

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
6 November 2008 (06.11.2008)

PCT

(10) International Publication Number
WO 2008/133373 A1

(51) International Patent Classification:
G06F 17/40 (2006.01)

(21) International Application Number:
PCT/KR2007/003626

(22) International Filing Date: 27 July 2007 (27.07.2007)

(25) Filing Language: Korean

(26) Publication Language: English

(30) Priority Data:
10-2007-0042220 30 April 2007 (30.04.2007) KR

(71) Applicant (for all designated States except US): **SEOUL NATIONAL UNIVERSITY INDUSTRY FOUNDATION** [KR/KR]; San 4-2, Bongchun-dong, Gwanak-gu, Seoul 151-818 (KR).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **LEE, Sang-Won** [KR/KR]; Sungkyunkwan University, of an information and communication, department of engineering, 300 Cheoncheon-dong, Jangan-gu, Suwon-si, Gyeonggi-do 440-330 (KR). **MOON, Bongki** [US/US]; Department of Computer Science, The University of Arizona, P.O.Box 210077, Tucson, AZ 85721-0077 (US). **MIN, Sang Lyul** [KR/KR]; Seoul National University, of Computer Science

and Engineering, San 56-1, Sillim-dong, Gwanak-gu, Seoul 151-010 (KR).

(74) Agent: **MUHANN PATENT & LAW FIRM**; 2, 5, 6th Floor, Myeonglim Bldg., 51-8 Nonhyeon-Dong, Gangnam-gu, Seoul 135-814 (KR).

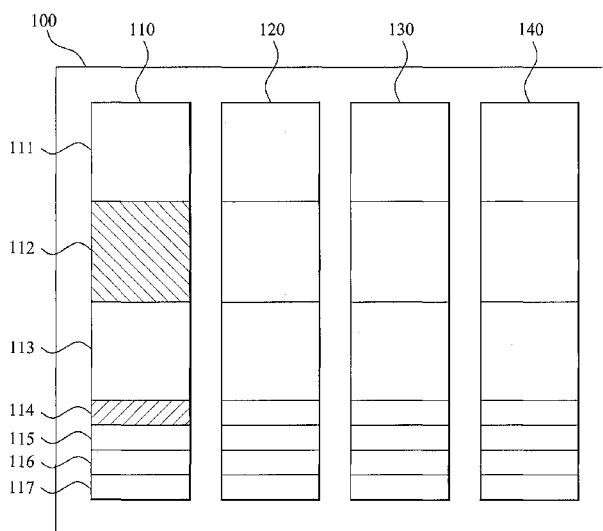
(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— with international search report

(54) Title: DATABASE MANAGEMENT SYSTEM BASED ON IN-PAGE LOGGING AND METHOD USING THE SYSTEM

FIG. 1



(57) Abstract: A non-volatile memory based database management system and method is provided. The database management system includes: a database including a plurality of non-volatile memory erase units; and a database manager allocating a data storage space and a log storage space to each of the erase units, and storing a change log associated with original data of the data storage space in the log storage space of an erase unit which include the data storage space, thereby reducing a non-volatile memory based database access time.

WO 2008/133373 A1

**DATABASE MANAGEMENT SYSTEM BASED ON IN-PAGE LOGGING AND
METHOD USING THE SYSTEM**

Technical Field

5 The present invention relates to a database management system, and more particularly, to a non-volatile memory based data management system, which is represented as a flash memory.

Background Art

10 This work was supported by the IT R&D program of Ministry of Information and Communication (MIC) and Institute of Information Technology Advancement (IITA) Republic of Korea. [2006-S-040-01, Development of Flash Memory-based Embedded Multimedia Software]

15 A database indicates all types of data storage media capable of storing data such as a magnetic disk and a semiconductor memory, and requires a data management system which conforms to different physical features since the data storage media have the different physical features according to their types.

20 A magnetic disk is used for a conventional database. As a general feature of the magnetic disk, the magnetic disk has read/write times of a couple of milliseconds per kilobyte. Also, the magnetic disk has a feature that the read/write times differ since arms' accessing times differ depending on physical locations of stored data.

25 Currently, a non-volatile memory device having shorter read/write times, requiring less power consumption, and having a smaller size in comparison to the magnetic disk is rapidly being substituted for the magnetic disk. This is possible since high capacity of the non-volatile memory device is achieved.

30 The non-volatile memory device is a semiconductor memory device which is electrically capable of reading, writing, and erasing, and is capable of maintaining programmed data when power is not supplied. The operation of storing data with respect to the non-volatile memory device is also referred to as 'programming' as well as 'writing'.

 The non-volatile memory device are variously applied to portable devices such

as a digital camera, a cellular phone, a personal data assistant (PDA), a lap-top computer, and the like, and is capable of storing a great amount of data in a small dimension on a semiconductor chip.

Among non-volatile memory devices, a NAND flash memory is broadly
5 utilized for mass storage devices. The NAND flash memory has a read time of dozens of microseconds per kilobyte and a write time of hundreds of microseconds per kilobyte. Also, a write time and an erase time are generally different from each other in the NAND flash memory, and since the erase time is up to a couple of milliseconds, a method of simultaneously erasing a plurality of memory cells is used to reduce an erase
10 time with respect to all data to be erased. In this instance, a unit of the plurality of memory cells, which are simultaneously erased, is referred to as an erase unit.

As described above, unlike the conventional magnetic disk, the non-volatile memory has a feature of asymmetric times of read/write/erase since required times for reading, writing, and erasing are respectively different in the non-volatile memory
15 which includes the NAND flash memory. Also, it is different from the conventional magnetic disk in that a location where data is stored scarcely affects read/write/erase features since the non-volatile memory electrically accesses data.

In the conventional magnetic disk, a data erase operation is included in a data write operation, therefore an additional erase time is not required. When there is a
20 need to update data, updated data is overwritten in a location of a previous data.

However, in the non-volatile memory, a data erase operation and a data write operation are separate, therefore an additional erase time is required, and when a method applied to the conventional magnetic disk is applied to the non-volatile memory, a great amount of time is required to update data. Specifically, when randomly
25 accessing data and when randomly updating data, a greater amount of time is required.

Accordingly, a more effective database management system is required in a non-volatile memory based database.

Disclosure of Invention

30 Technical Goals

The present invention provides a database management system and method in a non-volatile memory based database which requires less time to update data even when

the data is randomly accessed.

The present invention also provides a non-volatile memory based database management system and method which can respond to approval and abortion of a transaction without an additional overhead.

5 The present invention also provides a non-volatile memory based database management system and method which can be applied to portable devices due to less power consumption.

Technical solutions

10 According to an aspect of the present invention, there is provided a database management system including: a database including a plurality of non-volatile memory erase units; and a database manager allocating a data storage space and a log storage space to each of the erase units, and storing a change log associated with original data of the data storage space in the log storage space of an erase unit which
15 include the data storage space.

 According to another aspect of the present invention, there is provided a database management method including: allocating a data storage unit and a log storage space to each of erase units of a database including a plurality of non-volatile memory erase units; and storing a change log associated with original data of the data
20 storage space in the log storage space of an erase unit which includes the data storage space.

Brief Description of Drawings

 FIG. 1 is a diagram illustrating a database of a database management system
25 according to an exemplary embodiment of the present invention;

 FIG. 2 is a diagram a data flow of a database and a database buffer of a database management system according to an exemplary embodiment of the present invention;

 FIG. 3 is a block diagram illustrating a database management system according to an exemplary embodiment of the present invention;

30 FIG. 4 is a flowchart illustrating a database management method according to an exemplary embodiment of the present invention; and

 FIG. 5 is a flowchart illustrating merging operations provided by a database

management method according to an exemplary embodiment of the present invention.

Best Mode for Carrying Out the Invention

Hereinafter, embodiments of the present invention will be described in the
5 accompanying drawings.

FIG. 1 is a diagram illustrating a database 100 of a database management system according to an exemplary embodiment of the present invention.

Referring to FIG. 1, the database 100 of the database management system according to the exemplary embodiment of the present invention includes four erase
10 units 110, 120, 130, and 140. A data storage space and a log storage space are allocated to each of the four erase units 110, 120, 130, and 140.

There are three pages of data storage spaces 111, 112, and 113 and four sectors of log storage spaces 114, 115, 116, and 117 in the first erase unit 110.

Generally, in a non-volatile memory, a plurality of memory cells are
15 simultaneously erased when erasing the memory cells. In this instance, a minimum collection of the simultaneously erased plurality of memory cells is referred to as an erase unit. That is, memory cells in erase units are simultaneously erased in the erase units.

When an update to update contents of original data stored in the second data
20 storage space 112 included in the first erase unit 110 is requested by a processor, the original data is maintained, and a change log associated with the original data is stored in the first log storage space 114.

In this instance, when the processor accesses the original data, current data is calculated based on the original data and the change log associated with the original
25 data which is stored in the first log storage space 114. The calculated current data is outputted to the processor.

In the specification of the present invention, a database management method provided by the database management system of the present invention is referred to as an in-page logging method. In the database management method provided by the
30 database management system of the present invention, an erase unit includes a data storage space and a log storage space.

The change log is sequentially stored in a free space of the log storage space.

The change log includes address information of the original data corresponding to the change log.

When a log storage space has no free space to store the change log in the first erase unit 110, an available erase unit of other erase units 120, 130, and 140 is allocated, the current data corresponding to the original data of the first erase unit 110 is stored in the allocated erase unit, and the first erase unit 110 is erased. The erased first erase unit 110 switches to a storable state. The above operation is referred to as a merge operation.

Generally, a non-volatile memory has a feature of a uniform random access speed for data to be read regardless of an address of stored data.

Thus, according to the database management system of the present invention, a data access speed may not be affected even when having a log storage space scattered all over non-volatile memory area, without having a contiguous log storage space.

Also, the non-volatile memory generally has an asymmetric read/write speed, that is a data storage time is longer than a data read time, and a data erase time is longer than the data storage time.

Accordingly, it takes a great amount of time for a general non-memory management system to erase a stored data and to store the data which will be updated in an erased storage space to update the previously stored data. According to the database management system of the present invention, a data erase operation is not required and a data update time may be reduced since the stored data is not directly updated when updating the stored data and required information for the updating are sequentially stored in a log storage space of an erase unit, which includes the stored data.

Also, the in-page logging method of the present invention enables the merge operation, and does not require individual erase operations with respect to the data storage space and the log storage space due to the merge operation, thereby updating data via a single erase operation. Accordingly, overhead time may be minimized even when a log storage space has no free space.

A required time for the merge operation is calculated by,

[Equation 1]

$$C_{\text{merge}} = (K_d + K_l) \times C_{\text{read}} + K_d \times C_{\text{write}} + C_{\text{erase}}$$

where C_{merge} indicates a required time for the merge operation, K_d indicates a

number of sectors of a data storage space within a first erase unit 110, K_1 indicates a number of sectors of a log storage space within a first erase unit 110, C_{read} indicates a required time for a single read, C_{write} indicates a required time for a single write, and C_{erase} indicates a required time for a single erase.

5 Generally, a transaction indicating an update request for original data in a database 100 may be approved or aborted.

When the transaction is approved, the transaction is confirmed, and a result of the transaction is maintained. Conversely, when the transaction is aborted, a result of the transaction is required to be recovered to a state prior to the transaction.

10 When approval or abortion with respect to the transaction is not determined, it is referred to as an activated transaction.

According to the in-page logging method of the present invention, the result of the transaction may be applied using a minimized overhead without considering whether the transaction is approved or aborted. When the transaction is approved, the result of the transaction is confirmed without an additional operation. When the transaction is aborted, a change log corresponding to the aborted transaction is checked to be ignored.

FIG. 2 is a diagram of a data flow of a database and a database buffer 220 of a database management system according to an exemplary embodiment of the present invention.

Referring to FIG. 2, the database buffer 220 of the database management system according to the exemplary embodiment of the present invention includes a plurality of data storage spaces 221 and a plurality of log storage spaces 222.

25 An erase unit 210 includes a plurality of data storage space 211 and a plurality of log storage space 212.

The database buffer 220 stores one of original data, which is accessed by a processor, as in-memory data, in the data storage spaces 221.

When an update for the in-memory data is requested by the processor, the database management system in-place updates, the in-place updating overwrites the in-memory data in an in-memory data location. Also, the database management system stores an update log with respect to the updated in-memory data in the log storage spaces 222 of the database buffer 220.

When another original data in the database is accessed by the processor, the other accessed original data is stored in the data storage spaces 221 of the database buffer 220. When the data storage spaces 221 has no free space to store the update log, any one of the in-memory data stored in the data storage spaces 221 is evicted to the database instead of the accessed another original data, and the other accessed original data is stored as a new in-memory data in an area where the evicted in-memory data is stored.

It is assumed that the other accessed original data of the database corresponding to the evicted data is stored in the erase unit 210, a change log is stored in the log storage space 212 of the erase unit 210 from an update log associated with the evicted data, which is stored in the log storage spaces 222 of the database buffer 220. The evicted data is not required to be directly stored in the database since the change log is stored in the database.

According to the database management system of the present invention, since the change log is stored in the log storage space 212 of the erase unit 210 only when data evicted from the database buffer 220, a number of storing operations may be reduced with respect to the database. Since a storing operation with respect to the non-volatile memory generally requires a great amount of power, the database management system of the present invention may be easily applied to portable applications due to the reduced number of the storing operations.

It is assumed that the original data of the database corresponding to the in-memory data is stored in the erase unit 210 when the update for the in-memory data is requested by the processor and when the log storage spaces 222 of the database buffer 220 has no free space to store the update log, a change log is stored in the log storage space 212 of the erase unit 210 from an update log associated with the in-memory data. The update logs are required to be stored in the database when updates frequently occur with respect to the in-memory data since a defined number of update logs are stored in the log storage unit 222 of the database buffer 220.

According to the database management system of the present invention, since a change log is stored in the log storage space 212 of the erase unit 210 only when the log storage space 222 of the database buffer 220 has no free space to store the change log, a number of storing operations with respect to the database may be reduced. In this

instance, since the database management system of the present invention may reduce a number of storing operations with respect to a non-volatile memory and reduce power consumption, the database management system of the present invention may be easily applied to portable applications.

5 FIG. 3 is a block diagram illustrating a database management system according to an exemplary embodiment of the present invention.

Referring to FIG. 3, the database management system according to the exemplary embodiment of the present invention includes a database 310, a database manager 320, a database buffer 330, and a buffer manager 340.

10 The database 310 includes a plurality of erase units.

An erase unit in the non-volatile memory indicates a minimum collection of memory cells which are simultaneously erased.

The database manager 320 allocates a data storage space and a log storage space to each of the erase units, and stores a change log associated with original data of the data storage space in the log storage space of an erase unit, which includes the data storage space.

In this instance, the database manager 320 maintains the original data when an update for the original data is requested by a processor, and generates the change log associated with the original data according to the requested update for the original data.

20 The database manager 320 may store the generated change log in the log storage space of the erase unit, which includes the data storage space.

In this instance, when the update for the original data is requested from the processor, the database management system may calculate current data based on the original data and the change log associated with original data, and output the calculated current data to the processor. The calculating and outputting operations may be performed by the database manager 320.

In this instance, when the log storage space has no free space to store the change log, the database manager 320 may allocate another available erase unit, store the current data corresponding to the original data of a previous erase unit in the allocated erase unit, and erase the previous erase unit.

30

In this instance, the database manager 320 may select the other available erase unit based on a number of used times of each of erase units when allocating the other available erase unit.

5 A non-volatile memory generally uses a wear-leveling to increase life span. In the memory cell of the non-volatile memory, when storing and erasing operations are repeated, the memory cell may no longer be used since characteristics of the memory cell of the non-volatile memory may degrade, therefore management for a number of the storing and erase operations is required. Ideally, it may be desirable to repeat a same number of the storing operation as a same number of the erase operation.

10 Accordingly, the database manager 320 selects the other available erase unit based on the number of used times of each erase units when allocating the available erase unit, thereby enabling a number of used times of all memory cells in the database 310 to be uniform, and expanding a life span of the database 310.

15 In this instance, the database manager 320 may generate current data based on the original data and the change log associated with the original data of the previous erase unit, except for an active change log which corresponds to an activated transaction from the change log.

20 The database manager 320 applies a change log which corresponds to an approved transaction from the change logs to calculate the current data, and disregards another change log which corresponds to an aborted transaction. The database manager 320 determines whether to apply a change log, approved or aborted, to calculate the current data since it is not clear whether to approve the change log which corresponds to the activated transaction.

25 In this instance, when the log storage space has no free space to store the change log and when a ratio of an active change log, which corresponds to the activated transaction, to the change log exceeds a reference ratio, the database manager 320 allocates the other available erase unit as an overflow erase unit, and additionally stores the change log in the overflow erase unit.

30 Since efficiency of the merge operation is not high when the activated transaction exceeds the reference ratio, the database manager 320 allocates the other available erase unit as the overflow erase unit instead of performing the merge operation, and stores a change log in the overflow erase unit afterward.

The database buffer 330 stores one of original data, which is accessed by the processor, as in-memory data.

In this instance, the database buffer 330 may be a volatile memory having a shorter access time than the database 310. As an example, the database buffer 330
5 may be a dynamic random access memory (RAM) or a static RAM.

The buffer manager 340 updates the in-memory data when an update for the in-memory data is requested by the processor, stores the updated in-memory data in the database buffer 330, and stores an update log about the updated in-memory data in another space of the database buffer 330.

10 In this instance, when the in-memory data is evicted from the database buffer, the database manager 320 stores a change log associated with the in-memory data in the log storage space of the erase unit, which includes the data storage space where the original data is stored.

In this instance, when an update for the in-memory data is requested by the
15 processor and when the database buffer 330 has no free space to store the update log, the database manager 320 stores the change log associated with the in-memory data in the log storage space of the erase unit which includes the data storage space where the original data is stored, the original data corresponding to the in-memory data.

FIG. 4 is a flowchart illustrating database management method according to an
20 exemplary embodiment of the present invention.

Referring to FIG. 4, the database management method according to the exemplary embodiment of the present invention allocates a data storage space and a log storage space to each of erase units of a database, which includes a plurality of non-volatile memory erase units in operation S410.

25 The database management method stores a change log associated with original data of the data storage space in the log storage space of an erase unit, which include the data storage space in operation S420.

In this instance, when an update for the original data is requested by the processor, the database management method maintains the original data and stores the
30 change log associated with the original data in the log storage space of the erase unit, which includes the data storage space where the original data is stored.

When the processor accesses the original data, the database management

method calculates current data based on the original data and the change log associated with original data in operation S430.

The database management method outputs the calculated current data to the processor in operation S440.

5 According to another exemplary embodiment of the present invention, when the log storage space has no free space to store the change log, a database management method allocates another available erase unit, stores the current data corresponding to the original data of a previous erase unit in the allocated erase unit, and erases the previous erase unit.

10 In this instance, the database management method may select the other available erase unit based on a number of used times of each of erase units when allocating the other available erase unit.

FIG. 5 is a flowchart illustrating merging operations provided by a database management method according to an exemplary embodiment of the present invention.

15 Referring to FIG. 5, the database management method according to the exemplary embodiment of the present invention allocates another available erase unit when a log storage space has no free space to store a change log in operation S510.

 In this instance, the database management method may select the other available erase unit based on a number of used times of each of erase units when
20 allocating the other available erase unit.

The database management method determines whether a change log associated with original data of a previous erase unit exists in operation S520.

 When the change log associated with the original data of the previous erase unit exists, the database management method calculates current data based on the original
25 data and the change log associated with original data in operation S530.

 In this instance, the database management method may generate the current data based on the original data and the change log associated with the original data of the previous erase unit, except for an active change log which corresponds to an activated transaction from the change log.

30 The database management method stores the calculated current data in the allocated erase unit in operation S540.

The database management method stores the original data in the allocated erase

unit when the change log associated with the original data does not exist.

The database management method erases the previous erase unit after storing the current data in the allocated erase unit in operation S550.

A database management method according to another exemplary embodiment
5 of the present invention stores one of original data, which is accessed by a processor, as in-memory data in a database buffer.

The database management method updates the in-memory data when an update for the in-memory data is requested by the processor, stores the updated in-memory data in the database buffer, and stores the update log about the updated in-memory data in
10 another space of the database buffer.

In this instance, only when the in-memory data is evicted from the database buffer, the database management method may store a change log associated with the in-memory data in the log storage space of the erase unit which includes the data storage space where the original data is stored, the original data corresponding to the in-memory
15 data.

In this instance, when the update for the in-memory data is requested by the processor and when the database buffer has no free space to store the update log, the database management method stores the change log associated with the in-memory data in the log storage space of the erase unit which includes the data storage space where
20 the original data is stored, the original data corresponding to the in-memory data.

According to another exemplary embodiment of the present invention, when the log storage space has no free space to store the change log and when a ratio of an active change log, corresponding to the activated transaction, to the change log exceeds a reference ratio, the database management method allocates the other available erase unit
25 as an overflow erase unit, and additionally stores the change log in the overflow erase unit.

The database management method according to the above-described exemplary embodiments may be recorded in computer-readable media including program instructions to implement various operations embodied by a computer. The media
30 may also include, alone or in combination with the program instructions, data files, data structures, and the like. Examples of computer-readable media include magnetic media such as hard disks, floppy disks, and magnetic tape; optical media such as CD

ROM disks and DVD; magneto-optical media such as optical disks; and hardware devices that are specially configured to store and perform program instructions, such as read-only memory (ROM), random access memory (RAM), flash memory, and the like. The media may also be a transmission medium such as optical or metallic lines, wave
5 guides, and the like, including a carrier wave transmitting signals specifying the program instructions, data structures, and the like. Examples of program instructions include both machine code, such as produced by a compiler, and files containing higher level code that may be executed by the computer using an interpreter. The described hardware devices may be configured to act as one or more software modules in order to
10 perform the operations of the above-described embodiments of the present invention.

According to the present invention, it is possible to implement a database management system and database management method in a non-volatile memory based database which requires less time to update data even when the data is randomly accessed.

15 Also, according to the present invention, it is possible to implement a database management system and database management method which can respond to approval and abortion of a transaction without an additional overhead.

Also, according to the present invention, it is possible to embody a non-volatile memory based database management system and method which can be applied to
20 portable devices due to less power consumption.

Although a few embodiments of the present invention have been shown and described, the present invention is not limited to the described embodiments. Instead, it would be appreciated by those skilled in the art that changes may be made to these embodiments without departing from the principles and spirit of the invention, the scope
25 of which is defined by the claims and their equivalents.

CLAIMS

1. A database management system, comprising:
a database including a plurality of non-volatile memory erase units; and
a database manager allocating a data storage space and a log storage space to
5 each of the erase units, and storing a change log associated with original data of the data storage space in the log storage space of an erase unit which include the data storage space.
2. The system of claim 1, wherein, when a processor accesses the original data,
10 the database management system calculates current data based on the original data and the change log associated with the original data, and outputs the calculated current data to the processor.
3. The system of claim 1, wherein, when an update for the original data is
15 requested by the processor, the database manager maintains the original data, and stores the change log associated with the original data in the log storage space of the erase unit which includes the data storage space where the original data is stored.
4. The system of claim 1, wherein the database management system further
20 comprises:
a database buffer storing one of the original data, which is accessed by the processor, as in-memory data; and
a buffer manager updating the in-memory data when an update for the in-memory data is requested by the processor, storing the updated in-memory data in the
25 database buffer, and storing an update log about the updated in-memory data in another space of the database buffer.
5. The system of claim 4, wherein, when the in-memory data is evicted from the database buffer, the database manager stores a change log associated with the in-memory data in the log storage space of the erase unit which includes the data storage
30 space where the original data is stored, the original data corresponding to the in-memory data.

6. The system of claim 4, wherein, when an update for the in-memory data is requested by the processor and when the database buffer has no free space to store the update log, the database manager stores the change log associated with the in-memory
5 data in the log storage space of the erase unit which includes the data storage space where the original data is stored, the original data corresponding to the in-memory data.

7. The system of claim 1, wherein, when the log storage space has no free space to store the change log, the database manager allocates another available erase unit, stores
10 the current data corresponding to the original data of a previous erase unit in the allocated erase unit, and erases data from the previous erase unit.

8. The system of claim 7, wherein the database manager selects the other available erase unit based on a number of used times of each of the erase units.

15

9. The system of claim 7, wherein the database manager generates the current data based on the original data and the change log associated with the original data of the previous erase unit, except for an active change log which corresponds to an activated transaction from the change log.

20

10. The system of claim 1, wherein, when the log storage space has no free space to store the change log and when a ratio of an active change log, corresponding to the activated transaction, to the change log exceeds a reference ratio, the database manager allocates the other available erase unit as an overflow erase unit, and additionally stores
25 the change log in the overflow erase unit.

11. A database management method, comprising:
allocating a data storage unit and a log storage space to each of erase units of a database including a plurality of non-volatile memory erase units; and
30 storing a change log associated with original data of the data storage space in the log storage space of an erase unit which includes the data storage space.

12. The method of claim 11, wherein, when a processor accesses the original data, the database management method calculates current data based on the original data and the change log associated with original data, and outputs the calculated current data to the processor.

5

13. The method of claim 11, wherein, when an update for the original data is requested by the processor, the database management method maintains the original data and stores the change log associated with the original data in the log storage space of the erase unit which includes the data storage space where the original data is stored.

10

14. The method of claim 11, further comprising:

storing one of the original data, which is accessed by the processor, as in-memory data; and

15 updating the in-memory data when an update for the in-memory data is requested by the processor, storing the updated in-memory data in the database buffer, and storing an update log about the updated in-memory data in another space of the database buffer.

15. The method of claim 14, wherein, when the in-memory data is evicted from the database buffer and when the database buffer has no free space to store the update log, the database management method stores the change log associated with the in-memory data in the log storage space of the erase unit which includes the data storage space where the original data is stored, the original data corresponding to the in-memory data.

25 16. The method of claim 14, wherein, when an update for the in-memory data is requested by the processor and when the database buffer has no free space to store the update log, the database management method stores the change log associated with the in-memory data in the log storage space of the erase unit which includes the data storage space where the original data is stored, the original data corresponding to the in-memory data.

30

17. The method of claim 11, wherein, when the log storage space has no free space to store the change log, the database management method allocates another available erase unit, stores the current data corresponding to the original data of a previous erase unit in the allocated erase unit, and erases data from the previous erase unit.

5

18. The method of claim 17, wherein the database management method selects the other available erase unit based on a number of used times of each of the erase units.

19. The method of claim 17, wherein the database management method generates
10 the current data based on the original data and the change log associated with the original data of the previous erase unit, except for an active change log which corresponds to an activated transaction from the change log.

20. The method of claim 11, wherein, when the log storage space has no free space
15 to store the change log and when a ratio of an active change log, corresponding to the activated transaction, to the change log exceeds a reference ratio, the database management method allocates the other available erase unit as an overflow erase unit, and additionally stores the change log in the overflow erase unit.

20 21. At least one medium comprising computer readable instructions implementing the method of any one of claims 11 through 20.

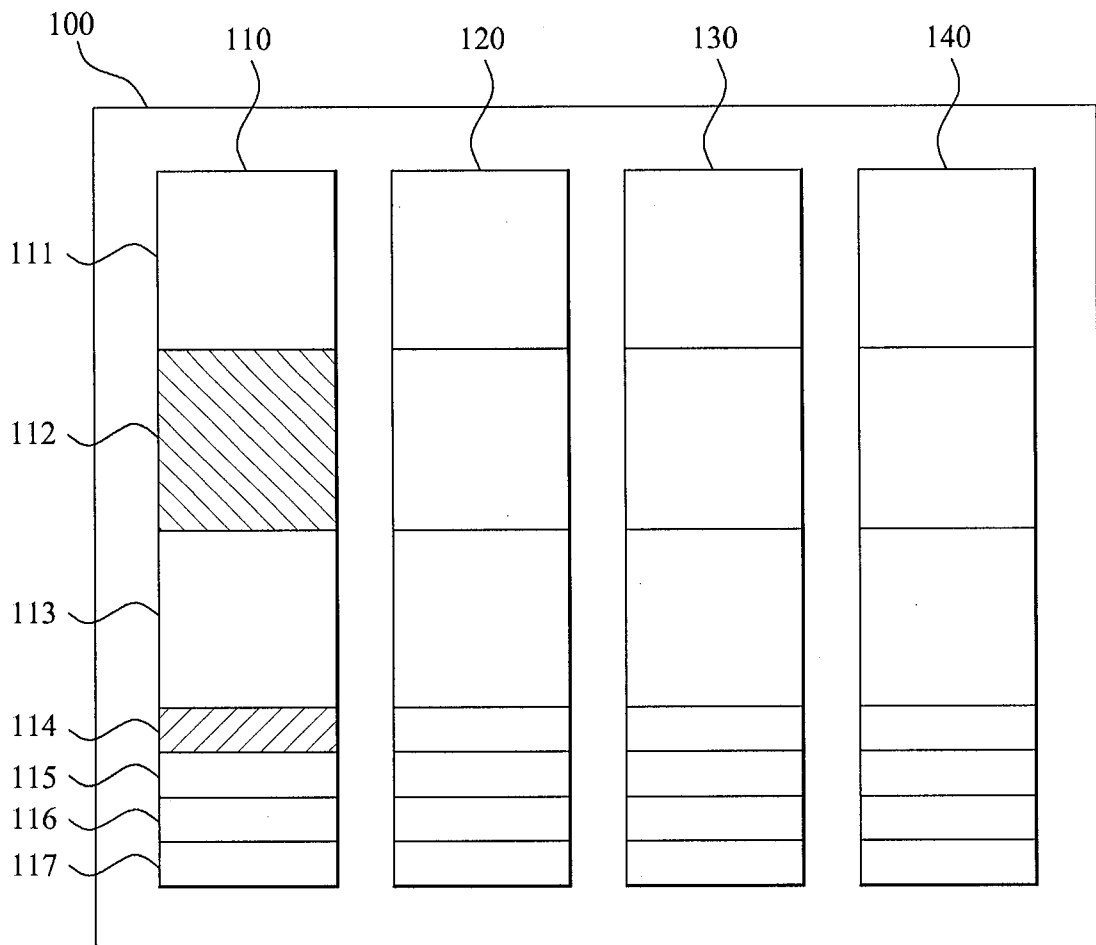
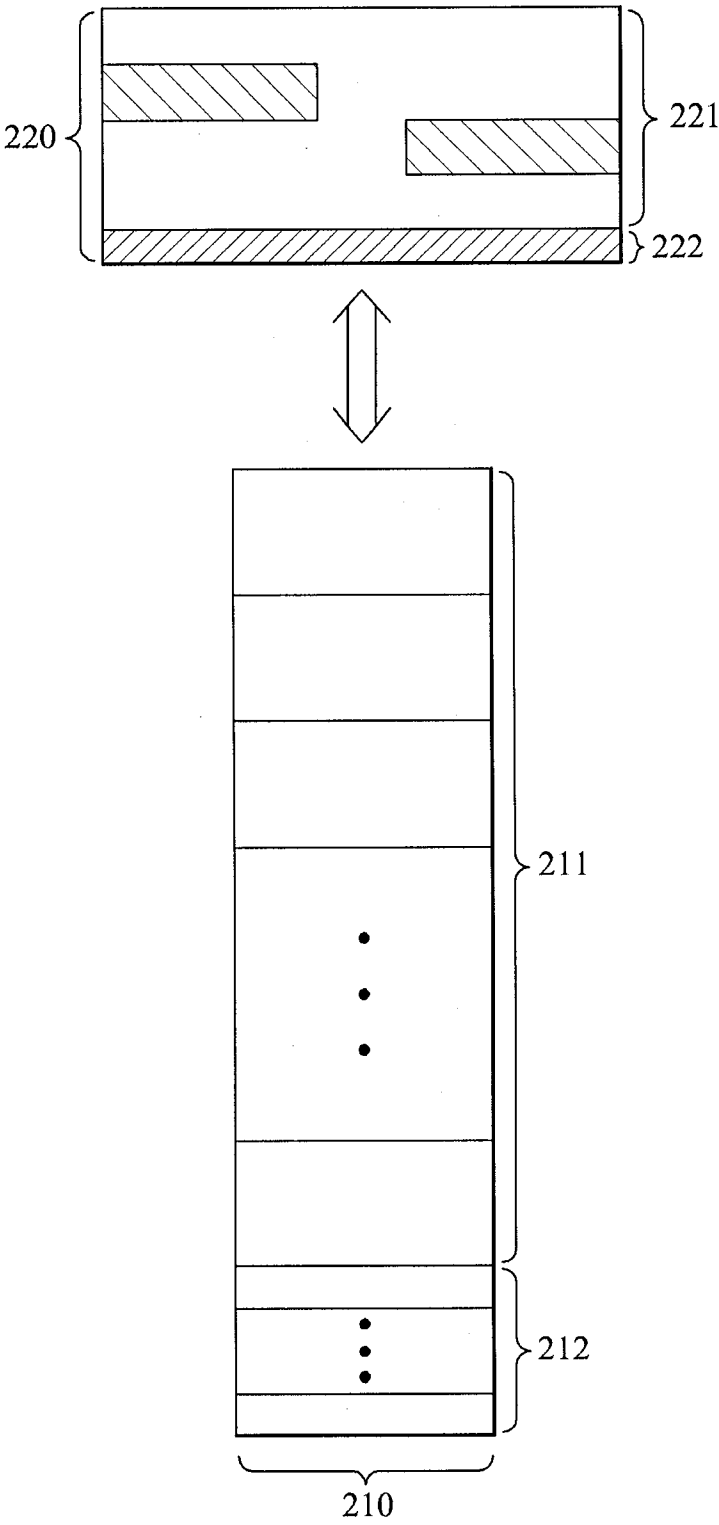
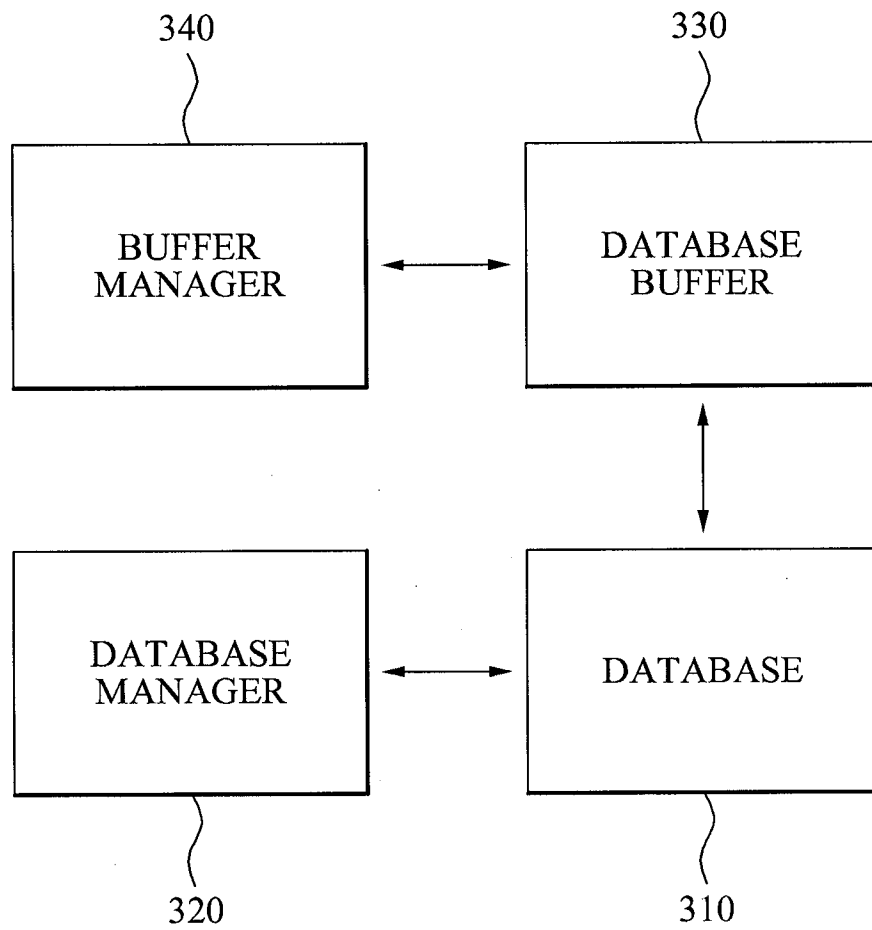
FIG. 1

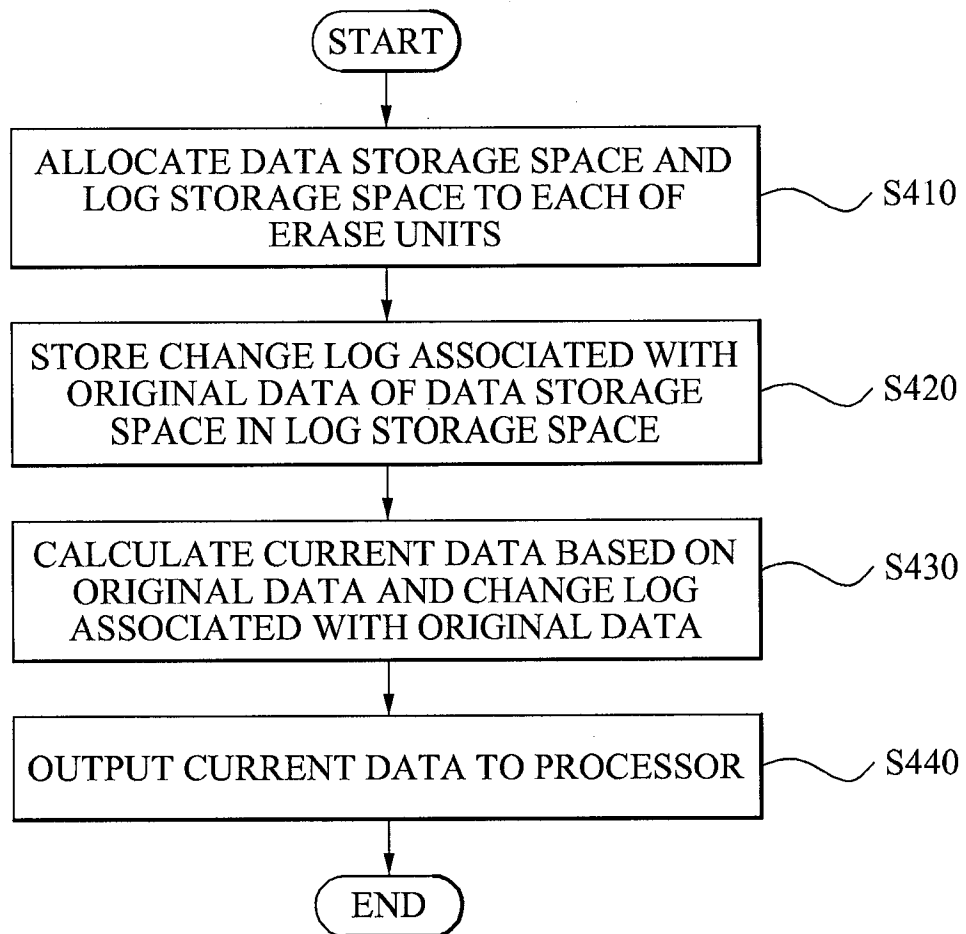
FIG. 2



3/5

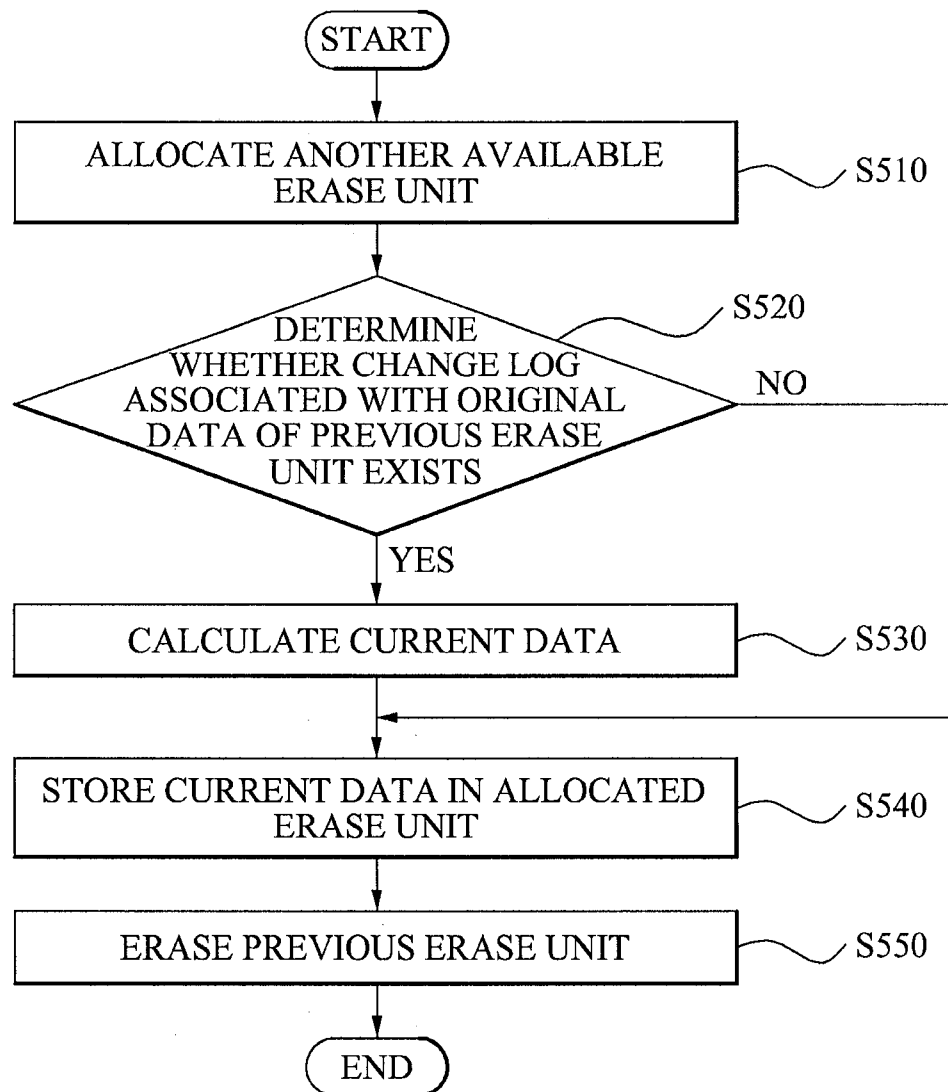
FIG. 3

4/5

FIG. 4

5/5

FIG. 5



INTERNATIONAL SEARCH REPORT

International application No.
PCT/KR2007/003626**A. CLASSIFICATION OF SUBJECT MATTER****G06F 17/40(2006.01)i**

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 8 : G06F

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

Korean Utility models and applications for Utility models since 1975

Japanese Utility models and applications for Utility models since 1975

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

PAJ, FPD, USPAT, eKIPASS, IEEE, YAHOO, GOOGLE , Keyword: "NAND Flash, DBMS, in-page logging"

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	GYE-JEONG KIM et al. 'LGeDBMS: A Small DBMS for Embedded System with Flash Memory' . In Proceedings of the 32nd International Conference on Very Large Data Bases, Seoul, Korea, September 12-15, 2006, pages 1255 - 1258. ACM, 2006 (See Abstract, Section 2. Overview of LGeDBMS)	1-21
A	CHRISTOPHE BOBINEAU et al. 'PicoDBMS: Scaling Down Database Techniques for the Smartcard' . In Proceedings of the 26th VLDB Conference, pages 11.20, Cairo, Egypt, September 2000. (See Abstract, Section4.PicoDBMS storage model)	1-21
A	HUI DAI et al., 'ELF:An Efficient Log-Structured Flash File System for Micro Sensor Nodes' . In the Second International Conference on Embedded Networked Sensor Systems(SenSys'03), pages 176-187, Baltimore, MD, USA, November 2004. (See Abstract, Section3.Design Overview)	1-21
A	US 07139199 B2 (SUJAYA SRINIVASAN et al) 21 November 2006 (See Abstract, Figs.3-5, column 3 line 15 - column 5 line 39)	1-21
A	US 2005/0271377 A1 (SEONG-JUN HONG) 08 December 2005 (See Abstract, Figs.4, paragraph [0037]-[0040])	1-21



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search

21 JANUARY 2008 (21.01.2008)

Date of mailing of the international search report

21 JANUARY 2008 (21.01.2008)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
920 Dunsan-dong, Seo-gu, Daejeon 302-701,
Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

Sun Dong Guk

Telephone No. 82-42-481-8248



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/KR2007/003626

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US07139199B2	21. 11. 2006	US20050286306A1	29. 12. 2005
US20050271377A1	08. 12. 2005	KR2005116042A	09. 12. 2005