

ERROR CORRECTING CODES FOR INCREASED STORAGE CAPACITY IN MULTILEVEL MEMORY DEVICES

Field

[0001] Embodiments of the present disclosure relate to the field of memory, and more particularly, to error correcting codes for increased storage capacity in multilevel memory devices.

Background

[0002] In the usual workflow for the development of a nonvolatile memory (NVM) device such as a flash device, error correcting codes (ECCs) are designed after fixing key parameters of the NVM device (e.g., size of technology node, number of bits per cell, etc.). At most, some rough estimate of ECC impact on latency times and logic area are performed.

Brief Description of the Drawings

[0003] Embodiments will be readily understood by the following detailed description in conjunction with the accompanying drawings. To facilitate this description, like reference numerals designate like structural elements. Embodiments are illustrated by way of example and not by way of limitation in the figures of the accompanying drawings.

[0004] **Figure 1** illustrates a nonvolatile memory device in accordance with an embodiment.

[0005] **Figure 2** illustrates a calculation system in accordance with an embodiment.

[0006] **Figure 3** is a number of flowcharts illustrating various calculation operations in accordance with an embodiment.

[0007] **Figure 4** illustrates a table of combinations of parameters in accordance with an embodiment.

[0008] **Figure 5** illustrates the nonvolatile memory device in accordance with an embodiment.

[0009] **Figure 6** is two flowcharts illustrating an encoding operation and a decoding operation in accordance with an embodiment.

[0010] **Figure 7** is a chart graphing error rates against signal-to-noise ratios in accordance with an embodiment.

[0011] **Figure 8** illustrates a host device incorporating an NVM device in accordance with an embodiment.

Detailed Description

[0012] In the following detailed description, reference is made to the accompanying drawings which form a part hereof wherein like numerals designate like parts throughout, and in which is shown by way of illustration embodiments in which the disclosure may be practiced. It is to be understood that other embodiments may be utilized and structural or logical changes may be made without departing from the scope of the present disclosure. Therefore, the following detailed description is not to be taken in a limiting sense, and the scope of embodiments in accordance with the present disclosure is defined by the appended claims and their equivalents.

[0013] Various operations may be described as multiple discrete operations in turn, in a manner that may be helpful in understanding embodiments of the present

disclosure; however, the order of description should not be construed to imply that these operations are order dependent.

[0014] For the purposes of the present disclosure, the phrase "A and/or B" means "(A), (B), or (A and B)." For the purposes of the present disclosure, the phrase "A, B, and/or C" means "(A), (B), (C), (A and B), (A and C), (B and C), or (A, B and C)."

[0015] Various modules and components may be introduced and described in terms of an operation provided by the modules and components. These modules and components may include hardware, software, and/or firmware elements in order to provide the described operations. While some of these modules and components may be shown with a level of specificity, e.g., providing discrete elements in a set arrangement, other embodiments may employ various modifications of elements/arrangements in order to provide the associated operations within the constraints/objectives of a particular embodiment.

[0016] The description may use the phrases "in an embodiment," or "in embodiments," which may each refer to one or more of the same or different embodiments. Furthermore, the terms "comprising," "including," "having," and the like, as used with respect to embodiments of the present disclosure, are synonymous.

[0017] The usual workflow for development of a nonvolatile memory (NVM) device described above may be efficient when the number of parity information bits is considerably less than the number of non-parity information bits. However, with the shrinking size of technology nodes and increasing number of bits stored in each cell, the raw bit error rate (BER) becomes greater and greater and, consequently, the number of parity information bits increases. In these situations the purported gain in storage

capacity from increasing a number of bits per cell may be offset by a corresponding increase in a number of parity information bits that are desired to achieve a reliability target. This is especially the case when hard-decision ECC solutions, e.g., Hamming and BCH codes, are used. Accordingly, embodiments discussed herein provide metrics and ECC schemes that may be used to increase storage capacity of NVM devices.

[0018] **Figure 1** illustrates an NVM device **100** in accordance with an embodiment. The NVM device **100** may include a programmer **104** that receives input data from, e.g., a host device, encodes the input data, and writes the encoded data into a cell matrix **108** with which it is coupled. The cell matrix **108** may be a matrix of multilevel memory cells that are each formed by a floating gate metal oxide semiconductor (MOS) transistor. In other embodiments, other transistor or transistor-like technologies may be used. In some embodiments, the multilevel memory cells (or simply "cells") may be phase change memory cells.

[0019] The NVM device **100** may also include a reader **112** coupled to the cell matrix **108**. The reader **112** may access the encoded data stored in the cell matrix **108** by using one or more sense amplifiers to read voltage levels, V_t , of the various cells and determine the logic level, or "cell state" by comparing the voltage levels to one or more threshold values. The cells may have more than two logic levels, or "cell states," determined by the one or more sense amplifiers comparing the V_t of a particular MLC to a plurality of voltage thresholds. Thus, a logic level may correspond to a discrete range of V_t .

[0020] After accessing the encoded data, the reader **112** may decode the data in a manner that is complementary to the encoding operation of the programmer **104**. The reader **112** may then provide the output data to, e.g., the host device.

[0021] In the encoding operation provided by the programmer **104**, a number of non-parity information bits, which represent the input data, may be generated along with a number of parity information bits, which may be used to correct errors that may occur in the subsequent retrieval of the non-parity information bits. The non-parity information bits may be written to non-parity cells while the parity information bits may be written to parity cells. There are a number of parameters of the NVM device **100**, including the ECC scheme used in the encoding process, that have a bearing on the percentage of parity cells of the cell matrix that are needed to meet target error rates of the NVM device **100**. Accordingly, embodiments of the present disclosure provide systems, methods, and apparatuses for selecting the parameters that increase storage capacity of the cell matrix **108** while maintaining desired operating characteristics.

[0022] **Figure 2** illustrates a calculation system **200** that may be used to select parameters of the NVM device **100** in accordance with some embodiments. The calculation system **200** may include four calculation modules: a non-parity calculation module **204**, an actual bit per cell (ABC) calculation module **208**, a parity calculation module **212**, and a nominal bit per cell (b/c) calculation module **216**, coupled to each other at least as shown.

[0023] Each of these calculation modules may receive respective inputs that are related to a selected combination of parameters of the NVM device **100**. In response, each of these calculation modules may provide a respective output that may be used as

a direct and/or indirect basis for selecting a desired combination of parameters of the NVM device **100**. The calculations provided by these calculation modules may account for the functional interdependence of the various parameters. The metric ABC may facilitate operational comparisons of the various combinations of parameters as will be described below.

[0024] **Figure 3** includes flowcharts describing calculation operations of the various modules of **Figure 2** in accordance with some embodiments. **Figure 4** illustrates a table **400** providing seven different combinations of parameters, and calculated values for each, in accordance with some embodiments.

[0025] Referring first to combination A, the b/c calculation module **216** may receive, as inputs, a cell quantum of 1 and a number of distributions of 4. A cell quantum refers to a number of cells that are read together to extract bit information and the number of distributions refers to the number of logical levels of each cell. Given these inputs, the b/c calculation module **216** may determine that the b/c value is 2. That is, each cell may represent an average of two bits.

[0026] Referring also to **Figure 3a**, the non-parity calculation module **204** may receive, at block **304**, a device size of the NVM device **100**. The device size may be given as a number of non-parity information bits that the NVM device **100** is capable of storing. In the combinations of table **400** the device size is given as 32,678 Megabits (Mbit). At block **308**, the non-parity calculation module **204** may receive the b/c from the b/c calculation module **216**. Given these inputs, the non-parity calculation module **204** may determine a number of non-parity cells at block **312**. With combination A, the number of non-parity cells may be $16,384 \times 10^6$.

[0027] Referring now to **Figure 3c**, the parity calculation module **212** may receive the device size of the NVM device **100** at block **316**; a number of distributions of the cells of the NVM device **100** at block **320**; an ECC scheme to be used in programming/reading data to/from the NVM device **100** at block **324**; and a b/c from the b/c calculation module **216** at block **328**. The parity calculation module **212** may use these inputs as a basis to determine a number of parity cells at block **332**.

[0028] In some embodiments, the parity calculation module **212** may determine an ECC value determined based on a BER associated with the number of distributions provided and a target device error rate (DER) for the NVM device **100**. Referring to parameters of combination A and assuming that all the distributions have the same width, e.g., 1.3 volts (V), a read window of 5.5 V may be used to accommodate the 4 distributions of a cell. This may result in a BER of around 10^{-5} . To achieve a DER of 10 parts per million (ppm) given this BER, the parity calculation module **212** may determine a 6 bit ECC value may be used. The ECC value may refer to the number of bits that are capable of being corrected per ECC block, which may be, e.g., 512 non-parity information bytes.

[0029] With combination A using BCH codes as the ECC scheme, the parity calculation module **212** may determine that there would be 78 parity information bits per ECC block of 512 non-parity information bytes. This may provide a parity overhead of approximately 2% and result in approximately 328×10^6 parity cells for the given embodiment.

[0030] While certain values may be described as being determined by certain modules in this disclosure, in other embodiments, the given modules may receive the values as inputs from other modules, and vice versa.

[0031] Referring now to **Figure 3b**, the ABC calculation module **208** may receive the size of the NVM device **100** at block **336**; the number of non-parity cells from the non-parity calculation module **204** at block **340**; and the number of parity cells from the parity calculation module **212** at block **344**. The ABC calculation module **208** may then use these inputs as a basis for determining an ABC at block **348** by the following equation:

[0032]
$$ABC = (\text{bits of the device}) / (\text{non-parity cells} + \text{parity cells}) \quad \text{Equation 1.}$$

[0033] Thus, the denominator of Equation 1 represents the true number of cells that are needed to store both the non-parity information bits and the parity information bits that are generated to meet the target DER. The numerator represents the size of the NVM device **100**, e.g., the number of non-parity information bits that it is capable of storing. In combination A, the determined ABC may be approximately 1.96.

[0034] The ABC values for combinations B – G may be determined in a similar manner. The main variables that may be adjusted among the different combinations are the b/c (including the cell quantum and the number of distributions) and the ECC scheme. Providing different values for these main variables and comparing the resulting ABCs may facilitate a determination of a desired NVM design that will accomplish design objectives of a given embodiment.

[0035] In combination B, the number of distributions is increased to 8 and the cell quantum remains at 1 to provide a b/c of 3. In this scenario, the BER becomes 0.02

and a 164 bit ECC value may be needed to achieve the same DER of 10 ppm. This would result in 2028 parity information bits per ECC block, which corresponds to a 49% parity overhead. The calculation system 200 may then determine that the ABC of this scenario is 2.01, which is only slightly larger than the 1.96 ABC of combination A. Thus, the purported gains by moving from 2 b/c to 3 b/c are almost entirely erased by the extra parity information bits needed to achieve the 10 ppm target DER.

[0036] In combination D, the number of distributions may be 7 and the cell quantum may be 1 resulting in a 2.67 b/c. In this scenario, a 100 bit ECC value may be needed to achieve the 10 ppm target DER. This results in a 31% parity overhead and an ABC of 2.03. So, while the nominal bits per cell of combination D is less than combination B, the actual bits per cell is greater.

[0037] A similar effect may be seen when considering combination F. In this combination, the calculation system 200 may determine that an ABC of 2.15 may be achieved using 2.5 b/c, which results from 6 distributions and a cell quantum of 1.

[0038] In this manner, the use of the ABC to factor in the parity costs associated with various parameter combinations may provide a true cost metric by which these combinations may be compared.

[0039] In some embodiments, as alluded to above, the ECC scheme may also be adjusted to provide further efficiencies. For example, while the combinations A, B, D, and F all rely on BCH codes, additional gains may be realized by use of a concatenated code as the ECC scheme. Concatenated coding may be used to construct long, powerful ECCs from short component codes. Combinations C, E, and G use a concatenated coding scheme with a trellis code modulation (TCM) as an inner code and

Reed Solomon (RS) as an outer code. In other embodiments, other concatenated coding schemes may be used.

[0040] **Figure 5** illustrates the NVM device **100** in further detail in accordance with an embodiment in which the ECC scheme is a concatenated coding scheme. The NVM device **100**, and the programmer **104**, in particular, may form a concatenated code by using two encoders: an outer encoder **504** that uses a non-binary code and an inner encoder **508** that uses a binary code. The programmer **104** may also include an interleaver **512** between the two encoders.

[0041] Operation of the programmer **104** may be briefly described in accordance with an embodiment with reference to the flowchart of **Figure 6a**. At block **604**, the programmer **104** may receive input data from a digital source **516**, e.g., a host device. The outer encoder **504** may generate an outer code sequence based at least in part on the input data at block **608**. At block **612**, the inner encoder **508** may generate an inner code sequence based at least in part on the outer code sequence generated by the outer encoder **504**. In some embodiments, as in **Figure 5**, an interleaver, e.g., interleaver **512** may interleave the outer code sequence and the inner encoder **508** may generate the inner code sequence based at least in part on the interleaved outer code sequence.

[0042] The inner code sequence, which may be referred to as "encoded data" in **Figure 5**, may include both parity and non-parity information bits. These parity and non-parity information bits may then be written to parity and non-parity cells, respectively, of the cell matrix **108** at block **616**.

[0043] The reader **112** may include an inner decoder **520**, a de-interleaver **524**, and an outer decoder **528** to operate in a complementary manner to the components of the programmer **104**. Operation of the reader **112** may be briefly described in accordance with an embodiment with reference to the flowchart of **Figure 6b**.

[0044] At block **624**, the reader **112** may receive the inner code sequence, e.g., the encoded data or concatenated code sequence, from the cell matrix **108**. This may be done by reading the parity information bits from the parity cells and reading the non-parity information bits from the non-parity cells. At block **628**, the inner decoder **520** of the reader **112** may decode the inner code sequence to provide an outer code sequence. This may be provided to the outer decoder **528**. In some embodiments, the outer code sequence may be de-interleaved by de-interleaver **524** prior to being provided to the outer decoder **528**. The outer decoder **528** may decode the outer code sequence to provide input data at block **632**. The input data may then be transmitted to a digital sink **532**, e.g., the host device.

[0045] The concatenated coding used in this embodiment may achieve high reliability with reduced complexity. The inner code sequence generated from the inner encoder **508** may be a short TCM sequence that is decoded by the inner decoder **520** with a soft-decision decoding algorithm, e.g., a Viterbi algorithm. The outer code sequence may be longer than the inner code sequence and may be decoded by the outer decoder **528** with an algebraic decoding method, e.g., an RS code.

[0046] The effectiveness of this combination comes from the fact that the inner soft code makes the coding channel, e.g., the programming and reading of values stored in the cell matrix **108**, less noisy. In essence, the inner code, e.g., the TCM

code, does this by combining ordinary rate $R = k / (k+1)$ binary convolutional codes with an M-ary signal constellation ($M = 2^{(k+1)} > 2$) in such a way that coding gain is achieved without increasing the rate at which symbols are transmitted. In other words, no additional parity check cells are needed. Then the outer code, by working on a better channel, becomes very effective in "crunching" all the remaining errors. The effectiveness may be further explained with reference to **Figure 7**.

[0047] **Figure 7** is a graph **700** showing error probabilities of a 512 byte ECC block in accordance with some embodiments. In particular, graph **700** charts signal-to-noise ratios (SNRs) to word error rates (WERs) for signals of varying ECC coding schemes and values. The SNRs may be related to the number of distributions and the distribution widths of the cells of the cell matrix **108**, while the WERs may be directly proportional to the BERs.

[0048] The vertical dashed line **704** represents a first SNR; the vertical dashed line **708** represents a second SNR; and the horizontal dashed line **712** is an error rate of the ECC block to be reached for achieving a target DER of 10 ppm. Lines **716**, **720**, **724**, **728**, **732**, **736**, and **740** respectively correspond to ECC values 164 b, 80 b, 66 b, 40 b, 24 b, 12 b, and 0 b.

[0049] Given the SNR of the vertical dashed line **704**, which may correspond to an embodiment having 8 distributions for a cell, an ECC value of 164 b (represented by line **716**) may be used to achieve the target DER. Thus, an embodiment using a standard BCH code would need an ECC value of 164 b as discussed above with respect to combination B.

[0050] However, when using convolutional codes, an inner code, e.g., a TCM code, may provide an SNR gain of approximately 2 dB on the coding channel. Thus, the outer code effectively acts on a coding channel with an SNR relating to the vertical dashed line **708** as opposed to the vertical dashed line **704**. This SNR provides the outer decoder **528** with the flexibility of using the 66 b code (represented by line **724**).

[0051] Lines **744** and **748** respectively show how the BER changes per SNR for an uncoded signal and a convolutionally coded signal. Line **744** represents a real channel condition and line **748** represents a channel conditions seen by a BCH code after application of convolutional codes. As can be seen, the convolutionally coded signal is associated with lower BERs over the range of SNRs.

[0052] Referring again to the table **400** of **Figure 4**, the combinations that use convolutional codes as an ECC scheme, e.g., combinations C, E, and G, all are associated with ABC values that are greater than 2.2, while the ABC values associated with the combinations that only use BCH codes are all less than 2.2.

[0053] In some embodiments, referring again to **Figure 2**, the system **200** may include a selector **220** that receives parameter combinations and associated ABC values and selects, based on the ABC values, desired parameters. It may be that the main variables of the parameter combinations may be ECC schemes and nominal bits per cell. Thus, these parameters and/or the underlying parameters that have an effect on these parameters (e.g., # of distributions and cell quantum) may be the selected parameters of these embodiments.

[0054] **Figure 8** illustrates a host device **800** that may host the NVM device **100** in accordance with some embodiments. The host device **800** may include one or more

processors **804**; system control logic **808** coupled to at least one of the processor(s) **804**; system memory **812** coupled to the system control logic **808**; the NVM device **100** coupled to the system control logic **808**; and one or more communication interface(s) **820** coupled to the system control logic **808**.

[0055] System control logic **808** for one embodiment may include any suitable interface controllers to provide for any suitable interface to the components with which it is coupled.

[0056] System memory **812** may be used to load and/or store data/instructions, for example, for the host device **800**. System memory **812** may include any suitable volatile memory, such as, but not limited to, suitable dynamic random access memory (DRAM).

[0057] The NVM device **100** may also be used to load and/or store data/instructions, for example, for the host device **800**. The NVM device **100** may include any suitable nonvolatile memory, such as, but not limited to, NOR flash memory, NAND flash memory, phase change memory, etc.

[0058] In some embodiments, logic may include instructions **824** that when executed by the processor(s) **804** result in the host device **800** and/or the NVM device **100** performing at least some of the programming, reading, and/or calculating operations described herein. The instructions may be located in the NVM device **100** and/or the system memory **812**. In some embodiments, the instructions **824** may additionally/alternatively be located in the system control logic **808**.

[0059] In some embodiments, the host device **800** may be used to implement the modules of calculation system **200** and not the programming/reading operations of the

NVM device **100**, itself. In these embodiments, the NVM device **100** may not be included in the host device **800**.

[0060] Communication interface(s) **820** may provide an interface for the host device **800** to communicate over one or more networks and/or with any other suitable device. Communication interface(s) **820** may include any suitable hardware and/or firmware. Communication interface(s) **820** for one embodiment may include, for example, a network adapter, a wireless network adapter, a telephone modem, and/or a wireless modem. For wireless communications, communication interface(s) **820** for one embodiment may use one or more antennas.

[0061] For one embodiment, at least one of the processor(s) **804** may be packaged together with logic for one or more controllers of system control logic **808**. For one embodiment, at least one processor of the processor(s) **804** may be packaged together with logic for one or more controllers of system control logic **808** to form a System in Package (SiP). For one embodiment, at least one processor of the processor(s) **804** may be integrated on the same die with logic for one or more controllers of system control logic **808**. For one embodiment, at least one processor of the processor(s) **804** may be integrated on the same die with logic for one or more controllers of system control logic **808** to form a System on Chip (SoC).

[0062] In various embodiments, the host device **800** may be a desktop or laptop computer, a server, a set-top box, a digital recorder, a game console, a personal digital assistant, a mobile phone, a digital media player, a digital camera, etc. The host device **800** may have more or less components and/or different architectures.

[0063] Although certain embodiments have been illustrated and described herein for purposes of description of the preferred embodiment, it will be appreciated by those of ordinary skill in the art that a wide variety of alternate and/or equivalent embodiments or implementations calculated to achieve the same purposes may be substituted for the embodiments shown and described without departing from the scope of the present disclosure. Similarly, memory devices of the present disclosure may be employed in host devices having other architectures. This application is intended to cover any adaptations or variations of the embodiments discussed herein. Therefore, it is manifestly intended that embodiments in accordance with the present disclosure be limited only by the claims and the equivalents thereof.

Claims

1. A method for writing input data to a nonvolatile memory device comprising:
receiving input data;
generating, with a concatenated coding scheme including an outer encoder, an outer code sequence based at least in part on the input data;
generating, with the concatenated coding scheme including an inner encoder, an inner code sequence based at least in part on the outer code sequence, the inner code sequence including a first plurality of parity information bits and a second plurality of non-parity information bits;
writing the first plurality of parity information bits to a third plurality of parity cells of a nonvolatile memory device; and
writing the second plurality of non-parity information bits to a fourth plurality of non-parity cells of the nonvolatile memory device.
2. The method of claim 1, further comprising:
using a non-binary code in said generating of the outer code sequence; and
using a binary code in said generating of the inner code sequence.
3. The method of claim 2, wherein the non-binary code is a Reed Solomon code and the binary code is a trellis code modulation code.
4. The method of claim 1, further comprising:
writing a nominal number of bits per cell of greater than two for the third plurality of parity cells and the fourth plurality of non-parity cells; and
using fifteen percent or less of a total number of cells of the nonvolatile memory device as parity cells.
5. The method of claim 1, wherein the nonvolatile memory device is capable of storing a first number of non-parity information bits and the first number divided by a sum of a second number of parity cells of the nonvolatile memory device and a third number of non-parity cells of the nonvolatile memory device is greater than 2.2.
6. The method of claim 5, wherein each of the second number of parity cells and the third number of non-parity cells includes eight or more distributions.

7. A method for providing input data extracted from a nonvolatile memory device comprising:

reading a first plurality of parity information bits from a second plurality of parity cells of a nonvolatile memory device;

reading a third plurality of non-parity information bits from a fourth plurality of non-parity cells of the nonvolatile memory device, the first plurality of parity information bits and the third plurality of non-parity information bits corresponding to an inner code sequence;

decoding, with a concatenated coding scheme including an inner decoder, the inner code sequence to provide an outer code sequence; and

decoding, with the concatenated coding scheme including an outer decoder, the outer code sequence to provide input data.

8. The method of claim 7, further comprising:

deinterleaving the outer code sequence prior to said decoding of the outer code sequence.

9. The method of claim 7, further comprising:

using a binary code in said decoding of the inner code sequence; and

using a non-binary code in said decoding of the outer code sequence.

10. A nonvolatile memory device comprising:

a matrix of nonvolatile memory cells; and

a programmer coupled to the matrix of nonvolatile memory cells, the programmer configured to receive input data and to generate a concatenated code sequence based at least in part on the input data, the concatenated code sequence including a first plurality of parity information bits and a second plurality of non-parity information bits, the programmer further configured to write the first plurality of parity information bits to a third plurality of parity cells and to write the second plurality of non-parity information bits to a fourth plurality of non-parity cells.

11. The nonvolatile memory device of claim 10, wherein the programmer comprises:

an outer encoder configured to receive the input data and to generate an outer code sequence based at least in part on the input data; and

an inner encoder configured to generate the concatenated code sequence based at least in part on the outer code sequence.

12. The nonvolatile memory device of claim 11, wherein the outer encoder is further configured to use a non-binary code to generate the outer code sequence, and the inner encoder is further configured to use a binary code to generate the inner code sequence.

13. The nonvolatile memory device of claim 10, further comprising:
an inner decoder configured to decode the concatenated code sequence to provide an outer code sequence; and
an outer decoder configured to decode the outer code sequence to provide the input data.

14. The nonvolatile memory device of claim 10, wherein the nonvolatile memory device is an integrated circuit.