



- (51) **International Patent Classification:**
G06F 11/14 (2006.01) *G06F 12/02* (2006.01)
- (21) **International Application Number:**
PCT/US2015/039754
- (22) **International Filing Date:**
9 July 2015 (09.07.2015)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant:** HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).
- (72) **Inventors:** KALER, Paul; 11445 Compaq Center Dr. W., Houston, Texas 77070 (US). YATES, James K.; 11445 Compaq Center Dr. W., Houston, Texas 77070 (US).
- (74) **Agents:** KWOK, Jonathan T. et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79, Fort Collins, CO 80528 (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, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, 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, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to the identity of the inventor (Rule 4.17(i))
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))

[Continued on next page]

- (54) **Title:** NON-VOLATILE MEMORY DIE DATA MIRRORING

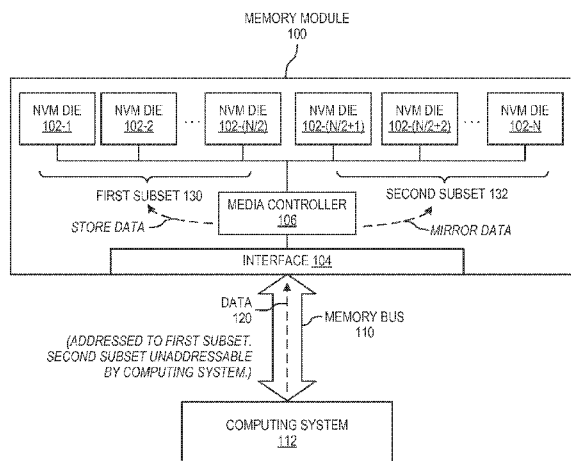


FIG. 1

- (57) **Abstract:** Example implementations relate to a memory module having a plurality of non-volatile memory dies and a media controller to mirror data independent of a computing system. For example, the media controller may receive data from the computing system, may store the data to a die of a first subset of the non-volatile memory dies, and may mirror the data to a die of a second subset of the non-volatile memory dies. The media controller may mirror the data according to a preprogrammed mirroring scheme. The second subset of the non-volatile memory dies may be unaddressable by the computing system.



Published:

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

- 1 -

NON-VOLATILE MEMORY DIE DATA MIRRORING

BACKGROUND

[0001] Memory modules may be installed in a computing system. Memory modules may include volatile or non-volatile memory. Some memory modules include error checking and correction features to mitigate or reduce read errors that may result in fatal system errors and cause the computing system to crash.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Various examples will be described below with reference to the following figures.

[0003] FIG. 1 is a block diagram of an example memory module according to an implementation.

[0004] FIG. 2A is a block diagram of an example mirroring configuration for the example memory module of FIG. 1 according to an implementation.

[0005] FIG. 2B is a block diagram of an example mirroring configuration for the example memory module of FIG. 1 according to an implementation.

[0006] FIG. 3 is a block diagram of an example memory module according to an implementation.

[0007] FIG. 4 is a flowchart of an example method for mirroring data to a non-volatile memory die.

[0008] FIG. 5 is a flowchart of an example method for receiving configuration instructions at a memory module.

[0009] FIG. 6 is a block diagram of an example memory module that includes a non-transitory, machine-readable medium encoded with instructions to mirror data.

[0010] FIG. 7 is a block diagram of an example memory module that includes a non-transitory, machine-readable medium encoded with instructions to receive configuration instructions.

- 2 -

DETAILED DESCRIPTION

[0011] A computing system may include memory for holding data and instructions. For example, such memory may be dynamic random-access memory (DRAM) in a dual in-line memory module (DIMM) form factor. Some data may be affected by memory errors, and such errors may be the result of defective or failing memory, overheating, power supply problems, or other causes. Reading errors may result in fatal computing system errors and may cause the computing system to crash. Some memory, including DRAM DIMMs, may implement error checking or correction functionality, such as error-correcting code and double device data correction.

[0012] Some memory may implement non-volatile memory dies on a memory module. Non-volatile memory (NVM) may store data persistently for retrieval even after power to the non-volatile memory has been cycled, whereas by contrast, volatile memory (including DRAM) maintains its data only while it is powered. Owing to the persistent property of non-volatile memory, it may be useful for the non-volatile memory to have a high level of robustness and reliability. For example, it may be desirable for non-volatile memory to mitigate data errors so that a computing system can have continued access to persistent data stored on non-volatile memory. As will be described herein, a non-volatile memory module with inherent or built-in redundancy may provide a high level of robustness and reliability.

[0013] Referring now to the figures, FIG. 1 is a block diagram of a memory module 100 according to an example implementation. The memory module 100 may include a plurality of non-volatile memory dies (e.g., 102-1 through 102-N; also referred to as NVM dies), an interface 104, and a media controller 106. In some implementations, the interface 104 may be for communicating with a memory bus 110 of a computing system 112. For example, the interface 104 may be compliant with a specification, such as a Double Data Rate (DDR) specification (e.g., DDR, DDR2, DDR3, DDR4, etc.) or other specification and, the form factor of the memory module 100 may be a single inline memory module (SIMM), dual inline memory module (DIMM), or other

- 3 -

form factor. In some implementations, the memory module 100 may be or form part of a system memory of the computing system 112. For example, the memory module 100 may be installed in the computing system 112. In some implementations, the computing system 112 may be a server, a desktop computer, a desktop computer, a workstation, a laptop computer, or the like. In some implementations, the memory module 100 may be enclosed in a chassis of the computing system 112.

[0014] In some implementations, the NVM dies 102-1 through 102-N may include flash memory (e.g., NAND type or NOR type). In some implementations, flash-type NVM dies may be coupled to dynamic random access memory (DRAM) also residing on the same memory module 100. In some implementations, the NVM dies 102-1 through 102-N may include phase change memory. In some implementations, the NVM dies 102-1 through 102-N may include spin-transfer torque memory. In some implementations, the NVM dies 102-1 through 102-N may include resistive random-access memory (also referred to as memristor memory). Other types of persistent or non-volatile memory may be implemented for the NVM dies 102-1 through 102-N. A total capacity of the memory module 100 (e.g., as measured in gigabytes, etc.) may be a function of the quantity of NVM dies and a memory density of the NVM dies included in memory module 100.

[0015] The media controller 106 may include a series of instructions encoded on a machine-readable storage medium and executable by a processor. For example, the processor may be a microprocessor, an application-specific integrated circuit, a field programmable gate array, and/or other hardware device suitable for retrieval and/or execution of instructions. In some implementations, the media controller 106 itself may be or include the processor. Additionally or alternatively, the media controller 106 may include one or more hardware devices including electronic circuitry for implementing functionality described herein.

[0016] The media controller 106 may receive data 120 over the interface 104 addressed to a first subset 130 of the plurality of NVM dies 102-1 through

- 4 -

102-N. For example, a processor or a memory controller of the computing system 112 may address the data 120 to the first subset 130 and may communicate the data to the interface 104 over the memory bus 110. In some implementations, the first subset 130 includes at least one NVM die. In the example illustrated in FIG. 1, the first subset 130 may include NVM dies 102-1, 102-2, through 102-(N/2). The media controller 106 may store the data 120 to a die of the first subset 130.

[0017] The media controller 106 may mirror (i.e., duplicate or copy) the data 120 to a die of a second subset 132 of the plurality of NVM dies 102-1 through 102-N according to a preprogrammed mirroring scheme. In the example illustrated in FIG. 1, the second subset 132 may include NVM dies 102-(N/2+1), 102-(N/2+2), through 102-N. In some implementations, the second subset 132 includes at least one NVM die. In some implementations, the second subset 132 may be unaddressable by the computing system 112, while all NVM dies 102-1 through 102-N are addressable by the media controller 106 (e.g., for mirroring data). For example, for a memory module 100 that is split into two subsets (e.g., the first subset 130 and the second subset 132), the memory module 100 may appear to have half the capacity of the actual total capacity of the NVM dies 102-1 through 102-N included on the memory module 100, from the viewpoint of the computing system 112 (more particularly, from the viewpoint of a basic input/output system (BIOS), a Unified Extensible Firmware Interface (UEFI), a kernel, an operating system, an application, or the like, of the computing system 112). Additionally, the computing system 112 (or more particularly, a processor or a memory controller thereof), may only be able to physically address the NVM dies of the first subset 130.

[0018] As described above, the media controller 106 may mirror the data 120 according to a preprogrammed mirroring scheme. For example, the preprogrammed mirroring scheme may include, for example, a direct mapping of a memory address in the first subset 130 to a memory address in the second subset 132. To illustrate, an address on NVM die 102-1 may be

- 5 -

mapped to an address on NVM die 102-(N/2+1), an address of NVM die 102-2 may be mapped to an address on NVM die 102-(N/2+2), an address on NVM die 102-3 may be mapped to an address on NVM die 102-N, and the like, although different preprogrammed mirroring schemes also may be implemented. In some implementations, the preprogrammed mirroring scheme may be deemed preprogrammed by virtue of being intrinsic to the electronic circuitry (e.g., being hard-wired) or to the encoded instructions (e.g., being hard-coded) included on the media controller 106. In some implementations, the preprogrammed mirroring scheme may be deemed to be a factory default.

[0019] In some implementations, the media controller 106 may mirror data to a number of subsets of the plurality of non-volatile memory dies 102-1 through 102-N. The number of mirror subsets may be an aspect of the preprogrammed mirroring scheme. For example, as described above, data may be mirrored to one subset (namely, the second subset 132) in a dual-mirroring configuration. As another implementation, FIG. 2A illustrates an example of memory module 100 where the data 120 may be stored to a first subset 210 (which includes NVM die 102-1 and 102-2) and mirrored to two different subsets of NVM dies, namely a second subset 212 (including NVM dies 102-3 and 102-4) and a third subset 214 (including NVM dies 102-5 and 102-6). The example of FIG. 2A may be deemed a triple-mirroring configuration. As another implementation, FIG. 2B illustrates an example of memory module 100 where the data 120 may be stored to a first subset 210 (which includes NVM die 102-1 and 102-2) and mirrored to three different subsets of NVM dies, namely a second subset 212 (including NVM dies 102-3 and 102-4), a third subset 214 (including NVM dies 102-5 and 102-6), and a fourth subset 216 (including NVM dies 102-7 and 102-8). The example of FIG. 2B may be deemed a quad-mirroring configuration. It should be understood that the foregoing examples are non-limiting illustrations, and different variations (e.g., number of subsets, number of NVM dies per subset, etc.) of mirroring configurations may be implemented.

- 6 -

[0020] Referring again to FIG. 1, the media controller 106 may mirror the data 120 independent of the computing system 112. For example, the media controller 106 may mirror the data 120 independent of (i.e., without involvement from) components of the computing system 112 such as a processor (e.g., central processing unit), a memory controller, BIOS, UEFI, a kernel, an operating system, or applications. For example, the media controller 106 may mirror the data 120 independent of the computing system 112 by virtue of the preprogrammed mirroring scheme described above. Accordingly, by virtue of the media controller 106 including a preprogrammed mirroring scheme and being capable of mirroring the data 120 independent of the computing system (in addition to the second subset being unaddressable by the computing system 112), the memory module 100 may store data 120 in non-volatile memory with persistence and reliability in a manner that is invisible to the computing system 112, and moreover, may be backwards compatible with a computing system 112 having an older generation architecture.

[0021] In some implementations, the mirroring of data 120 by the media controller 106 may be deemed analogous to a RAID-1 (Redundant Array of Inexpensive Disks) configuration in many respects. In some implementations, if the media controller 106 detects an error or failure of an NVM die in either the first subset 130 or the second subset 132 (e.g., via error checking routines), the media controller 106 may respond to a read request from the computer system 112 by retrieving redundant, mirrored data from the second subset 130 or the first subset 132, respectively. Accordingly, in some instances, the computing system 112 may continue to access data persisting on the memory module 100, thus reducing the probability of fatal errors of the computing system 112. Moreover, by virtue of dividing NVM dies 102-1 through 102-N into the first subset 130 and the second subset 132, the media controller 106 may compensate for a failure of an NVM die in one subset by reading/writing (without redundancy or mirroring) to a corresponding NVM die in the other subset. In some implementations, the memory module 100 may communicate its failure to the computing system 112, even though the

- 7 -

memory module 100 is able to utilize mirrored data and operational NVM dies, and the computing system 112 may offer a user an option to continue or safely shut down for replacement of the memory module 100. In this manner, a user may extend the time to replace the memory module 100 by operating the memory module 100 without mirroring for NVM dies that have failed.

[0022] In some implementations, the media controller 106 may stripe the data 120 across NVM dies of the first subset 130. That is, data 120 may be split up into blocks, and the media controller 106 may write each of the blocks to different NVM dies of the first subset 130. In some implementations, the data 120 is initially transmitted by the computing system 112 to the memory module 100 across the memory bus 110 in regularly-sized chunks (e.g., a 64-bit computing system 112 may transmit 64 bits at a time across the memory bus 110), and the media controller 106 may further divide those regularly-sized chunks of data 120 based on the number of NVM dies in the first subset 130. In some implementations, the data 120 may be divided evenly between the NVM dies in the first subset 130. To illustrate, if there are eight NVM dies in total on the memory module 100 ($N=8$, such that $102-N$ is $102-8$), with four NVM dies in the first subset 130 (e.g., $102-(N/2)$ is $102-4$), the media controller 106 may divide a 64-bit chunk of data 120 into four equal 16-bit blocks and write each of those four 16-bit blocks to a different one of the NVM dies of the first subset 130 (102-1 through 102-4). In some implementations, the striping of data 120 by the media controller 106 in this manner may be deemed analogous to a RAID-0 configuration in many respects. In some implementations, the data 120 as striped across the first subset 130 may also be mirrored to the second subset 132 (in the same striping configuration), in a manner that may be deemed analogous in many respects to a RAID-01 configuration.

[0023] In some implementations, the media controller 106 may generate parity data (or parity checksum) based on data striped across NVM dies of the first subset 130. For example, if data 120 is received in chunks (such as 64-bit chunks) and the first subset 130 includes $(N/2)$ number of NVM dies, the

- 8 -

media controller 106 may stripe data 120 in evenly-divided blocks across $(N/2 - 1)$ number of the NVM dies of the first subset 130 and store the parity data on the remaining NVM die. In some implementations, the media controller 106 may include an XOR engine (e.g., in the form of machine-readable instructions) to calculate, as the parity data, the XOR of the striped blocks of data 120. In some implementations, the NVM die to which parity data is written varies over time (e.g., for different 64-bit chunks of the data 120). In some implementations, generation of parity information with the striped data 120 by the media controller 106 in this manner may be deemed analogous to a RAID-5 configuration in many respects.

[0024] FIG. 3 is a block diagram of a memory module 300 according to an example implementation. Memory module 300 may include NVM dies 302-1 through 302-N, and each of the NVM dies 302-1 through 302-N may include flash memory, phase change memory, spin-transfer torque memory, resistive random-access memory (memristor memory), or other types of persistent or non-volatile memory. As will be described further herein below, the NVM dies 302-1 through 302-N may be logically deemed part of a subset of the plurality of NVM dies, such as a first subset, a second subset, or the like, for the purposes of describing a mirroring functionality. As with the memory module 100, a total capacity of the memory module 300 may be a function of the quantity of the NVM dies 302-1 through 302-N and a memory density of the NVM dies 302-1 through 302-N included in the memory module 300.

[0025] In some implementations, the NVM dies 302-1 through 302-N may be organized into physical NVM die packages (e.g., 303-1 through 303-M). For example, an NVM die package may include one NVM die, two NVM dies, quad NVM dies, or other configurations. The example illustrated in FIG. 3 shows but one implementation of two NVM dies in a package (e.g., NVM die package 303-1 may include two NVM dies 302-1, 302-2).

[0026] The memory module 300 may include an interface 304 for communicating with a memory bus 310 of a computing system 312. The interface 304, memory bus 310, and computing system 312 may be

- 9 -

analogous in many respects to the interface 104, memory bus 110, and computing system 112 of FIG. 1. The memory module 300 also may include a media controller 306. The media controller 306 may be analogous to the media controller 106 of FIG. 1 in many respects. The media controller 306 may include a series of instructions encoded on a machine-readable storage medium and executable by a processor (e.g., a microprocessor, an application-specific integrated circuit, a field programmable gate array, and/or other hardware device suitable for retrieval and/or execution of instructions). Additionally or alternatively, the media controller 306 may include one or more hardware devices including electronic circuitry for implementing functionality described herein. As with media controller 106, the media controller 306 may receive data 320 over the interface 304 addressed to a first subset of the plurality of NVM dies 302-1 through 302-N, may store the data 320 to an NVM die of the first subset, and may mirror the data 320 to a die of a second subset of the plurality of NVM dies 302-1 through 302-N. The media controller 306 may mirror the data 320 according to a preprogrammed mirroring scheme, independent of the computing system 312.

[0027] In some implementations, the NVM die of the first subset to which data 320 is stored and the NVM die of the second subset to which data 320 is mirrored may be in the same NVM package. In other implementations (as is illustrated in FIG. 2), NVM dies of the first subset (e.g., 330) are included in different NVM packages than the NVM dies of the second subset (e.g., 332). Accordingly, the NVM die of the first subset 330 to which the data 320 is stored is included in a different NVM die package than the NVM die of the second subset 332 to which the data 320 is copied. For example, as illustrated in FIG. 3, a first subset 330 may include at least one NVM die package 303-1 through 303-(M/2), which includes the NVM dies 302-1 through 302-(N/2), while the second subset 332 may include at least one NVM die package 303-(M/2+1) through 303-M, which includes NVM dies 302-(N/2+1) through 302-N. To illustrate, data stored into the NVM die 302-1 of the NVM die package 303-1 may be mirrored into the NVM die 302-(N/2+1) of the NVM die package 303-(N/2+1). By virtue of the first subset 330 including

- 10 -

different NVM die packages (and therefore different NVM dies) than the second subset 332, the memory module 300 may provide data redundancy that may be robust to failures of individual NVM die packages.

[0028] In some implementations, the memory module 300, or the media controller 306, may transmit or make available an identification 322 that indicates an architecture of the plurality of non-volatile memory dies 302-1 through 302-N (or more generally, the architecture of the memory module 300). For example, the architecture information may relate to a quantity of the NVM dies 302-1 through 302-N, a capacity of the NVM dies 302-1 through 302-N, a quantity of the NVM die packages 303-1 through 303-M, organization of the NVM dies 302-1 through 302-N into NVM die packages 303-1 through 303-M, or the like. In some implementations, the identification 322 may be a Stock Keeping Unit (SKU) or manufacturer part number of the memory module 300, and a different identification 322 may be assigned to each different architecture of memory module 300 (e.g., different capacities, different arrangements and capacities of NVM dies, etc.). In some implementations, the identification 322 may be encoded in Serial Presence Detect data of the memory module 300.

[0029] In some implementations, the identification 322 may be transmitted by the memory module 300 to a memory manager 314 installed on the computing system 312. The memory manager 314 may be, for example a firmware (such as BIOS or UEFI), a kernel, an operating system, an application, or the like (the foregoing examples may be understood to be instructions stored on a non-transitory machine readable medium and executable by a processor). In some implementations, the memory manager 314 may interpret the identification 322 (e.g., via a lookup table or the like) to determine the architecture of the associated memory module 300, and may present memory module configuration options based on the identification (or the determined architecture) to a user, via a display connected to the computing system 312 for example (not shown). In some implementations, the manufacturer of the memory module 300 may provide an extended

- 11 -

feature pack for BIOS or UEFI or the like to modify or replace an existing memory manager 314 and to enable the computing system 312 to recognize the identification 322 and present the related memory module configuration options. The memory module configuration options may relate to redundancy modes that the media controller 306 may implement within the memory module 300, in addition to the preprogrammed mirroring scheme. For example, such redundancy modes may include mirroring modes, striping modes, or parity modes, which will be described further herein below. In some implementations, the memory module configuration options presented by the memory manager 314 to the user may vary depending on the architecture of the memory module 300. For example, the memory manager 314 may only present a triple-mirroring configuration option if the number of NVM dies 302-1 through 302-N is divisible by three. The memory manager 314 may receive a user selection from among the presented memory module configuration options, and may transmit corresponding configuration instructions over the memory bus 310 to the memory module 300. For example, the configuration instructions may be instructions to the memory module 300 to mirror data, stripe data, or generate parity data on the plurality of NVM dies. The media controller 306 of the memory module 300 may receive the configuration instructions via the interface 304.

[0030] Examples of redundancy modes that the media controller 306 may implement in response to configuration instructions from the memory manager 314 will now be described. In an example redundancy mode, the configuration instructions may instruct the media controller 106 to mirror the data 320 to a number of subsets of the plurality of non-volatile memory dies 302-1 through 302-N, and the number of subsets may be a user-selectable via memory manager 314. For example, in a manner similar to that described above with respect to FIGS. 2A and 2B, the number of subsets may be two for a triple-mirroring configuration, three for a quad-mirroring configuration, and so on. In another example redundancy mode, the configuration instructions may instruct the media controller 306 to stripe the data 320 across NVM dies 302-1 through 302-(N/2) of the first subset 330 in a manner analogous to that

- 12 -

described above with regards to media controller 106 of FIG. 1. Similarly, the media controller 306 may also stripe the data 320 across NVM die packages 303-1 through 303-(M/2) of the first subset 330 in some implementations. In another example redundancy mode, the configuration instructions may instruct the media controller 306 to stripe the data 320 across the NVM dies 302-1 through 302-(N/2) of the first subset 330, and mirror the striped data 320 to the NVM dies 302-(N/2+1) through 302-N of the second subset 332, which may be deemed analogous to a RAID-01 configuration in many respects. In another example redundancy mode, the configuration instructions may instruct the media controller 306 to organize the plurality of NVM dies 302-1 through 302-N into a number of subsets, each subset including a pair of mirrored NVM dies, and to stripe the data 320 across the number of subsets. This foregoing redundancy mode may be deemed analogous to a RAID-10 configuration in many respects. In another example redundancy mode, the configuration instructions may instruct the media controller 306 to generate parity data (e.g., RAID-5) in a manner similar to that described above with respect to the media controller 106 of FIG. 1. It should be understood that some of the foregoing redundancy modes may be combined or nested to provide different levels of data protection and fault tolerance.

[0031] FIG. 4 is a flowchart of an example method 400 for mirroring data to a non-volatile memory die. Method 400 may be described below as being executed or performed by a media controller of a memory module, such as the media controller 106 of the memory module 100 described above with respect to FIG. 1. Various other media controllers may be used as well, such as, for example, the media controller 306. Method 400 may be implemented in the form of executable instructions stored on a machine-readable storage medium and executed by at least one processor of the media controller 106, and/or in the form of electronic circuitry. In some implementations of the present disclosure, one or more blocks of method 400 may be executed substantially concurrently or in a different order than shown in FIG. 4. In some implementations of the present disclosure, method 400 may include

- 13 -

more or less blocks than are shown in FIG. 4. In some implementations, one or more of the blocks of method 400 may, at certain times, be ongoing and/or may repeat.

[0032] The method 400 may begin at block 402, and continue to block 404, where the media controller 106 (of the memory module 100) may receive data from a computing system in which the memory module 100 is installed. At block 406, the media controller 106 may store the data received at block 404 to a die of a first subset of a plurality of non-volatile memory dies of the memory module 100. At block 408, the media controller 106 may mirror the data received at block 404 to a die of a second subset of a plurality of non-volatile memory dies of the memory module 100, according to a mirroring scheme preprogrammed in the media controller 106. The mirroring performed by the media controller 106 at block 408 may be performed independently of (e.g., without utilizing execution cycles of, or without intervention from) the computing system. The second subset may be unaddressable by the computing system. Additionally, the non-volatile memory dies of the memory module may be organized into non-volatile memory die packages, and the die of the first subset and the die of the second subset may be included in different non-volatile memory die packages (see e.g., the memory module 300). At block 410, the method 400 may end.

[0033] FIG. 5 is a flowchart of an example method 500 for receiving configuration instructions at a memory module. Method 500 may be described below as being executed or performed by a media controller of a memory module, such as the media controller 306 of the memory module 300 described above with respect to FIG. 3. Various other media controllers may be used as well, such as, for example, media controller 106. Method 500 may be implemented in the form of executable instructions stored on a machine-readable storage medium and executed by at least one processor of the media controller 306, and/or in the form of electronic circuitry. In some implementations of the present disclosure, one or more blocks of method 500 may be executed substantially concurrently or in a different order than shown

- 14 -

in FIG. 5. In some implementations of the present disclosure, method 500 may include more or less blocks than are shown in FIG. 5. In some implementations, one or more of the blocks of method 500 may, at certain times, be ongoing and/or may repeat.

[0034] The method 500 may begin at block 502, and continue to block 504, where the media controller 306 may receive configuration instructions to mirror data, stripe data, or generate parity data on a plurality of NVM dies of the memory module 300. The configuration instructions may be received by the media controller 306 from a memory manager installed on a computing system (e.g., a computing system where the memory module 300 may be installed). In some implementations, the memory manager may present memory module configuration options to a user based on an identification provided by the memory module 300 that indicates to the memory manager an architecture of the plurality of NVM dies of the memory module 300. At block 506, the media controller 306 may stripe data across NVM dies of a first subset of the plurality of NVM dies. In some implementations, the data may be data that is received at the media controller 306 from the computing system. For example, method 500 may be performed after block 404 of method 400, and the data may be received at block 404. At block 508, the media controller 306 may generate parity data based on data striped across NVM dies of the first subset, such as, for example, the data striped across the NVM dies of the first subset at block 506. In some implementations, the parity data may be generated by applying an XOR operation to the striped data. In some implementations, the parity data may also be stored on the NVM dies of the first subset. At block 510, the method 500 may end.

[0035] FIG. 6 is a block diagram illustrating a memory module 600 that includes a machine-readable medium encoded with instructions to mirror data to non-volatile memory dies according to an example implementation. In some implementations, the memory module 600 may serve as or form part of the memory module 100 of FIG. 1. In some implementations, the memory module 600 may include at least one controller 602 coupled to a machine-

- 15 -

readable medium 604. The controller 602 may include a single-core processor, a multi-core processor, an application-specific integrated circuit, a field programmable gate array, and/or other hardware device suitable for retrieval and/or execution of instructions from the machine-readable medium 604 (e.g., instructions 606, 608, 610) to perform functions related to various examples. Additionally or alternatively, the controller 602 may include electronic circuitry for performing the functionality described herein, including, but not limited to, the functionality of instructions 606, 608, and/or 610. With respect to the executable instructions represented as boxes in FIG. 6, it should be understood that part or all of the executable instructions and/or electronic circuits included within one box may, in alternate implementations, be included in a different box shown in the figures or in a different box not shown. In some implementations, the controller 602 and the machine-readable medium 604 may be or form part of the media controller 106 of FIG. 1. In some implementations, the memory module 600 may include a plurality of NVM dies 605, which may be coupled to the controller 602. In some example implementations, the memory module 600 may be installed in and in communication with a computing system 650.

[0036] The machine-readable medium 604 may be any medium suitable for storing executable instructions, such as random access memory (RAM), electrically erasable programmable read-only memory (EEPROM), flash memory, hard disk drives, optical discs, and the like. In some example implementations, the machine-readable medium 604 may be a tangible, non-transitory medium, where the term “non-transitory” does not encompass transitory propagating signals. As described further herein below, the machine-readable medium 604 may be encoded with a set of executable instructions 606, 608, 610.

[0037] Instructions 606, when executed by the controller 602, may receive data at the memory module 600 from a computing system 650 in which the memory module 600 is installed. Instructions 608, when executed by the controller 602, may store the data received by instructions 606 to a first

- 16 -

subset of the plurality of non-volatile memory dies 605 of the memory module 600. Instructions 610, when executed by the controller 602, may mirror the data received by instructions 606 to a number of subsets of the plurality of non-volatile memory dies 605 that are different from the first subset, according to a preprogrammed mirroring scheme and independently of the computing system 650. The number of subsets may include one subset for a dual-mirroring configuration, two subsets for a triple-mirroring configuration, three subsets for a quad-mirroring configuration, or the like. For example, the preprogrammed mirroring scheme may include a direct mapping of memory addresses in the first subset to memory addresses in the other subsets and also may include the number of subsets to which data is mirrored. The preprogrammed mirroring scheme may be preprogrammed by virtue of being included in the machine-readable medium 604, in some implementations. In some implementations, the mirrored subsets are unaddressable by the computing system. In some implementations, the NVM dies 605 may be organized into NVM die packages, and dies that contain corresponding mirrored data are included in different NVM die packages.

[0038] FIG. 7 is a block diagram illustrating a memory module 700 that includes a machine-readable medium encoded with example instructions to receive configuration instructions. The memory module 700 a controller 702 and a machine-readable medium 704, which may be analogous in many respects to the controller 602 and the machine-readable medium 604, respectively. The machine-readable medium 704 may be encoded with a set of executable instructions 706, 708, 710. Additionally or alternatively, the controller 702 may include electronic circuitry for performing the functionality described herein, including, but not limited to, the functionality of instructions 706, 708, and/or 710. The memory module 700 may also include a plurality of NVM dies 705, which may be coupled to the controller 702. In some implementations, the memory module 700 may be installed in and in communication with a computing system 750.

- 17 -

[0039] Instructions 706, when executed by the controller 702, may provide an identifier to a memory manager installed on the computing system 750, the identifier indicating to the memory manager an architecture of the plurality of NVM dies 705. For example, the architecture may relate to a quantity of the NVM dies 705, a capacity of the NVM dies 705, organization of the NVM dies 705 into NVM die packages, or the like. Instructions 708, when executed by the controller 702, may receive configuration instructions (from the memory manager of the computing system 750) to mirror data, stripe data, or generate parity data to the plurality of NVM dies 705 of the memory module 700. In some implementations, the configuration instructions may be related to a user selection from among memory module configuration options presented to the user, the options being based on the architecture indicated by the identifier provided by instructions 706. For example, the memory module configuration options may be to mirror data across a selectable number of mirrored subsets, to stripe data, to generate parity data, or a combination thereof. Instructions 710, when executed by the controller 702, may stripe data across dies of a first subset of the plurality of NVM dies 705. For example, the data to be striped by instructions 710 may be initially received by the memory module 700 from the computer system 750 (e.g., by a controller executing instructions 606 as described above). Instructions 712, when executed by the controller 702, may generate parity data based on data as striped across dies of the first subset of the plurality of NVM dies 705 (e.g., data striped by instructions 710). For example, instructions 712 may include instructions to perform an XOR operation on the striped data to generate the parity data.

[0040] In view of the foregoing description, it can be appreciated that a high level of reliability may be achieved for an NVM-based memory module on a memory bus of a computing system, by virtue of mirroring data within the memory module according to a preprogrammed mirroring scheme. Mirroring data within the memory module may increase mean-time between failure of NVM-based memory modules and reduce the probability of fatal system errors resulting from memory read errors. Additionally, by virtue of the memory module mirroring data independently of the computing system in

- 18 -

which it is installed, such reliability may be achieved with minimal user configuration and minimal performance impact to the computing system. Moreover, by virtue of receiving configuration instructions, the memory module may provide advanced redundancy features in addition to the preprogrammed mirroring scheme, if so desired by a user of the computing system.

[0041] In the foregoing description, numerous details are set forth to provide an understanding of the subject matter disclosed herein. However, implementation may be practiced without some or all of these details. Other implementations may include modifications and variations from the details discussed above. It is intended that the following claims cover such modifications and variations.

- 19 -

We claim:

1. A memory module comprising:
a plurality of non-volatile memory dies;
an interface for communicating with a memory bus of a computing system; and
a media controller to:
receive data over the interface addressed to a first subset of the plurality of non-volatile memory dies,
store the data to a die of the first subset, and
mirror, independent of the computing system, the data to a die of a second subset of the plurality of non-volatile memory dies according to a preprogrammed mirroring scheme, wherein the second subset is unaddressable by the computing system.
2. The memory module of claim 1, wherein the preprogrammed mirroring scheme includes a direct mapping of memory addresses in the first subset to memory addresses in the second subset, the direct mapping being intrinsic to the media controller.
3. The memory module of claim 1, wherein the media controller is to stripe the data across dies of first subset.
4. The memory module of claim 3, wherein the media controller is to generate parity data based on the data striped across dies of first subset.
5. The memory module of claim 1, wherein the non-volatile memory dies are organized into non-volatile memory die packages, and the die of the first subset and the die of the second subset are included in different non-volatile memory die packages.

6. The memory module of claim 1, wherein the non-volatile memory dies are organized into non-volatile memory die packages, and the die of the first subset and the die of the second subset are included in a same non-volatile memory die package.

7. The memory module of claim 1, wherein the media controller is to mirror the data to a number of subsets of the plurality of non-volatile memory dies, the number of subsets including two subsets for a triple-mirroring configuration and three subsets for a quad-mirroring configuration, and the subsets to which data is mirrored are unaddressable by the computing system.

8. The memory module of claim 1, wherein the media controller is to receive configuration instructions to mirror data, stripe data, or generate parity data on the plurality of non-volatile memory dies, the configuration instructions being received by the media controller over the interface from a memory manager installed on the computing system that presents memory module configuration options to a user based on an identification from the memory module that indicates to the memory manager an architecture of the plurality of non-volatile memory dies.

9. A method comprising:
receiving data at a media controller of a memory module from a computing system in which the memory module is installed;
storing, by the media controller, the received data to a die of a first subset of a plurality of non-volatile memory dies of the memory module;
and
mirroring, by the media controller, the received data to a die of a second subset of a plurality of non-volatile memory dies of the memory module, according to a mirroring scheme preprogrammed in the media controller and independently of the computing system,

- 21 -

wherein the second subset is unaddressable by the computing system, and

wherein the non-volatile memory dies are organized into non-volatile memory die packages, and the die of the first subset and the die of the second subset are included in different non-volatile memory die packages.

10. The method of claim 9, further comprising striping, by the media controller, the received data across dies of the first subset.

11. The method of claim 10, further comprising generating, by the media controller, parity data based on data striped across dies of the first subset.

12. The method of claim 9, further comprising receiving, at the media controller, configuration instructions to mirror data, stripe data, or generate parity data on the plurality of non-volatile memory dies, the configuration instructions being received from a memory manager installed on the computing system that presents memory module configuration options to a user based on an identification provided by the memory module that indicates to the memory manager an architecture of the plurality of non-volatile memory dies.

13. A non-transitory machine readable medium, storing instructions executable by a controller of a memory module, the non-transitory machine readable medium comprising:

instructions to receive data at the memory module from a computing system in which the memory module is installed;

instructions to store the received data to a first subset of a plurality of non-volatile memory dies of the memory module; and

instructions to mirror the received data to a number of subsets of the plurality of non-volatile memory dies different from the first subset, according to a preprogrammed mirroring scheme and independently of the

- 22 -

computing system, the number of subsets including one subset for a dual-mirroring configuration, two subsets for a triple-mirroring configuration, or three subsets for a quad-mirroring configuration,

wherein the mirrored subsets are unaddressable by the computing system.

14. The non-transitory machine readable medium of claim 13, further comprising:

instructions to stripe the received data across dies of the first subset; and

instructions to generate parity data based on data as striped across dies of the first subset.

15. The non-transitory machine readable medium of claim 13, further comprising

instructions to provide an identifier to a memory manager installed on the computing system, the identifier indicating to the memory manager an architecture of the plurality of non-volatile memory dies; and

instructions to receive, from the memory manager, configuration instructions to mirror data, stripe data, or generate parity data to the plurality of non-volatile memory dies of the memory module, the configuration instructions being related to a selection from among memory module configuration options presented to a user based on the indicated architecture.

1/7

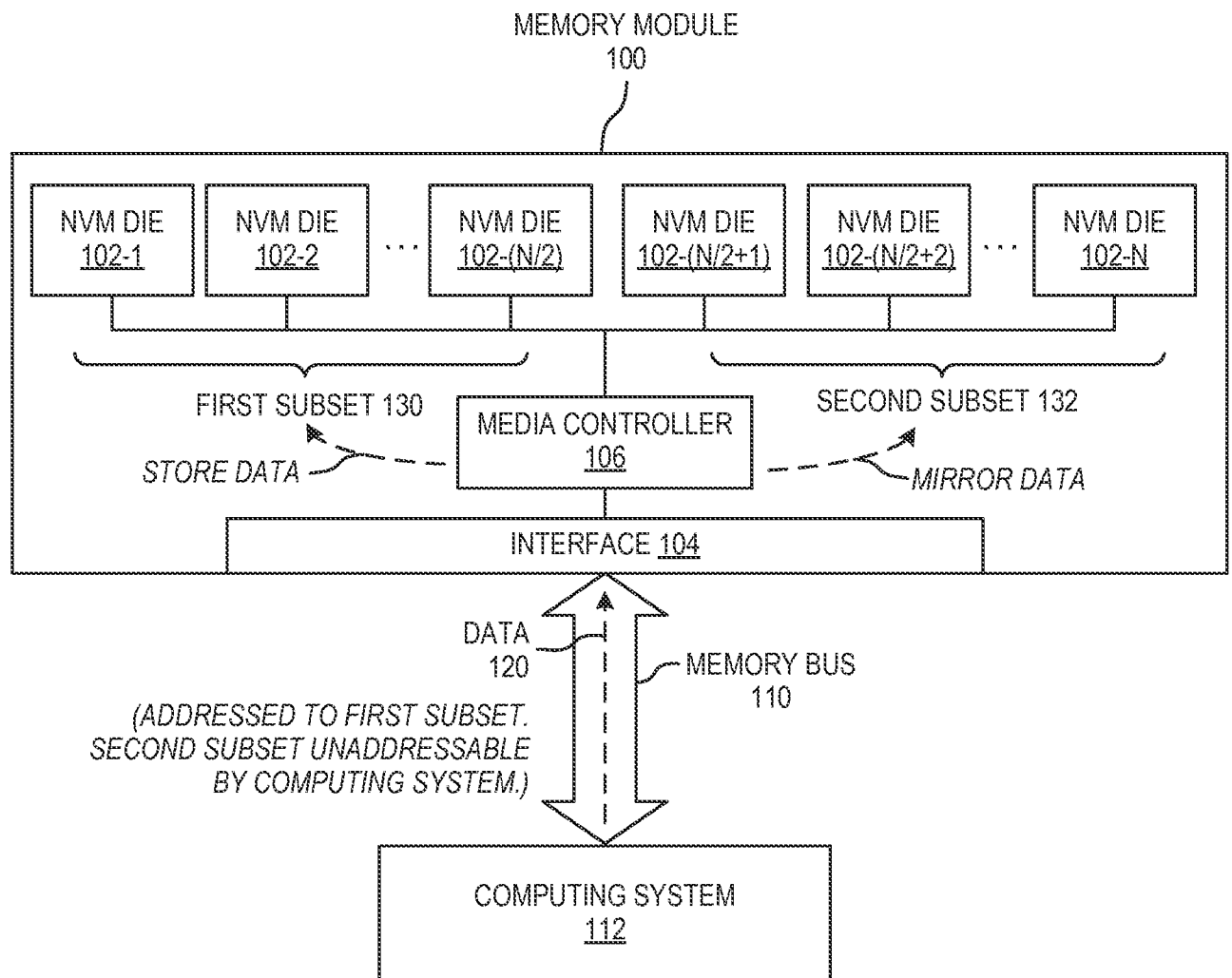


FIG. 1

2/7

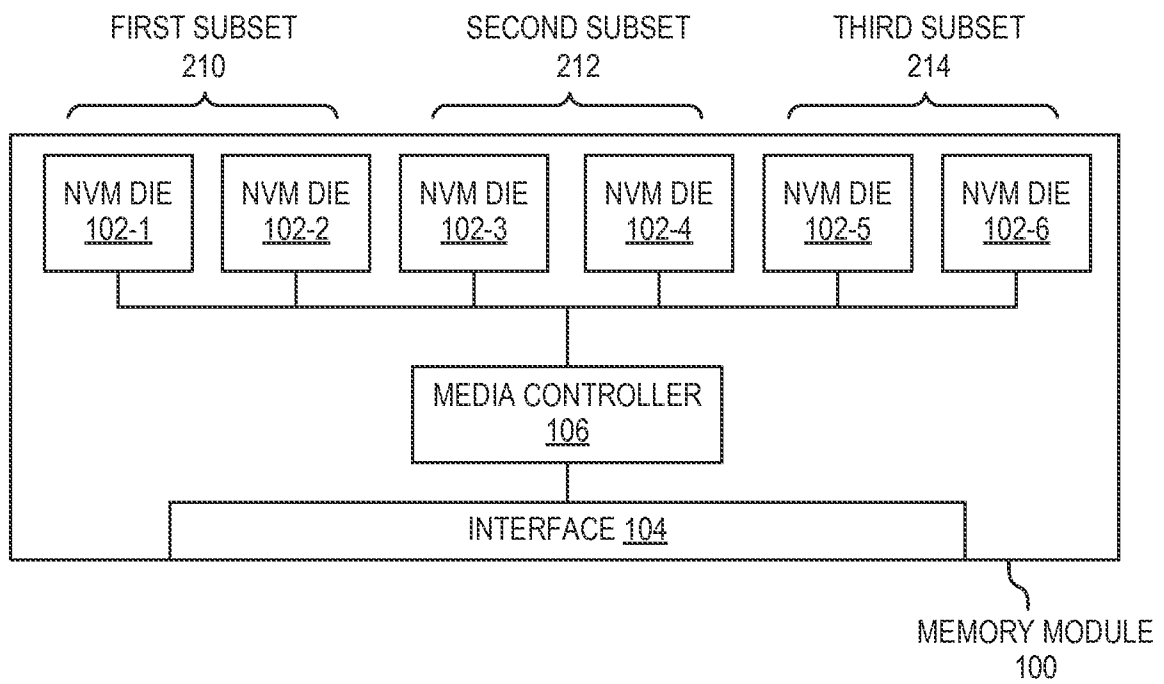


FIG. 2A

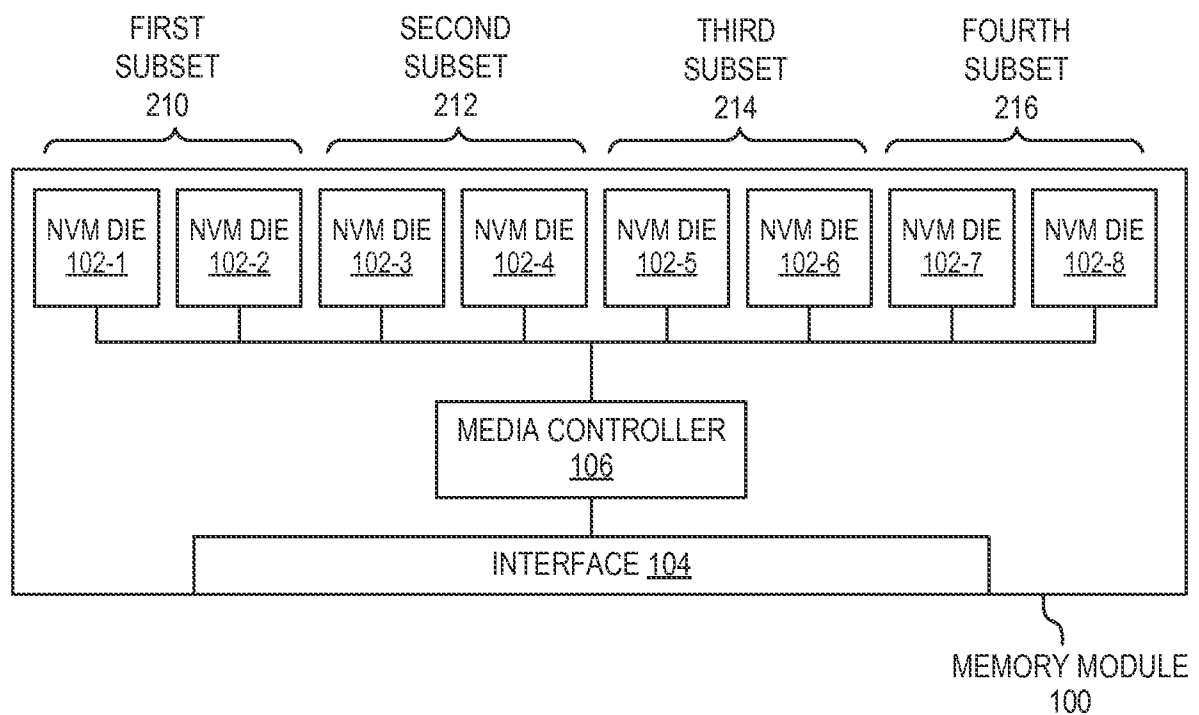


FIG. 2B

3/7

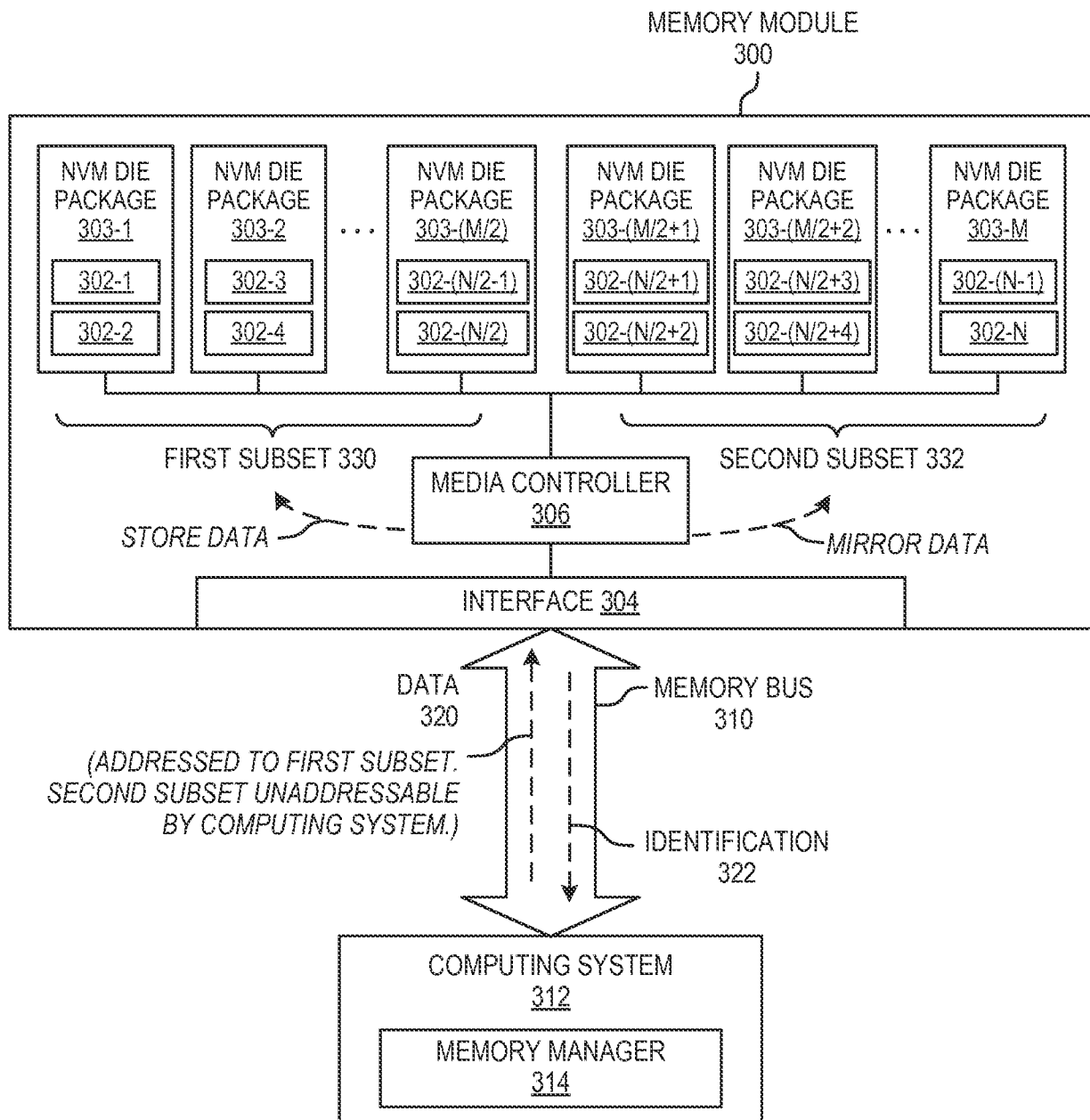


FIG. 3

4/7

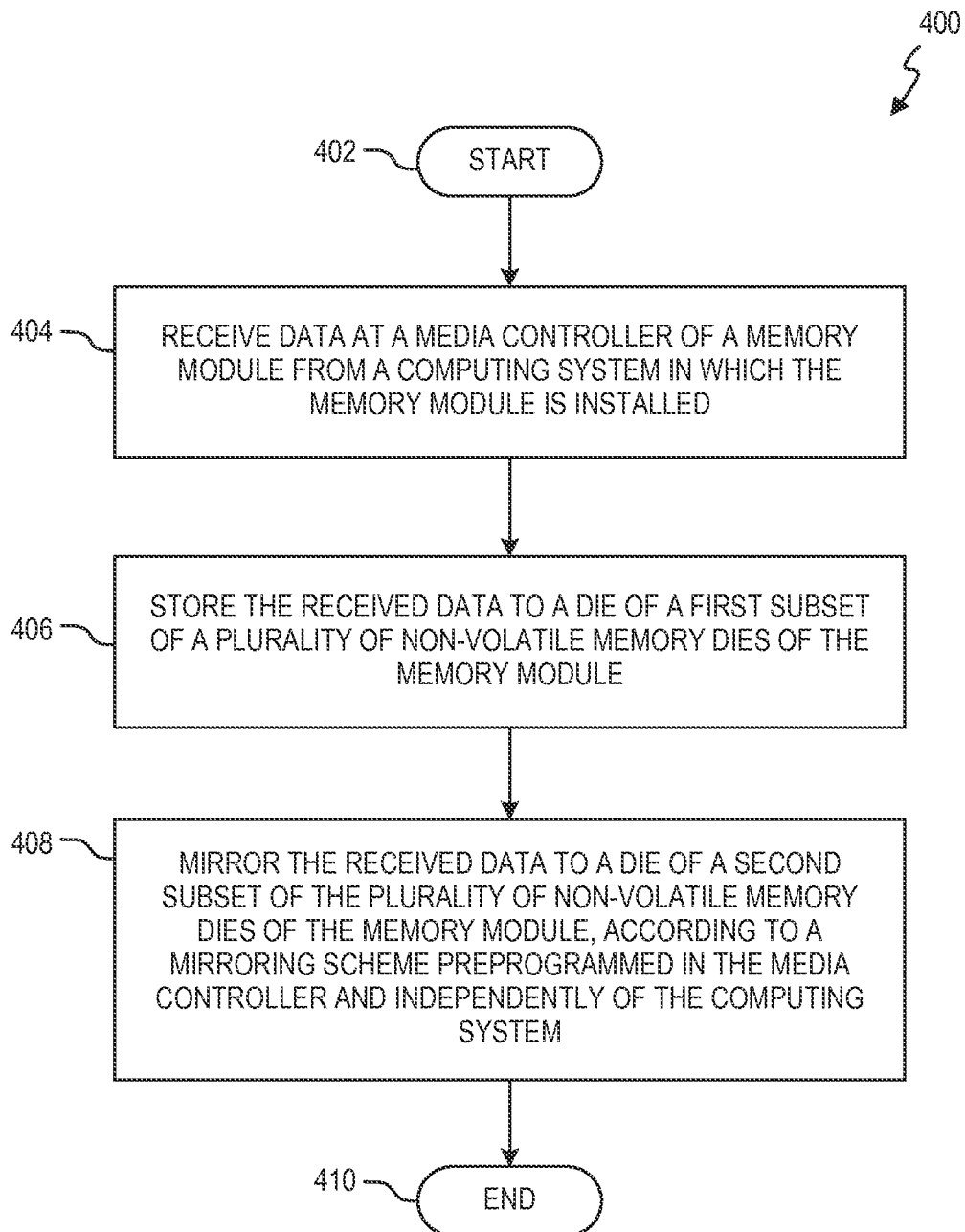


FIG. 4

5/7

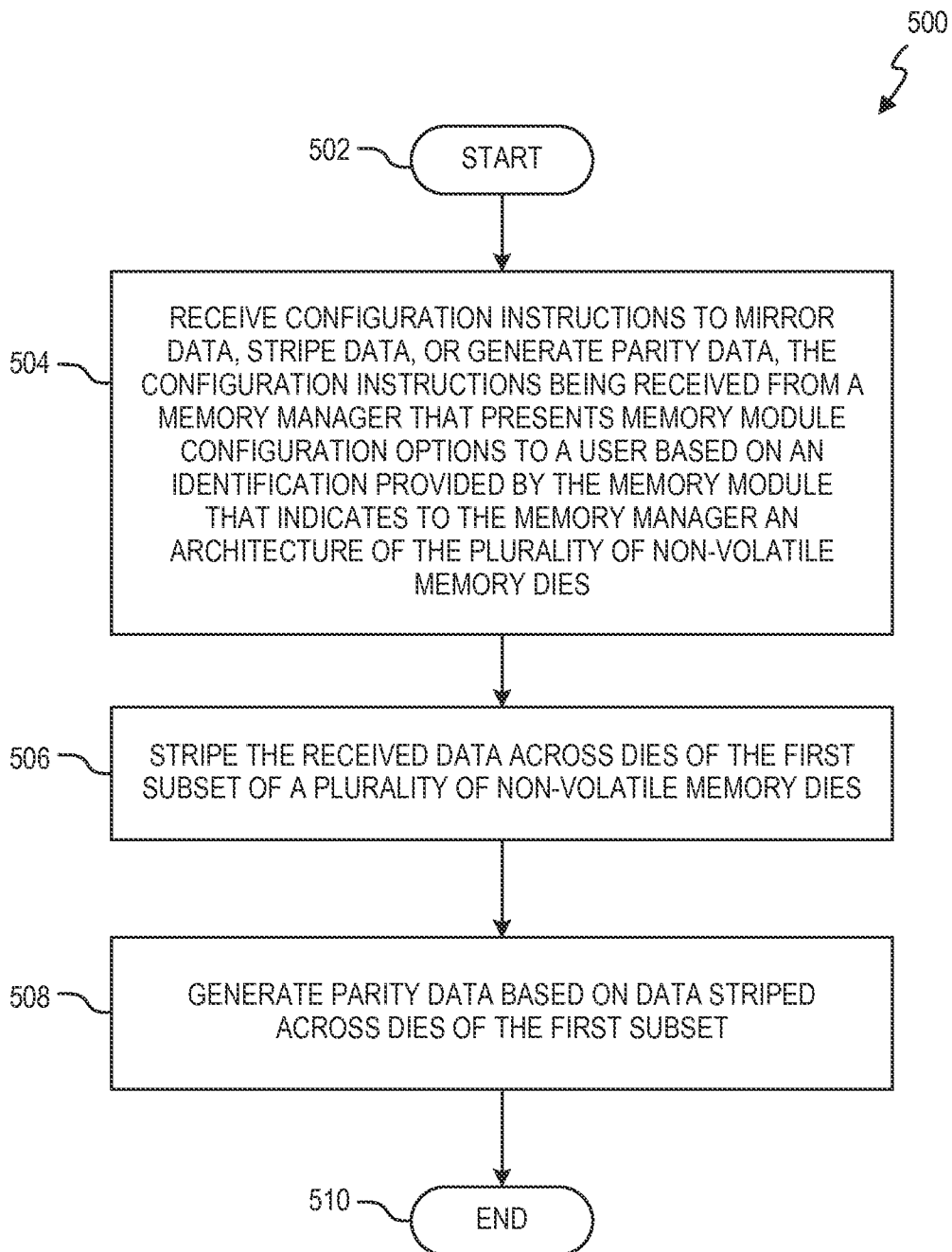


FIG. 5

6/7

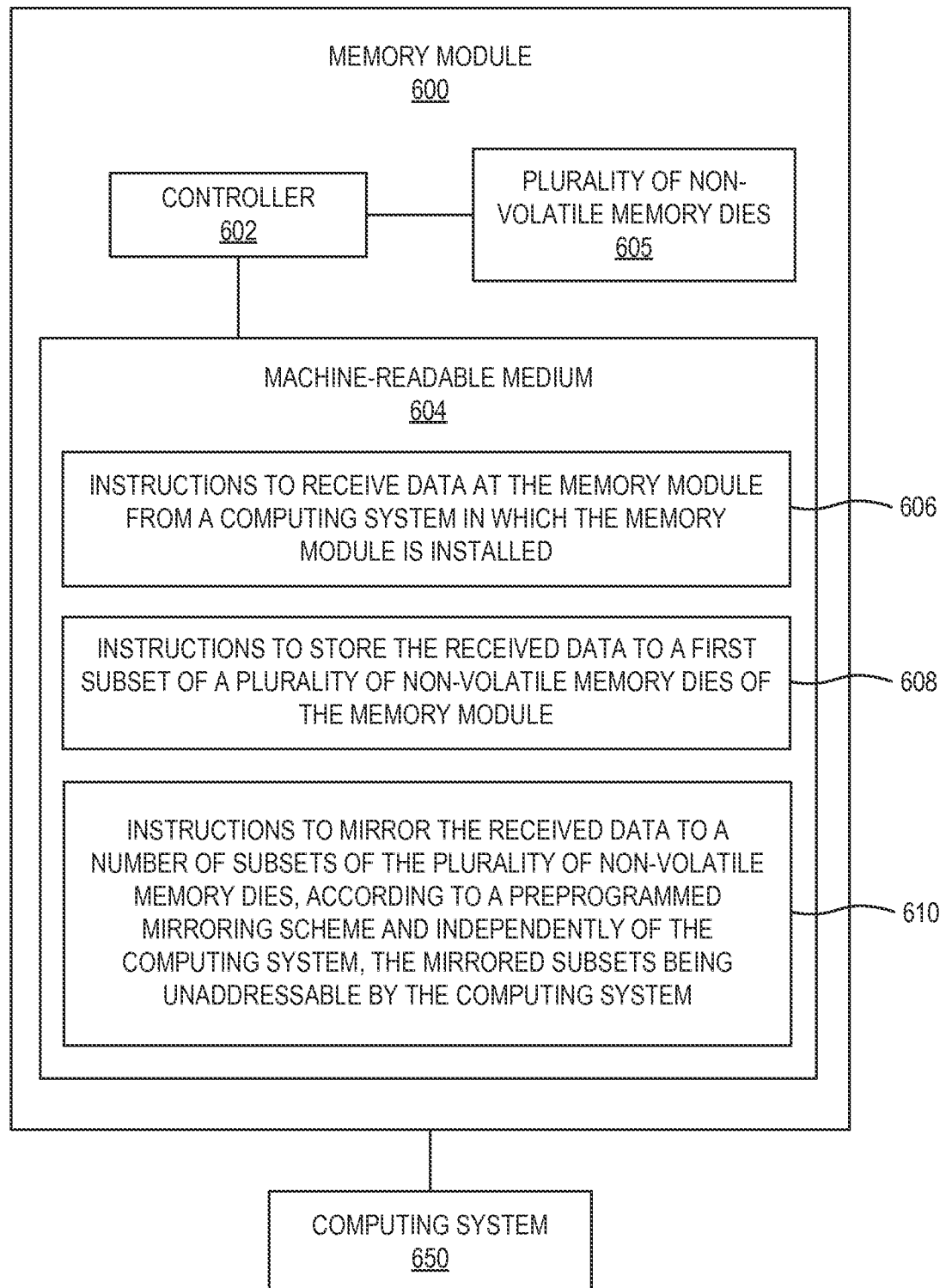


FIG. 6

7/7

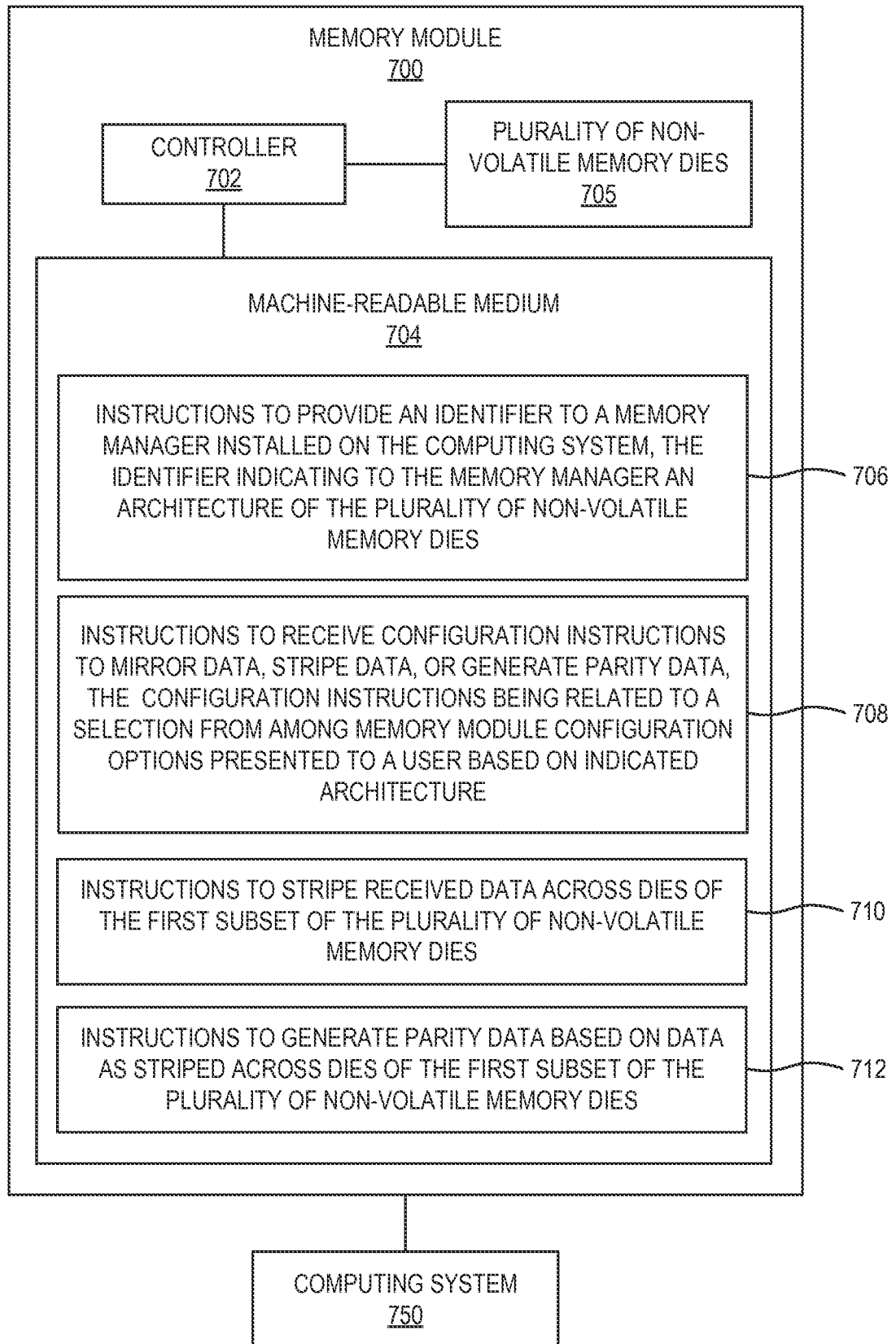


FIG. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/039754**A. CLASSIFICATION OF SUBJECT MATTER****G06F 11/14(2006.01)i, G06F 12/02(2006.01)i**

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

B. FIELDS SEARCHEDMinimum documentation searched (classification system followed by classification symbols)
G06F 11/14; H03M 13/09; G06F 12/02; G06F 12/16; G11C 16/04; G06F 11/10; G06F 11/20Documentation 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 modelsElectronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: memory module, die, mirroring, unaddressable, and similar terms.**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2009-0187806 A1 (JOHN C. PESCATORE) 23 July 2009 See paragraphs [0003], [0007], and [0015]-[0022]; claims 1 and 17; and figures 1-2 and 4.	1-6, 8-12
Y		7, 13-15
Y	US 2013-0019076 A1 (NETLIST, INC.) 17 January 2013 See paragraphs [0028]-[0033]; claim 1; and figure 2.	7, 13-15
A	US 2008-0140961 A1 (WILLIAM E. ATHERTON et al.) 12 June 2008 See paragraphs [0017]-[0018] and [0021]; and figures 1A-1B.	1-15
A	US 2012-0054543 A1 (ROLAND DREIER) 01 March 2012 See paragraphs [0020]-[0025] and [0032]-[0033]; and figures 2-3.	1-15
A	WO 2009-044302 A1 (NXP B. V.) 09 April 2009 See page 5, lines 2-10; and figure 1.	1-15



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

30 March 2016 (30.03.2016)

Date of mailing of the international search report

30 March 2016 (30.03.2016)

Name and mailing address of the ISA/KR

International Application Division
Korean Intellectual Property Office
189 Cheongsu-ro, Seo-gu, Daejeon, 35208, Republic of Korea

Facsimile No. +82-42-481-8578

Authorized officer

NHO, Ji Myong

Telephone No. +82-42-481-8528



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/039754

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2009-0187806 A1	23/07/2009	US 2006-0077750 A1 US 8341499 B2	13/04/2006 25/12/2012
US 2013-0019076 A1	17/01/2013	US 2015-248249 A1 US 8904098 B2	03/09/2015 02/12/2014
US 2008-0140961 A1	12/06/2008	EP 2097821 A1 US 9262284 B2 WO 2008-068176 A1	09/09/2009 16/02/2016 12/06/2008
US 2012-0054543 A1	01/03/2012	US 8601310 B2	03/12/2013
WO 2009-044302 A1	09/04/2009	None	