



(51) International Patent Classification:
G06F 13/16 (2006.01)

(21) International Application Number:

PCT/US2016/019974

(22) International Filing Date:

26 February 2016 (26.02.2016)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

14/670,250 26 March 2015 (26.03.2015) US

(71) Applicant: **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, California 95054 (US).

(72) Inventors: **GUO, Xin**; 6105 Castleknoll Drive, San Jose, California 95129 (US). **CARLTON, David B.**; 121 Embarcadero West, Apt 2222, Oakland, California 94607 (US). **NELSON, Scott**; 11-939 W 7th Ave, Vancouver, British Columbia V5Z 1C4 (CA). **PELSTER, David J.**;

1921 Corporate Center Cir., Ste. 3B, Longmont, Colorado 80501 (US). **SEBASTIAN, Donia**; 8437 Scenic Vista Way, Fair Oaks, California 95628 (US).

(74) Agent: **OSBORNE, David W.**; Thorpe North & Western, LLP, C/O CPA Global, P.O. Box 52050, Minneapolis, Minnesota 55402 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, 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,

[Continued on next page]

(54) Title: READ OPERATIONS IN MEMORY DEVICES

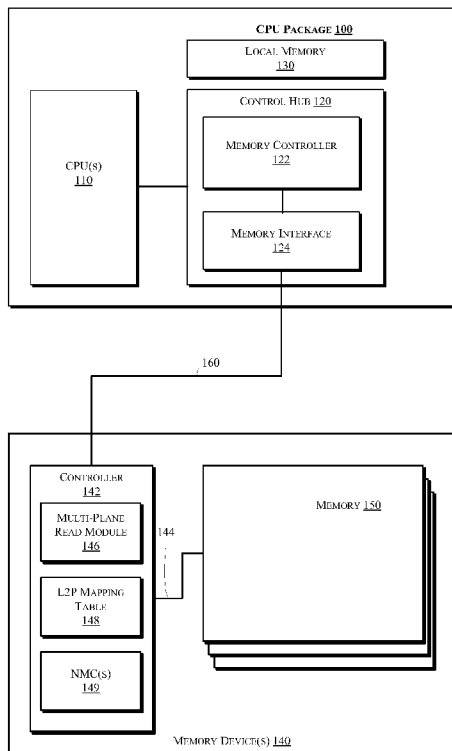


FIG. 1

(57) Abstract: Apparatus, systems, and methods to implement read operations in nonvolatile memory devices are described. In one example, a controller comprises logic to receive a first read request from a host device, place the first read request in a read queue comprising a plurality of read requests directed to the nonvolatile memory, determine a first target die and a first target plane in the nonvolatile memory for the first read request and combine the first read request with at least a second read request in the read queue to form a combined read request, wherein the second read request comprise a second target die, which is the same as the first target die, and a second target plane which is different from the first target plane. Other examples are also disclosed and claimed.



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).

Published:

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

READ OPERATIONS IN MEMORY DEVICES

FIELD

[0001] The present disclosure generally relates to the field of electronics. More particularly, aspects generally relate to read operations in memory devices.

BACKGROUND

[0002] Solid state drives (SSDs) or nonvolatile direct in-line memory modules (NV-DIMMs) provide high speed, nonvolatile memory capacity without the need for moving parts. SSDs rely on concurrency in read and write operations to increase performance. Memory in a SSD commonly comprises multiple physically separate “dies” that can be read from in parallel to improve performance. As manufacturers move to increasingly larger die sizes, smaller SSDs require fewer dies in them for a given memory density. This reduces the ability to utilize concurrency to improve read performance. This has a negative effect on performance. Accordingly, techniques to manage read operations in memory devices may find utility, e.g., in memory systems for electronic devices.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] The detailed description is provided with reference to the accompanying figures. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The use of the same reference numbers in different figures indicates similar or identical items.

[0004] Fig. 1 is a schematic, block diagram illustration of components of an apparatus in which read operations in memory devices may be implemented in accordance with various examples discussed herein.

[0005] Fig. 2 is a schematic, block diagram illustration of a memory in accordance with various examples discussed herein.

[0006] Fig. 3 is a schematic illustrations of mapping read operations into a memory in accordance with various examples discussed herein.

[0007] Figs. 4-5 are schematic illustrations of operations in methods to implement read operations in nonvolatile memory devices in accordance with various examples discussed herein.

[0008] Figs. 6-10 are schematic, block diagram illustrations of electronic devices which may be adapted to implement read operations in nonvolatile memory devices in accordance with various examples discussed herein.

DETAILED DESCRIPTION

[0009] In the following description, numerous specific details are set forth in order to provide a thorough understanding of various examples. However, various examples may be practiced without the specific details. In other instances, well-known methods, procedures, components, and circuits have not been described in detail so as not to obscure the particular examples. Further, various aspects of examples may be performed using various means, such as integrated semiconductor circuits (“hardware”), computer-readable instructions organized into one or more programs (“software”), or some combination of hardware and software. For the purposes of this disclosure reference to “logic” shall mean either hardware, software, or some combination thereof.

[0010] As described in detail below, by adroitly combining read operation in a read queue read operations directed to different planes on a single die may be executed concurrently, thereby increasing performance parameters of a SSD. Specific details of a systems and methods to manage read devices in electronic devices will be described below with reference to Figs. 1-10.

[0011] Fig. 1 is a schematic, block diagram illustration of components of an apparatus in which methods to manage nonvolatile memory devices may be implemented in accordance with various examples discussed herein. Referring to Fig. 1, in some examples a central processing unit (CPU) package 100 which may comprise one or more CPUs 110 coupled to a control hub 120 and a local

memory 130. Control hub 120 comprises a memory controller 122 and a memory interface 124. In some examples the control hub 120 may be integrated with the processor(s) 110.

[0012] Memory interface 124 is coupled to one or more remote memory devices 140 by a communication bus 160. Storage device 140 may be implemented as a solid state drive (SSD), a nonvolatile direct in-line memory module (NV-DIMM) or the like and comprise a controller 142 and memory 150. In various examples, at least some of the memory 150 may comprise nonvolatile memory, e.g., NAND (flash) memory, ferroelectric random-access memory (FeTRAM), nanowire-based non-volatile memory, memory that incorporates memristor technology, a static random access memory (SRAM), three dimensional (3D) cross point memory, phase change memory (PCM), spin-transfer torque memory (STT-RAM) or NAND memory or NOR memory. The specific configuration of the memory 150 in the memory device(s) 140 is not critical. In such embodiments the memory interface may comprise a Serial ATA interface, a PCI Express (PCIE) to 100 interface, or the like.

[0013] Controller 142 may comprise logic, at least partially including hardware logic, defining a multi-plane read module 146. Further, controller 142 may maintain a logical address to physical address mapping table 148 which maps a logical address received with a read request to a physical address in the nonvolatile memory and one or more NAND media channels 248, which are coupled to memory 150 via communication link 144

[0014] Fig. 2 is a schematic, block diagram illustration of a memory such as memory 150 in accordance with various examples discussed herein. Referring to Fig. 2, in some examples memory 150 may comprise a plurality of dies, each of which is subdivided into a number of planes. In the example depicted in Fig. 2 the memory 150 comprises a number M dies, each of which is divided into four planes. Thus, die 1 210A comprises plane 1 212A, plane 2 212B, plane 3 212C, and plane 4 212D. Similarly die 2 210B through die M 210M each comprise four planes.

[0015] Operations implemented by controller 142 will be described with reference to Figs. 3 and 4-5. Referring briefly to Fig. 3, read operations received in controller 142 may be placed in a read queue 310. Read queue may be implemented as a logical structure, e.g., an array, in controller 142, and controller 142 may select read operations from read queue in any order for execution against the memory 150.

[0016] Referring to Fig. 4, at operation 410 the controller 142 receives a read request from a host device, e.g., memory controller 122, directed to memory 150. At operation 415, the controller 142 places the read operation in read queue 310. Thus, in operation, read queue 310 may comprise a plurality of read requests directed to memory 150.

[0017] At operation 420 the controller 142 determines a target die and a target plane for read operations received from the host device. For example, read

operations may arrive from the host device with a logical memory address assigned by the host device. The controller 142 may reference the logical memory address in the logical to physical mapping table 148 to obtain a physical address in the memory 150 from which data in the read operation is to be retrieved. The die and plane information may be associated with the physical address in the read queue. Thus, referring to Fig. 3, the respective read requests in the read queue may each have a die and a plane on the die associated with them. For example, if the die includes four planes then two bits of the physical address may be used to identify the plane associated with the request. Similarly, if the memory includes 24 dies, then 5 bits of the physical address may be used to identify the die.

[0018] At operation 425 the controller 142 scans the read queue 310 for read requests that are directed to the same target die but to different planes on the target die. If, at operation 430 there are no matches then control passes to operation 440 and the controller 142 executes read operations in accordance with normal operations, e.g., the read operations may be executed in a first-in, first-out (FIFO) manner.

[0019] By contrast, if at operation 430 there are matches, then control passes to operation 435 and the controller 142 combines matching read requests to form a combined read request. For example, referring to Fig. 2, the controller 142 may scan the read queue 310 and identify that there are three read requests directed to die 15. A first read request directed to die 15 is directed to plane 2, a

second read request is directed to plane 1, and a third read request is directed to plane 3. In this scenario, at operation 430 the controller 142 will combine the three read requests directed to die 15 and execute the read requests concurrently (operation 440). In some examples the controller 142 may associate a unique identifier (e.g., a tag) with each of the separate read requests that are combined to form a combined read request.

[0020] The controller 142 must allow commands with four fully independent physical address fields (one for each plane) in order to permit multi plane reads to be sent to memory 150. In some examples controller 142 may enable a multiplane read mode in which the plane 2/3/4 transfer buffer destination address fields are used for the plane 2/3/4 page address.

[0021] Fig. 5 is a flowchart illustrating operations applied to data returned from a combined read request. Referring to Fig. 5, at operation 510 data retrieved from memory 150 as a result of a combined read request is received at controller 142. At operation 520 the controller 142 separates the data retrieved from the separate the data retrieved from the combined read request such that the data is associated with the original read request. For example, data retrieved from memory in response to a combined read request may be separated into first data associated with a first read request and second data associated with a second read request, third data associated with a third read request, and fourth data associated with a fourth read request. In some examples the controller 142 may use the

unique identifier (e.g., tag) associated with the data in the combined read request to separate the data retrieved from the combined read request.

[0022] At operation 520 the controller returns data associated with the first read request to the host which generated the first read request, and at operation 525 the controller 142 returns the second data associated with the second read request to a host which generated the second read request. One skilled in the art will recognize that in cases where more than two read requests are combined to form a combined read request the controller may return third data associated with a third read request and/or fourth data associated with a fourth read request.

[0023] Some existing controllers such as controller 142 may be limited to using a single same page address for each plane due to a matching limitation in the NAND media. In order to accommodate this limitation, an independent multi-plane read mode may be added to the NAND Media Channel (NMC) block. As described above, in some examples a multi-plane read mode may use the plane 2/3/4 transfer buffer destination address fields to use for the plane 2/3/4 page address.

[0024] As described above, in some examples the electronic device may be embodied as a computer system. Fig. 6 illustrates a block diagram of a computing system 600 in accordance with an example. The computing system 600 may include one or more central processing unit(s) (CPUs) 602 or processors that communicate via an interconnection network (or bus) 604. The processors

602 may include a general purpose processor, a network processor (that processes data communicated over a computer network 603), or other types of a processor (including a reduced instruction set computer (RISC) processor or a complex instruction set computer (CISC)). Moreover, the processors 602 may have a single or multiple core design. The processors 602 with a multiple core design may integrate different types of processor cores on the same integrated circuit (IC) die. Also, the processors 602 with a multiple core design may be implemented as symmetrical or asymmetrical multiprocessors. In an example, one or more of the processors 602 may be the same or similar to the processors 102 of Fig. 1. For example, one or more of the processors 602 may include the control unit 120 discussed with reference to Figs. 1-3. Also, the operations discussed with reference to Figs. 3-5 may be performed by one or more components of the system 600.

[0025] A chipset 606 may also communicate with the interconnection network 604. The chipset 606 may include a memory control hub (MCH) 608. The MCH 608 may include a memory controller 610 that communicates with a memory 612 (which may be the same or similar to the memory 130 of Fig. 1). The memory 612 may store data, including sequences of instructions, that may be executed by the CPU 602, or any other device included in the computing system 600. In one example, the memory 612 may include one or more volatile storage (or memory) devices such as random access memory (RAM), dynamic RAM (DRAM), synchronous DRAM (SDRAM), static RAM (SRAM), or other types

of storage devices. Nonvolatile memory may also be utilized such as a hard disk. Additional devices may communicate via the interconnection network 604, such as multiple CPUs and/or multiple system memories.

[0026] The MCH 608 may also include a graphics interface 614 that communicates with a display device 616. In one example, the graphics interface 614 may communicate with the display device 616 via an accelerated graphics port (AGP). In an example, the display 616 (such as a flat panel display) may communicate with the graphics interface 614 through, for example, a signal converter that translates a digital representation of an image stored in a storage device such as video memory or system memory into display signals that are interpreted and displayed by the display 616. The display signals produced by the display device may pass through various control devices before being interpreted by and subsequently displayed on the display 616.

[0027] A hub interface 618 may allow the MCH 608 and an input/output control hub (ICH) 620 to communicate. The ICH 620 may provide an interface to I/O device(s) that communicate with the computing system 600. The ICH 620 may communicate with a bus 622 through a peripheral bridge (or controller) 624, such as a peripheral component interconnect (PCI) bridge, a universal serial bus (USB) controller, or other types of peripheral bridges or controllers. The bridge 624 may provide a data path between the CPU 602 and peripheral devices. Other types of topologies may be utilized. Also, multiple buses may communicate with the ICH 620, e.g., through multiple bridges or controllers. Moreover, other

peripherals in communication with the ICH 620 may include, in various examples, integrated drive electronics (IDE) or small computer system interface (SCSI) hard drive(s), USB port(s), a keyboard, a mouse, parallel port(s), serial port(s), floppy disk drive(s), digital output support (e.g., digital video interface (DVI)), or other devices.

[0028] The bus 622 may communicate with an audio device 626, one or more disk drive(s) 628, and a network interface device 630 (which is in communication with the computer network 603). Other devices may communicate via the bus 622. Also, various components (such as the network interface device 630) may communicate with the MCH 608 in some examples. In addition, the processor 602 and one or more other components discussed herein may be combined to form a single chip (e.g., to provide a System on Chip (SOC)). Furthermore, the graphics accelerator 616 may be included within the MCH 608 in other examples.

[0029] Furthermore, the computing system 600 may include volatile and/or nonvolatile memory (or storage). For example, nonvolatile memory may include one or more of the following: read-only memory (ROM), programmable ROM (PROM), erasable PROM (EPROM), electrically EPROM (EEPROM), a disk drive (e.g., 628), a floppy disk, a compact disk ROM (CD-ROM), a digital versatile disk (DVD), flash memory, a magneto-optical disk, or other types of nonvolatile machine-readable media that are capable of storing electronic data (e.g., including instructions).

[0030] Fig. 7 illustrates a block diagram of a computing system 700, according to an example. The system 700 may include one or more processors 702-1 through 702-N (generally referred to herein as “processors 702” or “processor 702”). The processors 702 may communicate via an interconnection network or bus 704. Each processor may include various components some of which are only discussed with reference to processor 702-1 for clarity. Accordingly, each of the remaining processors 702-2 through 702-N may include the same or similar components discussed with reference to the processor 702-1.

[0031] In an example, the processor 702-1 may include one or more processor cores 706-1 through 706-M (referred to herein as “cores 706” or more generally as “core 706”), a shared cache 708, a router 710, and/or a processor control logic or unit 720. The processor cores 706 may be implemented on a single integrated circuit (IC) chip. Moreover, the chip may include one or more shared and/or private caches (such as cache 708), buses or interconnections (such as a bus or interconnection network 712), memory controllers, or other components.

[0032] In one example, the router 710 may be used to communicate between various components of the processor 702-1 and/or system 700. Moreover, the processor 702-1 may include more than one router 710. Furthermore, the multitude of routers 710 may be in communication to enable data routing between various components inside or outside of the processor 702-1.

[0033] The shared cache 708 may store data (e.g., including instructions) that are utilized by one or more components of the processor 702-1, such as the cores 706. For example, the shared cache 708 may locally cache data stored in a memory 714 for faster access by components of the processor 702. In an example, the cache 708 may include a mid-level cache (such as a level 2 (L2), a level 3 (L3), a level 4 (L4), or other levels of cache), a last level cache (LLC), and/or combinations thereof. Moreover, various components of the processor 702-1 may communicate with the shared cache 708 directly, through a bus (e.g., the bus 712), and/or a memory controller or hub. As shown in Fig. 7, in some examples, one or more of the cores 706 may include a level 1 (L1) cache 716-1 (generally referred to herein as “L1 cache 716”). In one example, the control unit 720 may include logic to implement the operations described above with reference to the memory controller 122 in Fig. 2.

[0034] Fig. 8 illustrates a block diagram of portions of a processor core 706 and other components of a computing system, according to an example. In one example, the arrows shown in Fig. 8 illustrate the flow direction of instructions through the core 706. One or more processor cores (such as the processor core 706) may be implemented on a single integrated circuit chip (or die) such as discussed with reference to Fig. 7. Moreover, the chip may include one or more shared and/or private caches (e.g., cache 708 of Fig. 7), interconnections (e.g., interconnections 704 and/or 112 of Fig. 7), control units, memory controllers, or other components.

[0035] As illustrated in Fig. 8, the processor core 706 may include a fetch unit 802 to fetch instructions (including instructions with conditional branches) for execution by the core 706. The instructions may be fetched from any storage devices such as the memory 714. The core 706 may also include a decode unit 804 to decode the fetched instruction. For instance, the decode unit 804 may decode the fetched instruction into a plurality of uops (micro-operations).

[0036] Additionally, the core 706 may include a schedule unit 806. The schedule unit 806 may perform various operations associated with storing decoded instructions (e.g., received from the decode unit 804) until the instructions are ready for dispatch, e.g., until all source values of a decoded instruction become available. In one example, the schedule unit 806 may schedule and/or issue (or dispatch) decoded instructions to an execution unit 808 for execution. The execution unit 808 may execute the dispatched instructions after they are decoded (e.g., by the decode unit 804) and dispatched (e.g., by the schedule unit 806). In an example, the execution unit 808 may include more than one execution unit. The execution unit 808 may also perform various arithmetic operations such as addition, subtraction, multiplication, and/or division, and may include one or more an arithmetic logic units (ALUs). In an example, a co-processor (not shown) may perform various arithmetic operations in conjunction with the execution unit 808.

[0037] Further, the execution unit 808 may execute instructions out-of-order. Hence, the processor core 706 may be an out-of-order processor core in

one example. The core 706 may also include a retirement unit 810. The retirement unit 810 may retire executed instructions after they are committed. In an example, retirement of the executed instructions may result in processor state being committed from the execution of the instructions, physical registers used by the instructions being de-allocated, etc.

[0038] The core 706 may also include a bus unit 714 to enable communication between components of the processor core 706 and other components (such as the components discussed with reference to Fig. 8) via one or more buses (e.g., buses 804 and/or 812). The core 706 may also include one or more registers 816 to store data accessed by various components of the core 706 (such as values related to power consumption state settings).

[0039] Furthermore, even though Fig. 7 illustrates the control unit 720 to be coupled to the core 706 via interconnect 812, in various examples the control unit 720 may be located elsewhere such as inside the core 706, coupled to the core via bus 704, etc.

[0040] In some examples, one or more of the components discussed herein can be embodied as a System On Chip (SOC) device. Fig. 9 illustrates a block diagram of an SOC package in accordance with an example. As illustrated in Fig. 9, SOC 902 includes one or more Central Processing Unit (CPU) cores 920, one or more Graphics Processor Unit (GPU) cores 930, an Input/Output (I/O) interface 940, and a memory controller 942. Various components of the SOC

package 902 may be coupled to an interconnect or bus such as discussed herein with reference to the other figures. Also, the SOC package 902 may include more or less components, such as those discussed herein with reference to the other figures. Further, each component of the SOC package 902 may include one or more other components, e.g., as discussed with reference to the other figures herein. In one example, SOC package 902 (and its components) is provided on one or more Integrated Circuit (IC) die, e.g., which are packaged into a single semiconductor device.

[0041] As illustrated in Fig. 9, SOC package 902 is coupled to a memory 960 (which may be similar to or the same as memory discussed herein with reference to the other figures) via the memory controller 942. In an example, the memory 960 (or a portion of it) can be integrated on the SOC package 902.

[0042] The I/O interface 940 may be coupled to one or more I/O devices 970, e.g., via an interconnect and/or bus such as discussed herein with reference to other figures. I/O device(s) 970 may include one or more of a keyboard, a mouse, a touchpad, a display, an image/video capture device (such as a camera or camcorder/video recorder), a touch screen, a speaker, or the like.

[0043] Fig. 10 illustrates a computing system 1000 that is arranged in a point-to-point (PtP) configuration, according to an example. In particular, Fig. 10 shows a system where processors, memory, and input/output devices are interconnected by a number of point-to-point interfaces. The operations discussed

with reference to Fig. 2 may be performed by one or more components of the system 1000.

[0044] As illustrated in Fig. 10, the system 1000 may include several processors, of which only two, processors 1002 and 1004 are shown for clarity. The processors 1002 and 1004 may each include a local memory controller hub (MCH) 1006 and 1008 to enable communication with memories 1010 and 1012. MCH 1006 and 1008 may include the memory controller 120 and/or logic of Fig. 1 in some examples.

[0045] In an example, the processors 1002 and 1004 may be one of the processors 702 discussed with reference to Fig. 7. The processors 1002 and 1004 may exchange data via a point-to-point (PtP) interface 1014 using PtP interface circuits 1016 and 1018, respectively. Also, the processors 1002 and 1004 may each exchange data with a chipset 1020 via individual PtP interfaces 1022 and 1024 using point-to-point interface circuits 1026, 1028, 1030, and 1032. The chipset 1020 may further exchange data with a high-performance graphics circuit 1034 via a high-performance graphics interface 1036, e.g., using a PtP interface circuit 1037.

[0046] As shown in Fig. 10, one or more of the cores 106 and/or cache 108 of Fig. 1 may be located within the processors 1002 and 1004. Other examples, however, may exist in other circuits, logic units, or devices within the system

1000 of Fig. 10. Furthermore, other examples may be distributed throughout several circuits, logic units, or devices illustrated in Fig. 10.

[0047] The chipset 1020 may communicate with a bus 1040 using a point-to-point PtP interface circuit 1041. The bus 1040 may have one or more devices that communicate with it, such as a bus bridge 1042 and I/O devices 1043. Via a bus 1044, the bus bridge 1043 may communicate with other devices such as a keyboard/mouse 1045, communication devices 1046 (such as modems, network interface devices, or other communication devices that may communicate with the computer network 803), audio I/O device, and/or a data storage device 1048. The data storage device 1048 (which may be a hard disk drive or a NAND flash based solid state drive) may store code 1049 that may be executed by the processors 1002 and/or 1004.

[0048] The following pertains to further examples.

[0049] Example 1 is an electronic device comprising at least one processor, at least one storage device comprising a nonvolatile memory, and a controller coupled to the memory and comprising logic , at least partially including hardware logic, to receive a first read request from a host device, place the first read request in a read queue comprising a plurality of read requests directed to the nonvolatile memory, determine a first target die and a first target plane in the nonvolatile memory for the first read request, and combine the first read request with at least a second read request in the read queue to form a

combined read request, wherein the second read request comprises a second target die, which is the same as the first target die, and a second target plane which is different from the first target plane.

[0050] In Example 2, the subject matter of Example 1 can optionally include logic, at least partially including hardware logic, to execute the combined read request to retrieve data from the nonvolatile memory requested in the combined read request.

[0051] In Example 3, the subject matter of any one of Examples 1–2 can optionally include logic, at least partially including hardware logic, to separate the data retrieved from the combined read request into to first data associated with the first read request and second data associated with the second read request

[0052] In Example 4, the subject matter of any one of Examples 1–3 can optionally include logic, at least partially including hardware logic, to return the first data associated with the first read request to a host which generated the first read request and return the second data associated with the second read request to a host which generated the second read request.

[0053] In Example 5, the subject matter of any one of Examples 1–4 can optionally include logic, at least partially including hardware logic, to combine the first read request and the second read request in the read queue with at least a third read request in the read queue.

[0054] In Example 6, the subject matter of any one of Examples 1–5 can optionally include logic, at least partially including hardware logic, to maintain a logical address to physical address mapping table which maps a logical address received with a read request to a physical address in the nonvolatile memory.

[0055] In Example 7, the subject matter of any one of Examples 1–6 can optionally include an arrangement in which one or more physical addresses in the nonvolatile memory are associated with a die and a plane in the nonvolatile memory.

[0056] Example 8 is a storage device, comprising a nonvolatile memory, and a controller coupled to the memory and comprising logic, at least partially including hardware logic, to receive a first read request from a host device, place the first read request in a read queue comprising a plurality of read requests directed to the nonvolatile memory, determine a first target die and a first target plane in the nonvolatile memory for the first read request, and combine the first read request with at least a second read request in the read queue to form a combined read request, wherein the second read request comprises a second target die, which is the same as the first target die, and a second target plane which is different from the first target plane.

[0057] In Example 9, the subject matter of Example 8 can optionally include logic, at least partially including hardware logic, to execute the combined

read request to retrieve data from the nonvolatile memory requested in the combined read request.

[0058] In Example 10, the subject matter of any one of Examples 8-9 can optionally include logic, at least partially including hardware logic, to separate the data retrieved from the combined read request into to first data associated with the first read request and second data associated with the second read request

[0059] In Example 11, the subject matter of any one of Examples 8-10 can optionally include logic, at least partially including hardware logic, to return the first data associated with the first read request to a host which generated the first read request and return the second data associated with the second read request to a host which generated the second read request.

[0060] In Example 12, the subject matter of any one of Examples 8-11 can optionally include logic, at least partially including hardware logic, to combine the first read request and the second read request in the read queue with at least a third read request in the read queue.

[0061] In Example 13, the subject matter of any one of Examples 8-12 can optionally include logic, at least partially including hardware logic, to maintain a logical address to physical address mapping table which maps a logical address received with a read request to a physical address in the nonvolatile memory.

[0062] In Example 14, the subject matter of any one of Examples 8-13 can optionally include an arrangement in which one or more physical addresses in the nonvolatile memory are associated with a die and a plane in the nonvolatile memory.

[0063] Example 15 is a controller coupled to the memory and comprising logic, at least partially including hardware logic, to receive a first read request from a host device, place the first read request in a read queue comprising a plurality of read requests directed to the nonvolatile memory, determine a first target die and a first target plane in the nonvolatile memory for the first read request, and combine the first read request with at least a second read request in the read queue to form a combined read request, wherein the second read request comprises a second target die, which is the same as the first target die, and a second target plane which is different from the first target plane.

[0064] In Example 16, the subject matter of Example 15 can optionally include logic, at least partially including hardware logic, to execute the combined read request to retrieve data from the nonvolatile memory requested in the combined read request.

[0065] In Example 17, the subject matter of any one of Examples 15-16 can optionally include logic, at least partially including hardware logic, to separate the data retrieved from the combined read request into to first data

associated with the first read request and second data associated with the second read request

[0066] In Example 18, the subject matter of any one of Examples 15-17 can optionally include logic, at least partially including hardware logic, to return the first data associated with the first read request to a host which generated the first read request and return the second data associated with the second read request to a host which generated the second read request.

[0067] In Example 19, the subject matter of any one of Examples 15-18 can optionally include logic, at least partially including hardware logic, to combine the first read request and the second read request in the read queue with at least a third read request in the read queue.

[0068] In Example 20, the subject matter of any one of Examples 15-19 can optionally include logic, at least partially including hardware logic, to maintain a logical address to physical address mapping table which maps a logical address received with a read request to a physical address in the nonvolatile memory.

[0069] In Example 21, the subject matter of any one of Examples 15-20 can optionally include an arrangement in which one or more physical addresses in the nonvolatile memory are associated with a die and a plane in the nonvolatile memory.

[0070] In various examples, the operations discussed herein, e.g., with reference to Figs. 1-10, may be implemented as hardware (e.g., circuitry), software, firmware, microcode, or combinations thereof, which may be provided as a computer program product, e.g., including a tangible (e.g., non-transitory) machine-readable or computer-readable medium having stored thereon instructions (or software procedures) used to program a computer to perform a process discussed herein. Also, the term “logic” may include, by way of example, software, hardware, or combinations of software and hardware. The machine-readable medium may include a storage device such as those discussed herein.

[0071] Reference in the specification to “one example” or “an example” means that a particular feature, structure, or characteristic described in connection with the example may be included in at least an implementation. The appearances of the phrase “in one example” in various places in the specification may or may not be all referring to the same example.

[0072] Also, in the description and claims, the terms “coupled” and “connected,” along with their derivatives, may be used. In some examples, “connected” may be used to indicate that two or more elements are in direct physical or electrical contact with each other. “Coupled” may mean that two or more elements are in direct physical or electrical contact. However, “coupled” may also mean that two or more elements may not be in direct contact with each other, but may still cooperate or interact with each other.

[0073] Thus, although examples have been described in language specific to structural features and/or methodological acts, it is to be understood that claimed subject matter may not be limited to the specific features or acts described. Rather, the specific features and acts are disclosed as sample forms of implementing the claimed subject matter.

CLAIMS

What is claimed is:

1. An electronic device, comprising:
at least one processor; and
at least one storage device comprising a nonvolatile memory; and
a controller coupled to the memory and comprising logic, at least partially including hardware logic, to:
 - receive a first read request from a host device;
 - place the first read request in a read queue comprising a plurality of read requests directed to the nonvolatile memory;
 - determine a first target die and a first target plane in the nonvolatile memory for the first read request; and
 - combine the first read request with at least a second read request in the read queue to form a combined read request, wherein the second read request comprises a second target die, which is the same as the first target die, and a second target plane which is different from the first target plane.
2. The electronic device of claim 1, wherein the controller comprises logic, at least partially including hardware logic, to:
 - execute the combined read request to retrieve data from the nonvolatile memory requested in the combined read request.
3. The electronic device of claim 2, wherein the controller comprises logic, at least partially including hardware logic, to:
 - separate the data retrieved from the combined read request into to first data associated with the first read request and second data associated with the second read request.

4. The electronic device of claim 3, wherein the controller comprises logic, at least partially including hardware logic, to:

return the first data associated with the first read request to a host which generated the first read request; and

return the second data associated with the second read request to a host which generated the second read request.

5. The electronic device of claim 1, wherein the controller comprises logic, at least partially including hardware logic, to:

combine the first read request and the second read request in the read queue with at least a third read request in the read queue.

6. The electronic device of claim 1, wherein the controller comprises logic, at least partially including hardware logic, to:

maintain a logical address to physical address mapping table which maps a logical address received with a read request to a physical address in the nonvolatile memory.

7. The electronic device of claim 1, wherein one or more physical addresses in the nonvolatile memory are associated with a die and a plane in the nonvolatile memory.

8. A storage device, comprising:

a nonvolatile memory; and

a controller coupled to the memory and comprising logic, at least partially including hardware logic, to:

receive a first read request from a host device;

place the first read request in a read queue comprising a plurality of read requests directed to the nonvolatile memory;

determine a first target die and a first target plane in the nonvolatile memory for the first read request; and

combine the first read request with at least a second read request in the read queue to form a combined read request, wherein the second read request comprise a second target die, which is the same as the first target die, and a second target plane which is different from the first target plane.

9. The storage device of claim 8, wherein the controller comprises logic, at least partially including hardware logic, to:

execute the combined read request to retrieve data from the nonvolatile memory requested in the combined read request.

10. The storage device of claim 9, wherein the controller comprises logic, at least partially including hardware logic, to:

separate the data retrieved from the combined read request into to first data associated with the first read request and second data associated with the second read request.

11. The storage device of claim 10, wherein the controller comprises logic, at least partially including hardware logic, to:

return the first data associated with the first read request to a host which generated the first read request; and

return the second data associated with the second read request to a host which generated the second read request.

12. The storage device of claim 8, wherein the controller comprises logic, at least partially including hardware logic, to:

combine the first read request and the second read request in the read queue with at least a third read request in the read queue.

13. The storage device of claim 8, wherein the controller comprises logic, at least partially including hardware logic, to:

maintain a logical address to physical address mapping table which maps a

logical address received with a read request to a physical address in the nonvolatile memory.

14. The storage device of claim 8, wherein wherein or more physical addresses in the nonvolatile memory are associated with a die and a plane in the nonvolatile memory.

15. A controller comprising logic, at least partially including hardware logic, to:
receive a first read request from a host device;

place the first read request in a read queue comprising a plurality of read requests directed to the nonvolatile memory;

determine a first target die and a first target plane in the nonvolatile memory for the first read request; and

combine the first read request with at least a second read request in the read queue to form a combined read request, wherein the second read request comprise a second target die, which is the same as the first target die, and a second target plane which is different from the first target plane.

16. The controller of claim 15, wherein the controller comprises logic, at least partially including hardware logic, to:

execute the combined read request to retrieve data from the nonvolatile memory requested in the combined read request.

17. The controller of claim 16, wherein the controller comprises logic, at least partially including hardware logic, to:

separate the data retrieved from the combined read request into to first data associated with the first read request and second data associated with the second read request.

18. The controller of claim 17, wherein the controller comprises logic, at least partially including hardware logic, to:

return the first data associated with the first read request to a host which generated the first read request; and

return the second data associated with the second read request to a host which generated the second read request.

19. The controller of claim 18, wherein the controller comprises logic, at least partially including hardware logic, to:

combine the first read request and the second read request in the read queue with at least a third read request in the read queue.

20. The controller of claim 15, wherein the controller comprises logic, at least partially including hardware logic, to:

maintain a logical address to physical address mapping table which maps a logical address received with a read request to a physical address in the nonvolatile memory.

21. The controller of claim 15, wherein one or more physical addresses in the nonvolatile memory are associated with a die and a plane in the nonvolatile memory.

1/10

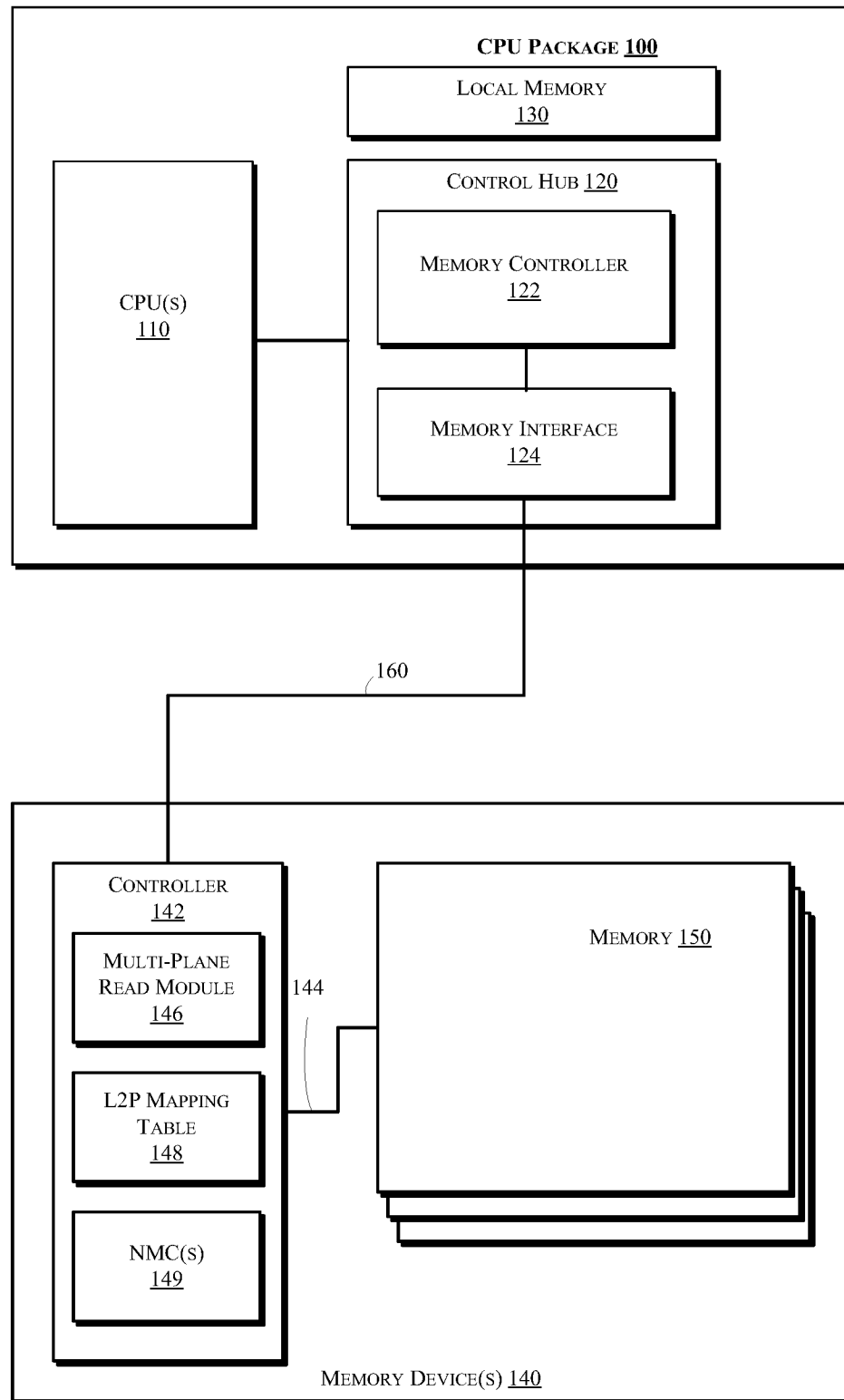


FIG. 1

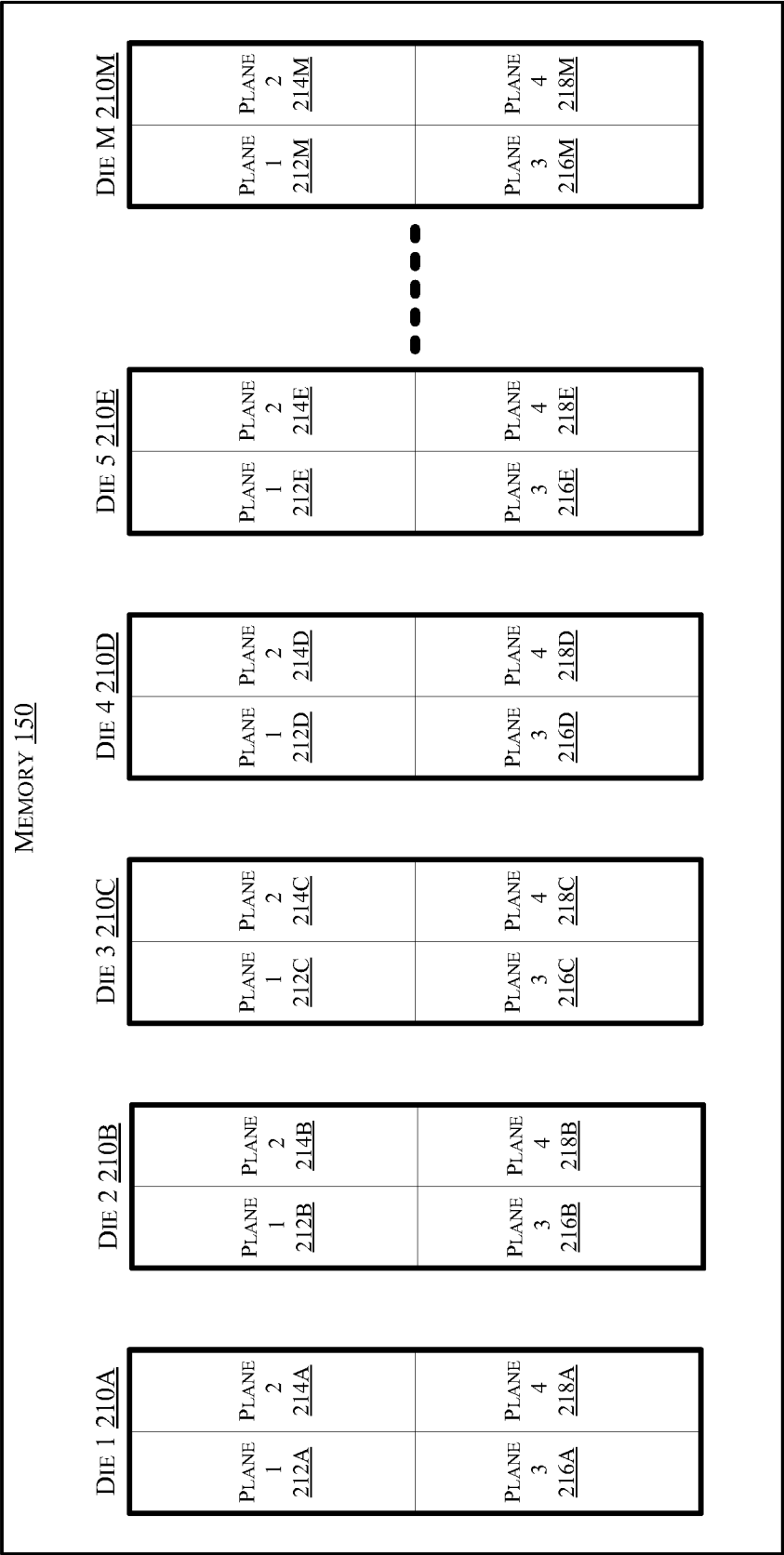


FIG. 2

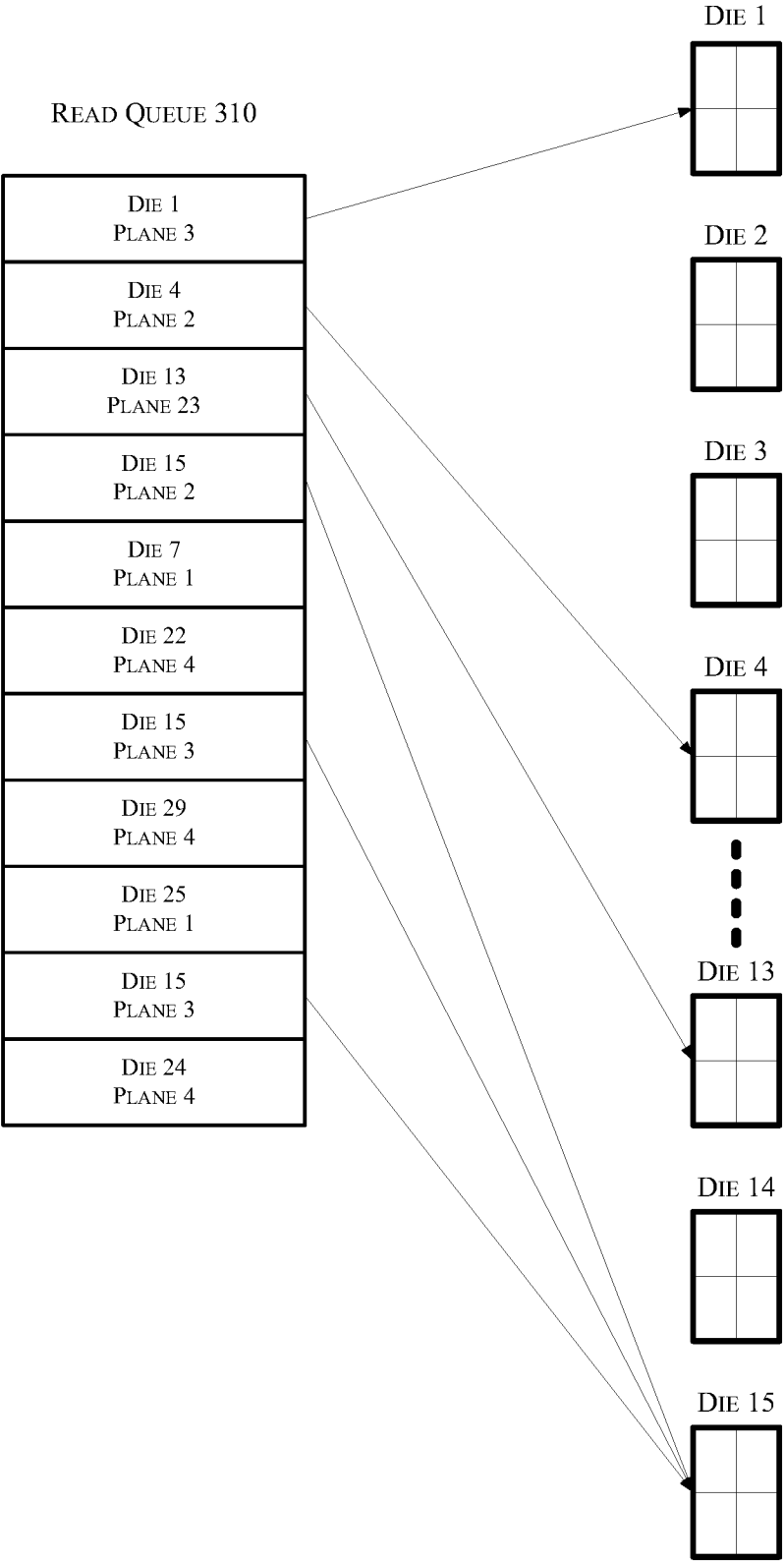


FIG. 3

4/10

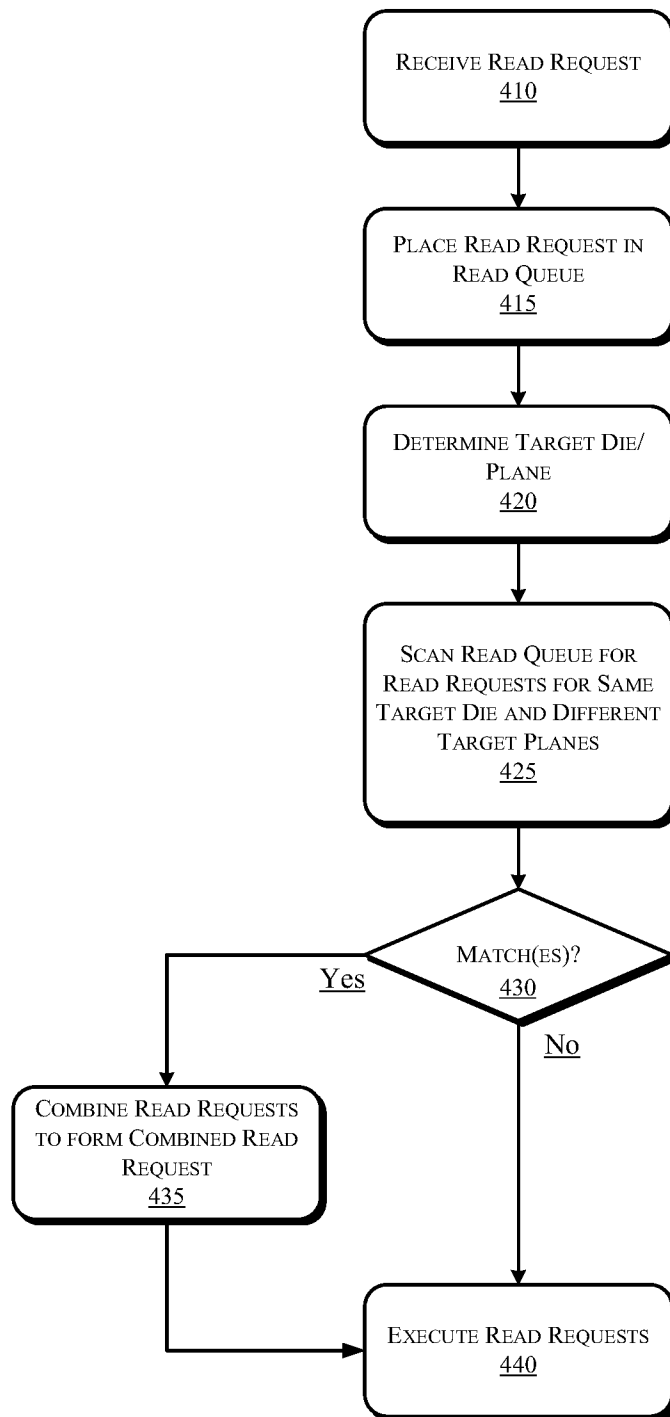


FIG. 4

5/10

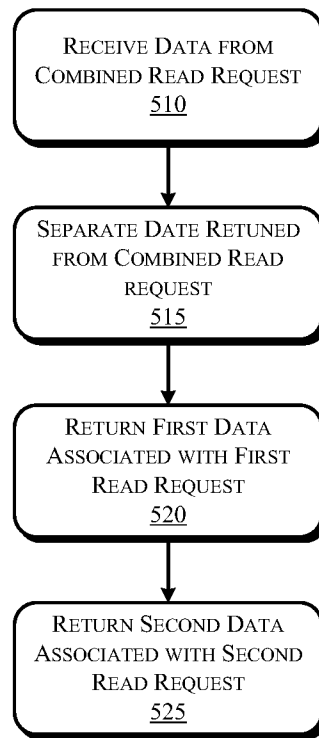
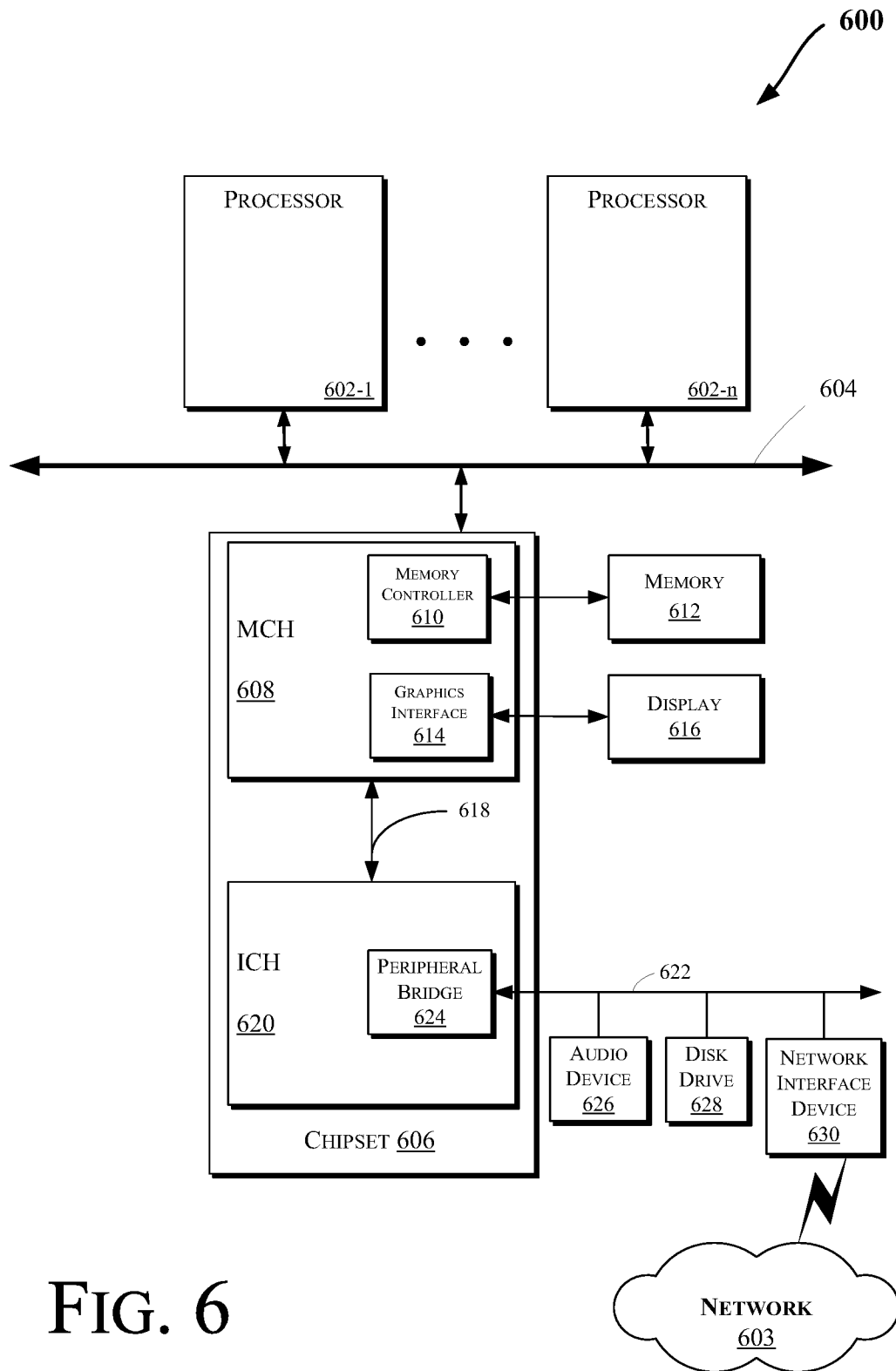


FIG. 5

6/10



7/10

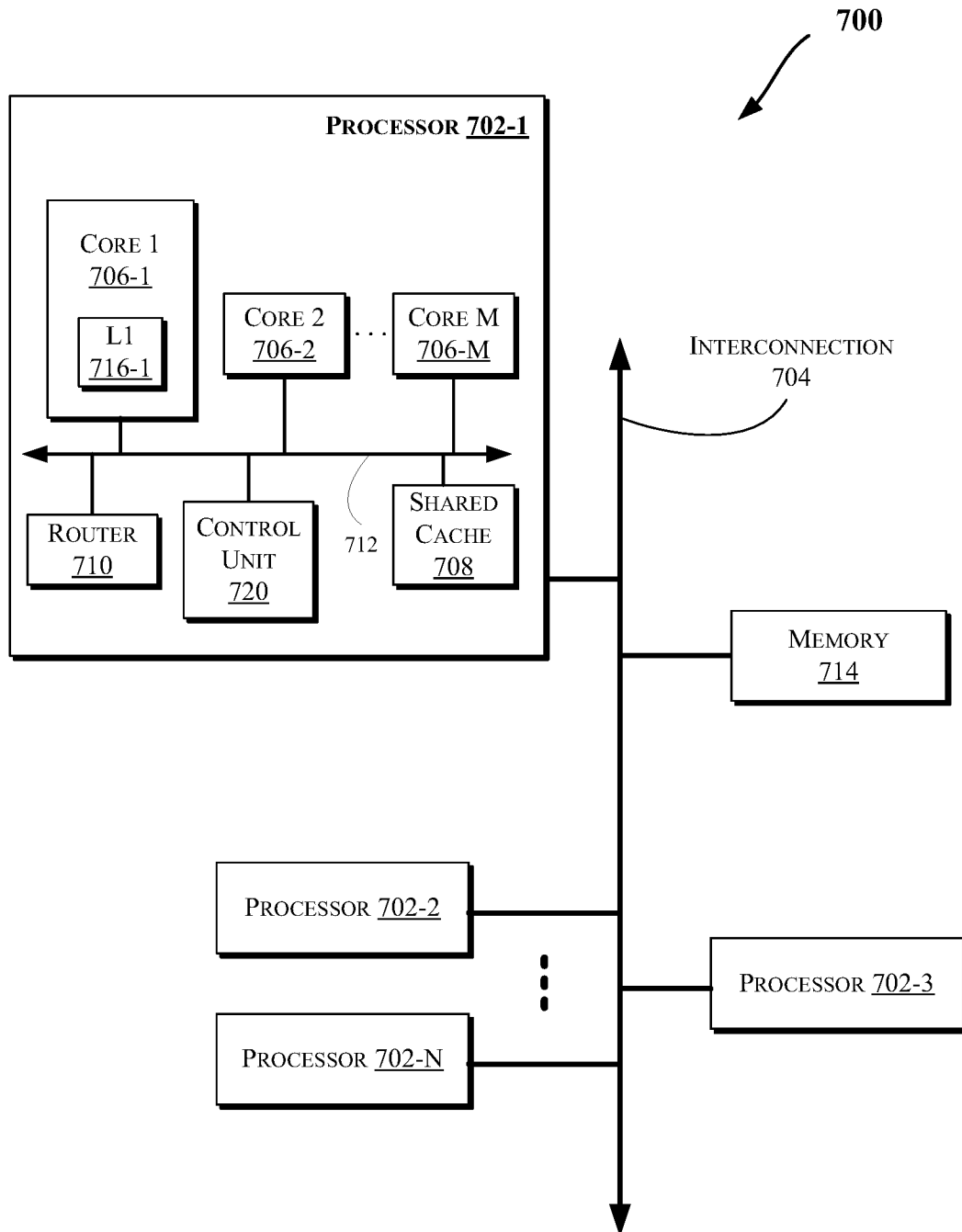


FIG. 7

8/10

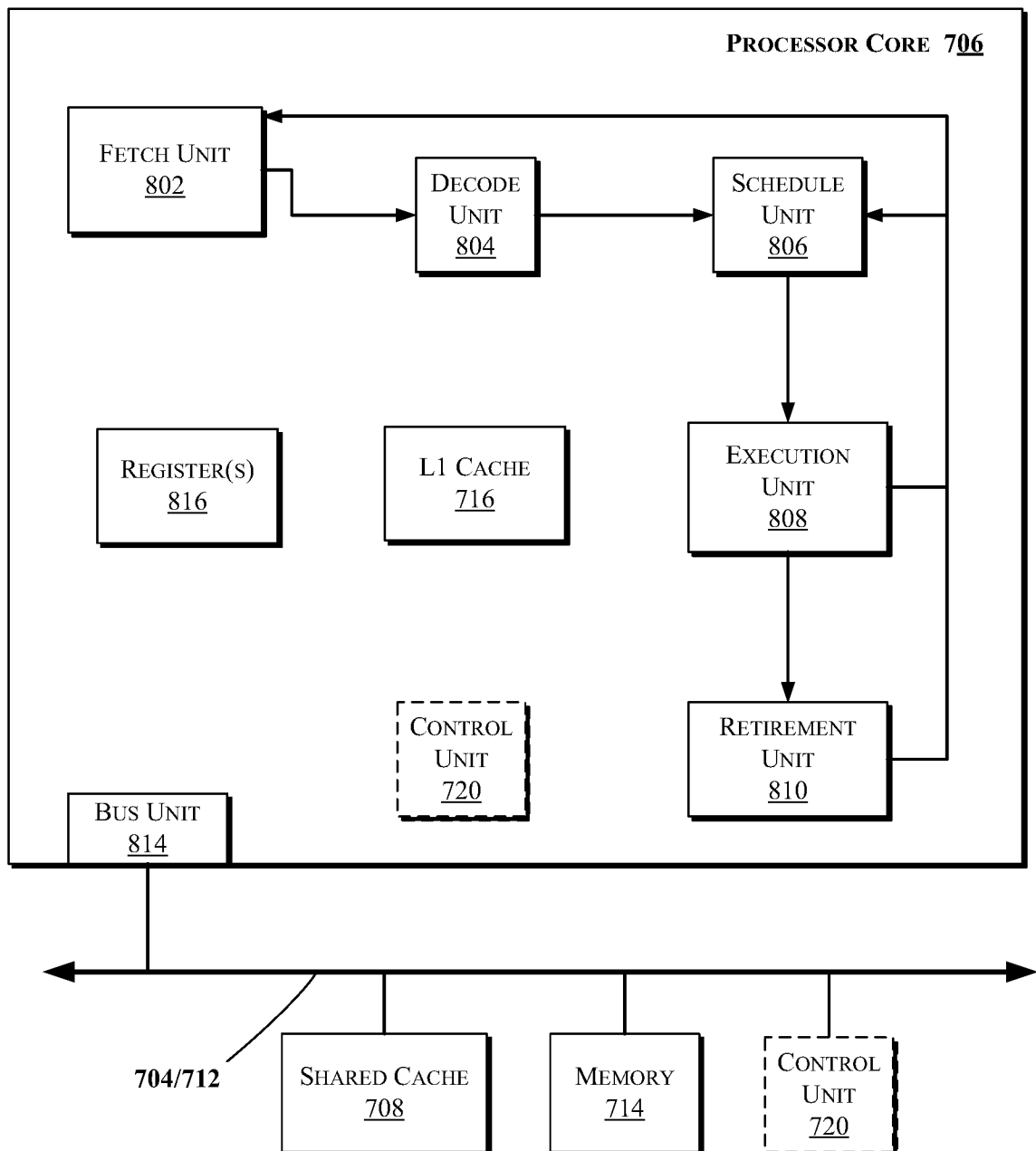


FIG. 8

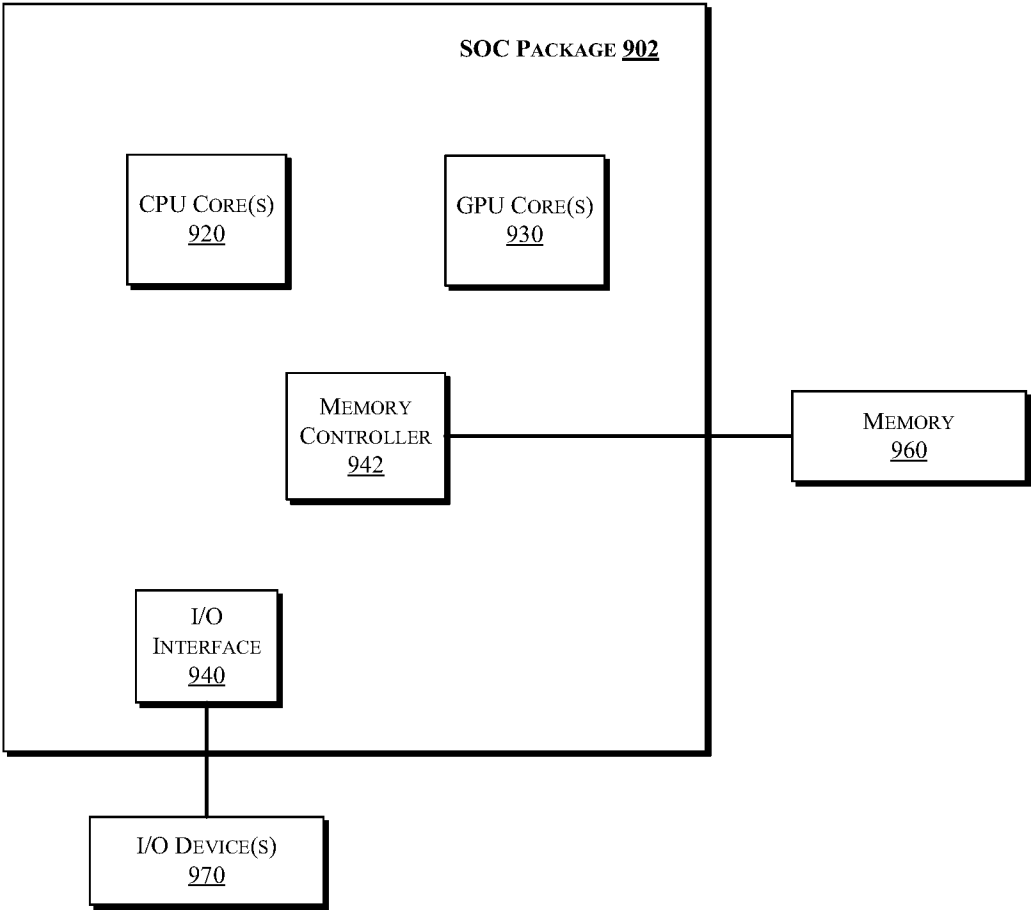


FIG. 9

10/10

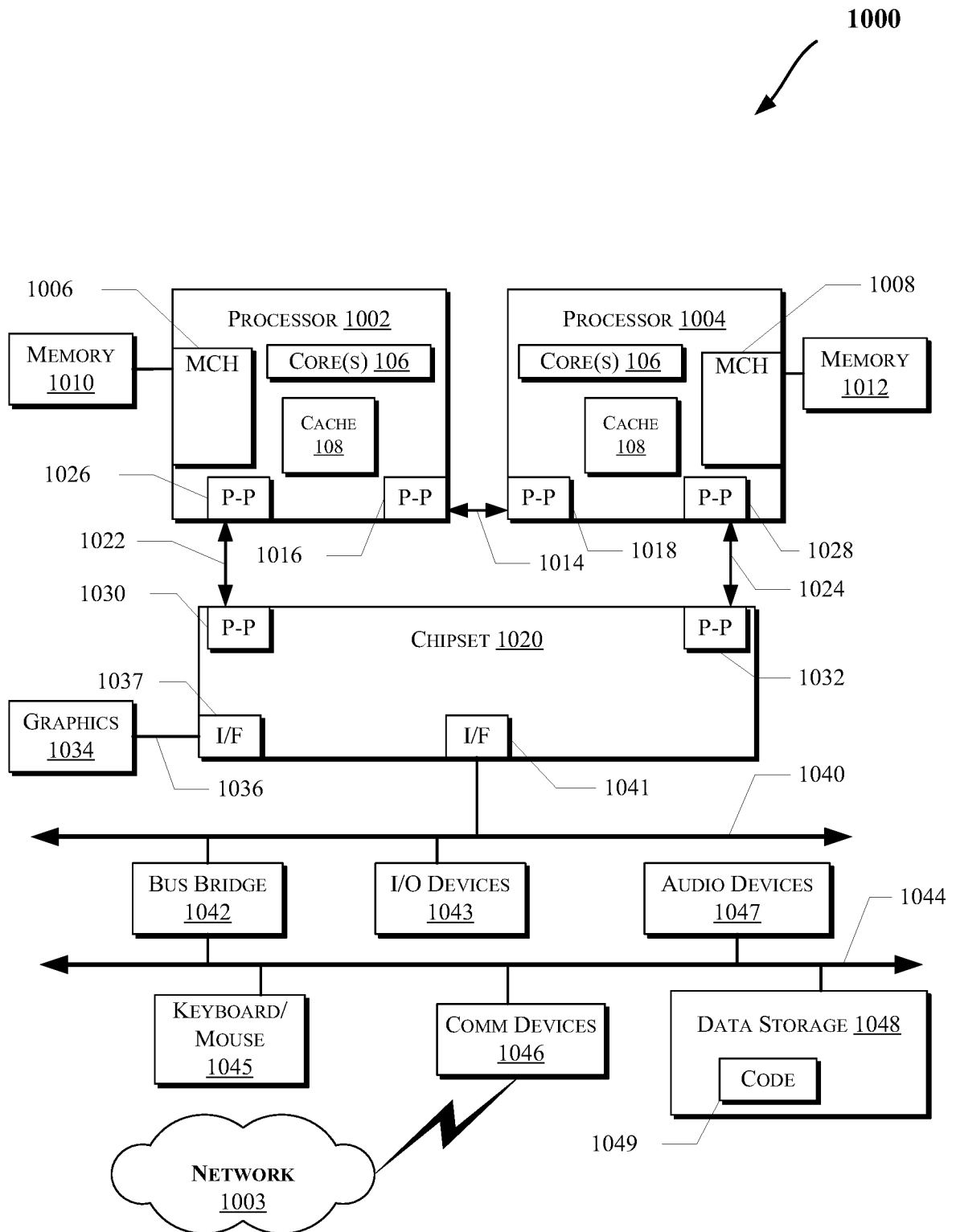


FIG. 10

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2016/019974

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F13/16
ADD.

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

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2013/262745 A1 (LIN GARY [US] ET AL) 3 October 2013 (2013-10-03) the whole document	1-21
A	----- Micron: "TN-29-25: Improving Performance Using Two-Plane Commands", 30 September 2008 (2008-09-30), XP055271241, Retrieved from the Internet: URL:www.micron.com [retrieved on 2016-05-09] the whole document -----	1-21



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

17 May 2016

Date of mailing of the international search report

25/05/2016

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Ghidini, Mario

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2016/019974

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2013262745	A1	03-10-2013	NONE
