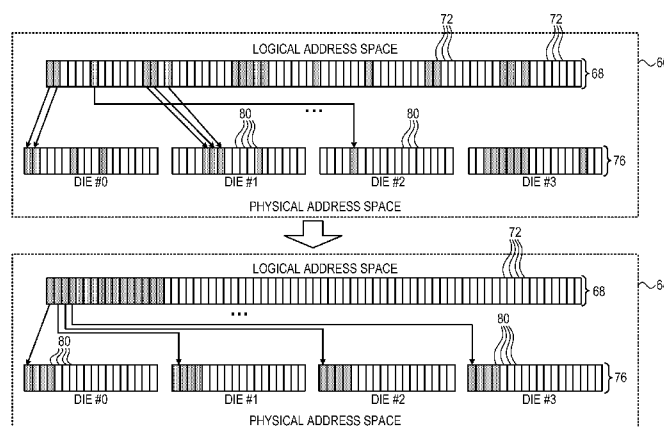


FIG. 2

- (57) **Abstract:** A method includes, for data items that are to be stored in a non-volatile memory in accordance with respective logical addresses, associating the logical addresses with respective physical storage locations in the non-volatile memory, and storing the data items in the respective associated physical storage locations. A remapping command, which specifies a group of source logical addresses that are associated with respective source physical storage locations, is received. In response to the remapping command, destination physical storage locations and destination logical addresses are selected jointly for replacing the source physical storage locations and the source logical addresses, respectively, so as to meet a joint performance criterion with respect to the logical addresses and the physical storage locations. The data items are copied from the source physical storage locations to the respective destination physical storage locations, and the destination physical storage locations are re-associated with the respective destination logical addresses.

JOINT LOGICAL AND PHYSICAL ADDRESS REMAPPING IN NON-VOLATILE MEMORY

FIELD OF THE INVENTION

The present invention relates generally to data storage, and
5 particularly to methods and systems for data storage management
in non-volatile memory.

BACKGROUND OF THE INVENTION

Various types of data storage systems use logical-to-
physical address translation. In such systems, data is provided
10 for storage in specified logical addresses, and the logical
addresses are translated into respective physical addresses in
which the data is physically stored. Address translation schemes
of this sort are used, for example, in Flash Translation Layers
(FTL) that manage data storage in Flash memory.

SUMMARY OF THE INVENTION

An embodiment of the present invention that is described
herein provides a method including, for data items that are to be
stored in a non-volatile memory in accordance with respective
logical addresses, associating the logical addresses with
20 respective physical storage locations in the non-volatile memory,
and storing the data items in the respective associated physical
storage locations. A remapping command, which specifies a group
of source logical addresses that are associated with respective
source physical storage locations, is received. In response to
25 the remapping command, destination physical storage locations and
destination logical addresses are selected jointly for replacing
the source physical storage locations and the source logical
addresses, respectively, so as to meet a joint performance
criterion with respect to the logical addresses and the physical
30 storage locations. The data items are copied from the source
physical storage locations to the respective destination physical
storage locations, and the destination physical storage locations

are re-associated with the respective destination logical addresses.

In some embodiments, jointly selecting the destination physical storage locations and the destination logical addresses includes reducing a first number of logical memory fragments occupied by the destination logical addresses relative to the source logical addresses, and reducing a second number of physical memory fragments occupied by the destination physical storage locations, relative to the source physical storage locations.

In an embodiment, jointly selecting the destination physical storage locations and the destination logical addresses includes increasing a throughput of accessing the data items in the non-volatile memory. In another embodiment, jointly selecting the destination physical storage locations and the destination logical addresses includes reducing a latency of accessing the data items in the non-volatile memory.

In a disclosed embodiment, jointly selecting the destination physical storage locations and the destination logical addresses includes selecting the destination logical addresses in a first contiguous sequence, and selecting the respective destination physical storage locations in a second contiguous sequence. In an alternative embodiment, the non-volatile memory includes multiple memory units, and jointly selecting the destination physical storage locations and the destination logical addresses includes selecting the destination logical addresses in a contiguous sequence, and selecting the respective destination physical storage locations in cyclical alternation among the multiple memory units.

In yet another embodiment, jointly selecting the destination physical storage locations and the destination logical addresses includes increasing a compressibility of a data structure used for storing respective associations between the logical addresses and the physical storage locations. In still another embodiment,

receiving the remapping command includes receiving an indication of the destination logical addresses in the command.

In some embodiments, the remapping command does not indicate the destination logical addresses, and jointly selecting the destination physical storage locations and the destination logical addresses includes deciding the destination logical addresses in response to receiving the command. The method may include outputting a notification of the decided destination logical addresses. In an embodiment, jointly selecting the destination physical storage locations and the destination logical addresses includes identifying an idle time period, and choosing the destination physical storage locations and the destination logical addresses during the idle time period.

There is additionally provided, in accordance with an embodiment of the present invention, apparatus including an interface and a processor. The interface is configured for communicating with a non-volatile memory. The processor is configured, for data items that are to be stored in the non-volatile memory in accordance with respective logical addresses, to associate the logical addresses with respective physical storage locations in the non-volatile memory and to store the data items in the respective associated physical storage locations, to receive a remapping command, which specifies a group of source logical addresses that are associated with respective source physical storage locations, to jointly select, in response to the remapping command, destination physical storage locations and destination logical addresses for replacing the source physical storage locations and the source logical addresses, respectively, so as to meet a joint performance criterion with respect to the logical addresses and the physical storage locations, to copy the data items from the source physical storage locations to the respective destination physical storage locations, and to re-associate the destination physical storage locations with the respective destination logical addresses.

The present invention will be more fully understood from the following detailed description of the embodiments thereof, taken together with the drawings in which:

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram that schematically illustrates a memory system, in accordance with an embodiment of the present invention;

Fig. 2 is a diagram that schematically illustrates a joint logical and physical address remapping process, in accordance with an embodiment of the present invention; and

Fig. 3 is a flow chart that schematically illustrates a method for joint logical and physical address remapping, in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF EMBODIMENTS

15 OVERVIEW

Embodiments of the present invention that are described herein provide methods and systems for arranging the logical and physical addresses of data stored in a non-volatile memory, in order to improve storage performance and simplify storage management tasks and data structures.

Consider, for example, an embodiment in which a host stores files in a Solid State Drive (SSD) or other non-volatile memory. The host and storage device use a logical addressing scheme, and the SSD translates between logical addresses and corresponding physical addresses. The terms "physical addresses" and "physical storage locations" are used interchangeably herein.

Over time, the logical addresses used for storing the data of a given file may become fragmented, i.e., non-contiguous and often scattered in multiple fragments across the logical address space. Fragmentation of the logical addresses may develop, for example, when changes are applied to the file after it is initially created. In addition to the logical address fragmentation, the physical addresses in which the data of the

file is stored in the non-volatile memory may also become fragmented. Physical address fragmentation may develop, for example, because of block compaction ("garbage collection") and other storage management processes performed in the non-volatile
5 memory.

Thus, over time, a given file often becomes fragmented both in the logical address space and in the physical address space. Fragmentation in the two domains (logical and physical) is often uncorrelated and caused by different reasons. Both types of
10 fragmentation, however, are undesirable and degrade the overall storage performance.

In some embodiments that are described herein, the storage device carries out a joint address remapping operation that reduces the fragmentation of a given file in both the logical and
15 the physical address spaces. The joint de-fragmentation process replaces both the logical addresses and the corresponding physical addresses of the file with new addresses, so as to meet a performance criterion defined over both the logical address space and the physical address space.

It is possible in principle to de-fragment the logical addresses and the physical addresses separately. Such a solution, however, will usually be sub-optimal and sometimes detrimental to the storage device performance. De-fragmenting the logical
20 addresses without considering the corresponding physical addresses is likely to worsen the physical address fragmentation, and *vice versa*.

Several examples of joint remapping schemes, and joint performance criteria that are met by these schemes, are described herein. In comparison with the naïve solution of independent
30 logical and physical de-fragmentation, the disclosed techniques are able to achieve superior storage throughput and latency, as well as reduced overhead and increased lifetime of the non-volatile memory.

Moreover, the disclosed techniques reduce the size and
35 complexity of the data structures used for storing the logical-

to-physical translation, as well as the data structures used by the host file system. Furthermore, the joint remapping operation is performed internally to the storage device without a need to transfer data between the storage device and the host. Therefore, communication load over the interface between the host and the storage device, as well as loading of host resources, are reduced.

SYSTEM DESCRIPTION

Fig. 1 is a block diagram that schematically illustrates a memory system, in accordance with an embodiment of the present invention. In the present example, the memory system comprises a computer 20 that stores data in a Solid state Drive (SSD) 24. Computer 20 may comprise, for example, a mobile, tablet or personal computer. The computer comprises a Central Processing Unit (CPU) 26 that serves as a host.

In alternative embodiments, the host may comprise any other suitable processor or controller, and the storage device may comprise any other suitable device. For example, the host may comprise a storage controller of an enterprise storage system, and the storage device may comprise an SSD or an array of SSDs. Other examples of hosts that store data in non-volatile storage devices comprise mobile phones, digital cameras, media players and removable memory cards or devices.

SSD 24 stores data for CPU 26 in a non-volatile memory, in the present example in one or more NAND Flash memory devices 34. In alternative embodiments, the non-volatile memory in SSD 24 may comprise any other suitable type of non-volatile memory, such as, for example, NOR Flash, Charge Trap Flash (CTF), Phase Change RAM (PRAM), Magnetoresistive RAM (MRAM) or Ferroelectric RAM (FeRAM).

An SSD controller 30 performs the various storage and management tasks of the SSD. The SSD controller is also referred to generally as a memory controller. SSD controller 30 comprises a host interface 38 for communicating with CPU 26, a memory interface 46 for communicating with Flash devices 34, and a

processor 42 that carries out the various processing tasks of the SSD.

SSD 24 further comprises a volatile memory, in the present example a Random Access Memory (RAM) 50. In the embodiment of Fig. 1 RAM 50 is shown as part of SSD controller 30, although the RAM may alternatively be separate from the SSD controller. RAM 50 may comprise, for example a Static RAM (SRAM), a Dynamic RAM (DRAM), a combination of the two RAM types, or any other suitable type of volatile memory.

SSD controller 30, and in particular processor 42, may be implemented in hardware. Alternatively, the SSD controller may comprise a microprocessor that runs suitable software, or a combination of hardware and software elements.

The configuration of Fig. 1 is an exemplary configuration, which is shown purely for the sake of conceptual clarity. Any other suitable SSD or other memory system configuration can also be used. Elements that are not necessary for understanding the principles of the present invention, such as various interfaces, addressing circuits, timing and sequencing circuits and debugging circuits, have been omitted from the figure for clarity. In some applications, e.g., non-SSD applications, the functions of SSD controller 30 are carried out by a suitable memory controller.

In the exemplary system configuration shown in Fig. 1, memory devices 34 and SSD controller 30 are implemented as separate Integrated Circuits (ICs). In alternative embodiments, however, the memory devices and the SSD controller may be integrated on separate semiconductor dies in a single Multi-Chip Package (MCP) or System on Chip (SoC), and may be interconnected by an internal bus. Further alternatively, some or all of the SSD controller circuitry may reside on the same die on which one or more of memory devices 34 are disposed. Further alternatively, some or all of the functionality of SSD controller 30 can be implemented in software and carried out by CPU 26 or other processor in the computer. In some embodiments, CPU 26 and SSD

controller 30 may be fabricated on the same die, or on separate dies in the same device package.

In some embodiments, processor 42 comprises a general-purpose processor, which is programmed in software to carry out the functions described herein. The software may be downloaded to the processor in electronic form, over a network, for example, or it may, alternatively or additionally, be provided and/or stored on non-transitory tangible media, such as magnetic, optical, or electronic memory.

CPU 26 of computer 20 typically runs a File System (FS - not shown in the figure), which stores one or more files in SSD 24. The FS stores the files in the SSD using a logical addressing scheme. In such a scheme, the FS assigns each file a group of one or more logical addresses (also referred to as Logical Block Addresses - LBAs), and sends the file data to SSD 24 for storage in accordance with the LBAs.

Processor 42 of SSD controller 30 typically maintains a logical-to-physical address translation, which associates the logical addresses specified by the host with respective physical storage locations (also referred to as physical addresses) in Flash devices 34, and stores the data in the appropriate physical storage locations. The logical-to-physical address translation (also referred to as Virtual-to-Physical mapping - V2P) may be stored in RAM 50, in Flash devices 34, or in both.

JOINT REMAPPING OF LOGICAL AND PHYSICAL ADDRESSES

Fig. 2 is a diagram that schematically illustrates a joint logical and physical address remapping process, in accordance with an embodiment of the present invention. The top of the figure shows an association (mapping) 60 of logical addresses 72 with corresponding physical addresses 80, before applying joint address remapping. The bottom of the figure shows an improved association (mapping) 64, which is produced by the disclosed joint remapping operation. In the figure, shaded logical and physical addresses denote mark the data of a particular file of

the host FS, and arrows connect the logical addresses to the respective associated physical addresses.

In the present example, each logical address 72 corresponds to a respective logical page in a logical address space 68. Each physical address 80 corresponds to a respective physical page in a physical address space 76 of Flash devices 34. In the example of Fig. 2, the physical address space spans four Flash dies denoted Die#0...Die#3. In alternative embodiments, the logical-to-physical address mapping may be defined using any other suitable mapping unit, e.g., block or sector, and the logical and physical address spaces may have any other suitable configuration.

Consider mapping 60 at the top of Fig. 2. In this example, logical addresses 72 of the file in question are severely fragmented across logical address space 68. At the same time, physical addresses 80 of the file are severely fragmented across physical address space 76.

At some point in time, processor 42 of SSD controller 30 receives from CPU 26 a remapping command. In response to the command, processor 42 jointly remaps the logical and physical addresses of the file, so as to produce mapping 64 at the bottom of the figure. (In a typical Flash memory, data cannot be overwritten in-place, and therefore the new physical addresses of the data will typically reside in new memory blocks. This feature is not shown in Fig. 2 for the sake of clarity.)

As can be seen in the figure, both the logical addresses and the physical addresses of the file are considerably less fragmented in mapping 64 in comparison with mapping 60. The remapping operation considers fragmentation in the logical address space and in the physical address space jointly, rather than trying to de-fragment each address space separately from the other.

In the present context, the logical and physical addresses of the file in mapping 60 (before remapping) are referred to as source logical and physical addresses, respectively. The logical and physical addresses of the file in mapping 64 (after

remapping) are referred to as destination logical and physical addresses, respectively. The remapping operation thus selects the destination logical and physical addresses for replacing the source logical and physical addresses of the file.

5 Processor 42 typically remaps the source logical and physical addresses so as to meet a certain performance criterion that is defined over both the logical and physical domains, i.e., over both the logical and physical addresses. In various embodiments, processor 42 may use different performance criteria
10 for selecting the destination logical and physical addresses for the remapping operation.

In one example embodiment, the remapping is performed so as to reduce or minimize the amount of fragmentation in the two domains. In other words, processor 42 selects the destination
15 logical and physical addresses so as to reduce the number of fragments of logical address space 68 in which the file data is stored, and at the same time to reduce the number of fragments of physical address space 76 in which the file data is stored.

In another embodiment, processor 42 selects the remapping
20 operation so as to maximize the storage (write and/or read) throughput of SSD 30. Such a criterion typically depends on the structure of the SSD. The remapping operation of Fig. 2, for example, is suitable for an SSD that supports multi-die read and write commands, which read and write multiple corresponding pages
25 in multiple respective dies in parallel. In order to best utilize these commands mapping 64 maps successive logical addresses 72 are mapped to physical addresses that alternate cyclically among the four dies. A similar alternation can be applied among other types of physical memory units, such as memory devices, memory
30 planes or even memory blocks. In yet another embodiment, processor 42 configures the remapping operation so as to minimize the storage (write and/or read) latency of SSD 24.

In other embodiments, the remapping operation is chosen so as to reduce the size and/or complexity of a data structure in
35 the host or in the storage device. For example, the remapping may

be selected so as to make the V2P mapping of the SSD as compressible as possible. High compressibility is typically achieved by reducing fragmentation, but may also depend on the specific configuration of the data structure used for storing the V2P mapping. As another example, the remapping may be selected so as to simplify the data structure used for storing the mapping of files to LBAs in the host.

Further alternatively, processor 42 may remap the logical and physical addresses so as to meet any other suitable performance criterion.

As explained above, the remapping command is typically sent from CPU 26 (or more generally from the host) to processor 42 (or more generally to the storage device). The command typically indicates the group of source logical addresses of the file that is to be remapped. In some embodiments, the destination logical addresses are selected by the host FS. In such an implementation the destination logical addresses are specified in the remapping command in addition to the source logical addresses.

In alternative embodiments, the command specifies only the source logical addresses, and the storage device (e.g., processor 42) selects the destination logical addresses. The storage device thus notifies the host of the selected destination logical addresses. These embodiments are typically used when the host and storage device use trim commands, which indicate to the storage device which logical addresses are not in use by the host FS. In either case, the destination physical addresses are selected by processor 42.

JOINT REMAPPING METHOD DESCRIPTION

Fig. 3 is a flow chart that schematically illustrates a method for joint logical and physical address remapping, in accordance with an embodiment of the present invention. The method begins with processor 42 receiving from CPU 26 data items for storage in Flash devices 34, at an input step 90. The data

items are received via interface 38 for storage in respective logical addresses.

Processor 42 associates the logical addresses of the data items with respective physical addresses, at a mapping step 94,
5 and stores the data items in the respective physical addresses, at a storage step 98. The storage process of steps 90-98 is typically carried out whenever CPU 26 (or more generally the host) has data items to store in the SSD.

At some point in time, CPU 26 sends to SSD 24 a remapping
10 command for a particular file, at a remapping command step 102. The remapping command indicates the group of logical addresses in which the data items of the file are stored (i.e., the source logical addresses). The source logical addresses of the file are associated (in accordance with the mapping of step 94 above) with
15 respective source physical addresses.

In response to the remapping command, processor 42 selects destination logical addresses to replace the respective source logical addresses, at a logical remapping step 106, and selects destination physical addresses to replace the respective source
20 physical addresses, at a physical remapping step 110. The selection of destination logical and physical addresses (steps 106 and 110) is performed jointly, so as to meet a performance criterion with respect to the logical and physical addresses.

Processor 42 copies the data items of the file from the
25 source physical addresses to the destination physical addresses, at a copying step 114. Processor 42 associates the destination logical addresses with the corresponding destination physical addresses, at a logical re-association step 118. Typically, processor 42 updates the V-P mapping to reflect the improved
30 mapping.

In some embodiments, processor 42 carries out the remapping operation in a background task, which is executed during idle time periods in which the processor is not busy executing storage commands. Processor 42 typically identifies such idle time
35 periods, and carries out the remapping task during these periods.

Background operation of this sort enables processor 42, for example, to copy and remap large bodies of data so as to occupy large contiguous address ranges in both the logical and physical domains.

5 It will be appreciated that the embodiments described above are cited by way of example, and that the present invention is not limited to what has been particularly shown and described hereinabove. Rather, the scope of the present invention includes
10 both combinations and sub-combinations of the various features described hereinabove, as well as variations and modifications thereof which would occur to persons skilled in the art upon reading the foregoing description and which are not disclosed in the prior art. Documents incorporated by reference in the present patent application are to be considered an integral part of the
15 application except that to the extent any terms are defined in these incorporated documents in a manner that conflicts with the definitions made explicitly or implicitly in the present specification, only the definitions in the present specification should be considered.

20

CLAIMS

1. A method, comprising:

for data items that are to be stored in a non-volatile memory in accordance with respective logical addresses, associating the logical addresses with respective physical storage locations in the non-volatile memory, and storing the data items in the respective associated physical storage locations;

receiving a remapping command, which specifies a group of source logical addresses that are associated with respective source physical storage locations;

in response to the remapping command, jointly selecting destination physical storage locations and destination logical addresses for replacing the source physical storage locations and the source logical addresses, respectively, so as to meet a joint performance criterion with respect to the logical addresses and the physical storage locations; and

copying the data items from the source physical storage locations to the respective destination physical storage locations, and re-associating the destination physical storage locations with the respective destination logical addresses.

2. The method according to claim 1, wherein jointly selecting the destination physical storage locations and the destination logical addresses comprises reducing a first number of logical memory fragments occupied by the destination logical addresses relative to the source logical addresses, and reducing a second number of physical memory fragments occupied by the destination physical storage locations, relative to the source physical storage locations.

3. The method according to claim 1, wherein jointly selecting the destination physical storage locations and the destination

logical addresses comprises increasing a throughput of accessing the data items in the non-volatile memory.

4. The method according to claim 1, wherein jointly selecting the destination physical storage locations and the destination logical addresses comprises reducing a latency of accessing the data items in the non-volatile memory.

5. The method according to claim 1, wherein jointly selecting the destination physical storage locations and the destination logical addresses comprises selecting the destination logical addresses in a first contiguous sequence, and selecting the respective destination physical storage locations in a second contiguous sequence.

6. The method according to claim 1, wherein the non-volatile memory comprises multiple memory units, and wherein jointly selecting the destination physical storage locations and the destination logical addresses comprises selecting the destination logical addresses in a contiguous sequence, and selecting the respective destination physical storage locations in cyclical alternation among the multiple memory units.

7. The method according to claim 1, wherein jointly selecting the destination physical storage locations and the destination logical addresses comprises increasing a compressibility of a data structure used for storing respective associations between the logical addresses and the physical storage locations.

8. The method according to claim 1, wherein receiving the remapping command comprises receiving an indication of the destination logical addresses in the command.

9. The method according to claim 1, wherein the remapping command does not indicate the destination logical addresses, and wherein jointly selecting the destination physical storage
5 locations and the destination logical addresses comprises deciding the destination logical addresses in response to receiving the command.

10. The method according to claim 9, and comprising outputting a
10 notification of the decided destination logical addresses.

11. The method according to claim 1, wherein jointly selecting the destination physical storage locations and the destination logical addresses comprises identifying an idle time period, and
15 choosing the destination physical storage locations and the destination logical addresses during the idle time period.

12. Apparatus, comprising:

an interface for communicating with a non-volatile memory;
20 and

a processor, which is configured, for data items that are to be stored in the non-volatile memory in accordance with respective logical addresses, to associate the logical addresses with respective physical storage locations in the non-volatile
25 memory and to store the data items in the respective associated physical storage locations, to receive a remapping command, which specifies a group of source logical addresses that are associated with respective source physical storage locations, to jointly select, in response to the remapping command, destination
30 physical storage locations and destination logical addresses for replacing the source physical storage locations and the source logical addresses, respectively, so as to meet a joint performance criterion with respect to the logical addresses and

the physical storage locations, to copy the data items from the source physical storage locations to the respective destination physical storage locations, and to re-associate the destination physical storage locations with the respective destination logical addresses.

13. The apparatus according to claim 12, wherein, by jointly selecting the destination physical storage locations and the destination logical addresses, the processor is configured to reduce a first number of logical memory fragments occupied by the destination logical addresses relative to the source logical addresses, and to reduce a second number of physical memory fragments occupied by the destination physical storage locations, relative to the source physical storage locations.

15

14. The apparatus according to claim 12, wherein, by jointly selecting the destination physical storage locations and the destination logical addresses, the processor is configured to increase a throughput of accessing the data items in the non-volatile memory.

20

15. The apparatus according to claim 12, wherein, by jointly selecting the destination physical storage locations and the destination logical addresses, the processor is configured to reduce a latency of accessing the data items in the non-volatile memory.

25

16. The apparatus according to claim 12, wherein the processor is configured to select the destination logical addresses in a first contiguous sequence, and to select the respective destination physical storage locations in a second contiguous sequence.

30

17. The apparatus according to claim 12, wherein the non-volatile memory comprises multiple memory units, and wherein the processor is configured to select the destination logical
5 addresses in a contiguous sequence, and to select the respective destination physical storage locations in cyclical alternation among the multiple memory units.

18. The apparatus according to claim 12, wherein, by jointly
10 selecting the destination physical storage locations and the destination logical addresses, the processor is configured to increase a compressibility of a data structure used for storing respective associations between the logical addresses and the physical storage locations.

15

19. The apparatus according to claim 12, wherein the interface is configured to receive an indication of the destination logical addresses in the remapping command.

20. The apparatus according to claim 12, wherein the remapping
20 command does not indicate the destination logical addresses, and wherein the interface is configured to decide the destination logical addresses in response to receiving the command.

25

1/3

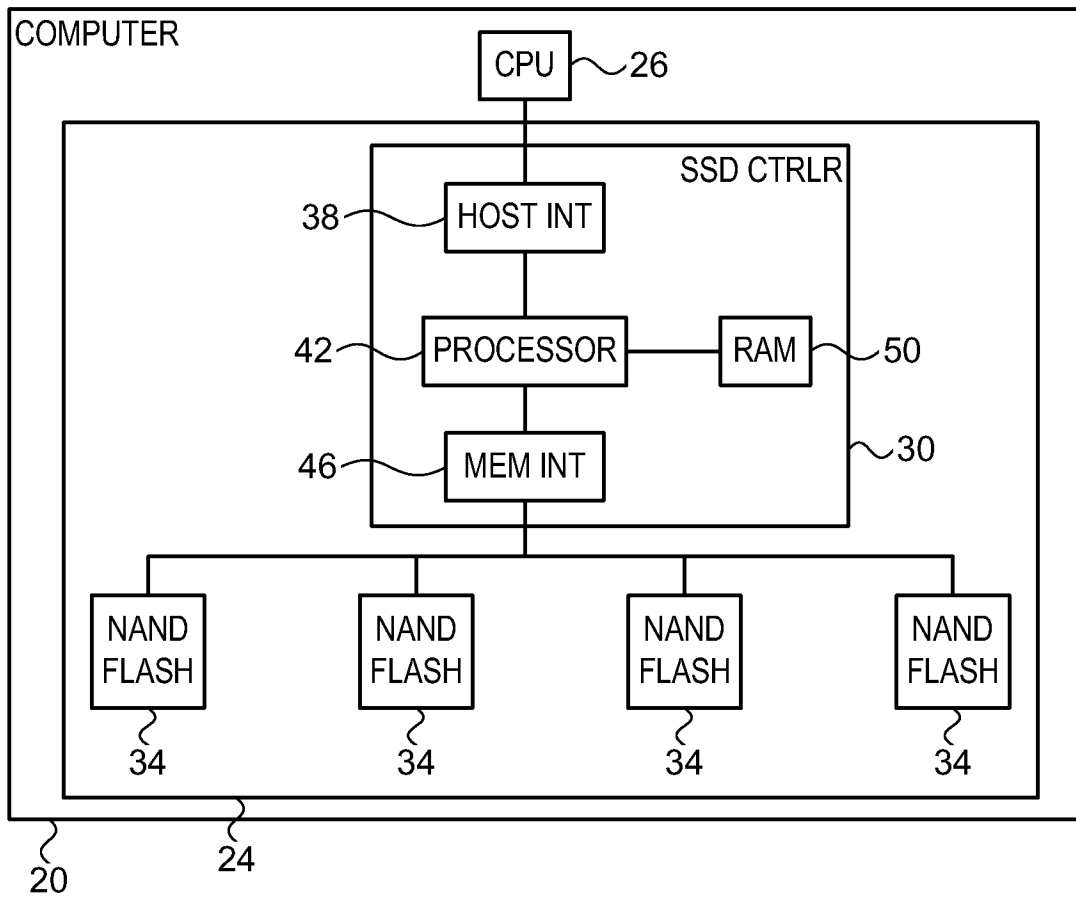


FIG. 1

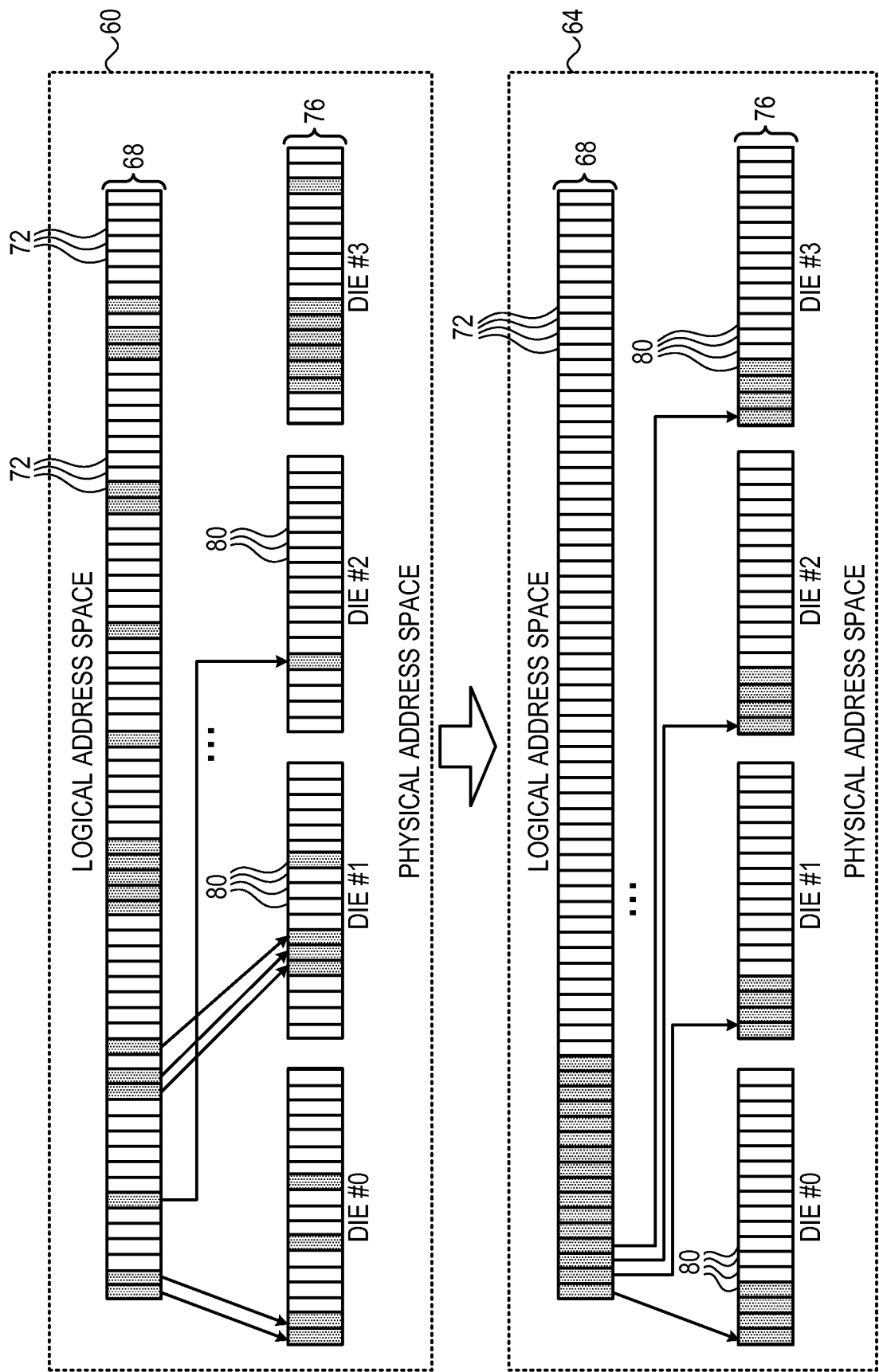


FIG. 2

3/3

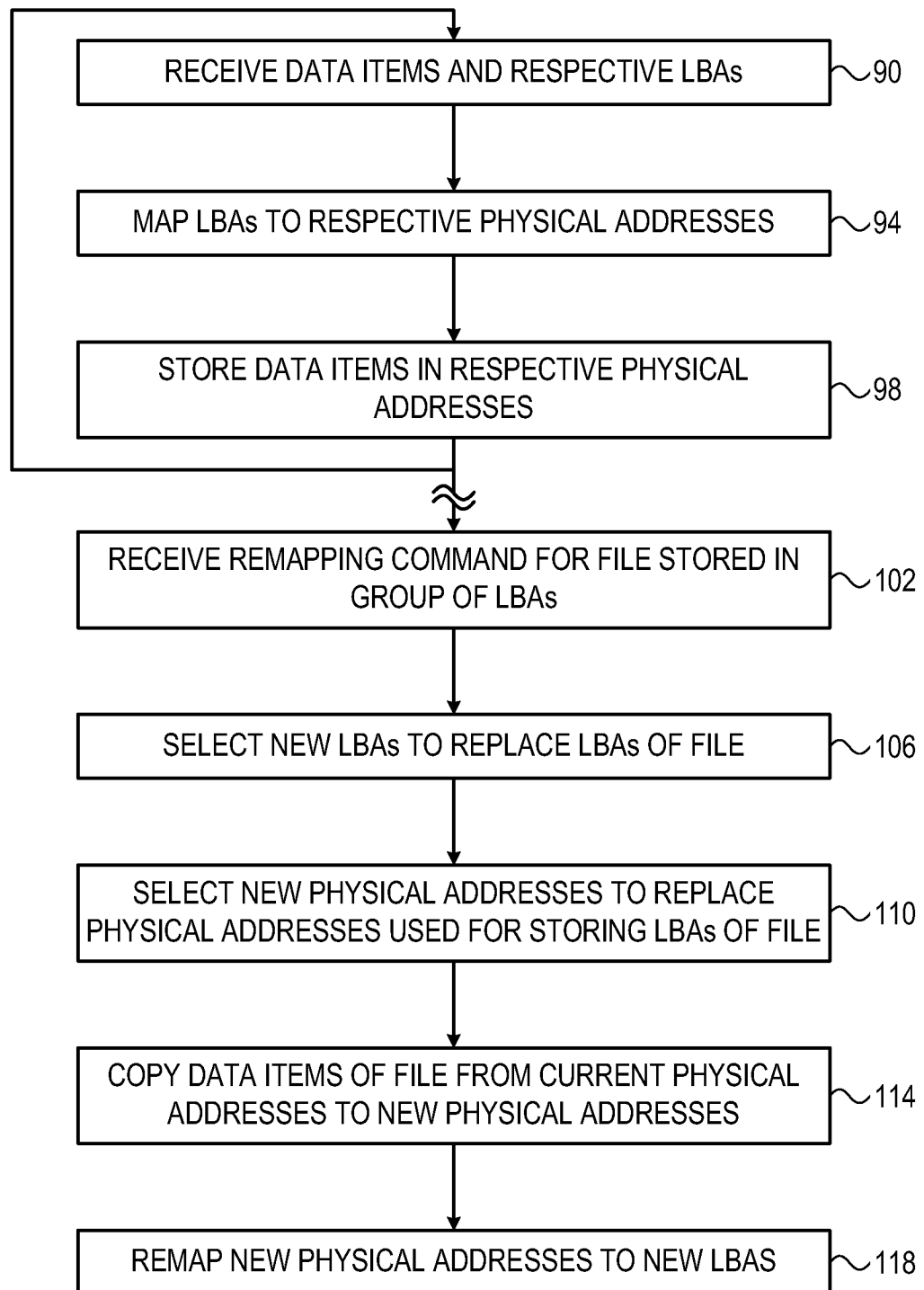


FIG. 3

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2013/069481

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06F3/06 G06F12/02
ADD.

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

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

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

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2011/099326 A1 (JUNG DAWOON [KR] ET AL) 28 April 2011 (2011-04-28) figures 2-17 paragraph [0039] - paragraph [0098] -----	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

"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

13 January 2014

Date of mailing of the international search report

20/01/2014

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Andlauer, J

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2013/069481

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2011099326 A1	28-04-2011	KR 20110046233 A US 2011099326 A1	04-05-2011 28-04-2011
