

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
29 January 2009 (29.01.2009)

PCT

(10) International Publication Number  
**WO 2009/014576 A1**

(51) International Patent Classification:

G06F 15/16 (2006.01)

(21) International Application Number:

PCT/US2008/006752

(22) International Filing Date: 27 May 2008 (27.05.2008)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

11/828,042 25 July 2007 (25.07.2007) US

(71) Applicant (for all designated States except US):

HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P. [US/US]; 20555 S.h. 249, Houston, TX 77070 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): PURCELL, Brian, T. [US/US]; 20555 Tomball Parkway, Houston, TX 77070 (US). BENEDICT, Melvin, K. [US/US]; 20555 Tomball Parkway, Houston, TX 77070 (US).

(74) Agents: WEBB, Steven, L et al.; Hewlett-Packard

Company, Intellectual Property Administration, P.O. Box 272400, Mail Stop 35, Fort Collins, CO 80527-2400 (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, BR, BW, BY, BZ, CA, CH, 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, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every

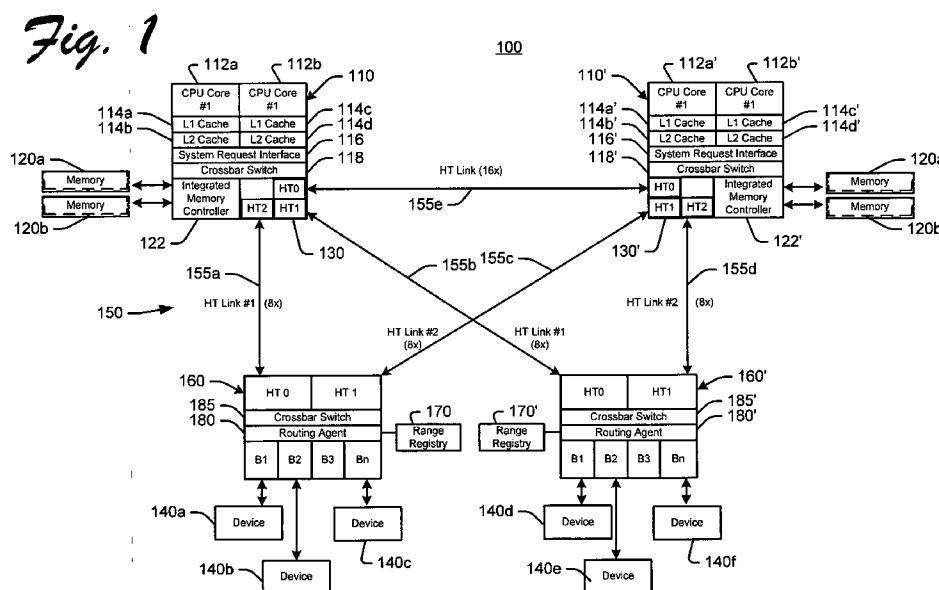
kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI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

Declarations under Rule 4.17:

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

[Continued on next page]

(54) Title: SYSTEMS AND METHODS FOR IMPROVING PERFORMANCE OF A ROUTABLE FABRIC



(57) Abstract: Systems and methods for improving performance of a routable fabric are disclosed. In an exemplary embodiment a system may comprise a plurality of compute nodes, a routable fabric, and a plurality of chipsets connected by the routable fabric to the plurality of compute nodes. The chipsets have range registers dynamically directing traffic from any device to any of the plurality of compute nodes over the routable fabric.



**Published:**

- *with international search report*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

- 1 -

**SYSTEMS AND METHODS FOR IMPROVING PERFORMANCE  
OF A ROUTABLE FABRIC**

**BACKGROUND**

[0001] Data center environments place a tremendous demand on computing systems (e.g., servers, blades, etc.) to provide significant amounts of I/O bandwidth. A server typically provides the required I/O bandwidth needs by supporting provisions for adding multiple I/O cards/devices (also referred to as “adapters”) and/or directly incorporating embedded devices within the server. The embedded devices and/or add-in adapter interfaces are typically, but not limited to, PCI Express, PCI / PCI-X, and HyperTransport. The adapters represent a variety of device classes, including storage (SCSI, SATA, SAS, RAID, backup, etc), networking (Ethernet, ATM), clustering (Infiniband, ServerNet), multimedia (video, audio), and others.

[0002] It is oftentimes impractical (e.g., due to reliability, cost, and component yield) to connect many I/O devices directly to a compute node or other processing element due to the large number of component pins that would be required. Processor, chipset, and component vendors have addressed these issues by partitioning the various functions and interfaces (e.g., as computing, memory, and

- 2 -

[0003]

[0004] I/O interfaces) into multiple devices. The architecture and partitioning scheme provides a generic and simple way to construct multiple platforms that range from small, simple systems that have one or two components to large systems with one or more instances of each component.

[0003][0005] Larger systems (e.g., Opteron-based systems) may include multiple processor cores/sockets, multiple chipset components, and many I/O expansion slots. These systems are designed to optimize the CPU-CPU and CPU-memory bandwidth. Accordingly, most of the compute node's (or processor's) buses/interconnects are dedicated to connect memory, memory controllers, and/or other processors. Depending upon the complexity of the system, one or more processors may either have no additional interfaces available, or a very limited/restricted (perhaps in bandwidth) interface available to connect to the I/O subsystem (or other parts of the compute grid in a multi-compute node environment). This scenario can force the I/O or expansion chipsets ("chipset") to the "corners" or periphery of the processing elements within the compute node.

[0004][0006] Another side effect of reduced/limited connectivity between the chipset and the processor/memory elements is that there may be a large disparity between the amount of bandwidth on either side of the protocol translator (or "chipset"). For example, a system configuration may have a chipset component that

- 3 -

[0007]

[0008] supports over thirty-two lanes of PCI Express (PCIe), while the chipset to processor/memory interface only has at most eight lanes. Chipset vendors, on behalf of system vendors, have opted to include additional interfaces (e.g., HyperTransport) between the chipset and processor/memory components. The additional interfaces not only provide additional bandwidth, but also provide better balance between the various interfaces (chipsets, protocols, etc). The inclusion of additional interfaces to the chipset can reduce the number of chipset components required for a given design, resulting in cost savings.

[0005][0009] Chipsets may have a very different “view” of the nodes (e.g., the processor and memory components). As mentioned previously, the optimization of the CPU-CPU and CPU-memory interconnect may not allow the chipset to be connected directly to each node. Chipset transactions to/from nodes must traverse from one node to another node, until the destination node is reached. Each link between nodes and/or the chipset represents one “hop”. From the chipset perspective, different nodes within the compute environment may be a different number of hops away. Nodes that have fewer hops are more “near,” whereas nodes with a higher number of hops from the chipset are more “far.” System performance is directly related to the amount of active chipset (e.g., I/O) bandwidth and the number of hops that the chipset-to-target-node. The chipset transactions are

- 4 -

[0010]

[0011] replicated at each node along the chipset-to-target-node path. The chipset transactions consume bandwidth from each local node's available bandwidth (e.g., memory) and thereby limit the amount of bandwidth of the processor(s) and other devices within that node.

[0006][0012] When the chipset supports multiple links into the compute node environment, additional chipset bandwidth is available. Currently planned chipset architectures provide either soft or hard partitioning between the upstream (compute node) interfaces and downstream (compute node, fabric, or I/O) interfaces. Traffic (DMA, interrupts, messages, management/etc) is pinned from a downstream interface to only one upstream interface. This pinning (via software and/or hardware configuration/strapping) of a downstream interface to a single upstream interface may not provide the optimal system performance due to the number of hops particular traffic encounters between the chipset and target node.

[0007][0013] The problem is very apparent when the operating system scheduler moves tasks/processes (e.g., drivers, applications) from one node to another within the compute environment. The dynamic movement of these processes can either improve or hinder system performance, depending upon the traffic profile (e.g., number of hops) of the chipset.

[0014]

- 5 -

### **BRIEF DESCRIPTION OF THE DRAWINGS**

[0008][0015] **Figure 1** is a schematic diagram of an exemplary processing system having a routable fabric.

[0009][0016] **Figure 1a and 1b** illustrate exemplary address translation.

[0010][0017] **Figure 2** is a high level diagram of an exemplary 2S design showing improved performance using a bifurcated design in a routable fabric.

[0011][0018] **Figure 3** is a high level diagram of an exemplary 4S design showing improved performance using a bifurcated design in a routable fabric.

[0012][0019] **Figure 4** is a high level diagram of an exemplary 8S design showing improved performance using a bifurcated design in a routable fabric.

[0013][0020] **Figure 5** is a flowchart illustrating exemplary operations for improving performance of a routable fabric.

### **DETAILED DESCRIPTION**

[0014][0021] It is beneficial to provide an added level of intelligence within the chipset device(s) to route their traffic to the target node using the most direct path (e.g., least number of hops). Providing multiple paths to the CPU complex can reduce the number of hops, thereby reducing or altogether eliminating the replication of data on the CPU buses. Exemplary embodiments increase I/O

- 6 -

**[0022]**

**[0023]** bandwidth connectivity and utilize all of the available I/O links. This is particularly desirable for high bandwidth devices (e.g., GPUs, 10Gbps NICs, and RAID adapters).

### **Exemplary System**

**[0015][0024]** Figure 1 is a schematic diagram of an exemplary processing system 100 having a routable fabric 150. In an exemplary embodiment, the processing system 100 may include a plurality of compute nodes 110 and 110'. Compute node 110 may have one or more CPU core 112a-b, cache 114a-d, and/or system request interface(s) 116. A crossbar switch 118 may be provided to connect the CPU cores 112a-b to the shared memory resources 120a-b via an integrated memory controller 122. Although not called out herein, like components are also shown in Figure 1 for compute node 110' and are designated with the "prime" symbol. Of course, more than two compute nodes 110 and 110' may also be provided in processing system 100, as will be readily apparent from the following description.

**[0016][0025]** Each compute node 110 and 110' may also include one or more interfaces to the routable fabric. In the exemplary embodiment, these interfaces may consist of one or more HT interfaces 130 and 130', respectively. Generally, HT interfaces 130 and 130' may include circuitry and program code (e.g.,

- 7 -

[0026]

[0027] firmware) to connect a plurality of devices 140a-f (directly or via one or more suitable interconnect translation (e.g., bridge) devices) over the routable fabric 150 to one or more of the compute nodes 110 and 110'. Of course other point-to-point links may also be implemented and are not limited to the HT link (the HT link being only exemplary in this embodiment). The devices may include storage media (e.g., RAID), network interface cards (NICs), multimedia devices, PCIe cards, both embedded and non-embedded devices, and/or other devices now known or later developed.

[0017][0028] Any of the devices 140a-f may connect via a chipset 160 or 160' over a direct path 155a-e through the routable fabric 150 to any of the compute nodes 110 and 110'. Although all of the devices 140a-f may simultaneously connect to separate compute nodes 110 and 110' and multiple devices 140a-f can connect/communicate across the routable fabric concurrently.

[0018][0029] In the exemplary embodiment, devices that connect to the routable fabric may contain one or more logical and/or physical fabric interfaces. A single physical interface may be divided into multiple logical interfaces, each providing unique "ports" or "paths" into the routable fabric. In exemplary embodiments, multiple paths 155a-e may be established using HyperTransport links. The HyperTransport 3.0 specification allows links to be divided (e.g., bifurcated) into

- 8 -

[0030]

[0031] multiple, independent connections. This allows a HT-based device with the ability to connect to multiple/other HT devices. For example, a 16x HT link can be split or divided (e.g., bifurcated) into two logical 8x HT links. That is, a single HT port from a processor may be split into logical ports (e.g., HT0, HT1, HT2, etc.) to provide multiple paths 155a and 155b. Another single HT link from chipset 160' may also be split (e.g., bifurcated) into logical ports HT0 and HT1 to provide paths 155c and 155d. It is noted that in the exemplary implementation each chipset 160, 160' has two independent HT links. In other embodiments, however, a bifurcated link may be employed. It is noted that the bifurcated HT link is capable of operating as either two 8x HT links or as a single 16x HT link, depending on bandwidth requirements.

[0019][0032] Dual HT interfaces enable each chipset 160 and 160' to be connected to both compute nodes 110 and 110' in the 2S design shown in Figure 1, and traffic to be routed between both compute nodes 110 and 110'. In 4S (see Figure 3) and 8S (see Figure 4) designs, bifurcation (and/or multiple HT interfaces) allows use of the same I/O subsystem between the two platforms. For example, in the 4S design, each chipset component connects to two processors via independent 8x links which allows the chipset twice the number of connections to the compute node 110, or allows simultaneous connection between multiple compute nodes 110 and 110'.

- 9 -

[0033]

[0020][0034] It is noted that there are no theoretical limits on the number of paths that can be included in a routable fabric and bifurcation is shown only as an exemplary embodiment. The number of paths is limited primarily by the connectivity implemented in the routable fabric and devices that interface to the routable fabric. It is also noted that the processing system is not limited to 1S, 2S, 4S, or 8S designs, and can also be extended to other processing environments.

[0021][0035] It is also noted that bifurcation and dual HT links is described only to illustrate one example of a representative chipset and processor architecture which may be implemented. Other embodiments are also contemplated to allow the chipset to directly connect to 2+ compute nodes (CPUs).

[0022][0036] In exemplary embodiments, a range registry 170 may be implemented as a physical registry set up in the interface stack of each chipset 160 and 160' to dynamically direct traffic through the multiple paths 155a-e in the routable fabric 150. The BIOS or software may set up the chipset routing tables. For example, the BIOS may provide extensions to a routing table in order to provide a "weighted" view of CPU/memory to chipset and endpoint affinity. This provides a mechanism to tell the OS and software how many hops for a particular I/O device, thereby enabling the OS and drivers to affinitize the I/O device (or

[0037]

- 10 -

[0038]

[0039] compute node) traffic to the other fabric endpoint (DRAM memory or other device) based upon the most direct path.

[0023][0040] Additionally, the range registry 170 and associated logic/circuitry directs traffic (DMA requests, messages, etc.) from device ports B1-Bn to one or more of the fabric interconnect ports 155. The range registry directs the traffic in order to optimize the traffic's path (by minimizing the number of hops).

[0024][0041] In an exemplary embodiment, each routing table entry includes a descriptor defining the physical base address, a physical address limit (or aperture size), and configurable access parameters including read, write, prefetch, and write posting. Each routing table entry may also contain a configuration mechanism that associates/ties the descriptor to a specific upstream port. Each descriptor can be routed to one upstream port (but not both). Multiple descriptors are available to allow several address ranges to be mapped between the two upstream ports.

[0025][0042] When a bus master (DMA) request occurs, the routing agent 180 first performs virtual to physical address translation if needed. In an exemplary embodiment, the routing agent may be an I/O Memory Mapped Unit (IOMMU). The routing agent (IOMMU) is responsible for virtual to physical address translation and/or security/authentication tasks. The routing agent may also perform virtual-to-virtual address translation, in cases where multiple translations

- 11 -

[0043]

[0044] may be required to determine the final physical address. In any event, the translated (if required) address is then compared to the routing table entries to determine which upstream port the cycle should be forwarded to via the crossbar switch 185. The upstream port then forwards the transaction to the routable fabric interconnect 155.

[0026][0045] Address translation may be better understood with reference to Figures 1a and 1b. **Figure 1a** shows an exemplary physical memory map from each compute node's perspective. In this example, memory map 190 corresponds to compute node 1 and memory map 192 corresponds to compute node 2 (although any number of compute nodes may be present). Compute node 1 cannot directly access the memory/resources of compute node 2 without some kind of address translation. Compute node 1 does have the ability to directly access memory at compute node 2 (shown at the top of the addressable space), but some form of address translation is needed to re-map down into the physical range (e.g., available DRAM) at compute node 2.

[0027][0046] **Figure 1b** shows an exemplary system architecture with two compute nodes (as shown in Figure 1), each with their own local memory and view of system resources (as shown in Figure 1a). The switches 195a-b (e.g., HT bridges/switches, PCIe, or other fabric interconnect), serve as a fabric interconnect.

- 12 -

[0047]

[0048] The fabric interconnect provides address translation services (if required) and directs traffic to the appropriate ports or paths within the fabric. In the example shown in Figure 1b, the arrows and text illustrate what type of address translations may occur at the various stages. In this example, only the device 170 (e.g., storage and/or network controller) to target compute node access is detailed.

[0028][0049] Before continuing, it is noted that dynamically directing traffic through the multiple paths 155a-e in the routable fabric 150, as just described, is for purposes of illustration and is not intended to be limiting. In addition, other functional components may also be provided and are not limited to those shown and described herein.

[0029][0050] It is also noted that the exemplary embodiments described herein are particularly well-suited for use within the PCI configuration space. The configuration space can be accessed via standard PCI configuration cycles, memory-mapped cycles, or other type of messaging or management protocol. Each upstream link (e.g., HT0 and HT1) connects to a unique port on one or more processors (e.g., compute nodes). By common architectural implementations (e.g., PCI based systems), each I/O link contains its own set of unique PCI bus numbers. The integrated devices (e.g., PCI Express bridges, P2P bridges, embedded devices) can be configured ("pinned") to either port, but not both (in order to conform to

- 13 -

[0051]

[0052] common software device discovery algorithms). Individual devices must only appear once in the PCI configuration space (unless software/hardware is included to handle devices which would be discovered multiple times by one or more fabric-connected devices). It is noted that each CPU can have a unique PCI configuration routing table to optimize the routing of PCI configuration cycles to the chipset. This allows multiple entry points for configuration cycles (e.g., a PCI configuration cycle to bus 0xYY can enter a device from either routable fabric port, depending on which device initiated/forwarded the configuration cycle), and provides the most direct (eg. shortest) routing path between the initiator and target devices.

[0030][0053] In addition, it is noted that the exemplary embodiments described herein allow the entire I/O to be available in all product numbers (or SKUs). In 4S and 8S designs, bifurcation enables use of the same I/O subsystem between the two platforms. For example, the 4S system can be configured as either 2S or 4S (each chipset component connects to two processors via a unique 8x link, enabling full I/O connectivity in either a 2S or 4S configuration). The 8S system can be configured as either 4S or 8S (the first four processors hook to the four chipset components by 8x links and the remaining 4 processors are hooked to the same chipset components via 8x links).

- 14 -

[0054]

[0031][0055] The exemplary embodiments described herein also enable higher concurrency between I/O (chipset) and system memory (i.e., there are more active HT links). All links can be utilized within a system without having to add chipset components.

[0032][0056] Furthermore, the average number of "hops" the data takes between I/O (chipset) and system memory 2S design can be reduced if the devices (e.g., chipsets) within the routable fabric direct traffic to the closest and/or most-optimal port. For example, the average number of hops goes from 1.5 to 1.0 (a 33% improvement) in a 2S design. The average number of hops goes from 1.75 to 1.5 (a 14% improvement) in a 4S design. And the average number of hops goes from 1.875 to 1.75 (a 6.7% improvement) in an 8S design, as will be better understood with reference to the illustrations shown in Figures 2-4.

[0033][0057] **Figure 2** is a high level diagram of an exemplary 2S design showing improved performance using a bifurcated design in a routable fabric. In a traditional 2S interconnect 200, all I/O is connected to CPU1 via 16x HT links. The crossbar capabilities of CPU socket 2 are not fully realized, and an average 1.5 hops from the chipset are required to reach the memory resources at either of the CPUs.

- 15 -

**[0034][0058]** In a bifurcated 2S design 250, however, each processor is connected to a chipset via an 8x HT link. All I/O is available in both the 1S and 2S configurations, and I/O is balanced across all crossbars. Accordingly, an average 1.0 hops are required to reach the memory resources at either of the CPUs, resulting in a 33% reduction in number of hops as compared to the traditional 2S interconnect 200.

**[0035][0059]** Figure 3 is a high level diagram of an exemplary 4S design showing improved performance using a bifurcated design in a routable fabric. In a traditional 4S interconnect 300, all I/O is connected to CPU1 via 16x HT links, and all CPUs are interconnected via 16x HT links. An average 1.75 hops from the chipset are required to reach the memory resources at either of the CPUs.

**[0036][0060]** In a bifurcated 4S design 350, however, each processor is connected to a chipset via an 8x HT link, and all CPUs are interconnected via 16x HT links. All I/O is available in both the 2S and 4S configurations, and I/O is balanced across all crossbars. Accordingly, an average 1.5 hops are required to reach the memory resources at either of the CPUs, resulting in a 14% reduction in number of hops as compared to the traditional 4S interconnect 300.

**[0037][0061]** Figure 4 is a high level diagram of an exemplary 8S design showing improved performance using a bifurcated design in a routable fabric. In a

- 16 -

[0062]

[0063] traditional 8S interconnect 400, all I/O is connected to CPU1 via 8x HT links, and all CPUs are interconnected via 8x HT links. However, only 50% of the HT bandwidth is available to the I/O compared to a traditional 4S design. That is, there are four unused HT links (CPU5-CPU8). An average 1.875 hops from the chipset are required to reach the memory resources at either of the CPUs.

[0038][0064] In a bifurcated 8S design 450, however, each processor is connected to a chipset via an 8x HT link, and all CPUs are interconnected via 8x HT links. The full I/O subsystem is available in both the 4S and 8S configurations, and all HT links are used (i.e., there are 8 links to the I/O). Accordingly, an average 1.75 hops are required to reach the memory resources at either of the CPUs, resulting in a 6.7% reduction in number of hops as compared to the traditional 8S interconnect 400.

[0039][0065] It is noted that the exemplary systems discussed above are provided for purposes of illustration. Still other implementations are also contemplated.

---

### **Exemplary Operations**

[0040][0066] Figure 5 is a flowchart illustrating exemplary operations for improving performance of a routable fabric. Operations 500 may be embodied as logic and/or logic instructions on one or more computer-readable medium. When

- 17 -

[0067]

[0068] executed the logic instructions cause one or more devices to be programmed for implementing the described operations. In an exemplary embodiment, the components and connections depicted in the figures may be used to improve performance of a routable fabric.

[0041][0069] In operation 510, data packets (such as, but not limited to, DMA requests, management, configuration, interrupt, and messages) are received from any of a plurality of devices connected to a compute complex. The compute complex may contain any combination of processors, chipsets, memory, routable fabric interfaces, and/or devices. In operation 520, an available path is identified for the data packets through a routable fabric based on data stored in a plurality of range registers. For example, the range registers or virtualization agent (e.g., IOMMU) may be implemented to translate addresses of the data packets to identify the most efficient and/or direct path through the routable fabric. In operation 530, the data packets are dynamically directed to any of a plurality of compute nodes in the compute complex.

[0042][0070] Dynamically directing the data packets to any of a plurality of compute nodes in the compute complex reduces the number of CPU-CPU or CPU-I/O hops. Dynamically directing the data packets to any of a plurality of compute nodes in the compute complex also reduces data replication on CPU buses. This

- 18 -

[0071]

[0072] also allows the scalability/extendibility/expandability of the fabric to include multiple compute nodes.

[0043][0073] The operations shown and described herein are provided to illustrate exemplary embodiments for improving performance of a routable fabric. It is noted that the operations are not limited to the ordering shown and described. In addition, still other operations may also be implemented.

[0044][0074] By way of example, operations may implement physical-to-virtual, physical-to-physical, and/or virtual-to-virtual address translation (e.g., based on data stored in the range registers and/or virtualization agent such as the IOMMU). The translation features may be handled by one or more I/O Memory Management Units (IOMMUs) utilizing IOMMU entries. The initial translation services and access privileges (e.g., security) is performed by the IOMMU. The translated address (either virtual to physical, or virtual to virtual) is sent to the appropriate "fabric port" based upon the resulting translated address.

[0075] In addition to the specific embodiments explicitly set forth herein, other aspects and implementations will be apparent to those skilled in the art from consideration of the specification disclosed herein. It is intended that the specification and illustrated implementations be considered as examples only, with a true scope and spirit of the following claims.

**REPLACEMENT CLAIM SET**

1. A system comprising (100):  
a plurality of compute nodes (110);  
a routable fabric (150); and  
a plurality of chipsets (160) connected by the routable fabric to the plurality of compute nodes (110), the chipsets (160) having range registers dynamically directing traffic from any device to any of the plurality of compute nodes (110) over the routable fabric (150).
2. The system (100) of claim 1 further comprising:  
at least one table and logic to translate physical addresses for directing traffic from any device to any of the plurality of compute nodes (110) over the routable fabric (150).
3. The system (100) of claim 1 further comprising at least one table and logic to translate virtual addresses for directing traffic from any device to any of the plurality of compute nodes (110) over the routable fabric (150).
4. The system (100) of claim 1 wherein the routable fabric (150) includes at least one of the following: point-to-point link links, HT links, split HT links, and bifurcated split HT links.
5. The system (100) of claim 1 wherein the plurality of chipsets (160) include bridges, and the bridges include at least one of the following: PCI controllers, PCIe controllers, storage controllers, video controllers, audio controllers, I/O Memory Management Units (IOMMUs) and network interface controllers.
6. The system (100) of claim 1 wherein the plurality of chipsets (160) include endpoint devices, and the endpoint devices include at least one of the following: an add-in adapter or embedded component, storage media, network controllers, storage devices, video devices, and audio devices.
7. A method (500) for improving performance of a routable fabric, comprising:

receiving (510) data packets from any of a plurality of devices connected to a compute complex;

identifying (520) an available path for the data packets through a routable fabric (150) based on data stored in a plurality of range registers; and

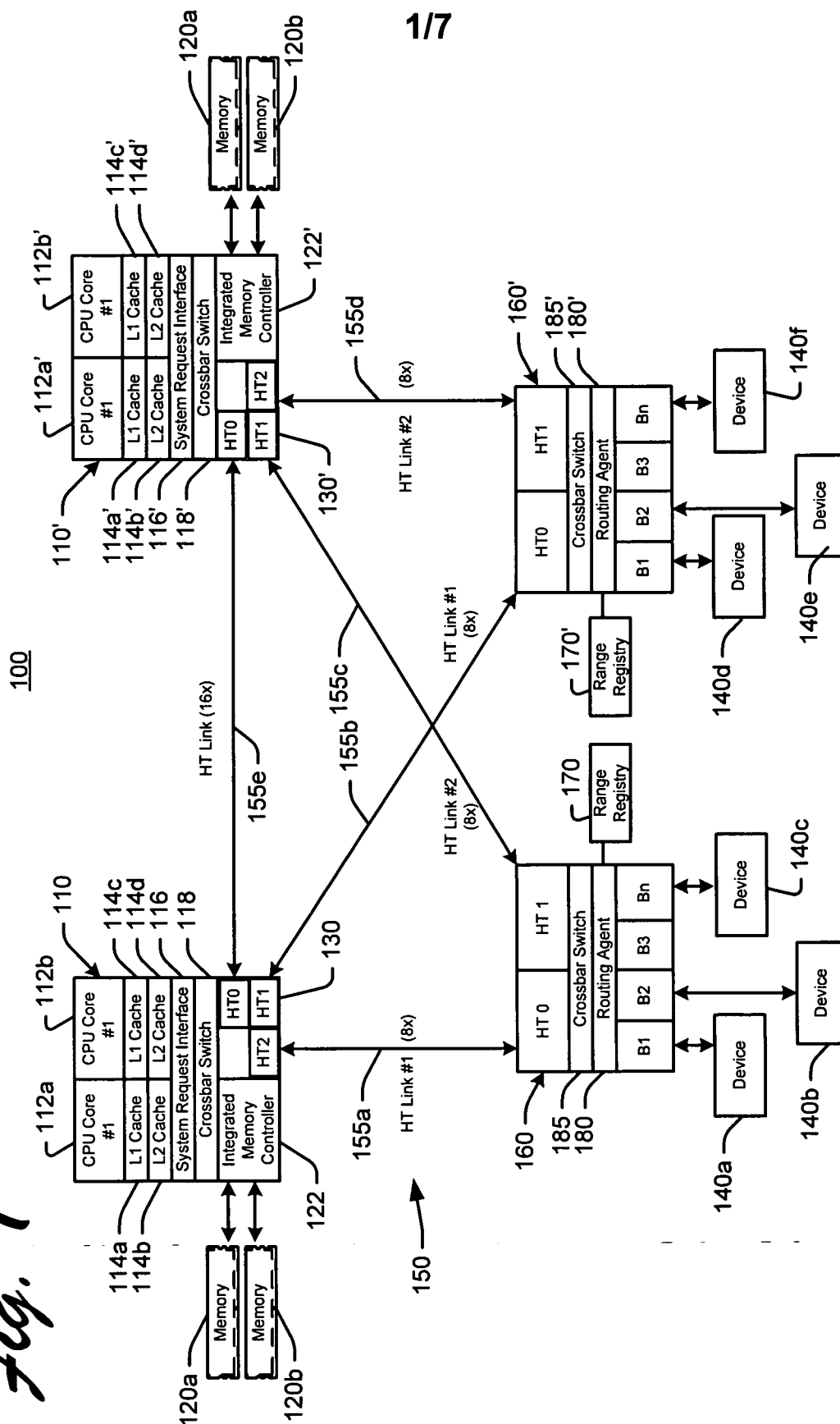
dynamically (530) directing the data packets to any of a plurality of compute nodes (110) in the compute complex.

8. The method (500) of claim 7 wherein dynamically directing the data packets to any of a plurality of compute nodes (110) in the compute complex reduces the number of CPU-CPU or CPU-I/O hops.

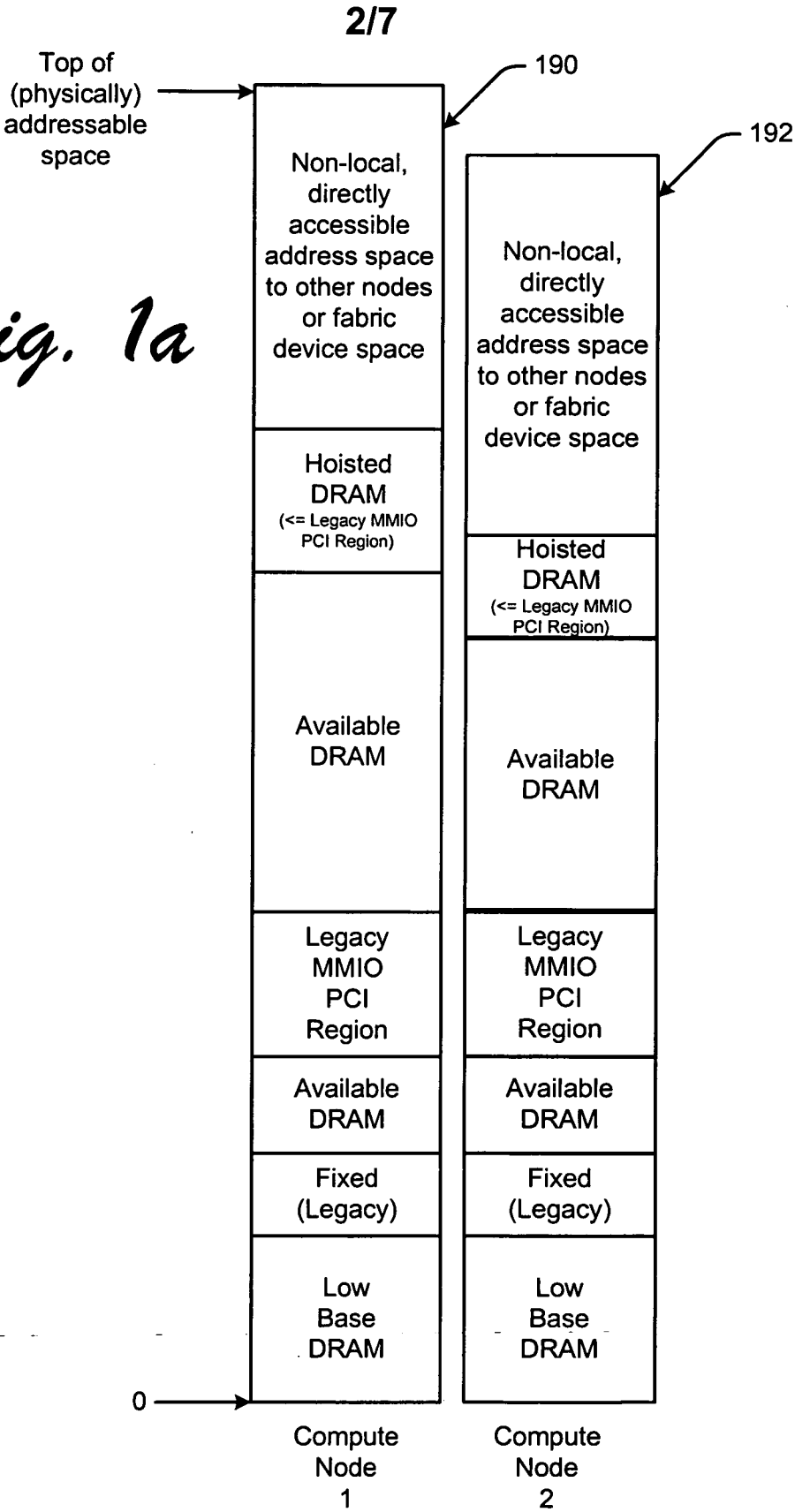
9. The method (500) of claim 7 wherein reducing the number of CPU-CPU or CPU-I/O hops reduces data replication on CPU buses.

10. The method (500) of claim 7 wherein the routable fabric (150) includes a split point-to-point link, point-to-point links, or a split point-to-point bifurcated HT link.

Fig. 1



*Fig. 1a*



3/7

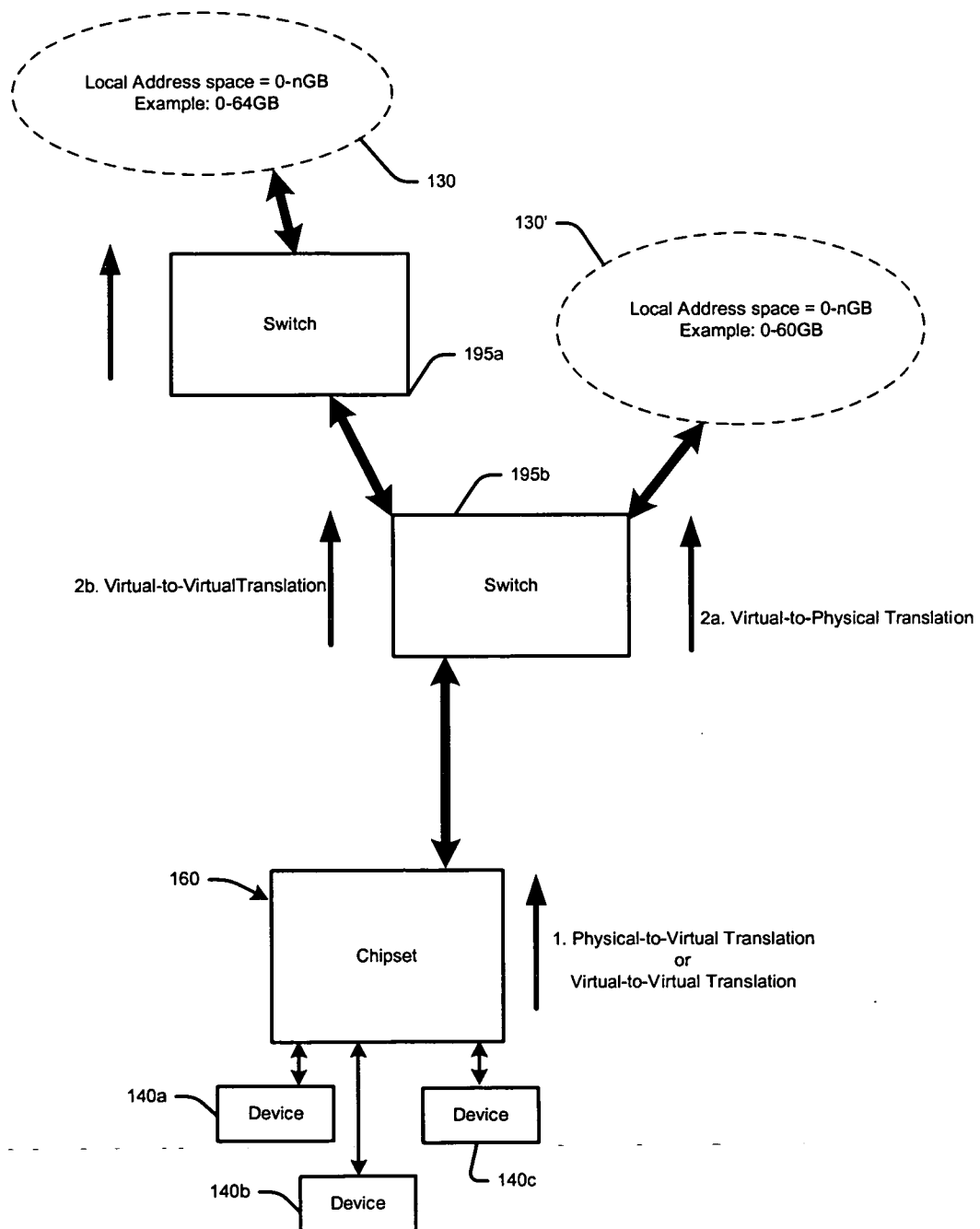
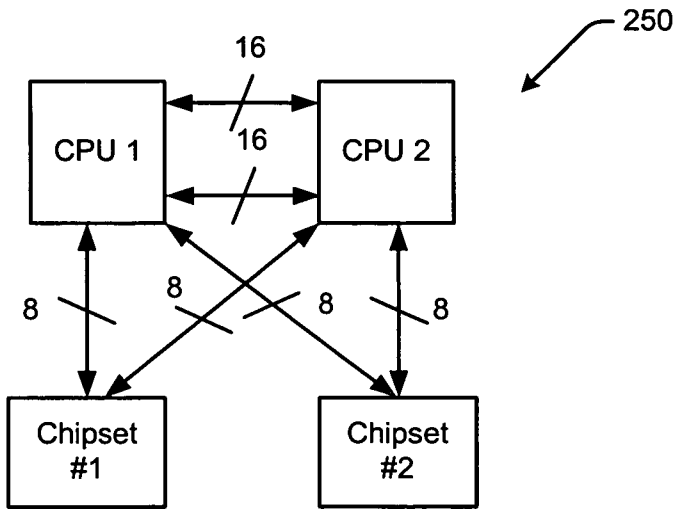
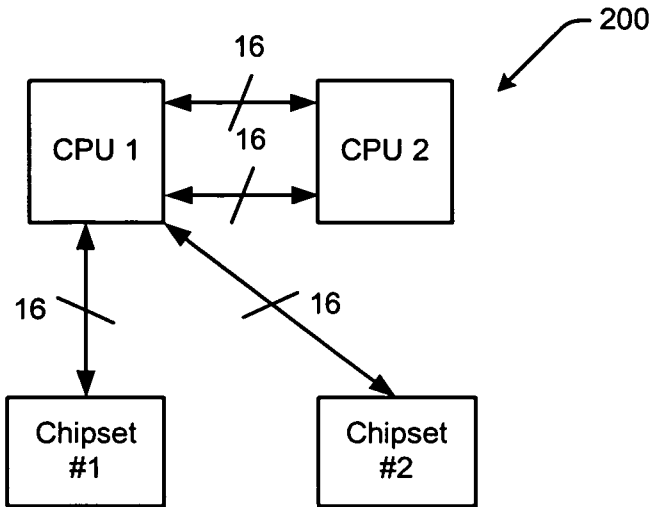
*Fig. 16*

Fig. 2



5/7

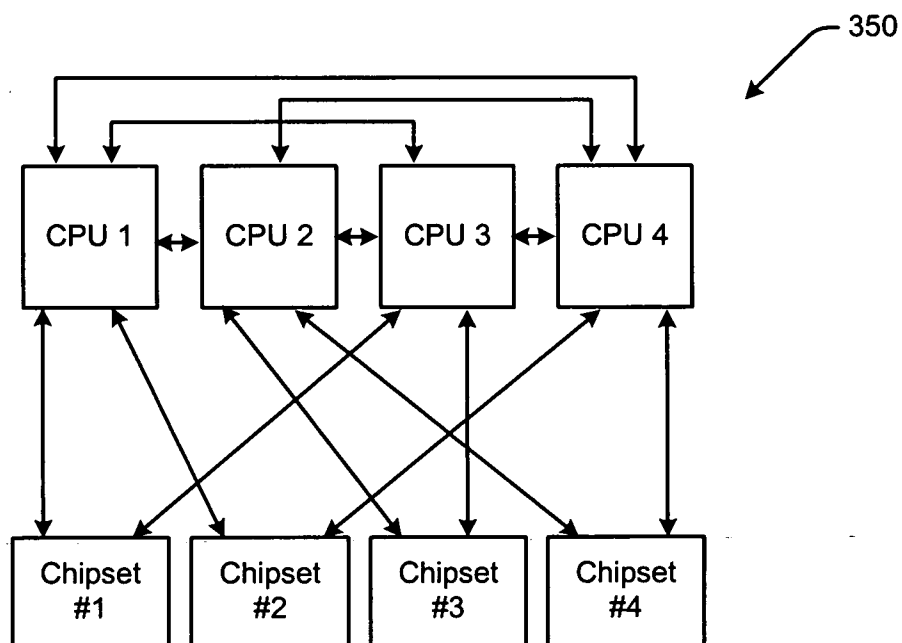
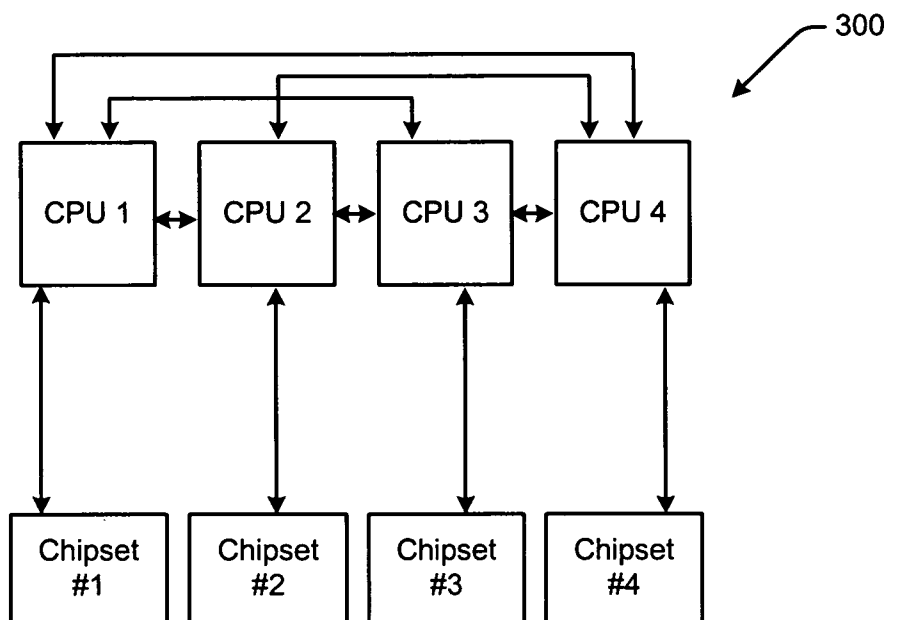
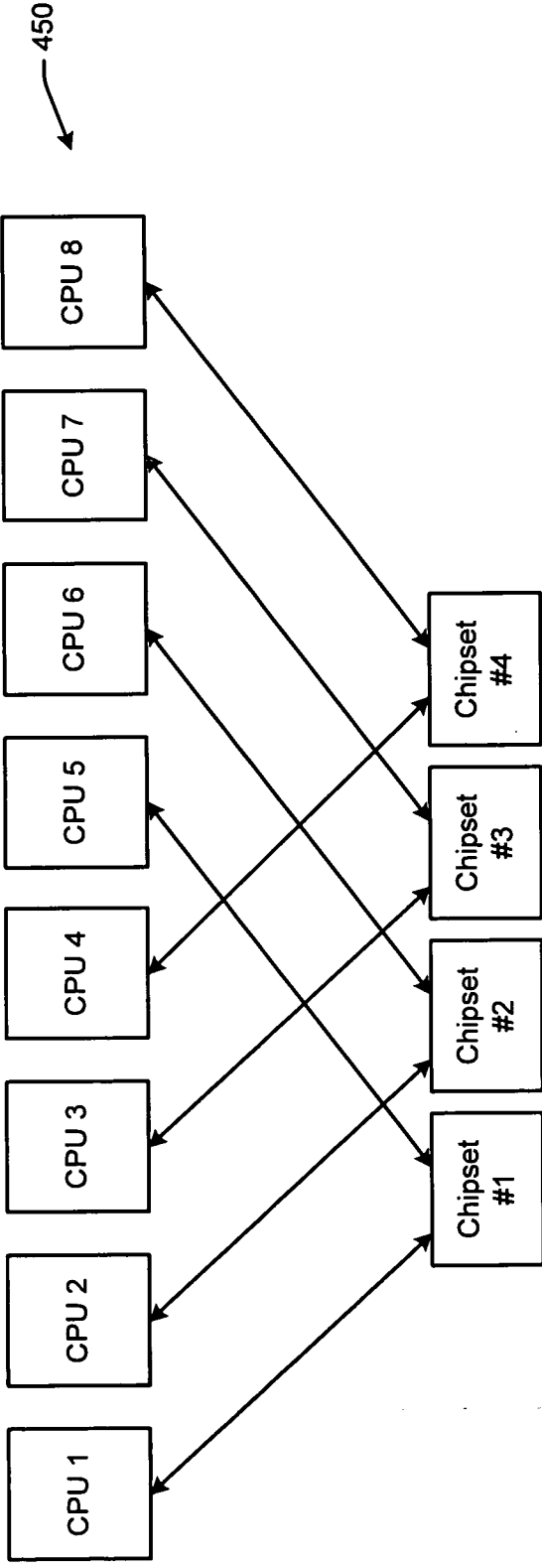
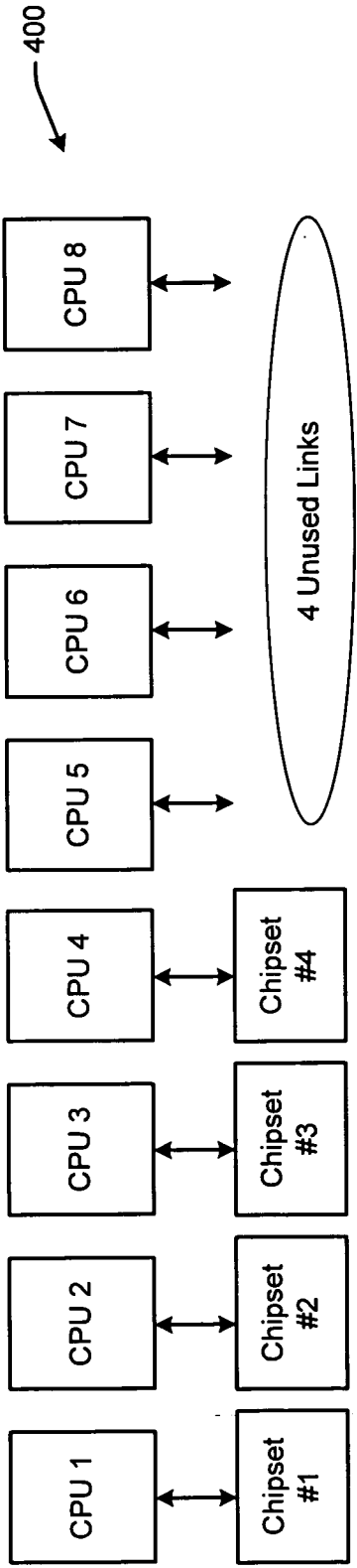
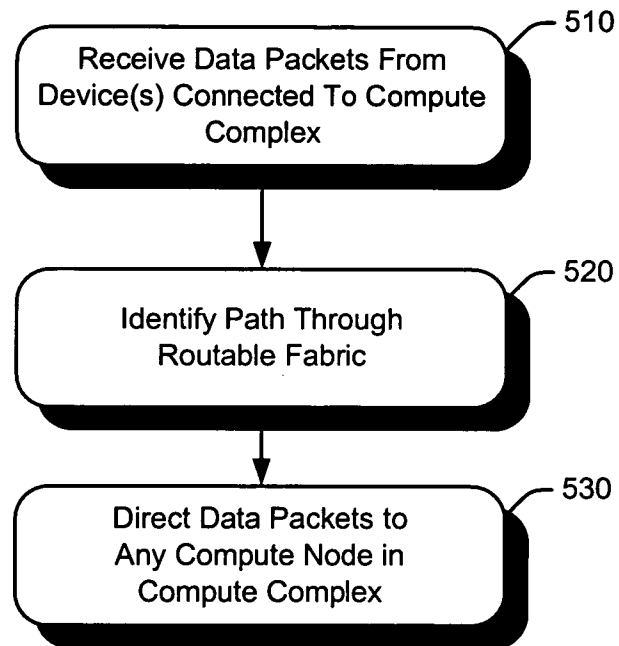
*Fig. 3*

Fig. 4



7/7

*Fig. 5*500

**A. CLASSIFICATION OF SUBJECT MATTER****G06F 15/16(2006.01)i**

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

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

IPC 8 G06F

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 since 1975

Japanese Utility models and applications for Utility Models since 1975

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

eKIPASS(KIPO internal) &amp; keywords : "dual, multi\*", "chipset\*", "bridge control"

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

| Category*       | Citation of document, with indication, where appropriate, of the relevant passages                                                  | Relevant to claim No.    |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| X<br>-----<br>Y | EP 1088272 B1 (SUN MICROSYSTEMS, INC.) 23 OCTOBER 2002<br>see paragraphs [0023]-[0025], [0029]-[0032], [0039], [0074], figures 2, 6 | 1, 5-9<br>-----<br>4, 10 |
| Y               | WO 2004-062209 A2 (ADVANCED MICRO DEVICES INC.) 22 JULY 2004<br>see abstract                                                        | 4, 10                    |
| A               | US 2002-0156958 A1 (CHUNG-CHING HUANG) 24 OCTOBER 2002<br>see abstract, figure 2 and its related description                        | 1-10                     |
| A               | KR 10-1999-0012939 A (SAMSUNG ELECTRONICS CO., LTD.) 25 FEBRUARY 1999<br>see figures 2-3 and their related descriptions             | 1-10                     |
| A               | KR 10-2001-0088224 A (SAMSUNG ELECTRONICS CO., LTD.) 26 SEPTEMBER 2001<br>see abstract, claim 1 and its related description         | 1-10                     |



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 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

"&amp;" document member of the same patent family

Date of the actual completion of the international search

23 DECEMBER 2008 (23.12.2008)

Date of mailing of the international search report

**23 DECEMBER 2008 (23.12.2008)**

Name and mailing address of the ISA/KR

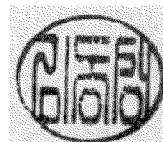
Korean Intellectual Property Office  
Government Complex-Daejeon, 139 Seonsa-ro, Seo-  
gu, Daejeon 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

KIM, Dong Sung

Telephone No. 82-42-481-8519



**INTERNATIONAL SEARCH REPORT**

Information on patent family members

International application No.

**PCT/US2008/006752**

| Patent document<br>cited in search report | Publication<br>date | Patent family<br>member(s) | Publication<br>date |
|-------------------------------------------|---------------------|----------------------------|---------------------|
| EP 1088272 B1                             | 23.10.2002          | DE 69903630 T2             | 31.07.2003          |
|                                           |                     | EP 1088272 A1              | 04.04.2001          |
|                                           |                     | JP 2002-518739 A           | 25.06.2002          |
|                                           |                     | KR 10-2001-0052813 A       | 25.06.2001          |
|                                           |                     | US 6173351 B1              | 09.01.2001          |
|                                           |                     | WO 99-66407 A1             | 23.12.1999          |
| WO 2004-062209 A2                         | 22.07.2004          | AU 2003-291317 A1          | 29.07.2004          |
|                                           |                     | CN 1729662 A               | 01.02.2006          |
|                                           |                     | EP 01573978 A2             | 14.09.2005          |
|                                           |                     | JP 2006-511878 A           | 06.04.2006          |
|                                           |                     | KR 10-2005-0085642 A       | 29.08.2005          |
|                                           |                     | US 2004-0122973 A1         | 24.06.2004          |
| US 2002-0156958 A1                        | 24.10.2002          | WO 2004-062209 A3          | 20.01.2005          |
|                                           |                     |                            |                     |
|                                           |                     |                            |                     |
| US 2002-0156958 A1                        | 24.10.2002          | TW 498213 B                | 11.08.2002          |
|                                           |                     | TW 498213 A                | 11.08.2002          |
|                                           |                     | US 6711642 B2              | 23.03.2004          |
| KR 10-1999-0012939 A                      | 25.02.1999          |                            |                     |
|                                           |                     |                            |                     |
|                                           |                     |                            |                     |
| KR 10-1999-0012939 A                      | 25.02.1999          | NONE                       |                     |
|                                           |                     |                            |                     |
|                                           |                     |                            |                     |
| KR 10-2001-0088224 A                      | 26.09.2001          |                            |                     |
|                                           |                     |                            |                     |
|                                           |                     |                            |                     |
| KR 10-2001-0088224 A                      | 26.09.2001          | NONE                       |                     |
|                                           |                     |                            |                     |
|                                           |                     |                            |                     |