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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 January 2018 (25.01.2018)

WIPO | PCT

(10) International Publication Number
WO 2018/017208 A1

(51) International Patent Classification:
*G06F 13/42* (2006.01)

(21) International Application Number:
PCT/US2017/037408

(22) International Filing Date:
14 June 2017 (14.06.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

| | | |
|---|---|---|
| 62/365,969 | 22 July 2016 (22.07.2016) | US |
| 62/376,859 | 18 August 2016 (18.08.2016) | US |
| 62/427,268 | 29 November 2016 (29.11.2016) | US |
| 15/396,501 | 31 December 2016 (31.12.2016) | US |


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

(72) Inventors: **WAGH, Mahesh**; 13527 NW Hogan St, Portland, Oregon 97229 (US). **MYERS, Mark S.**; 1414 SW 3rd Ave, Apt 2904, Portland, Oregon 97201 (US). **VAN DOREN, Stephen R.**; 8927 NW Bartholomew Drive, Portland, Oregon 97229 (US). **ZIAKAS, Dimitrios**; Jones Farm 5, Hillsboro, Oregon 97124 (US). **COURY, Bassam**; 2800 Center Drive N., Dupont, Washington 98327 (US).

(74) Agent: **DYER, Richard A**; Kacvinsky Daisak Bluni PLLC, 430 Davis Drive, Suite 150, Morrisville, North Carolina 27560 (US).

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

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

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

(54) Title: TECHNIQUES TO SUPPORT MULTIPLE INTERCONNECT PROTOCOLS FOR A COMMON SET OF INTERCONNECT CONNECTORS

(57) Abstract: Embodiments may be generally direct to apparatuses, systems, method, and techniques to determine a configuration for a plurality of connectors, the configuration to associate a first interconnect protocol with a first subset of the plurality of connectors and a second interconnect protocol with a second subset of the plurality of connectors, the first interconnect protocol and the second interconnect protocol are different interconnect protocols and each comprising one of a serial link protocol, a coherent link protocol, and an accelerator link protocol, cause processing of data for communication via the first subset of the plurality of connectors in accordance with the first interconnect protocol, and cause processing of data for communication via the second subset of the plurality of connector in accordance with the second interconnect protocol.

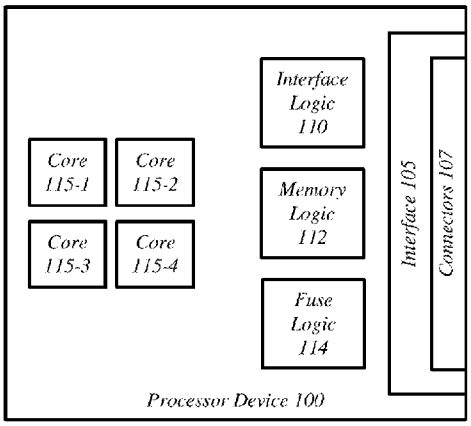



*FIG. 1*

WO 2018/017208 A1

# TECHNIQUES TO SUPPORT MULTIPLE INTERCONNECT PROTOCOLS FOR A COMMON SET OF INTERCONNECT CONNECTORS

## CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of United States Patent Application Serial No. 15/396,501 filed December 31, 2016, entitled "TECHNIQUES TO SUPPORT MULTIPLE INTERCONNECT PROTOCOLS FOR A COMMON SET OF INTERCONNECT CONNECTORS," which claims priority to United States Provisional Patent Application Number 62/365,969, filed July 22, 2016, United States Provisional Patent Application Number 62/376,859, filed August 18, 2016, and United Provisional Patent Application Number 62/427,268, filed November 29, 2016, each of which are hereby incorporated by reference in their entirety.

## TECHNICAL FIELD

Embodiments described herein generally include techniques to support multiple interconnect protocols for a common set of interconnect connectors.

## BACKGROUND

As computing systems are advancing, the components therein are becoming more complex. As a result, the interconnect architecture to couple and communicate between the components is also increasing in complexity to ensure bandwidth and latency requirements are met for optimal component operation. Furthermore, different market segments need different interconnect architectures to suit the market's needs and different interconnect connections. For example, these computing systems may provide various processing capabilities that require different add-in cards having physical resources. These add-in cards that may be coupled with the baseboard and may require any number of different interconnect protocols. However, connector space on the baseboard may be limited and a single connector typical supports on a single or limited number of interconnect protocols. Thus, embodiments may be directed to solving these and other problems.

## BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention are illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings in which like reference numerals refer to similar elements.

**FIG. 1** illustrates an example of a processor device.

**FIGs. 2A-2D** illustrate examples of a computer system.

**FIG. 3A** illustrates an example of an interconnect protocol layer system.

**FIG. 3B** illustrates an example of another interconnect protocol layer system.

**FIG. 4A** illustrates an example of a first logic flow diagram.

**FIG. 4B** illustrates an example of second and third logic flow diagrams.

**FIG. 5** illustrates an example of a forth flow diagram.

**FIG. 6** illustrates an example of a computer architecture.

## DETAILED DESCRIPTION

Various embodiments may be generally directed to providing multiple interconnect protocols for a common set of interconnect connectors. For example, embodiments may include determining a configuration for a plurality of connectors, the configuration to associate a first interconnect protocol with a first subset of the plurality of connectors and a second interconnect protocol with a second subset of the plurality of connectors. In some instances, the configuration may associate an interconnect protocol, such as a serial link protocol, a coherent link protocol, and/or an accelerator link protocol, with the first and second subset of connector. The configuration may be based on the setting of one or more fuses, an auto discovery operation, and platform straps, as will be discussed in more detail below.

Further, embodiments causing processing of data for communication via the first subset of the plurality of connectors in accordance with the first interconnect protocol and causing processing of data for communication via the second subset of the plurality of connector in accordance with the second interconnect protocol. Said differently, embodiments include directing data for processing by a particular interconnect protocol stack based on the configuration. The interconnect protocol stacks may be specific to the interconnect protocol. Examples of interconnect protocol stacks may include a serial link protocol stack, a coherent link protocol stack, and an accelerator link protocol. Thus, embodiments including processing data using one of these or other interconnect protocols based on the configuration of the connectors. Moreover, embodiments may include sending and receiving data via the plurality of connectors based on a configuration. Embodiments are not limited in this manner. These and other details will become more apparent in the following description.

Reference is now made to the drawings, wherein like reference numerals are used to refer to like elements throughout. In the following description, for purposes of explanation, numerous specific details are set forth to provide a thorough understanding thereof. It may be evident, however, that the novel embodiments can be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form to facilitate a description thereof. The intention is to cover all modifications, equivalents, and alternatives consistent with the claimed subject matter.

Reference is now made to the drawings, wherein like reference numerals are used to refer to like elements throughout. In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding thereof. It may be

evident, however, that the novel embodiments can be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form to facilitate a description thereof. The intention is to cover all modifications, equivalents, and alternatives consistent with the claimed subject matter.

**FIG. 1** illustrates a general overview of a processor device 100 which may be part of the system, such as a computer system, compute system, networking system, distributed system, and so forth. In some instances, the processor device 100 may be any type of computational element, such as but not limited to, a microprocessor, a processor, central processing unit, digital signal processing unit, dual-core processor, a quad-core processor, a multi-core processor, mobile device processor, desktop processor, single core processor, a system-on-chip (SoC) device, complex instruction set computing (CISC) microprocessor, a reduced instruction set (RISC) microprocessor, a very long instruction word (VLIW) microprocessor, a field-programmable gate array (FPGA) circuit, application specific integrated circuit (ASIC) or any other type of processor or processing circuit on a single chip or integrated circuit. The processor device 100 may have a number of elements, including one or more cores 115, interface logic 110, memory logic 112, fuse logic 114, and one or more interfaces 105 having a number of connectors 107.

In the illustrated example, the processor device 100 includes four cores 115-1 through 115-4. However, embodiments are not limited in this manner, and the processor device 100 may include any number of cores including a single core. Moreover, the multiple core design can integrate different types of processor cores on the same integrated circuit (IC) die, *e.g.* in a heterogeneous design. Thus, the one or more cores 115-1 through 115-4 may be different from each other. In some instances, each core of the multiple core design may be the same in a homogeneous.

The cores 115 of the processor device 100 can read and execute program instructions. The instructions are typically central processing unit CPU instructions (such as add, move data, and branch). The multiple core design enables the processor device 100 to execute or run multiple instructions at the same time, increasing overall speed for programs and applications. In some instances, the cores 115 may be integrated onto a single integrated circuit die (known as a chip multiprocessor or CMP), or onto multiple dies in a single chip package. Also, the processor device 100 with a multiple core design can be implemented as symmetrical or asymmetrical multiprocessors.

In some embodiments, the processor device 100 includes an interface 105 and connectors 107. The connectors 107 and interface 105 may provide physical connections to couple with other devices, such as interface components, memory, processing cards, networking interface components, accelerator cards, and so forth. The interface 105 and connectors 107 can include

one or more wires or signal traces capable of communicating information and data via electrical signaling. In some instances, the interface 105 and connectors 107 may be coupled with a physical slot capable of accepting processing cards. These processing cards typically provide additional processing and memory, which may be directed to a specific task, *e.g.* graphics processing, network processing, storage processing, interface processing, and so forth. As will be discussed in more detail below, the interface 105 and connectors 107 provide a common set of pins that support communication via a number of interconnect protocols.

In embodiments, the interface 105 and connectors 107 may be part of and enable one or more interconnects or logical links to send and receive data. The information and data communicated via the interconnect, and logical links may include data, control messages, interrupts, and so forth. These interconnect include a number or lanes, such as signaling pairs, and each lane may be made up of four wires or signal traces of the connectors 107 to send and receive data with another device, for example. Embodiments are not limited in this manner. In other words, each connector 107 may be coupled to a single wire or trace and four connectors 107 may make up a lane. In one example, the interface 105 may include a total of twenty-six (26) lanes to communicate data with another device. However, embodiments are not limited in this manner, and an interface may include more or fewer lanes based on computing requirements, for example.

The processor device 100 also includes interface logic 110 to enable and cause communication of data via the interface 105 in accordance with one or more of the interconnect protocols, such as a serial link protocol, a coherent link protocol, an accelerator link protocol, or any other interconnect protocol. Examples of a serial link protocol may include a Peripheral Component Interconnect Express (PCIe or PCI-E) protocol, PCI protocol, Universal Serial Bus (USB) protocol, Serial Peripheral Interface (SPI) protocol, Serial AT Attachment (SATA) protocol, and any other protocol for communicating serially via a link. Examples of a coherent link protocol may include a high performance interconnect protocol, such as Intel's® QuickPath Interconnect (QPI) protocol and Intel's UltraPath Interconnect (UPI) protocol, and so forth. Further, examples of an accelerator link protocol may include Intel's® Optimized Accelerator Protocol (OAP), a networking accelerator protocol, a storage accelerator protocol, a display accelerator protocol, and so forth. Embodiments are not limited to these examples. In some embodiments, the interface logic 110 may include a multiplexer such that a common set of the connectors 107 of the interface 105 supports the interconnect protocols. Thus, different devices or cards requiring different interconnect protocols may use same interface 105 and connectors 107.

In some embodiments, the interface logic 110 may be configured such that the lanes of the interface 105 are capable of communicating via one or more of the interconnect protocols, such as one or more of the serial link protocols, the coherent link protocols, and the accelerator link protocols. Moreover, and in some instances, the interface logic 110 may enable and cause communications via the interface 105 and the connectors 107 in accordance with more than one interconnect protocol at a time. For example, a first subset of the connectors 107 may be configured by the interface logic 110 to communicate in accordance with a first interconnect protocol while a second subset of the connectors 107 may be configured by the interface logic 110 to communicate in accordance with a second interconnect protocol. The second interconnect protocol may be different than the first interconnect protocol. In another example, a third subset of the connectors 107 may be configured by the interface logic 110 to communicate in accordance with a third interconnect protocol. The third interconnect protocol may be different than the first and the second interconnect protocols.

In embodiments, the interconnect protocol used for communication via specific connectors 107 may be programmed by a user or a computer system statically, or dynamically. Further, the programming may occur by setting one or more fuses via the fuse logic 114 to configure the interface logic 110, for example. In some instances, the interconnect protocol for use by the interface logic 110 may be originally programmed at the time of manufacture, during startup of a compute system, or during run-time. Embodiments are not limited in this manner.

In embodiments, the interface logic 110 may be reprogrammed to reprovision one or more of the connectors 107 of the interface 105 from one interconnect protocol to another interconnect protocol. For example, the interface logic 110 may be reprogrammed or set such that one or more connectors 107 currently operating in accordance with a serial link protocol may be reprogrammed to operate in accordance with a coherent link protocol or an accelerator link protocol. In another example, the interface logic 110 may be reprogrammed such that one or more connectors 107 programmed to operate in accordance with a coherent link protocol operate in accordance with a serial link protocol or an accelerator link protocol. In a third example, the interface logic 110 may be reprogrammed such that one or more connectors 107 currently operating in accordance with an accelerator link protocol may operate in accordance with a coherent link protocol or a serial link protocol. Note that the programming and reprogramming may occur at the interface logic 110, at the interface 105, at the connector 107, or some combination thereof. In other words, setting and configurations changes may occur at any of the above-mentioned elements, *e.g.* interface logic 110, interface 105, and connectors 107. Thus, the same set of connectors 107 or pins may support different connection types and provide

communication via different interconnect protocols based on the interface logic 110. Embodiments are not limited to these examples.

In some embodiments, the interface logic 110 may be statically and dynamically programmed and reprogrammed to support the interconnect protocols for the connectors 107 or a portion thereof. In one example, the interface logic 110 may be programmed to operate in accordance with one or more of the interconnect protocols at the time of manufacturing and may not be changed once programmed. The interface logic 110 may be programmed based on the needs of the manufacturer and the devices incorporated in a computer system, for example. In another example, the interface logic 110 may be programmed at bootup or run-time of a computing system, *e.g.* initialization performed by a Basic Input/Output System (BIOS), startup or operation of an operating system, during application run-time, etc. This programming may occur during the first time a computer system boots/loads or during the first time a device is detected when coupled with one or more of the connectors 107, for example. In another example, the interface logic 110 may be reprogrammed while a compute system is running or operating based on a different device being coupled to one or more of the connectors 107. Embodiments are not limited in this manner.

The one or more connectors 107 may be programmed, statically and dynamically, by a user and by a computer system, as previously mentioned. For example, a graphical user interface (GUI) may be presented by a BIOS, operating system, or application to enable a user to set one or more interconnect protocols for one or more of the connectors 107. In some embodiments, one or more of a BIOS, operating system, or application may set one or more interconnect protocols for the one or more of the connectors 107 without user interaction. For example, a device coupled to the connectors 107 may be detected, *e.g.* via a signal from the device or a polling of the interface 105. One or more of the BIOS, operating system, and application may determine an appropriate interconnect protocol for the connectors 107 based on the detection, and configure one or more of the interface logic 110, interface 105, and connectors 107 accordingly. Embodiments are not limited in this manner.

In embodiments, the processor device 100 also includes memory logic 112, such as coherency and cache logic. The memory logic 112 may perform operations for a memory of the processor device 100, such as cache, or a different memory not on the same die as the processor device 100, for example. The memory logic 112 may enable a device coupled to the interface 105 via the connectors 107 to read and write data to and from memory. In some embodiments, the memory logic 112 may enable a coherency protocol for memory usage of the cores 115 of the processor device 100. Since there may be two or more processing elements or cores 115 working at the same time, it is possible that they simultaneously access the same memory

location of a cache. If one of the cores 115 changes data in a memory location, the memory logic 112 may notify all the other cores 115 of changes to shared values in memory, for example. Embodiments are not limited in this manner, and any one of a number of coherency protocols may be utilized by the memory logic 112.

In embodiments, the processor device 100 also includes fuse logic 114 which may be used to set one or more interconnect protocols for the interface logic 110. The fuse logic 114 may be on-die of the processor device 100 and user programmable. For example, the fuse logic 114 may be coupled to a fuse mapping to individual fuse buses to interact and program one or more fuses. The fuses may be set and control which of the interconnect protocols are enabled and provided by the interface logic 110 for each of the one or more connectors 107. For example, the fuse logic 114 may program one or more fuses to enable one or more of a serial link protocol, a coherent link protocol, and an accelerator link protocol for the connectors 107. The fuse logic 114 may program one or more fuses based on user input. For example, the fuses to configure an interconnect protocol for the interface logic 110 may be programmed by a user at the time of manufacturer or during run-time. In another example, a user may program the fuses to set an interconnect protocol for the interface logic 110 via at least one of a BIOS, an operating system, and an application. The fuse logic 114 may also program one or more fuses to set an interconnect protocol based on signals received from the BIOS, an application, or operating system directly. In some instances, a discovery operation may occur to detect a device or a card to configure the interface logic 110. The discovery operation may include automatically detecting a card or device plugged into a slot, for example. In another example, platform straps may be used to determine a configuration of the interface logic 110. The platform straps may include detecting signalling on one or more pins during a bootup, for example.

In some embodiments, a combination of one or more methods maybe used to configure the interface logic 110. For example, fuse logic 114 and a discovery operation maybe used to set the interface logic 110. Fuses might be used to enable a link as PCIe/OAP (*i.e.* Flexbus® with no UPI) link, and an auto discovery operation can be used to determine if an attached device is PCIe or OAP. Note that in some instances, the fuse logic 114 may also be used to disable or disallow certain interconnect protocols. A fuse could, for example, prevent a link being transition to an OAPmode orUPI mode. Therefore effectively render a link PCIe only.

**FIGs 2A-2D** illustrates various configurations of a computer system 200 having one or more interconnect protocols enabled for a processor device 100. FIG. 2A illustrates computer system 200 having a processor device 100 coupled with a BIOS 216 and a plurality of interconnects 205, which may include a number of lanes over the connectors 107 or pins. The

interconnects 205 may be logical connections coupling one or more other devices, components, elements, and so forth with the processor device 100.

In embodiments, the interconnects 205 may operate in accordance with the one or more interconnect protocols, *e.g.* a serial link protocol, a coherent link protocol, and an accelerator link protocol. Moreover, the interconnects 205 may include a serial link interconnect and operate in accordance with a serial link protocol, a coherent link interconnect and operate in accordance with a coherent link protocol, an accelerator link interconnect and operate in accordance with an accelerator link protocol, and so forth.

FIG. 2B illustrates one example configuration of a computer system 200 having a number of connectors 107 set to operate as a serial interconnect 205-1 and a coherent interconnect 205-2. The serial interconnect 205-1 may include a number of lanes, such as sixteen (16) lanes for PCIe providing thirty-two (32) Gigabytes (GB)/second (s) bandwidth or throughput. Similarly, the coherent interconnect 205-2 may include a number of lanes, such as twenty-four (24) to provide 40.8 GB/s throughput. In this configuration, the interface logic 110 may enable communication of data via the serial interconnect 205-1 in accordance with a serial link protocol and via the coherent interconnect 205-2 in accordance with a coherent link protocol. This configuration may be based on setting one or more fuses by the fuse logic 114. Moreover, the settings may be set at a time of manufacturer by programming one or more fuses (or antifuses) by using known programming techniques, such as "blowing" one or more fuses by applying a high voltage pulse. In some embodiments,the settings may be set at run-time or boot time of the computer system 200 by configuring information in the BIOS 216, a operating system configuration, or an application configuration. The settings may be set statically or dynamically. In another example,the configuration may be hardcoded into the BIOS 216 or the configuration could be stored in a configuration read-only memory (ROM) and passed to the BIOS 216 by power management code. Embodiments are not limited in this manner. Note that FIG. 2B illustrates only one of many different configurations that may be applied to the interconnects 205.

For example, FIG. 2C illustrates another example configuration of a computer system 200 having a number of connectors 107 set to operate as a serial interconnect 205-1, a coherent interconnect 205-2, and an accelerator interconnect 205-3. The serial interconnect 205-1 and coherent interconnect 205-2 may include a number of lanes as previously discussed. The accelerator interconnect 205-3 may also include a number of lanes, such as twenty-six (26) providing forty-eight (48) GB/sec throughput. In this configuration, the interface logic 110 may enable communication of data via the serial interconnect 205-1 in accordance with a serial link protocol, via the coherent interconnect 205-2 in accordance with a coherent link protocol, and

via the accelerator interconnect 205-3 in accordance with an accelerator link protocol. This configuration may be based on setting one or more fuses by the fuse logic 114.

FIG. 2D illustrates another example configuration of a computer system 200 having a number of connectors 107 programmed to operate as a serial interconnect 205-1. In this illustrated example, the serial interconnect 205-1 may provide additional lanes compared to what is illustrated in FIGs. 2B and 2C. The additional lanes configured as a serial interconnect 205-1 may provide additional bandwidth for additional devices to couple with the processor device 100 via serial links. For example, the serial interconnect 205-1 illustrated in FIG. 2D may enable two or more devices to couple with processor device 100 via serial interconnects 205-1. Further, the serial interconnect 205-1 in FIG. 2D may provide two sets of 16 lanes, each operating at 32GB/s, for example. As previously discussed, the interconnects 205-1 and 205-2 may be programmed and reprogrammed statically and dynamically and based on setting one or more fuses by the fuse logic 114, for example. Embodiments are not limited to these examples illustrated in FIGs 2A-2D. Other configurations may be set statically or dynamically based on the needs of a user or manufacture for a computer system.

**FIG. 3A** illustrate an embodiment of a interconnect protocol stack 300. The interconnect protocol stack 300 may generally include or represent the previously discussed interconnect protocols, *e.g.* a serial link protocol, a coherent link protocol, and an accelerator link protocol. Embodiments are not limited in this manner.

The interconnect protocol stack 300 may include a number of layers, such as a transaction layer 302, a link layer 304, and a physical layer (PHY) 306. In embodiments, portions of the interconnect protocol stack 300 may be implemented as part of the interface logic 110, the interface 105, connectors 107, or combination thereof. However, embodiments are not limited in this manner, and portions of the interconnect protocol stack 300 may be implemented in different elements of the processor device 100.

In some embodiments, the interconnect protocol stack 300 and interconnect protocols may communicate data between the coherent fabric 310 and a device. The coherent fabric 310 may connect and include the cores 115, the memory logic 112, memory, the processor cache, and so forth with the interface logic 110. The coherent fabric 310 may include protocol agents, as will be discussed in more detail below in FIG. 3B. The transaction layer 306 may handle data and action requests and messages. The transaction layer 306 may parse the action requests and messages and initiates the appropriate actions in the processor's memory system according to protocol specific rules, such as ordering rules. The transaction layer 306 may also process data and action requests which may include read and write instructions. Action requests may also include cache coherency actions for UPI and OAP, for example, and address transaction actions

for PCIe, for example. The messages processed by the transaction layer 306 may include error messages, interrupts, and so forth.

The transaction layer 306 may provide an interface between the one or more cores 115, and interconnect architecture including at least portions of the PHY layer 302, which may include the interface 105, and connectors 107 coupled to another device. The transaction layer 306 may also communicate information between the cores 115 and the processor's memory system and another device via the link layer 304 and PHY layer 302 in transaction layer packets (TLPs). As mentioned, this information may include memory reads, memory writes, input/output (I/O), I/O writes, messages, completion, and so forth.

The link layer 304, also referred to as a data link layer, acts as an intermediate stage between the transaction layer 306 and the PHY 302. In one embodiment, the link layer 304 may provide a reliable mechanism for exchanging TLPs between two components in a link. The link layer 304 may append information, *e.g.* packet sequence identification, to the TLPs when sending data and may remove the information from packets when receiving data. The link layer 304 may also determine and append an error detection code (CRC) to the packet header/payload of the TLPs. The link layer 304 sends the modified TLPs to the PHY 306 for transmission across a physical, *e.g.* interface 105 and connectors 107, to an external device.

In one embodiment, the interconnect protocol stack 300 may also include a PHY 302, which may include a logical sub-block 314 and an electrical sub-block 318 to physically transmit a packet to an external device. In some embodiments, the PHY 302 may include portions of the interface logic 110, the interface 105, and the connectors 107 or pins, as illustrated in FIGs. 1 and 2A-2D.

In some instances, the logical sub-block 314 may be divided into a media access control (MAC) sublayer and a physical coding sublayer (PCS). In some instances, the PHY Interface for PCI Express (PIPE), published by Intel® Corp., defines the MAC/PCS functional partitioning and the interface between these two sub-layers. The PIPE specification also identifies the physical media attachment (PMA) layer, which includes the serializer/deserializer (SerDes) circuitry and other analog circuitry.

The logical sub-block 314 may also responsible for the logical functions of the PHY 302. The logical sub-block 314 includes a buffer that may function either as a drift buffer or an elastic buffer. Further, the logical sub-block 314 includes a data encoding section, which can encode data using a 128b/130b transmission code, where 130-bit symbols are transmitted/received. In some embodiments, the logical sub-block 314 includes a transmit section to prepare outgoing information for transmission by electrical sub-block 318, and a receiver section to identify and prepare received information before passing it to the link layer 304. The electrical sub-block 318

includes a transmitter and a receiver to send and receive data. The transmitter is supplied by logical sub-block 314 with symbols and transmits on to an external device. The receiver is supplied with symbols from an external device and transforms the received signals into a bit-stream. The bit-stream is supplied to the logical sub-block 314.

**FIG. 3B** illustrates an embodiment of an interconnect block diagram 350 to support multiple interconnect protocols on a common set of pins or connectors 107 for a processor device 100. In the illustrated example, the interface logic 110 may control and determine one or more interconnect protocols, *e.g.* a serial link protocol, a coherent link protocol, and an accelerator link protocol, to support sending and receiving information and data between the coherent fabric 310 components and another device coupled via the PHY layer 302. In embodiments, each interconnect protocol may include a separate interconnect protocol stack having a transaction layer 306, a link layer 304 and at least a portion of the PHY layer, *i.e.* the PHY layer logic 314 or Logical PHY. The interconnect protocols may share the electrical sub-block 318, *i.e.* the Analog PHY, of the PHY layer 302. Further, a multiplexer 362 may be between the electrical sub block 318 and each of the interconnect protocols stacks to direct information and data to the appropriate interconnect protocol stack based on a configuration, as previously discussed.

In embodiments, each of the interconnect protocol stacks may be coupled with a protocol agent which may be part of the coherent fabric 310. The protocol agents, including but not limited to, a serial link protocol agent 352, a coherent link protocol agent 354, and an accelerator link protocol agent 356 may be activated and deactivated based on the configuration, as discussed above. Activating a protocol agent may set up various protocol maps and routing tables in the coherent fabric 310 to steer action requests and messages associated with the selected protocol to the activated protocol agent and steer action requests and messages from the activated protocol agent to other components of the coherent fabric 310, *e.g.* cores 115, processor cache, memory logic 112, memory, and so forth.

In one example, if the serial link protocol agent 352 is activated, one or more protocol maps and routing tables may be configured to send action requests and messages from components of the coherent fabric 310 to the serial link protocol agent 352 for further processing by the serial link protocol stack. The serial link protocol stack may include the transaction layer 306-1, the link layer 304-1, the PHY layer logic 314-1. The layers 306-1, 304-1, and 314-1 may process information and cause communication to another device via the electrical sub-block 318 over one or more connectors 107. Similarly, information received via one or more connectors 107 configured for a serial link may be processed by electrical sub-block 318, PHY layer logic 314-1, link layer 304-1, and the transaction layer 306-1. The serial link protocol agent 352 may

send the information including action requests and messages to components of the coherent fabric 310.

In a second example, the coherent link protocol agent 354 and coherent link stack may process information in a similar manner as the serial link protocol agent 352. For example, when the coherent link protocol agent 354 is activated, one or more protocol maps and routing tables are configured to send information from the components of the coherent fabric 310 to the coherent link protocol agent 354 for processing by the transaction layer 306-2, link layer 304-2, the PHY layer logic 314-2, and for communication via a device coupled with connectors 107 via the electrical sub block 318. Information received via the connectors 107 configured for a coherent link may be processed by electrical sub block 318, PHY layer logic 314-2, link layer 304-2, and the transaction layer 306-2. The coherent link protocol agent 354 may send the information including action requests and messages to components of the coherent fabric 310.

The accelerator link protocol agent 356 may operate in a similar manner when activated. Information from the components of the coherent fabric 310 sent to the accelerator link protocol agent 356 may be processed by the transaction layer 306-3, link layer 304-3, the PHY layer logic 314-3, and communicated to a device coupled with connectors 107 via the electrical sub-block 318. Information received via the connectors 107 configured for an accelerator link may be processed by electrical sub-block 318, PHY layer logic 314-3, link layer 304-3, and the transaction layer 306-3. The accelerator link protocol agent 356 may send the information including action requests and messages to components of the coherent fabric 310.

As previously discussed, the configuration of the interface logic 110 may be determined by setting one or more fuses in the fuse logic 114. In some embodiments, interface logic 110 includes a multiplexer 362 configured to cause the application of a particular interconnect protocol for particular connectors 107 based on one or more settings in fuse logic 114, for example. More specifically, one or more fuses (or antifuses) may be set and configure the multiplexer and activate the serial link protocol agent 352 to send and receive data via the first set of connectors 107. In another example, one or more fuses (or antifuses) may be set to activate a coherent link protocol agent 354 and configure a multiplexer 362 to send and receive data via the second set of connectors 107. In the third example, one or more fuses (or antifuses) may be set to activate an accelerator link protocol agent 356 and configure a multiplexer 362 to communicate information vai the third set of connectors 107. Embodiments are not limited to these examples, and different interconnect protocols may be applied to one or more of the sets of connectors 107. Moreover, the fuse logic 110 may be configured such that one or more of the protocol agents are deactivated. These fuses and configurations may be changed and

reprogrammed to cause different interconnect protocols to be applied to one or more sets of the connectors 107.

In operation, a processor device 100 may receive data or information via one or more connectors 107 from a coupled device. A coupled device may be any type of device that may send information to a processor device 100 for processing by one or more cores 115 and a coherent fabric that may connect the cores 115, the processor's memory system including the memory logic 112 and memory, the interface logic 110 and interconnect protocol stack. The PHY layer 302 may receive the data or information and process it in accordance with the appropriate interconnect protocol stack. The interconnect protocol stack used may be based on the one or more connectors 107 or pins on which the data or information was received on and the multiplexer 362.

The PHY layer logic 314 may receive the information or data and perform decoding and descramble to ensure the integrity of the information or data is maintained. Different interconnect protocols may use different encoding/decoding techniques to ensure bit integrity. For example, PCIe 2.0 utilizes 8b/10b encoding meaning that for every eight payload bits or encoded with ten encoded bits to transmit data, causing a 20% overhead. In another example, PCIe 3.0 utilizes 128b/130b encoding using less overhead. The PHY layer logic 314 may perform processing based on the interconnect protocol utilized. Embodiments are not limited in this manner.

The PHY layer logic 314 may pass or send the data or information to other layers of the interconnect protocol stacks, such as the link layer 304 and the transaction layer 306 for further processing. Each of the layers may process the data or information and in accordance with a particular interconnect protocol. The information or data may be sent to an appropriate protocol agent and coherent fabric 310 for processing.

In a similar manner, a processor device 100 may send or communicate data or information to another device. For example, information or data may be received by a coherent fabric 310 and an appropriate protocol agent based on configured mappings and routing tables. The information or data may be processed by the interconnect protocols layers including the transaction layer 306, the link layer 304, and the PHY layer logic 314, as previously discussed above. The information and data may be communicated to the destination device by the multiplexer 362 and the electrical sub-block 318 via the appropriate connectors 107.

**FIG. 4A** illustrates a first logic flow diagram 400 for configuring interface logic 110 and one or more connectors 107 with one or more interconnect protocols. Although logic flow diagram 400 illustrates certain operations are occurring in a particular order, embodiments are not limited in this manner. Some operations may occur before or after other operations, and some

may occur in parallel. Moreover, the logic flow 400 may be representative of some or all of the operations executed by one or more embodiments described herein.

At block 402, embodiments include determining a configuration for interconnects and which interconnect protocols are active and de-active for one or more connectors. In embodiments, the determination may be made based on the configuration of fuse logic and the setting of one or more fuses. The configuration may also be determined based on a discovery operation performed to detect a card or device coupled with the interconnects. For example, a discovery operation may be performed and detect that a device is a PCIe device, an OAP device and so forth. In some embodiments, the configuration may also be determined or detected using platform straps which may include signaling on one or more pins or connectors during a bootup.

At block 404, the logic flow 400 includes activating and deactivating protocol agents based on the configuration of the system. For example, if the configuration includes having one or more connectors are to be configured as a serial link, the serial link protocol agent may be activated. Similarly, if one or more connectors 107 are to be configured as coherent link, the coherent link protocol agent is activated. The accelerator link protocol agent may be activated if one or more connectors are to be configured as an accelerator link. If the one or more connectors are not to be configured as a particular link, *e.g.* serial link, coherent link, and accelerator link, the corresponding protocol may be deactivated. For example, if the connectors are not to be configured as serial links, the serial link protocol agent may be activated.

At block 406, the logic flow 400 may include configuring one or more protocol maps and routing tables based on the active protocol agents. For example, protocol maps and routing tables may be configured to steer request and messages to the serial link protocol agent, and the appropriate interconnect protocol layers if one or more of the connectors are configured as a serial link. Similarly, protocol maps and routing tables may be configured to steer particular requests and messages to a coherent link protocol agent and an accelerator link protocol agent if one or more connectors are configured as coherent links and accelerator links. Embodiments are not limited in this manner.

**FIG. 4B** illustrates a second logic flow diagram 450 for receiving information via connectors and a third logic flow diagram 470 to send information to another device or component. Although logic flow diagrams 450, and 470 illustrate certain operations are occurring in a particular order, embodiments are not limited in this manner. Some operations may occur before or after other operations, and some may occur in parallel. Moreover, the logic flows 450 and 470 may be representative of some or all of the operations executed by one or more embodiments described herein.

At block 452, embodiment includes receiving information via an interface and one or more connectors. The data may be received from another device or component coupled with the one or more connectors, for example. At block 454, the logic flow 450 includes sending the information to the appropriate interconnect protocol stack. In one instance, interface logic may determine the interconnect protocol stack based on the connectors and a configuration, as discussed in FIG. 4A. In one example, the configuration may cause a multiplexer to steer the information to the appropriate interconnect protocol stack. For example, the information may be sent to a serial link protocol stack based on a configuration and processed by the PHY layer, a link layer, and a transaction layer associated with the serial link protocol stack. At block 458, the data may be processed in accordance with the interconnect protocol. Similarly, the coherent protocol stack and the accelerator protocol stack may be utilized based on the configuration.

At block 456, the information may be processed by the PHY layer, the link layer, and the transaction, as discussed in FIGs 3A and 3B. Also, the protocol agent for the interconnect protocol stack may determine one or more components of the coherent fabric to send the information. For example, embodiments may include communicating information to cores, memory, memory logic, and so forth. At block 458, information may be sent to the components of the coherent fabric.

Logic flow 470 may include receiving information from components to send to a device coupled vid connectors. At block 472, a protocol agent may receive information, *e.g.* requests, messages, and so forth, to send to a device from a component of a coherent fabric. The protocol agent, which may be one of a serial link protocol agent, a coherent link protocol agent, and an accelerator link protocol agent may send the information to a corresponding interconnect protocol at block 474. The information may be processed by the transaction layer, the link layer, and the PHY layer at block 476 and communicated to the device at block 478.

**FIG. 5** illustrates an example a first processing flow 500 for process data via one or more connectors. The processing flow 500 may be representative of some or all of the operations executed by one or more embodiments described herein. For example, the processing flow 500 is illustrated with the components illustrated in FIGs. 1 and 2A through 2D. However, embodiments are not limited in this manner.

At block 505, the processing flow 500 includes sending and receiving data via a plurality of connectors of a PHY layer interface. For example, data may be received from another device or component for processing by one or more cores for processing via the plurality of connectors. In another example, data may be sent to another device via the plurality of connectors. Embodiments are not limited in this manner.

At block 510, the logic flow 500 may include causing processing of the data for communication via a first subset of the plurality of connectors in accordance with a first interconnect protocol and causing processing of the data for communication via a second subset of the plurality of connector in accordance with a second interconnect protocol, the first interconnect protocol and the second interconnect protocol are different protocols and each comprising one of a serial link protocol, a coherent link protocol, and an accelerator link protocol.

**FIG. 6** illustrates an embodiment of an exemplary computing architecture 600 suitable for implementing various embodiments as previously described. In one embodiment, the computing architecture 600 may comprise or be implemented as part one or more systems and devices previously discussed.

As used in this application, the terms "system" and "component" are intended to refer to a computer-related entity, either hardware, a combination of hardware and software, software, or software in execution, examples of which are provided by the exemplary computing architecture 600. For example, a component can be, but is not limited to being, a process running on a processor, a processor, a hard disk drive, multiple storage drives (of optical and/or magnetic storage medium), an object, an executable, a thread of execution, a program, and/or a computer. By way of illustration, both an application running on a server and the server can be a component. One or more components can reside within a process and thread of execution, and a component can be localized on one computer and/or distributed between two or more computers. Further, components may be communicatively coupled to each other by various types of communications media to coordinate operations. The coordination may involve the uni-directional or bi-directional exchange of information. For instance, the components may communicate information in the form of signals communicated over the communications media. The information can be implemented as signals allocated to various signal lines. In such allocations, each message is a signal. Further embodiments, however, may alternatively employ data messages. Such data messages may be sent across various connections. Exemplary connections include parallel interfaces, serial interfaces, and bus interfaces.

The computing architecture 600 includes various common computing elements, such as one or more processors, multi-core processors, co-processors, memory units, chipsets, controllers, peripherals, interfaces, oscillators, timing devices, video cards, audio cards, multimedia input/output (I/O) components, power supplies, and so forth. The embodiments, however, are not limited to implementation by the computing architecture 600.

As shown in Figure 6, the computing architecture 600 comprises a processing unit 604, a system memory 606 and a system bus 608. The processing unit 604 can be any of various

commercially available processors, such as those described concerning the processing circuitry is shown in Figure 1A.

The system bus 608 provides an interface for system components including, but not limited to, the system memory 606 to the processing unit 604. The system bus 608 can be any of several types of bus structure that may further interconnect to a memory bus (with or without a memory controller), a peripheral bus, and a local bus using any of a variety of commercially available bus architectures. Interface adapters may connect to the system bus 608 via a slot architecture. Example slot architectures may include without limitation Accelerated Graphics Port (AGP), Card Bus, (Extended) Industry Standard Architecture ((E)ISA), Micro Channel Architecture (MCA), NuBus, Peripheral Component Interconnect (Extended) (PCI(X)), PCI Express, Personal Computer Memory Card International Association (PCMCIA), and the like.

The computing architecture 600 may comprise or implement various articles of manufacture. An article of manufacture may comprise a computer-readable storage medium to store logic. Examples of a computer-readable storage medium may include any tangible 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. Examples of logic may include executable computer program instructions implemented using any suitable type of code, such as source code, compiled code, interpreted code, executable code, static code, dynamic code, object-oriented code, visual code, and the like. Embodiments may also be at least partly implemented as instructions contained in or on a non-transitory computer-readable medium, which may be read and executed by one or more processors to enable performance of the operations described herein.

The system memory 606 may include various types of computer-readable storage media in the form of one or more higher speed memory units, such as read-only memory (ROM), random-access memory (RAM), dynamic RAM (DRAM), Double-Data-Rate DRAM (DDRAM), synchronous DRAM (SDRAM), static RAM (SRAM), programmable ROM (PROM), erasable programmable ROM (EPROM), electrically erasable programmable ROM (EEPROM), flash memory, polymer memory such as ferroelectric polymer memory, ovonic memory, phase change or ferroelectric memory, silicon-oxide-nitride-oxide-silicon (SONOS) memory, magnetic or optical cards, an array of devices such as Redundant Array of Independent Disks (RAID) drives, solid state memory devices (e.g., USB memory, solid state drives (SSD) and any other type of storage media suitable for storing information. In the illustrated embodiment shown in FIGURE 6, the system memory 606 can include non-volatile memory 610 and volatile memory 612. A basic input/output system (BIOS) can be stored in the non-volatile memory 610.

The computer 602 may include various types of computer-readable storage media in the form of one or more lower speed memory units, including an internal (or external) hard disk drive (HDD) 614, a magnetic floppy disk drive (FDD) 616 to read from or write to a removable magnetic disk 618, and an optical disk drive 620 to read from or write to a removable optical disk 622 (e.g., a CD-ROM or DVD). The HDD 614, FDD 616 and optical disk drive 620 can be connected to the system bus 608 by a HDD interface 624, an FDD interface 626 and an optical drive interface 628, respectively. The HDD interface 624 for external drive implementations can include at least one or both of Universal Serial Bus (USB) and IEEE 1394 interface technologies.

The drives and associated computer-readable media provide volatile and nonvolatile storage of data, data structures, computer-executable instructions, and so forth. For example, a number of program modules can be stored in the drives and memory units 610, 612, including an operating system 630, one or more application programs 632, other program modules 634, and program data 636. In one embodiment, the one or more application programs 632, other program modules 634, and program data 636 can include, for example, the various applications and components of the system 100.

A user can enter commands and information into the computer 602 through one or more wired/wireless input devices, for example, a keyboard 638 and a pointing device, such as a mouse 640. Other input devices may include microphones, infra-red (IR) remote controls, radio-frequency (RF) remote controls, game pads, stylus pens, card readers, dongles, finger print readers, gloves, graphics tablets, joysticks, keyboards, retina readers, touch screens (e.g., capacitive, resistive, etc.), trackballs, trackpads, sensors, styluses, and the like. These and other input devices are often connected to the processing unit 604 through an input device interface 642 that is coupled to the system bus 608, but can be connected by other interfaces such as a parallel port, IEEE 1394 serial port, a game port, a USB port, an IR interface, and so forth.

A monitor 644 or another type of display device is also connected to the system bus 608 via an interface, such as a video adaptor 646. The monitor 644 may be internal or external to the computer 602. In addition to the monitor 644, a computer typically includes other peripheral output devices, such as speakers, printers, and so forth.

The computer 602 may operate in a networked environment using logical connections via wired and wireless communications to one or more remote computers, such as a remote computer 648. The remote computer 648 can be a workstation, a server computer, a router, a personal computer, portable computer, microprocessor-based entertainment appliance, a peer device or other common network node, and typically includes many or all of the elements described relative to the computer 602, although, for purposes of brevity, only a memory/storage device 650 is illustrated. The logical connections depicted include wired/wireless connectivity to

a local area network (LAN) 652 and larger networks, for example, a wide area network (WAN) 654. Such LAN and WAN networking environments are commonplace in offices and companies, and facilitate enterprise-wide computer networks, such as intranets, all of which may connect to a global communications network, for example, the Internet.

When used in a LAN-networking environment, the computer 602 is connected to the LAN 652 through a wire and wireless communication network interface or adaptor 656. The adaptor 656 can facilitate wire and wireless communications to the LAN 652, which may also include a wireless access point disposed of thereon for communicating with the wireless functionality of the adaptor 656.

When used in a WAN-networking environment, the computer 602 can include a modem 658, or is connected to a communications server on the WAN 654, or has other means for establishing communications over the WAN 654, such as by way of the Internet. The modem 658, which can be internal or external and a wire and wireless device, connects to the system bus 608 via the input device interface 642. In a networked environment, program modules depicted relative to the computer 602, or portions thereof, can be stored in the remote memory/storage device 650. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers can be used.

The computer 602 is operable to communicate with wire and wireless devices or entities using the IEEE 802 family of standards, such as wireless devices operatively disposed of in wireless communication (e.g., IEEE 802.11 over-the-air modulation techniques). This includes at least WiFi (or Wireless Fidelity), WiMax, and Bluetooth™ wireless technologies, 3G, 4G, LTE wireless technologies, among others. Thus, the communication can be a predefined structure as with a conventional network or simply an ad hoc communication between at least two devices. WiFi networks use radio technologies called IEEE 802.11x (a, b, g, n, etc.) to provide secure, reliable, fast wireless connectivity. A WiFi network can be used to connect computers to each other, to the Internet, and to wire networks (which use IEEE 802.3-related media and functions).

The various elements and components as previously described with reference to FIGS. 1-5 may comprise various hardware elements, software elements, or a combination of both. Examples of hardware elements may include devices, logic devices, components, processors, microprocessors, circuits, processors, circuit elements (e.g., transistors, resistors, capacitors, inductors, and so forth), integrated circuits, application specific integrated circuits (ASIC), programmable logic devices (PLD), digital signal processors (DSP), field programmable gate array (FPGA), memory units, logic gates, registers, semiconductor device, chips, microchips, chip sets, and so forth. Examples of software elements may include software components, programs, applications, computer programs, application programs, system programs, software

development programs, machine programs, operating system software, middleware, firmware, software modules, routines, subroutines, functions, methods, procedures, software interfaces, application program interfaces (API), instruction sets, computing code, computer code, code segments, computer code segments, words, values, symbols, or any combination thereof. However, determining whether an embodiment is implemented using hardware elements and/or software elements may vary in accordance with any number of factors, such as desired computational rate, power levels, heat tolerances, processing cycle budget, input data rates, output data rates, memory resources, data bus speeds and other design or performance constraints, as desired for a given implementation.

The detailed disclosure now turns to providing examples that pertain to further embodiments. Examples one through twenty-five (1-25) provided below are intended to be exemplary and non-limiting.

In a first example, a system, a device, an apparatus, and so forth may include a physical (PHY) layer interface comprising a plurality of connectors configured to couple with one or more other devices, and to send and receive data via the plurality of connectors, and interface logic coupled with the PHY layer interface, the interface logic to cause processing of the data for communication via a first subset of the plurality of connectors in accordance with a first interconnect protocol and to cause processing of the data for communication via a second subset of the plurality of connector in accordance with a second interconnect protocol, the first interconnect protocol and the second interconnect protocol are different protocols and each comprising one of a serial link protocol, a coherent link protocol, and an accelerator link protocol.

In a second example and in furtherance of the first example, a system, a device, an apparatus, and so forth may include the interface logic to cause processing of the data for communication via a third subset of the plurality of connectors in accordance with a third interconnect protocol, the third interconnect protocol different than the first interconnect protocol and the second interconnect protocol and the third interconnect protocol comprising one of the serial link protocol, the coherent link protocol, and the accelerator link protocol.

In a third example and in furtherance of the previous examples, a system, a device, an apparatus, and so forth may include fuse logic, and one or more fuses coupled with the interface logic, the one or more fuses programmable to associate the first interconnect protocol with the first subset of the plurality of connectors and to associated the second interconnect protocol with the second subset of the plurality of connectors.

In a fourth example and in furtherance of the previous examples, a system, a device, an apparatus, and so forth may include the one or more fuses statically programmable to prevent

changes to which interconnect protocols are associated with particular subsets of the plurality of connectors.

In a fifth example and in furtherance of the previous examples, a system, a device, an apparatus, and so forth may include the one or more fuses reprogrammable to associate a third interconnect protocol with at least one of the first subset of the plurality of connectors and the second subset of the plurality of connectors, the third interconnect protocol different than at least one of the first interconnect protocol and the second interconnect protocol.

In a sixth example and in furtherance of the previous examples, a system, a device, an apparatus, and so forth may include the interface logic to cause processing of the data for communication via the plurality of connectors in accordance with the first interconnect protocol and the second interconnect protocol based on a setting in a Basic Input/Output System (BIOS).

In a seventh example and in furtherance of the previous examples, a system, a device, an apparatus, and so forth may include the BIOS, and one or more fuses coupled with the BIOS, the one or more fuses programmable to associate the first interconnect protocol with the first subset of the plurality of connectors and the second interconnect protocol with the second subset of the plurality of connectors.

In an eighth example and in furtherance of the previous examples, a system, a device, an apparatus, and so forth may include the interface logic to associate the first interconnect protocol with the first subset of the plurality of connectors and the second interconnect protocol with the second subset of the plurality of connectors based on a signal generated based on a first device coupled with eth first subset and a second device coupled with the second subset.

In a ninth example and in furtherance of the previous examples, a system, a device, an apparatus, and so forth may include the interface logic to dynamically change at least one of an association of the first interconnect protocol with the first subset of the plurality of connectors and the second interconnect protocol with the second subset of the plurality of connectors to a different interconnect protocol based on a signal generated by an application during run-time.

In a tenth example and in furtherance of the previous examples, embodiments may include a non-transitory computer-readable storage medium, comprising a plurality of instructions, that when executed, enable processing circuitry to determine a configuration for a plurality of connectors, the configuration to associate a first interconnect protocol with a first subset of the plurality of connectors and a second interconnect protocol with a second subset of the plurality of connectors, the first interconnect protocol and the second interconnect protocol are different interconnect protocols and each comprising one of a serial link protocol, a coherent link protocol, and an accelerator link protocol, cause processing of data for communication via the first subset of the plurality of connectors in accordance with the first interconnect protocol, and

cause processing of data for communication via the second subset of the plurality of connector in accordance with the second interconnect protocol.

In an eleventh example and in furtherance of the previous examples, embodiments may include a non-transitory computer-readable storage medium, comprising a plurality of instructions, that when executed, enable processing circuitry to associate a third interconnect protocol with a third subset of the plurality of connectors based on the configuration, the third interconnect protocol different than the first interconnect protocol and the second interconnect protocol and the third interconnect protocol comprising one of the serial link protocol, the coherent link protocol, and the accelerator link protocol, and cause processing of data for communication via the third subset of the plurality of connectors in accordance with the third interconnect protocol.

In a twelfth example and in furtherance of the previous examples, embodiments may include a non-transitory computer-readable storage medium, comprising a plurality of instructions, that when executed, enable processing circuitry to detect one or more fuses to determine the configuration to associate the first interconnect protocol with the first subset of the plurality of connectors and to associated the second interconnect protocol with the second subset of the plurality of connectors.

In a thirteenth example and in furtherance of the previous examples, embodiments may include a non-transitory computer-readable storage medium, wherein the one or more fuses are statically programmable to prevent changes to which interconnect protocols are associated with particular subsets of the plurality of connectors.

In a fourteenth example and in furtherance of the previous examples, embodiments may include a non-transitory computer-readable storage medium, wherein one or more fuses reprogrammable to associate a different interconnect protocol with at least one of the first subset of the plurality of connectors and the second subset of the plurality of connectors, the different interconnect protocol different than at least one of the first interconnect protocol and the second interconnect protocol.

In a fifteenth example and in furtherance of the previous examples, embodiments may include a non-transitory computer-readable storage medium, comprising a plurality of instructions, that when executed, enable processing circuitry to determine the configuration based on a setting in a Basic Input/Output System (BIOS) or in a read-only memory.

In a sixteenth example and in furtherance of the previous examples, embodiments may include a non-transitory computer-readable storage medium, comprising a plurality of instructions, that when executed, enable processing circuitry to associate the first interconnect protocol with the first subset of the plurality of connectors and the second interconnect protocol

with the second subset of the plurality of connectors based on a signal generated from a first device coupling with the first subset and a second device coupling with the second subset.

In a seventeenth example and in furtherance of the previous examples, embodiments may include a non-transitory computer-readable storage medium, comprising a plurality of instructions, that when executed, enable processing circuitry to dynamically change at least one of an association of the first interconnect protocol with the first subset of the plurality of connectors and the second interconnect protocol with the second subset of the plurality of connectors to a different interconnect protocol based on a signal generated by an application during run-time.

In an eighteenth example and in furtherance of the previous examples, embodiments may include a computer-implemented method including determining a configuration for a plurality of connectors, the configuration to associate a first interconnect protocol with a first subset of the plurality of connectors and a second interconnect protocol with a second subset of the plurality of connectors, the first interconnect protocol and the second interconnect protocol are different interconnect protocols and each comprising one of a serial link protocol, a coherent link protocol, and an accelerator link protocol, causing processing of data for communication via the first subset of the plurality of connectors in accordance with the first interconnect protocol, and causing processing of data for communication via the second subset of the plurality of connector in accordance with the second interconnect protocol.

In a nineteenth example and in furtherance of the previous examples, embodiments may include a computer-implemented method including associating a third interconnect protocol with a third subset of the plurality of connectors based on the configuration, the third interconnect protocol different than the first interconnect protocol and the second interconnect protocol and the third interconnect protocol comprising one of the serial link protocol, the coherent link protocol, and the accelerator link protocol, and causing processing of data for communication via the third subset of the plurality of connectors in accordance with the third interconnect protocol.

In a twentieth example and in furtherance of the previous examples, embodiments may include a computer-implemented method including detecting one or more fuses to determine the configuration to associate the first interconnect protocol with the first subset of the plurality of connectors and to associated the second interconnect protocol with the second subset of the plurality of connectors.

In a twenty-first example and in furtherance of the previous examples, embodiments may include a computer-implemented method wherein the one or more fuses are statically programmable to prevent changes to which interconnect protocols are associated with particular subsets of the plurality of connectors.

In a twenty-second example and in furtherance of the previous examples, embodiments may include a computer-implemented method wherein one or more fuses reprogrammable to associate a different interconnect protocol with at least one of the first subset of the plurality of connectors and the second subset of the plurality of connectors, the different interconnect protocol different than at least one of the first interconnect protocol and the second interconnect protocol.

In a twenty-third example and in furtherance of the previous examples, embodiments may include a computer-implemented method including determining the configuration based on a setting in a Basic Input/Output System (BIOS) or in a read-only memory.

In a twenty-fourth example and in furtherance of the previous examples, embodiments may include a computer-implemented method including associating the first interconnect protocol with the first subset of the plurality of connectors and the second interconnect protocol with the second subset of the plurality of connectors based on a signal generated from a first device coupling with the first subset and a second device coupling with the second subset.

In a twenty-fifth example and in furtherance of the previous examples, embodiments may include a computer-implemented method including dynamically changing at least one of an association of the first interconnect protocol with the first subset of the plurality of connectors and the second interconnect protocol with the second subset of the plurality of connectors to a different interconnect protocol based on a signal generated by an application during run-time.

Some embodiments may be described using the expression “one embodiment” or “an embodiment” along with their derivatives. These terms mean that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment. The appearances of the phrase “in one embodiment” in various places in the specification are not necessarily all referring to the same embodiment. Further, some embodiments 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, some embodiments may be described using the terms “connected” and “coupled” to 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.

It is emphasized that the Abstract of the Disclosure is provided to allow a reader to quickly ascertain the nature of the technical disclosure. It is submitted with the understanding that it will not be used to interpret or limit the scope or meaning of the claims. Also, in the preceding Detailed Description, it can be seen that various features are grouped together in a single embodiment for the purpose of streamlining the disclosure. This method of disclosure is not to

be interpreted as reflecting an intention that the claimed embodiments require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive subject matter lies in less than all features of a single disclosed embodiment. Thus the following claims are at this moment incorporated into the Detailed Description, with each claim standing on its own as a separate embodiment. In the appended claims, the terms "including" and "in which" are used as the Plain-English equivalents of the respective terms "comprising" and "wherein," respectively. Moreover, the terms "first," "second," "third," and so forth, are used merely as labels and are not intended to impose numerical requirements on their objects.

What has been described above includes examples of the disclosed architecture? It is, of course, not possible to describe every conceivable combination of components and methodologies, but one of ordinary skill in the art may recognize that many further combinations and permutations are possible. Accordingly, the novel architecture is intended to embrace all such alterations, modifications, and variations that fall within the spirit and scope of the appended claims.

## Claims

What is claimed is:

1. An apparatus, comprising:

a physical (PHY) layer interface comprising a plurality of connectors configured to couple with one or more other devices, and to send and receive data via the plurality of connectors; and

interface logic coupled with the PHY layer interface, the interface logic to cause processing of the data for communication via a first subset of the plurality of connectors in accordance with a first interconnect protocol and to cause processing of the data for communication via a second subset of the plurality of connector in accordance with a second interconnect protocol, the first interconnect protocol and the second interconnect protocol are different protocols and each comprising one of a serial link protocol, a coherent link protocol, and an accelerator link protocol.

2. The apparatus of claim 1, the interface logic to cause processing of the data for communication via a third subset of the plurality of connectors in accordance with a third interconnect protocol, the third interconnect protocol different than the first interconnect protocol and the second interconnect protocol and the third interconnect protocol comprising one of the serial link protocol, the coherent link protocol, and the accelerator link protocol.

3. The apparatus of claim 1, comprising:

fuse logic; and

one or more fuses coupled with the interface logic, the one or more fuses programmable to associate the first interconnect protocol with the first subset of the plurality of connectors and to associated the second interconnect protocol with the second subset of the plurality of connectors.

4. The apparatus of claim 3, the one or more fuses statically programmable to prevent changes to which interconnect protocols are associated with particular subsets of the plurality of connectors.

5. The apparatus of claim 3, the one or more fuses reprogrammable to associate a third interconnect protocol with at least one of the first subset of the plurality of connectors and the second subset of the plurality of connectors, the third interconnect protocol different than at least one of the first interconnect protocol and the second interconnect protocol.

6. The apparatus of claim 1, the interface logic to cause processing of the data for communication via the plurality of connectors in accordance with the first interconnect protocol and the second interconnect protocol based on a setting in a Basic Input/Output System (BIOS).

7. The apparatus of claim 6, comprising:

the BIOS; and

one or more fuses coupled with the BIOS, the one or more fuses programmable to associate the first interconnect protocol with the first subset of the plurality of connectors and the second interconnect protocol with the second subset of the plurality of connectors.

8. The apparatus of claim 1, the interface logic to associate the first interconnect protocol with the first subset of the plurality of connectors and the second interconnect protocol with the second subset of the plurality of connectors based on a signal generated based on a first device coupled with eth first subset and a second device coupled with the second subset.

9. The apparatus of claim 1, the interface logic to dynamically change at least one of an association of the first interconnect protocol with the first subset of the plurality of connectors and the second interconnect protocol with the second subset of the plurality of connectors to a different interconnect protocol based on a signal generated by an application during run-time.

10. A non-transitory computer-readable storage medium, comprising a plurality of instructions, that when executed, enable processing circuitry to:

determine a configuration for a plurality of connectors, the configuration to associate a first interconnect protocol with a first subset of the plurality of connectors and

a second interconnect protocol with a second subset of the plurality of connectors, the first interconnect protocol and the second interconnect protocol are different interconnect protocols and each comprising one of a serial link protocol, a coherent link protocol, and an accelerator link protocol;

cause processing of data for communication via the first subset of the plurality of connectors in accordance with the first interconnect protocol; and

cause processing of data for communication via the second subset of the plurality of connector in accordance with the second interconnect protocol.

11. The non-transitory computer-readable storage medium of claim 10, comprising a plurality of instructions, that when executed, enable processing circuitry to:

associate a third interconnect protocol with a third subset of the plurality of connectors based on the configuration, the third interconnect protocol different than the first interconnect protocol and the second interconnect protocol and the third interconnect protocol comprising one of the serial link protocol, the coherent link protocol, and the accelerator link protocol; and

cause processing of data for communication via the third subset of the plurality of connectors in accordance with the third interconnect protocol.

12. The non-transitory computer-readable storage medium of claim 10, comprising a plurality of instructions, that when executed, enable processing circuitry to detect one or more fuses to determine the configuration to associate the first interconnect protocol with the first subset of the plurality of connectors and to associated the second interconnect protocol with the second subset of the plurality of connectors.

13. The non-transitory computer-readable storage medium of claim 12, wherein the one or more fuses are statically programmable to prevent changes to which interconnect protocols are associated with particular subsets of the plurality of connectors.

14. The non-transitory computer-readable storage medium of claim 12, wherein one or more fuses reprogrammable to associate a different interconnect protocol with at least one of the first subset of the plurality of connectors and the second subset of the plurality

of connectors, the different interconnect protocol different than at least one of the first interconnect protocol and the second interconnect protocol.

15. The non-transitory computer-readable storage medium of claim 10, comprising a plurality of instructions, that when executed, enable processing circuitry to determine the configuration based on a setting in a Basic Input/Output System (BIOS) or in a read-only memory.

16. The non-transitory computer-readable storage medium of claim 10, comprising a plurality of instructions, that when executed, enable processing circuitry to associate the first interconnect protocol with the first subset of the plurality of connectors and the second interconnect protocol with the second subset of the plurality of connectors based on a signal generated from a first device coupling with the first subset and a second device coupling with the second subset.

17. The non-transitory computer-readable storage medium of claim 10, comprising a plurality of instructions, that when executed, enable processing circuitry to dynamically change at least one of an association of the first interconnect protocol with the first subset of the plurality of connectors and the second interconnect protocol with the second subset of the plurality of connectors to a different interconnect protocol based on a signal generated by an application during run-time.

18. A computer-implemented method, comprising:

determining a configuration for a plurality of connectors, the configuration to associate a first interconnect protocol with a first subset of the plurality of connectors and a second interconnect protocol with a second subset of the plurality of connectors, the first interconnect protocol and the second interconnect protocol are different interconnect protocols and each comprising one of a serial link protocol, a coherent link protocol, and an accelerator link protocol;

causing processing of data for communication via the first subset of the plurality of connectors in accordance with the first interconnect protocol; and

causing processing of data for communication via the second subset of the plurality of connector in accordance with the second interconnect protocol.

19. The computer-implemented method of claim 18, comprising:

associating a third interconnect protocol with a third subset of the plurality of connectors based on the configuration, the third interconnect protocol different than the first interconnect protocol and the second interconnect protocol and the third interconnect protocol comprising one of the serial link protocol, the coherent link protocol, and the accelerator link protocol; and

causing processing of data for communication via the third subset of the plurality of connectors in accordance with the third interconnect protocol.

20. The computer-implemented method of claim 18, comprising:

detecting one or more fuses to determine the configuration to associate the first interconnect protocol with the first subset of the plurality of connectors and to associated the second interconnect protocol with the second subset of the plurality of connectors.

21. The computer-implemented method of claim 20, wherein the one or more fuses are statically programmable to prevent changes to which interconnect protocols are associated with particular subsets of the plurality of connectors.

22. The computer-implemented method of claim 20, wherein one or more fuses reprogrammable to associate a different interconnect protocol with at least one of the first subset of the plurality of connectors and the second subset of the plurality of connectors, the different interconnect protocol different than at least one of the first interconnect protocol and the second interconnect protocol.

23. The computer-implemented method of claim 18, comprising:

determining the configuration based on a setting in a Basic Input/Output System (BIOS) or in a read-only memory.

24. The computer-implemented method of claim 18, comprising:

associating the first interconnect protocol with the first subset of the plurality of connectors and the second interconnect protocol with the second subset of the plurality of connectors based on a signal generated from a first device coupling with the first subset and a second device coupling with the second subset.

25. The computer-implemented method of claim 18, comprising:

dynamically changing at least one of an association of the first interconnect protocol with the first subset of the plurality of connectors and the second interconnect protocol with the second subset of the plurality of connectors to a different interconnect protocol based on a signal generated by an application during run-time.

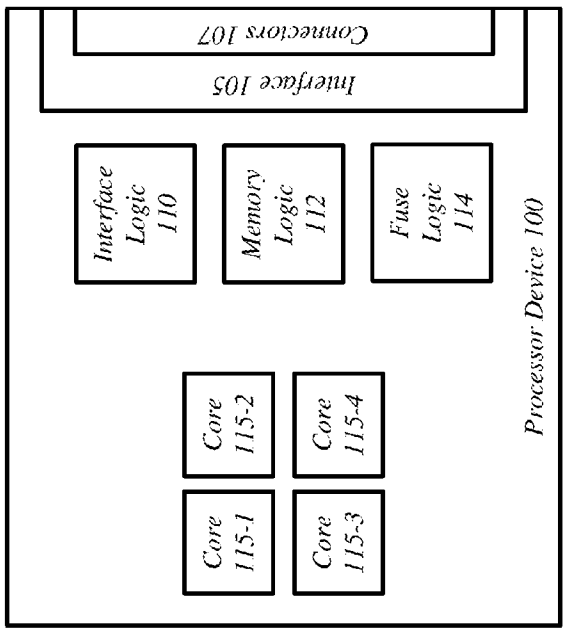

Connectors 107
Interface 105
Interface Logic 110
Memory Logic 112
Fuse Logic 114
Core 115-1
Core 115-2
Core 115-3
Core 115-4
Processor Device 100

FIG. 1

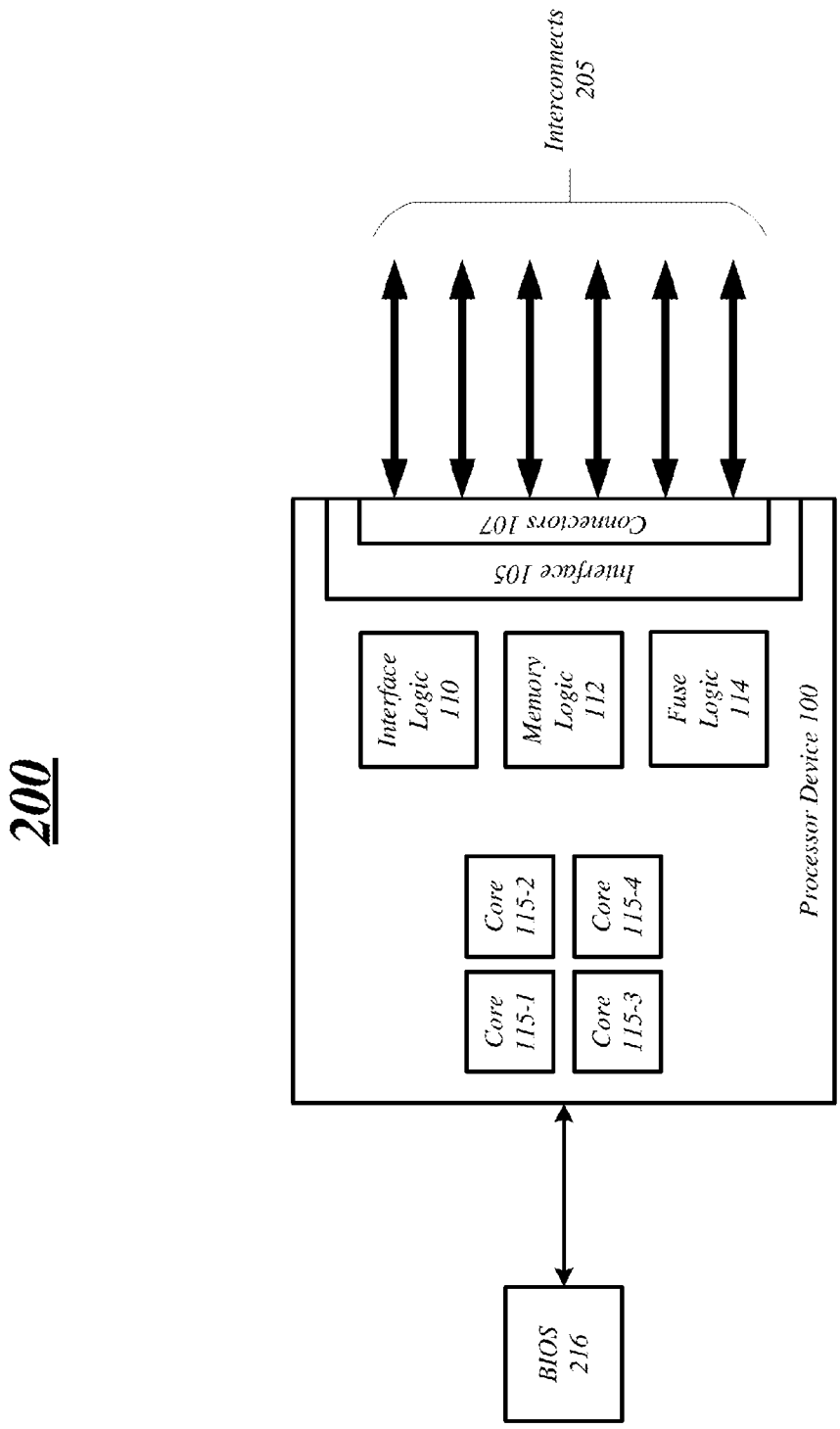

200
Interconnects 205
Connectors 107
Interface 105
Interface Logic 110
Memory Logic 112
Fuse Logic 114
Core 115-2
Core 115-4
Core 115-1
Core 115-3
Processor Device 100
BIOS 216

FIG. 2A

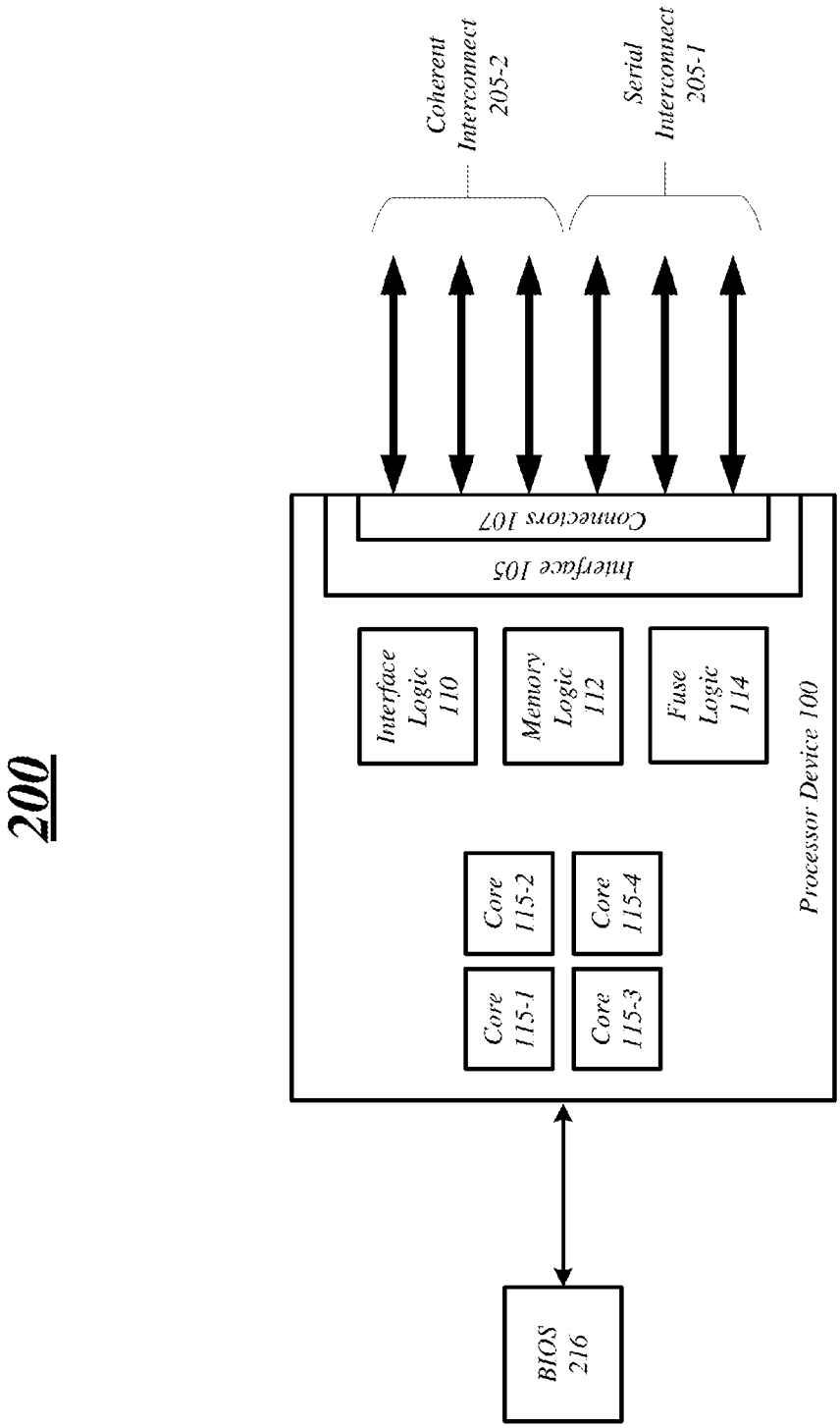

200
Coherent Interconnect 205-2
Serial Interconnect 205-1
Connectors 107
Interface 105
Interface Logic 110
Memory Logic 112
Fuse Logic 114
Core 115-2
Core 115-4
Core 115-1
Core 115-3
Processor Device 100
BIOS 216

FIG. 2B

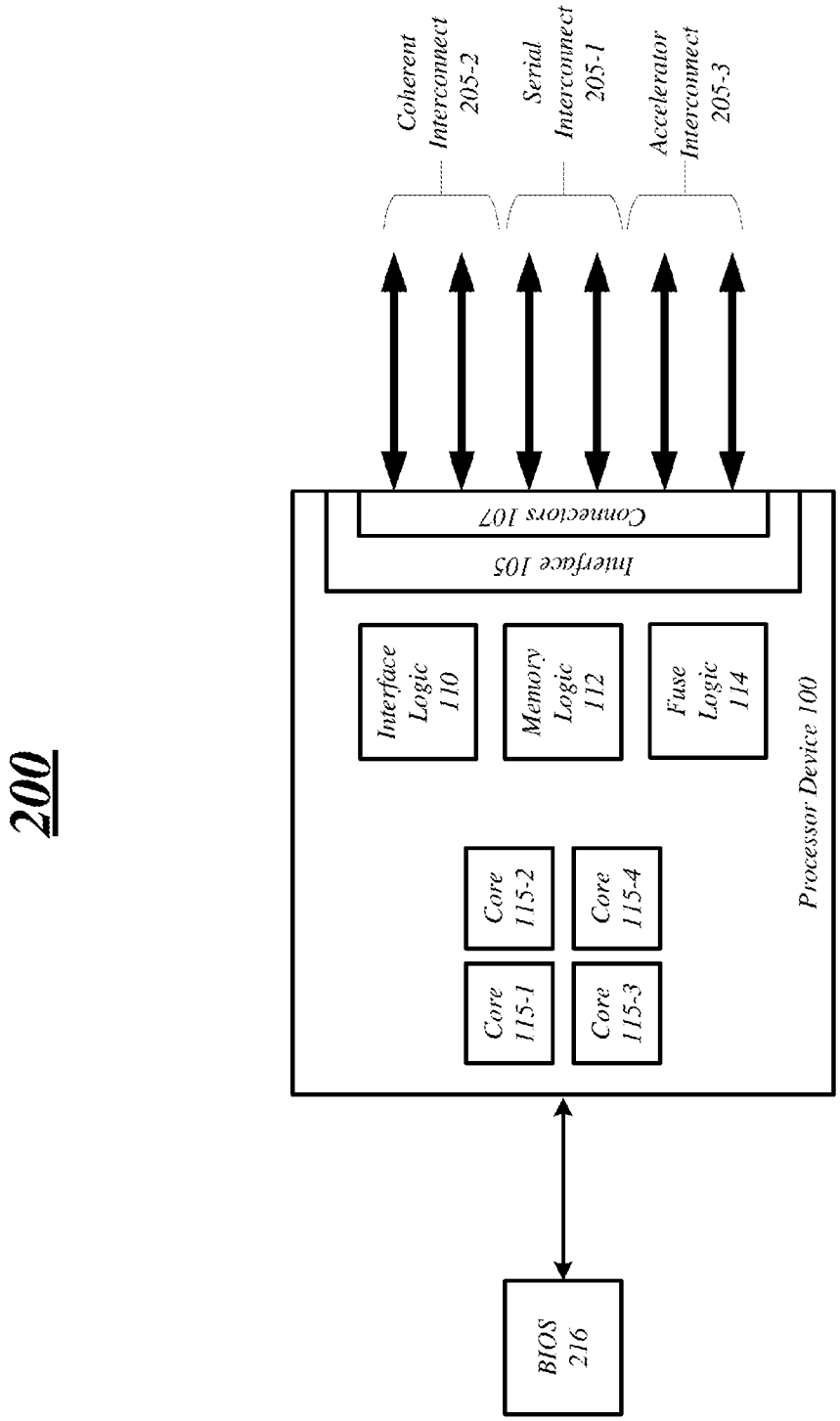

200
Coherent Interconnect 205-2
Serial Interconnect 205-1
Accelerator Interconnect 205-3
Connectors 107
Interface 105
Interface Logic 110
Memory Logic 112
Fuse Logic 114
Core 115-2
Core 115-4
Core 115-1
Core 115-3
Processor Device 100
BIOS 216

FIG. 2C

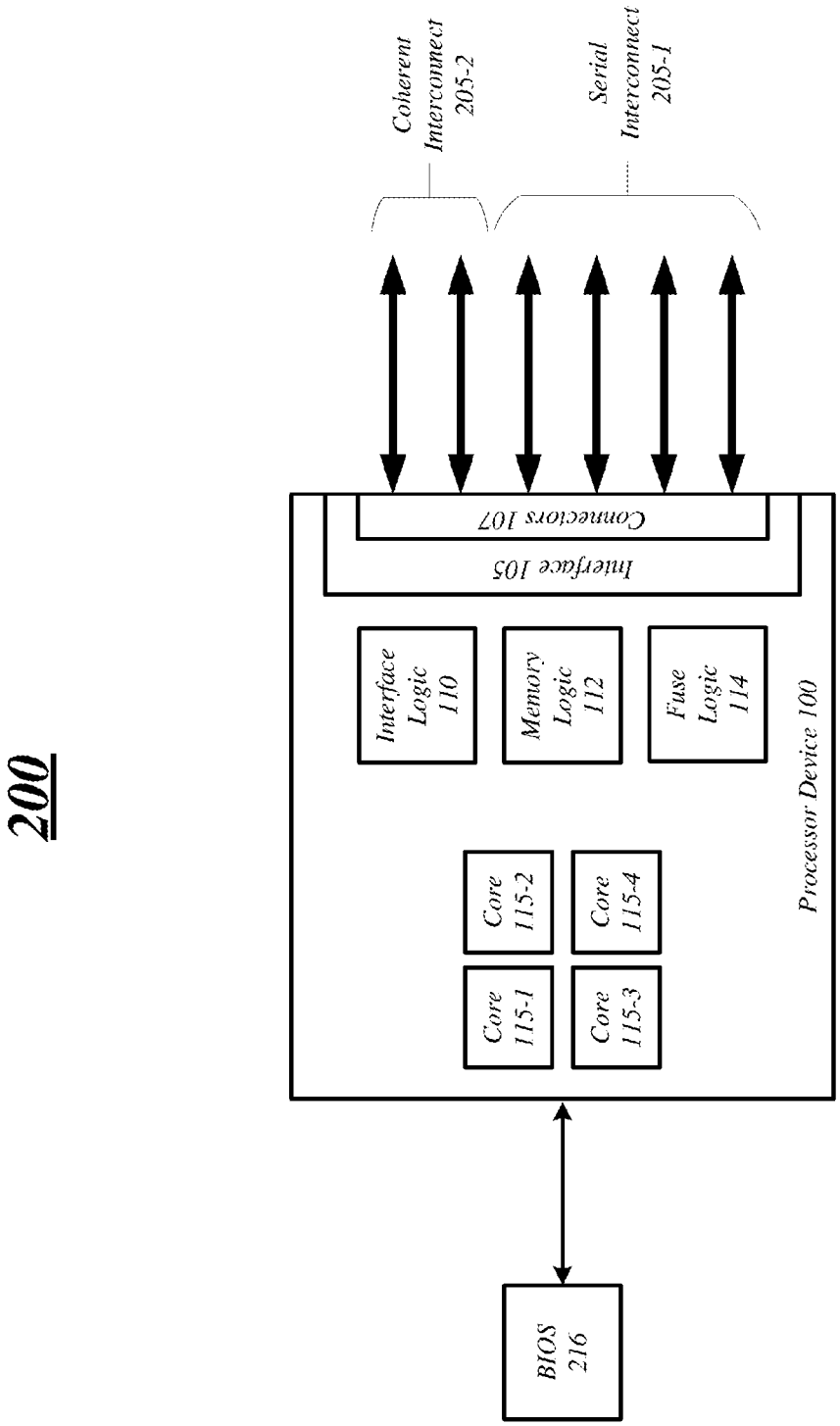

200
Coherent Interconnect 205-2
Serial Interconnect 205-1
Connectors 107
Interface 105
Interface Logic 110
Memory Logic 112
Fuse Logic 114
Core 115-2
Core 115-4
Core 115-1
Core 115-3
Processor Device 100
BIOS 216

FIG. 2D

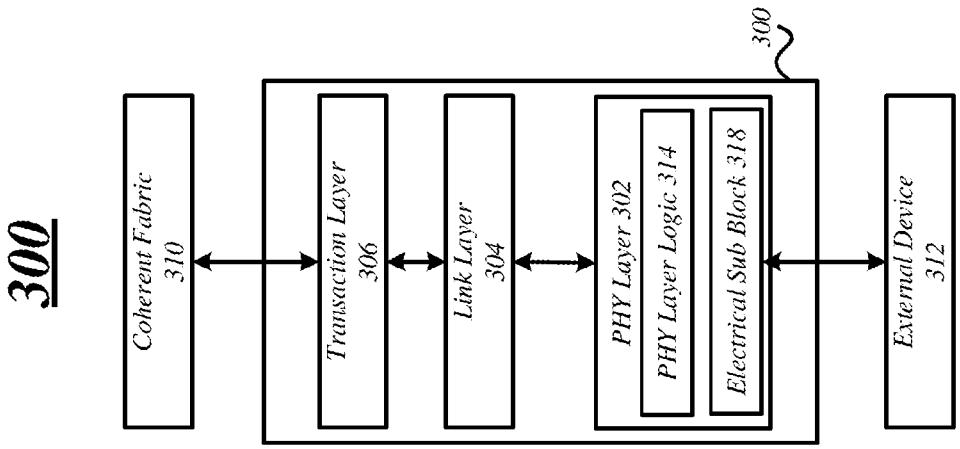

300
Coherent Fabric 310
Transaction Layer 306
Link Layer 304
PHY Layer 302
PHY Layer Logic 314
Electrical Sub Block 318
External Device 312
300

FIG. 3A

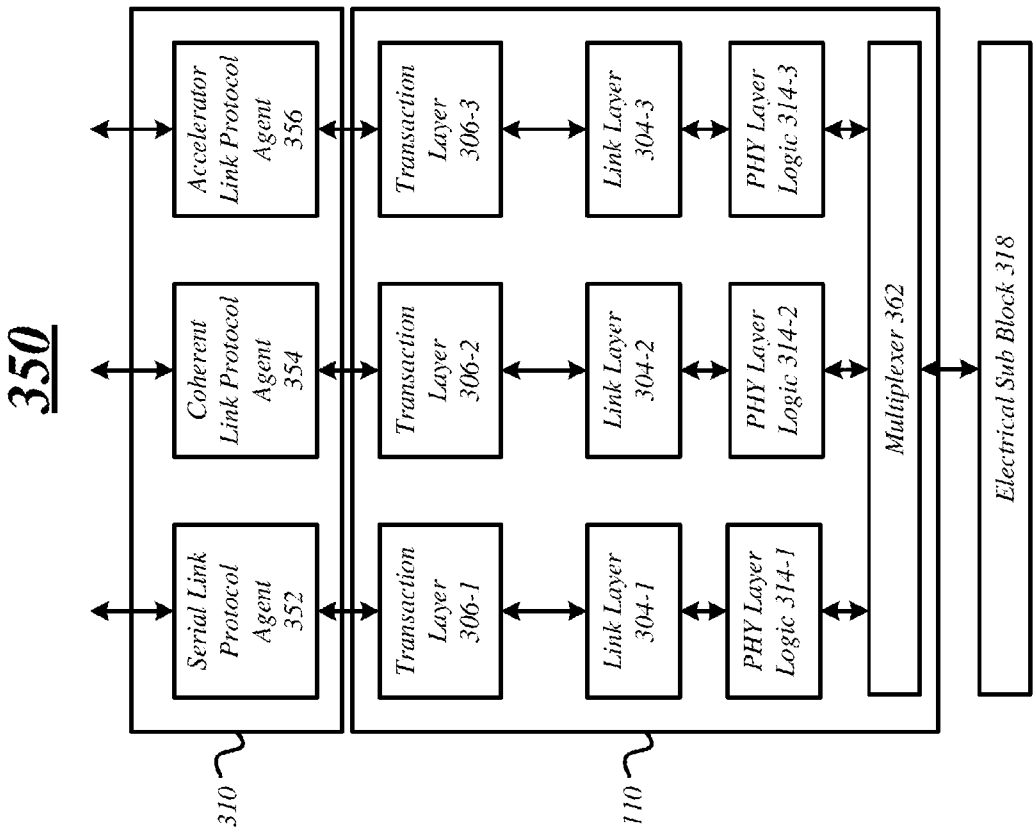

350
Serial Link Protocol Agent 352
Coherent Link Protocol Agent 354
Accelerator Link Protocol Agent 356
Transaction Layer 306-1
Transaction Layer 306-2
Transaction Layer 306-3
Link Layer 304-1
Link Layer 304-2
Link Layer 304-3
PHY Layer Logic 314-1
PHY Layer Logic 314-2
PHY Layer Logic 314-3
Multiplexer 362
Electrical Sub Block 318
310
110

FIG. 3B

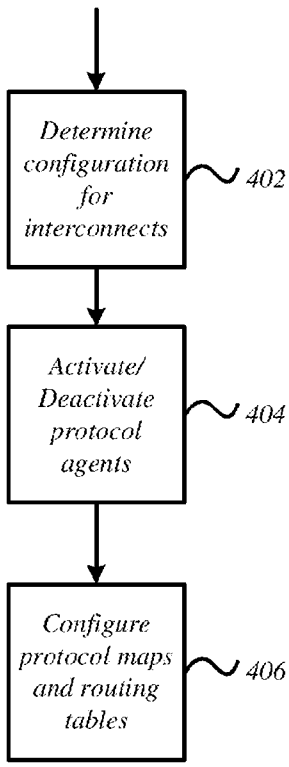

400
Determine configuration for interconnects
402
Activate/ Deactivate protocol agents
404
Configure protocol maps and routing tables
406

FIG. 4A

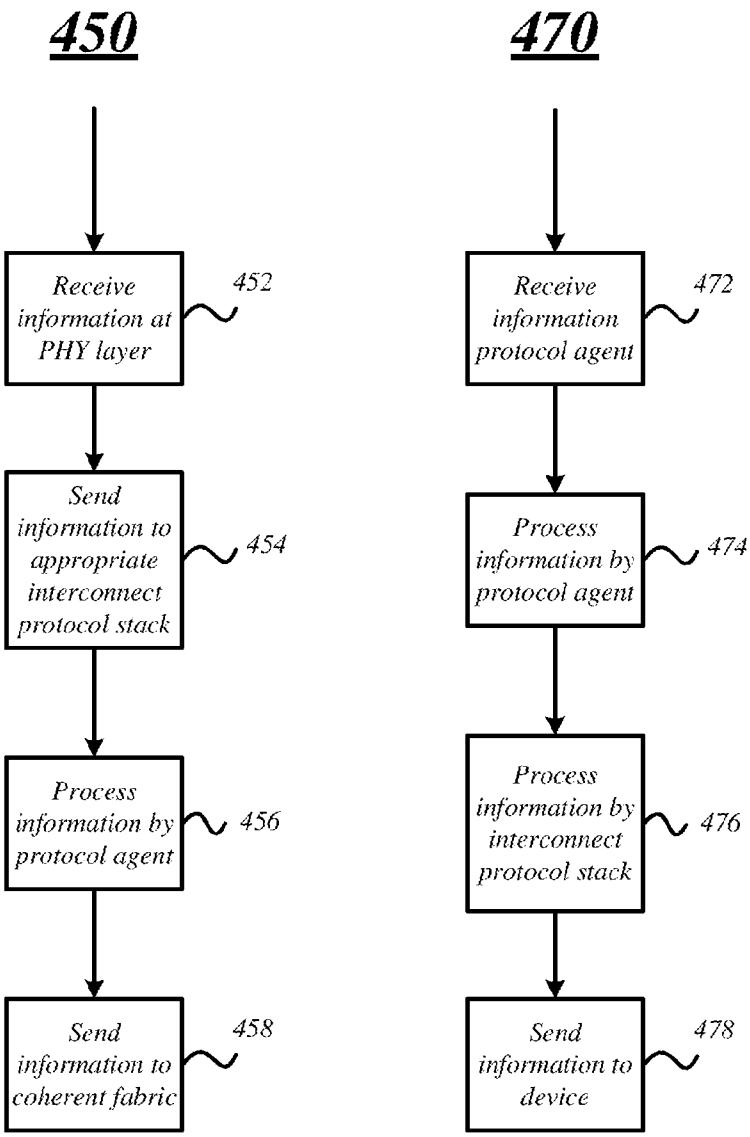

450
Receive information at PHY layer
452
Send information to appropriate interconnect protocol stack
454
Process information by protocol agent
456
Send information to coherent fabric
458
470
Receive information protocol agent
472
Process information by protocol agent
474
Process information by interconnect protocol stack
476
Send information to device
478

*FIG. 4B*

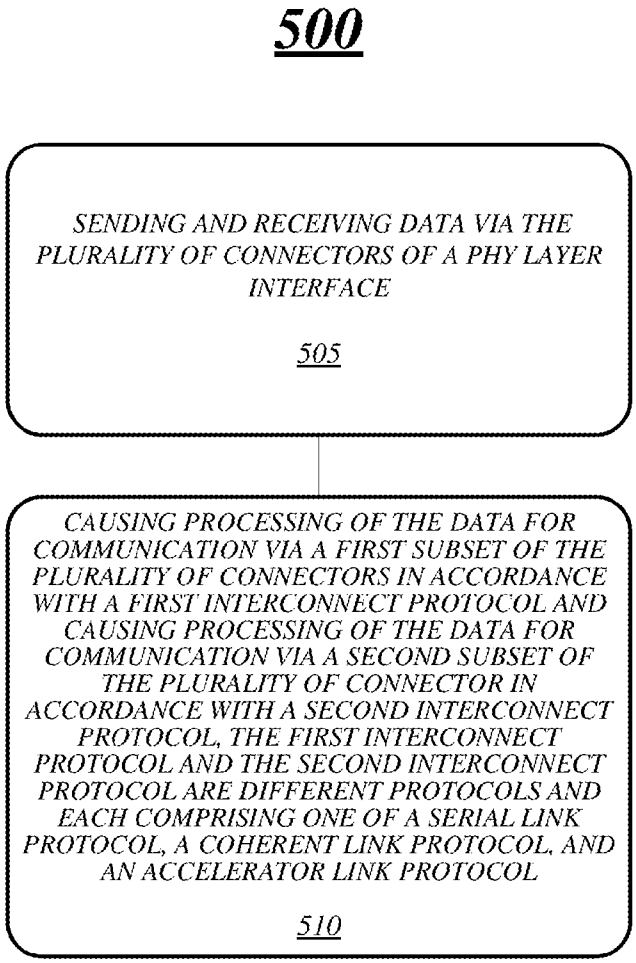

500
SENDING AND RECEIVING DATA VIA THE PLURALITY OF CONNECTORS OF A PHY LAYER INTERFACE
505
CAUSING PROCESSING OF THE DATA FOR COMMUNICATION VIA A FIRST SUBSET OF THE PLURALITY OF CONNECTORS IN ACCORDANCE WITH A FIRST INTERCONNECT PROTOCOL AND CAUSING PROCESSING OF THE DATA FOR COMMUNICATION VIA A SECOND SUBSET OF THE PLURALITY OF CONNECTOR IN ACCORDANCE WITH A SECOND INTERCONNECT PROTOCOL, THE FIRST INTERCONNECT PROTOCOL AND THE SECOND INTERCONNECT PROTOCOL ARE DIFFERENT PROTOCOLS AND EACH COMPRISING ONE OF A SERIAL LINK PROTOCOL, A COHERENT LINK PROTOCOL, AND AN ACCELERATOR LINK PROTOCOL
510

*FIG. 5*

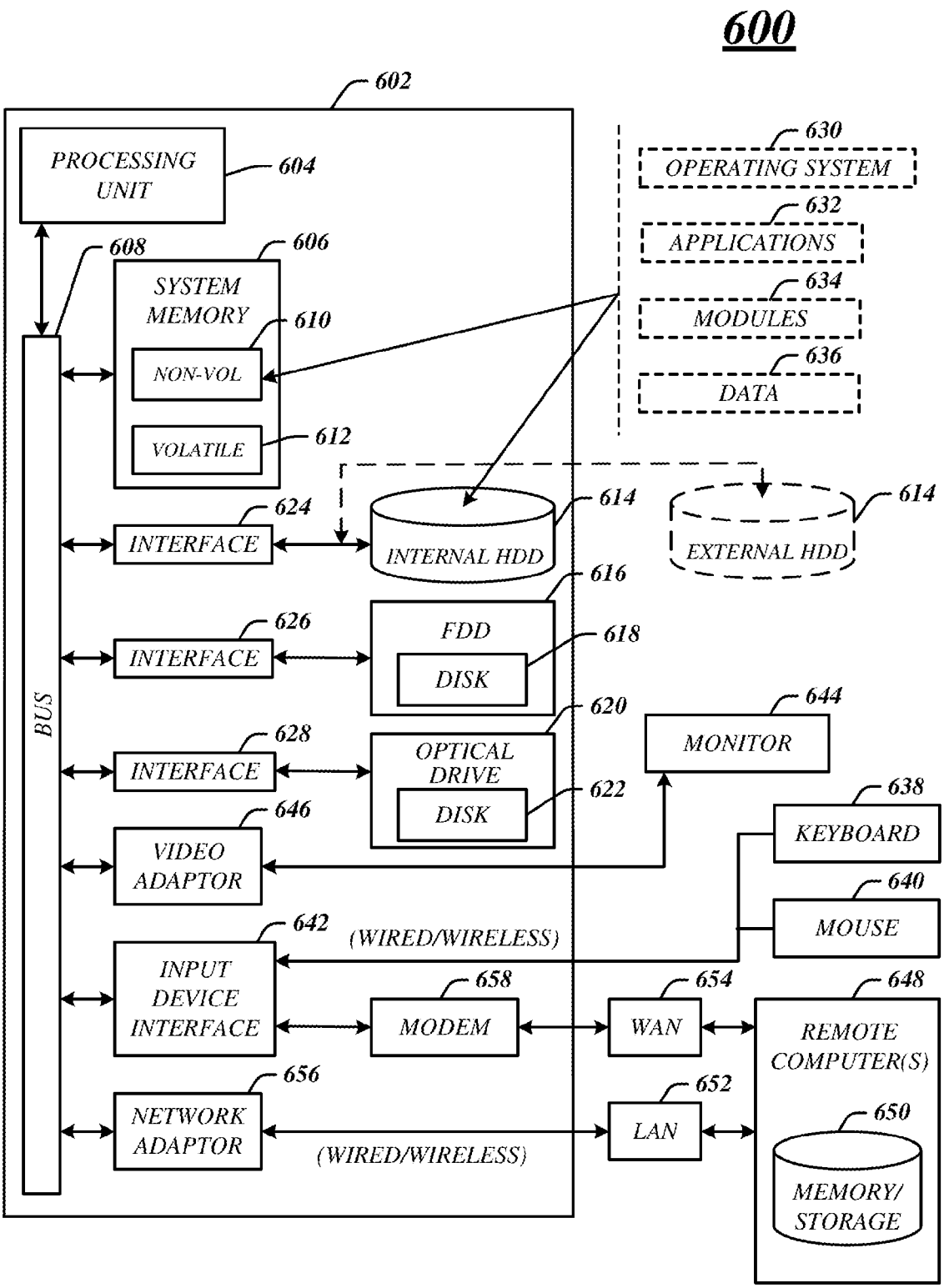

600
602
PROCESSING UNIT
604
608
606
SYSTEM MEMORY
610
NON-VOL
VOLATILE
612
BUS
630
OPERATING SYSTEM
632
APPLICATIONS
634
MODULES
636
DATA
624
INTERFACE
614
INTERNAL HDD
614
EXTERNAL HDD
616
626
INTERFACE
FDD
618
DISK
620
644
MONITOR
628
INTERFACE
OPTICAL DRIVE
622
DISK
638
KEYBOARD
646
VIDEO ADAPTOR
640
MOUSE
642
(WIRED/WIRELESS)
INPUT DEVICE INTERFACE
658
MODEM
654
WAN
648
REMOTE COMPUTER(S)
656
NETWORK ADAPTOR
652
LAN
650
(WIRED/WIRELESS)
MEMORY/ STORAGE

*FIG. 6*

**INTERNATIONAL SEARCH REPORT**

International application No.
**PCT/US2017/037408**

**A. CLASSIFICATION OF SUBJECT MATTER**

**G06F 13/42(2006.01)i**

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

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)
G06F 13/42; G06F 13/40; H01R 13/66; H04J 3/22; G06F 1/32

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: interface, interconnect, protocol, PCIe, QPI, connector, subset, and similar terms.

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| X | US 2013-0297845 A1 (DIMITRIOS ZIAKAS et al.) 07 November 2013<br>See paragraphs [0018]-[0024], [0028], and [0034];<br>claims 8-9, 11, 15, and 18; and figures 1-3. | 1-25 |
| A | US 2011-0183546 A1 (WAEL WILLIAM DIAB et al.) 28 July 2011<br>See paragraphs [0047]-[0048] and figure 3. | 1-25 |
| A | US 2014-0019653 A1 (MIKHAIL AMCHISLAVSKY et al.) 16 January 2014<br>See paragraphs [0041]-[0045]; claim 1; and figure 1. | 1-25 |
| A | US 2014-0181352 A1 (SHAUN M. CONRAD et al.) 26 June 2014<br>See paragraphs [0028]-[0030] and figure 6. | 1-25 |
| A | US 2011-0206063 A1 (WAEL WILLIAM DIAB) 25 August 2011<br>See paragraphs [0045]-[0047] and figures 3A-3B. | 1-25 |

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

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

| Date of the actual completion of the international search | Date of mailing of the international search report |
|---|---|
| 25 September 2017 (25.09.2017) | **25 September 2017 (25.09.2017)** |

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea
Facsimile No. +82-42-481-8578

Authorized officer
NHO, Ji Myong
Telephone No. +82-42-481-8528

**INTERNATIONAL SEARCH REPORT**

Information on patent family members

International application No.
**PCT/US2017/037408**

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| US 2013-0297845 A1 | 07/11/2013 | CN 103946822 A | 23/07/2014 |
| | | CN 103946822 B | 09/11/2016 |
| | | EP 2761479 A1 | 06/08/2014 |
| | | US 9361257 B2 | 07/06/2016 |
| | | WO 2013-048494 A1 | 04/04/2013 |
| US 2011-0183546 A1 | 28/07/2011 | CN 102136851 A | 27/07/2011 |
| | | EP 2348677 A1 | 27/07/2011 |
| | | TW 201203942 A | 16/01/2012 |
| US 2014-0019653 A1 | 16/01/2014 | CN 104620564 A | 13/05/2015 |
| | | EP 2873220 A1 | 20/05/2015 |
| | | JP 2015-528243 A | 24/09/2015 |
| | | KR 10-2015-0036539 A | 07/04/2015 |
| | | TW 201403341 A | 16/01/2014 |
| | | TW I579705 B | 21/04/2017 |
| | | US 9087163 B2 | 21/07/2015 |
| | | WO 2014-011317 A1 | 16/01/2014 |
| US 2014-0181352 A1 | 26/06/2014 | CN 104769567 A | 08/07/2015 |
| | | DE 112013005370 T5 | 23/07/2015 |
| | | US 2017-0010648 A1 | 12/01/2017 |
| | | US 9477627 B2 | 25/10/2016 |
| | | WO 2014-105198 A1 | 03/07/2014 |
| US 2011-0206063 A1 | 25/08/2011 | CN 102164038 A | 24/08/2011 |
| | | EP 2360868 A1 | 24/08/2011 |
| | | TW 201203932 A | 16/01/2012 |