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

(19) **World Intellectual Property Organization**
International Bureau

(43) **International Publication Date**
22 September 2011 (22.09.2011)

PCT

(10) **International Publication Number**
WO 2011/116071 A2

(51) **International Patent Classification**:
G06F 12/02 (2006.01) *G06F 12/16* (2006.01)

(21) **International Application Number**:
PCT/US2011/028637

(22) **International Filing Date**:
16 March 2011 (16.03.2011)

(25) **Filing Language**: English

(26) **Publication Language**: English

(30) **Priority Data**:
12/726,200 17 March 2010 (17.03.2010) US

(71) **Applicant** *(for all designated States except US)*: **PLIANT TECHNOLOGY, INC.** [US/US]; 630 Alder Drive, Suite 202, Milpitas, California 95035 (US).

(72) **Inventors; and**
(75) **Inventors/Applicants** *(for US only)*: **OLBRICH, Aaron K.** [US/US]; 14525 Atherton Circle, Morgan Hill, California 95037 (US). **PRINS, Douglas A.** [US/US]; 27096 Hidden Trail Road, Laguna Hills, California 92653 (US).

(74) **Agent**: **KWOK, Edward C.**; Haynes and Boone, LLP, 2323 Victory Avenue, Suite 700, Dallas, Texas 75219 (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, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, 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, SD, SL, SZ, TZ, UG,

[Continued on next page]

(54) **Title**: MLC SELF-RAID FLASH DATA PROTECTION SCHEME

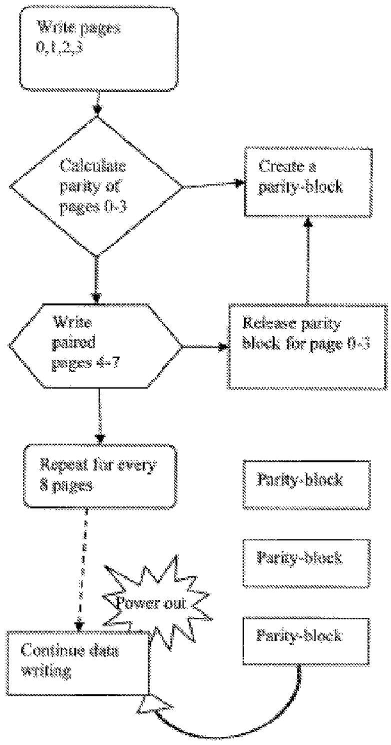


FIG. 14

(57) **Abstract**: A two-dimensional self-RAID method of protecting page-based storage data in a MLC multiple-level-cell flash memory device. The protection scheme includes reserving one parity sector across each data page, reserving one parity page as the column parity, selecting a specific number of pages to form a parity group, writing into the parity page a group parity value for data stored in the pages of the parity group. The parity sector represents applying a RAID technique in a first dimension. The group parity represents applying a RAID technique in a second dimension. Data protection is achieved because a corrupted data sector can likely be recovered by the two dimensional RAID data.

WO 2011/116071 A2

ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, 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, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

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

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

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

MLC SELF-RAID FLASH DATA PROTECTION SCHEME

Aaron K. Olbrich and Douglas A. Prins

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] The present application is related to and claims priority of U.S. Patent Application No. 12/726,200 filed on March 17, 2010.

FIELD OF THE INVENTION

[0002] The invention described herein relates to data storage management in semiconductor flash memories, and in particular to a data storage protection method that prevents data corruption in multiple level cell (MLC) memory devices in the event of a power interruption.

BACKGROUND OF THE INVENTION

[0003] Current enterprise-level mass storage relies on hard drives that are typically characterized by a 3.5” form factor, a 15,000 rpm spindle motor and a storage capacity between 73GB and 450GB. The mechanical design follows the traditional hard drive with a single actuator and 8 read/write heads moving across 8 surfaces. The constraints of the head/media technology limit the read/write capabilities to using only one active head at a time. All data requests that are sent to the drive are handled in a serial manner, with long delays between operations, as the actuator moves the read/write head to the required position and the media rotates to place the data under the read/write head.

[0004] A solid state memory device is attractive in an enterprise mass-storage environment. For that environment, the flash memory is a good candidate among various solid state memory devices, since it does not have the mechanical delays associated with hard drives, thereby allowing higher performance and commensurately lower cost, and better usage of power and space.

[0005] The flash memory is a form of non-volatile memory, i.e. EEPROM (electronically erasable programmable read-only memory). A memory cell in a flash memory array generally includes a transistor having a control gate and drain and source diffusion regions formed in a substrate. The transistor has a floating gate under the control gate, thus forming an electron storage device. A channel region lies under the floating gate, isolated by an insulation layer (e.g. a tunnel oxide layer) between the channel and the floating gate. The energy barrier imposed by the insulating layer against charge carriers movement into or out of the floating gate can be overcome by applying a sufficiently high electric field across the insulating layer. The charge stored in the floating gate determines the threshold voltage (Vt) of the cell, which represents the stored data of the cell. Charge stored in the floating gate causes the cell to have a higher Vt. To change the Vt of a cell to a higher or lower value, the charge stored in the floating gate is increased or decreased by applying appropriate voltages at the control gate, the drain and source diffusion regions, and the channel region. The appropriate voltages cause charge to move between one or more of these regions and through the insulation layer to the floating gate.

[0006] A single-level cell (SLC) flash memory device has a single threshold voltage Vt and can store one bit of data per cell. A memory cell in a multiple-level cell (MLC) flash memory device has multiple threshold voltages, and depending on the amount of charge stored in the floating gate, can represent more than one bit of data. Because a MLC flash memory device enables the storage of multiple data bits per cell, high density mass storage applications (such as 512 Mb and beyond) are readily achievable. In a typical four-level two-bit MLC flash memory device, the cell threshold voltage Vt can be set at any of four levels to represent data “00”, “01”, “10”, and “11”. To program the memory cell to a given level, the cell may be programmed multiple times. Before each write, a flash memory array is erased to reset every cell in the array to a default state. As a result, multiple data bits that share the same cell and their electronic states, (hence their threshold voltage Vt’s), are interdependent to a point that an unexpected power interruption can generate unpredictable consequences. Variations in the electronic states of the memory cells also generate variations within ranges of threshold voltages in a real system. Table 1 below shows the electronic states and the threshold voltage ranges in a two-bit MLC.

[0007] Table 1 Threshold voltages and bit values in a two-bit MLC memory cell

Vt	Bit 1	Bit 2
-4.25V to -1.75V	1	1
-1.75V to 0.75V	1	0
0.75V to 3.25V	0	1
3.25V to 5.75V	0	0

[0008] In spite of the advantages of MLC over SLC, MLC flash memory devices have not traditionally been used because of certain technical constraints, among which data corruption presents one of the most severe challenges.

[0009] All flash memories have a finite number of erase-write cycles. MLC flash memory devices are more vulnerable to data corruption than SLC flash memory devices. The specified erase cycle limit for each flash memory page is typically in the order of 100,000 cycles for SLC flash memory devices and typically in the order of 10,000 cycles for MLC devices. The lower cycle limit in the MLC flash memory devices poses particular problems for data centers that operate with unpredictable data streams. The unpredictable data streams may cause “hot spots”, resulting in certain highly-used areas of memory being subject to a large number of erase cycles.

[0010] In addition, various factors in normal operation can also affect flash memory integrity, including read disturbs or program disturbs. These disturbs lead to unpredictable loss of data bits in a memory cell, as a result of the reading or writing of memory cells adjacent to the disturbed cell. Sudden data losses in MLC flash memory devices due to unexpected power interruptions require frequent data recoveries. Because some data levels require more than one write operations to achieve and because more than one bit of data share the same memory cell, a power change or a program error during a write data operation leaves the data in a wrong state. When the power returns, the memory cell can be in an erratic state. Therefore a power interruption is a major risk to the integrity of data stored in MLC flash memory devices.

[0011] Flash media typically are written in units called “pages”; each page typically includes between 2000 bytes and 8000 bytes. Flash media typically are erased in units called “blocks”. Each block typically includes between 16 and 64 pages. Pages in MLC flash memory devices are coupled into paired pages. The number of paired pages maybe two for the 2-bit MLC and may go up to 3 to 4 or higher for higher bit MLCs. The paired pages may reside in shared MLC flash memory cells. If the power failure occurs while the MLC is in the middle of an operation that changes the contents of the flash media (e.g., in the middle of writing a page of data or in the middle of erasing a block of data), the electrical states of the interrupted page or block are unpredictable after the device is powered up again. The electrical states can even be random, because some of the affected bits may already be in the states assigned to them by the operation, at the time power is interrupted. However, other bits may be lagging behind and have not yet reached their target values yet. Furthermore, some bits might be caught in intermediate states and thus be in an unreliable mode, so that reading these bits returns different results under different read operations. Therefore power losses while programming a certain page can corrupt a paired page.

[0012] In the prior arts, error correction codes (ECC) and Redundant Array of Inexpensive Disk (RAID) techniques have been used to mitigate data corruption. In one instance, data corruption is prevented by writing parity pages at a different page address. Those techniques require either additional memory or complicated error-searching and data rebuilding procedures after power returns. Such requirements or solutions make the process costly to implement and place significant strain on the processing power of a conventional flash memory controller, which generally includes only a single processor. Furthermore, if a power failure occurs during the writing of a page, the paired page data can become corrupt in a MLC flash memory device. Therefore even the conventional paired page technique is susceptible to a sudden power interruption. As a matter of fact, the severity of the possible corruption is high; in some cases, every 10th data bit can be lost. Relying on conventional ECC techniques to make a MLC flash memory system reliable would be impractical to implement.

[0013] NAND flash memory data corruption can also result from program erase cycle wear outs. Electrons are injected and removed by tunneling through thin film oxide insulators. Repeated program/erase cycles damage the oxide and reduce its effectiveness. As device dimensions (e.g., oxide film thickness) shrink, data integrity problems from device wearing out can become more severe. One factor that influences this wearing out process is the speed at which the program and erase cycles are performed. However, if the speed of programming and erasing is slowed to avoid wearing out, overall performance can be impacted significantly.

[0014] Currently, a technique exists which applies a lower sense voltage to measure the charge states of the flash memory, in order to extend the lifetime of the memory device. A flash memory device is a charge-trap device that uses sense circuits to detect if a cell contains a given charge level. However, as the device wears out, its ability to store a charge is compromised. A worn out memory device allows the stored charge on the floating gate to leak. Consequently a sense circuit will detect a reduced voltage from the device. One current recovery mechanism reduces the sense voltage that is used to determine the logic value a cell contains. However, a lower sense voltage also returns a lower detected voltage, thus resulting in an incorrect charge tracking.

SUMMARY OF THE INVENTION

[0015] The present invention provides a two-dimensional self-RAID method of protecting, following a power loss, page-based storage data in a multiple-level-cell flash memory device. The process includes reserving a parity sector in each data page under an application of RAID ("First dimensional RAID") technique, thereby forming a parity group containing a predetermined number of pages, and repeating the parity grouping for every subsequent data pages under a second application of a RAID technique ("Second dimensional RAID"). Thus if a subsequent write corrupts a paired page, the lost data can be recovered using the two dimensional RAID data.

[0016] The first dimensional parity in the present invention is associated with a data page. One sector within the page is reserved for the first dimensional RAID data. This parity sector allows the recovery of any single sector within the ECC capability of that sector. This level of RAID data can be calculated from the available data at the time the controller transfers the data to a chip buffer.

[0017] The second dimensional parity in the present invention is calculated across a column of sectors in a predetermined number of pages. When the specific page number is selected carefully, paired page faults can be recovered.

[0018] Full data protection against power interruption is achieved because any corrupted data sector can be recovered by the RAID data either from the within page sector parity or from the crossed sector page parity.

[0019] The present invention provides a method of preserving page-based flash memory integrity during writing, in the event of a power loss. The present invention can be used to manage a flash memory having multiple level cells (MLC).

[0020] According to an embodiment of the present invention, a method is provided to protect a MLC flash memory data which includes numerous memory pages. A method of managing a multiple level cell flash memory that includes a plurality of pages, each page including a plurality of sequentially numbered sectors, the method comprising: (a) choosing a sector in each page as a parity sector; (b) writing data into each page and calculating a parity value of the sectors in each page and storing the parity value in the reserved parity sectors; (c) dividing data pages into a plurality of groups, wherein each group, except the first group and the last group, consists of a first predetermined number of pages; (d) reserving a page in each group as a group parity page and writing data into each page of the group, calculating a parity value of the group and storing the parity value in the reserved group parity page; (e) repeating (a) to (d) for each group; (f) reserving a new page to store a column parity of all sectors sharing the same sector number.

[0021] According to another embodiment of the present invention, a method is provided to protect a MLC flash memory data which includes numerous memory pages. A method of managing a multiple level cell flash memory that includes a plurality of pages, each page including a plurality of sequentially numbered sectors, the method comprising: (a) reserving a parity block; (b) choosing a sector in each page as a parity sector; (c) writing data into each page and calculating a parity value of the sectors in each page and storing the parity value in the chosen parity sector; (d) dividing data pages into a plurality of groups, wherein each group, except the last group, consists of a second predetermined number of pages; (e) writing data into pages in a subset of a group and calculating the subset group parity; (f) storing the subset group parity value in the reserved parity block; (g) writing data into the remaining pages of the group; (h) repeating (b) to (g); (l) reserving a new page to store a column parity of sectors sharing the same sector number but residing in different pages.

[0022] According to the present invention, a data storage system is provided in which the above methods can be operated upon.

[0023] According to the present invention, a method for reducing data corruption from device wearing out is provided by extending the programming and erasing time on selected weak cells. Weak cells are identified by the rate they generate errors, and then the blocks and pages associated with the weak cells can be programmed and erased at a slower rate than other cells . By tracking the weaker blocks and treating them differently than other more robust blocks, endurance can be enhanced. Because the slower programming and erasure processes are only performed on the relatively few weak blocks, overall performance is not significantly compromised. A method of managing a multiple level cell flash memory with numerous pages, the method comprising: (a) programming and erasing data on a page at a predetermined speed; (b) detecting an error rate for each of the pages and identifying the pages associated with error rates that exceed a predetermined value; (c) programming and erasing the identified high error page set a speed that is slower than the predetermined speed.

[0024] According to another embodiment of the present invention, a method for overcoming the leakage induced charge level shifts is provided. A method of managing a multiple level cell flash memory that includes a sense circuitry, the method comprising: (a) selecting a sense voltage; (b) detecting charge levels of memory cells in the multiple level cell flash memory with the selected sense voltage, and making a first table that correlates the predetermined sense voltage and the sensed charge levels; (c) reducing the sense voltage and detecting charge levels of memory cells in the multiple level cell flash memory using the reduced sense voltage and making a second table that correlates the reduced sensed voltage with the sensed charge levels; (d) replacing the first table with the second table.

[0025] The present invention is better understood upon consideration of the detailed description below in conjunction with the accompanying drawings.

[0026] DESCRIPTION OF THE DRAWINGS

[0027] FIG. 1 shows a schematic block diagram of a device controller using a flash medium for non-volatile data storage.

[0028] FIG. 2 is a cross sectional view of a flash memory device.

[0029] FIG. 3 shows a cross sectional view of a multi-level flash memory cell and its equivalent circuit diagram.

[0030] FIG. 4 shows a two bit-MLC threshold voltage ranges in a MLC flash memory device.

[0031] FIG. 5 is a flash memory paired page diagram.

[0032] FIG. 6 illustrates the concept of column parity without group parity.

[0033] FIG. 7 illustrates the concept of sector parity or row parity.

[0034] FIG. 8 shows group parity with 8 pages in each parity sector group.

[0035] FIG. 9 is a table that summarizes the 2-dimensional RAID group parity scheme showing a flash memory data set of 63 pages (rows) and 31 sectors (columns).

[0036] FIG. 10 shows 8-page group parity configuration in accordance with an embodiment of the present invention.

[0037] FIG. 11 shows 8-page group parity configuration using a set aside parity block in accordance with another embodiment of the present invention.

[0038] FIG. 12 shows a protection scheme using a two-dimensional RAID technique (example 1).

[0039] FIG. 13 shows a protection scheme using a two-dimensional RAID technique (example 2).

[0040] FIG. 14 is a flow chart of an algorithm which protects a MLC flash memory device against corruption resulting from a power interruption during a write operation.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0041] FIG. 1 is a high level schematic block diagram of a conventional NAND flash media device in a non-volatile data storage unit.

[0042] FIG. 2 shows a cross sectional view of a conventional flash memory device.

[0043] An example of a conventional multiple level (MLC) flash memory cell is illustrated in FIG. 3, where a split channel device has two different threshold voltages, Vt1 and Vt2. The referenced cross sectional view and equivalent circuit are taken from US Pat. 5,045,940 (Harari).

[0044] Variations of the electronic states generate ranges of threshold voltages in a real MLC system. FIG. 4 depicts the threshold voltage spans in a conventional two-bit MLC device.

[0045] Pages of data sharing the same multiple level cells are called “shared pages”. Each manufacturer may use a different distance between its shared pages. Many memory vendors prefer to set the distance at four. For example, at a pair distance of 4, page 0 is paired with page 4, page 1 is paired with page 5, page 2 is paired with page 6, and page 3 is paired with page 7. FIG. 5 shows the paired pages concept at the pair distance of 4.

[0046] The paired pages may share the same memory cells in a MLC flash memory system (e.g., in a 2-bit MLC flash memory, bit 0 and bit 1 of a memory cell are bits from the first and second pages of the paired pages, respectively). When a program operation is abnormally aborted, for example, during a power down or a reset, not only is the page data that is being programmed damaged, the data in the paired page may also be damaged, even though it may have been written correctly at a previous time.

[0047] According to one embodiment of the present invention, RAID techniques are applied in a method along two dimensions. In the first dimension, “the first dimensional RAID”), the method preserves parity information on the same page. The first dimensional RAID uses row parity or the sector parity, which is calculated using data from the first sector to the last sector in the same page. As shown in FIG. 7, there are 31 sectors of data in each page, respectively labeled Sec 00 to Sec 30. A single sector within each page is reserved for storing the first dimensional RAID data (i.e., the row or sector parity). In Fig. 7, p0 sector is the parity sector for Page 0, and sector pX is the parity sector for Page X. One common error correction code (ECC) allows single-bit error correction. With an extra parity bit, double-bit errors can be detected. Therefore the parity data in the parity sector allows recovery of any sector in a page within the ECC capability of the sector, in the event of a power interruption. Sector parity may be calculated from all the data in the page at the time the data is moved from the controller to a chip buffer.

[0048] In the second dimension ("the second dimensional RAID"), the method preserves parity data calculated over a number of pages in a parity group. Such parity data is referred to as group parity. The number of pages in each parity group is variable. In one implementation, for example, the number of pages in a parity group is 8 pages.

[0049] Group parity that is calculated for corresponding sectors over all pages in a block is referred to column parity. FIG. 6 illustrates column parity in a 64 page data set. In FIG. 6, each column parity value is calculated over the same sectors, for example, Sec X, from all the pages. The parity value is calculated and written in a parity page designated for the block, with a column parity calculated for the same sector in all the pages of the block.

[0050] Group parity in the second dimensional RAID provides additional parity protection in a flash memory device. A group parity for a parity group that includes less than all pages of a block sets a higher level of protection than column parity which is illustrated in FIG. 6. As a first step, the number of pages in the parity group is selected as the group size. This number is chosen carefully in order to provide adequate protection from paired page faults and is often selected by the MLC flash memory manufacturer. For the pairing chosen, a typical group size is 8. FIG. 8 shows group parity with 8 pages in each parity sector group. In FIG. 8, pages p3, p11, ..., p59 are parity pages.

[0051] Figure 10 illustrates a parity grouping configuration applicable to the configuration shown in FIG. 8, according to one embodiment of the present invention. As shown in FIG. 10, the first group (i.e., Group 1) is a half group. In each subsequent group of 8 pages, the first 4 pages are paired with the 4 pages in the previous group. For instance, if the page pairing sequence at the multiple level cells is 0-4, 1-5, 2-6 and 3-7 (i.e. page 0 is paired to page 4, page 1 is paired to page 5, page 2 is paired to page 6, and page 3 is paired to page 7), then for each write of any of pages 0, 1, 2, a corresponding 2-dimensional RAID group parity write can be performed on group parity page 3. In the event that a subsequent write of a page in any of pages 4 to 6 corrupts the corresponding paired page 0, 1, or 2 the data loss may be recovered using the RAID data from the previously written group parity page 3. Thus the group parity page protects page 0-2 in parity group 1 of 4 pages. The group 2 parity page is page 11, protecting page 8, 9, 10 from damages caused by power interruption during writing pages 12 to 14. FIG. 9 is a table that summarizes the 2-dimensional RAID group parity scheme showing a flash memory data set of 63 pages (rows) and 31 sectors (columns). As shown in FIG. 9, Page 3, Page 11 ... and Page 59 are such selected to be the group parity pages for 8-page groups. First half of each group is paired, or co-resided with the second half of the previous group, on the same MLC flash memory device.

[0052] FIG. 11 illustrates a parity grouping configuration in anther embodiment of the present invention. In FIG. 11, all groups contain 8 pages. The page pairing sequence at the multiple level cells is 0-4, 1-5, 2-6 and 3-7 (i.e. page 0 is paired to page 4, page 1 is paired to page 5, page 2 is paired to page 6, and page 3 is paired to page 7). The group parity for pages 0-3 is written to a set-aside parity page in parity block outside the data pages. If there is power interruption during a subsequent writing of any of pages 4-7, the parity page in the set-aside parity block may be used to recover the damaged data in pages 0-3. If pages 4-7 are written successfully, the parity page for pages 0-3 in the parity block outside the data pages can be saved for future reference, and also can be erased if desirable. This process is repeated for every 8 pages.

[0053] Another advantage of this type of RAID protection is that it does not require reads to generate the parity data on writes. All that is required is a parity cache for the pages being written. This simplifies the algorithm required for parity generation and does not cause a write performance penalty. The only time performance is affected is during the rebuild of data in the event that a hard error is encountered.

[0054] Although in the detailed description of the current invention, an exemplary number of 8 are used as the number of pages in a group, the invention does not limit the number of pages in a group to 8.

[0055] To implement the scheme shown in FIG. 11, in the first dimension, each of the 64 pages is provided with a parity sector (located, for example, in the second sector of each page). The parity sector in each page provides the first dimensional RAID. In the second dimension, there are two levels of parity protection. First, there is column parity for all pages. In the column parity page, each sector stores the column parity calculated based on all the corresponding sectors of all the pages. Second, group parity is also recorded according for pages grouped according to a selected group size. The group parity may be written to set-aside parity blocks. Alternatively, the group parity may use the 4th, 11th, 19th ... 59th pages to store group parity pages along with the data pages. In one embodiment using set-aside parity blocks, after the second half group of the paired pages are written without any power interruptions, the set-aside parity block written for the first half of the paired pages may be erased or may be saved for future reference.

[0056] FIG. 12 illustrates how the two dimensional RAID protection works when a number of uncorrectable sectors occur in one embodiment of the present invention (the corrupted sectors are represented in FIG. 12 by densely lined blocks). For example, Sec 00 in Page 4 and Sec 02 in Page 5 are single errors in the respective pages; therefore, the data loss in the corresponding sector can be recovered from the parity sector on each of those pages respectively. However, as Page 7 is corrupted in two sectors, the corruption in Sec 00 in Page 7 can be recovered from the group parity page 11. Similarly, the corruption in Sec 02 in Page 9 can also be recovered from the group parity page 11. Subsequent to the above corrections, both Sec 30 in Page 7 and Sec 30 in Page 9 can be recovered from the parity sectors p7 and p9 respectively, since the other sectors in those pages have been recovered. Thus it is possible to recover data even though there are two bad sectors in one page or in one column.

[0057] In a more severe power interruption scenario when a number of paired pages are affected in a single page or in a single column, the group sector parity is able to recover corrupted data by combining the row parity, column parity and group parity. One example of the recovery scheme is illustrated in FIG. 13, where multiple hard errors exist (the error sectors are represented with dense-lined blocks). Sec 00 in Page 0, Sec 02 in Page 1 and Sec 02 in Page 5 are single errors in the corresponding pages; therefore, they can be recovered from the parity sectors in those pages respectively. Sec 29 in Page 9 can be recovered from the group parity in Sec 29 of Page 11. After these corrections are made, Sec 00 in Page 3 can be recovered by recalculating group parity of Sec 00 over pages 0-2, and Sec 02 in Page 9 can be recovered from the group parity of Sec 02 in Page 11. Subsequent to those corrections, Sec 00 in Page 7 is correctable from the group parity page 11, Sec 02 in Page 13, and Sec 29 in Page 13 can be recovered from the group parity page 19. Following the above corrections, Sec 30 in Page 3, Sec 30 in Page 7, Sec 30 in Page 9 and Sec 30 in Page 13 can be recovered from the corresponding parity sectors since the other bad sectors in relevant pages have been recovered. Thus it is possible to recover data even though there are four sectors with hard errors residing in one column.

[0058] In one embodiment of the present invention where a set aside parity block outside the data pages is used to reserve the group parity, an algorithm can be written for the process of data writing, when data is protected against the write-corrupt at a power interruption. This algorithm comprises the following steps for a parity group of 8 pages:

a. Write 4 pages from pages 0 to page 3 and calculate the group parity for pages 0-3;

b. Write the group parity page to a Parity-Block outside the data memory;

c. Write pages 4-7;

d. Release the Parity-Block of pages 0-3, used Parity-Block can be saved or erased;

e. Repeat the process a) to d) for every subsequent group of 8 pages;

f. Treat the last group the same way even if the last group has less than 8 pages.

[0059] This algorithm is also illustrated in FIG. 14.

[0060] The foregoing description is intended to illustrate, but not to limit, the scope of the invention, which is defined by the scope of the appended claims. Other embodiments are within the scope of this disclosure.

WHAT IS CLAIMED IS:

1. A method of managing a multiple level cell flash memory that is logically organized to include a block having a plurality of pages, each page including a plurality of sectors, the method comprising:

(a) choosing a parity sector in each page;

(b) assigning the pages of the block into one or more groups and selecting a parity page in each group;

(c) for each page, calculating a sector parity value for data stored in the sectors of the page and storing the sector parity value into the parity sector of the page; and

(d) calculating a group parity value of data stored in the pages of the group and storing the group parity value in the parity page.

2. The method of managing flash memory as in Claim 1, further comprising:

(e) repeating steps (c) and (d) for each group.

3. The method of managing flash memory as in Claim 2, wherein the sectors of each page are each assigned a sector number, the method further comprising:

(f) selecting a column parity page and calculating, for each sector number, a column parity for all sectors of the sector number in the pages of the block.

4. The method as in Claim 1, wherein each multiple level cell is shared by at least two pages of the block.

5. The method as in Claim 1, wherein each group consists of 8 pages.

6. The methodas in Claim 4, wherein the pages sharing each multiple level cells are assigned to different groups.

7. The method as in Claim 6, wherein the groups are each assigned a group number and wherein the pagessharing a multiple level cell are assigned consecutive group numbers.

8. The method as in Claim 7, wherein one or more of thegroupsare assigned half the number of pages as assigned to each of the remainder of the groups of the block.

9. A method of managing a multiple level cell flash memory that is organized logically into one or more blocks each having a plurality of pages, each page including a plurality of sectors, the method comprising:

(a) selecting from the blocks a parity block;

(b) choosing a parity sector in each page;

(c) assigning the pages of the block into a plurality of groups;

(d) for each page, calculating a sector parity value for data stored in the sectors in each page and storing the sector parity value in the parity sector;

(e) prior to completing data writing into all pages of a group, calculating a subset group parity for a subset of pages in the group; and

(f) storing the subset group parity in the parity block.

10. The method as in Claim 9, further comprising the steps of:

(h) repeating steps (d) to (f).

11. The method as in Claim 9, wherein the sectors of each page are each assigned a sector number, the method further comprising:

(f) selecting a column parity page and calculating, for each sector number, a column parity for all sectors of the sector number in the pages of the block.

12. The method as in Claim 9, wherein each group consists of 8 pages.

13. The method as in Claim 9, wherein each multiple level cell is shared by two pages of the same group.

14. The method as in Claim 9, wherein the parity block resides outside the flash memory.

15. The method as in Claim 9, wherein the parity block resides on a non-volatile memory.

16. The method as in Claim 9, wherein the subset group parity of a group

written into the parity block is erased after data is completely written into the pages of the group.

17. The method as in Claim 9, the subset group parity of a group written into the parity block is saved after data is completely written into the pages of the group.

18. The method as in Claim 9, wherein the subset group parity is calculated from data in half of the pages of the group.

19. A flash memory data storage system operable in accordance with the method of Claim 1.

20. A flash memory data storage system operable in accordance with the method of Claim 9.

21. A method of managing a multiple level cell flash memory that is organized logically into blocks each including a plurality of pages, each page including a plurality of sequentially numbered sectors, the method comprising:

(a) programming and erasing data on a page at a predetermined speed;

(b) detecting an error rate for each page of a block and identifying a group of high error pages based on the error rates; and

(c) applying a speed slower than the predetermined speed in programming and erasing data on the identified high error pages.

22. A method of managing a multiple level cell flash memory that includes a sense circuitry, comprising:

(a) detecting charge levels of the multiple level cells using a first sense voltage, and compiling a first correlation table between the first sense voltage and each of the charge level of each of the multiple level cell;

(b) selecting a second sense voltage that is lower than the first sense voltage and detecting charge levels of the multiple level cells using the second sense voltage, and compiling a correlation table between the second sense voltage and each of the charge level of each of the multiple level cells;

(c) replace the first correlation table by the second correlation table.

Prior Art

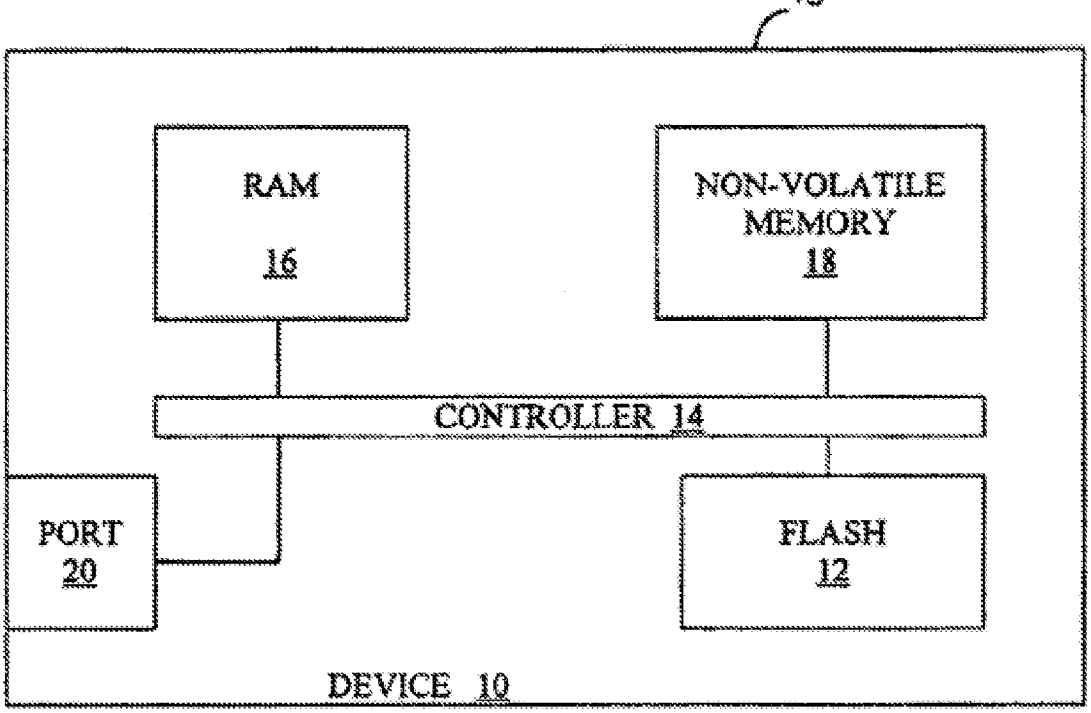
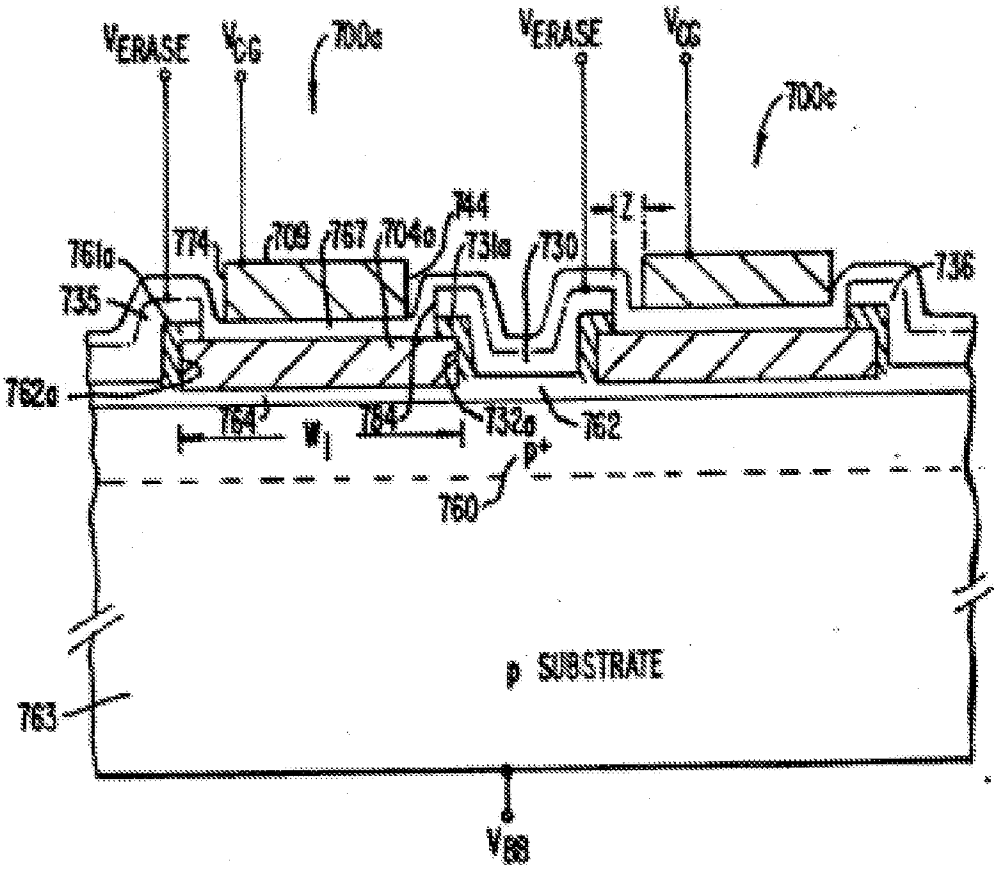


FIG. 1

Prior Art



Note: 709 is the control gate and 704a is the isolated floating gate.

FIG. 2

Prior Art

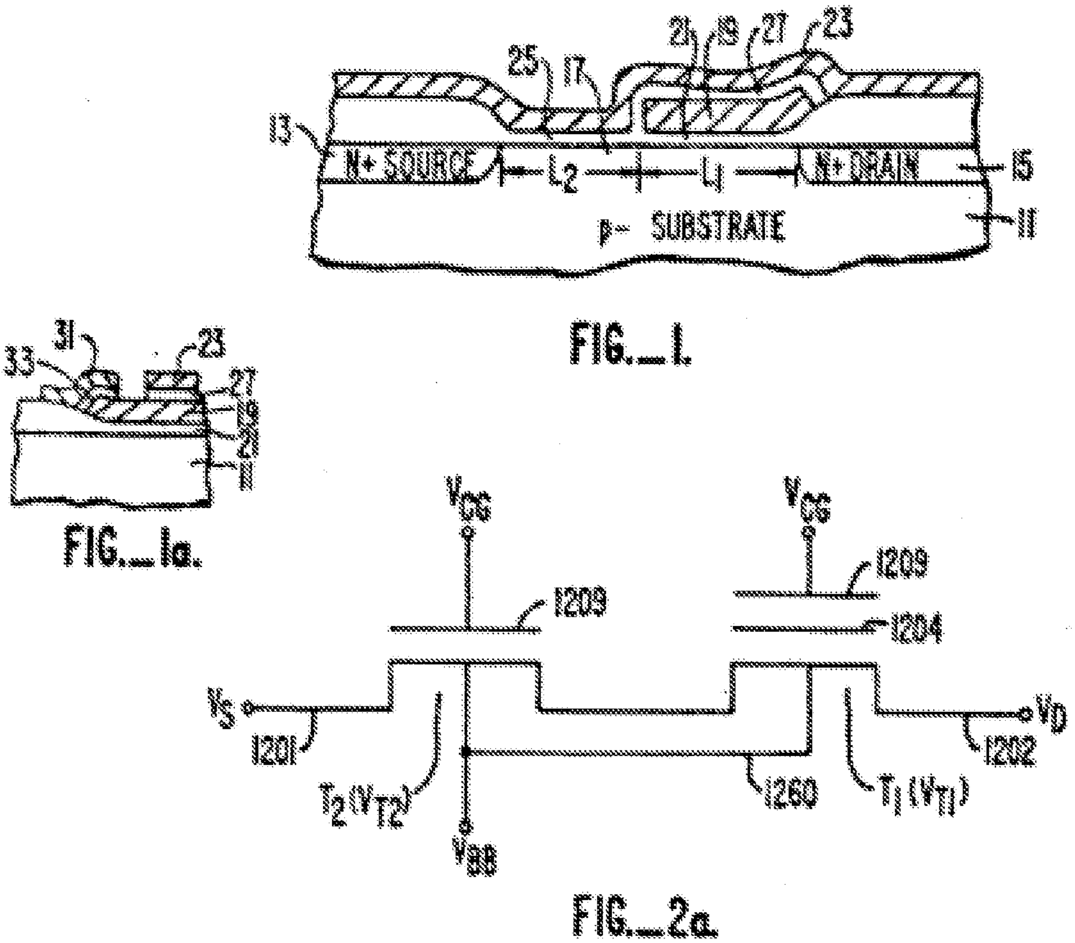


FIG._1.

FIG._1a.

FIG._2a.

FIG. 3

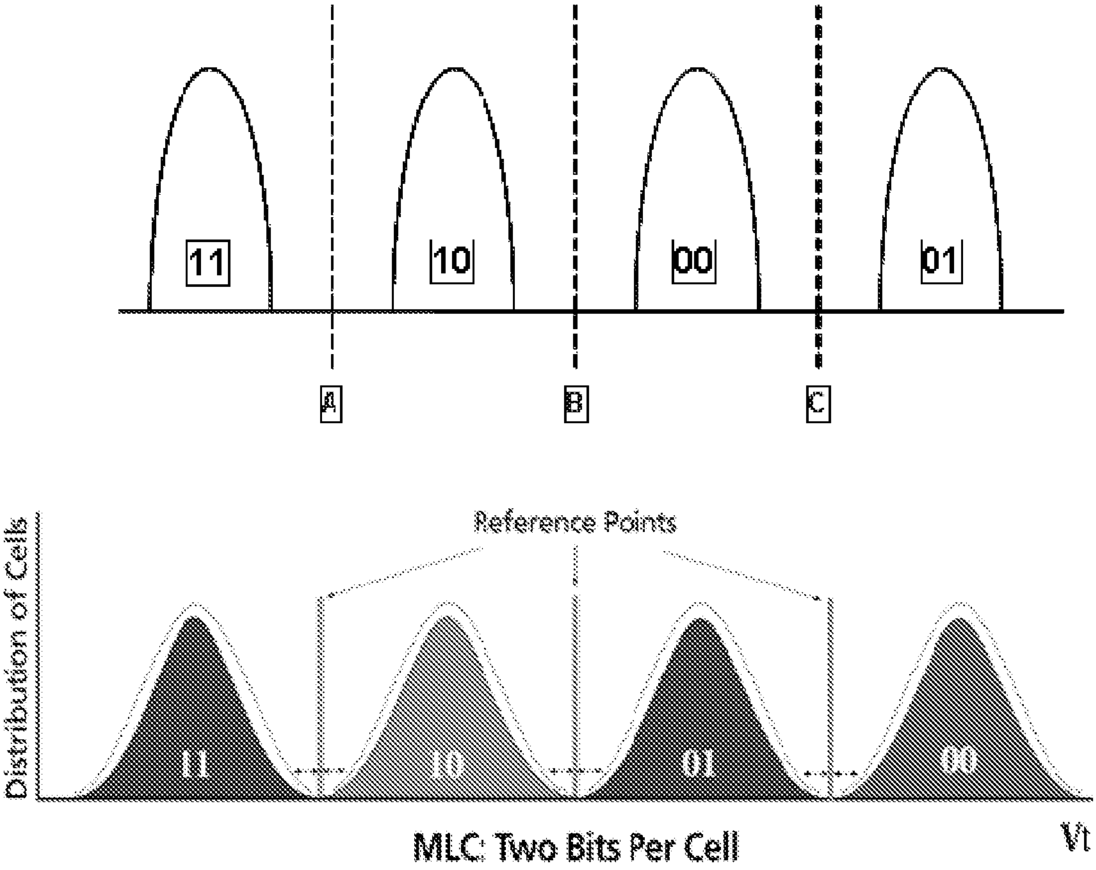

11
10
00
01
A
B
C
Reference Points
Distribution of Cells
11
10
01
00
MLC: Two Bits Per Cell
Vt

FIG. 4

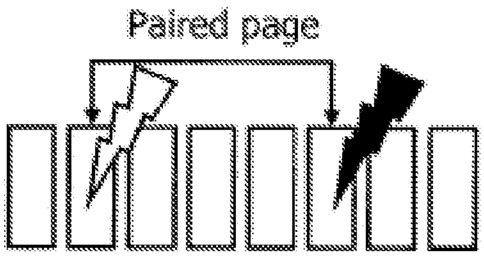
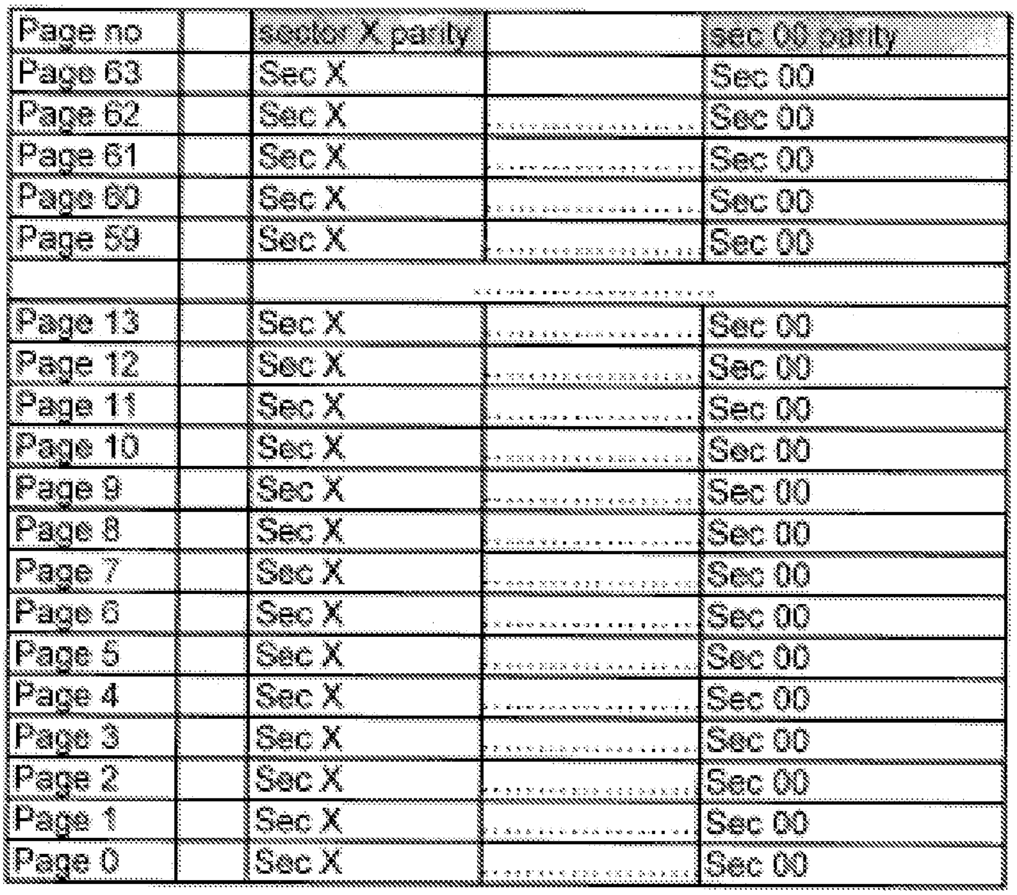


FIG. 5

Column parity

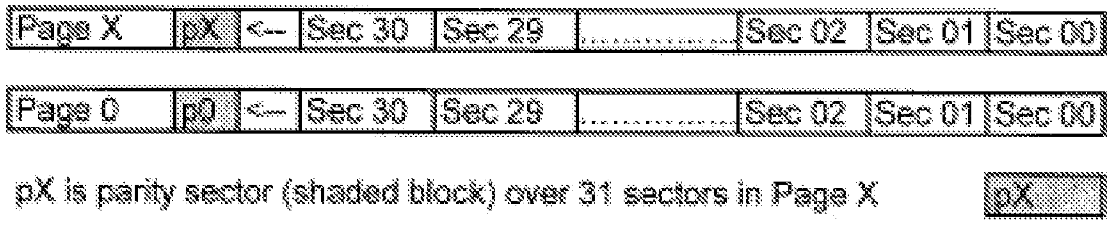


Page no		sector X parity		sec 00 parity
Page 63		Sec X		Sec 00
Page 62		Sec X	....	Sec 00
Page 61		Sec X	....	Sec 00
Page 60		Sec X	....	Sec 00
Page 59		Sec X	....	Sec 00
		....		
Page 13		Sec X	....	Sec 00
Page 12		Sec X	....	Sec 00
Page 11		Sec X	....	Sec 00
Page 10		Sec X	....	Sec 00
Page 9		Sec X	....	Sec 00
Page 8		Sec X	....	Sec 00
Page 7		Sec X	....	Sec 00
Page 6		Sec X	....	Sec 00
Page 5		Sec X	....	Sec 00
Page 4		Sec X	....	Sec 00
Page 3		Sec X	....	Sec 00
Page 2		Sec X	....	Sec 00
Page 1		Sec X	....	Sec 00
Page 0		Sec X	....	Sec 00

Sector X parity summarizes all data in Sec X of all pages

FIG. 6

Flash memory first dimensional RAID - sector parity



Page X	pX	<--	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 0	p0	<--	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00

pX is parity sector (shaded block) over 31 sectors in Page X ▨ pX

Sector parity id in shaded blocks as ▨

FIG. 7

Group parity with 8 pages in each group

Page no		sector X parity		sector 00 parity
Page 63		Sec X		Sec 00
Page 62		Sec X	....	Sec 00
Page 61		Sec X	....	Sec 00
Page 60		Sec X	....	Sec 00
Page 59		Sec X	....	Sec 00
....			....	
Page 13		Sec X	....	Sec 00
Page 12		Sec X	....	Sec 00
Page 11		Sec X	....	Sec 00
Page 10		Sec X	....	Sec 00
Page 9		Sec X	....	Sec 00
Page 8		Sec X	....	Sec 00
Page 7		Sec X	....	Sec 00
Page 6		Sec X	....	Sec 00
Page 5		Sec X	....	Sec 00
Page 4		Sec X	....	Sec 00
Page 3		Sec X	....	Sec 00
Page 2		Sec X	....	Sec 00
Page 1		Sec X	....	Sec 00
Page 0		Sec X	....	Sec 00

Group parity pages are in shaded blocks as ▨ .

FIG. 8

2 dimensional RAID table

Page no	group	page parity		sector 30 parity	sector 29 parity		sector 02 parity	sector 01 parity	sector 00 parity
Page 63		p63	<--	Sec 30	Sec 29	…………………	Sec 02	Sec 01	Sec 00
Page 62		p62	<--	Sec 30	Sec 29	…………………	Sec 02	Sec 01	Sec 00
Page 61		p61	<--	Sec 30	Sec 29	…………………	Sec 02	Sec 01	Sec 00
Page 60		p60	<--	Sec 30	Sec 29	…………………	Sec 02	Sec 01	Sec 00
Page 59	52 to 58	p59	<--	Sec 30	Sec 29		Sec 02	Sec 01	Sec 00
			<--			…………………			
Page 13		p13	<--	Sec 30	Sec 29	…………………	Sec 02	Sec 01	Sec 00
Page 12		p12	<--	Sec 30	Sec 29	…………………	Sec 02	Sec 01	Sec 00
Page 11	4 to 10	p11	<--	Sec 30	Sec 29		Sec 02	Sec 01	Sec 00
Page 10		p10	<--	Sec 30	Sec 29	…………………	Sec 02	Sec 01	Sec 00
Page 9		p9	<--	Sec 30	Sec 29	…………………	Sec 02	Sec 01	Sec 00
Page 8		p8	<--	Sec 30	Sec 29	…………………	Sec 02	Sec 01	Sec 00
Page 7		p7	<--	Sec 30	Sec 29	…………………	Sec 02	Sec 01	Sec 00
Page 6		p6	<--	Sec 30	Sec 29	…………………	Sec 02	Sec 01	Sec 00
Page 5		p5	<--	Sec 30	Sec 29	…………………	Sec 02	Sec 01	Sec 00
Page 4		p4	<--	Sec 30	Sec 29	…………………	Sec 02	Sec 01	Sec 00
Page 3	0 to 2	p3	<--	Sec 30	Sec 29		Sec 02	Sec 01	Sec 00
Page 2		p2	<--	Sec 30	Sec 29	…………………	Sec 02	Sec 01	Sec 00
Page 1		p1	<--	Sec 30	Sec 29	…………………	Sec 02	Sec 01	Sec 00
Page 0		p0	<--	Sec 30	Sec 29	…………………	Sec 02	Sec 01	Sec 00

Parity sectors and group parity pages are in shaded blocks as ▨ .

FIG. 9

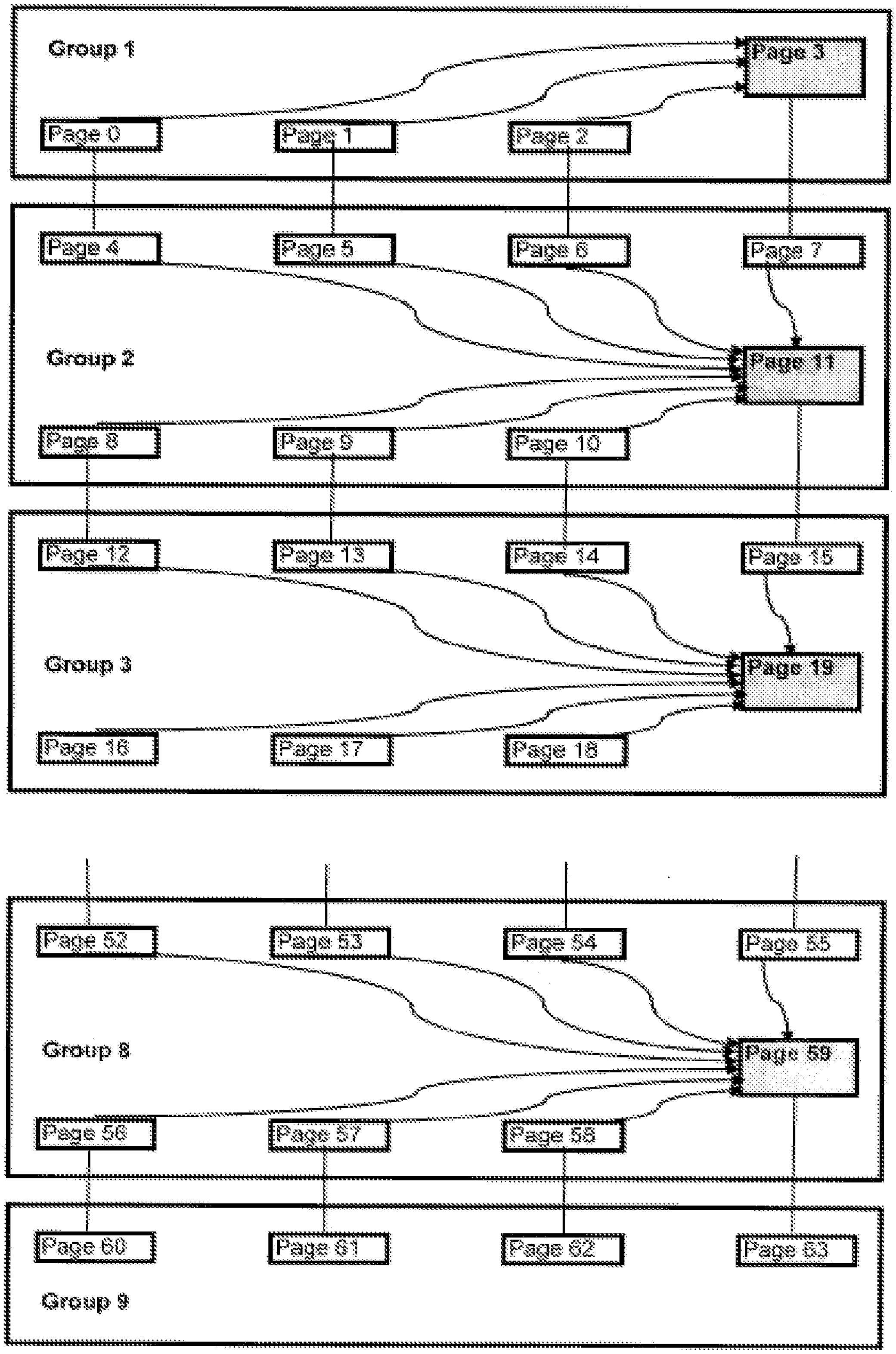

Group 1
Page 3
Page 0
Page 1
Page 2
Page 4
Page 5
Page 6
Page 7
Group 2
Page 11
Page 8
Page 9
Page 10
Page 12
Page 13
Page 14
Page 15
Group 3
Page 19
Page 16
Page 17
Page 18
Page 52
Page 53
Page 54
Page 55
Group 8
Page 59
Page 56
Page 57
Page 58
Page 60
Page 61
Page 62
Page 63
Group 9

FIG. 10

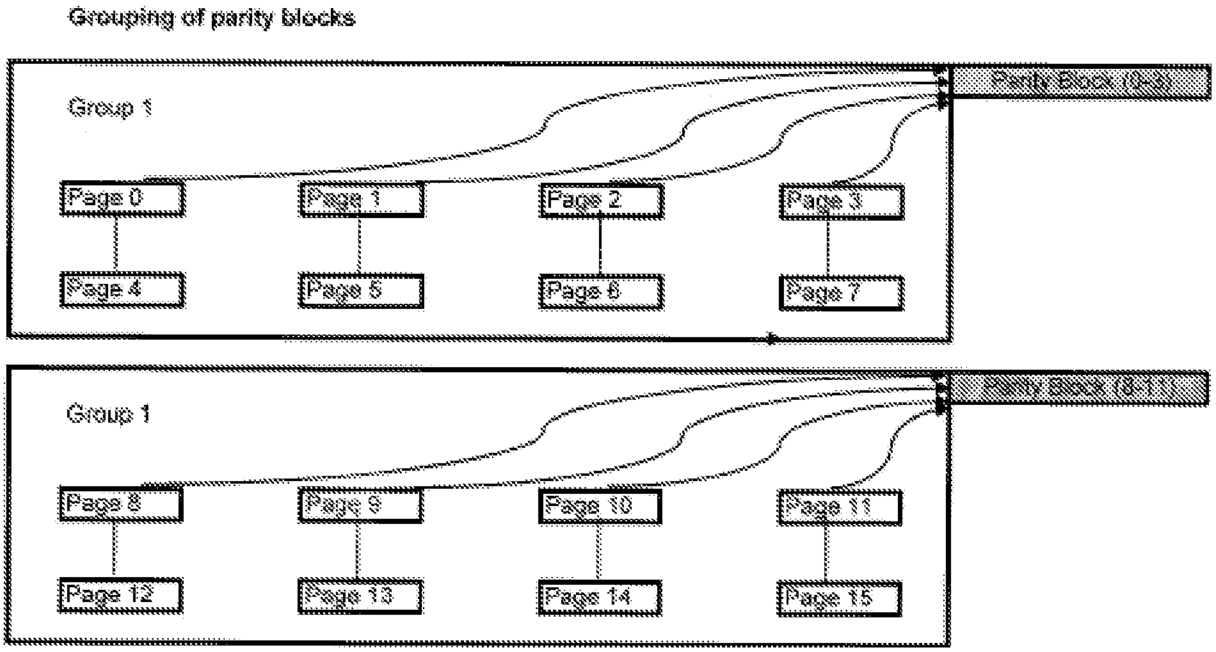
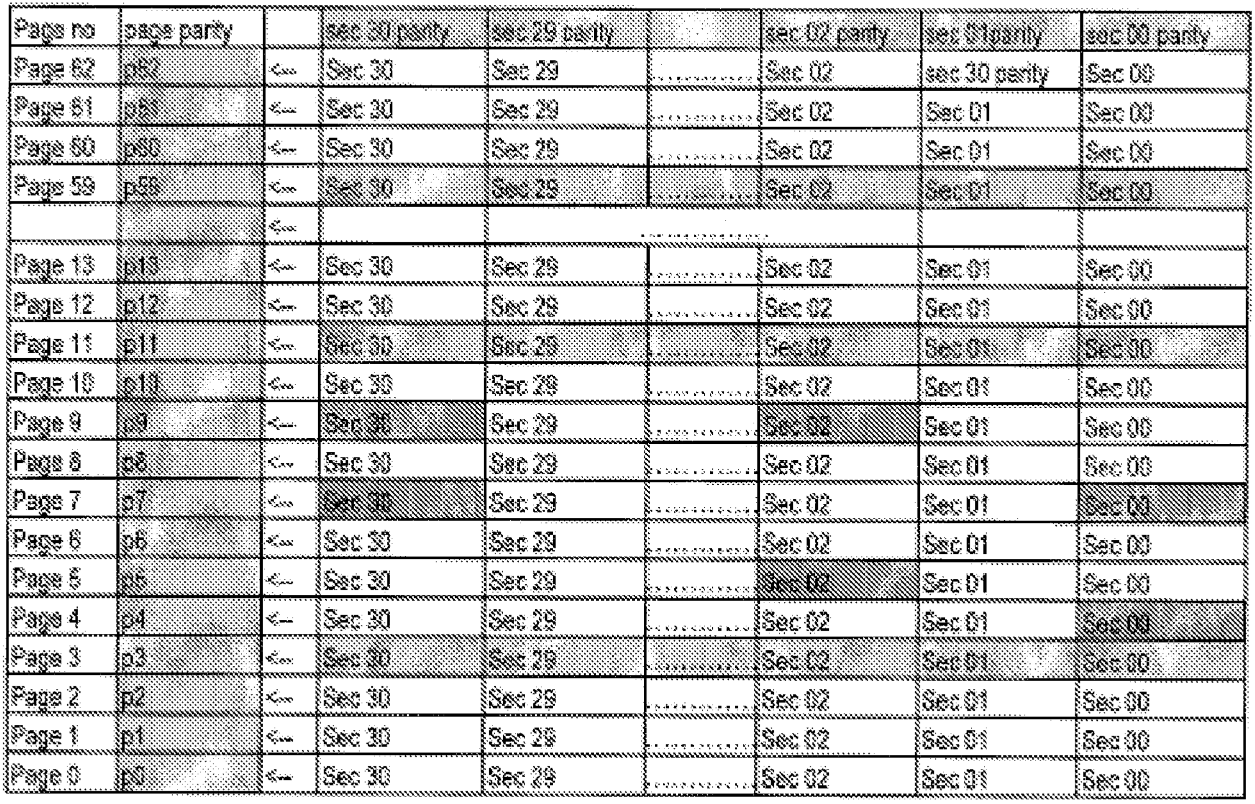

Grouping of parity blocks
Group 1
Parity Block (0-3)
Page 0
Page 1
Page 2
Page 3
Page 4
Page 5
Page 6
Page 7
Group 1
Parity Block (8-11)
Page 8
Page 9
Page 10
Page 11
Page 12
Page 13
Page 14
Page 15

FIG. 11

2 dimensional RAID protection - 1



Page no	page parity		sec 30 parity	sec 29 parity		sec 02 parity	sec 01parity	sec 00 parity
Page 62	p62	<--	Sec 30	Sec 29	....	Sec 02	sec 30 parity	Sec 00
Page 61	p61	<--	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 60	p60	<--	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 59	p59	<--	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
		<--		....				
Page 13	p13	<--	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 12	p12	<--	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 11	p11	<--	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 10	p10	<--	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 9	p9	<--	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 8	p8	<--	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 7	p7	<--	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 6	p6	<--	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 5	p5	<--	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 4	p4	<--	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 3	p3	<--	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 2	p2	<--	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 1	p1	<--	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 0	p0	<--	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00

Parity sectors and pages are in shaded blocks as ▒ and corrupted sectors are in blocks as ▨

FIG. 12

2 dimensional RAID protection -2

Page no	page parity		sec 30 parity	sec 29 parity		sec 02 parity	sec 01parity	sec 00 parity
Page 62	p62	<–	Sec 30	Sec 29	....	Sec 02	sec 30 parity	Sec 00
Page 61	p61	<–	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 60	p60	<–	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 59	p59	<–	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
		<–			....			
Page 13	p13	<–	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 12	p12	<–	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 11	p11	<–	Sec 30	Sec 29		Sec 02	Sec 01	Sec 00
Page 10	p10	<–	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 9	p9	<–	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 8	p8	<–	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 7	p7	<–	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 6	p6	<–	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 5	p5	<–	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 4	p4	<–	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 3	p3	<–	Sec 30	Sec 29		Sec 02	Sec 01	Sec 00
Page 2	p2	<–	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 1	p1	<–	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00
Page 0	p0	<–	Sec 30	Sec 29	....	Sec 02	Sec 01	Sec 00

Parity sectors and pages are in shaded blocks as ░ and corrupted sectors are in blocks as ▨

FIG. 13

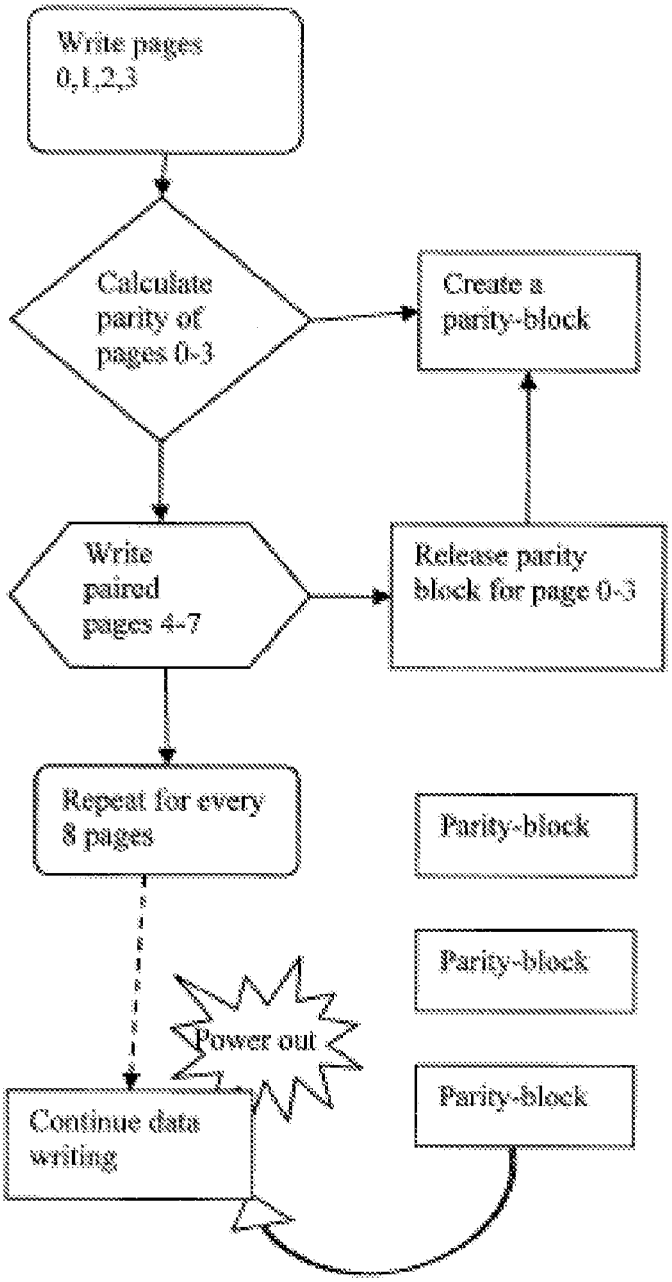

Write pages 0,1,2,3
Calculate parity of pages 0-3
Create a parity-block
Wrise paired pages 4-7
Release parity block for page 0-3
Repeat for every 8 pages
Parity-block
Parity-block
Power out
Parity-block
Continue data writing

FIG. 14