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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 November 2012 (22.11.2012)

WIPO | PCT

(10) International Publication Number
WO 2012/156889 A1

(51) International Patent Classification:
G11C 16/04 (2006.01) *G06F 12/02* (2006.01)

(21) International Application Number:
PCT/IB2012/052375

(22) International Filing Date:
13 May 2012 (13.05.2012)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/486,330 16 May 2011 (16.05.2011) US
13/338,335 28 December 2011 (28.12.2011) US

(71) Applicant *(for all designated States except US)*: **APPLE INC.** [US/US]; 1 Infinite Loop, Cupertino, CA 95014 (US).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: **GURGI, Eyal** [IL/IL]; 10 Hameginim Street, 49442 Petach Tikva (IL).

(74) Agent: **KIVLIN, Noel, B.**; Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C, P.O. Box 398, Austin, TX 78767-0398 (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, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

Published:

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

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

(54) **Title**: SPARSE PROGRAMMING OF ANALOG MEMORY CELLS

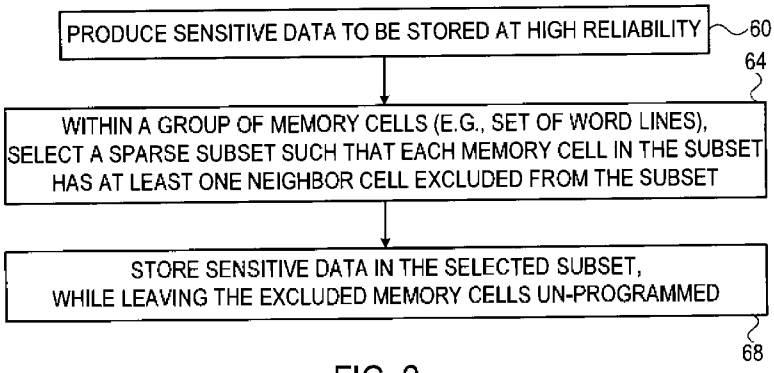

FIG. 2

(57) **Abstract**: A method for data storage in a memory (24) including an array (28) of analog memory cells (32), includes selecting a group of the memory cells such that each memory cell in the group has one or more neighbor memory cells in the array that are excluded from the group. Data is stored in the group of the memory cells while excluding the neighbor memory cells from programming as long as the data is stored in the group of the memory cells.

WO 2012/156889 A1

SPARSE PROGRAMMING OF ANALOG MEMORY CELLS

FIELD OF THE INVENTION

The present invention relates generally to memory devices, and particularly to methods and systems for data storage in analog memory cells.

BACKGROUND OF THE INVENTION

Some memory devices, such as Flash devices, comprise arrays of analog memory cells. In some cases, neighboring memory cells in the array may cause cross-coupling interference to one another. Various techniques for reducing interference in memory devices are known in the art. Example techniques are described in U.S. Patent Application Publications 2008/0198650, 2009/0024905, 2009/0158126, 2010/0131826, whose disclosures are incorporated herein by reference.

Some memory systems employ means for protecting critical information. For example, U.S. Patent Application Publication 2009/0193184, whose disclosure is incorporated herein by reference, describes a hybrid Solid-State Disk (SSD) having Multi-Level-Cell (MLC) or Single-Level-Cell (SLC) Flash memory, or both. SLC Flash may be emulated by MLC that uses fewer cell states. Data reliability is improved since fewer MLC states are used, and noise margins may be relaxed.

SUMMARY OF THE INVENTION

An embodiment of the present invention that is described herein provides a method for data storage in a memory that includes an array of analog memory cells. A group of the memory cells is selected such that each memory cell in the group has one or more neighbor memory cells in the array that are excluded from the group. Data is stored in the group of the memory cells while excluding the neighbor memory cells from programming as long as the data is stored in the group of the memory cells.

In some embodiments, selecting the group includes excluding from the group all the immediate neighbor memory cells of each memory cell in the group. Selecting the group may include, for at least a part of the array, excluding from the group all the memory cells belonging to odd-order word lines or all the memory cells belonging to even-order word lines. In another embodiment, selecting the group includes, for at least a part of the array, excluding from the group all the memory cells belonging to odd-order bit lines or all the memory cells belonging to even-order bit lines.

In a disclosed embodiment, selecting the group includes, for each memory cell in the group, excluding from the group any neighbor memory cells that is adjacent to the memory cell and belongs to a same word line or bit line as the memory cell. In yet another embodiment, the memory cells in the array are assigned to hold n bits per cell, and storing the data includes programming the memory cells in the group with only k bits per cell, $k<n$.

In some embodiments, storing the data includes writing the data using programming commands that store a given data page in all the memory cells of a respective word line. In alternative embodiments, storing the data includes writing the data using programming commands that store a given data page either in odd-order or even-order memory cells of a respective word line.

In some embodiments, storing the data includes storing sensitive information in the group of the memory cells. In an embodiment, the sensitive information includes management information and/or power-down information. In a disclosed embodiment, storing the data includes applying to the data an additional protection mechanism that is not applied to other data stored outside the group. Applying the additional protection mechanism may include storing the data at a lower storage density relative to the other data; encoding the data with a stronger Error Correction Code (ECC) relative to the other data; and/or programming the memory cells in the group using a sequence of programming pulses that increase by an increment that is smaller relative to increments used outside the group.

There is additionally provided, in accordance with an embodiment of the present invention, a data storage apparatus including an interface and storage circuitry. The interface is configured to communicate with a memory including an array of analog memory cells. The storage circuitry is configured to select a group of the memory cells such that each memory cell in the group has one or more neighbor memory cells in the array that are excluded from the group, and to store data in the group of the memory cells while excluding the neighbor memory cells from programming as long as the data is stored in the group of the memory cells.

There is also provided, in accordance with an embodiment of the present invention, a data storage apparatus including a memory and storage circuitry. The memory includes an array of analog memory cells. The storage circuitry is configured to select a group of the memory cells such that each memory cell in the group has one or more neighbor memory cells in the array that are excluded from the group, and to store data in the group of the memory cells while excluding the neighbor memory cells from programming as long as the data is stored in the group of the memory cells.

The present invention will be more fully understood from the following detailed description of the embodiments thereof, taken together with the drawings in which:

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram that schematically illustrates a memory system, in accordance with an embodiment of the present invention; and

Fig. 2 is a flow chart that schematically illustrates a method for data storage, in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF EMBODIMENTS

OVERVIEW

Data is typically stored in analog memory cells by programming the memory cells to certain analog values, such as electrical charge levels or voltages. The data is retrieved from the memory cells by sensing the analog values. In some cases, memory cells may inflict cross-coupling interference on one another. This interference may distort the analog values read from the memory cells and cause read errors.

Embodiments of the present invention that are described herein provide improved methods and systems for storing data in analog memory cells. The disclosed techniques store data by programming a group of memory cells sparsely, such that each memory cell has one or more neighbor cells that remain un-programmed with data. This sort of programming helps to reduce cross-coupling interference, since the level of interference in a memory cell typically grows with the magnitudes of the analog values of its neighbors.

Several examples of sparse storage schemes are described hereinbelow. The disclosed schemes are simple to implement and do not incur considerable computational load, for example in comparison with schemes that cancel cross-coupling interference at readout time. The sparse storage schemes described herein differ from one another in the number of un-programmed neighbors per memory cell, and thus provide different trade-offs between data reliability and storage density.

The methods and systems described herein can be used both in memory devices that store different pages in different subsets of the memory cells in a word line (e.g., different pages in odd-order and even-order cells) and in All Bit Line (ABL) memory devices that store a page in all memory cells of a word line. The disclosed techniques are particularly suitable for storing sensitive information, such as management information, whose corruption would affect large amounts of stored data.

SYSTEM DESCRIPTION

Fig. 1 is a block diagram that schematically illustrates a memory system 20, in accordance with an embodiment of the present invention. System 20 can be used in various host systems and devices, such as in computing devices, cellular phones or other communication terminals, removable memory modules (sometimes referred to as "USB Flash Drives"), Solid State Disks (SSD), digital cameras, music and other media players and/or any other system or device in which data is stored and retrieved.

System 20 comprises a memory device 24, which stores data in a memory cell array 28. The memory array comprises multiple memory blocks 34. Each memory block 34 comprises multiple analog memory cells 32. In the context of the present patent application and in the claims, the term "analog memory cell" is used to describe any memory cell that holds a continuous, analog value of a physical parameter, such as an electrical voltage or charge. Array 28 may comprise analog memory cells of any kind, such as, for example, NAND, NOR and Charge Trap Flash (CTF) Flash cells, phase change RAM (PRAM, also referred to as Phase Change Memory - PCM), Nitride Read Only Memory (NROM), Ferroelectric RAM (FRAM), magnetic RAM (MRAM) and/or Dynamic RAM (DRAM) cells.

The charge levels stored in the cells and/or the analog voltages or currents written into and read out of the cells are referred to herein collectively as analog values, analog storage values or storage values. The storage values may comprise, for example, threshold voltages or any other suitable kind of storage values. System 20 stores data in the analog memory cells by programming the cells to assume respective programming states, which are also referred to as programming levels. The programming states are selected from a finite set of possible states, and each programming state corresponds to a certain nominal storage value. For example, a 3 bit/cell MLC can be programmed to assume one of eight possible programming states by writing one of eight possible nominal storage values into the cell.

Memory device 24 comprises a reading/writing (R/W) unit 36, which converts data for storage in the memory device to analog storage values and writes them into memory cells 32. In alternative embodiments, the R/W unit does not perform the conversion, but is provided with voltage samples, i.e., with the storage values for storage in the cells. When reading data out of array 28, R/W unit 36 converts the storage values of memory cells 32 into digital samples having a resolution of one or more bits. Data is typically written to and read from the memory cells in groups that are referred to as pages. In some embodiments, the R/W unit can erase a group of cells 32 by applying one or more negative erasure pulses to the cells. Erasure is typically performed in entire memory blocks.

The storage and retrieval of data in and out of memory device 24 is performed by a memory controller 40. The memory controller comprises an interface 44 for communicating with memory device 24, and a processor 48 that carries out the various memory management functions. In particular, processor 48 carries stores at least some of the data using sparse, high-reliability storage schemes that are described in detail below.

Memory controller 40 communicates with a host 52, for accepting data for storage in the memory device and for outputting data retrieved from the memory device. Memory controller 40, and in particular processor 48, may be implemented in hardware. Alternatively, the memory controller may comprise a microprocessor that runs suitable software, or a combination of hardware and software elements.

The configuration of Fig. 1 is an exemplary system configuration, which is shown purely for the sake of conceptual clarity. Any other suitable memory system configuration can also be used. Elements that are not necessary for understanding the principles of the present invention, such as various interfaces, addressing circuits, timing and sequencing circuits and debugging circuits, have been omitted from the figure for clarity.

Although the example of Fig. 1 shows a single memory device 24, system 20 may comprise multiple memory devices that are controlled by memory controller 40. In the exemplary system configuration shown in Fig. 1, memory device 24 and memory controller 40 are implemented as two separate Integrated Circuits (ICs). In alternative embodiments, however, the memory device and the memory controller may be integrated on separate semiconductor dies in a single Multi-Chip Package (MCP) or System on Chip (SoC), and may be interconnected by an internal bus. Further alternatively, some or all of the memory controller circuitry may reside on the same die on which the memory array is disposed. Further alternatively, some or all of the functionality of memory controller 40 can be implemented in software and carried out by a processor or other element of the host system. In some embodiments, host 44 and memory controller 40 may be fabricated on the same die, or on separate dies in the same device package.

In some embodiments, memory controller 40 comprises a general-purpose processor, which is programmed in software to carry out the functions described herein. The software may be downloaded to the processor in electronic form, over a network, for example, or it may, alternatively or additionally, be provided and/or stored on non-transitory tangible media, such as magnetic, optical, or electronic memory.

In an example configuration of array 28, memory cells 32 are arranged in multiple rows and columns, and each memory cell comprises a floating-gate transistor. The gates of the

transistors in each row are connected by word lines, and the sources of the transistors in each column are connected by bit lines. The memory array is typically divided into multiple pages, i.e., groups of memory cells that are programmed and read simultaneously. Pages are sometimes sub-divided into sectors. In some embodiments, each page comprises an entire row of the array. In alternative embodiments, each row (word line) can be divided into two or more pages. For example, in some devices each row is divided into two pages, one comprising the odd-order cells and the other comprising the even-order cells.

Typically, memory controller 40 programs data in page units, but erases entire memory blocks 34. Typically although not necessarily, a memory block is on the order of 10^6 memory cells, whereas a page is on the order of 10^3-10^4 memory cells.

HIGH-RELIABILITY STORAGE USING SPARSE PROGRAMMING

A given memory cell 32 in memory array 28 may be subject to cross-coupling interference from neighboring memory cells in the array. In the present context, the term "neighboring memory cell" is used broadly and refers to any memory cell that potentially inflicts cross-coupling interference on the memory cell in question. Neighboring memory cells may be immediately adjacent to the interfered memory cell (e.g., vertical neighbors on the same bit line, horizontal neighbors on the same word line or diagonal neighbors), or non-adjacent memory cells that nevertheless potentially cause interference.

Typically, the level of interference caused by a certain neighboring cell grows with the magnitude of the interfering cell's analog value. Therefore, a neighboring cell that is programmed with data (and particularly a neighboring cell programmed with data that corresponds to a high analog value) will typically cause stronger interference than an un-programmed memory cell. Un-programmed memory cells usually cause little or no interference, because they are set to an erased state represented by a relatively small negative analog value.

In some embodiments that are described herein, processor 48 stores certain data using a storage scheme that retains, for each programmed memory cell, one or more neighboring memory cells un-programmed with data. In other words, processor 48 stores the data such that each memory cell being programmed with data has one or more neighbor cells that remain un-programmed as long as the data is stored is held in the programmed cells. Such storage schemes are referred to herein as sparse storage schemes. In the description that follows, the sparse storage scheme is defined and applied by processor 48 in the memory controller. In

alternative embodiments, however, the storage scheme may be applied by R/W unit 36 in the memory device. Thus, processor 48 and/or unit 36 are referred to herein collectively as storage circuitry that carries out the disclosed storage techniques.

In some embodiments, processor 48 applies the sparse storage scheme to a selected group of memory cells, e.g., a set of word lines or a memory block. Within this group of memory cells, processor 48 selects a partial subset of memory cells for programming, such that each memory cell in the subset has at least one neighboring cell that is excluded from the subset. Processor 48 stores data in the group by programming the memory cells in the partial subset while retaining the excluded memory cells un-programmed with data. As a result of the sparse programming, each programmed memory cell suffers relatively low cross-coupling interference, and the readout error probability is therefore reduced.

In various embodiments, processor 48 may use various kinds of sparse programming schemes. In the examples given below, memory cells 32 comprise four-level memory cells, each storing two bits of data. In each memory cell, one bit is referred to as Least Significant Bit (LSB) and one bit is referred to as Most Significant Bit (MSB). The memory cells along each word line are divided into even-order and odd-order cells (i.e., the memory cells belonging to the even- and odd-order bit lines, respectively). The memory controller stores two memory pages (referred to as LSB and MSB pages) in the even-order cells, and two additional pages memory pages (LSB and MSB) in the odd-order cells. This configuration, however, is chosen purely by way of example. In alternative embodiments, any other suitable memory configuration can be used.

In one embodiment, processor 48 applies a sparse storage scheme that is defined in the following table:

Word line / Bit lines		Even-order bit lines	Odd-order bit lines
0	LSB page	0	
	MSB page	1	
1	LSB page		
	MSB page		
2	LSB page	2	
	MSB page	3	
3	LSB page		
	MSB page		

…	…	…

In this scheme, a page is first written to the LSB page of the even-order bit lines on word line 0, then to the MSB page of the even-order bit lines on word line 0, then to the LSB page of the even-order bit lines on word line 2, then to the MSB page of the even-order bit lines on word line 2, and so on. Odd-order word lines and odd-order bit lines are not programmed at all.

As can be seen in the table, the programmed memory cells (on the intersections of the even-order word lines and even-order bit lines) have no immediate neighbors (i.e., cells that are directly adjacent to the memory cell, either diagonally, vertically or horizontally) that are programmed with data. Therefore, the memory cells experience little or no cross-coupling interference. On the other hand, this scheme stores data at only 25% of the capacity of the memory – since only a quarter of the memory cells are actually programmed.

In another embodiment, processor 48 applies a sparse storage scheme that is defined in the following table:

Word line	Bit lines	Even-order bit lines	Odd-order bit lines
0	LSB page	0	
	MSB page		
1	LSB page		
	MSB page		
2	LSB page	1	
	MSB page		
3	LSB page		
	MSB page		
…		…	…

In this scheme, data is programmed only to the LSB pages of the even-order bit lines on the even-order word lines. As in the previous scheme, the programmed memory cells in the present scheme have no adjacent neighbors that are programmed with data. Therefore, the memory cells experience little or no cross-coupling interference. Moreover, the present scheme stores data at a reduced density of 1 bit/cell instead of 2 bits/cell (by storing only the LSB page and not the MSB page), and therefore is more resilient to read errors. The high

resilience comes at the expense of reduced capacity – The present scheme utilizes 12.5% of the memory capacity.

In yet another embodiment, processor 48 applies a sparse storage scheme that is defined in the following table:

Word line	Bit lines	Even-order bit lines	Odd-order bit lines
0	LSB page	0	1
	MSB page	2	3
1	LSB page		
	MSB page		
2	LSB page	4	5
	MSB page	6	7
3	LSB page		
	MSB page		
…		…	…

In this scheme too, only the even-order word lines are programmed. Memory pages are written alternately to the even-order and odd-order bit lines until the word line is fully programmed, and then programming advances to the next word line. In the present example, word line 0 is first programmed with four pages: A page is written to the LSB page of the even-order bit lines, a second page is written to the LSB page of the odd-order bit lines, then a page is written to the MSB page of the even-order bit lines, and a fourth page is written to the MSB page of the odd-order bit lines. The process is repeated on word line 2 (skipping word line 1), and so on.

In this sparse storage scheme, memory cells may suffer from some horizontal cross-coupling interference, since the horizontal neighbors of each programmed memory cell are also programmed. Vertical and diagonal interference, however, are eliminated. This scheme uses 50% of the memory capacity.

In another alternative embodiment, processor 48 applies a sparse storage scheme that is defined in the following table:

Word line	Bit lines	Even-order bit lines	Odd-order bit lines
0	LSB page	0	

	MSB page	1	
1	LSB page		2
	MSB page		3
2	LSB page	4	
	MSB page	5	
3	LSB page		6
	MSB page		7
…		…	…

In this scheme, processor 48 programs the even-order bit lines on the even-order word lines, and the odd-order bit lines on the odd-order word lines. Both LSB and MSB pages are programmed. In this sparse storage scheme, the vertical and horizontal neighboring cells are not programmed, but the diagonal neighbors are programmed. This scheme uses 50% of the memory capacity. (In an alternative but equivalent embodiment, processor 48 may program the even-order bit lines on the odd-order word lines, and the odd-order bit lines on the even-order word lines.)

In still another embodiment, processor 48 applies a sparse storage scheme that is defined in the following table:

Word line / Bit lines		Even-order bit lines	Odd-order bit lines
0	LSB page	0	
	MSB page		
1	LSB page		1
	MSB page		
2	LSB page	2	
	MSB page		
3	LSB page		3
	MSB page		
…		…	…

This scheme is similar to the previous example, with the addition that only the LSB pages are programmed. This sparse storage scheme has improved resilience to interference, at the expense of using only 25% of the memory capacity. As in the previous scheme, in an

alternative embodiment processor 48 may program the even-order bit lines on the odd-order word lines, and the odd-order bit lines on the even-order word lines.

The example sparse programming schemes described above are chosen purely for the sake of conceptual clarity. In alternative embodiments, any other suitable programming scheme, which retains at least one un-programmed neighboring cell for each programmed memory cell, can be used. For example, the above-described schemes are applied in a memory that supports separate page write commands for programming the even- and odd-order memory cells in a given word line.

In some embodiments, the disclosed techniques are used in memory devices that program all the memory cells of a given word line in the same command (sometimes referred to as All Bit Lines – ABL – devices), e.g., by programming every second memory cell. This technique helps to reduce horizontal cross-coupling interference, i.e., interference from horizontal neighbors on the same word line as the interfered cell.

In some embodiments, processor 48 uses the sparse programming schemes described herein for increasing the storage reliability of sensitive information, as opposed to non-sensitive information. In these embodiments, the use of sparse programming is restricted to sensitive information because these techniques usually come at the expense of memory capacity. As such, the disclosed techniques may be used in specific areas of array 28, for example in specific sets of word lines in each block 34. In other areas of the memory, processor 48 may store the non-sensitive data while utilizing all the memory cells. Nevertheless, in alternative embodiments the disclosed techniques can be used for storing some or even all non-sensitive data, if desired.

The disclosed techniques enable processor 48 to store data with high reliability but with little or no Error Correction Code (ECC) redundancy. For example, the disclosed techniques, using an ECC rate on the order of 97%, can achieve the effective strength of an ECC having 25% rate and conventional programming schemes. This sort of configuration has performance benefits both during programming (less data is programmed, less resources and latency spent on ECC encoding) and during readout (less data is read, less resources and latency spent on ECC decoding).

Sensitive information may comprise any information whose loss or corruption would affect large amounts of stored data, or whose loss or corruption would cause additional damage or performance degradation beyond the loss of that specific information. Specific examples of sensitive information may comprise management data such as boot code (particularly if the boot time is restricted, so as to limit the boot code read time) or power-

down data for which reliability should typically be high and program time is of importance. Non-sensitive information may comprise, for example, user data that is received from the host. Alternatively, processor 48 may use sparse programming for storing any other suitable type of information.

In some embodiments, processor 48 combines a sparse programming scheme with one or more additional protection mechanisms that increase storage reliability. Typically, the additional protection mechanism is not used other than for sparse storage. For example, processor 48 may store the data sparsely, and also using less than the number of bits per cell designated to memory cells 32. Consider, for example, a system in which processor 48 stores user data in memory cells 32 using n bits/cell, e.g., using 2^n programming levels. In some embodiments, processor 48 may store sensitive information using sparse programming and also using only k bits/cell (2^k programming levels), $k<n$.

As another example, consider a system in which processor 48 encodes the user data with an ECC having a certain redundancy level (e.g., a certain code rate). In some embodiments, processor 48 may store sensitive information using sparse programming and also encode the sensitive information with an ECC having a higher redundancy level (e.g., lower code rate).

As yet another example, consider a system in which R/W unit 36 programs memory cells 32 in an iterative Programming and Verification (P&V) process. In such a process, the R/W unit applies to a group of memory cells a sequence of programming pulses that progressively increase in magnitude and/or duration. The analog values of the memory cells are verified during the programming sequence, and subsequent pulses are applied only to memory cells that have not yet reached their intended analog values.

In some embodiments, processor 48 may store user data by programming the memory cells using a pulse sequence that increases from pulse to pulse by a certain magnitude and/or duration increment, and store sensitive information by programming the memory cells using a pulse sequence that increases from pulse to pulse by a smaller increment. The smaller increment typically increases the storage accuracy, at the expense of programming speed.

In some embodiments, processor 48 may apply any suitable combination of the above-described protection mechanisms, or any other suitable protection mechanism for increasing storage reliability along with sparse programming.

Fig. 2 is a flow chart that schematically illustrates a method for data storage, in accordance with an embodiment of the present invention. The method begins with processor

48 of memory controller 40 producing sensitive information that is to be stored in memory device 24, at an information generation step 60.

Processor 48 selects a sparse subset of memory cells for storing the sensitive information, at a selection step 64. In an embodiment, processor 48 assigns a certain group of memory cells, e.g., a set of word lines, for storing the sensitive information. Within this group, processor 48 selects a partial subset of the memory cells that will actually be programmed. The other memory cells in the group will be excluded from programming as long as the sensitive information is stored in the partial subset.

The partial subset is selected such that each cell in the subset (which will be programmed) has at least one neighbor cell in the group that will be excluded from programming. Processor 48 may select the partial subset according to any suitable sparse programming scheme, such as the example schemes described above. Processor 48 stores the sensitive information in the selected subset of the memory cells, at a sparse programming step 68.

Although the embodiments described herein mainly address devices of analog memory cells such as Flash memory, the methods and systems described herein can also be used in other applications, such as in three-dimensional memory devices.

It will thus be appreciated that the embodiments described above are cited by way of example, and that the present invention is not limited to what has been particularly shown and described hereinabove. Rather, the scope of the present invention includes both combinations and sub-combinations of the various features described hereinabove, as well as variations and modifications thereof which would occur to persons skilled in the art upon reading the foregoing description and which are not disclosed in the prior art. Documents incorporated by reference in the present patent application are to be considered an integral part of the application except that to the extent any terms are defined in these incorporated documents in a manner that conflicts with the definitions made explicitly or implicitly in the present specification, only the definitions in the present specification should be considered.

CLAIMS

1. A method for data storage, comprising:

in a memory comprising an array of analog memory cells, selecting a group of the memory cells such that each memory cell in the group has one or more neighbor memory cells in the array that are excluded from the group; and

storing data in the group of the memory cells while excluding the neighbor memory cells from programming as long as the data is stored in the group of the memory cells.

2. The method according to claim 1, wherein selecting the group comprises excluding from the group all the immediate neighbor memory cells of each memory cell in the group.

3. The method according to claim 1, wherein selecting the group comprises, for at least a part of the array, excluding from the group all the memory cells belonging to odd-order word lines or all the memory cells belonging to even-order word lines.

4. The method according to claim 1, wherein selecting the group comprises, for at least a part of the array, excluding from the group all the memory cells belonging to odd-order bit lines or all the memory cells belonging to even-order bit lines.

5. The method according to claim 1, wherein selecting the group comprises, for each memory cell in the group, excluding from the group any neighbor memory cells that is adjacent to the memory cell and belongs to a same word line or bit line as the memory cell.

6. The method according to any of claims 1-5, wherein the memory cells in the array are assigned to hold n bits per cell, and wherein storing the data comprises programming the memory cells in the group with only k bits per cell, $k<n$.

7. The method according to any of claims 1-5, wherein storing the data comprises writing the data using programming commands that store a given data page in all the memory cells of a respective word line.

8. The method according to any of claims 1-5, wherein storing the data comprises writing the data using programming commands that store a given data page either in odd-order or even-order memory cells of a respective word line.

9. The method according to any of claims 1-5, wherein storing the data comprises storing sensitive information in the group of the memory cells.

10. The method according to claim 9, wherein the sensitive information comprises at least one information type selected from a group of types consisting of management information and power-down information.

11. The method according to any of claims 1-5, wherein storing the data comprises applying to the data an additional protection mechanism that is not applied to other data stored outside the group.

12. The method according to claim 11, wherein applying the additional protection mechanism comprises performing at least one action selected from a group of actions consisting of:

storing the data at a lower storage density relative to the other data;

encoding the data with a stronger Error Correction Code (ECC) relative to the other data; and

programming the memory cells in the group using a sequence of programming pulses that increase by an increment that is smaller relative to increments used outside the group.

13. A data storage apparatus, comprising:

an interface, which is configured to communicate with a memory comprising an array of analog memory cells; and

storage circuitry, which is configured to select a group of the memory cells such that each memory cell in the group has one or more neighbor memory cells in the array that are excluded from the group, and to store data in the group of the memory cells while excluding the neighbor memory cells from programming as long as the data is stored in the group of the memory cells.

14. The apparatus according to claim 13, wherein the storage circuitry is configured to exclude from the group all the immediate neighbor memory cells of each memory cell in the group.

15. The apparatus according to claim 13, wherein, for at least a part of the array, the storage circuitry is configured to exclude from the group all the memory cells belonging to odd-order word lines or all the memory cells belonging to even-order word lines.

16. The apparatus according to claim 13, wherein, for at least a part of the array, the storage circuitry is configured to exclude from the group all the memory cells belonging to odd-order bit lines or all the memory cells belonging to even-order bit lines.

17. The apparatus according to claim 13, wherein the storage circuitry is configured to exclude from the group, for each memory cell in the group, any neighbor memory cells that is adjacent to the memory cell and belongs to a same word line or bit line as the memory cell.

18. The apparatus according to any of claims 13-17, wherein the memory cells in the array are assigned to hold n bits per cell, and wherein the storage circuitry is configured to store the data by programming the memory cells in the group with only k bits per cell, $k<n$.

19. The apparatus according to any of claims 13-17, wherein the storage circuitry is configured to store the data using programming commands that store a given data page in all the memory cells of a respective word line.

20. The apparatus according to any of claims 13-17, wherein the storage circuitry is configured to store the data using programming commands that store a given data page either in odd-order or even-order memory cells of a respective word line.

21. The apparatus according to any of claims 13-17, wherein the storage circuitry is configured to store sensitive information in the group of the memory cells while excluding the neighbor memory cells.

22. The apparatus according to claim 21, wherein the sensitive information comprises at least one information type selected from a group of types consisting of management information and power-down information.

23. The apparatus according to any of claims 13-17, wherein the storage circuitry is configured to apply to the data an additional protection mechanism that is not applied to other data stored outside the group.

24. The apparatus according to claim 23, wherein the storage circuitry is configured to apply the additional protection mechanism by performing at least one action selected from a group of actions consisting of:

storing the data at a lower storage density relative to the other data;

encoding the data with a stronger Error Correction Code (ECC) relative to the other data; and

programming the memory cells in the group using a sequence of programming pulses that increase by an increment that is smaller relative to increments used outside the group.

25. A data storage apparatus, comprising:

a memory comprising an array of analog memory cells; and

storage circuitry, which is configured to select a group of the memory cells such that each memory cell in the group has one or more neighbor memory cells in the array that are excluded from the group, and to store data in the group of the memory cells while excluding the neighbor memory cells from programming as long as the data is stored in the group of the memory cells.

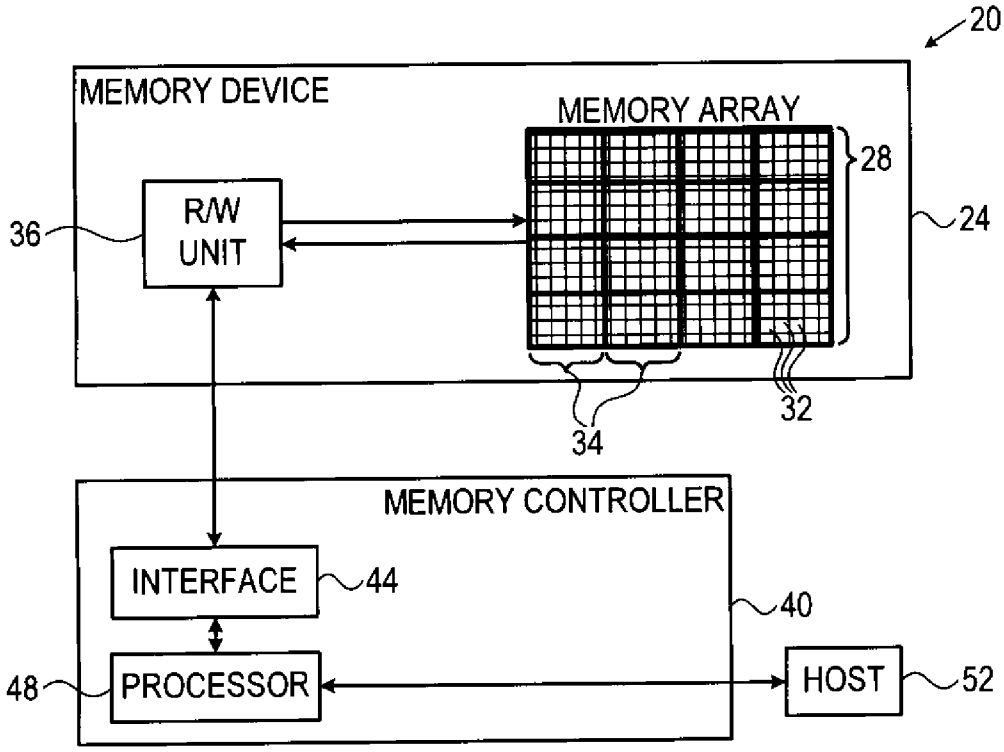
20
MEMORY DEVICE
MEMORY ARRAY
28
24
R/W
UNIT
36
34
32
MEMORY CONTROLLER
INTERFACE
44
40
PROCESSOR
48
HOST
52

FIG. 1

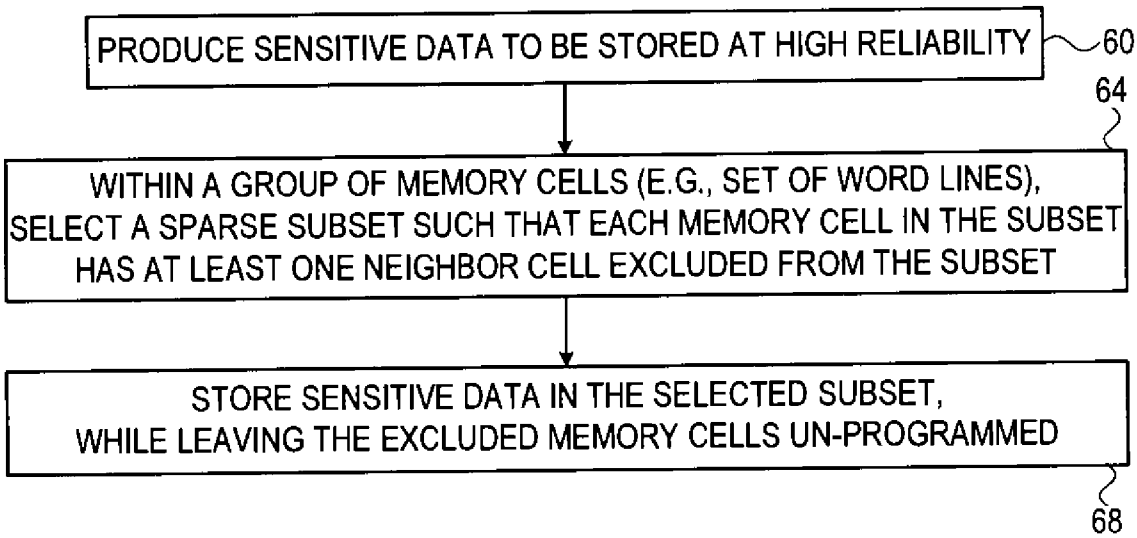
PRODUCE SENSITIVE DATA TO BE STORED AT HIGH RELIABILITY
60
64
WITHIN A GROUP OF MEMORY CELLS (E.G., SET OF WORD LINES),
SELECT A SPARSE SUBSET SUCH THAT EACH MEMORY CELL IN THE SUBSET
HAS AT LEAST ONE NEIGHBOR CELL EXCLUDED FROM THE SUBSET
STORE SENSITIVE DATA IN THE SELECTED SUBSET,
WHILE LEAVING THE EXCLUDED MEMORY CELLS UN-PROGRAMMED
68

FIG. 2

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IB12/52375

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G11C 16/04; G06F 12/02 (2012.01)
USPC - 365/185.03; 711/1/173
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)
IPC(8) Classification(s): G11C 15/26, 16/04, 15/26, 7/16; G06F 12/02, 12/06 (2012.01)
USPC Classification(s): 365/185.01,185.03, 185.18, 185.25; 711/1,5,170,173,212

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
MicroPatent (US-G, US-A, EP-A, EP-B, WO, JP-bib, DE-C,B, DE-A, DE-T, DE-U, GB-A, FR-A); Google Patent Search; DialogPro (Derwent, INSPEC, NTIS, PASCAL, Current Contents Search, Dissertation Abstracts Online, Inside Conferences); IP.com; Search Terms Used: memory, cell, neighbor, interference, program

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2011/0044102 A1 (DONG, Y. et al) February 24, 2011, Figure 11A and Paragraphs [0038] and [0039]	1-5, 6/1, 6/2, 6/3, 6/4, 6/5, 7/1, 7/2, 7/3, 7/4, 7/5, 8/1, 8/2, 8/3, 8/4, 8/5, 11/1, 11/2, 11/3, 11/4, 11/5, 12/11/1, 12/11/2, 12/11/3, 12/11/4, 12/11/5, 13-17, 18/13, 18/14, 18/15, 18/16, 18/17, 19/13, 19/14, 19/15, 19/16, 19/17, 20/13, 20/14, 20/15, 20/16, 20/17, 23/13, 23/14, 23/15, 23/16, 23/17, 24/23/13, 24/23/14, 24/23/15, 24/23/16, 24/23/17 and 25
---		---
Y		9/1, 9/2, 9/3, 9/4, 9/5, 10/9/1, 10/9/2, 10/9/3, 10/9/4, 10/9/5, 21/13, 21/14, 21/15, 21/16, 21/17, 22/21/13, 22/21/14, 22/21/15, 22/21/16 and 22/21/17

☒ Further documents are listed in the continuation of Box C. ☐

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
16 September 2012 (16.09.2012)	16 OCT 2012
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer: Shane Thomas PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IB12/52375

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 7,333,364 B2 (YU, F. et al.) February 19, 2008, Column 5, Lines 39-45	9/1, 9/2, 9/3, 9/4, 9/5, 10/9/1, 10/9/2, 10/9/3, 10/9/4, 10/9/5, 21/13, 21/14, 21/15, 21/16, 21/17, 22/21/13, 22/21/14, 22/21/15, 22/21/16 and 22/21/17