(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 2 291 746 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
29.10.2014 Bulletin 2014/44

(21) Application number: **09767334.7**

(22) Date of filing: **28.05.2009**

(51) Int Cl.:
G06F 12/00 *(2006.01)* ***G06F 12/02*** *(2006.01)*
G11C 16/34 *(2006.01)*

(86) International application number:
PCT/US2009/045386

(87) International publication number:
WO 2009/155022 (23.12.2009 Gazette 2009/52)

(54) **HYBRID MEMORY MANAGEMENT**

HYBRID-SPEICHERVERWALTUNG

GESTION DE MÉMOIRE HYBRIDE

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO SE SI SK TR

(30) Priority: **28.05.2008 US 127945**

(43) Date of publication of application:
09.03.2011 Bulletin 2011/10

(73) Proprietor: **Round Rock Research, LLC**
Jersey City, NJ 07302 (US)

(72) Inventors:
- **RADKE, William, H.**
  Los Gatos
  CA 95030 (US)
- **MURRAY, Michael**
  Mountain View
  CA 94041 (US)
- **FURUHJELM, Martin, Ragnar**
  Grass Valley, CA 95949 (US)
- **GELDMAN, John**
  Los Gatos
  CA 65030 (US)

(74) Representative: **Beresford, Keith Denis Lewis et al**
Beresford & Co.
16 High Holborn
London
WC1V 6BX (GB)

(56) References cited:
WO-A1-2007/037757
KR-A- 20080 040 426
US-A1- 2008 071 973
US-A1- 2008 104 309
US-A1- 2008 104 309

- **LI-PIN CHANG ED - TADAYOSHI ENOMOTO ET AL: "Hybrid solid-state disks: Combining heterogeneous NAND flash in large SSDs", DESIGN AUTOMATION CONFERENCE, 2008. ASPDAC 2008. ASIA AND SOUTH PACIFIC, IEEE, PISCATAWAY, NJ, USA, 21 March 2008 (2008-03-21), pages 428-433, XP031241387, ISBN: 978-1-4244-1921-0**
- **'Asia and South Pacifie Design Automation Conference(ASPD AC) IEEE, 21-24 March 2008', article LI-PIN CHANG: 'Hybrid Solid-State Disks: Combining Heterogeneous NAND Flash in Large SSDs.', pages 428 - 433, XP031241387**

EP 2 291 746 B1

Description

TECHNICAL FIELD

[0001] The present disclosure relates generally to memory devices and in particular the present disclosure relates to managing data stored in a memory device having single and multi level memory cell storage capability.

BACKGROUND

[0002] Memory devices are typically provided as internal, semiconductor, integrated circuits in computers or other electronic devices. There are many different types of memory including random-access memory (RAM), read only memory (ROM), dynamic random access memory (DRAM), synchronous dynamic random access memory (SDRAM), and flash memory.

[0003] Flash memory devices have developed into a popular source of non-volatile memory for a wide range of electronic applications. Flash memory devices typically use a one-transistor memory cell that allows for high memory densities, high reliability, and low power consumption. Common uses for flash memory include personal computers, personal digital assistants (PDAs), digital cameras, and cellular telephones. Program code and system data, such as a basic input/output system (BIOS), are typically stored in flash memory devices for use in personal computer systems.

[0004] Flash memory typically utilizes one of two basic architectures known as NOR Flash and NAND Flash. The designation is derived from the logic used to read the devices. In a NAND type flash memory array architecture, the floating gate memory cells of the memory array are arranged in a matrix of rows and columns. The memory cells of the array are also arranged together in strings, typically of 8, 16, 32, or more each, where the memory cells in the string are connected together in series, source to drain, between a common source line and a column transfer line, often referred to as a bit line. The array is then accessed by a row decoder activating a row of floating gate memory cells by selecting the word line connected to their gates. In addition, bit lines can also be driven high or low depending on the current operation being performed.

[0005] As the performance and complexity of electronic systems increase, the requirement for additional memory in a system also increases. However, in order to continue to reduce the costs of the system, it is desirable to keep the parts count low. This can be accomplished by increasing the memory density of an integrated circuit by using such technologies as multilevel cells (MLC). For example, MLC NAND flash memory is a cost effective non-volatile memory.

[0006] Multilevel memory cells assign a data state (e.g., as represented by a bit pattern) to a specific range of threshold voltages (Vt) stored on the memory cell. Single level memory cells (SLC) permit the storage of a single bit of data on each memory cell. Meanwhile, MLC technology permits the storage of two or more bits per cell (e.g., 2, 4, 8, 16 bits), depending on the quantity of threshold voltage ranges assigned to the cell and the stability of the assigned threshold voltage ranges during the lifetime operation of the memory cell. The number of threshold voltage ranges (e.g., levels), which are sometimes referred to as Vt distribution windows, used to represent a bit pattern comprised of N-bits is 2^N. For example, one bit may be represented by two levels, two bits by four levels, three bits by eight levels, etc.

[0007] For example, a cell may be assigned a Vt that falls within one of four different voltage ranges of 200 mV, each being used to represent a data state corresponding to a bit pattern comprised of two bits. Typically, a dead space (which is sometimes referred to as a margin) of 0.2V to 0.4V is maintained between each range to keep the Vt distributions from overlapping. If the voltage stored on the cell is within the first of the four Vt distributions, the cell in this case is storing a logical '11' state and is typically considered the erased state of the cell. If the voltage is within the second of the four Vt distributions, the cell in this case is storing a logical '10' state. A voltage in the third distribution of the four Vt distributions would indicate that the cell in this case is storing a logical '00' state. Finally, a Vt residing in the fourth Vt distribution indicates that a logical '01' state is stored in the cell.

[0008] There are advantages and disadvantages associated with using SLC or MLC memory. MLC memory is generally considered more cost effective in regards to memory density as MLC memory can, for example, store multiple bits of data in a single memory cell as opposed to SLC memory which is conventionally used to store one bit of data per cell. However, conventional SLC memory can be written to many (e.g., by an order magnitude) more times than conventional MLC memory. For example, a characteristic of conventional MLC memory is that after data has been erased and re-written about 10,000 times the memory may become subject to significant read and write errors. Conventional SLC memory on the other hand typically may be erased and re-written about 100,000 times before the reliability of the data begins to deteriorate. These density and performance characteristics also apply between different types of MLC arrays. MLC devices with four and eight levels exist today, while denser memories are being researched. While an MLC with more levels would be more efficient (higher density) than an MLC with less levels (lower density), these higher density devices may have performance penalties over the lower density devices. The case of a device built with SLC (two level) and MLC (more than two level) can be generalized as a device with multiple memory arrays, each with its own density and performance trade-offs. An example is a device built with an array of MLC (four level) and an array of MLC (eight level). There could even be more than two arrays of memories, such as SLC, MLC (four level) and MLC (eight level). A common naming

convention is to refer to SLC memory as MLC(two level) memory as SLC memory utilizes two levels in order to store one bit of data as represented by a 0 or a 1, for example. MLC memory configured to store two bits of data can be represented by MLC(four level), three bits of data by MLC(eight level), etc. An MLC(four level) memory cell is typically referred to as a lower density memory cell than an MLC(eight level) memory due to the lower number of bits stored per memory cell, for example. SLC (e.g., MLC(two level)) is typically referred to as a lower density memory than MLC (four level) memory and so on.

[0009] United States Patent Application Publication No.US 2008/0126680 A1 entitled "Non-volatile Memory System Storing Data In Single-Level Cell or Multi-level Cell According to Data Characteristics" by Yang-sup Lee et. al. describes a system storing data received from an application or file system in a non-volatile memory system of single-level cells and multi-level cells in accordance with one or more data characteristics.

[0010] The document entitled "Hybrid solid-state disks: Combining heterogeneous NAND flash in large SSDs" by Li-Pin Chang et. al. published in the proceedings for the "Design Automation Conference", 2008. ASPDAC 2008. Asia and South Pacific, IEEE, Piscataway, NJ, USA, 21 March 2008, pages 428 - 433, XP031241387 describes a hybrid approach to large SSDs, which combines MLC flash and SLC flash. The aim is to complement the drawbacks of the two kinds of NAND flash with each other's advantages. The technical issues of the design of a hybrid SSD pertain to data placement and wear leveling over heterogeneous NAND flash. The experimental results show that, by adding a 256 MB SLC flash to a 20 GB MLC-flash array, the hybrid SSD improves over a conventional SSD by 4.85 times in terms of average response. The average throughput and energy consumption are improved by 17 % and 14%, respectively. The hybrid SSD is only 2% more expensive than a purely MLC-flash-based SSD, for which the extra cost is limited and very rewarded.

[0011] United States Patent Application Publication No.US 2008/010 4309 A1 entitled "Flash Memory Device with Multi-level Cells and Method of Writing Data Therein" by Won-Moon Cheon et. al. describes a method of writing data in a flash memory system. The flash memory system forms an address mapping pattern according to a log block mapping scheme. The method includes determining a writing pattern of data to be written in a log block, and allocating one of SLC and MLC blocks to the log block in accordance with the writing pattern of the data.

[0012] The document WO 2007-037757 A1 discloses a hybrid SLC/MLC flash memory, wherein access to the different memory arrays can be done simultaneously (via a separate data bus) or alternatively (via a common bus), for optimising the different operation speeds and storage capacities.

[0013] The document US 2008-0071973 A1 discloses a flash memory card, whereby a memory controller uses an index for converting logical addresses into physical addresses.

[0014] For the reasons stated above, and for other reasons stated below which will become apparent to those skilled in the art upon reading and understanding the present specification, there is a need in the art for hybrid memory devices that are adapted to manage the utilization of memory of different densities, such as SLC and MLC memory, to take advantage of preferred operating characteristics associated with each type of memory.

Summary of the invention

[0015] The present invention is defined in the appended independent claims to which reference should be made. Advantageous features are set out in the appended dependent claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016]

Figure 1 is a functional block diagram of a memory module according to an embodiment of the present disclosure.

Figure 2 is a flow chart illustrating multiple operations according to various embodiments of the present disclosure.

Figure 3 illustrates one storage configuration of a memory device according to an embodiment of the present disclosure.

FIG. 4 illustrates a move operation of data according to an embodiment of the present disclosure.

FIG. 5 illustrates a write operation of data according to an embodiment of the present disclosure.

FIG. 6 is a functional block diagram of an electronic system having at least one memory device according to an embodiment of the present disclosure.

DETAILED DESCRIPTION

[0017] In the following detailed description of the embodiments, reference is made to the accompanying drawings that form a part hereof. In the drawings, like numerals describe substantially similar components throughout the several views. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention. Other embodiments may be utilized and structural, logical, and electrical changes may be made without departing from the scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present disclosure is defined only by the appended claims.

[0018] As discussed above, conventional SLC and MLC memory have both positive and negative attributes associated with them. SLC memory allows for faster write operations and can withstand far more write operations before reliability of the memory cells begin to degrade as

compared to MLC memory. However, SLC memory is not as efficient as MLC memory in that MLC memory can store multiple bits of data on each memory cell whereas SLC is used to store only a single bit per cell. Various embodiments of the present disclosure store data that is frequently updated in SLC memory and store data that is updated less frequently in MLC memory. This can be used to enhance the efficiency and reliability of memory devices. Although some embodiments are disclosed with reference to utilizing SLC and MLC memory, the various embodiments are not so limited. For example, one embodiment may utilize SLC memory, sometimes referred to as MLC (two level) memory, and MLC (four level) memory. Another embodiment may utilize MLC (four level) and MLC (eight level) memory. Still other embodiments can utilize three or more memory arrays such as SLC, MLC (four level) and MLC (eight level) memory in the same memory device, for example. Other combinations are possible according to the various embodiments of the present disclosure. Thus the embodiments can utilize a combination of different level (e.g., density) memory in a memory device.

[0019] Data in a non-volatile memory device can be accessed through an abstraction called a logical block address (LBA) that does not define where the data physically resides in the device. The device can also have physical block addresses (PBA) that define a physical location, but does not define or imply what data is kept in this location. In a magnetic disk device, a physical block address translates to a specific cylinder, head, and sector. In a solid state non-volatile memory device, the physical block address typically refers to a specific memory block address in a specific memory array. Logical block addresses and physical block addresses are well known to those skilled in the art. Memory devices maintain look up tables which map LBAs to their assigned PBAs. Conventional memory devices that have both SLC and MLC memory store data in either SLC memory or MLC memory. The data does not move between the SLC and MLC memory. This is in contrast with various embodiments of the present disclosure which cause the data stored in the memory device to move between SLC and MLC, such as based on the usage (e.g., number of write operations performed) of a given LBA. This usage is determined (e.g., tracked) and maintained for the LBAs of the memory device according to various embodiments of the present disclosure. Various embodiments of the present disclosure might also predict the usage for a given LBA and assign it to SLC or MLC memory accordingly. For example many file systems keep a data structure which is used to reference files to LBAs, such as a File Allocation Table (FAT), LBAs associated with such structures are likely to be subject to write operations on every file write operation. Thus, LBAs associated with the FAT could initially be assigned to SLC memory instead of MLC memory. Other embodiments adjust the location of a given LBA in SLC or MLC memory by acting in response to the actual tracked usage of each LBA. According to some embodiments of the present disclosure, an LBA may move between the SLC and MLC memory based on the actual usage of the LBA during operation of the memory device regardless of how the LBA was initially assigned.

[0020] Figure 1 illustrates a memory device according to an embodiment of the present disclosure. The memory device in Figure 1 has been simplified to focus on various embodiments of the present disclosure. The memory device 100 comprises a controller 170 for controlling and managing the operations of the memory device 100 according to various embodiments of the present disclosure including managing the assignment of LBAs to either SLC or MLC (or between different density MLC) memory. The controller 170 can take the form of discrete logic or a state machine, for example. The controller 170 also incorporates various means for communicating with a host, such as a processor 110. For example, the controller 170 may incorporate a Universal Serial Bus (USB), SATA, PATA, ATAB-ACS, SD, MMC, Compact Flash, Memory Stick, IEEE 1394 or BA-NAND interface as are well known in the art. Physical wear leveling of the SLC and MLC memory can also be handled by the controller 170.

[0021] The memory device 100 also comprises an array of SLC memory 132 and an array of MLC memory 134. In some embodiments, the SLC 132 and MLC 134 memory may be separate flash memory chips while in others the SLC and MLC memory may be contained on one chip. The memory arrays 132 and 134 may also be comprised of different density MLC memory. Array 132 might be MLC(four level) and array 134 might be MLC(eight level), for example. The SLC 132 and MLC 134 memory illustrated in Figure 1 may each be further comprised of multiple banks and blocks of memory. Each of the SLC 132 and MLC 134 memory blocks shown in Figure 1 can be comprised of multiple memory chips. The amount (e.g. capacity) of SLC and MLC memory of the memory device 100 may or may not be equal. In some embodiments (e.g. a single chip comprising SLC and MLC memory) the size of the SLC array and MLC array may be assignable depending on the desired amount of SLC vs. MLC memory for a given application. In another embodiment according to the present disclosure, the memory device of Figure 1 may be a two-way interleaved memory device having two SLC chips and two MLC chips. Other quantities of SLC and MLC chips are also possible according to various embodiments of the present disclosure.

[0022] The memory device of Figure 1 also includes a usage table 136. The usage table is utilized by the controller 170 to store usage data (e.g. write operations performed) for the various LBAs of the memory device 100. For example, each time a write operation is performed on an LBA, the associated usage data for that LBA will updated. The usage table 136 may also store usage information (e.g., a time stamp) indicating when a given LBA was last written to as opposed to the number of times it has been written to. For example, LBAs may be

mapped based on how much time has elapsed (e.g., a minute, a day, a month, etc.) since a particular LBA was last written to instead of a cumulative number of times it has been written to. In addition to assigning a time stamp to an LBA, LBAs may instead be assigned to a time group. LBAs assigned to a first time group may represent LBAs used in the current month whereas LBAs assigned to a second time group may represent LBAs used last month, for example. According to various embodiments, usage might also comprise tracking which LBAs are utilized at power up of the memory device or during a particular time frame following power up. The particular time frame may also follow a reset operation, for example. These LBAs may then be assigned to lower density (e.g., SLC) memory which can typically be accessed faster than higher density (e.g., MLC(four level)) memory. This can improve start up performance such as reducing access time during a boot load operation, for example. Usage data stored in the usage table 136 is not limited to usage of individual LBAs. Usage data may also be stored pertaining to use of multiple or ranges of LBAs. The usage table can also be cleared if desired. For example, a host 110 connected to the memory device 100 might send a particular command instructing that all or part of the usage table 136 be cleared. An example of a usage table is shown in block 136 of Figure 1.

[0023] The usage table 136 may be stored in a standalone component such as an integrated circuit device having one or both of a volatile and non-volatile memory portion. For embodiments having both volatile and non-volatile memory, the volatile memory can maintain the current usage table during operation of the memory device. The usage table could then periodically be copied from the volatile memory to the non-volatile memory. The current usage table 136 can also be loaded into the volatile memory at power up and transferred back to non-volatile memory during power down of the memory device. Other embodiments allow for the usage table 136 data to be stored in the memory array 132/134 of the memory device 100. For example, the usage table data may be stored along with (e.g. appended to) the data associated with the LBA in the memory device. In other embodiments, the usage table may be stored in a dedicated location of either the SLC 132 or MLC 134 memory.

[0024] Figure 1 also illustrates the memory device 100 coupled to a processor 110. The memory device 100 is coupled to the processor 110 by way of an interface 174, which may comprise multiple busses and signals. For example, control signals generated by the processor can be coupled to the memory device by a control bus 172. Additionally, an address bus 142 and a data bus 162 are also shown in the figure. The interface 174 may conform to one of the interface protocols discussed above (e.g. USB, SATA, PATA, et al.).

[0025] Figure 2 illustrates a flow chart of some of the various activities performed by the controller 170 in implementing embodiments of the present disclosure. In some embodiments of the present disclosure, the controller of the memory device determines where an LBA 200 is to be assigned in the memory device 100. As discussed above, the FAT table is likely to be frequently used. Thus, according to one embodiment of the present disclosure, the controller 170 of the memory device 100 may assign LBAs associated with the FAT to SLC (e.g., MLC(two level) memory 214. The FAT LBAs according to this embodiment may also be permanently assigned (e.g. 'pinned') to SLC memory. In other embodiments, the controller may assign the FAT LBAs to MLC memory 206 and the controller 170 can move the FAT tables based on usage. Thus, according to one embodiment, all LBAs may be initially written to MLC memory 206. In other embodiments, all LBAs may be initially assigned to SLC memory 214. Still other embodiments may assign all LBAs having a write operation to be performed thereon to SLC memory. In these embodiments, should the LBA currently being written to be already assigned to MLC memory, the controller 170 can determine if the LBA being written to should ultimately be assigned to SLC or MLC memory.

[0026] The controller also tracks and maintains (e.g., updates) 208/216 the usage table data for LBAs of the memory device during operation. If the usage of an LBA currently assigned to MLC memory exceeds some threshold value 210, the controller will attempt to move the data associated with the LBA (and reassign the LBA to the location in) to SLC memory 212. In one embodiment, this threshold value may be 1000 write operations performed on a given LBA. The embodiments however are not limited to a single threshold value. For example, the threshold value may be some fraction of the total write operations performed on the memory 212. If the usage for all the LBAs assigned to MLC memory remain less than the threshold value, then those LBAs will continue to be assigned to the MLC memory.

[0027] According to one embodiment, each time an LBA is assigned to SLC memory, either from the controller 214 or because an LBA assigned to MLC memory has exceeded some threshold value 212, a determination is made regarding whether a sufficient number of spare locations remain in the SLC memory 218. Spare locations are desirable to allow for data handling and housekeeping functions to be performed on the memory device. If sufficient spares will exist in the SLC after the current write operation to SLC takes place, no further action by the controller is necessary and the LBA is assigned to the SLC memory. If however, the current assignment of an LBA to SLC memory will result in too few remaining spare locations in SLC memory, the controller will perform data management functions according to various embodiments of the present disclosure in order to preserve a minimum number of spare locations in the SLC memory. For example, the controller will review the usage data for each LBA currently assigned to SLC. Whichever LBA currently assigned to SLC memory has the least usage associated with it, the data associated with that LBA will be moved to MLC memory 220 (e.g., the next

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.