

STORAGE DEVICE, METHOD FOR OPERATING THE SAME, AND STORAGE SYSTEM INCLUDING STORAGE DEVICES

CROSS-REFERENCE TO RELATED APPLICATION

5 **[0001]** This application claims priority to Korean Patent Application No. 10-2017-0121184 filed on September 20, 2017 in the Korean Intellectual Property Office, the disclosure of which is incorporated by reference in its entirety herein.

TECHNICAL FIELD

10 **[0002]** Exemplary embodiments of the present inventive concept relate to a storage device, a method for operating the same, and a storage system including storage devices.

BACKGROUND

15 **[0003]** Semiconductor memory devices are typically divided into a volatile memory device that fails to retain stored information when the power is interrupted, and a non-volatile memory device that retains stored information even if the power is interrupted. A NAND flash memory device is widely used as a non-volatile memory device.

20 **[0004]** A reading, programming, or erasing operation can be performed on a memory bank of a NAND flash memory device in response to corresponding commands. The delay time for the operation is represented by the sum of an execution time of the operation and a waiting time until a previous operation in the same memory bank has completed. However, the delay time of the operation may adversely affect a user's perception of the performance of the memory device. Thus, it is desirable to reduce latency in responding to read, program and erase commands.

25

SUMMARY

[0005] At least one embodiment of the present inventive concept provides a storage device including non-volatile memory devices each storing mirrored data.

30 **[0006]** At least one embodiment of the present inventive concept also provides a method for operating a storage device including non-volatile memory devices each storing mirrored data.

[0007] At least one embodiment of the present inventive concept also provides a storage system including storage devices each storing mirrored data.

[0008] According to an exemplary embodiment of the present inventive concept, a storage device includes a first non-volatile memory comprising a plurality of first banks having a

plurality of first addresses assigned thereto according to a first assignment policy, a second non-volatile memory comprising a plurality of second banks having a plurality of second addresses assigned thereto according to a second assignment policy different from the first assignment policy, wherein the plurality of second addresses corresponds to the plurality of first addresses, and wherein the second non-volatile memory mirrors data items stored in the first addresses to store them in the second addresses, and a controller configured to receive a command from a host to control the first non-volatile memory and the second non-volatile memory, wherein the controller provides a read command received from the host simultaneously to the first non-volatile memory and the second non-volatile memory, and outputs to the host an earlier one between data provided from the first non-volatile memory and data provided from the second non-volatile memory based on the read command.

[0009] According to an exemplary embodiment of the present inventive concept, a storage system includes a first storage device comprising a plurality of first non-volatile memories having a plurality of first addresses assigned thereto according to a first assignment policy, a second storage device comprising a plurality of second non-volatile memories having a plurality of second addresses assigned thereto according to a second assignment policy different from the first assignment policy, wherein the plurality of second addresses corresponds to the plurality of first addresses, and wherein the second storage device mirrors data items stored in the first addresses to store them in the second addresses, and a host configured to control the first storage device and the second storage device, wherein the host issues a read request simultaneously to the first storage device and the second storage device, and outputs an earlier one between data provided from the first storage device and data provided from the second storage device.

[0010] According to an exemplary embodiment of the present inventive concept, a method for operating a storage device includes a first non-volatile memory comprising a plurality of first banks having a plurality of first addresses assigned thereto according to a first assignment policy, and a second non-volatile memory comprising a plurality of second banks having a plurality of second addresses assigned thereto according to a second assignment policy different from the first assignment policy, wherein the plurality of second addresses corresponds to the plurality of first addresses, and wherein the second non-volatile memory mirrors data items stored in the first addresses to store them in the second addresses, the method comprising a host issuing a read request to the storage device, a controller of the storage device applying a read command simultaneously to the first non-volatile memory and the second non-volatile memory based on the read request, and the controller outputting to

the host an earlier one between data provided from the first non-volatile memory and data provided from the second non-volatile memory based on the read command.

[0011] According to an exemplary embodiment of the inventive concept, a storage device includes: a first non-volatile memory comprising M rows of N first banks having a plurality of first addresses, a second non-volatile memory comprising M rows of N second banks having a plurality of second addresses, and controller. The plurality of second addresses corresponds to the plurality of first addresses. The second non-volatile memory mirrors data items stored in the first addresses to store them in the second addresses. The controller is configured to provide a read command received from a host simultaneously to the first non-volatile memory and the second non-volatile memory, and output to the host an earlier one between data provided from the first non-volatile memory and data provided from the second non-volatile memory based on the read command. The controller shifts the first addresses belonging to a given row of the first banks by $M \bmod N$ banks to determine the corresponding second banks of the second addresses.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] The present inventive concept will become more apparent by describing in detail exemplary embodiments thereof with reference to the attached drawings, in which:

[0013] FIG. 1 is a block diagram for illustrating a storage device according to an exemplary embodiment of the present inventive concept;

[0014] FIG. 2 is a block diagram for illustrating the storage device shown in FIG. 1 according to an exemplary embodiment of the present inventive concept;

[0015] FIG. 3 is a block diagram for illustrating the controller of FIG. 1 according to an exemplary embodiment of the present inventive concept;

[0016] FIG. 4A is a diagram for conceptually illustrating addresses assigned to a plurality of first banks included in the first non-volatile memory;

[0017] FIG. 4B is a diagram for conceptually illustrating addresses assigned to a plurality of second banks included in the second non-volatile memory;

[0018] FIG. 5 is a flowchart illustrating an operation of a storage device according to an exemplary embodiment of the present inventive concept;

[0019] FIG. 6 is a diagram for conceptually illustrating an operation of the controller when a data read request is provided to the first non-volatile memory and the second non-volatile memory according to an exemplary embodiment of the present inventive concept;

[0020] FIG. 7A is a diagram for conceptually illustrating addresses and banks accessing the first non-volatile memory in response to a data read request;

[0021] FIG. 7B is a diagram for conceptually illustrating addresses and banks accessing the second non-volatile memory in response to a data read request according to an exemplary embodiment of the present inventive concept;

[0022] FIG. 8 is a timing diagram for illustrating an operation of the storage device according to an exemplary embodiment of the present inventive concept;

[0023] FIG. 9 is a flowchart illustrating an operation of a storage device according to an exemplary embodiment of the present inventive concept;

[0024] FIGS. 10A and 10B are diagrams for illustrating inter-process communications (IPC) between the first non-volatile memory and the second non-volatile memory according to an exemplary embodiment of the present inventive concept;

[0025] FIG. 11 is a block diagram for illustrating operation time synchronization of a first non-volatile memory and a second non-volatile memory according to an exemplary embodiment of the present inventive concept;

[0026] FIGS. 12A and 12B are tables for illustrating operation time synchronization between the first non-volatile memory and the second non-volatile memory according to an exemplary embodiment of the present inventive concept;

[0027] FIG. 13 is a flowchart for illustrating an erase/program operation of a first non-volatile memory and a second non-volatile memory according to an exemplary embodiment of the present inventive concept; and

[0028] FIG. 14 is a block diagram for illustrating a storage system according to an exemplary embodiment of the present inventive concept.

DETAILED DESCRIPTION OF THE EXEMPLARY EMBODIMENTS

[0029] Hereinafter, a storage device according to an exemplary embodiment of the present inventive concept, and a storage system including the same will be described with reference to FIGS. 1 to 14.

5 [0030] FIG. 1 is a block diagram for illustrating a storage device according to an exemplary embodiment of the present inventive concept. FIG. 2 is a block diagram of the storage device shown in FIG. 1 in more detail.

[0031] Referring to FIGS. 1 and 2, a storage device 100 according to an exemplary embodiment of the present inventive concept includes a controller 110, a first non-volatile
10 memory 120_1, and a second non-volatile memory 120_2.

[0032] The storage device 100 may be connected to a host. The storage device 100 may receive a read command READ, a program command PROGRAM, and an erase command ERASE from the host and may perform an operation associated with each of the commands.

[0033] The storage device 100 may be, for example, a solid-state drive (SSD) composed of
15 the first non-volatile memory 120_1 and the second non-volatile memory 120_2.

[0034] Alternatively, the storage device 100 may be, but is not limited to, a memory card such as a PC card or PCMCIA (personal computer memory card international association) card, a compact flash (CF) card, a smart media (SM) card or SMC, a memory stick, a multimedia card (MMC, RS-MMC or MMCmicro), an SD card, (SD, miniSD, microSD or
20 SDHC) and a universal flash storage (UFS).

[0035] The controller 110 (e.g., a control circuit) may control the operation of the first non-volatile memory 120_1 and the second non-volatile memory 120_2. The controller 110 may control the first non-volatile memory 120_1 and the second non-volatile memory 120_2 based on a command received from a host. As will be described below, the first non-volatile
25 memory 120_1 and the second non-volatile memory 120_2 are mirrored to each other. Accordingly, the controller 110 may provide the command received from the host equally to the first non-volatile memory 120_1 and the second non-volatile memory 120_2.

[0036] FIG. 3 is a block diagram illustrating the controller of FIG. 1 according to an exemplary embodiment of the inventive concept.

30 [0037] Referring to FIG. 3, the controller 110 includes a command queue 111, a command decoder 112, and an address assignment map 113.

[0038] The command queue 111 may temporarily store commands received by the controller 110 from the host and addresses on the first and second non-volatile memories 120_1 and 120_2 that the commands are to access. Accordingly, the command queue 111 may include

a storage element (e.g., storage device, registers, buffers, etc.). The command queue 111 may provide the stored commands and addresses to the command decoder 112.

[0039] The command decoder 112 may decode commands and addresses received from the command queue 111. The command decoder 112 may provide decoded commands and addresses to the address assignment map 113. In an embodiment, the command decoder 112 is implemented by a decoder circuit.

[0040] The host may provide the controller 110 with logical addresses for accessing the first and second non-volatile memories 120_1 and 120_2. The logical addresses may include, for example, logical page numbers (LPNs).

[0041] In an embodiment, the controller 110 converts the logical addresses received from the host into actual physical addresses for accessing the first memory cell 130 and the second memory cell 140. The physical addresses may include, for example, physical page numbers (PPNs).

[0042] In an embodiment, the command decoder 112 converts the logical addresses received from the host to the physical addresses by using the address assignment map 113, and provides the converted physical addresses to the first and second non-volatile memories 120_1 and 120_2. The command decoder 112 may additionally provide a command (COM) to the non-volatile memories 120_1 and 120_2 corresponding to the converted physical addresses.

[0043] The address assignment map 113 includes a first map 114 and a second map 115 for use in converting a logical address received from a host into a physical address.

[0044] In the first map 114, the relationship between the physical addresses of the first non-volatile memory 120_1 and the logical addresses may be defined, for example, in the form of a table. Likewise, in the second map 115, the relationship between the physical addresses of the second non-volatile memory 120_1 and the logical addresses may be defined in the form of a table.

[0045] The command decoder 112 provides a first address Addr1 received from the host to the address assignment map 113. As described above, the first address Addr1 may include one or more logical addresses for accessing the first non-volatile memory 120_1 and the second non-volatile memory 120_2.

[0046] The address assignment map 113 converts the first address Addr1 into the second address Addr2 using the first map 114. The second address Addr2 may include one or more physical addresses for accessing the first non-volatile memory 120_1. Similarly, the address assignment map 113 converts the first address Addr1 into the third address Addr3 using the

second map 115. The third address Addr3 may be one or more physical addresses for accessing the second non-volatile memory 120_2.

[0047] The command decoder 112 may provide the second address Addr2 and the third address Addr3 received from the address assignment map 113 to the first non-volatile memory 120_1 and the second non-volatile memory 120_2, respectively.

[0048] In summary, the controller 110 may convert the first address Addr1 included in a data access request received from the host into the second address Addr2 for accessing the first non-volatile memory 120_1, and may convert the first address Addr1 into the third address Addr3 for accessing the second non-volatile memory 120_2. In an embodiment, the two maps 114 and 115 are replaced with a single map that include entries indexable by the first address Addr1, where each entry include the corresponding second address Addr2 and third address Addr3.

[0049] The controller 110 may provide the data access request received from the host simultaneously to the first non-volatile memory 120_1 and the second non-volatile memory 120_2. For example, if the host issues a data read command for the first address Addr1, the controller 110 provides a read command for the second address Addr2 to the first non-volatile memory 120_1 and provides a read command for the third address Addr3 to the second non-volatile memory 120_2. The controller 110 may provide the read command simultaneously to the first non-volatile memory 120_1 and the second non-volatile memory, and may return read data from the first non-volatile memory 120_1 or read data from the second non-volatile memory 120_2, whichever has been completed earlier.

[0050] Referring again to FIG. 2, the first non-volatile memory 120_1 includes a first NVM controller 125_1 (e.g., a memory controller or a control circuit) and a first memory cell 130. The first NVM controller 125_1 may control the operation of the first memory cell 130 based on a command received from the controller 110. The first NVM controller 125_1 is connected to a first bus 135_1.

[0051] The first memory cell 130 may include, but is not limited to, NAND flash memory cells. That is, the first memory cell 130 may include a non-transitory memory cell that is capable of storing data on a semiconductor device, and any memory cell may be employed as long as it can cause a delay due to mutual interference of read, program and erase operations on the same bank.

[0052] In the following description, the first memory cell 130 is described as including a NAND flash memory cell. Although not explicitly shown in FIG. 2, the first memory cell 130 may be, but is not limited to, comprised of a plurality of NAND flash memory chips (i.e.,

integrated memory chips packaged into a single package). The first memory cell 130 may include a single NAND flash memory chip.

[0053] When the first memory cell 130 includes NAND flash memory cells, the first memory cell 130 may include a single-level cell (SLC). A SLC stores one bit of information per memory cell. Although a SLC has smaller capacity per memory cell than a multi-level cell (MLC) or a triple-level cell (TLC), it may have a lower latency required for accessing data than the MLC and the TLC. It is to be understood that the first memory cell 130 is not limited to the SLC.

[0054] The first non-volatile memory 120_1 includes a plurality of banks 130_1, 130_2, ..., 130_n. Each of the plurality of banks 130_1 to 130_n is a unit of address with which the controller 110 performs a read, a program or an erase operation.

[0055] The second non-volatile memory 120_2 includes a second NVM controller 125_2 and a second memory cell 140. The second NVM controller 125_2 may control the operation of the second memory cell 140 based on a command received from the controller 110. The second NVM controller 125_2 is connected to a second bus 135_2.

[0056] The second memory cell 140 may have the same configuration as the first memory cell 130. Thus, the second memory cell 140 may include a non-transitory memory cell that is capable of storing data on a semiconductor device, such as a NAND flash memory cell. Likewise, the second memory cell 140 is not limited to a NAND flash cell, and may include any memory device as long as it can cause a delay due to mutual interference of read, program and erase operations on the same bank.

[0057] In the following description, the second memory cell 140 will be described as a NAND flash memory cell like the first memory cell 130 for convenience of discussion.

[0058] In an embodiment, the first memory cell 130 and the second memory cell 140 are mirrored to each other. Accordingly, the first memory cell 130 and the second memory cell 140 stores the same data. The controller 110 may access areas on the first memory cell 130 and the second memory cell 140 in which the same data is stored according to the command and address provided from the host, and may perform the same command for the first memory cell 130 and the second memory cell 130. In an embodiment, a first part of the data stored in the first memory cell 130 in a first location is stored in the second memory cell 140 in a second location different from the first location, and a second part of the data is a stored in a same third location in both the second memory cell 130 and the third memory cell 140.

[0059] FIG. 4A is a diagram for conceptually illustrating addresses assigned to a plurality of first banks included in the first non-volatile memory. FIG. 4B is a diagram for conceptually

illustrating addresses assigned to a plurality of second banks included in the second non-volatile memory.

[0060] First, referring to FIG. 4A, the first non-volatile memory 120_1 includes a plurality of banks 130_0, 130_1, 130_2, and 130_3. Although the first non-volatile memory 120_1 is shown as including four banks of the zeroth bank 130_0 to the third bank 130_3 for convenience of illustration, it is to be understood that this is merely illustrative. For example, the first non-volatile memory 120_1 may include fewer or more than four banks.

[0061] In addition, the first non-volatile memory 120_1 may be divided into a plurality of rows R1_0, R1_1, R1_2, R1_3, R1_4. Each of the plurality of banks 130_0, 130_1, 130_2, and 130_3 may include respective ones of logical addresses 0 to 19 each assigned to one of the plurality of rows R1_0 to R1_4.

[0062] For example, addresses 0, 4, 8, 12 and 16 are included in the zeroth bank 130_0 of the first non-volatile memory 120_1. Addresses 1, 5, 9, 13 and 17 are included in the first bank 130_1. Addresses 2, 6, 10, 14 and 18 are included in the second bank 130_2. Addresses 3, 7, 11, 15 and 19 are included in the third bank 130_3. In this manner, the logical addresses of the first non-volatile memory 120_1 may be sequentially assigned to the plurality of banks 130_0 to 130_3.

[0063] Among the logical addresses assigned to the plurality of banks 130_0 to 130_3, the address 4 may be represented by the physical address of, for example, the first row of the zeroth bank (BANK0, #1). In another example, the address 11 may be represented by the physical address of the second row of the third bank (BANK3, #2).

[0064] As described above, the first non-volatile memory 120_1 performs read, program and erase commands bank-by-bank. If a read command for addresses 0 and 4 are received from the host, the first non-volatile memory 120_1 cannot process it at once. That is, the first non-volatile memory 120_1 has to read the data in the zeroth bank 130_0 over two data cycles and then provide the data to the host. This is a so-called bank collision. For example, the first non-volatile memory 120_1 reads the data associated with address 0 in the zeroth bank 130_0 during a first data cycle and reads the data associated with address 4 in the zeroth bank 130_0 during a second data cycle after the read associated with address 0 has completed.

[0065] On the contrary, let us assume that a read command for the data stored in address 0 and address 1 is received in the first non-volatile memory 120_1 from the host. Since the address 0 is assigned to the zeroth bank 130_0 and the address 1 is assigned to the first bank

130_1, the first non-volatile memory 120_1 may read the data stored in the address 0 and the address 1 in a single data cycle and provide the data to the host.

[0066] In a typical NAND flash memory device, it is known that a memory cell processes read and program requests page by page. Thus, as described herein, processing a read request bank-by-bank in the first and second non-volatile memories 120_1 and 120_2 may be interpreted as processing a read request page-by-page. It is to be understood that this is merely illustrative and the plurality of banks included in the first and second non-volatile memories 120_1 and 120_2 may refer to the plurality of pages.

[0067] Referring to FIG. 4B, the second non-volatile memory 120_2 has a configuration similar to that of the first non-volatile memory 120_1. It is to be understood that the second non-volatile memory 120_2 mirrors and stores the data stored in the first non-volatile memory 120_1.

[0068] Specifically, the second non-volatile memory 120_2 includes the same number of banks as the first non-volatile memory 120_1, and the same number of logical addresses assigned to the banks as the first non-volatile memory 120_1. However, the assignment policy for assigning logical addresses to the same banks is different. For example, when different assignment policies are used for different non-volatile memories, a given read request applied to both non-volatile memories may result in fewer bank collisions in one of the non-volatile memories.

[0069] Although the second non-volatile memory 120_2 is shown in FIG. 4B as including four banks of the zeroth bank 140_0 to the third bank 140_3 for convenience of illustration, it is to be understood that this is merely illustrative. For example, the second non-volatile memory 120_2 may include fewer or more than four banks.

[0070] The second non-volatile memory 120_2 is divided into a plurality of rows R2_0 to R2_4. Each of the plurality of banks 140_0, 140_1, 140_2, and 140_3 may include respective ones of logical addresses 0 to 19 each assigned to one of the plurality of rows R2_0 to R2_4.

[0071] For example, addresses 0, 7, 10, 13, 12 and 16 are included in the zeroth bank 140_0 of the second non-volatile memory 120_2. Addresses 1, 4, 11, 14 and 17 are included in the first bank 140_1. Addresses 2, 5, 8, 15 and 18 are included in the second bank 140_2. Addresses 3, 6, 9, 12 and 19 are included in the third bank 140_3.

[0072] Among the logical addresses assigned to the plurality of banks 140_0 to 140_3, the address 4 may be represented by the physical address of, for example, the first row of the first

bank (BANK1, #1). For another example, the address 12 may be represented by the physical address of the third row of the third bank (BANK3, #3).

[0073] Compared to the logical addresses assigned to the plurality of banks 130_0 to 130_3 of the first non-volatile memory 120_1, the logical addresses of the second non-volatile memory 120_2 assigned to the plurality of banks 140_0 to 140_3 have been shifted to neighboring banks by several logical addresses.

[0074] Specifically, the addresses 0 to 3 in the zeroth row R2_0 of the second non-volatile memory 120_2 are sequentially assigned to the zeroth to third banks 140_0 to 140_3. The addresses 4 to 7 in the first row R2_1 are shifted by one bank to be assigned to the next banks. The addresses 8 to 11 in the second row R2_2 are shifted by two banks to be assigned to the adjacent banks. The addresses 13 and 14 in the third row R2_3 are shifted by three banks to be assigned to the adjacent banks. To generalize it, when a non-volatile memory includes n banks, addresses belonging to the m^{th} row are shifted by $m \bmod n$ banks to be assigned, where mod is the modulo operation.

[0075] As such, the addresses included in the second non-volatile memory 120_2 may be assigned to the plurality of banks 140_1 to 140_3 in a manner different from the manner that addresses of the first non-volatile memory 120_1 are assigned to the plurality of banks 130_1 to 130_3.

[0076] Although the bank assignment policy of addresses included in the second non-volatile memory 120_2 has been described above with reference to FIG. 4B, the present disclosure is not limited thereto. The addresses included in the second non-volatile memory 120_2 may be assigned to the plurality of banks 140_0 to 140_3 according to a different policy as long as it has sequentiality.

[0077] The first non-volatile memory 120_1 and the second non-volatile memory 120_2 have different address assignment policies, which relates to the method for operating a storage device according to an exemplary embodiment of the present inventive concept. The operation of the storage device will be described in more detail with reference to FIGS. 5 to 7B.

[0078] FIG. 5 is a flowchart illustrating an operation of a storage device according to an exemplary embodiment of the present inventive concept. FIG. 6 is a diagram for conceptually illustrating an operation of the controller when a data read request is provided to the first non-volatile memory and the second non-volatile memory according to an exemplary embodiment of the present inventive concept. FIG. 7A is a diagram for conceptually illustrating addresses and banks accessing the first non-volatile memory in response to a data

read request. FIG. 7B is a diagram for conceptually illustrating addresses and banks accessing the second non-volatile memory in response to a data read request according to an exemplary embodiment of the present inventive concept.

[0079] Referring to FIG. 5, the storage device 100 according to an exemplary embodiment of the present inventive concept sends a read request to the first non-volatile memory 120_1 and the second non-volatile memory 120_2 simultaneously (step S110). Then, the data received earlier from the first non-volatile memory 120_1 is provided to the host (step S120), while the data received later from the second non-volatile memory 120_2 is ignored (step S130).

[0080] Typically, data mirroring is used for data backup purposes. In the storage device according to an exemplary embodiment of the present inventive concept, by providing a read command simultaneously to the first non-volatile memory 120_1 and the second non-volatile memory 120_2 that mirrors the first non-volatile memory 120_1 (step S110), the latency due to the read request can be improved. In an embodiment, the controller 110 does not provide the ignored data to the host. If the data is instead received earlier from the second non-volatile memory 120_2, this data can be provided to the host, while the data received later from the first non-volatile memory 120_1 is ignored.

[0081] Referring to FIG. 6, the operation of the command decoder 112 and the address assignment map 113 is shown when the controller 110 receives a read command for the logical addresses 0, 4, 8, 12 and 16 from the host.

[0082] The command decoder 112 provides a first addresses Addr1 including the addresses 0, 4, 8, 12 and 16 to the address assignment map 113. The address assignment map 113 returns to the command decoder 112 the second address Addr2 converted from the first address Addr1 using the first map 114. FIG. 6 shows an example in which the address assignment map 113 returns physical addresses corresponding to the zeroth to fourth rows of the zeroth bank (BANK0, #0 to #4).

[0083] The address assignment map 113 returns to the command decoder 112 the third address Addr3 converted from the first address Addr1 using the second map 115. FIG. 6 shows an example in which the address assignment map 113 returns physical addresses of the zeroth row of the zeroth bank (BANK0, #0), the first row of the first bank (BANK1 #1), the second row of the second bank (BANK2, #2), the third row or and the third bank (BANK3, #3), and the fourth row of the zeroth bank (BANK0, #4).

[0084] The controller 110 may access the first non-volatile memory 120_1 and the second non-volatile memory 120_2 based on the physical addresses provided to the command decoder 112 from the address assignment map 113.

[0085] Referring to FIG. 7A, the logical addresses 0, 4, 8, 12 and 16 are assigned to the zeroth bank 130_0. Since the physical addresses corresponding to the logical addresses 0, 4, 8, 12 and 16, respectively, are located in the zeroth to 4th rows of the zeroth bank (BANK0, # 0 to # 4), at least four bank collisions occur for the read request from the host. As a result, the latency for the read request increases, which may adversely affect data parallelism.

[0086] Referring to FIG. 7B, the logical addresses 0, 4, 8, 12 and 16 are assigned across the zeroth to third banks 130_0 to 130_3 appropriately. In the example shown in FIG. 7B, only one data collision occurs for the read request from the host. If the second non-volatile memory 120_2 also assigned the logical addresses to the plurality of banks 140_1 to 140_3 according to the same policy, there may be four bank collisions like in the first non-volatile memory 120_1.

[0087] Accordingly, the second non-volatile memory 120_2 may first process the read request from the host and provide the data read earlier to the controller 110. The controller 110 may provide the data received from the second non-volatile memory 120_2 to the host. On the other hand, the data read from the first non-volatile memory 120_1 may be ignored by the controller 110.

[0088] It is to be understood that the above example is ideal for the second non-volatile memory 120_2. If the host issues a read command for different logical addresses, there may be a different latency.

[0089] However, according to an exemplary embodiment of the present inventive concept, the second non-volatile memory 120_2 that mirrors the first non-volatile memory 120_1 employs a policy for assigning logical addresses to the plurality of banks 140_1 to 140_3 that is different from the policy of the first non-volatile memory 120_1, thereby suppressing bank collisions for the same read request, less than in the first non-volatile memory 120_1.

[0090] Accordingly, the controller 110 can return the data read from the first non-volatile memory 120_1 or the data read from the second non-volatile memory 120_2, whichever is received earlier, in response to the read request from the host. In other words, the controller 110 can provide the host with data received from the first non-volatile memory 120_1 or the data received from the second non-volatile memory 120_2, whichever has less bank collisions.

[0091] The latency of the first non-volatile memory 120_1 and the second non-volatile memory 120_2 with respect to the read request from the host may be determined based on other factors, in addition to the bank collision described above. In a storage device according to an exemplary embodiment of the present inventive concept, it is assumed that

the processing of the read request is delayed by a previously processed erase or program request.

[0092] FIG. 8 is a timing diagram for illustrating an operation of the storage device according to an exemplary embodiment of the present inventive concept.

5 **[0093]** Referring to FIG. 8, an example is shown in which there is a read request for the zeroth bank 130_0 of the first non-volatile memory 120_1 and the zeroth bank 140_0 of the second non-volatile memory 120_2. Prior to each read request, an erase or program command from the host is sent to the storage device 100, and the storage device 100 may process it.

10 **[0094]** For example, it is assumed that a read request is received from the host to the zeroth banks 130_0 and 140_0 of the first non-volatile memory 120_1 and the second non-volatile memory 120_2 at a second time point t2, and the first non-volatile memory 120_1 and the second non-volatile memory 120_2 are processing an erase command for the zeroth banks 130_0 and 140_0 from the host at the second time point t2.

15 **[0095]** As described above, since the first non-volatile memory 120_1 cannot simultaneously process different commands for the same bank, the first non-volatile memory 120_1 cannot process the read command until the erase command for the zeroth bank has completed. Likewise, the second non-volatile memory 120_2 cannot simultaneously process different commands for the same bank, the second non-volatile memory 120_2 cannot process the read
20 command until the erase command for the zeroth bank has completed.

[0096] In order to avoid a situation in which the two non-volatile memories 120_1 and 120_2 are waiting for the completion of processing of the previous erase/program command for the same bank, the first non-volatile memory 120_1 and the second non- 2 non-volatile memory 120_2 process the erase/program command mutually exclusively. For example,
25 while the first non-volatile memory 120_1 is processing an erase command for the zeroth bank, the second non-volatile memory 120_2 does not process the erase command for the zeroth bank.

[0097] Specifically, as shown in FIG. 8, the first non-volatile memory 120_1 starts to process an erase command C1_0 at a first time point t1, and completes the processing of the
30 erase command C1_0 at the second time point t2. The second non-volatile memory 120_2 starts to process an erase command C1_1 at the second time point t2 and completes the processing of the erase command C1_1 at the third time point t3. The erase command C1_0 issued to the first non-volatile memory 120_1 is identical to the erase command C1_1 issued

to the second non-volatile memory 120_2, and includes an erase request for the same logical address.

[0098] If a read command from the host is issued at the second time point t2, the second non-volatile memory 120_2, which has not completed the processing of the erase command C1_1, cannot start the processing of the read command at the second time point t2. On the other hand, the first non-volatile memory 120_1, which has completed the processing of the erase command C1_0, can start and complete the processing of the read command earlier.

[0099] Hereinafter, the operation of the storage device 100 in which the first non-volatile memory 120_1 and the second non-volatile memory 120_2 mutually exclusively process an erase/program command issued from the host will be described.

[00100] FIG. 9 is a flowchart illustrating an operation of a storage device according to an exemplary embodiment of the present inventive concept.

[00101] Referring to FIG. 9, it is checked whether the erase/program command is being processed in the first non-volatile memory 120_1 (step S210). When the erase/program command is being processed in the first non-volatile memory 120_1, the second non-volatile memory 120_2 waits without processing the erase/program command (step S220). If the erase/program command is not being processed in the first non-volatile memory 120_1 or the processing has completed, the second non-volatile memory 120_2 processes the erase/program command (step S230).

[00102] The operations of the first non-volatile memory 120_1 and the second non-volatile memory 120_2 described in FIG. 9 are merely illustrative. That is, as shown in FIG. 9, the second non-volatile memory 120_2 can check whether the first non-volatile memory 120_1 is processing the erase/program command. It is to be understood that the first non-volatile memory 120_1 can also check whether the second non-volatile memory 120_2 is processing the erase/program command.

[00103] FIGS. 10A and 10B are diagrams for illustrating inter-process communications (IPC) between the first non-volatile memory and the second non-volatile memory according to an exemplary embodiment of the present inventive concept.

[00104] Referring to FIG. 10A, the second non-volatile memory 120_2 may use the IPC to check whether the first non-volatile memory 120_1 is processing the erase/program command. Specifically, a channel is formed between the first NVM controller 125_1 and the second NVM controller 125_2, such that it may be checked whether the first non-volatile memory 120_1 or the second non-volatile memory 120_2 is processing an erase/program command via the channel.

[00105] For example, while the first non-volatile memory 120_1 is processing an erase/program command, the first NVM controller 125_1 sends a busy signal to the second NVM controller 125_2. The first NVM controller 125_1 stops sending the busy signal when the processing of the erase/program command has completed. While the first NVM controller 125_1 is sending the busy signal, the second NVM controller 125_2 recognizes that the first non-volatile memory 120_1 is processing an erase/program command and waits until the processing of the erase/program command has completed.

[00106] Likewise, while the second non-volatile memory 120_2 is processing an erase/program command, the second NVM controller 125_2 sends a busy signal to the first NVM controller 125_1. The second NVM controller 125_2 stops sending the busy signal when it completes the processing of the erase/program command. While the second NVM controller 125_2 is sending the busy signal, the first NVM controller 125_1 recognizes that the second non-volatile memory 120_2 is processing an erase/program command and waits until the processing of the erase/program command has completed.

[00107] The IPC between the first non-volatile memory 120_1 and the second non-volatile memory 120_2 described above may be performed without the controller 110, for example. For example, the IPC may be handled by the controllers 125_1 and 125_2.

[00108] Referring to FIG. 10B, the storage device 100 further includes an internal memory 150 for the IPC in an alternate embodiment. The internal memory 150 stores a busy flag indicating whether the first non-volatile memory 120_1 or the second non-volatile memory 120_2 is processing an erase/program command.

[00109] For example, while the first non-volatile memory 120_1 is processing an erase/program command, the first NVM controller 125_1 sets a busy flag to one. In an embodiment, the first NVM controller 125_1 sets the busy flag to zero when it completes the processing of the erase/program command. The second NVM controller 125_2 checks the busy flag before processing an erase/program command. While the busy flag is set to one, the second NVM controller 125_2 recognizes that the first non-volatile memory 120_1 is processing an erase/program command and waits until the processing of the erase/program command has completed.

[00110] For example, while the second non-volatile memory 120_2 is processing an erase/program command, the second NVM controller 125_1 sets the busy flag to one. The second NVM controller 125_2 sets the busy flag to zero when the processing of the erase/program command has completed. The first NVM controller 125_1 checks the busy flag before processing an erase/program command. While the busy flag is set to one, the

first NVM controller 125_1 recognizes that the second non-volatile memory 120_2 is processing an erase/program command and waits until the processing of the erase/program command has completed. In an embodiment, a mechanism is used to prevent the controllers 125_1 and 125_2 from simultaneously setting the busy flag. For example, the timing at which the controllers 125_1 and 125_2 attempt to set the busy flag can be adjusted so they cannot simultaneously set the busy flag.

[00111] Likewise, the IPC between the first non-volatile memory 120_1 and the second non-volatile memory 120_2 using the internal memory 150 may be performed without the controller 110, for example.

[00112] FIG. 11 is a block diagram for illustrating an operation time synchronization of a first non-volatile memory and a second non-volatile memory according to an exemplary embodiment of the present inventive concept. FIGS. 12A and 12B are tables for illustrating the operation time synchronization between the first non-volatile memory and the second non-volatile memory according to an exemplary embodiment of the present inventive concept.

[00113] Referring to FIGS. 11 to 12B, an operation of processing an erase/program command received from a host exclusively by the first non-volatile memory 120_1 or the second non-volatile memory 120_2 of the storage device 100 according to an exemplary embodiment of the present inventive concept will be described.

[00114] The storage device 100 includes a synchronization timer 160 (e.g., a timer circuit). The synchronization timer 160 provides a time synchronization signal to the first non-volatile memory 120_1 and the second non-volatile memory 120_2. The time synchronization signal provided by the synchronization timer 160 to the first and second non-volatile memories 120_1 and 120_2 may be, for example, a clock signal, and more specifically, may be a real-time clock (RTC) signal.

[00115] The first NVM controller 125_1 receives a time synchronization signal from the synchronization timer 160. The first NVM controller 125_1 starts the processing of the erase/program command based on the time synchronization signal.

[00116] The second NVM controller 125_2 receives a time synchronization signal from the synchronization timer 160. The second NVM controller 125_2 starts the processing of the erase/program command based on the time synchronization signal.

[00117] FIG. 12A depicts a table showing an example of start time points of the processing of erase commands by the first non-volatile memory 120_1 and the second non-volatile memory 120_2. The unit of the numbers is milliseconds (ms). The first and second

non-volatile memories 120_1 and 120_2 process the erase command after times indicated by the numbers have elapsed from a reference time point. It is assumed that it takes 2 ms for the first and second non-volatile memories 120_1 and 120_2 to process the erase command.

[00118] The first non-volatile memory 120_1 starts the processing of the erase command after the times of 0, 4, 8, 12, 16, ... , 4N ms have elapsed from the reference time point. Referring back to FIG. 8, for example, the first time point t1 is 0 ms after the reference time point, and the third time point t3 is 4 ms after the reference time point. The first non-volatile memory 120_1 starts the processing of the erase command at predetermined time intervals from the reference time point.

[00119] The second non-volatile memory 120_2 starts the processing of the erase command after the times of 2, 6, 10, 14, 18, ... , 4N ms have elapsed from the reference time point. Referring to FIG. 8, the second time point t2 is 2 ms after the reference time point, and the fourth time point t4 is 6 ms after the reference time point. That is, the second non-volatile memory 120_2 may start the processing of the erase command at predetermined time intervals after an offset time (for example, 2 ms) has elapsed from the reference time point.

[00120] The first NVM controller 125_1 sets the reference time point based on the time synchronization signal provided from the synchronization timer 160. The first NVM controller 125_1 may issue an erase/program command to the first memory cell 130 at predetermined time intervals from the reference time point.

[00121] The second NVM controller 125_2 sets the reference time point based on the time synchronization signal provided from the synchronization timer 160. The second NVM controller 125_2 may issue an erase/program command to the second memory cell 130 at predetermined time intervals from the reference time point. The offset time may be, for example, a time required for the first non-volatile memory 120_1 or the second non-volatile memory 120_2 to process the erase/program command.

[00122] By doing so, the second non-volatile memory 120_2 waits without processing the erase/program command while the first non-volatile memory 120_1 processes the erase/program command. That is, the processing times of erase/program commands of the first non-volatile memory 120_1 and the second non-volatile memory 120_2 are mutually exclusive, without overlapping each other.

[00123] FIG. 12B depicts a table showing an example in which the processing times of erase commands by the first non-volatile memory 120_1 and the second non-volatile memory 120_2 partially overlap with one another, unlike the example shown in FIG. 12A.

[00124] The operation of the first non-volatile memory 120_1 is the same as that described above with reference to FIG. 12A; and, therefore, the redundant description will be omitted.

[00125] The second non-volatile memory 120_2 starts the processing of the erase commands after times of $2 - k$, $6 - k$, $10 - k$, $14 - k$, $18 - k$, ..., $(4N + 2) - k$ ms have been elapsed from the reference time point (e.g., k may be an integer such as 1). That is, the times that the second non-volatile memory 120_2 processes the erase/program commands may overlap with the times that the first non-volatile memory 120_1 processes the erase/program commands by k ms.

[00126] In other words, in the example shown in FIG. 12B, the processing times of the erase/program commands by the first non-volatile memory 120_1 and the second non-volatile memory 120_2 are partially overlapped.

[00127] In an exemplary embodiment of the present inventive concept, the k value satisfies the condition $0 \leq k < m$, depending on the user's settings, where m is a time taken for processing an erase/program command.

[00128] FIG. 13 is a flowchart for illustrating erase/program operation of a first non-volatile memory and a second non-volatile memory according to an exemplary embodiment of the present inventive concept.

[00129] Referring to FIG. 13, according to an exemplary embodiment of the present inventive concept, the second non-volatile memory 120_2 performs an erase/program command even while the first non-volatile memory 120_1 is performing the erase/program command. As described above, the second non-volatile memory 120_2 can check whether the first non-volatile memory 120_1 is processing the erase/program command, and the first non-volatile memory 120_1 can also check whether the second non-volatile memory 120_2 is processing the erase/program command.

[00130] Initially, the erase/program operation is started in the first non-volatile memory 120_1 (step S310).

[00131] Subsequently, a command corresponding to the same erase/program operation is provided to the second non-volatile memory 120_2. The second non-volatile memory 120_2 checks if there is a read command for the second non-volatile memory 120_2 in a preset range among the commands stored in the command queue 111 (step S320). For example, the second NVM controller 125_2 may access the command queue 111 to check whether there is a read command provided to the second non-volatile memory 120_2 within the preset range.

[00132] If there is no read command provided to the second non-volatile memory 120_2 within the preset range, the second non-volatile memory 120_2 performs the erase/program operation immediately (S340). In this case, the second non-volatile memory 120_2 does not check whether the first non-volatile memory 120_1 is performing the erase/program operation.

[00133] If there is a read command provided to the second non-volatile memory 120_2 within the preset range, the second non-volatile memory 120_2 performs an erase/program operation exclusively according to a predetermined rule (S330). For example, the predetermined rule corresponds to the rules described above with reference to FIGS. 9 to 12B.

In an embodiment, a read command is in the preset range when an address stored with the read command in the command queue 111 is greater than or equal to a first address and less than or equal to a second address.

[00134] As mentioned earlier, the second non-volatile memory 120_2 checks whether the first non-volatile memory 120_1 is performing the erase/program operation so that the first non-volatile memory 120_1 and the second non-volatile memory 120_2 can process the erase/program command mutually exclusively, thereby reducing the latency of the read command. However, if there is no read command in the preset range among the commands stored in the command queue 111, it is not necessary to process the erase/program command mutually exclusively between the first and second non-volatile memories 120_1 and 120_2.

Thus, the second non-volatile memory 120_2 may process the erase/program command immediately.

[00135] FIG. 14 is a block diagram for illustrating a storage system according to an exemplary embodiment of the present inventive concept.

[00136] Referring to FIG. 14, a storage system according to an exemplary embodiment of the present inventive concept includes a host 10 (e.g., a host device such a computer), a first storage device 200, and a second storage device 300.

[00137] The host 10 may generate a request to control read, program and erase operations of the first storage device 200 and the second storage device 300.

[00138] The first storage device 200 may be, for example, a solid-state drive (SSD) including a plurality of non-volatile memories 220_1, 220_2, ..., 220_n. The plurality of non-volatile memories 220_1 to 220_n may be packaged into, for example, a single chip.

[00139] Alternatively, the first storage device 200 may be, but is not limited to, a memory card such as a PC card or PCMCIA (personal computer memory card international association) card, a compact flash (CF) card, a smart media (SM) card or SMC, a memory

stick, a multimedia card (MMC, RS-MMC or MMCmicro), an SD card, (SD, miniSD, microSD or SDHC) and universal flash storage (UFS), and the like.

[00140] The second storage device 300 may mirror the data stored in the first storage device 200. Thus, the second storage device 300 may include a configuration similar to that of the first storage device 200. The second storage device 300 may be a plurality of non-volatile memories 320_1, 320_2, ..., 320_n, an SSD or a memory card such as a compact flash card, a smart media card, a memory stick, a multimedia card, an SD card, and a universal flash storage.

[00141] The host 10 may perform an operation similar to the controller 110 included in the above-described exemplary embodiments. Accordingly, the host may send a read request simultaneously to the first storage device 200 and the second storage device 300 that mirror each other, and may receive data provided from the first storage device 200 or the second storage device 300, whichever is earlier, while ignoring the data provided later.

[00142] The first storage device 200 includes a first controller 210 for controlling a plurality of non-volatile memories 220_1 to 220_n. The first controller 210 may process a command received from the host 10 for each of the plurality of non-volatile memories 220_1 to 220_n. That is, for two different commands that access the first non-volatile memory 220_1, the first controller 210 may process them one by one over two data cycles, instead of processing them simultaneously. The first controller 210 may include an address assignment map for converting logical addresses received from the host 10 into physical addresses corresponding to the plurality of non-volatile memories 220_1 to 220_n, respectively.

[00143] The second storage device 300 includes a second controller 310 for controlling a plurality of non-volatile memories 320_1 to 320_n. The second controller 310 may process a command received from the host 10 for each of the plurality of non-volatile memories 320_1 to 320_n. That is, for two different commands that access the second non-volatile memory 320_1, the second controller 310 may process them one by one over two data cycles, instead of processing them simultaneously. The second controller 310 may include an address assignment map for converting logical addresses received from the host 10 into physical addresses corresponding to the plurality of non-volatile memories 320_1 to 320_n, respectively.

[00144] At least one embodiment of the inventive concept provides an improvement to the functioning of a computer. For example, a computer that incorporates the storage device 100 of FIG. 1 can apply read commands to the storage device 100 and complete those read commands more quickly than a conventional computer since the controller of the storage

device 100 applies address mapping rules of two mirrored memory devices in the storage device 100 differently to prevent a read operation from being performed on the same bank or to reduce the instances where the read operation is performed on the same bank. For example, when a read command for a same logical address is provided by a computer that includes a conventional storage device, the execution of the read command is delayed by a bank collision, which does not occur or is less likely to occur when the computer uses the storage device 100 of the inventive concept. Further, a computer that incorporates the storage device 100 can process erase, program, and read operations with less delay time since the controller of the storage device 100 may prevent a read operation from being performed simultaneously with an erase or a program operation. Therefore, a computer that incorporates the storage device 100 can complete operations that access (e.g., read, write, delete, etc.) the storage device 100 more quickly than a computer that includes a conventional storage device. Accordingly, a computer that incorporates the storage 100 of the inventive concept performs better than a conventional computer.

[00145] The embodiments of the inventive concept have been described with reference to the attached drawings, but it is to be understood by one of ordinary skill in the art that the inventive concept may be performed one of ordinary skill in the art in other specific forms without changing the technical concept or major features of the inventive concept.

What is claimed is:

1. A storage device comprising:

a first non-volatile memory comprising a plurality of first banks having a plurality of
5 first addresses assigned thereto according to a first assignment policy;

a second non-volatile memory comprising a plurality of second banks having a
plurality of second addresses assigned thereto according to a second assignment policy
different from the first assignment policy, wherein the plurality of second addresses
corresponds to the plurality of first addresses, and wherein the second non-volatile memory
10 mirrors data items stored in the first addresses to store them in the second addresses; and

a controller configured to receive a command from a host to control the first
non-volatile memory and the second non-volatile memory,

wherein the controller provides a read command received from the host
simultaneously to the first non-volatile memory and the second non-volatile memory, and
15 outputs to the host an earlier one between data provided from the first non-volatile memory
and data provided from the second non-volatile memory based on the read command.

2. The storage device of claim 1, wherein the controller receives logical
addresses from the host for accessing the first non-volatile memory and the second
20 non-volatile memory, and comprises an address assignment map for converting the logical
addresses into physical addresses to assign them to the first banks and the second banks.

3. The storage device of claim 2, wherein the address assignment map comprises:
a first map for converting the logical addresses into first physical addresses
25 assigned to the first banks; and

a second map for converting the logical addresses into second physical
addresses assigned to the second banks.

4. The storage device of claim 3, wherein the first map and the second map are
30 based on different assignment rules.

5. The storage device of claim 1, wherein the host issues an erase/program
command that accesses the storage device,

wherein the controller generates from the erase/program command, a first erase/program command that accesses the first non-volatile memory and a second erase/program command that accesses the second non-volatile memory, and

5 wherein the controller processes the first erase/program command and the second erase/program command mutually exclusively.

6. The storage device of claim 5, wherein the second non-volatile memory performs a check to determine whether the first non-volatile memory is performing the first erase/program command, and performs the second erase/program command when the check
10 determines the first erase/program command is not being performed.

7. The storage device of claim 6, further comprising:

a channel for inter-process communication (IPC) between the first non-volatile memory and the second non-volatile memory,

15 wherein the second non-volatile memory performs a check to determine whether the first non-volatile memory is performing the first erase/program command via the IPC.

8. The storage device of claim 6, further comprising:

20 an internal memory connected between the first non-volatile memory and the second non-volatile memory,

wherein the internal memory comprises a busy flag indicating whether the erase/program command is being performed, and

wherein the first non-volatile memory sets the busy flag when the first erase/program command is being performed.

25

9. The storage device of claim 5, wherein the controller comprises a command queue for storing commands issued from the host,

wherein the second non-volatile memory performs a check to determine whether a read command provided to the second non-volatile memory is present among the commands
30 stored in the command queue, and performs the second erase/program command immediately when the check determines the read command is not present.

10. The storage device of claim 1, wherein the controller ignores a later one between data provided from the first non-volatile memory and data provided from the second non-volatile memory.

11. The storage device of claim 1, wherein the controller processes a command that is issued from the host and provided to the first non-volatile memory in each of the first banks sequentially, and processes the command in each of the second banks sequentially.

12. A method for operating a storage device comprising a first non-volatile memory comprising a plurality of first banks having a plurality of first addresses assigned thereto according to a first assignment policy, and a second non-volatile memory comprising a plurality of second banks having a plurality of second addresses assigned thereto according to a second assignment policy different from the first assignment policy, wherein the plurality of second addresses corresponds to the plurality of first addresses, and wherein the second non-volatile memory mirrors data items stored in the first addresses to store them in the second addresses, the method comprising:

issuing, by a host, a read request to the storage device;

applying, by a controller of the storage device, a read command simultaneously to the first non-volatile memory and the second non-volatile memory based on the read request;

outputting, by the controller, to the host an earlier one between data provided from the first non-volatile memory and data provided from the second non-volatile memory based on the read command.

13. The method of claim 12, further comprising:

issuing, by the host, an erase/program command that accesses the storage device;

generating, by the controller, a first erase/program command that accesses the first non-volatile memory and a second erase/program command that accesses the second non-volatile memory, based on the issued erase/program command; and

processing, by the controller, the first erase/program command and the second erase/program command mutually exclusively.

14. The method of claim 13, wherein the processing of the first erase/program command and the second erase/program command mutually exclusively comprises:

performing, by the controller a check to determine whether the first non-volatile memory is performing the first erase/program command, and

performing, by the controller, the second erase/program command when the check determines the first erase/program command is not being performed.

5

15. The method of claim 12, further comprising:

ignoring, by the controller, a later one between the data provided from the first non-volatile memory and the data provided from the second non-volatile memory.

10

16. A storage device comprising:

a first non-volatile memory comprising M rows of N first banks having a plurality of first addresses;

a second non-volatile memory comprising M rows of N second banks having a plurality of second addresses, wherein the plurality of second addresses corresponds to the plurality of first addresses, and wherein the second non-volatile memory mirrors data items stored in the first addresses to store them in the second addresses; and

a controller configured to provide a read command based on a read request received from a host simultaneously to the first non-volatile memory and the second non-volatile memory, and output to the host an earlier one between data provided from the first non-volatile memory and data provided from the second non-volatile memory based on the read command, and

wherein the controller shifts the first addresses belonging to a given row of the first banks by $M \bmod N$ banks to determine the corresponding second banks of the second addresses.

25

17. The storage device of claim 16, wherein the read request includes logical addresses, and the controller includes a map that maps the logical addresses to first physical addresses in the first non-volatile memory and second physical addresses in the second non-volatile memory.

30

18. The storage device of claim 16, wherein the host issues an erase/program command that accesses the storage device,

wherein the controller generates from the erase/program command, a first erase/program command that accesses the first non-volatile memory and a second erase/program command that accesses the second non-volatile memory, and

5 wherein the controller processes the first erase/program command and the second erase/program command mutually exclusively.