(51) International Patent Classification:
G06F 11/10 (2006.01) G11C 16/10 (2006.01)
(21) International Application Number:
PCT/US20 12/042775
(22) International Filing Date:
15 June 2012 (15.06.2012)
(25) Filing Language:
English
(26) Publication Language:
English
(30) Priority Data:
61/498,583 19 June 2011 (19.06.2011) US
13/285,892 31 October 2011 (31.10.2011) US
(71) Applicant (for all designated States except US):
SANDISK ENTERPRISE IP LLC [US/US]; 350 North St. Paul Street, Suite 2900, Dallas, TX 75201 (US).
(72) Inventors; and
WESTON-LEWIS, Graeme, Moffat [GB/US]; 46925 Zapotec Drive, Fremont, CA 94539 (US). PRINS, Douglas, Alan [US/US]; 27096 Hidden Trail Road, Laguna Hills, CA 92653 (US).
OLBRICH, Aaron, Keith [US/US]; 14525 Atherton Circle, Morgan Hill, CA 95037 (US).

[Continued on nextpage]

(54) Title: SYSTEM AND METHOD FOR DETECTING COPYBACK PROGRAMMING PROBLEMS

(57) Abstract: In a method for detecting problems related to copyback programming, after the copyback data is read into the internal flash buffer, a part of the copyback data stored in the internal flash buffer (such as spare data) is analyzed to determine whether there are any errors in a part of the copyback data read. The analysis may be used by the flash memory in one or more ways related to the current copyback operation, subsequent copyback operations, subsequent treatment of the data in the current copyback operation, and subsequent treatment of the section in memory associated with the source page.
Published:

— with international search report (Art. 21(3))
SYSTEM AND METHOD FOR DETECTING COPYBACK
PROGRAMMING PROBLEMS

TECHNICAL FIELD

[0001] This application relates generally to managing data in a memory device. More specifically, this application relates to a flash memory implementing an improved programming sequence for detecting copyback programming problems.

BACKGROUND

[0002] Copyback is an operation used to read and copy data stored in one page (a source page) and reprogram it in another page (a destination page) in a flash memory. Unlike standard read and program operations, data retrieved from the source page is not streamed out (such as streamed out of the flash memory chip), but is saved in an internal buffer in the flash memory and then programmed directly into the destination page without using an external memory. In this way, the data is kept entirely within the flash memory chip and not sent to any external chip within the flash memory (such as to an external controller in the flash memory). Thus, the copyback operation is faster and more efficient than standard operations because reading out the data and then re-loading the data to be programmed are not required. The operation is particularly useful when a portion of a page is updated and the rest of the page needs to be copied to a newly assigned block.

[0003] Copyback operations are "blind" in that the data is moved without being checked. In particular, as the data read from the source page is internal to the flash memory during a copyback operation, the Error Correction Code (ECC) cannot be evaluated before copying the source page to the destination page. As a result, any bit error (either caused by the copyback read operation or other error) cannot be detected and will be copied into the destination page, resulting in propagation and accumulation of the bit error.

SUMMARY

[0004] In order to address problems related to copyback programming, methods and systems are disclosed herein for detecting problems related to copyback programming.

[0005] According to a first aspect, a method of detecting errors in a copyback operation in a flash memory device is disclosed. The method includes, in the flash memory device with a controller, performing: internally copying data from a first non-volatile portion in a flash memory chip of the flash memory device to a volatile portion of the flash memory...
device; using a part of the data copied to the volatile portion to detect the presence of one or more errors; internally copying the data from the volatile portion to a second non-volatile portion of the flash memory chip; and modifying some or all of the data based on the detected presence of the one or more errors. In one embodiment, the copyback operation is performed on the flash memory chip, with copying from a first non-volatile portion on the flash memory chip to a volatile portion of the flash memory chip, and then copying from the volatile portion on the flash memory chip to a second portion on the flash memory chip. In another embodiment, part of the data copied to the volatile portion is copied external to the flash memory chip, such as to a volatile memory associated with the controller. In still another embodiment, depending on the detection of the one or more errors, the data associated with the copyback operation is modified, such as by verifying the data copied to the second portion of the flash memory chip and/or modifying the part of the data stored in the volatile portion prior to copying to the second portion of the flash memory chip. In yet another embodiment, depending on the detection of the one or more errors, a subsequent copyback operation is modified, such as disabling all subsequent copyback operations for the first non-volatile portion.

[0006] In another aspect, a storage device is disclosed. The storage device may comprise a flash memory device that is configured to detect errors in a copyback operation. The flash memory device comprises: a flash memory chip that includes a first non-volatile portion, a second non-volatile portion, and a volatile portion; and a controller in communication with the flash memory chip. The controller is configured to: command the flash memory chip to internally copy data from the first non-volatile portion to the volatile portion; use part of the data copied to the volatile portion to detect the presence of one or more errors; command the flash memory chip to internally copy the data from the volatile portion to the second non-volatile portion; and modify some or all the data associated with the copyback operation based on the detected presence of the one or more errors.

[0007] Other features and advantages will become apparent upon review of the following drawings, detailed description and claims. Additionally, other embodiments are disclosed, and each of the embodiments can be used alone or together in combination. The embodiments will now be described with reference to the attached drawings.
BRIEF DESCRIPTION OF THE DRAWINGS

[0008] FIG. 1 illustrates a host connected with a flash memory device having a multi-bank non-volatile memory containing multiple die.

[0009] FIG. 2 is a flow chart of one method for analyzing data during a copyback operation and potential applications of the analysis.

[0010] FIG. 3 is a flow chart of a method to program the spare area of the flash array with program integrity check data.

[0011] FIG. 4 is a flow chart of another method for analyzing data during a copyback operation and potential applications of the analysis.

[0012] FIG. 5 illustrates charge levels in a MLC memory and migration of charge due to the effect of over-programming.

[0013] FIG. 6 illustrates charge levels in the MLC memory migration of charge due to the effect of retention loss.

DETAILED DESCRIPTION

[0014] As discussed in the background, copyback operations are "blind" in that the data is moved without being checked. In one embodiment, at least part of the data copied in the copyback process is checked during the copyback process. For example, after the copyback data is read into the internal flash buffer, a part of the copyback data stored in the internal flash buffer is analyzed to determine whether there are any errors in a part of the copyback data read.

[0015] One example of copyback data read into the internal flash is data in the source page. The data in the source page may include user data, metadata, and potentially spare data. Examples of metadata include logical block addresses (LBAs) of the user data and the relative age of the LBAs. Spare data includes one or more unused bytes in the source page.

[0016] In one aspect, part (or all) of the spare data stored in the internal flash buffer is analyzed to determine whether there is a problem in the copyback programming. To perform the analysis, part (or all) of the spare data may be moved external to the flash memory chip, such as from the internal flash buffer to a buffer within the controller of the flash memory, as discussed in more detail below. The controller may then analyze the spare data in order to determine whether there is a problem with the copyback programming.
The controller’s analysis may include comparing an expected value of the spare data with the actual value of the spare data. In one embodiment, the expected value is a predetermined integrity check data value (such as 'FF00'), sometimes herein called a predetermined sequence of bits, which may have been programmed previously into the source page prior to the start of the copyback operation. In another embodiment, the expected value is an initialization value that was programmed as part of an initialization. For example, prior to use of the source page, the source page is initialized with 'FF' values. In this way, the expected value of 'FF' need not be programmed in an operation separate from the initialization.

The comparison of the expected value with the actual value may indicate whether there is a problem with the copyback programming. For example, the analysis may indicate that one or more bytes of spare data include errors. The number of bytes that include errors may indicate that other data in the copyback programming, including the user data and the metadata, may have errors as well.

The controller's analysis may be used by the flash memory device in one or more ways related to the data associated with the current copyback operation, subsequent copyback operations, and subsequent treatment of the section in memory associated with the source page.

In one aspect, the controller's analysis may be used to modify or change the data associated with the current copyback operation, such as modify the data stored in the internal flash buffer or modify the data after it is stored back into non-volatile memory. For example, if the number of bytes in error is above a predetermined number, one or more bytes stored in the internal flash buffer (such as part (or all) of the spare data in the internal flash buffer) may be modified. As another example, the controller's analysis may be used to modify subsequent treatment of the data in the current copyback operation. For example, if the number of bytes in error is above a predetermined number, it may be determined that some (or all) of the data in the current copyback operations should be verified. So that, after copying of the data to the destination page in flash memory, the data may be verified by the controller using ECC.

In another aspect, the controller's analysis may be used to modify a subsequent copyback operation. For example, if the number of bytes in error is above a predetermined number, subsequent or future copyback operations for a particular part of the
flash memory device (such as the block associated with the source page) may be prevented. Instead, updating of the pages in the block associated with the source page may be performed by copying to the buffer within the controller and by using the controller to perform ECC on the copied data.

[0022] In yet another aspect, the controller's analysis may be used to modify subsequent treatment of the section in flash memory associated with the source page. For example, if the number of bytes in error is above a predetermined number, it may be determined that part of the non-volatile memory (such as the block associated with the source page) may be reclaimed.

[0023] A flash memory device 102 suitable for use in implementing a copyback operation is shown in FIG. 1. A host system 100 of FIG. 1 stores data into and retrieves data from the flash memory device 102. The flash memory device 102 may be flash memory embedded within the host, such as in the form of a solid state disk (SSD) drive installed in a personal computer, a computer server, or a storage array. Alternatively, the flash memory device 102 may be in the form of a card that is removably connected to the host through mating parts 104 and 106 of a mechanical and electrical connector as illustrated in FIG. 1. A flash memory device configured for use as an internal or embedded SSD drive may look similar to the schematic of FIG. 1, with the primary difference being the location of the flash memory device 102 internal to the host. SSD drives may be in the form of discrete modules that are drop-in replacements for rotating magnetic disk drives.

[0024] The host system 100 of FIG. 1 may be viewed as having two major parts, insofar as the flash memory device 102 is concerned, made up of a combination of circuitry and software. They are an applications portion 108 and a driver portion 110 that interfaces with the flash memory device 102. In a PC, for example, the applications portion 108 can include a processor, such as CPU 112, running word processing, graphics, control or other popular application software, as well as the file system 114 for managing data on the host 100. In a camera, cellular telephone or other host system that is primarily dedicated to perform a single set of functions, the applications portion 108 includes the software that operates the camera to take and store pictures, the cellular telephone to make and receive calls, and the like.

[0025] The flash memory device 102 of FIG. 1 may include non-volatile memory, such as flash memory 116, and a system controller 118. The system controller 118 controls
the flash memory 116 and communicates with the host 100 to which the flash memory device 102 is connected in order to pass data back and forth. The system controller 118 may convert between logical addresses of data used by the host 100 and physical addresses of the flash memory 116 during data programming and reading.

[0026] The flash memory 116 may include one or more flash memory chips 130. The flash memory chip 130 includes a flash memory array 120, which is a non-volatile memory, and a flash memory buffer 128, which is a volatile memory. FIG. 1 illustrates a single flash memory chip 130 with a single flash memory array 120 and a single flash memory buffer 128 by way of example. Further, the flash memory 116 may include multiple die, each containing an array of memory cells organized into multiple planes, and a volatile memory buffer. Alternatively, the memory cell array may not be divided into planes. In an alternate embodiment, the flash memory chips 130 include the flash memory array 120, and a separate chip may include the flash memory buffer 128.

[0027] Functionally, the system controller 118 may include a controller 122, which may comprises a processor, control logic, or the like. The system controller may also include controller firmware 124 for coordinating operation of the flash memory 116, such as monitoring copyback programming as disclosed below in FIGS. 2-4. The system controller 118 may further include controller RAM 126 (or other volatile memory associated with the controller 122). The system controller 118 may be implemented on a single integrated circuit chip, such as an application specific integrated circuit (ASIC), or may be also be incorporated into flash memory 116.

[0028] For example, the system controller 118 may initiate the copyback operation by sending a command to a particular flash memory chip in the flash memory 116 to move data from a source page of non-volatile memory into the flash memory buffer of the particular flash memory chip. This is discussed, for example, at 202 in FIG. 2 and 402 in FIG. 4. The system controller 118 may further send a command (such as a direct memory access (DMA) command) to the particular flash memory chip in order to copy data stored in the flash memory buffer for storage in the controller RAM 126. In one embodiment, the command to copy data stored in the flash memory buffer for storage in the controller RAM 126 moves the data off-chip (from the particular flash memory chip to the controller chip). As discussed below, "spare" data is copied from the flash memory buffer into the controller RAM. This is discussed, for example, at 404 in FIG. 4. Moreover, the system controller 118 may command the particular flash chip to store data sent from the system controller into the flash memory.
buffer. This is discussed, for example, at 212 in FIG. 2 and 412 in FIG. 4. In addition, the system controller 118 may command the particular flash chip to store the data in the flash memory buffer into the destination page in the non-volatile flash memory array 120. This is discussed, for example, at 226 in FIG. 2.

[0029] The memory cells may be operated to store more than two detectable levels of charge in each charge storage element or region, thereby to store more than one bit of data in each. This configuration is referred to as multi level cell (MLC) memory. Alternatively, the memory cells may be operated to store two levels of charge so that a single bit of data is stored in each cell. This is typically referred to as a binary or single level cell (SLC) memory. Both types of memory cells may be used in a memory, for example binary flash memory may be used for caching data and MLC memory may be used for longer term storage. The charge storage elements of the memory cells are most commonly conductive floating gates but may alternatively be non-conductive dielectric charge trapping material.

[0030] In implementations of MLC memory operated to store two bits of data in each memory cell, each memory cell is configured to store four levels of charge corresponding to values of "11," "01," "10," and "00." Each bit of the two bits of data may represent a page bit of a lower page or a page bit of an upper page, where the lower page and upper page span across a series of memory cells sharing a common word line. Typically, the less significant bit of the two bits of data represents a page bit of a lower page and the more significant bit of the two bits of data represents a page bit of an upper page.

[0031] As discussed above, copyback operations are typically "blind" in that the data is moved without being checked. As a result, any bit error from the source page cannot be detected and will be copied into the destination page, resulting in propagation and accumulation of the bit error. Errors may be due, for example, to over-programming, which may cause bits in the memory cells to gain charge, and to retention loss, which may cause bits in the memory cells to lose charge. The charge in the cells of the flash memory may thus move due to the various errors, resulting in the bits to move as well, such as to the adjacent right or left state by over-programming or retention.

[0032] As discussed below, a part of the data (such as the spare data) stored in the flash buffer memory 128 is analyzed. The analysis of the spare data may be used to determine if the majority movement is towards charge gain or charge loss in a flash memory cell. For example, in a flash memory with a bit assignment, by upper page and lower page
bit, of 11, 01, 00, and 10, over-programming may cause ‘1’s to tend to ‘0’ and retention loss may cause ‘0’s to tend to ‘1’s. For the analysis of the spare data to account for both over-programming and retention loss, the spare bits may be programmed with both ‘1’s and ‘0’s (such as FF). Alternatively, if the analysis is only focusing on one of the errors, then the spare bits may include only ‘1’ s, so that no preprogramming of a special sequence is necessary.

[0033] Referring to FIG. 5, there is shown the effect of errors due to over-programming on a specific type of MIX. In particular, FIG. 5 illustrates ideal charge levels 502, 504, 506, 508 for bits 11, 01, 00, 10, respectively. Further, there is shown curves 510, 512, 514, 516 which illustrate the potential charge levels due to effect of over-programming. The V_a, V_b, and V_c are read reference voltages. Thus, FIG. 5 illustrates one example of the effect of over-programming on the values of the bits in 2-bit MLC flash memory. FIG. 5 is for illustration purposes. Other types of flash memory layouts may be affected differently by over-programming.

[0034] Similar to FIG. 5, FIG. 6 illustrates ideal charge levels 502, 504, 506, 508 for bits 11, 01, 00, 10, respectively. Further, there is shown curves 602, 604, 606, 608 which illustrate the potential charge levels due to effect of retention loss. Thus, FIG. 6 illustrates one example of the effect of retention-loss on the values of the bits in 2-bit MLC flash memory. FIG. 6 is for illustration purposes. Other types of flash memory layouts may be affected differently by retention loss.

[0035] FIG. 2 is a flow chart 200 of one method for analyzing data during a copyback operation and potential applications of the analysis. At 202, data (such as a source page) is copied from flash non-volatile memory (such as flash memory array 120) to flash volatile memory (such as flash memory buffer 128). At 204, the controller 122 analyzes at least a part of the data in the flash volatile memory. As discussed above, the spare data copied into the flash memory buffer 128 is analyzed. At 206, the controller 122 determines whether the analysis finds one or more errors. If not, at 226, the data in the flash volatile memory is written to the flash non-volatile memory. If so, at 208, the controller 122 logs the details of the one or more errors, such as the bits in the spare data that were in error. At 210, the controller 122 determines whether to fix errors in the data in the flash volatile memory. If so, at 212, at least a part of the data in the flash volatile memory is modified. As discussed above, the data in the flash volatile memory may include user data, metadata, and spare data. In one aspect, part (or all) of the spare data is modified.
[0036] At 214, the controller 122 determines whether to disable subsequent copyback operations. If so, at 216, the controller 122 logs disabling of subsequent copyback operations for a part of the flash non-volatile memory, such as for the source page or for the entire block associated with the source page. At 218, the controller 122 determines whether to correct the data in the copyback operation. If so, at 220, the data is corrected and moved to flash non-volatile memory.

[0037] At 222, the controller 122 determines whether to reclaim part of the non-volatile memory. If the analysis indicates serious errors in the copyback data (such as numerous errors in the spare data stored in the flash volatile memory), the controller 122 may determine that a section of memory should be reclaimed and no longer used. If so, at 224, a part of the flash non-volatile memory (such as the block associated with the source page) is indicated to be reclaimed. At 226, the data in the flash volatile memory is written to the flash non-volatile memory.

[0038] FIG. 3 is a flow chart 300 of a method to program the spare area of the flash array with program integrity check data. As discussed above, in certain instances, the flash non-volatile memory may be programmed with a special code for later checking during the copyback process. At 302, a page of user data is moved into the flash memory buffer (such as flash memory buffer 128). At 304, program integrity check data is moved into "spare" area of the flash memory buffer. As discussed above, the spare area of memory may include the space in the flash memory buffer that is not used even with a full page of data loaded into the flash memory buffer. At 306, the contents of the flash memory buffer are written to the flash memory array.

[0039] FIG. 4 is a flow chart 400 of another method for analyzing data during a copyback operation and potential applications of the analysis. At 402, a copyback read is performed. At 404, the "spare" area is moved from the flash memory buffer into the stage buffer (which may be controller RAM 126). At 406, the controller 122 may perform a quick check of the "spare" bytes. At 408, the controller 122 may determine whether the "spare" bytes include more than 1 byte in error. If so, at 410, the controller 122 may perform a detailed check of the "spare" area and log the details. Optionally, at 412, the controller 122 may fix any bytes in error in the stage buffer and move the bytes into the flash memory buffer. For example, the controller 122 may input the correct predetermined code (such as the program integrity check data) into the stage buffer, which in turn is sent to the flash.
memory buffer, and ultimately programmed into the flash memory array. In this way, the errors in the "spare" area need not be propagated.

[0040] Optionally, at 414, copyback is disabled for the block associated with the source page. The determination whether to disable copyback for the block associated with the source page may be based on a first predetermined number of "spare" bytes that are in error (as shown in FIG. 4, the first predetermined number of "spare" bytes that are in error is greater than 1). Optionally, at 416, the controller 122 may force verification of the page after the write to the flash memory array. The determination whether to disable copyback for the block associated with the source page may be based on a second predetermined number of "spare" bytes that are in error (as shown in FIG. 4, the second predetermined number of "spare" bytes that are in error is greater than 1). In one embodiment, the first predetermined number of error bytes may be the same as the second predetermined number of error bytes (such as shown in FIG. 4) or may be different. Further, at 418, if more than a third predetermined number of bytes are in error (3, as shown in FIG. 4), the controller 122 forces immediate reclaim of the block associated with the source page, and logs an immediate reclaim request. If the errors are considered severe enough, the controller 122 may determine that the block associated with the source page is unreliable enough that it should not be used further, leading to the block being retired from further use. FIG. 4, including the various steps and the number of error bytes, are merely for illustration purposes. Other steps, and other numbers of error bytes, may be used.

[0041] It is intended that the foregoing detailed description be understood as an illustration of selected forms that the invention can take and not as a definition of the invention. It is only the following claims, including all equivalents, which are intended to define the scope of this invention. Also, some of the following claims may state that a component is operative to perform a certain function or configured for a certain task. It should be noted that these are not restrictive limitations. It should also be noted that the acts recited in the claims can be performed in any order and not necessarily in the order in which they are recited.
WHAT IS CLAIMED IS:

1. A method of detecting errors in a copyback operation in a flash memory device, the method comprising:
   in the flash memory device with a controller, performing:
   internally copying data from a first non-volatile portion in a flash memory chip of the flash memory device to a volatile portion of the flash memory device;
   using a part of the data copied to the volatile portion to detect the presence of one or more errors;
   internally copying the data from the volatile portion to a second non-volatile portion of the flash memory chip; and
   modifying some or all of the data based on the detected presence of the one or more errors.

2. The method of claim 1, wherein the volatile portion is in the flash memory chip; and
   further comprising copying part of the data from the volatile portion to an associated memory of the controller, the associated memory of the controller on a chip separate from the flash memory chip; and
   wherein detecting the presence of the one or more errors comprises analyzing the part of the data in the associated memory for the one or more errors.

3. The method of claim 2, wherein internally copying data from the first non-volatile portion to the volatile portion comprises internally copying a page of data; and
   wherein copying part of the data from the volatile portion to the associated memory of the controller comprises copying less than the entire page of data.

4. The method of claim 2 or 3, wherein the page of data comprises user data, meta data, and spare data;
   wherein the spare data is copied to the associated memory of the controller; and
   wherein analyzing the part of the data comprises analyzing the spare data copied to the associated memory for the one or more errors.

5. The method of claim 4, further comprising, prior to the copyback operation, programming the page of data with a predetermined sequence of bits so that the spare data includes the predetermined sequence of bits; and
wherein analyzing the spare data copied to the associated memory for the one or more-
errors includes determining whether the spare data includes the predetermined sequence.

6. The method of any preceding claim, further comprising determining whether to disable copyback operations.

7. The method of any preceding claim, wherein modifying some or all of the data comprises, after copying the data to the second non-volatile portion, modifying some or all of the data copied to the second non-volatile portion.

8. The method of claim 7, wherein modifying some or all of the data copied to the second non-volatile portion of the flash memory chip comprises verifying the data copied to the second non-volatile portion.

9. The method of any preceding claim, wherein modifying some or all of the data comprises modifying, prior to copying the data to the second non-volatile portion, the part of the data copied to the volatile portion.

10. The method of claim 9, wherein the modifying corrects the one or more errors detected.

11. The method of any preceding claim, further comprising determining, based on the detected one or more errors, whether to reclaim the first non-volatile portion of the flash memory chip.

12. The method of any of claims 1-10, further comprising determining, based on the detected one or more errors, whether to retire the first non-volatile portion of the flash memory chip.

13. A flash memory device configured to detect errors in a copyback operation, the flash memory device comprising:

   a flash memory chip comprising a first non-volatile portion, a second non-volatile portion, and a volatile portion; and

   a controller in communication with the flash memory chip, the controller configured to:

   command the flash memory chip to internally copy data from the first non-
   volatile portion to the volatile portion;

   use part of the data copied to the volatile portion to detect the presence of one
   or more errors;
command the flash memory chip to internally copy the data from the volatile portion to the second non-volatile portion; and
modify some or all of the data based on the detected presence of the one or more errors.

14. The flash memory device of claim 13, wherein the controller is on a chip separate from the flash memory chip;
further comprising an associated memory of the controller;
wherein the controller is further configured to copy part of the data from the volatile portion to the associated memory of the controller; and
wherein the controller is configured to detect the presence of one or more errors by analyzing the part of the data in the associated memory for the one or more errors.

15. The flash memory device of claim 14, wherein the controller is configured to command the flash memory chip to internally copy data from the first non-volatile portion to the volatile portion by commanding the flash memory chip to internally copy a page of data; and
wherein the controller is configured to command the flash memory chip to copy part of the data from the volatile portion to the associated memory of the controller by copying less than the entire page of data.

16. The flash memory device of claim 14 or 15, wherein the page of data comprises user data, meta data, and spare data;
wherein the spare data is copied to the associated memory of the controller; and
wherein the controller is configured to analyze the part of the data by analyzing the spare data copied to the associated memory for the one or more errors.

17. The flash memory device of claim 16, wherein, prior to the copyback operation, the controller is further configured to program the page of data with a predetermined sequence of bits so that the spare data includes the predetermined sequence of bits; and
wherein the controller is configured to analyze the spare data copied to the associated memory for the one or more errors by determining whether the spare data includes the predetermined sequence.

18. The flash memory device of any of claims 13-17, wherein the controller is further configured to determine whether to disable subsequent copyback operations.
19. The flash memory device of any of claims 13-18, wherein the controller is configured to modify some or all of the data by, after copying the data to the second non-volatile portion, modifying some or all of the data copied to the second non-volatile portion.

20. The flash memory device of claim 19, wherein the controller is configured to modify some or all of the data copied to the second non-volatile portion of the flash memory chip by verifying the data copied to the second non-volatile portion.

21. The flash memory device of any of claims 13-19, wherein the controller is configured to modify some or all of the data by modifying, prior to copying the data to the second non-volatile portion, the part of the data copied to the volatile portion.

22. The flash memory device of any of claims 21, wherein the controller is configured to modify by correcting the one or more errors detected.

23. The flash memory device of any of claims 13-22, wherein the controller is further configured to determine, based on the detected one or more errors, whether to reclaim the first non-volatile portion of the flash memory chip.

24. The flash memory device of any of claims 13-22, wherein the controller is further configured to determine, based on the detected one or more errors, whether to retire the first non-volatile portion of the flash memory chip.
Start

Move page of user data into flash memory buffer

Move program integrity check data into "spare" area of flash memory buffer

Write flash memory buffer to flash memory array

End

FIG. 3
Start

Do Copyback Read

Move "spare" area from flash memory buffer to Stage Buffer

Do quick check of "spare" bytes

More than 1 byte in error?

Yes

Do detailed check of "spare" area and log the details

Fix any bytes in error in the stage buffer and move to flash memory buffer

Disable copyback for this block

No

Force verify of page after write

If more than 3 bytes in error, force immediate reclaim of this block and log immediate reclaim request

End

FIG. 4

4/6

SUBSTITUTE SHEET (RULE 26)
## A. CLASSIFICATION OF SUBJECT MATTER

**INV.** G06F11/10 G11C16/10

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 G11C

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

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

EPO-Internal

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>wo 2009/134576 Al (APPLE INC [US]; WAKRAT NI R JACOB [US]; HELM MARK ALAN [US]) 5 November 2009 (2009-11-05) page 1, paragraph 1 - paragraph 2 page 2, paragraph 12 - page 5, paragraph 23; figures 1A, IB, 2, 3A, 3B</td>
<td>1-24</td>
</tr>
<tr>
<td>X</td>
<td>US 2005/172065 Al (KEAYS BRADY L [US]) 4 August 2005 (2005-08-04) page 4, paragraph 31 page 6, paragraph 42 - paragraph 45 page 7, paragraph 53 - page 8, paragraph 56; figures 3A, 3B, 4</td>
<td>1-24</td>
</tr>
</tbody>
</table>

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

* Special categories of cited documents:
  - "P" document published prior to the international filing date but later than the priority date claimed
  - "L" document which may throw doubts on priority claim(s) one of which is cited to establish the publication date of another cited or other special reason (as specified).
  - "D" document referring to an oral disclosure, use, exhibition or other means
  - "T" 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
  - "F" 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
  - "S" document member of the same patent family

Date of the actual completion of the international search: 18 September 2012

Date of mailing of the international search report: 26/09/2012

Name and mailing address of the ISA: European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016

Authorized officer: Bauer, Regine

Form PCT/ISA/210 (second sheet) (April 2005)
<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>US 2005/289314 Al (ADUSUMI LLI VIJAYA P [US] ET AL) 29 December 2005 (2005-12-29) page 1, paragraph 3 - paragraph 10 page 2, paragraph 25 - page 3, paragraph 29; figures 2,3</td>
<td>1-24</td>
</tr>
<tr>
<td>X</td>
<td>EP 1 465 203 Al (SAMSUNG ELECTRONICS CO LTD [KR]) 6 October 2004 (2004-10-06) page 1, paragraph 4 - paragraph 7 page 4, paragraph 25 - paragraph 27 claim 1; figures 2,3</td>
<td>1-24</td>
</tr>
<tr>
<td>X</td>
<td>US 2009/193058 Al (REID ROBERT ALAN [US]) 30 July 2009 (2009-07-30) page 1, paragraph 6 - paragraph 14 page 3, paragraph 32 - paragraph 37 page 4, paragraph 41 - paragraph 42; figures 5,8</td>
<td>1-24</td>
</tr>
<tr>
<td>Patent document cited in search report</td>
<td>Publication date</td>
<td>Patent family member(s)</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-----------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>WO 2009134576 A1</td>
<td>05-11-2009</td>
<td>AT 535866 T</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CN 102077176 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 2297642 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 2407883 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ES 2378371 T3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 2011520188 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KR 20110008301 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KR 20120059658 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2009276560 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2012233387 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO 2009134576 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KR 20080083474 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2008229000 A1</td>
</tr>
<tr>
<td>US 2005172065 A1</td>
<td>04-08-2005</td>
<td>NONE</td>
</tr>
<tr>
<td>US 2005289314 A1</td>
<td>29-12-2005</td>
<td>CN 101031895 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 1769362 A2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 2008504637 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KR 20070024624 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2005289314 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO 2006007264 A2</td>
</tr>
<tr>
<td>EP 1465203 A1</td>
<td>06-10-2004</td>
<td>CN 1551244 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CN 102034545 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DE 602004003275 T2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 1465203 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 2004311010 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KR 20040086923 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2004202034 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2008163030 A1</td>
</tr>
<tr>
<td>US 2009193058 A1</td>
<td>30-07-2009</td>
<td>NONE</td>
</tr>
</tbody>
</table>