

I, STEPHANIE BOX, hereby declare:

- a) TransPerfect Translations, a Delaware corporation, with its principal office at 1250 Broadway 32nd Floor, New York, NY 10001, is a professional translation company of multiple languages, including English and Japanese;
- b) I am a translator that handles translations for TransPerfect Translations;
- c) I am fluent in the Japanese and English languages;
- d) I have translated from the original document to the translated document identified below, and to the best of my knowledge, information, and belief the translation of that document is accurate; and

I declare under the penalty of perjury under the laws of the United States that the foregoing is true and correct; and further declare that I am aware that willful false statements and the like so made are punishable by fine and imprisonment, or both, under 1001 of Title 19 of the United States Code.

Original Document Identifier: JP2005018490A

Translated Document Identifier: JP2005018490A_EN

A handwritten signature in black ink, appearing to read 'Steph', followed by a long horizontal line and a small upward tick at the end.

Stephanie Box

Signed: May 2, 2025

(19) Japan Patent Office (JP)

**(12) Japanese Unexamined
Patent Application
Publication (A)**(11) Japanese Unexamined Patent
Application Publication Number
**Japanese Unexamined Patent
Application Publication 2005-
018490
(P2005-18490A)**(43) Publication date **January 20, 2005 (01/20/2005)**

(51) Int. Cl. ⁷	F1	Theme codes (reference)
G06F 12/02	G06F 12/02 570A	5B018
G06F 12/00	G06F 12/00 597U	5B060
G06F 12/16	G06F 12/16 310A	
	G06F 12/16 310M	
	G06F 12/16 310R	
Request for examination Not yet requested Number of claims 8 OL (Total of 18 pages)		

(21) Application number	Japanese Patent Application 2003-183502 (P2003-183502)	(71) Applicant	000003067 TDK Corporation 1-13-1 Nihonbashi, Chuo-ku, Tokyo
(22) Date of application	June 26, 2003 (06/26/2003)	(72) Inventor	Yukio Terazaki c/o TDK Corporation, 1-13-1 Nihonbashi, Chuo-ku, Tokyo
		F terms (reference)	5B018 GA04 HA03 KA15 NA06 QA155B060 AB26

(54) (TITLE OF THE INVENTION) Memory controller and a flash memory system equipped with a memory controller

(57) (ABSTRACT)

(PROBLEM) To provide a memory controller that is optimal for use in a system having frequent occurrence of the continuous rewriting of data for consecutive host addresses and that can improve the speed of rewrite processing, and a flash memory system that is equipped with this memory controller.

(MEANS FOR SOLVING) A memory controller that manages a plurality of blocks having consecutive physical block addresses within a flash memory as a zone, wherein this memory controller is provided with a function to store information related to bad blocks contained within a zone in a management information storage block set within the blocks constituting said zone as well as to store information related to whether or not data has been stored in said zone, and a function to store data such that the logical block addresses of the data stored within adjacent blocks will be continuous when aligning the blocks constituting said zone in sequence of the physical block address excluding said management information storage blocks and said bad blocks.

(REPRESENTATIVE DRAWING/FIGURE) Figure 8

Zone	Logical block address
Zone 1	LBA10000
Zone 2	Erase request
Zone 3	LBA12000
Zone 4	LBA13000
Zone 5	LBA14000
Zone 6	LBA15000
Zone 7	LBA11000
Zone 8	Erased
Zone 9	Erased
Zone 10	Erased
Zone 11	Erased
Zone 12	Erased
Zone 13	Erased
Zone 14	Erased
Zone 15	Erased
Zone 16	Erased
Zone 17	Erased
Zone 18	Erased
Zone 19	Erased
Zone 20	Erased

(SCOPE OF PATENT CLAIMS)

(CLAIM 1)

A memory controller that manages a plurality of blocks having consecutive physical block addresses within a flash memory as a zone,

wherein said memory controller is characterized in that it is provided with a function to store information related to bad blocks contained within said zone in a management information storage block set within the blocks constituting said zone as well as to store information related to whether or not data has been stored in said zone, and a function to store data such that the logical block addresses of the data stored within adjacent blocks will be continuous when aligning the blocks constituting said zone in sequence of the physical block address excluding said management information storage blocks and said bad blocks.

(CLAIM 2)

The memory controller described in Claim 1 characterized in that it is provided with a function to perform data writing processing in relation to said zone in which there has been no preparation of an address translation table that shows the correspondence of the addresses supplied from the host system and the addresses within the flash memory.

(CLAIM 3)

The memory controller described in Claim 1 or Claim 2 characterized in that it is provided with a function to store information related to the physical block addresses where data has been stored in correspondence to the initial logical block address of the data stored in said zone and to store information related to the physical block address where data has been stored in correspondence to the last logical block address.

(CLAIM 4)

The memory controller described in Claim 1 through Claim 3 characterized in that it is provided with a function to store information related to the initial logical block address of the data stored in said zone within said management information storage block.

(CLAIM 5)

The memory controller described in Claim 1 through Claim 4 characterized in that it is provided with a function to store information indicating that data has not been stored in said zone in said management information storage block when data stored in a block other than said management information storage block and said bad block has been erased.

(CLAIM 6)

The memory controller described in Claim 1 through Claim 5 characterized in that it is provided with a function to store information related to the number of erase operations that have been performed in said zone in said management information storage block when data stored in a block other than said management information storage block and said bad block has been erased.

(CLAIM 7)

The memory controller described in Claim 1 through Claim 6 characterized in that it is provided with a function to store the information that may be stored within management information storage block in at least two regions within said zone.

(CLAIM 8)

A flash memory system that is equipped with the memory controller and the flash memory described in any one of Claim 1 through Claim 7.

(DETAILED DESCRIPTION OF THE INVENTION)

(0001)

(FIELD OF INDUSTRIAL APPLICATION)

The present invention relates to a memory controller and a flash memory system equipped with this memory controller.

(0002)

(PRIOR ART)

In recent years, it has become common for flash memory to be used as the semiconductor memory that may be used in memory systems such as memory cards and silicon disks. This flash memory is a type of non-volatile memory that requires that data be maintained regardless of whether the power is turned on or not.

(0003)

However, the flash memory that may be used in devices such as those described above is constructed of a page that is a unit of reading and writing processing (such as a data area of 512 bytes) and a block that is a unit of erase processing (such as a data area of 32 pages). Also, because the flash memory cannot overwrite data, writing new data that is different from that on pages on which data has already been written requires processing to write the new data after the data that was already written has been erased.

(0004)

However, because erasing is processed on a block-by-block basis as described above, the erasing of data on a page onto which the new data will be written means that all of the data on all of the pages in the blocks contained within that page will be deleted. Therefore, when rewriting data that has been written to a page, the data on all of the pages in the blocks contained within the page to be rewritten will be subjected to processing to be written onto the blocks undergoing the erase processing (hereinafter, the blocks undergoing the erase processing will be referred to as the erased blocks) and to erase the original blocks.

(0005)

In the above-noted processing, the data after rewriting will be written to a different block than the block prior to the rewriting; therefore, each time the data is rewritten, the block address in the flash memory (hereinafter referred to as the physical block address) will change. Therefore, the correspondence between the block address that may be supplied from the host system on which the flash memory system is installed (hereinafter, the block address that may be supplied from the host system will be referred to as the logical block address) and the physical block address corresponding to this logical block address will change each time the data is rewritten.

(0006)

In order to manage the correspondence between the logical and physical block addresses that will change each time the data is rewritten as described above, Japanese Unexamined Patent Application Publication 2001-243110 describes the preparation of an address translation table that shows the correspondence between logical block addresses and physical block addresses. In this literature, an address translation table is created for each area that is constructed of multiple blocks in order to manage the correspondence between the logical block addresses and the physical block addresses.

(0007)

In this type of system, when a write command for data is received from the host system, said address translation table is used as the basis to determine whether or not there is a block where the same data as that data has been stored in a logical address (the address that may be supplied from the host system). Therefore, if said address translation table was not created when the write command was received, the address translation table was created before the processing was performed, or the address translation table was updated after the processing was performed.

(0008)

(PROBLEM TO BE SOLVED BY THE INVENTION)

However, in systems in which there is frequent continuous rewriting of large volumes of data with consecutive host addresses, the processing time for the preparation or updating of the address translation table will prevent any improvement in the processing speed.

(0009)

Therefore, the present invention aims to provide a memory controller that is optimal for use in a system in which there is frequent continuous rewriting of large volumes of data with consecutive host addresses and that can improve the speed of rewrite processing, and to provide a flash memory system that is equipped with this memory controller.

(0010)

(MEANS FOR SOLVING THE PROBLEM)

The objective of the present invention may be achieved as a result of a memory controller that manages a plurality of blocks having consecutive physical block addresses within a flash memory as a zone, wherein said memory controller is characterized in that it is provided with a function to store information related to

bad blocks contained within said zone in a management information storage block set within the blocks constituting said zone as well as to store information related to whether or not data has been stored in said zone, and a function to store data such that the logical block addresses of the data stored within adjacent blocks will be continuous when aligning the blocks constituting said zone in sequence of the physical block address excluding said management information storage blocks and said bad blocks.

(0011)

Further, the objective of the present invention may be achieved by a flash memory system that is provided with this memory controller and a flash memory.

(0012)

Also, according to the present invention, it is even more preferable for the memory controller to be provided with a function to perform data writing processing in relation to said zone in which there has been no preparation of an address conversion table that shows the correspondence of the addresses supplied from the host system and the addresses within the flash memory.

(0013)

Also, according to the present invention, it is even more preferable for the memory controller to be provided with a function to store information related to the physical block addresses where data has been stored in correspondence to the initial logical block address of the data stored in said zone and to store information related to the physical block address where data has been stored in correspondence to the last logical block address.

(0014)

Also, according to the present invention, it is even more preferable for the memory controller to be provided with a function to store information related to the initial logical block address of the data stored in said zone in said management information storage block.

(0015)

Also, according to the present invention, it is even more preferable for the memory controller to be provided with a function to store information indicating that data has not been stored in said zone in said management information storage block when data stored in a block other than said management information storage block and said bad block has been erased.

(0016)

Also, according to the present invention, it is even more preferable for the memory controller to be provided with a function to store information related to the number of erase operations that have been performed in said zone in said management information storage block when data stored in a block other than said management information storage block and said bad block has been erased.

(0017)

Also, according to the present invention, it is even more preferable for the memory controller to be provided with a function to store the information that may be stored in management information storage block in at least two regions within said zone.

(0018)

(EXAMPLES OF EMBODIMENT OF THE INVENTION)

The examples of embodiment of the present invention will be described in detail based on the drawings below.

(Explanation of flash memory system 1)

Figure 1 is a block diagram that schematically illustrates flash memory system 1 according to the present invention. As shown in Figure 1, flash memory system 1 is constructed of flash memory 2 and controller 3 that controls this flash memory. Also, flash memory system 1 is typically removably attached to host system 4 and is used as a type of external storage relative to host computer system 4.

(0019)

Examples of host computer system 4 include various types of information processing devices such as personal computers or digital still cameras that can process various information such as text, audio, or image information.

(0020)

Flash memory 2 is a device that performs page-by-page reading or writing and block-by-block erasing, wherein a single block may be constructed of 32 pages, while a single page may be constructed of a 512-byte user region and a 16-byte redundant region.

(0021)

Controller 3 may be constructed of host interface control block 5, microprocessor 6, host interface block 7, work area 8, buffer 9, flash memory interface block 10, error collection code (ECC) block 11, and flash sequencer block 12. Controller 3 that may be constructed of these functional blocks may be integrated on a single semiconductor chip. The functions of each block are described below.

(0022)

Microprocessor 6 is a functional block that controls the overall operations of each functional block that constitutes controller 3.

(0023)

Host interface control block 5 is a functional block that controls the operations of host interface block 7. Here, host interface control block 5 is equipped with an operation configuration register (not shown in the figure) that configures the operation of host interface block 7, and host interface block 7 operates based on this operation configuration register.

(0024)

Host interface block 7 is a functional block that exchanges data, address information, status information, and external command information with host system 4. In other words, when flash memory system 1 is mounted onto host system 4, flash memory system 1 and host system 4 are connected to each other via external bus 13. In such a state, data and the like that is supplied from host system 4 to flash memory system 1 is taken into the interior of controller 3 using host interface block 7 as an inlet, and data and the like that is supplied from flash memory system 1 to host system 4 is fed into host system 4 using host interface block 7 as an outlet.

(0025)

In addition, host interface block 7 includes a task file register (not shown in the figure) that temporarily stores host addresses and external commands supplied by host system 4, and an error register (not shown in the figure) that may be set in the event of an error.

(0026)

Work area 8 is a work area that temporarily stores the data necessary for controlling flash memory 2, and is a functional block that may be constructed of a plurality of SRAM cells.

(0027)

Buffer 9 is a functional block that temporarily stores the data that was read out from flash memory 2 and the data to be written onto flash memory 2. In other words, the data that was read out of flash memory 2 is held in buffer 9 until host system 4 is in a receivable state, and the data to be written to flash memory 2 is held in buffer 9 until flash memory 2 is in a writable state.

(0028)

Flash sequencer block 12 is a functional block that controls the operation of flash memory 2 based on internal commands. Flash sequencer block 12 is equipped with a plurality of registers (not shown in the figure), and the information required to execute internal commands on this plurality of registers may be configured here. When the required information for executing internal commands is set in this plurality of registers, flash sequencer block 12 performs processing based on that information. Here, an "internal command" is a command that may be provided by controller 3 to flash memory 2, and it may be differentiated from an "external command," which is provided by host system 4 to flash memory system 1.

(0029)

Flash memory interface block 10 is a functional block that performs the provision and receipt of data, address information, status information, and internal command information with flash memory 2 via internal bus 14.

(0030)

ECC block 11 is a functional block that generates an error collection code to be appended to the data to be written to flash memory 2, and it also detects and corrects errors contained in the retrieved data based on the error collection code appended to the data that was read out.

(Description of the memory cell)

Next, the specific structure of memory cell 16 that constitutes flash memory 2 shown in Figure 1 will be described while referring to Figure 2 and Figure 3.

(0031)

Figure 2 is a cross-sectional view that schematically illustrates the structure of memory cell 16 constituting the flash memory. As shown in this figure, memory cell 16 is constructed of N-type source diffusion region 18 and drain diffusion region 19 formed on P-type semiconductor substrate 17, tunnel oxide film 20 formed over P-type semiconductor substrate 17 between source diffusion region 18 and drain diffusion region 19, floating gate electrode 21 formed on tunnel oxide film 20, insulating film 22 formed on floating gate electrode 21, and control gate electrode 23 formed on insulating film 22. Memory cell 16 having this type of configuration may be connected in series within a plurality of flash memories.

(0032)

Memory cell 16 will display either an "Erase state (a state in which no electrons are being accumulated)" or a "Write state (a state in which electrons are being accumulated)" depending on whether or not electrons have been injected into floating gate electrode 21. Here, a single memory cell 16 corresponds to one bit of data, while the "Erase state" of memory cell 16 corresponds to the data having a logical value of "1," and the "Write state" of memory cell 16 corresponds to the data having a logical value of "0."

(0033)

In the "Erase state," because no electrons are being accumulated in floating gate electrode 21, when no read voltage is applied to control gate electrode 23, no channels are formed on the surface of P-type semiconductor substrate 17 between source diffusion region 18 and the drain diffusion region 19, meaning that source diffusion region 18 and drain diffusion region 19 are electrically isolated. On the other hand, when a read voltage is applied to control gate electrode 23, a channel (not shown in the figure) is formed on the surface of P-type semiconductor substrate 17 between source diffusion region 18 and the drain diffusion region 19, and source diffusion region 18 and drain diffusion region 19 are electrically connected by this channel.

(0034)

In other words, in the "Erase state," control gate electrode 23 is electrically isolated from source diffusion region 18 and drain diffusion region 19 when no read voltage is applied, while source diffusion region 18 and drain diffusion region 19 will be electrically connected in the state in which a read voltage has been applied to control gate electrode 23.

(0035)

Figure 3 is a cross-sectional view that schematically illustrates memory cell 16 in the "Write state." As shown in this figure, the "Write state" refers to a state in which electrons have accumulated in floating gate electrode 21. Because floating gate electrode 21 is sandwiched between tunnel oxide film 20 and insulating film 22, the electrons injected into floating gate electrode 21 remain within floating gate electrode 21 for a very long time. In this "Write state," because electrons are accumulated in floating gate electrode 21, channel 24 is formed on the surface of P-type semiconductor substrate 17 between source diffusion region 18 and drain diffusion region 19, regardless of whether or not a read voltage is being applied to control gate electrode 23. Therefore, in the "Write state," source diffusion region 18 and drain diffusion region 19 are always electrically connected by channel 24, regardless of whether or not a readout voltage is being applied to control gate electrode 23.

(0036)

In addition, it is possible to read out whether said memory cell 16 is in the erase state or the write state as described

below. Memory cell 16 may be connected in series within the flash memory. A low level voltage is applied to memory cell 16 selected among this series and a high level voltage is applied to control gate electrode 23 of the other memory cells 16. In this state, the system will detect whether the series unit of memory cell 16 is in continuity. As a result, if this series is in continuity, the selected memory cell 16 is determined to be in a write state, and if it is in an isolated state, the selected flash memory cell 16 is determined to be in an erase state. In this way, it is possible to read out whether the data retained in any of memory cells 16 included within the series unit is "0" or "1".

(0037)

In addition, when changing memory cell 16, which is in the erase state, to the write state, a high voltage is applied, ensuring that control gate electrode 23 will be on the high potential side, and electrons are injected into floating gate electrode 21 via tunnel oxide film 20. At this time, the FN (Fowler-Nordheim) tunnel current will flow and electrons will be injected into floating gate electrode 21. On the other hand, if flash memory cell 16, which is in the write state, is changed to the erase state, a high voltage is applied, ensuring that control gate electrode 23 will be on the low potential side, and the electrons that have accumulated in floating gate electrode 21 will be expelled through tunnel oxide film 20.

(Description of the memory structure of the flash memory)

The memory structure of flash memory 2 will be described next. Figure 4 is a diagram that schematically illustrates the memory structure of flash memory 2. As shown in Figure 4, flash memory 2 is constructed of pages that are processing units for reading and writing data, and blocks that are erase units of data.

(0038)

The above-noted page may be constructed of, for example, a 512-byte data region 25 and a 16-byte redundant region 26. Data region 25 is primarily an area in which the data that may be supplied from host system 4 is stored, while redundant region 26 is an area in which additional information, such as error collection codes, corresponding logical block addresses, and the block status, may be stored.

(0039)

An error collection code is additional information for correcting errors contained in the data that may be stored in user region 25, and it is generated by ECC block 11. Based on this error collection code, if the number of errors contained in the data stored in user area 25 is less than or equal to a predetermined number, the errors will be corrected.

(0040)

The corresponding logical block address indicates to which logical block address the block corresponds to when data is stored in the block. If data is not stored in the block, the corresponding logical block address is not stored, either, so it is possible to determine whether the block is an erased block based on whether the corresponding logical block address has been stored or not. In other words, if the corresponding logical block address is not stored, it is judged to be an erased block.

(0041)

The block status is a flag that indicates whether the block is a bad block (a block that cannot normally write data, etc.), and if it is determined that the block is a bad block, a flag that indicates it is a bad block (hereinafter, a flag that indicates it is a bad block is called a bad block flag) will be set for the block.

(Description of the processing according to the present invention)

The flash memory system according to the present invention manages data in zones consisting of multiple blocks. The explanation will describe the instance in which a single zone consists of 1024 blocks while referring to the figures.

(0042)

Figure 5 is a diagram illustrating the relationship between the zones, blocks, and pages of the flash memory. As shown in Figure 5, there is a 512-byte data region that may be called a page, which is a unit of reading and writing

processing, while the block that is a unit of erase processing is constructed of a data region with 32 pages. In addition, a single zone is constructed of 1024 blocks.

(0043)

Here, if the data region of a single page is 512 bytes and a single block is constructed of 32 pages, the lower 5 bits of the host address will be used to specify the page, and if the remaining upper bits are used as the block address, the block address will be used to specify the block. Because the capacity of the data region on a single page and the number of pages that make up a single block will vary depending on the specifications of the flash memory, the corresponding part of the host address may also change accordingly. In addition, each zone may store up to 1,000 blocks of data, and 24 blocks are spare blocks that take into account the occurrence of bad blocks or the like.

(0044)

The flash memory system according to the present invention performs write processing in zone units in order to enable the high-speed processing of data in a single group with consecutive host addresses in continuous write processing (hereinafter, the continuous write processing of the data of a single group with consecutive host addresses is referred to as group data write processing). Therefore, it is preferable to set the number of blocks that constitute the zone as appropriate in consideration of the number of blocks, etc., written in a single group data write process.

(0045)

The zone management table to manage the data storage status of each zone will be described next. Figure 6 through Figure 8 are diagrams illustrating zone management tables. This zone management table manages the logical block addresses of the data written onto each zone by zone. For example, zone 1 through zone 5 of the zone management table shown in Figure 6 are written with data corresponding to the following logical block addresses: In order to indicate this storage status, the initial logical block address of the stored data is written onto the zone management table.

Zone 1: LBA10000 to LBA10999

Zone 2: LBA11000 to LBA11999

Zone 3: LBA12000 to LBA12999

Zone 4: LBA13000 to LBA13999

Zone 5: LBA14000 to LBA14999

Further, for Zone 7 through Zone 20, all of the blocks excluding bad blocks will be in an erase state (hereinafter, the zones for which all of the blocks excluding bad blocks are in an erase state will be referred to as erased zones), so "Erased" flags indicating this will be set.

(0046)

In this state, if an external command to write data corresponding to a logical block address for LBA15000 to LBA15999 is received from the host system, the erased zone will be found within the zone management table, and the data will be written onto that erased zone. For example, if Zone 6, which is an erased zone, is written with data corresponding to a logical block address for LBA15000 through LBA15999, the storage status of Zone 6 in the zone management table will be revised from the erased flag to LBA15000 (the initial logical block address of the stored data), as shown in Figure 7.

(0047)

In addition, when revising data that has already been written, the zone management table will search for the zone in which the original data was written, and will change the storage status of that zone to an erase request flag (a flag indicating that erase processing is necessary to convert the zone to an erased zone). On the other hand, for new data, the erased zone is found in the zone management table, and after the data is written to the erased zone, the erased flag of the zone is changed to the first logical block address of the written data.

(0048)

For example, if the data corresponding to a logical block address for LBA11000 to LBA11999 is revised, as shown in Figure 8, Zone 2 in which the original data was written will be changed to an erase request flag, and the storage

status of Zone 7 in which the new data corresponding to a logical block address for LBA11000 to LBA11999 will be revised from the erased flag to LBA11000 (the initial logical block address of the stored data).

(0049)

Next, referring to Figure 9 to Figure 13, the writing processing for data into a zone constructed of 1024 blocks will be described. In the flash memory system according to the present invention, information indicating the state of each zone (write state, erase state, erase request state, etc.) and information on the data stored in each zone (information on the physical block address of the block where the data is stored, information on the logical block address of the stored data, etc.) are written into the management information storage block (such as the first page of the first or last good block) that has been set in each zone. In addition, when managing the erase count, etc., for each zone to ensure that each zone is used equally, it is preferable to write the erase count, etc., for that zone in the management information storage block. As long as information showing the state of each zone, information related to the data that has been stored in each zone, or the zone management information such as the erase count performed for a zone, etc., is written in the management information storage block set within each zone in this manner, it will be possible to prepare the zone management table based on that information and to manage the order of priority of the erased zones to be used.

(0050)

In order to improve the reliability of the zone management information that has been written in the above-noted management information storage block, it is preferable to write the zone management information in two or more regions. For example, if the zone management information is written on the initial page of the management information storage block, and the same backup data as the data written on the initial page is written on the next page, then even if the data written on the initial page fails, the correct zone management information can be obtained using the data written on the next page. In addition, when a backup management information storage block is set and the same backup data as the zone management information written in the regular management information storage block is written, it will be possible to improve the reliability of the zone management information as well.

(0051)

Figure 9 to Figure 13 are examples of the use of the above-noted management information storage block as the initial page of the initial good block, and these figures show the data region and redundant region of the blocks that constitute the zone, as well as the zone management information written on the first page of the initial good block.

(0052)

Figure 9 is a diagram that shows the data region and redundant region of the erased zone as well as the zone management information written on the first page of the initial good block. Here, the first page of B0000, which is the initial good block, includes an erase flag indicating that it is an erased zone, information showing the physical block address of the block for the start of data writing (hereinafter, the information showing the physical block address of the block where data writing will be started will be referred to as the write start pointer), information showing the physical block address of the block where the last data was written in the event that the data has been written (hereinafter, the information showing the physical block address of the block where the last data was written will be referred to as the write end pointer), and information on bad blocks present within the zone (hereinafter, the information on bad blocks present within the zone will be referred to as bad block information).

(0053)

If an external command is received from the host system to write data corresponding to a logical block address for LBA10000 through LBA10999 in this erased zone, the data will be written in order from B0001 set to the write start pointer. At this time, the data will be written, skipping bad blocks without writing any data onto the bad blocks listed in the bad block information. In addition, the data will be written, skipping over the initial good block without writing any data onto the initial good block, either.

(0054)

In this write processing, the data corresponding to a logical block address for LBA10000 to LBA10999 will be sequentially captured into buffer 9 via host interface block 7 under the control of host interface control block 5

shown in Figure 1. On the other hand, the data captured in buffer 9 will be sequentially written from B0001 of the physical block address by setting the following write processing in the register (not shown in the figure) of flash sequencer block 12.

- 1) An internal write command is set as an internal command for a predetermined register (not shown in the figure) within flash sequencer block 12.
- 2) The page address in the physical block address set to the write start pointer is set to a predetermined register (not shown in the figure) within the flash sequencer block 12, starting from the beginning.

(0055)

Thereafter, based on the above-noted write processing settings, flash sequencer block 12 will sequentially execute the processing. When this processing is executed, information will be provided from flash memory interface block 10 to flash memory 2 via internal bus 14 in order to execute the internal commands. As a result, the data captured in buffer 9 will be supplied sequentially to flash memory 2 via internal bus 14 and stored sequentially in the data region of the page specified in the above-noted write processing settings.

(0056)

As a result of the above-noted write processing, data corresponding to a logical block addresses for LBA10000 through LBA10999 is stored in the data area of blocks B0001 to B1002 of the physical block addresses, in the order of the logical block addresses, as shown in Figure 10. At this time, the data will be written, skipping the blocks without writing any data onto B0004 and B0009 listed in the bad block information. In addition, the logical block address corresponding to the data stored in the data region will be written in the redundant region.

(0057)

After completing the above-noted writing process, the zone management information that has been written on the first page of the initial good block will be changed to the zone management information as shown in Figure 10. Here, the erase flag is changed to a written flag indicating that the data has been written into this zone, and B1002 of the physical block address is set to the write end pointer. If a written flag has been set, the above-noted write start pointer indicates the block that contains data corresponding to the initial logical block address of the data that has been written in that zone.

(0058)

When changing the zone management information written on the first page of the initial good block, the data that has been written is read out into buffer 9, then the data of the initial good block is deleted. Next, the data that was read out into buffer 9 is changed under the control of microprocessor 6, and the data is written to the first page of the initial good block. In addition, when writing data such as logical block addresses into the redundant region, under the control of microprocessor 6, the data to be written to the area corresponding to the redundant region of buffer 9 is set, and the data is written to the redundant region.

(0059)

When updating the zone management information written on the first page of the initial good block, the updated data may be written to a page that does not have any data written in the block, and furthermore, information indicating the validity of the zone management information that has been written to a page may be written within that block. In this case, the zone management information will be written to a page other than the first page, but there will be no need to perform erase processing each time an update is made.

(0060)

Next, the write processing of data corresponding to a logical block address for LBA11000 to LBA11999 after the erasure of the data corresponding to a logical block address for LBA10000 to LBA10999 that has been written into the zone shown in Figure 10 will be described. In this processing, first, the zone to which the data corresponding to a logical block address for LBA10000 to LBA10999 is changed to an erased zone (hereinafter, the processing to change a zone to which data has been written into an erased zone will be referred to as zone erase processing), and

then, write processing is performed for the data corresponding to a logical block address for LBA11000 to LBA11999 is performed.

(0061)

During the zone erase processing, the bad block information written on the first page of the initial good block of each zone is read out, and the erase processing is sequentially performed for blocks other than the bad blocks and the initial good block. In other words, bad blocks are skipped and the erase processing is performed only for the good blocks (excluding the initial good block). During this zone erase processing, it is also acceptable to perform the erase processing on only the good blocks between the write start pointer and the write end pointer that have been written onto the first page of the initial good block, without performing the erase processing on all of the good blocks (excluding the initial good block).

(0062)

If zone erase processing is performed on the zone where data corresponding to a logical block address for LBA10000 through LBA10999 is written, as shown in Figure 10, the zone erase processing will sequentially be performed on all blocks except B0000, B0004, B0009 and B1021, or on blocks B0001 through B1002 excluding B0004 and B0009.

(0063)

Here, the erase processing for each block may be performed sequentially by setting the erase processing as follows in the register (not shown in the figure) of flash sequencer block 12.

- 1) An internal erase command is set as an internal command for a predetermined register (not shown in the figure) within flash sequencer block 12.
- 2) The physical block address of the block to be erased is sequentially set to a predetermined register (not shown in the figure) within flash sequencer block 12.

(0064)

Thereafter, based on the above-noted erase processing settings, flash sequencer block 12 will sequentially execute the processing. When this processing is executed, information will be provided from flash memory interface block 10 to flash memory 2 via internal bus 14 in order to execute the internal commands. As a result, the data that has been written to the block specified in the above-noted erase processing settings will be deleted sequentially.

(0065)

After the above-noted zone erase has been completed, the zone management information that has been written on the first page of the initial good block is read out into buffer 9, and then the data of the initial good block is erased, the information that was read out into buffer 9 is changed, and the information shown in Figure 11 is written into the first page of the initial good block. In this change, the erase request flag or the written flag is changed to the erased flag, and the write start pointer and the write end pointer are changed to the physical block address corresponding to the next block in the final data storage block (the block corresponding to the physical block address set to the end pointer) prior to the zone erase. In Figure 11, because the physical block address of the final data storage block (the physical block address set to the write end pointer) before the zone erase was B1002, the write start pointer and the write end pointer after the zone erase are set to B1003.

(0066)

Also, if the final data storage block before the zone erase is the last block of the zone (in the case of blocks B1023), the write start pointer and write end pointer after the zone erase are set to the physical block address (B0000) of the initial block. However, no data is written onto the initial good block or bad blocks, and the data will be written, skipping over those blocks. Therefore, if the next block of the final data storage block before the zone erase is the initial good block or a bad block, it is preferable to set the write start pointer and the write end pointer by skipping the initial good block or the bad block.

(0067)

For example, if B1004, B0008, and B1021 are bad blocks, and if the write end pointer before the zone erase is B0003, then B0005 is set for the write start pointer and the write end pointer after the zone erase, while if the write end pointer before the zone erase is B1023, then it is preferable to set B0001 for the write start pointer and the write end pointer after the zone erase.

(0068)

If the data is rewritten, the zone erase process described above will be performed on the zone in which the original data was written. Therefore, if the data is rewritten, it is preferable to change the written flag written on the first page of the initial good block in the zone in which the original data was written to the erase request flag, and then to change the erase request flag to the erased flag when the above-noted zone erase processing has been performed.

(0069)

Next, the write processing for data corresponding to a logical block address for LBA11000 to LBA11999 according to an external command from the host system to the erased zone shown in Figure 11 will be described.

(0070)

As with the case above, in this write processing as well, the data corresponding to a logical block address for LBA11000 to LBA11999 will be sequentially captured into buffer 9 via host interface block 7 under the control of host interface control block 5 shown in Figure 1. On the other hand, the data captured in buffer 9 will be sequentially written from B1004 of the physical block address by setting the following write processing in the register (not shown in the figure) of flash sequencer block 12 as described above.

(0071)

As a result of the above-noted write processing, data corresponding to a logical block addresses for LBA11000 through LBA11999 is stored in the data area of blocks B1004 to B0982 of the physical block addresses, in the order of the logical block addresses, as shown in Figure 12. At this time, the data will be written, skipping the blocks without writing any data onto B0004, B0009, and B1021 listed in the bad block information. In addition, the data will be written, skipping over the initial good block without writing any data onto B0000, which is the initial good block, either. In addition, the logical block address corresponding to the data stored in the data region will be written in the redundant region.

(0072)

After completing the above-noted writing process, the zone management information that has been written on the first page of the initial good product block will be changed to the information as shown in Figure 12. Here, the erase flag is changed to a written flag indicating that the data has been written into this zone, and B0982 of the physical block address is set to the write end pointer.

(0073)

In the examples shown in Figure 10 and Figure 12, the logical block address corresponding to the data stored in the data region was written into the redundant region, but as shown in Figure 13, it is also acceptable not to write the logical block address corresponding to data stored in the data region into the redundant region. However, in this case, based on the zone management information written on the first page of the initial good block, an assessment must be made as to whether or not the block in the zone is an erased block.

(0074)

In Figure 13, the data is stored in the data region of a block having a physical block address from B0001 to B1002 (the block denoted by an * symbol). In addition, the first page of the initial good block contains information about the logical block address of the data stored in this zone along with the written flag. In addition, the write start pointer, write end pointer, and bad block information are written in the same manner as in the example shown in Figure 10.

(0075)

Information about the above-noted logical block address includes the initial logical block address of the stored data, the first and last logical block address of the stored data, and the like. Here, the data stored in the physical block address indicated by the write start pointer corresponds to the data of the initial logical block address, while the data stored in the physical block address indicated by the write end pointer corresponds to the data of the last logical block address.

(0076)

For example, if the initial logical block address of the data stored as shown in Figure 13 is written on the first page of the initial good block, the correspondence between the logical block address and the physical block address can be ascertained based on the write start pointer, the write end pointer, and the bad block information.

(0077)

In other words, as shown below, the data in B0001 of the physical block address indicated by the write start pointer corresponds to the data in LBA10000 of the logical block address, and the data of the good block after B0001 corresponds to the data of the logical block address after LBA10000.

B0001 data: LBA10000 data

B0002 data: LBA10001 data

B0003 data: LBA10002 data

B0005 data: LBA10003 data

•

•

•

B1002 data: LBA10999 data

In this way, it is possible to ascertain the logical block address corresponding to the data from the physical block address indicated by the write start pointer and up to the physical block address indicated by the write end pointer.

(0078)

Next, the explanation will focus on the case in which data that has been written into a zone as described above is read out. When reading out data, first, the above-noted zone management table is used as reference to find the zone in which the data corresponding to the logical block address to be read out has been stored. Here, the zone management table may be created based on the zone management information that has been written on the first page of the initial good block for each zone at startup. However, if information about the logical block address of the stored data is not written to the first page of the initial good block, the logical block address written into the redundant region of the physical block address set to the read start point will be used.

(0079)

Next, the zone management information written on the first page of the initial good block in the zone containing the data corresponding to the logical block address to be read out will be read out. Based on this zone management information, the following read processing settings can be made for the register (not shown in the figure) of flash sequencer block 12 in order to read out the data corresponding to the logical block address to be read out.

1) An internal read command is set as an internal command for a predetermined register (not shown in the figure) within flash sequencer block 12.

2) The page address within the physical block address where there is storage of the data corresponding to the logical block address to be read out as calculated based on the zone management information written onto the first page of the initial good block is set in the predetermined register (not shown in the figure) within flash sequencer block 12 in sequence from the top.

(0080)

Thereafter, based on the above-noted read processing settings, flash sequencer block 12 will sequentially execute

the processing. When this processing is executed, information will be provided from flash memory interface block 10 to flash memory 2 via internal bus 14 in order to execute the internal commands. As a result, the data stored in flash memory 2 is sequentially read out to buffer 9 via internal bus 14.

(0081)

On the other hand, the data read out into buffer 9 is sequentially transmitted to host system 1 via host interface block 7 under the control of host interface control block 5 shown in Figure 1.

(0082)

(EFFECT OF THE INVENTION)

As explained above, according to the present invention, data can be stored in the blocks constituting a zone in sequence of the logical block address, skipping bad blocks and management information storage blocks (blocks to which zone management information has been written), and therefore, there will be no need to update the address translation table during write processing, making it possible to improve the speed of rewrite processing. In addition, because the storage status of data can be ascertained simply using the information indicating the state of each zone, information on the data stored in each zone, or simple zone management information such as the number of erase operations that have been performed for a zone, it is suitable for systems in which there are frequent continuous rewriting operations on a large volume of data with consecutive host addresses.

(BRIEF DESCRIPTION OF THE DRAWINGS)

(FIGURE 1) Figure 1 is a block diagram that schematically illustrates flash memory system 1 according to the present invention.

(FIGURE 2) Figure 2 is a cross-sectional view that schematically illustrates the structure of memory cell 16 constituting the flash memory.

(FIGURE 3) Figure 3 is a cross-sectional view that schematically illustrates memory cell 16 in the "Write state".

(FIGURE 4) Figure 4 is a diagram that schematically shows the structure of the address space of flash memory 2.

(FIGURE 5) Figure 5 is a diagram that schematically illustrates the structure of flash memory 2.

(FIGURE 6) Figure 6 is a diagram showing an example of a zone management table.

(FIGURE 7) Figure 7 is a diagram showing an example of a zone management table.

(FIGURE 8) Figure 8 is a diagram showing an example of a zone management table.

(FIGURE 9) Figure 9 is a diagram that shows the data region and redundant region of the blocks constituting the zone as well as the zone management information written on the first page of the initial good block.

(FIGURE 10) Figure 10 is a diagram that shows the data region and redundant region of the blocks constituting the zone as well as the zone management information written on the first page of the initial good block.

(FIGURE 11) Figure 11 is a diagram that shows the data region and redundant region of the blocks constituting the zone as well as the zone management information written on the first page of the initial good block.

(FIGURE 12) Figure 12 is a diagram that shows the data region and redundant region of the blocks constituting the zone as well as the zone management information written on the first page of the initial good block.

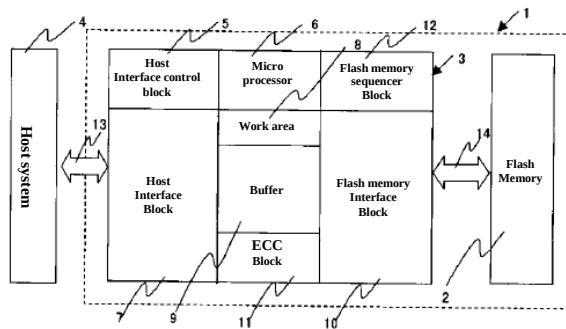
(FIGURE 13) Figure 13 is a diagram that shows the data region and redundant region of the blocks constituting the zone as well as the zone management information written on the first page of the initial good block.

(EXPLANATION OF REFERENCES)

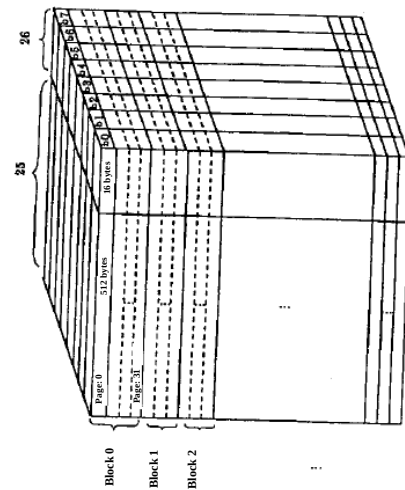
- 1 Flash memory system
- 2 Flash memory
- 3 Controller
- 4 Host computer
- 5 Host interface control block
- 6 Microprocessor
- 7 Host interface block
- 8 Work area
- 9 Buffer
- 10 Flash memory interface block

- 11 ECC block
- 12 Flash sequencer block
- 13 External bus
- 14 Internal bus
- 16 Memory cell
- 17 P-type semiconductor substrate
- 18 Source diffusion area
- 19 Drain diffusion area
- 20 Tunnel oxide film
- 21 Floating gate electrode
- 22 Insulating film
- 23 Control gate electrode
- 24 Channel
- 25 Data region
- 26 Redundant region

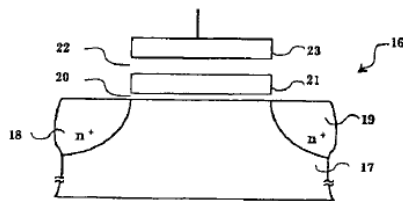
(FIGURE 1)



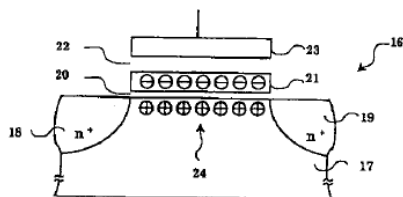
(FIGURE 4)



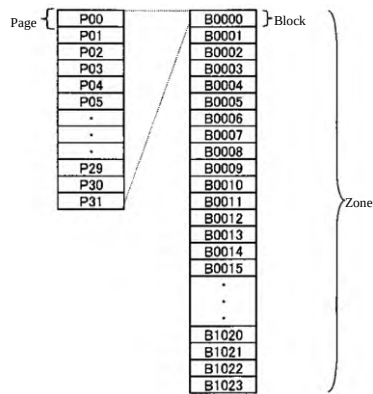
(FIGURE 2)



(FIGURE 3)



(FIGURE 5)



(FIGURE 6)

Zone	Logical block address
Zone 1	LBA10000
Zone 2	LBA11000
Zone 3	LBA12000
Zone 4	LBA13000
Zone 5	LBA14000
Zone 6	Erased
Zone 7	Erased
Zone 8	Erased
Zone 9	Erased
Zone 10	Erased
Zone 11	Erased
Zone 12	Erased
Zone 13	Erased
Zone 14	Erased
Zone 15	Erased
Zone 16	Erased
Zone 17	Erased
Zone 18	Erased
Zone 19	Erased
Zone 20	Erased

(FIGURE 7)

Zone	Logical block address
Zone 1	LBA10000
Zone 2	LBA11000
Zone 3	LBA12000
Zone 4	LBA13000
Zone 5	LBA14000
Zone 6	LBA15000
Zone 7	Erased
Zone 8	Erased
Zone 9	Erased
Zone 10	Erased
Zone 11	Erased
Zone 12	Erased
Zone 13	Erased
Zone 14	Erased
Zone 15	Erased
Zone 16	Erased
Zone 17	Erased
Zone 18	Erased
Zone 19	Erased
Zone 20	Erased

(FIGURE 8)

Zone	Logical block address
Zone 1	LBA10000
Zone 2	Erase request
Zone 3	LBA12000
Zone 4	LBA13000
Zone 5	LBA14000
Zone 6	LBA15000
Zone 7	LBA11000
Zone 8	Erased
Zone 9	Erased
Zone 10	Erased
Zone 11	Erased
Zone 12	Erased
Zone 13	Erased
Zone 14	Erased
Zone 15	Erased
Zone 16	Erased
Zone 17	Erased
Zone 18	Erased
Zone 19	Erased
Zone 20	Erased

(FIGURE 9)

	Data region	Redundant region
Erased	B0000	
Start	B0001	
B0001	B0002	
End	B0003	
B0001	B0004	Bad
Bad	B0005	
B0004	B0006	
B0009	B0007	
B1021	B0008	
	B0009	Bad
	B0010	
	B0011	
	B0012	
	B0013	
	B0014	
	B0015	
	.	.
	.	.
	.	.
	.	.
	.	.
	B1020	
	B1021	Bad
	B1022	
	B1023	

(FIGURE 10)

	Data region	Redundant region
Written	B0000	
Start	B0001	LB10000
B0001	B0002	LB10001
End	B0003	LB10002
B1002	B0004	Bad
Bad	B0005	LB10003
B0004	B0006	LB10004
B0009	B0007	LB10005
B1021	B0008	LB10006
	B0009	Bad
	B0010	LB10007
	B0011	LB10008
	.	.
	.	.
	.	.
	B1001	LB10998
	B1002	LB10999
	B1003	
	B1004	
	.	.
	.	.
	B1020	
	B1021	Bad
	B1022	
	B1023	

(FIGURE 11)

	Data region	Redundant region
Erased	B0000	
Start	B0001	
B1003	B0002	
End	B0003	
B1003	B0004	Bad
Bad	B0005	
B0004	B0006	
B0009	B0007	
B1021	B0008	
	B0009	Bad
	B0010	
	B0011	
	B0012	
	B0013	
	B0014	
	B0015	
	.	.
	.	.
	.	.
	.	.
	.	.
	B1020	
	B1021	Bad
	B1022	
	B1023	

(FIGURE 12)

	Data region	Redundant region
Written	B0000	
Start	B0001	LB11020
B1003	B0002	LB11021
End	B0003	LB11022
B0982	B0004	Bad
Bad	B0005	LB11023
B0004	B0006	LB11024
B0009	B0007	LB11025
B1021	B0008	LB11026
	B0009	Bad
	.	.
	.	.
	B0981	LB11998
	B0982	LB11999
	B0983	
	.	.
	.	.
	B1002	
	B1003	LB11000
	B1004	LB11001
	.	.
	.	.
	B1021	Bad
	B1022	LB11018
	B1023	LB11019

(FIGURE 13)

		Data region		Redundant region
Written		B0000		
LBA10000	*	B0001		
Start	*	B0002		
B0001	*	B0003		
End		B0004	Bad	
B1003	*	B0005		
Bad	*	B0006		
B0003	*	B0007		
B0009	*	B0008		
B1021	*	B0009	Bad	
	*	B0010		
	*	B0011		
		.	.	
		.	.	
		.	.	
	*	B1001		
	*	B1002		
		B1003		
		B1004		
		.	.	
		.	.	
		B1020		
		B1021	Bad	
		B1022		
		B1023		