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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 July 2005 (21.07.2005)

PCT

(10) International Publication Number
WO 2005/066763 A2

(51) International Patent Classification[7]: G06F 3/14

(21) International Application Number:
PCT/US2004/043650

(22) International Filing Date:
22 December 2004 (22.12.2004)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/746,422 24 December 2003 (24.12.2003) US

(71) Applicant *(for all designated States except US)*: **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, CA 95052 (US).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: **KULKARNI, Sunil, A.** [IN/US]; 13347 NW Keeton Park Lane, Portland, OR 97229 (US).

(74) Agents: **VINCENT, Lester, J.** et al.; Blakely, Sokoloff, Taylor & Zafman LLP, 7th Floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, 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, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: GRAPHICS MEMORY SWITCH

(57) Abstract: A graphics device delivers a graphics address to a graphics memory switch that includes a graphics random access memory translator and a graphics memory page table. The graphics memory address is delivered to the graphics memory switch via a point-to-point, packet based interconnect. The graphics memory switch generates a physical system memory address and delivers the physical address to a root complex. The physical system memory address is delivered to the root complex via a point-to-point, packet based interconnect.

WO 2005/066763 A2

GRAPHICS MEMORY SWITCH

Field Of The Invention

[0001] The present invention pertains to the field of semiconductor devices. More particularly, this invention pertains to the field of using a graphics memory switch to provide a graphics device access to system memory.

Background of the Invention

[0002] The rapid and efficient transfer of information between a graphics device and system memory has been and will continue to be one of the most challenging tasks faced by computer system component designers. Through the years, different interface protocols have been used to accomplish these transfers. Several years ago, the Peripheral Component Interconnect (PCI) bus was a commonly used implementation to couple graphics devices to memory controllers. As graphics memory bandwidth requirements increased, the Accelerated Graphics Port (AGP) specification was created and adopted by a large segment of the computer industry.

[0003] One of the main advantages of the AGP implementations is the ability of the graphics device to view a large, contiguous graphics memory space where multi-megabyte textures, bitmaps, and graphics commands are stored. A graphics address remapping table is used to generate addresses to system memory from

graphics memory addresses. There is no actual memory behind the graphics memory space, but the graphics address remapping table and associated translation circuitry provides access to actual system memory pages that may be scattered throughout the system memory.

[0004] Graphics memory bandwidth requirements continue to increase, and faster interconnect technologies are being developed to keep ahead of the growing requirements. One such interconnect technology is based on the PCI Express specification (PCI Express Base Specification, revision 1.0a). It would be desirable to provide a large, contiguous, graphics memory space for use with these emerging interconnect technologies.

Brief Description of the Drawings

[0005] The invention will be understood more fully from the detailed description given below and from the accompanying drawings of embodiments of the invention which, however, should not be taken to limit the invention to the specific embodiments described, but are for explanation and understanding only.

[0006] Figure 1 is a block diagram of one embodiment of a computer system including a graphics memory switch.

[0007] Figure 2 is a block diagram of a graphics memory switch including a graphics random access memory translator and a graphics memory page table.

[0008] Figure 3 is a block diagram demonstrating a conversion from a virtual graphics memory address to a physical system memory address.

[0009] Figure 4 is a block diagram of a graphics memory switch including a closer look at a graphics random access memory translator.

[0010] Figure 5 is a block diagram of a graphics memory switch that includes a virtual PCI-PCI bridge.

[0011] Figure 6 is a block diagram of several graphics components coupled to a root complex through a graphics memory switch.

[0012] Figure 7 is a flow diagram of one embodiment of a method for generating a physical memory address from a virtual graphics memory address received over a point-to-point, packet based interconnect.

Detailed Description

[0013] In general, a graphics device delivers a virtual graphics address to a graphics memory switch that includes a graphics random access memory translator and a graphics memory page table. The virtual graphics memory address is delivered to the graphics memory switch via a point-to-point, packet based interconnect. The graphics memory switch generates a physical system memory address and delivers the physical address to a root complex. The physical system memory address is delivered to the root complex via a point-to-point, packet based interconnect.

[0014] For the embodiments described herein, virtual graphics addresses are defined as graphics addresses that are physical, but where no real physical memory exists at these addresses. In other words, converting virtual graphics

addresses to physical memory addresses involves only a graphics memory switch and a graphics memory page table, and no system page tables are required. Another way to look at the conversion of virtual graphics addresses to physical system memory addresses is to see the conversion as including converting physical graphics addresses (contiguous, non-existent) to physical system memory addresses (non-contiguous, existent).

[0015] Figure 1 is a block diagram of one embodiment of a computer system 100 including a graphics memory switch 130. The system 100 includes a processor 110 coupled to a root complex 140. The root complex 140 includes a memory controller (not shown) to provide communication with a system memory 150. The root complex 140 is further coupled to a switch 160. The switch 160 is coupled to an endpoint device 170 via an interconnect 165. The switch 160 is also coupled to an endpoint device 180 via an interconnect 163. The endpoint devices 170 and 180 may be any of a wide variety of computer system components, including hard disk drives, optical storage devices, communications devices, etc.

[0016] For this example embodiment, the links 163 and 165 adhere to the PCI Express specification. The root complex 140 and the switch 160 also comply with the PCI Express specification.

[0017] The system 100 further includes a graphics device 120 that is coupled to a graphics memory (GM) switch 130 via a point-to-point, packet based

interconnect, which for this example embodiment is a PCI Express interconnect 125. The GM switch 130 is further coupled to the root complex 140 via another point-to-point interconnect, which for this example embodiment is a PCI Express Link 135.

[0018] The graphics device 120 may be a component soldered to a motherboard, or may be located on a graphics card, or may be integrated into a larger component.

[0019] Although the system 100 is shown with the graphics device 120, the GM switch 130, and the root complex 140 as separate devices, other embodiments are possible where the GM switch 130 is integrated into one device along with the root complex 140. Yet other embodiments are possible where the graphics device 120, the GM switch 130, and the root complex 140 are integrated into a single device.

[0020] For the system 100, a contiguous memory called graphics random access memory (GRAM) is allocated in system address space. However, there is no real memory behind the GRAM. The GRAM is seen by the graphics device 120 as a large, contiguous memory space. An operating system will allocate the GRAM as pages scattered all over the system memory 150, wherever it can find space.

[0021] Figure 2 is a block diagram of the GM switch 130. The GM switch includes a GRAM translator 132 and a graphics memory page (GMP) table 134. The GMP Table 134 is loaded with physical addresses under software control

(device driver, operating system, etc.). The GRAM translator 132 receives virtual graphics memory addresses over the PCI Express link 125. The GRAM translator 132 uses the virtual addresses to access the GMP table 134. The GRAM translator 132 generates physical addresses which are delivered to the root device 140 via the PCI Express link 135.

[0022] The GMP table 134 is an address translation table. As previously mentioned, the GMP table 134 holds the addresses of the physical memory allocated by the operating system. The size of the table 134 may depend on the size of the GRAM. For example, if the GRAM is 2GB, using 32-bit addresses for the pages and 4kbytes per page, the GMP Table 134 will be (2*1024*1024*1024)/(4*1024) entries * 4 bytes per entry = 2Mbytes. Although the GMP Table 134 is shown in this example embodiment as being integrated into the GM switch 130, other embodiments are possible where the GMP Table is located in memory separate from but local to the GM switch 130 or in system memory 150.

[0023] Figure 3 is a block diagram demonstrating a conversion from a virtual graphics memory address to a physical system memory address. The input to the GRAM translator 132 arrives over the PCI Express link 125. The input is a GRAM address "X" that the graphics device 120 needs to access. The GRAM space exists outside the system memory range. The GRAM space begins at an address denoted as GRAM Base. Several address locations in GRAM space are

shown; addresses X, X+1, and X+2. The translator 132 takes the virtual graphics address X and converts it into an index to the GMP Table 134. The address at the specified GMP Table entry gives the actual physical address of the page of memory that the operating system has allocated. For this example, only three entries of the GMP Table 134 are shown; entries A, B, and C. The addresses stored in the A, B, and C entries correspond to regions A, B, and C of the system memory 150. For this example, the virtual address "X" provides an index to the C entry of the GMP Table 134. The GMP Table 134 delivers the physical address from the C entry to the root complex 140, which allows access to region C of the system memory.

[0024] Figure 4 is a block diagram of the GM switch 130 including a closer look at the GRAM Translator 132. As described above, a virtual graphics address "X" arrives from the graphics device. The GRAM translator 132 receives the address and uses the portion of the virtual address that denotes a page number to form an index into the GMP Table 134. The GRAM Translator 132 generates the index by subtracting the GRAM Base address from the address "X". The physical address stored at the entry C of the GMP table 134 is combined with the portion of the virtual address that indicates an offset into the page. The resulting address is delivered to the root complex 140 via the PCI Express link 135.

[0025] The overall functioning environment of the GRAM Translator may be such that the same operating system drivers that are used for AGP

implementations can be used for managing the GMP Table and for allocating and releasing GRAM pages. In AGP, this driver is often referred to as the GART (graphics address remapping table) driver. Being able to reuse the existing GART drivers may ease the transition from AGP to PCI Express.

[0026] A video device driver may request N number of GRAM pages to the operating system. The GMP Table driver may allocate these pages in the memory and populate the GMP Table 134. The video driver will reserve the pages it needs to use for a particular application. The graphics device's view of the GRAM will be starting from the GRAM Base address and extending as far as is required. When the graphics device 120 needs to use the GRAM, it will issue a transaction for an address with the GRAM range. The GRAM translator 132, after checking to be sure that the request is within an appropriate range, will calculate an index into the GMP Table 134 and picks up an address of the actual page in the system memory 150. This address is sent over the PCI Express link 135 to the root complex 140 so that the system memory 150 can be accessed.

[0027] Figure 5 is a block diagram of a graphics memory switch that includes a virtual PCI-PCI bridge 136. When the PCI-PCI bridge 136 is encountered by an operating system during enumeration, an appropriate driver (perhaps a GART driver) is loaded. The GM switch 130 also includes a configuration space 138 which includes registers which are used for setting up the GMP Table for proper operation during runtime. The registers in the configuration space 138 may

comply with the AGP specification so that no change in existing software is necessary.

[0028] Figure 6 is a block diagram of one example embodiment of several graphics components 610, 620, and 630 coupled to a root complex 630 through a graphics memory switch 620. A configuration of this type can provide a system that allows multiple graphics devices. Each of the graphics devices may or may not support multiple displays. A single driver can be loaded when the operating system encounters the virtual PCI-PCI bridge 628 that connects to the root complex 630. The multiple graphics devices 610, 620, and 630 can each have the same contiguous view of GRAM space and can share the information stored in GRAM space.

[0029] The graphics drivers 610, 620, and 630 are coupled to the virtual PCI-PCI bridge 628 via virtual PCI-PCI bridges 622, 624, and 626, respectively.

[0030] Figure 7 is a flow diagram of one embodiment of a method for generating a physical memory address from a virtual graphics memory address received over a point-to-point, packet based interconnect. At block 710, a virtual graphics memory address is received from a graphics device over a point-to-point, packet based interconnect. A physical memory address is generated using a graphics memory translator at block 720. Then, at block 730, the physical memory address is delivered to a root complex device.

[0031] In the foregoing specification the invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than in a restrictive sense.

[0032] Reference in the specification to "an embodiment," "one embodiment," "some embodiments," or "other embodiments" means that a particular feature, structure, or characteristic described in connection with the embodiments is included in at least some embodiments, but not necessarily all embodiments, of the invention. The various appearances of "an embodiment," "one embodiment," or "some embodiments" are not necessarily all referring to the same embodiments.

CLAIMS

What is claimed is:

1. An apparatus, comprising:

an input to receive a virtual graphics memory address over a point-to-point, packet-based interconnect; and

a graphics address translator to receive the virtual graphics memory address and to generate a physical memory address.

2. The apparatus of claim 1, the graphics address translator including a graphics memory page table.

3. The apparatus of claim 2, the graphics memory page table to store a plurality of physical addresses that are allocated by an operating system.

4. The apparatus of claim 3, the graphics memory page table including a plurality of entries, each of the entries to store 32-bit addresses.

5. The apparatus of claim 4, wherein the point-to-point, packet based interconnect adheres to a PCI Express specification.

6. The apparatus of claim 5, further comprising an output to deliver the physical address to a root complex device over a second point-to-point, packet based interconnect.

7. The apparatus of claim 1, further comprising a root complex function to receive the physical address and to deliver the physical address to a memory controller.

8. The apparatus of claim 1, the graphics address translator to access an external graphics memory page table.

9. An apparatus, comprising:

a graphics controller to generate a virtual graphics memory address;

a graphics address translator to receive the virtual graphics memory address and to generate a physical memory address; and

an output to deliver the physical address to a root complex device over a point-to-point, packet based interconnect.

10. The apparatus of claim 9, the graphics address translator including a graphics memory page table.

11. The apparatus of claim 10, the graphics memory page table to store a plurality of physical addresses that are allocated by an operating system.

12. The apparatus of claim 11, the graphics memory page table including a plurality of entries, each of the entries to store 32-bit addresses.

13. The apparatus of claim 12, wherein the point-to-point, packet based interconnect adheres to a PCI Express specification.

14. A system, comprising:

a graphics device;

a graphics memory switch device to receive a virtual graphics memory address from the graphics device over a first point-to-point, packet-based interconnect, the graphics memory switch device including a graphics memory translator to receive the virtual graphics memory address and to generate a physical memory address; and

a root complex device to receive the physical memory address from the graphics memory switch device over a second point-to-point, packet based interconnect.

15. The system of claim 14, the graphics address translator including a graphics memory page table.

16. The system of claim 15, wherein the first and second point-to-point, packet based interconnects adhere to a PCI Express specification.

17. A system, comprising:

a graphics device, including a graphics memory switch device that includes a graphics memory translator to receive a virtual graphics memory address and to generate a physical memory address; and

a root complex device to receive the physical memory address from the graphics memory switch device over a point-to-point, packet based interconnect.

18. The system of claim 17, the graphics address translator including a graphics memory page table.

19. The system of claim 18, wherein the point-to-point, packet based interconnect adheres to a PCI Express specification.

20. A system, comprising:

a graphics device; and

a memory controller hub including

a graphics memory switch device to receive a virtual graphics memory

address from the graphics device over a point-to-point, packet-

based interconnect, the graphics memory switch device including a

graphics memory translator to receive the virtual graphics memory

address and to generate a physical memory address,

a memory controller, and

a root complex device to receive the physical memory address from the

graphics memory switch device and to deliver the physical memory

address to the memory controller.

21. The system of claim 20, the graphics address translator including a graphics memory page table.

22. The system of claim 21, wherein the point-to-point, packet based interconnect adheres to a PCI Express specification.

23. A method, comprising:

receiving a virtual graphics memory address from a graphics device over a point-to-point, packet based interconnect;

generating a physical memory address using a graphics memory translator; and

delivering the physical memory address to a root complex device.

24. The method of claim 23, wherein receiving a virtual graphics memory address from a graphics device over a point-to-point, packet based interconnect includes receiving a virtual graphics memory address from a graphics device over a point-to-point, packet based interconnect that adheres to a PCI Express specification.

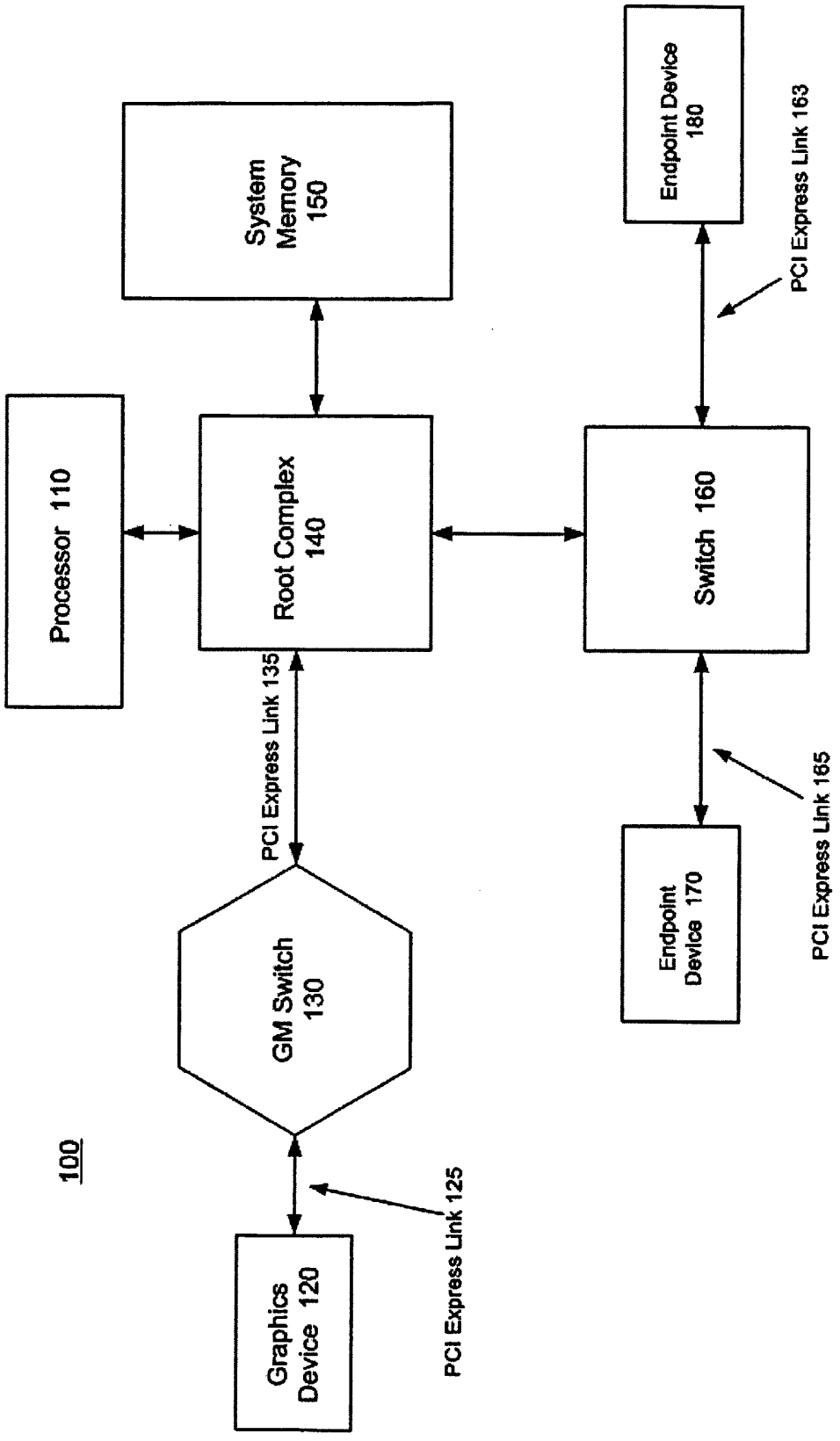
100
Processor 110
System Memory 150
Root Complex 140
Switch 160
Endpoint Device 180
PCI Express Link 163
PCI Express Link 135
GM Switch 130
Endpoint Device 170
PCI Express Link 165
Graphics Device 120
PCI Express Link 125

Figure 1

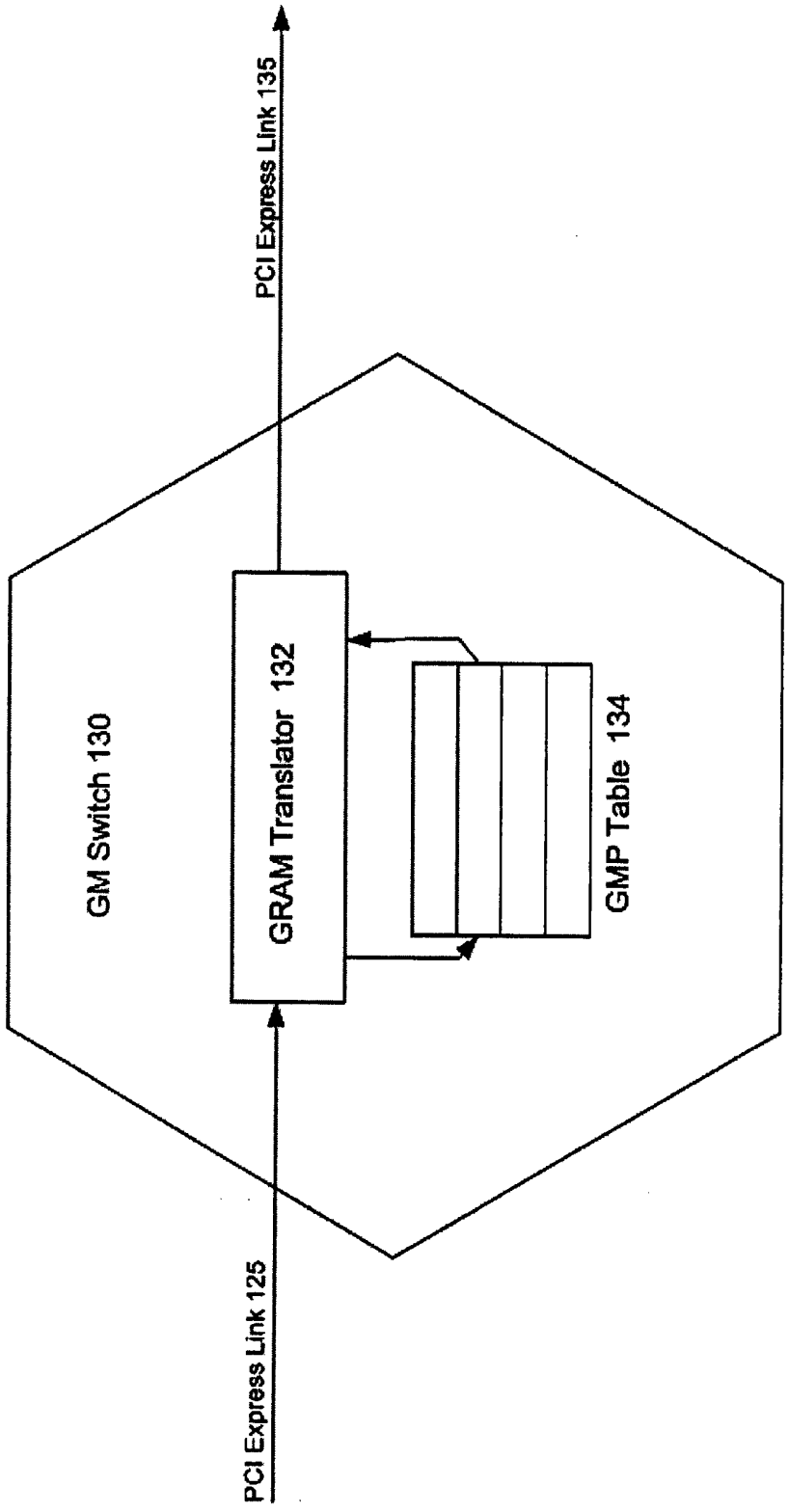
PCI Express Link 135
GM Switch 130
GRAM Translator 132
GMP Table 134
PCI Express Link 125

Figure 2

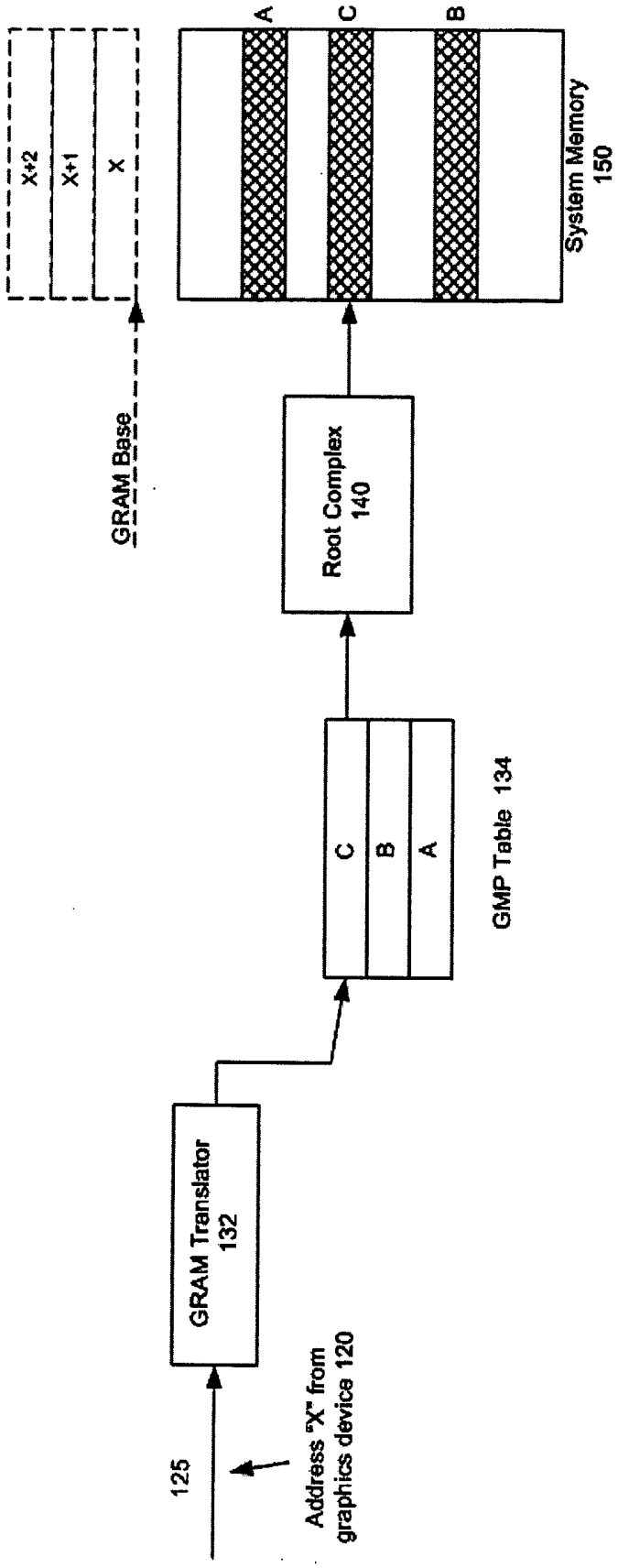
X+2
X+1
X
GRAM Base
A
C
B
System Memory
150
Root Complex
140
C
B
A
GMP Table 134
GRAM Translator
132
125
Address "X" from
graphics device 120

Figure 3

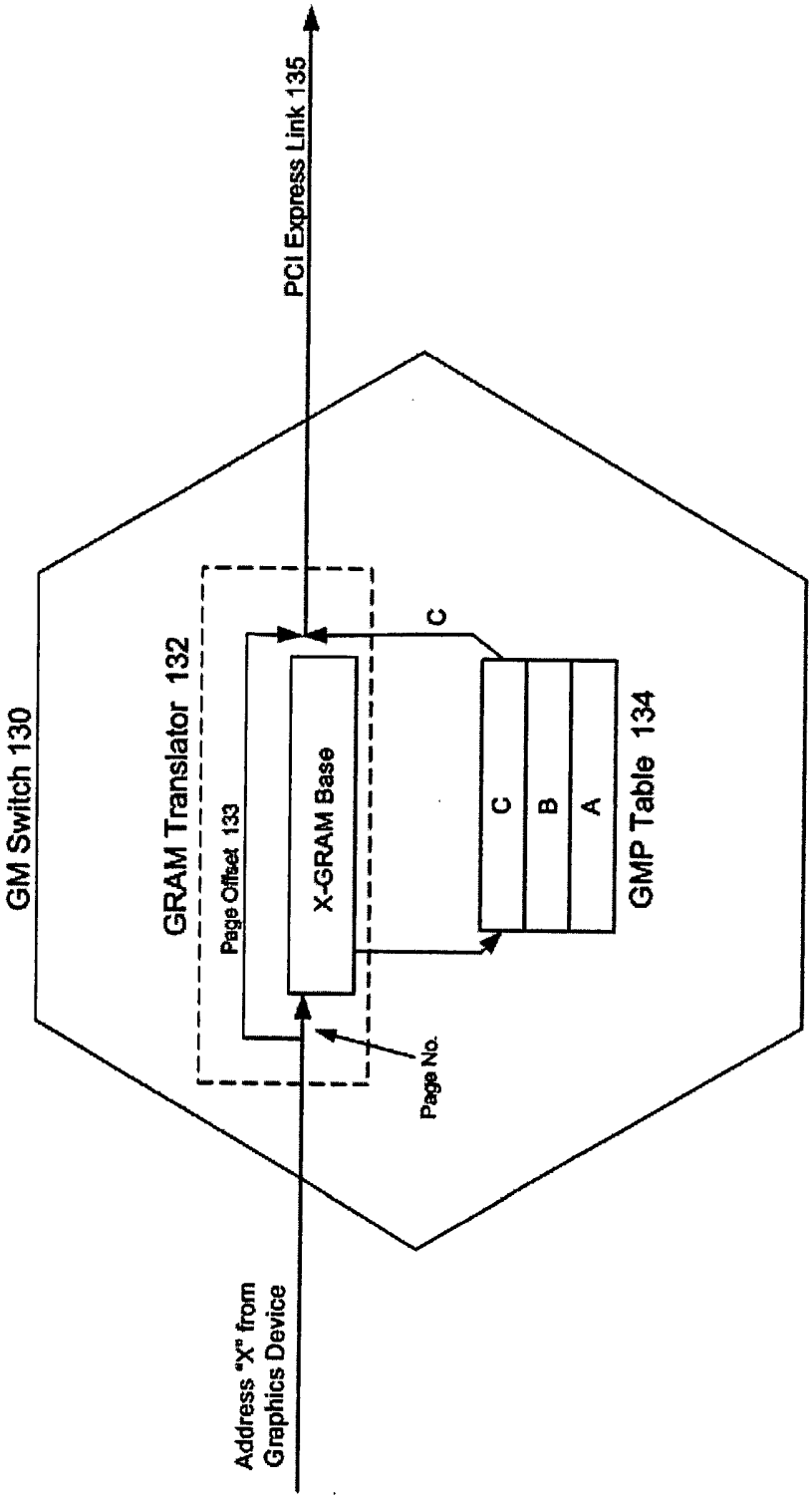
PCI Express Link 135
GM Switch 130
GRAM Translator 132
Page Offset 133
X-GRAM Base
C
C
B
A
GMP Table 134
Page No.
Address "X" from Graphics Device

Figure 4

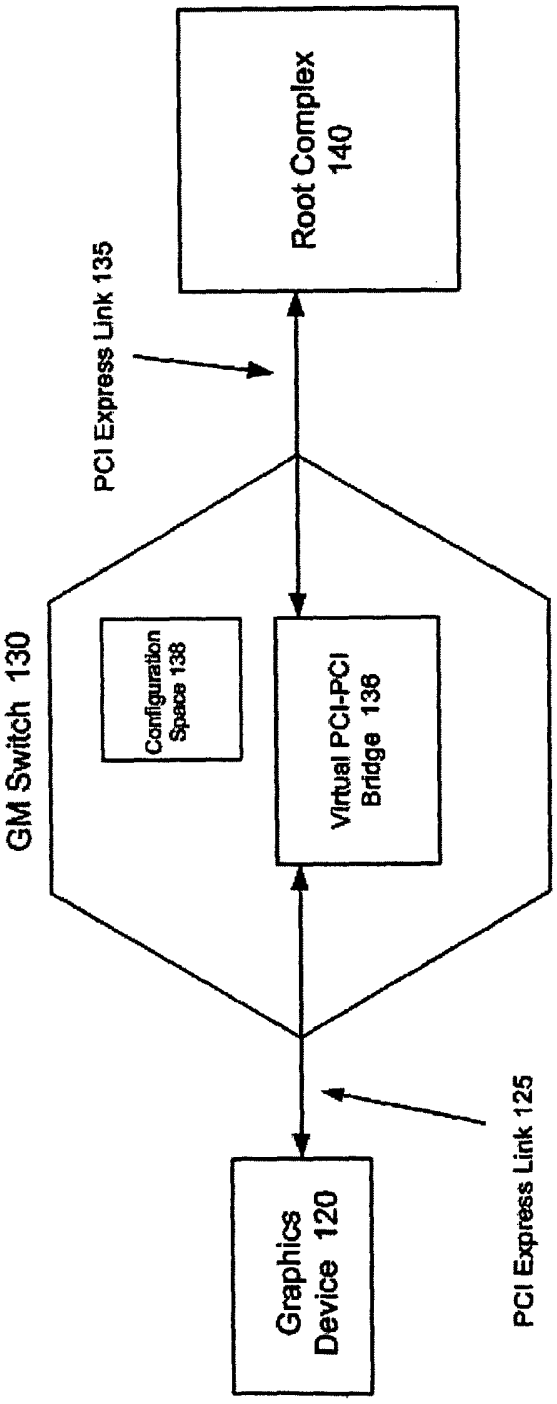
Root Complex
140
PCI Express Link 135
GM Switch 130
Configuration
Space 138
Virtual PCI-PCI
Bridge 136
Graphics
Device 120
PCI Express Link 125

Figure 5

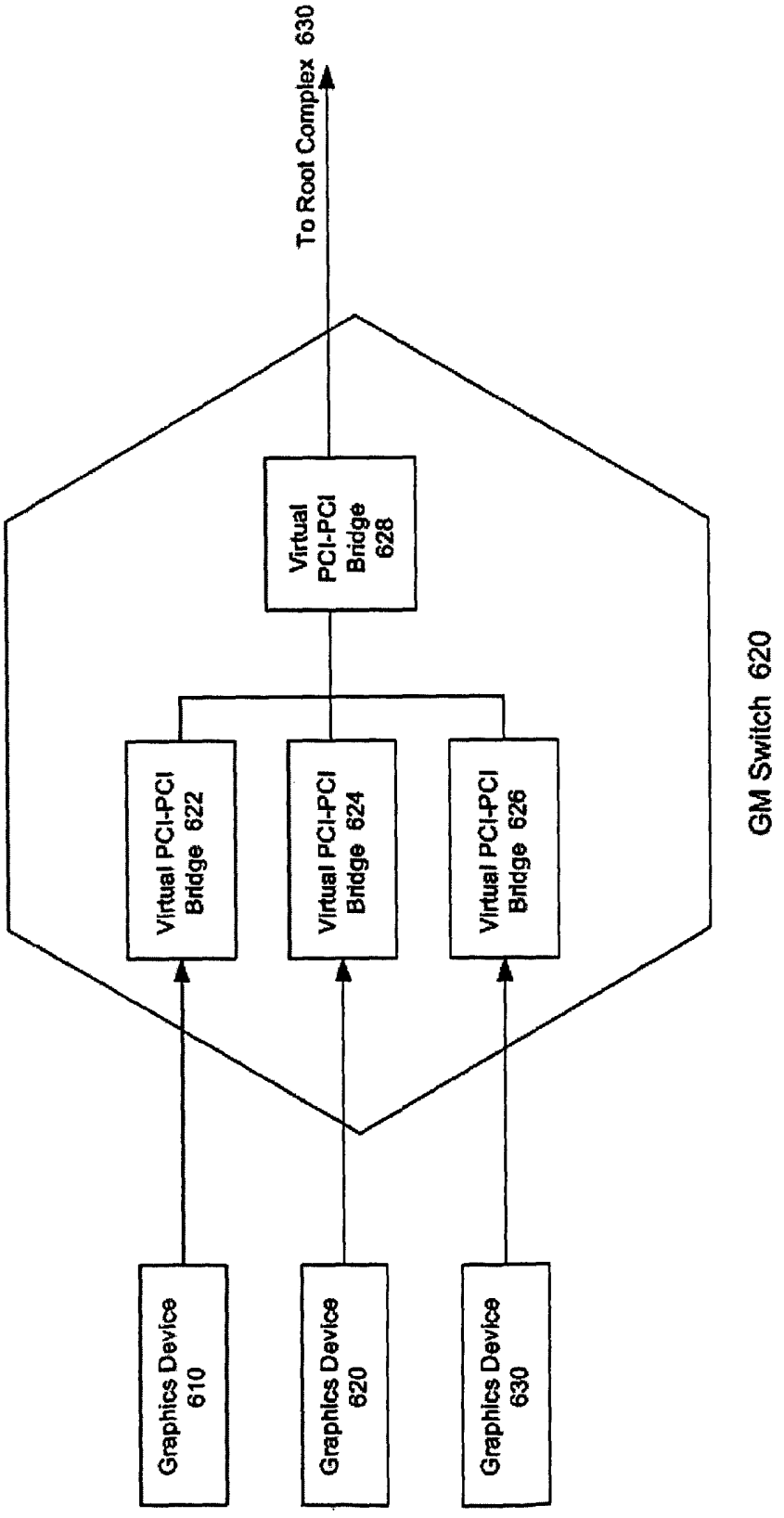
To Root Complex 630
Virtual PCI-PCI Bridge 628
Virtual PCI-PCI Bridge 622
Virtual PCI-PCI Bridge 624
Virtual PCI-PCI Bridge 626
GM Switch 620
Graphics Device 610
Graphics Device 620
Graphics Device 630

Figure 6

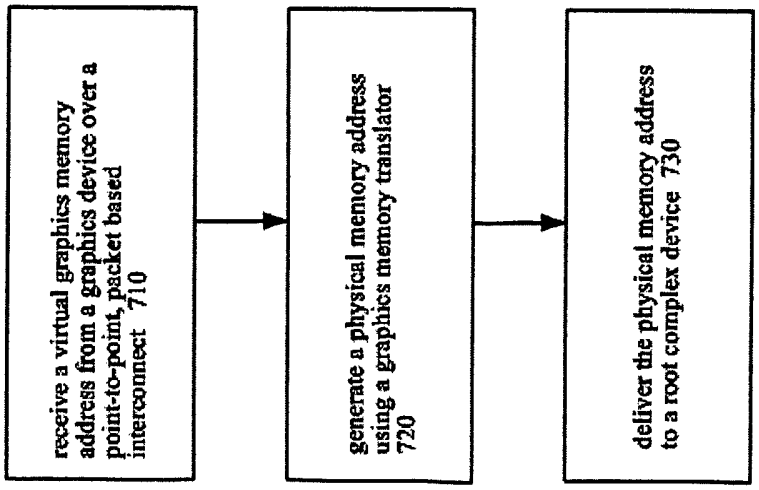
receive a virtual graphics memory address from a graphics device over a point-to-point, packet based interconnect 710
generate a physical memory address using a graphics memory translator 720
deliver the physical memory address to a root complex device 730

Figure 7