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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
12 December 2024 (12.12.2024)

WIPO | PCT

(10) International Publication Number
WO 2024/253710 A1

(51) International Patent Classification:
G06F 3/06 (2006.01) *G06F 12/06* (2006.01)

(21) International Application Number:
PCT/US2024/011238

(22) International Filing Date:
11 January 2024 (11.01.2024)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
63/471,411 06 June 2023 (06.06.2023) US
18/226,385 26 July 2023 (26.07.2023) US

(71) Applicant: **SANDISK TECHNOLOGIES, INC.** [US/US]; 951 Sandisk Drive, Legal Dep., Milpitas, California 95035 (US).

(72) Inventors: **NAIR, Devika**; c/o Western Digital Technologies, Inc., 5601 Great Oaks Parkway, San Jose, California 95119 (US). **SHARMA, Amit**; c/o Western Digital Technologies, Inc., 5601 Great Oaks Parkway, San Jose, California 95119 (US).

(74) Agent: **HETZ, Joseph F.** et al.; Crowell & Moring LLP, 455 North Cityfront Plaza Drive, Suite 3600, Chicago, Illinois 60611 (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, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, 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, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, 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, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, 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))*

(54) Title: DATA STORAGE DEVICE AND METHOD FOR PREDICTABLE LOW-LATENCY IN A TIME-SENSITIVE ENVIRONMENT

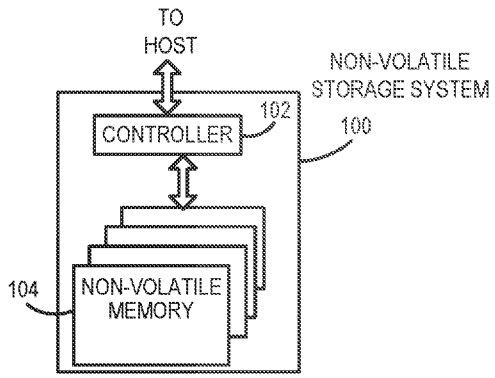

FIG. 1A

(57) Abstract: A data storage device and method are provided for predictable low-latency in a time-sensitive environment. In one embodiment, a data storage device is provided comprising a memory and a controller configured to communicate with the memory. The controller is further configured to: receive, from a host, an indication of a logical block address range that the host will later read; and in response to receiving the indication: read data from the logical block address range; and perform an action on the data to reduce a read latency when the host later reads the logical block address range. Other embodiments are disclosed.

WO 2024/253710 A1

Data Storage Device and Method for Predictable Low-Latency in a Time-Sensitive Environment

Cross-Reference to Related Application

[0001] This application claims the benefit of and hereby incorporates by reference, for all purposes, the entirety of the contents of U.S. Nonprovisional Application No 18/226,385, entitled "Data Storage Device and Method for Predictable Low-Latency in a Time-Sensitive Environment" and filed in the United States Patent & Trademark Office on July 26, 2023, which is claims priority of U.S. Provisional application number 63/471,411 filed June 6, 2023.

Background

[0002] A host can retrieve data from and/or store data in a data storage device. Some hosts may require data be retrieved from the data storage device in a certain amount of time. For example, advancements in autonomous vehicles (e.g., autonomous cars) and robotics can demand real-time (or near real-time), low-latency data processing. In such environments, it may be expected that the data storage device will provide the data to the host (e.g., surgical robot) in a predictable and time-sensitive (e.g., time-critical) manner.

Brief Description of the Drawings

[0003] Figure 1A is a block diagram of a data storage device of an embodiment.

[0004] Figure 1B is a block diagram illustrating a storage module of an embodiment.

[0005] Figure 1C is a block diagram illustrating a hierarchical storage system of an embodiment.

[0006] Figure 2A is a block diagram illustrating components of the controller of the data storage device illustrated in Figure 1A according to an embodiment.

[0007] Figure 2B is a block diagram illustrating components of the memory data storage device illustrated in Figure 1A according to an embodiment.

[0008] Figure 3 is a block diagram of a host and data storage device of an embodiment.

[0009] Figure 4 is an illustration of communications between a host and a data storage device of an embodiment for time-sensitive data reads.

[0010] Figure 5 is an illustration of communications between a host and a data storage device of an embodiment where the data storage device cannot provide a time-sensitive data read.

[0011] Figure 6 is an illustration of a method of an embodiment where data is moved from a block having a high bit-error rate (BER) to a block having a low BER rate.

[0012] Figure 7 is an illustration of a method of an embodiment for improving performance of a direct memory access (DMA) operation.

Detailed Description

[0013] The following embodiments generally relate to a data storage device and method for predictable low-latency in a time-sensitive environment. In one embodiment, a data storage device is provided comprising a memory and a controller configured to communicate with the memory. The controller is further configured to: receive, from a host, an indication of a logical block address range that the host will later read; and in response to receiving the indication: read data from the logical block address range; and perform an action on the data to reduce a read latency when the host later reads the logical block address range.

[0014] In some embodiments, the controller is further configured to determine whether the read latency is above a threshold when reading the data from the logical block address range; and the action is performed in response to the read latency being above the threshold.

[0015] In some embodiments, the controller is further configured to, after the action is performed, confirm that the read latency is not above the threshold.

[0016] In some embodiments, the controller is further configured to inform the host in response to the read latency still being above the threshold after the action is performed.

[0017] In some embodiments, the threshold is provided by the host.

[0018] In some embodiments, the action comprises performing an error correction operation on the data.

[0019] In some embodiments, the action comprises moving the data to a different block in the memory.

[0020] In some embodiments, the action comprises moving the data from a multi-level cell (MLC) block to a single-level cell (SLC) block.

[0021] In some embodiments, the action comprises moving the data to an area of the memory reserved for time-sensitive data.

[0022] In some embodiments, the host comprises an automated vehicle.

[0023] In some embodiments, the host comprises a surgical robot.

[0024] In some embodiments, the memory comprises a three-dimensional memory.

[0025] In another embodiment, a method is provided that is performed in a data storage device comprising a memory. The method comprises: receiving, from a host, an indication of a logical block address range that the host will later read; in response to receiving the indication: reading original data from the logical block address range, wherein the original data is stored in the memory with a set of parity bits; and storing a copy of the data in the memory with fewer, if any, parity bits than what is stored with the original data; and in response to receiving a command from the host to read the logical block address range, reading the copy of the data instead of the original data, wherein because the copy of the data is stored with fewer, if any, parity bits than

what is stored with the original data, reading the copy of the data instead of the original data reduces read latency.

[0026] In some embodiments, the data is time-sensitive data.

[0027] In some embodiments, the copy of the data is stored in a single-level cell (SLC) memory.

[0028] In some embodiments, the copy of the data is stored in an area of the memory reserved for time-sensitive data.

[0029] In some embodiments, the host comprises an automated vehicle.

[0030] In some embodiments, the host comprises a surgical robot.

[0031] In some embodiments, the memory comprises a three-dimensional memory.

[0032] In another embodiment, a data storage device is provided comprising: a memory; means for receiving, from a host, an indication of an area of the memory that the host will later read; and means for, in response to receiving the indication: reading data from the area of the memory; and performing an action on the data to reduce a read latency when the host later reads the area of the memory.

[0033] 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.

[0034] Embodiments

[0035] The following embodiments relate to a data storage device (DSD). As used herein, a "data storage device" refers to a device that stores data. Examples of DSDs include, but are not limited to, hard disk drives (HDDs), solid state drives (SSDs), tape drives, hybrid drives, etc. Details of example DSDs are provided below.

[0036] Data storage devices suitable for use in implementing aspects of these embodiments are shown in Figures 1A-1C. Figure 1A is a block diagram illustrating a data storage device 100 according to an embodiment of the subject matter described herein. Referring to Figure 1A, data storage device 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.

[0037] The controller 102 (which may be a non-volatile memory controller (e.g., a flash, resistive random-access memory (ReRAM), phase-change memory (PCM), or magnetoresistive random-access memory (MRAM) 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.

[0038] As used herein, a non-volatile memory controller is a device that manages data stored on non-volatile memory and communicates with a host, such as a computer or electronic device. A non-volatile memory controller can have various functionality in addition to the specific functionality described herein. For example, the non-volatile memory controller can format the non-volatile memory to ensure the memory is operating properly, map out bad non-volatile 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 non-volatile memory controller and implement other features. In operation, when a host needs to read data from or write data to the non-volatile memory, it can communicate with the non-volatile memory controller. If the host provides a logical address to which data is to be read/written, the non-volatile memory controller can convert the logical address received from the host to a physical address in the non-volatile memory. (Alternatively, the host can provide the physical address.) The non-volatile 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).

[0039] Non-volatile memory die 104 may include any suitable non-volatile storage medium, including resistive random-access memory (ReRAM), magnetoresistive random-access memory (MRAM), phase-change memory (PCM), 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) (e.g., dual-level cells, triple-level cells (TLC), quad-level cells (QLC), etc.) 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.

[0040] 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, the data storage device 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, the data storage device 100 may be part of an embedded data storage device.

[0041] Although, in the example illustrated in Figure 1A, the data storage device 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 architectures (such as the ones shown in Figures 1B and 1C), two, four, eight or more memory channels may exist between the controller and the 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.

[0042] Figure 1B illustrates a storage module 200 that includes plural non-volatile data storage devices 100. As such, storage module 200 may include a storage controller 202 that interfaces with a host and with data storage device 204, which includes a plurality of data storage devices 100. The interface between storage controller 202 and data storage devices 100 may be a bus interface, such as a serial advanced technology attachment (SATA), peripheral component interconnect express (PCIe) interface, or double-data-rate (DDR) interface. Storage module 200, in one embodiment, may be a solid state drive (SSD), or non-volatile dual in-line memory module (NVDIMM), such as found in server PC or portable computing devices, such as laptop computers, and tablet computers.

[0043] 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 data storage device 204. Host systems 252 may access memories within the storage system 250 via a bus interface. In one embodiment, the bus interface may be a Non-Volatile Memory Express (NVMe) or Fibre 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.

[0044] 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. Also, "means" for performing a function can be implemented with at least any of the structure noted herein for the controller and can be pure hardware or a combination of hardware and computer-readable program code.

[0045] 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.

[0046] 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, serially attached small computer system interface (SAS), Fibre Channel, universal serial bus (USB), PCIe, and NVMe. The host interface 120 typically facilitates transfer for data, control signals, and timing signals.

[0047] Back-end module 110 includes an error correction code (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.

[0048] The data storage device 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.

[0049] 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 ReRAM, MRAM, PCM, 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.

[0050] Returning again to Figure 2A, the flash control layer 132 (which will be referred to herein as the flash translation layer (FTL) or, more generally, the "media management layer," as the memory may not be flash) handles flash errors and interfaces with the host. In particular, the FTL, which may be an algorithm in firmware, is responsible for the internals of memory management and translates writes from the host into writes to the memory 104. The FTL may be needed because the memory 104 may have limited endurance, may be written in only multiples of pages, and/or may not be written unless it is erased as a block. The FTL understands these potential limitations of the memory 104, which may not be visible to the host. Accordingly, the FTL attempts to translate the writes from host into writes into the memory 104.

[0051] The FTL may include a logical-to-physical address (L2P) map (sometimes referred to herein as a table or data structure) and allotted cache memory. In this way, the FTL translates logical block addresses ("LBAs") from the host to physical addresses in the memory 104. The FTL can include other features, such as, but not limited to, power-off recovery (so that the data structures of the FTL can be recovered in the event of a sudden power loss) and wear leveling (so that the wear across memory blocks is even to prevent certain blocks from excessive wear, which would result in a greater chance of failure).

[0052] Turning again to the drawings, Figure 3 is a block diagram of a host 300 and data storage device 100 of an embodiment. The host 300 can take any suitable form, including, but not limited to, a computer, a mobile phone, a tablet, a wearable device, a digital video recorder, an autonomous vehicle, a medical robot etc. The host 300 in this embodiment (here, a computing device) comprises a processor 330 and a memory 340. In one embodiment, computer-readable program code stored in the host memory 340 configures the host processor 330 (which can be one or more processors) to perform the acts described herein. So, actions performed by the host 300 are sometimes referred to herein as being performed by an application (computer-readable program code) run on the host 300. For example, the host 300 can be configured to send data

(e.g., initially stored in the host's memory 340) to the data storage device 100 for storage in the data storage device's memory 104.

[0053] As mentioned above, advancements in autonomous vehicles and robotics can demand real-time (or near real-time), low-latency data processing. In such environments, it may be expected that the data storage device will provide the data to the host (e.g., car, surgical robot) in a predictable and time-sensitive (e.g., time-critical) manner.

[0054] For example, automated robots are being explored in a variety of fields, where the robot would perform any operation autonomously. One such example is the surgical robotic systems that can classify autonomy for medical robots on a scale from 0 to 5, with 0 corresponding to no autonomy, with the surgeon remaining in full control, and 5 to a system fully capable of performing entire surgeries with no human input.

[0055] As can be seen from these examples, some environments may require that images/data be read from the data storage device in real-time (or near real-time) without delay. Any latencies induced by the data storage device may be critical for the proper functioning of the system. However, a memory (e.g., NAND storage media) of a data storage device can encounter errors due to several reasons. Data storage devices can often correct these errors by various built-in error-correction mechanisms, such as low-density parity check (LDPC) and exclusive-or (XOR) operations, but these correction schemes can induce some latencies and unpredictability in the response of the data storage device. These latencies can be critical for the environments mentioned above. In some situations, even a few microseconds/milliseconds of delay in reading the data from the data storage device can be of the utmost importance and impact basic functionality of the system.

[0056] The following embodiments can be used to avoid these data storage device latencies. In one embodiment, before reading data from the data storage device 100, the host 300 indicates to the data storage device 100 the highly time-critical logical block address (LBA) range, and the controller 102 in the data storage device 100 takes action to ensure that the host-indicated time-critical LBA range can be read in a fast and a predictable manner.

[0057] More specifically, these embodiments can be used to ensure that the error correction mechanism of the data storage device 100 does not get invoked while reading time-critical data. As mentioned above, it is common to have errors in the memory 104 for several reasons, and the data storage device 100 can have a correction mechanism to correct those errors. However, those correction mechanisms take time to correct the data, which can add unpredictability to the device response time, which can be critical in some situations. So, in one embodiment, error correction is not performed when reading time-critical data. For example, before reading the actual data, the host 300 can indicate to the data storage device 100 that it is going to later require a given LBA

range to be read in a time-sensitive manner. On this hint, the controller 102 of the data storage device 100 can scan the LBA range in the memory 104 and respond back to the host 300 with either an acknowledgment or a message that the data storage device 100 may potentially induce time delays with the data is eventually read. The affirmative result of this interaction can be valid for a timeframe, and the host 300 may need to re-trigger this mechanism before repeating time-sensitive operations.

[0058] In one example implementation, before performing a time-sensitive operation, the host 300 (e.g., a robot/self-operating device) can initiate a communication with the data storage device 100 asking whether or not given data can later be read in a predictable, time-sensitive manner. During such communication, the host 300 can also provide the data storage device with a time frame for preparing the data. The host-device communication can be achieved in multiple ways, and one such way is illustrated in Figure 4.

[0059] As shown in Figure 4, the host 300 (e.g., a robot/self-operating device) identifies time-sensitive data that it may read in the future and provides the data storage device 100 with an indication of the LBA range for that data (act 400). This indication is not an actual read command for the data, as the actual read command will come later. In response, the controller 102 of the data storage device 100 can scan through the LBA range and make sure that the LBA range will not induce any delays due to memory errors. This can involve the controller 102 taking corrective action on the data stored in the LBA range to give faster/predictable access to the LBA range when the host 300 eventually reads the data (act 405). For example, the controller 102 can perform error correction, which can be achieved through moving data to a different block. After attempting such data movement, the indicated LBA range can be scanned to ensure the new location is able to provide data with faster/predictable manner.

[0060] The controller 102 then responds to the host 300 that the LBA range is ready to be read (act 410). Next, the host 300 sends read command(s) to the data storage device 100 to read the data (act 420), and the controller 102 in the data storage device 100 responds by returning the data (act 430). Again, because the data in the LBA has already been error corrected, the data can be read from the data storage device 100 in a relatively-fast and predictable manner without unpredictable delays.

[0061] As illustrated in Figure 5, there may be situations where the memory 104 is so worn out (e.g., due to a high number of program/erase cycles of the memory 104) that the data storage device 100 cannot satisfy the host's request. So, after receiving the host's indication (act 500) and performing the memory scan and attempting data movement to a different block (act 505), if the new data location also exhibits higher BER, the data storage device 100 can respond back negatively to the host 300 indicating that reading the given LBA range could have some

unpredictable delays (act 510). In this situation, the error correction can be done at run-time during the operation of the read command, which can induce delays. The host 300 can then decide either to continue or to replace the data storage device 100. The host 300 can repeat the above procedures before performing any time-sensitive operation.

[0062] As mentioned above, the controller 102 of the data storage device 100 can perform a bit error rate (BER) scan on the indicated LBA range of the memory 104. If the BER is above a threshold, the controller 102 can trigger an error-correction mechanism, which would induce delays. In one embodiment, the controller 102 is provided with one or both of the following mechanisms – one to ensure memory errors are not encountered and the other to reduce the amount of data to be toggled out of the memory 104.

[0063] Starting first with the mechanism to reduce errors, errors get induced in the memory 104 over time or with repetitive memory operations. During the error-detection phase, the controller 102 can identify the BER of time-sensitive data and use the error-correction mechanism to correct the errors. Depending on the error severity (e.g., if the BER is above a certain threshold), certain correction mechanisms (e.g., LDPC operations with longer decode times, BER estimation scan (BES) operations, soft-bit reads, etc.) may be required. The additional time to correct such errors could range from tens of microseconds to milliseconds. So, it may be desired to reduce the BER below the threshold, so such time-consuming error-correction mechanisms may not need to be used.

[0064] To reduce the BER, the controller 102 can move data to a new block (the new block probably does not have the same error-causing issues as the original block) (see Figure 6), move the data from multi-level cell (MLC) blocks to single-level cell (SLC) blocks (which typically have fewer error-causing issues), and/or ensure that certain healthy memory blocks are reserved for time-sensitive data. After refreshing/moving the data, the controller 102 can check the BER of the data to ensure it is under the threshold. If the BER is still above the threshold (or not low enough under the threshold), the controller 102 can repeat the above process.

[0065] Turning now to the method to reduce data-transfer time, for smaller read operations in low queue-depth scenarios, direct memory access (DMA) operations can come into the foreground. In one embodiment, the controller 102 can improve or optimize the performance of these operations by reducing the amount of data to be toggled out through DMA operations. This is illustrated in Figure 7. As shown in Figure 7, error correction code (ECC) parity size can range up to 10% of the total data written. While reading data, this parity can be toggled out alongside host data, and an ECC engine can operate on this parity to find and correct the errors. In one embodiment, the host 300 passes the LBA range for which it wants to reduce the DMA transfer overheads (e.g., the highest read latency critical data), and the controller 102 of the data storage

device 100 reduces the parity size for the LBA range (e.g., by reducing the parity to a minimal checksum or by reducing LDPC parity). This LBA range can be stored into the healthiest block of the memory 105 as a faster copy, with the original (full parity) version retained in the memory 104. When the host 300 later reads the LBA range, the controller 102 can use a data structure that associates a physical address of the copy of the data with the LBA range, so that the copy of the data is returned instead of the original data. This can reduce read latency.

[0066] There are several alternatives that can be used with these embodiments. For example, in response to receiving a signal in advance from the host 300 of the LBA range that will be read, the controller 102 of the data storage device 100 can mark the LBA range for a speculative read to reduce future latency. Also, in another alternative, elements of the NVMe protocol can be used. For example, the dataset management context attributes feature can be used to mark an LBA range for an upcoming read and request low latency on that read. Also, the get LBA status (rebuild assist) feature can be used to mark LBA ranges in order to check for potential errors.

[0067] 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 of EEPROM), ferroelectric random access memory ("FRAM"), and magnetoresistive random access memory ("MRAM"), 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.

[0068] 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 ReRAM 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.

[0069] 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. A NAND 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.

[0070] 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.

[0071] 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.

[0072] 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 wordlines.

[0073] 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 the y direction is substantially perpendicular and the x and z directions are substantially parallel to the major surface of the substrate).

[0074] 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.

[0075] 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.

[0076] 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.

[0077] 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.

[0078] 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.

[0079] 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.

[0080] 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.

What is claimed is:

1. A data storage device comprising:

a memory; and

a controller configured to communicate with the memory and further configured to:

receive, from a host, an indication of a logical block address range that the host will later read; and

in response to receiving the indication:

read data from the logical block address range; and

perform an action on the data to reduce a read latency when the host later reads the logical block address range.

2. The data storage device of Claim 1, wherein:

the controller is further configured to determine whether the read latency is above a threshold when reading the data from the logical block address range; and

the action is performed in response to the read latency being above the threshold.

3. The data storage device of Claim 2, wherein the controller is further configured to, after the action is performed, confirm that the read latency is not above the threshold.

4. The data storage device of Claim 3, wherein the controller is further configured to inform the host in response to the read latency still being above the threshold after the action is performed.

5. The data storage device of Claim 3, wherein the threshold is provided by the host.

6. The data storage device of Claim 1, wherein the action comprises performing an error correction operation on the data.

7. The data storage device of Claim 1, wherein the action comprises moving the data to a different block in the memory.

8. The data storage device of Claim 1, wherein the action comprises moving the data from a multi-level cell (MLC) block to a single-level cell (SLC) block.

9. The data storage device of Claim 1, wherein the action comprises moving the data to an area of the memory reserved for time-sensitive data.

10. The data storage device of Claim 1, wherein the host comprises an automated vehicle.

11. The data storage device of Claim 1, wherein the host comprises a robot.

12. The data storage device of Claim 1, wherein the memory comprises a three-dimensional memory.

13. A method comprising;

performing in a data storage device comprising a memory:

receiving, from a host, an indication of a logical block address range that the host will later read;

in response to receiving the indication:

reading original data from the logical block address range, wherein the original data is stored in the memory with a set of parity bits; and

storing a copy of the data in the memory with fewer, if any, parity bits than what is stored with the original data; and

in response to receiving a command from the host to read the logical block address range, reading the copy of the data instead of the original data, wherein because the copy of the data is stored with fewer, if any, parity bits than what is stored with the original data, reading the copy of the data instead of the original data reduces read latency.

14. The method of Claim 13, wherein the data is time-sensitive data.

15. The method of Claim 13, wherein the copy of the data is stored in a single-level cell (SLC) memory.

16. The method of Claim 13, wherein the copy of the data is stored in an area of the memory reserved for time-sensitive data.

17. The method of Claim 13, wherein the host comprises an automated vehicle.

18. The method of Claim 13, wherein the host comprises a robot.

19. The method of Claim 13, wherein the memory comprises a three-dimensional memory.

20. A data storage device comprising:

a memory;

means for receiving, from a host, an indication of an area of the memory that the host will later read; and

means for, in response to receiving the indication:

reading data from the area of the memory; and

performing an action on the data to reduce a read latency when the host later reads the area of the memory.

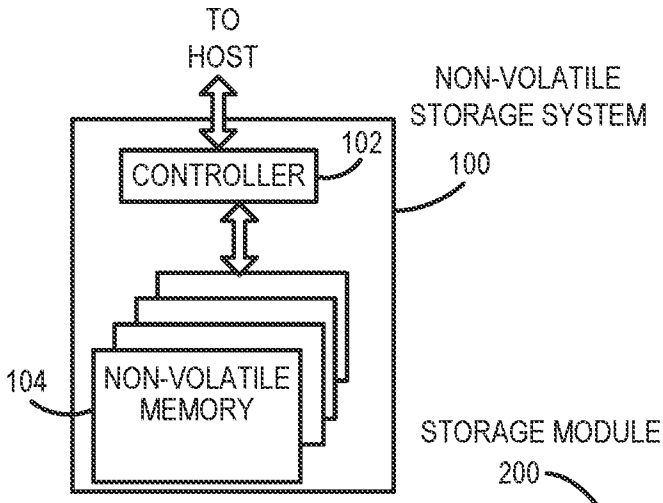

FIG. 1A

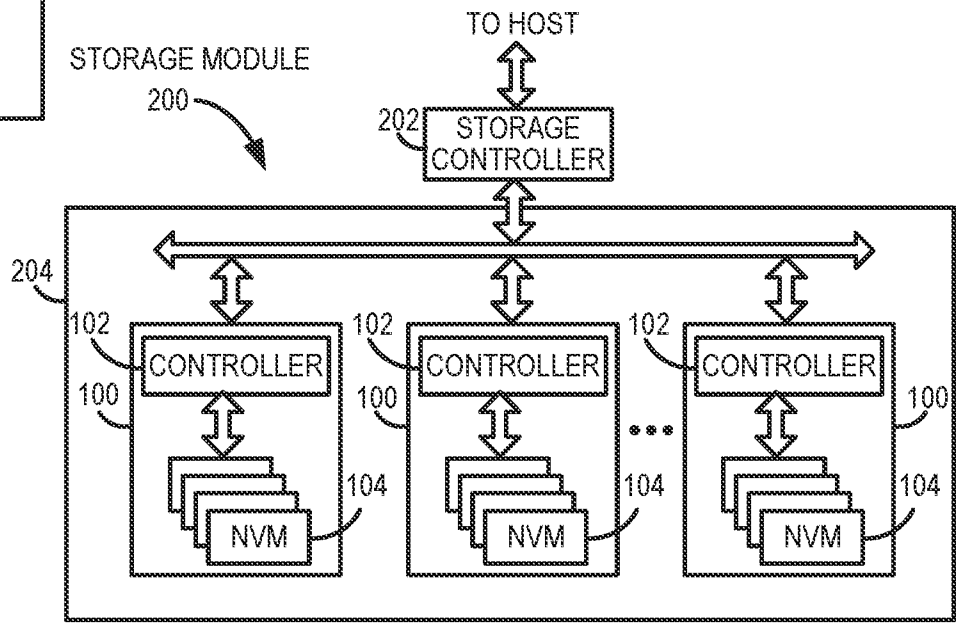

FIG. 1B

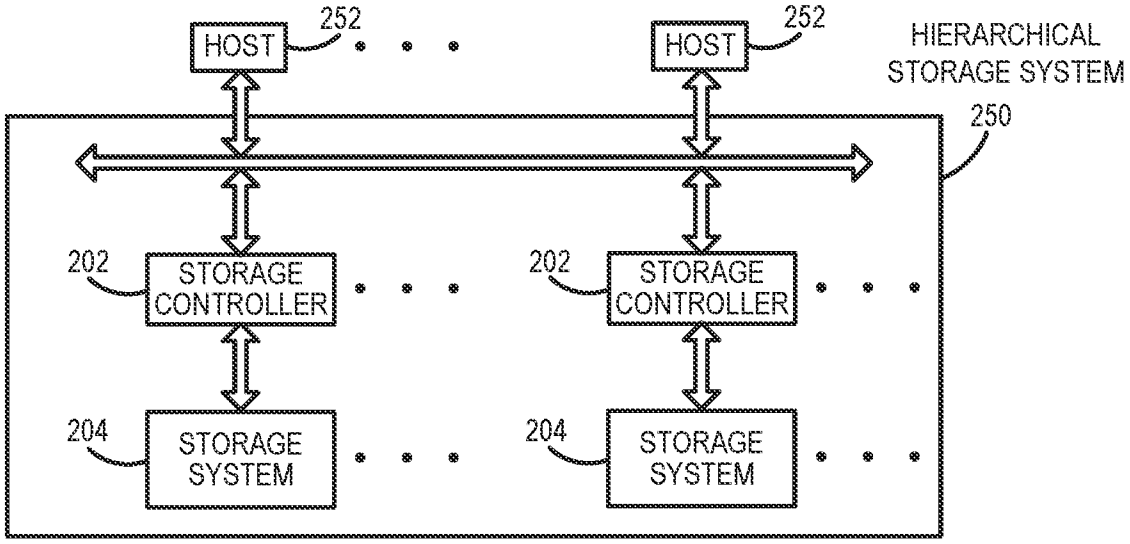

FIG. 1C

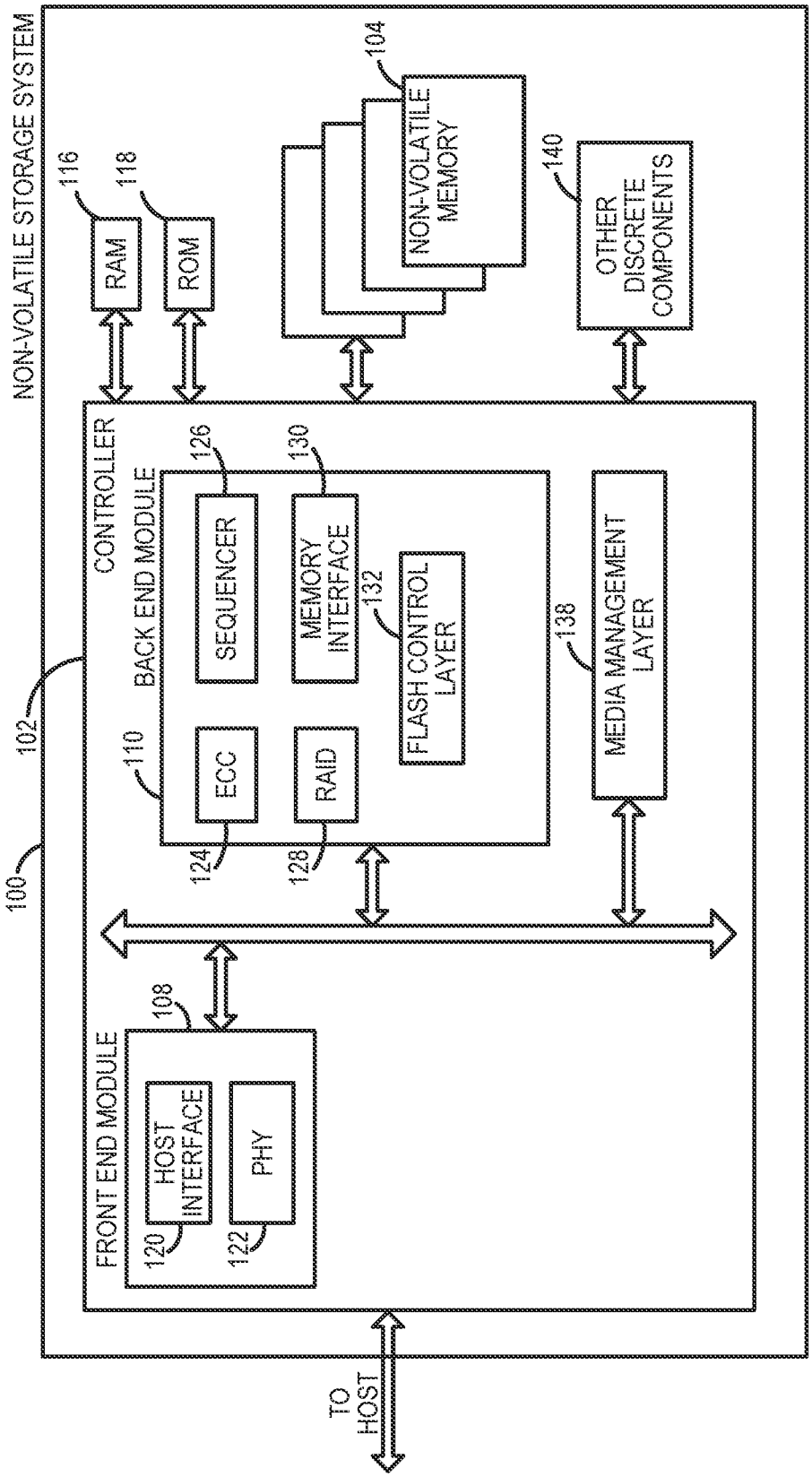
NON-VOLATILE STORAGE SYSTEM
100
102
RAM
116
ROM
118
NON-VOLATILE MEMORY
104
OTHER DISCRETE COMPONENTS
140
CONTROLLER
BACK END MODULE
110
SEQUENCER
126
MEMORY INTERFACE
130
FLASH CONTROL LAYER
132
ECC
124
RAID
128
MEDIA MANAGEMENT LAYER
138
FRONT END MODULE
108
HOST INTERFACE
120
PHY
122
TO HOST

FIG. 2A

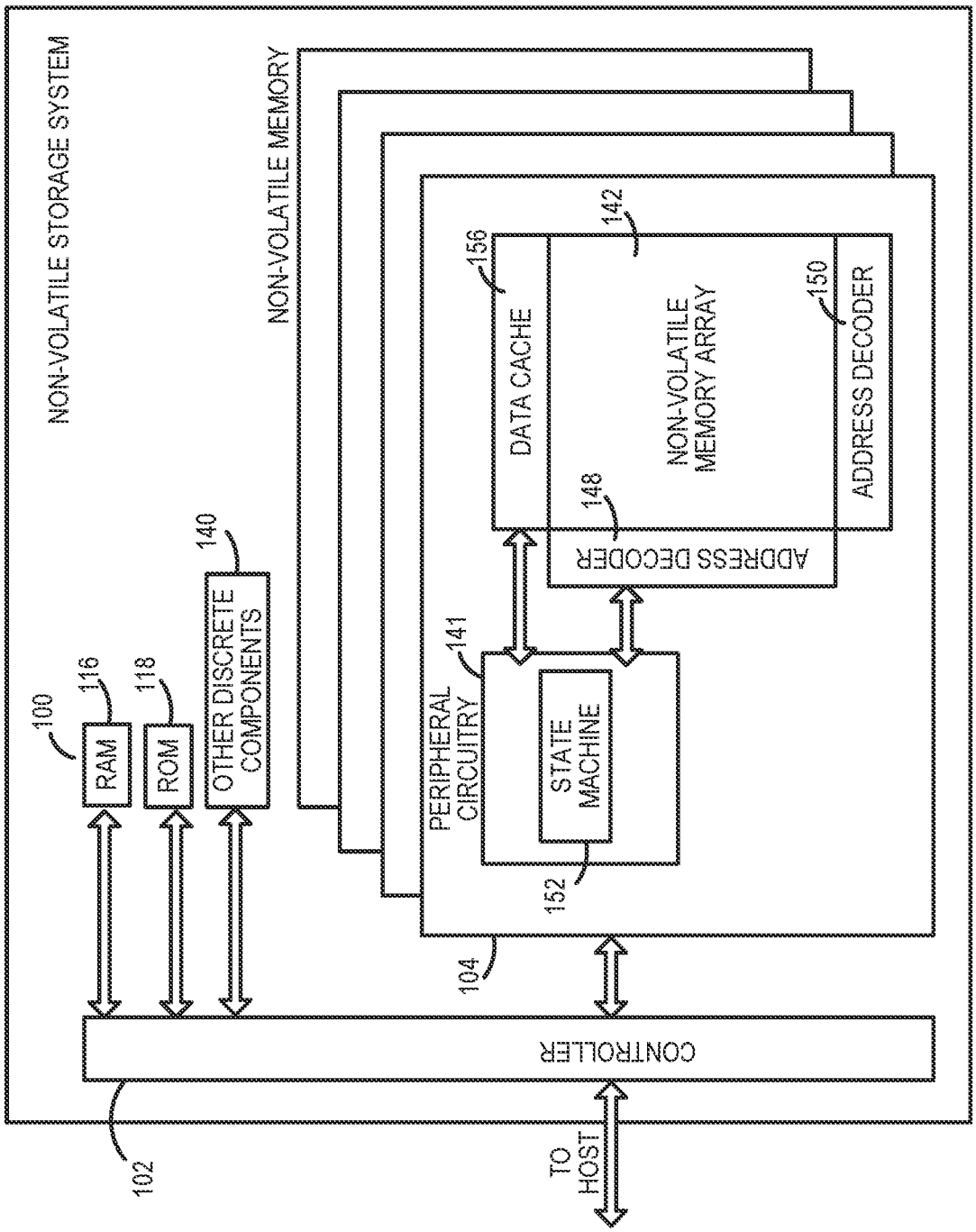
NON-VOLATILE STORAGE SYSTEM
NON-VOLATILE MEMORY
100
RAM
116
ROM
118
OTHER DISCRETE COMPONENTS
140
PERIPHERAL CIRCUITRY
141
STATE MACHINE
152
DATA CACHE
156
NON-VOLATILE MEMORY ARRAY
142
ADDRESS DECODER
148
ADDRESS DECODER
150
104
CONTROLLER
102
TO HOST

FIG. 2B

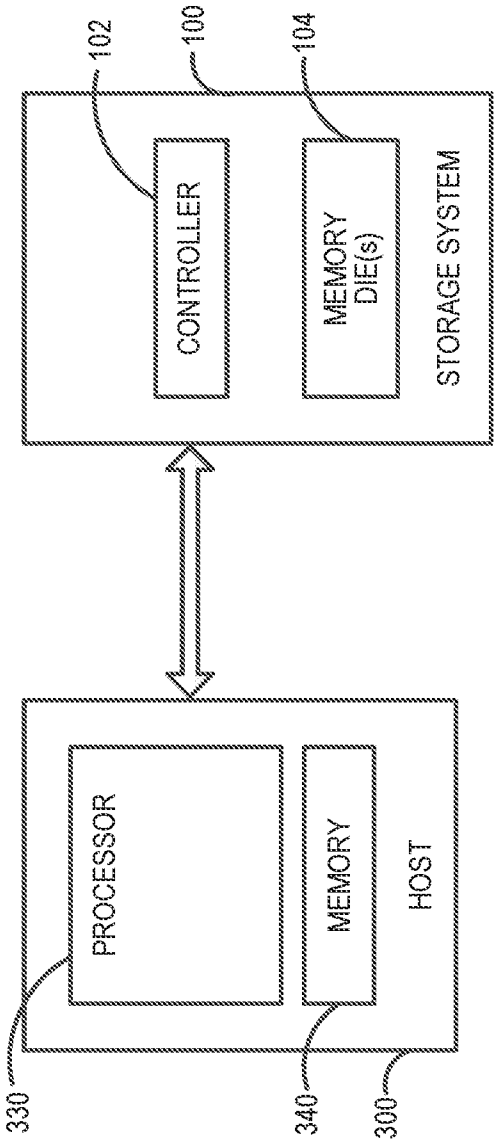
102
100
104
CONTROLLER
MEMORY DIE(s)
STORAGE SYSTEM
PROCESSOR
MEMORY
HOST
330
340
300

FIG. 3

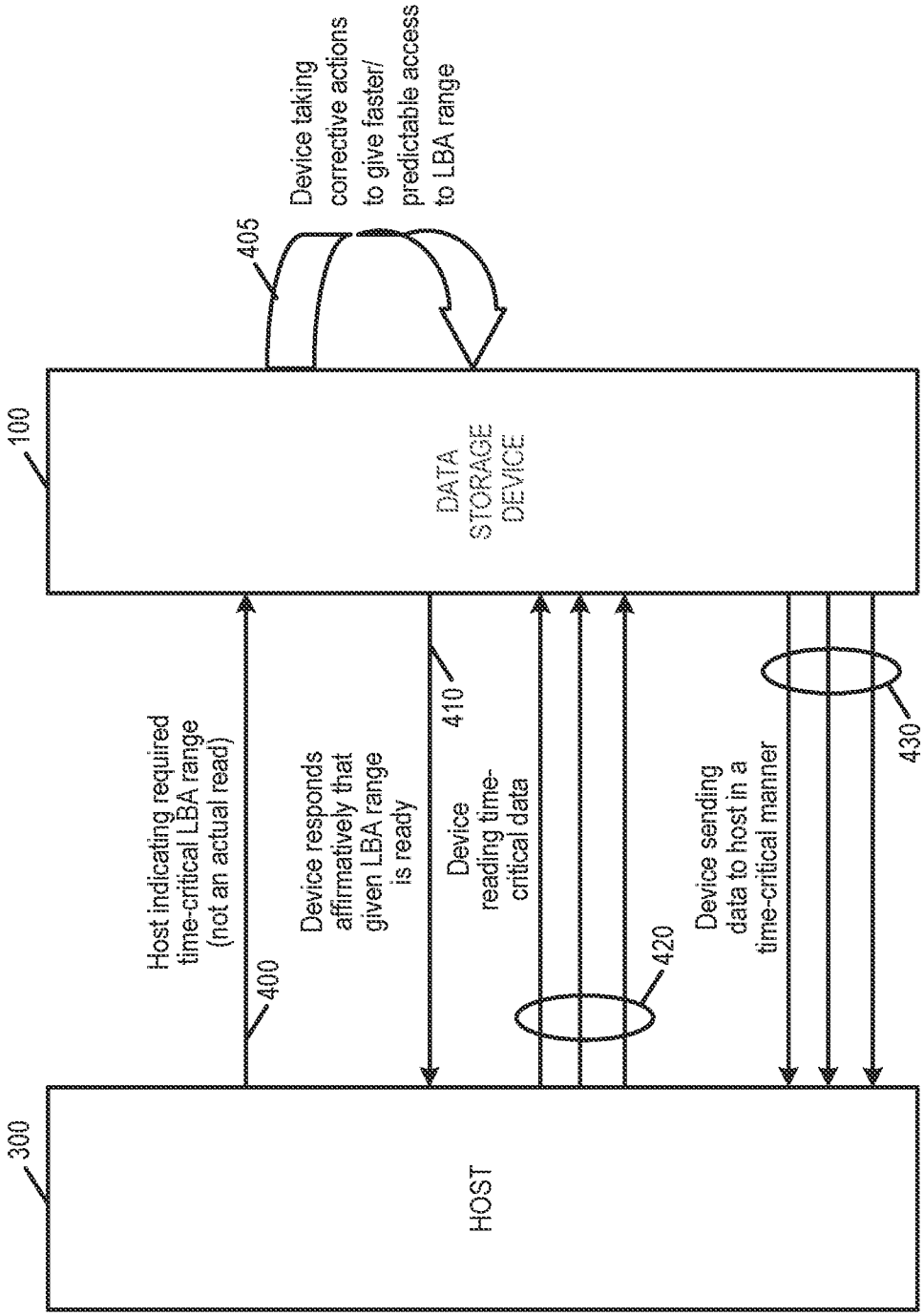
Device taking corrective actions to give faster/ predictable access to LBA range
405
100
DATA STORAGE DEVICE
Host indicating required time-critical LBA range (not an actual read)
400
Device responds affirmatively that given LBA range is ready
410
Device reading time-critical data
420
Device sending data to host in a time-critical manner
430
300
HOST

FIG. 4

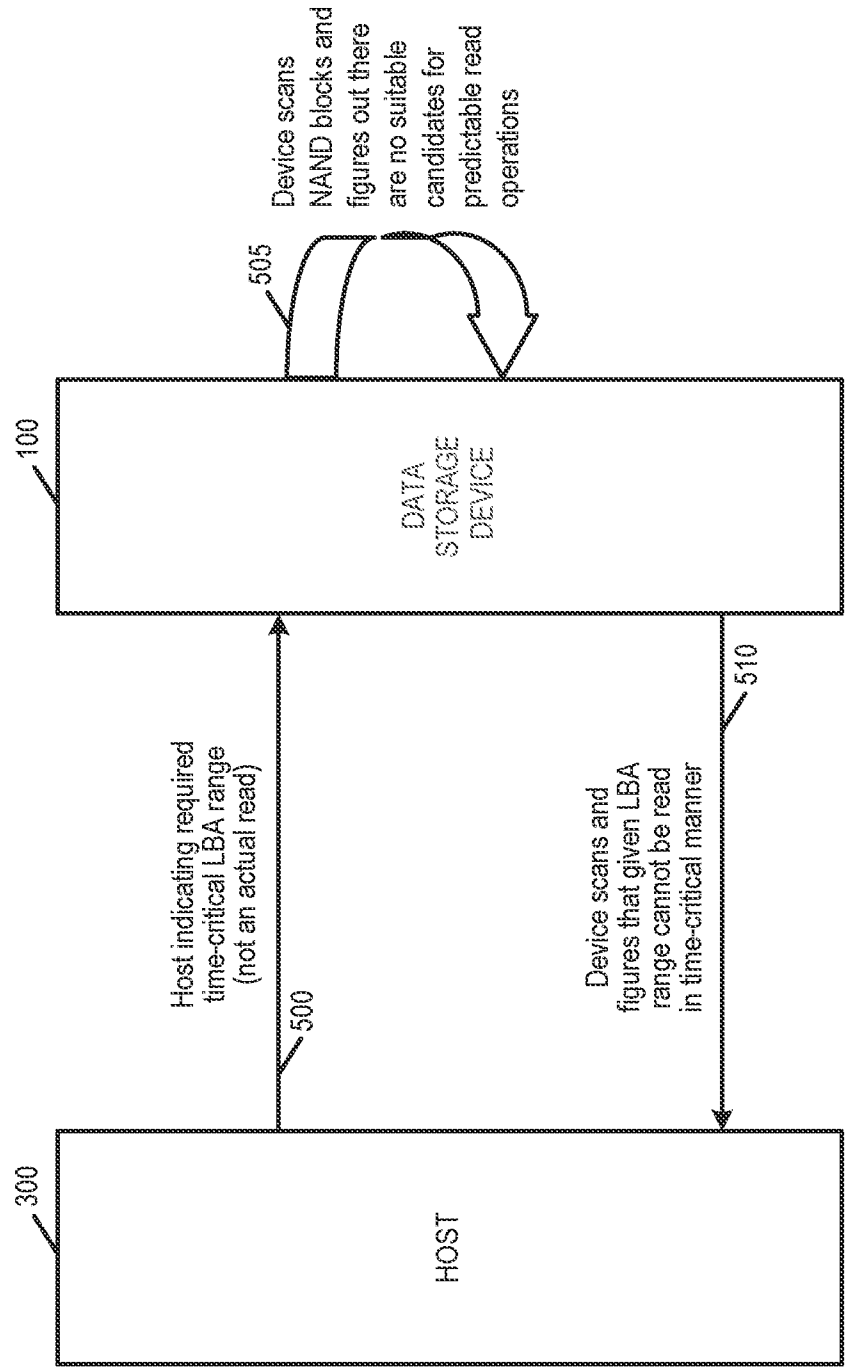
Device scans NAND blocks and figures out there are no suitable candidates for predictable read operations
505
100
DATA STORAGE DEVICE
Host indicating required time-critical LBA range (not an actual read)
500
Device scans and figures that given LBA range cannot be read in time-critical manner
510
300
HOST

FIG. 5

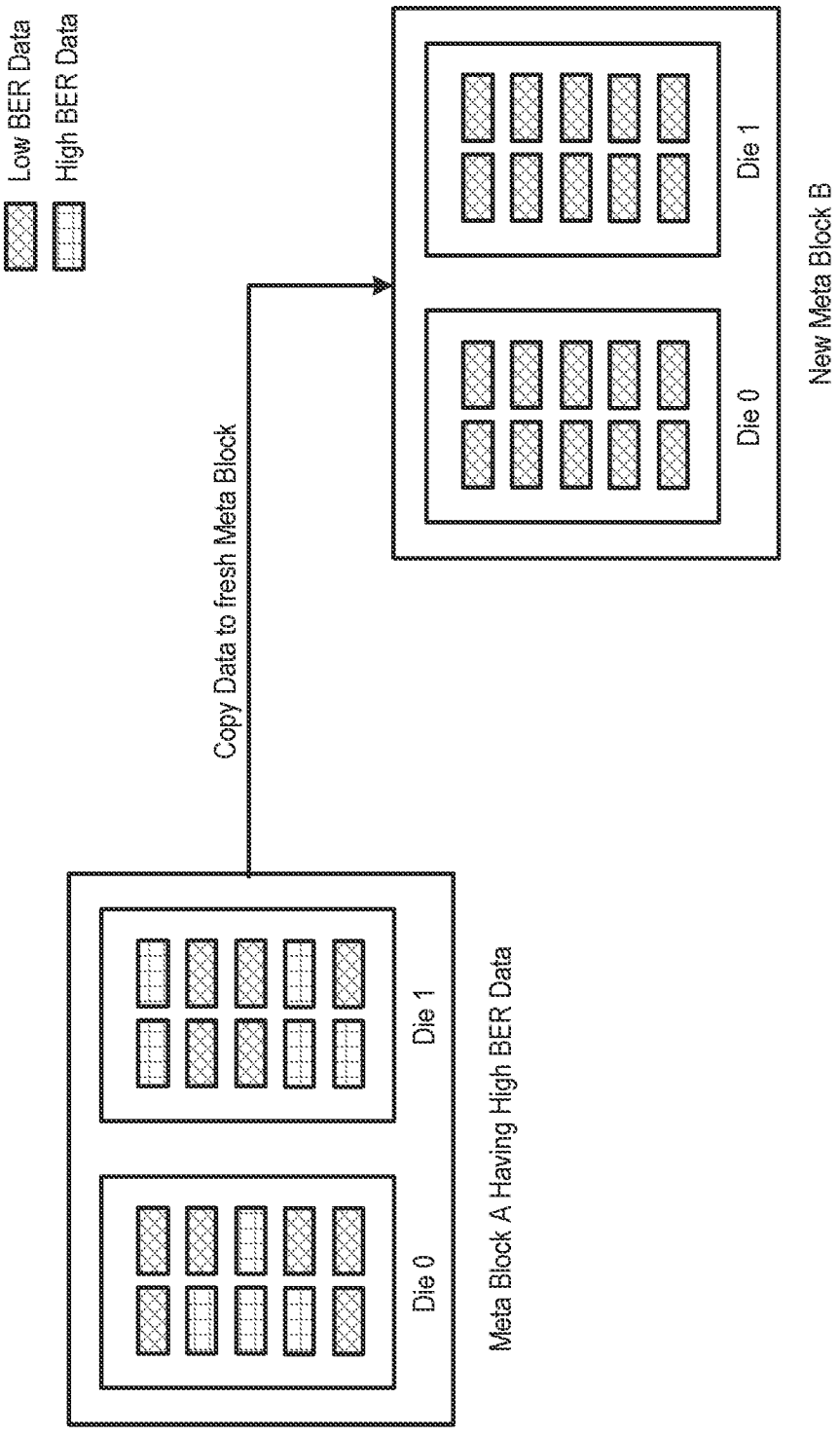
Low BER Data
High BER Data
Copy Data to fresh Meta Block
Meta Block A Having High BER Data
Die 0
Die 1
New Meta Block B
Die 0
Die 1

FIG. 6

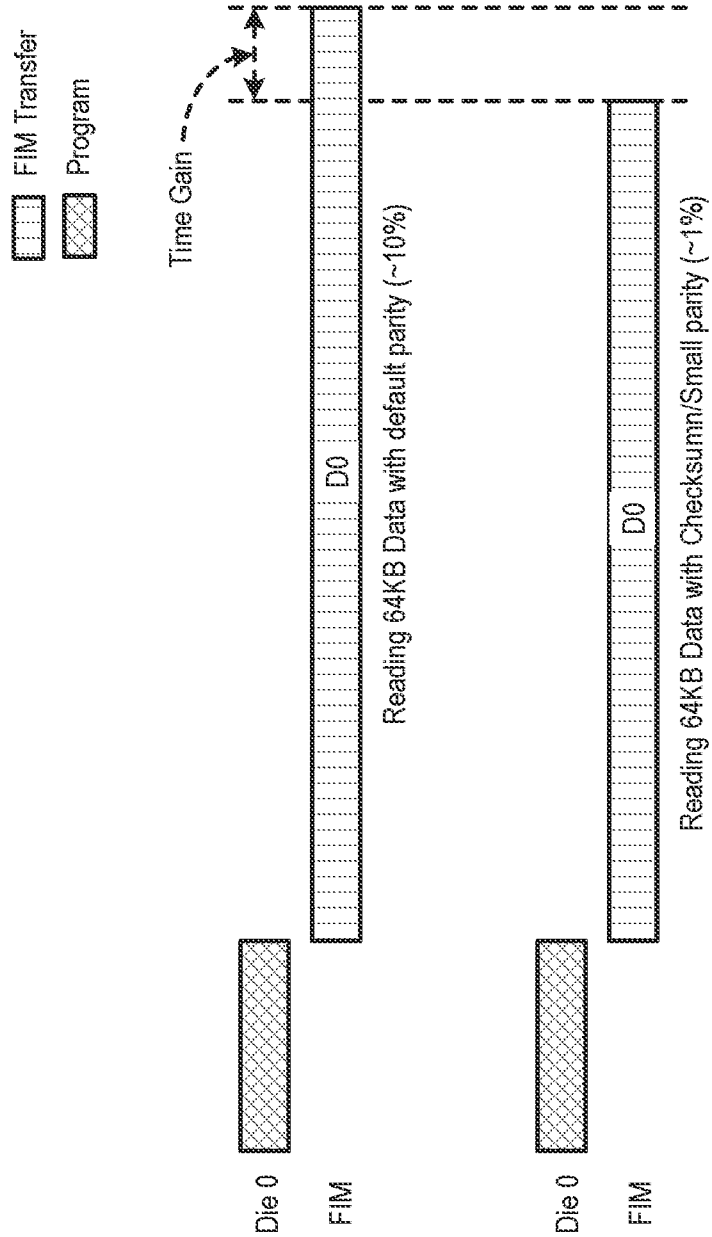
FIM Transfer
Program
Time Gain
Die 0
FIM
D0
Reading 64KB Data with default parity (~10%)
Die 0
FIM
D0
Reading 64KB Data with Checksumn/Small parity (~1%)

FIG. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2024/011238

A. CLASSIFICATION OF SUBJECT MATTER

G06F 3/06(2006.01)i; **G06F 12/06**(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(2006.01); G06F 11/10(2006.01); G06F 12/00(2006.01); G06F 12/02(2006.01); G06F 12/0862(2016.01); G06F 12/10(2006.01); G06F 17/30(2006.01); G11C 29/00(2006.01)

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: storage, data, logical block address range, host, later, read, latency

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
	US 2020-0301841 A1 (MICRON TECHNOLOGY, INC.) 24 September 2020 (2020-09-24)	
X	paragraphs [0046]-[0047], [0073], [0075], [0085]; and claims 1, 4, 7	1,20
Y		6-12
A		2-5,13-19
	US 9208018 B1 (PMC-SIERRA, INC.) 08 December 2015 (2015-12-08)	
Y	claim 1	6-12
	US 2008-0275928 A1 (GARY STEPHEN SHUSTER) 06 November 2008 (2008-11-06)	
A	paragraphs [0042]-[0056]; claims 13-19; and figure 2	1-20
	US 2014-0082261 A1 (LSI CORPORATION) 20 March 2014 (2014-03-20)	
A	paragraphs [0350]-[0477]; claims 1-3; and figure 11	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
"D" document cited by the applicant in the international application
"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
02 May 2024	**02 May 2024**

Name and mailing address of the ISA/KR	Authorized officer
Korean Intellectual Property Office **189 Cheongsa-ro, Seo-gu, Daejeon** **35208, Republic of Korea**	**YANG, JEONG ROK**
Facsimile No. **+82-42-481-8578**	Telephone No. **+82-42-481-5709**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2024/011238

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2019-0243908 A1 (ELECTRONICS AND TELECOMMUNICATIONS RESEARCH INSTITUTE) 08 August 2019 (2019-08-08) paragraphs [0072]-[0095]; claims 1-9; and figures 2-4	1-20

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/US2024/011238

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
US 2020-0301841 A1	24 September 2020	CN 111868679 A	30 October 2020
		EP 3769202 A1	27 January 2021
		JP 2021-516402 A	01 July 2021
		JP 7038227 B2	17 March 2022
		KR 10-2443593 B1	16 September 2022
		US 10705963 B2	07 July 2020
		US 11327892 B2	10 May 2022
		US 2019-0294547 A1	26 September 2019
		WO 2019-182931 A1	26 September 2019
US 9208018 B1	08 December 2015	None	
US 2008-0275928 A1	06 November 2008	US 2011-0238912 A1	29 September 2011
		US 2015-0149719 A1	28 May 2015
		US 2016-0019115 A1	21 January 2016
		US 7958303 B2	07 June 2011
		US 8819365 B2	26 August 2014
		US 9405627 B2	02 August 2016
		US 9448886 B2	20 September 2016
US 2014-0082261 A1	20 March 2014	CN 104375956 A	25 February 2015
		CN 104679446 A	03 June 2015
		EP 2646922 A2	09 October 2013
		EP 2666091 A2	27 November 2013
		EP 2750018 A2	02 July 2014
		EP 2757480 A1	23 July 2014
		EP 2764516 A1	13 August 2014
		JP 2015-036981 A	23 February 2015
		JP 2015-036982 A	23 February 2015
		JP 6285709 B2	28 February 2018
		JP 6443794 B2	26 December 2018
		KR 10-1564569 B1	03 November 2015
		KR 10-2307382 B1	29 September 2021
		KR 10-2328612 B1	17 November 2021
		US 10613929 B2	07 April 2020
		US 10740011 B2	11 August 2020
		US 11144389 B2	12 October 2021
		US 11379301 B2	05 July 2022
		US 11449252 B2	20 September 2022
		US 2020-0019463 A1	16 January 2020
		US 2020-0272537 A1	27 August 2020
		US 2020-0333969 A1	22 October 2020
		WO 2013-052562 A1	11 April 2013
		WO 2014-028183 A1	20 February 2014
US 2019-0243908 A1	08 August 2019	KR 10-2551601 B1	06 July 2023