(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
12 June 2014 (12.06.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/088682 A1

(51) International Patent Classification:
G06F 12/16 (2006.01) *G06F 11/08* (2006.01)

(21) International Application Number:
PCT/US2013/061608

(22) International Filing Date:
25 September 2013 (25.09.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13/708,873 7 December 2012 (07.12.2012) US

(71) Applicant: WESTERN DIGITAL TECHNOLOGIES, INC. [US/US]; 3355 Michelson Drive, Suite 100, Irvine, CA 92612 (US).

(72) Inventors: **SUN, Yongke**; c/o Western Digital Technologies, Inc., 3355 Michelson Drive, Suite 100, Irvine, CA 92612 (US). **ZHAO, Dengtao**; c/o Western Digital Technologies, Inc., 3355 Michelson Drive, Suite 100, Irvine, CA 92612 (US). **YANG, Jui-yao**; c/o Western Digital Technologies, Inc., 3355 Michelson Drive, Suite 100, Irvine, CA 92612 (US).

(74) Agent: **ALTMAN, Daniel, E.**; Knobbe Martens Olson & Bear, LLP, 2040 Main Street, 14th Floor, Irvine, CA 92614 (US).

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

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *with international search report (Art. 21(3))*

(54) **Title**: SYSTEM AND METHOD FOR LOWER PAGE DATA RECOVERY IN A SOLID STATE DRIVE

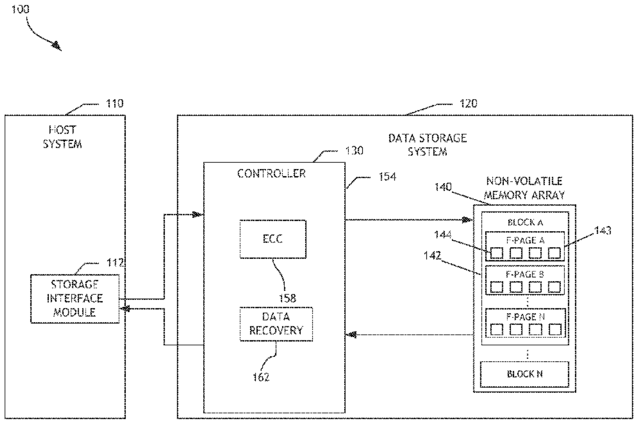


FIGURE 1

(57) **Abstract**: In some embodiments of the present invention, a data storage system includes a controller and a non-volatile memory array having a plurality of memory pages. The controller performs a method that efficiently resolves the lower page corruption problem. In one embodiment, the method selects programmed lower page(s) for which paired upper page(s) have not been programmed, reads data from those selected lower page(s), corrects the read data, and reprograms the read data into those lower page(s). Since the number of lower pages in this condition is typically low (e.g., several pages in a block with hundreds or thousands of pages), this is a much more efficient method than reprogramming the entire block, in another embodiment, a similar reprogramming method is applied as a data recovery scheme in situations in which only lower pages are programmed (e.g., SLC memory, MLC memory in SLC mode, etc.).

SYSTEM AND METHOD FOR LOWER PAGE DATA RECOVERY IN A SOLID STATE DRIVE

BACKGROUND

Technical Field

This disclosure relates to data storage systems, such as solid state drives, for computer systems. More particularly, the disclosure relates to lower page data recovery.

Description of the Related Art

Memory arrays with multi-level cell (MLC) NAND media are now commonplace in solid state drives (SSDs). MLC allows multiple possible states to exist in a single memory cell, thereby enabling the storing of more than one bit of information per cell (e.g., 2, 3, 4 or more). For example, in a 2-bit-per cell MLC flash, 4 possible states (4 voltage (Vt) levels) are possible which enable storage of 2 bits. Based on the data the memory cells are going to store and the coding for different states, the cells are programmed to 4 possible and distinctive Vt zones. Typically, data stored in lower pages and upper pages are logically paired together, with the lower pages being programmed first.

BRIEF DESCRIPTION OF THE DRAWINGS

Systems and methods that embody the various features of the invention will now be described with reference to the following drawings, in which:

Figure 1 illustrates a storage system that reprograms data according to one embodiment of the invention.

Figures 2A and 2B show the voltage distribution of memory cells to illustrate the factors that contribution to lower page corruption and a solution according to one embodiment of the invention.

Figure 3 is a flow diagram showing a method of reprogramming according to one embodiment of the invention.

Figures 4A-4B and 5 show the different memory cell configuration in which open lower pages may be selected for reprogramming according to embodiments of the invention.

Figure 6 is a flow diagram illustrating a process of reprogramming lower page programmed only cells according to one embodiment of the invention.

DETAILED DESCRIPTION OF SPECIFIC EMBODIMENTS

While certain embodiments are described, these embodiments are presented by way of example only, and are not intended to limit the scope of protection. Indeed, the novel methods and systems described herein may be embodied in a variety of other forms. Furthermore, various omissions, substitutions, and changes in the form of the methods and systems described herein may be made without departing from the scope of protection.

Overview

In MLC flash memory, even though the lower pages and upper pages are physically paired, they are generally decoupled in programming. The data stored in lower pages and upper pages can be programmed at different time and from different sources. There are often cases where the upper pages are programmed much later and at different temperature. Most time, there are no problems for this programming scheme. However, various factors including temperature and aging effects of the memory cells can degrade retention of the data in the cells and may lead to the problem of lower page corruption when the storage system attempts to program the upper page.

One way to overcome the lower page corruption problem is to move the old data of the entire open block to another new address when there is a need to program the upper page. In this manner, the lower page and upper page are programmed at the same time under the similar condition. However, this method is inefficient.

In some embodiments of the present invention, a data storage system includes a controller and a non-volatile memory array having a plurality of memory pages. The controller performs a method that may require less system load while still preventing the lower page corruption problem discussed above. In one embodiment, the method selects programmed lower page(s) for which paired upper page(s) have not been programmed, reads data from those selected lower page(s), corrects the read data, and reprograms the read data into those lower page(s). Since the number of lower pages in this condition is typically low (e.g., several pages in a block with hundreds or thousands of pages), this is a much more efficient method than reprogramming the entire block. In another embodiment, a similar reprogramming method is applied as a data recovery scheme in situations in which only lower pages are programmed (e.g., SLC (single-level cell) memory, MLC memory in SLC mode, etc.).

System Overview

Fig. 1 illustrates a storage system 120 that performs reprogramming for data recovery according to one embodiment of the invention. As is shown, a storage system 120 (e.g., solid state drive, hybrid hard drive, etc.) includes a controller 130 and a non-volatile memory array 140, which includes one or more blocks of memory storage, identified as Block “A” 142 through Block “N”. Each block includes flash pages (F-pages). For example, Block A 142 of Figure 1 includes F-pages identified as F-pages A 153, B, through N. In some embodiments, each “F-page” is a smallest grouping of memory cells in the non-volatile memory array 140 that can be programmed in a single operation or as a unit. Further, each F-page includes error correcting code pages (E-pages). In the illustrated embodiment, each F-page includes four E-pages that are illustrated as four boxes, including E-page 144. Other embodiments may use F-pages or E-pages that are defined differently or each F-page may include greater or fewer than four E-pages.

The controller 130 can receive data and/or storage access commands from a storage interface module 112 (e.g., a device driver) in a host system 110. Storage access commands communicated by the storage interface 112 can include write and read commands issued by the host system 110. The commands can specify a logical block address in the storage system 120, and the controller 130 can execute the received commands in the non-volatile memory array 140. In a hybrid hard drive, data may be stored in magnetic media storage component (not shown in Fig. 1) in addition to the non-volatile memory array 140.

In one embodiment, the controller 130 includes an ECC module 158. In one embodiment, the ECC module 158 handles error correction of data read from the memory array 140. In one embodiment, it encodes data to be written to memory pages, such as E-pages, of the non-volatile memory array 140 and decodes the data when they are read out. The controller 130 in one embodiment also includes a data recovery module 162 which performs the reprogramming methods in accordance with one or more embodiments of the invention, as further described below.

Voltage Distribution Illustration

Figures 2A and 2B show the voltage distribution of memory cells to illustrate the factors that contribution to lower page corruption and a solution according to one embodiment of the invention. Figure 2A shows the voltage distribution of MLC memory cells in which only the lower pages have been programmed. Line 200 shows the voltage distribution at the point of initial programming. Line 202 shows the voltage distribution after some time has lapsed. In Figure 2A, those cells have undergone a testing process where they are heated to simulate a time lapse of approximately 12.6 months at 40 °C. It can be seen that the distribution has drifted to the left on the voltage scale, such that some cells are now in the error region 208. These cells, if read, would likely result in a different bit value than the initial programmed value, because their voltage level is now below threshold 206. Thus these hundreds of bits are now in error. Line 204 shows the distribution after the cells have been reprogrammed in according to one or more embodiments of the invention. The distribution is now nearly identical to the distribution at the time of initial programming, and the failing bit count (FBC) is zero.

Figure 2B shows the voltage distribution of MLC memory cells in which both the lower and upper pages have been programmed. This graph does not show the effect of the voltage drifts. Rather, it illustrates two scenarios. First, line 210 shows the voltage distribution of the Vt states of the cells in which both the upper and lower pages are programmed at the same time or nearly the same time. This would be considered an ideal condition since there is little to no time lapsed, and little to no temperature differences between the two programming. However, this is not always how the cells are programmed. If the upper page is programmed at a different time and/or temperature than the lower page, the lower page corruption problem can occur. However, the reprogramming of the lower page reduces the risk of this problem. Line 212 shows the voltage distribution after the cells have been reprogrammed according to one or more embodiments of the invention. As shown, after reprogramming all three states are aligned very closely to the distribution shown in line 210, which, as mentioned above reflects the distribution of cells where upper and lower pages have been programmed at the same or nearly the same time.

Reprogramming

Figure 3 is a flow diagram showing a method 250 of reprogramming according to one embodiment of the invention. The method 250 may be executed as part of a start-up sequence, on a periodic basis, or on an as-needed basis. In one embodiment, the method 250 is performed by the controller 130 shown in Figure 1. At block 255, the method selects programmed lower page(s) with paired upper pages that are not programmed. These lower page(s) may be referred to as "open" lower pages. For example, the method may select such pages from one or more "open" blocks in which data is currently being programmed. To illustrate further, if the method 250 is executed as part of a start-up sequence, such open blocks may be blocks that were being programmed when the storage system was last shut down, and they have not been closed since their capacity have not been filled. Since the system is likely to resume programming in these open blocks, the lower page corruption problem may occur in these open lower page(s). Therefore, in one embodiment, the method selects such open lower pages for reprogramming.

At block 260, the data from the selected lower pages are read, and then the read data is corrected (e.g., through the application of Error Correction Code (ECC)) at block 265. Then, at block 270, the corrected data is reprogrammed back to the selected lower pages. As previously shown in Figure 2A, the voltage distribution of the cells in these lower pages now closely resembles the distribution when they were initially programmed. Thus, when their paired upper pages are programmed, the lower page corruption problem can be avoided.

Selecting Lower Pages

Figures 4A-4B and 5 show the different memory cell configuration in which open lower pages may be selected. In Figure 4A, an MLC configuration is presented and pages 0-7 are shown. The "U" denotes an upper page and the "L" denotes a lower page. From top to bottom, the horizontal lines represent word lines, and they are labeled (between Figures 4A and 4B) WL (Word Line) 0, WL 1, WL 2, and so on.

The page number denotes the order in which pages are programmed. Here, pages 0-7 have been programmed. Page 0, a lower page in WL 0 is first programmed, and then page 1, a lower page in the same WL 0 is programmed, and so on. Note that page 0 is paired with upper page 4, and page 1 is paired with upper page 5, and so on. In this programming sequence, pages 2, 3, 6, and 7 are open lower pages that do not have their paired upper pages programmed. In one embodiment, if the controller finds the state of the memory as shown in Figure 4A, these pages will be selected for reprogramming, for example, according to the method shown in Figure 2. In this typical configuration, there is a maximum of 4 such open pages at a given time. Thus, to reprogram these open pages is much more efficient than reprogram data from the entire memory block. In one embodiment, these open pages may be flagged (e.g., through metadata) by the controller as part of a shutdown sequence, so that any such open lower pages in open blocks can be quickly identified at start-up. If the indication for such open lower pages is not available (e.g., a previous ungraceful/unexpected shutdown has occurred or the system does not support marking such pages), the controller may perform scanning through the memory blocks to locate such open lower pages.

Figure 4B shows the same memory cells after the programming of pages 8 and 9. In the current example, just before pages 8 and 9 are programmed, a reprogram has taken place to read out data from pages 2 and 3, the corrected data is reprogrammed back to pages 2 and 3. Thus, lower page corruption of pages 2 and 3 can be avoided when pages 8 and 9 are programmed. With pages 8 and 9 programmed, pages 2 and 3 are no longer open. If the system shut downs at this point, at the next start-up the controller may select pages 6 and 7 for reprogramming.

Figure 5 shows another MLC configuration in which each cell is configured to encode 3 bits. This configuration is commonly referred to as TLC (Three-Level Cell) memory. The same notations apply here as with Figures 4A-4B. The programmed pages are labeled pages 0-11 with U and L designations and the word lines are labeled accordingly. Here, each lower page is paired with two upper pages. For example, page 0 is paired with upper pages 4 and 10. Pages 2, 3, 6, and 7 are open lower pages. Accordingly to one embodiment, if the controller finds the memory in this illustrated state, these pages would be selected for reprogramming. Note that pages 2 and 3 are considered open pages even though they are already paired with programmed upper pages 8 and 9. This is because each of pages 2 and 3 has one more upper page to be programmed. Again, reprogramming these few open pages is more efficient than reprogramming the entire block of memory.

Lower Page Only Reprogramming

Figure 6 is a flow diagram illustrating a process of reprogramming lower page programmed only cells according to one embodiment of the invention. Another situation the lower page recovery program may be useful is where only lower pages are programmed for data storage. Typically these pages are in memory blocks that have been designated to operate in a lower page only or SLC mode. Alternatively, the same scenario can occur in SLC memory. Because the upper pages are not programmed, lower page corruption problem does not occur. However, there are instances where reprogramming may nonetheless be useful to preserve data integrity. For example, in one embodiment, when the storage system finds that the data are close to some predefined criteria, recovery program may be then applied to bring the programmed voltage level back up to where it is supposed to be.

In Figure 6, the method 600 may be performed by the controller 130. The method starts at block 610 where one or more data integrity conditions are checked. Error rate such as bit failure rate and/or error correction effort applied (e.g., in the LDPC decoding and/or RAID recovery) from recent reads may be indications of the data integrity condition. The rates/conditions may be obtained from reads that are performed as part of a scanning process. In addition, other useable indications of integrity condition may include program erase cycle counter values and time lapsed approximated by voltage reference drift measured in reference pages/blocks. These conditions may be compared against a threshold metric at block 315, and when certain pre-defined conditions for triggering recovery are met, the method goes to block 320 to read data from these pages where only lower page are programmed. The read data is corrected (e.g., through the application of ECC) at block 325. Then at block 330 the corrected data is reprogrammed back to the pages. As previously shown in Figure 2A, the voltage distribution of the cells in these pages now closely resembles the distribution when they were initially programmed.

Other Variations

Those skilled in the art will appreciate that in some embodiments, other approaches and methods can be used. For example, if multi-pass programming is allowed for upper pages by finite-state machine on NAND flash chips, it is also possible to apply the methods in the various embodiments for upper page data recovery. For example, some upper pages may be causing an amount of errors that is near the correction limit of the ECC and may benefit from reprogramming to move the voltage levels to near the original programmed levels. In addition, the non-volatile memory array 140 can be implemented using memory devices other than NAND flash memory devices. Other types of solid-state memory devices can alternatively be used, such as array of flash integrated circuits, Chalcogenide RAM (C-RAM), Phase Change Memory (PC-RAM or PRAM), Programmable Metallization Cell RAM (PMC-RAM or PMCm), Ovonic Unified Memory (OUM), Resistance RAM (RRAM), NOR memory, EEPROM, Ferroelectric Memory (FeRAM), Magnetoresistive RAM (MRAM), other discrete NVM (non-volatile memory) chips, or any combination thereof. In one embodiment, the non-volatile memory array 140 preferably includes multi-level cell (MLC) devices having multi-level cells capable of storing more than a single bit of information, although single-level cell (SLC) memory devices or a combination of SLC and MLC devices may be used. In one embodiment, the storage system 120 can include other memory modules, such as one or more magnetic memory modules. The storage system 120 can further include other types of storage media, such as magnetic storage. Accordingly, the scope of the present disclosure is intended to be defined only by reference to the appended claims.

While certain embodiments have been described, these embodiments have been presented by way of example only, and are not intended to limit the scope of the protection. Indeed, the novel methods and systems described herein may be embodied in a variety of other forms. Furthermore, various omissions, substitutions and changes in the form of the methods and systems described herein may be made without departing from the spirit of the protection. The accompanying claims and their equivalents are intended to cover such forms or modifications as would fall within the scope and spirit of the protection. For example, the systems and methods disclosed herein can be applied to hybrid hard drives and the like. In addition, other forms of storage (e.g., DRAM or SRAM, battery backed-up volatile DRAM or SRAM devices, EPROM, EEPROM memory, etc.) may additionally or alternatively be used. As another example, the various components illustrated in the figures may be implemented as software and/or firmware on a processor, ASIC/FPGA, or dedicated hardware. Also, the features and attributes of the specific embodiments disclosed above may be combined in different ways to form additional embodiments, all of which fall within the scope of the present disclosure. Although the present disclosure provides certain preferred embodiments and applications, other embodiments that are apparent to those of ordinary skill in the art, including embodiments which do not provide all of the features and advantages set forth herein, are also within the scope of this disclosure. Accordingly, the scope of the present disclosure is intended to be defined only by reference to the appended claims.

WHAT IS CLAIMED IS:

1. A solid-state storage system, including:

a non-volatile memory array including a plurality of memory blocks, each memory block including lower memory pages paired with upper memory pages; and

a controller configured to:

select, from the memory blocks, a programmed lower memory page that has a paired upper memory page that has not been programmed;

read data from the selected lower memory page;

apply error correction to the read data to generated corrected data; and

program the corrected data to the selected lower memory page.

2. The solid-state storage system of claim 1, wherein the non-volatile memory array comprises multi-level memory (MLC) cells.

3. The solid-state storage system of claim 1, wherein the controller is configured to select the programmed lower memory page from an open memory block of the memory blocks.

4. A solid-state storage system, including:

a non-volatile memory array including a plurality of memory blocks, each including a plurality of memory pages; and

a controller configured to:

determine whether a state of the storage system meets a pre-defined condition to initiate data recovery;

in response to determining that the state of the storage system meets a pre-defined condition to initiate data recovery, select, from the memory blocks, a memory page from a memory block;

read data from the selected memory page;

apply error correction to the read data to generated corrected data; and

program the corrected data to the selected memory page.

5. The solid-state storage system of claim 4, wherein the state of the storage system includes a state of a memory page in a reference memory block.

6. The solid-state storage system of claim 5, wherein the state of the memory page in the reference memory block comprises a failure bit rate.

7. The solid-state storage system of claim 6, wherein failure bit rate is used to approximate time lapsed since last programmed.

8. The solid-state storage system of claim 6, wherein the state of the storage system includes a failure bit rate that is obtained from a scanning process.

9. The solid-state storage system of claim 4, wherein the non-volatile memory array comprises multi-level memory (MLC) cells and the memory page is in a memory block that is lower-page only programmed.

10. The solid-state storage system of claim 4, wherein the non-volatile memory array comprises single-level memory (SLC) cells.

11. The solid-state storage system of claim 4, wherein the controller is configured to determine whether the state of the storage system meets the pre-defined condition to initiate data recovery as part of a startup sequence.

12. A method of preserving data integrity in a solid-state storage system that includes a non-volatile memory array including a plurality of memory blocks, each including a plurality of memory pages, the method including:

selecting, from the memory blocks, a programmed lower memory page that has a paired upper memory page that has not been programmed;

reading data from the selected lower memory page;

applying error correction to the read data to generated corrected data; and

programming the corrected data to the selected lower memory page.

13. The method of claim 12, wherein the non-volatile memory array comprises multi-level memory (MLC) cells.

14. The method of claim 12, wherein selecting comprises selecting the programmed lower memory page from an open memory block of the memory blocks.

15. A method of preserving data integrity in a solid-state storage system that includes a non-volatile memory array including a plurality of memory blocks, each including a plurality of memory pages, the method including:

determining whether a state of the storage system meets a pre-defined condition to initiate data recovery;

in response to determining that the state of the storage system meets a pre-defined condition to initiate data recovery, selecting, from the memory blocks, a memory page from a memory block;

reading data from the selected memory page;

applying error correction to the read data to generated corrected data; and

programming the corrected data to the selected memory page.

16. The method of claim 15, wherein the state of the storage system includes a state of a memory page in a reference memory block.

17. The method of claim 16, wherein the state of the memory page in the reference memory block comprises a failure bit rate.

18. The method of claim 17, wherein failure bit rate is used to approximate time lapsed since last programmed.

19. The method of claim 17, wherein the state of the storage system includes a failure bit rate that is obtained from a scanning process.

20. The method of claim 15, wherein the non-volatile memory array comprises multi-level memory (MLC) cells and the memory page is in a memory block that is lower-page only programmed.

21. The method of claim 15, wherein the non-volatile memory array comprises single-level memory (SLC) cells.

22. The method of claim 15, wherein determining whether the state of the storage system meets the pre-defined condition to initiate data recovery is performed as part of a startup sequence.

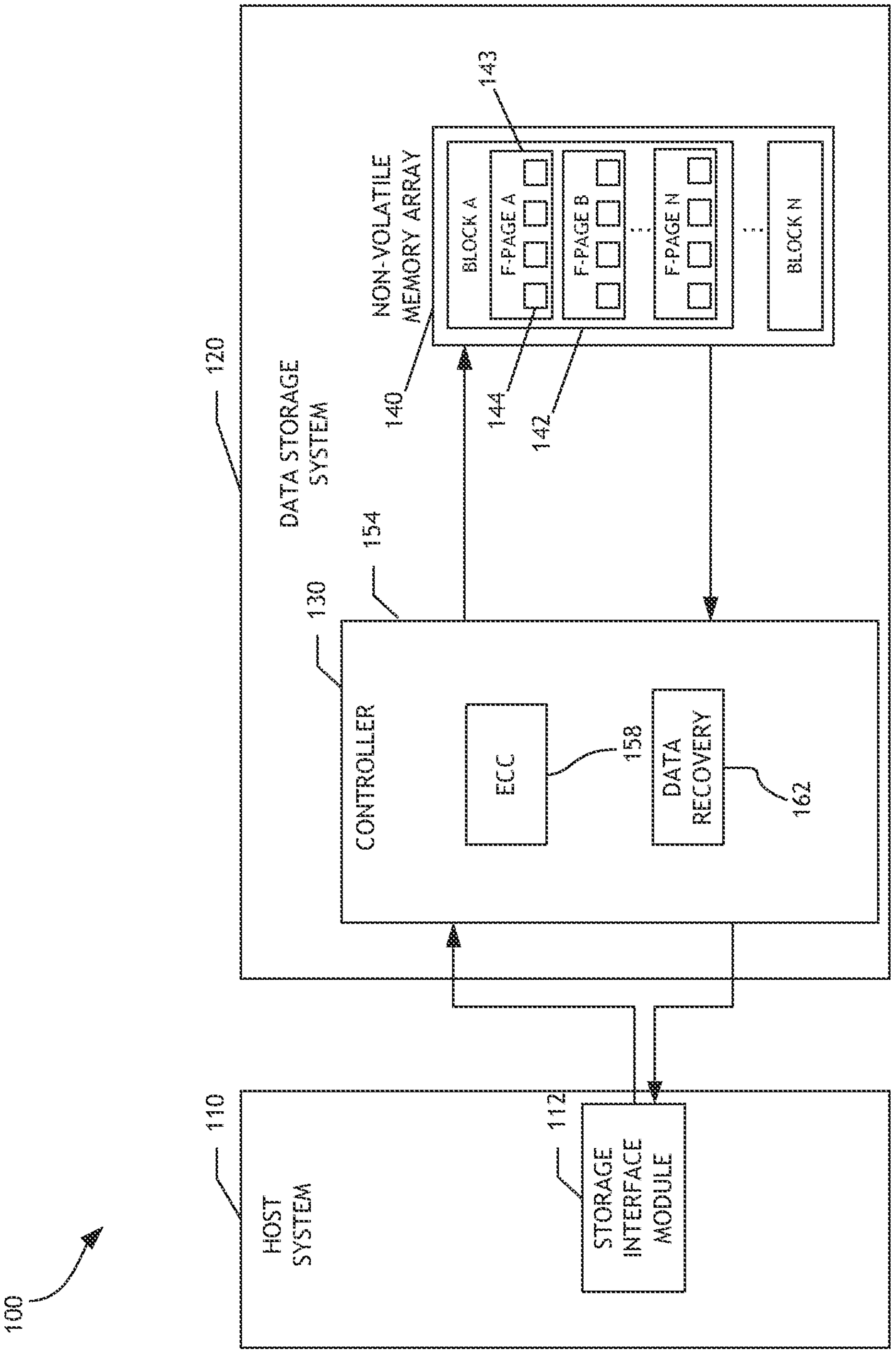

100
HOST SYSTEM
110
STORAGE INTERFACE MODULE
112
DATA STORAGE SYSTEM
120
CONTROLLER
130
154
ECC
158
DATA RECOVERY
162
NON-VOLATILE MEMORY ARRAY
140
BLOCK A
F-PAGE A
143
144
F-PAGE B
142
F-PAGE N
BLOCK N

FIGURE 1

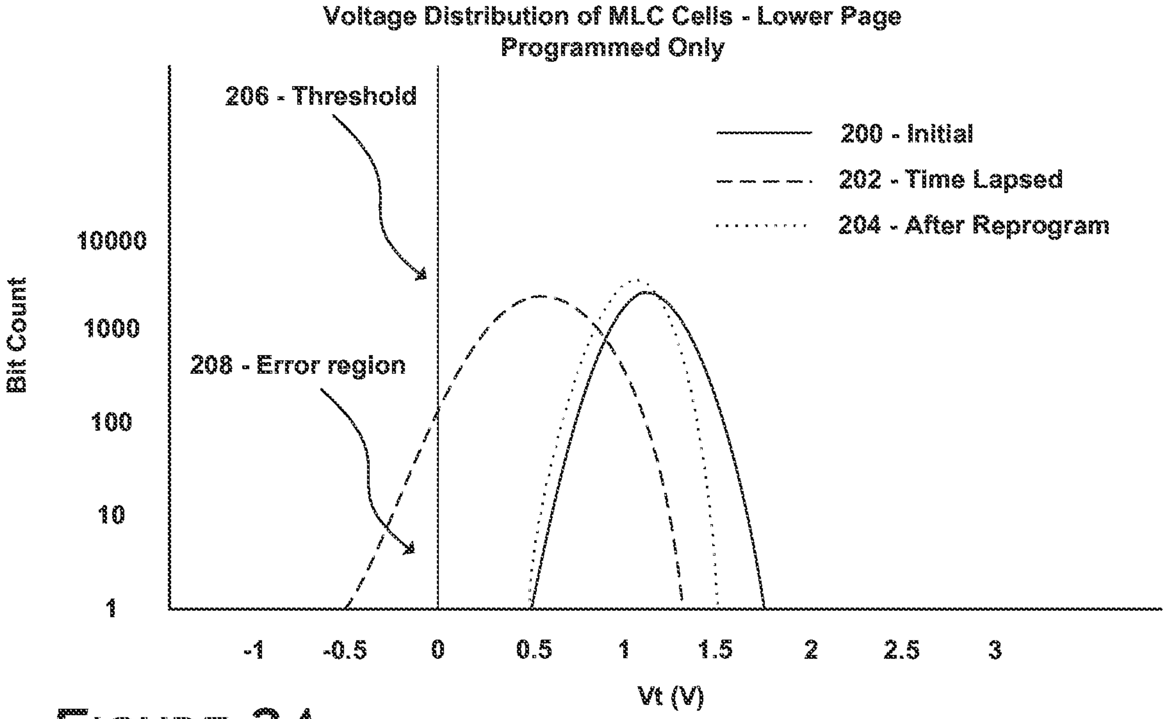


FIGURE 2A

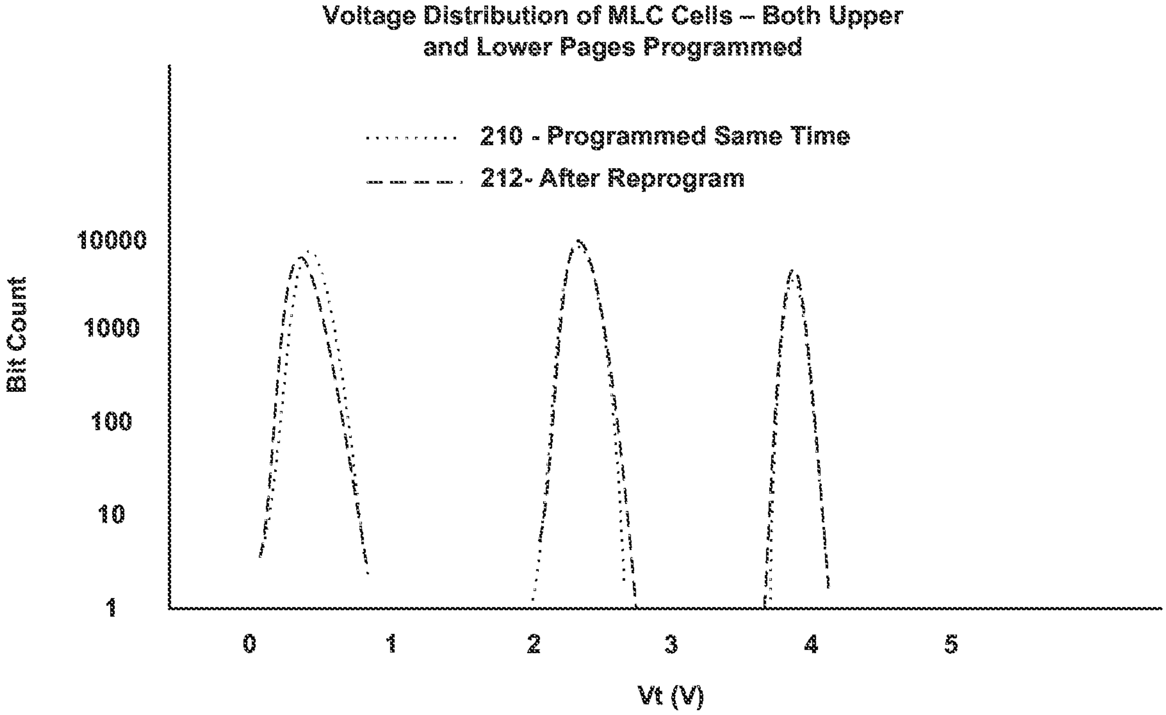


FIGURE 2B

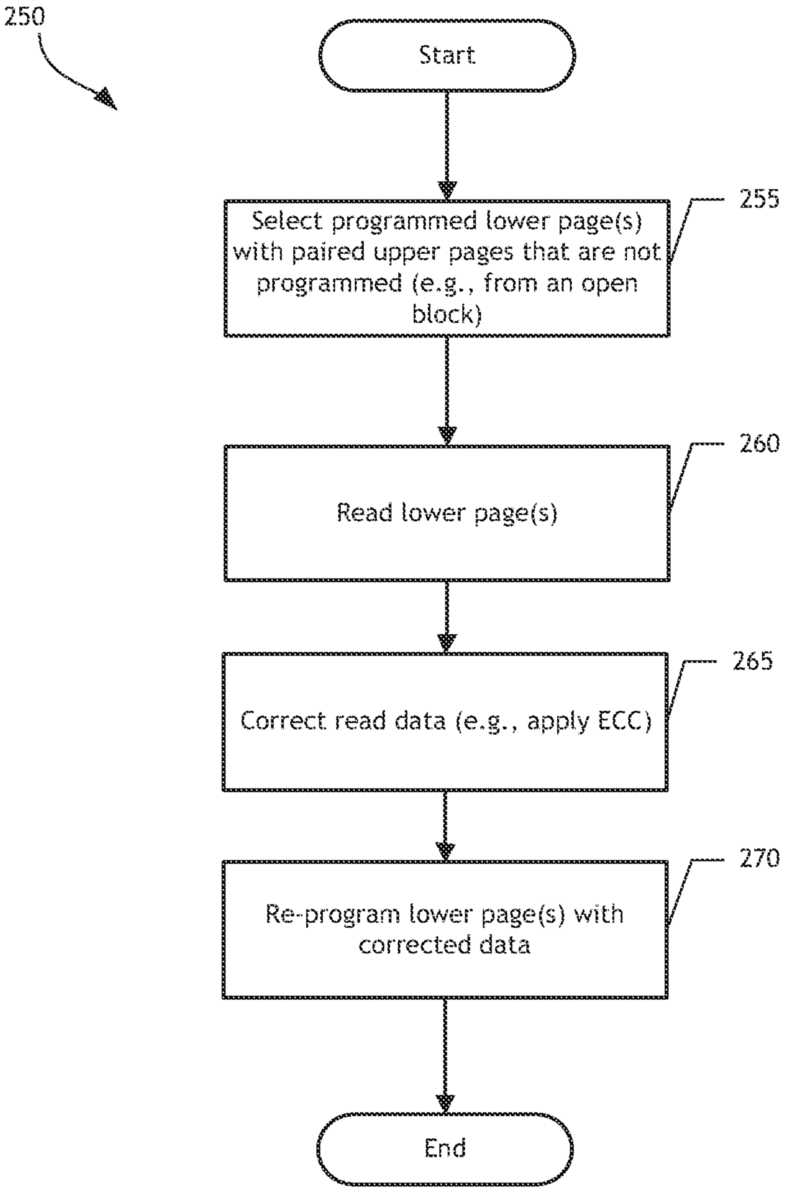

250
Start
Select programmed lower page(s) with paired upper pages that are not programmed (e.g., from an open block)
255
Read lower page(s)
260
Correct read data (e.g., apply ECC)
265
Re-program lower page(s) with corrected data
270
End

FIGURE 3

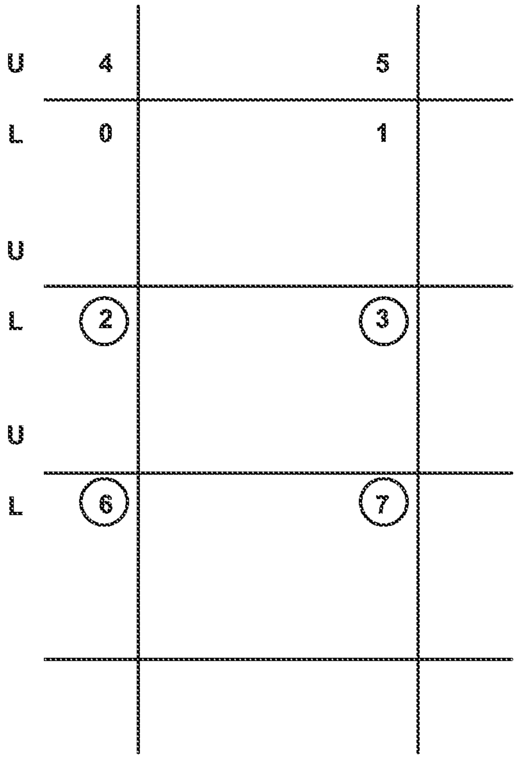


FIGURE 4A

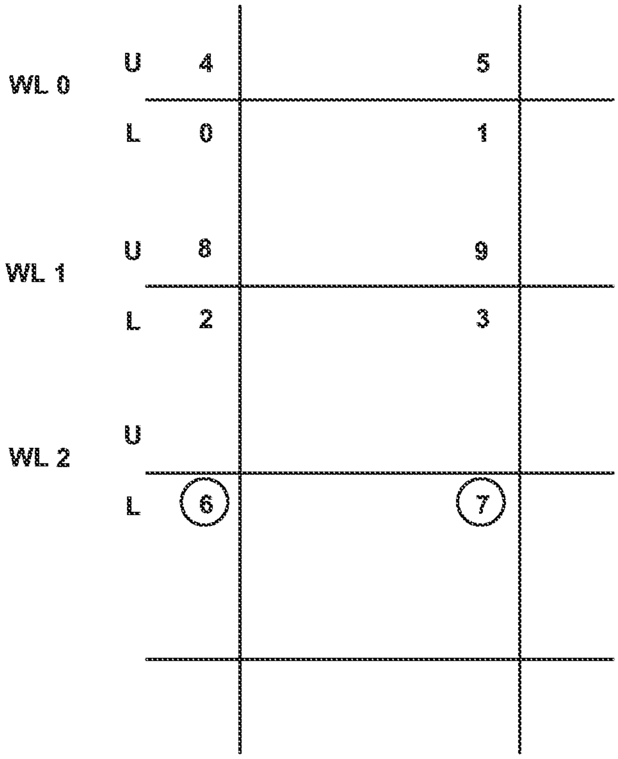


FIGURE 4B

○ = open lower page

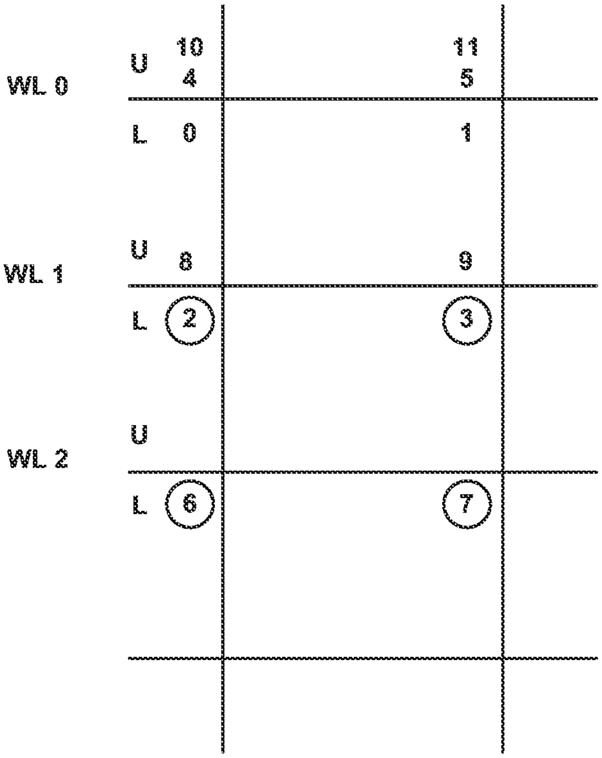


FIGURE 5

○ = open lower page

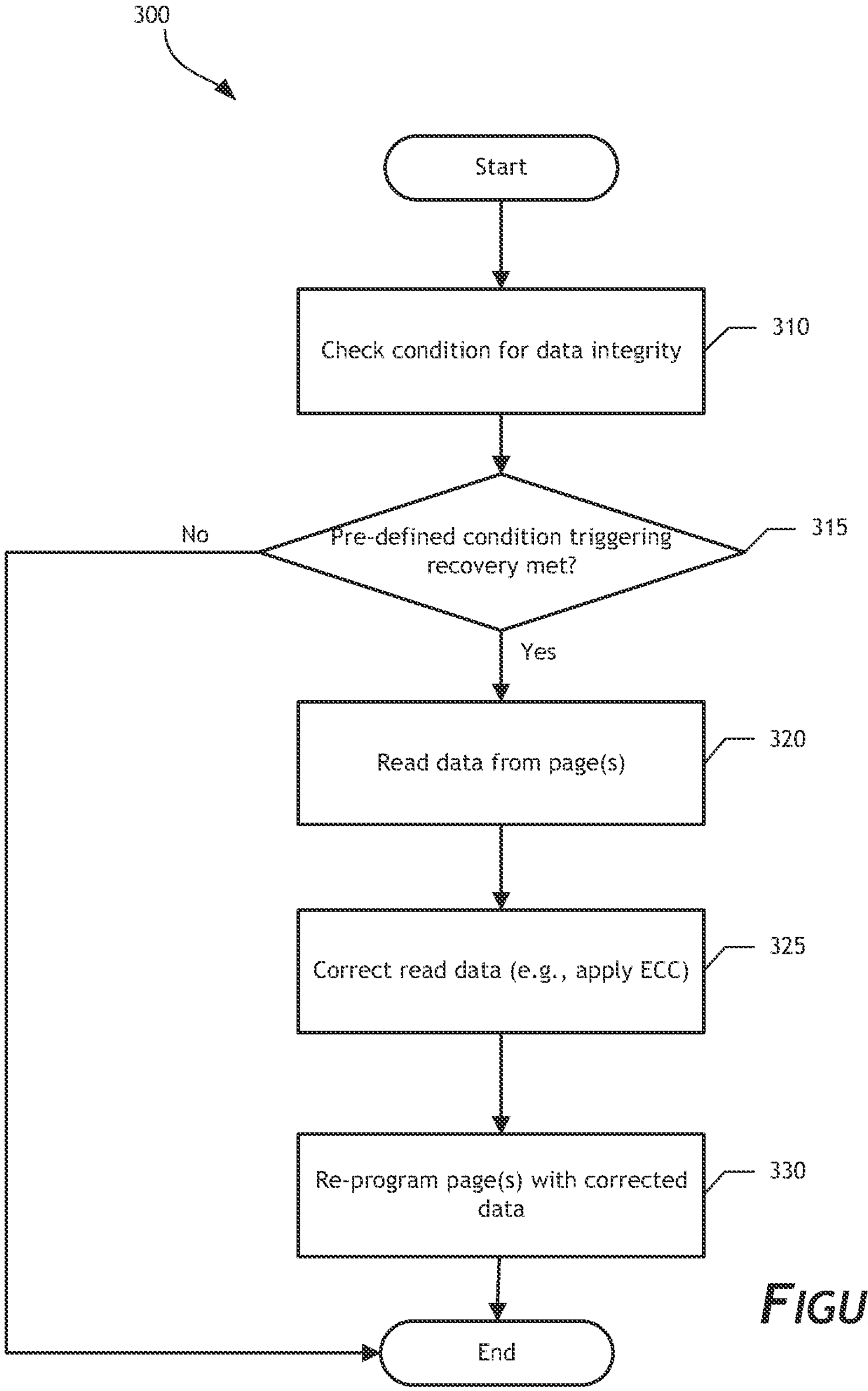

300
Start
Check condition for data integrity
310
Pre-defined condition triggering recovery met?
315
No
Yes
Read data from page(s)
320
Correct read data (e.g., apply ECC)
325
Re-program page(s) with corrected data
330
End

FIGURE 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2013/061608

A. CLASSIFICATION OF SUBJECT MATTER

G06F 12/16(2006.01)i, G06F 11/08(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)
G06F 12/16; H03M 13/05; G11C 16/04; G06F 11/10; G06F 11/14; G06F 12/00; G06F 12/08; G06F 11/08

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
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: lower page, upper page, non-volatile memory, data recovery, error correction, and similar terms.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2012-0054582 A1 (BYOM, MATTHEW et al.) 01 March 2012 See paragraphs [0032]-[0047] and [0061]-[0074]; claim 9; and figures 2 and 6.	1-22
A	US 2010-0318839 A1 (AVILA, CHRIS NGA YEE et al.) 16 December 2010 See paragraphs [0033]-[0037] and [0049]; claims 1 and 12; and figures 3 and 8.	1-22
A	US 2011-0202710 A1 (ZHAO, QUN et al.) 18 August 2011 See paragraphs [0018]-[0028]; claim 1; and figures 1-3.	1-22
A	US 2008-0177934 A1 (YU, JAE-SUNG et al.) 24 July 2008 See paragraphs [0058]-[0081] and [0104]-[0120]; claim 15; and figures 3-4 and 7.	1-22
A	US 2010-0246260 A1 (KANG, DONGKU) 30 September 2010 See paragraphs [0089]-[0094] and [0113]-[0122]; claim 14; and figures 5 and 10.	1-22
A	US 2010-0157675 A1 (SHALVI, OFIR et al.) 24 June 2010 See paragraphs [0058]-[0083]; claim 17; and figures 1-2.	1-22

□ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search 30 December 2013 (30.12.2013)	Date of mailing of the international search report **30 December 2013 (30.12.2013)**
Name and mailing address of the ISA/KR Korean Intellectual Property Office 189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701, Republic of Korea Facsimile No. +82-42-472-7140	Authorized officer NHO, Ji Myong Telephone No. +82-42-481-8528

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012-0054582 A1	01/03/2012	US 8341500 B2	25/12/2012
US 2010-0318839 A1	16/12/2010	US 8307241 B2	06/11/2012
US 2011-0202710 A1	18/08/2011	CN 102792381 A	21/11/2012
		US 8549214 B2	01/10/2013
		WO 2011-103168 A1	25/08/2011
US 2008-0177934 A1	24/07/2008	KR 10-0850515 B1	05/08/2008
		KR 10-2008-0069822 A	29/07/2008
		US 7755950 B2	13/07/2010
US 2010-0246260 A1	30/09/2010	KR 10-2010-0107294 A	05/10/2010
		US 2013-0088923 A1	11/04/2013
		US 8331145 B2	11/12/2012
		US 8537612 B2	17/09/2013
US 2010-0157675 A1	24/06/2010	US 2012-163079 A1	28/06/2012
		US 8174905 B2	08/05/2012
		US 8437185 B2	07/05/2013
		WO 2009-037691 A2	26/03/2009
		WO 2009-037691 A3	04/03/2010

(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 104919434 A
(43)申请公布日 2015.09.16

(21)申请号 201380064141.3

(22)申请日 2013.09.25

(30)优先权数据
13/708,873 2012.12.07 US

(85)PCT国际申请进入国家阶段日
2015.06.08

(86)PCT国际申请的申请数据
PCT/US2013/061608 2013.09.25

(87)PCT国际申请的公布数据
WO2014/088682 EN 2014.06.12

(71)申请人 西部数据技术公司
地址 美国加利福尼亚

(72)发明人 Y·孙 D·赵 J-y·杨

(74)专利代理机构 永新专利商标代理有限公司 72002

代理人 刘瑜 王英

(51)Int.Cl.
G06F 12/16(2006.01)
G06F 11/08(2006.01)

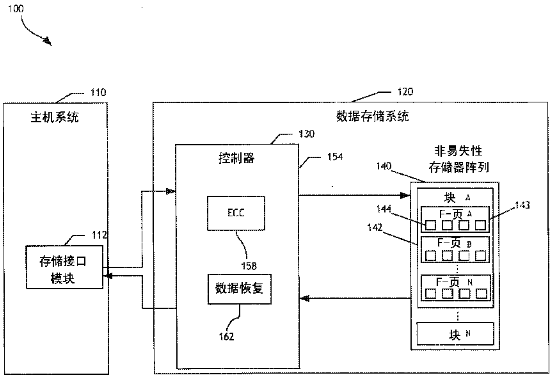
权利要求书2页 说明书5页 附图6页

(54)发明名称

用于在固态驱动器中进行较低页数据恢复的系统和方法

(57)摘要

在本发明的某些实施例中，数据存储系统包括控制器和具有多个存储页的非易失性存储器阵列。该控制器执行有效地解决较低页讹误问题的方法。在一个实施例中，该方法选择所配对的较高页未被编程的已编程的较低页，从那些所选择的较低页中读取数据，校正所读取的数据，并且将所读取的数据重编程到那些较低页。由于这种条件下的较低页的数量典型地低（例如，在具有成百上千页的块中的若干页），所以这是比对整个块进行重编程更有效的方法，在另一个实施例中，类似的重编程方法在仅较低页已编程的情况下（例如，SLC存储器、SLC模式下的MLC存储器等等）被应用作为一种数据恢复方案。



1.一种固态存储系统，包括：

包括多个存储块的非易失性存储器阵列，每一个存储块包括与较高存储页配对的较低存储页；以及

控制器，其被配置为：

从所述多个存储块中选择具有未被编程的配对的较高存储页的已编程的较低存储页；

从所选择的较低存储页中读取数据；

将误差校正应用到所读取的数据以生成校正的数据；以及

将所校正的数据编程到所选择的较低存储页。

2.如权利要求1所述的固态存储系统，其中，所述非易失性存储器阵列包括多级存储(MLC)单元。

3.如权利要求1所述的固态存储系统，其中，所述控制器被配置为从所述多个存储块的开放存储块中选择所述已编程的较低存储页。

4.一种固态存储系统，包括：

包括多个存储块的非易失性存储器阵列，所述多个存储块的每一个包括多个存储页；以及

控制器，其被配置为：

确定存储系统的状态是否满足发起数据恢复的预定义的条件；

响应于确定所述存储系统的状态满足发起数据恢复的预定义的条件，从所述多个存储块中选择来自一个存储块的存储页；

从所选择的存储页中读取数据；

将误差校正应用到所读取的数据以生成校正的数据；以及

将所校正的数据编程到所选择的存储页。

5.如权利要求4所述的固态存储系统，其中，所述存储系统的状态包括在参考存储块中的存储页的状态。

6.如权利要求5所述的固态存储系统，其中，在所述参考存储块中的存储页的状态包括失败比特率。

7.如权利要求6所述的固态存储系统，其中，失败比特率被用于估计自从上一次被编程后所流逝的时间。

8.如权利要求6所述的固态存储系统，其中，所述存储系统的状态包括从扫描过程中获取的失败比特率。

9.如权利要求4所述的固态存储系统，其中，所述非易失性存储器阵列包括多级存储(MLC)单元，并且所述存储页位于仅较低页被编程的存储块中。

10.如权利要求4所述的固态存储系统，其中，所述非易失性存储器阵列包括单级存储(SLC)单元。

11.如权利要求4所述的固态存储系统，其中，所述控制器被配置为作为启动序列的部分来确定所述存储系统的状态是否满足发起数据恢复的预定义的条件。

12.一种在包括非易失性存储器阵列的固态存储系统中保持数据完整性的方法，所述非易失性存储器阵列包括多个存储块，所述多个存储块的每一个包括多个存储页，所述方

法包括：

从所述多个存储块中选择具有未被编程的配对的较高存储页的已编程的较低存储页；

从所选择的较低存储页中读取数据；

将误差校正应用到所读取的数据以生成校正的数据；以及

将所校正的数据编程到所选择的较低存储页。

13.如权利要求12所述的方法，其中，所述非易失性存储器阵列包括多级存储(MLC)单元。

14.如权利要求12所述的方法，其中，选择包括从所述多个存储块的开放存储块中选择已编程的较低存储页。

15.一种在包括非易失性存储器阵列的固态存储系统中保持数据完整性的方法，所述非易失性存储器阵列包括多个存储块，所述多个存储块的每一个包括多个存储页，所述方法包括：

确定存储系统的状态是否满足发起数据恢复的预定义的条件；

响应于确定所述存储系统的状态满足发起数据恢复的预定义的条件，从所述多个存储块中选择来自一个存储块的存储页；

从所选择的存储页中读取数据；

将误差校正应用到所读取的数据以生成校正的数据；以及

将所校正的数据编程到所选择的存储页。

16.如权利要求15所述的方法，其中，所述存储系统的状态包括参考存储块中的存储页的状态。

17.如权利要求16所述的方法，其中，在所述参考存储块中的存储页的状态包括失败比特率。

18.如权利要求17所述的方法，其中，失败比特率被用于估计自从上一次被编程后所流逝的时间。

19.如权利要求17所述的方法，其中，所述存储系统的状态包括从扫描过程中获取的失败比特率。

20.如权利要求15所述的方法，其中，所述非易失性存储器阵列包括多级存储(MLC)单元，并且所述存储页位于仅较低页被编程的存储块中。

21.如权利要求15所述的方法，其中，所述非易失性存储器阵列包括单级存储(SLC)单元。

22.如权利要求15所述的方法，其中，确定所述存储系统的状态是否满足发起数据恢复的预定义的条件被作为启动序列的部分而执行。

用于在固态驱动器中进行较低页数据恢复的系统和方法

技术领域

[0001] 本公开涉及数据存储系统，例如，用于计算机系统的固态驱动器。更具体地，该公开涉及较低页数据恢复。

背景技术

[0002] 具有多层单元(MLC)NAND媒体的存储器阵列现在在固态驱动器(SSD)中是常见的。MLC允许多个可能的状态存在于单个存储单元中，因此能够每单元存储多于1个比特的信息(例如，2、3、4或更多)。例如，在每单元2个比特的MLC闪存中，4个可能的状态(4伏特(Vt)水平)是可能的，其能够存储2个比特。基于数据的存储单元将对不同状态进行存储以及编码，该单元被编程为4个可能和明显不同的Vt区域。典型地，存储在较低页和较高页的数据被逻辑地配对在一起，较低页首先被编程。

附图说明

[0003] 体现本发明的各种特征的系统和方法现在将参考附图进行详细示出，其中：

[0004] 图1示出了根据本发明的一个实施例的对数据进行重编程的存储系统。

[0005] 图2A和2B示出了根据本发明的一个实施例的用于说明有助于较低页讹误和解决方法的因素的存储单元的电压分布。

[0006] 图3是示出了根据本发明一个实施例的重编程的方法的流程图。

[0007] 图4A-图4B和图5示出了根据本发明的实施例的在其中开放较低页可以被选择用于重编程的不同的存储单元配置。

[0008] 图6是示出了根据本发明的一个实施例的对仅较低页被编程的单元进行重编程的过程的流程图。

具体实施方式

[0009] 尽管示出了某些实施例，但是这些实施例仅通过示例进行呈现，并不是要限制保护的范围。事实上，在本文中描述的新方法和系统可以以各种其他形式体现。此外，可以对本文描述的系统和方法的形式做出各种省略、替代和改变而不偏离保护的范围。

[0010] 概述

[0011] 在MLC闪存中，尽管较低页和较高页被物理地配对，但是它们通常在编程中被解耦。存储在较低页和较高页的数据可以在不同的时间以及从不同的源被编程。经常会出现较高页被编程得更晚并且处于不同温度的情况。大部分时间，这种编程方案是没问题的。但是，当该存储系统尝试对较高页进行编程时，包括存储单元的温度和老化效应的各种因素会降低单元中数据的保持力并且可能会导致较低页讹误的问题。

[0012] 一种克服较低页讹误问题的方法是当需要对较高页进行编程时将整个开放块的旧数据移动到另一个新地址。以这种方式，在类似的情况下较低页和较高页被同时编程。但是，此方法是低效的。

[0013] 在本发明的某些实施例中，数据存储系统包括控制器和具有多个存储页的非易失性存储器阵列。尽管仍然阻止上述讨论的较低页讹误问题，但是该控制器执行可能需要较少量系统负载的方法。在一个实施例中，该方法选择配对的较高页未被编程的已编程的较低页，从那些所选择的较低页中读取数据，校正所读取的数据，并且将所读取的数据重编程到那些较低页。由于这种条件下的较低页的数量典型地很低（例如，在具有成百上千页的块中的若干页），这是比对整个块进行重编程更有效的方法。在另一个实施例中，类似的重编程方法被应用为在仅较低页被编程的情况下（例如，SLC（单级单元）存储器、SLC模式下的MLC存储器等等）的数据恢复方案。

[0014] 系统概述

[0015] 图1示出了根据本发明的一个实施例的针对数据恢复执行重编程的存储系统120。如所示，存储系统120（例如，固态驱动器、混合硬盘等）包括控制器130和包括被标识为块“A”142到块“N”的存储器存储的一个或多个存储块在内的非易失性存储器阵列140。每一个块包括闪存页（F-页）。例如，图1的块A 142包括被标识为F-页A 153、B到N的F-页。在某些实施例中，每一个“F-页”是能够在单个操作中进行编程或被编程为单元的非易失性存储器阵列140中存储单元的最小分组。此外，每一个F-页包括误差校正代码页（E-页）。在所示出的实施例中，每一个F-页包括被示出为四个框的四个E-页，其包括E-页144。其他实施例可以使用被不同地定义的F-页或E-页，或每一个F-页可以包括多于或少于四个E-页。

[0016] 控制器130可以从主机系统110中的存储接口模块112（例如，设备驱动器）中接收数据和/或存储访问命令。由存储接口112传送的存储访问命令可以包括由主机系统110发出的写和读命令。该命令可以指定存储系统120中的逻辑块地址，并且控制器130可以执行在非易失性存储器阵列140中所接收的命令。在混合硬盘中，数据可以存储在除非易失性存储器阵列140外的磁媒介存储部件中（图1中未示出）。

[0017] 在一个实施例中，控制器130包括ECC模块158。在一个实施例中，该ECC模块158处理从存储阵列140读取的数据进行误差校正。在一个实施例中，其对要被写入非易失性存储器阵列140的存储页面（例如，E-页）的数据进行编码，并且当数据被读出时对它们进行解码。在一个实施例中，控制器130还包括根据本发明的一个或多个实施例执行重编程方法的数据恢复模块162，如下被进一步所描述的。

[0018] 电压分布图

[0019] 图2A和图2B示出了根据本发明的一个实施例的用于示出有助于较低页讹误和解决方法的因素的存储单元的电压分布。图2A示出了其中仅较低页被编程的MLC存储单元的电压分布。线200示出了在初始编程点的电压分布。线202示出了一段时间流逝后的电压分布。在图2A中，那些单元已经经历了一个测试过程，在其中它们被加热以模拟40℃大约12.6个月的时间流逝。可以看出分布已经漂移到电压范围左侧，使得某些单元现在位于误差区域208中。这些单元，如果被读取，将可能导致与初始编程值不同的比特值，这是因为它们的电压值现在低于阈值206。因此这数百个比特现在是有误差的。线204示出了根据本发明的一个或多个实施例的在这些单元已经被重新编程后的分布。该分布现在几乎与初始编程时的分布相同，并且失效比特数(FBC)是零。

[0020] 图2B示出了较高页和较低页二者都已经被编程的MLC存储单元的电压分布。所

述图没有示出电压漂移的影响。而是示出了两个场景。首先，线 210 示出了较高页和较低页二者同时或几乎同时被编程的单元 Vt 状态的电压分布。这可能被认为是理想状态，因为只有几乎没有时间流逝，并且在两次编程之间几乎没有温度差别。但是，这不是通常单元被编程的方式。如果较高页在与较低页不同的时间和 / 或温度进行编程，则较低页讹误问题可能发生。但是，较低页的重新编程减少了这个问题的风险。线 212 示出了根据本发明的一个或多个实施例的单元被重编程后的电压分布。如所示，在对所有三个状态进行重新编程后，三个状态被排列为与在线 210 中示出的分布非常接近，如上所述，其反映了较高页和较低页二者同时或几乎同时被编程的单元的分布。

[0021] 重编程

[0022] 图 3 是示出了根据本发明的一个实施例的重编程方法 250 的流程图。方法 250 可以在定期的基础上或按需的基础上被执行作为启动序列的部分。在一个实施例中，方法 250 由图 1 中示出的控制器 130 执行。在块 255，该方法选择具有未被编程的配对的较高页的已编程的较低页。这些较低页可以被称为“开放”较低页。例如，该方法可以从在其中数据目前正被编程的一个或多个“开放”块中选择这样的页。为了进一步示出，如果方法 250 被执行作为启动序列的部分，则这样的开放块可以是当存储系统最后被关闭时正被编程的块，并且它们没有被关闭因为它们的容量未被填满。由于系统在这些开放块中很有可能重新开始编程，所以较低页讹误问题可能在这些开放较低页中发生。因此，在一个实施例中，该方法选择这样的开放较低页以用于重新编程。

[0023] 在块 260，来自所选较低页的数据被读取，并且接着在块 265 所读取的数据被校正 (例如，通过误差校正码 (ECC) 的应用)。然后，在块 270，被校正的数据被重新编程回所选的较低页。如之前在图 2A 中示出的，现在在这些较低页中的单元的电压分布非常类似于当它们初始被编程时的分布。因此，当它们的配对的较高页被编程时，较低页讹误问题可以被避免。

[0024] 选择较低页

[0025] 图 4A- 图 4B 和图 5 示出了在其中可以选择开放较低页的不同的存储单元配置。在图 4A 中，呈现了 MLC 配置并且示出了 0-7 页。“U”表示较高页并且“L”表示较低页。从顶到底，水平线表示字线，并且它们被标记 (在图 4A 和 4B 之间) 为 WL(字线)0、WL1、WL2 等等。

[0026] 页码表示页面被编程的顺序。在此，0-7 页已经被编程。页 0，WL0 中的较低页首先被编程，并且接着是页 1，在相同 WL0 中的较低页被编程等等。注意，页 0 与较高页 4 配对，而页 1 与较高页 5 配对等等。在这个编程序列里，页 2、3、6 和 7 是它们配对的较高页没有被编程的开放较低页。在一个实施例中，如果控制器发现存储器的状态如图 4A 中所示，则这些页面会被选择用于重新编程，例如，根据图 2 中所示的方法。在这个典型的配置中，在给定的时间里最多有 4 个这样的开放页面。因此，对这些开放页面进行重编程比对来自整个存储块的数据进行重编程更高效。在一个实施例中，这些开放页面可以由控制器标记 (例如，通过元数据) 作为关闭序列的部分，使得在开放块中的任何这样的开放较低页能在启动时被快速地识别。如果用于这样的开放较低页的指示是不可用的 (例如，之前非正常 / 意外关闭已发生或系统不支持标记这样的页)，则控制器可以通过存储块执行扫描以定位这样的开放较低页。

[0027] 图4B示出了在对页8和9进行编程之后相同的存储单元。在当前的示例中，只是在页8和9被编程之前，发生重编程以从页2和3中读出数据，校正的数据被重编程回页2和3。因此，当页8和9被编程时，页2和3的较低页讹误能够被避免。随着页8和9被编程，页2和3不再开放。如果在这一点系统关闭，则在下次启动时控制器可以选择页6和7以用于重编程。

[0028] 图5示出了在其中每一个单元被配置用于对3个比特进行编码的另一个MLC配置。这个配置通常被称为TLC（三级单元）存储器。在文中与图4A-4B一样应用相同的符号。已编程的页面用具有U和L的符号页0-11进行标记，并且字线被相应地标记。在此，每一个较低页与两个较高页配对。例如，页0与较高页4和10配对。页2、3、6和7是开放较低页。根据一个实施例，如果控制器发现存储器处于这个所示出的状态，则这些页面将被选择用于重编程。注意即使页2和3已经与编程的较高页8和9配对，页2和3也被认为是开放页面。这是因为页2和3中的每一个具有多于一个的要被编程的较高页。再次，对这些少量开放页进行重编程比对存储器的整个存储块进行重编程更高效。

[0029] 仅对较低页进行重编程

[0030] 图6是示出了根据本发明的一个实施例的对仅较低页被编程的单元进行重编程的过程的流程图。较低页恢复程序可能有用的另一个情况是仅仅较低页被编程用于数据存储。典型地这些页在已经被指定为仅在较低页或SLC模式下操作的存储块中。可替换地，同样的场景可以发生在SLC存储器中。由于较高页未被编程，所以较低页讹误问题不会发生。但是，尽管如此存在重编程对保持数据完整性有用的实例。例如，在一个实施例中，当存储系统发现数据接近于某些预定义标准时，恢复程序随后可能被应用以促使所编程的电压水平回到其被假定所到的水平。

[0031] 在图6中，方法600可以由控制器130执行。方法在块610处开始，其中检查一个或多个数据完整性条件。来自最近读取的误差率（例如，被应用的比特失败率和／或误差校正工作）（例如，在LDPC解码和／或RAID恢复中）可能是数据完整性条件的指示。该比率／条件可以从读取中获得被执行作为扫描过程的部分。此外，完整性条件的其他可用的指示可以包括程序擦除循环计数器值和由在参考页／块中测量的电压参考漂移进行估计(approximate)的流逝时间。在块315，这些条件可能与阈值度量做比较，并且当用于触发恢复的特定预定义的条件满足时，方法转到块320以从这些仅较低页被编程的页中读取数据。所读取的数据在块325被校正（例如，通过ECC的应用）。接着在块330，被校正的数据被重编程回这些页。如之前在图2A中所示，现在这些页中单元的电压分布非常类似于它们被初始编程时的分布。

[0032] 其他变型

[0033] 本领域的技术人员将意识到在某些实施例中，可以使用其他的方式和方法。例如，如果多遍编程在NAND闪存芯片上由有限状态机允许较高页，则将各种实施例中的方法应用于较高页数据恢复也是可能的。例如，某些较高页可能导致接近ECC的校正限制的大量的误差，并且可能从重编程中获益以将电压水平移动到接近原始编程水平。此外，非易失性存储器阵列140可以使用存储器设备而不是NAND闪存设备来实现。其他类型的固态存储器设备能够可替换地使用，所述其他类型的固态存储器设备例如，闪存集成电路阵列、硫属化物RAM(C-RAM)、相变存储器 (PC-RAM或PRAM)、可编程金属单元RAM(PMC-RAM或PMCm)、

双向统一存储器 (OUM)、电阻 RAM(RRAM)、NOR 存储器、EEPROM、铁电存储器 (FeRAM)、磁阻 RAM(MRAM)、其他分立的 NVM(非易失性存储器) 芯片、或其任意组合。在一个实施例中,非易失性存储器阵列 140 优选地包括具有能够存储多于单个比特信息的多级单元的多级单元 (MLC) 设备,但是可以使用单级单元 (SLC) 存储器设备或 SLC 和 MLC 设备的组合。在一个实施例中,存储系统 120 可以包括其他存储器模块,例如,一个或多个磁存储器模块。存储系统 120 可以进一步包括其他类型的存储介质,例如,磁存储装置。相应地,本公开的范围是要仅通过参考所附的权利要求来进行限定。

[0034]　尽管已经描述了特定实施例,但是这些实施例仅通过示例的方式进行呈现,并不是要限制保护的范围。事实上,本文描述的新方法和系统可以以各种其他形式体现。此外,可以对本文描述的方法和系统形式做出各种省略、代替和改变,而不偏离所保护的精神。所附的权利要求和它们的等同物是要覆盖这些形式或修改,其将都落在保护的范围和精神内。例如,本文所公开的系统和方法可以被应用于混合硬盘以及类似物。此外,可以额外或可替换地使用其他形式的存储装置 (例如,DRAM 或 SRAM、电池备份的易失性 DRAM 或 SRAM 设备、EPROM、EEPROM 存储器等等)。作为另一个示例,在附图中示出的各种部件可以被实现为处理器上的软件和 / 或固件、ASIC/FPGA、或专用硬件。同样,以上公开的具体实施例的特征和属性能够以不同的方式组合以形成额外的实施例,所有的这些都落在本公开的范围内。尽管本公开提供了某些优选的实施例和应用,但是其他的实施例对本领域的普通技术人员而言是显而易见的,包括不提供本文所述的所有特征和优点的实施例,也落在本公开的范围内。相应地,本公开的范围是要仅通过参考所附的权利要求来进行限定。

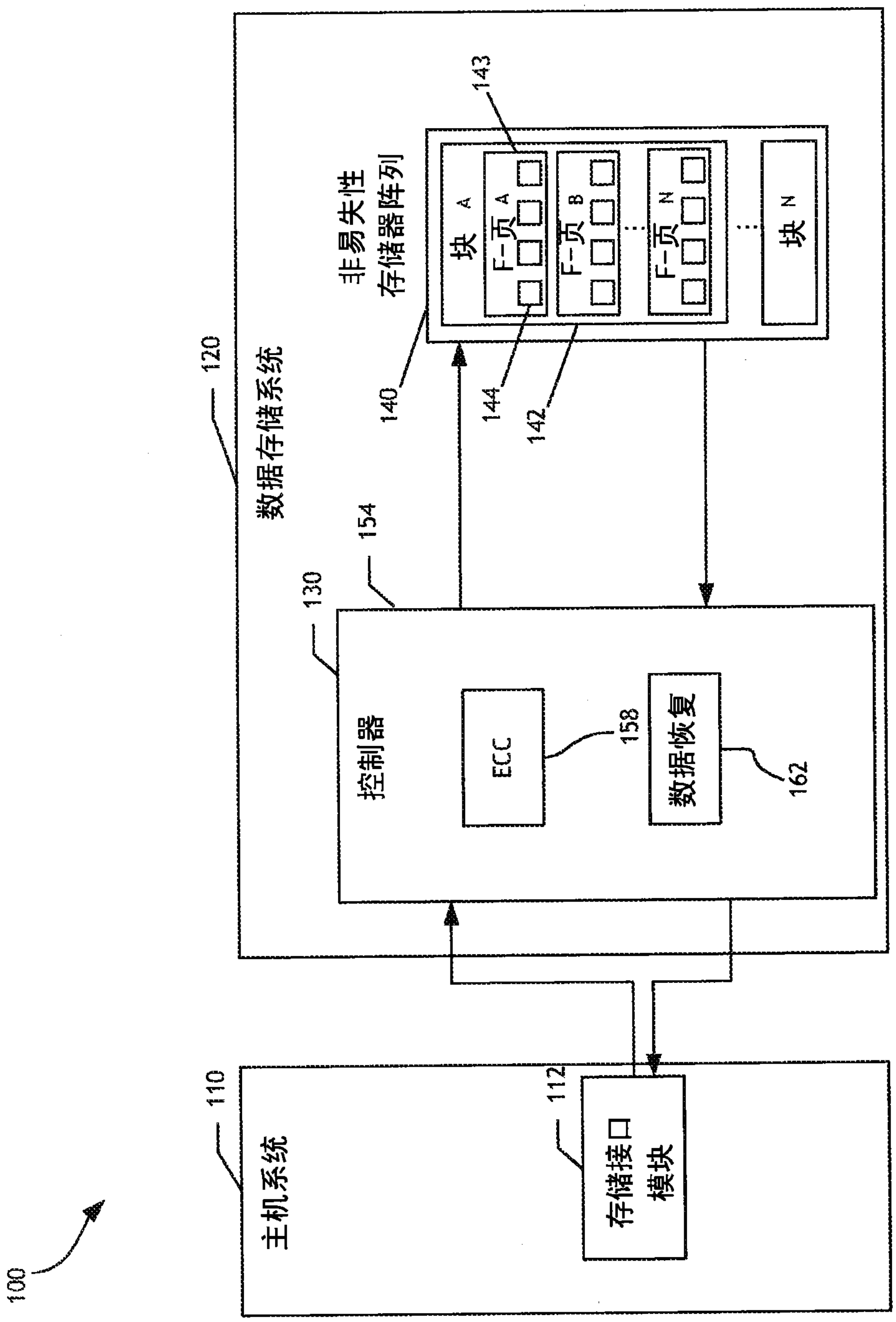

100
110
主机系统
112
存储接口模块
120
数据存储系统
130
控制器
154
ECC
158
数据恢复
162
140
非易失性存储器阵列
块 A
F-页 A
F-页 B
F-页 N
块 N
143
144
142

图 1

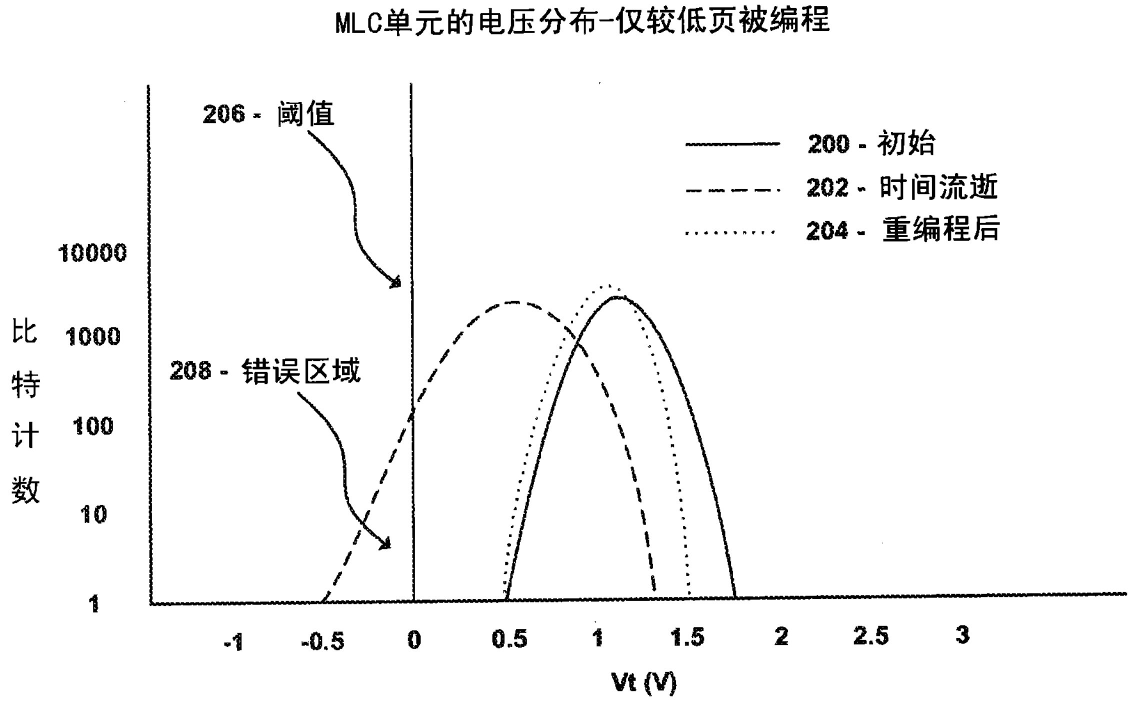

MLC单元的电压分布-仅较低页被编程
206 - 阈值
200 - 初始
202 - 时间流逝
204 - 重编程后
208 - 错误区域
比特计数
10000
1000
100
10
1
-1
-0.5
0
0.5
1
1.5
2
2.5
3
Vt (V)

图 2A

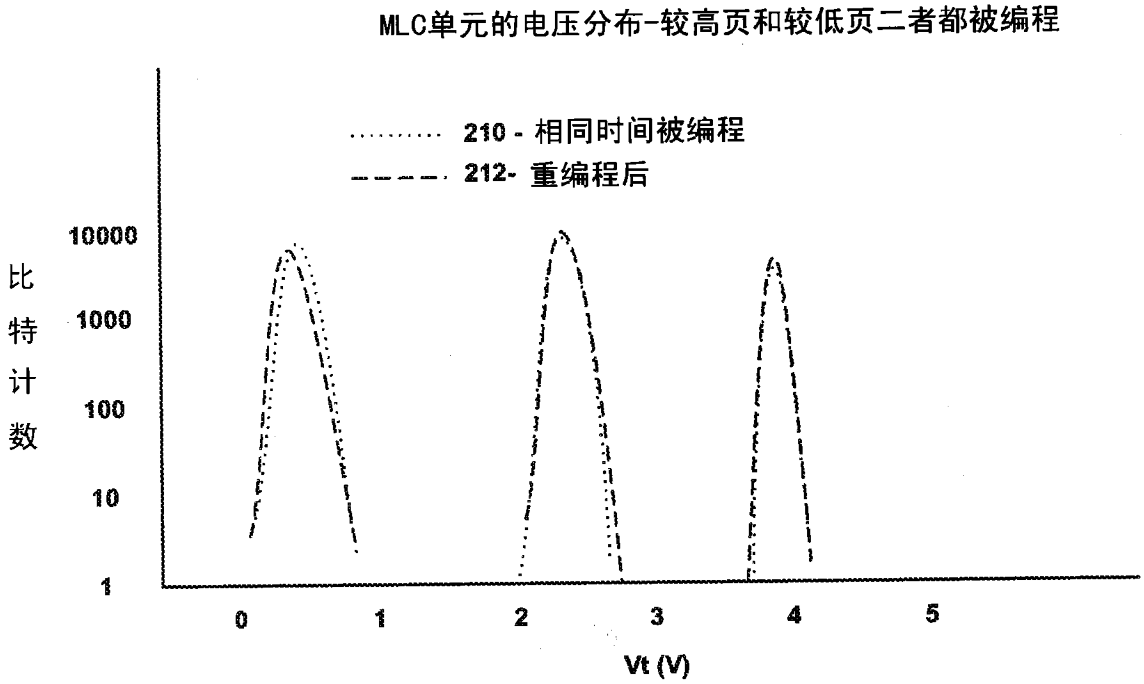

MLC单元的电压分布-较高页和较低页二者都被编程
210 - 相同时间被编程
212- 重编程后
比特计数
10000
1000
100
10
1
0
1
2
3
4
5
Vt (V)

图 2B

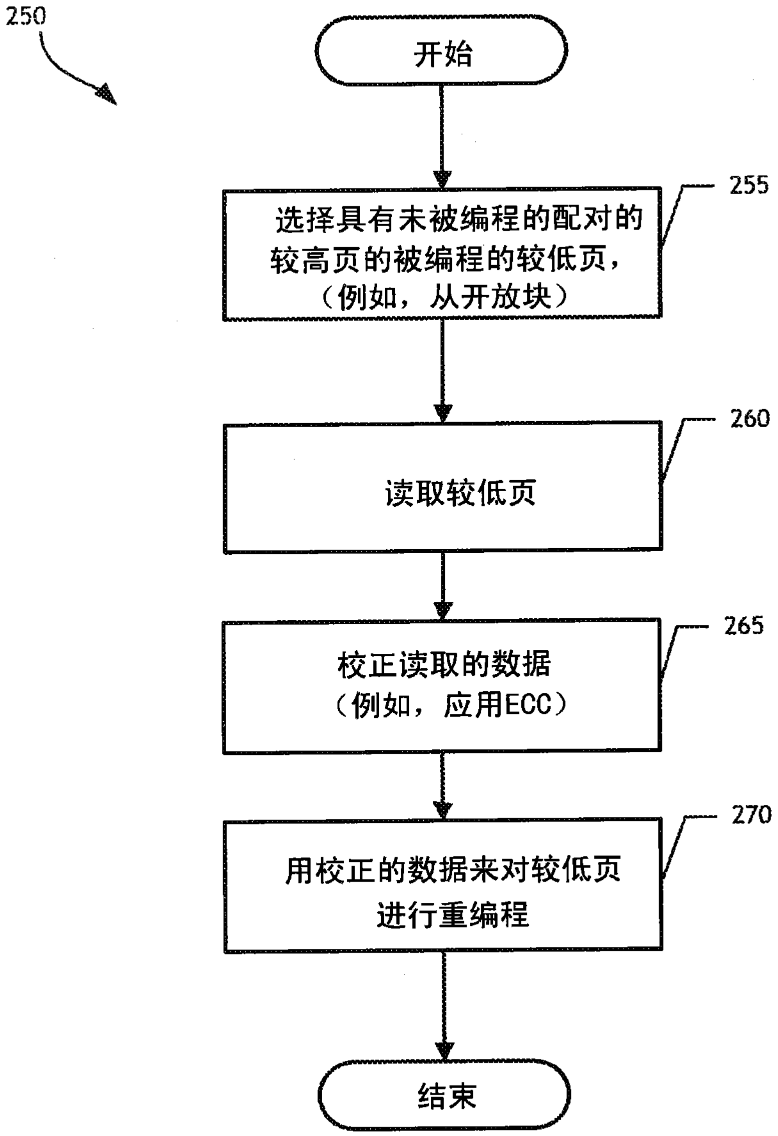

250
开始
选择具有未被编程的配对的较高页的被编程的较低页，（例如，从开放块）
255
读取较低页
260
校正读取的数据（例如，应用ECC）
265
用校正的数据来对较低页进行重编程
270
结束

图 3

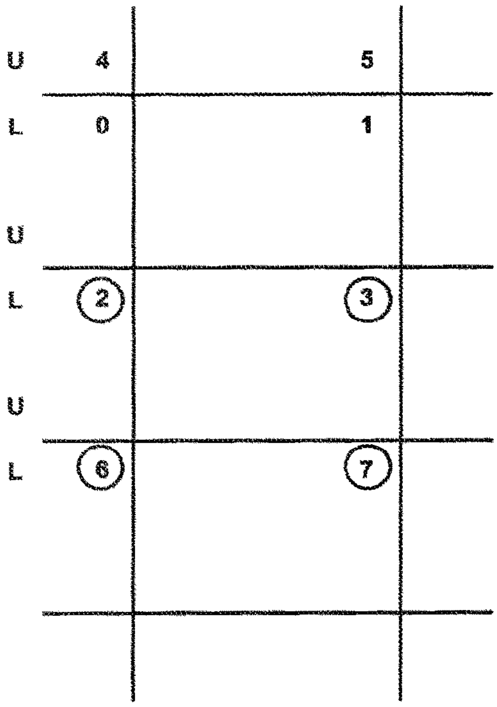


图4A

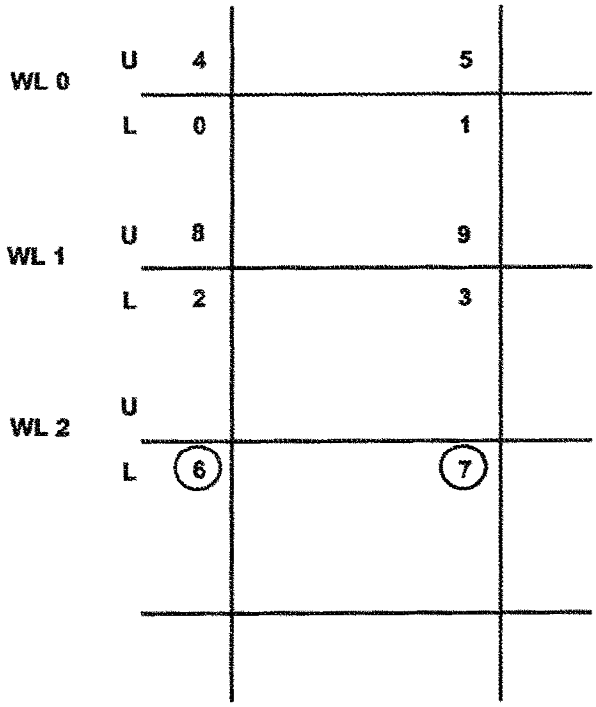


图4B

○ = 开放较低页

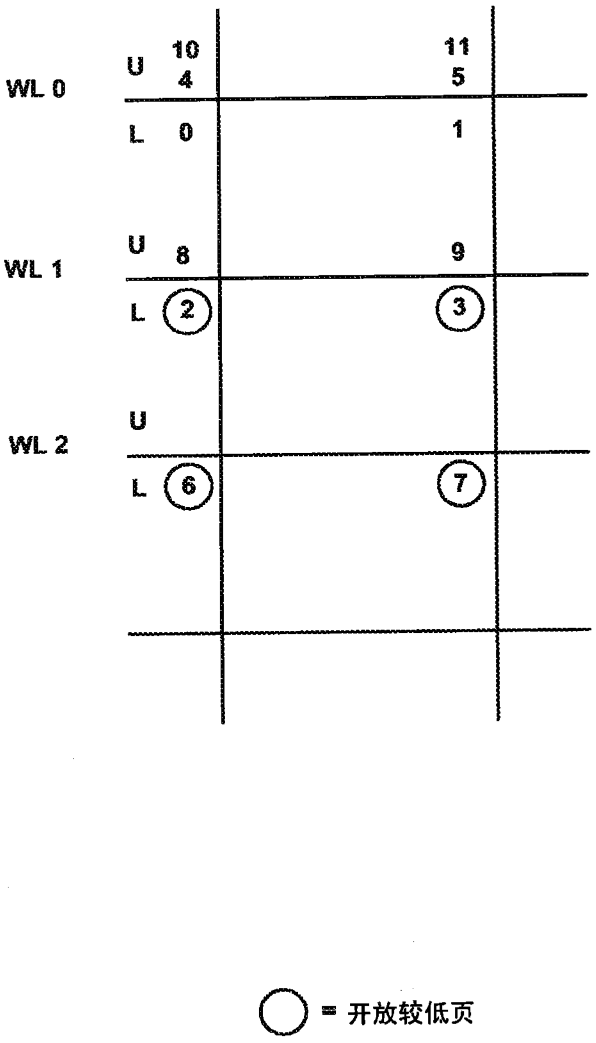


图 5

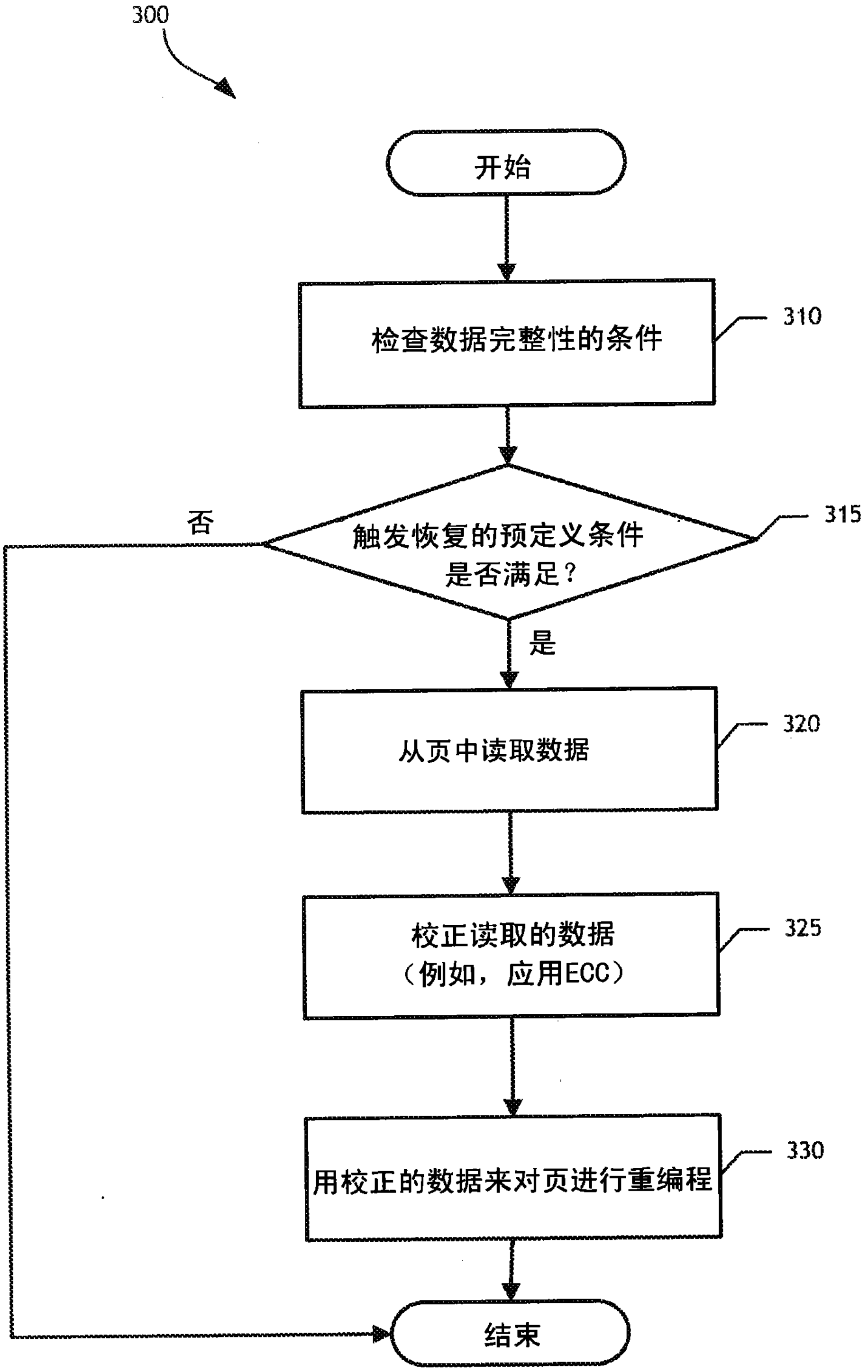

300
开始
检查数据完整性的条件
310
触发恢复的预定义条件是否满足？
315
否
是
从页中读取数据
320
校正读取的数据
（例如，应用ECC）
325
用校正的数据来对页进行重编程
330
结束

图 6

Abstract

In some embodiments of the present invention, a data storage system includes a controller and a non-volatile memory array having a plurality of memory pages. The controller performs a method that efficiently resolves the lower page corruption problem. In one embodiment, the method selects programmed lower page(s) for which paired upper page(s) have not been programmed, reads data from those selected lower page(s), corrects the read data, and reprograms the read data into those lower page(s). Since the number of lower pages in this condition is typically low (e.g., several pages in a block with hundreds or thousands of pages), this is a much more efficient method than reprogramming the entire block. In another embodiment, a similar reprogramming method is applied as a data recovery scheme in situations in which only lower pages are programmed (e.g., SLC memory, MLC memory in SLC mode, etc.).