(12) STANDARD PATENT (11) Application No. **AU 2010234647 B2**
(19) AUSTRALIAN PATENT OFFICE

(54) Title
Partitioning a flash memory data storage device

(51) International Patent Classification(s)
G06F 13/38 (2006.01) *G06F 3/06* (2006.01)

(21) Application No: **2010234647** (22) Date of Filing: **2010.04.05**

(87) WIPO No: **WO10/117929**

(30) Priority Data

(31) Number	(32) Date	(33) Country
12/537,741	**2009.08.07**	**US**
61/167,709	**2009.04.08**	**US**
61/187,835	**2009.06.17**	**US**

(43) Publication Date: **2010.10.14**
(44) Accepted Journal Date: **2013.08.15**

(71) Applicant(s)
Google Inc.

(72) Inventor(s)
Borchers, Albert T.;Swing, Andrew T.;Sprinkle, Robert S.

(74) Agent / Attorney
Pizzeys, PO Box 291, WODEN, ACT, 2606

(56) Related Art
US 2008/0320214 A1

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 October 2010 (14.10.2010)

PCT

(10) International Publication Number
WO 2010/117929 A1

(51) International Patent Classification:
G06F 13/38 (2006.01) *G06F 3/06* (2006.01)

(21) International Application Number:
PCT/US2010/029917

(22) International Filing Date:
5 April 2010 (05.04.2010)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

61/167,709	8 April 2009 (08.04.2009)	US
61/187,835	17 June 2009 (17.06.2009)	US
12/537,741	7 August 2009 (07.08.2009)	US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:
US 12/537,741 (CON)
Filed on 7 August 2009 (07.08.2009)

(71) Applicant *(for all designated States except US)*: **GOOGLE INC.** [US/US]; 1600 Amphitheatre Parkway, Mountain View, California 94043 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **BORCHERS, Albert, T.** [US/US]; 60 Nichols Drive, Santa Cruz, California 95060 (US). **SWING, Andrew, T.** [US/US]; 16370 Lucky Road, Los Gatos, California 95030 (US). **SPRINKLE, Robert, S.** [US/US]; 549 Oak Street, Mountain View, California 94041 (US).

(74) Agents: **BELLERMANN, Mark, R., W.** et al.; Brake Hughes Bellermann LLP, c/o CPA Global, P.O. Box 52050, 900 2nd Ave S, Suite 1560, Minneapolis, Minnesota 55402 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, 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, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (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, 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))*

(54) Title: PARTITIONING A FLASH MEMORY DATA STORAGE DEVICE

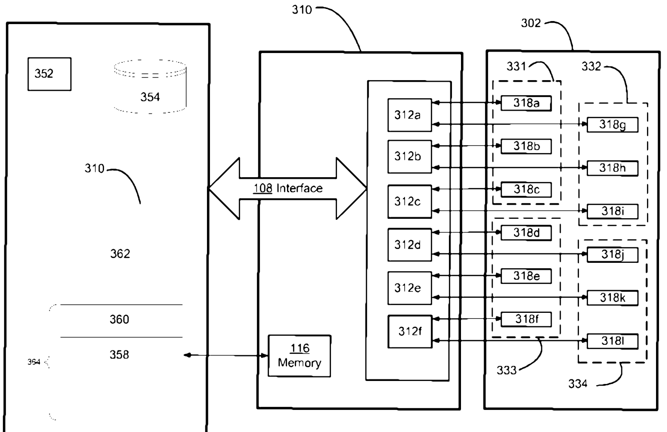

FIG. 3B

(57) **Abstract**: A method of partitioning a data storage device (100, 302) that has a plurality of memory chips (118a, 118b, 218, 318) includes determining a number memory chips (118a, 118b, 218, 318) in the data storage device (100, 302), defining, via a host (106, 350) coupled to the data storage device (100, 302), a first partition (321, 331) of the data storage device (100, 302), where the first partition (321, 331) includes a first subset of the plurality of memory chips (118a, 118b, 218, 318), defining a second partition (322, 332) of the data storage device (100, 302) via the host (106, 350) where the second partition (322, 332) includes a second subset of the plurality of memory chips (118a, 118b, 218, 318), such that the first subset does not include any memory chips (118a, 118b, 218, 318) of the second subset and wherein the second subset does not include any memory chips (118a, 118b, 218, 318) of the first subset.

PARTITIONING A FLASH MEMORY DATA STORAGE DEVICE

Inventors:

Albert T. Borchers, Andrew T. Swing, Robert S. Sprinkle

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of U.S. Provisional Application No. 61/167,709, filed April 8, 2009, and titled “DATA STORAGE DEVICE” and U.S. Provisional Application No. 61/187,835, filed June 17, 2009, and titled “PARTITIONING AND STRIPING IN A FLASH MEMORY DATA STORAGE DEVICE,” both of which are hereby incorporated by reference in entirety.

TECHNICAL FIELD

[0002] This description relates to a data storage device.

BACKGROUND

[0003] Data storage devices may be used to store data. A data storage device may be used with a computing device to provide for the data storage needs of the computing device. In certain instances, it may be desirable to store large amounts of data on a data storage device. Also, it may be desirable to execute commands quickly to read data from and to write data to the data storage device.

SUMMARY

[0004] According to one general aspect, a method of partitioning a data storage device that includes a plurality of memory chips is disclosed. The method includes determining a number memory chips in the data storage device. A host coupled to the data storage device defines a first partition and a second of the data storage device, where the first partition includes a first subset of the plurality of memory chips and where the second partition includes a second subset of the plurality of memory chips. The first subset does not include any memory chips of the second subset and wherein the second subset does not include any memory chips of the first subset.

[0005] In another general aspect, a method of partitioning a data storage device that includes a plurality of memory chips is disclosed, in which a physical configuration of the data storage device, including the number memory chips in the data storage device, and a partitioning scheme for the data storage device are read. A host coupled to the data storage device defines a first partition and a second partition of the data storage device, where the first partition includes a first subset of the plurality of memory chips and the second partition includes a second subset of the plurality of memory chips, and where the first subset does not include any memory chips of the second subset and wherein the second subset does not include any memory chips of the first subset. A logical to physical memory map is allocated for the first partition, and a logical to physical memory map is allocated for the first partition.

[0006] Implementations can include one or more of the following features. Data can be written to the first partition while reading data from the second partition. Determining a number memory chips in the data storage device can include transmitting information from the data storage device to the host indicating the number of memory chips in the data storage device. The host can define an address location in the data storage device to which to write data from the host, where the address location specifies that the data be written to a specific one of the plurality of memory chips.

[0007] The data storage device can include a plurality of physical channels for communication of data between the host and the plurality of memory chips, each channel being operably connected to a different plurality of the memory chips, and then the number of physical channels can be determined. A first subset and a second subset of the channels can be defined, where channels of the first subset of the channels are operably connected only to memory chips of the first subset of memory chips and where channels of the second subset of the channels are operably connected only to memory chips of the second subset of memory chips. And the host can define an address location in the data storage device to which to write data from the host, where the address location specifies that the data be written to a specific one of the plurality of memory chips through a specific channel. The first partition can include memory chips that are operably connected to a single channel.

[0008] The host can re-define the first partition of the data storage device to include a third subset of the plurality of memory chips, where the third subset is different from the first subset, and where the third subset does not include any memory chips of the second subset and where the second subset does not include any memory chips of the third subset. An indication that one of the memory chips in the first subset has failed or is approaching failure can be received, and then re-defining the first partition can include defining the third subset as the first subset of memory chips but for the memory chip that has failed or that is approaching failure.

[0009] In another general aspect, an apparatus includes a data storage device that includes a plurality of memory chips, and a host operably coupled to the data storage device via an interface. The host includes a configuration detection engine configured to detect the number of memory chips in the data storage device, and a partition engine. The partition engine is configured to define a first partition of the data storage device, where the first partition includes a first subset of the plurality of memory chips and to define a second partition of the data storage device, where the second partition includes a second subset of the plurality of memory chips, and where the first subset does not include any memory chips of the second subset and where the second subset does not include any memory chips of the first subset.

[0010] In another general aspect, an apparatus includes a data storage device that includes a plurality of memory chips, and a host operably coupled to the data storage device via an interface. The host includes a configuration detection engine configured to read a physical configuration of the data storage device, including the number memory chips in the data storage device and configured to read a partitioning scheme for the data storage device. The host also includes a partition engine configured to define a first partition of the data storage device, where the first partition includes a first subset of the plurality of memory chips and to define a second partition of the data storage device, where the second partition includes a second subset of the plurality of memory chips, and configured to allocate a logical to physical memory map for the first partition and to allocate a logical to physical memory map for the second partition. The first subset does not include any memory chips of the second subset, and the second subset does not include any memory chips of the first subset.

[0011] Implementations can include one or more of the following features.

For example, the data storage device can be configured to transmit, upon receiving a command from the host, information from the data storage device to the host indicating the number of memory chips in the data storage device. The host can also include an address assignment engine configured to assign a memory address to data to be written to the data storage device, where the assigned memory address specifies that the data be written to a specific one of the plurality of memory chips.

[0012] The data storage device can include a plurality of physical channels for communication of data between the host and the plurality of memory chips, with each channel being operably connected to a different plurality of the memory chips. Then, the configuration detection engine can be further configured to detect the number of channels in the data storage device, and the partition engine can be further configured to define a first subset of the channels, where channels of the first subset of the channels are operably connected only to memory chips of the first subset of memory chips, and where the partition engine can be further configured to define a second subset of the channels, where channels of the second subset of the channels are operably connected only to memory chips of the second subset of memory chips.

[0013] The host can further include an address assignment engine configured to assign a memory address to data to be written to the data storage device, where the assigned memory address specifies that the data be written to a specific one of the plurality of memory chips through a specific channel. The first partition can include memory chips that are operably connected to a single channel. The partition engine can be further configured to re-define the first partition of the data storage device to include a third subset of the plurality of memory chips, where the third subset is different from the first subset, and where the third subset does not include any memory chips of the second subset and wherein the second subset does not include any memory chips of the third subset. The partition engine can be further configured to receive an indication that one of the memory chips in the first subset has failed or is approaching failure; and re-defining the first partition to include the third subset of the plurality of memory chips can include defining the third subset as the first subset of memory chips but for the memory chip that has failed or that is approaching failure.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] FIG. 1 is an exemplary block diagram of a data storage device.

[0015] FIG. 2 is an exemplary block diagram of a FPGA controller that can be used in the data storage device of FIG. 1.

[0016] FIG. 3A is an exemplary block diagram of exemplary computing devices for use with the data storage device of FIG. 1.

[0017] FIG. 3B is an exemplary block diagram of exemplary computing devices for use with the data storage device of FIG. 1.

[0018] FIG. 4 is an exemplary flowchart illustrating an example process of partitioning the data storage device of FIG. 1.

[0019] FIG. 5 is another exemplary flowchart illustrating an example process of partitioning the data storage device of FIG. 1.

DETAILED DESCRIPTION

[0020] This document describes an apparatus, system(s) and techniques for data storage. Such a data storage apparatus may include a controller board having a controller that may be used with one or more different memory boards, with each of the memory boards having multiple flash memory chips. The data storage apparatus may communicate with a host using an interface on the controller board. In this manner, the controller on the controller board may be configured to receive commands from the host using the interface and to execute those commands using the flash memory chips on the memory boards.

[0021] FIG. 1 is a block diagram of a data storage device 100. The data storage device 100 may include a controller board 102 and one or more memory boards 104a and 104b. The data storage device 100 may communicate with a host 106 over an interface 108. The interface 108 may be between the host 106 and the controller board 102. The controller board 102 may include a controller 110, a DRAM 111, multiple channels 112, a power module 114, and a memory module 116. The memory boards 104a and 104b may include multiple flash memory chips 118a and 118b on each of the memory boards. The memory boards 104a and 104b also may include a memory device 120a and 120b.

[0022] In general, the data storage device 100 may be configured to store data on the flash memory chips 118a and 118b. The host 106 may write data to and read data from the flash memory chips 118a and 118b, as well as cause other operations to be performed with respect to the flash memory chips 118a and 118b. The reading and writing of data between the host 106 and the flash memory chips 118a and 118b, as well as the other operations, may be processed through and controlled by the controller 110 on the controller board 102. The controller 110 may receive commands from the host 106 and cause those commands to be executed using the flash memory chips 118a and 118b on the memory boards 104a and 104b. The communication between the host 106 and the controller 110 may be through the interface 108. The controller 110 may communicate with the flash memory chips 118a and 118b using the channels 112.

[0023] The controller board 102 may include DRAM 111. The DRAM 111 may be operably coupled to the controller 110 and may be used to store information. For example, the DRAM 111 may be used to store logical address to physical address maps and bad block information. The DRAM 111 also may be configured to function as a buffer between the host 106 and the flash memory chips 118a and 118b.

[0024] In one exemplary implementation, the controller board 102 and each of the memory boards 104a and 104b are physically separate printed circuit boards (PCBs). The memory board 104a may be on one PCB that is operably connected to the controller board 102 PCB. For example, the memory board 104a may be physically and/or electrically connected to the controller board 102. Similarly, the memory board 104b may be a separate PCB from the memory board 104a and may be operably connected to the controller board 102 PCB. For example, the memory board 104b may be physically and/or electrically connected to the controller board 102.

[0025] The memory boards 104a and 104b each may be separately disconnected and removable from the controller board 102. For example, the memory board 104a may be disconnected from the controller board 102 and replaced with another memory board (not shown), where the other memory board is operably connected to controller board 102. In this example, either or both of the memory boards 104a and 104b may be swapped out with other memory boards such that the other memory boards may operate with the same controller board 102 and controller 110.

[0026] In one exemplary implementation, the controller board 102 and each of the memory boards 104a and 104b may be physically connected in a disk drive form factor. The disk drive form factor may include different sizes such as, for example, a 3.5" disk drive form factor and a 2.5" disk drive form factor.

[0027] In one exemplary implementation, the controller board 102 and each of the memory boards 104a and 104b may be electrically connected using a high density ball grid array (BGA) connector. Other variants of BGA connectors may be used including, for example, a fine ball grid array (FBGA) connector, an ultra fine ball grid array (UBGA) connector and a micro ball grid array (MBGA) connector. Other types of electrical connection means also may be used.

[0028] The interface 108 may include a high speed interface between the controller 110 and the host 106. The high speed interface may enable fast transfers of data between the host 106 and the flash memory chips 118a and 118b. In one exemplary implementation, the high speed interface may include a Peripheral Component Interconnect Express ("PCIe") interface. For instance, the PCIe interface may be a PCIe x4 interface or a PCIe x8 interface. The PCIe interface 108 may include a PCIe connector cable assembly to the host 106. In this example, the 110 may include an interface controller configured to interface between the host 106 and the interface 108. The interface controller may include a PCIe endpoint controller. Other high speed interfaces, connectors, and connector assemblies also may be used.

[0029] In one exemplary implementation, the communication between the controller board 102 and the flash memory chips 118a and 118b on the memory boards 104a and 104b may be arranged and configured into multiple channels 112. Each of the channels 112 may communicate with one or more flash memory chips 118a and 118b. The controller 110 may be configured such that commands received from the host 106 may be executed by the controller 110 using each of the channels 112 simultaneously or at least substantially simultaneously. In this manner, multiple commands may be executed simultaneously on different channels 112, which may improve throughput of the data storage device 100.

[0030] In the example of FIG. 1, twenty (20) channels 112 are illustrated. The completely solid lines illustrate the ten (10) channels between the controller 110 and the flash memory chips 118a on the memory board 104a. The mixed solid and dashed lines illustrate the ten (10) channels between the controller 110 and the flash memory chips 118b on the memory board 104b. As illustrated in FIG. 1, each of the channels 112 may support multiple flash memory chips. For instance, each of the channels 112 may support up to 32 flash memory chips. In one exemplary implementation, each of the 20 channels may be configured to support and communicate with 6 flash memory chips. In this example, each of the memory boards 104a and 104b would include 60 flash memory chips each. Depending on the type and the number of the flash memory chips 118a and 118b, the data storage 100 device may be configured to store up to and including multiple terabytes of data.

[0031] The controller 110 may include a microcontroller, a FPGA controller, other types of controllers, or combinations of these controllers. In one exemplary implementation, the controller 110 is a microcontroller. The microcontroller may be implemented in hardware, software, or a combination of hardware and software. For example, the microcontroller may be loaded with a computer program product from memory (e.g., memory module 116) including instructions that, when executed, may cause the microcontroller to perform in a certain manner. The microcontroller may be configured to receive commands from the host 106 using the interface 108 and to execute the commands. For instance, the commands may include commands to read, write, copy and erase blocks of data using the flash memory chips 118a and 118b, as well as other commands.

[0032] In another exemplary implementation, the controller 110 is a FPGA controller. The FPGA controller may be implemented in hardware, software, or a combination of hardware and software. For example, the FPGA controller may be loaded with firmware from memory (e.g., memory module 116) including instructions that, when executed, may cause the FPGA controller to perform in a certain manner. The FPGA controller may be configured to receive commands from the host 106 using the interface 108 and to execute the commands. For instance, the commands may include commands to read, write, copy and erase blocks of data using the flash memory chips 118a and 118b, as well as other commands.

[0033] The memory module 116 may be configured to store data, which may be loaded to the controller 110. For instance, the memory module 116 may be configured to store one or more images for the FPGA controller, where the images include firmware for use by the FPGA controller. The memory module 116 may interface with the host 106 to communicate with the host 106. The memory module 116 may interface directly with the host 106 and/or may interface indirectly with the host 106 through the controller 110. For example, the host 106 may communicate one or more images of firmware to the memory module 116 for storage. In one exemplary implementation, the memory module 116 includes an electrically erasable programmable read-only memory (EEPROM). The memory module 116 also may include other types of memory modules.

[0034] The memory boards 104a and 104b may be configured to operate with different types of flash memory chips 118a and 118b. In one exemplary implementation, the flash memory chips 118a and the flash memory chips 118b may be the same type of flash memory chips including requiring the same voltage from the power module 114 and being from the same flash memory chip vendor. The terms vendor and manufacturer are used interchangeably throughout this document.

[0035] In another exemplary implementation, the flash memory chips 118a on the memory board 104a may be a different type of flash memory chip from the flash memory chips 118b on the memory board 104b. For example, the memory board 104a may include SLC NAND flash memory chips and the memory board 104b may include MLC NAND flash memory chips. In another example, the memory board 104a may include flash memory chips from one flash memory chip manufacturer and the memory board 104b may include flash memory chips from a different flash memory chip manufacturer. The flexibility to have all the same type of flash memory chips or to have different types of flash memory chips enables the data storage device 100 to be tailored to different applications being used by the host 106.

[0036] In another exemplary implementation, the memory boards 104a and 104b may include different types of flash memory chips on the same memory board. For example, the memory board 104a may include both SLC NAND chips and MLC NAND chips on the same PCB. Similarly, the memory board 104b may include both SLC NAND chips and MLC NAND chips. In this manner, the data storage device 100 may be advantageously tailored to meet the specifications of the host 106.

[0037] In another exemplary implementation, the memory board 104a and 104b may include other types of memory devices, including non-flash memory chips. For instance, the memory boards 104a and 104b may include random access memory (RAM) such as, for instance, dynamic RAM (DRAM) and static RAM (SRAM) as well as other types of RAM and other types of memory devices. In one exemplary implementation, the both of the memory boards 104a and 104 may include RAM. In another exemplary implementation, one of the memory boards may include RAM and the other memory board may include flash memory chips. Also, one of the memory boards may include both RAM and flash memory chips.

[0038] The memory modules 120a and 120b on the memory boards 104a and

104b may be used to store information related to the flash memory chips 118a and 118b, respectively. In one exemplary implementation, the memory modules 120a and 120b may store device characteristics of the flash memory chips. The device characteristics may include whether the chips are SLC chips or MLC chips, whether the chips are NAND or NOR chips, a number of chip selects, a number of blocks, a number of pages per block, a number of bytes per page and a speed of the chips.

[0039] In one exemplary implementation, the memory modules 120a and 120b may include serial EEPROMs. The EEPROMs may store the device characteristics. The device characteristics may be compiled once for any given type of flash memory chip and the appropriate EEPROM image may be generated with the device characteristics. When the memory boards 104a and 104b are operably connected to the controller board 102, then the device characteristics may be read from the EEPROMs such that the controller 110 may automatically recognize the types of flash memory chips 118a and 118b that the controller 110 is controlling. Additionally, the device characteristics may be used to configure the controller 110 to the appropriate parameters for the specific type or types of flash memory chips 118a and 118b.

[0040] As discussed above, the controller 110 may include a FPGA controller. Referring to FIG. 2, an exemplary block diagram of a FPGA controller 210 is illustrated. The FPGA controller may be configured to operate in the manner described above with respect to controller 110 of FIG. 1. The FPGA controller 210 may include multiple channel controllers 250 to connect the multiple channels 112 to the flash memory chips 218. The flash memory chips 218 are illustrated as multiple flash memory chips that connect to each of the channel controllers 250. The flash memory chips 218 are representative of the flash memory chips 118a and 118b of FIG. 1, which are on the separate memory boards 104a and 104b of FIG. 1. The separate memory boards are not shown in the example of FIG. 2. The FPGA controller 210 may include a PCIe interface module 208, a bi-directional direct memory access (DMA) controller 252, a dynamic random access memory (DRAM) controller 254, a command processor/queue 256 and an information and configuration interface module 258.

[0041] Information may be communicated with a host (e.g., host 106 of FIG. 1) using an interface. In this example, FIG. 2, the FPGA controller 210 includes a PCIe interface to communicate with the host and a PCIe interface module 208. The

PCIe interface module 208 may be arranged and configured to receive commands from the host and to send commands to the host. The PCIe interface module 208 may provide data flow control between the host and the data storage device. The PCIe interface module 208 may enable high speed transfers of data between the host and the controller 210 and ultimately the flash memory chips 218. In one exemplary implementation, the PCIe interface and the PCIe interface module 208 may include a 64-bit bus. The bi-directional direct memory access (DMA) controller 252 may be arranged and configured to control the operation of the bus between the PCIe interface module 208 and the command processor/queue 256.

[0042] The bi-directional DMA controller 252 may be configured to interface with the PCIe interface 208, and each of the channel controllers 250. The bi-directional DMA controller 252 enables bi-directional direct memory access between the host 106 and the flash memory chips 218.

[0043] The DRAM controller 254 may be arranged and configured to control the translation of logical to physical addresses. For example, in an implementation in which the host addresses the memory space using logical addresses, the DRAM controller 254 may assist the command processor/queue 256 with the translation of the logical addresses used by the host to the actual physical addresses in the flash memory chips 218 related to data being written to or read from the flash memory chips 218. A logical address received from the host may be translated to a physical address for a location in one of the flash memory chips 218. Similarly, a physical address for a location in one of the flash memory chips 218 may be translated to a logical address and communicated to the host.

[0044] The command processor/queue 256 may be arranged and configured to receive the commands from the host through the PCIe interface module 208 and to control the execution of the commands through the channel controllers 250. The command processor/queue 256 may maintain a queue for a number of commands to be executed and order the commands using an ordered list to ensure that the oldest commands may be processed first. The command processor 100 may maintain the order of the commands designated for the same flash memory chip and may reorder the commands designated for different flash memory chips. In this manner, multiple commands may be executed simultaneously and each of the channels 112 may be used simultaneously or at least substantially simultaneously.

[0045] The command processor/queue 256 may be configured to process commands for different channels 112 out of order and preserve per-channel command ordering. For instance, commands that are received from the host and that are designated for different channels may be processed out of order by the command processor/queue 256. In this manner, the channels may be kept busy. Commands that are received from the host for processing on the same channel may be processed in the order that the commands were received from the host by the command processor/queue 256. In one exemplary implementation, the command processor/queue 256 may be configured to maintain a list of commands received from the host in an oldest-first sorted list to ensure timely execution of the commands.

[0046] The channel controllers 250 may be arranged and configured to process commands from the command processor/queue 256. Each of the channel controllers 250 may be configured to process commands for multiple flash memory chips 218. In one exemplary implementation, each of the channel controllers 250 may be configured to process commands for up to and including 32 flash memory chips 218.

[0047] The channel controllers 250 may be configured to process the commands from the command processor/queue 256 in order as designated by the command processor/queue 256. Examples of the commands that may be processed include, but are not limited to, reading a flash page, programming a flash page, copying a flash page, erasing a flash block, reading a flash block's metadata, mapping a flash memory chip's bad blocks, and resetting a flash memory chip.

[0048] The information and configuration interface module 258 may be arranged and configured to interface with a memory module (e.g., memory module 116 of FIG. 1) to receive configuration information for the FPGA controller 210. For example, the information and configuration interface module 258 may receive one or more images from the memory module to provide firmware to the FPGA controller 210. Modifications to the images and to the firmware may be provided by the host to the controller 210 through the information and configuration interface module 258. Modifications received through the information and configuration interface module 258 may be applied to any of the components of the controller 210 including, for example, the PCIe interface module 208, the bi-directional direct memory access (DMA) controller 252, the DRAM controller 254, the command processor/queue 256 and the channel controllers 250. The information and configuration interface module 258 may include one or more registers, which may be modified as necessary by instructions from the host.

[0049] The FPGA controller 210 may be arranged and configured to cooperate and process commands in conjunction with the host. The FPGA controller 210 may perform or at least assist in performing error correction, bad block management, logical to physical mapping, garbage collection, wear levelling, partitioning and low level formatting related to the flash memory chips 218.

[0050] FIG. 3A is a schematic block diagram of an apparatus 300 including a data storage device 302 having a plurality of flash memory chips 318a, 318b, 318c, 318d, 318e, 318f, 318g, 318h, 318i, 318j, 318k, 318l that are organized into a first partition 321 and a second partition 322. The first and second partition 321 and 322 define different physical areas of storage space in the data storage device 302, such that directories and files of different categories can be stored in the different partitions, or so that one partition can be used for different purposes than the other partition. The first partition can include a first subset of the flash memory chips 318a-f, while the second partition can include a second subset of the flash memory chips 318g-l, where there are not any flash memory chips that are part of both partitions. That is, the boundary between the partitions 321 and 322 is drawn between individual flash memory chips to ensure that an individual flash memory chip does not belong to more than one partition.

[0051] Organizing the data storage device into two or more partitions can serve a number of purposes. For example, operating system file stored on one partition can be kept separate from user files stored on another partition. Cache and log files that can change size dynamically and rapidly, potentially making a file system full, can be stored on one partition and kept separate from other files stored on a different partition. Partitions can be used for multi-booting setups, which allow users to have more than one operating system on a single computer. For example, a user could install Linux, Mac OS X, and Microsoft Windows or operating systems on different partitions of the same data storage device and have a choice of booting into any operating system (supported by the hardware) at power-up. Partitions can be used to protect or isolate files to make it easier to recover a corrupted file system or operating system installation. For example if one partition is corrupted but none of the other file systems are affected, the data on the storage device may still be salvageable. Using a separate partition for read-only data also reduces the chances of the file system on that partition becoming corrupted. Partitions also can raise overall computer performance on systems where smaller file systems are more efficient. For example, large hard drives with only one NTFS file system typically have a very large sequentially-accessed Master File Table (MFT), and it generally takes more time to read this MFT than the smaller MFTs of smaller partitions.

[0052] In another example embodiment, the data storage device 302 may be used to store large amounts of data (e.g., many Gigabytes or Terabytes of data) that must be read quickly from the data storage device and supplied to the host. For example, the data storage device can be used to cache large volumes of publicly accessible information (e.g., a large corpus of web pages from the World Wide Web, a large library of electronic versions of books, or digital information representing a large volume of telecommunications, etc.) that can be fetched by the host in response to a query. Thus, it can be important that the relevant data be accessed and returned very quickly in response to a read command issued by the host. However, the information stored in the data storage device also may need to be constantly updated to keep the information up to date as the relevant information changes. For example, if the information on the storage device relates to a corpus of web pages, the information stored on the storage device may need to be updated as the web pages change and as new web pages are created.

[0053] In such a system, a partitioned flash memory data storage device 302 can offer exceptional performance. In a flash memory storage device, write operations to a flash memory chip take much longer (e.g., 10 - 100 times longer) than read operations from a flash memory chip. Therefore, organizing the chips 318a-l of the data storage device into two or more partitions, where the partitions are defined at boundaries between different chips, offers a way to ensure fast read operations while also allowing the information stored on the data storage device to be updated in real time. For example, both partitions 321 and 322 can be used to store a corpus of data (e.g., a corpus of web pages) to be served in response to queries and the individual partitions can alternate between serving the requests and being updated with new information. For instance, in a first time period the first partition 321 can be used to provide the information to the host (e.g., information that may be requested in response to a user query), while the data on the second partition 322 is updated (e.g., in response to changes or additions to the web pages of the corpus). Then, in a second time period, the recently updated second partition 322 can be used to provide the information to the host, while the data on the first partition 321 is updated. This process can be repeated so that data is always served from a partition that acts as a read-only device, and therefore provides very fast responses to read commands from the host without being slowed down by write commands, while the other partition is being updated with new information. Defining the partitions such that an individual flash memory chip is included in only one partition ensures that no flash chip will have data written to it and read from it at substantially the same time, which would cause a delay is responding to a read request from the host 350.

[0054] As discussed above, the memory chips 318a-l can be connected to a controller that may include a FPGA controller 310. The FPGA controller may be configured to operate in the manner described above with respect to controller 110 of FIG. 1 or of FPGA 210 of FIG. 2. The FPGA controller 310 may include multiple channel controllers 312a, 312b, 312c, 312d, 312e, 312f to connect the multiple channels 112 to the flash memory chips 318a-l. Of course, as described above, the storage device can include more than 12 flash memory chips, more than six channel controllers, and many more than two flash memory chips may be operably connected to a channel controller across a physical channel. Thus, the implementation shown in FIGS. 3A and 3B is merely schematic for clarity of illustration.

[0055] In one implementation, channel controllers 312a, 312b, 312c, 312d, 312e, 312f can control channels that are operably connected to flash memory chips that are part of each partition 321 and 322. For example, channel controller 312a can be operably connected to memory chip 318a, which is part of the first partition 321, and also to memory chip 318g, which is part of the second partition 322. In such a configuration, at least one memory chip in the first partition 321 is connected to each communication channel between the data storage device 302 and the host, and at least one memory chip in the second partition 322 is connected to each communication channel between the data storage device 302 and the host 350. Such a configuration results in maximum parallelism of communication between a partition 321 or 322 and the host, which can result in fast read access and fast write times from and to the data storage device 302.

[0056] In another implementation, approximately half the channel controllers can be operably connected to flash memory chips in a first partition and approximately half the channel controllers can be operably connected to flash memory chips in the second partition.

[0057] In another implementation, shown in FIG 3B, flash memory chips 318a, 318b, 318c, 318d, 318e, 318f, 318g, 318h, 318i, 318j, 318k, 318l can be organized into a first partition 331, a second partition 332, a third partition 333, and a fourth partition 334, where the different partitions define different physical areas of storage space in the data storage device 302, such that directories and files of different categories can be stored in the different partitions, or so that one partition can be used for different purposes than the other partition. The first partition 331 can include a first subset of the flash memory chips 318a-c. The second partition 332 can include a second subset of the flash memory chips 318d-f. The third partition 333 can include a third subset of the flash memory chips 318g-i. The fourth partition 334 can include a fourth subset of the flash memory chips 318j-l. Among the different partitions 331, 332, 333, and 334 there are not any individual flash memory chips whose physical memory address space is part of two or more partitions. That is, the boundaries between the partitions 331, 332, 333, and 334 are drawn between individual flash memory chips to ensure that an individual flash memory chip does not belong to more than one partition.

[0058] In the system of FIG. 3B, a partitioned flash memory data storage device 302 can offer exceptional performance, e.g., when used to store a corpus of data (e.g., a corpus of web pages) to be served in response to queries, and the individual partitions can alternate between serving the requests and being updated with new information. For instance, in a first time period the first, second, and third partitions 331, 332, and 333 can be used to provide the information to the host (e.g., information that may be requested in response to a user query), while the data on the fourth partition 334 is updated (e.g., in response to changes or additions to the web pages of the corpus). Then, in a second time period, the recently updated fourth partition 334, along with the second and third partitions 332 and 332 can be used to provide the information to the host, while the data on the first partition 331 is updated. Thus, data on each partition can be updated in round robin fashion, while query requests are served by the other partitions. This process can be repeated so that data is always served from partitions that act as read-only devices, and therefore provides very fast responses to read commands from the host without being slowed down by write commands, while the other partition is being updated with new information. Defining four partitions results in redundancy of information stored on the data storage device, so that if a partition, channel, or individual memory chip fails, such that one partition is no longer usable, the remaining three partitions can continue to be used to provide a data storage device in which each of the remaining partitions takes turns being updated while the other remaining partitions serve data requests.

[0059] As described above, the data storage device 302 can be connected to a host 350 though an interface 308, which can be a high speed interface, such as, for example a PCIe interface. The host can include, for example, a processor 352, a first memory 354, a second memory 356, and a partition engine 360. The first memory 354 can include, for example, a non-volatile memory device (e.g., a hard disk) adapted for storing machine-readable, executable code instructions that can be executed by the processor 352. The code instructions stored on the first memory 354 can be loaded into the second memory (e.g., a volatile memory, such as, a random access memory) 356 where they can be executed by the processor 352 to create the memory device detection engine 358 and the partition engine 360. The second memory can include logical blocks of "user space" devoted to user mode applications and logical blocks of "kernel space" 364 devoted to running the lower-level resources that user-level applications must control to perform their functions. The memory device detection engine 358 and the partition engine 360 can reside in the kernel space 364 of the second memory 356.

[0060] The configuration detection engine 358 can be configured to detect the number of flash memory chips 318 on the data storage device 302, and the partition engine 360 can be configured to define the first partition 321 and the second partition 322 of the data storage device. Thus, the configuration detection engine 358 and the partition engine 360, which run on the host 350, can be used by the host to discover hardware device properties of the data storage device 302 and then to define, via the host, the partitions 321 and 322. In one implementation, the configuration detection engine 358 can issue a query command to the data storage device, and in response to the query command the data storage device can return information to the host about, for example, the number of flash memory chips 318, the size (e.g., as measured in bytes) of each chip, the number of channels in the data storage device, the flash memory chips to which each the channel controller 312a-e is operably connected. Such information can be stored on the EEPROM 116 on the FPGA 310 and/or on the EEPROM 120a of the flash board of the data storage device 302. The configuration detection engine can poll the EEPROM 116 or the EEPROM 120a (e.g., during a boot-up operation of the host 350) to cause the data storage device to return such information to the host 350. In another implementation, the host may poll the flash memory chips 318 to provide the information about, for example, the number of flash memory chips 318, the size (e.g., as measured in bytes) of each chip, the number of channels in the data storage device, the flash memory chips to which each the channel controller 312a-e is operably connected.

[0061] The partition engine 360 can receive the information from the memory device detection engine 358 about the number of flash chips 318, the size of each flash chip, the number of channels and the memory chips to which each channels is operably connected, and, based on this information, the partition engine can define a first partition 321 and second partition 322 in the data storage device 302 The partition engine running on the host 350 can define the first partition to include memory blocks drawn from a first subset of the memory chips 318 and the second partition memory blocks drawn from a second subset of the memory chips 318, where the first subset does not include any individual flash chips of the second subset and the second subset does not include any individual flash chips of the first subset. The partition engine 360 then can map the physical memory block addresses (which may include, for example, a unique channel number, a unique flash memory chip number, and a block address within the flash memory chip) to logical addresses that can be used by application programs running the in the user space, such that the user space applications running on the host 350 can read data from the data storage device 302 and write data to the data storage device 302 with reference to the logical space addresses.

[0062] After a partition scheme of multiple partitions has been defined and data has been stored on the flash memory chips of the data storage device 100, the device can store information about the partitioning scheme, e.g., on the memory 116, so that the when the device is booted at a later time, it can communicate the partitioning scheme to the host 106 for the host to use. For example, the device may maintain information about the physical configuration of the data storage device, including a number of flash memory chips in the device and about the partitioning scheme, including which flash memory storage chips and channels are associated with which partitions on the memory 116. Then, when the system including the host 106 and the data storage device 100 is booted, the storage device 100 can communicate this information to the host 106, e.g., in response to a read operation performed by the configuration detection engine 358 of the host 106. The partitioning engine 360 of the host 106 then can define the partitions for the operating system and applications running on the host. For example, the partitioning engine 360 can define a first and second partition based on the information read from the storage device 100, where the first and second partitions do not include any of the same memory chips. The partitioning engine 360 also can allocate a logical to physical memory map for the first and second partitions, so that they user-level application programs can use logical addresses that then are mapped to physical memory addresses of the flash memory chips of the storage device 100.

[0063] The partition engine 360 also can be used to re-define the first partition of the data storage device to include a third subset of the plurality of flash memory chips, where the third subset is different from the first subset, and where the third subset does not include any flash memory chips of the second subset and wherein the second subset does not include any flash memory chips of the third subset. For example, with reference to FIG. 3A and FIG. 3B, a user may decide that the original partition scheme shown in FIG. 3A does not suit his or her needs, and therefore may use the host to redefine the partitions 321 and 322 (e.g., to include more or fewer flash memory chips in the particular partitions) or to add additional partitions to the scheme. In one implementation, the first partition 321 can be redefined as partitions 331 and 333. Allowing the user to define the partitions through the host rather that forcing the user to accept a partition scheme that is pre-defined by, or pre-loaded in, the controller 310 gives the user flexibility to define partitions as he or she desires and to change the partition scheme when the need arises. In another implementation, the imminent failure of one of the flash memory chips, e.g., 318a, may be detected by the host, and in response to this information, the partition engine may re-define the first partition 321 to exclude the flash memory chip 318a from the partition, i.e., as the originally defined first partition but for the memory chip 318a. Thus, any number of partitions can be defined (up to the number of flash memory chips 118a and 118b in the storage device 100), and different partitions within a partition scheme can include different numbers of flash memory chips and can include different amounts of memory space.

[0064] The host also may include an address assignment engine 366 that can exist in the kernel 364 and that can assign physical memory addresses to data to be written to the data storage device 302. For example, an application running in user space 362 may call for data to be written from the host 350 to the data storage device 302, and the user space application may specify that the data be written to a particular logical memory address. The address assignment engine 366 may translate logical addresses into physical addresses that can include, for example, a particular channel that the data should be written to, a particular flash memory chip operably connected to the specified channel to which the data should be written, and a particular physical block address of the specified memory chip to which the data should be written. In such an implementation, the translation of logical addresses to physical memory space addresses can be performed by the address assignment engine 366, such that the role of the DRAM controller 254 of the FPGA 210 is reduced or irrelevant.

[0065] FIG. 4 is an exemplary flowchart illustrating an example process 400 of partitioning the data storage device of FIG. 1, where the device includes a plurality of flash memory chips. The process 400 can include determining a number of flash memory chips in the data storage device (402). For example, the configuration detection engine can query the data storage device to gather information about the number of flash memory chips in the data storage device. A first partition of the data storage device can be defined, via a host coupled to the data storage device, where the first partition includes a first subset of the plurality of flash memory chips (404). A second partition of the data storage device can be defined, via the host, where the second partition includes a second subset of the plurality of flash memory chips (406). As a result of this process it is ensured that the first subset does not include any flash memory chips of the second subset and that the second subset does not include any flash memory chips of the first subset.

[0066] Optionally, the process 400 can include writing data to the first partition while reading data from the second partition (408). Determining the number flash memory chips in the data storage device can include transmitting information from the data storage device to the host indicating the number of flash memory chips in the data storage device (410). An address location in the data storage device to which to write data from the host can be defined in the host, where the address location specifies that the data be written to a specific one of the plurality of memory chips (412).

[0067] When the data storage device includes a plurality of physical channels for communication of data between the host and the plurality of flash memory chips, with each channel being operably connected to a different plurality of the memory chips, the process 400 can further include determining the number of physical channels (414), determining a first subset of the channels, where channels of the first subset of the channels are operably connected only to memory chips of the first subset of memory chips (416), determining a second subset of the channels, where channels of the second subset of the channels are operably connected only to memory chips of the second subset of memory chips (418), and defining, in the host, an address location in the data storage device to which to write data from the host, wherein the address location specifies that the data be written to a specific one of the plurality of memory chips through a specific channel (420). In addition, the process 400 can include re-defining, via the host coupled to the data storage device, the first partition of the data storage device to include a third subset of the plurality of flash memory chips (422).

[0068] FIG. 5 is another exemplary flowchart illustrating an example process 500 of partitioning a data storage device that includes a plurality of flash memory chips,, The process 500 can include reading a physical configuration and of the data storage device, including the number flash memory chips in the data storage device (502). The process can also include reading a partitioning scheme for the data storage device (504). For example, the configuration detection engine can read the physical configuration and the partitioning scheme from the memory 116 of the data storage device. A host coupled to the data storage device can define a first partition of the data storage device, where the first partition includes a first subset of the plurality of flash memory chips (506), and a logical to physical memory map for the first partition can be allocated (508). The host can define a second partition of the data storage device, where the second partition includes a second subset of the plurality of flash memory chips (510) and where the first subset does not include any flash memory chips of the second subset and wherein the second subset does not include any flash memory chips of the first subset, and a logical to physical memory map for the second partition can be allocated (512).

[0069] Implementations of the various techniques described herein may be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. Implementations may be implemented as a computer program product, i.e., a computer program tangibly embodied in an information carrier, e.g., in a machine-readable storage device, for execution by, or to control the operation of, data processing apparatus, e.g., a programmable processor, a computer, or multiple computers. A computer program, such as the computer program(s) described above, can be written in any form of programming language, including compiled or interpreted languages, and can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program can be deployed to be executed on one computer or on multiple computers at one site or distributed across multiple sites and interconnected by a communication network.

[0070] Method steps may be performed by one or more programmable processors executing a computer program to perform functions by operating on input data and generating output. Method steps also may be performed by, and an apparatus may be implemented as, special purpose logic circuitry, e.g., a FPGA or an ASIC (application-specific integrated circuit).

[0071] Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read-only memory or a random access memory or both. Elements of a computer may include at least one processor for executing instructions and one or more memory devices for storing instructions and data. Generally, a computer also may include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto-optical disks, or optical disks. Information carriers suitable for embodying computer program instructions and data include all forms of non-volatile memory, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks. The processor and the memory may be supplemented by, or incorporated in special purpose logic circuitry.

[0072] To provide for interaction with a user, implementations may be implemented on a computer having a display device, e.g., a cathode ray tube (CRT) or liquid crystal display (LCD) monitor, for displaying information to the user and a keyboard and a pointing device, e.g., a mouse or a trackball, by which the user can provide input to the computer. Other kinds of devices can be used to provide for interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback, e.g., visual feedback, auditory feedback, or tactile feedback; and input from the user can be received in any form, including acoustic, speech, or tactile input.

[0073] Implementations may be implemented in a computing system that includes a back-end component, e.g., as a data server, or that includes a middleware component, e.g., an application server, or that includes a front-end component, e.g., a client computer having a graphical user interface or a Web browser through which a user can interact with an implementation, or any combination of such back-end, middleware, or front-end components. Components may be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication networks include a local area network (LAN) and a wide area network (WAN), e.g., the Internet.

[0074] While certain features of the described implementations have been illustrated as described herein, many modifications, substitutions, changes and equivalents will now occur to those skilled in the art. It is, therefore, to be understood that the appended claims are intended to cover all such modifications and changes as fall within the scope of the embodiments.

[0075] Throughout this specification and the claims which follow, unless the context requires otherwise, the word "comprise", and variations such as "comprises" and "comprising", will be understood to imply the inclusion of a stated integer or step or group of integers or steps but not the exclusion of any other integer or step or group of integers or steps.

[0076] The reference to any prior art in this specification is not, and should not be taken as, an acknowledgement or any form of suggestion that the prior art forms part of the common general knowledge in Australia.

WHAT IS CLAIMED IS:

1. A method of partitioning a data storage device, wherein the device includes a plurality of memory chips, the method comprising:

determining a number of memory chips in the data storage device, wherein a first portion of the plurality of memory chips is implemented in a first memory channel and a second portion of the plurality of memory chips is implemented in a second memory channel;

defining, via a host coupled to the data storage device, a first partition of the data storage device, wherein the first partition includes a first subset of the plurality of memory chips; and

defining a second partition of the data storage device via the host coupled to the data storage device, wherein the second partition includes a second subset of the plurality of memory chips,

wherein:

the first subset includes one or more memory chips of the first memory channel and one or more memory chips of the second memory channel;

the first subset does not include any memory chips of the second subset;

the second subset does not include any memory chips of the first subset;

the first partition and the second partition implement different data storage areas in the data storage device; and

a boundary between the first partition and second partition is defined by the first subset of memory chips and the second subset of memory chips.

2. The method as in claim 1, further comprising writing data to the first partition while reading data from the second partition.

3. The method as in any of the preceding claims, wherein determining a number of memory chips in the data storage device includes transmitting information from the data storage device to the host indicating the number of memory chips in the data storage device.

4. The method as in any of the preceding claims, further comprising defining, in the host, an address location in the data storage device to which to write data from the host, wherein the address location specifies that the data be written to a specific one of the plurality of memory chips.

5. The method as in any of the preceding claims, wherein the data storage device includes a plurality of physical channels for communication of data between the host and the plurality of memory chips, each channel being operably connected to a different plurality of the memory chips, the method further comprising:

determining the number of physical channels;

determining a first subset of the channels, wherein channels of the first subset of the channels are operably connected only to memory chips of the first subset of memory chips;

determining a second subset of the channels, wherein channels of the second subset of the channels are operably connected only to memory chips of the second subset of memory chips; and

defining, in the host, an address location in the data storage device to which to write data from the host, wherein the address location specifies that the data be written to a specific one of the plurality of memory chips through a specific channel.

6. The method as in any of the preceding claims, wherein the first partition includes memory chips that are operably connected to a single channel.

7. The method as in any of the preceding claims, further comprising:

re-defining, via the host coupled to the data storage device, the first partition of the data storage device to include a third subset of the plurality of memory chips,

wherein the third subset is different from the first subset, and

wherein the third subset does not include any memory chips of the second subset and wherein the second subset does not include any memory chips of the third subset.

8. The method as in any of the preceding claims, further comprising:

receiving an indication that one of the memory chips in the first subset has failed or is approaching failure, wherein re-defining the first partition includes defining the third subset as the first subset of memory chips but for the memory chip that has failed or that is approaching failure.

9. An apparatus comprising:

a data storage device including a plurality of memory chips, wherein a first portion of the plurality of memory chips is implemented in a first memory channel and a second portion of the plurality of memory chips is implemented in a second memory channel;

a host operably coupled to the data storage device via an interface, the host comprising:

a configuration detection engine configured to detect the number of memory chips in the data storage device; and

a partition engine configured to define a first partition of the data storage device, wherein the first partition includes a first subset of the plurality of memory chips and to define a second partition of the data storage device, wherein the second partition includes a second subset of the plurality of memory chips,

wherein:

the first subset includes one or more memory chips of the first memory channel and one or more memory chips of the second memory channel;

the first subset does not include any memory chips of the second subset;

the second subset does not include any memory chips of the first subset;

the first partition and the second partition implement different data storage areas in the data storage device; and

a boundary between the first partition and second partition is defined by the first subset of memory chips and the second subset of memory chips.

10. The apparatus as in claim 9, wherein the data storage device is configured to transmit, upon receiving a command from the host, information from

the data storage device to the host indicating the number of memory chips in the data storage device.

11. The apparatus as in any one of claims 9 to 10, wherein the host further comprises an address assignment engine configured to assign a memory address to data to be written to the data storage device, wherein the assigned memory address specifies that the data be written to a specific one of the plurality of memory chips.

12. The apparatus as in any one of claims 9 to 11, wherein the data storage device includes a plurality of physical channels for communication of data between the host and the plurality of memory chips, each channel being operably connected to a different plurality of the memory chips,

wherein the configuration detection engine is further configured to detect the number of channels in the data storage device,

wherein the partition engine is further configured to define a first subset of the channels, wherein channels of the first subset of the channels are operably connected only to memory chips of the first subset of memory chips, and

wherein the partition engine is further configured to define a second subset of the channels, wherein channels of the second subset of the channels are operably connected only to memory chips of the second subset of memory chips.

13. The apparatus as in any one of claims 9 to 12, wherein the host further comprises an address assignment engine configured to assign a memory address to data to be written to the data storage device, wherein the assigned memory address specifies that the data be written to a specific one of the plurality of memory chips through a specific channel.

14. The apparatus as in any one of claims 9 to 13, wherein the first partition includes memory chips that are operably connected to a single channel.

15. The apparatus as in any one of claims 9 to 14, wherein the partition engine is further configured to re-define the first partition of the data storage device to include a third subset of the plurality of memory chips,

wherein the third subset is different from the first subset, and wherein the third subset does not include any memory chips of the second subset and wherein the second subset does not include any memory chips of the third subset.

16. The apparatus as in claim 15, wherein the partition engine is further configured to receive an indication that one of the memory chips in the first subset has failed or is approaching failure; and

wherein re-defining the first partition to include the third subset of the plurality of memory chips includes defining the third subset as the first subset of memory chips but for the memory chip that has failed or that is approaching failure.

17. A method of partitioning a data storage device, wherein the device includes a plurality of memory chips, the method comprising:

reading a physical configuration of the data storage device, including the number of memory chips in the data storage device, wherein a first portion of the plurality of memory chips is implemented in a first memory channel and a second portion of the plurality of memory chips is implemented in a second memory channel;

reading a partitioning scheme for the data storage device;

defining, via a host coupled to the data storage device, a first partition of the data storage device, wherein the first partition includes a first subset of the plurality of memory chips;

allocating a logical to physical memory map for the first partition;

defining, via the host coupled to the data storage device, a second partition of the data storage device, wherein the second partition includes a second subset of the plurality of memory chips; and

allocating a logical to physical memory map for the first partition,

wherein:

the first subset includes one or more memory chips of the first memory channel and one or more memory chips of the second memory channel;

the first subset does not include any memory chips of the second subset;

the second subset does not include any memory chips of the first subset;

2010234647 01 Jul 2013

the first partition and the second partition implement different data storage areas in the data storage device; and

a boundary between the first partition and second partition is defined by the first subset of memory chips and the second subset of memory chips.

18. The method as in claim 17, further comprising writing data to the first partition while reading data from the second partition.

19. The method as in any one of claims 17 to 18, wherein the data storage device includes a plurality of physical channels for communication of data between the host and the plurality of memory chips , each channel being operably connected to a different plurality of the memory chips, wherein the physical configuration of the data storage device further includes the number of memory chips in the data storage device, the method further comprising:

determining a first subset of the channels, wherein channels of the first subset of the channels are operably connected only to memory chips of the first subset of memory chips;

determining a second subset of the channels, wherein channels of the second subset of the channels are operably connected only to memory chips of the second subset of memory chips; and

defining, in the host, an address location in the data storage device to which to write data from the host, wherein the address location specifies that the data be written to a specific one of the plurality of memory chips through a specific channel.

20. An apparatus comprising:

a data storage device including a plurality of memory chips, wherein a first portion of the plurality of memory chips is implemented in a first memory channel and a second portion of the plurality of memory chips is implemented in a second memory channel;

a host operably coupled to the data storage device via an interface, the host comprising:

a configuration detection engine configured to read a physical configuration of the data storage device, including the number of memory chips in the data storage device and configured to read a partitioning scheme

2010234647 01 Jul 2013

for the data storage device; and

a partition engine configured to define a first partition of the data storage device, wherein the first partition includes a first subset of the plurality of memory chips and to define a second partition the data storage device, wherein the second partition includes a second subset of the plurality of memory chips, and configured to allocate a logical to physical memory map for the first partition and to allocate a logical to physical memory map for the second partition,

wherein:

the first subset includes one or more memory chips of the first memory channel and one or more memory chips of the second memory channel;

the first subset does not include any memory chips of the second subset;

the second subset does not include any memory chips of the first subset;

the first partition and the second partition implement different data storage areas in the data storage device; and

a boundary between the first partition and second partition is defined by the first subset of memory chips and the second subset of memory chips.

21. The apparatus as in claim 20, wherein the data storage device is configured to transmit, upon receiving a command from the host, information from the data storage device to the host indicating the physical configuration of the data storage device.

22. The apparatus as in any of claims 20 to 21, wherein the data storage device includes a plurality of physical channels for communication of data between the host and the plurality of memory chips, each channel being operably connected to a different plurality of the memory chips,

wherein the physical configuration of the data storage device further includes the number channels in the data storage device,

wherein the partition engine is further configured to define a first subset of the channels, wherein channels of the first subset of the channels are operably connected only to memory chips of the first subset of memory chips, and

wherein the partition engine is further configured to define a second subset of the channels, wherein channels of the second subset of the channels are operably connected only to memory chips of the second subset of memory chips.

23. The apparatus as in any of claims 21 to 22, wherein the first partition includes memory chips that are operably connected to a single channel.

24. A method of partitioning a data storage device, wherein the device includes a plurality of memory chips substantially as herein described.

25. An apparatus substantially as herein described.

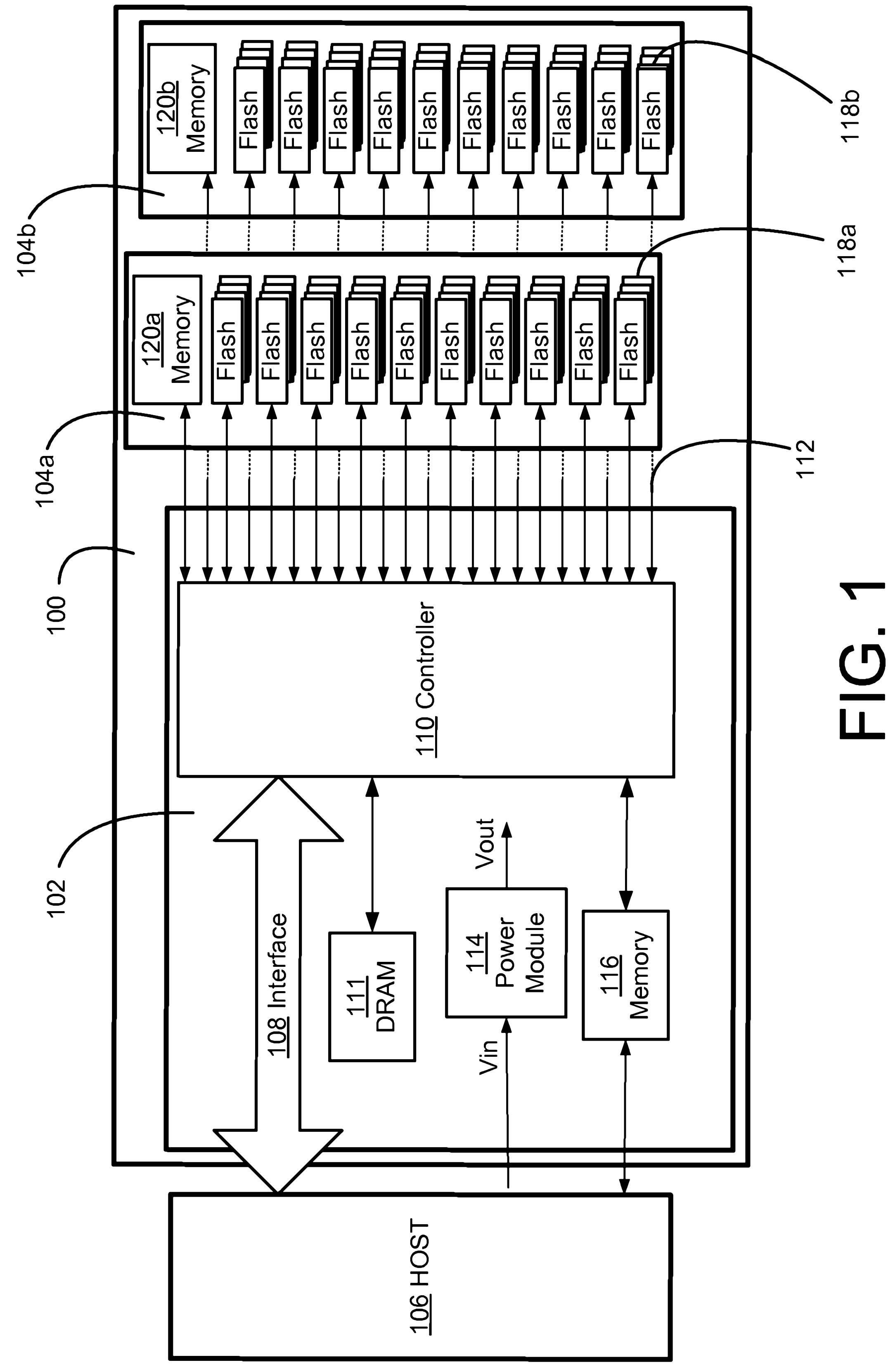
104b
120b Memory
Flash
Flash
Flash
Flash
Flash
Flash
Flash
Flash
Flash
Flash
118b
104a
120a Memory
Flash
Flash
Flash
Flash
Flash
Flash
Flash
Flash
Flash
Flash
118a
112
100
110 Controller
102
108 Interface
111 DRAM
Vout
114 Power Module
Vin
116 Memory
106 HOST

FIG. 1

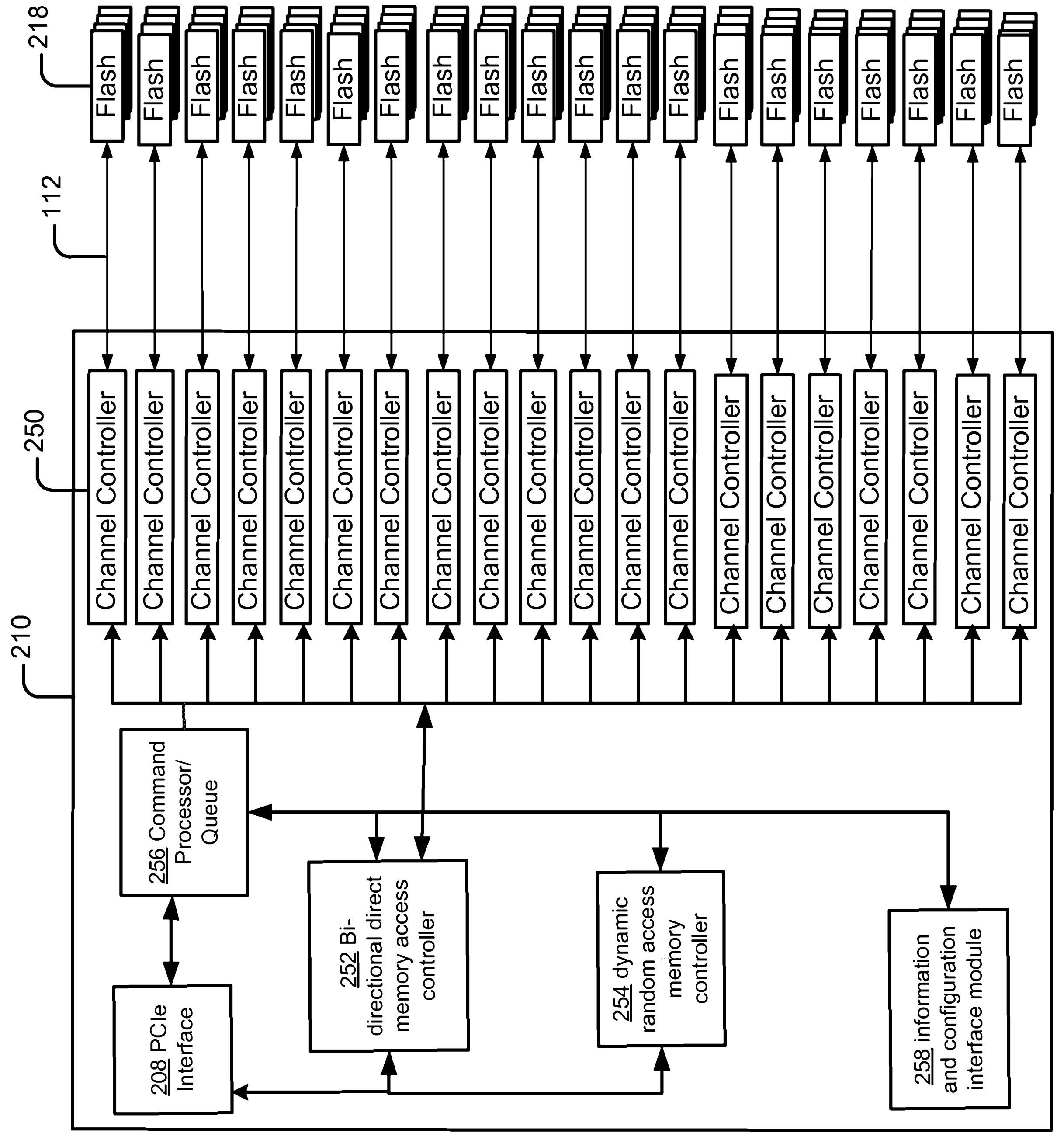
218
Flash
Flash
Flash
Flash
Flash
Flash
Flash
Flash
Flash
Flash
Flash
Flash
Flash
Flash
Flash
Flash
Flash
Flash
Flash
Flash
112
250
210
Channel Controller
Channel Controller
Channel Controller
Channel Controller
Channel Controller
Channel Controller
Channel Controller
Channel Controller
Channel Controller
Channel Controller
Channel Controller
Channel Controller
Channel Controller
Channel Controller
Channel Controller
Channel Controller
Channel Controller
Channel Controller
Channel Controller
Channel Controller
256 Command Processor/ Queue
252 Bi-directional direct memory access controller
254 dynamic random access memory controller
208 PCIe Interface
258 information and configuration interface module

FIG. 2

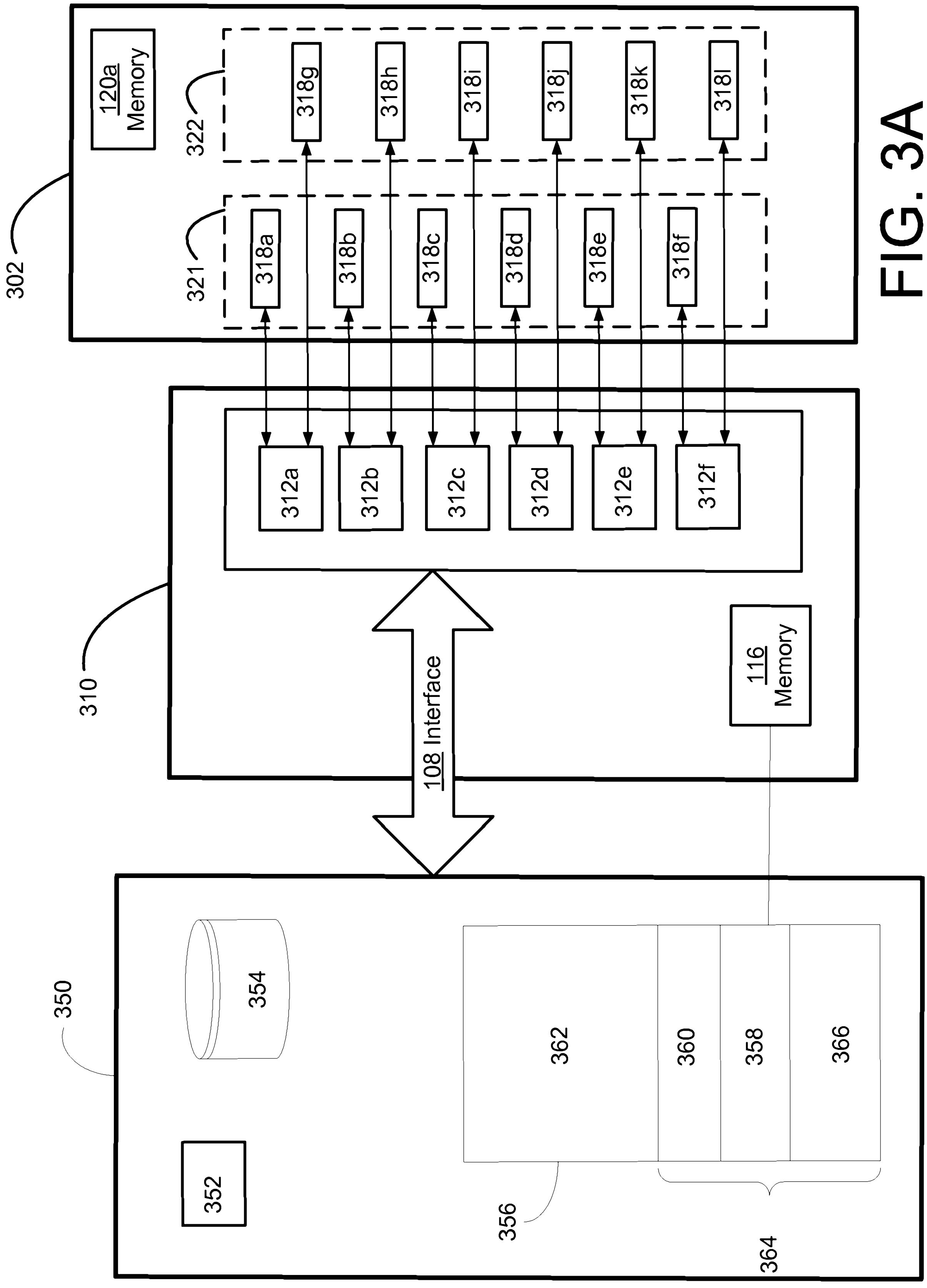
302
120a Memory
321
322
318a
318b
318c
318d
318e
318f
318g
318h
318i
318j
318k
318l
310
312a
312b
312c
312d
312e
312f
116 Memory
108 Interface
350
354
352
362
360
358
366
356
364

FIG. 3A

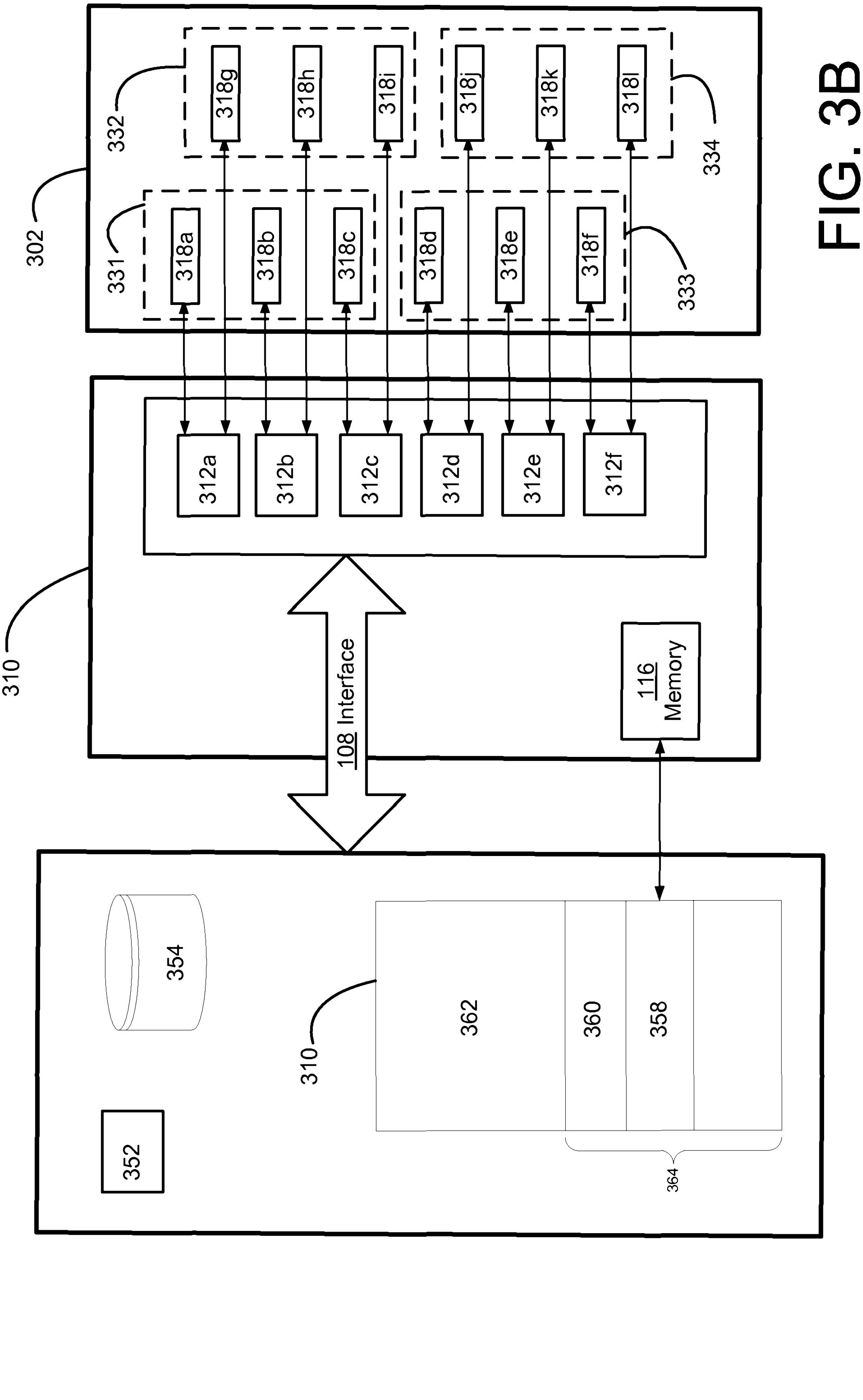
302
331
332
333
334
318a
318b
318c
318d
318e
318f
318g
318h
318i
318j
318k
318l
310
312a
312b
312c
312d
312e
312f
108 Interface
116 Memory
310
354
362
360
358
352
364

FIG. 3B

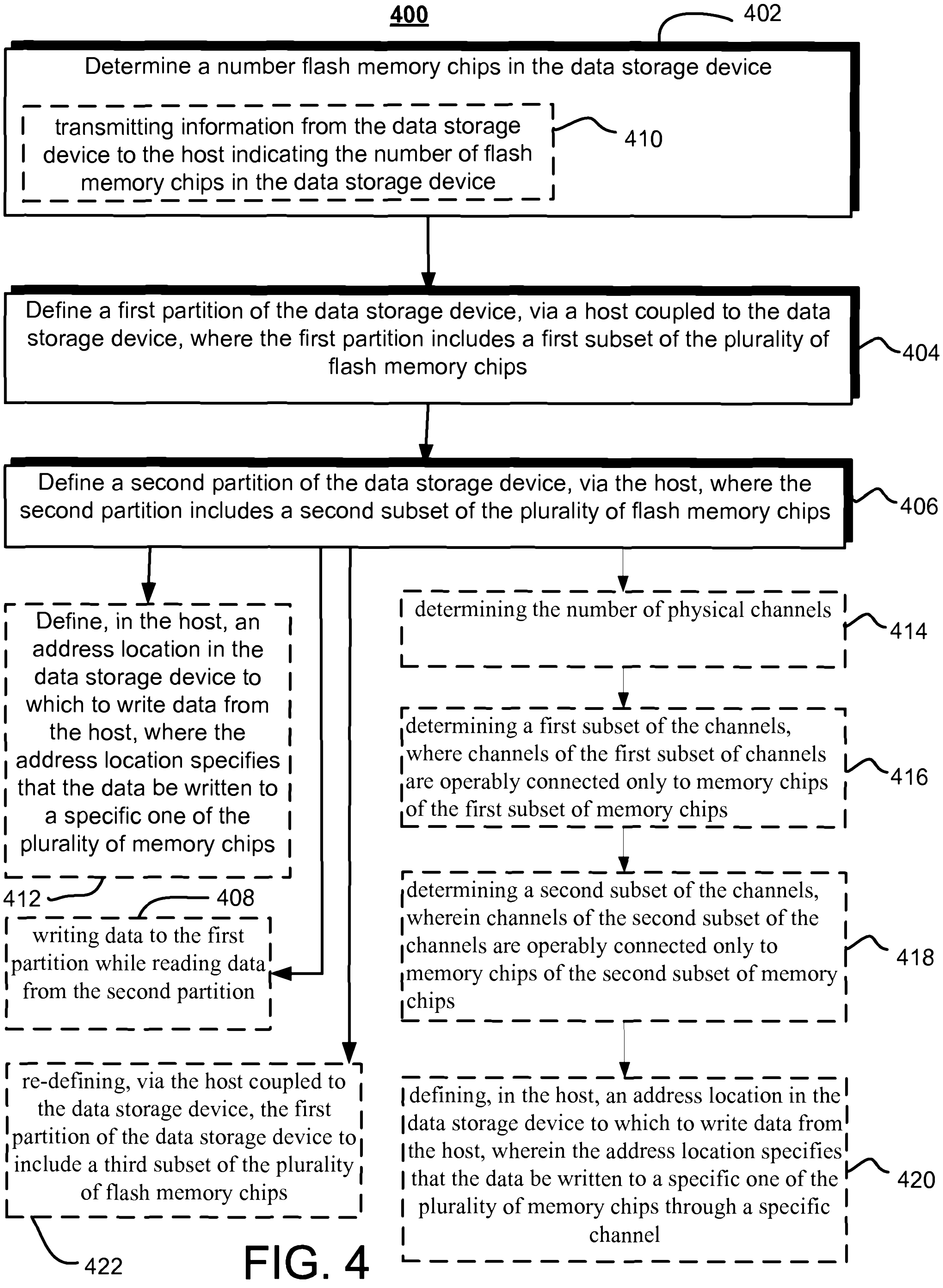
400
402
Determine a number flash memory chips in the data storage device
transmitting information from the data storage device to the host indicating the number of flash memory chips in the data storage device
410
Define a first partition of the data storage device, via a host coupled to the data storage device, where the first partition includes a first subset of the plurality of flash memory chips
404
Define a second partition of the data storage device, via the host, where the second partition includes a second subset of the plurality of flash memory chips
406
Define, in the host, an address location in the data storage device to which to write data from the host, where the address location specifies that the data be written to a specific one of the plurality of memory chips
412
408
writing data to the first partition while reading data from the second partition
re-defining, via the host coupled to the data storage device, the first partition of the data storage device to include a third subset of the plurality of flash memory chips
422
determining the number of physical channels
414
determining a first subset of the channels, where channels of the first subset of channels are operably connected only to memory chips of the first subset of memory chips
416
determining a second subset of the channels, wherein channels of the second subset of the channels are operably connected only to memory chips of the second subset of memory chips
418
defining, in the host, an address location in the data storage device to which to write data from the host, wherein the address location specifies that the data be written to a specific one of the plurality of memory chips through a specific channel
420

FIG. 4

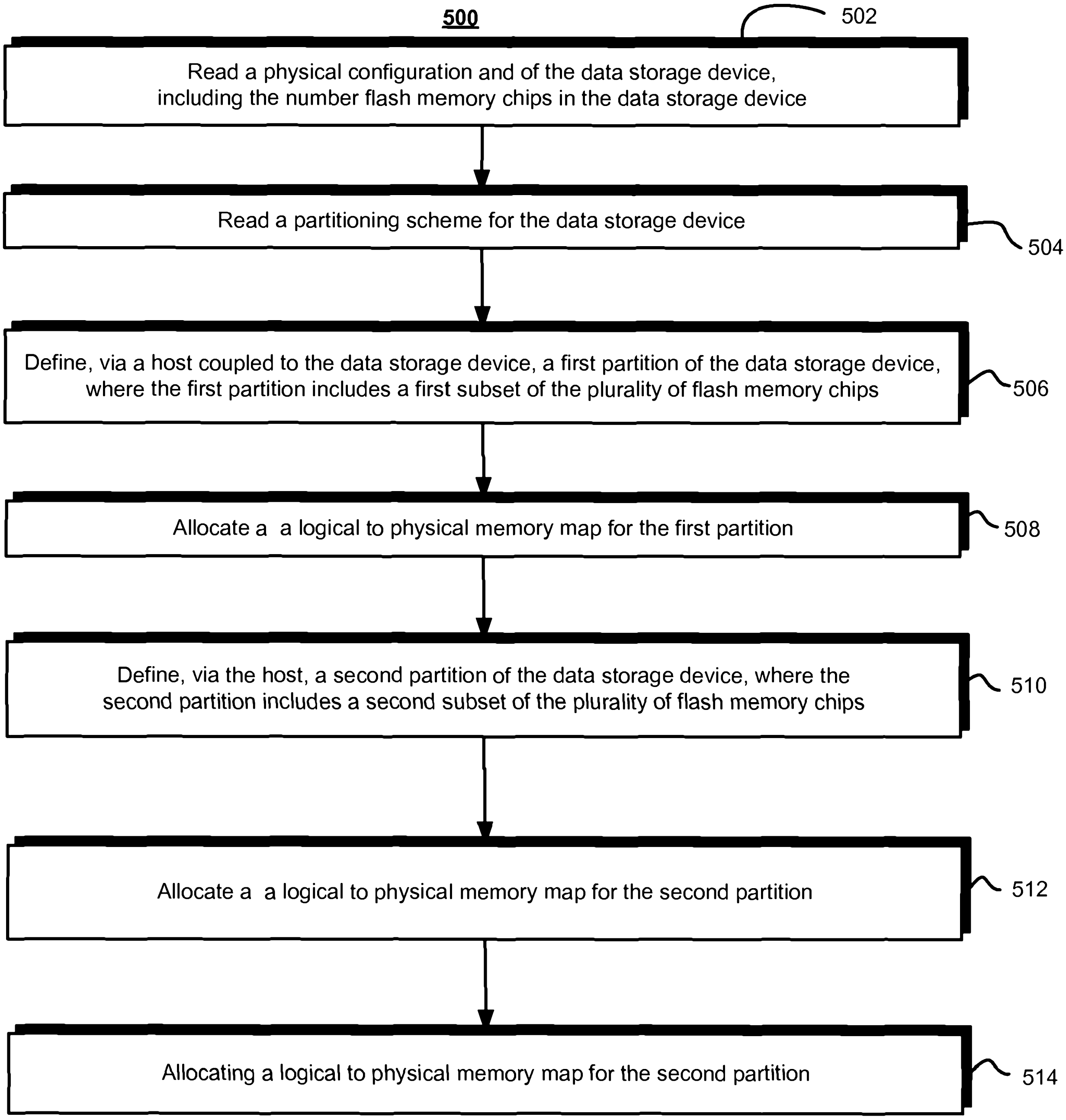
500
502
Read a physical configuration and of the data storage device, including the number flash memory chips in the data storage device
Read a partitioning scheme for the data storage device
504
Define, via a host coupled to the data storage device, a first partition of the data storage device, where the first partition includes a first subset of the plurality of flash memory chips
506
Allocate a a logical to physical memory map for the first partition
508
Define, via the host, a second partition of the data storage device, where the second partition includes a second subset of the plurality of flash memory chips
510
Allocate a a logical to physical memory map for the second partition
512
Allocating a logical to physical memory map for the second partition
514

FIG. 5