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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 December 2009 (10.12.2009)

PCT

(10) International Publication Number
WO 2009/148432 A1

(51) International Patent Classification:
G06F 13/14 (2006.01)

(21) International Application Number:
PCT/US2008/065460

(22) International Filing Date:
1 June 2008 (01.06.2008)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.** [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **NORDEN, Hahn, Vo** [US/US]; 4 M74d186 MS M0704-705, 11445 Compaq Center Dr. W., Houston, TX 77070 (US).

(74) Agents: **JONES, Kevin** et al.; Hewlett-Packard Company, Intellectual Property Administration, Mail Stop 35, P.o.box 272400, Fort Collins, CO 80527-2400 (US).

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

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

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i))*

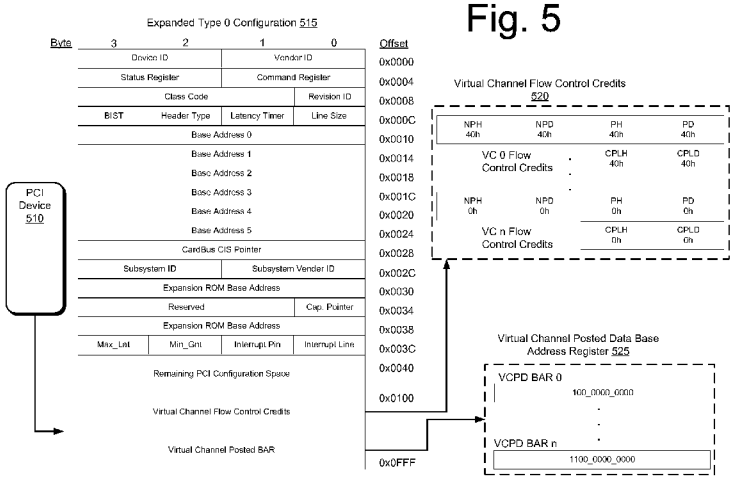
— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

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

(54) Title: QUEUE SHARING AND RECONFIGURATION IN PCI EXPRESS LINKS

Fig. 5



(57) Abstract: In one embodiment an electronic device comprises at least one processor, at least one PCI express link, a virtual channel/sub-link flow control module, and a memory module communicatively connected to the one or more processors and comprising logic instructions which, when executed on the one or more processors configure the one or more processors to determine, in an electrical device, whether a virtual channel/sub-link is inactive, and in response to a determination that at least one virtual channel/sub-link is inactive, reallocate queue space from the at least one inactive channel to at least one active channel.

WO 2009/148432 A1

QUEUE SHARING AND RECONFIGURATION IN PCI EXPRESS LINKS

BACKGROUND

[0001] In many computer environments, a fast and flexible interconnect system can be desirable to provide connectivity to devices capable of high levels of data throughput. In the fields of data transfer between devices in a computing environment, PCI Express (PCI-E) can be used to provide connectivity between a host and one or more client devices or endpoints.

[0002] Many high-speed links, such as PCI Express, support multiple virtual channels for multiple traffic classes and multiple division of “fat” link into multiple sublinks to support multiple end devices. If a virtual channel, or sublink, is not active, memory associated with an inactive channel is also unused.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] Fig. 1 is a flowchart illustrating operations implementing a virtual channel/sub-link flow control module, according to embodiments.

[0004] Fig. 2 is a schematic illustration of a computing system adapted to incorporate a virtual channel/sub-link flow control module, according to embodiments.

[0005] Fig. 3 is a schematic illustration of an electronic device adapted to incorporate a virtual channel/sub-link flow control module, according to embodiments.

[0006] Fig. 4 is a schematic representation of a PCI-Express connection, according to embodiments.

[0007] Fig. 5 is a schematic illustration of one embodiment in which queue sharing and reconfiguring in a point to point link may be implemented..

DETAILED DESCRIPTION

[0008] Generally, many high-speed links, such as PCI Express, support multiple virtual channels for multiple traffic classes. Each traffic type may require its own resources for each type of transaction, such as but not limited to: Posted/none-posted/completion, Non-Posted/none-posted/completion and Completion. Traditionally, a PCI Express device may have separate and fixed queues for each traffic type and each virtual channel. If a virtual channel/sub-link is not active, then unused queues may be wasted. Described herein are methods and apparatus to reconfigure and reallocate memory queues from inactive channels to active channels to optimize performance across a PCI express link.

[0009] Fig. 1 is a flowchart illustrating operations implementing a virtual channel/sub-link flow control module, according to embodiments. Referring to Fig. 1, at operation 105, virtual channel/sub-link flow control credits are allocated. In some embodiments, these virtual channel/sub-link flow control credits are referenced in a virtual channel/sub-link flow control register. In some embodiments, the allocation may take place as part of power-on self-test (POST). In some embodiments, non-zero values with numbers indicating a queue size may be used for active channels, while zeros may be used for inactive channels.

[0010] During usage of a computer system, electronic device or the like, a virtual channel/sub-link flow control module may reallocate queue space from inactive channels to active channels. At operation 110, virtual channel/sub-link

flow control registers are initialized. At operation 115, the virtual channel/sub-link flow control credits register is read to determine the status of virtual channels. If at operation 120, all the virtual channels are in use then at operation 125 the queue configuration for virtual channels may remain in a default state, and data may be transmitted across a PCI express link through virtual channels at operation 140.

[0011] By contrast, if at operation 120, it is determined that at least one virtual channel/sub-link is inactive, then at operation 130 the memory queue allocated to the inactive channels may be reallocated to active channels. In operation, inactive memory queue may be reallocated in any number of ways. The reallocation does not need to be equal among active channels.

[0012] At operation 135, offset information may be written to a virtual channel/sub-link's posted/none-posted/completion data base address register. At operation 140 data may be transmitted across a PCI express link through virtual channels. Finally, at operation 145 data is transmitted from reconfigured offset information in the virtual channel/sub-link's posted/none-posted/completion data base address register.

[0013] Fig. 2 is a schematic illustration of a computing system 200 adapted to include a virtual channel/sub-link flow control module, according to embodiments. The computing system 200 includes a computing engine 208 and possibly one or more accompanying input/output devices 206 including, but not limited to, a display 202 having a screen 204, a keyboard 210, and other I/O device(s) 212. The other device(s) 212 may, by way of example, and not by

limitation, include a touch screen, a voice-activated input device, a track ball, a mouse and any other device that allows the system 200 to receive input from a developer and/or a user.

[0014] The computing engine 208 includes system hardware 220 commonly implemented on a motherboard and at least one auxiliary circuit board. System hardware 220 includes a processor 222 and a basic input/output system (BIOS) 226. BIOS 226 may be implemented in flash memory and may comprise logic operations to boot the computer device and a power-on self-test (POST) module for performing system initialization and tests. In operation, when activation of computing system 200 begins processor 222 accesses BIOS 226 and shadows the instructions of BIOS 226, such as power-on self-test module, into operating memory. Processor 222 then executes power-on self-test operations to implement POST processing.

[0015] Computing system 200 further includes a file store 280 communicatively connected to computing engine 208. File store 280 may be internal such as, e.g., one or more hard drives, or external such as, e.g., one or more external hard drives, network attached storage, or a separate storage network. In some embodiments, the file store 280 may include one or more partitions 282, 284, 286.

[0016] Memory 230 includes an operating system 240 for managing operations of computing engine 208. In one embodiment, operating system 240 includes a hardware abstraction layer 254 that provides an interface to system hardware 220. In addition, operating system 240 includes a kernel 244, one or

more file systems 246 that manage files used in the operation of computing engine 208 and a process control subsystem 248 that manages processes executing on computing engine 208. Operating system 240 further includes one or more device drivers 250 and a system call interface module 242 that provides an interface between the operating system 240 and one or more application modules 262 and/or libraries 264. The various device drivers 250 interface with and generally control the hardware installed in the computing system 200.

[0017] In operation, one or more application modules 262 and/or libraries 264 executing on computing engine 208 make calls to the system call interface module 242 to execute one or more commands on the computer's processor. The system call interface module 242 invokes the services of the file systems 246 to manage the files required by the command(s) and the process control subsystem 248 to manage the process required by the command(s). The file system(s) 246 and the process control subsystem(s) 248, in turn, invoke the services of the hardware abstraction layer 254 to interface with the system hardware 220. The operating system kernel 244 can be generally considered as one or more software modules that are responsible for performing many operating system functions.

[0018] The particular embodiment of operating system 240 is not critical to the subject matter described herein. Operating system 240 may, for example, be embodied as a UNIX operating system or any derivative thereof (e.g., Linux, Solaris, etc.) or as a Windows® brand operating system or another operating system.

[0019] In some embodiments, computing system 200 includes at least one virtual channel/sub-link flow control module 228, 266, which may comprise operational logic and may include or invoke hardware that can communicate with at least one remote device or component. In the embodiment depicted in Fig. 2, BIOS 226 includes a virtual channel/sub-link flow control module 228 and system memory 230 includes a virtual channel/sub-link flow control module 266.

[0020] Virtual channel/sub-link flow control module 228,266 may determine whether each of n virtual channels is active or inactive and may write that information to a virtual channel/sub-link flow control credits register. The virtual channel/sub-link flow control module 228, 266 may then read the virtual channel/sub-link flow control credits register to determine if any virtual channels are inactive and may reallocate memory queue space in the memory 230 from the inactive channels to active channels. After reconfiguring the queues, the virtual channel/sub-link flow control module 228, 266 may write new memory offset of the channels' queue region into its virtual channel/sub-link's posted/none-posted/completion base address register. Operations implemented by the virtual channel/sub-link flow control modules 228, 266 will be discussed in greater detail below, with reference to Figs. 3 and 4.

[0021] Fig. 3 is a schematic illustration of an electronic device 300 adapted to incorporate a virtual channel/sub-link flow control module, according to embodiments. Referring to Fig. 3, electronic device 300 may comprise, among other components, at least one PCI express device 310, at least one processor 345, at least one memory module 350 and a virtual channel/sub-link

flow control module 355. In some embodiments, memory 350 may provide a storage location for data to be queued before transmission or after data is received. In some embodiments, to optimize performance, a quad/oct-port RAM memory may be used so as to allow all channels to access their region of memory/queue concurrently and independently.

[0022] PCI Express device 310 may comprise at least one virtual channel/sub-link 315a, 315b, 315c, and 315n for data communication across a PCI express link. As depicted in Fig. 3, n virtual channels are represented as virtual channel/sub-link 1, virtual channel/sub-link 2 to virtual channel/sub-link n. In some embodiments, each virtual channel/sub-link may comprise n number of traffic types. Furthermore, each traffic type may comprise header information 320a, 320b, 320n and data 325a, 325b, 325n.

[0023] In some embodiments, a virtual channel/sub-link flow control module 355 may be initialized to analyze the virtual channels 315 a, 315b, 315c, and 315n. Virtual channel/sub-link flow control module 355 determines flow control credits for each type of traffic type for each virtual channel/sub-link and writes the flow control credits to a virtual channel/sub-link flow control credit register 330. By way of example and not limitation, an inactive channel may have values of zero, while active channels may have non-zero values with the numbers indicating the size of the active channels' queue.

[0024] In some embodiments, a virtual channel/sub-link flow control module 330 may read the virtual channel/sub-link flow control credit register 330 to determine if there are inactive channels, and then reallocate the memory

queue from the inactive virtual channels 315 a, 315b, 315c, and 315n to active virtual channels 315 a, 315b, 315c, and 315n. In some embodiments, after reconfiguring queues, the virtual channel/sub-link flow control module 330 may write offset of the channels' queue region into its virtual channel/sub-link Posted/none-posted/completion Base Address Register 340.

[0025] In operation, a PCI express device 310 in an electronic device 300 may be capable of supporting a number (n) of virtual channels represented in Fig. 3 by reference numerals 315a, 315b, 315c, and 315n. A virtual channel/sub-link flow control module 355 may determine whether each of n virtual channels is active or inactive and writes that information to a virtual channel/sub-link flow control credits register 330. An example of the Type 0 configuration space header in which virtual channel/sub-link flow control credits may be placed is shown in Fig. 5. The virtual channel/sub-link flow control module 355 then reads the virtual channel/sub-link flow control credits register 330 to determine if any virtual channels are inactive and reallocates queue space in the memory 350 from the inactive channels to active channels. After reconfiguring the queues, the virtual channel/sub-link flow control module 355 writes offset of the channels' queue region into its virtual channel/sub-link Posted/none-posted/completion Base Address Register 340. An example of the Type 0 configuration space header in which a virtual channel/sub-link Posted/none-posted/completion Base Address Register may be placed is shown in Fig. 5.

[0026] By way of example and not limitation, a PCI express device may have eight virtual channels, with each channel having six registers allocated to it. The six registers may be divided between three traffic types with each traffic type having a register for header information and a register for data.

[0027] Fig. 4 is a schematic representation of a PCI-Express connection, according to embodiments. The components shown in Fig. 4 are only examples, and are not intended to suggest any limitation as to the scope of the functionality of the current invention; the current invention is not necessarily dependent on the features shown in Fig. 4.

[0028] With reference to Fig. 4, there will now be described the basic point-to-point communications channel provided by PCI-Express. A component collection consisting of two ports and the lanes connecting those ports can be referred to as a link. A link represents a dual-simplex communications channel between two components. As shown in FIG. 4, in a simplified form, a link 410 may include components 412 and 414, each including respective transmit and receive PCI device pairs 413 and 415. Two unidirectional, low-voltage, differentially driven channels 416a, 416b, 416n and 418a, 418b, 418n connect each of the ports of the components, one channel in each direction. The channel pair can be referred to as a lane. The channels 416a, 416b, 416n and 418a, 418b, 418n may carry packets 417a, 417b, 417n and 419a, 419b, 419n between the components. In some implementations, each lane provides an effective data transfer rate of 2.5 Gigabits per second per lane in each direction. For circumstances where this data bandwidth is insufficient, to scale bandwidth

a link may aggregate multiple Lanes denoted by ×N where N may be any of the supported Link widths. A ×8 Link represents an aggregate bandwidth of 20 Gigabits/second of raw bandwidth in each direction. This base specification 1.0 describes operations for ×1, ×2, ×4, ×8, ×12, ×16, and ×32 Lane widths. In some implementations only symmetrical links may be permitted, such that a link includes the same number of lanes in each direction.

[0029] Various classes of data may be transmitted across a PCI-express link. By way of example and not limitation, transaction layer packets (TLPs) may be transmitted from one link to another as necessary, subject to the outing mechanisms and rules discussed below. As transactions are carried out between PCI Express requesters and completers, four separate address spaces are used: Memory, IO, Configuration, and Message. Accesses to the four address spaces in PCI Express may be accomplished using split-transaction requests and completions. In PCI Express, the completion following a request may be initiated by the completer when it has data and/or status ready for delivery. The fact that the completion is separated in time from the request which caused it also means that two separate TLPs are generated, with independent routing for the request TLP and the Completion TLP. While a link is free for other activity in the time between a request and its subsequent completion, a split-transaction protocol involves some additional overhead as two complete TLPs must be generated to carry out a single transaction.

[0030] To mitigate the penalty of the request-completion latency, messages and some write transactions in PCI Express are posted/none-

posted/completion, meaning the write request (including data) is sent, and the transaction is over from the requester's perspective as soon as the request is sent out of an egress port. In PCI Express, write posting to memory is considered acceptable in exchange for the higher performance. On the other hand, writes to IO and configuration space may change device behavior, and write posting is not permitted. A completion will always be sent to report status of the IO or configuration write operation.

[0031] Fig. 5 is a schematic illustration of one embodiment in which queue sharing and reconfiguring in a point to point link may be implemented. The components shown in Fig. 5 are only examples, and are not intended to suggest any limitation as to the scope of the functionality of the current invention; the current invention is not necessarily dependent on the features shown in Fig. 5.

[0032] With reference to Fig. 5, in some instances, a PCI device 510 may be capable of supporting eight virtual channels. By way of example and not limitation, if one were to assume a PCI device 510 uses 4KB queue for Posted/none-posted/completion traffic, which provides 512 bytes for each virtual channel/sub-link if all eight channels are active. In some embodiments, when flow control initialization is complete, PCI device 510 may read from virtual channel/sub-link flow control registers 515 to determine the virtual channel/sub-link flow control credits 520. In the depicted example, the PCI device may read values of 40h (64 decimal) for four active virtual channels and zeros for the remaining 4 inactive channels. Thereafter, queue space from the four inactive

channels may be reallocated to the four active channels thereby allowing each active channel to have 1024 byte queues, assuming the redistribution is even across the four active channels. After reconfiguring the queues, offset of the channels' queue region may be written to a virtual channel/sub-link posted/none-posted/completion data base address register (VCPD BAR) 625.

[0033] As described above, traffic types such as but not limited to post, non-post and completion are used with PCI express links. By way of example and not limitation, Memory Write requests are posted/none-posted/completion. No completion is expected or sent. Configuration Read and Write requests are non-posted/none-posted/completion. A completion without data (Cpl) may be returned by the completer to report status of the configuration space write operation. A completion with data (CplD) may be returned by the completer with requested data and to report status of the read operation.

[0034] Thus, described herein are exemplary system and methods for implementing queue sharing and queue optimization in PCI express linked systems. The methods described herein may be embodied as logic instructions on a computer-readable medium. When executed on a processor, the logic instructions cause a general purpose computing device to be programmed as a special-purpose machine that implements the described methods. The processor, when configured by the logic instructions to execute the methods recited herein, constitutes structure for performing the described methods.

[0035] Reference in the specification to "one embodiment" or "an embodiment" means that a particular feature, structure, or characteristic

described in connection with the embodiment is included in at least an implementation. The appearances of the phrase “in one embodiment” in various places in the specification are not necessarily all referring to the same embodiment.

CLAIMS

What is claimed is:

1. A method for queue sharing among virtual channels in a point to point link comprising:

determining, in an electrical device, whether a virtual channel/sub-link is inactive; and

in response to a determination that at least one virtual channel/sub-link is inactive, reallocating queue space from the at least one inactive channel to at least one active channel.

2. The method of claim 1 further comprising initializing at least one virtual channel/sub-link on the point-to-point link, wherein initializing at least one virtual channel/sub-link on the point-to-point link comprises writing, a number of flow control credits for at least one traffic type on the at least one virtual channel/sub-link to a virtual channel/sub-link flow control credits register.

3. The method of claim 2, wherein initializing the virtual channel/sub-link comprises:

writing a zero value to the virtual channel/sub-link flow control credits register for inactive virtual channels; and

writing a non-zero value to the virtual flow control credits register for active virtual channels.

4. The method of claim 3, wherein writing a non-zero value to the virtual flow control credits register for active virtual channels comprises writing a value that corresponds to the size of the queue space for the active virtual channels.

5. The method of claim 1, wherein determining, in an electrical device, whether a virtual channel/sub-link is inactive comprises:

initiating a virtual channel/sub-link flow control module; and

reading the flow control credits register for the virtual channel.

6. The method of claim 1, wherein reallocating queue space from the at least one inactive channel to at least one active channel comprises reallocating all queue space from the at least one inactive channel to the at least one active channel.

7. The method of claim 1, wherein reallocating queue space from the at least one inactive channel to at least one active channel comprises reallocating a first portion of queue space from the at least one inactive channel to the a first active channel and a second portion of queue space from the at least one inactive channel to the a second active channel.

8. The method of claim 1, further comprising:

writing, in an electrical device, offset information of a virtual channels' queue region in a virtual channel/sub-link posted/none-posted/completion data base address register for data to be transmitted across a known number of virtual channels; and

transmitting data, in an electrical device, across the known number of virtual channels.

9. The method of claim 8, wherein the data is transmitted through a PCI express link.

10. The method of claim 2, wherein writing, a number of flow control credits for at least one traffic type on the at least one virtual channel/sub-link to a virtual channel/sub-link flow control credits register comprises allocating six registers for traffic types, named virtual channel/sub-link flow control credits registers, for each of eight possible virtual channels.

11. The method of claim 10, wherein the six registers for traffic types comprises three traffic types with each of the three traffic type using two registers comprising one register for a header and one register for data.

12. A computer program product comprising logic instructions stored on a computer-readable medium which, when executed by a computer processor, configure the processor to:

determine, in an electrical device, whether a virtual channel/sub-link is inactive; and

in response to a determination that at least one virtual channel/sub-link is inactive, reallocate queue space from the at least one inactive channel to at least one active channel.

13. An electronic device comprising:

at least one processor;

at least one PCI express link;

a virtual channel/sub-link flow control module; and

a memory module communicatively connected to the one or more processors and comprising logic instructions which, when executed on the one or more processors configure the one or more processors to:

determine, in an electrical device, whether a virtual channel/sub-link is inactive; and

in response to a determination that at least one virtual channel/sub-link is inactive, reallocate queue space from the at least one inactive channel to at least one active channel.

14. The electronic device of claim 13, further comprising logic to initiate at least one virtual channel/sub-link on the point-to-point link, wherein logic to initiate at least one virtual channel/sub-link on the point-to-point link comprises logic to write, a number of flow control credits for at least one traffic type on the at least one virtual channel/sub-link to a virtual channel/sub-link flow control credits register.

15. The electronic device of claim 14, wherein logic to initiate the virtual channel/sub-link comprises logic to

write a zero value to the virtual channel/sub-link flow control credits register for inactive virtual channels; and

write a non-zero value to the virtual flow control credits register for active virtual channels.

16. The electronic device of claim 13, wherein logic to determine, in an electrical device, whether a virtual channel/sub-link is inactive comprises logic to:

initiate a virtual channel/sub-link flow control module; and

read the flow control credits register for the virtual channel.

17. The electronic device of claim 13, wherein logic to reallocate queue space from the at least one inactive channel to at least one active channel comprises logic to reallocate all queue space from the at least one inactive channel to the at least one active channel.

18. The electronic device of claim 13, wherein logic to reallocate queue space from the at least one inactive channel to at least one active channel comprises logic to reallocate a first portion of queue space from the at least one inactive channel to the a first active channel and a second portion of queue space from the at least one inactive channel to the a second active channel.

19. The electronic device of claim 13, further comprising logic to:

write offset information of a virtual channels' queue region in a virtual channel/sub-link posted/none-posted/completion data base address register for data to be transmitted across a known number of virtual channels; and

transmit data, in an electrical device, across the known number of virtual channels.

20. The electronic device of claim 14, wherein logic to write, a number of flow control credits for at least one traffic type on the at least one virtual channel/sub-link to a virtual channel/sub-link flow control credits register comprises logic to allocate six registers for traffic types, named virtual channel/sub-link flow control credits registers, for each of eight possible virtual channels, wherein the six registers for traffic types comprise three traffic types with each of the three traffic type using two registers comprising one register for a header and one register for data

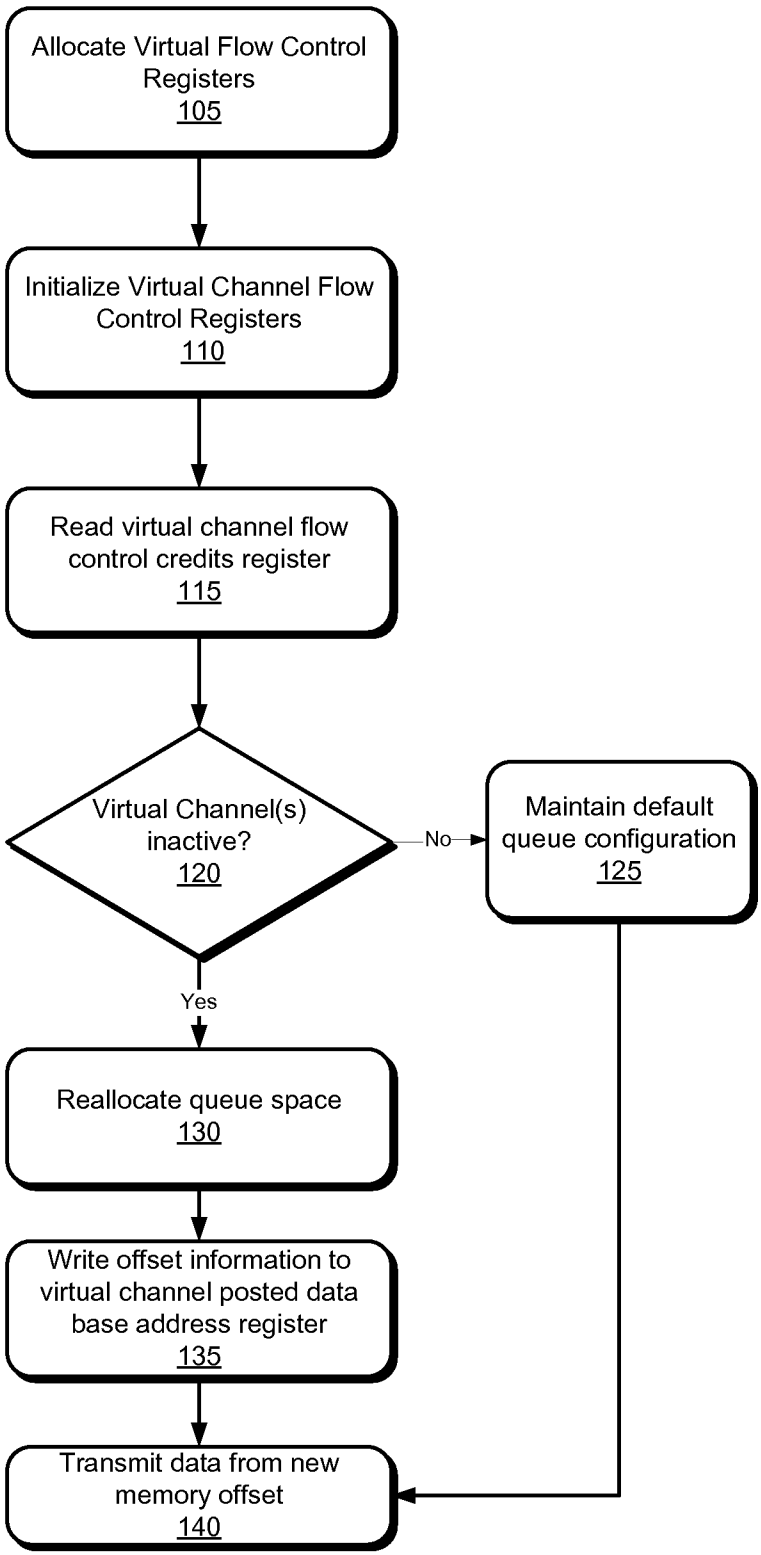

Allocate Virtual Flow Control Registers
105
Initialize Virtual Channel Flow Control Registers
110
Read virtual channel flow control credits register
115
Virtual Channel(s) inactive?
120
No
Maintain default queue configuration
125
Yes
Reallocate queue space
130
Write offset information to virtual channel posted data base address register
135
Transmit data from new memory offset
140

Fig. 1

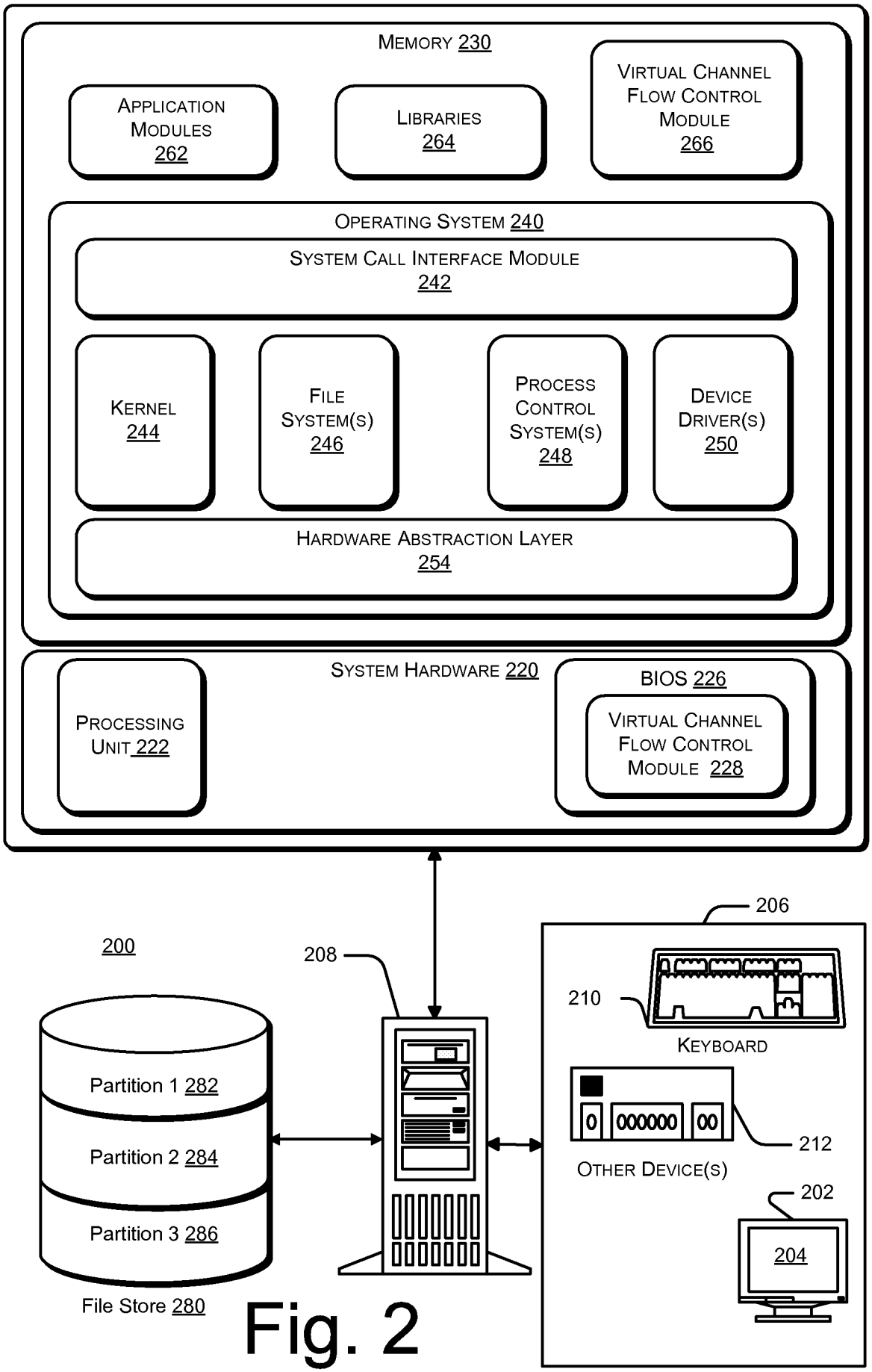

MEMORY 230
APPLICATION MODULES 262
LIBRARIES 264
VIRTUAL CHANNEL FLOW CONTROL MODULE 266
OPERATING SYSTEM 240
SYSTEM CALL INTERFACE MODULE 242
KERNEL 244
FILE SYSTEM(S) 246
PROCESS CONTROL SYSTEM(S) 248
DEVICE DRIVER(S) 250
HARDWARE ABSTRACTION LAYER 254
SYSTEM HARDWARE 220
PROCESSING UNIT 222
BIOS 226
VIRTUAL CHANNEL FLOW CONTROL MODULE 228
200
208
206
210
KEYBOARD
212
OTHER DEVICE(S)
202
204
Partition 1 282
Partition 2 284
Partition 3 286
File Store 280

Fig. 2

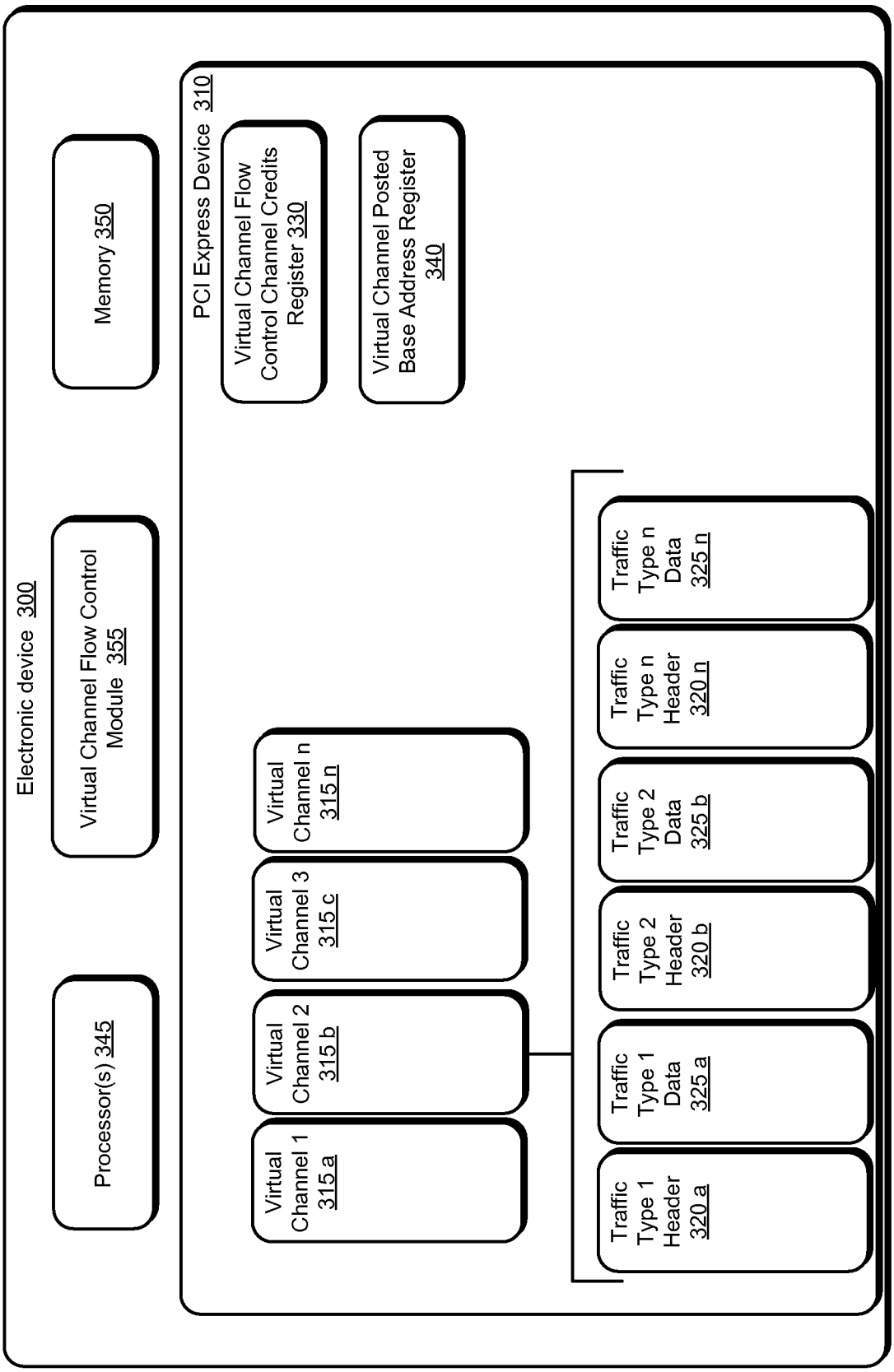

Electronic device 300
Processor(s) 345
Virtual Channel Flow Control Module 355
Memory 350
PCI Express Device 310
Virtual Channel Flow Control Channel Credits Register 330
Virtual Channel Posted Base Address Register 340
Virtual Channel 1 315 a
Virtual Channel 2 315 b
Virtual Channel 3 315 c
Virtual Channel n 315 n
Traffic Type 1 Header 320 a
Traffic Type 1 Data 325 a
Traffic Type 2 Header 320 b
Traffic Type 2 Data 325 b
Traffic Type n Header 320 n
Traffic Type n Data 325 n

Fig. 3

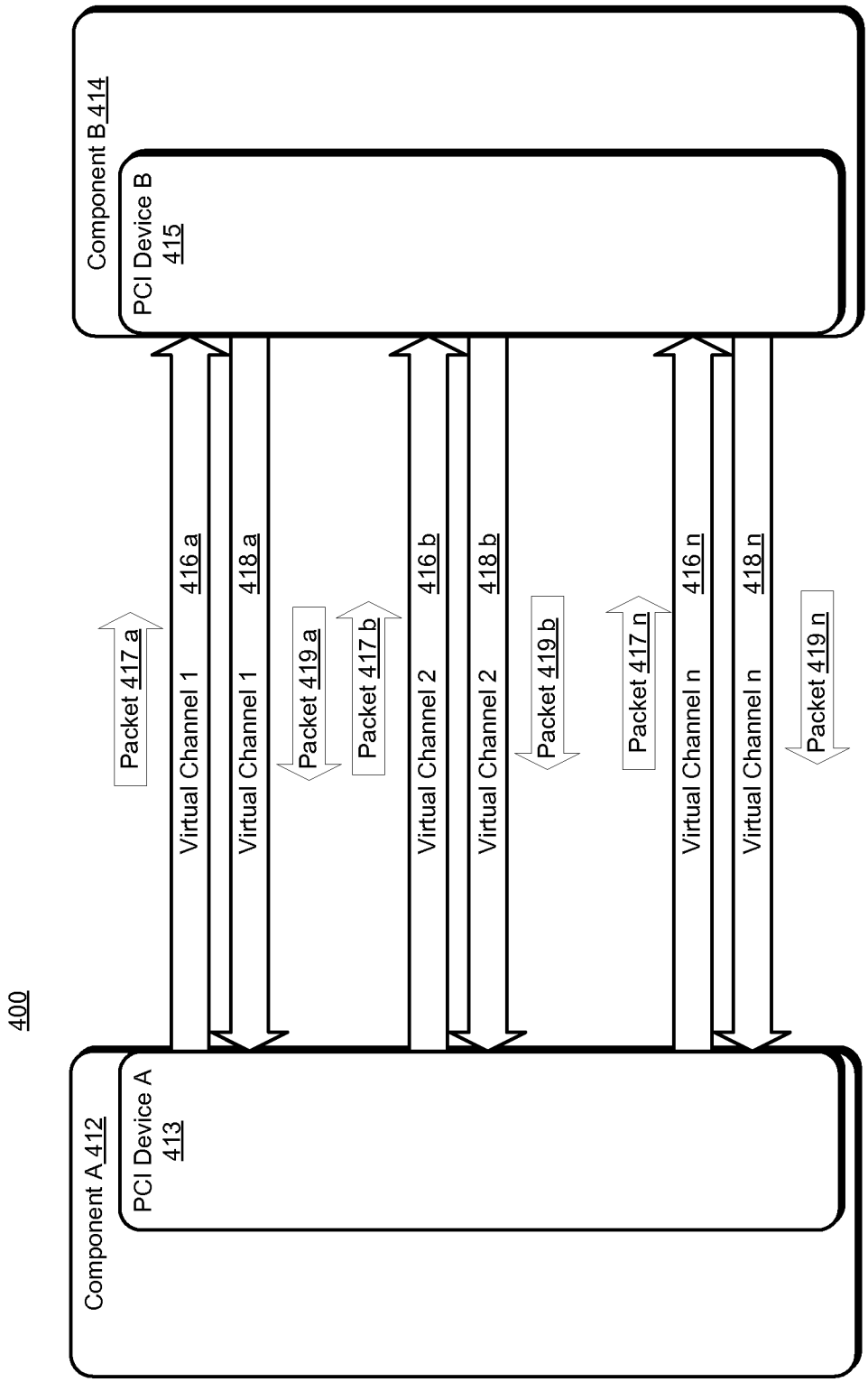
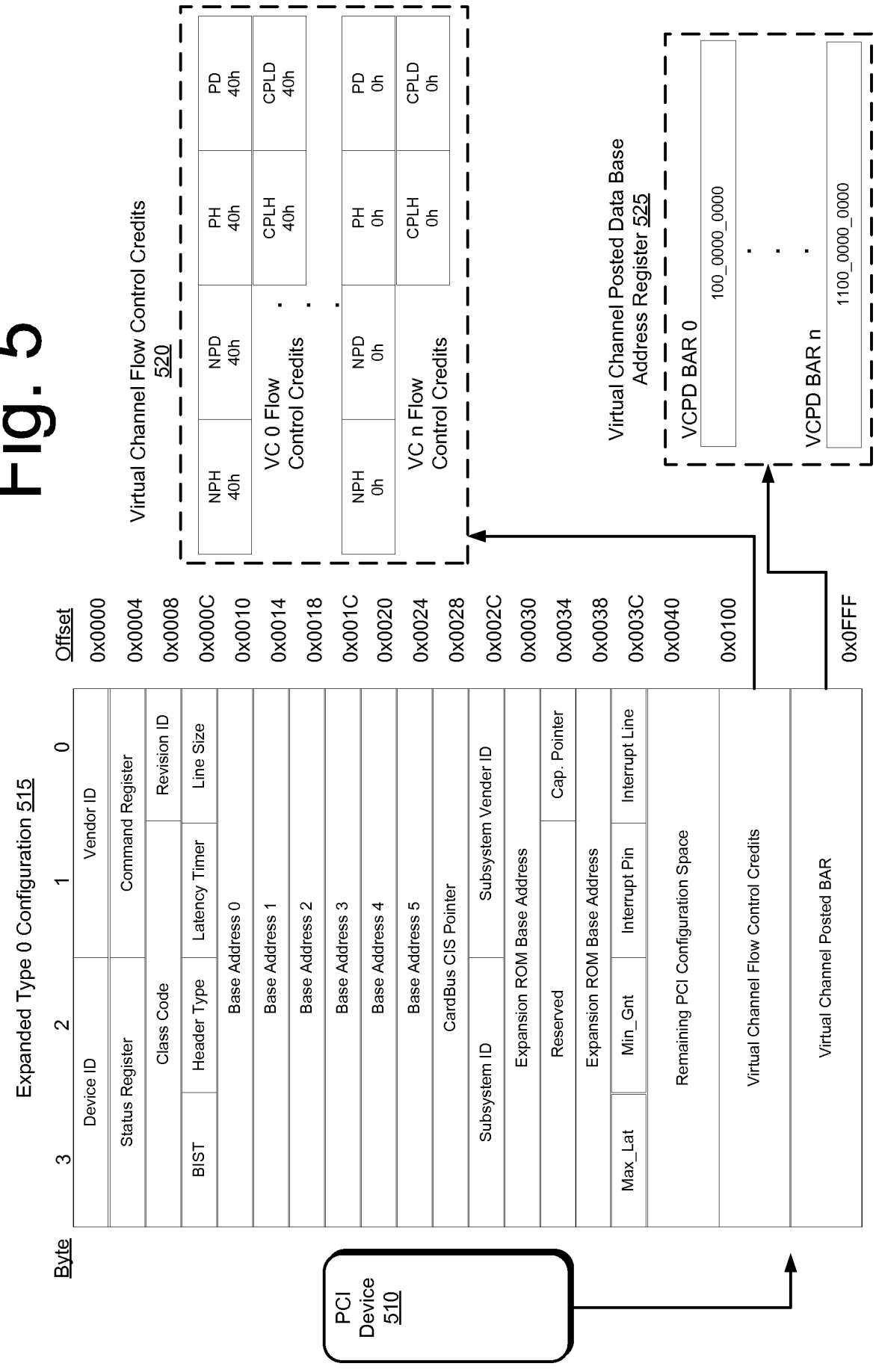

400
Component A 412
PCI Device A 413
Component B 414
PCI Device B 415
Packet 417 a
Virtual Channel 1 416 a
Virtual Channel 1 418 a
Packet 419 a
Packet 417 b
Virtual Channel 2 416 b
Virtual Channel 2 418 b
Packet 419 b
Packet 417 n
Virtual Channel n 416 n
Virtual Channel n 418 n
Packet 419 n

Fig. 4

Fig. 5


Byte
Offset
3
2
1
0
Expanded Type 0 Configuration 515
Device ID
Vendor ID
Status Register
Command Register
Class Code
Revision ID
BIST
Header Type
Latency Timer
Line Size
Base Address 0
Base Address 1
Base Address 2
Base Address 3
Base Address 4
Base Address 5
CardBus CIS Pointer
Subsystem ID
Subsystem Vender ID
Expansion ROM Base Address
Reserved
Cap. Pointer
Expansion ROM Base Address
Max_Lat
Min_Gnt
Interrupt Pin
Interrupt Line
Remaining PCI Configuration Space
Virtual Channel Flow Control Credits
Virtual Channel Posted BAR
0x0000
0x0004
0x0008
0x000C
0x0010
0x0014
0x0018
0x001C
0x0020
0x0024
0x0028
0x002C
0x0030
0x0034
0x0038
0x003C
0x0040
0x0100
0x0FFF
PCI Device 510
Virtual Channel Flow Control Credits 520
VC 0 Flow Control Credits
NPH 40h
NPD 40h
PH 40h
CPLH 40h
CPLD 40h
PD 40h
VC n Flow Control Credits
NPH 0h
NPD 0h
PH 0h
CPLH 0h
CPLD 0h
PD 0h
Virtual Channel Posted Data Base Address Register 525
VCPD BAR 0
100_0000_0000
VCPD BAR n
1100_0000_0000

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2008/065460

A. CLASSIFICATION OF SUBJECT MATTER

G06F 13/14(2006.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 8 : G06F

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

Korean Utility models and applications for Utility models since 1975

Japanese Utility models and application for Utility models since 1975

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

e-KIPASS(KIPO internal) : "virtual", "channel", "active", "allocate", "queue"

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 7,194,271 B2 (GOLESTANI et al.) 20 March 2007 See the abstract, column 26, line 37 - column 4, line 13, and figure 2	1-20
A	US 7,218,638 B2 (MOLL) 15 May 2007 See the abstract, column 4, line 45 - column 6, line 27, and figures 2-3	1-20
A	US 7,181,578 B1 (GUHA et al.) 20 February 2007 See the abstract, column 11, line 30 - column 14, line 17, and figures 2-3	1-20
A	US 6,353,867 B1 (QURESHI et al.) 5 March 2002 See the abstract, column 4, line 10 - column 7, line 30, and figures 1-2	1-20

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

☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art

"&" 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 FEBRUARY 2009 (25.02.2009)	**25 FEBRUARY 2009 (25.02.2009)**
Name and mailing address of the ISA/KR Korean Intellectual Property Office Government Complex-Daejeon, 139 Seonsa-ro, Seo-gu, Daejeon 302-701, Republic of Korea Facsimile No. 82-42-472-7140	Authorized officer KIM, JONG KWON Telephone No. 82-42-481-8584

Form PCT/ISA/210 (second sheet) (July 2008)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2008/065460

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 7194271 B2	20.03.2007	CN 1684439 A	19.10.2005
		EP 1587254 A1	19.10.2005
		JP 2005-304036 A	27.10.2005
		KR 10-2006-0046662	17.05.2006
		US 2005-0226191 A1	13.10.2005
US 7218638 B2	15.05.2007	US 7272151 B2	18.09.2007
		US 7313146 B2	25.12.2007
		US 7319702 B2	15.01.2008
		US 2004-0037313 A1	26.02.2004
		US 2004-0078459 A1	22.04.2004
		US 2004-037313 A1	26.02.2004
		US 2004-078459 A1	22.04.2004
US 7181578 B1	20.02.2007	NONE	
US 6353867 B1	05.03.2002	NONE	