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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
1 May 2014 (01.05.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/065967 A1

(51) International Patent Classification:
*G11C 29/42* (2006.01) *G11C 16/34* (2006.01)

(21) International Application Number:
PCT/US2013/061249

(22) International Filing Date:
23 September 2013 (23.09.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13/659,099 24 October 2012 (24.10.2012) US

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

(72) Inventor: **LU, Guangming**; c/o Western Digital Technologies, Inc., 3355 Michelson Drive, Suite 100, Irvine, CA 92612 (US).

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

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

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

**Declarations under Rule 4.17**:

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

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

**Published**:

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

(54) Title: ADAPTIVE ERROR CORRECTION CODES FOR DATA STORAGE SYSTEMS

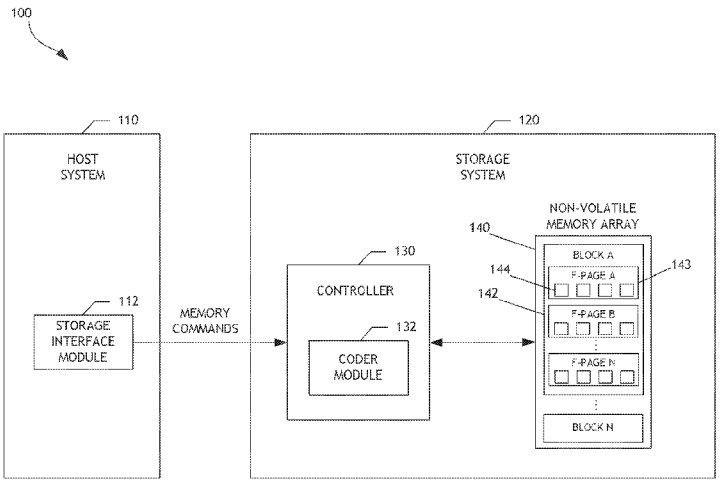



FIGURE 1

(57) Abstract: A data storage system configured to adaptively code data is disclosed, in one embodiment, a data storage system controller determines a common memory page size, such as an E-page size, for a non-volatile memory array. Based on the common memory page size, the controller selects a low-density parity-check (LDPC) code word length from a plurality of pre-defined LDPC code word lengths. The controller determines LDPC coding parameters for coding data written to or read from the memory array based on the selected LDPC code word length. By using the plurality of pre-defined LDPC code word lengths, the data storage system can support multiple non-volatile memory page formats, including memory page formats in which the common memory page size does not equal any LDPC code word length of the plurality of pre-defined LDPC code word lengths. Flexibility and efficiency of data coding can thereby be achieved.

WO 2014/065967 A1

# ADAPTIVE ERROR CORRECTION CODES FOR DATA STORAGE SYSTEMS

## BACKGROUND

### Technical Field

**[0001]** This disclosure relates to data storage systems, such as solid state drives, for computer systems. More particularly, the disclosure relates to adaptive error correction codes for data storage systems.

### Description of the Related Art

**[0002]** Non-volatile memory arrays often have limited endurance. The endurance of the memory array is typically contingent on usage pattern and wear. In addition, endurances depend on a particular type of the non-volatile memory array. For example, memory arrays with multi-level cell (MLC) NAND media typically have a lower endurance than memory arrays with single-level cell (SLC) NAND media. To protect user data stored to memory arrays from corruption, which may be caused by a diminished endurance, parity data can be determined and stored along with user data to facilitate error detection and/or correction.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0003]** Systems and methods that embody the various features of the invention will now be described with reference to the following drawings, in which:

**[0004]** Figure 1 illustrates a storage system that adaptively codes data according to one embodiment of the invention.

**[0005]** Figure 2 is a flow diagram illustrating a process of determining coding parameters for coding data according to one embodiment of the invention.

**[0006]** Figure 3 is a table illustrating relationships between low-density parity-check (LDPC) coding parameters and LDPC code word lengths according to one embodiment of the invention.

**[0007]** Figures 4A-4C are diagrams illustrating padded user data and parity data according to one embodiment of the invention.

**[0008]** Figure 5 is a flow diagram illustrating a process of adaptive code shortening according to one embodiment of the invention.

**[0009]** Figure 6 is a flow diagram illustrating a process of adjusting a code rate according to one embodiment of the invention.

## DETAILED DESCRIPTION OF SPECIFIC EMBODIMENTS

**[0010]** While certain embodiments are described, these embodiments are presented by way of example only, and are not intended to limit the scope of protection. Indeed, the novel methods and systems described herein may be embodied in a variety of other forms. Furthermore, various omissions, substitutions, and changes in the form of the methods and systems described herein may be made without departing from the scope of protection.

**[0011]** In some embodiments, "coding" or "to code" data as used in this disclosure refer to the process of encoding data and/or the process of decoding data.

### Overview

**[0012]** Storage systems such as solid state drives typically include one or more controllers coupled with non-volatile memory arrays. It is commonplace for such controllers to be designed/manufactured by one party and the non-volatile memory arrays to be designed/manufactured by another party. In addition, memory arrays from different manufacturers tend to have different internal formatting, including different memory page formats, where each memory page format corresponds to one of multiple memory pages sizes. Also, for costs and various other competitive reasons, storage system manufacturers typically use memory arrays from different manufacturers. For example, a storage system manufacturer may use one brand of memory array in a current production cycle and another brand in the next production cycle. Also, different memory arrays may be used for different storage system models priced at different levels.

**[0013]** Therefore, one common design approach is to have storage systems include one or more controllers with multiple specialized hardware, firmware, and/or software that encode/decode data read from or written to the memory arrays from

different manufacturers, with a subset of such hardware, firmware, and/or software actually used depending on the actual memory arrays paired with the controller(s) at time of assembly or at the final design stage. As a result, storage systems can include multiple controllers or extra, unused hardware, firmware, and/or software to support multiple memory page formats for the one or more memory arrays that are not included in the finally assembled storage systems.

**[0014]** In the alternative, to reduce the amount of hardware, firmware, and/or software included in storage systems, storage systems and controllers can be constructed to support only one specific, known memory page format having a known memory page size and may not to support other memory page formats. Such an approach, however, limits the usefulness of storage systems and controllers to other types of memory arrays, and multiple different storage systems and controllers may need to be constructed to support multiple memory array types. In addition, in some instances, a memory page format of a memory array may not be known when storage systems and controllers are designed/constructed. Accordingly, improved systems and methods for adaptively coding data for multiple known or unknown memory storage formats are desired.

**[0015]** In some embodiments of the present invention, a storage system includes a controller and a non-volatile memory array having a plurality of memory pages with a common memory page size. The controller determines the common memory page size for the non-volatile memory array and selects a code word length, such as a low-density parity-check (LDPC) code word length, from multiple pre-defined code word lengths based on the common memory page size. By selectively using an appropriate code word length (chosen from multiple pre-defined code word lengths) and corresponding coding parameters, the controller uses at least some common hardware, firmware, and/or software to support multiple known or unknown memory page formats. Thus, the controller can selectively adapt its operations, so it can be paired with memory arrays of different formats made by different manufacturers.

**[0016]** In some embodiments, a controller of a storage system adjusts coding parameters for coding user data as memory pages, blocks, or dies of the non-volatile memory array age and/or wear out due, in part, to use of the memory array. The

mechanism of adjusting coding parameters enables the controller to support additional parity per unit data over time, thereby improving the error correction or detection capabilities as the quality (e.g., data retention capabilities) of the memory array diminishes. Moreover, the mechanism of adjusting coding parameters facilitates a balancing of decoding time due to additional parity per unit data with error correction or detection benefits of additional parity data. In addition, the controller can store adjusted coding parameters for memory pages, blocks, or dies of the memory array to permit different code or parity rates for different memory pages, blocks, or dies.

System Overview

**[0017]** Fig. 1 illustrates a storage system 120 that adaptively codes data according to one embodiment of the invention. As is shown, a storage system 120 (e.g., hybrid hard drive, solid state drive, etc.) includes a controller 130 and a non-volatile memory array 140, which comprises one or more blocks of storage, identified as Block "A" 142 through Block "N". Each block comprises a plurality of flash pages (F-pages). For example, Block A 142 of Figure 1 includes a plurality of F-pages, identified as F-pages A 153, B, through N. In some embodiments, each "F-page" is a smallest grouping of memory cells in the non-volatile memory array 140 that can be programmed in a single operation or as a unit. Further, each F-page includes a plurality of error correcting code pages (E-pages). In the illustrated embodiment, each F-page includes four E-pages that are illustrated as four boxes, including E-page 144. Other embodiments may use F-pages or E-pages that are defined differently or each F-page may include greater or fewer than four E-pages.

**[0018]** The controller 130 can receive data and/or storage access commands from a storage interface module 112 (e.g., a device driver) in a host system 110. Storage access commands communicated by the storage interface 112 can include write and read commands issued by the host system 110. The commands can specify a logical block address in the storage system 120, and the controller 130 can execute the received commands in the non-volatile memory array 140. In a hybrid hard drive, data may be stored in magnetic media storage component (not shown in Fig. 1) in addition to the non-volatile memory array 140.

**[0019]** The storage system 120 can store data received from the host system 110 so that the storage system 120 can act as memory storage for the host system 110. To facilitate this function, the controller 130 can implement a logical interface. The logical interface can present to the host system 110 storage system memory as a set of logical addresses (e.g., contiguous address) where data can be stored. Internally, the controller 130 can map logical addresses to various physical memory addresses in the non-volatile memory array 140 and/or other memory module(s).

**[0020]** The controller 130 includes a coder module 132. In one embodiment, the coder module 132 determines coding parameters for decoding/encoding data (e.g., user data) read from or written to memory pages, such as E-pages, of the non-volatile memory array 140. The coding parameters can be used for decoding user data read from the non-volatile memory array 140, encoding user data for storage to the non-volatile memory array 140, and other uses such as error detection or correction. The coding parameters can include LDPC coding parameters, such as the column weight of a G or H coding matrix, the row weight of a G or H coding matrix, a P matrix size (e.g., where the P matrix is a sub-matrix of a G or H coding matrix), and the like. Further, the coder module 132 can determine parity data for unpadded or padded user data, as well as decode user data having corresponding parity data and padding. In addition, the coder module 132 can adjust a code or parity rate for coding data by adjusting the coding parameters. The controller 130 and/or coder module 132 can further include internal memory (not shown), which may be of one or more suitable memory types.

**[0021]** The non-volatile memory array 140 can be implemented using NAND flash memory devices. Other types of solid-state memory devices can alternatively be used, such as array of flash integrated circuits, Chalcogenide RAM (C-RAM), Phase Change Memory (PC-RAM or PRAM), Programmable Metallization Cell RAM (PMC-RAM or PMCm), Ovonic Unified Memory (OUM), Resistance RAM (RRAM), NOR memory, EEPROM, Ferroelectric Memory (FeRAM), Magnetoresistive RAM (MRAM), other discrete NVM (non-volatile memory) chips, or any combination thereof. In one embodiment, the non-volatile memory array 140 preferably includes multi-level cell (MLC) devices having multi-level cells capable of storing more than a single bit of information, although single-level cell (SLC) memory devices or a combination of SLC

and MLC devices may be used. In one embodiment, the storage system 120 can include other memory modules, such as one or more magnetic memory modules. The storage system 120 can further include other types of storage media, such as magnetic storage.

Adaptive Data Coding

**[0022]** Figure 2 is a flow diagram illustrating a process 200 of determining coding parameters for coding data according to one embodiment of the invention. The process 200 can be executed by the controller 130 and/or the coder module 132. Advantageously, the process 200 can enable the controller 130 and/or the coder module 132 to support multiple non-volatile memory page formats using a plurality of code word lengths.

**[0023]** At block 205, the process 200 determines a memory page size for a non-volatile memory array, such as the non-volatile memory array 140. The memory page size can be, for instance, provided by the memory array vendor or calculated based on another known memory size. For example, the memory page size can correspond to an E-page size of the non-volatile memory array, and the process 200 can calculate the E-page size by looking-up a vendor provided F-page size of the memory array and dividing the F-page size by an appropriate constant for the non-volatile memory array, such as 4 or 8. In other embodiments, the memory page size can correspond to an F-page size.

**[0024]** At block 210, the process 200 selects a code word length that equals or exceeds the memory page size from a plurality of code word lengths. For example, the memory page size can be 2164 bytes or octets, and the plurality of pre-defined code word lengths can include lengths of 2176 bytes and 2304 bytes. In one embodiment, the process 200 selects the LDPC code word length having a minimum size or number of bytes of data equal to or greater than the memory page size from a plurality of pre-defined LDPC code word lengths. For example, the process 200 can select the code word length of 2176 bytes, which exceeds 2164 byte memory page size and has a minimum size of the plurality of pre-defined LDPC code word lengths.

**[0025]** At block 215, the process 200 determines coding parameters based at least in part on the selected code word length. The coding parameters can be used when coding data read from or written to the non-volatile memory array and enable the process 200 to manage a code rate (e.g., an amount of user data per total data of a data unit, where total data includes user data and parity data) for data. In one embodiment, the coding parameters for a LDPC code word length include a column weight, a P matrix size, and a row weight, and further include at least one of a code rate, an amount of user data, and an amount of parity data.

**[0026]** At block 220, the process 200 stores the coding parameters. For example, the process 200 can store the coding parameters in the non-volatile memory array 140 and/or in one or more other storage media of the storage system 120. The process 200 can store the coding parameters in internal memory of the controller 130 and/or the coder module 132. The stored coding parameters can facilitate using different coding parameters for coding different pages, blocks, or other divisions and/or subdivisions of the non-volatile memory array 140 and for tracking and adjusting coding parameters over time as portions of the non-volatile memory array age and/or wear out.

**[0027]** Figure 3 is a table 300 illustrating relationships between LDPC coding parameters and LDPC code word lengths according to one embodiment of the invention. The table 300 can designate the LDPC code word lengths and LDPC coding parameters supported by a controller, such as the controller 130 and/or coder module 132. The table 300 can be stored in the non-volatile memory array 140, one or more other storage media of the storage system 120, and/or in the internal memory of the controller 130 and/or the coder module 132. The table 300 includes two code length columns indicating two supported LDPC code word lengths. One LDPC code word length equals 2176 bytes (2048 + 128·1 bytes), and the other LDPC code word length equals 2304 bytes (2048 + 128·2 bytes). It has been discovered that various code rates around 2 kbytes can provide an optimal trade-off between complexity and performance.

**[0028]** For each code length, the LDPC coding parameters of column weight, P matrix size, and row weight can be varied to enable coding of data using different designed code rates (e.g., different amounts of user data per total data of a data unit, where total data includes user data and parity data) as listed in table 300. For example,

if the LDPC code word length equals 2176 bytes, the applied LDPC coding parameters can be one of the LDPC coding parameter sets at circles C1, C2, C3, or C4. In one instance, the LDPC coding parameters can be chosen at circle C3, which correspond to a column weight of 4, P matrix size of 512, and row weight of 34 and further to a code rate of 0.882 (1920 bytes of user data for 2176 byes of total code length). Further, in one embodiment, the code length, which may be defined as 2048 + 128x$\Delta$, can be adjusted based on selecting $\Delta$ as 1, 2, etc. As is illustrated, selecting the value of $\Delta$ also affects the code rate. For example, at circle C1, selecting $\Delta$ to be 1 results in the code rate of 0.941, and if the column weight, P matrix size, and row weight remain unchanged, selecting $\Delta$ to be 2 results in the code rate of 0.944. As will be further explained, the various pre-defined code rates can be used to accommodate memory arrays of different page sizes. For example, as shown, the two code lengths of 2176 and 2304 bytes can accommodate page sizes of 2176 and 2304 bytes. It is to be noted that in practice the number of pre-defined code lengths can be much higher than two to accommodate various page sizes.

**[0029]** A controller supporting the LDPC coding parameters of table 300 can advantageously select one code length for a non-volatile memory array and adjust the LDPC coding parameters to code data at different code rates. For instance, when a memory array with an E-page size of 2176 bytes is relatively new (e.g., lightly used) and/or experiences or exhibits few coding errors, the LDPC coding parameters at circle C1 can be selected for coding data. The LDPC coding parameters at circle C1 correspond to a column weight of 4, P matrix size of 256, and row weight of 68 and further to a code rate of 0.941. A total of 128 bytes of parity can be used for coding 2048 bytes of data at circle C1. As memory pages, blocks, or dies of the memory array age and/or wear out, the LDPC coding parameters at circles C2, C3, and C4 can instead be selected for coding data. The controller can, accordingly, gradually increase the amount of parity per unit data from a code rate of 0.941 to codes rates of 0.926, 0.882, and 0.853 at circles C2, C3, and C4, respectively, in response to changes in the quality (e.g., loss of quality) of the non-volatile memory array.

Adaptive Code Shortening

**[0030]** Through adaptive code shortening, some embodiments of the invention can accommodate memory arrays with page sizes that do not match up exactly with the pre-defined code lengths. Figures 4A-4C are diagrams illustrating padded user data and parity data used in adaptive code shortening according to one embodiment of the invention. In particular, Figures 4A-4C illustrate how shortening may be used to adapt code word lengths to match the memory page size for a non-volatile memory array. Advantageously, shortening enables the controller 130 and/or coder module 132 to support a memory page format in which the memory page size does not equal any code word length of a plurality of pre-defined code word lengths. For instance, if the controller 130 and/or coder module 132 support pre-defined LDPC code word lengths equal to 2176 and 2304 bytes, shortening can be used so that the controller 130 and/or coder module 132 further support a memory page size of 2164 bytes. Moreover, shortening can permit the controller 130 and/or coder module 132 to adaptively code data to match the non-volatile memory array format without sacrificing a large bit error rate performance.

**[0031]** In one embodiment, shortening comprises three operations. First, padding data is added to the user data to be encoded. The padding data in one embodiment is sized to be the difference between the pre-defined code length and memory page size. Second, parity data is generated based on the padding and user data. Third, the user data and parity data, which add up to the memory page size, are stored in the memory page. The padding data is not stored, but will be appended to the user data upon decoding (e.g., when the user data and parity data is read out from the memory page at a later time).

**[0032]** Figure 4A illustrates padding 410 and user data 420 of a data unit 400a. The user data 420 corresponds to an amount of user data for data unit 400a, and the padding 410 corresponds to a padding that facilitates code shortening. The padding 410 can include a data set of entirely zeros, entirely ones, etc., or any known or pre-defined data pattern. Continuing the example of the previous paragraph, if the memory page size is 2164 bytes for the non-volatile memory array, the controller 130 and/or coder module 132 can select, from various pre-defined code lengths such as those

shown in table 300, the shortest LDPC code word length having a size equal to or greater than 2164 bytes. In this case, the code length of 2176 bytes is selected. The controller 130 and/or coder module 132 can determine that the padding 410 should include a data set having length that is equal to the difference between the code word length (2176) and the memory page size (2164) of the non-volatile memory array, or 2176 – 2164 = 12 bytes of padding data. Depending on the LDPC coding parameters for the data unit 400a, the controller 130 and/or coder module 132 can further determine an amount of bytes reserved for user data 420 and an appropriate G coding matrix for determining and/or generating parity data.

**[0033]** Figure 4B illustrates an example encoding process according to one embodiment. In effect, the user data 420 is "padded" so that the combined user data and padding meet the amount of bytes reserved for user data in the chosen pre-defined code length. The parity 430 is then generated for the combined padding 410 and user data 420 of a data unit 400b. Continuing the example of the previous paragraph, the appropriate G coding matrix can be used to determine LDPC parity data for parity 430. Note that as described above, the actual distribution of the 2176 bytes between padding 410, user data 420, and parity data 430 may vary. The following table illustrates some of the possible configurations (all sizes in bytes):

**Memory Page Size 2164 bytes**

| Configuration C1 (Figure 3) | Code Length - 2176 | | |
|---|---|---|---|
| | Reserved for User Data | | Reserved for Parity |
| | 2048 | | 128 |
| With Shortening Scheme | Padding (Figure 4 – 410) | Actual User Data (Figure 4 – 420) | Parity (Figure 4 - 430) |
| | 12 | 2036 | 128 |
| Memory Page Size | Not written to memory | 2164 (2036 + 128) | |

**Memory Page Size 2164 bytes**

| Configuration C3 (Figure 3) | Code Length - 2176 | | |
|---|---|---|---|
| | Reserved for User Data | | Reserved for Parity |
| | 1920 | | 256 |
| With Shortening Scheme | Padding (Figure 4 – 410) | Actual User Data (Figure 4 – 420) | Parity (Figure 4 - 430) |
| | 12 | 1908 | 256 |
| Memory Page Size | Not written to memory | 2164 (1908 + 256) | |

**[0034]** Figure 4C illustrates a data unit 400c, having user data 420 and parity 430 of the data unit 400b with padding 410 removed. The user data 420 and the parity 430 can be written to and subsequently read from a memory page of the non-volatile memory array 140. As shown in the tables above, the amount of user data 420 and parity data 430 can equal the memory page size of the non-volatile memory array, and the padding data is not written to the page. When the page is read out later, as part of the decoding, the padding is appended back to the user data read from the page. In

this manner, coding for arbitrary page sizes can be performed using one of the plurality of pre-defined code word lengths at the cost of a small loss in coding efficiency.

**[0035]** Figure 5 is a flow diagram illustrating a process 500 of adaptive code shortening according to one embodiment of the invention. The process 500 can be executed by the controller 130 and/or the coder module 132. Advantageously, the process 500 can enable the controller 130 and/or the coder module 132 to support memory page sizes that do not equal any code word length of a plurality of pre-defined code word lengths supported by the controller 130 and/or coder module 132. The process 500 can be used to construct and manage data units 400a, 400b, and 400c described in Figures 4A-4C.

**[0036]** At block 505, the process 500 receives user data. The user data can be received from the storage interface module 112 along with a write command to write the user data to a non-volatile memory array, such as the non-volatile memory array 140.

**[0037]** At block 510, the process 500 pads the user data with padding data. The padding data can include a data set of entirely zeros, entirely ones, or a known or pre-defined data pattern. In addition, at block 510, the process 500 can further divide the user data into units having a size equal to an amount of user per data unit, which depends on corresponding coding parameters. For example, if the non-volatile memory array has a memory page size equal to 2164 bytes and the LDPC coding parameters correspond to the parameters at circle C3 of Figure 3, the user data can be divided into units having a size equal to 1908 bytes.

**[0038]** At block 515, the process 500 determines parity data for the padded user data using coding parameters. Continuing the example of the previous paragraph, if the LDPC coding parameters correspond to the parameters at circle C3, an appropriate G coding matrix can be selected and used to determine LDPC parity data for the padded user data.

**[0039]** At block 520, the process 500 outputs the user data and parity data. For example, the process 500 can output the user data and parity data for storage to E-page 144 of F-page 143 of the non-volatile memory array 140. It can be noted that the padding described with to respect to blocks 510 and 515 can be characterized as "virtual padding" since the padding itself may not be written to the memory page.

Code Rate Adjusting

**[0040]** Figure 6 is a flow diagram illustrating a process 600 of adjusting a code rate according to one embodiment of the invention. The process 600 can be executed by the controller 130 and/or the coder module 132. Advantageously, the process 600 can enable the controller 130 and/or the coder module 132 to adjust the code rate (e.g., an amount of parity per unit data) of memory pages, blocks, or other divisions of a non-volatile memory array as the memory pages, blocks, or other divisions wear out and/or experience decreased quality.

**[0041]** At block 605, the process 600 reads user data and parity data stored in a memory page. For example, the process 600 can perform a read of F-page 143 in response to a read command from the host system 110.

**[0042]** At block 610, the process 600 detects a number of bit errors when decoding the user data using parity data and coding parameters. For instance, the process 600 can determine a number of detected bit errors when decoding the user data using stored parity data and LDPC coding parameters corresponding to the memory page.

**[0043]** At block 615, the process 600 determines whether the number of bit errors exceeds a bit error threshold. The bit error threshold can depend or vary based on the coding parameters for coding data to the memory page. For example, the bit error threshold for the LDPC coding parameters at circle C1 of Figure 3 may be lower than the bit error threshold for the LDPC coding parameters at circle C2. If the process 600 determines that the number of bit errors does not exceed the bit error threshold, the process 600 terminates. On the other hand, if the process 600 determines that the number of bit errors exceeds the bit error threshold, the process 600 moves to block 620.

**[0044]** At block 620, the process 600 checks whether the coding parameters can be adjusted to reduce a code rate. In other words, the process 600 can determine whether more parity data can be used for coding. In one embodiment, the process 600 can determine whether the LDPC coding parameters may be adjusted while keeping a LDPC code word length unchanged. For instance, if E-page 144 is currently coded

using the LDPC coding parameters at circle C3 of Figure 3, the LDPC coding parameters can be adjusted to the parameters at circle C4. Alternatively, if E-page 144 currently is coded using LDPC coding parameters at circle C4 and the table 300 contains the only available LDPC coding parameters, the parameters may not be further adjusted to a lower code rate. If the process 600 determines that the coding parameters may not be adjusted to reduce the code rate, the process 600 terminates. On the other hand, if the process 600 determines that the coding parameters can be adjusted to reduce a code rate, the process 600 moves to block 625. In one embodiment, the change of code rate may be managed at a block level where the pages in the block are switched to a new code rate at the same time. In one embodiment, where MLC memory is used, upon a determination that a further reduced code rate cannot be used, rather than terminating the process 600, the page (or block of pages) may be configured to operate in a lower-page only mode.

**[0045]** At block 625, the process 600 adjusts the coding parameters and stores the adjusted coding parameters to reduce a code rate for a next write operation. The process 600 can store the adjusted coding parameters in the non-volatile memory array 140, other memory module of the storage system 120, and/or internal memory of the controller 130 and/or coder module 132. The process 600 can store an indication of the code rate or LDPC coding parameters for coding data to facilitate management of LDPC coding parameters on a memory page, block, or other level division of a non-volatile memory array. Further, the adjusted coding parameters can be used for coding user data associated with a subsequent write command received from the host system 110.

Other Variations

**[0046]** Those skilled in the art will appreciate that in some embodiments, other approaches and methods can be used. For example, the coding techniques disclosed herein can apply to codes besides LDPC codes, such as other iterative codes like turbo codes. In addition, although the coding parameters and other values disclosed in the table 300 of Figure 3 illustrate an example set of relationships between coding parameters and code word lengths, other or additional coding relationships can be used. The table 300 can include column weights with values less than 4 and greater

than 5 (such as 3 or 6), P matrix sizes with values less than 256 bits or greater than 512 (such as 128 or 1024), Δ values of less than 1 or greater than 2 (such as -1, 0, 3, or 4), Δ values having a corresponding granularity other than 128 bytes (such as 64 bytes), and a base code length with a value of less or greater than 2048 (such as 2176). Further, an amount of parity data in each data unit can be set to different values or varied depending on a quality of a storage medium. Additionally, quality metrics other than or in addition to bit errors can be used to determine whether to adjust coding parameters for coding data. Moreover, depending on the embodiment, certain of the steps described above may be removed, and others may be added. Accordingly, the scope of the present disclosure is intended to be defined only by reference to the appended claims.

**[0047]** While certain embodiments have been described, these embodiments have been presented by way of example only, and are not intended to limit the scope of the protection. Indeed, the novel methods and systems described herein may be embodied in a variety of other forms. Furthermore, various omissions, substitutions and changes in the form of the methods and systems described herein may be made without departing from the spirit of the protection. The accompanying claims and their equivalents are intended to cover such forms or modifications as would fall within the scope and spirit of the protection. For example, the systems and methods disclosed herein can be applied to hard disk drives, hybrid hard drives, and the like. In addition, other forms of storage (e.g., DRAM or SRAM, battery backed-up volatile DRAM or SRAM devices, EPROM, EEPROM memory, etc.) may additionally or alternatively be used. As another example, the various components illustrated in the figures may be implemented as software and/or firmware on a processor, ASIC/FPGA, or dedicated hardware. Also, the features and attributes of the specific embodiments disclosed above may be combined in different ways to form additional embodiments, all of which fall within the scope of the present disclosure. Although the present disclosure provides certain preferred embodiments and applications, other embodiments that are apparent to those of ordinary skill in the art, including embodiments which do not provide all of the features and advantages set forth herein, are also within the scope of this disclosure.

Accordingly, the scope of the present disclosure is intended to be defined only by reference to the appended claims.

WHAT IS CLAIMED IS:

1. A solid-state storage system, comprising:

a non-volatile memory array comprising a plurality of memory pages, each memory page having a common memory page size; and

a controller configured to:

determine the common memory page size for the non-volatile memory array;

select a low-density parity-check (LDPC) code word length from a plurality of pre-defined LDPC code word lengths, the LDPC code word length having a size equal to or greater than the common memory page size for the non-volatile memory array; and

determine LDPC coding parameters for coding data written to or read from one or more memory pages of the non-volatile memory array based at least in part on the LDPC code word length,

wherein the controller is configured to support multiple non-volatile memory page formats using the plurality of pre-defined LDPC code word lengths and to support a memory page format in which the common memory page size does not equal any LDPC code word length of the plurality of pre-defined LDPC code word lengths.

2. The solid-state storage system of claim 1, wherein the LDPC coding parameters comprise a P matrix size, a column weight, and a row weight, and further comprise at least one of a code rate, an amount of user data, and an amount of parity data.

3. The solid-state storage system of claim 1, wherein the controller is configured to select the LDPC code word length having a minimum size equal to or greater than the common memory page size for the non-volatile memory array.

4. The solid-state storage system of claim 1, wherein the memory pages comprise error-correcting code pages (E-pages).

5. The solid-state storage system of claim 1, wherein the controller is further configured to:

determine parity data for padded user data using the LDPC coding parameters, the padded user data comprising user data and padding data; and

store the user data and the parity data in a memory page of the non-volatile memory array.

6. The solid-state storage system of claim 5, wherein an amount of the padding data of the padded user data depends at least in part on a difference between the LDPC code word length and the common memory page size for the non-volatile memory array.

7. In a data storage system comprising a controller, a method of coding data, the method comprising:

determining a common memory page size for a non-volatile memory array, the non-volatile memory array comprising a plurality of memory pages, each memory page having the common memory page size;

selecting a low-density parity-check (LDPC) code word length from a plurality of pre-defined LDPC code word lengths, the LDPC code word length having a size equal to or greater than the common memory page size for the non-volatile memory array; and

determining LDPC coding parameters for coding data written to or read from one or more memory pages of the non-volatile memory array based at least in part on the LDPC code word length.

8. The method of claim 7, wherein the LDPC coding parameters comprise a P matrix size, a column weight, and a row weight, and further comprise at least one of a code rate, an amount of user data, and an amount of parity data.

9. The method of claim 7, wherein said selecting the LDPC code word length comprises selecting the LDPC code word length having a minimum size equal to or greater than the common memory page size for the non-volatile memory array.

10. The method of claim 7, wherein the memory pages comprise error-correcting code pages (E-pages).

11. The method of claim 7, further comprising:

determining parity data for padded user data using the LDPC coding parameters, the padded user data comprising user data and padding data; and

storing the user data and the parity data in a memory page of the non-volatile memory array.

12. The method of claim 11, wherein an amount of the padding data of the padded user data depends at least in part on a difference between the LDPC code word length and the common memory page size for the non-volatile memory array.

13. A solid-state storage system, comprising:

a non-volatile memory array comprising a plurality of memory pages, each memory page having a common memory page size; and

a controller configured to:

determine the common memory page size for the non-volatile memory array;

select a code word length from a plurality of pre-defined code word lengths, the code word length having a size equal to or greater than the common memory page size for the non-volatile memory array; and

determine coding parameters for coding data written to or read from one or more memory pages of the non-volatile memory array based at least in part on the code word length,

wherein the controller is configured to support multiple non-volatile memory page formats using the plurality of pre-defined code word lengths.

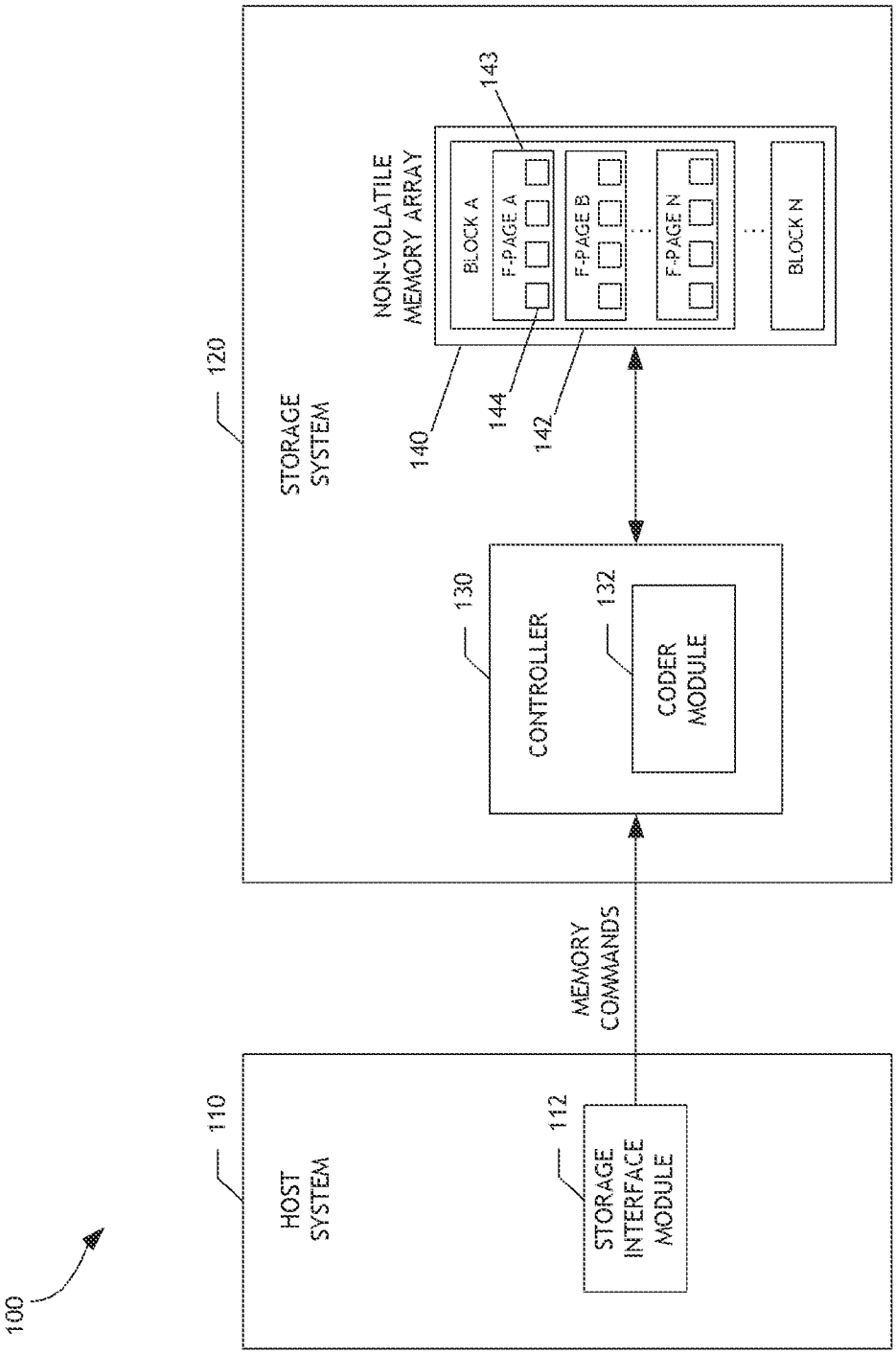

100
HOST SYSTEM
110
STORAGE INTERFACE MODULE
112
MEMORY COMMANDS
STORAGE SYSTEM
120
CONTROLLER
130
CODER MODULE
132
NON-VOLATILE MEMORY ARRAY
BLOCK A
F-PAGE A
F-PAGE B
F-PAGE N
BLOCK N
140
142
143
144

FIGURE 1

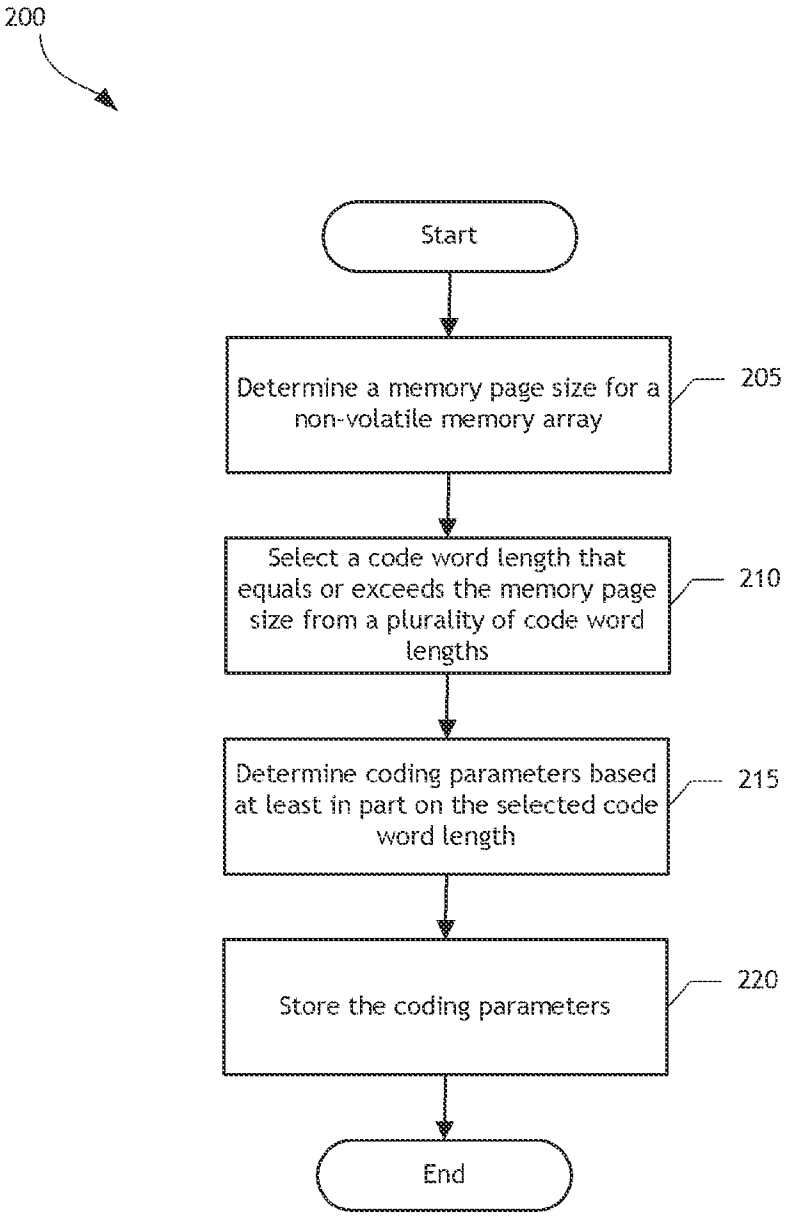

200
Start
Determine a memory page size for a non-volatile memory array
205
Select a code word length that equals or exceeds the memory page size from a plurality of code word lengths
210
Determine coding parameters based at least in part on the selected code word length
215
Store the coding parameters
220
End

FIGURE 2

300

| Column Weight | P Matrix Size (bits) | Code Length = 2048 + 128·Δ (bytes) | | |
|---|---|---|---|---|
| | | Δ = 1 | Δ = 2 | |
| 4 | 256 | C1 68 | 72 | Row Weight |
| | | 2176 | 2304 | Code Length (bytes) |
| | | 2048 | 2176 | User Data (bytes) |
| | | 0.941 | 0.944 | Code Rate |
| | 512 | C3 34 | 36 | Row Weight |
| | | 2176 | 2304 | Code Length (bytes) |
| | | 1920 | 2048 | User Data (bytes) |
| | | 0.882 | 0.889 | Code Rate |
| 5 | 256 | C2 68 | 72 | Row Weight |
| | | 2176 | 2304 | Code Length (bytes) |
| | | 2016 | 2144 | User Data (bytes) |
| | | 0.926 | 0.931 | Code Rate |
| | 512 | C4 34 | 36 | Row Weight |
| | | 2176 | 2304 | Code Length (bytes) |
| | | 1856 | 1984 | User Data (bytes) |
| | | 0.853 | 0.861 | Code Rate |

FIGURE 3

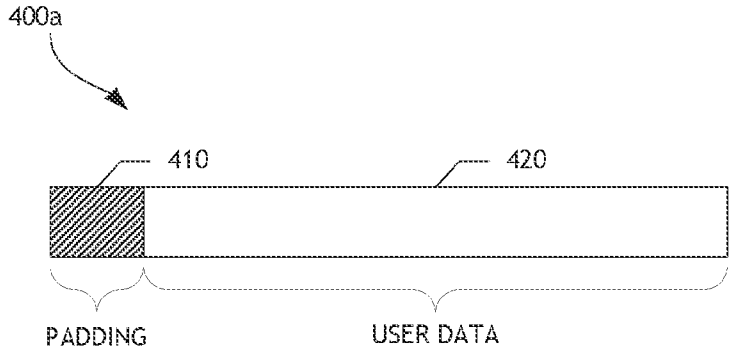



FIGURE 4A

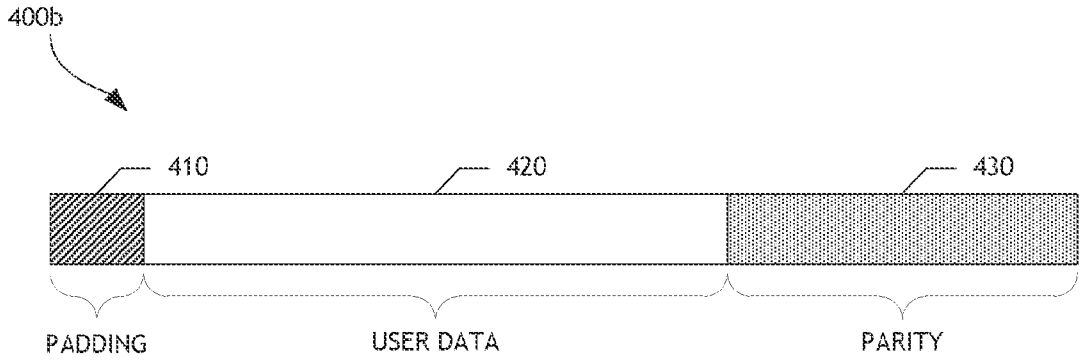



FIGURE 4B

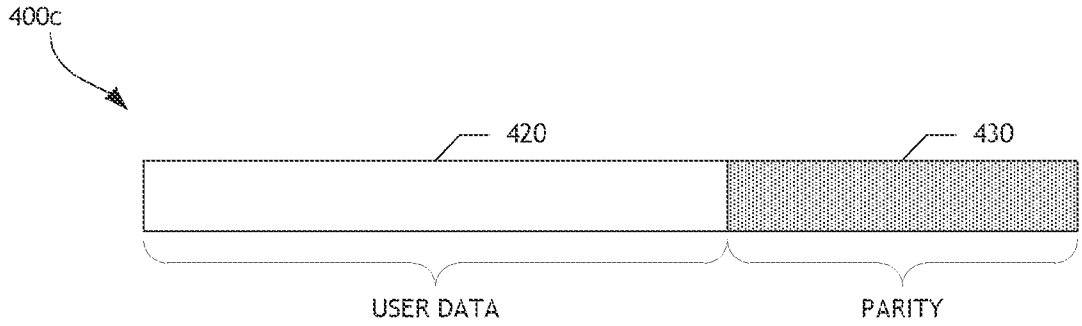



FIGURE 4C

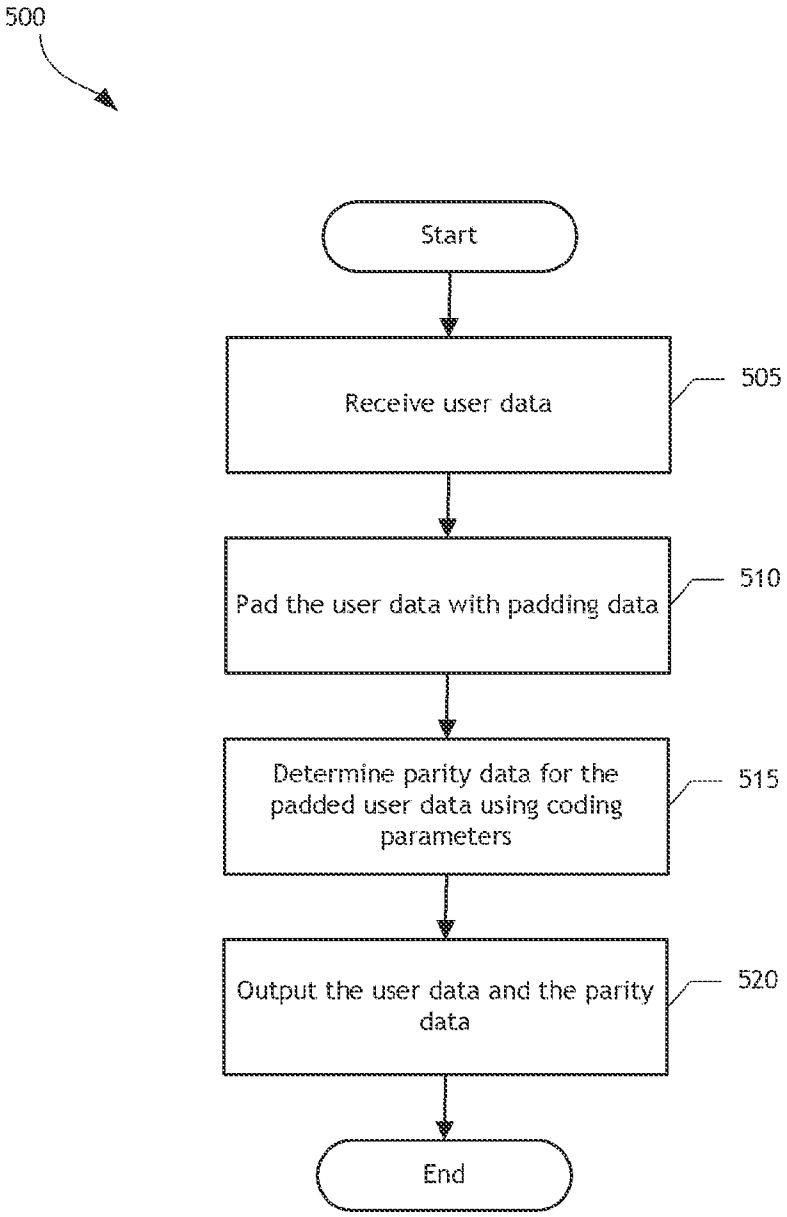

500
Start
Receive user data
505
Pad the user data with padding data
510
Determine parity data for the padded user data using coding parameters
515
Output the user data and the parity data
520
End

FIGURE 5

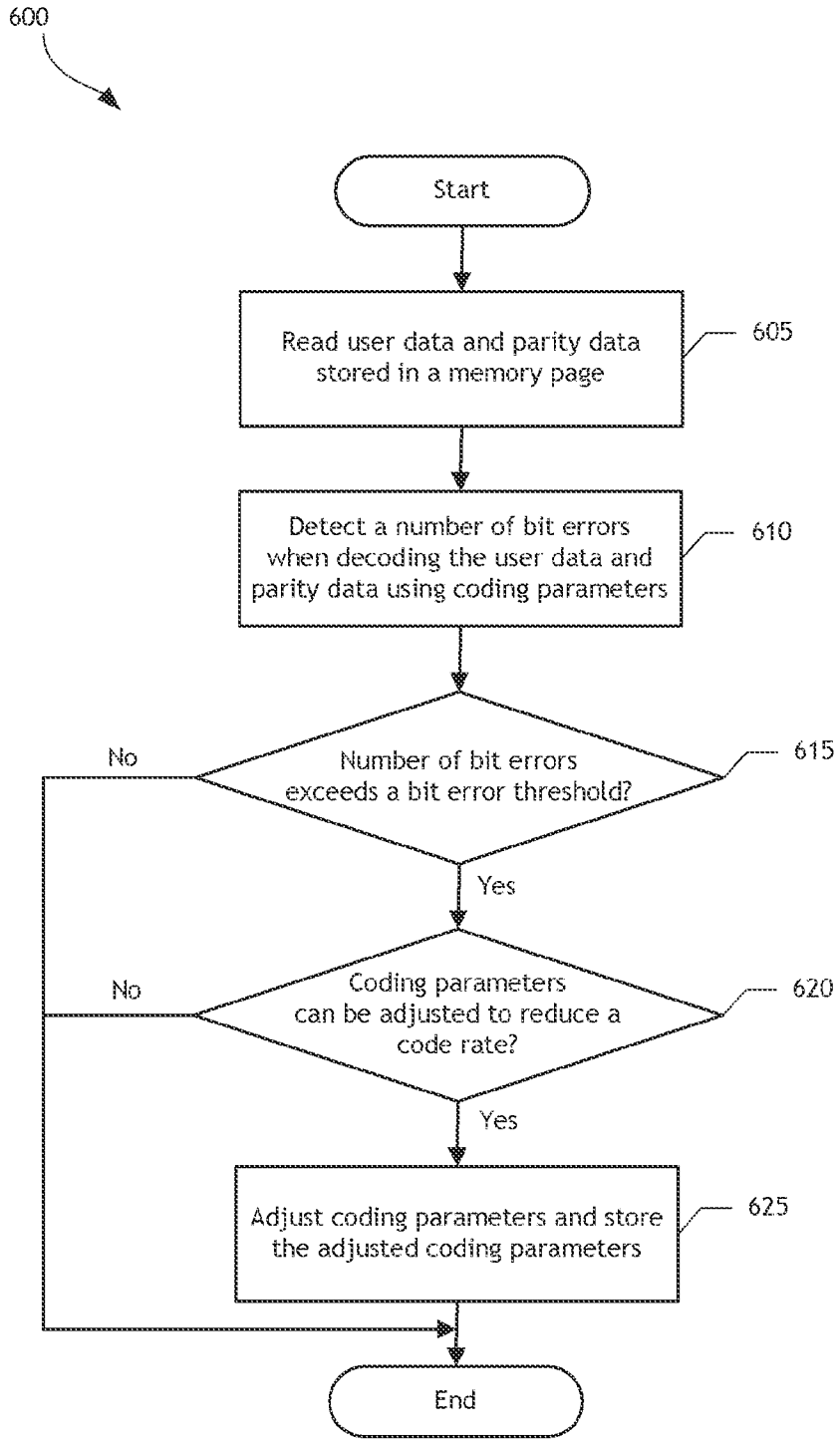

600
Start
Read user data and parity data stored in a memory page
605
Detect a number of bit errors when decoding the user data and parity data using coding parameters
610
Number of bit errors exceeds a bit error threshold?
615
No
Yes
Coding parameters can be adjusted to reduce a code rate?
620
No
Yes
Adjust coding parameters and store the adjusted coding parameters
625
End

FIGURE 6

**INTERNATIONAL SEARCH REPORT**

International application No.
**PCT/US2013/061249**

**A. CLASSIFICATION OF SUBJECT MATTER**

**G11C 29/42(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 29/42; G06F 12/16; G11C 16/04; H03M 13/05; G06F 11/10; G06F 3/08; G11C 16/06; 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: low-density parity-check, code, word, length, error, correction, memory, parameter, format, size and similar terms.

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| A | WO 2012-058328 A1 (SANDFORCE INC.) 3 May 2012<br>See paragraphs 29-32, 80, 84; figure 1; and claims 1-2. | 1-13 |
| A | US 2010-0315874 A1 (RAMIN GHODSI) 16 December 2010<br>See paragraphs 4-5; figure 4; and claims 1, 8. | 1-13 |
| A | US 2011-0231737 A1 (KENSHI DACHIKU) 22 September 2011<br>See paragraphs 21-36; figure 1; and claims 1-3. | 1-13 |
| A | JP 2008-102819 A (HITACHI LTD.) 1 May 2008<br>See paragraphs 22-26; figures 1-2; and claim 1. | 1-13 |
| A | US 2009-0070652 A1 (SE HO MYUNG et al.) 12 March 2009<br>See paragraphs 70-76; figure 5; and claim 1. | 1-13 |

☐ 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 |
|---|---|
| 17 January 2014 (17.01.2014) | **20 January 2014 (20.01.2014)** |


Name and mailing address of the ISA/KR
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701, Republic of Korea
Facsimile No. +82-42-472-7140

Authorized officer
BYUN, Sung Cheal
Telephone No. +82-42-481-8262

**INTERNATIONAL SEARCH REPORT**

Information on patent family members

International application No.
**PCT/US2013/061249**

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| WO 2012-058328 A1 | 03/05/2012 | CN 103329103 A | 25/09/2013 |
| | | EP 2633409 A1 | 04/09/2013 |
| | | KR 10-2013-0096753 A | 30/08/2013 |
| | | TW 201234170 A | 16/08/2012 |
| US 2010-0315874 A1 | 16/12/2010 | CN 102804276 A | 28/11/2012 |
| | | KR 10-1271912 B1 | 05/06/2013 |
| | | KR 10-2012-0027521 A | 21/03/2012 |
| | | TW 201110124 A | 16/03/2011 |
| | | US 2011-0141813 A1 | 16/06/2011 |
| | | US 2013-003460 A1 | 03/01/2013 |
| | | US 2013-242657 A1 | 19/09/2013 |
| | | US 7898859 B2 | 01/03/2011 |
| | | US 8345478 B2 | 01/01/2013 |
| | | US 8462552 B2 | 11/06/2013 |
| | | WO 2010-147770 A2 | 23/12/2010 |
| | | WO 2010-147770 A3 | 10/02/2011 |
| US 2011-0231737 A1 | 22/09/2011 | JP 04940322 B2 | 30/05/2012 |
| | | JP 2011-192206 A | 29/09/2011 |
| | | KR 10-1252382 B1 | 08/04/2013 |
| | | KR 10-2011-0104428 A | 22/09/2011 |
| | | US 8495468 B2 | 23/07/2013 |
| JP 2008-102819 A | 01/05/2008 | EP 1914757 A2 | 23/04/2008 |
| | | EP 1914757 A3 | 01/09/2010 |
| | | JP 04932427 B2 | 16/05/2012 |
| | | US 2008-0098158 A1 | 24/04/2008 |
| | | US 2010-0241797 A1 | 23/09/2010 |
| | | US 7757038 B2 | 13/07/2010 |
| US 2009-0070652 A1 | 12/03/2009 | KR 10-2009-0026709 A | 13/03/2009 |
| | | US 8261152 B2 | 04/09/2012 |

(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 104871249 A
(43)申请公布日 2015.08.26

(21)申请号 201380067610.7
(22)申请日 2013.09.23
(30)优先权数据
13/659,099 2012.10.24 US
(85)PCT国际申请进入国家阶段日
2015.06.23
(86)PCT国际申请的申请数据
PCT/US2013/061249 2013.09.23
(87)PCT国际申请的公布数据
WO2014/065967 EN 2014.05.01
(71)申请人 西部数据技术公司
地址 美国加利福尼亚
(72)发明人 G·陆
(74)专利代理机构 永新专利商标代理有限公司 72002
代理人 刘瑜 王英
(51)Int.Cl.
*G11C 29/42*(2006.01)
*G11C 16/34*(2006.01)

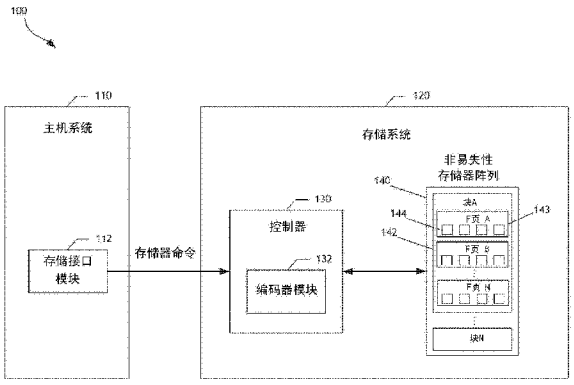
权利要求书2页 说明书8页 附图6页

(54)发明名称

数据存储系统的自适应错误纠正码

(57)摘要

本发明公开了一种被配置为自适应地编码数据的数据存储系统，在一个实施例中，数据存储系统控制器确定非易失性存储器阵列的共同的存储器页大小，例如，E 页大小。基于所述共同的存储器页大小，所述控制器从多个预定义的低密度奇偶校验 (LDPC) 码字长中选择 LDPC 码字长。所述控制器基于所选择的 LDPC 码字长来确定用于对写入到所述存储器阵列的数据或从所述存储器阵列读取的数据进行编码的 LDPC 编码参数。通过使用所述多个预定义的 LDPC 码字长，所述数据存储系统能够支持多个非易失性存储器页格式，包括这样的存储器页格式：其中所述共同的存储器页大小不等于所述多个预定义的 LDPC 码字长中的任何 LDPC 码字长。从而能够获得数据编码的灵活性和高效性。




CN 104871249 A

1. 一种固态存储系统，包括：

非易失性存储器阵列，所述非易失性存储器阵列包括多个存储器页，每个存储器页具有共同的存储器页大小；以及

控制器，所述控制器被配置为：

确定所述非易失性存储器阵列的所述共同的存储器页大小；

从多个预定义的低密度奇偶校验（LDPC）码字长中选择LDPC码字长，所述LDPC码字长的大小等于或大于所述非易失性存储器阵列的所述共同的存储器页大小；以及

至少部分地基于所述LDPC码字长来确定LDPC编码参数以用于对写入到所述非易失性存储器阵列的一个或多个存储器页的数据或从所述一个或多个存储器页中读取的数据进行编码，

其中，所述控制器被配置为使用所述多个预定义的LDPC码字长来支持多个非易失性存储器页格式，并且被配置为支持这样的存储器页格式：其中所述共同的存储器页大小不等于所述多个预定义的LDPC码字长中的任何LDPC码字长。

2. 根据权利要求1所述的固态存储系统，其中，所述LDPC编码参数包括P矩阵大小、列权值和行权值，并且进一步包括码率、用户数据量和奇偶校验数据量的至少其中之一。

3. 根据权利要求1所述的固态存储系统，其中，所述控制器被配置为选择等于或大于所述非易失性存储器阵列的所述共同的存储器页大小的具有最小大小的LDPC码字长。

4. 根据权利要求1所述的固态存储系统，其中，所述存储器页包括错误纠正码页（E页）。

5. 根据权利要求1所述的固态存储系统，其中所述控制器进一步被配置为：

采用所述LDPC编码参数来确定针对已填充用户数据的奇偶校验数据，所述已填充用户数据包括用户数据和填充数据；以及

将所述用户数据和所述奇偶校验数据存储在所述非易失性存储器阵列的存储器页中。

6. 根据权利要求5所述的固态存储系统，其中，所述已填充用户数据的所述填充数据的量至少部分地取决于所述LDPC码字长与所述非易失性存储器阵列的所述共同的存储器页大小之差。

7. 一种在包括控制器的数据存储系统中编码数据的方法，所述方法包括：

确定非易失性存储器阵列的共同的存储器页大小，所述非易失性存储器阵列包括多个存储器页，每个存储器页具有所述共同的存储器页大小；

从多个预定义的低密度奇偶校验（LDPC）码字长中选择LDPC码字长，所述LDPC码字长的大小等于或大于所述非易失性存储器阵列的所述共同的存储器页大小；以及

至少部分地基于所述LDPC码字长来确定LDPC编码参数以用于对写入到所述非易失性存储器阵列的一个或多个存储器页的数据或从所述一个或多个存储器页中读取的数据进行编码。

8. 根据权利要求7所述的方法，其中，所述LDPC编码参数包括P矩阵大小、列权值和行权值，并且进一步包括码率、用户数据量和奇偶校验数据量的至少其中之一。

9. 根据权利要求7所述的方法，其中，所述的选择所述LDPC码字长包括选择等于或大于所述非易失性存储器阵列的所述共同的存储器页大小的具有最小大小的所述LDPC码字长。

10. 根据权利要求 7 所述的方法，其中，所述存储器页包括错误纠正码页 (E 页 )。

11. 根据权利要求 7 所述的方法，进一步包括 ：

采用所述 LDPC 编码参数来确定针对已填充用户数据的奇偶校验数据，所述已填充用户数据包括用户数据和填充数据 ；以及

将所述用户数据和所述奇偶校验数据存储在所述非易失性存储器阵列的存储器页中。

12. 根据权利要求 11 所述的方法，其中，所述已填充用户数据的所述填充数据的量至少部分地取决于所述 LDPC 码字长与所述非易失性存储器阵列的所述共同的存储器页大小之差。

13. 一种固态存储系统，包括 ：

非易失性存储器阵列，所述非易失性存储器阵列包括多个存储器页，每个存储器页具有共同的存储器页大小 ；以及

控制器，所述控制器被配置为 ：

确定所述非易失性存储器阵列的所述共同的存储器页大小 ；

从多个预定义的码字长中选择码字长，所述码字长的大小等于或大于所述非易失性存储器阵列的所述共同的存储器页大小 ；以及

至少部分地基于所述码字长来确定编码参数以用于对写入到所述非易失性存储器阵列的一个或多个存储器页的数据或从所述一个或多个存储器页中读取的数据进行编码，

其中，所述控制器被配置为使用所述多个预定义的码字长来支持多个非易失性存储器页格式。

# 数据存储系统的自适应错误纠正码

## 技术领域

[0001] 本公开涉及计算机系统的诸如固态硬盘等的数据存储系统。更具体地，本公开涉及数据存储系统的自适应错误纠正码。

## 背景技术

[0002] 非易失性存储器阵列常常具有有限的耐久性。存储器阵列的耐久性通常视使用模式和磨损而定。另外，耐久性取决于非易失性存储器阵列的特定类型。例如，与具有单层单元 (SLC)NAND 介质的存储器阵列相比，具有多层单元 (MLC)NAND 介质的存储器阵列通常具有较低的耐久性。为了保护存储到存储器阵列的用户数据免于可能由减少的耐久性而导致的讹误，能够确定奇偶校验数据并且将奇偶校验数据与用户数据一起存储以便于错误检测和 / 或错误纠正。

## 附图说明

[0003] 现在参考下面的附图对实施本发明的各种特征的系统和方法进行描述，其中 ：

[0004] 图 1 示出了根据本发明的一个实施例的自适应地编码数据的存储系统。

[0005] 图 2 是示出了根据本发明的一个实施例的确定用于编码数据的编码参数的过程的流程图。

[0006] 图 3 是示出了根据本发明的一个实施例的低密度奇偶校验 (LDPC) 编码参数与 LDPC 码字长之间关系的表。

[0007] 图 4A-4C 是示出了根据本发明的一个实施例的已填充用户数据和奇偶校验数据的图。

[0008] 图 5 是示出了根据本发明的一个实施例的自适应码缩短的过程的流程图。

[0009] 图 6 是示出了根据本发明的一个实施例的调整码率的过程的流程图。

## 具体实施方式

[0010] 尽管描述了某些实施例，但是这些实施例仅以示例的方式给出，而不是要限制保护范围。实际上，本文所描述的新颖方法和系统可以以多种其他形式来实施。而且，可以进行本文所描述的方法和系统的形式上的各种省略、替换和改变，而不会脱离保护范围。

[0011] 在一些实施例中，本公开中所使用的“编码”数据指的是编码数据的过程和 / 或解码数据的过程。

[0012] 概述

[0013] 诸如固态硬盘等的存储系统通常包括与非易失性存储器阵列耦合的一个或多个控制器。通常，这些控制器由一方设计 / 制造而非易失性存储器阵列由另一方设计 / 制造。另外，来自不同制造商的存储器阵列往往具有不同的内部格式，包括不同的存储器页格式，其中每个存储器页格式对应于多个存储器页大小中的一个。而且，由于成本和多种其他竞争原因，存储系统制造商通常使用来自不同的制造商的存储器阵列。例如，存储系统制造商

可能在当前生产周期中使用一种品牌的存储器阵列，而在下一个生产周期中使用另一种品牌的存储器阵列。此外，对于标价在不同的水平的不同的存储系统型号可以使用不同的存储器阵列。

[0014] 因此，常见的设计方法是：令存储系统包括一个或多个控制器，所述一个或多个控制器具有多个专用的硬件、固件和/或软件以用于对从来自不同制造商的存储器阵列读取的数据或写入到来自不同制造商的存储器阵列的数据进行编码/解码，而实际使用的这些硬件、固件和/或软件的子集取决于在装配时或在最终设计阶段与（多个）控制器配对的实际存储器阵列。结果是，存储系统可能包括多个控制器或额外的、不使用的硬件、固件和/或软件以支持未包括在最终装配的存储系统中的一个或多个存储器阵列的多个存储器页格式。

[0015] 替代地，为减少包括在存储系统中的硬件、固件和/或软件的数量，能够将存储系统和控制器构建为仅支持一个特定的、已知的、具有已知的存储器页大小的存储器页格式，而不支持其他存储器页格式。然而，这样的方法限制了存储系统和控制器对其他类型的存储器阵列的可用性，并且可能需要构建多个不同的存储系统和控制器以支持多个存储器阵列类型。另外，在一些实例中，在设计/构建存储系统和控制器时，可能不知道存储器阵列的存储器页格式。因此，期望有针对多个已知或未知的存储器存储格式自适应地编码数据的改善的系统和方法。

[0016] 在本发明的一些实施例中，存储系统包括控制器和非易失性存储器阵列，非易失性存储器阵列具有多个存储器页，多个存储器页具有共同的存储器页大小。控制器确定非易失性存储器阵列的共同的存储器页大小，并且基于该共同的存储器页大小从多个预定义的码字长中选择码字长，例如低密度奇偶校验（LDPC）码字长。通过选择性地使用适当的码字长（从多个预定义的码字长中选择）和对应的编码参数，控制器使用至少一些共同的硬件、固件和/或软件来支持多个已知的或未知的存储器页格式。因而，控制器能够选择性地调节其操作，所以控制器能够与不同制造商所制造的不同格式的存储器阵列配对。

[0017] 在一些实施例中，随着部分地由于存储器阵列的使用而引起存储器页、块或非易失性存储器阵列的管芯老化和/或逐渐耗尽，存储设备的控制器调整用于编码用户数据的编码参数。调整编码参数的机制使控制器能够支持随着时间推移每单元数据的附加奇偶校验，从而随着存储器阵列的质量（例如，数据保持能力）降低而改善错误纠正或错误检测能力。此外，调整编码参数的机制便于平衡由每单元数据的附加奇偶校验所引起的解码时间与由附加奇偶校验数据所引起的错误纠正或错误检测益处。另外，控制器能够存储针对存储器页、块或存储器阵列的管芯所调整的编码参数，以准许不同的码率或奇偶校验率适于不同的存储器页、块或管芯。

[0018] 系统概述

[0019] 图1示出了根据本发明的一个实施例的自适应地编码数据的存储系统120。如所示出的，存储系统120（例如，混合硬盘、固态硬盘等）包括控制器130和非易失性存储器阵列140，非易失性存储器阵列140包括一个或多个存储块，标识为块“A”142一直到块“N”。每个块包括多个闪存页（F页）。例如，图1中的块A 142包括多个F页，标识为F页A 153、F页B一直到F页N。在一些实施例中，每个“F页”是在非易失性存储器阵列140中能够在单个操作中或作为单元来编程的存储单元的最小组。此外，每个F页包括多个错误纠正码

页（E 页）。在所示出的实施例中，每个 F 页包括被示出为四个方框的四个 E 页，包括 E 页 144。其他实施例可以采用以不同方式定义的 F 页或 E 页，或者每个 F 页可以包括多于或少于四个 E 页。

[0020] 控制器 130 可以从主机系统 110 中的存储接口模块 112（例如，设备驱动）接收数据和 / 或存储访问命令。由存储接口 112 所传送的存储访问命令能够包括由主机系统 110 所发布的写入命令和读取命令。命令能够指定存储系统 120 中的逻辑块地址，并且控制器 130 能够在非易失性存储器阵列 140 中执行所接收的命令。在混合硬盘中，除了非易失性存储器阵列 140 外，数据还可以存储在磁介质存储部件（图 1 中未示出）中。

[0021] 存储系统 120 能够存储从主机系统 110 接收的数据，以使得存储系统 120 能够充当主机系统 110 的存储装置。为了便于该功能，控制器 130 能够实施逻辑接口。逻辑接口能够以其中能够存储数据的一组逻辑地址（例如，连续地址）的形式向主机系统 110 呈现存储系统存储器。在内部，控制器 130 能够将逻辑地址映射到非易失性存储器阵列 140 和 / 或其他（多个）存储器模块中的多种物理存储器地址。

[0022] 控制器 130 包括编码器模块 132。在一个实施例中，编码器模块 132 确定编码参数以用于对从非易失性存储器阵列 140 中的存储器页（例如，E 页）读取的数据（例如，用户数据）或写入到存储器页的数据进行解码 / 编码。编码参数能够用于 :对从非易失性存储器阵列 140 读取的用户数据进行解码，对存储到非易失性存储器阵列 140 的用户数据进行编码，以及诸如错误检测或错误纠正等的其他用途。编码参数能够包括 LDPC 编码参数，例如，G 或 H 编码矩阵的列权值、G 或 H 编码矩阵的行权值、P 矩阵大小（例如，其中 P 矩阵是 G 或 H 编码矩阵的子矩阵）等。此外，编码器模块 132 能够确定针对未填充或已填充的用户数据的奇偶校验数据，也能够对具有对应的奇偶校验数据和填充数据的用户数据进行解码。另外，编码器模块 132 能够通过调整编码参数来对用于编码数据的码率或奇偶校验率进行调节。控制器 130 和 / 或编码器模块 132 还能够包括内部存储器（未示出），其可以属于一个或多个适当的存储器类型。

[0023] 非易失性存储器阵列 140 能够使用 NAND 闪速存储器设备来实施。能够替代地使用其他类型的固态存储器设备，例如，闪速集成电路的阵列、硫系 RAM(C-RAM)、相变存储器 (PC-RAM 或 PRAM)、可编程金属化单元 RAM(PMC-RAM 或 PMCm)、双向统一存储器 (OUM)、阻变 RAM(RRAM)、NOR 存储器、EEPROM、铁电存储器 (FeRAM)、磁阻 RAM(MRAM)、其他分立的 NVM（非易失性存储器）芯片或其任意组合。在一个实施例中，尽管可以使用单层单元 (SLC) 存储器设备或 SLC 设备和多层单元 (MLC) 设备的组合，但非易失性存储器阵列 140 优选地包括具有能够存储多于单个位信息的多层单元的 MLC 设备。在一个实施例中，存储系统 120 能够包括诸如一个或多个磁存储器模块等其他存储器模块。存储系统 120 还能够包括诸如磁存储等其他类型的存储介质。

[0024] 自适应数据编码

[0025] 图 2 是示出了根据本发明的一个实施例的确定用于编码数据的编码参数的流程图。过程 200 能够由控制器 130 和 / 或编码器模块 132 来执行。有利地，过程 200 能够使控制器 130 和 / 或编码器模块 132 能够使用多个码字长来支持多个非易失性存储器页格式。

[0026] 在方框 205，过程 200 确定诸如非易失性存储器阵列 140 等非易失性存储器阵列的存储器页大小。例如，存储器页大小能够由存储器阵列厂商来提供或基于其他已知存储

器大小来计算。例如，存储器页大小可以与非易失性存储器阵列的 E 页大小相对应，而过程 200 能够通过查询厂商提供的存储器阵列的 F 页大小并将 F 页的大小除以诸如 4 或 8 等对于非易失性存储器阵列适当的常数，来计算 E 页大小。在其他实施例中，存储器页大小能够与 F 页大小相对应。

[0027] 在方框 210，过程 200 从多种码字长中选择等于或超过存储器页大小的码字长。例如，存储器页大小能够是 2164 字节或八位位组，多个预定义的码字长能够包括 2176 字节和 2304 字节的长度。在一个实施例中，过程 200 从多个预定义的 LDPC 码字长中选择等于或大于存储器页大小的具有最小大小或数据字节数的 LDPC 码字长。例如，过程 200 能够选择 2176 字节的码字长，该码字长超过 2164 字节的存储器页大小并且具有多个预定义 LDPC 码字长中的最小大小。

[0028] 在框 215，过程 200 至少部分地基于所选择的码字长来确定编码参数。在对从非易失性存储器阵列读取的数据或写入到非易失性存储器阵列的数据进行编码时能够使用编码参数，并且编码参数使过程 200 能够管理数据的码率（例如，每数据单元的总数据的用户数据量，其中总数据包括用户数据和奇偶校验数据）。在一个实施例中，针对 LDPC 码字长的编码参数包括列权值、P 矩阵大小和行权值，并且进一步包括码率、用户数据量和奇偶校验数据量中的至少一个。

[0029] 在方框 220，过程 200 存储编码参数。例如，过程 200 能够在非易失性存储器阵列 140 中和 / 或在存储系统 120 的一个或多个其他存储介质中存储编码参数。过程 200 能够在控制器 130 和 / 或编码器模块 132 的内部存储器中存储编码参数。所存储的编码参数能够便于使用不同的编码参数来编码非易失性存储器 140 中不同的页、块或其他划分和 / 或细分，以及便于随着非易失性存储器阵列的部分老化和 / 或逐渐耗尽来随着时间推移跟踪和调整编码参数。

[0030] 图 3 是示出了根据本发明的一个实施例的 LDPC 编码参数和 LDPC 码字长之间关系的表 300。表 300 能够标明由诸如控制器 130 和 / 或编码器模块 132 等控制器所支持的 LDPC 码子长和 LDPC 编码参数。表 300 能够存储在非易失性存储器阵列 140、存储系统 120 中的一个或多个其他存储介质和 / 或控制器 130 和 / 或编码器模块 132 的内部存储器中。表 300 包括两个码长列，指示两种支持的 LDPC 码字长。一种 LDPC 码字长等于 2176 字节 (2048+128 • 1 字节），而另一种 LDPC 码字长等于 2304 字节 (2048+128 • 2 字节）。已经发现，约 2K 字节的多种码率能够提供复杂性和性能之间的最佳折衷。

[0031] 对于每个码长，包括列权值、P 矩阵大小和行权值的 LDPC 编码参数可以是多样的，以使得能够对采用不同设计的码率（例如，不同的每数据单元的总数据的用户数据量，其中总数据包括用户数据和奇偶校验数据）的数据进行编码，如表 300 中所列出的那样。例如，如果 LDPC 码字长等于 2176 字节，则适用的 LDPC 编码参数可以是在圈 C1、C2、C3 或 C4 处的 LDPC 编码参数集合的其中之一。在一个实例中，能够选择圈 C3 处的 LDPC 编码参数，其对应列权值为 4、P 矩阵大小为 512 和行权值为 34，并且还对应码率为 0.882（对于 2176 字节的总码长，有 1920 字节的用户数据）。此外，在一个实施例中，可以被定义为 2048+128x Δ 的码长，能够被基于选择 Δ 为 1 或 2 等来进行调整。如所示出的那样，对 Δ 的值的选择也影响码率。例如，在圈 C1，选择 Δ 为 1 使得码率为 0.941，并且如果列权值、P 矩阵大小和行权值保持不变，选择 Δ 为 2 使得码率为 0.944。如将进一步解释的那样，能够使用多种预

定义码率来适应不同页大小的存储器阵列。例如，如所示出的，两种码长 2176 字节和 2304 字节能够适应 2176 字节和 2304 字节的页大小。应当注意的是，在实践中预定义的码长的数量可以大大高于二，以适应多种页大小。

[0032] 支持表 300 中的 LDPC 编码参数的控制器能够有利地为非易失性存储器阵列选择一个码长，并且调整 LDPC 编码参数来以不同码率对数据进行编码。例如，当 E 页大小为 2176 字节的存储器阵列相对较新（例如，轻度使用）和 / 或经历或展现了很少的编码错误时，可以选择在圈 C1 处的 LDPC 编码参数来编码数据。在圈 C1 处的 LDPC 编码参数对应列权值为 4、P 矩阵大小为 256 和行权值为 68，以及还对应码率为 0.941。在圈 C1 处，能够使用总共 128 字节的奇偶校验来编码 2048 字节的数据。随着存储器页、块或存储器阵列的管芯老化和 / 或逐渐耗尽，能够选择在圈 C2、C3 和 C4 处的 LDPC 编码参数作为代替来编码数据。因此，响应于非易失性存储器阵列质量的改变（例如，质量的损失），控制器能够逐渐增加每单元数据的奇偶校验量，从 0.941 的码率分别到在圈 C2、C3 和 C4 处的 0.926、0.882 和 0.853 的码率。

[0033] 自适应码缩短

[0034] 整个自适应码缩短中，本发明的一些实施例能够适应具有与预定义码长不完全匹配的页大小的存储器阵列。图 4A-4C 是示出了根据本发明的一个实施例的在自适应码缩短中所使用的已填充用户数据和奇偶校验数据的图。特别地，图 4A-4C 示出了可以怎样使用缩短来调整码字长以匹配非易失性存储器阵列的存储器页大小。有利地，缩短使控制器 130 和 / 或编码器模块 132 能够支持其中存储器页大小不等于多个预定义的码字长中的任何码字长的存储器页格式。例如，如果控制器 130 和 / 或编码器模块 132 支持等于 2176 和 2304 字节的预定义的 LDPC 码字长，则能够使用缩短以使得控制器 130 和 / 或编码器模块 132 还支持 2164 字节的存储器页大小。此外，缩短能够准许控制器 130 和 / 或编码器模块 132 自适应地编码数据以使其匹配非易失性存储器阵列格式，而不牺牲大的位错误率性能。

[0035] 在一个实施例中，缩短包括三个操作。第一，将填充数据加到将要被编码的用户数据上。在一个实施例中的填充数据的大小是预定义码长与存储器页大小之差。第二，基于填充数据和用户数据产生奇偶校验数据。第三，将合计达存储器页大小的用户数据和奇偶校验数据存储在存储器页中。填充数据不被存储，但是在解码时（例如，稍后当用户数据和奇偶校验数据被从存储器页读出时）将被附加到用户数据上。

[0036] 图 4A 示出了数据单元 400a 中的填充数据 410 和用户数据 420。用户数据 420 对应于数据单元 400a 的用户数据量，而填充数据 410 对应于便于码缩短的填充数据。填充数据 410 能够包括全零、全一等的数据组或任何已知的或预定义的数据模式。继续先前段中的示例，如果非易失性存储器阵列的存储器页大小为 2164 字节，那么控制器 130 和 / 或编码器模块 132 能够从诸如表 300 中所示出的那些多种预定义码长中选择大小等于或大于 2164 字节的最短的 LDPC 码字长。在该示例中，选择 2176 字节的码长。控制器 130 和 / 或编码器模块 132 能够确定：填充数据 410 应该包括长度等于码字长 (2176) 与非易失性存储器阵列的存储器页大小 (2164) 之差的数据组，即 2176 - 2164 = 12 字节的填充数据。取决于数据单元 400a 的 LDPC 编码参数，控制器 130 和 / 或编码器模块 132 还能够确定：为用户数据 420 保留的字节量和用于确定和 / 或生成奇偶校验数据的适当的 G 编码矩阵。

[0037] 图 4B 示出了根据一个实施例的一个示例性编码过程。实际上，用户数据 420 被

“填充”，以使得组合在一起的用户数据和填充数据满足所选择的预定义字长中为用户数据保留的字节量。然后为数据单元 400b 中组合在一起的填充数据 410 和用户数据 420 生成奇偶校验 430。继续先前段的示例，能够使用适当的 G 编码矩阵来确定 LDPC 奇偶校验数据以用于奇偶校验 430。注意，如上文所描述的，2176 字节在填充数据 410、用户数据 420 和奇偶校验数据 430 中的实际分配可以变化。下表示出了一些可能的配置（所有大小单位为字节）：

[0038] 存储器页大小 2164 字节

[0039]

<table>
<tr><td rowspan="3">配置 C1（图 3）</td><td colspan="3">码长 - 2176</td></tr>
<tr><td colspan="2">为用户数据保留</td><td>为奇偶校验保留</td></tr>
<tr><td colspan="2">2048</td><td>128</td></tr>
<tr><td rowspan="2">具有缩短方案</td><td>填充数据（图 4 - 410）</td><td>实际用户数据（图 4 -4 20）</td><td>奇偶校验（图 4 - 430）</td></tr>
<tr><td>12</td><td>2036</td><td>128</td></tr>
<tr><td>存储器页大小</td><td>未写入存储器</td><td colspan="2">2164 (2036 + 128)</td></tr>
</table>

[0040] 存储器页大小 2164 字节

[0041]

<table>
<tr><td rowspan="3">配置 C3（图 3）</td><td colspan="3">码长 - 2176</td></tr>
<tr><td colspan="2">为用户数据保留</td><td>为奇偶校验保留</td></tr>
<tr><td colspan="2">1920</td><td>256</td></tr>
<tr><td rowspan="2">具有缩短方案</td><td>填充数据（图 4 - 410）</td><td>实际用户数据（图 4 - 420）</td><td>奇偶校验（图 4 - 430）</td></tr>
<tr><td>12</td><td>1908</td><td>256</td></tr>
<tr><td>存储器页大小</td><td>未写入存储器</td><td colspan="2">2164 (1908 + 256)</td></tr>
</table>

[0042] 图 4C 示出了数据单元 400c，其具有数据单元 400b 的用户数据 420 和奇偶校验 430，移除了填充数据 410。能够将用户数据 420 和奇偶校验 430 写入到非易失性存储器阵列 140 的存储器页，并随后从非易失性存储器阵列 140 的存储器页读取用户数据 420 和奇偶校验 430。如上表所示，用户数据 420 和奇偶校验数据 430 的量可以等于非易失性存储器阵列的存储器页大小，并且填充数据不被写入到页中。当该页稍后被读出时，作为解码的一部分，填充数据被附加回从该页读取的用户数据。照这样，能够以在编码效率上很小的损失为代价使用多个预定义码字长中的一个来执行针对任意页大小的编码。

[0043] 图 5 是示出了根据本发明的一个实施例的自适应码缩短的过程 500 的流程图。过程 500 可以由控制器 130 和 / 或编码器模块 132 来执行。有利地，过程 500 可以使控制器 130 和 / 或编码器模块 132 能够支持这样的存储器页大小 :其不等于控制器 130 和 / 或编

码器模块 132 所支持的多个预定义码字长中的任意码字长。过程 500 能够用于构建和管理图 4A-4C 中所描述的数据单元 400a、400b 和 400c。

[0044] 在方框 505，过程 500 接收用户数据。用户数据能够连同将用户数据写入到非易失性存储器阵列（例如，非易失性存储器阵列 140）的写入命令一起，从存储接口模块 112 接收。

[0045] 在方框 510，过程 500 用填充数据来对用户数据进行填充。填充数据能够包括全零、全一的数据组或者已知的或预定义的数据模式。另外，在方框 510，过程 500 还能够将用户数据分成大小等于每数据单元的用户数据量的单元，这取决于对应的编码参数。例如，如果非易失性存储器阵列具有等于 2164 字节的存储器页大小并且 LDPC 编码参数对应图 3 中圈 C3 处的参数，那么能够将用户数据分成大小等于 1908 字节的单元。

[0046] 在方框 515，过程 500 使用编码参数确定用于已填充用户数据的奇偶校验数据。继续先前段的示例，如果 LDPC 编码参数对应圈 C3 处的参数，那么能够选择并使用适当的 G 编码矩阵来确定用于已填充用户数据的 LDPC 奇偶校验数据。

[0047] 在方框 520，过程 500 输出用户数据和奇偶校验数据。例如，过程 500 能够输出用户数据和奇偶校验数据以使其存储到非易失性存储器阵列 140 中的 F 页 143 中的 E 页 144。能够注意到的是，参照方框 510 和方框 515 所描述的填充可以被表征为“虚拟填充”，这是因为填充数据自身可以不被写入到存储器页。

[0048] 码率调整

[0049] 图 6 是示出了根据本发明的一个实施例的调整码率的过程 600 的流程图。过程 600 能够由控制器 130 和 / 或编码器模块 132 来执行。有利地，过程 600 可以使控制器 130 和 / 或编码器模块 132 能够随着存储器页、块或非易失性存储器阵列的其他划分逐渐耗尽和 / 或经历质量降低，来调整存储器页、块或其他划分的码率（例如，每单元数据的奇偶校验的量）。

[0050] 在方框 605，过程 600 读取存储在存储器页中的用户数据和奇偶校验数据。例如，过程 600 能够响应于来自主机系统 110 的读取命令而执行对 F 页 143 的读取。

[0051] 在方框 610，当使用奇偶校验数据和编码参数来解码用户数据时，过程 600 检测到若干位错误。例如，当使用所存储的奇偶校验数据和对应于存储器页的 LDPC 编码参数来解码用户数据时，过程 600 能够确定若干检测到的位错误。

[0052] 在方框 615，过程 600 对位错误的数量是否超过位错误阈值进行确定。位错误阈值能够取决于用于将数据编码到存储器页的编码参数或者基于编码参数而变化。例如，针对图 3 中圈 C1 处的 LDPC 编码参数的位错误阈值可以低于针对圈 C2 处的 LDPC 编码参数的位错误阈值。如果过程 600 确定位错误的数量不超过位错误阈值，那么过程 600 终止。另一方面，如果过程 600 确定位错误的数量超过位错误阈值，那么过程 600 进行到方框 620。

[0053] 在方框 620，过程 600 检查是否能够调整编码参数以减少码率。换言之，过程 600 能够确定是否可以使用更多奇偶校验数据来进行编码。在一个实施例中，过程 600 能够确定是否可以调整 LDPC 编码参数同时保持 LDPC 码字长不变。例如，如果当前使用图 3 中圈 C3 处的 LDPC 编码参数来编码 E 页 144，那么能够将 LDPC 编码参数调整为圈 C4 处的参数。替代地，如果当前使用圈 C4 处的 LDPC 编码参数来编码 E 页 144 并且表 300 包含唯一可用的 LDPC 编码参数，那么参数不可以被进一步调整为更低码率。如果过程 600 确定不可以调

整编码参数以减小码率，那么过程 600 终止。另一方面，如果过程 600 确定能够调整编码参数以减小码率，那么过程 600 进行到方框 625。在一个实施例中，可以在块一级上管理码率的改变，其中块中的页被同时转换到新码率。在使用 MLC 存储器的一个实施例中，在确定了不能使用进一步减少的码率时，页（或页构成的块）可以被配置为在仅较低页模式下操作，而不是终止过程 600。

[0054] 在方框 625，过程 600 调整编码参数并存储所调整的编码参数以减少下一次写入操作的码率。过程 600 能够将所调整的编码参数存储在非易失性存储器阵列 140 中、存储系统 120 的其他存储器模块中和 / 或控制器 130 和 / 或编码器模块 132 的内部存储器中。过程 600 能够存储关于用于编码数据的码率或 LDPC 编码参数的指示，以便于管理存储器页、块或非易失性存储器阵列的其他级别的划分上的 LDPC 编码参数。此外，所调整的编码参数能够用于对与随后从主机系统 110 接收到的写入命令相关联的用户数据进行编码。

[0055] 其他变化

[0056] 本领域技术人员将会意识到，在一些实施例中能够使用其他方法和途径。例如，本文所披露的编码技术能够应用到除 LDPC 码之外的码，诸如像 Turbo 码等其他迭代码。另外，尽管图 3 中表 300 所披露的编码参数和其他值示出了一组编码参数和码字长之间关系的示例，但其他的或附加的编码关系也能够被使用。表 300 能够包括值小于 4 和大于 5（例如，3 或 6）的列权值、值小于 256 位或大于 512 位（例如，128 或 1024）的 P 矩阵大小、小于 1 或大于 2（例如，-1、0、3 或 4）的 Δ 值、对应的粒度不同于 128 字节（例如，64 字节）的 Δ 值以及具有小于或大于 2048（例如，2176）的值的基码长。此外，在每个数据单元中奇偶校验数据量能够被设置为不同的值，或者取决于存储介质的质量而变化。另外，能够采用除位错误之外的或不同于位错误的质量测度来确定是否调整用于编码数据的编码参数。而且，可以取决于实施例而移除上文所描述的特定步骤以及可以加入其它步骤。因此，旨在仅参照所附权利要求来定义本公开的范围。

[0057] 尽管已经描述了某些实施例，但这些实施例只是以示例的方式给出，而并不是要限制保护范围。实际上，可以以多种其他形式实施本文所描述的新颖方法和系统。另外，可以进行本文所描述的方法和系统的形式上的各种省略、替换和改变，而不脱离本保护的精神。所附权利要求及其等价物旨在涵盖这样的将落入本保护的范围和精神的形式或变型。例如，能够将本文所披露的系统和方法应用到硬盘驱动器、混合硬盘等。另外，可以额外地或可替代地使用其他形式的存储（例如，DRAM 或 SRAM、电池备用易失性 DRAM 或 SRAM 设备、EPROM、EEPROM 存储器等等）。作为另一个示例，可以将在附图中示出的各种部件实施为处理器上的软件和 / 或固件、ASIC/FPGA 或专用硬件。同样，上文所披露的特定实施例的特征和属性可以以不同的方式组合以形成另外的实施例，其全部落入本公开的范围内。尽管本公开提供了某些优选的实施例和应用，但对本领域技术人员显而易见的是，其他实施例，包括没有提供本文所阐述的全部特征和优点的实施例，也在本公开的范围内。因此，旨在仅参考所附权利要求来定义本公开的范围。

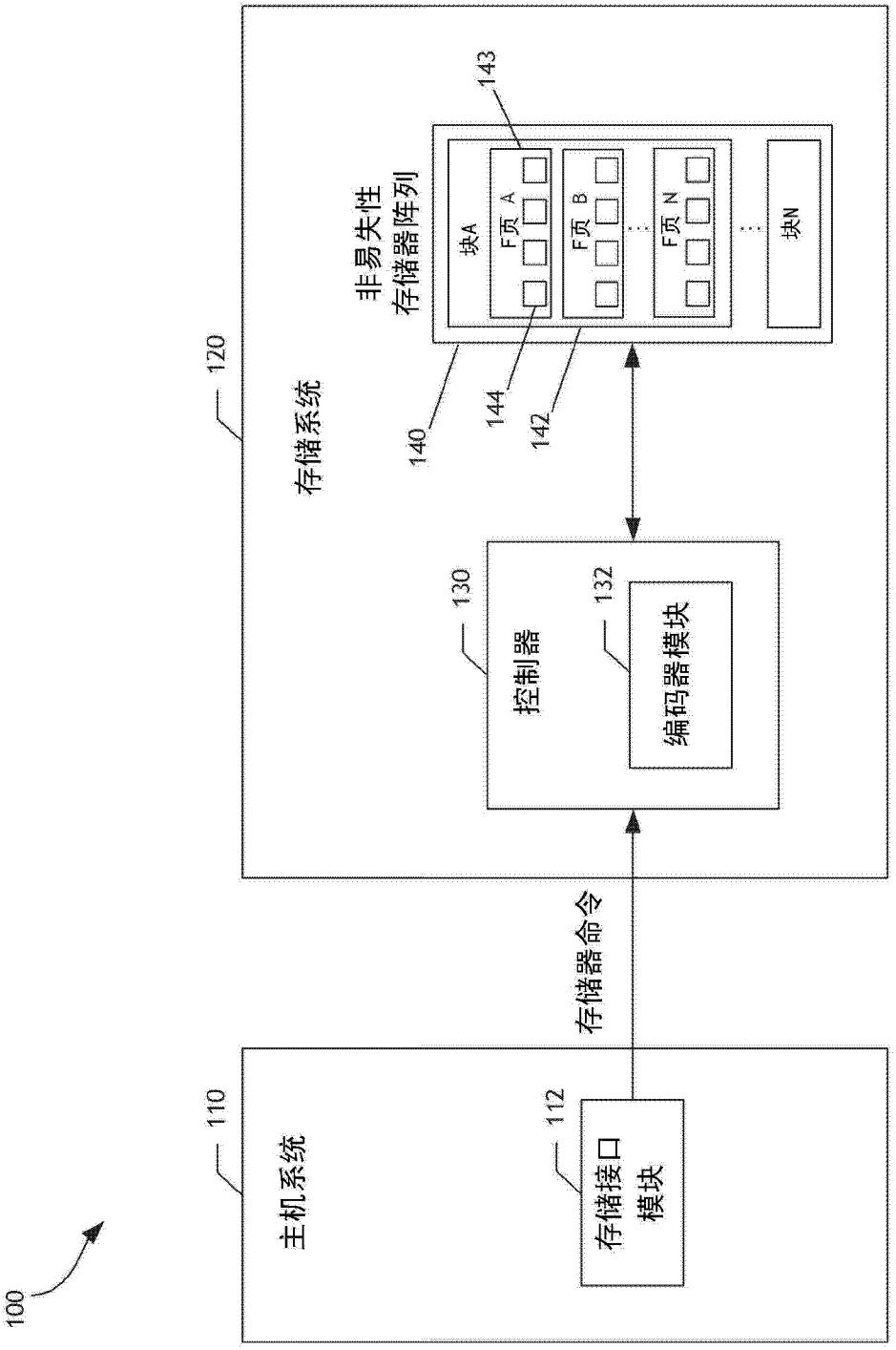

100
主机系统
110
存储接口
模块
112
存储器命令
存储系统
120
非易失性
存储器阵列
143
块A
F页 A
F页 B
F页 N
块N
140
144
142
控制器
130
编码器模块
132

图 1

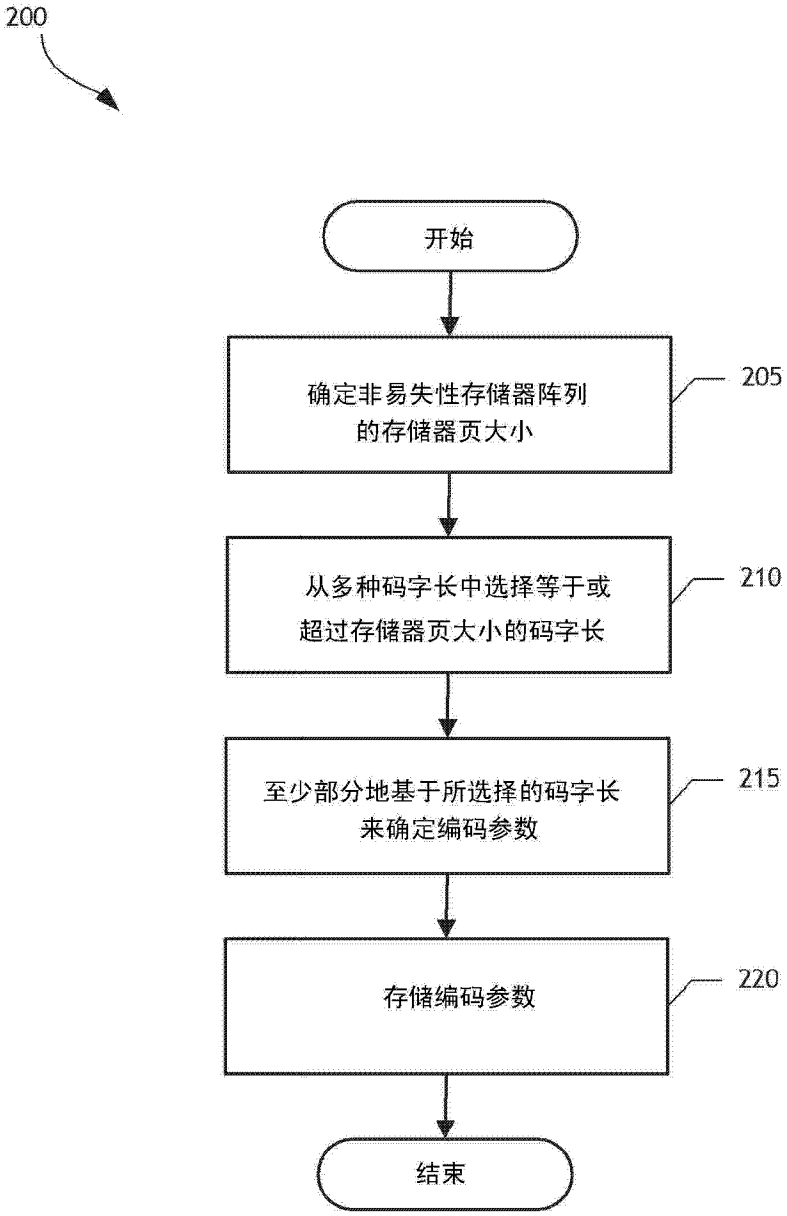

200
开始
确定非易失性存储器阵列的存储器页大小
205
从多种码字长中选择等于或超过存储器页大小的码字长
210
至少部分地基于所选择的码字长来确定编码参数
215
存储编码参数
220
结束

图 2

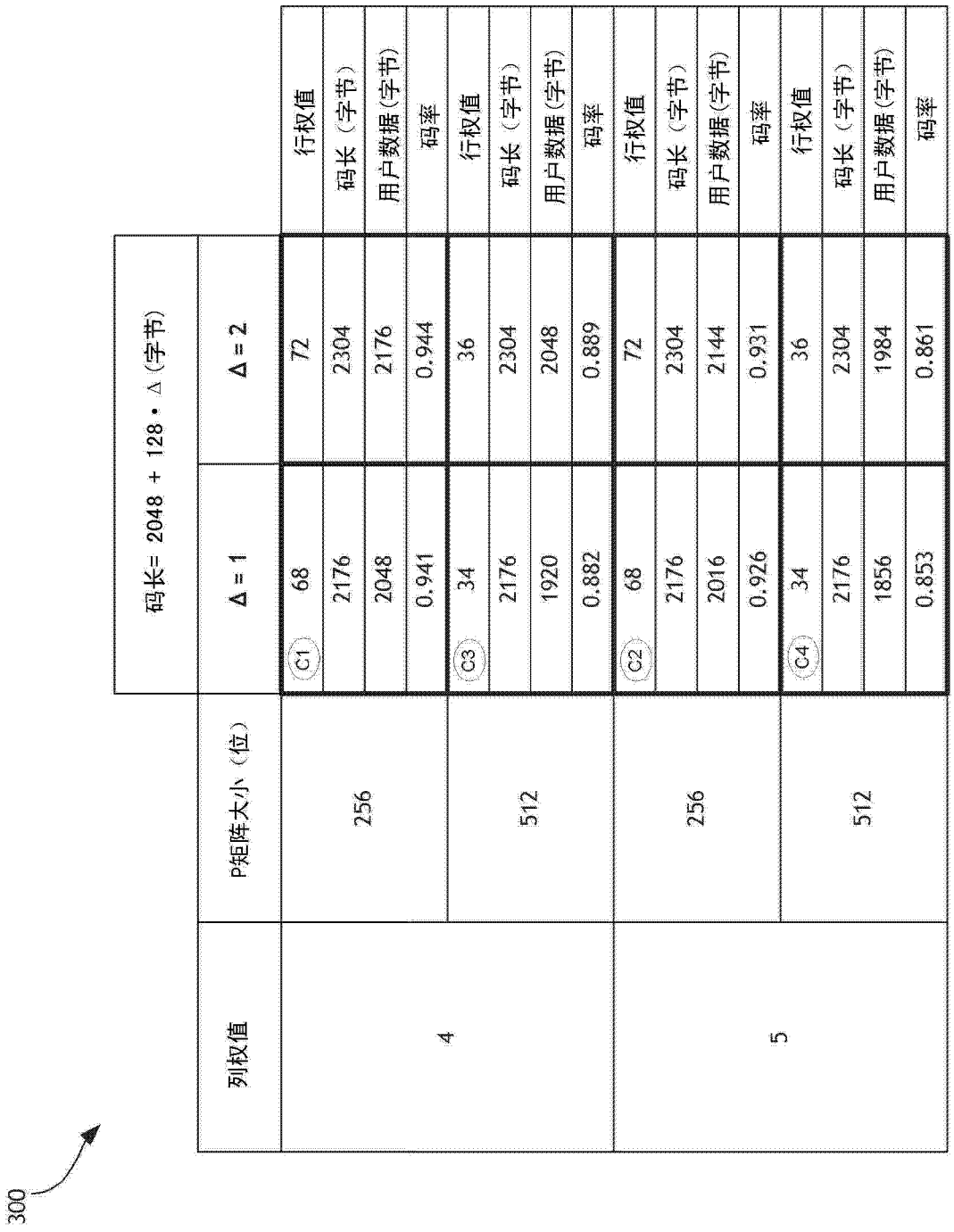

300
列权值
P矩阵大小（位）
码长= 2048 + 128 • Δ（字节）
Δ = 1
Δ = 2
4
256
C1
68
72
2176
2304
2048
2176
0.941
0.944
512
C3
34
36
2176
2304
1920
2048
0.882
0.889
5
256
C2
68
72
2176
2304
2016
2144
0.926
0.931
512
C4
34
36
2176
2304
1856
1984
0.853
0.861
行权值
码长（字节）
用户数据(字节)
码率

图 3

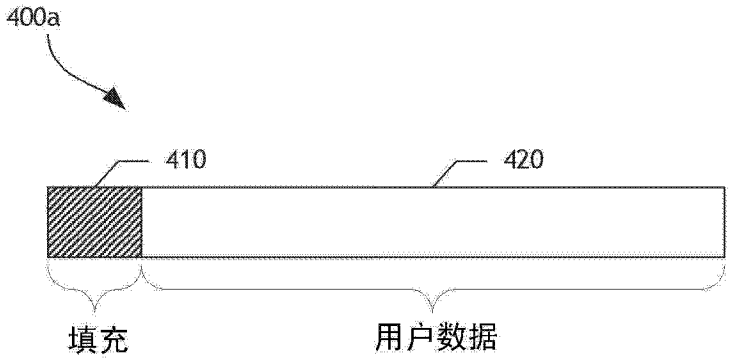

400a
410
420
填充
用户数据

图 4A

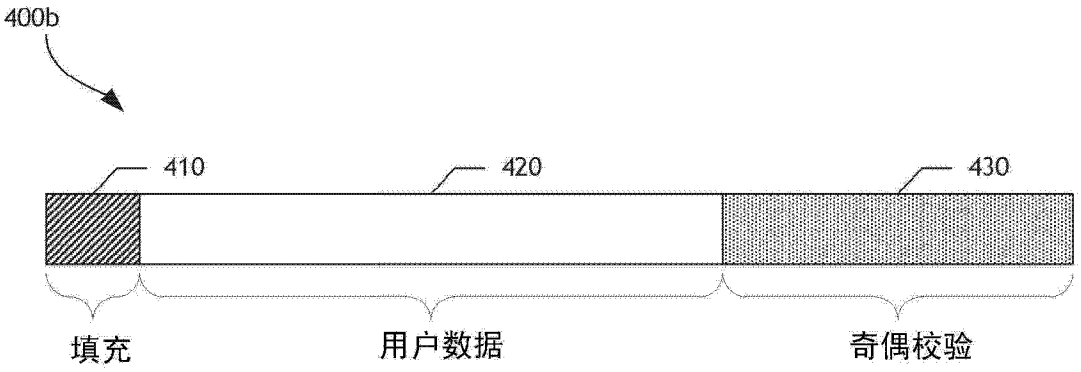

400b
410
420
430
填充
用户数据
奇偶校验

图 4B

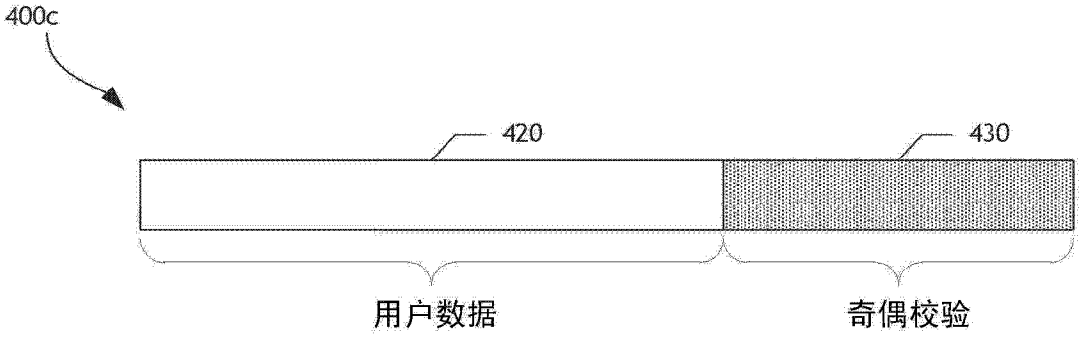

400c
420
430
用户数据
奇偶校验

图 4C

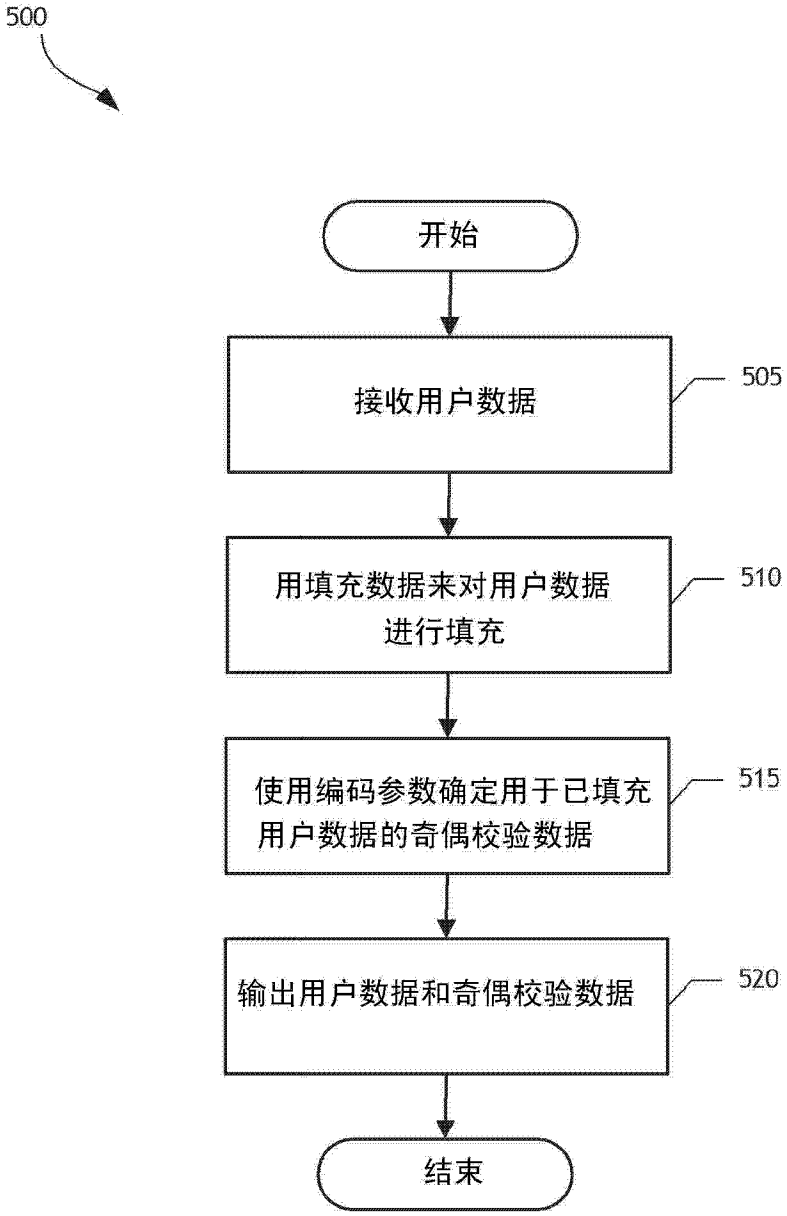

500
开始
接收用户数据
505
用填充数据来对用户数据进行填充
510
使用编码参数确定用于已填充用户数据的奇偶校验数据
515
输出用户数据和奇偶校验数据
520
结束

图 5

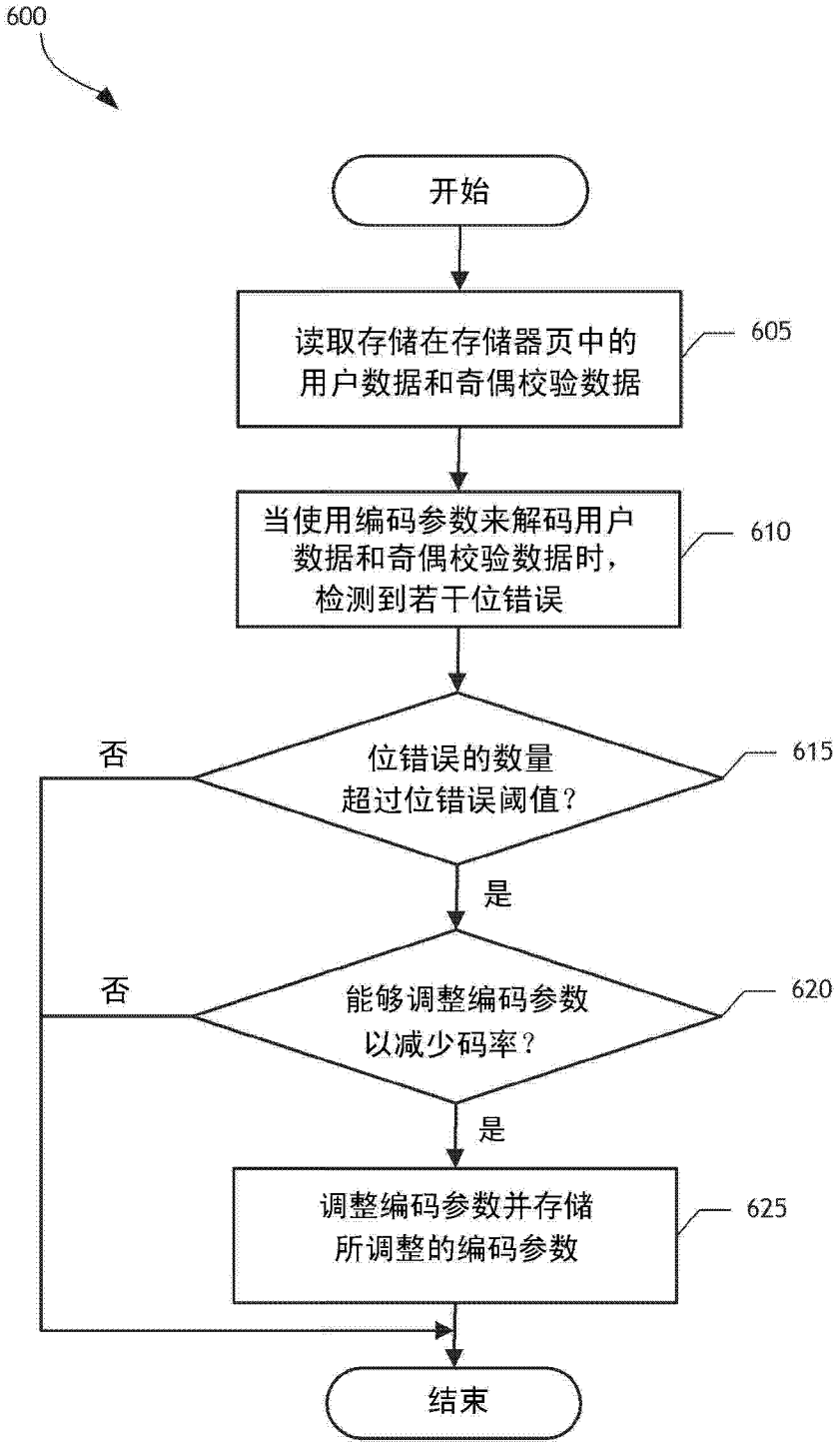

600
开始
读取存储在存储器页中的
用户数据和奇偶校验数据
605
当使用编码参数来解码用户
数据和奇偶校验数据时，
检测到若干位错误
610
否
位错误的数量
超过位错误阈值？
615
是
否
能够调整编码参数
以减少码率？
620
是
调整编码参数并存储
所调整的编码参数
625
结束

图 6

## Abstract

A data storage system configured to adaptively code data is disclosed. In one embodiment, a data storage system controller determines a common memory page size, such as an E-page size, for a non-volatile memory array. Based on the common memory page size, the controller selects a low-density parity-check (LDPC) code word length from a plurality of pre-defined LDPC code word lengths. The controller determines LDPC coding parameters for coding data written to or read from the memory array based on the selected LDPC code word length. By using the plurality of pre-defined LDPC code word lengths, the data storage system can support multiple non-volatile memory page formats, including memory page formats in which the common memory page size does not equal any LDPC code word length of the plurality of pre-defined LDPC code word lengths. Flexibility and efficiency of data coding can thereby be achieved.