

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: JP2008176606A

Translated Document Identifier: JP2008176606A_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

(12) **Japanese Unexamined Patent
Application Publication (A)**

(19) Japan Patent Office (JP)

(11) Japanese Unexamined Patent
Application Publication Number**2008-176606****(P2008-176606A)**(43) Publication date **July 31, 2008 (07.31.2008)**

(51) Int. Cl.	F1	Theme codes (reference)
G06F 12/00 (01.2006)	G06F 12/00 560B	5B060
G11C 16/02 (01.2006)	G11C 17/00 601T	5B065
G06F 3/08 (01.2006)	G06F 12/00 597U	5B125
	G06F 3/08 H	
	G11C 17/00 601E	
	Request for examination	Requested Number of claims 4 OL (Total of 14 pages)

(21) Application number Japanese Patent Application 2007-009970 (P2007-9970)

(22) Date of application January 19, 2007 (01.19.2007)

(71) Applicant 000003067

TDK Corporation

1-13-1 Nihonbashi, Chuo-ku, Tokyo

(74) Agent 100074930

Patent attorney

Keiichi Yamamoto

(74) Agent 100131886

Patent attorney

Takashi Sakamoto

(72) Inventor Takeo Kikuchi

c/o TDK Corporation, 1-13-1 Nihonbashi, Chuo-ku, Tokyo

(72) Inventor Sukenao Kobayashi

c/o TDK Corporation, 1-13-1 Nihonbashi, Chuo-ku, Tokyo

F terms 5B060 AA02 AA14

(reference) 5B065 BA05 CA15 CE01 CE23 EK05

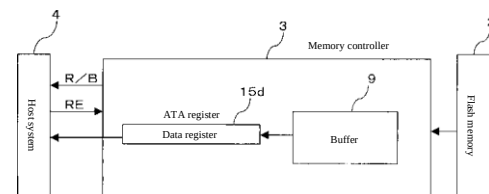
Continued on the last page

(54) (TITLE OF THE INVENTION) Memory controller and a flash memory system provided with a memory controller, as well as a flash memory control method

(57) (ABSTRACT)

(PROBLEM) To efficiently transfer internal information within a flash memory system that is connected to an ATA interface to the host system side in response to a request from the host system side. Further, to enable the host system to acquire status information that is equal to or greater than the length of the status register from the flash memory using a simple means. (MEANS FOR SOLVING) In response to instruction information that may be provided from the host system, the internal information related to the status within the flash memory will be written to the buffer, and upon the completion of the writing of the internal information to the buffer, the host system is notified of the fact that transferable data is being held within the buffer, and the internal information that is being held in the buffer will be transferred to the host system. The host system will be able to obtain internal information in an operation that is similar to that when reading out data that is being stored in flash memory.

(REPRESENTATIVE DRAWING/FIGURE) Figure 5



(SCOPE OF PATENT CLAIMS)

(CLAIM 1)

A memory controller that controls access to flash memory in which the erasing of stored data may be performed in physical block units based on instruction information that may be provided from the host system, wherein this memory controller is characterized by the fact that it is provided with

a data retention means to hold the write data to be written to said flash memory or the read data that was read out from said flash memory in one or more sector units;

a notification means to provide said host system with notification of the fact that transferable data is being stored in said data retention means;

a data transfer means to transfer the data being held in said data retention means to said host system; 10

and a data setting means to write internal information data regarding the status within said flash memory to said data retention means in response to instruction information that may be provided by said host system;

and, when said read data that was read out and that can be transferred to said host system has been held in said data retention means, or when there has been completion of the writing of said internal information data to said data retention means by said information setting means, said notification means will submit notification of the fact that transferable data is being held in said data retention means.

(CLAIM 2)

The memory controller described in Claim 1 that is characterized by the fact that said internal information is the number of bad blocks and the number of empty blocks within said flash memory. 20

(CLAIM 3)

A flash memory system that is characterized by the fact that it is provided with the memory controller described in Claim 1 or Claim 2 and flash memory.

(CLAIM 4)

A flash memory control method that controls access to flash memory in which the erasing of stored data may be performed in physical block units based on instruction information that may be provided from the host system, wherein this flash memory control method is characterized by the fact that it includes

a data setting step to write the internal information data related to the status within said flash memory into the data retention means that will hold the data to be written to said flash memory or the data that was read out from said flash memory in units of one or more sectors in response to the instruction information that may be provided from said host system; 30

a notification step to notify said host system of the fact that transferable data is being held in said data retention means when the writing of said internal information data to said data retention means has been completed;

and a data transfer step to transfer said internal information data being held in said data retention means to said host system.

(DETAILED DESCRIPTION OF THE INVENTION)

(TECHNICAL FIELD)

40

(0001)

The present invention relates to a memory controller, a flash memory system that is equipped with this memory controller, and a flash memory control method.

(BACKGROUND ART)

(0002)

In recent years, there has been active development of flash memory, which is a non-volatile storage medium, and flash memory has become popular as a storage medium for information devices (host systems) such as digital cameras. Accompanying the increase in the amount of data handled by these types of information devices, the storage capacity of flash memory is also increasing. Therefore, NAND flash memory that boasts excellent integration and for which the capacity is easily increased is garnering attention. 50

(0003)

NAND flash memory can be performed on a memory cell basis if the memory cell changes from an erased state (logical value = 1) to a written state (logical value = 0). However, if a memory cell is changed from a write state (logical value = 0) to an erasure state (logical value = 1), it can only be done in a predetermined erasure unit (block unit) consisting of multiple memory cells. Data will be read out and written in page units, and the blocks that represent the erase units are constructed of a plurality of pages.

(0004)

A page of NAND flash memory corresponds to one or more sectors in a magnetic disk device. Therefore, memory systems using NAND flash memory are increasingly used in applications that replace the conventional magnetic disk devices. For this reason, the interface of memory systems using NAND flash memory typically uses AT Attachment (ATA), which is used in magnetic disk devices.

10

(0005)

An ATA-compliant interface (hereinafter referred to as an ATA interface) is provided with ATA register 15, such as command register 15a, sector count register 15b, Logical Block Address (LBA) register 15c, data register 15d, or status register 15e, as shown in Figure 7. By writing to or reading out from this ATA register 15, it will be possible to perform access to peripherals connected to the ATA interface. Therefore, even in the case of a flash memory system in which a peripheral device uses flash memory as a storage medium, when accessing the flash memory system from the host system side, access may be performed by setting command, address, and other instruction information in ATA register 15.

20

(0006)

For example, when reading out or writing is performed on a flash memory system from the host system side, the first address of the sector where the read or write is performed is set in LBA register 15c, the number of sectors to read out or the number of sectors to be written to is set in sector count register 15b, and the command to read and write is set in command register 15a. Based on these settings, data will be written to the flash memory in the case of writing, and data will be read out from the flash memory in the case of reading.

(0007)

The host system can also learn the status of the peripherals connected to the ATA interface based on the information that has been set in status register 15e. In other words, when wishing to determine the status of peripheral devices connected to the ATA interface, as disclosed in the following Patent Literature 1, the information that has been set in status register 15e (hereinafter referred to as the status information) may be read out. Each bit in status register 15e is allocated specific information. Therefore, the state of a peripheral device may be determined based on the logical values of each bit of status information that has been read out from status register 15e.

30

(0008)

(PATENT LITERATURE 1) Japanese Unexamined Patent Application Publication 2005-071560

(DISCLOSURE OF THE INVENTION)

(PROBLEM TO BE SOLVED BY THE INVENTION)

40

(0009)

However, using the status information that has been read out from status register 15e, it will only be possible to learn the information that has been allocated to each bit of status register 15e. Therefore, in a flash memory system connected to an ATA interface, it was not possible to acquire information showing the internal state, such as the number of bad blocks or the number of empty blocks within the flash memory (hereinafter to be referred to as internal information).

(0010)

Therefore, the aim of the present invention is to provide a memory controller that is optimal for use in acquiring internal information of the flash memory system connected to an ATA interface from the host system side, as well as to provide a flash memory system using this memory controller and a flash memory control method.

(MEANS FOR SOLVING THE PROBLEM)

(0011)

In order to achieve the above-noted aim, the memory controller according to the present invention is a memory controller that controls access to flash memory in which the erasing of stored data may be performed in physical block units based on instruction information that may be provided from the host system, wherein this memory controller is characterized by the fact that it is provided with a data retention means to hold the write data to be written to said flash memory or the read data that was read out from said flash memory in one or more sector units; a notification means to provide said host system with notification of the fact that transferable data is being stored in said data retention means; a data transfer means to transfer the data being held in said data retention means to said host system; and a data setting means to write internal information data regarding the status within said flash memory to said data retention means in response to instruction information that may be provided by said host system; and, when said read data that was read out and that can be transferred to said host system has been held in said data retention means, or when there has been completion of the writing of said internal information data to said data retention means by said information setting means, said notification means will submit notification of the fact that transferable data is being held in said data retention means.

10

20

(0012)

It is preferable for said internal information to be the number of bad blocks and the number of empty blocks within said flash memory.

(0013)

In order to achieve the above-noted aim, the flash memory system according to the present invention is provided with one of said memory controllers and a flash memory.

(0014)

In order to achieve the above-noted aim, the flash memory control method according to the present invention is a flash memory control method that controls access to flash memory in which the erasing of stored data may be performed in physical block units based on instruction information that may be provided from the host system, wherein this flash memory control method is characterized by the fact that it includes a data setting step to write the internal information data related to the status within said flash memory into the data retention means that will hold the data to be written to said flash memory or the data that was read out from said flash memory in units of one or more sectors in response to the instruction information that may be provided from said host system; a notification step to notify said host system of the fact that transferable data is being held in said data retention means when the writing of said internal information data to said data retention means has been completed; and a data transfer step to transfer said internal information data being held in said data retention means to said host system.

30

40

(EFFECT OF THE INVENTION)

(0015)

According to the present invention, it will be possible to efficiently transfer internal information within a flash memory system that is connected to an ATA interface to the host system side in response to a request from the host system side. Also, the host system will be able to obtain internal information in an operation that is similar to that when reading out data that is being stored in flash memory.

(BEST EMBODIMENTS OF THE INVENTION)

(0016)

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

50

(0017)

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 that is a NAND flash memory, and controller 3 that controls this flash memory.

(0018)

Flash memory system 1 may be connected to host system 4 via external bus 13. Host system 4 is constructed of a CPU (Central Processing Unit) to control the operation of host system 4 as a whole, as well as a companion chip, etc., responsible for providing and receiving information from flash memory system 1. Host computer system 4 may be 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.

(0019)

Flash memory 2 is a non-volatile memory and is constructed of a register and a memory cell array. Flash memory 2 performs the writing of data to or reading out of data between the register and the memory cell array.

(0020)

The memory cell array is provided with a plurality of memory cell groups and a word line. In each memory cell group, a plurality of memory cells are connected in series. The word line is used to select a particular memory cell within a memory cell group. The writing or reading out of data may be performed between the selected memory cell and the register via this word line.

(0021)

The memory cells that constitute the memory cell array are provided with control gates and floating gates, and the writing of data or the erasing of data may be performed by injecting a charge (electrons) into a floating gate or by discharging the charge (electrons) from a floating gate. Here, the erase state corresponds to a logical value of "1", while the write state corresponds to a logical value of "0".

(0022)

The structure of flash memory 2, which is a NAND flash memory, is shown in Figure 2. Flash memory 2 is constructed of "pages" and "blocks (physical blocks)".

(0023)

A page is a processing unit for the data reading operations and data writing operations that may be performed in flash memory 2. A physical block is a processing unit for the data erase operations that may be performed in flash memory 2, and a physical block is constructed of a plurality of pages.

(0024)

Although the configuration of the above-noted physical blocks and pages depends on the specifications of the flash memory, it is common to use a small block of flash memory as shown in Figure 2 (a) and a large block of flash memory as shown in Figure 2 (b). In the small block configuration, as shown in Figure 2 (a), a block is constructed of 32 pages (P0 to P31), and each page is constructed of a 512-byte (1 sector) user region 25 and a 16-byte redundant region 26. In the large block configuration, as shown in Figure 2 (b), a block is constructed of 64 pages (P0 to P63), and each page is constructed of a 2048-byte (4 sectors) user region 25 and a 64-byte redundant region 26.

(0025)

Here, user region 25 is a region to store user data that may be supplied from host system 4.

(0026)

In addition, redundant region 26 is an area for storing additional data such as error correction codes, logical address information, and block status.

(0027)

Error correction codes are data that may be used to detect and correct errors contained in the user data stored in user region 25.

(0028)

The logical address information is information that may be used to identify a logical block that corresponds to the user data when valid user data is stored in a physical block. A logical block is a collection of multiple sectors in the address space on the host system 4 side, and by managing the

correspondence between this logical block and the physical block, it will be possible to manage the correspondence between the logical address managed by host system 4 and the physical address in flash memory 2. The address space on the side of host system 4 is managed by the Logical Block Address (LBA), which is a number assigned to the area divided into sector (512 bytes) units. Host system 4 specifies an access area within flash memory 2 using this LBA.

(0029)

If no valid user data is stored in the physical block, no logical address information will be written to redundant region 26 of that physical block. Therefore, it is possible to determine whether or not a physical block is an empty block (an erased block) based on whether or not the logical address information has been written to redundant region 26. In other words, if no logical address information has been written, it is determined to be an empty block (erased block).

10

(0030)

The block status is information that indicates whether the physical block is a bad block (a physical block to which data cannot be written normally), and based on this information, it is possible to determine whether it is a bad block or not. Therefore, redundant region 26 of the physical block that is determined to be a bad block will have information showing that it is a bad block set as its block status.

(0031)

This type of flash memory 2 will receive data, address information, or internal commands, etc., from controller 3, and will execute each operation such as the reading, writing, and erasing, etc., of data.

20

(0032)

As shown in Figure 1, controller 3 is provided with microprocessor 6, host interface block 7, work area 8, buffer 9, flash memory interface block 10, ECC (error correcting code) block 11, and Read Only Memory (ROM) 12. Controller 3 that may be constructed of these functional blocks may be integrated on a single semiconductor chip. Each function block will be described below.

(0033)

Microprocessor 6 controls the overall operation of controller 3 according to the program that may be stored in ROM 12. For example, microprocessor 6 reads a command set (sequence command) defining the various processing, etc., from ROM 12 and feeds it to flash memory interface block 10 in order to cause flash memory interface block 10 to execute the processing.

30

(0034)

Host interface block 7 exchanges data, address information, status information, and external commands, etc., to and from host system 4. The data, etc., that may be supplied from host system 4 to flash memory system 1 will be taken within flash memory system 1 (such as buffer 9) using host interface block 7 as the inlet. The data, etc., that may be supplied from flash memory system 1 to host system 4 is also supplied to host system 4 using host interface block 7 as the outlet.

(0035)

Work area 8 is a work area that temporarily stores the data required to control flash memory 2, and is constructed of multiple Static Random Access Memory (SRAM) cells. Work area 8 may store, for example, an address translation table for converting logical and physical addresses, or an empty block search table for searching for empty blocks, etc.

40

(0036)

Buffer 9 temporarily holds the data that was read out from flash memory 2 and the data to be written to flash memory 2 in one sector or multiple sector units. 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.

(0037)

Flash memory interface block 10 is a block that controls the write processing from buffer 9 to flash memory 2 and the read processing from flash memory 2 to buffer 9. The series of control operations when performing write or read operations will be executed based on a sequence command indicating a series of control operations stored in ROM 12.

(0038)

ECC block 11 generates an Error Correcting Code (ECC) that is appended to the data to be written to flash memory 2, and detects and corrects errors contained in the data that was read out based on the ECC that was appended to the data that was read out from flash memory 2.

(0039)

ROM 12 is a non-volatile storage element that stores the programs, etc., that define the procedures for the processing to be executed by microprocessor 6. More specifically, ROM 12 stores the program that defines the processing procedures, such as, for example, creating an address translation table.

(0040)

Next, the method of managing the storage area of flash memory will be described. Due to the increasing storage capacity of flash memory, a method of dividing and managing this storage space into multiple zones has been used in order to smoothly manage the storage space of the large amount of flash memory.

(0041)

Figure 3 is an example of the management of the storage area of flash memory 2 following division into multiple zones (physical zones). Each physical block, which is the erase unit for stored data, is assigned a unique physical block address (PBA). Each physical block may be classified into one of several physical zones, each of which will be assigned a physical zone number (PZN). The number of physical blocks included in each physical zone is called the Physical Zone Internal Block Number (PZIBN). In the example shown in Figure 3, the physical block is assigned a PBA consecutively starting from #0, and a total of 1024 physical blocks with PBA numbers from #0 to #1023 are included within the physical zone of PZN #0, while a total of 1024 physical blocks with PBA numbers from #1024 to #2047 are included in the physical zone of PZN #1, and a total of 1024 physical blocks with PBA numbers from PBA #2048 to #3071 are included in the physical zone of PZN #2.

(0042)

On the other hand, the address space on the host system 4 side is also divided into multiple zones (logical zones), and a single physical zone is assigned to a single logical zone. The data corresponding to each logical block in the logical zone will be written to the physical block in the physical zone assigned to that logical zone. The number of logical blocks in the logical zone will be set to be less than the number of physical blocks in the physical zone. For example, a logical zone containing 1000 logical blocks will be assigned to a physical zone containing 1024 physical blocks. In this way, the reason why the number of logical blocks contained in the logical zone will be less than the number of physical blocks contained in the physical zone is because it takes into account the innate number of bad blocks and the number of blocks that became bad over time.

(0043)

In addition, logical address information corresponding to the data that has been written will be written to the redundant region of the physical block where the data was written. This logical address information may be the logical block number (LBN), which is the number assigned to the logical block, or the logical block number in the logical zone (LZIBN), which is the number within each logical zone of the logical block contained in each logical zone.

(0044)

Next, the procedure of obtaining information (internal information) indicating the internal state of flash memory 2 from the host system 4 side will be described. In the following description, as the internal information, the number of bad blocks and the number of empty blocks per physical zone as shown in Figure 4 will be obtained. Below, the explanation will refer to Figure 4 to Figure 6 when describing the process from the preparation of the information related to the number of bad blocks and the number of empty blocks in each physical zone by memory controller 3 and up to the read out of the information that was prepared from the host system 4 side (up to the transfer of the information that was prepared to the host system 4 side).

(0045)

10

In this example of embodiment, a command code that is not specified for operation in the ATA specifications will be assigned to the command to acquire the number of bad blocks and the number of empty blocks in each physical zone. This type of vendor-specific special command that is not specified in the ATA specifications is called a vendor command. In the following description, the vendor command to acquire the number of bad blocks and the number of empty blocks in each physical zone is referred to as the internal information acquisition command.

(0046)

Referring to Figure 5, the preparation of the internal information and the read out of the internal information that was prepared from the host system 4 side (transfer to the host system 4 side) in the first example of embodiment of the present invention will be described. In the first example of embodiment, first, host system 4 will write the command code of the internal information acquisition command to the command register in memory controller 3 (command register 15a of ATA register 15 shown in Figure 7). Memory controller 3 will initiate the preparation of information regarding the number of bad and the number of empty blocks in each physical zone in response to this internal information acquisition command. In this preparation processing, the block status and logical address information that has been written to the redundant region of each physical block will be read out sequentially, and the number of bad blocks and the number of empty blocks in each physical zone will be counted based on the block status and logical address information that was read out.

20

(0047)

First, the block status and logical address information that has been written to the redundant region of the physical blocks from PBA #0 to #1023 will be read out, and the number of bad blocks and the number of empty blocks in the physical zone of PZN #0 will be counted. Here, for the number of bad blocks, the number of physical blocks with information that indicates that the block status is a bad block will be counted. For the number of free blocks, the number of physical blocks with no information written indicating that the block status is a bad block and with no logical address information written will be counted. When the count of bad blocks and empty blocks in the PZN #0 physical zone has been completed, the number of bad blocks and the number of empty blocks in the PZN #0 physical zone will be written to buffer 9.

30

(0048)

Next, the block status and logical address information that has been written to the redundant region of the physical blocks from PBA #1024 to #2047 will be read out, and the number of bad blocks and the number of empty blocks in the physical zone of PZN #1 will be counted. When the count of bad blocks and empty blocks in the PZN #1 physical zone has been completed, the number of bad blocks and the number of empty blocks in the PZN #1 physical zone will be written to buffer 9. In this way, the number of bad blocks and the number of empty blocks in the physical zones from PZN #0 to #7 will be counted, and the number of bad blocks and the number of empty blocks per physical zone will be written to buffer 9 as the internal information. When counting the number of bad blocks and the number of empty blocks in each physical zone, areas to write the number of bad blocks and the number of empty blocks in each physical zone may be allocated on buffer 9, and "0" may be written as the initial value for the allocated area, with the number increased in increments of 1 for writing to these areas when there has been detection of bad blocks or empty blocks. For example, if "3" has been written to an area to write the number of bad blocks in the physical zone of PZN #1 when a bad block contained in the physical zone of PZN #1 has been detected, the value will be rewritten to be "4".

40

50

(0049)

Next, the internal information that was written to buffer 9 (information on the number of bad blocks and the number of empty blocks in each physical zone) will be read from the host system 4 side (transferred to the host system 4 side).

(0050)

When all of the internal information has been written to buffer 9, memory controller 3 notifies host system 4 of this fact. This notification uses a signal output from the Ready/Busy (R/B) terminal of memory controller 3. In other words, the level of the signal that may be outputted from the Ready/Busy (R/B) terminal (high or low level) indicates the Ready or Busy state (hereinafter, the signal indicating the Ready state will be called the Ready signal, while the signal that indicates the Busy state will be called the Busy signal). For example, if the high level is set to the Ready state and the low level is set to the Busy state, the low level (Busy signal) will be outputted from the Ready/Busy (R/B) terminal until the writing of the internal information to buffer 9 has been finished, and the high level (Ready signal) will be outputted from the Ready/Busy (R/B) terminal when the writing of the internal information to buffer 9 has been finished.

10

(0051)

The method of notifying host system 4 that all of the internal information has been written to buffer 9 may be a method of outputting an interrupt signal or rewriting the logical value of a particular bit in status register 15e.

(0052)

When memory controller 3 notifies host system 4 that all of the internal information has been written to buffer 9, host system 4 will alternately input high-level and low-level signals to the read enable (RE) terminal of memory controller 3. Memory controller 3 will transfer the internal information that was written to buffer 9 at the timing of the transition of the level of the signal that may be inputted into the read enable (RE) terminal (from high level to low level, or from low level to high level) to data register 15d. Here, the internal information that was written to buffer 9 will be transferred in units of 1 byte if the number of bits in data register 15d is 8 bits, and in units of 2 bytes if the number of bits in data register 15d is 16 bits. On the other hand, host system 4 will alternately input high-level and low-level signals to the read enable (RE) terminals and will read out the information held in data register 15d.

20

(0053)

In this way, the internal information that was written to buffer 9 will be transferred to data register 15d and read out by host system 4. When all of the internal information that was written to buffer 9 is set in data register 15d, memory controller 3 will output a busy signal from the ready busy (R/B) terminal, and host system 4 will terminate the input of the signal to the read enable (RE) terminal and the retrieval of information held in data register 15d.

30

(0054)

Referring to Figure 6, the preparation of the internal information and the read out of the internal information that was prepared from the host system 4 side (transfer to the host system 4 side) in the second example of embodiment of the present invention will be described. In the second example of embodiment, instead of starting counting the number of bad blocks and the number of empty blocks when an internal information acquisition command has been provided from host system 4, when creating an address translation table and an empty block table, the number of bad blocks and the number of empty blocks are counted in parallel, and the number of bad blocks and the number of empty blocks in each physical zone is held in work area 8.

40

(0055)

Then, when an internal information acquisition command has been provided by host system 4, memory controller 3 will write the internal information that has been held in work area 8 (the information on the number of bad blocks and the number of empty blocks in each physical zone) to buffer 9 and will output the ready signal from the Ready/Busy (R/B) terminal. Subsequently, as in the first example of embodiment, the internal information that was written to buffer 9 will be transferred to data register 15d and read out by host system 4. The reason for counting the number of bad blocks and the number of empty blocks in parallel when preparing the address translation table and empty block table is because the block status and logical address information that has been

50

written to the redundant region in each physical block will be read out when preparing the address translation table and empty block table.

(0056)

Here, address translation table is a table that shows the correspondence between the logical block and the physical block, while the empty block table is a table that shows the physical block address (PBA) of the empty blocks. This address translation table and empty block table may be created for each logical zone. Therefore, there may be instances in which no address translation table or an empty block table is created for a logical zone that was not accessed. Therefore, in this type of case, when an internal information acquisition command has been provided from host system 4, the counting of the number of bad blocks and the number of empty blocks will be performed only in the physical zones corresponding to logical zones in which no address translation table or empty block table will be prepared.

10

(0057)

In this example of embodiment, because the internal information is held in work area 8, as was the case in the first example of embodiment, there is the advantage that it is not necessary to count the number of bad blocks and the number of empty blocks when the internal information acquisition command is provided by host system 4, making it possible to respond quickly to the internal information acquisition command.

(0058)

However, in the case of this example of embodiment, the number of empty blocks and the number of bad blocks will change over time, so it will be necessary to update the internal information with each change. For example, if user data has been written to an empty block, the number of empty blocks in the corresponding physical zone will be decremented (decreased by one), and if the block to which user data was written in the past is erased and has become empty, the number of empty blocks in the corresponding physical zone will be incremented (increased by one). Similarly, in the case of bad blocks, if a new bad block has occurred, the number of bad blocks in the corresponding physical zone will be incremented (increased by one).

20

(0059)

As described above, in the present invention, in response to a command that has been provided by host system 4, memory controller 3 will write internal information to buffer 9, and memory controller 3 will notify host system 4 when all of the internal information has been written to buffer 9. Host system 4 will alternately input high-level and low-level signals to the read enable (RE) terminals in response to this notification, and will read out the information held in data register 15d. Because the internal information that was written to buffer 9 will be transferred to data register 15d at the timing of the transition of the signal input to the read enable (RE) terminal, host system 4 can obtain the internal information in a manner similar to that of reading the data that has been stored in flash memory 2.

30

(0060)

Also, in the present example of embodiment, the internal information was the number of bad blocks and the number of empty blocks in each physical zone, but the internal information is not limited to these types of information. Accordingly, the internal information may be information such as the physical block address (PBA) of a bad block.

40

(0061)

In addition, both of the examples of embodiment described above are illustrative and the present invention is not limited to these examples. The present invention may be implemented in various other variations and modifications. Accordingly, the scope of the present invention is defined only by the claims and their equivalent scope.

(BRIEF DESCRIPTION OF THE DRAWINGS)

(0062)

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

50

(FIGURE 2) This is an illustration showing the relationship between a block of flash memory and a page.

(FIGURE 3) This is an example of the storage area in flash memory that was divided into zones for management.

(FIGURE 4) This is an example of internal information in the example of embodiment of the present invention.

(FIGURE 5) This is an illustration to illustrate the operation of the memory controller in the first example of embodiment of the present invention.

(FIGURE 6) This is an illustration to illustrate the operation of the memory controller in the second example of embodiment of the present invention.

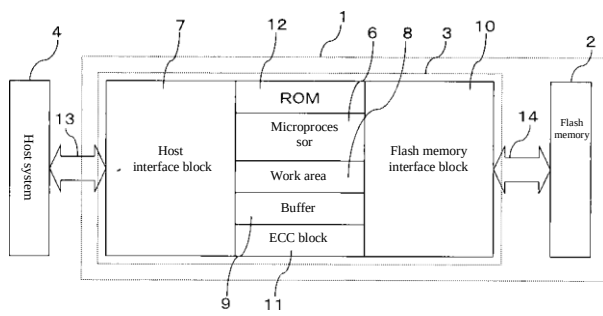
(FIGURE 7) This is a diagram schematically showing the ATA register.

(EXPLANATION OF REFERENCES)

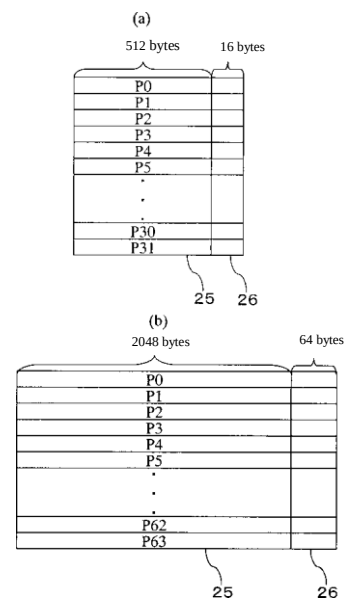
(0063)

1	Flash memory system	
2	Flash memory	
3	Memory controller	10
4	Host system	
6	Microprocessor	
7	Host interface block	
8	Work area	
9	Buffer	
10	Flash memory interface block	
11	ECC block	
12	ROM	
13	External bus	
14	Internal bus	20
15	ATA register	
15a	Command register	
15b	Sector count register	
15c	LBA register	
15d	Data register	
15e	Status register	
25	User region	
26	Redundant region	

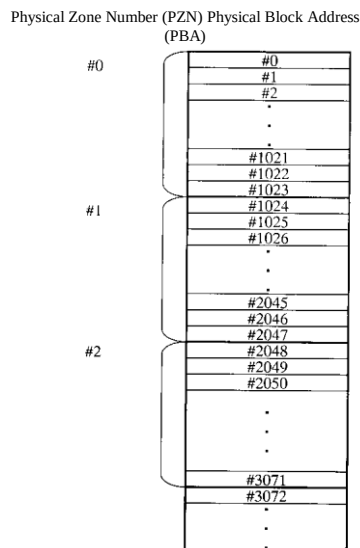
(FIGURE 1)



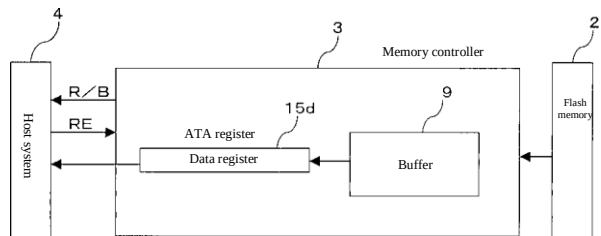
(FIGURE 2)



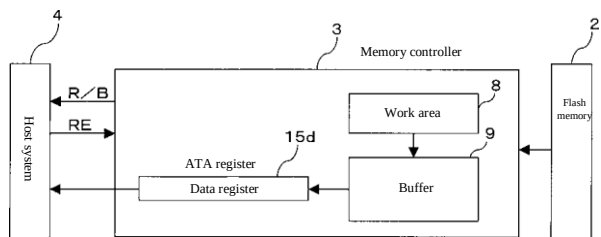
(FIGURE 3)



(FIGURE 5)



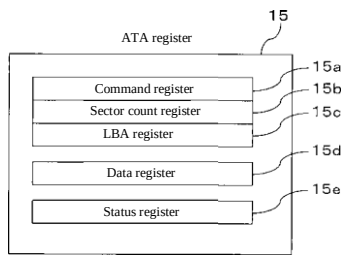
(FIGURE 6)



(FIGURE 4)

PZN	Number of bad blocks	Number of empty blocks
#0	3	10
#1	2	17
#2	4	12
#3	1	11
#4	3	15
#5	2	16
#6	3	14
#7	2	22

(FIGURE 7)



Continued from the front page

F terms (reference) 5B125 CA08 DD03 DD09 DE07 DE08 DE20 EA05 EA10 EK01 EK02
FA04

(12) **Japanese Unexamined Patent
Application Publication (A)**

(19) Japan Patent Office (JP)

(11) Japanese Unexamined Patent
Application Publication Number

2008-176606

(P2008-176606A)

(43) Publication date **July 31, 2008 (07.31.2008)**

(51) Int. Cl.			F1				Theme codes (reference)
G06F	12/00	(01.2006)	G06F	12/00	560B		5B060
G11C	16/02	(01.2006)	G11C	17/00	601T		5B065
G06F	3/08	(01.2006)	G06F	12/00	597U		5B125
			G06F	3/08	H		
			G11C	17/00	601E		
				Request for examination	Requested Number of claims 4 OL (Total of 14 pages)		

(21) Application number Japanese Patent Application 2007-009970 (P2007-9970)

(22) Date of application January 19, 2007 (01.19.2007)

(71) Applicant 000003067

TDK Corporation

1-13-1 Nihonbashi, Chuo-ku, Tokyo

(74) Agent 100074930

Patent attorney

Keiichi Yamamoto

(74) Agent 100131886

Patent attorney

Takashi Sakamoto

(72) Inventor Takeo Kikuchi

c/o TDK Corporation, 1-13-1 Nihonbashi, Chuo-ku, Tokyo

(72) Inventor Sukenao Kobayashi

c/o TDK Corporation, 1-13-1 Nihonbashi, Chuo-ku, Tokyo

F terms 5B060 AA02 AA14

(reference) 5B065 BA05 CA15 CE01 CE23 EK05

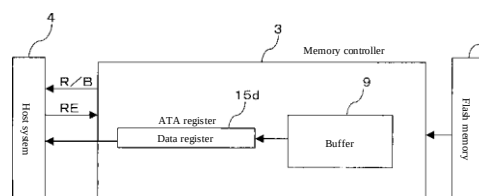
Continued on the last page

(54) (TITLE OF THE INVENTION) Memory controller and a flash memory system provided with a memory controller, as well as a flash memory control method

(57) (ABSTRACT)

(PROBLEM) To efficiently transfer internal information within a flash memory system that is connected to an ATA interface to the host system side in response to a request from the host system side. Further, to enable the host system to acquire status information that is equal to or greater than the length of the status register from the flash memory using a simple means. (MEANS FOR SOLVING) In response to instruction information that may be provided from the host system, the internal information related to the status within the flash memory will be written to the buffer, and upon the completion of the writing of the internal information to the buffer, the host system is notified of the fact that transferable data is being held within the buffer, and the internal information that is being held in the buffer will be transferred to the host system. The host system will be able to obtain internal information in an operation that is similar to that when reading out data that is being stored in flash memory.

(REPRESENTATIVE DRAWING/FIGURE) Figure 5



(SCOPE OF PATENT CLAIMS)

(CLAIM 1)

A memory controller that controls access to flash memory in which the erasing of stored data may be performed in physical block units based on instruction information that may be provided from the host system, wherein this memory controller is characterized by the fact that it is provided with

a data retention means to hold the write data to be written to said flash memory or the read data that was read out from said flash memory in one or more sector units;

a notification means to provide said host system with notification of the fact that transferable data is being stored in said data retention means;

a data transfer means to transfer the data being held in said data retention means to said host system;

and a data setting means to write internal information data regarding the status within said flash memory to said data retention means in response to instruction information that may be provided by said host system;

and, when said read data that was read out and that can be transferred to said host system has been held in said data retention means, or when there has been completion of the writing of said internal information data to said data retention means by said information setting means, said notification means will submit notification of the fact that transferable data is being held in said data retention means.

(CLAIM 2)

The memory controller described in Claim 1 that is characterized by the fact that said internal information is the number of bad blocks and the number of empty blocks within said flash memory.

(CLAIM 3)

A flash memory system that is characterized by the fact that it is provided with the memory controller described in Claim 1 or Claim 2 and flash memory.

(CLAIM 4)

A flash memory control method that controls access to flash memory in which the erasing of stored data may be performed in physical block units based on instruction information that may be provided from the host system, wherein this flash memory control method is characterized by the fact that it includes

a data setting step to write the internal information data related to the status within said flash memory into the data retention means that will hold the data to be written to said flash memory or the data that was read out from said flash memory in units of one or more sectors in response to the instruction information that may be provided from said host system;

a notification step to notify said host system of the fact that transferable data is being held in said data retention means when the writing of said internal information data to said data retention means has been completed;

and a data transfer step to transfer said internal information data being held in said data retention means to said host system.

(DETAILED DESCRIPTION OF THE INVENTION)

(TECHNICAL FIELD)

(0001)

The present invention relates to a memory controller, a flash memory system that is equipped with this memory controller, and a flash memory control method.

(BACKGROUND ART)

(0002)

In recent years, there has been active development of flash memory, which is a non-volatile storage medium, and flash memory has become popular as a storage medium for information devices (host systems) such as digital cameras. Accompanying the increase in the amount of data handled by these types of information devices, the storage capacity of flash memory is also increasing. Therefore, NAND flash memory that boasts excellent integration and for which the capacity is easily increased is garnering attention.

(0003)

NAND flash memory can be performed on a memory cell basis if the memory cell changes from an erased state (logical value = 1) to a written state (logical value = 0). However, if a memory cell is changed from a write state (logical value = 0) to an erasure state (logical value = 1), it can only be done in a predetermined erasure unit (block unit) consisting of multiple memory cells. Data will be read out and written in page units, and the blocks that represent the erase units are constructed of a plurality of pages.

(0004)

A page of NAND flash memory corresponds to one or more sectors in a magnetic disk device. Therefore, memory systems using NAND flash memory are increasingly used in applications that replace the conventional magnetic disk devices. For this reason, the interface of memory systems using NAND flash memory typically uses AT Attachment (ATA), which is used in magnetic disk devices.

10

(0005)

An ATA-compliant interface (hereinafter referred to as an ATA interface) is provided with ATA register 15, such as command register 15a, sector count register 15b, Logical Block Address (LBA) register 15c, data register 15d, or status register 15e, as shown in Figure 7. By writing to or reading out from this ATA register 15, it will be possible to perform access to peripherals connected to the ATA interface. Therefore, even in the case of a flash memory system in which a peripheral device uses flash memory as a storage medium, when accessing the flash memory system from the host system side, access may be performed by setting command, address, and other instruction information in ATA register 15.

20

(0006)

For example, when reading out or writing is performed on a flash memory system from the host system side, the first address of the sector where the read or write is performed is set in LBA register 15c, the number of sectors to read out or the number of sectors to be written to is set in sector count register 15b, and the command to read and write is set in command register 15a. Based on these settings, data will be written to the flash memory in the case of writing, and data will be read out from the flash memory in the case of reading.

(0007)

The host system can also learn the status of the peripherals connected to the ATA interface based on the information that has been set in status register 15e. In other words, when wishing to determine the status of peripheral devices connected to the ATA interface, as disclosed in the following Patent Literature 1, the information that has been set in status register 15e (hereinafter referred to as the status information) may be read out. Each bit in status register 15e is allocated specific information. Therefore, the state of a peripheral device may be determined based on the logical values of each bit of status information that has been read out from status register 15e.

30

(0008)

(PATENT LITERATURE 1) Japanese Unexamined Patent Application Publication 2005-071560

(DISCLOSURE OF THE INVENTION)

(PROBLEM TO BE SOLVED BY THE INVENTION)

40

(0009)

However, using the status information that has been read out from status register 15e, it will only be possible to learn the information that has been allocated to each bit of status register 15e. Therefore, in a flash memory system connected to an ATA interface, it was not possible to acquire information showing the internal state, such as the number of bad blocks or the number of empty blocks within the flash memory (hereinafter to be referred to as internal information).

(0010)

Therefore, the aim of the present invention is to provide a memory controller that is optimal for use in acquiring internal information of the flash memory system connected to an ATA interface from the host system side, as well as to provide a flash memory system using this memory controller and a flash memory control method.

(MEANS FOR SOLVING THE PROBLEM)

(0011)

In order to achieve the above-noted aim, the memory controller according to the present invention is a memory controller that controls access to flash memory in which the erasing of stored data may be performed in physical block units based on instruction information that may be provided from the host system, wherein this memory controller is characterized by the fact that it is provided with a data retention means to hold the write data to be written to said flash memory or the read data that was read out from said flash memory in one or more sector units; a notification means to provide said host system with notification of the fact that transferable data is being stored in said data retention means; a data transfer means to transfer the data being held in said data retention means to said host system; and a data setting means to write internal information data regarding the status within said flash memory to said data retention means in response to instruction information that may be provided by said host system; and, when said read data that was read out and that can be transferred to said host system has been held in said data retention means, or when there has been completion of the writing of said internal information data to said data retention means by said information setting means, said notification means will submit notification of the fact that transferable data is being held in said data retention means.

(0012)

It is preferable for said internal information to be the number of bad blocks and the number of empty blocks within said flash memory.

(0013)

In order to achieve the above-noted aim, the flash memory system according to the present invention is provided with one of said memory controllers and a flash memory.

(0014)

In order to achieve the above-noted aim, the flash memory control method according to the present invention is a flash memory control method that controls access to flash memory in which the erasing of stored data may be performed in physical block units based on instruction information that may be provided from the host system, wherein this flash memory control method is characterized by the fact that it includes a data setting step to write the internal information data related to the status within said flash memory into the data retention means that will hold the data to be written to said flash memory or the data that was read out from said flash memory in units of one or more sectors in response to the instruction information that may be provided from said host system; a notification step to notify said host system of the fact that transferable data is being held in said data retention means when the writing of said internal information data to said data retention means has been completed; and a data transfer step to transfer said internal information data being held in said data retention means to said host system.

(EFFECT OF THE INVENTION)

(0015)

According to the present invention, it will be possible to efficiently transfer internal information within a flash memory system that is connected to an ATA interface to the host system side in response to a request from the host system side. Also, the host system will be able to obtain internal information in an operation that is similar to that when reading out data that is being stored in flash memory.

(BEST EMBODIMENTS OF THE INVENTION)

(0016)

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

(0017)

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 that is a NAND flash memory, and controller 3 that controls this flash memory.

(0018)

Flash memory system 1 may be connected to host system 4 via external bus 13. Host system 4 is constructed of a CPU (Central Processing Unit) to control the operation of host system 4 as a whole, as well as a companion chip, etc., responsible for providing and receiving information from flash memory system 1. Host computer system 4 may be 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.

(0019)

Flash memory 2 is a non-volatile memory and is constructed of a register and a memory cell array. Flash memory 2 performs the writing of data to or reading out of data between the register and the memory cell array.

(0020)

The memory cell array is provided with a plurality of memory cell groups and a word line. In each memory cell group, a plurality of memory cells are connected in series. The word line is used to select a particular memory cell within a memory cell group. The writing or reading out of data may be performed between the selected memory cell and the register via this word line.

(0021)

The memory cells that constitute the memory cell array are provided with control gates and floating gates, and the writing of data or the erasing of data may be performed by injecting a charge (electrons) into a floating gate or by discharging the charge (electrons) from a floating gate. Here, the erase state corresponds to a logical value of "1", while the write state corresponds to a logical value of "0".

(0022)

The structure of flash memory 2, which is a NAND flash memory, is shown in Figure 2. Flash memory 2 is constructed of "pages" and "blocks (physical blocks)".

(0023)

A page is a processing unit for the data reading operations and data writing operations that may be performed in flash memory 2. A physical block is a processing unit for the data erase operations that may be performed in flash memory 2, and a physical block is constructed of a plurality of pages.

(0024)

Although the configuration of the above-noted physical blocks and pages depends on the specifications of the flash memory, it is common to use a small block of flash memory as shown in Figure 2 (a) and a large block of flash memory as shown in Figure 2 (b). In the small block configuration, as shown in Figure 2 (a), a block is constructed of 32 pages (P0 to P31), and each page is constructed of a 512-byte (1 sector) user region 25 and a 16-byte redundant region 26. In the large block configuration, as shown in Figure 2 (b), a block is constructed of 64 pages (P0 to P63), and each page is constructed of a 2048-byte (4 sectors) user region 25 and a 64-byte redundant region 26.

(0025)

Here, user region 25 is a region to store user data that may be supplied from host system 4.

(0026)

In addition, redundant region 26 is an area for storing additional data such as error correction codes, logical address information, and block status.

(0027)

Error correction codes are data that may be used to detect and correct errors contained in the user data stored in user region 25.

(0028)

The logical address information is information that may be used to identify a logical block that corresponds to the user data when valid user data is stored in a physical block. A logical block is a collection of multiple sectors in the address space on the host system 4 side, and by managing the

correspondence between this logical block and the physical block, it will be possible to manage the correspondence between the logical address managed by host system 4 and the physical address in flash memory 2. The address space on the side of host system 4 is managed by the Logical Block Address (LBA), which is a number assigned to the area divided into sector (512 bytes) units. Host system 4 specifies an access area within flash memory 2 using this LBA.

(0029)

If no valid user data is stored in the physical block, no logical address information will be written to redundant region 26 of that physical block. Therefore, it is possible to determine whether or not a physical block is an empty block (an erased block) based on whether or not the logical address information has been written to redundant region 26. In other words, if no logical address information has been written, it is determined to be an empty block (erased block).

10

(0030)

The block status is information that indicates whether the physical block is a bad block (a physical block to which data cannot be written normally), and based on this information, it is possible to determine whether it is a bad block or not. Therefore, redundant region 26 of the physical block that is determined to be a bad block will have information showing that it is a bad block set as its block status.

(0031)

This type of flash memory 2 will receive data, address information, or internal commands, etc., from controller 3, and will execute each operation such as the reading, writing, and erasing, etc., of data.

20

(0032)

As shown in Figure 1, controller 3 is provided with microprocessor 6, host interface block 7, work area 8, buffer 9, flash memory interface block 10, ECC (error correcting code) block 11, and Read Only Memory (ROM) 12. Controller 3 that may be constructed of these functional blocks may be integrated on a single semiconductor chip. Each function block will be described below.

(0033)

Microprocessor 6 controls the overall operation of controller 3 according to the program that may be stored in ROM 12. For example, microprocessor 6 reads a command set (sequence command) defining the various processing, etc., from ROM 12 and feeds it to flash memory interface block 10 in order to cause flash memory interface block 10 to execute the processing.

30

(0034)

Host interface block 7 exchanges data, address information, status information, and external commands, etc., to and from host system 4. The data, etc., that may be supplied from host system 4 to flash memory system 1 will be taken within flash memory system 1 (such as buffer 9) using host interface block 7 as the inlet. The data, etc., that may be supplied from flash memory system 1 to host system 4 is also supplied to host system 4 using host interface block 7 as the outlet.

(0035)

Work area 8 is a work area that temporarily stores the data required to control flash memory 2, and is constructed of multiple Static Random Access Memory (SRAM) cells. Work area 8 may store, for example, an address translation table for converting logical and physical addresses, or an empty block search table for searching for empty blocks, etc.

40

(0036)

Buffer 9 temporarily holds the data that was read out from flash memory 2 and the data to be written to flash memory 2 in one sector or multiple sector units. 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.

(0037)

Flash memory interface block 10 is a block that controls the write processing from buffer 9 to flash memory 2 and the read processing from flash memory 2 to buffer 9. The series of control operations when performing write or read operations will be executed based on a sequence command indicating a series of control operations stored in ROM 12.

(0038)

ECC block 11 generates an Error Correcting Code (ECC) that is appended to the data to be written to flash memory 2, and detects and corrects errors contained in the data that was read out based on the ECC that was appended to the data that was read out from flash memory 2.

(0039)

ROM 12 is a non-volatile storage element that stores the programs, etc., that define the procedures for the processing to be executed by microprocessor 6. More specifically, ROM 12 stores the program that defines the processing procedures, such as, for example, creating an address translation table.

(0040)

Next, the method of managing the storage area of flash memory will be described. Due to the increasing storage capacity of flash memory, a method of dividing and managing this storage space into multiple zones has been used in order to smoothly manage the storage space of the large amount of flash memory.

(0041)

Figure 3 is an example of the management of the storage area of flash memory 2 following division into multiple zones (physical zones). Each physical block, which is the erase unit for stored data, is assigned a unique physical block address (PBA). Each physical block may be classified into one of several physical zones, each of which will be assigned a physical zone number (PZN). The number of physical blocks included in each physical zone is called the Physical Zone Internal Block Number (PZIBN). In the example shown in Figure 3, the physical block is assigned a PBA consecutively starting from #0, and a total of 1024 physical blocks with PBA numbers from #0 to #1023 are included within the physical zone of PZN #0, while a total of 1024 physical blocks with PBA numbers from #1024 to #2047 are included in the physical zone of PZN #1, and a total of 1024 physical blocks with PBA numbers from PBA #2048 to #3071 are included in the physical zone of PZN #2.

(0042)

On the other hand, the address space on the host system 4 side is also divided into multiple zones (logical zones), and a single physical zone is assigned to a single logical zone. The data corresponding to each logical block in the logical zone will be written to the physical block in the physical zone assigned to that logical zone. The number of logical blocks in the logical zone will be set to be less than the number of physical blocks in the physical zone. For example, a logical zone containing 1000 logical blocks will be assigned to a physical zone containing 1024 physical blocks. In this way, the reason why the number of logical blocks contained in the logical zone will be less than the number of physical blocks contained in the physical zone is because it takes into account the innate number of bad blocks and the number of blocks that became bad over time.

(0043)

In addition, logical address information corresponding to the data that has been written will be written to the redundant region of the physical block where the data was written. This logical address information may be the logical block number (LBN), which is the number assigned to the logical block, or the logical block number in the logical zone (LZIBN), which is the number within each logical zone of the logical block contained in each logical zone.

(0044)

Next, the procedure of obtaining information (internal information) indicating the internal state of flash memory 2 from the host system 4 side will be described. In the following description, as the internal information, the number of bad blocks and the number of empty blocks per physical zone as shown in Figure 4 will be obtained. Below, the explanation will refer to Figure 4 to Figure 6 when describing the process from the preparation of the information related to the number of bad blocks and the number of empty blocks in each physical zone by memory controller 3 and up to the read out of the information that was prepared from the host system 4 side (up to the transfer of the information that was prepared to the host system 4 side).

(0045)

10

In this example of embodiment, a command code that is not specified for operation in the ATA specifications will be assigned to the command to acquire the number of bad blocks and the number of empty blocks in each physical zone. This type of vendor-specific special command that is not specified in the ATA specifications is called a vendor command. In the following description, the vendor command to acquire the number of bad blocks and the number of empty blocks in each physical zone is referred to as the internal information acquisition command.

(0046)

Referring to Figure 5, the preparation of the internal information and the read out of the internal information that was prepared from the host system 4 side (transfer to the host system 4 side) in the first example of embodiment of the present invention will be described. In the first example of embodiment, first, host system 4 will write the command code of the internal information acquisition command to the command register in memory controller 3 (command register 15a of ATA register 15 shown in Figure 7). Memory controller 3 will initiate the preparation of information regarding the number of bad and the number of empty blocks in each physical zone in response to this internal information acquisition command. In this preparation processing, the block status and logical address information that has been written to the redundant region of each physical block will be read out sequentially, and the number of bad blocks and the number of empty blocks in each physical zone will be counted based on the block status and logical address information that was read out.

20

(0047)

First, the block status and logical address information that has been written to the redundant region of the physical blocks from PBA #0 to #1023 will be read out, and the number of bad blocks and the number of empty blocks in the physical zone of PZN #0 will be counted. Here, for the number of bad blocks, the number of physical blocks with information that indicates that the block status is a bad block will be counted. For the number of free blocks, the number of physical blocks with no information written indicating that the block status is a bad block and with no logical address information written will be counted. When the count of bad blocks and empty blocks in the PZN #0 physical zone has been completed, the number of bad blocks and the number of empty blocks in the PZN #0 physical zone will be written to buffer 9.

30

(0048)

Next, the block status and logical address information that has been written to the redundant region of the physical blocks from PBA #1024 to #2047 will be read out, and the number of bad blocks and the number of empty blocks in the physical zone of PZN #1 will be counted. When the count of bad blocks and empty blocks in the PZN #1 physical zone has been completed, the number of bad blocks and the number of empty blocks in the PZN #1 physical zone will be written to buffer 9. In this way, the number of bad blocks and the number of empty blocks in the physical zones from PZN #0 to #7 will be counted, and the number of bad blocks and the number of empty blocks per physical zone will be written to buffer 9 as the internal information. When counting the number of bad blocks and the number of empty blocks in each physical zone, areas to write the number of bad blocks and the number of empty blocks in each physical zone may be allocated on buffer 9, and "0" may be written as the initial value for the allocated area, with the number increased in increments of 1 for writing to these areas when there has been detection of bad blocks or empty blocks. For example, if "3" has been written to an area to write the number of bad blocks in the physical zone of PZN #1 when a bad block contained in the physical zone of PZN #1 has been detected, the value will be rewritten to be "4".

40

50

(0049)

Next, the internal information that was written to buffer 9 (information on the number of bad blocks and the number of empty blocks in each physical zone) will be read from the host system 4 side (transferred to the host system 4 side).

(0050)

When all of the internal information has been written to buffer 9, memory controller 3 notifies host system 4 of this fact. This notification uses a signal output from the Ready/Busy (R/B) terminal of memory controller 3. In other words, the level of the signal that may be outputted from the Ready Busy (R/B) terminal (high or low level) indicates the Ready or Busy state (hereinafter, the signal indicating the Ready state will be called the Ready signal, while the signal that indicates the Busy state will be called the Busy signal). For example, if the high level is set to the Ready state and the low level is set to the Busy state, the low level (Busy signal) will be outputted from the Ready/Busy (R/B) terminal until the writing of the internal information to buffer 9 has been finished, and the high level (Ready signal) will be outputted from the Ready/Busy (R/B) terminal when the writing of the internal information to buffer 9 has been finished.

10

(0051)

The method of notifying host system 4 that all of the internal information has been written to buffer 9 may be a method of outputting an interrupt signal or rewriting the logical value of a particular bit in status register 15e.

(0052)

When memory controller 3 notifies host system 4 that all of the internal information has been written to buffer 9, host system 4 will alternately input high-level and low-level signals to the read enable (RE) terminal of memory controller 3. Memory controller 3 will transfer the internal information that was written to buffer 9 at the timing of the transition of the level of the signal that may be inputted into the read enable (RE) terminal (from high level to low level, or from low level to high level) to data register 15d. Here, the internal information that was written to buffer 9 will be transferred in units of 1 byte if the number of bits in data register 15d is 8 bits, and in units of 2 bytes if the number of bits in data register 15d is 16 bits. On the other hand, host system 4 will alternately input high-level and low-level signals to the read enable (RE) terminals and will read out the information held in data register 15d.

20

(0053)

In this way, the internal information that was written to buffer 9 will be transferred to data register 15d and read out by host system 4. When all of the internal information that was written to buffer 9 is set in data register 15d, memory controller 3 will output a busy signal from the ready busy (R/B) terminal, and host system 4 will terminate the input of the signal to the read enable (RE) terminal and the retrieval of information held in data register 15d.

30

(0054)

Referring to Figure 6, the preparation of the internal information and the read out of the internal information that was prepared from the host system 4 side (transfer to the host system 4 side) in the second example of embodiment of the present invention will be described. In the second example of embodiment, instead of starting counting the number of bad blocks and the number of empty blocks when an internal information acquisition command has been provided from host system 4, when creating an address translation table and an empty block table, the number of bad blocks and the number of empty blocks are counted in parallel, and the number of bad blocks and the number of empty blocks in each physical zone is held in work area 8.

40

(0055)

Then, when an internal information acquisition command has been provided by host system 4, memory controller 3 will write the internal information that has been held in work area 8 (the information on the number of bad blocks and the number of empty blocks in each physical zone) to buffer 9 and will output the ready signal from the Ready/Busy (R/B) terminal. Subsequently, as in the first example of embodiment, the internal information that was written to buffer 9 will be transferred to data register 15d and read out by host system 4. The reason for counting the number of bad blocks and the number of empty blocks in parallel when preparing the address translation table and empty block table is because the block status and logical address information that has been

50

written to the redundant region in each physical block will be read out when preparing the address translation table and empty block table.

(0056)

Here, address translation table is a table that shows the correspondence between the logical block and the physical block, while the empty block table is a table that shows the physical block address (PBA) of the empty blocks. This address translation table and empty block table may be created for each logical zone. Therefore, there may be instances in which no address translation table or an empty block table is created for a logical zone that was not accessed. Therefore, in this type of case, when an internal information acquisition command has been provided from host system 4, the counting of the number of bad blocks and the number of empty blocks will be performed only in the physical zones corresponding to logical zones in which no address translation table or empty block table will be prepared.

10

(0057)

In this example of embodiment, because the internal information is held in work area 8, as was the case in the first example of embodiment, there is the advantage that it is not necessary to count the number of bad blocks and the number of empty blocks when the internal information acquisition command is provided by host system 4, making it possible to respond quickly to the internal information acquisition command.

(0058)

However, in the case of this example of embodiment, the number of empty blocks and the number of bad blocks will change over time, so it will be necessary to update the internal information with each change. For example, if user data has been written to an empty block, the number of empty blocks in the corresponding physical zone will be decremented (decreased by one), and if the block to which user data was written in the past is erased and has become empty, the number of empty blocks in the corresponding physical zone will be incremented (increased by one). Similarly, in the case of bad blocks, if a new bad block has occurred, the number of bad blocks in the corresponding physical zone will be incremented (increased by one).

20

(0059)

As described above, in the present invention, in response to a command that has been provided by host system 4, memory controller 3 will write internal information to buffer 9, and memory controller 3 will notify host system 4 when all of the internal information has been written to buffer 9. Host system 4 will alternately input high-level and low-level signals to the read enable (RE) terminals in response to this notification, and will read out the information held in data register 15d. Because the internal information that was written to buffer 9 will be transferred to data register 15d at the timing of the transition of the signal input to the read enable (RE) terminal, host system 4 can obtain the internal information in a manner similar to that of reading the data that has been stored in flash memory 2.

30

(0060)

Also, in the present example of embodiment, the internal information was the number of bad blocks and the number of empty blocks in each physical zone, but the internal information is not limited to these types of information. Accordingly, the internal information may be information such as the physical block address (PBA) of a bad block.

40

(0061)

In addition, both of the examples of embodiment described above are illustrative and the present invention is not limited to these examples. The present invention may be implemented in various other variations and modifications. Accordingly, the scope of the present invention is defined only by the claims and their equivalent scope.

(BRIEF DESCRIPTION OF THE DRAWINGS)

(0062)

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

50

(FIGURE 2) This is an illustration showing the relationship between a block of flash memory and a page.

(FIGURE 3) This is an example of the storage area in flash memory that was divided into zones for management.

(FIGURE 4) This is an example of internal information in the example of embodiment of the present invention.

(FIGURE 5) This is an illustration to illustrate the operation of the memory controller in the first example of embodiment of the present invention.

(FIGURE 6) This is an illustration to illustrate the operation of the memory controller in the second example of embodiment of the present invention.

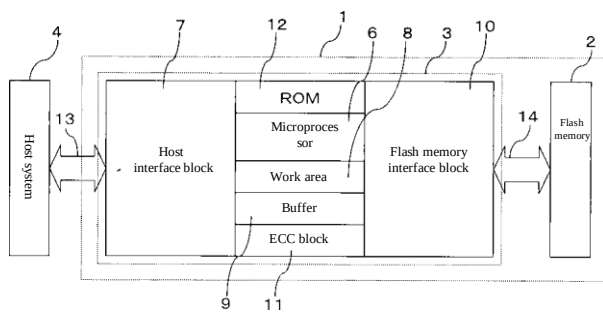
(FIGURE 7) This is a diagram schematically showing the ATA register.

(EXPLANATION OF REFERENCES)

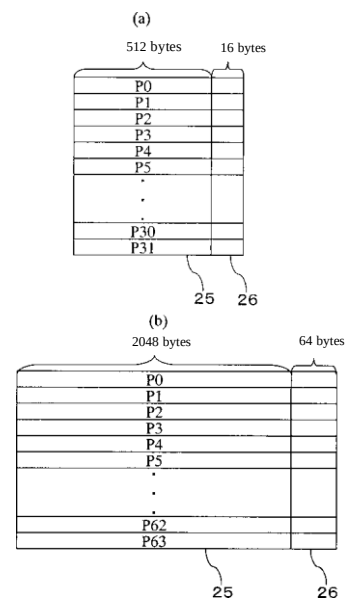
(0063)

1	Flash memory system	
2	Flash memory	
3	Memory controller	10
4	Host system	
6	Microprocessor	
7	Host interface block	
8	Work area	
9	Buffer	
10	Flash memory interface block	
11	ECC block	
12	ROM	
13	External bus	
14	Internal bus	20
15	ATA register	
15a	Command register	
15b	Sector count register	
15c	LBA register	
15d	Data register	
15e	Status register	
25	User region	
26	Redundant region	

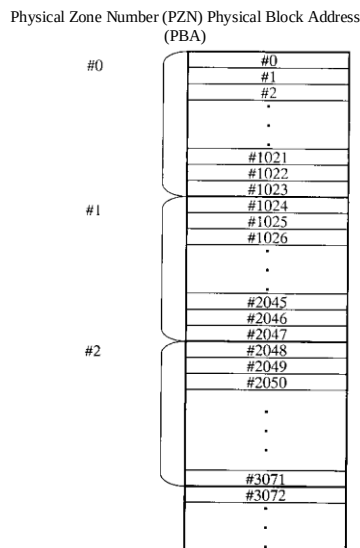
(FIGURE 1)



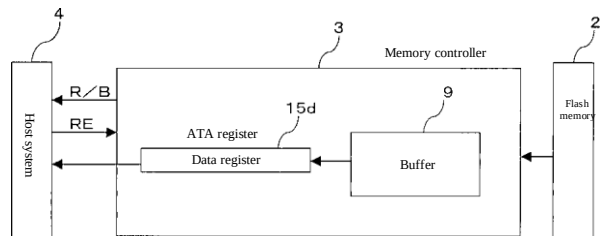
(FIGURE 2)



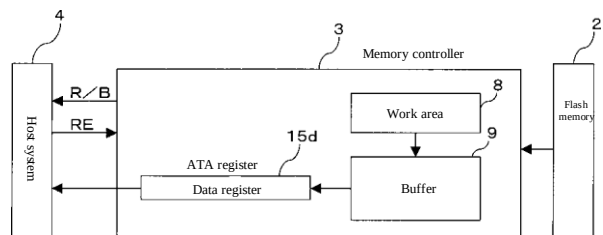
(FIGURE 3)



(FIGURE 5)



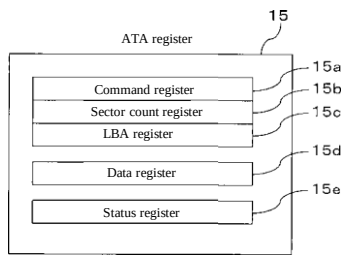
(FIGURE 6)



(FIGURE 4)

PZN	Number of bad blocks	Number of empty blocks
#0	3	10
#1	2	17
#2	4	12
#3	1	11
#4	3	15
#5	2	16
#6	3	14
#7	2	22

(FIGURE 7)



Continued from the front page

F terms (reference) 5B125 CA08 DD03 DD09 DE07 DE08 DE20 EA05 EA10 EK01 EK02
FA04