



- (51) **International Patent Classification:**
G11C 16/08 (2006.01) *G11C 16/34* (2006.01)
- (21) **International Application Number:**
PCT/US2016/017228
- (22) **International Filing Date:**
10 February 2016 (10.02.2016)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
14/671,140 27 March 2015 (27.03.2015) US
- (71) **Applicant:** INTEL CORPORATION [US/US]; 2200 Mission College Blvd., Santa Clara, CA 95054 (US).
- (72) **Inventors:** MOTWANI, Ravi H.; 12047 Alta Carmel Ct, Unit 195, San Diego, CA 92128 (US). KALAVADE, Pranav; 6385 Slida Drive, San Jose, CA 95129 (US).
- (74) **Agent:** MUGHAL, Usman A.; Green, Howard, & Mughal, LLP, c/o CPA Global, P.O. Box 52050, Minneapolis, Minnesota 55402 (US).
- (81) **Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM,

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

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

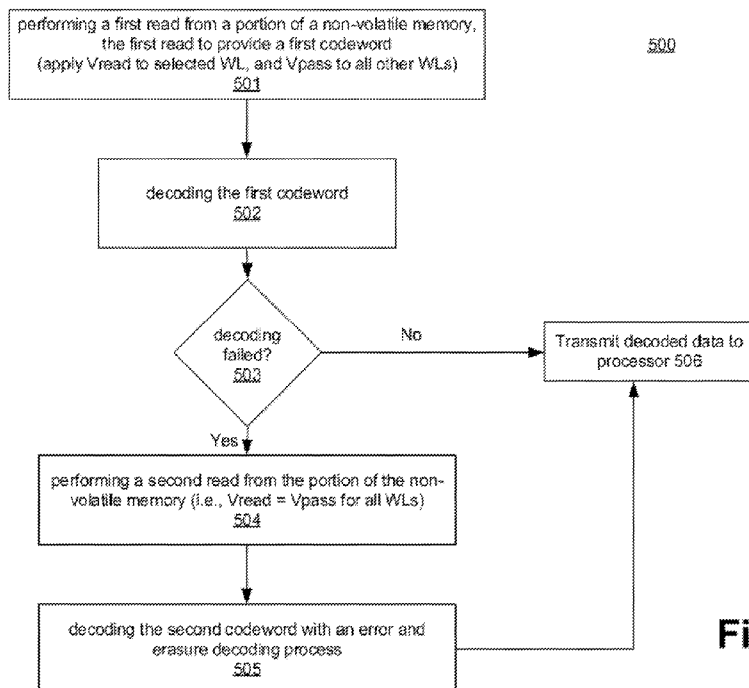
Declarations under Rule 4.17:

— of inventorship (Rule 4.17(iv))

Published:

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

- (54) **Title:** APPARATUS AND METHOD FOR DETECTING AND MITIGATING BIT-LINE OPENS IN FLASH MEMORY

**Fig. 5**

(57) **Abstract:** Described is a method which comprises performing a first read from a portion of a non-volatile memory, the first read to provide a first codeword; decoding the first codeword; determining whether the decoding operation failed; performing a second read from the portion of the non-volatile memory when it is determined that the decoding operation failed, the second read to provide a second codeword; and decoding the second codeword with an errors-and-erasures decoding process.

APPARATUS AND METHOD FOR DETECTING AND MITIGATING BIT-LINE OPENS IN FLASH MEMORY

CLAIM FOR PRIORITY

[0001] This application claims priority to U.S. Provisional Patent Application Serial No. 14/671,140, filed on 27 March 2015, titled “APPARATUS AND METHOD FOR DETECTING AND MITIGATING BIT-LINE OPENS IN FLASH MEMORY,” and which is incorporated by reference in entirety.

BACKGROUND

[0002] Defects in the form of open lines or shorts are prevalent in non-volatile memories such as NAND flash memories. These defects can be present at the time of manufacture, or can be generated through the operating life of the memories. The defects can also be caused by regular wear and tear (e.g., periodic refresh operations over a long time). These defects contribute towards Raw Bit Error rate (RBER). The term “RBER” here generally refers to the rate of errors when reading from a non-volatile memory. If Low-Density Parity-Check Code (LDPC) is used to encode input data which is stored in the NAND flash memory, and when a soft read operation is performed, the open lines or circuits (generally referred here as “opens”) in the read memory may manifest as errors with high confidence (or errors with high reliability value, where reliability is the magnitude of the Log Likelihood Ratio (LLR)).

Having open circuits or open lines means that the memory cell cannot be used to store data. These high confidence errors are more detrimental to LDPC performance than soft errors. One reason for such detriment is that soft errors can allow LDPC decoding to continue (and the decoded result has a high chance of being corrected by error correction schemes) while hard (or high confidence errors) may cause LDPC decoding to fail with low or no chance of being corrected by the error correction schemes.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] The embodiments of the disclosure will be understood more fully from the detailed description given below and from the accompanying drawings of various embodiments of the disclosure, which, however, should not be taken to limit the disclosure to the specific embodiments, but are for explanation and understanding only.

[0004] **Fig. 1** illustrates an architecture with an apparatus to detect and mitigate defects caused by bit-line opens in a non-volatile memory, according to some embodiments of the disclosure.

[0005] **Fig. 2** illustrates a block of non-volatile NAND flash memory with an open associated with one of the memory cells.

[0006] **Fig. 3** illustrates a plot showing distribution of cells for a two threshold level (e.g., 2-level, 1bit/cell) NAND memory cell.

[0007] **Fig. 4** illustrates a plot showing distribution of cells for a multi-threshold level (e.g., 4-levels, 2bits/cell) NAND memory cell.

[0008] **Fig. 5** illustrates a flowchart of a method for detecting and mitigating defects caused by bit-line opens in a non-volatile memory, according to some embodiments of the disclosure.

[0009] **Fig. 6** illustrates a plot comparing defect probabilities versus soft Raw Bit Error rate (RBER) for Low-Density Parity-Check Code (LDPC) decoders that are informed of the open and for LDPC decoders without the information about the open, according to some embodiments.

[0010] **Fig. 7** illustrates a smart device or a computer system or a SoC (System-on-Chip) with apparatus to detect and mitigate defects caused by bit-line opens in a non-volatile memory, according to some embodiments.

DETAILED DESCRIPTION

[0011] Various embodiments describe a method and apparatus to handle (i.e., detect and mitigate) opens in a non-volatile memory gracefully by maintaining high error correction capability by appropriately tailoring decoding algorithms and media (e.g., logic and registers) providing additional information. In some embodiments, errors caused by opens are converted to erasures by logic and registers providing real-time information on opens, and by a decoding mechanism using this information to identify opens as erasures. Converting errors to erasures significantly improves error correction capability, which reduces the codeword failure probability.

[0012] The term “codeword failure probability” here generally refers to the probability that an encoded codeword would fail decoding. An error caused by an open appears as high confidence incorrect information which is worse than a soft error which is usually wrong

information with low confidence (of being wrong). It will be understood that soft errors are a type of transient errors that may be those caused by cosmic radiation or some other anomaly or environmental condition that randomly affects bits. Such soft errors are typically limited to a single bit or a column. Such soft errors may occur during one read and not a subsequent read. In contrast, persistent errors or hard errors or hard failures are those caused by a device defect. An error caused by an open appears as high confidence incorrect information. The high confidence error has a higher chance of causing the decoding algorithm to fail. As such, converting errors to erasures reduces the probability of codeword failure in accordance to some embodiments. Various embodiments described here process hard errors in the form of opens instead of soft errors.

[0013] An erasure is generally defined as a location in the codeword (which refers to a location in the memory array) whose bit value is lost (i.e., no bit information exists about that location and so it is referred to as an erasure). The erasures are received by a decoder (e.g., a Low-Density Parity-Check Code (LDPC) decoder) which applies a decoding algorithm with knowledge about the erasures (i.e., locations in the codeword with lost bit values). The decoding algorithm, when armed with information about erasures instead of errors, has a higher chance of continuing to operate. For example, decoding algorithm can work with twice as many erasures than errors for error correction algebraic codes like Reed-Solomon. The output of the decoder is then corrected by a correction logic. Conversely, errors may cause the decoding algorithm to fail, and as such the correction logic may not correct the read data (or codeword).

[0014] Here, the term “read voltage” or “Vread” generally refers to a threshold voltage applied to a gate terminal of the selected memory cell to turn it on and to determine the state of that selected cell. Here the term “pass voltage” or “Vpass” generally refers to a threshold voltage applied to all word-lines (i.e., gate terminals of all un-selected cells in a string of memory cells) to cause the memory cells to be turned on so as to pass down their values to a ground node.

[0015] In some embodiments, a method and apparatus is provided in which a read logic performs a first read operation from a portion of a non-volatile memory (e.g., a block in an array of NAND flash memory), where the first read provides a first codeword (e.g., a LDPC). In some embodiments, the first read operation is performed by applying a read voltage (i.e., Vread) to a

selected word-line (WL) of the portion of the non-volatile memory, and then by applying a pass voltage (i.e., V_{pass}) to all other word-lines in the portion of the non-volatile memory.

[0016] In some embodiments, a decoder is used for decoding the first codeword. In some embodiments, a logic is provided which determines whether the decoding operation failed. For example, if the number of soft errors obtained from the decoding process are above a threshold level then the decoding operation is considered to have failed. In some embodiments, when the logic determines that the decoding operation failed, the logic requests the read logic to perform a second read operation. In some embodiments, the second read operation is performed by applying the pass voltage to the selected word-line (i.e., now all word-lines in the block (i.e., portion) of the non-volatile memory receive the pass voltage). The second read operation provides a second codeword. In some embodiments, the second codeword ascertains a location of an open circuit. In some embodiments, the location of the open circuit is stored in a register so that the decoder can use that information when performing the errors-and-erasures decoding process. In some embodiments, the errors-and-erasures decoding process flags the open circuit as erasures.

[0017] There are many technical effects of various embodiments. The various embodiments permit detection of these defects (i.e., opens) with one extra read command. In some embodiments, this extra read command is the only extra read command. This detection of opens leads to higher endurance. For example, typically, at a defect probability of $2e^{-3}$ (i.e., 10-20% of total error handling capability), various embodiments result in a 2x reduction in RBER by identifying opens as erasures. As such, converting errors to erasures significantly improves error correction capability which reduces codeword failure probability.

[0018] In the following description, numerous details are discussed to provide a more thorough explanation of embodiments of the present disclosure. It will be apparent, however, to one skilled in the art, that embodiments of the present disclosure may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form, rather than in detail, in order to avoid obscuring embodiments of the present disclosure.

[0019] Note that in the corresponding drawings of the embodiments, signals are represented with lines. Some lines may be thicker, to indicate more constituent signal paths, and/or have arrows at one or more ends, to indicate primary information flow direction. Such

indications are not intended to be limiting. Rather, the lines are used in connection with one or more exemplary embodiments to facilitate easier understanding of a circuit or a logical unit. Any represented signal, as dictated by design needs or preferences, may actually comprise one or more signals that may travel in either direction and may be implemented with any suitable type of signal scheme.

[0020] Throughout the specification, and in the claims, the term "connected" means a direct electrical or wireless connection between the things that are connected, without any intermediary devices. The term "coupled" means either a direct electrical, physical, or wireless connection between the things that are connected or an indirect electrical, physical, or wireless connection through one or more passive or active intermediary devices. The term "signal" means at least one current signal, voltage signal or data/clock signal. The meaning of "a," "an," and "the" include plural references. The meaning of "in" includes "in" and "on."

[0021] The terms "substantially," "close," "approximately," "near," and "about," generally refer to being within +/- 20% of a target value. Unless otherwise specified the use of the ordinal adjectives "first," "second," and "third," etc., to describe a common object, merely indicate that different instances of like objects are being referred to, and are not intended to imply that the objects so described must be in a given sequence, either temporally, spatially, in ranking or in any other manner.

[0022] For the purposes of the present disclosure, phrases "A and/or B" and "A or B" mean (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).

[0023] **Fig. 1** illustrates architecture 100 with an apparatus to detect and mitigate defects caused by bit-line opens in a non-volatile memory, according to some embodiments of the disclosure. In some embodiments, architecture 100 includes storage unit 101 and Processor 102. In some embodiments, storage unit 101 is a Solid State Drive (SSD). In other embodiments, other types of storage units may be used. In some embodiments, Processor 102 is a microprocessor (such as those designed by Intel Corporation of Santa Clara, California), Digital Signal Processors (DSPs), Field-Programmable Gate Arrays (FPGAs), Application Specific Integrated Circuits (ASICs), or Radio-Frequency Integrated Circuits (RFICs), etc.

[0024] So as not to obscure the embodiments, a simplified version of memory architecture 100 is shown. A person skilled in the art would appreciate that there are other logic

and circuits needed for complete operation of architecture 100. For example, syndrome calculators, queues, input-output buffers, multiplexers, sense amplifiers, etc., are not shown.

[0025] In some embodiments, SSD 101 includes I/O interface 103, Memory Controller 104, and a plurality of memory banks logically combined as Memory Block 105. In some embodiments, I/O interface 103 is a Serial Advanced Technology Attachment (SATA) interface and interconnect 106 is a SATA compliant bus coupling SSD 101 to Processor 102. A person skilled in the art would appreciate that in some cases, Processor 102 is indirectly coupled to SSD 101 via a bus controller (not shown). In other embodiments, other types of I/O interfaces may be used for I/O interface 103. For example, Serial Attached Small Computer System Interface (SCSI) (or simply SAS) may be used for I/O interface 103, and interconnect 106 is a SAS compliant interface; or Peripheral Component Interconnect Express (PCIe) as described in the PCI Express Base 3.0 Specification may be used for I/O interface 103.

[0026] While the embodiments of **Fig. 1** are illustrated with two distinct components in SSD 101 and Processor 102, in some embodiments, storage unit 101 and Processor 102 can be packaged together as a single unit. In some embodiments, storage unit 101 and Processor 102 are implemented using three dimensional integrated circuit (3D IC) technology where various dies are stacked on each other. For example, various dies or components of SSD 101 may be implemented as dies that are stacked on a die of Processor 102 to form a stacked or 3D IC.

[0027] Here, memory banks (i.e., Memory bank-1 to Memory bank-(N+1), where ‘N’ is an integer) in Memory Block 105 are shown as a group of memory banks in one area. The terms “die” and “bank” here are interchangeably used and generally refer to an integrated circuit that may be packaged separately or integrated with other circuits. In some embodiments, the memory banks (or die 1 to die (N+1)) may be distributed in SSD 101. In some embodiments, each memory bank is a non-volatile memory. For example, each memory bank is a single or multi-threshold level NAND flash memory (as described with reference to **Figs. 3-4**), NOR flash memory, single or multi-level Phase Change Memory (PCM), a three dimensional cross point memory, a resistive memory, nanowire memory, ferro-electric transistor random access memory (FeTRAM), magnetoresistive random access memory (MRAM) memory that incorporates memristor technology, spin transfer torque (STT)-MRAM, or a combination of any of the above, etc.

[0028] In some embodiments, Memory Controller 104 includes Encoder 104a, Write Logic 104b, Read Logic 104c, Decoder 104d, Error Logic 104e, and Correction Logic 104f. In some embodiments, Encoder 104a applies an error correction scheme to the input data (data_in) and generates encoded data (i.e., codewords). Each codeword includes a data portion and a corresponding parity portion. In some embodiments, the error correction scheme is one of LDPC scheme, or a Reed-Solomon code scheme. In other embodiments, other types of error correction schemes may be used, such as, Cyclic Redundancy Check (CRC) scheme, Hamming Code scheme, or Convolutional code scheme.

[0029] In some embodiments, the encoded codeword is written to Memory Block 105 by Write Logic 104b. In some embodiments, Write Logic 104b includes any suitable driver to drive data for writing to Memory Block 105. In some embodiments, the encoded codewords are distributed to memory banks (i.e., Memory bank-1 through Memory bank-N). In some embodiments, Memory Controller 104 communicates with Memory Block 105 (i.e., the various memory dies) via an Input/Output (I/O) interface and corresponding compliant bus. In some embodiments, Memory Controller 104 communicates with Memory Block 105 via an interface which is compliant to the Open NAND Flash Interface (ONFI) specification (e.g., Revision 4.0 released April 02, 2014). In other embodiments, other types of interfaces may be used for communicating between Memory Controller 104 and Memory Block 105.

[0030] So as not to obscure various embodiments, Memory Block 105 is described as a NAND flash memory, Encoder 104a is described as an LDPC encoder, and Decoder 104d is described as a LDPC decoder. However, other types of suitable non-volatile memories, encoders, and decoders may be used. In some embodiments, Read Logic 104c is operable to read the encoded data from the one or more Memory banks from Memory Block 105. Read Logic 104c may be implemented using any suitable read circuits.

[0031] In some embodiments, Read Logic 104c performs a first read operation from a block in an array of NAND flash memory, where the result of the first read operation provides a first codeword (i.e., an LDPC codeword). In some embodiments, the first read operation is performed by applying a read voltage (Vread) to a selected word-line of the portion of the NAND memory (e.g., Memory bank-1), and then by applying a pass voltage (Vpass) to all other word-lines (i.e., to unselected cells) in the portion of the NAND memory as described with reference to **Fig. 2**.

[0032] Referring back to **Fig. 1**, in some embodiments, Decoder 104c decodes the first codeword by applying an error correction scheme (e.g., LDPC scheme) to decode the codewords encoded by Encoder 104a. In some embodiments, Error Logic 104e determines whether the decoding operation failed. For example, if the number of soft errors obtained from the decoding process (executed by Decoder 104d) are above a threshold level then the decoding operation is considered to have failed. In some embodiments, when Error Logic 104e determines that the decoding process has failed, Error Logic 104e requests Read Logic 104c to perform a second read.

[0033] Continuing with this example, if the number of soft errors obtained from the decoding process (executed by Decoder 104d) for the first read operation are equal to or below the threshold level, then the decoded codeword is processed by Correction Logic 104f. In some embodiments, the threshold level is programmable. In other embodiments, the threshold level is predetermined at the time of manufacture.

[0034] In some embodiments, the second read operation is performed by applying the pass voltage to the selected word-line (i.e., now all word-lines in the block of NAND memory receive the pass voltage). The result from the second read operation provides a second codeword. In some embodiments, the second codeword ascertains a location of an open circuit because the output of selected NAND string does not change even when the selected word-line is biased with the pass voltage. In some embodiments, this output of the selected NAND string is an output of a sense amplifier (not shown) coupled to the NAND flash memory.

[0035] In some embodiments, the location of the open circuit is stored in a register so that Decoder 104d can use that information when performing the errors-and-erasures decoding process which flags the open circuit as an erasure (instead of an error). In some embodiments, since the open bit-lines are block dependent and not word-line dependent, the locations for the opens can be stored for each block of the NAND flash memory. For example, the locations of the opens can be stored in registers outside of the NAND flash memory arrays. In some embodiments, once the locations of the opens are known, no extra sensing at V_{pass} is used. One technical effect of saving the extra sensing is that read latency is saved. In some embodiments, errors detected while reading the memory block are converted into erasures which allows for high correction probability. Erasures can be handled by LDPC Decoder 104d far better than errors. As such, the performance of LDPC Decoder 104d improves.

[0036] In some embodiments, a periodic update of open detection is performed and locations of the opens are detected and stored regularly. In some embodiments, open detection is performed before or after a refresh operation (i.e., background data refresh operation). For example, when data in the NAND flash memory is about to be routinely refreshed, the process of detecting opens is performed and locations of the detected opens are stored in registers. In some embodiments, during refresh two read operations are performed—the first read operation (i.e., the normal read for a word-line) followed by the second read operation (i.e., applying V_{pass} for that word-line). As such, opens are identified for an entire block of memory which is being refreshed. In some embodiments, Correction Logic 104f corrects faults in the one or more memory banks (i.e., Memory banks-1 to Memory banks-N) using the decoded codeword (or part of it) obtained from the second read operation. The corrected output is then provided as output data `data_out` for processing by Processor 102.

[0037] **Fig. 2** illustrates a block of non-volatile NAND flash memory 200 with an open associated with one of the memory cells. It is pointed out that those elements of **Fig. 2** having the same reference numbers (or names) as the elements of any other figure can operate or function in any manner similar to that described, but are not limited to such.

[0038] NAND flash memory block 200 comprises strings of floating gate transistors (or cells), such that the transistors are coupled in series. The gate terminal of the first transistor in each string is coupled to a bit-line (e.g., $BL_{(0)-(n)}$) and controllable by a bit-line select signal. The last transistor in each string is coupled to ground (or a source-line) and controllable by a source line select signal (or ground select signal). The remaining transistors in the string are coupled to word-lines (e.g., $WL_{(0)-(n)}$).

[0039] In the absence of any defects, to read a value from a memory cell in a string, the bit-line select and ground select signals are asserted and one of the BLs is selected (e.g., $BL_{(0)}$). To select the memory cell of interest in the enabled or selected string, WL for that cell is set to a read voltage (V_{read}) and the remaining WLs are set to pass voltage (V_{pass}). The pass voltage is the voltage which is high enough to allow the memory cell (i.e., transistor) to pass its value down to ground, while the read voltage is the threshold voltage for the memory cell. Depending on the charge stored in the memory cell of interest and the read voltage, the BL is discharged or retains a certain charge. The BL is then sensed by Sense Amplifier 202 and the stored value (or state) in the selected memory cell is determined.

[0040] In the case of a defect, such as an open circuit in one of the memory cells of the selected string, BL is not discharged to ground. In this example, memory cell 201 has an open circuit (i.e., an open line). This open line can generate soft errors with high confidence during LDPC decoding. Various embodiments described here identify the open, the location of the open, and mitigate its effects by converting the error into an erasure, which is easier to handle by Decoder 104d when it applies an errors-and-erasures decoding process. The various embodiments are applicable to a variety of types of memory cells.

[0041] For example, some embodiments are applicable to NAND memories with Single Level Cells (SLCs), in which each memory cell stores two states (i.e., 1 bit programmed/read per memory cell). Here, the term “state” and threshold “level” are interchangeably used. **Fig. 3** illustrates plot 300 showing distribution of cells for a two threshold level (i.e., 2-level, 1bit/cell) NAND memory cell. Here, x-axis is threshold (i.e., programming voltage) and y-axis represents normalized number of memory cells. Vread is the read voltage that determines the state stored in the memory cell. When the voltage applied to the gate of the selected memory cell is Vread, then L0 (i.e., level zero) is read from the selected cell. To read L1 (i.e., level one), the voltage on Vread is increased until it is greater than the programmed threshold L1 for the cell.

[0042] In some embodiments, Vread is set by Read Logic 104b. In some embodiments, during the first read operation, the selected memory cell is biased by Vread. Depending on the value programmed in the memory cell, the Vread voltage is set to be above that threshold level to read from that memory cell. In some embodiments, during the second read operation, the selected memory cell is biased by Vpass instead of Vread. The second read ascertains whether there is an open in the selected memory cell or associated with the selected memory cell in accordance to some embodiments. Here, the erasure is the location of the memory cell whose data value is lost because of an open. In some embodiments, Decoder 104d applies an errors-and-erasures decoding process on the second codeword (which is obtained as a result of the second read operation). As such, Decoder 104d has a higher chance to decode the second codeword. During, errors-and-erasures decoding process for LDPC, the LLR for the erasures is set to zero. As such, an iterative decoding of LDPC codes is configured to decode errors and erasures.

[0043] Some embodiments are also applicable to NAND memories with Multi-level Cells (MLCs), in which each memory cell stores multiple states. For example, for a 4-level

MLC NAND, four states (i.e., 2 bits programmed/read per memory cell) are stored in each memory cell, for 8-level MLC NAND, eight states (i.e., 3 bits programmed/read per memory cell) are stored in each memory cell, and so on. For 3 bits/cell to 4 bits/cell, individual memory cells are programmed to 8 or 16 distinct levels, respectively.

[0044] **Fig. 4** illustrates plot 400 showing distribution of memory cells for a multi-threshold level (i.e., 4-levels, 2bits/cell) NAND memory cell. Here, x-axis is threshold (i.e., programming voltage) and y-axis represents normalized number of memory cells. For MLC, multiple Vread levels are used to read the state stored in the MLC memory cell. In this example, four levels (i.e., L0, L1, L2, and L3) can be read by three different threshold levels of Vread—Vread₀, Vread₁, and Vread₂.

[0045] In some embodiments, Vread₀₋₂ is set by Read Logic 104b. In some embodiments, during the first read operation, the selected memory cell (e.g., memory cell 201) is biased by Vread. Depending on the value programmed in the memory cell, Vread is set to be above that threshold level to read the state from that memory cell. For example, Vread₂ is set to determine whether the selected memory cell stores L4 (i.e., one) or zero. In some embodiments, during the second read operation, the selected memory cell is biased by Vpass₂ instead of Vread₀ or Vread₁. The second read ascertains whether there is an open in the selected memory cell or associated with the selected memory cell. In some embodiments, Decoder 104d applies an errors-and-erasures decoding process on the second codeword (obtained as a result of the second read operation). As such, Decoder 104d has a higher chance to decode the second codeword.

[0046] **Fig. 5** illustrates 500 flowchart of a method for mitigating defects caused by bit-line opens in a non-volatile memory, according to some embodiments of the disclosure. It is pointed out that those elements of **Fig. 5** having the same reference numbers (or names) as the elements of any other figure can operate or function in any manner similar to that described, but are not limited to such.

[0047] Although the blocks in the flowchart with reference to **Fig. 5** are shown in a particular order, the order of the actions can be modified. Thus, the illustrated embodiments can be performed in a different order, and some actions/blocks may be performed in parallel. Some of the blocks and/or operations listed in **Fig. 5** are optional in accordance with certain embodiments. The numbering of the blocks presented is for the sake of clarity and is not

intended to prescribe an order of operations in which the various blocks must occur.

Additionally, operations from the various flows may be utilized in a variety of combinations.

[0048] At block 501, a first read operation is performed by Read Logic 104b from a portion (i.e., block) of a non-volatile memory 105 (e.g., NAND flash memory). The first read provides a first codeword by applying Vread to the selected WL and Vpass to all other WLs. At block 502, Decoder 104d decodes the first codeword. For example, LDPC Decoder 104d applies an LDPC decoding scheme to decode the first codeword. At block 503, Error Logic 104e determines whether the decoding process failed. If a number of soft errors is greater than a threshold, then the decoding fails. If the decoding fails, the process proceeds to block 504, or else it proceeds to process to block 506. At block 506, the decoded codeword is transmitted to processor 102 for further processing.

[0049] At block 504, a second read operation is performed by Read Logic 104b from the portion of the non-volatile memory. The second read operation provides a second codeword by applying Vpass to the selected WL, where Vpass is provided to all WLs. By applying Vpass during the second read operation, the location of the open is determined. In some embodiments, the location of the open is stored in a register and passed on to Decoder 104d. The erasure is defined as the location of the open (i.e., location of the memory cell whose bit is lost).

[0050] At block 505, Decoder 104d performs errors-and-erasures decoding process on the second codeword (obtained as a result of the second read operation), and then provides the decoded codeword to processor 102 as illustrated by block 506. In some embodiments, a periodic update of open detection is performed and locations of the opens are detected and stored regularly. In some embodiments, open detection is performed during refresh operation (i.e., background data refresh operation).

[0051] **Fig. 6** illustrates plot 600 comparing defect probabilities versus soft RBER for LDPC decoders that are informed of the open and for LDPC decoders without the information about the open, according to some embodiments. Here, x-axis is soft RBER and y-axis is defect probability as Misplacement (i.e., ratio of hard RBER to Erasure RBER).

[0052] Plot 600 illustrates LDPC performance with misplacement marked as erasures. Waveform 601 illustrates misplacement with default Log-likelihood ratios (LLRs). In this example, Waveform 602 illustrates misplacement with optimal LLRs. Waveform 603 illustrates erasures with optimal LLRs. Waveform 604 illustrates erasures with LLR as 0. In this example,

comparing waveforms 603/604 and 601/602 shows that LDPC decoding continues even when erasures are 1.5 times more than hard errors. By converting errors into erasures, Decoder 104d can apply errors-and-erasures decoding process and has a higher chance to continue to decode.

[0053] Fig. 7 illustrates a smart device or a computer system or a SoC (System-on-Chip) with apparatus to detect and mitigate defects caused by bit-line opens in a non-volatile memory, according to some embodiments. It is pointed out that those elements of Fig. 7 having the same reference numbers (or names) as the elements of any other figure can operate or function in any manner similar to that described, but are not limited to such.

[0054] Fig. 7 illustrates a block diagram of an embodiment of a mobile device in which flat surface interface connectors could be used. In some embodiments, computing device 1600 represents a mobile computing device, such as a computing tablet, a mobile phone or smart-phone, a wireless-enabled e-reader, or other wireless mobile device. It will be understood that certain components are shown generally, and not all components of such a device are shown in computing device 1600.

[0055] In some embodiments, computing device 1600 includes a first processor 1610 with apparatus of memory controller 104 to detect and mitigate defects caused by bit-line opens in a non-volatile memory, according to some embodiments discussed. Other blocks of the computing device 1600 may also include apparatus of memory controller 104 to detect and mitigate defects caused by bit-line opens in a non-volatile memory, according to some embodiments. The various embodiments of the present disclosure may also comprise a network interface within 1670 such as a wireless interface so that a system embodiment may be incorporated into a wireless device, for example, cell phone or personal digital assistant.

[0056] In some embodiments, processor 1610 (and/or processor 1690) can include one or more physical devices, such as microprocessors, application processors, microcontrollers, programmable logic devices, or other processing means. The processing operations performed by processor 1610 include the execution of an operating platform or operating system on which applications and/or device functions are executed. The processing operations include operations related to I/O (input/output) with a human user or with other devices, operations related to power management, and/or operations related to connecting the computing device 1600 to another device. The processing operations may also include operations related to audio I/O and/or display I/O.

[0057] In some embodiments, computing device 1600 includes audio subsystem 1620, which represents hardware (e.g., audio hardware and audio circuits) and software (e.g., drivers, codecs) components associated with providing audio functions to the computing device. Audio functions can include speaker and/or headphone output, as well as microphone input. Devices for such functions can be integrated into computing device 1600, or connected to the computing device 1600. In one embodiment, a user interacts with the computing device 1600 by providing audio commands that are received and processed by processor 1610.

[0058] In some embodiments, computing device 1600 includes Display subsystem 1630. Display subsystem 1630 represents hardware (e.g., display devices) and software (e.g., drivers) components that provide a visual and/or tactile display for a user to interact with the computing device 1600. Display subsystem 1630 includes display interface 1632, which includes the particular screen or hardware device used to provide a display to a user. In one embodiment, display interface 1632 includes logic separate from processor 1610 to perform at least some processing related to the display. In one embodiment, display subsystem 1630 includes a touch screen (or touch pad) device that provides both output and input to a user.

[0059] In some embodiments, computing device 1600 includes I/O controller 1640. I/O controller 1640 represents hardware devices and software components related to interaction with a user. I/O controller 1640 is operable to manage hardware that is part of audio subsystem 1620 and/or display subsystem 1630. Additionally, I/O controller 1640 illustrates a connection point for additional devices that connect to computing device 1600 through which a user might interact with the system. For example, devices that can be attached to the computing device 1600 might include microphone devices, speaker or stereo systems, video systems or other display devices, keyboard or keypad devices, or other I/O devices for use with specific applications such as card readers or other devices.

[0060] As mentioned above, I/O controller 1640 can interact with audio subsystem 1620 and/or display subsystem 1630. For example, input through a microphone or other audio device can provide input or commands for one or more applications or functions of the computing device 1600. Additionally, audio output can be provided instead of, or in addition to display output. In another example, if display subsystem 1630 includes a touch screen, the display device also acts as an input device, which can be at least partially managed by I/O controller

1640. There can also be additional buttons or switches on the computing device 1600 to provide I/O functions managed by I/O controller 1640.

[0061] In some embodiments, I/O controller 1640 manages devices such as accelerometers, cameras, light sensors or other environmental sensors, or other hardware that can be included in the computing device 1600. The input can be part of direct user interaction, as well as providing environmental input to the system to influence its operations (such as filtering for noise, adjusting displays for brightness detection, applying a flash for a camera, or other features).

[0062] In some embodiments, computing device 1600 includes power management 1650 that manages battery power usage, charging of the battery, and features related to power saving operation. Memory subsystem 1660 includes memory devices for storing information in computing device 1600. In some embodiments, Memory subsystem 1600 includes apparatus of memory controller 104 to detect and mitigate defects caused by bit-line opens in a non-volatile memory, according to some embodiments. In some embodiments, Memory subsystem 1600 is an SSD such as SSD 101 with apparatus to detect and mitigate defects caused by bit-line opens in a non-volatile memory.

[0063] Memory can include nonvolatile (state does not change if power to the memory device is interrupted) and/or volatile (state is indeterminate if power to the memory device is interrupted) memory devices. Memory subsystem 1660 can store application data, user data, music, photos, documents, or other data, as well as system data (whether long-term or temporary) related to the execution of the applications and functions of the computing device 1600.

[0064] Elements of embodiments are also provided as a machine-readable medium (e.g., memory 1660) for storing the computer-executable instructions (e.g., instructions to implement any other processes discussed herein). The machine-readable medium (e.g., memory 1660) may include, but is not limited to, flash memory, optical disks, CD-ROMs, DVD ROMs, RAMs, EPROMs, EEPROMs, magnetic or optical cards, PCM, or other types of machine-readable media suitable for storing electronic or computer-executable instructions. For example, embodiments of the disclosure may be downloaded as a computer program (e.g., BIOS) which may be transferred from a remote computer (e.g., a server) to a requesting computer (e.g., a client) by way of data signals via a communication link (e.g., a modem or network connection).

[0065] In some embodiments, computing device 1600 comprises connectivity 1670. Connectivity 1670 includes hardware devices (e.g., wireless and/or wired connectors and communication hardware) and software components (e.g., drivers, protocol stacks) to enable the computing device 1600 to communicate with external devices. The computing device 1600 could be separate devices, such as other computing devices, wireless access points or base stations, as well as peripherals such as headsets, printers, or other devices.

[0066] Connectivity 1670 can include multiple different types of connectivity. To generalize, the computing device 1600 is illustrated with cellular connectivity 1672 and wireless connectivity 1674. Cellular connectivity 1672 refers generally to cellular network connectivity provided by wireless carriers, such as provided via GSM (global system for mobile communications) or variations or derivatives, CDMA (code division multiple access) or variations or derivatives, TDM (time division multiplexing) or variations or derivatives, or other cellular service standards. Wireless connectivity (or wireless interface) 1674 refers to wireless connectivity that is not cellular, and can include personal area networks (such as Bluetooth, Near Field, etc.), local area networks (such as Wi-Fi), and/or wide area networks (such as WiMax), or other wireless communication.

[0067] In some embodiments, computing device 1600 comprises peripheral connections 1680. Peripheral connections 1680 include hardware interfaces and connectors, as well as software components (e.g., drivers, protocol stacks) to make peripheral connections. It will be understood that the computing device 1600 could both be a peripheral device ("to" 1682) to other computing devices, as well as have peripheral devices ("from" 1684) connected to it. The computing device 1600 commonly has a "docking" connector to connect to other computing devices for purposes such as managing (e.g., downloading and/or uploading, changing, synchronizing) content on computing device 1600. Additionally, a docking connector can allow computing device 1600 to connect to certain peripherals that allow the computing device 1600 to control content output, for example, to audiovisual or other systems.

[0068] In addition to a proprietary docking connector or other proprietary connection hardware, the computing device 1600 can make peripheral connections 1680 via common or standards-based connectors. Common types can include a Universal Serial Bus (USB) connector (which can include any of a number of different hardware interfaces), DisplayPort including

MiniDisplayPort (MDP), High Definition Multimedia Interface (HDMI), Firewire, or other types.

[0069] Reference in the specification to "an embodiment," "one embodiment," "some embodiments," or "other embodiments" means that a particular feature, structure, or characteristic described in connection with the embodiments is included in at least some embodiments, but not necessarily all embodiments. The various appearances of "an embodiment," "one embodiment," or "some embodiments" are not necessarily all referring to the same embodiments. If the specification states a component, feature, structure, or characteristic "may," "might," or "could" be included, that particular component, feature, structure, or characteristic is not required to be included. If the specification or claim refers to "a" or "an" element, that does not mean there is only one of the elements. If the specification or claims refer to "an additional" element, that does not preclude there being more than one of the additional element.

[0070] Furthermore, the particular features, structures, functions, or characteristics may be combined in any suitable manner in one or more embodiments. For example, a first embodiment may be combined with a second embodiment anywhere the particular features, structures, functions, or characteristics associated with the two embodiments are not mutually exclusive.

[0071] While the disclosure has been described in conjunction with specific embodiments thereof, many alternatives, modifications and variations of such embodiments will be apparent to those of ordinary skill in the art in light of the foregoing description. For example, other memory architectures e.g., Dynamic RAM (DRAM) may use the embodiments discussed. The embodiments of the disclosure are intended to embrace all such alternatives, modifications, and variations as to fall within the broad scope of the appended claims.

[0072] In addition, well known power/ground connections to integrated circuit (IC) chips and other components may or may not be shown within the presented figures, for simplicity of illustration and discussion, and so as not to obscure the disclosure. Further, arrangements may be shown in block diagram form in order to avoid obscuring the disclosure, and also in view of the fact that specifics with respect to implementation of such block diagram arrangements are highly dependent upon the platform within which the present disclosure is to be implemented (i.e., such specifics should be well within purview of one skilled in the art). Where specific

details (e.g., circuits) are set forth in order to describe example embodiments of the disclosure, it should be apparent to one skilled in the art that the disclosure can be practiced without, or with variation of, these specific details. The description is thus to be regarded as illustrative instead of limiting.

[0073] The following examples pertain to further embodiments. Specifics in the examples may be used anywhere in one or more embodiments. All optional features of the apparatus described herein may also be implemented with respect to a method or process.

[0074] For example, a method is provided which comprises: performing a first read from a portion of a non-volatile memory, the first read to provide a first codeword; decoding the first codeword; determining whether the decoding operation failed; performing a second read from the portion of the non-volatile memory when it is determined that the decoding operation failed, the second read to provide a second codeword; and decoding the second codeword with an errors-and-erasures decoding process. In some embodiments, the method comprises: identifying a location of an open circuit in response to performing the second read; and storing the location of the open circuit.

[0075] In some embodiments, performing the first read comprises: applying a read voltage to a selected word-line; and applying a pass voltage to all other word-lines in the portion of the non-volatile memory. In some embodiments, performing the second read comprises: applying the pass voltage to the selected word-line. In some embodiments, determining whether the decoding operation failed comprises: determining whether a number of soft errors, associated with the decoding operation, exceeds a threshold.

[0076] In some embodiments, the non-volatile memory is a NAND memory. In some embodiments, decoding the first and second codewords is performed by at least one of: a LDPC decoder; or a Reed-Solomon decoder. In some embodiments, the first and second codewords are at least one of: LDPCs; or Reed-Solomon codes. In some embodiments, the errors-and-erasures decoding process comprises setting a Log Likelihood Ratio (LLR) for erasures to zero.

[0077] In another example, an apparatus is provided which comprises: read logic operable to perform a first read from a portion of a non-volatile memory, the first read to provide a first codeword; a decoder operable to decode the first codeword; and logic operable to determine whether the decoder operation failed; wherein the read logic is operable to perform a second read from the portion of the non-volatile memory when it is determined that the decoder

operation failed, the second read to provide a second codeword, and wherein the decoder is operable to decode the second codeword with an errors-and-erasures decoding process.

[0078] In some embodiments, the read logic is operable to apply: a read voltage to a selected word-line to perform a first read; and a pass voltage to all other word-lines in the portion of the non-volatile memory. In some embodiments, the read logic is operable to perform the second read by applying the pass voltage to the selected word-line. In some embodiments, the decoder is at least one of: a LDPC decoder; or a Reed-Solomon decoder. In some embodiments, the non-volatile memory is a NAND memory. In some embodiments, the decoder is operable to determine whether the decoding operation failed by comparing a number of soft errors, associated with the decoding operation, against a threshold. In some embodiments, the apparatus a storage unit for storing a location of an open circuit in the non-volatile memory in response to the logic determining that the decoder operation failed.

[0079] In another example, a system is provided which comprises: a processor; a non-volatile memory; a memory controller coupled to the non-volatile memory, the memory controller including: read logic operable to perform a first read from a portion of the non-volatile memory, the first read to provide a first codeword; a decoder operable to decode the first codeword; and logic operable to determine whether the decoder operation failed; wherein the read logic is operable to perform a second read from the portion of the non-volatile memory when it is determined that the decoder operation failed, the second read to provide a second codeword, and wherein the decoder is operable to decode the second codeword with an errors-and-erasures decoding process; and a wireless interface for allowing the processor to communicate with another device.

[0080] In some embodiments, the read logic is operable to apply: a read voltage to a selected word-line to perform a first read; and a pass voltage to all other word-lines in the portion of the non-volatile memory. In some embodiments, the read logic is operable to perform the second read by applying the pass voltage to the selected word-line. In some embodiments, the non-volatile memory is a NAND memory.

[0081] In another example, a machine readable storage media is provided having machine readable instructions that, when executed, cause one or more processors to perform an operation according to the method described above.

[0082] In another example, an apparatus is provided which comprises means for performing a first read from a portion of a non-volatile memory, the first read to provide a first codeword; means for decoding the first codeword; means for determining whether the decoding operation failed; means for performing a second read from the portion of the non-volatile memory when it is determined that the decoding operation failed, the second read to provide a second codeword; and means for decoding the second codeword with an errors-and-erasures decoding process.

[0083] In some embodiments, the apparatus comprises: means for identifying a location of an open circuit in response to performing the second read; and means for storing the location of the open circuit. In some embodiments, the means for performing the first read comprises: means for applying a read voltage to a selected word-line; and means for applying a pass voltage to all other word-lines in the portion of the non-volatile memory. In some embodiments, the means for performing the second read comprises: means for applying the pass voltage to the selected word-line. In some embodiments, the means for determining whether the decoding operation failed comprises: means for determining whether a number of soft errors, associated with the decoding operation, exceeds a threshold. In some embodiments, the non-volatile memory is a NAND memory. In some embodiments, the means for decoding the first and second codewords is performed by at least one of: a LDPC decoder; or a Reed-Solomon decoder. In some embodiments, the first and second codewords are at least one of: LDPCs; or Reed-Solomon codes. In some embodiments, the errors-and-erasures decoding process comprises setting a LLR for erasures to zero.

[0084] An abstract is provided that will allow the reader to ascertain the nature and gist of the technical disclosure. The abstract is submitted with the understanding that it will not be used to limit the scope or meaning of the claims. The following claims are hereby incorporated into the detailed description, with each claim standing on its own as a separate embodiment.

CLAIMS

We claim:

1. A method comprising:
 - performing a first read from a portion of a non-volatile memory, the first read to provide a first codeword;
 - decoding the first codeword;
 - determining whether the decoding operation failed;
 - performing a second read from the portion of the non-volatile memory when it is determined that the decoding operation failed, the second read to provide a second codeword;
 - and
 - decoding the second codeword with an errors-and-erasures decoding process.
2. The method claim 1 comprising:
 - identifying a location of an open circuit in response to performing the second read; and
 - storing the location of the open circuit.
3. The method of claim 1, wherein performing the first read comprises:
 - applying a read voltage to a selected word-line; and
 - applying a pass voltage to all other word-lines in the portion of the non-volatile memory.
4. The method of claim 3, wherein performing the second read comprises: applying the pass voltage to the selected word-line.
5. The method of claim 1, wherein determining whether the decoding operation failed comprises: determining whether a number of soft errors, associated with the decoding operation, exceeds a threshold.
6. The method of claim 1, wherein the non-volatile memory is a NAND memory.

7. The method of claim 1, wherein decoding the first and second codewords is performed by at least one of: a Low-Density Parity-Check Code (LDPC) decoder; or a Reed-Solomon decoder.
8. The method of claim 1, wherein the first and second codewords are at least one of: Low-Density Parity-Check Codes (LDPCs); or Reed-Solomon codes.
9. The method of claim 1, wherein the errors-and-erasures decoding process comprises setting a Log Likelihood Ratio (LLR) for erasures to zero.
10. An apparatus comprising:
 - read logic operable to perform a first read from a portion of a non-volatile memory, the first read to provide a first codeword;
 - a decoder operable to decode the first codeword; and
 - logic operable to determine whether the decoder operation failed, wherein the read logic is operable to perform a second read from the portion of the non-volatile memory when it is determined that the decoder operation failed, the second read to provide a second codeword, and wherein the decoder is operable to decode the second codeword with an errors-and-erasures decoding process.
11. The apparatus of claim 10, wherein the read logic is operable to apply:
 - a read voltage to a selected word-line to perform a first read; and
 - a pass voltage to all other word-lines in the portion of the non-volatile memory.
12. The apparatus of claim 11, wherein the read logic is operable to perform the second read by applying the pass voltage to the selected word-line.
13. The apparatus of claim 10, wherein the decoder is at least one of: a Low-Density Parity-Check Code (LDPC) decoder; or a Reed-Solomon decoder.
14. The apparatus of claim 10, wherein the non-volatile memory is a NAND memory.

15. The apparatus of claim 10, wherein the decoder is operable to determine whether the decoding operation failed by comparing a number of soft errors, associated with the decoding operation, against a threshold.
16. The apparatus of claim 10 comprises a storage unit for storing a location of an open circuit in the non-volatile memory in response to the logic determining that the decoder operation failed.
17. A system comprising:
 - a processor;
 - a non-volatile memory;
 - a memory controller coupled to the non-volatile memory, the memory controller including:
 - read logic operable to perform a first read from a portion of the non-volatile memory, the first read to provide a first codeword;
 - a decoder operable to decode the first codeword; and
 - logic operable to determine whether the decoder operation failed; wherein the read logic is operable to perform a second read from the portion of the non-volatile memory when it is determined that the decoder operation failed, the second read to provide a second codeword, and wherein the decoder is operable to decode the second codeword with an errors-and-erasures decoding process; and
 - a wireless interface for allowing the processor to communicate with another device.
18. The system of claim 17, wherein the read logic is operable to apply:
 - a read voltage to a selected word-line to perform a first read; and
 - a pass voltage to all other word-lines in the portion of the non-volatile memory.
19. The system of claim 18, wherein the read logic is operable to perform the second read by applying the pass voltage to the selected word-line.
20. The system of claim 16, wherein the non-volatile memory is a NAND memory.

21. A machine readable storage media having machine readable instructions that, when executed, cause one or more processors to perform an operation according to any one of the method claims 1 to 9.

22. An apparatus comprising:

means for performing a first read from a portion of a non-volatile memory, the first read to provide a first codeword;

means for decoding the first codeword;

means for determining whether the decoding operation failed;

means for performing a second read from the portion of the non-volatile memory when it is determined that the decoding operation failed, the second read to provide a second codeword; and

means for decoding the second codeword with an errors-and-erasures decoding process.

23. The apparatus claim 22 comprising:

means for identifying a location of an open circuit in response to performing the second read; and

means for storing the location of the open circuit.

24. The apparatus of claim 22, wherein the means for performing the first read comprises:

means for applying a read voltage to a selected word-line; and

means for applying a pass voltage to all other word-lines in the portion of the non-volatile memory.

25. The apparatus of claim 24, wherein the means for performing the second read comprises:

means for applying the pass voltage to the selected word-line

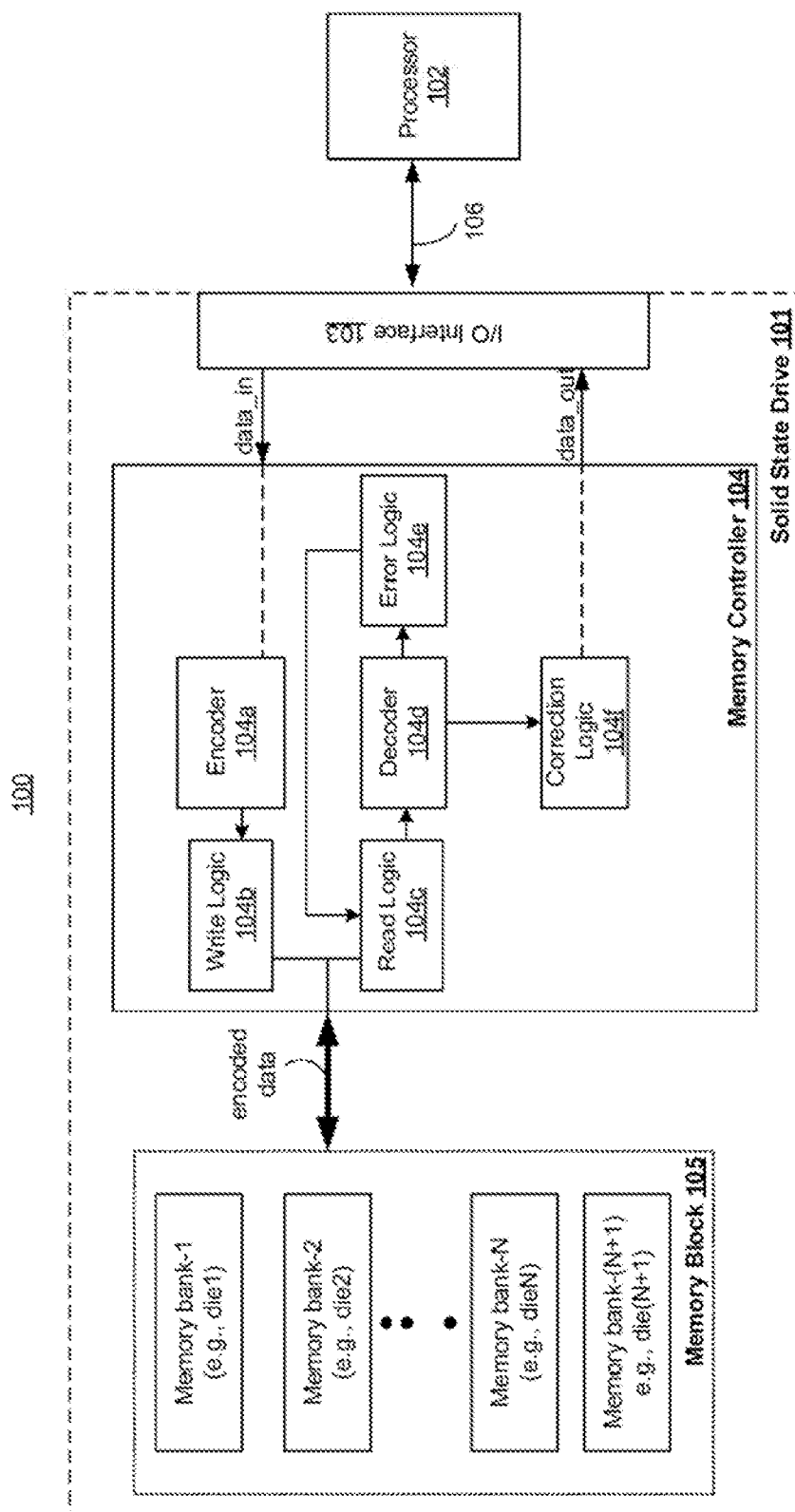


Fig. 1

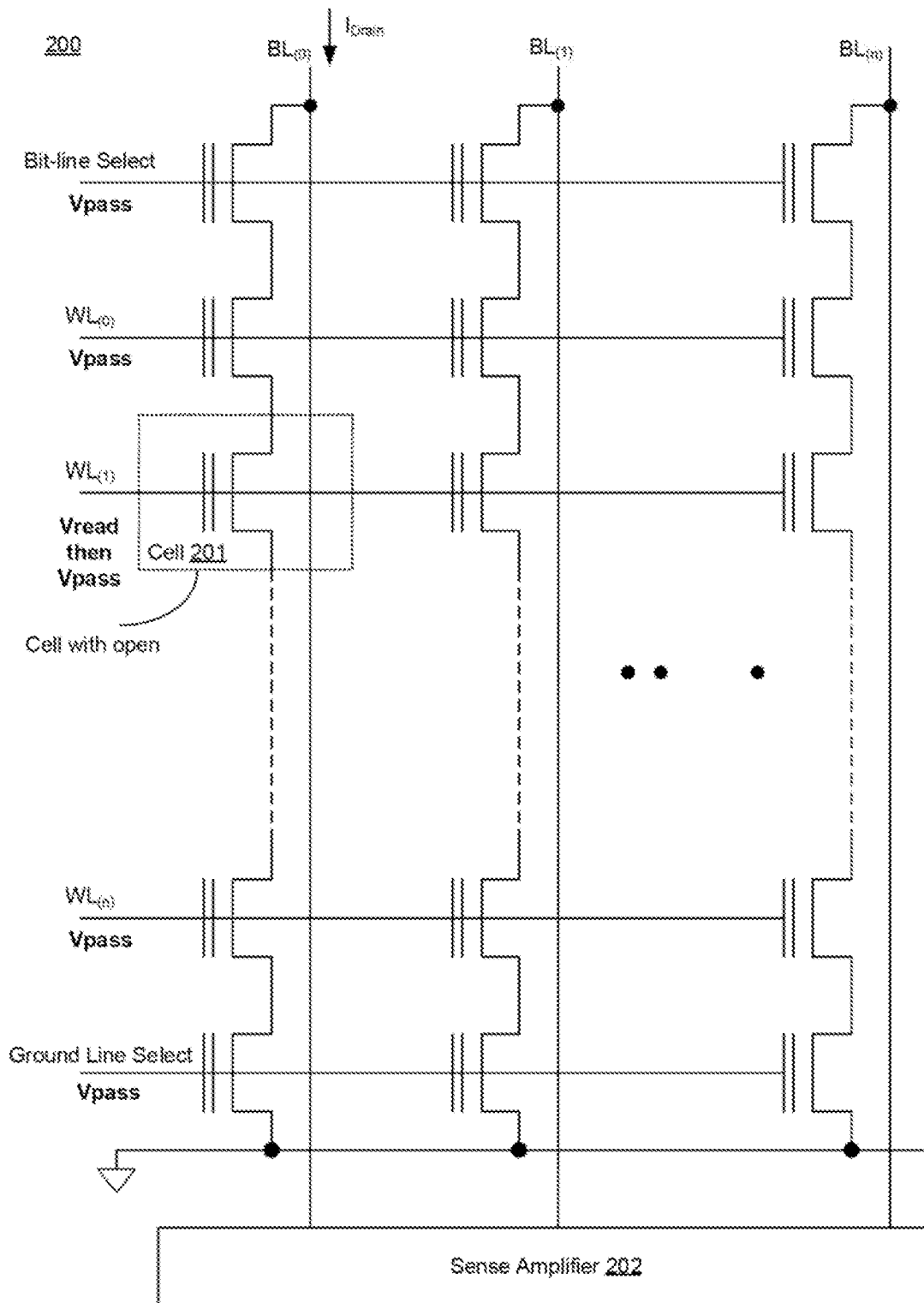
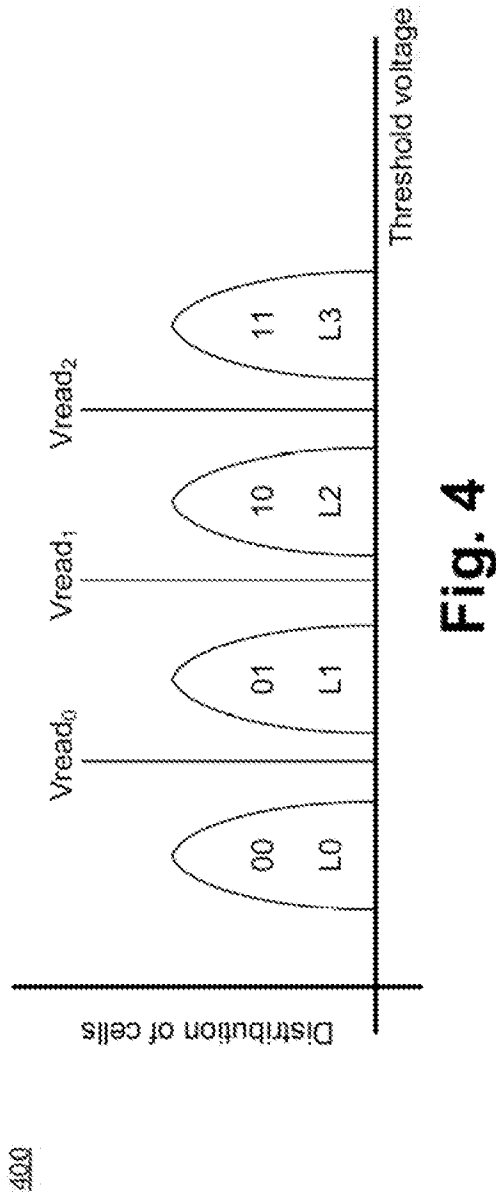
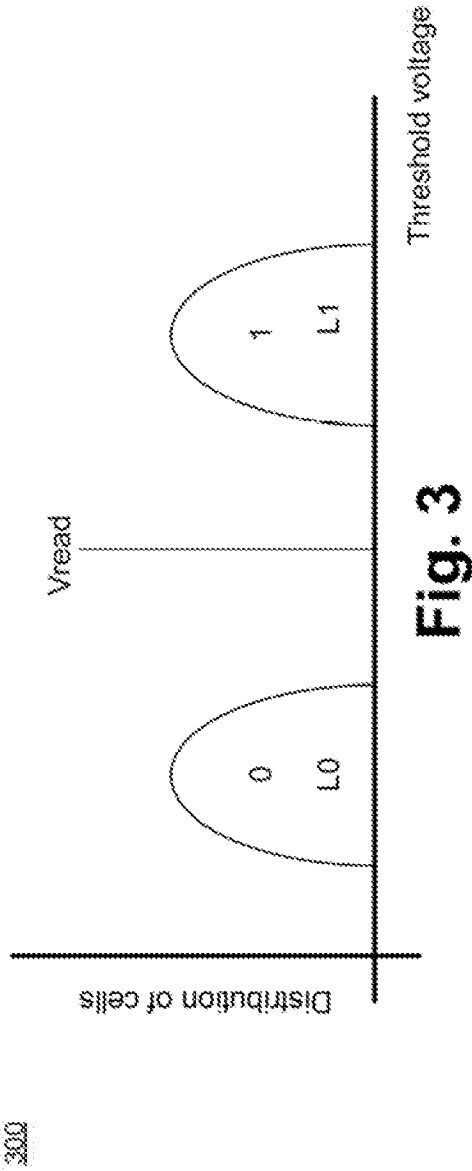


Fig. 2



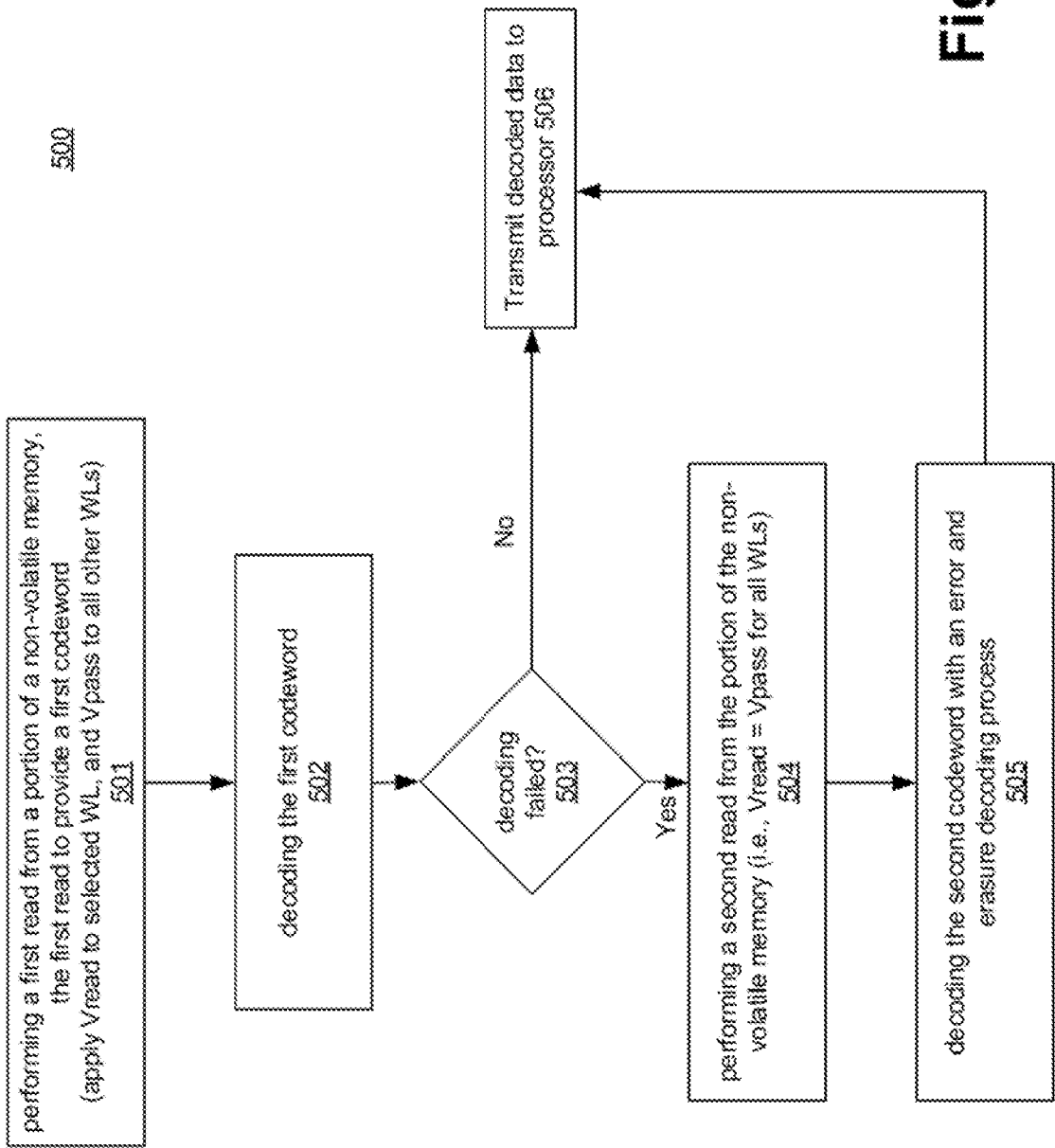


Fig. 5

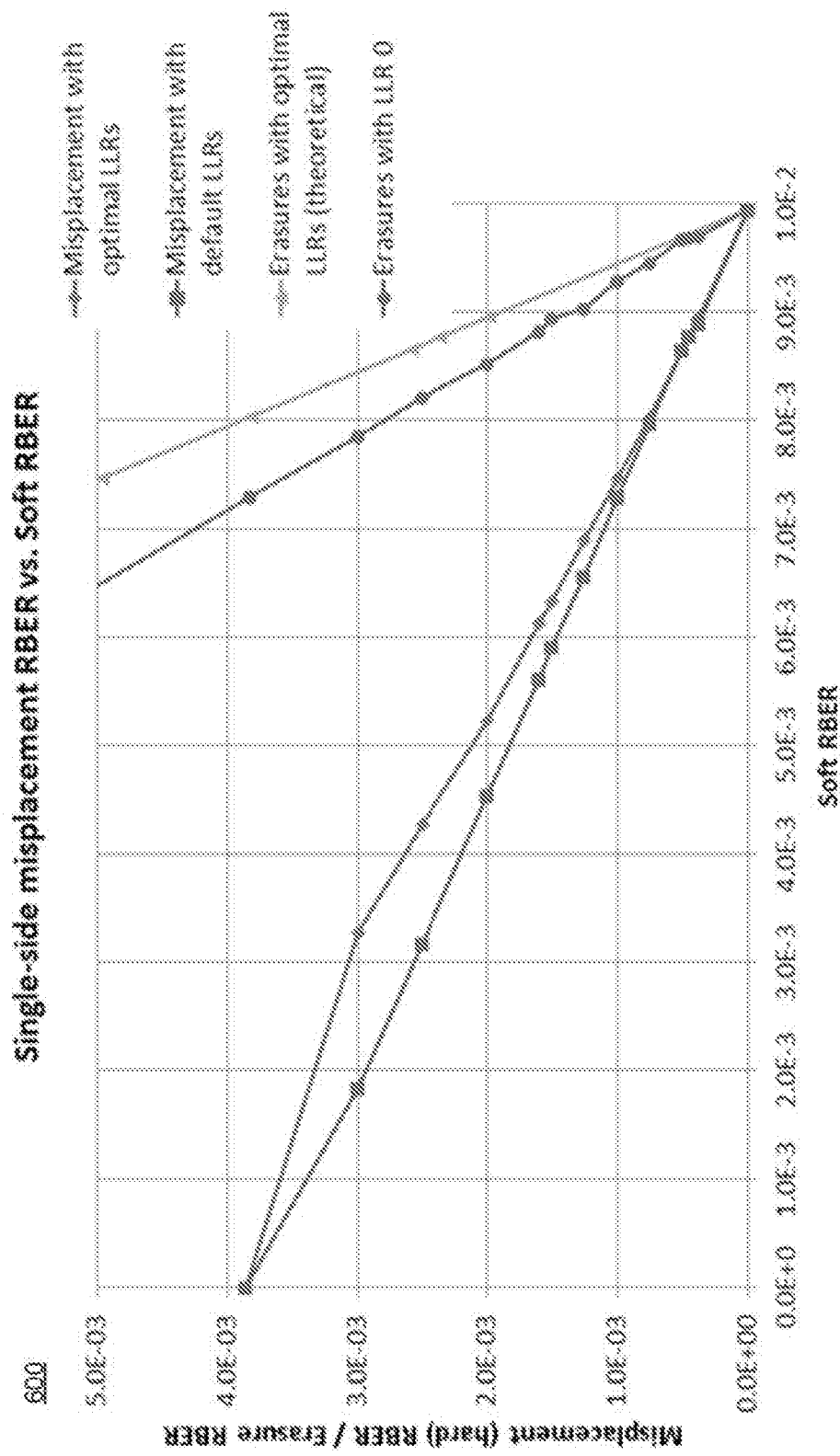


Fig. 6

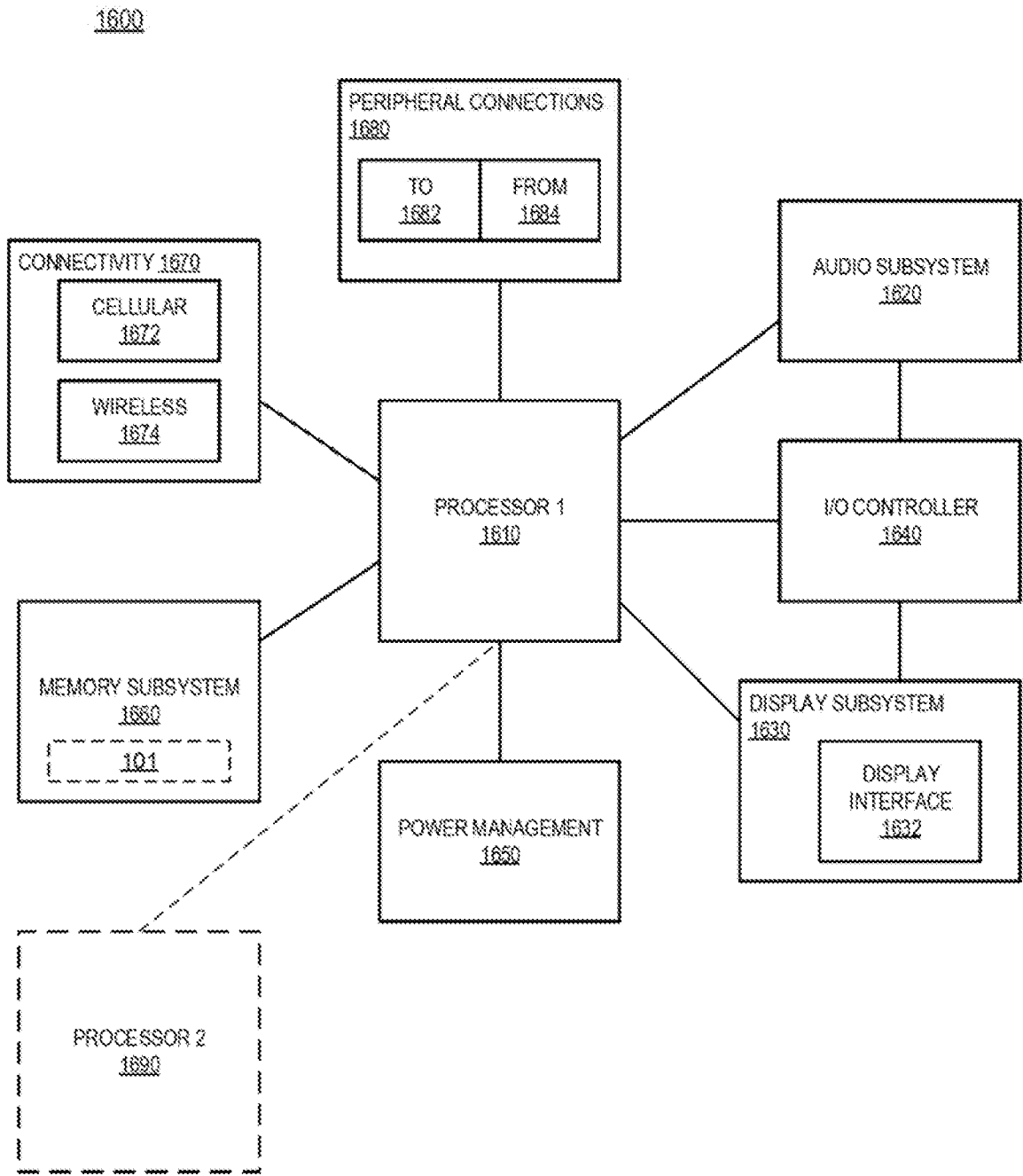


Fig. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2016/017228**A. CLASSIFICATION OF SUBJECT MATTER****G11C 16/08(2006.01)i, G11C 16/34(2006.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G11C 16/08; H04L 1/00; G06F 11/25; H03M 13/15; G11C 29/50; G06F 11/10; G06F 11/08; G11C 29/52; 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: non-volatile, memory, first, second, read, codeword, decoding, determining, operation, failed, error-and-erasures, low density parity check code(LDPC), log likelihood ratio(LLR)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2010-0077279 A1 (YONG JUNE KIM et al.) 25 March 2010 See paragraphs [0011]-[0012], [0068], [0073], [0079]; and figures 3, 12, 14, 16.	1,3,5-6,10-11 ,14-15,17-18,20-22 ,24
Y		7-9,13
A		2,4,12,16,19,23,25
Y	US 2015-0089323 A1 (YU KOU et al.) 26 March 2015 See paragraphs [0015], [0021]; and figures 1-2.	7-9,13
A	US 2014-0281839 A1 (ZION S. KWOK) 18 September 2014 See paragraphs [0045]-[0054]; and figure 4.	1-25
A	US 2013-0326296 A1 (SEONGHYEONG CHOI et al.) 05 December 2013 See paragraphs [0063]-[0069]; and figure 6.	1-25
A	US 2011-0252283 A1 (NIMA MOKHLESI et al.) 13 October 2011 See paragraphs [0094]-[0102]; and figures 17A-17B.	1-25



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

07 June 2016 (07.06.2016)

Date of mailing of the international search report

08 June 2016 (08.06.2016)

Name and mailing address of the ISA/KR

International Application Division

Korean Intellectual Property Office

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

Facsimile No. +82-42-481-8578

Authorized officer

BYUN, Sung Cheal

Telephone No. +82-42-481-8262



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2016/017228

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2010-0077279 A1	25/03/2010	KR 10-1466270 B1 KR 10-2010-0033195 A US 8370710 B2	28/11/2014 29/03/2010 05/02/2013
US 2015-0089323 A1	26/03/2015	US 8898546 B1 US 9021340 B2	25/11/2014 28/04/2015
US 2014-0281839 A1	18/09/2014	US 9054742 B2	09/06/2015
US 2013-0326296 A1	05/12/2013	KR 10-2013-0136271 A US 9105359 B2	12/12/2013 11/08/2015
US 2011-0252283 A1	13/10/2011	US 2008-244338 A1 US 7966550 B2 US 8145981 B2	02/10/2008 21/06/2011 27/03/2012