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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 August 2019 (29.08.2019)

WIPO | PCT

(10) International Publication Number
WO 2019/164573 A1

(51) International Patent Classification:
G06F 3/06 (2006.01) *G11C 16/26* (2006.01)
G11C 16/10 (2006.01)

(21) International Application Number:
PCT/US2018/066434

(22) International Filing Date:
19 December 2018 (19.12.2018)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
15/903,930 23 February 2018 (23.02.2018) US

(71) Applicant: **WESTERN DIGITAL TECHNOLOGIES, INC.** [US/US]; 5601 Great Oaks Parkway, San Jose, CA 95119 (US).

(72) Inventor: **YANG, Nian, Niles**; c/o Western Digital Technologies, Inc., 5601 Great Oaks Parkway, San Jose, CA 95119 (US).

(74) Agent: **HURLES, Steven, C.** et al.; Dickinson Wright PLLC, 2600 W. Big Beaver Road, Suite 300, Troy, MI 48084-3312 (US).

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

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

(54) Title: STORAGE SYSTEM AND METHOD FOR PERFORMING HIGH-SPEED READ AND WRITE OPERATIONS

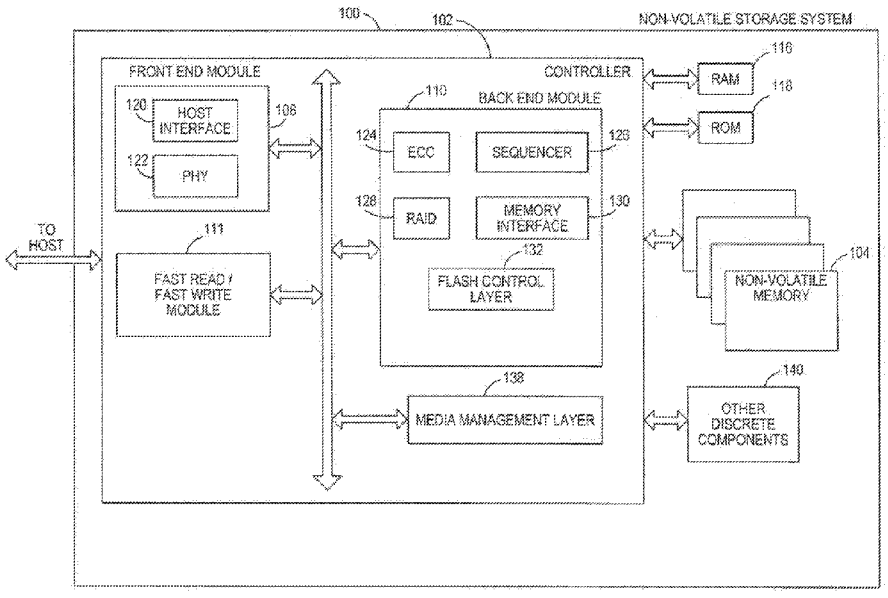

FIG. 2A

(57) Abstract: A storage system and method for performing high-speed read and write operations are disclosed. In general, these embodiments discuss ways for performing a fast read in response to determining that the fast read will probably not have a negative impact on performance due to error correction and performing a fast write in response to determining that a storage system criterion is satisfied.

WO 2019/164573 A1

[Continued on next page]

TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

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

STORAGE SYSTEM AND METHOD FOR PERFORMING HIGH-SPEED READ AND WRITE OPERATIONS

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of U.S. Utility Application Serial No. 15/903,930, filed February 23, 2018, which is incorporated herein by reference in its entirety.

BACKGROUND

[0002] It is sometimes desired to have faster-than-normal reads and writes in a storage system (e.g., in a solid state drive (SSD)). For example, as random reads take longer than sequential reads, it may be desired to increase the read speed of random reads. To increase read speed, the design of the storage system can be changed to provide more parallelism, but this can comes at the expense of increased volatile memory (e.g., RAM), power, and complexity. Also, for a low-queue-depth read, more parallelism may not be an effective design choice. As another option to improve read speed, memories can be screened for speed during manufacturing to "cherry pick" the fastest memories of a certain storage system. However, this may be at the expense of memory yield and wafer cost. As yet another option, the trim on the memory can be tuned to reduce the read margin. While this can provide a faster read, it can come at the cost of data integrity, as reducing the read margin (e.g., by reading the memory before a preferred read voltage is reached) can increase the bit error rate.

[0003] Increasing write speed may also be desired. For example, some storage systems use meta-block and jumbo block designs to operate on multiple memory dies via different channels in parallel to enhance the system's sequential write performance. Other storage systems use large volatile memory to cache data for programing and read the data in response to a host request. Some storage systems provide program speed improvement

via a special trim command to enhance memory cell behavior for high-program speed at the memory level.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] Figure 1A is a block diagram of a non-volatile storage system of an embodiment.

[0005] Figure 1B is a block diagram illustrating a storage module of an embodiment.

[0006] Figure 1C is a block diagram illustrating a hierarchical storage system of an embodiment.

[0007] Figure 2A is a block diagram illustrating components of the controller of the non-volatile storage system illustrated in Figure 1A according to an embodiment.

[0008] Figure 2B is a block diagram illustrating components of the non-volatile memory storage system illustrated in Figure IA according to an embodiment.

[0009] Figure 3 is a block diagram of a memory system of an embodiment.

[0010] Figure 4 is an illustration of bit-error-rate differences between a normal read and a fast read of an embodiment.

[0011] Figure 5 is a graph of LDPC throughput versus bit error rate of an embodiment.

[0012] Figure 6 is a flow chart of an embodiment for optimizing storage system performance with respect to memory behavior.

[0013] Figure 7 is flow chart of a method of an embodiment for high speed read.

[0014] Figure 8 is a diagram showing regular and high (fast) writes of an embodiment.

[0015] Figure 9 is a system data path view of an embodiment.

[0016] Figure 10 is flow chart of a method of an embodiment for high speed write.

DETAILED DESCRIPTION

[0017] Overview

[0018] By way of introduction, the below embodiments relate to a storage system and method for performing high-speed read and write operations.

[0019] In one embodiment, a method for reading a memory of a storage system is provided that is performed in a storage system comprising a memory configured to operate in a first read mode and a second read mode, wherein the first read mode is faster than the second read mode but can result in an increased bit error rate. The method comprises determining whether to operate the memory in the first read mode based on whether operating the memory in the first read mode will likely increase a bit error rate above a threshold; and in response to determining that operating the memory in the first read mode will not likely increase the bit error rate above the threshold, operating the memory in the first read mode.

[0020] In some embodiments, the method further comprises increasing a power supply to the memory when operating the memory in the first read mode.

[0021] In some embodiments, determining whether to operate the memory in the first read mode is performed in response to receiving a request for a random read.

[0022] In some embodiments, a likelihood that the bit error rate will increase above the threshold is based on a temperature of the memory.

[0023] In some embodiments, a likelihood that the bit error rate will increase above the threshold is based on a location of a word line to be read in the memory.

[0024] In some embodiments, the memory comprises a multi-level cell (MLC) memory, and wherein a likelihood that the bit error rate will increase above the threshold is based on a location of page to be read in the memory.

[0025] In some embodiments, a likelihood that the bit error rate will increase above the threshold is based on a power supply level.

[0026] In some embodiment , the storage system is embedded in a host.

[0027] In another embodiment, a storage system is provided comprising: a memory configured to operate in a first write mode and a second write mode, wherein the first write mode is faster than the second write mode; and a controller configured to: determine whether to instruct the memory to operate in the first write mode based on whether a storage system criterion is satisfied; and in response to determining that the storage system criterion is satisfied, instruct the memory to operate in the first write mode.

[0028] In some embodiments, the storage system criterion is satisfied when a program- erase count of the memory is under a threshold.

[0029] In some embodiments, the storage system comprises an error correction code encoder configured to operate at a first encoding speed, the memory comprises a plurality of memory dies, and the storage system criterion is satisfied when a memory die parallel operating speed is less than the first encoding speed.

[0030] In some embodiments, the memory comprises a plurality of memory dies, and wherein the storage system criterion is satisfied when a number of memory dies being operated in parallel is below a threshold.

[0031] In some embodiments, the storage system criterion is satisfied when the memory comprising single-level cells (SLCs).

[0032] In some embodiments, the second write mode performs a verify operation to check that data has been written correctly in the memory, whereas the first write mode does not.

[0033] In some embodiments, the memory comprises a three-dimensional memory.

[0034] In some embodiments, the storage system is embedded in a host.

[0035] In another embodiment, a storage system is provided comprising a memory; and at least one of the following: means for performing a fast read in response to determining that the fast read will probably not have a negative impact on performance due to error correction; and means for performing a fast write in response to a storage system criterion being satisfied.

[0036] In some embodiments, the memory comprises a three dimensional memory.

[0037] In some embodiments, the storage system is embedded in a host.

[0038] Other embodiments are possible, and each of the embodiments can be used alone or together in combination. Accordingly, various embodiments will now be described with reference to the attached drawings.

[0039] Embodiments

[0040] Storage systems suitable for use in implementing aspects of these embodiments are shown in Figures IA -1C. Figure IA is a block diagram illustrating a non-volatile storage system 100 according to an embodiment of the subject matter described herein. Referring to Figure IA, non-volatile storage system 100 includes a controller 102 and non-volatile memory that may be made up of one or more non-volatile memory die 104. As used herein, the term die refers to the collection of non-volatile memory cells, and associated circuitry for managing the physical operation of those non-volatile memory cells, that are formed on a single semiconductor substrate. Controller 102 interfaces with a host system and transmits command sequences for read, program, and erase operations to non-volatile memory die 104.

[0041] The controller 102 (which may be a flash memory controller) can take the form of processing circuitry, a microprocessor or processor, and a computer-readable medium that stores computer-readable program code (e.g., firmware) executable by the (micro)processor, logic gates, switches, an application specific integrated circuit (ASIC), a programmable logic controller, and an embedded microcontroller, for example. The

controller 102 can be configured with hardware and/or firmware to perform the various functions described below and shown in the flow diagrams. Also, some of the components shown as being internal to the controller can also be stored external to the controller, and other components can be used. Additionally, the phrase "operatively in communication with" could mean directly in communication with or indirectly (wired or wireless) in communication with through one or more components, which may or may not be shown or described herein.

[0042] As used herein, a flash memory controller is a device that manages data stored on flash memory and communicates with a host, such as a computer or electronic device. A flash memory controller can have various functionality in addition to the specific functionality described herein. For example, the flash memory controller can format the flash memory to ensure the memory is operating properly, map out bad flash memory cells, and allocate spare cells to be substituted for future failed cells. Some part of the spare cells can be used to hold firmware to operate the flash memory controller and implement other features. In operation, when a host needs to read data from or write data to the flash memory, it will communicate with the flash memory controller. If the host provides a logical address to which data is to be read/written, the flash memory controller can convert the logical address received from the host to a physical address in the flash memory. (Alternatively, the host can provide the physical address.) The flash memory controller can also perform various memory management functions, such as, but not limited to, wear leveling (distributing writes to avoid wearing out specific blocks of memory that would otherwise be repeatedly written to) and garbage collection (after a block is full, moving only the valid pages of data to a new block, so the full block can be erased and reused).

[0043] Non-volatile memory die 104 may include any suitable non-volatile storage medium, including NAND flash memory cells and/or NOR flash memory cells. The

memory cells can take the form of solid-state (e.g., flash) memory cells and can be one-time programmable, few-time programmable, or many-time programmable. The memory cells can also be single-level cells (SLC), multiple-level cells (MLC), triple-level cells (TLC), or use other memory cell level technologies, now known or later developed. Also, the memory cells can be fabricated in a two-dimensional or three-dimensional fashion.

[0044] The interface between controller 102 and non-volatile memory die 104 may be any suitable flash interface, such as Toggle Mode 200, 400, or 800. In one embodiment, storage system 100 may be a card based system, such as a secure digital (SD) or a micro secure digital (micro-SD) card. In an alternate embodiment, storage system 100 may be part of an embedded storage system.

[0045] Although, in the example illustrated in Figure lA, non-volatile storage system 100 (sometimes referred to herein as a storage module) includes a single channel between controller 102 and non-volatile memory die 104, the subject matter described herein is not limited to having a single memory channel. For example, in some NAND storage system architectures (such as the ones shown in Figures lB and lC), 2, 4, 8 or more NAND channels may exist between the controller and the NAND memory device, depending on controller capabilities. In any of the embodiments described herein, more than a single channel may exist between the controller and the memory die, even if a single channel is shown in the drawings.

[0046] Figure 1B illustrates a storage module 200 that includes plural non volatile storage systems 100. As such, storage module 200 may include a storage controller 202 that interfaces with a host and with storage system 204, which includes a plurality of non-volatile storage systems 100. The interface between storage controller 202 and non-volatile storage systems 100 may be a bus interface, such as a serial advanced technology attachment (SATA) or peripheral component interface express (PCie) interface. Storage

module 200, in one embodiment, may be a solid state drive (SSD), such as found in portable computing devices, such as laptop computers, and tablet computers.

[0047] Figure 1C is a block diagram illustrating a hierarchical storage system. A hierarchical storage system 250 includes a plurality of storage controllers 202, each of which controls a respective storage system 204. Host systems 252 may access memories within the storage system via a bus interface. In one embodiment, the bus interface may be an NVMe or fiber channel over Ethernet (FCoE) interface. In one embodiment, the system illustrated in Figure 1C may be a rack mountable mass storage system that is accessible by multiple host computers, such as would be found in a data center or other location where mass storage is needed.

[0048] Figure 2A is a block diagram illustrating components of controller 102 in more detail. Controller 102 includes a front end module 108 that interfaces with a host, a back end module 110 that interfaces with the one or more non-volatile memory die 104, and various other modules that perform functions which will now be described in detail. A module may take the form of a packaged functional hardware unit designed for use with other components, a portion of a program code (e.g., software or firmware) executable by a (micro)processor or processing circuitry that usually performs a particular function of related functions, or a self-contained hardware or software component that interfaces with a larger system, for example. Modules of the controller 102 may include an fast read/ fast write module 111, which is discussed in more detail below, and can be implemented in hardware or software/firmware.

[0049] Referring again to modules of the controller 102, a buffer manager/bus controller 114 manages buffers in random access memory (RAM) 116 and controls the internal bus arbitration of controller 102. A read only memory (ROM) 118 stores system boot code. Although illustrated in Figure 2A as located separately from the controller 102,

in other embodiments one or both of the RAM 116 and ROM 118 may be located within the controller. In yet other embodiments, portions of RAM and ROM may be located both within the controller 102 and outside the controller.

[0050] Front end module 108 includes a host interface 120 and a physical layer interface (PHY) 122 that provide the electrical interface with the host or next level storage controller. The choice of the type of host interface 120 can depend on the type of memory being used. Examples of host interfaces 120 include, but are not limited to, SATA, SATA Express, SAS, Fibre Channel, USB, PCie, and NVMe. The host interface 120 typically facilitates transfer for data, control signals, and timing signals.

[0051] Back end module 110 includes an error correction controller (ECC) engine 124 that encodes the data bytes received from the host, and decodes and error corrects the data bytes read from the non-volatile memory. A command sequencer 126 generates command sequences, such as program and erase command sequences, to be transmitted to non-volatile memory die 104. A RAID (Redundant Array of Independent Drives) module 128 manages generation of RAID parity and recovery of failed data. The RAID parity may be used as an additional level of integrity protection for the data being written into the memory device 104. In some cases, the RAID module 128 may be a part of the ECC engine 124. A memory interface 130 provides the command sequences to non-volatile memory die 104 and receives status information from non-volatile memory die 104. In one embodiment, memory interface 130 may be a double data rate (DDR) interface, such as a Toggle Mode 200,400, or 800 interface. A flash control layer 132 controls the overall operation of back end module 110.

[0052] The storage system 100 also includes other discrete components 140, such as external electrical interfaces, external RAM, resistors, capacitors, or other components that may interface with controller 102. In alternative embodiments, one or more of the physical

layer interface 122, RAID module 128, media management layer 138 and buffer management/bus controller 114 are optional components that are not necessary in the controller 102.

[0053] Figure 2B is a block diagram illustrating components of non-volatile memory die 104 in more detail. Non-volatile memory die 104 includes peripheral circuitry 141 and non-volatile memory array 142. Non-volatile memory array 142 includes the non-volatile memory cells used to store data. The non-volatile memory cells may be any suitable non-volatile memory cells, including NAND flash memory cells and/or NOR flash memory cells in a two dimensional and/or three dimensional configuration. Non- volatile memory die 104 further includes a data cache 156 that caches data. Peripheral circuitry 141 includes a state machine 152 that provides status information to the controller 102 and a temperature sensor 153 that can detect the temperature of the memory. Any suitable technology can be used to implement the temperature sensor 153, including technology currently employed in conventional memory temperature sensors. Also, it should be noted that while the temperature sensor 153 is located in the memory die 104 in this embodiment, the temperature sensor 153 can be located in another component in the storage system 100, such as the controller 102, or can be a separate component in the storage system 100.

[0054] Turning again the drawings, Figure 3 shows another view of the storage system 100 of one embodiment, which illustrates the storage system's power circuitry with respect to other components and its control signals and output interrupt signals. As shown in Figure 3, the storage system 100 comprises a memory 104 (here, a plurality of NAND packages) with one or more temperature sensors 153, a controller 102 (here, including an application-specific integrated circuit (ASIC), an analog "top" layer (ATOP), and RAM), and a power supply 300 (here, a power integrated circuit (IC), which may also be referred to as the power management integrated circuit (PMIC). The power supply 300 receives

power from a host, regulates the power, and supplies it to the NAND dies 104 and to the controller ASIC 102 (via internal and external interrupt requests (IRQs). The controller storage system 100 also comprises always-on logic 310 for low-voltage operations and firmware 302 to determine the fast read for the NAND power level. The firmware 302 can take the form of the fast read/fast write module 111 mentioned previously.

[0055] As mentioned above, it is sometimes desired to have faster-than-normal reads and writes in a storage system (e.g., in a solid state drive (SSD)). For example, as random reads take longer than sequential reads, it may be desired to increase the read speed of random reads. To increase read speed, the design of the storage system can be changed to provide more parallelism, but this can comes at the expense of increased volatile memory (e.g., RAM), power, and complexity. Also, for a low-queue-depth read, more parallelism may not be an effective design choice. As another option to improve read speed, memories can be screened for speed during manufacturing to "cherry pick" the fastest memories of a certain storage system. However, this may be at the expense of memory yield and wafer cost.

[0056] As yet another option, the trim on the memory can be tuned to reduce the read margm. In this situation, the memory can be operated in one of two read modes. In a "normal" read mode, internal circuitry in the memory die waits until the read voltage applied to a wordline in the memory reaches a predetermined voltage level before reading memory cells on the wordline, as the memory cells can be reliably read at the a predetermined voltage level. The read voltage is supplied by a voltage pump, and, due to the physics involved in generating voltage, it can take some time for the voltage pump to supply the predetermined voltage level. For example, if the predetermined voltage level is 3.0 V, the voltage pump may go through a series of overshoots (e.g., regulating up to 3.3

V) and undershoots (e.g., regulating down to 2.8 V) before settling on 3.0 V. In some storage system, this settling process can take several microseconds.

[0057] For a fast read, the controller can instruct the memory to operate differently (e.g., to tune the trim on the memory to reduce the read margin). In a fast read, the memory does not wait until read voltage reaches the predetermined level before reading memory cells on the wordline. For example, if the normal read mode needs to wait 20 microseconds to generate the read voltage and have it settle to the predetermined level, the fast read mode can be performed the read at 10 microseconds. Because the fast read mode does not wait for the read voltage to settle, the read voltage in a fast read may not be reliably at 3.0 V (e.g., it might be at 2.8 V or at 3.3 V). As a result, if the read voltage is not at the predetermined level, it is possible that a memory cell read during the fast read mode will not be read correctly (e.g., a "0" might be read as a "l," and a "1" might be read as a "0"). Accordingly, a faster read can come at the cost of data integrity, as reducing the read margin (e.g., by reading the memory before the preferred read voltage is reached) can increase the bit error rate (BER). This is illustrated in Figure 4, which is an illustration 400 of bit-error-rate differences between a normal read and a fast read of an embodiment. As shown in Figure 4, fast read, which be as much as 29% faster than a normal read mode in an embodiment, comes at the cost of a higher bit error rate. For example, a bit-error rate of 0.2 is more likely during a fast read than a normal read. To help reduce the bit error rate, the controller 102 can increase the power supply to the memory 104 when operating the memory 104 in the fast read mode. This is because the higher the power supply, the faster it is for the voltage pump to regulate the read voltage (e.g., increasing Vcc by the PMIC can shorten the settling time and get to predetermined read voltage more quickly).

[0058] As mentioned above, the storage system 100 comprises error correction functionality 124 (e.g., low-density parity-check code (LDPC)), which can correct a certain number of errors. This is illustrated in Figure 5, which is a graph 500 of LDPC throughput versus bit error rate. As shown in Figure 5, the performance of the storage system 100 is at an acceptable level when the bit-error rate is below a certain level (here, about 0.36%). That is, the system will perform poorly when the bit-error rate is higher than this level due to the slower LDPC throughput. However, after this level, the LDPC throughput starts to drop quickly, negatively impacting the performance of the storage system 100. Accordingly, in this example, if the fast read mode will likely result in a bit-error rate increase that is not greater than the 0.36% threshold, the fast read mode can come without sacrificing performance even though it increases the bit-error rate.

[0059] Further, there are certain system conditions that can make it more likely that the bit-error rate will increase more than the threshold. If these system conditions exist, it may be preferred not to perform a fast read, as they can increase the likelihood of the bit-error rate exceeding the performance threshold. For example, as the temperature of the memory 104 increases, the likelihood of errors also increases. As another example, in some memory designs, certain wordlines may be different than other wordlines and more prone to errors. For instance, in a three-dimensional memories, the first five wordlines, which are on the bottom of the physical stack, may be different from the other wordlines in the memory and have less read margin, resulting in a higher likelihood of biterrors.

As yet another example, in a multi-level cell (MLC) memory, the likelihood of a bit error can be based on the location of the page being read in the memory. For instance, in a triple-level cell (TLC) memory that has three pages, the middle page tends to have a higher bit-error rate based on memory physics. As yet another example, a bit error can be more likely

depending on the power level supplied to the memory 104, with bit errors less likely the greater the power level.

[0060] In summary, because of the error correction functionality in the storage system 100, an increase in bit error rate caused by a fast read does not necessarily negatively impact performance if the increased bit-error rate is below a certain threshold. And, as noted above, various system conditions can make it more or less likely that the increased bit-error rate will exceed the threshold. Putting these various factors together, in one embodiment, the controller 102 determines that it should instruct the memory 104 to operate in the fast read mode only if doing so will not likely increase a bit error rate above a threshold. In this way, the controller 102 intelligently determines when to enable the fast read in order to maximize system performance.

[0061] It should be noted that the terms "likely" or "likelihood" refer to the fact that the controller 102 may not know with certainty whether or not the bit error rate will go above the threshold based on the various conditions or variables present in the system 100. "Likelihood" can be determined in any suitable way. For example, in one embodiment, the storage system 100 contains a table or other data structure with values associated with various system conditions and variables and compares the total value of the system conditions and variables present at a given time to a number to determine "likelihood." As another example, which will be illustrated in more detail below, different thresholds are set for various system conditions and variables, and the controller 102 checks these thresholds in various combinations in making its determination.

[0062] This process is generally illustrated in the diagram 600 in Figure 6. As shown in Figure 6, the controller 102 checks various system conditions (act 610), such as, but not limited to, the read mode requested by the host (e.g., random or sequential), the memory temperature, the power supply level, and/or the type of memory 104 used in the system

100. The controller 102 can set the memory read operation mode according to these conditions (act 620). For example, if the controller 102 determines that it is not likely that the bit-error rate will exceed the threshold, the controller 102 can instruct the memory 104 to perform a fast read. On the other hand, if the controller 102 determines that it is likely that the bit-error rate will exceed the threshold, the controller 102 can let the memory 104 operate in a normal read mode. In some embodiments, the controller 102 only needs to instruct the memory 104 to enter fast read mode (e.g., to change the memory die's internal timing reduction to speed the sensing operation for a certain page to be read out), and no instruction is needed for the memory 104 to perform a normal read, as that is the memory's default. However, in other embodiments, the controller 102 instructs the memory 104 which read mode to operate in (e.g., fast or normal).

[0063] Figure 7 is a more-detailed flow chart 700 of an embodiment illustrating this method. In this embodiment, this method is performed in the controller 102 in response to receiving a request for a random read (e.g., a series of reads in non-consecutive memory locations), as fast reads may be more desirable in this situation than in a sequential read situation (e.g., a series ofreads in consecutive memory locations). As shown in Figure 7, in this example, the controller 102 first looks to see if the read is within the first five wordlines (act 710). As mentioned above, in some memory designs, the first five wordlines (or some other number N wordlines) are more susceptible to bit errors and, thus, may be undesirable for a fast read. If the read is to the first five wordlines, the controller 102 then checks to see if the read is to the middle of a page (act 720). As mentioned above, in TLC memories, a read to the middle of a page can be more error prone.

[0064] In this example, if the controller 102 determines that the read is to the first five word lines and is to the middle of a page, the controller 102 determines that it is likely that the bit-error rate will exceed the threshold and that a normal read should be used

instead of a fast read (act 730). If not, the controller 102 determines if the temperature of the memory 104 (e.g., as determined by the temperature sensors 153) is higher than a certain temperature threshold (75 degree C, in this example) (act 740). As mentioned above, an increased temperature can increase the likelihood of bit errors. If the memory temperature is above the temperature threshold, the controller 102 determines that it is likely that the bit-error rate will exceed the threshold, and a normal read should be used instead of a fast read (act 730). In contrast, if the memory temperature is below the temperature threshold, the controller 102 determines that it is not likely that the bit-error rate will exceed the threshold. In this case, the controller 102 instructs the memory 104 to operate in the fast read mode (act 760). Optionally, the controller 102 can increase power supply to the memory 104 (act 750), as the bit-error rate in fast read is dependent on the internal Vcc power supply noise, and temporarily increasing the PMIC Vcc supply to the targeted memory channel(s) / die(s) by a certain amount (e.g., 250mV~1000mV) can help reduce the bit error rate. When normal read mode is resumed, the PMIC can reduce the power supply to the nominal values for the non-fast read case on the memory die.

[0065] There are several advantages associated with these embodiments. In general, these embodiments provide faster system read with higher data integrity from a system perspective, resulting in a more-reliable system with overall higher-performance fast reads as compared to other storage systems. These embodiments can be used to provide a storage system design approach for achieving the fastest possible read speed with minimum loss of data integrity by identifying when and how to best use the fast read mode without elevating the bit error rate over a threshold. Further, these embodiments provide a system-side solution to a memory die problem. That is, fast read relies on the memory die's internal timing reduction to speed the sensing operation, leading to an intrinsically-higher bit-error rate for the read and, thus, error correction decoding time, which in tum diminishes the

performance of the random read. Using multiple variables or conditions at the system level to control how/when to use the memory's high speed (but noisier) read mode, the controller 102 can set the conditions for the system to take the advantages of the memory's high speed read operation without sacrificing the bit- error rate threshold beyond which the system performance will suffer. As noted above, these variables/conditions can include being aware that a higher power supply leads to less noisy memory operation (thus increasing the fast read integrity for less bit error rate), that certain physical locations and page types are sensitive to fast reads where higher BER is incurred compared to other location, and that fast read is sensitive to temperature. This awareness by the controller 102 provides a high-performance fast read operation that can meet fast random access time requirements for a high-performance storagesystem. That is, making system-level decisions on how to configure the system for optimum performance with respect to the memory behavior can help storage system vendors maximize system performance within the ultimate limits of the memory read operation and the memory's bit error rate.

[0066] There are several alternatives that can be used with these embodiments. For example, as noted above, in general, the higher the temperature, the more likely it is that there will be a bit-error due to increased noise. However, in some embodiments, if programming occurred at a high temperature, it may be preferred to read at that same high temperature. This is because, in certain situations, the bigger the temperature delta between write and read, the higher the bit-error rate. Accordingly, in an alternate embodiment, the storage system 100 stores the temperature value of a memory location when it is written. That way, the controller 102 can compare the temperature of the memory location at read with the stored temperature value and make the decision on whether or not to perform a fast read based on whether the temperature delta exceeds a threshold.

[0067] Other alternatives are possible. For example, instead of or in addition to the fast read embodiments discussed above, the storage system 100 can include a fast write embodiment. In some memory devices, the memory receives a command from the controller instructing the memory to program data into a location in the memory. After the memory programs the data, it reads the data from the location to verify that the data has been written correctly. This is shown in the top line in Figure 8. If the data was not written correctly, the memory can re-program the memory cells. If the data was written correctly, the memory can provide a response back to the controller that the program operation was successful.

[0068] In a fast write situation, the controller can instruct the memory (e.g., via a trim command) to skip the verify operation. This is shown in the bottom line in Figure 8. Skipping the verify operation results in a faster overall programming operation (in some embodiments, the verify operation takes about 30% of the program time). Fast writes may be preferred in many situations, such as in a high-capacity SSD system that requires very-high sequential-write performance with overall higher performance of the programming operation. However, skipping the verify step can result in a higher bit- error rate because errors that could have been caught during the verify operation are not. This embodiment addresses this by recognizing that certain conditions and variables in the storage system can determine the performance impact of a fast write. More specifically, in the following embodiments, the controller 102 (e.g., the fast read/ fast write module 111) can be configured to determine whether to instruct the memory 104 to operate in a fast write mode based on whether a storage system criterion is satisfied. As used herein, a "storage system criterion" (sometimes called a "system configuration") can refer to a structural or operating conditions of the storage system 100, such as, but not limited to, a memory type (e.g., single-level cell (SLC) memory), a program-erase count of a block of memory, where a

bottleneck is in the system 100, how many channels are used, type of write mode, memory temperature, and power supply level.

[0069] For example, the storage system criterion can be satisfied when the memory 104 comprises single-level cells (SLCs). Unlike multi-level cells, only a single programming pulse is needed to program an SLC cell. As a result, there is not a significant chance of the memory cell not being programmed correctly. Even if there is an error, it is likely that error correction functionality (e.g., LDPC) can handle the error. Further, in some storage systems, there would not be enough time during the allotted duration for a program operation to apply a second program pulse to correct the error. Accordingly, SLC memory is a good candidate for a fast write.

[0070] As another example, the storage system criterion can be satisfied when the program-erase count (PEC) of the memory 104 is under a threshold. This example recognizes that programming errors are less prone in "fresh" memory (e.g., at beginning of life (BOL)) than in memory that has endured a lot of program and erase cycles (e.g., at end of life (EOL)). Consider, for example, a situation in which the high-speed programming mode is at "M" MB/sat PEC < Threshold (e.g., M=32SMB/s) and the regular-speed mode is at "N" MB/sat the PEC > Threshold (e.g., N=2S0MB/s).

[0071] In high-speed programming mode, an SLC block in the memory 104 uses only a program-pulse step (no program-verify step). In this example, it is assumed that the program-verify step is redundant for PEC threshold< SOK, as that is considered beginning of life (BOL) in this example. In contrast, if the PEC is over the threshold, the memory will use the regular-speed programming mode in which a program-verify step is performed after the program pulse. In this example, it is assumed that PEC > SOK implies that the memory is at end oflife (EOL) and there may be a greater likelihood of programming errors. In this way, by looking at the PEC of the memory 104, the controller 102 can

accommodate system requirements for performance enhancement via a high-speed write without sacrificing data integrity (e.g., by sacrificing the bit error rate threshold beyond which system performance will suffer).

[0072] As yet another example, the storage system criterion can be satisfied by determining that the fast-write mode will achieve higher performance versus there being a performance bottleneck upstream of the memory 104. This embodiment will be illustrated in conjunction with the diagram in Figure 9, which shows several components inside the storage system 100 of an embodiment. As shown in Figure 9, in this example, the memory 104 comprises a plurality of NAND dies that are organized into several channels, as established by the flash interface manager 130. Here, each channel has at least one die. However, other configurations are possible. For example, instead of being a multi-die, multi-channel system (e.g., n being multiples of 2), the storage system 100 can include a single channel (e.g., n=0).

[0073] As also shown in Figure 9, error correction circuitry 124 (here, an LDPC encoder) encodes data before it is supplied to the flash interface manager 130 for transmission to the memory 104 via the channels. In some situations, the encoding speed of the encoder 124 can be a bottleneck, meaning that a fast write will not incur a performance increase (although it still can incur a data integrity penalty because the verify step is skipped). Accordingly, in one embodiment, the controller 102 conducts this analysis before instructing the memory 104 to enter the fast write mode. The following example will illustrate how the controller 102 can gauge the LDPC engine 124 encoding speed in order to match up the memory cell programming performance, thereby becoming intelligent regarding the capacity configuration for the program operation.

[0074] In this example, the SLC program speed is 250MB/s per die. If there are eight channels, the eight-die parallel operation will lead to 2000MB/s. The storage system has a given spec for the LDPC engine 124 throughput speed (e.g., including scrambling, encoding, and header combinations). If the LDPC encoding speed is 3200MB/s, then the bottleneck will be the memory cell performance, which in theory should go up to 400MB/s. In this situation, the controller 102 can determine that a fast write is warranted. On the other hand, if the system 100 has 32 dies operating in parallel, then at a 250MB/s program speed on the memory, the LDPC engine 124 will become the bottleneck as it cannot keep up with encoding data at the speed required by the memory, resulting in idle die(s) for programming. In this situation, the controller 102 can determine that a fast write is not warranted.

[0075] In the above example, the controller 102 looked at whether the memory die parallel operating speed is less than the encoding speed. In a variation of this embodiment, the controller 102 can instead perform this analysis by comparing a number of memory dies being operated in parallel against a threshold. For example, in an eight- channel system, a fast write is warranted when fewer than eight dies are used. That is, when $8*M < X (=3200)$, then the memory performance is the bottleneck for performance, and a high-speed write will increase performance. However, when more than eight dies are used (e.g., n=l 6), $16*N > X (=3200)$, the LDPC throughput - not memory performance - is the bottleneck for performance. In this case, a regular-speed write is warranted.

[0076] It should be noted that the controller 102 can consider other storage system criterion as well, including, but not limited to power (the higher the power, the more stable the program operation), temperature (the lower the temperature, the faster the program operation), and write mode (sequential may be preferred for fast writes). Also,

as with the fast read embodiments, these various factors can considered alone or in combination. As an illustration of this, Figure 10 shows a flow chart 1000 in which several factors are considered. As shown in Figure 10, at the start of an SLC programming operation (act 1005), the controller 102 checks the program-erase count (PEC) of a block to be programmed (act 1010). If the PEC is above a threshold (here, 50) (act 1020), the controller 102 lets the memory 104 operate in the normal read mode (with verify) (act 1030), and the program operation is initiated (act 1035). However, if the PEC is below the threshold, the controller 102 checks the parallelism for the program (act 1040). If the parallelism is below a threshold (here, 8 channels) (act 1050), the controller 102 trims the setting in the memory 104 for the high-speed mode (without verify) (act 1060). If the parallelism is above the threshold, the controller 102 lets the memory 104 operate in the normal read mode (with verify) (act 1070).

[0077] Finally, as mentioned above, any suitable type of memory can be used. Semiconductor memory devices include volatile memory devices, such as dynamic random access memory ("DRAM") or static random access memory ("SRAM") devices, non-volatile memory devices, such as resistive random access memory ("ReRAM"), electrically erasable programmable read only memory ("EEPROM"), flash memory (which can also be considered a subset ofEEPROM), ferroelectric random access memory ("FRAM"), magnetoresistive random access memory ("MRAM"), phase-change memory ("PCM"), and other semiconductor elements capable of storing information. Each type of memory device may have different configurations. For example, flash memory devices may be configured in a NAND or a NOR configuration.

[0078] The memory devices can be formed from passive and/or active elements, in any combinations. By way of non-limiting example, passive semiconductor memory elements include ReR_A.M device elements, which in some embodiments include a

resistivity switching storage element, such as an anti-fuse, phase change material, etc., and optionally a steering element, such as a diode, etc. Further by way of non-limiting example, active semiconductor memory elements include EEPROM and flash memory device elements, which in some embodiments include elements containing a charge storage region, such as a floating gate, conductive nanoparticles, or a charge storage dielectric material.

[0079] Multiple memory elements may be configured so that they are connected in series or so that each element is individually accessible. By way of non-limiting example, flash memory devices in a NAND configuration (NAND memory) typically contain memory elements connected in series. ANAND memory array may be configured so that the array is composed of multiple strings of memory in which a string is composed of multiple memory elements sharing a single bit line and accessed as a group. Alternatively, memory elements may be configured so that each element is individually accessible, e.g., a NOR memory array. NAND and NOR memory configurations are examples, and memory elements may be otherwise configured.

[0080] The semiconductor memory elements located within and/or over a substrate may be arranged in two or three dimensions, such as a two dimensional memory structure or a three dimensional memory structure.

[0081] In a two dimensional memory structure, the semiconductor memory elements are arranged in a single plane or a single memory device level. Typically, in a two dimensional memory structure, memory elements are arranged in a plane (e.g., in an x-z direction plane) which extends substantially parallel to a major surface of a substrate that supports the memory elements. The substrate may be a wafer over or in which the layer of the memory elements are formed or it may be a carrier substrate which is attached to the

memory elements after they are formed. As a non-limiting example, the substrate may include a semiconductor such as silicon.

[0082] The memory elements may be arranged in the single memory device level in an ordered array, such as in a plurality of rows and/or columns. However, the memory elements may be arrayed in non-regular or non-orthogonal configurations. The memory elements may each have two or more electrodes or contact lines, such as bit lines and word lines.

[0083] A three dimensional memory array is arranged so that memory elements occupy multiple planes or multiple memory device levels, thereby forming a structure in three dimensions (i.e., in the x, y and z directions, where they direction is substantially perpendicular and the x and z directions are substantially parallel to the major surface of the substrate).

[0084] As a non-limiting example, a three dimensional memory structure may be vertically arranged as a stack of multiple two dimensional memory device levels. As another non-limiting example, a three dimensional memory array may be arranged as multiple vertical columns (e.g., columns extending substantially perpendicular to the major surface of the substrate, i.e., in the y direction) with each column having multiple memory elements in each column. The columns may be arranged in a two dimensional configuration, e.g., in an x-z plane, resulting in a three dimensional arrangement of memory elements with elements on multiple vertically stacked memory planes. Other configurations of memory elements in three dimensions can also constitute a three dimensional memory array.

[0085] By way of non-limiting example, in a three dimensional NAND memory array, the memory elements may be coupled together to form a NAND string within a single horizontal (e.g., x-z) memory device levels. Alternatively, the memory elements

may be coupled together to form a vertical NAND string that traverses across multiple horizontal memory device levels. Other three dimensional configurations can be envisioned wherein some NAND strings contain memory elements in a single memory level while other strings contain memory elements which span through multiple memory levels. Three dimensional memory arrays may also be designed in a NOR configuration and in a ReRAM configuration.

[0086] Typically, in a monolithic three dimensional memory array, one or more memory device levels are formed above a single substrate. Optionally, the monolithic three dimensional memory array may also have one or more memory layers at least partially within the single substrate. As a non-limiting example, the substrate may include a semiconductor such as silicon. In a monolithic three dimensional array, the layers constituting each memory device level of the array are typically formed on the layers of the underlying memory device levels of the array. However, layers of adjacent memory device levels of a monolithic three dimensional memory array may be shared or have intervening layers between memory device levels.

[0087] Then again, two dimensional arrays may be formed separately and then packaged together to form a non-monolithic memory device having multiple layers of memory. For example, non-monolithic stacked memories can be constructed by forming memory levels on separate substrates and then stacking the memory levels atop each other. The substrates may be thinned or removed from the memory device levels before stacking, but as the memory device levels are initially formed over separate substrates, the resulting memory arrays are not monolithic three dimensional memory arrays. Further, multiple two dimensional memory arrays or three dimensional memory arrays (monolithic or non-monolithic) may be formed on separate chips and then packaged together to form a stacked-chip memory device.

[0088] Associated circuitry is typically required for operation of the memory elements and for communication with the memory elements. As non-limiting examples, memory devices may have circuitry used for controlling and driving memory elements to accomplish functions such as programming and reading. This associated circuitry may be on the same substrate as the memory elements and/or on a separate substrate. For example, a controller for memory read-write operations may be located on a separate controller chip and/or on the same substrate as the memory elements.

[0089] One of skill in the art will recognize that this invention is not limited to the two dimensional and three dimensional structures described but cover all relevant memory structures within the spirit and scope of the invention as described herein and as understood by one of skill in the art.

[0090] It is intended that the foregoing detailed description be understood as an illustration of selected forms that the invention can take and not as a definition of the invention. It is only the following claims, including all equivalents, that are intended to define the scope of the claimed invention. Finally, it should be noted that any aspect of any of the embodiments described herein can be used alone or in combination with one another.

CLAIMS

What is claimed is:

1. A method for reading a memory of a storage system, the method comprising:

performing the following in a storage system comprising a memory configured to

operate in a first read mode and a second read mode, wherein the first read mode is faster than the second read mode but can result in an increased bit error rate:

determining whether to operate the memory in the first read mode based on whether operating the memory in the first read mode will likely increase a bit error rate above a threshold; and

in response to determining that operating the memory in the first read mode will not likely increase the bit error rate above the threshold, operating the memory in the first read mode.

2. The method of Claim 1 further comprising increasing a power supply to the memory when operating the memory in the first read mode.

3. The method of Claim 1, wherein determining whether to operate the memory in the first read mode is performed in response to receiving a request for a random read.

4. The method of Claim 1, wherein a likelihood that the bit error rate will increase above the threshold is based on a temperature of the memory.

5. The method of Claim 1, wherein a likelihood that the bit error rate will increase above the threshold is based on a location of a word line to be read in the memory.

6. The method of Claim 1, wherein the memory comprises a multi-level cell (MLC) memory, and wherein a likelihood that the bit error rate will increase above the threshold is based on a location of page to be read in the memory.

7. The method of Claim 1, wherein a likelihood that the bit error rate will increase above the threshold is based on a power supply level.

8. The method of Claim 1, wherein the memory comprises a three-dimensional memory.

9. The method of Claim 1, wherein the storage system is embedded in a host.

10. A storage system comprising:

a memory configured to operate in a first write mode and a second write mode, wherein the first write mode is faster than the second write mode; and

a controller configured to:

determine whether to instruct the memory to operate in the first write mode based on whether a storage system criterion is satisfied; and

in response to determining that the storage system criterion is satisfied, instruct the memory to operate in the first write mode.

11. The storage system of Claim 10, wherein the storage system criterion is satisfied when a program-erase count of the memory is under a threshold.

12. The storage system of Claim 10 further comprising an error correction code encoder configured to operate at a first encoding speed, wherein the memory comprises a plurality of memory dies, and wherein the storage system criterion is satisfied when a memory die parallel operating speed is less than the first encoding speed.

13. The storage system of Claim 10, wherein the memory comprises a plurality of memory dies, and wherein the storage system criterion is satisfied when a number of memory dies being operated in parallel is below a threshold.

14. The storage system of Claim 10, wherein the storage system criterion is satisfied when the memory comprising single-level cells (SLCs).

15. The storage system of Claim 10, wherein the second write mode performs a verify operation to check that data has been written correctly in the memory, whereas the first write mode does not.

16. The storage system of Claim 10, wherein the memory comprises a three-dimensional memory.

17. The storage system of Claim 10, wherein the storage system is embedded in a host.

18. A storage system comprising: a memory; and

at least one of the following:

means for performing a fast read in response to determining that the fast read will probably not have a negative impact on performance due to error correction; and

means for performing a fast write in response to a storage system criterion being satisfied.

19. The storage system of Claim 18, wherein the memory comprises a three-dimensional memory.

20. The storage system of Claim 18, wherein the storage system is embedded in a host.

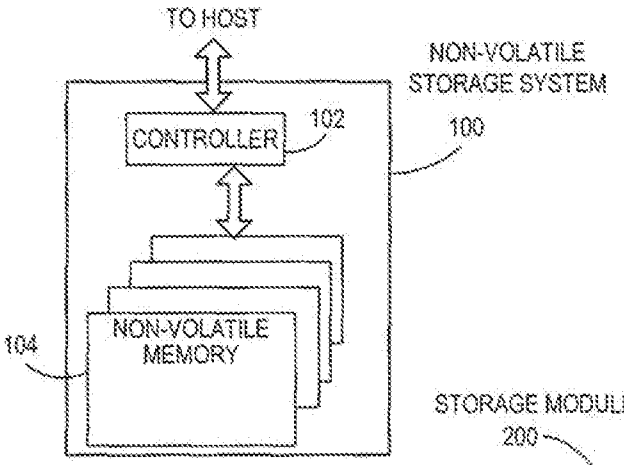

FIG. 1A

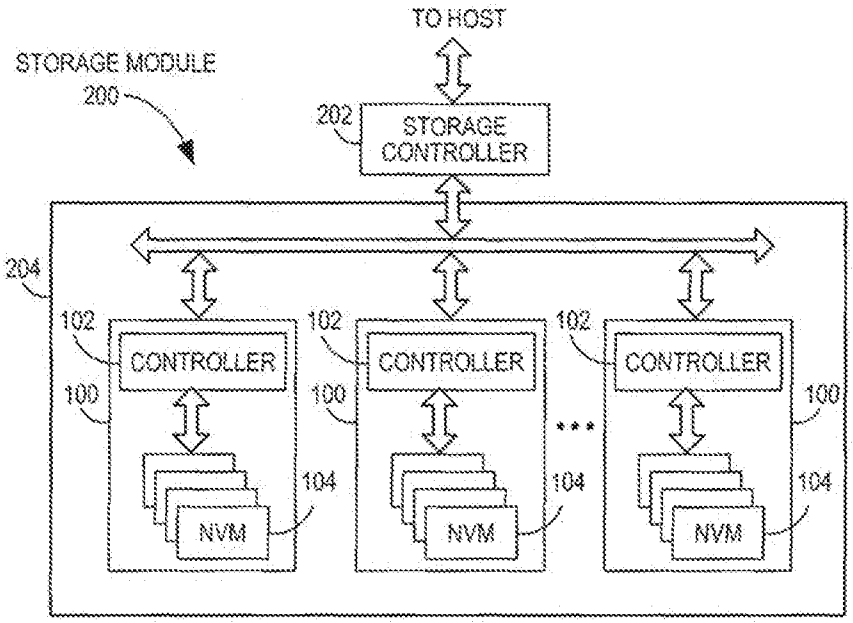

FIG. 1B

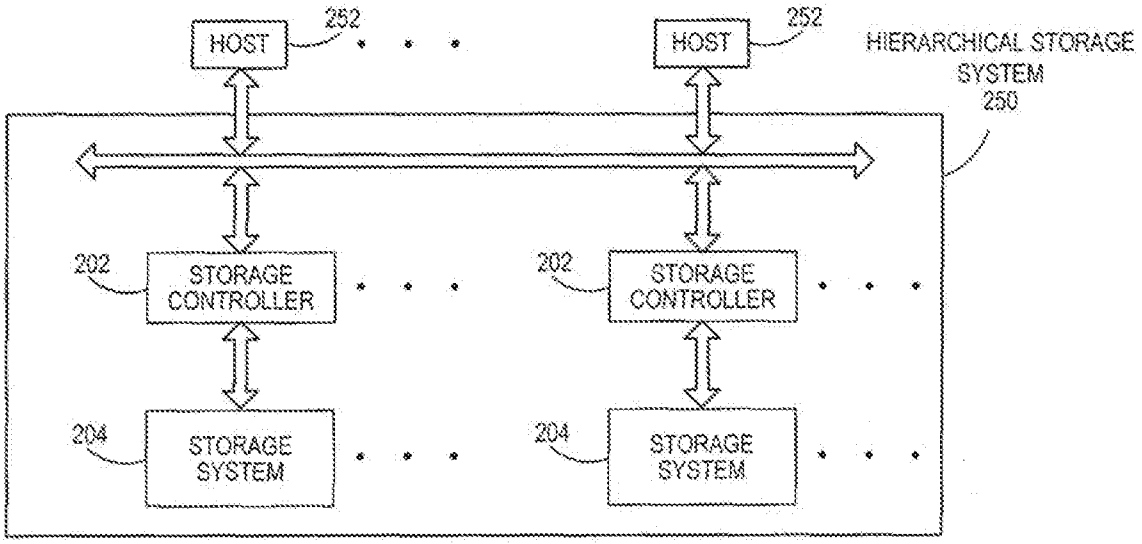

FIG. 1C

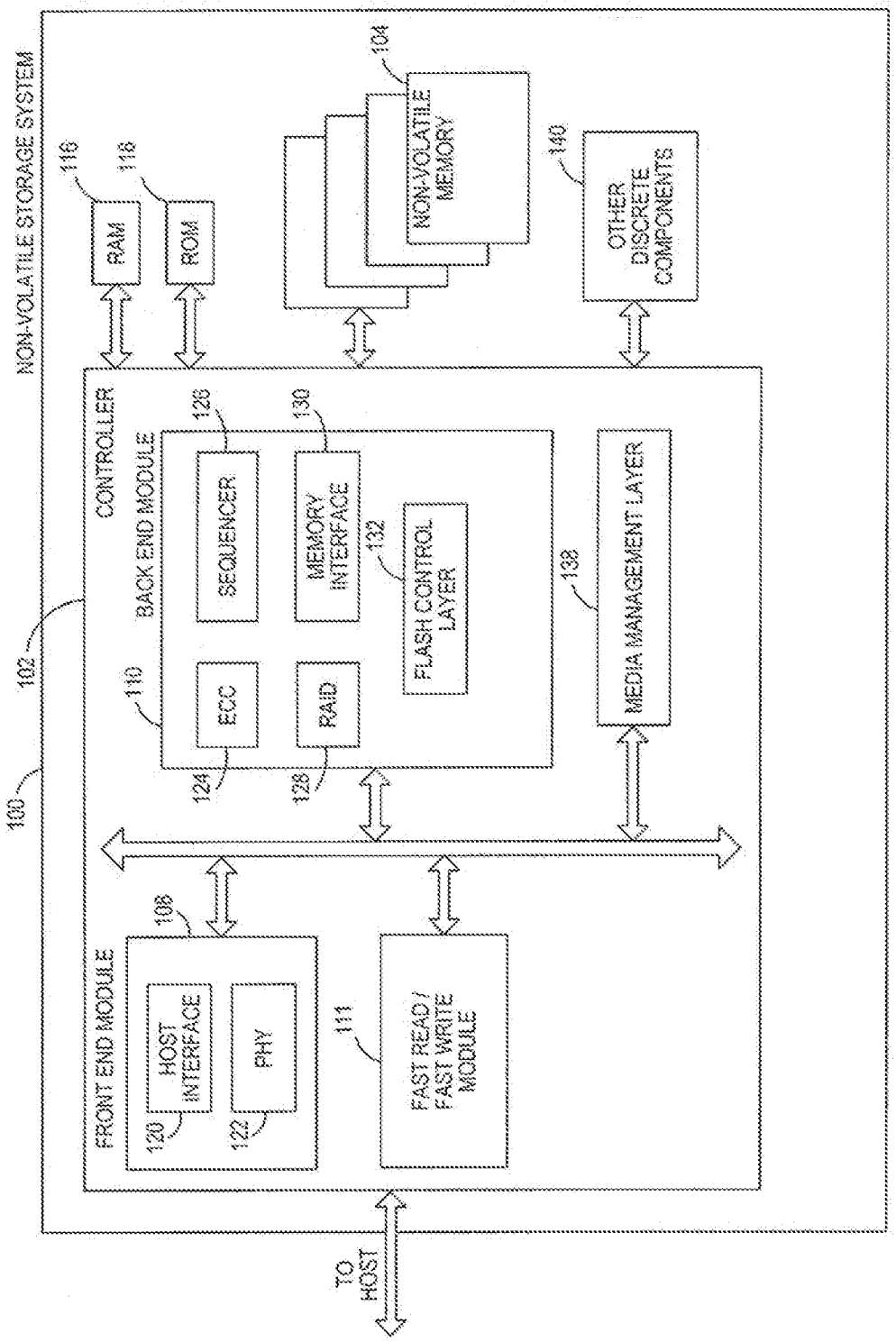
NON-VOLATILE STORAGE SYSTEM
100
102
CONTROLLER
FRONT END MODULE
108
HOST INTERFACE
120
PHY
122
FAST READ / FAST WRITE MODULE
111
BACK END MODULE
110
ECC
124
RAID
126
SEQUENCER
128
MEMORY INTERFACE
130
FLASH CONTROL LAYER
132
MEDIA MANAGEMENT LAYER
138
RAM
116
ROM
118
NON-VOLATILE MEMORY
104
OTHER DISCRETE COMPONENTS
140
TO HOST

FIG. 2A

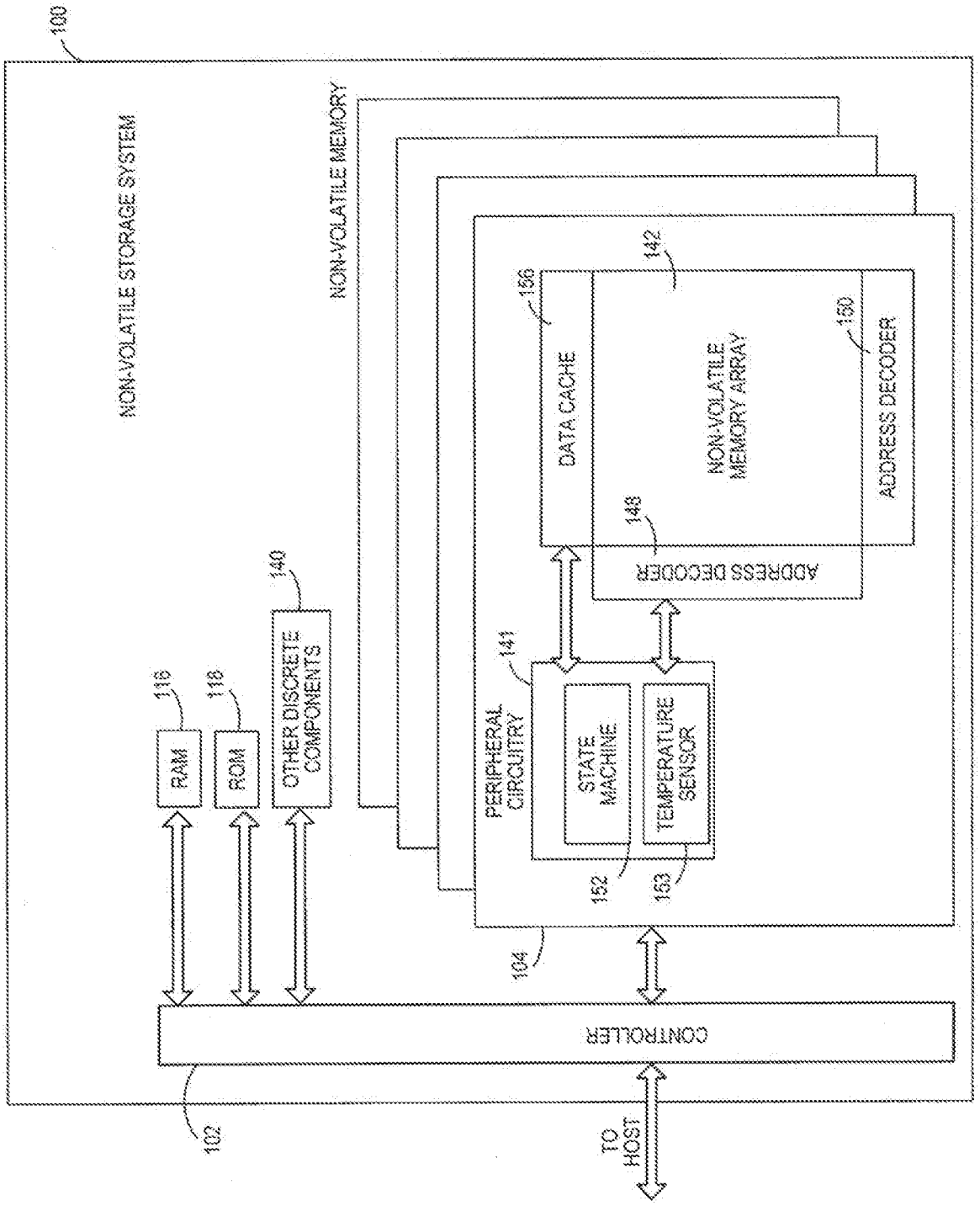
100
NON-VOLATILE STORAGE SYSTEM
NON-VOLATILE MEMORY
142
156
DATA CACHE
NON-VOLATILE MEMORY ARRAY
150
ADDRESS DECODER
148
ADDRESS DECODER
140
OTHER DISCRETE COMPONENTS
116
RAM
118
ROM
141
PERIPHERAL CIRCUITRY
STATE MACHINE
TEMPERATURE SENSOR
152
163
104
CONTROLLER
102
TO HOST

FIG. 2B

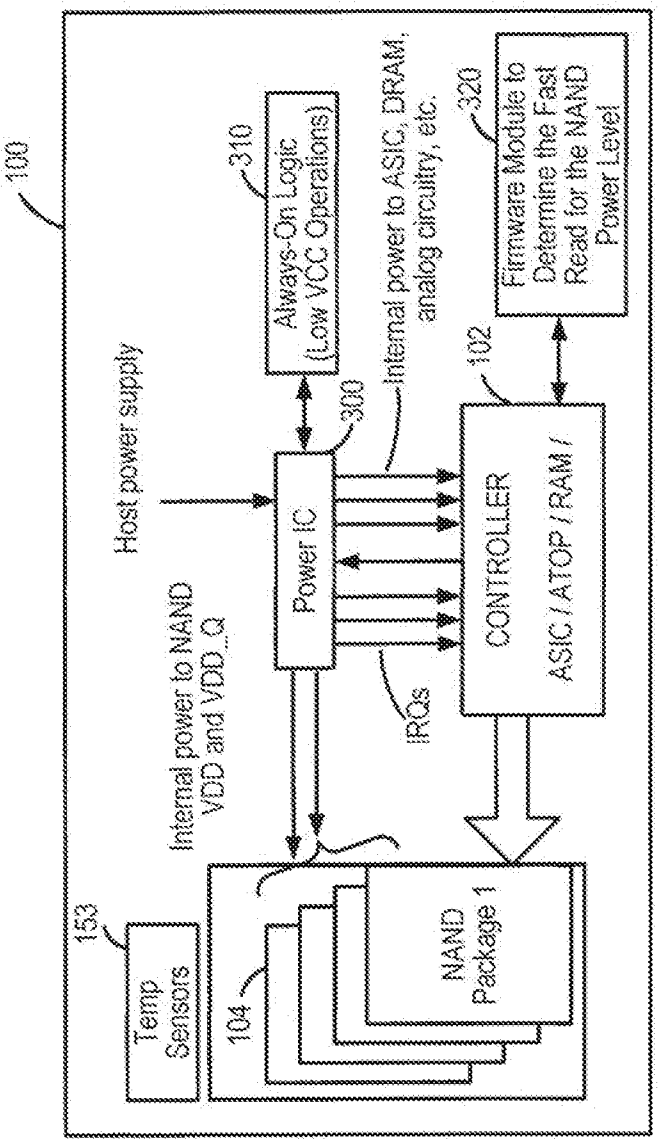
100
Host power supply
Internal power to NAND
VDD and VDD_Q
Power IC
300
310
Always-On Logic
(Low VCC Operations)
Internal power to ASIC, DRAM,
analog circuitry, etc.
IRQs
102
CONTROLLER
ASIC / ATOP / RAM /
320
Firmware Module to
Determine the Fast
Read for the NAND
Power Level
153
Temp
Sensors
104
NAND
Package 1

FIG. 3

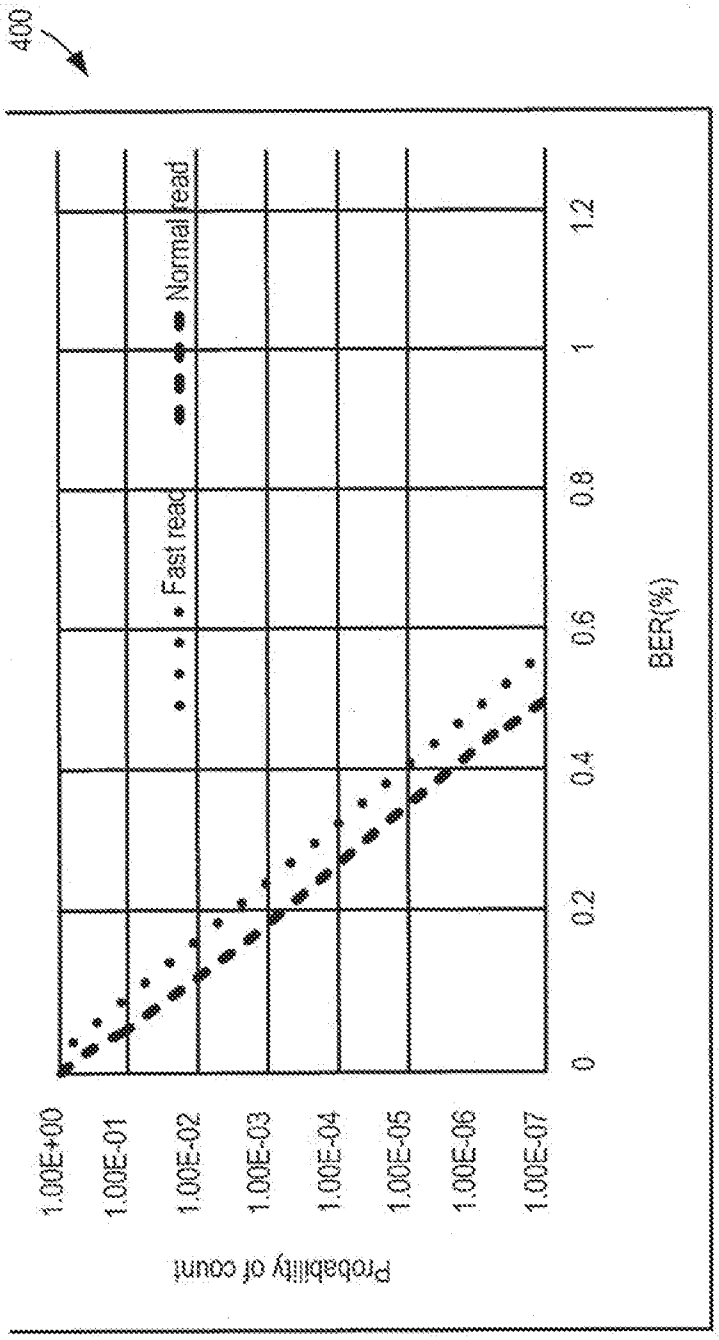
400
Probability of count
1.00E+00
1.00E-01
1.00E-02
1.00E-03
1.00E-04
1.00E-05
1.00E-06
1.00E-07
0
0.2
0.4
0.6
0.8
1
1.2
BER(%)
Fast read
Normal read

FIG. 4

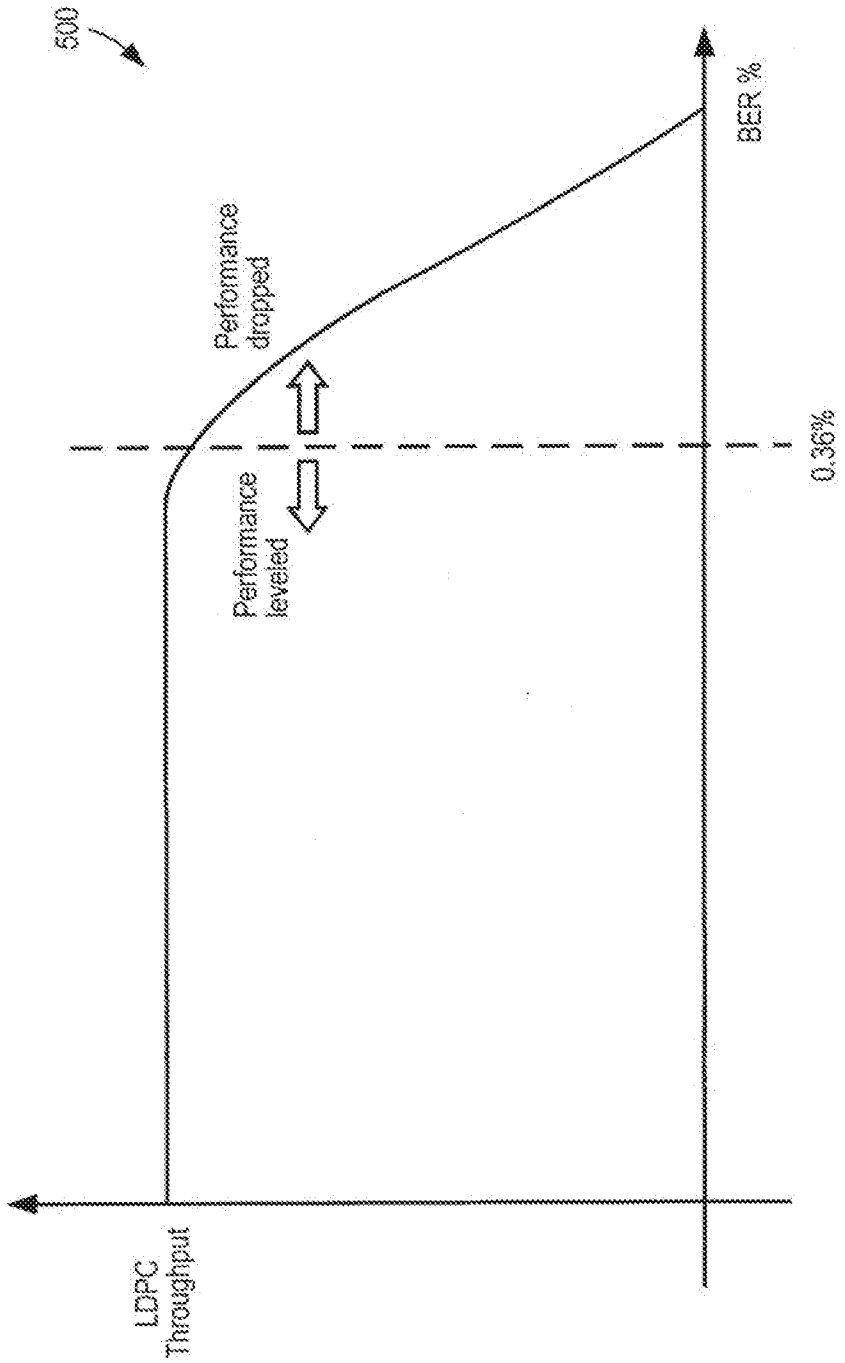
500
BER %
Performance dropped
0.36%
Performance leveled
LDPC Throughput

FIG. 5

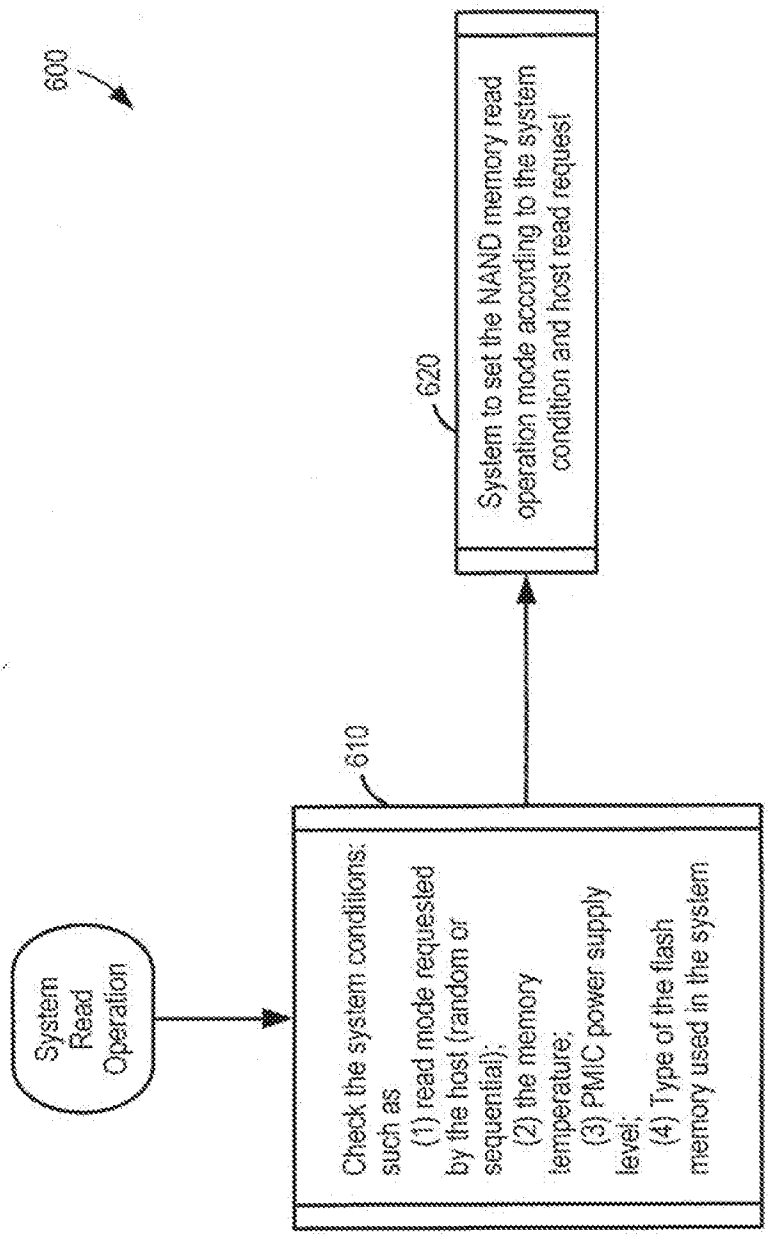
600
System Read Operation
Check the system conditions: such as
(1) read mode requested by the host (random or sequential);
(2) the memory temperature;
(3) PMIC power supply level;
(4) Type of the flash memory used in the system
610
System to set the NAND memory read operation mode according to the system condition and host read request
620

FIG. 6

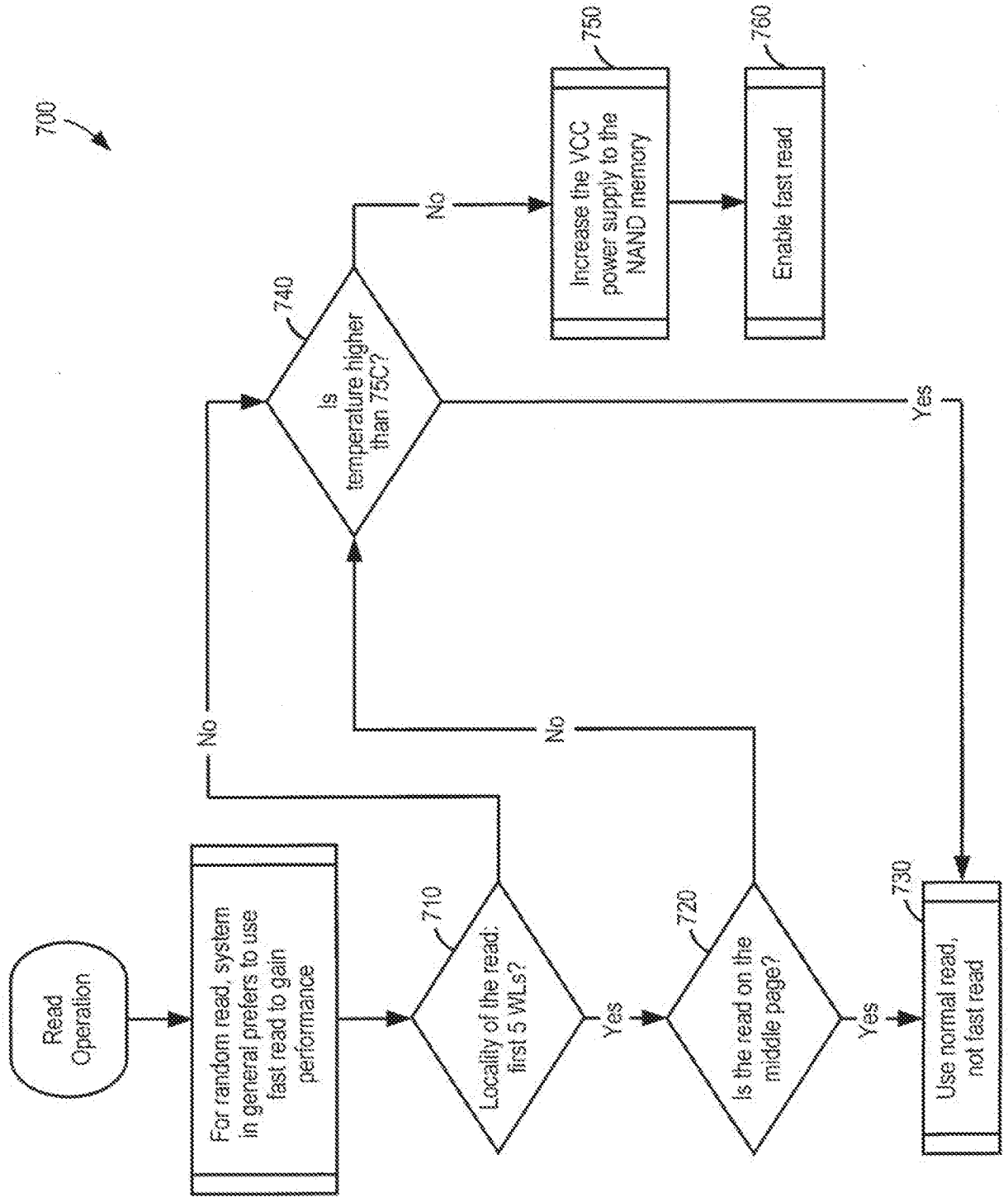
700
Read Operation
For random read, system in general prefers to use fast read to gain performance
710
Locality of the read: first 5 WLs?
Yes
No
720
Is the read on the middle page?
Yes
No
730
Use normal read, not fast read
740
Is temperature higher than 75C?
Yes
No
750
Increase the VCC power supply to the NAND memory
760
Enable fast read

FIG. 7

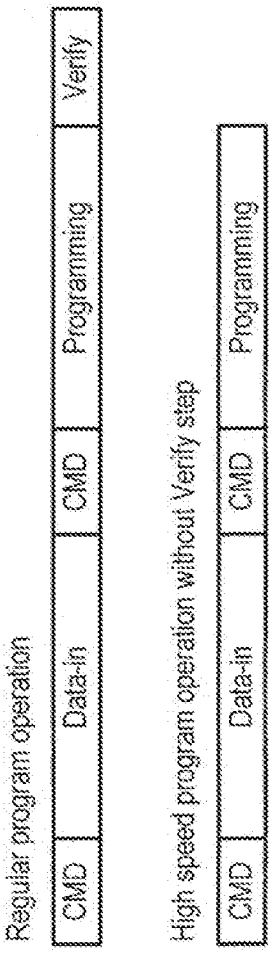
Regular program operation
CMD
Data-in
CMD
Programming
Verify
High speed program operation without Verify step
CMD
Data-in
CMD
Programming

FIG. 8

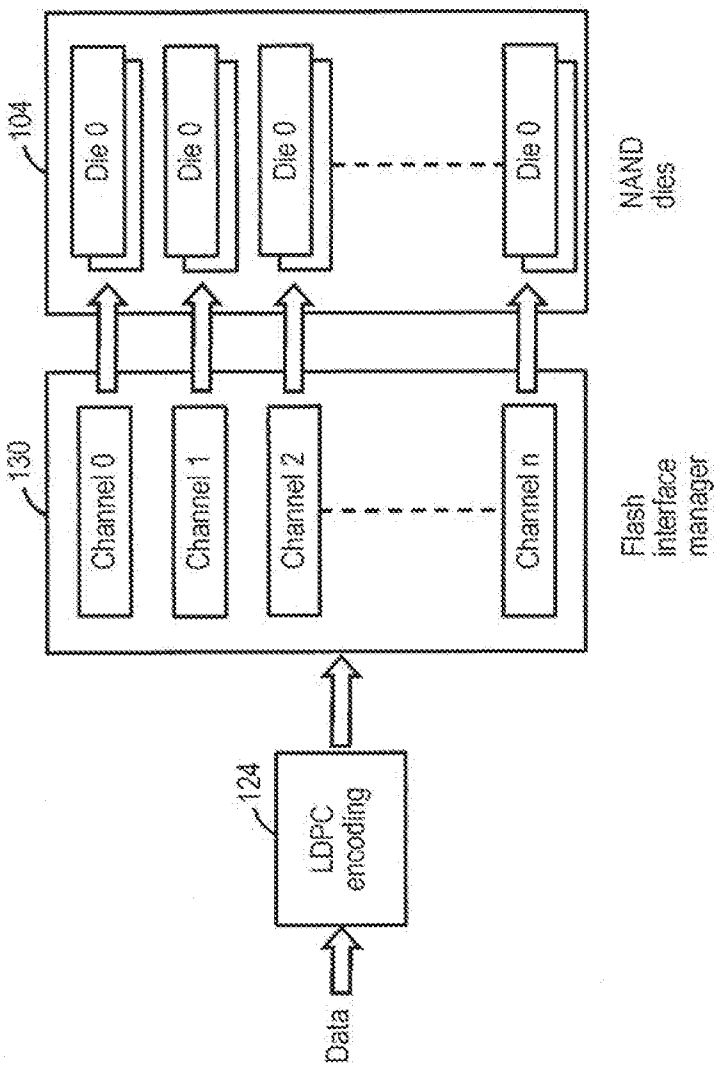
104
Die 0
Die 0
Die 0
Die 0
NAND dies
130
Channel 0
Channel 1
Channel 2
Channel n
Flash interface manager
124
LDPC encoding
Data

FIG. 9

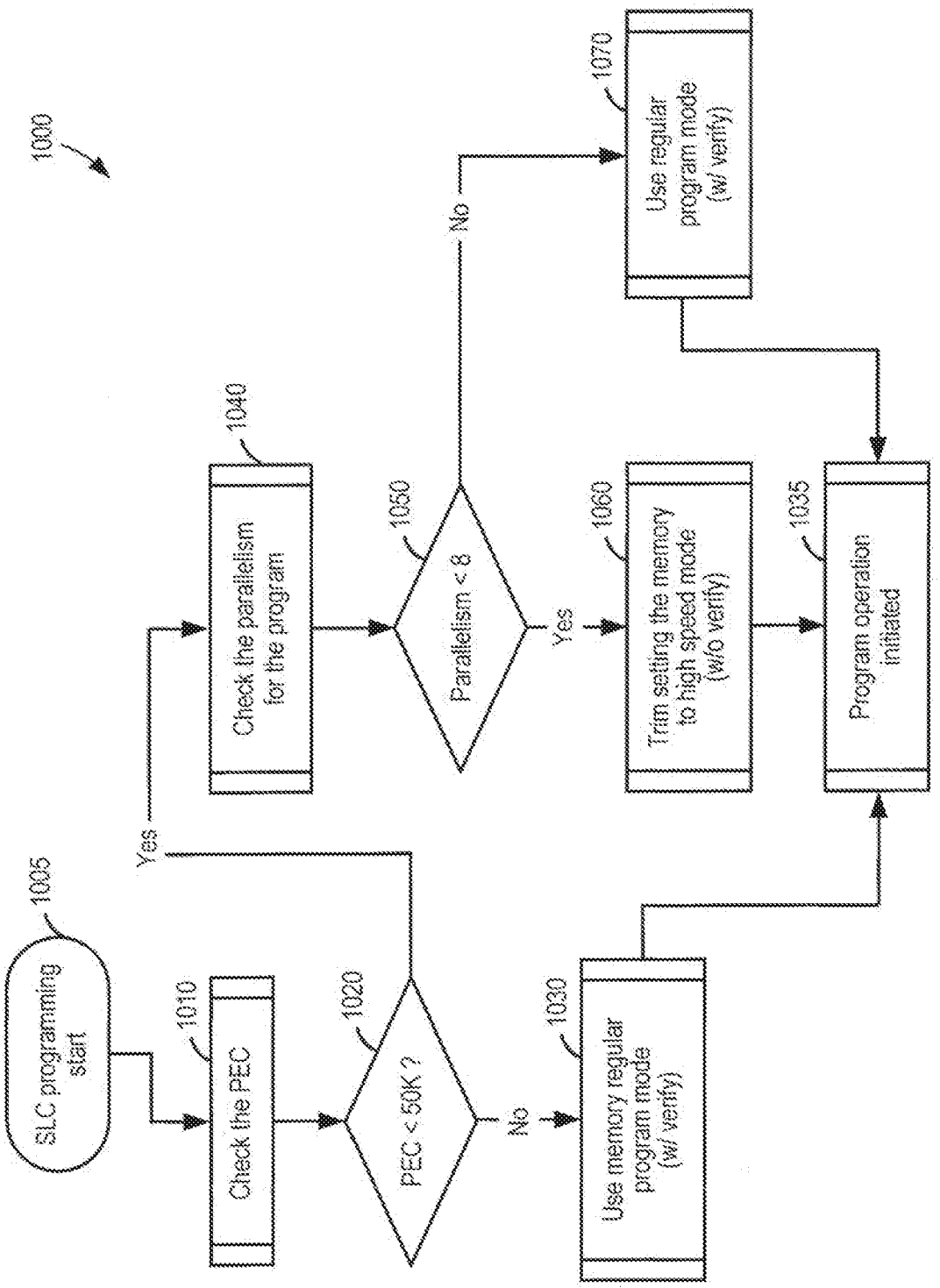
1000
1005
SLC programming start
1010
Check the PEC
1020
PEC < 50K ?
Yes
No
1030
Use memory regular program mode (w/ verify)
1040
Check the parallelism for the program
1050
Parallelism < 8
Yes
No
1060
Trim setting the memory to high speed mode (w/o verify)
1070
Use regular program mode (w/ verify)
1035
Program operation initiated

FIG. 10

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2018/066434

A. CLASSIFICATION OF SUBJECT MATTER

G06F 3/06(2006.01)i, G11C 16/10(2006.01)i, G11C 16/26(2006.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F 3/06; G06F 11/00; G11C 16/04; G11C 29/04; G11C 5/02; G11C 5/06; G11C 7/00; G11C 8/00; G11C 16/10; G11C 16/26

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & keywords: read, write, speed, memory, threshold, bit error rate, temperature, 3-D memory

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 8499206 B2 (BEOM-SIG CHO et al.) 30 July 2013 See column 3, line 65 - column 7, line 33; claim 1; and figure 8.	1-2,4,7,10,18
Y		3,5-6,8-9,11,14,16-17,19-20
A		12-13,15
Y	US 6349059 B1 (SIMONE BARTOLI et al.) 19 February 2002 See column 2, line 10 - column 3, line 2; and claim 6.	3
Y	US 8750042 B2 (ERAN SHARON et al.) 10 June 2014 See column 1, line 41 - column 47, line 17.	5-6,9,11,14,17,20
Y	WO 2011-025679 A1 (MICRON TECHNOLOGY INC.) 03 March 2011 See paragraph [0008].	8,16,19
A	US 6944067 B2 (TREVOR NIGEL MUDGE et al.) 13 September 2005 See column 9, lines 4-26; and figure 7.	1-20

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

☒ See patent family annex.

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

Date of the actual completion of the international search	Date of mailing of the international search report
01 April 2019 (01.04.2019)	**02 April 2019 (02.04.2019)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea
Facsimile No. +82-42-481-8578

Authorized officer
KANG, Hee Gok
Telephone No. +82-42-481-8264

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2018/066434

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 8499206 B2	30/07/2013	KR 10-2011-0100465 A	14/09/2011
		US 2011-0219274 A1	08/09/2011
US 6349059 B1	19/02/2002	EP 1103977 A1	30/05/2001
		EP 1103977 B1	25/02/2009
US 8750042 B2	10/06/2014	CN 103814409 A	21/05/2014
		CN 103827971 A	28/05/2014
		CN 103827971 B	08/02/2017
		EP 2737486 A2	04/06/2014
		EP 2737486 B1	06/06/2018
		EP 2737487 A1	04/06/2014
		EP 2737488 A2	04/06/2014
		KR 10-1835174 B1	06/03/2018
		KR 10-1838340 B1	13/03/2018
		KR 10-2014-0053207 A	07/05/2014
		KR 10-2014-0064785 A	28/05/2014
		KR 10-2014-0065393 A	29/05/2014
		TW 201314698 A	01/04/2013
		TW 201319801 A	16/05/2013
		US 2013-0028021 A1	31/01/2013
		US 2013-0031430 A1	31/01/2013
		US 2013-0031431 A1	31/01/2013
		US 2014-0254272 A1	11/09/2014
		US 8726104 B2	13/05/2014
		US 8873288 B2	28/10/2014
		WO 2013-016390 A2	31/01/2013
		WO 2013-016390 A3	20/06/2013
		WO 2013-016397 A2	31/01/2013
		WO 2013-016397 A3	18/04/2013
		WO 2013-016401 A1	31/01/2013
		WO 2013-016401 A4	31/01/2013
WO 2011-025679 A1	03/03/2011	CN 102625948 A	01/08/2012
		CN 102625948 B	11/03/2015
		KR 10-1722800 B1	05/04/2017
		KR 10-2012-0067349 A	25/06/2012
		TW 201126531 A	01/08/2011
		TW I470638 B	21/01/2015
		US 2011-0051512 A1	03/03/2011
		US 8320181 B2	27/11/2012
US 6944067 B2	13/09/2005	CN 100449651 C	07/01/2009
		CN 100468286 C	11/03/2009
		CN 1761927 A	19/04/2006
		CN 1761945 A	19/04/2006
		CN 1761945 C	03/09/2008
		CN 1761946 A	19/04/2006
		CN 1761946 C	09/07/2008

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2018/066434

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		CN 1762028 A	19/04/2006
		DE 602004001228 T2	03/05/2007
		DE 602004001679 T2	02/08/2007
		DE 602004001869 T2	03/05/2007
		EP 1604265 A1	14/12/2005
		EP 1604265 B1	14/06/2006
		EP 1604281 A1	14/12/2005
		EP 1604281 B1	09/08/2006
		EP 1604282 A1	14/12/2005
		EP 1604282 B1	22/08/2018
		EP 1604371 A1	14/12/2005
		EP 1604371 B1	26/07/2006
		IL 168453 A	31/05/2010
		IL 168453 B	31/05/2010
		IL 168514 B	31/10/2010
		IL 168928 A	31/05/2010
		IL 168928 B	31/05/2010
		IL 169151 A	31/05/2010
		IL 169151 B	31/05/2010
		IL 169151 D0	04/07/2007
		JP 2006-520952 A	14/09/2006
		JP 2006-520953 A	14/09/2006
		JP 2006-520954 A	14/09/2006
		JP 2006-520955 A	14/09/2006
		JP 4279874 B2	17/06/2009
		JP 4317212 B2	19/08/2009
		JP 4335253 B2	30/09/2009
		JP 4426571 B2	03/03/2010
		KR 10-0955285 B1	30/04/2010
		KR 10-0981999 B1	13/09/2010
		KR 10-0982461 B1	16/09/2010
		KR 10-0994188 B1	12/11/2010
		KR 10-2005-0117559 A	14/12/2005
		KR 10-2005-0118184 A	15/12/2005
		KR 10-2005-0118185 A	15/12/2005
		KR 10-2006-0009236 A	31/01/2006
		MY 136842 A	28/11/2008
		RU 2005129253 A	10/04/2006
		RU 2005129257 A	27/01/2006
		RU 2005129270 A	27/05/2006
		RU 2005129281 A	27/01/2006
		TW 200507396 A	16/02/2005
		TW I309904 A	11/05/2009
		TW I309904 B	11/05/2009
		US 2004-0199821 A1	07/10/2004
		US 2004-0223386 A1	11/11/2004
		US 2004-0239397 A1	02/12/2004
		US 2004-0243893 A1	02/12/2004
		US 2005-0022094 A1	27/01/2005
		US 2005-0207521 A1	22/09/2005

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2018/066434

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		US 2005-0246613 A1	03/11/2005
		US 2006-0018171 A1	26/01/2006
		US 2006-0280002 A1	14/12/2006
		US 2007-0162798 A1	12/07/2007
		US 2007-0288798 A1	13/12/2007
		US 2010-0058107 A1	04/03/2010
		US 2011-0093737 A1	21/04/2011
		US 2011-0107166 A1	05/05/2011
		US 2011-0126051 A1	26/05/2011
		US 2014-0013178 A1	09/01/2014
		US 2014-0181581 A1	26/06/2014
		US 2016-0034339 A1	04/02/2016
		US 2016-0378588 A1	29/12/2016
		US 7072229 B2	04/07/2006
		US 7162661 B2	09/01/2007
		US 7260001 B2	21/08/2007
		US 7278080 B2	02/10/2007
		US 7310755 B2	18/12/2007
		US 7320091 B2	15/01/2008
		US 7337356 B2	26/02/2008
		US 7401273 B2	15/07/2008
		US 7650551 B2	19/01/2010
		US 8060814 B2	15/11/2011
		US 8185786 B2	22/05/2012
		US 8185812 B2	22/05/2012
		US 8407537 B2	26/03/2013
		US 8650470 B2	11/02/2014
		US 9164842 B2	20/10/2015
		US 9448875 B2	20/09/2016
		WO 2004-084053 A1	30/09/2004
		WO 2004-084070 A1	30/09/2004
		WO 2004-084072 A1	30/09/2004
		WO 2004-084233 A1	30/09/2004