



(51) International Patent Classification:

G11B 20/10 (2006.01)

G06F 11/07 (2006.01)

G06F 3/06 (2006.01)

(21) International Application Number:

PCT/US2019/020607

(22) International Filing Date:

04 March 2019 (04.03.2019)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

16/020,853

27 June 2018 (27.06.2018)

US

(71) Applicant: **WESTERN DIGITAL TECHNOLOGIES, INC.** [US/US]; 5601 Great Oaks Parkway, San Jose, CA 95119 (US).

(72) Inventors: **LAMBERTS, Bernd**; 5601 Great Oaks Parkway, San Jose, California 95119 (US). **PIT, Remmelt**; 5601 Great Oaks Parkway, San Jose, California 95119 (US).

LIEBER, Timothy; 5601 Great Oaks Parkway, San Jose, California 95119 (US). **LESTER, Robert M.**; 5601 Great Oaks Parkway, San Jose, California 95119 (US). **POPOV, Zarko**; 5601 Great Oaks Parkway, San Jose, California 95119 (US).

(74) Agent: **HENKHAUS, John, D.**; 7052 Santa Teresa Blvd. #203, San Jose, CA 95139 (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, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, 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.

(54) Title: SYSTEM-LEVEL ERROR CORRECTION CODING ALLOCATION BASED ON DEVICE POPULATION DATA INTEGRITY SHARING

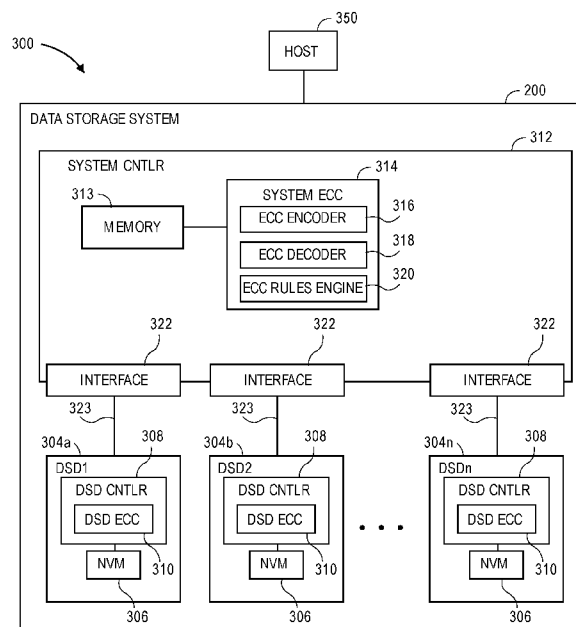


FIG. 3

(57) Abstract: A dynamic scalable error correction coding (ECC) scheme for a data storage system involves a system controller predicting a type and/or amount of ECC needed to reconstruct data to be stored on a particular data storage device(s) based on operational data integrity information accessed from the array of data storage devices. Thus, redundancy does not need to be allocated unless required. The devices may be logically grouped into subsets according to common characteristics, whereby the prediction made for a device in a subset may be based on the data integrity information from that subset, as well as from other relevant subsets.



(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).

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))*

SYSTEM-LEVEL ERROR CORRECTION CODING ALLOCATION BASED ON DEVICE POPULATION DATA INTEGRITY SHARING

FIELD OF EMBODIMENTS

- 5 **[0001]** Embodiments of the invention may relate generally to data storage systems and, more particularly, to error correction coding (ECC) based on data integrity information from the system storage devices.

BACKGROUND

- 10 **[0002]** There is a commercial demand for high-capacity digital data storage systems, in which multiple data storage devices (DSDs) are housed in a common enclosure. Data storage systems often include large enclosures that house multiple shelves on which rows of DSDs are mounted. Historically, storage systems have strictly autonomous independently designed error redundancy schemes. This leads to overlap and reduced efficiency in the overall system. For example, the device may allocate parity to protect
15 any error from any component for any data, regardless of whether this is required or whether the host might need those data. Thus, this error correction coding on the device may increase the failure footprint (i.e., the smallest error event), the latency invoked, and the cost of the system.
- 20 **[0003]** Any approaches described in this section are approaches that could be pursued, but not necessarily approaches that have been previously conceived or pursued. Therefore, unless otherwise indicated, it should not be assumed that any of the approaches described in this section qualify as prior art merely by virtue of their inclusion in this section.

25

BRIEF DESCRIPTION OF THE DRAWINGS

- [0004]** Embodiments are illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:
- 30 **[0005]** FIG. 1A is a plan view illustrating a hard disk drive (HDD), according to an embodiment;
- [0006]** FIG. 1B is a block diagram illustrating a solid-state drive (SSD), according to an embodiment;

[0007] FIG. 2 is a perspective view illustrating a data storage system, according to an embodiment;

[0008] FIG. 3 is a block diagram illustrating a data storage system architecture, according to an embodiment; and

5 [0009] FIG. 4 is a flow diagram illustrating a method for allocating system-level error correction coding (ECC) based on device data integrity information, according to an embodiment.

DETAILED DESCRIPTION

10 [0010] Approaches to error correction coding (ECC) based on data integrity information from the system storage devices are described. In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the embodiments of the invention described herein. It will be apparent, however, that the embodiments of the invention described herein may be
15 practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the embodiments of the invention described herein.

DATA STORAGE SYSTEM

[0011] There is a commercial demand for high-capacity digital data storage systems, in which multiple data storage devices (DSDs) are housed in a common enclosure. Data
20 storage systems often include large enclosures that house multiple shelves on which rows of DSDs are mounted. FIG. 2 is a perspective view illustrating a data storage system, according to embodiments. A data storage system 200 comprises a system enclosure 202 (or “rack 202”), in which multiple data storage system trays 204 are housed. Each tray
25 204 may be placed or slid into a corresponding slot within the rack 202. Rack 202 further houses a system controller, and may further house switches, storage server(s), application server(s), a power supply, cooling fans, etc.

[0012] FIG. 3 is a block diagram illustrating a data storage system architecture, according to an embodiment. Example architecture 300 illustrates a data storage system
30 200 that comprises multiple data storage devices (DSDs) 304a (DSD1), 304b (DSD2), and 304n (DSDn), where n represents an arbitrary number of DSDs that may vary from implementation to implementation. Each DSD 304a-304n is under the control of and communicative with a system controller 312, via a communication interface 322

according to a corresponding communication protocol 323. Each DSD 304a, 304b, 304n includes corresponding non-volatile memory (NVM) 306 controlled by a respective DSD controller 308, which includes a DSD ECC module 310. Each DSD ECC module 310 (which may be, for example, constituent to a corresponding system on chip (SoC), such as part of the data channel or the hard disk controller in the context of an HDD) is capable of performing “track ECC” processing, i.e., ECC encoding/decoding at the data track level, using procedures such as those known in the art. System controller 312 includes memory 313, a system ECC module 314, which includes an ECC encoder 316, an ECC decoder 318, and an ECC rules engine 320, each of which represents procedural functionality involving ECC parity encoding and decoding, as described in more detail elsewhere herein.

[0013] Processing, functions, procedures, actions, method steps, and the like, that are described herein as being performed or performable by system controller 312, or by a DSD 304a-304n, may include enactment by execution of one or more sequences of instructions stored in one or more memory units and which, when executed by one or more processors, cause such performance. System controller 312 may be embodied in any form of and/or combination of hardware, software, and firmware. For example, and according to an embodiment, the system controller 312 comprises an application-specific integrated circuit (ASIC) comprising at least one memory unit for storing such instructions (such as firmware, for a non-limiting example) and at least one processor for executing such instructions, enabling ECC parity allocation, encoding, decoding, and application to recover corrupt stored data by transforming corrupt or lost data to its original form as write requested.

[0014] The data storage system 200 may be communicatively coupled with a host 350, which may be embodied in a hardware machine on which executable code is executable (for non-limiting examples, a computer or hardware server, and the like), or as software instructions executable by one or more processors (for non-limiting examples, a software server such as a database server, application server, media server, and the like). Host 350 generally represents a client of the data storage system 200, and has the capability to make read and write requests to the data storage system 200. Note that the system controller 312 may also be referred to as a “host” because the term is often generally used in reference to any device that makes I/O calls to a data storage device or an array of devices.

INTRODUCTION

[0015] Is used herein, the term “substantially” will be understood to describe a feature that is largely or nearly structured, configured, dimensioned, etc., but with which manufacturing tolerances and the like may in practice result in a situation in which the structure, configuration, dimension, etc. is not always or necessarily precisely as stated. For example, describing a structure as “substantially vertical” would assign that term its plain meaning, such that the sidewall is vertical for all practical purposes but may not be precisely at 90 degrees.

[0016] Error correction coding (ECC) (or “error correction code”, “error-correcting code”, etc.) generally refers to schemes, techniques, procedures, encoding/decoding, and the like, for correcting (i.e., reconstructing, or “recovering”) corrupt data. Corrupt data may be the result of incorrect transmission of data, or unwanted bit-flips (bit errors) that occur after data is stored in (i.e., written) to non-volatile memory, or unwanted bit erasures that occur after data is stored to non-volatile memory, such as NVM 306 of DSDs 304a, 304b, 304n (FIG. 3). ECC is typically implemented by adding or appending extra encoded data (at times generally referred to as “parity”, “parity data”, “parity information”, “parity bits”, and the like) to associated data, so that the data can be recovered even when a number of errors (up to the capability of the parity being used) are introduced into the data.

[0017] As an example, a “write verify” function may be utilized in certain data storage contexts to ensure data reliability (so-called “data integrity”), but can negatively affect system performance by decreasing write command throughput. Thus, maintaining data integrity has its own trade-offs and challenges, and one may consider committing resources to enhanced ECC schemes to recover corrupt data over committing resources to data integrity at the storage device. Stated otherwise, one may be willing to accept less data integrity at the storage device, e.g., at the non-volatile memory 306 of the DSD 304a-304n (FIG. 3), in order to increase performance, while relying further on recovery of corrupt data via more independent, intelligent, and scenario-specific ECC schemes.

INTELLIGENT SYSTEM-LEVEL ERROR CORRECTION CODING

[0018] As discussed, historical storage systems have strictly autonomous independently designed error redundancy schemes, which can lead to overlap and reduced efficiency in the overall system. By contrast, an expanded holistic view of the

data integrity at and among different storage devices as distributed throughout the storage system may lead to a more efficient and more reliable overall storage system.

[0019] The data storage devices, such as DSDs 304a-304n (FIG. 3) that are constituent to a data storage system 200 (FIGS. 2 and 3) can be considered sensitive sensors of factors that may affect the data integrity associated with each DSD, such as temperature, airflow, vibration, and the like in the context of HDDs, and age, storage cycles, and the like in the context of SSDs. Furthermore, DSDs 304a-304n in the form of HDDs may be sources of information regarding position error signals (PES) and the factors contributing thereto, corresponding to a specific head or region of media. Hence, data integrity information that each DSD may be internally privy to, and combinations thereof, when shared with a system controller such as controller 312 (FIG. 3), can provide an inexpensive but valuable expanded vision of the symptoms of data degradation at the device level, for use as an informed and efficient ECC controlling point at the system level. There may also be additional information about the DSDs available at the system level, e.g., available to the system controller, that can contribute to and/or be combined to contribute to intelligent predictive ECC decision-making at the system level. For example, in the context of HDDs, the system controller may be made privy to information such as the storage device form factor, the number of disks (and/or actuators), the disks' RPM (revolution per minute), and other common variations among HDD models, and possibly as well as the location of each storage device within a particular shelf and within the system rack (e.g., rack 202 of FIG. 2), relative proximity to fans, fan speed, relative proximity to heat sources, and other factors external to each device itself. Note that the DSDs 304a-304n, as well as the controller 312, may be programmed to "learn" over time. Thus, factors that may affect the data integrity associated with each DSD may not be static, i.e., such factors (and ECC decisions based thereon) may be dynamic.

[0020] FIG. 4 is a flow diagram illustrating a method for allocating system-level error correction coding (ECC) based on device data integrity information, according to an embodiment.

[0021] At block 402, a plurality of data storage devices constituent to a data storage system provide access to respective data integrity information to system controller circuitry. For example and according to an embodiment, the DSDs 304a-304n (FIG. 3) of data storage system 200 (FIGS. 2, 3) may initiate a messaging protocol (e.g., which may be existing or which may be specifically tailored to this purpose and overlaid onto an

existing or other communication protocol) to the system controller 312 (FIG. 3) (or the system controller 312 may operate to initiate the messaging protocol), e.g., via a common communications channel or bus and/or a dedicated communications channel or bus, by which the DSDs 304a-304n (or at least a subset of the DSDs 304a-304n) transmit or
5 report respective data integrity, data fitness, data risk information to the controller 312 via the respective communication interface 322 (FIG. 3). Alternatively, for example and according to an embodiment, the data integrity, data fitness, data risk information may be stored on each corresponding DSD 304a-304n, or on any other of the non-corresponding DSDs 304a-304n, or elsewhere, from which the controller 312 may access/read such data
10 integrity information.

[0022] Generally, the data integrity, fitness, risk information characterizes the probability of or risk of a read error (e.g., corrupt data due to bit flips, bit erasures, and the like) occurring when the respective DSD performs a read operation, according to an embodiment. Stated otherwise, each DSD 304a-304n may provide to the controller 312 a
15 set of operational metrics that measure its respective read signal integrity. Furthermore, the system controller 312 may at times choose to change the metrics that it utilizes in ECC decision-making based on, for non-limiting examples, the significance of the corresponding stored data, the cost associated with replacing or rebuilding the stored data from other sources, and the like.

[0023] For one non-limiting example, each or any DSD 304a-304n may provide raw signal integrity corresponding to an HDD sector, such as in the form of a local iteration counter, which is a signal quality metric regarding how difficult or processing-intensive it is for the read channel to decode bits. For another non-limiting example, each or any DSD 304a-304n may provide data integrity, fitness, risk information regarding adjacent
25 track interference (ATI) and/or far track interference (FTI) and/or track squeeze, and the effects thereof (e.g., amplitude change, erase band width, byte error rate), to name a few. Other data integrity metrics sharable by the DSDs 304a-304n to the controller 312 for this purpose may be, for non-limiting examples, per head or per zone signal to noise ratio (SNR) margin, information about write inhibit feed forward and intelligent write verify
30 operations in the context of shingled magnetic recording (SMR) HDDs, HDD disk load/unload zones, and the like.

[0024] Generally, the types of data integrity, fitness, risk information that the DSDs 304a-304n may provide to the controller 312 are relatively unbounded, within the context

of local knowledge that the DSD has about itself, and with the goal of providing information to enable the controller 312 to perform relevant risk assessment and prediction with respect to the quality of a read signal generated within the DSD. This is not to say the information provided by the DSDs 304a-304n to the controller 312 is
5 arbitrary, rather the overall system design, development, and implementation of the techniques described herein would benefit from an integrated approach to what types of information is available to each DSD and valuable to the system controller for assessment and predictive purposes.

[0025] At block 404, the controller circuitry determines, based on the data integrity
10 information from at least a subset of the plurality of data storage devices, a type and/or amount of error correction coding (ECC) corresponding to data to be stored on one or more of the data storage devices. For example, based on the data integrity, fitness, risk information that the population of DSDs 304a-304n has shared with the controller 312 at block 402 (or at least a subset of such shared information), the ECC rules engine 320
15 (FIG. 3) of the controller 312 evaluates such information to predict the probability or risk of error corresponding to a read operation of a given DSD 304a-304n. Then, based on the results of its evaluation (which may be further based on information other than the information collected at block 402), the controller 312 can make an informed decision as to what type and/or amount, level, degree of ECC (for non-limiting examples, low density
20 parity check (LDPC) code at the lowest level, block level code over N logical block addresses (LBAs) at the next level, erasure coding implemented as distributed code over multiple DSDs at the next level, a RAID scheme, and the like) is suitable for handling the associated risk of data corruption, and the controller 312 can then track for which data additional parity is allocated. For example, supposing a baseline parity of 1x bytes of
25 encoded parity for data on a particular DSD, if the data integrity, fitness, risk information shared by that DSD leads to a conclusion or prediction by the controller 312 that the data on that particular DSD is compromised or aging (e.g., by way of thermal decay, squeeze, etc.), then the rules engine 320 of the controller 312 may determine that 2x bytes or even 4x bytes, for example, of encoded parity is suitable for new data written to that particular
30 DSD.

[0026] According to an embodiment, the plurality of DSDs 304a-304n are logically grouped into subsets according to one or more common characteristics (e.g., vibration environment, age, drive type, etc.), and the determining performed at block 404 for a

particular subset of DSDs 304a-304n or for a particular DSD 304a-304n within the subset is based on the shared data integrity information from that particular subset.

Alternatively, according to an embodiment, the determining performed at block 404 for a particular subset of DSDs 304a-304n or for a particular DSD 304a-304n within the subset is based on the shared data integrity information from that particular subset and from one or more additional subsets of DSDs 304a-304n.

[0027] The system controller 312 likely has knowledge that is not available to the DSDs 304a-304n, which it may utilize to make the determination at block 404. For a non-limiting example, controller 312 may have knowledge of the time criticality of certain data, such as based on information in a write request from the host 350 (FIG. 3), based on which the controller 312 determines desired ECC. For example, data for which read latency is more undesirable may in turn be allocated maximum ECC protection. For another non-limiting example, controller 312 has knowledge as to whether redundancy (e.g., ECC parity) for certain data already exists in a different context. For example, the controller 312 may know that erasure coding protection is already available on a larger scale, such as in an archive context, and that read latency is not paramount, based on which the controller 312 foregoes additional ECC protection because the data is already covered by this other form of ECC protection. According to an embodiment, the processing associated with block 404 is performed as a background process, continuously or as processing bandwidth is available. Alternatively, according to an embodiment, the processing associated with block 404 is performed in response to a write request from the host 350, whereby the write request may be queued for writing out to the storage device array while such processing is being performed.

[0028] At block 406, the controller 312 computes the ECC determined at block 404, and at block 408, data is stored on (written to) and the corresponding computed ECC is stored on (written to) one or more of the DSDs 304a-304n. For example and according to an embodiment, the ECC may be stored on the corresponding DSD 304a-304n on which the corresponding data is stored. Alternatively, for example and according to an embodiment, the data may be stored on one DSD 304a-304n while the corresponding ECC is stored on a different DSD 304a-304n from which the data itself is stored (or elsewhere accessible to the controller 312). According to an embodiment, at block 408 the ECC is stored on a virtual track, i.e., a track logically associated with a certain number of HDD sectors of a larger sector grouping. For example, the ECC may be stored

on a track separate from the track on which the corresponding data is stored, and for every 100 sectors of a 500-sector group, for a non-limiting example. Thus, an entire track need not be read in order to get to the parity, which is typically stored at the end of the track on which the corresponding data is stored.

5 **[0029]** Thus, in response to a data read request from a client, such as host 350, the controller 312 makes a read request to the one or more DSDs 304a-304n on which the data is stored, the one or more DSDs 304a-304n return to the controller 312 the requested data and the corresponding ECC, and the controller 312 decodes the ECC in order to reconstruct or perfect the requested data. According to an embodiment, the DSDs 304a-
10 304n return some form of the requested data independent of whether the read operation was successful. That is, in the event of a partial failed read operation, the DSDs 304a-304n may return “dummy” data for the failed sectors and the controller 312 detects and corrects the data errors based on the available ECC. Furthermore, redundancy or parity that is used often may be cached by the controller 312 (e.g., in memory 313) and/or the
15 DSDs 304a-304n to improve its availability.

[0030] The dynamic and scalable approach to error correction coding (ECC) based on data integrity information from the system storage devices as described herein provides for a more reliable data storage system and, therefore a more available data storage system over its intended or specified or advertised or warranted operational life cycle,
20 than with conventional approaches. This is at least in part because the system controller 312 gains a holistic view of the data signal integrity of the array of data storage devices 304a-304n in view of the informative and practical data integrity feedback that the population of devices 304a-304n reports to or otherwise shares with the controller 312. Hence, a more optimal schema of ECC is employed within the storage system, e.g., only
25 a necessary amount of ECC need be generated in scenarios in which the data integrity is strong and as much ECC as needed can be generated in scenarios in which the data integrity is compromised, and read failures are inhibited. Stated otherwise, the type and amount of ECC more closely maps to the amount of data redundancy actually needed because the type and amount of ECC needed is determined/predicted based on “real”
30 operational knowledge from and about the device array. Thus, this flexible and adaptable approach to ECC further provides for a lower computational cost and more efficient ECC framework in which coding is only generated when and where the need arises. Furthermore, determining at the system level when and where and how much coding is

needed provides a better performing data storage system because the ECC computing and processing has a minimal effect on device I/O operations.

PHYSICAL DESCRIPTION OF AN ILLUSTRATIVE OPERATING CONTEXT

HARD DISK DRIVE CONFIGURATION

5 **[0031]** Embodiments may be used in the context of a data storage system in which multiple data storage devices (DSDs) such as hard disk drives (HDDs), as well as solid state memory devices (or “solid state drives” (SSDs)) and/or hybrid drives, are employed. Thus, in accordance with an embodiment, a plan view illustrating an HDD 100 is shown in FIG. 1A to illustrate an exemplary operating context.

10 **[0032]** FIG. 1A illustrates the functional arrangement of components of the HDD 100 including a slider 110b that includes a magnetic read-write head 110a. Collectively, slider 110b and head 110a may be referred to as a head slider. The HDD 100 includes at least one head gimbal assembly (HGA) 110 including the head slider, a lead suspension 110c attached to the head slider typically via a flexure, and a load beam 110d attached to the lead suspension 110c. The HDD 100 also includes at least one recording medium 120 rotatably mounted on a spindle 124 and a drive motor (not visible) attached to the spindle 124 for rotating the medium 120. The read-write head 110a, which may also be referred to as a transducer, includes a write element and a read element for respectively writing and reading information stored on the medium 120 of the HDD 100. The medium 120 or
20 a plurality of disk media may be affixed to the spindle 124 with a disk clamp 128.

[0033] The HDD 100 further includes an arm 132 attached to the HGA 110, a carriage 134, a voice-coil motor (VCM) that includes an armature 136 including a voice coil 140 attached to the carriage 134 and a stator 144 including a voice-coil magnet (not visible). The armature 136 of the VCM is attached to the carriage 134 and is configured
25 to move the arm 132 and the HGA 110 to access portions of the medium 120, all collectively mounted on a pivot shaft 148 with an interposed pivot bearing assembly 152. In the case of an HDD having multiple disks, the carriage 134 may be referred to as an “E-block,” or comb, because the carriage is arranged to carry a ganged array of arms that gives it the appearance of a comb.

30 **[0034]** An assembly comprising a head gimbal assembly (e.g., HGA 110) including a flexure to which the head slider is coupled, an actuator arm (e.g., arm 132) and/or load beam to which the flexure is coupled, and an actuator (e.g., the VCM) to which the actuator arm is coupled, may be collectively referred to as a head stack assembly (HSA).

An HSA may, however, include more or fewer components than those described. For example, an HSA may refer to an assembly that further includes electrical interconnection components. Generally, an HSA is the assembly configured to move the head slider to access portions of the medium 120 for read and write operations.

5 **[0035]** With further reference to FIG. 1A, electrical signals (e.g., current to the voice coil 140 of the VCM) comprising a write signal to and a read signal from the head 110a, are transmitted by a flexible cable assembly (FCA) 156 (or “flex cable”). Interconnection between the flex cable 156 and the head 110a may include an arm-electronics (AE) module 160, which may have an on-board pre-amplifier for the read signal, as well as
10 other read-channel and write-channel electronic components. The AE module 160 may be attached to the carriage 134 as shown. The flex cable 156 may be coupled to an electrical-connector block 164, which provides electrical communication, in some configurations, through an electrical feed-through provided by an HDD housing 168. The HDD housing 168 (or “enclosure base” or “baseplate” or simply “base”), in conjunction with an HDD
15 cover, provides a semi-sealed (or hermetically sealed, in some configurations) protective enclosure for the information storage components of the HDD 100.

[0036] Other electronic components, including a disk controller and servo electronics including a digital-signal processor (DSP), provide electrical signals to the drive motor, the voice coil 140 of the VCM and the head 110a of the HGA 110. The electrical signal
20 provided to the drive motor enables the drive motor to spin providing a torque to the spindle 124 which is in turn transmitted to the medium 120 that is affixed to the spindle 124. As a result, the medium 120 spins in a direction 172. The spinning medium 120 creates a cushion of air that acts as an air-bearing on which the air-bearing surface (ABS) of the slider 110b rides so that the slider 110b flies above the surface of the medium 120
25 without making contact with a thin magnetic-recording layer in which information is recorded. Similarly in an HDD in which a lighter-than-air gas is utilized, such as helium for a non-limiting example, the spinning medium 120 creates a cushion of gas that acts as a gas or fluid bearing on which the slider 110b rides.

[0037] The electrical signal provided to the voice coil 140 of the VCM enables the
30 head 110a of the HGA 110 to access a track 176 on which information is recorded. Thus, the armature 136 of the VCM swings through an arc 180, which enables the head 110a of the HGA 110 to access various tracks on the medium 120. Information is stored on the medium 120 in a plurality of radially nested tracks arranged in sectors on the medium

120, such as sector 184. Correspondingly, each track is composed of a plurality of sectored track portions (or “track sector”) such as sectored track portion 188. Each sectored track portion 188 may include recorded information, and a header containing error correction code information and a servo-burst-signal pattern, such as an ABCD-servo-burst-signal pattern, which is information that identifies the track 176. In accessing the track 176, the read element of the head 110a of the HGA 110 reads the servo-burst-signal pattern, which provides a position-error-signal (PES) to the servo electronics, which controls the electrical signal provided to the voice coil 140 of the VCM, thereby enabling the head 110a to follow the track 176. Upon finding the track 176 and identifying a particular sectored track portion 188, the head 110a either reads information from the track 176 or writes information to the track 176 depending on instructions received by the disk controller from an external agent, for example, a microprocessor of a computer system.

[0038] An HDD’s electronic architecture comprises numerous electronic components for performing their respective functions for operation of an HDD, such as a hard disk controller (“HDC”), an interface controller, an arm electronics module, a data channel, a motor driver, a servo processor, buffer memory, etc. Two or more of such components may be combined on a single integrated circuit board referred to as a “system on a chip” (“SOC”). Several, if not all, of such electronic components are typically arranged on a printed circuit board that is coupled to the bottom side of an HDD, such as to HDD housing 168.

SOLID STATE DRIVE CONFIGURATION

[0039] FIG. 1B is a block diagram illustrating an example operating context with which embodiments of the invention may be implemented. FIG. 1B illustrates a generic SSD architecture 150, with an SSD 152 communicatively coupled with a host 154 through a primary communication interface 156. Embodiments are not limited to a configuration as depicted in FIG. 1B, rather, embodiments may be implemented with SSD configurations other than that illustrated in FIG. 1B. For example, embodiments may be implemented to operate in other environments that rely on non-volatile memory storage components for writing and reading of data.

[0040] Host 154 broadly represents any type of computing hardware, software, or firmware (or any combination of the foregoing) that makes, among others, data I/O requests or calls to one or more memory device. For example, host 154 may be an

operating system executing on a computer, a tablet, a mobile phone, or generally any type of computing device that contains or interacts with memory, such as host 350 (FIG. 3).

The primary interface 156 coupling host 154 to SSD 152 may be, for example, a storage system's internal bus or a communication cable or a wireless communication link, or the

5 like.

[0041] The example SSD 152 illustrated in FIG. 1B includes an interface 160, a controller 162 (e.g., a controller having firmware logic therein), an addressing 164 function block, data buffer cache 166, error correction code (ECC) 168, and one or more non-volatile memory components 170a, 170b-170n.

10 **[0042]** Interface 160 is a point of interaction between components, namely SSD 152 and host 154 in this context, and is applicable at the level of both hardware and software. This allows a component to communicate with other components via an input/output system and an associated protocol. A hardware interface is typically described by the mechanical, electrical and logical signals at the interface and the protocol for sequencing
15 them. Some non-limiting examples of common and standard interfaces include SCSI (Small Computer System Interface), SAS (Serial Attached SCSI), and SATA (Serial ATA).

[0043] An SSD 152 includes a controller 162, which incorporates the electronics that bridge the non-volatile memory components (e.g., NAND flash) to the host, such as non-
20 volatile memory 170a, 170b, 170n to host 154. The controller is typically an embedded processor that executes firmware-level code and is an important factor in SSD performance.

[0044] Controller 162 interfaces with non-volatile memory 170a, 170b, 170n via an addressing 164 function block. The addressing 164 function operates, for example, to
25 manage mappings between logical block addresses (LBAs) from the host 154 to a corresponding physical block address on the SSD 152, namely, on the non-volatile memory 170a, 170b, 170n of SSD 152. Because the non-volatile memory page and the host sectors are different sizes, an SSD has to build and maintain a data structure that enables it to translate between the host writing data to or reading data from a sector, and
30 the physical non-volatile memory page on which that data is actually placed. This table structure or "mapping" may be built and maintained for a session in the SSD's volatile memory 172, such as DRAM or some other local volatile memory component accessible to controller 162 and addressing 164. Alternatively, the table structure may be

maintained more persistently across sessions in the SSD's non-volatile memory such as non-volatile memory 170a, 170b-170n.

[0045] Addressing 164 interacts with data buffer cache 166 and error correction code (ECC) 168, in addition to non-volatile memory 170a, 170b-170n. Data buffer cache 166 of an SSD 152 typically uses DRAM as a cache, similar to the cache in hard disk drives. Data buffer cache 166 serves as a buffer or staging area for the transmission of data to and from the non-volatile memory components, as well as serves as a cache for speeding up future requests for the cached data. Data buffer cache 166 is typically implemented with volatile memory so the data stored therein is not permanently stored in the cache, i.e., the data is not persistent.

[0046] ECC 168 is a system of adding redundant data, or parity data, to a message, such that it can be recovered by a receiver even when a number of errors were introduced, either during the process of transmission, or on storage.

[0047] Finally, SSD 152 includes one or more non-volatile memory 170a, 170b-170n components. For a non-limiting example, the non-volatile memory components 170a, 170b-170n may be implemented as flash memory (e.g., NAND or NOR flash), or other types of solid-state memory available now or in the future. The non-volatile memory 170a, 170b-170n components are the actual memory electronic components on which data is persistently stored. The non-volatile memory 170a, 170b-170n components of SSD 152 can be considered the analogue to the hard disks in hard-disk drive (HDD) storage devices.

[0048] Furthermore, references herein to a data storage device may encompass a multi-medium storage device (or "multi-medium device", which may at times be referred to as a "multi-tier device" or "hybrid drive"). A multi-medium storage device refers generally to a storage device having functionality of both a traditional HDD (see, e.g., HDD 100) combined with an SSD (see, e.g., SSD 150) using non-volatile memory, such as flash or other solid-state (e.g., integrated circuits) memory, which is electrically erasable and programmable. As operation, management and control of the different types of storage media typically differ, the solid-state portion of a hybrid drive may include its own corresponding controller functionality, which may be integrated into a single controller along with the HDD functionality. A multi-medium storage device may be architected and configured to operate and to utilize the solid-state portion in a number of ways, such as, for non-limiting examples, by using the solid-state memory as cache

memory, for storing frequently-accessed data, for storing I/O intensive data, for storing metadata corresponding to payload data (e.g., for assisting with decoding the payload data), and the like. Further, a multi-medium storage device may be architected and configured essentially as two storage devices in a single enclosure, i.e., a traditional HDD
5 and an SSD, with either one or multiple interfaces for host connection.

EXTENSIONS AND ALTERNATIVES

[0049] In the foregoing description, embodiments of the invention have been described with reference to numerous specific details that may vary from implementation to implementation. Therefore, various modifications and changes may be made thereto
10 without departing from the broader spirit and scope of the embodiments. Thus, the sole and exclusive indicator of what is the invention, and is intended by the applicants to be the invention, is the set of claims that issue from this application, in the specific form in which such claims issue, including any subsequent correction. Any definitions expressly set forth herein for terms contained in such claims shall govern the meaning of such terms
15 as used in the claims. Hence, no limitation, element, property, feature, advantage or attribute that is not expressly recited in a claim should limit the scope of such claim in any way. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

[0050] In addition, in this description certain process steps may be set forth in a
20 particular order, and alphabetic and alphanumeric labels may be used to identify certain steps. Unless specifically stated in the description, embodiments are not necessarily limited to any particular order of carrying out such steps. In particular, the labels are used merely for convenient identification of steps, and are not intended to specify or require a particular order of carrying out such steps.

25

CLAIMS

What is claimed is:

1. A method comprising:
a plurality of data storage devices constituent to a data storage system providing access to respective data integrity information to a controller circuitry of the data storage system;
the controller circuitry determining, based at least in part on the data integrity information from at least a subset of the plurality of data storage devices, a type and/or amount of error correction coding (ECC) corresponding to data to be stored on one or more of the plurality of data storage devices to inhibit read operation failures;
the controller circuitry computing the determined ECC; and
storing data and storing the corresponding computed ECC on one or more of the plurality of data storage devices.
2. The method of Claim 1, wherein the determining is performed in response to a write request from a client.
3. The method of Claim 1, wherein the determining is performed as a background process.
4. The method of Claim 1, further comprising:
in response to a data read request from a client, the controller circuitry requesting a read operation from one or more of the data storage devices on which the requested data is stored;
the one or more data storage devices returning to the controller circuitry the requested data and the corresponding ECC; and
the controller circuitry decoding the ECC.
5. The method of Claim 1, wherein:
the plurality of data storage devices are logically grouped into subsets according to one or more common characteristics; and

the determining for a particular subset of data storage devices is based on the data integrity information from that particular subset.

6. The method of Claim 1, wherein:
the plurality of data storage devices are logically grouped into subsets according to one or more common characteristics; and
the determining for a particular subset of data storage devices is based on the data integrity information from that particular subset and from one or more additional subsets of data storage devices.
7. The method of Claim 1, wherein the data integrity information from a data storage device characterizes signal integrity corresponding to that data storage device.
8. The method of Claim 7, wherein the data integrity information from a data storage device comprises a local iteration counter corresponding to at least a portion of a storage medium of that data storage device.
9. The method of Claim 7, wherein the data integrity information from a data storage device comprises a metric characterizing adjacent track interference corresponding to at least a portion of a storage medium of that data storage device.
10. The method of Claim 1, wherein the determining is further based on time criticality of the data to be stored on the one or more data storage devices.
11. The method of Claim 1, wherein the determining is further based on availability of other error correction coding corresponding to the data to be stored on the one or more data storage devices.
12. The method of Claim 1, wherein the determining is further based on ensuring operational availability of the data storage system during its specified life cycle.
13. The method of Claim 1, wherein:

the storing of the ECC is stored on an ECC track logically associated with a certain number of hard disk drive data sectors of a larger data sector grouping; and

the ECC track is a different track than a track on which corresponding data is stored.

14. The method of Claim 1, wherein the providing access to the respective data integrity information includes one or more of the plurality of data storage devices reporting respective data integrity information to the controller circuitry.
15. The method of Claim 1, wherein the providing access to the respective data integrity information includes the controller circuitry reading data integrity information stored on one or more of the plurality of data storage devices.
16. The method of Claim 1, wherein the storing includes storing the computed ECC on the data storage device on which the corresponding data is stored.
17. The method of Claim 1, wherein the storing includes storing the data on a first data storage device and storing the corresponding computed ECC on a different second data storage device.
18. A data storage system comprising:
 - a plurality of data storage devices; and
 - a system controller circuitry comprising memory and one or more processors and embodying one or more sequences of instructions which, when executed by the one or more processors, cause performance of:
 - accessing respective data fitness information corresponding to data stored on at least a subset of the plurality of data storage devices;
 - determining, based at least in part on the accessed data fitness information, a type and/or amount of error correction coding (ECC) corresponding to data to be stored on one or more of the plurality of data storage devices to inhibit read operation failures;
 - computing the determined ECC; and

in response to a data write command from a client, commanding writing data and writing the corresponding computed ECC onto one or more of the plurality of data storage devices.

19. The data storage system of Claim 18, wherein the one or more sequences of instructions cause further performance of:
in response to a data read request from the client, requesting a read operation from one or more of the data storage devices on which the requested data is stored;
receiving from the one or more data storage devices the requested data and the corresponding ECC; and
decoding the ECC.
20. The data storage system of Claim 18, wherein:
the one or more sequences of instructions cause further performance of logically grouping the plurality of data storage devices into subsets based on one or more common characteristics; and
determining for a particular subset of data storage devices is based on the data fitness information from at least that particular subset.
21. The data storage system of Claim 18, wherein the data fitness information from a data storage device characterizes a risk of data read errors corresponding to that data storage device.
22. The data storage system of Claim 18, wherein the plurality of data storage devices comprises at least two types of devices from a group consisting of rotating magnetic storage medium devices, solid state memory devices, and multi-medium devices.
23. The data storage system of Claim 18, wherein the determining is further based on ensuring operational availability of the data storage system during its warranted life cycle.

24. A data storage system electronic controller manufacture embodying one or more sequences of instructions which, when executed by one or more processors, cause performance of:
- accessing respective data risk information corresponding to data stored on at least a subset of a plurality of data storage devices under the control of the controller manufacture; and
 - predicting, based at least in part on the data risk information from at least a subset of the plurality of data storage devices providing access to the data risk information, whether additional error correction coding (ECC) is desired for data to be stored on one or more of the plurality of data storage devices to inhibit read operation failures.
25. The data storage system electronic controller manufacture of Claim 24, wherein the one or more sequences of instructions cause further performance of:
- in response to predicting that additional ECC is desired, computing the additional ECC;
 - in response to a data write command from a system client, commanding writing data and writing the corresponding additional ECC onto one or more of the data storage devices;
 - in response to a data read request from the system client, requesting a read operation from one or more of the data storage devices on which the requested data is stored;
 - receiving from the one or more data storage devices the requested data and the corresponding additional ECC; and
 - decoding the additional ECC to reconstruct the requested data.
26. The data storage system electronic controller manufacture of Claim 24, wherein the one or more sequences of instructions cause further performance of:
- in response to predicting that additional ECC is not desired, and in response to a data read request from a system client, requesting a read operation from one or more of the data storage devices on which the requested data is stored;

receiving from the one or more data storage devices the requested data and ECC already in place for protecting the data; and
decoding the ECC already in place to reconstruct the requested data.

1/5

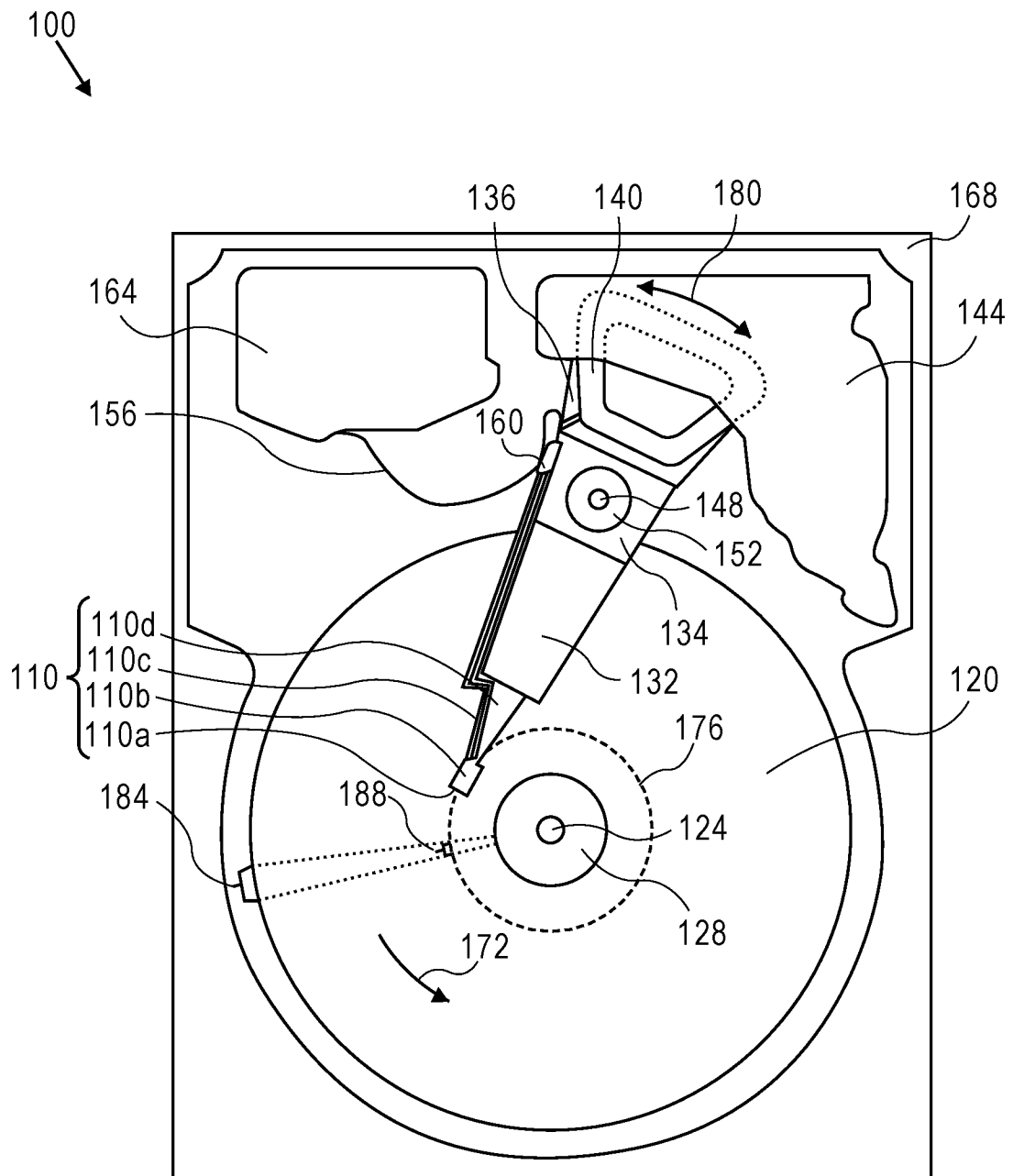
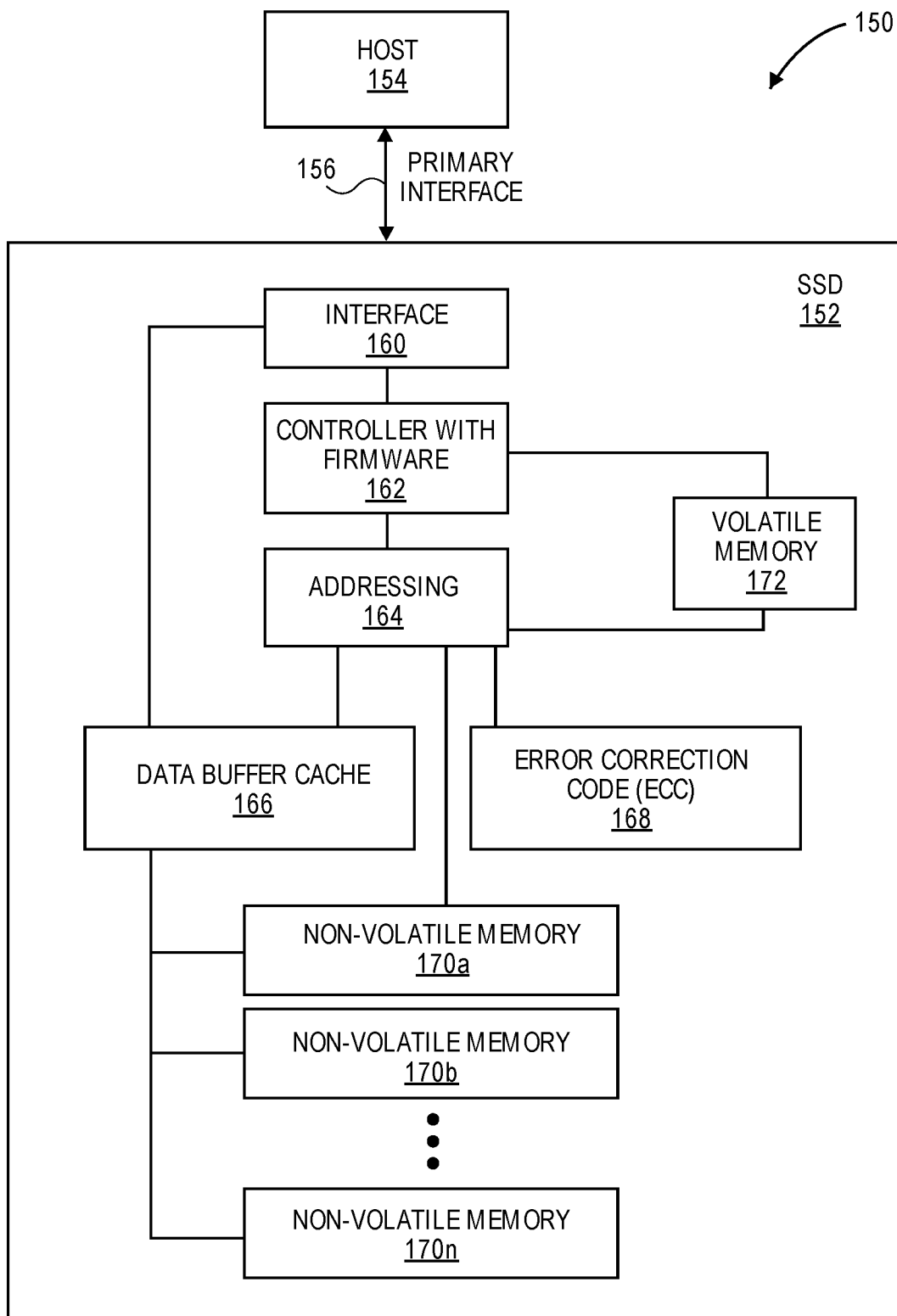


FIG. 1A

2/5

**FIG. 1B**

3/5

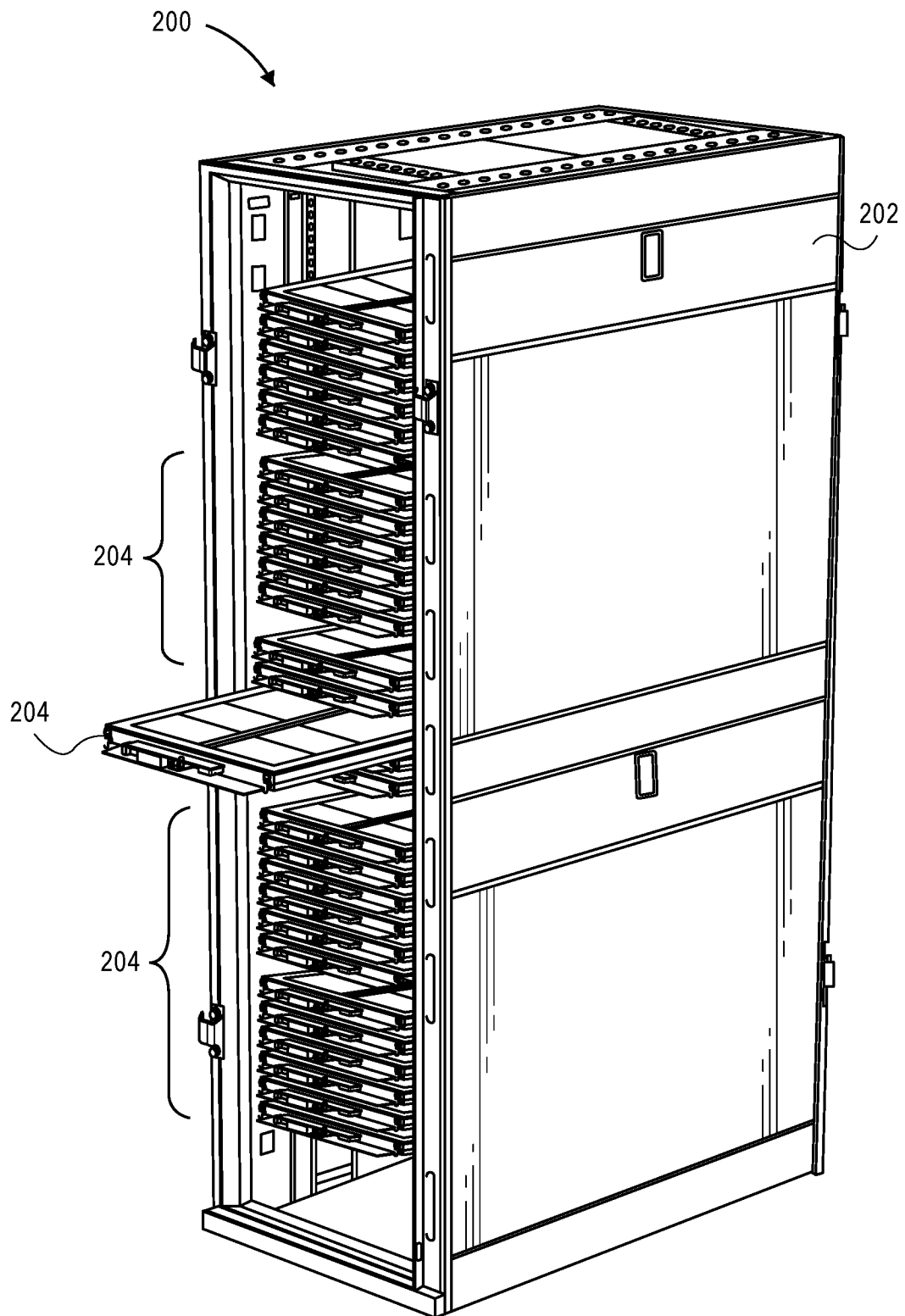


FIG. 2

4/5

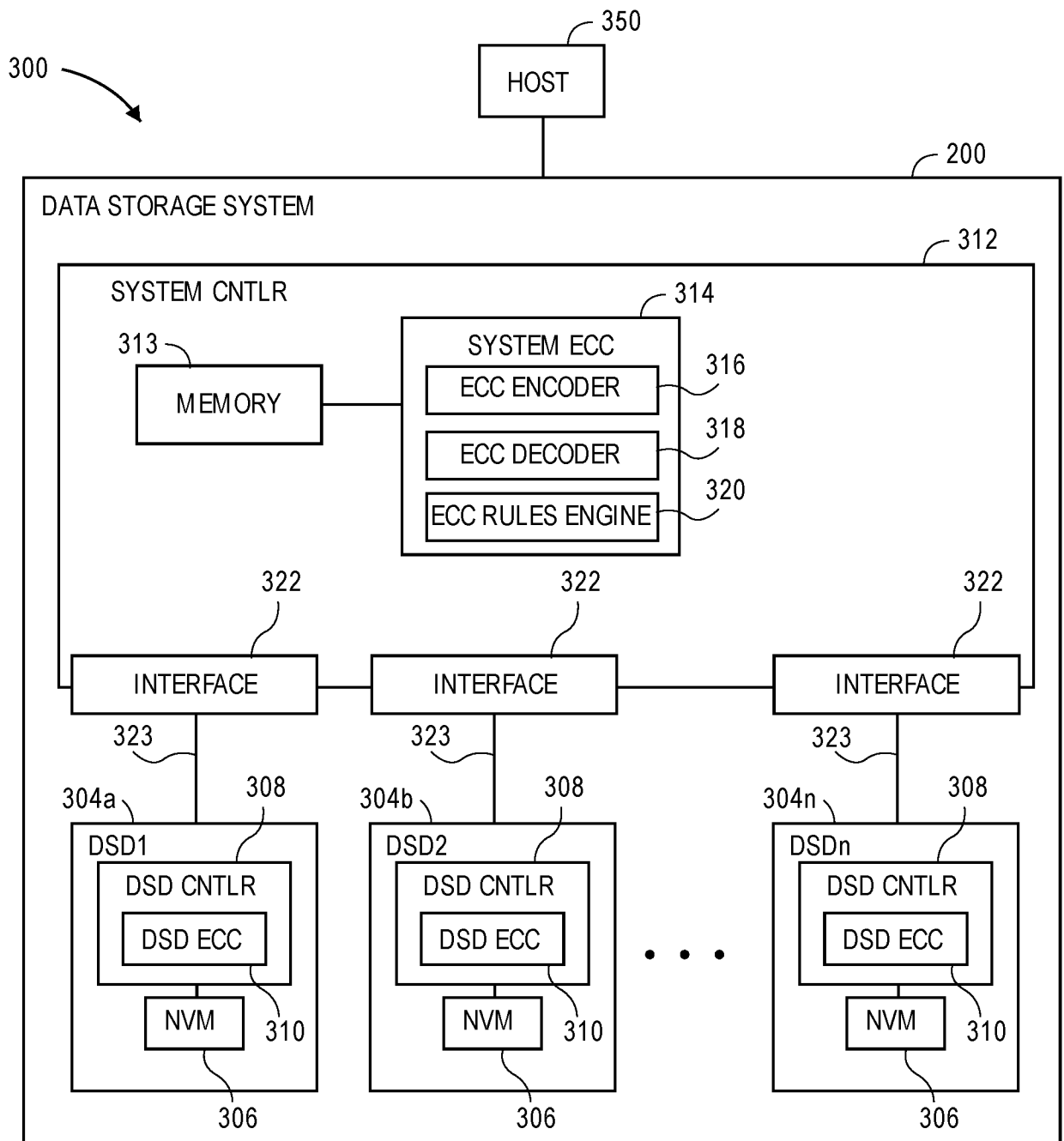
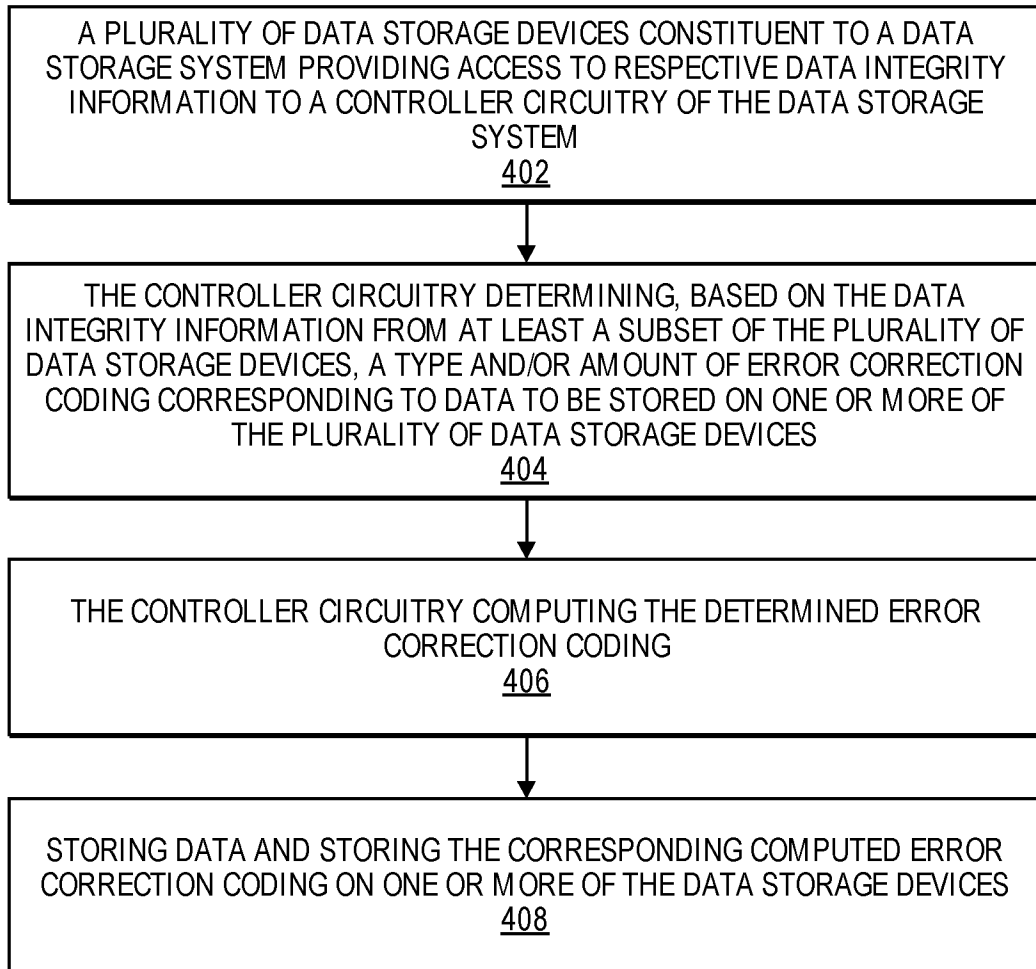


FIG. 3

5/5

**FIG. 4**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2019/020607**A. CLASSIFICATION OF SUBJECT MATTER****G11B 20/10(2006.01)i, G06F 3/06(2006.01)i, G06F 11/07(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)

G11B 20/10; G06F 11/00; G06F 11/08; G06F 11/10; G06F 12/08; G11C 8/02; H03M 13/09; H04L 1/00; G06F 3/06; G06F 11/07

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: data storage system, ECC(error correction coding), type, amount, data integrity information, fitness, risk

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2017-0249242 A1 (LONGITUDE ENTERPRISE FLASH S.A.R.L.) 31 August 2017 See paragraphs [0009], [0027], [0071], [0164], [0168]–[0169], [0181], [0190], [0288]; claim 18; and figures 1, 4A, 6, 17.	1–26
Y	US 7730344 B1 (ANGEL G. PEROZO) 01 June 2010 See column 7, lines 17–26, 57–67; column 9, lines 19–33; claims 8, 16; and figures 1–2, 4.	1–26
A	US 2018-0069658 A1 (WESTERN DIGITAL TECHNOLOGIES, INC.) 08 March 2018 See paragraphs [0036]–[0056]; and figure 1.	1–26
A	US 2012-0124448 A1 (TED EMERSON et al.) 17 May 2012 See paragraphs [0016]–[0027]; and figures 1–2.	1–26
A	US 2011-0047437 A1 (DAVID FLYNN) 24 February 2011 See paragraphs [0058]–[0070]; and figure 2.	1–26



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

20 June 2019 (20.06.2019)

Date of mailing of the international search report

20 June 2019 (20.06.2019)

Name and mailing address of the ISA/KR

International Application Division

Korean Intellectual Property Office

189 Cheongsu-ro, Seo-gu, Daejeon, 35208, Republic of Korea



Facsimile No. +82-42-481-8578

Authorized officer

BYUN, Sung Cheal

Telephone No. +82-42-481-8262



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2019/020607

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2017-0249242 A1	31/08/2017	US 2010-0293440 A1 US 2012-0131413 A1 US 2014-0380130 A1 US 2016-0217033 A1 US 8281227 B2 US 8832528 B2 US 9304853 B2 US 9684560 B2 WO 2010-135370 A2 WO 2010-135370 A3	18/11/2010 24/05/2012 25/12/2014 28/07/2016 02/10/2012 09/09/2014 05/04/2016 20/06/2017 25/11/2010 03/03/2011
US 7730344 B1	01/06/2010	US 7945812 B1	17/05/2011
US 2018-0069658 A1	08/03/2018	US 10055164 B2 US 10069597 B2 US 2018-0067684 A1 US 2018-0067800 A1 WO 2018-048485 A1	21/08/2018 04/09/2018 08/03/2018 08/03/2018 15/03/2018
US 2012-0124448 A1	17/05/2012	TW 201117219 A TW I511150 B US 8738995 B2 WO 2011-031260 A1	16/05/2011 01/12/2015 27/05/2014 17/03/2011
US 2011-0047437 A1	24/02/2011	US 2009-0132760 A1 US 2011-0022801 A1 US 2011-0066808 A1 US 8019938 B2 US 8443134 B2 US 8706968 B2 US 8719501 B2	21/05/2009 27/01/2011 17/03/2011 13/09/2011 14/05/2013 22/04/2014 06/05/2014