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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
16 July 2015 (16.07.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/106263 A1

(51) International Patent Classification:
G11C 16/26 (2006.01) *G11C 16/34* (2006.01)

(21) International Application Number:
PCT/US2015/011189

(22) International Filing Date:
13 January 2015 (13.01.2015)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/926,516 13 January 2014 (13.01.2014) US
14/156,693 16 January 2014 (16.01.2014) US

(71) Applicant: **SEAGATE TECHNOLOGY LLC** [US/US]; 10200 S. De Anza Blvd., Cupertino, California 95014 (US).

(72) Inventors: **CHEN, Ning**; 1320 Ridder Park Drive, San Jose, California 95131 (US). **CAI, Yu**; 1320 Ridder Park Drive, San Jose, California 95131 (US). **WU, Yunxiang**; 1320 Ridder Park Drive, San Jose, California 95131 (US).

(74) Agent: **MAIORANA, Christopher**; Suite 100, 24840 Harper Avenue, St. Clair Shores, Michigan 48080 (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, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, 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.

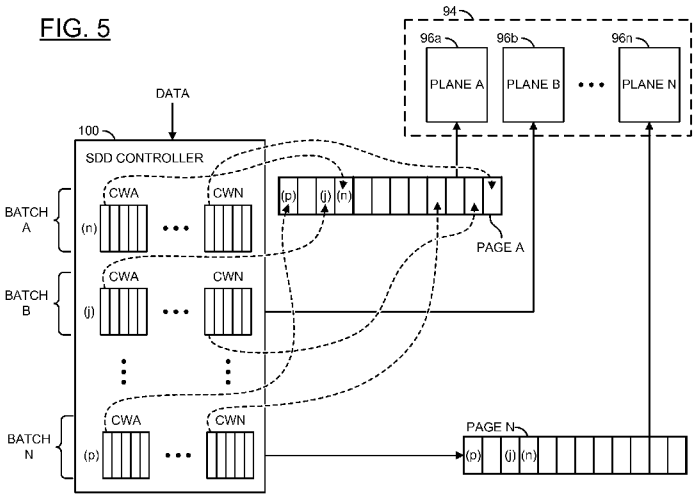
(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, 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).

Published:

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

(54) Title: INTERLEAVING CODEWORDS OVER MULTIPLE FLASH PLANES

FIG. 5



(57) Abstract: An apparatus having an interface to a plurality of memories and a circuit is disclosed. Each memory generally has a plurality of planes and is nonvolatile. The circuit is configured to (i) generate a plurality of codewords by encoding a plurality of data units, (ii) generate a plurality of slices by parsing the codewords, (iii) generate a plurality of pages by interleaving the slices and (iv) write the pages in parallel into respective ones of the planes.

WO 2015/106263 A1

INTERLEAVING CODEWORDS OVER MULTIPLE FLASH PLANES

This application relates to U.S. Provisional Application No. 61/926,516, filed January 13, 2014, which is hereby incorporated by reference in its entirety.

Field of the Invention

The invention relates to solid-state drives generally and, more particularly, to a method and/or apparatus for interleaving codewords over multiple flash planes.

Background

As flash memory geometries scale down, an amount of electrons stored in each flash memory cell decreases. The smaller number of electrons is more susceptible to process variations and so reliability differences among the various flash planes and die become more pronounced. In conventional designs, each codeword is stored on the same die. Therefore, if the error correction coding cannot recover the errors of certain codewords on the worst flash die/plane, an uncorrectable code correction failure happens.

Summary

The invention concerns an apparatus having an interface to a plurality of memories and a circuit. Each memory generally has a plurality of planes and is nonvolatile. The circuit is configured to (i) generate a plurality of codewords by encoding a plurality of data units, (ii) generate a plurality of slices by parsing the codewords, (iii) generate a plurality of pages by interleaving the slices and (iv) write the pages in parallel into respective ones of the planes.

Brief Description of the Figures

Embodiments of the invention will be apparent from the following detailed description and the appended claims and drawings in which:

FIG. 1 is a block diagram of an apparatus;

FIG. 2 is a block diagram of a redundancy block;

FIG. 3 is a flow diagram of a method for writing in accordance with an embodiment of the invention;

FIG. 4 is a diagram of a codeword divided into slices;

FIG. 5 is a diagram for mapping the slices into memory planes;

FIG. 6 is a flow diagram of a method for reading; and

FIG. 7 is a diagram for mapping data from the memory planes into the multiple slices.

Detailed Description of the Embodiments

Embodiments of the invention include interleaving codewords over multiple flash planes that may (i) distribute the codewords across multiple flash devices, (ii) improve reliability compared to existing techniques, (iii) average error rates over multiple flash devices, (iv) decrease process variation induced errors inside the codewords and/or (v) be implemented as one or more integrated circuits.

In various embodiments, the codewords are sliced and distributed across multiple flash devices (or die) and/or multiple planes within the flash devices. An error number inside a single codeword is thus an average error rate over multiple die multiplied by the codeword size. Therefore, a worst-case number of raw errors inside the codewords are reduced. The slicing and distributing result in improved reliability compared with common techniques. A premature failure caused by an unreliable flash plane is delayed or potentially eliminated. As such, a program/erase cycle lifetime and a maximum retention time of the flash devices are increased. The slicing and distributing also results in a more predictable performance. Because the errors inside the codewords are averaged, the variations of the errors inside each logical codeword are decreased. Thus, the performance of decoding variations is decreased making the performance more predictable.

Referring to FIG. 1, a block diagram of an example implementation of an apparatus 90 is shown. The apparatus (or circuit or device or integrated circuit) 90 implements a computer having a nonvolatile memory circuit. The apparatus 90 generally comprises a block (or circuit) 92, a block (or circuit) 94 and a block (or circuit) 100. The circuits 94 and 100 form a drive (or device) 102. The circuits 92 to 102 may represent modules and/or blocks that may be implemented as hardware, software, a combination of hardware and software, or other implementations.

One or more signals (e.g., HOSTIO) are exchanged between an interface of the circuit 92 and an interface of the circuit 100. The host input/output signal HOSTIO generally includes, but is not limited to, a logical address component used to access data in the circuit 102, a host command component that controls the circuit 102, a write data component that transfers write data units from

the circuit 92 to the circuit 100 and a read data component that transfers error corrected read data units from the circuit 100 to the circuit 92. One or more signals (e.g., NVMIO) are exchanged between an interface of the circuit 100 and another interface of the circuit 94. The nonvolatile memory input/output signal NVMIO generally includes, but is not limited to, a physical address component used to access data in the circuit 94, a memory command component that controls the circuit 94 (e.g., read or write commands), a write codeword component that carries error correction coded and cyclical redundancy check protected write codewords written from the circuit 100 into the circuit 94 and a read codeword component that carries the error correction coded codewords read from the circuit 94 to the circuit 100.

The circuit 92 is shown implementing a host circuit. The circuit 92 is generally operational to read and write data to and from the circuit 94 via the circuit 100. When reading or writing, the circuit 92 transfers a logical address value in the signal HOSTIO to identify which set of data is to be written or to be read from the circuit 94. The address generally spans a logical address range of the circuit 102. The logical address can address individual data units, such as SATA (e.g., serial-ATA) sectors.

The circuit 94 is shown implementing one or more nonvolatile memory circuits (or devices). According to various embodiments, the circuit 94 comprises one or more nonvolatile semiconductor devices. The circuit 94 is generally operational to store data in a nonvolatile condition. When data is read from the circuit 94, the circuit 94 accesses a set of data (e.g., multiple bits) identified by the address (e.g., a physical address) in the signal NVMIO. The address generally spans a physical address range of the circuit 94.

In some embodiments, the circuit 94 is implemented as one or more flash memories. The circuit 94 may be implemented as a single-level cell (e.g., SLC) type circuit. A single-level cell type circuit generally stores a single bit per memory cell (e.g., a logical 0 or 1). In other embodiments, the circuit 94 may be implemented as a multi-level cell type circuit. A multi-level cell type circuit is capable of storing multiple (e.g., two) bits per memory cell (e.g., logical 00, 01, 10 or 11). In still other embodiments, the circuit 94 may implement a triple-level cell type circuit. A triple-level cell circuit stores multiple (e.g., three) bits per memory cell (e.g., a logical 000, 001, 010, 011, 100, 101, 110 or 111). A four-level cell type circuit may also be implemented.

Data within the circuit 94 is generally organized in a hierarchy of units. A block is a smallest quantum of erasing. A page is a smallest quantum of writing. A codeword (or read unit

or Epage or ECC-page) is a smallest quantum of reading and error correction. Each block includes an integer number of pages. Each page includes an integral number of codewords.

The circuit 100 is shown implementing a controller circuit. The circuit 100 is generally operational to control reading to and writing from the circuit 94. The circuit 100 generates write codewords by encoding data units received from the circuit 92. The circuit 100 includes an ability to decode the read codewords received from the circuit 94. The resulting decoded data is presented to the circuit 92 via the signal HOSTIO and/or re-encoded and written back into the circuit 94 via the signal NVMIO. The circuit 100 comprises one or more integrated circuits (or chips or die) implementing the controller of one or more solid-state drives, embedded storage, or other suitable control applications.

As part of the writing, the circuit 100 is generally configured to generate multiple codewords by encoding respective data units. Several codewords are assembled together to form multiple batches, each batch containing two or more codewords. A size of each batch substantially matches a size of a plane in the circuit 94. The circuit 100 also generates multiple slices by parsing the batches/codewords. The slices are interleaved and gathered together into several pages. The circuit 100 uses multiple pages to create a redundancy block that is written into the circuit 94.

As part of the reading, the circuit 100 is generally configured to regenerate the pages by reading the redundancy block from the circuit 94. The pages are parsed to regenerate the slices. The circuit 100 subsequently regenerates the codewords from the slices. The codewords are decoded to regenerate the original data units.

The circuit 102 is shown implementing a solid-state drive. The circuit 102 is generally operational to store data generated by the circuit 92 and return the data to the circuit 92. According to various embodiments, the circuit 102 comprises one or more: nonvolatile semiconductor devices, such as NAND Flash devices, phase change memory (e.g., PCM) devices, or resistive RAM (e.g., ReRAM) devices; portions of a solid-state drive having one or more nonvolatile devices; and any other volatile or nonvolatile storage media. The circuit 102 is generally operational to store data in a nonvolatile condition.

Referring to FIG. 2, a block diagram of an example redundancy block N is shown. The redundancy block N generally comprises multiple blocks 120a-120n (e.g., N0-N63). Each block 120a-120n has multiple pages. Each block 120a-120n is stored on a different one of several die 96a-96n (e.g., Die 0-Die 63) of the circuit 94. For example, 8 kilobits of data cover 8 die 96a-96n at 1

kilobit per page. In some situations, the redundancy block N has fewer blocks 120a-120n than the number of die 96a-96n. In other situations, the redundancy block N has a larger number of blocks 120a-120n than the number of die 96a-96n.

At a start of the life of the redundancy block N, all of the blocks 120a-120n are read using a single set of average channel parameters. Over time and as the redundancy block N is programmed and erased, one or more blocks (e.g., blocks 120c and 120m) may be identified as outlier blocks having a higher error rate. Therefore, the error correction coding of the outlier blocks is increased to account for the increased error rate.

By applying the redundancy block as a coarse granularity, circuit 100 optionally provides a fault tolerant capability that allows for the loss of one or more blocks 120a-120n (or the corresponding die 96a-96n). In various embodiments, the circuit 100 is operational to generate redundant information (e.g., parity information) from the data being stored in the redundancy block N. The redundant information generally allows reconstruction of the data in the event that one or more of the blocks 120a-120n fails and/or loses power. The data reconstruction is similar to the reconstruction in a redundant array of independent disk (e.g., RAID) hard disk drive. The redundant information is stored in one or more of the blocks 120a-120n of the redundancy block N. The fault tolerance of the redundant information is adjustable. For example, a single redundant block (e.g., 120a) is used to store redundant information sufficient to recover from the loss of a single block 120b-120n. Two redundant blocks (e.g., 120a-120b) are used to recover from the loss of two blocks 120c-120n. Where the redundant information is a mirror copy of the data (e.g., RAID 0), half the blocks 120a-120n may store the data and the other half stores the mirrored copy of the data.

Referring to FIG. 3, a flow diagram of an example method 140 for writing is shown in accordance with an embodiment of the invention. The method (or process) 140 is implemented by the circuit 100. The method 140 generally comprises a step (or state) 142, a step (or state) 144, a step (or state) 146, a step (or state) 148, a step (or state) 150, and a step (or state) 152. The steps 142 to 152 may represent modules and/or blocks that may be implemented as hardware, software, a combination of hardware and software, or other implementations. The sequence of the steps 142 to 152 is shown as a representative example. Other step orders may be implemented to meet the criteria of a particular application.

In the step 142, the circuit 100 encodes data units received in a signal (e.g., DATA) from the circuit 92 (via the signal HOSTIO), read from internal to the circuit 100 and/or read from

the circuit 94 as part of maintenance operations. The codewords are assembled into batches by the circuit 100 in the step 144. Each batch has a bit-width that is as large as (substantially matches) or is smaller than the bit-width of a plane in the devices 96a-96n. In the step 146, the batches and/or the codewords within the batches are divided into slices by the circuit 100.

Referring to FIG. 4, a diagram of an example codeword 160 is shown. The data of each codeword 160 is parsed (or divided) into slices 162a-162n with similar sizes. Each slice 162a-162n is eventually stored in a different flash plane. As a result, the error rate inside the codeword 160 is the average number over all flash planes. The number of errors inside the codeword 160 is smaller than the number of errors if the codeword 160 is stored in the worst plane, instead of distributing the codeword 160 over multiple planes. A size of the codeword 160 varies depending on the level of error correction coding (or code rate) performed on the corresponding data unit. Higher error correctable coding generally results in larger codewords 160 than lower error correctable coding.

Referring to FIG. 5, a diagram of an example mapping of the slices 162a-162n of multiple codewords into the devices 96a-96n is shown. Returning to FIG. 3, the batches (e.g., batch A to batch N) created in the step 144 are gathered (or assembled) together by the circuit 100 in the form of a redundancy block having multiple pages (e.g., page A to page N). The slicing in the step 146 divides the codewords (e.g., CWA to CWN) in each batch A-N (and thus divides the batches A-N) into smaller pieces.

In the step 148, the slices 162a-162n from different codewords CWA-CWN in each batch A-N and/or the slices 162a-162n from different batches A-N are interleaved into the pages A-N. In the example shown in FIG. 5, the page A contains multiple (e.g., two) slices from the batch A, multiple (e.g., two) slices from the batch B and multiple (e.g., two) slices from the batch N. In various embodiments, one or more of the batches A-N may become a page without interleaving the slices 162a-162n. As a result, the non-interleaved page is stored in a plane (e.g., plane B) of a device (e.g., circuit 96b).

In the step 150, the pages A-N containing the interleaved and/or non-interleaved slices 162a-162n are assembled into a redundancy block. The redundancy block is subsequently written into the circuit 94 in the step 152. As part of the write operation, each page A-N is stored in a respective plane among the planes A-N. In various embodiments, each of the respective planes

A-N are stored in a different device 96a-96n. In some embodiments, two or more of the respective planes are in the same device 96a-96n (e.g., 2 or 4 planes in a die can be accessed in parallel).

Consider an example where each plane (e.g., physical page) has a size of m codewords and each codeword has n slices. A k-th slice of all codewords on the same column form a virtual codeword. At a slice per column, the m columns form up to m virtual codewords. The m virtual codewords are written to the k-th flash plane in the circuit 94. The interleave mapping causes no additional hardware costs and no read/write performance degradation. Using the mapping, each physical codeword is distributed to multiple flash planes and thus the worst-flash-plane problem is overcome.

Referring to FIG. 6, a flow diagram of an example method 180 for reading is shown. The method (or process) 180 is implemented by the circuit 100. The method 180 generally comprises a step (or state) 182, a step (or state) 184, a step (or state) 186, a step (or state) 188, a step (or state) 190, and a step (or state) 192. The steps 182 to 192 may represent modules and/or blocks that may be implemented as hardware, software, a combination of hardware and software, or other implementations. The sequence of the steps 182 to 192 is shown as a representative example. Other step orders may be implemented to meet the criteria of a particular application.

Referring to FIG. 7, a diagram of an example mapping of data in the devices 96a-96n into the multiple slices 162a-162n of multiple codewords CWA-CWN is shown. Returning to FIG. 6, a redundancy block is read from the circuit 94 in the step 182. The redundancy block that is read from a given address in the circuit 94 is the same as the redundancy block previously written to the given address, apart from errors. The redundancy block is parsed by the circuit 100 in the step 184 to regenerate the pages A-N. The read pages are the same as the write pages of the previously written redundancy block.

In various embodiments, the pages A-N are parsed by the circuit 100 in the step 186 to regenerate the slices 162a-162n. The slices 162a-162n are de-interleaved and assembled in the step 188 to regenerated the batches A-N. In some situations where the pages were not interleaved prior to writing, the de-interleaving step may be skipped. The batches A-N are parsed by the circuit 100 to regenerate the codewords CWA-CWN in the step 190. Each codeword CWA-CWN is decoded in the step 192 to recover the data units. The data units are presented in the signal DATA. The data units are subsequently transferred to the circuit 92 in the signal HOSTIO and/or to the source internal to the circuit 100 that initiated the read.

Following the write example involving the k-th column, reading the k-th flash plane returns a physical page that contains m virtual codewords to the circuit 100. Each virtual codeword has n slices. The i-th virtual codeword and j-th slice map to the i-th column and j-th row of a 2-dimensional read codeword buffer inside the circuit 100. Data units in the read codeword buffer can be transferred in the signal DATA to a host computer (via the signal HOSTIO) or other destination (e.g., internal to the circuit 100) that requested the read data.

The circuit 100 is generally useful for applications in which the dominant access patterns are sequential reads and/or accesses for a large data size. For example, accessing video files and audio file generally access large data sizes that are sensitive to the latency variation of different accesses. The circuit 100 is also useful for random read intensive applications and other random applications.

In some implementations, common hardware designs may be unaltered and the interleaving is enabled or disabled by software/firmware. For example, embodiments of the invention can be combined with current controller designs. Where the program/erase cycle count is low, the current designs are used to access the circuit 94. As the program/erase cycle count increases, the error rates in different die/planes generally increase. The program/erase cycle lifetime and maximum retention time of the different die/planes may also become different. For example, the codewords stored in the worst plane/die commonly fail to be corrected after many (e.g., 10,000) program/erase cycles. In a common design, a worst die is marked as unusable when that die fails. Thus, the program/erase cycle lifetime of the common flash-based solid-state drive with certain guaranteed capacity and without the interleaving is just 10,000 program/erase cycles. By enabling the interleaving, the errors in the codewords are the average of multiple die/planes. Thus, the codewords remain correctable even after 10,000 program/erase cycles. The solid-state drive remains usable for more that the 10,000 program/erase cycles without a loss in the capacity.

The interleaving can be implemented orthogonally to many other techniques, such as adaptive error correction coding over program/erase cycles. If the adaptive error correction coding is applied, the code rate is initially adapted over the program/erase cycles. As the program/erase cycles increase, the flash controller switches gradually to stronger error correction coding (e.g., lower coding rate error correction codes) until eventually the strongest error correction coding is applied (e.g., a lowest coding rate error correction code). As the program/erase cycles continue to increase, the lowest coding rate error correction code on the weakest flash block/die/plane eventually fails.

By enabling the interleaving, the weakest die may be still remain usable, and thus the program/erase cycle lifetime is extended without a capacity loss.

Various implementations of the invention cause the performance of the circuit 102 to be more predicable compared with conventional techniques. The interleaving reduces the cases where more errors exist within some codewords than other codewords stored in different die/planes. Distributing each codeword across multiple physical flash planes avoids a bottleneck caused by the worst flash plane. The reliability and lifetime are thus improved.

The functions performed by the diagrams of FIGS. 1-7 may be implemented using one or more of a conventional general purpose processor, digital computer, microprocessor, microcontroller, RISC (reduced instruction set computer) processor, CISC (complex instruction set computer) processor, SIMD (single instruction multiple data) processor, signal processor, central processing unit (CPU), arithmetic logic unit (ALU), video digital signal processor (VDSP) and/or similar computational machines, programmed according to the teachings of the specification, as will be apparent to those skilled in the relevant art(s). Appropriate software, firmware, coding, routines, instructions, opcodes, microcode, and/or program modules may readily be prepared by skilled programmers based on the teachings of the disclosure, as will also be apparent to those skilled in the relevant art(s). The software is generally executed from a medium or several media by one or more of the processors of the machine implementation.

The invention may also be implemented by the preparation of ASICs (application specific integrated circuits), Platform ASICs, FPGAs (field programmable gate arrays), PLDs (programmable logic devices), CPLDs (complex programmable logic devices), sea-of-gates, RFICs (radio frequency integrated circuits), ASSPs (application specific standard products), one or more monolithic integrated circuits, one or more chips or die arranged as flip-chip modules and/or multi-chip modules or by interconnecting an appropriate network of conventional component circuits, as is described herein, modifications of which will be readily apparent to those skilled in the art(s).

The invention thus may also include a computer product which may be a storage medium or media and/or a transmission medium or media including instructions which may be used to program a machine to perform one or more processes or methods in accordance with the invention. Execution of instructions contained in the computer product by the machine, along with operations of surrounding circuitry, may transform input data into one or more files on the storage medium and/or one or more output signals representative of a physical object or substance, such as

an audio and/or visual depiction. The storage medium may include, but is not limited to, any type of disk including floppy disk, hard drive, magnetic disk, optical disk, CD-ROM, DVD and magneto-optical disks and circuits such as ROMs (read-only memories), RAMs (random access memories), EPROMs (erasable programmable ROMs), EEPROMs (electrically erasable programmable ROMs), UVPROM (ultra-violet erasable programmable ROMs), Flash memory, magnetic cards, optical cards, and/or any type of media suitable for storing electronic instructions.

The elements of the invention may form part or all of one or more devices, units, components, systems, machines and/or apparatuses. The devices may include, but are not limited to, servers, workstations, storage array controllers, storage systems, personal computers, laptop computers, notebook computers, palm computers, personal digital assistants, portable electronic devices, battery powered devices, set-top boxes, encoders, decoders, transcoders, compressors, decompressors, pre-processors, post-processors, transmitters, receivers, transceivers, cipher circuits, cellular telephones, digital cameras, positioning and/or navigation systems, medical equipment, heads-up displays, wireless devices, audio recording, audio storage and/or audio playback devices, video recording, video storage and/or video playback devices, game platforms, peripherals and/or multi-chip modules. Those skilled in the relevant art(s) would understand that the elements of the invention may be implemented in other types of devices to meet the criteria of a particular application.

The terms "may" and "generally" when used herein in conjunction with "is(are)" and verbs are meant to communicate the intention that the description is exemplary and believed to be broad enough to encompass both the specific examples presented in the disclosure as well as alternative examples that could be derived based on the disclosure. The terms "may" and "generally" as used herein should not be construed to necessarily imply the desirability or possibility of omitting a corresponding element.

While the invention has been particularly shown and described with reference to embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made without departing from the scope of the invention.

CLAIMS

1. An apparatus comprising:

an interface to a plurality of memories each having a plurality of planes, wherein said memories are nonvolatile; and

a circuit configured to (i) generate a plurality of codewords by encoding a plurality of data units, (ii) generate a plurality of slices by parsing said codewords, (iii) generate a plurality of pages by interleaving said slices and (iv) write said pages in parallel into respective ones of said planes.

2. The apparatus according to claim 1, wherein said memories are flash memories.

3. The apparatus according to claim 1, wherein (i) said circuit is further configured to generate a plurality of batches each containing a plurality of said codewords and (ii) a size of each of said batches substantially matches one of said planes.

4. The apparatus according to claim 3, wherein one or more of said pages contain said slices from two or more of said batches.

5. The apparatus according to claim 1, wherein said circuit is further configured to (i) regenerate said pages by reading said respective ones of said planes in parallel from said memories, (ii) regenerate said slices by parsing said pages, (iii) regenerate said codewords from said slices and (iv) regenerate said data units by decoding said codewords.

6. The apparatus according to claim 5, wherein (i) said circuit is further configured to generate a plurality of batches each containing a plurality of said codewords and (ii) one or more of said batches contains said slices from two or more of said pages.

7. The apparatus according to claim 6, wherein said codewords are regenerated by parsing said batches.

8. The apparatus according to claim 1, wherein each of said pages is written to said respective one of said planes in a different one of said memories.

9. The apparatus according to claim 1, wherein said apparatus is implemented as one or more integrated circuits.

10. A method for interleaving codewords over multiple planes, comprising the steps of:

(A) generating a plurality of said codewords by encoding a plurality of data units;

(B) generating a plurality of slices by parsing said codewords;

(C) generating a plurality of pages by interleaving said slices; and

(D) writing said pages in parallel into respective ones of a plurality of said planes in a plurality of memories, wherein said memories are nonvolatile.

11. The method according to claim 10, wherein said memories are flash memories.

12. The method according to claim 10, further comprising the step of:

generating a plurality of batches each containing a plurality of said codewords, wherein a size of each of said batches substantially matches one of said planes.

13. The method according to claim 12, wherein one or more of said pages contain said slices from two or more of said batches.

14. The method according to claim 10, further comprising the steps of:

regenerating said pages by reading said respective ones of said planes in parallel from said memories;

regenerating said slices by parsing said pages;

regenerating said codewords from said slices; and

regenerating said data units by decoding said codewords.

15. The method according to claim 14, further comprising the step of:

generating a plurality of batches each containing a plurality of said codewords, wherein one or more of said batches contains said slices from two or more of said pages.

16. The method according to claim 15, wherein said codewords are regenerated by parsing said batches.

17. The method according to claim 10, wherein each of said pages is written to said respective one of said planes in a different one of said memories.

18. An apparatus comprising:

means for generating a plurality of codewords by encoding a plurality of data units;

means for generating a plurality of slices by parsing said codewords;

means for generating a plurality of pages by interleaving said slices; and

means for writing said pages in parallel into respective ones of a plurality of planes in a plurality of memories, wherein said memories are nonvolatile.

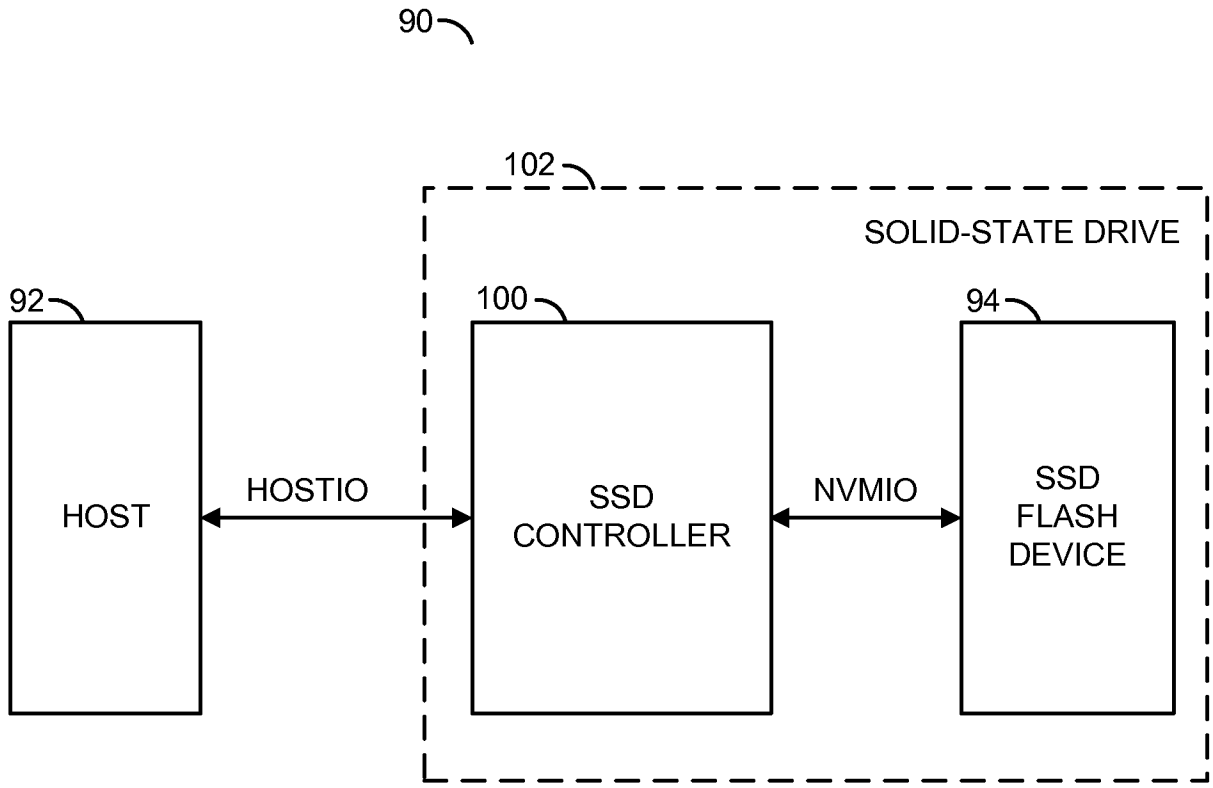

90
102
SOLID-STATE DRIVE
92
HOST
HOSTIO
100
SSD
CONTROLLER
NVMIO
94
SSD
FLASH
DEVICE

FIG. 1

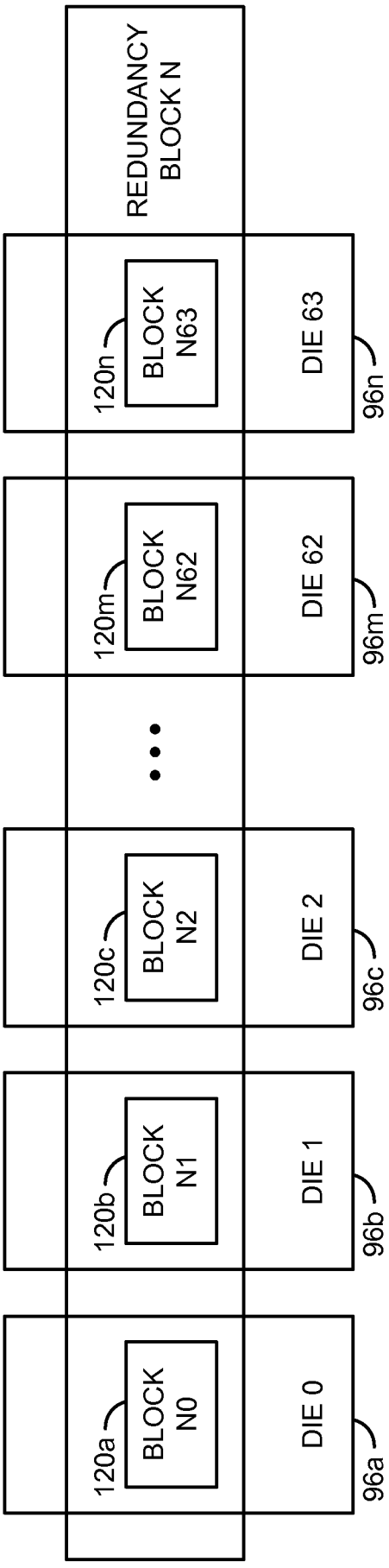

REDUNDANCY BLOCK N
120n
BLOCK N63
DIE 63
96n
120m
BLOCK N62
DIE 62
96m
120c
BLOCK N2
DIE 2
96c
120b
BLOCK N1
DIE 1
96b
120a
BLOCK N0
DIE 0
96a

FIG. 2

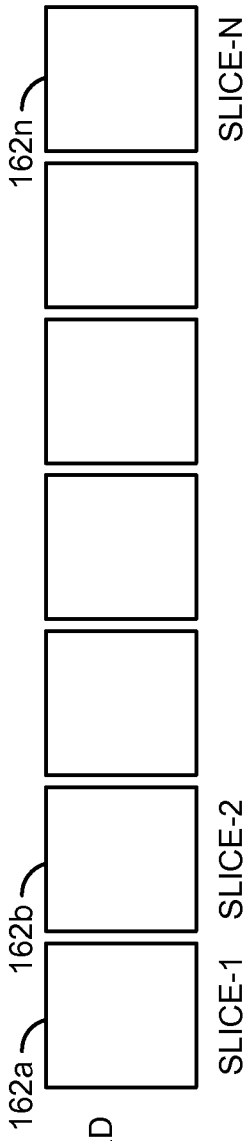

162n
SLICE-N
162b
SLICE-2
162a
SLICE-1
CODEWORD
160

FIG. 4

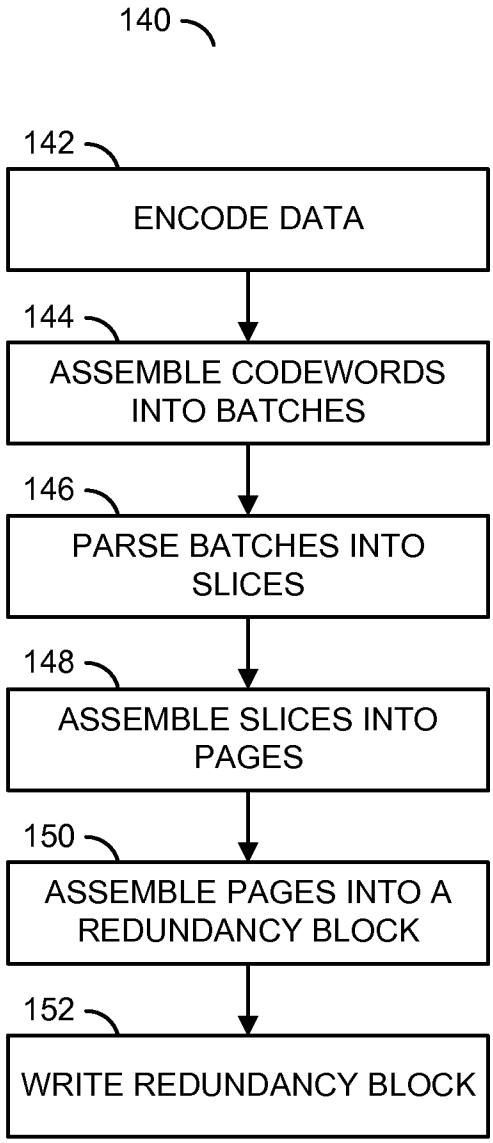

140
142
ENCODE DATA
144
ASSEMBLE CODEWORDS INTO BATCHES
146
PARSE BATCHES INTO SLICES
148
ASSEMBLE SLICES INTO PAGES
150
ASSEMBLE PAGES INTO A REDUNDANCY BLOCK
152
WRITE REDUNDANCY BLOCK

FIG. 3

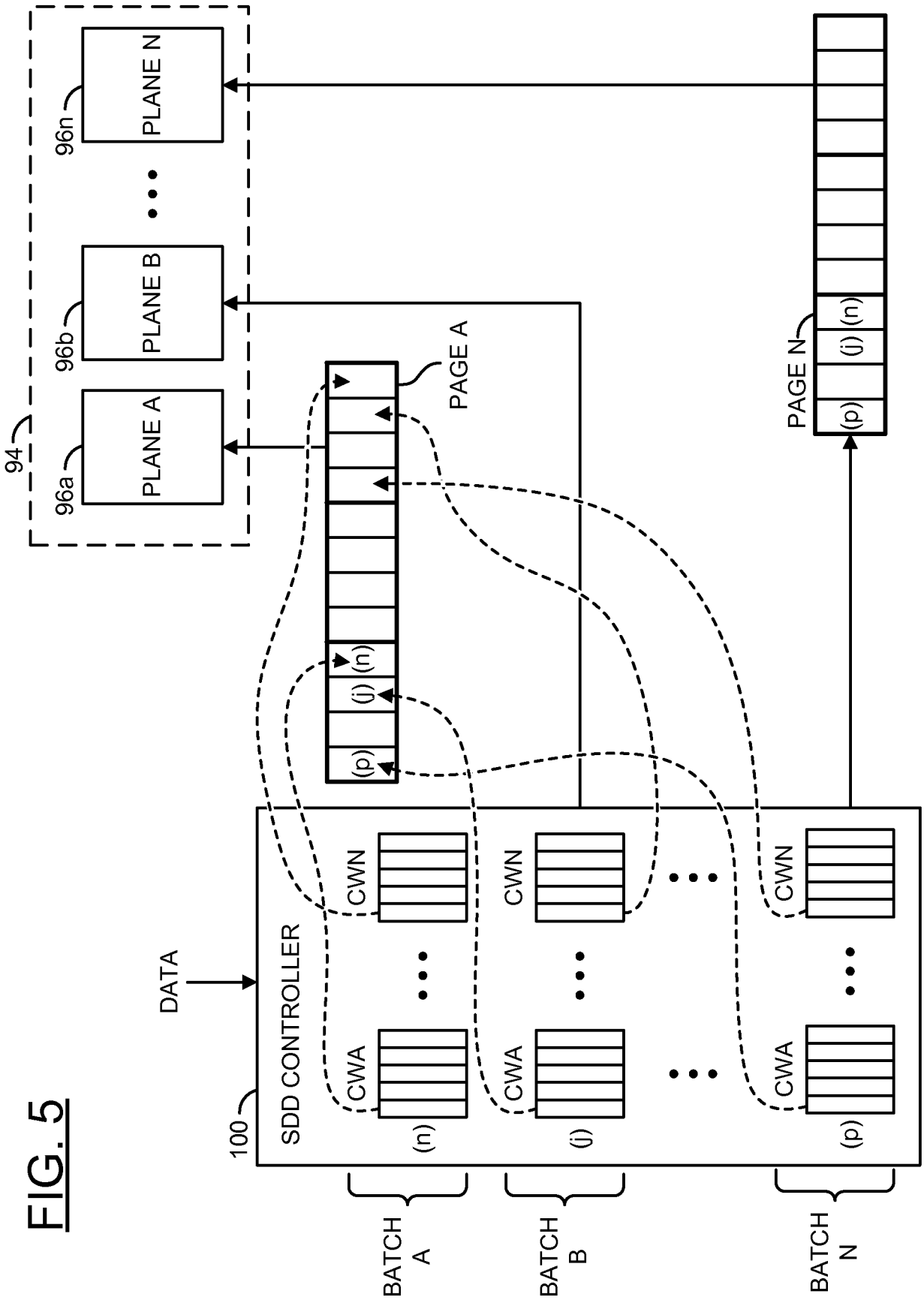

94
96a
PLANE A
96b
PLANE B
96n
PLANE N
PAGE A
(p)
(j)
(n)
PAGE N
(p)
(j)
(n)
DATA
100
SDD CONTROLLER
CWA
CWN
(n)
(j)
(p)
BATCH A
BATCH B
BATCH N

FIG. 5

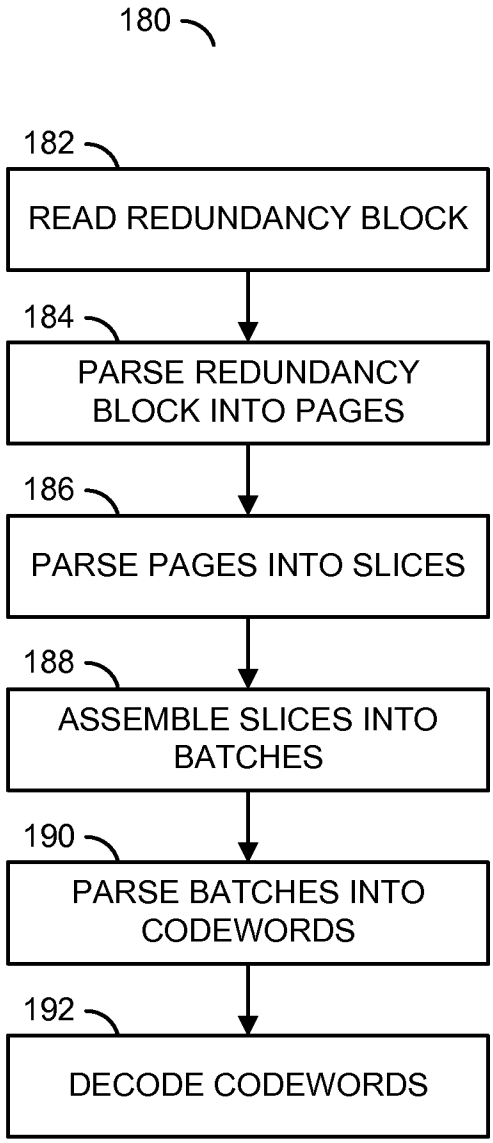

180
182
READ REDUNDANCY BLOCK
184
PARSE REDUNDANCY BLOCK INTO PAGES
186
PARSE PAGES INTO SLICES
188
ASSEMBLE SLICES INTO BATCHES
190
PARSE BATCHES INTO CODEWORDS
192
DECODE CODEWORDS

FIG. 6

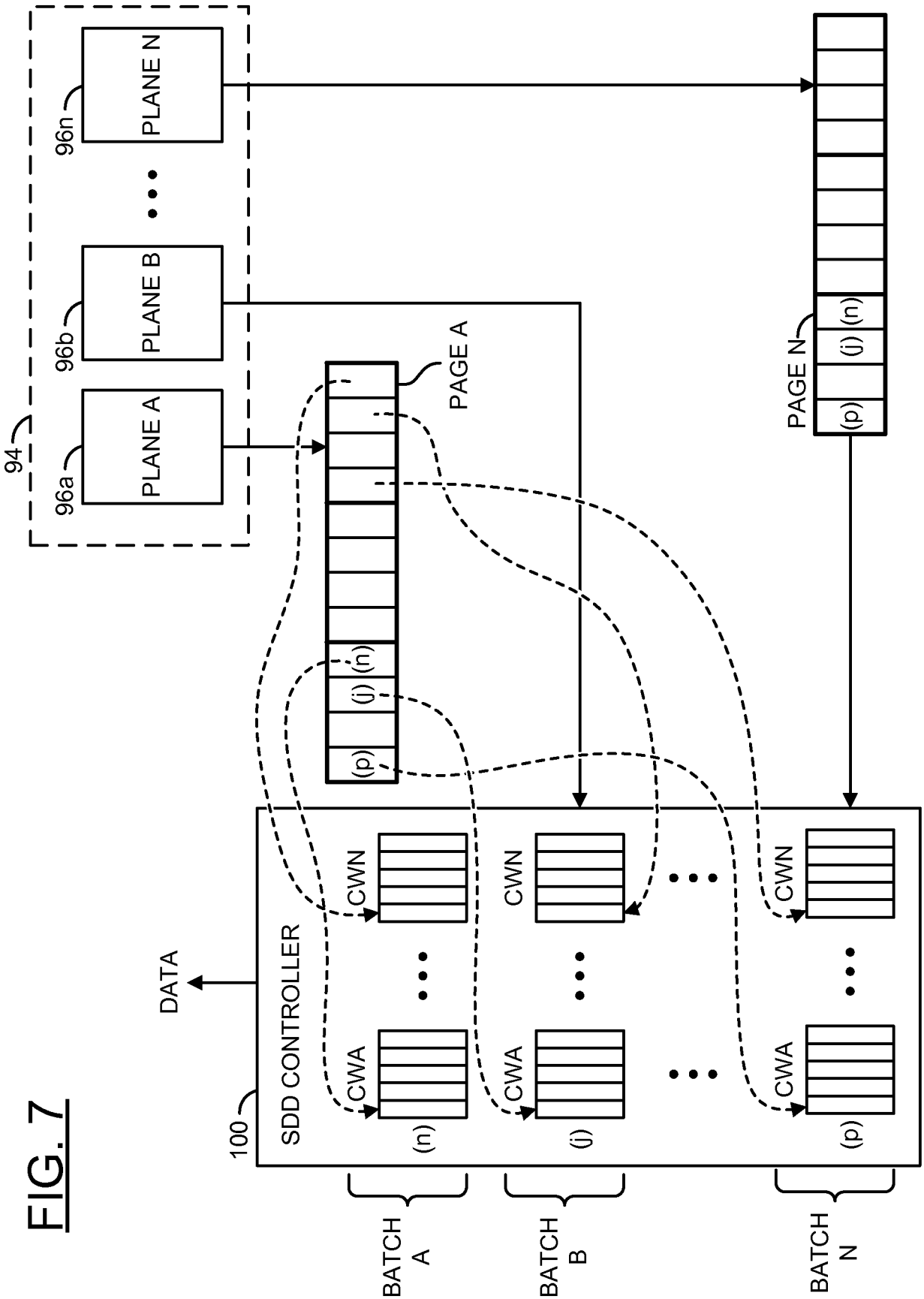

94
96a
PLANE A
96b
PLANE B
96n
PLANE N
PAGE A
(p)
(j)
(n)
PAGE N
100
SDD CONTROLLER
DATA
CWA
CWN
(n)
(j)
(p)
BATCH A
BATCH B
BATCH N

FIG. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/011189

A. CLASSIFICATION OF SUBJECT MATTER

G11C 16/26(2006.01)i, G11C 16/34(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)
G11C 16/26; G06F 12/02; G06F 11/10; G06F 12/00; G06F 11/08; H03M 13/29; G06F 11/00; G11C 16/34

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: memory, nonvolatile, plane, codeword, slice, parsing, page, interleaving, batch, flash

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2012-0144093 A1 (AVI STEINER et al.) 07 June 2012 See paragraphs [0032], [0083], [0149], [0178], [0191], [0193], [0207]; and figures 1, 10, 12, 14, 16.	1-2,5,8-11,14,17-18
A		3-4,6-7,12-13,15-16
Y	US 2010-0211833 A1 (HANAN WEINGARTEN) 19 August 2010 See paragraph [0023]; and figure 14.	1-2,5,8-11,14,17-18
A	US 2011-0119562 A1 (AVI STEINER et al.) 19 May 2011 See paragraph [0034]; and figure 4.	1-18
A	US 2012-0311406 A1 (SAMPATH K. RATNAM et al.) 06 December 2012 See paragraph [0054]; and figure 4.	1-18
A	US 2014-0006898 A1 (ERAN SHARON et al.) 02 January 2014 See paragraph [0026]; and figure 5.	1-18

☐ 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	Date of mailing of the international search report
18 March 2015 (18.03.2015)	**20 March 2015 (20.03.2015)**

Name and mailing address of the ISA/KR
International Application Division
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
BYUN, Sung Cheal
Telephone No. +82-42-481-8262

Form PCT/ISA/210 (second sheet) (January 2015)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/011189

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012-0144093 A1	07/06/2012	US 8850100 B2	30/09/2014
US 2010-0211833 A1	19/08/2010	US 2010-0058146 A1	04/03/2010
		US 2010-0064096 A1	11/03/2010
		US 2010-0088557 A1	08/04/2010
		US 2010-0131809 A1	27/05/2010
		US 2010-0131831 A1	27/05/2010
		US 2010-0146191 A1	10/06/2010
		US 2010-0146192 A1	10/06/2010
		US 2010-0180073 A1	15/07/2010
		US 2010-0199149 A1	05/08/2010
		US 2010-0211856 A1	19/08/2010
		US 8276051 B2	25/09/2012
		US 8321625 B2	27/11/2012
		US 8335977 B2	18/12/2012
		US 8341335 B2	25/12/2012
		US 8359516 B2	22/01/2013
		US 8427867 B2	23/04/2013
		US 8443242 B2	14/05/2013
		US 8453022 B2	28/05/2013
		US 8607128 B2	10/12/2013
		US 8694715 B2	08/04/2014
		US 8751726 B2	10/06/2014
		WO 2009-053961 A2	30/04/2009
		WO 2009-053961 A3	04/03/2010
		WO 2009-053962 A2	30/04/2009
		WO 2009-053962 A3	04/03/2010
		WO 2009-053963 A2	30/04/2009
		WO 2009-053963 A3	04/03/2010
		WO 2009-072100 A2	11/06/2009
		WO 2009-072100 A3	04/03/2010
		WO 2009-072101 A2	11/06/2009
		WO 2009-072101 A3	04/03/2010
		WO 2009-072102 A2	11/06/2009
		WO 2009-072102 A3	04/03/2010
		WO 2009-072103 A2	11/06/2009
		WO 2009-072103 A3	04/03/2010
		WO 2009-072104 A2	11/06/2009
		WO 2009-072104 A3	04/03/2010
		WO 2009-072104 A8	11/06/2009
		WO 2009-072105 A2	11/06/2009
		WO 2009-072105 A3	04/03/2010
		WO 2009-074978 A2	18/06/2009
		WO 2009-074978 A3	04/03/2010
		WO 2009-074979 A2	18/06/2009
		WO 2009-074979 A3	04/03/2010
US 2011-0119562 A1	19/05/2011	US 8626988 B2	07/01/2014

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/011189

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012-0311406 A1	06/12/2012	TW 201303595 A	16/01/2013
		US 2014-325316 A1	30/10/2014
		US 8732557 B2	20/05/2014
		WO 2012-166723 A2	06/12/2012
		WO 2012-166723 A3	28/02/2013
US 2014-0006898 A1	02/01/2014	US 8910017 B2	09/12/2014