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

(19) World Intellectual Property Organization
International Bureau

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

WIPO | PCT

(10) International Publication Number
WO 2024/253700 A1

(51) International Patent Classification:
G06F 12/0877 (2016.01)

(21) International Application Number:
PCT/US2023/078576

(22) International Filing Date:
03 November 2023 (03.11.2023)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
18/207,602 08 June 2023 (08.06.2023) US

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

(72) Inventors: **POLISHUK, Leon**; Mivtza Yonatan 6 Apt. 10, 3467701 Haifa (IL). **MANDAL, Ayan**; #285, Sobha Hibiscus Palm Avenue, Green Glen Layout, Bangaluru 560103 (IN). **JINDAL, Neetu**; 434-2, Ward 7, Adarsh Gali, New Colony, Kurukshetra 136118 (IN). **SHIFER, Eran**; Wohlman Street, 9/24, 6940226 Tel Aviv (IL). **MELAMED, Keren**; Etz HaEgoz 3a, 3057803 Binyamina (IL).

(74) Agent: **CHOI, Glen B.** et al.; Compass IP Law PC, 4804 NW Bethany Blvd, Ste. I-2 #237, Portland, Oregon 97229 (US).

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

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

(54) Title: CACHE ACCESS FABRIC

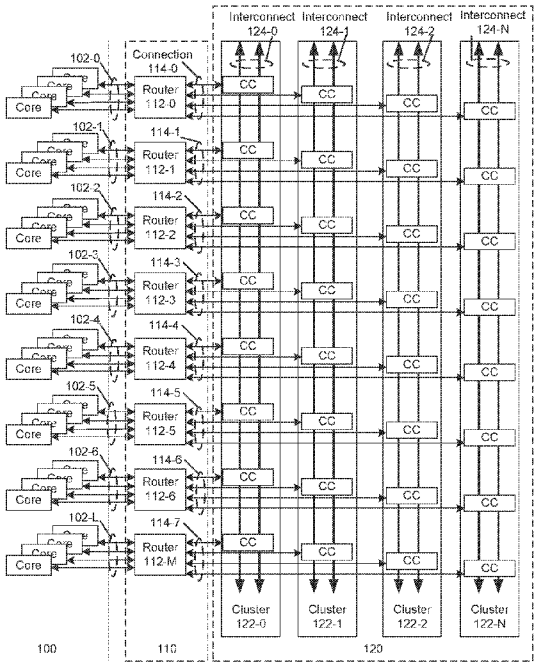

FIG. 1B

(57) Abstract: Examples described herein relate to a cache fabric that includes a first layer of a group of routers and includes a second layer of a plurality of clusters of cache controllers. A router of the group of routers can be accessible via an interface that is to receive a memory access request from a processor and select from a group of cache controllers based on a cluster identifier and memory address and provide the memory access request to the selected group of cache controllers. The selected group of cache controllers can receive the memory access request and service a memory access request from a cache device or forward the memory access request to a second cache controller associated with the cache device or a second cache device.

WO 2024/253700 A1

[Continued on next page]

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

CACHE ACCESS FABRIC

CLAIM OF PRIORITY

[0001] This application claims priority under 35 U.S.C. § 365(c) to U.S. Application No. 18/207,602, filed June 8, 2023, the contents of which is hereby incorporated by reference in its entirety.

BACKGROUND

[0002] With constant growth of a number of cores in a server, fabrics that communicatively couple cores to cache devices have increasing bandwidth requirements. A mesh fabric can allow connection of cores to a cache device to access an entirety of the cache. For example, the mesh fabric allows a Last Level Cache (LLC) to be accessible by connected cores.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] FIGs. 1A and 1B depict example systems.

[0004] FIG. 2 depicts an example system.

[0005] FIGs. 3A and 3B depict example systems.

[0006] FIG. 4 depicts an example process.

[0007] FIG. 5 depicts an example system.

DETAILED DESCRIPTION

[0008] Power usage of mesh fabrics can increase as mesh fabrics connect increasing numbers of processors to cache devices. At least to potentially reduce or limit power usage of a cache fabric that connects processor cores to one or more cache devices, various examples provide a cache fabric with a first layer of a group of routers and a second layer of a plurality of clusters of cache controllers. A router of the group of routers can be accessible via an interface that is to receive a memory access request from a processor and select from a group of cache controllers based on a cluster identifier and memory address and provide the memory access request to the selected group of cache controllers. The selected group of cache controllers can receive the memory access request and service a memory access request from a cache device or forward the memory access request to a second cache controller associated with the cache device or a second cache device.

[0009] FIG. 1A depicts an example overview of a system. Processors 100 can include one or more processor cores configured to process a specific instruction set. In some examples, instruction sets may facilitate Complex Instruction Set Computing (CISC), Reduced Instruction

Set Computing (RISC), or computing via a Very Long Instruction Word (VLIW). Processors 100 may process different instruction sets, which may include instructions to facilitate the emulation of other instruction sets. Processors 100 may also include other processing devices, such as a Digital Signal Processor (DSP). Processors 100 can be part of a central processing unit (CPU) or graphics processing unit (GPU).

[0010] A register file (not shown) can be additionally included in processors 100 and may include different types of registers for storing different types of data (e.g., integer registers, floating point registers, status registers, and an instruction pointer register). Some registers may be general-purpose registers. In some examples, one or more of the cores can include a memory controller (not shown) that is to provide memory access request communications to a memory device and other components (e.g., routers 110). In some examples, a memory access request can specify one or more of: an virtual or physical address associated with a memory device (e.g., memory 130) to read-from or write-to or a cache cluster identifier to identify a cluster among cache clusters 120.

[0011] In some examples, one or more of processors 100 can be coupled with one or more interface buses 102-0 to 102-L, where L is an integer, to transmit communication signals such as address, data, or control signals between a core and one or more of routers of routers 110. The one or more interface buses 102-0 to 102-L can include a processor bus, such as a version of the Direct Media Interface (DMI) bus. However, processor busses are not limited to the DMI bus, and may include one or more Peripheral Component Interconnect interfaces (e.g., Peripheral Component Interconnect express (PCIe), Universal Chiplet Interconnect Express (UCIe)), memory busses or interfaces (e.g., Compute Express Link (CXL), or other types of interfaces.

[0012] Cache clusters 120 can include one or more cache controllers and cache (e.g., one or more registers, one or more cache devices (e.g., level 1 cache (L1), level 2 cache (L2), level 3 cache (L3), last level cache (LLC)), volatile memory device, non-volatile memory device, or persistent memory) (not shown), which may be shared among processors 100 using cache coherency techniques. If data referenced by a memory access request is stored in a cache device (e.g., cache hit), the cache device can provide the data to a requester processor. If data referenced by a memory access request is not stored in a cache device (e.g., cache miss), the cache controller or cache device can forward the request to a memory controller of memory device 130. A home agent (HA) or caching and home agent (CHA) (not shown) can perform data coherency among cache devices and memory devices. In some examples, for particular addressable memory ranges accessible to particular CPUs, HA can attempt to achieve data consistency among the memory devices and caches. Memory device 130 can include one or more volatile or non-volatile memory devices. Memory 130 can provide the data to a requester processor via cache clusters 120 or

another connection.

[0013] Communications between components of FIG. 1A can take place using conductors that provide die-to-die communications; chip-to-chip communications; circuit board-to-circuit board communications; and/or package-to-package communications. A die-to-die communications can utilize Embedded Multi-Die Interconnect Bridge (EMIB) or an interposer. Components of FIG. 1A can be enclosed in one or more semiconductor packages. A semiconductor package can include metal, plastic, glass, and/or ceramic casing that encompass and provide communications within or among one or more semiconductor devices or integrated circuits. Various examples can be implemented in a die, in a package, or between multiple packages, in a server, or among multiple servers.

[0014] FIG. 1B depicts an example system. One or more cores of processors 100 can provide a memory access request to an associated router among routers 112-0 to 112-M, where M is an integer. In some examples, a source core can calculate a destination cache controller and cache device that manages a specific address associated with the memory access request. For example, routers 112-0 to 112-M can perform decoding of memory access requests to identify a cluster among clusters 122-0 to 122-N, where N is an integer, that corresponds to a cluster of cache controllers and cache devices (e.g., cluster 122-0 to 122-N, where N is an integer) that potentially store data associated with a memory address provided by the memory access request. A router can utilize a cross bar, one or more multiplexers, or other interface to forward the memory access request to the identified cluster.

[0015] Clusters 120 can provide an interconnect between the output of routers 110 and cache controllers and cache devices and, potentially, memory devices. One or more of clusters 122-0 to 122-N can include one or more cache controllers and cache devices (CC). A cluster can be mapped to a subset of memory addresses, based on hash function or other allocation, to allocate memory addresses or spans of contiguous memory addresses to particular clusters. In some examples, the receiver cache controller of a CC can manage a specific address associated with the memory access request. A cache controller of a CC can receive the memory access request and determine if the cache controller manages memory addresses in a range that includes a memory address associated with the memory access request or another cache controller manages memory addresses in a range that includes a memory address associated with the memory access request. Based on the cache controller managing memory addresses in a range that includes a memory address associated with the memory access request, the cache controller can determine if an associated cache stores or does not store data associated with the memory access request. For a cache hit, the cache controller can provide the data to the requester core. In the case of a cache hit, data can traverse a path to the

core via an interconnect through the cluster (e.g., one of 124-0 to 124-N) and then through the originating router (e.g., one of 112-0 to 122-M) to the requestor core. For a cache miss, the cache controller can provide the memory access request, or portion thereof, to a home agent or caching and home agent (not shown) to retrieve and provide the data to the source core. A home agent or caching and home agent (not shown) can perform data coherency among cache devices and memory devices so that data provided to a requester core is a current version.

[0016] FIG. 2 depicts an example router circuitry. Decoder 202 can receive a memory access request and based on a configuration from an operating system (OS), system administrator, orchestrator, or other entity, select an output line connected to a target cluster based on an address associated with a memory access request. For example, decoder 202 can access a configuration that specifies: a cluster and output line for a particular virtual or physical memory address range or values. In some examples, a core can specify a memory address and cluster number and decoder 202 can select an output line in crossbar circuitry 204 to the target cluster based on address and cluster number.

[0017] Crossbar circuitry 204 can connect one or more of a number of cores to a number of different cache controllers in different clusters. For example, crossbar circuitry 204 can connect one or more of 4 cores to 4 different cache controllers in different clusters. Crossbar circuitry 204 can include one or more 4x4 crossbars, four 2x2 crossbars, or other configurations. Crossbar circuitry 204 can enqueue received memory access requests prior to output to a target cluster and output memory access requests based on ordering (e.g., ordering of requests received from different transfers such as first in first out), arbitration (e.g., round robin selection from queues or source cores, round robin selection from queues or source cores without head of line blocking, higher priority memory access requests are released before lower priority memory access requests), or other policies. Crossbar circuitry 204 can enqueue received responses to memory access requests (e.g., data or other control messages) prior to output to a core based on ordering (e.g., first in first out), arbitration (e.g., higher priority memory access requests are released before lower priority memory access requests), or other policies.

[0018] FIG. 3A depicts an example cache controller system. Input 302 can include a U2C (e.g., uncore to core) circuitry to provide communications (e.g., memory access requests, data, snoops, responses, interrupts, or control signals) to a router or crossbar for forwarding to a processor via an interface. Input 302 can include a C2U (e.g., core to uncore) circuitry to provide communications (e.g., memory access requests, data, snoops, responses, interrupts, or control signals) from a router or crossbar to connections 304 via an interface. An uncore (e.g., system agent) can include or more of a memory controller, a shared cache (e.g., last level cache (LLC)),

a cache coherency manager, arithmetic logic units, floating point units, core or processor interconnects, Caching/Home Agent (CHA), interface circuitry (e.g., fabric, memory, device), and/or bus or link controllers. An uncore can provide one or more of: direct memory access (DMA) engine connection, non-cached coherent master connection, data cache coherency between cores and arbitrates cache requests, or Advanced Microcontroller Bus Architecture (AMBA) capabilities.

[0019] Connections 304 can provide communication between different cache controllers of a cluster using an interface. For example, one or more interfaces can operate in a manner consistent with Inter-Integrated Circuit (I2C) Protocol, Serial Peripheral Interface (SPI), or other protocols. At ingress, cache controller and cache 308 can receive memory access requests or snoops from processors. A snoop can be used to determine if cache coherency is maintained among cache devices and identify whether cache devices store content identified by a memory address. At egress, cache controller and cache 308 can forward memory access requests to another cache controller and cache in a same cluster, provide the memory access request to a home agent or caching and home agent, or provide data from cache 312 to be sent to a processor that issued a memory access request for the data. Cache controller 310 can include or access HA or CHA circuitry to perform data coherence checks to identify and request and receive revised data from other cache devices or memory devices. Cache controller 310 can utilize scoreboard circuitry to track received memory access requests, attempt to maintain order of operations, and track status of memory access requests or snoops (e.g., completed, forwarded, incomplete, time out (e.g., no response received), or others).

[0020] FIG. 3B depicts an example cache controller system. The operation of cache controller and cache (e.g., LLC) can be similar to that of the system of FIG. 3A.

[0021] FIG. 4 depicts an example process. The process can be performed by a processor using a group of routers and a plurality of clusters of cache controllers. At 402, a processor can issue memory access request to an associated router. In some examples, the memory access request is associated with one or more of: a read or write request, memory address, and target cluster. At 404, the router can route the memory access request to a target cluster. In some examples, the router can be one of a plurality of routers, where one or more of the plurality of routers is coupled to different cache controllers in different clusters. For example, the router can select a particular output line that is connected to the target cluster to forward the memory access request to a cache controller in the target cluster.

[0022] At 406, a determination can be made if the cache controller in the target cluster can service the memory access request. For example, the cache controller in the target cluster can

service the memory access request based on the memory access request being associated with an address assigned to the cache controller. Based on the cache controller in the target cluster servicing the memory access request, the process can proceed to 408. At 408, based on a cache hit in a cache device associated with the cache controller, the cache controller can service the memory access request by providing data from the cache device to the requester processor.

[0023] At 406, based on the cache controller in the target cluster not servicing the memory access request, the process can proceed to 410. At 410, based on a cache miss in a cache device associated with the cache controller, the cache controller can forward the memory access request to a home agent (HA) or caching and home agent (CHA) to manage access of the data for the memory access request from a memory or storage device. In some examples, at 410, the cache controller in the target cluster can forward the memory access request to a second cache controller in the same cluster using an output line in the target cluster for the second cache controller to service the memory access request. For example, the target cluster can include one or more output lines that communicatively couple together different cache controllers.

[0024] FIG. 5 depicts a system. The system can use a multiple routers and multiple cluster systems to forward memory access requests, as described herein. System 500 includes processors 510, which provides processing, operation management, and execution of instructions for system 500. Processors 510 can include any type of microprocessor, central processing unit (CPU), graphics processing unit (GPU), XPU, processing core, or other processing hardware to provide processing for system 500, or a combination of processors. An XPU can include one or more of: a CPU, a graphics processing unit (GPU), general purpose GPU (GPGPU), and/or other processing units (e.g., accelerators or programmable or fixed function FPGAs). Processors 510 controls the overall operation of system 500, and can be or include, one or more programmable general-purpose or special-purpose microprocessors, digital signal processors (DSPs), programmable controllers, application specific integrated circuits (ASICs), programmable logic devices (PLDs), or the like, or a combination of such devices. Processors 510 can include one or more processor sockets.

[0025] In some examples, interface 512 and/or interface 514 can include a switch (e.g., CXL switch) that provides device interfaces between processors 510 and other devices (e.g., memory subsystem 520, graphics 540, accelerators 542, network interface 550, and so forth).

[0026] In one example, system 500 includes interface 512 coupled to processors 510, which can represent a higher speed interface or a high throughput interface for system components that needs higher bandwidth connections, such as memory subsystem 520 or graphics interface components 540, or accelerators 542. Interface 512 represents an interface circuit, which can be a standalone component or integrated onto a processor die. In some examples, interface 512 and/or

interface 512 can include a multiple routers and multiple cluster systems to forward memory access requests to memory subsystem 520 and/or storage subsystem 580, as described herein.

[0027] Accelerators 542 can be a programmable or fixed function offload engine that can be accessed or used by a processors 510. For example, an accelerator among accelerators 542 can provide compression (DC) capability, cryptography services such as public key encryption (PKE), cipher, hash/authentication capabilities, decryption, or other capabilities or services. In some embodiments, in addition or alternatively, an accelerator among accelerators 542 provides field select controller capabilities as described herein. In some cases, accelerators 542 can be integrated into a CPU socket (e.g., a connector to a motherboard or circuit board that includes a CPU and provides an electrical interface with the CPU). For example, accelerators 542 can include a single or multi-core processor, graphics processing unit, logical execution unit single or multi-level cache, functional units usable to independently execute programs or threads, application specific integrated circuits (ASICs), neural network processors (NNPs), programmable control logic, and programmable processing elements such as field programmable gate arrays (FPGAs). Accelerators 542 can provide multiple neural networks, CPUs, processor cores, general purpose graphics processing units, or graphics processing units can be made available for use by artificial intelligence (AI) or machine learning (ML) models. For example, the AI model can use or include any or a combination of: a reinforcement learning scheme, Q-learning scheme, deep-Q learning, or Asynchronous Advantage Actor-Critic (A3C), combinatorial neural network, recurrent combinatorial neural network, or other AI or ML model. Multiple neural networks, processor cores, or graphics processing units can be made available for use by AI or ML models.

[0028] Memory subsystem 520 represents the main memory of system 500 and provides storage for code to be executed by processors 510, or data values to be used in executing a routine. Memory subsystem 520 can include one or more memory devices 530 such as read-only memory (ROM), flash memory, one or more varieties of random access memory (RAM) such as DRAM, or other memory devices, or a combination of such devices. Memory 530 stores and hosts, among other things, operating system (OS) 532 to provide a software platform for execution of instructions in system 500. Additionally, applications 534 can execute on the software platform of OS 532 from memory 530. Applications 534 represent programs that have their own operational logic to perform execution of one or more functions. Applications 534 and/or processes 536 can refer instead or additionally to a virtual machine (VM), container, microservice, processor, or other software. Processes 536 represent agents or routines that provide auxiliary functions to OS 532 or one or more applications 534 or a combination. OS 532, applications 534, and processes 536 provide software logic to provide functions for system 500. In one example, memory subsystem

520 includes memory controller 522, which is a memory controller to generate and issue commands to memory 530. Memory controller 522 could be a physical part of processors 510 or a physical part of interface 512. For example, memory controller 522 can be an integrated memory controller, integrated onto a circuit with processors 510.

[0029] In some examples, OS 532 can be Linux®, Windows® Server or personal computer, FreeBSD®, Android®, MacOS®, iOS®, VMware vSphere, openSUSE, RHEL, CentOS, Debian, Ubuntu, or any other operating system. The OS and driver can execute on one or more processors sold or designed by Intel®, ARM®, AMD®, Qualcomm®, IBM®, Nvidia®, Broadcom®, Texas Instruments®, among others. In some examples, OS 532 and/or a driver can configure multiple routers and multiple cluster systems to forward memory access requests to memory subsystem 520 or storage subsystem 580, as described herein.

[0030] While not specifically illustrated, it will be understood that system 500 can include one or more buses or bus systems between devices, such as a memory bus, a graphics bus, interface buses, or others. Buses or other signal lines can communicatively or electrically couple components together, or both communicatively and electrically couple the components. Buses can include physical communication lines, point-to-point connections, bridges, adapters, controllers, or other circuitry or a combination. Buses can include, for example, one or more of a system bus, a Peripheral Component Interconnect (PCI) bus, a Hyper Transport or industry standard architecture (ISA) bus, a small computer system interface (SCSI) bus, a universal serial bus (USB), or an Institute of Electrical and Electronics Engineers (IEEE) standard 1394 bus (Firewire).

[0031] In one example, system 500 includes interface 514, which can be coupled to interface 512. In one example, interface 514 represents an interface circuit, which can include standalone components and integrated circuitry. In one example, multiple user interface components or peripheral components, or both, couple to interface 514. Network interface 550 provides system 500 the ability to communicate with remote devices (e.g., servers or other computing devices) over one or more networks. Network interface 550 can include an Ethernet adapter, wireless interconnection components, cellular network interconnection components, USB (universal serial bus), or other wired or wireless standards-based or proprietary interfaces. Network interface 550 can transmit data to a device that is in the same data center or rack or a remote device, which can include sending data stored in memory.

[0032] In some examples, network interface 550 can be implemented as a network interface controller, network interface card, a host fabric interface (HFI), or host bus adapter (HBA), and such examples can be interchangeable. Network interface 550 can be coupled to one or more servers using a bus, PCIe, CXL, or DDR. Network interface 550 may be embodied as part of a

system-on-a-chip (SoC) that includes one or more processors, or included on a multichip package that also contains one or more processors.

[0033] Some examples of network device 550 are part of an Infrastructure Processing Unit (IPU) or data processing unit (DPU) or utilized by an IPU or DPU. An xPU can refer at least to an IPU, DPU, GPU, GPGPU, or other processing units (e.g., accelerator devices). An IPU or DPU can include a network interface with one or more programmable pipelines or fixed function processors to perform offload of operations that could have been performed by a CPU. The IPU or DPU can include one or more memory devices. In some examples, the IPU or DPU can perform virtual switch operations, manage storage transactions (e.g., compression, cryptography, virtualization), and manage operations performed on other IPUs, DPUs, servers, or devices.

[0034] In one example, system 500 includes one or more input/output (I/O) interface(s) 560. Peripheral interface 570 can include any hardware interface not specifically mentioned above. Peripherals refer generally to devices that connect dependently to system 500. A dependent connection is one where system 500 provides the software platform or hardware platform or both on which operation executes.

[0035] In one example, system 500 includes storage subsystem 580 to store data in a nonvolatile manner. In one example, in certain system implementations, at least certain components of storage 580 can overlap with components of memory subsystem 520. Storage subsystem 580 includes storage device(s) 584, which can be or include any conventional medium for storing large amounts of data in a nonvolatile manner, such as one or more magnetic, solid state, or optical based disks, or a combination. Storage 584 holds code or instructions and data 586 in a persistent state (e.g., the value is retained despite interruption of power to system 500). Storage 584 can be generically considered to be a "memory," although memory 530 is typically the executing or operating memory to provide instructions to processors 510. Whereas storage 584 is nonvolatile, memory 530 can include volatile memory (e.g., the value or state of the data is indeterminate if power is interrupted to system 500). In one example, storage subsystem 580 includes controller 582 to interface with storage 584. In one example controller 582 is a physical part of interface 514 or processors 510 or can include circuits or logic in processors 510 and interface 514.

[0036] In an example, system 500 can be implemented using interconnected compute sleds of processors, memories, storages, network interfaces, and other components. High speed interconnects can be used such as: Ethernet (IEEE 802.3), remote direct memory access (RDMA), InfiniBand, Internet Wide Area RDMA Protocol (iWARP), Transmission Control Protocol (TCP), User Datagram Protocol (UDP), quick UDP Internet Connections (QUIC), RDMA over

Converged Ethernet (RoCE), Peripheral Component Interconnect express (PCIe), Intel QuickPath Interconnect (QPI), Intel Ultra Path Interconnect (UPI), Intel On-Chip System Fabric (IOSF), Omni-Path, Compute Express Link (CXL), HyperTransport, high-speed fabric, NVLink, Advanced Microcontroller Bus Architecture (AMBA) interconnect, OpenCAPI, Gen-Z, Infinity Fabric (IF), Cache Coherent Interconnect for Accelerators (CCIX), 3GPP Long Term Evolution (LTE) (4G), 3GPP 5G, and variations thereof. Data can be copied or stored to virtualized storage nodes or accessed using a protocol such as Non-volatile Memory Express (NVMe) over Fabrics (NVMe-oF) or NVMe.

[0037] In some examples, system 500 can be implemented using interconnected compute nodes of processors, memories, storages, network interfaces, and other components. High speed interconnects can be used such as PCIe, Ethernet, or optical interconnects (or a combination thereof).

[0038] Embodiments herein may be implemented in various types of computing and networking equipment, such as switches, routers, racks, and blade servers such as those employed in a data center and/or server farm environment. The servers used in data centers and server farms comprise arrayed server configurations such as rack-based servers or blade servers. These servers are interconnected in communication via various network provisions, such as partitioning sets of servers into Local Area Networks (LANs) with appropriate switching and routing facilities between the LANs to form a private Intranet. For example, cloud hosting facilities may typically employ large data centers with a multitude of servers. A blade comprises a separate computing platform that is configured to perform server-type functions, that is, a "server on a card." Accordingly, each blade includes components common to conventional servers, including a main printed circuit board (main board) providing internal wiring (e.g., buses) for coupling appropriate integrated circuits (ICs) and other components mounted to the board.

[0039] Various examples may be implemented using hardware elements, software elements, or a combination of both. In some examples, hardware elements may include devices, components, processors, microprocessors, circuits, circuit elements (e.g., transistors, resistors, capacitors, inductors, and so forth), integrated circuits, ASICs, PLDs, DSPs, FPGAs, memory units, logic gates, registers, semiconductor device, chips, microchips, chip sets, and so forth. In some examples, software elements may include software components, programs, applications, computer programs, application programs, system programs, machine programs, operating system software, middleware, firmware, software modules, routines, subroutines, functions, methods, procedures, software interfaces, APIs, instruction sets, computing code, computer code, code segments, computer code segments, words, values, symbols, or any combination thereof.

[0040] Some examples may be implemented using or as an article of manufacture or at least one computer-readable medium. A computer-readable medium may include a non-transitory storage medium to store logic. In some examples, the non-transitory storage medium may include one or more types of computer-readable storage media capable of storing electronic data, including volatile memory or non-volatile memory, removable or non-removable memory, erasable or non-erasable memory, writeable or re-writeable memory, and so forth. In some examples, the logic may include various software elements, such as software components, programs, applications, computer programs, application programs, system programs, machine programs, operating system software, middleware, firmware, software modules, routines, subroutines, functions, methods, procedures, software interfaces, API, instruction sets, computing code, computer code, code segments, computer code segments, words, values, symbols, or any combination thereof.

[0041] One or more aspects of at least one example may be implemented by representative instructions stored on at least one machine-readable medium which represents various logic within the processor, which when read by a machine, computing device or system causes the machine, computing device or system to fabricate logic to perform the techniques described herein. Such representations, known as "IP cores" may be stored on a tangible, machine readable medium and supplied to various customers or manufacturing facilities to load into the fabrication machines that actually make the logic or processor.

[0042] The appearances of the phrase "one example" or "an example" are not necessarily all referring to the same example or embodiment. Any aspect described herein can be combined with any other aspect or similar aspect described herein, regardless of whether the aspects are described with respect to the same figure or element. Division, omission or inclusion of block functions depicted in the accompanying figures does not infer that the hardware components, circuits, software and/or elements for implementing these functions would necessarily be divided, omitted, or included in embodiments.

[0043] Some examples may be described using the expression "coupled" and "connected" along with their derivatives. These terms are not necessarily intended as synonyms for each other. For example, descriptions using the terms "connected" and/or "coupled" may indicate that two or more elements are in direct physical or electrical contact with each other. The term "coupled," however, may also mean that two or more elements are not in direct contact with each other, but yet still co-operate or interact with each other.

[0044] The terms "first," "second," and the like, herein do not denote any order, quantity, or importance, but rather are used to distinguish one element from another. The terms "a" and "an" herein do not denote a limitation of quantity, but rather denote the presence of at least one of the

referenced items. The term "asserted" used herein with reference to a signal denote a state of the signal, in which the signal is active, and which can be achieved by applying any logic level either logic 0 or logic 1 to the signal. The terms "follow" or "after" can refer to immediately following or following after some other event or events. Other sequences of operations may also be performed according to alternative embodiments. Furthermore, additional operations may be added or removed depending on the particular applications. Any combination of changes can be used and one of ordinary skill in the art with the benefit of this disclosure would understand the many variations, modifications, and alternative embodiments thereof.

[0045] Disjunctive language such as the phrase "at least one of X, Y, or Z," unless specifically stated otherwise, is otherwise understood within the context as used in general to present that an item, term, etc., may be either X, Y, or Z, or any combination thereof (e.g., X, Y, and/or Z). Thus, such disjunctive language is not generally intended to, and should not, imply that certain embodiments require at least one of X, at least one of Y, or at least one of Z to each be present. Additionally, conjunctive language such as the phrase "at least one of X, Y, and Z," unless specifically stated otherwise, should also be understood to mean X, Y, Z, or any combination thereof, including "X, Y, and/or Z."'

[0046] Illustrative examples of the devices, systems, and methods disclosed herein are provided below. An embodiment of the devices, systems, and methods may include any one or more, and any combination of, the examples described below.

[0047] Flow diagrams as illustrated herein provide examples of sequences of various process actions. The flow diagrams can indicate operations to be executed by a software or firmware routine, as well as physical operations. In some embodiments, a flow diagram can illustrate the state of a finite state machine (FSM), which can be implemented in hardware and/or software. Although shown in a particular sequence or order, unless otherwise specified, the order of the actions can be modified. Thus, the illustrated embodiments should be understood only as an example, and the process can be performed in a different order, and some actions can be performed in parallel. Additionally, one or more actions can be omitted in various embodiments; thus, not all actions are required in every embodiment. Other process flows are possible.

[0048] Example 1 includes one or more examples and includes an apparatus that includes: a semiconductor package comprising: a plurality of cores and a fabric to provide the plurality of cores with access to at least one cache device, wherein: the fabric comprises a group of routers and a plurality of groups of cache controllers, the fabric comprises a group of connections between the group of routers and the plurality of groups of cache controllers, the fabric comprises a second group of connections between cache controllers in a group of cache controllers, and based on

receipt of a memory access request from a core of the plurality of cores: a router of the group of routers is to select a group from among the plurality of groups of cache controllers based on the memory access request and forward the memory access request to the selected group and a cache controller in the selected group is to: based on data associated with the memory access request stored in the at least one cache device, service the forwarded memory access request at a selected cache device of the at least one cache device and based on data associated with the memory access request managed by a second cache controller, forward the forwarded memory access request to the second cache controller in the selected group.

[0049] Example 2 includes one or more examples, wherein the group of connections and the second group of connections are physically arranged in different directions.

[0050] Example 3 includes one or more examples, wherein the memory access request comprises a memory address and identifier of a group of cache controllers of the plurality of groups of cache controllers to which to forward the memory access request.

[0051] Example 4 includes one or more examples, and includes the at least one cache device communicatively coupled to the fabric.

[0052] Example 5 includes one or more examples, wherein based on a cache miss in the at least one cache device, the second cache controller is to request data associated with the memory access request from a memory device.

[0053] Example 6 includes one or more examples, wherein the at least one cache device comprises one or more of: level 1 cache (L1), level 2 cache (L2), level 3 cache (L3), or last level cache (LLC).

[0054] Example 7 includes one or more examples, and includes a server, wherein the server comprises the plurality of cores and the fabric.

[0055] Example 8 includes one or more examples, and includes a method comprising: in a fabric connected to a plurality of processor cores: based on receipt of a memory access request from a core of the plurality of cores: selecting a group from among a plurality of groups of cache controllers based on the memory access request, forwarding the memory access request to the selected group, based on data associated with the memory access request stored in at least one cache device, servicing the forwarded memory access request at a selected cache device of the at least one cache device, and based on data associated with the memory access request managed by a second cache controller, forward the forwarded memory access request to the second cache controller in the selected group.

[0056] Example 9 includes one or more examples, wherein: the fabric comprises a group of routers and the plurality of groups of cache controllers, the fabric comprises a group of connections

between the group of routers and the plurality of groups of cache controllers, and the fabric comprises a second group of connections between cache controllers in a group of cache controllers.

[0057] Example 10 includes one or more examples, wherein the memory access request comprises a memory address and identifier of a group of cache controllers of the plurality of groups of cache controllers to which to forward the memory access request.

[0058] Example 11 includes one or more examples, wherein based on a cache miss in the at least one cache device, requesting data associated with the memory access request from a memory device.

[0059] Example 12 includes one or more examples, and includes performing cache coherency to retrieve updated data associated with the memory access request.

[0060] Example 13 includes one or more examples, wherein a package is to enclose the plurality of cores and the fabric.

[0061] Example 14 includes one or more examples, wherein a router performs the selecting a group from among a plurality of groups of cache controllers based on the memory access request and the forwarding the memory access request to the selected group.

[0062] Example 15 includes one or more examples, wherein the selecting a group from among a plurality of groups of cache controllers based on the memory access request comprises selecting an output line to the selected group.

CLAIMS

What is claimed is:

1. An apparatus comprising:

a semiconductor package comprising:

a plurality of cores and

a fabric to provide the plurality of cores with access to at least one cache device, wherein:

the fabric comprises a group of routers and a plurality of groups of cache controllers,

the fabric comprises a group of connections between the group of routers and the plurality of groups of cache controllers, and

the fabric comprises a second group of connections between cache controllers in a group of cache controllers, wherein:

based on receipt of a memory access request from a core of the plurality of cores:

a router of the group of routers is to select a group from among the plurality of groups of cache controllers based on the memory access request and forward the memory access request to the selected group and

a cache controller in the selected group is to:

based on data associated with the memory access request stored in the at least one cache device, service the forwarded memory access request at a selected cache device of the at least one cache device and

based on data associated with the memory access request managed by a second cache controller, forward the forwarded memory access request to the second cache controller in the selected group.

2. The apparatus of claim 1, wherein the group of connections and the second group of connections are physically arranged in different directions.

3. The apparatus of any of claims 1-2, wherein the memory access request comprises a memory address and identifier of a group of cache controllers of the plurality of groups of cache controllers to which to forward the memory access request.

4. The apparatus of any of claims 1-3, comprising the at least one cache device communicatively coupled to the fabric.

5. The apparatus of any of claims 1-4, wherein based on a cache miss in the at least one cache device, the second cache controller is to request data associated with the memory access request from a memory device.

6. The apparatus of any of claims 1-5, wherein the at least one cache device comprises one or more of: level 1 cache (L1), level 2 cache (L2), level 3 cache (L3), or last level cache (LLC).

7. The apparatus of any of claims 1-6, comprising a server, wherein the server comprises the plurality of cores and the fabric.

8. A method comprising:

in a fabric connected to a plurality of processor cores:

based on receipt of a memory access request from a core of the plurality of cores:

selecting a group from among a plurality of groups of cache controllers based on the memory access request,

forwarding the memory access request to the selected group,

based on data associated with the memory access request stored in at least one cache device, servicing the forwarded memory access request at a selected cache device of the at least one cache device, and

based on data associated with the memory access request managed by a second cache controller, forward the forwarded memory access request to the second cache controller in the selected group.

9. The method of claim 8, wherein:

the fabric comprises a group of routers and the plurality of groups of cache controllers,

the fabric comprises a group of connections between the group of routers and the plurality of groups of cache controllers, and

the fabric comprises a second group of connections between cache controllers in a group of cache controllers.

10. The method of any of claims 8-9, wherein the memory access request comprises a memory address and identifier of a group of cache controllers of the plurality of groups of cache controllers to which to forward the memory access request.

11. The method of any of claims 8-10, wherein based on a cache miss in the at least one cache device, requesting data associated with the memory access request from a memory device.

12. The method of any of claims 8-11, comprising:

performing cache coherency to retrieve updated data associated with the memory access request.

13. The method of any of claims 8-12, wherein a package is to enclose the plurality of cores and the fabric.

14. The method of any of claims 8-13, wherein a router performs the selecting a group from among a plurality of groups of cache controllers based on the memory access request and the forwarding the memory access request to the selected group.

15. The method of any of claims 8-14, wherein the selecting a group from among a plurality of groups of cache controllers based on the memory access request comprises selecting an output line to the selected group.

16. An apparatus comprising:

a plurality of cores and

circuitry to provide the plurality of cores with access to at least one cache device, wherein:

the circuitry comprises a group of routers and a plurality of groups of cache controllers,

the circuitry comprises a group of connections between the group of routers and the plurality of groups of cache controllers, and

the circuitry comprises a second group of connections between cache controllers in a group of cache controllers, wherein:

based on receipt of a memory access request from a core of the plurality of cores:

a router of the group of routers is to select a group from among the plurality of groups of cache controllers based on the memory access request and forward the memory access request to the selected group and

a cache controller in the selected group is to:

based on data associated with the memory access request stored in the at least one cache device, service the forwarded memory access request at a selected cache device of the at least one cache device and

based on data associated with the memory access request managed by a second cache controller, forward the forwarded memory access request to the second cache controller in the selected group.

17. The apparatus of claim 16, wherein the group of connections and the second group of connections are physically arranged in different directions.

18. The apparatus of any of claims 16-17, wherein the memory access request comprises a memory address and identifier of a group of cache controllers of the plurality of groups of cache controllers to which to forward the memory access request.

19. The apparatus of any of claims 16-18, comprising the at least one cache device communicatively coupled to the circuitry.

20. The apparatus of any of claims 16-19, wherein based on a cache miss in the at least one cache device, the second cache controller is to request data associated with the memory access request from a memory device.

21. The apparatus of any of claims 16-20, wherein the at least one cache device comprises one or more of: level 1 cache (L1), level 2 cache (L2), level 3 cache (L3), or last level cache (LLC).

22. The apparatus of any of claims 16-21, comprising a server, wherein the server comprises the plurality of cores and the circuitry.

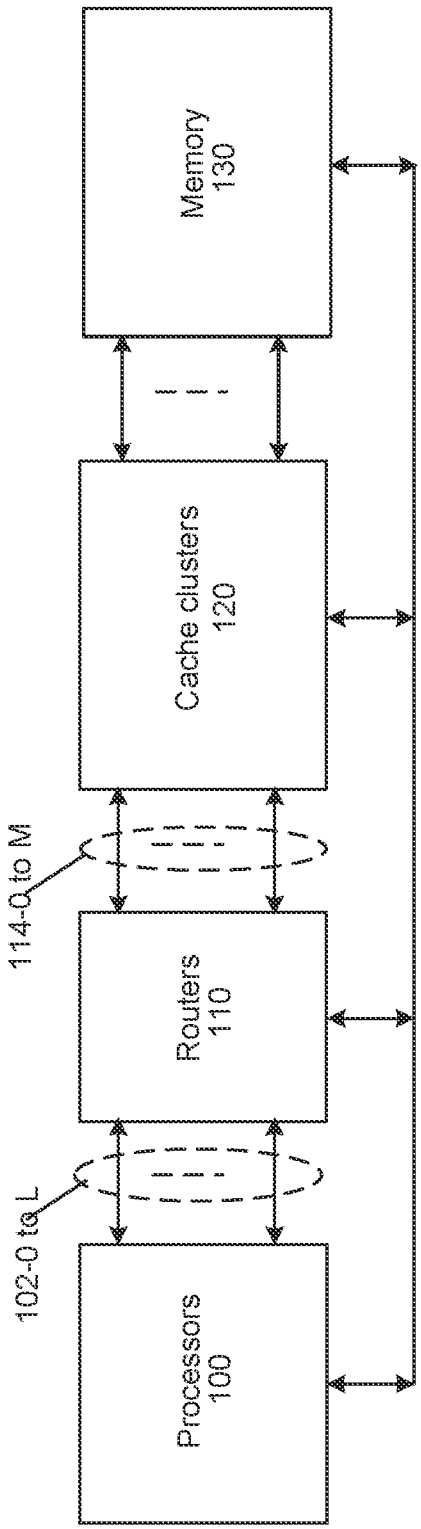
Memory
130
Cache clusters
120
114-0 to M
Routers
110
102-0 to L
Processors
100

FIG. 1A

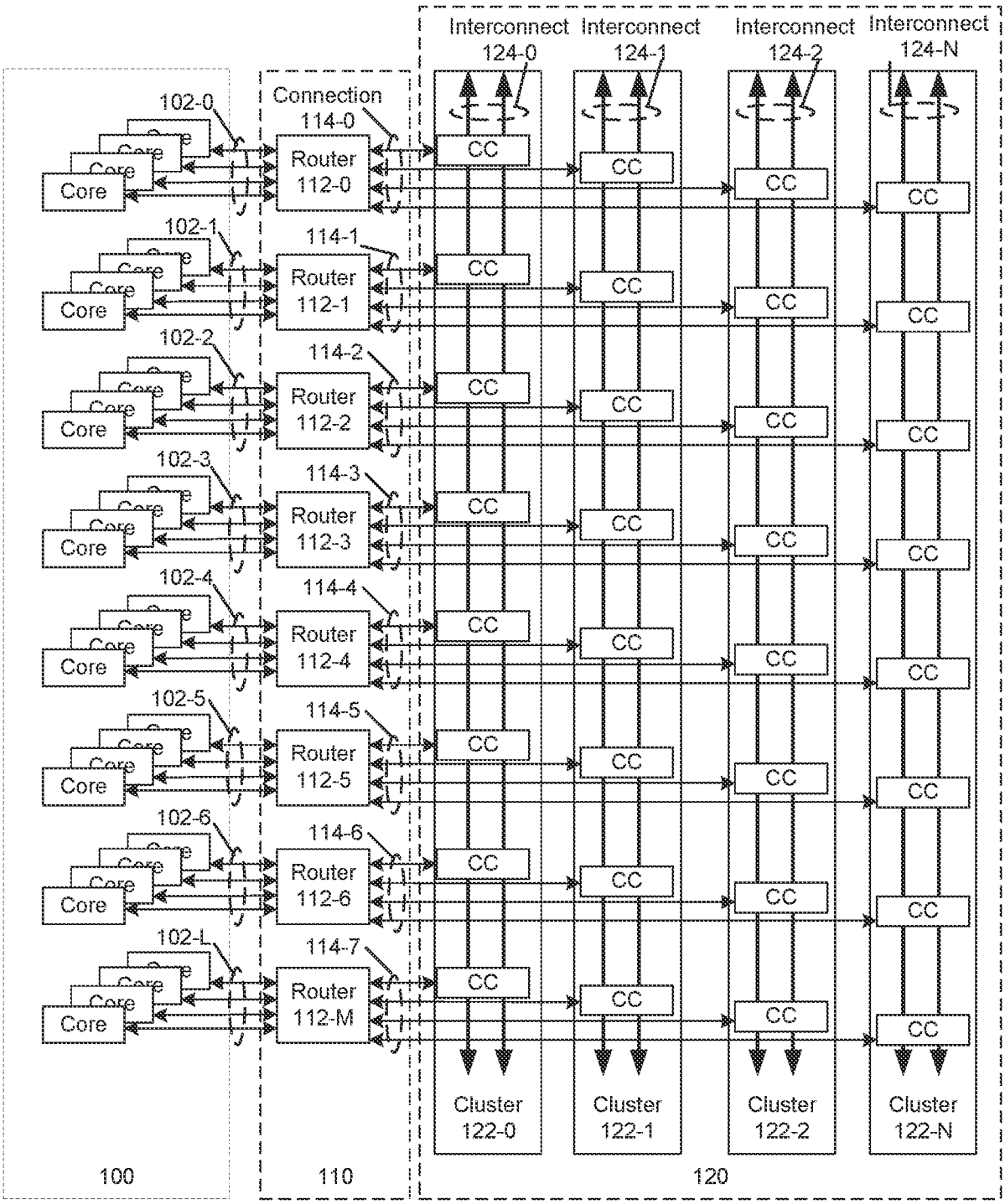
Interconnect 124-0
Interconnect 124-1
Interconnect 124-2
Interconnect 124-N
Connection 114-0
102-0
Core
Router 112-0
114-1
102-1
Router 112-1
114-2
102-2
Router 112-2
114-3
102-3
Router 112-3
114-4
102-4
Router 112-4
114-5
102-5
Router 112-5
114-6
102-6
Router 112-6
102-L
114-7
Router 112-M
CC
Cluster 122-0
Cluster 122-1
Cluster 122-2
Cluster 122-N
100
110
120

FIG. 1B

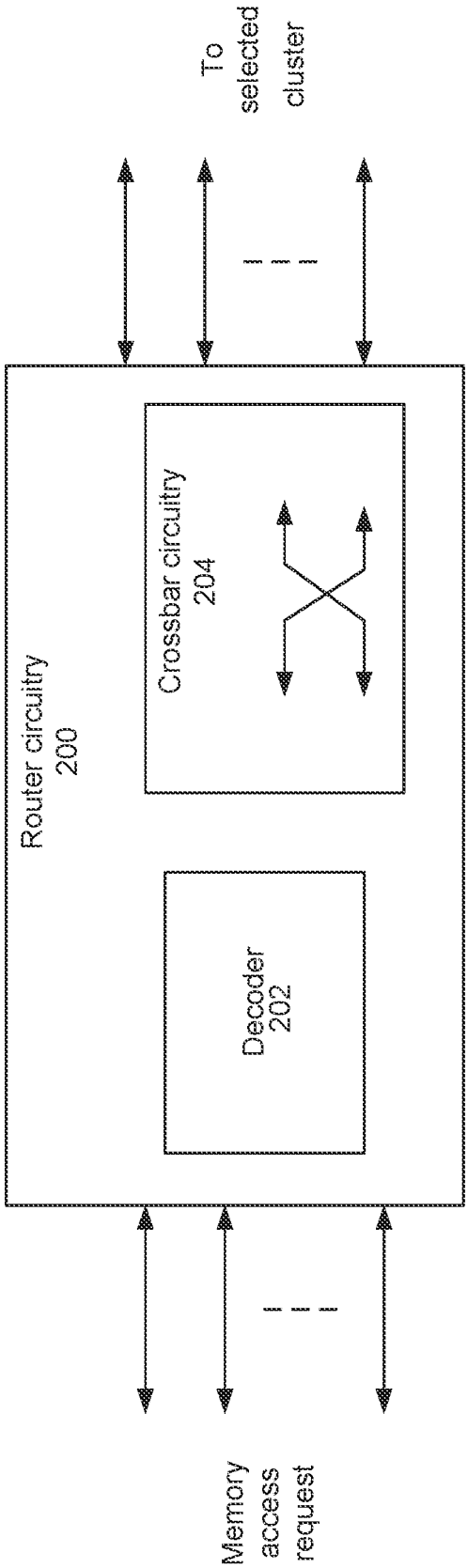
To selected cluster
Router circuitry 200
Crossbar circuitry 204
Decoder 202
Memory access request

FIG. 2

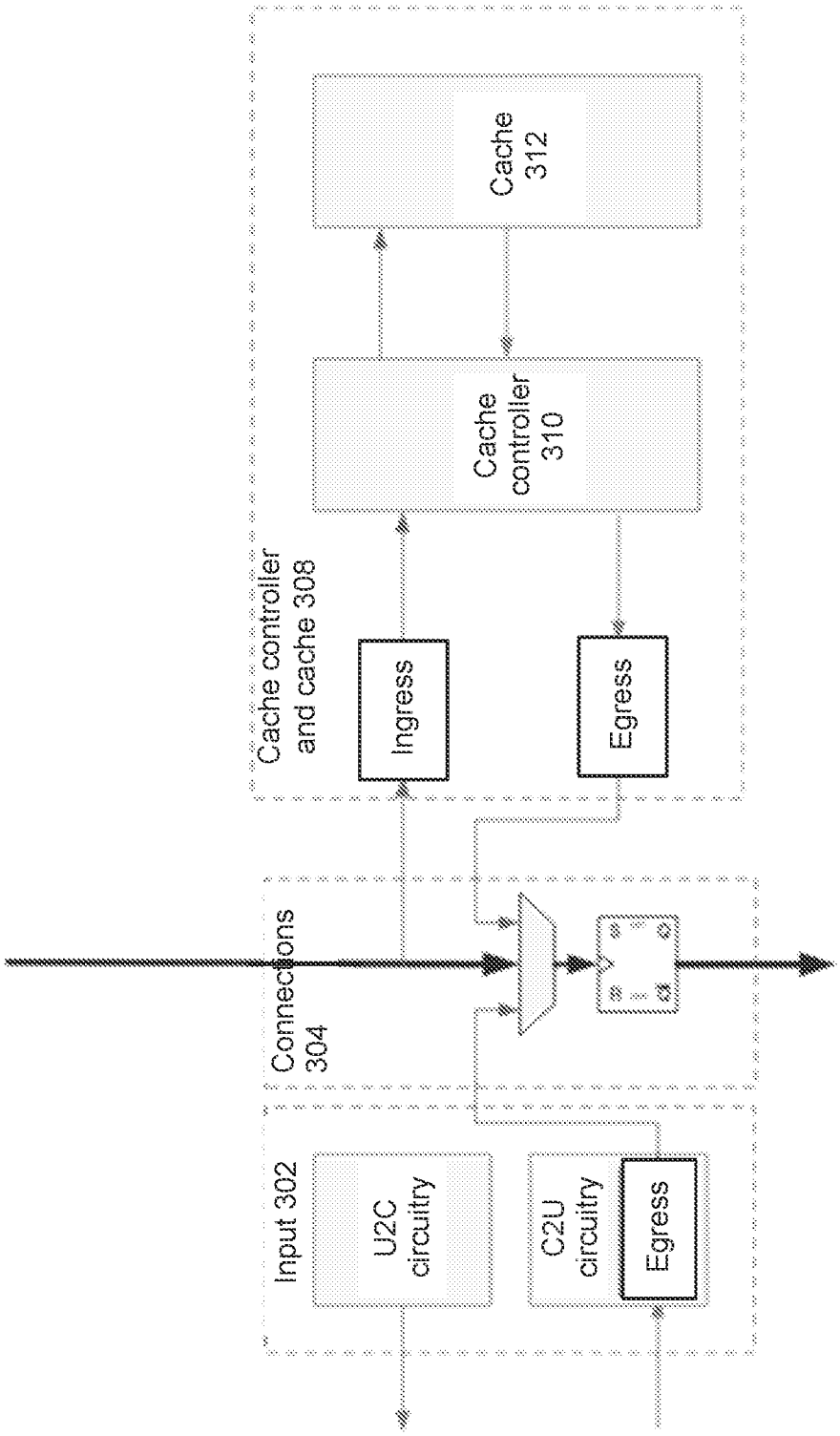
Cache controller and cache 308
Cache 312
Cache controller 310
Ingress
Egress
Connections 304
Input 302
U2C circuitry
C2U circuitry
Egress

FIG. 3A

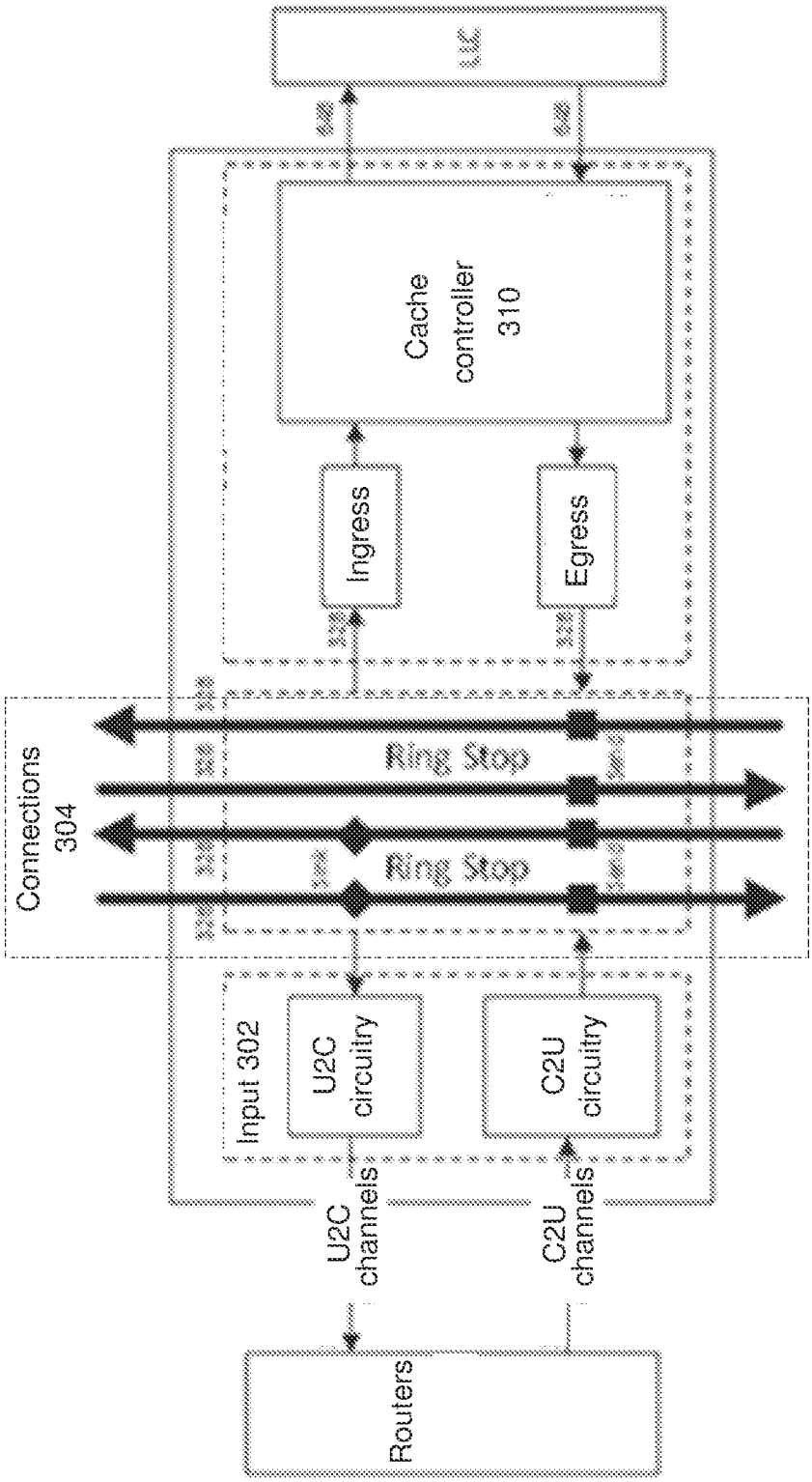
UC
Cache controller 310
Ingress
Egress
Connections 304
Ring Stop
Ring Stop
Input 302
U2C circuitry
C2U circuitry
U2C channels
C2U channels
Routers

FIG. 3B

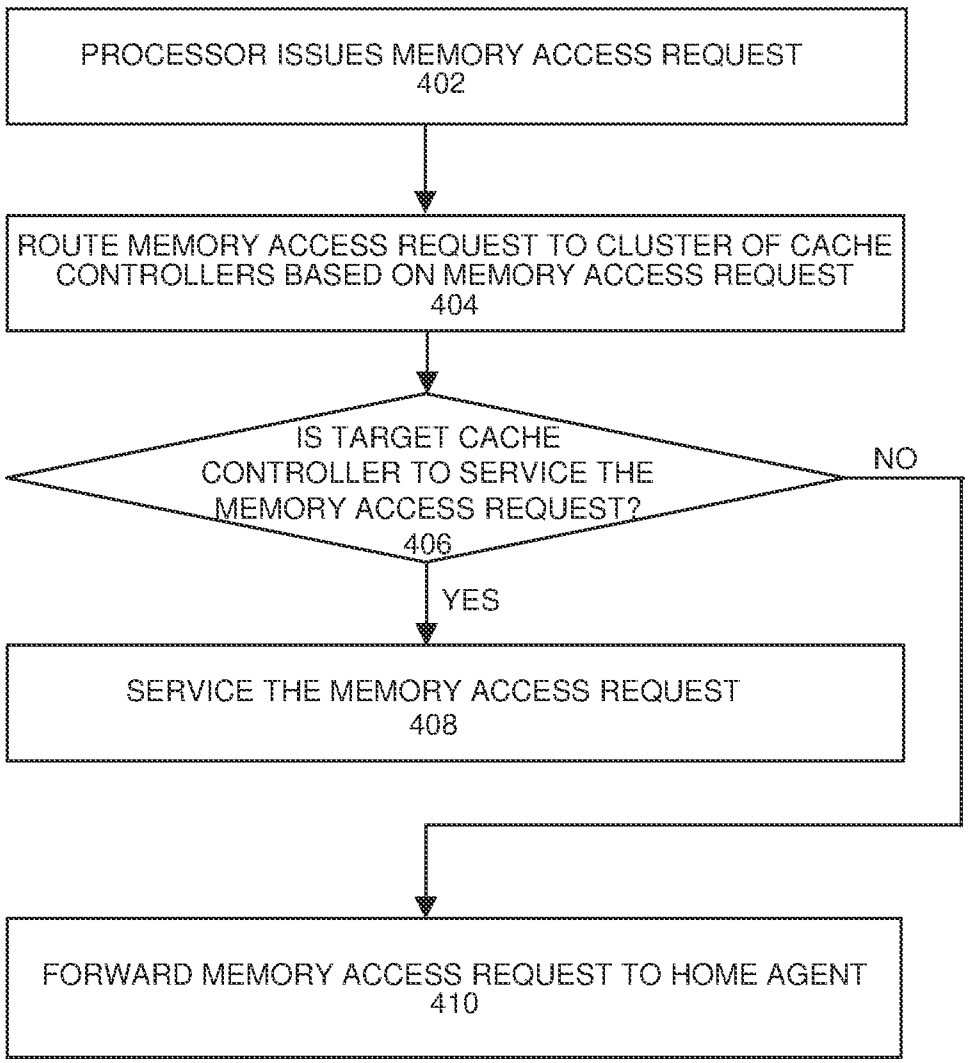
PROCESSOR ISSUES MEMORY ACCESS REQUEST
402
ROUTE MEMORY ACCESS REQUEST TO CLUSTER OF CACHE CONTROLLERS BASED ON MEMORY ACCESS REQUEST
404
IS TARGET CACHE CONTROLLER TO SERVICE THE MEMORY ACCESS REQUEST?
406
NO
YES
SERVICE THE MEMORY ACCESS REQUEST
408
FORWARD MEMORY ACCESS REQUEST TO HOME AGENT
410

FIG. 4

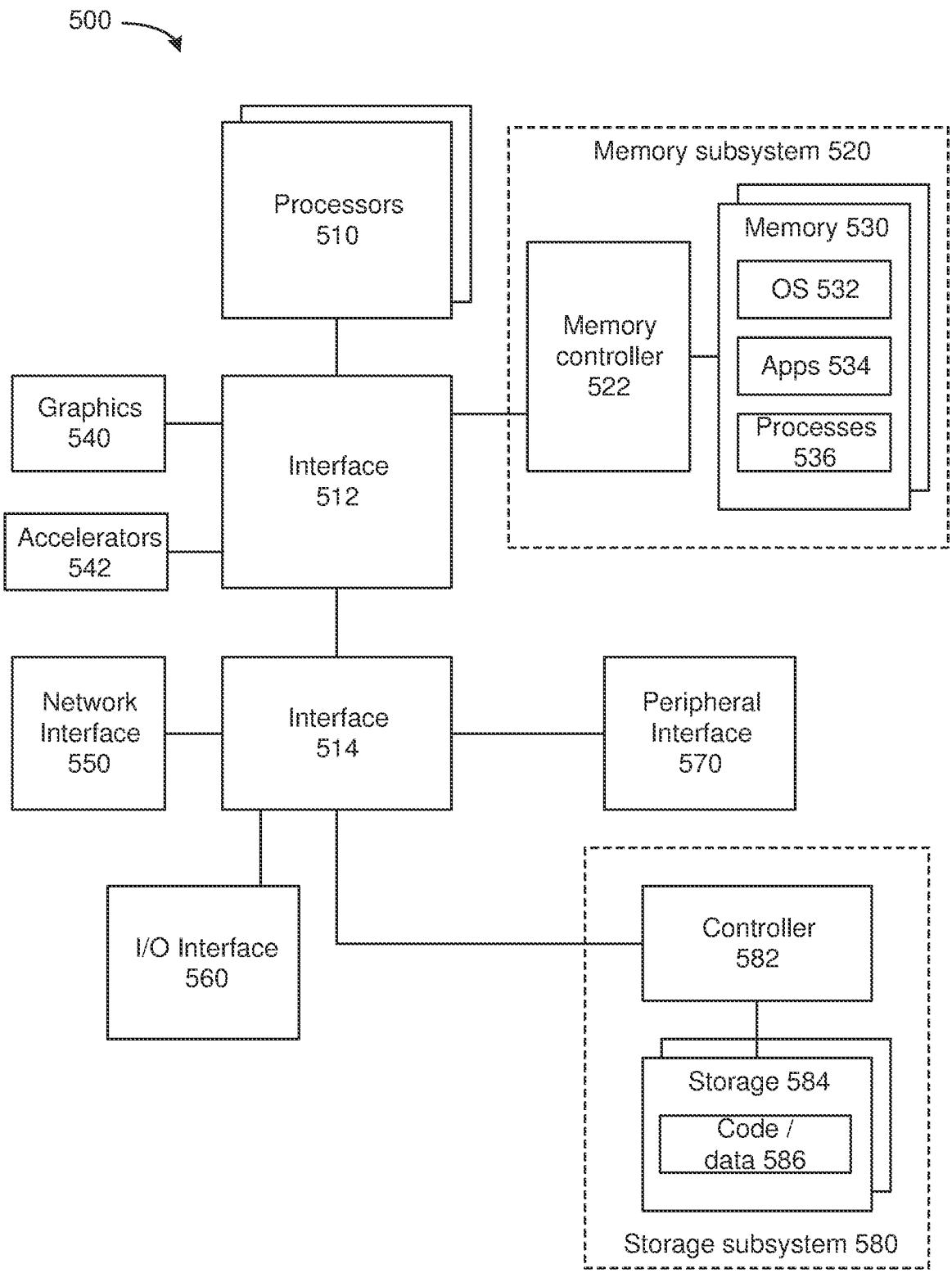
500
Processors
510
Memory subsystem 520
Memory 530
OS 532
Memory
controller
522
Apps 534
Processes
536
Graphics
540
Interface
512
Accelerators
542
Network
Interface
550
Interface
514
Peripheral
Interface
570
I/O Interface
560
Controller
582
Storage 584
Code /
data 586
Storage subsystem 580

FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2023/078576

A. CLASSIFICATION OF SUBJECT MATTER

G06F 12/0877(2016.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 12/0877(2016.01); G06F 12/02(2006.01); G06F 12/0811(2016.01); G06F 12/084(2016.01); G06F 12/0842(2016.01); G06F 12/0871(2016.01); G06F 13/28(2006.01); H04L 12/26(2006.01); H04L 12/931(2013.01)

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

Korean utility models and applications for utility models
Japanese utility models and applications for utility models

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

eKOMPASS(KIPO internal) & Keywords: cache, fabric, access, core, router, group, controller, request, forward, select

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2021-0374057 A1 (MARVELL ASIA PTE, LTD.) 02 December 2021 (2021-12-02) paragraphs [0022], [0025], [0027]-[0028], [0037]-[0038], [0040]; claim 1; and figure 2	1-22
Y	US 2022-0091980 A1 (ADVANCED MICRO DEVICES, INC.) 24 March 2022 (2022-03-24) paragraph [0023]; and figure 1	1-22
A	US 2023-0064369 A1 (APPLE INC.) 02 March 2023 (2023-03-02) paragraphs [0019]-[0048]; claims 1, 9, 14; and figures 1-5	1-22
A	US 2006-0117159 A1 (SHIGEYOSHI OHARA et al.) 01 June 2006 (2006-06-01) paragraphs [0061]-[0064], [0127]-[0135]; claim 1; and figures 1, 8-10	1-22
A	US 2014-0286179 A1 (EMPIRE TECHNOLOGY DEVELOPMENT, LLC) 25 September 2014 (2014-09-25) paragraphs [0026]-[0038]; claims 1, 5; and figures 2-5	1-22

☑ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

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

Date of the actual completion of the international search	Date of mailing of the international search report
27 February 2024	**27 February 2024**

Name and mailing address of the ISA/KR	Authorized officer
Korean Intellectual Property Office **189 Cheongsa-ro, Seo-gu, Daejeon** **35208, Republic of Korea**	**KIM, Sung Hee**
Facsimile No. **+82-42-481-8578**	Telephone No. **+82-42-481-3516**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2023/078576

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	US 2023-0315642 A1 (INTEL CORPORATION) 05 October 2023 (2023-10-05) paragraphs [0008]-[0061]; claims 1-15; and figures 1A-5 * The above document is a publication of the earlier application whose priority has been claimed in this international application.	1-22

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/US2023/078576

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
US 2021-0374057 A1	02 December 2021	US 11119929 B2	14 September 2021
		US 11620223 B2	04 April 2023
		US 2020-0250088 A1	06 August 2020
		US 2023-0229595 A1	20 July 2023
US 2022-0091980 A1	24 March 2022	US 11403221 B2	02 August 2022
US 2023-0064369 A1	02 March 2023	US 11755489 B2	12 September 2023
US 2006-0117159 A1	01 June 2006	EP 1662369 A2	31 May 2006
		EP 1662369 A3	12 November 2008
		EP 1662369 B1	06 December 2017
		EP 2296085 A1	16 March 2011
		EP 2296085 B1	15 May 2013
		KR 10-0736645 B1	09 July 2007
		KR 10-2006-0060534 A	05 June 2006
		US 2014-0223097 A1	07 August 2014
US 2014-0286179 A1	25 September 2014	CN 105164664 A	16 December 2015
		CN 105164664 B	15 June 2018
		TW 201507408 A	16 February 2015
		TW I565265 B	01 January 2017
		US 9473426 B2	18 October 2016
		WO 2014-149040 A1	25 September 2014
US 2023-0315642 A1	05 October 2023	None	