(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 102007479 B
(45)授权公告日 2015.01.14

(21)申请号 200980111272.6
(22)申请日 2009.03.27
(30)优先权数据
61/041,210 2008.03.31 US
(85)PCT国际申请进入国家阶段日
2010.09.28
(86)PCT国际申请的申请数据
PCT/US2009/038637 2009.03.27
(87)PCT国际申请的公布数据
WO2009/123933 EN 2009.10.08
(73)专利权人 先进微装置公司
地址 美国加利福尼亚州
(72)发明人 S·莫雷恩 M·S·格罗斯曼
W·F·克鲁格 B·艾特沙伊德
(74)专利代理机构 北京戈程知识产权代理有限公司 11314
代理人 程伟 王锦阳
(51)Int.Cl.
G06F 13/42(2006.01)
G06F 15/163(2006.01)

(56)对比文件
US 6956579 B1,2005.10.18, 说明书第4栏第59行-第9栏60行,图1.
CN 101149719 A,2008.03.26, 说明书第7页第1段,图8.
US 5745913 A,1998.04.28, 全文.
US 6956579 B1,2005.10.18, 说明书第4栏第59行-第9栏60行,图1.

审查员 孔昕

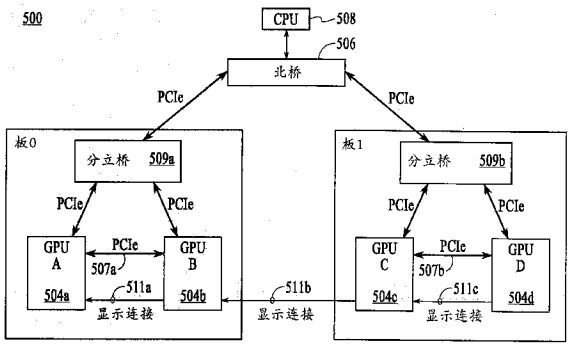
权利要求书2页 说明书13页 附图8页

(54)发明名称

点对点专用处理器结构和方法

(57)摘要

本发明描述点对点专用处理器结构和方法。实施例包括:多个专用处理器,通过主桥总线耦接至中央处理单元;直接总线,将所述各专用处理器直接耦接至其余专用处理器中的至少一个;以及存储控制器,耦接至所述多个专用处理器。其中,所述至少一存储控制器决定是否通过该主机总线或直接总线传输数据以及是否通过该主机总线或直接总线接收数据。



CN 102007479 B

1. 一种传输数据的系统，包括：

多个专用处理器，耦接至主桥总线；

直接总线，将所述多个专用处理器的每一个直接耦接至所述多个专用处理器中的至少另一个；

存储控制器，耦接至所述多个专用处理器中的至少一个，其中，所述至少一个存储控制器决定是否通过该主桥总线或直接总线传输数据以及是否通过该主桥总线或直接总线接收数据；以及

至少一个本地存储器，至少一个系统存储器，以及至少一个对等存储器，其中所述至少一个对等存储器驻留于专用处理器上，并且所述存储控制器经配置而基于所述各存储器的地址范围决定访问哪个存储器。

2. 如权利要求 1 所述的系统，其中，所述存储控制器驻留于所述多个专用处理器的至少一个上。

3. 如权利要求 1 所述的系统，其中，所述多个专用处理器包括图形处理单元 (GPU)。

4. 如权利要求 1 所述的系统，其中，所述主桥总线包括高速周边元件接口（PCIe®）总线。

5. 如权利要求 1 所述的系统，其中，所述直接总线包括高速周边元件接口（PCIe®）总线。

6. 如权利要求 1 所述的系统，其中，所述直接总线包括专有总线。

7. 如权利要求 1 所述的系统，其中，当地址落在该对等存储器范围内时，该存储控制器基于所述主桥总线和所述直接总线的每一个上的流量决定使用哪个端口。

8. 如权利要求 1 所述的系统，其中，当地址落在该对等存储器范围内时，该存储控制器基于所述主桥总线和直接总线目前是否满负荷而决定使用哪个端口。

9. 如权利要求 1 所述的系统，其中，当地址落在该对等存储器范围内时，该存储控制器的决定基于向所述主桥总线或直接总线分配专门地址。

10. 如权利要求 1 所述的系统，还包括至少一个本地存储器，至少一个系统存储器，以及至少一个对等存储器，其中，所述至少一个对等存储器驻留于专用处理器上，并且其中所述存储控制器经配置而基于所述各存储器的地址范围决定访问哪个存储器，并且其中该存储控制器还基于所述主桥总线和直接总线的每一个上的流量执行动态负载平衡。

11. 一种在多处理器系统中传输数据的方法，包括：

决定是否通过由主机处理器和多个专用处理器共享的主桥总线或者通过该多个专用处理器之间的直接总线传输或接收数据，其中，所做决定取决于传输的存储地址范围，其中，所述地址范围指定驻留于专用处理器上的对等存储器、该主桥总线上的系统存储器以及该专用处理器之间耦接的本地存储器中的其中之一。

12. 如权利要求 11 所述的方法，其中，落在指定所述对等存储器的范围内的专门地址被分配至所述主桥总线或直接总线的其中之一。

13. 如权利要求 11 所述的方法，其中，当地址落在指定该对等存储器的范围内时，所做决定基于所述各主桥总线和直接总线的每一个上的流量。

14. 如权利要求 13 所述的方法，还包括执行动态负载平衡以基于历史流量数据调节所

述主桥总线和直接总线的每一个上的百分比流量。

15. 如权利要求 11 所述的方法，还包括向该主机处理器分配存储器的图圈并分配所述多个专用处理器的每一个上的各自流量，其中，该存储器的图圈应用于该对等存储器、系统存储器和本地存储器，并且该图圈决定各个处理器如何查看存储器。

16. 如权利要求 15 所述的方法，还包括执行图圈的压缩，该执行图圈的压缩包括使用偏移寄存器通过改变偏移而使所述对等存储器的全部都是可见的。

17. 如权利要求 11 所述的方法，其中，该方法是通过可用于创建专用集成电路 (ASIC) 的硬件描述语言指令来执行。

18. 一种在多处理器系统中传输数据的方法，包括 ：

决定是否通过由主机处理器与多个专用处理器共享的主桥总线或者通过该多个专用处理器之间的直接总线传输或接收数据，其中，所做决定取决于传输的存储地址范围 ；以及

通过所述直接总线在该多个专用处理器之间直接传输数据，其中，所述专用处理器包括具有框型缓冲图圈的图形处理单元，并且直接传输状态是包括提供经由该直接总线在该框型缓冲图圈之外的访问。

点对点专用处理器结构和方法

技术领域

[0001] 本发明涉及计算机及其他数字系统中的数据传输领域。

[0002] 相关申请

[0003] 本申请要求 2008 年 3 月 31 日提交的美国临时专利申请号 61/041,210 的优先权，其全部内容通过参考整体纳入本申请。

背景技术

[0004] 随着计算机以及其他数字系统变得更加复杂而强大，用以增强系统元件之间数据传输的方法和硬件也在不断发展。要传输的数据包括代表数据的信号、指令或任意其他信号。数据传输的速度和效率在运行数据密集型应用例如图形应用的系统中极为重要。在典型的系统中，图形处理功能是作为中央处理单元 (central processing unit ;CPU) 功能的一部分来提供的，或者由独立的专用处理器例如图形处理单元 (graphics processing unit ;GPU) 提供，该专用处理器与 CPU 通信并辅助处理视频游戏等应用的图形数据。系统中可包括一个或多个 GPU。在传统的多 GPU 系统中，桥接主机接口 (例如高速周边元件接口 (peripheral component interface express ;PCIe)) 必须与主机流量共享带宽。图 1 为现有系统 100 的方块图，该系统 100 包括根处理器 108(例如 CPU)、桥 106 以及两个端点 EP0 104a 和 EP1 104b(例如 GPU0 和 GPU1)。各端点 104 分别耦接至存储器元件 102a 和 102b。

[0005] 为了充分利用系统中存在的多个 GPU，该多个 GPU 共享处理任务，从而需要在其之间传输数据。在多 GPU 系统中实现性能提升的挑战在于潜在有大量数据需要在这些 GPU 之间来回传输。例如，若应用要求绘制背景或其他纹理，而该绘制需要花费大量时间，则宜将该背景或纹理复制到另一 GPU，以使其能够用于下一帧或同一帧的另一部分。在系统 100 中，在 GPU104 之间进行数据传输的唯一路径是分别通过路径 105a 和 105b 到达桥 106。抵达根 108 的此路径必然承载有自各 GPU104 至根 108 的流量以及两个 GPU104 之间的流量。这样，由于物理线路共享而可形成瓶颈。

附图说明

[0006] 图 1 显示现有技术中多处理器系统的方块示意图。

[0007] 图 2 显示依据一实施例加入点对点输入输出路径的多处理器系统的部分方块示意图。

[0008] 图 3 显示依据一实施例加入点对点输入输出路径的多处理器系统的详细方块示意图。

[0009] 图 4 显示依据一实施例由该输入输出系统执行的方法流程示意图。

[0010] 图 5 显示依据一实施例在不对称输入输出系统中传输相位信息的方法流程图。

[0011] 图 6 显示依据一实施例在不对称输入输出系统中向主机设备传输客户端相位信息的时序图。

具体实施方式

[0012] 这里描述点对点 (peer-to-peer) 专用处理器结构和方法的实施例。实施例包括通过主桥总线耦接至至少一个中央处理单元的多个专用处理器。直接总线将所述各专用处理器直接耦接至其余专用处理器中的至少一个。存储控制器耦接至该多个专用处理器并决定是否通过该主机总线或直接总线传输数据

[0013] 图 2 显示依据一实施例的多处理器系统 200 的部分方块示意图。系统 200 包括耦接至桥 026 的根处理器 208。桥 206 分别通过总线 205a 和 205b 耦接至总线端点 EP0 204a 和 EP1 204b。各端点 204 分别耦接至存储器设备 202a、202b。

[0014] 一实施例中，根处理器 208 为 CPU，各端点 204 为 GPU，但并不以此为限。如下文所述，实施例包括的端点 204 分别为具有侧端口 (side port) 的 GPU，所述侧端口类似 PCIe 端口。PCIe 的寻址特征和各种负载平衡特征应用于系统 200 的结构中以提升性能。在典型的结构中，自一 GPU 向另一 GPU 发出的“读”操作极其昂贵。“读”操作中，发出请求，并于一段时间后接收所请求的数据。相较包括“读”操作在内的操作的典型性能，实施例作出了改进。

[0015] 图 3 显示依据一实施例的多处理器系统 300 的方块示意图。系统 300 与系统 200 类似，但所示更为详细。CPU308 为系统 300 的根处理器。CPU308 通过桥 306 耦接至系统存储器 309。CPU308 还通过桥 306 经由主桥总线 305a 耦接至 GPU0304a，并经由主桥总线 305b 耦接至 GPU1304b。

[0016] GPU304 通过各自的侧端口经由直接总线 307 彼此耦接。GPU304 还分别通过各自独立的端口耦接至各自的本地存储器 313a 和 313b。GPU0304a 包括多个客户端 311a。客户端 311a 与存储控制器 303a 通信。存储控制器 303a 控制客户端 311 对本地存储器 313a（例如视频存储器）的访问，以及客户端 311 对总线 305a 的访问。更具体地说，存储控制器 303a 决定哪个客户端 311 与本地存储器 313a 通讯，哪个客户端 311 与总线 305a 通信，以及何时通信。

[0017] 类似地，GPU1304b 包括耦接至存储控制器 303b 的多个客户端 311b。存储控制器 303b 与本地存储器 313b 通信。GPU0304a 与 GPU1304b 经由直接总线 307 彼此耦接。直接总线 307 是位于 GPU304 的新的侧端口之间的点对点路径。

[0018] 一实施例中，客户端 311 对不同总线的访问由地址范围决定。对存储器 309 和 313 的访问基于所述地址范围。一实施例中，存储地址空间分为三部分：系统存储器 309 部分；本地存储器 313a 部分；以及本地存储器 313b 部分。从 GPU304b 的角度看，存储器 313a 为对等存储器，反之亦然。对等地址涉及另一 GPU 上的存储器。对本地存储器的访问由桥 306 以一种典型方式处理。一实施例中，当地址落在对等地址范围内时，使用两条可行路径（直接总线 307 和主桥总线 305）。经由一条路径或另一条路径发送的数据量取决于不同实施例中的不同条件。例如，依据一模式，数据被平均分配在两条路径中：50％的数据由总线 307 发送，50％的数据由总线 305 发送。该模式提升了性能，而这在传统结构中是不可能实现。不过，本领域的技术人员应了解，总线 305 仍在被其他流量（如上所述的正在传输的数据之外）共享。为此，还可在该模式中采用动态负载平衡以进一步提升性能。例如，所述直接路径（利用直接路径 307）可为优选路径，但当该路径满负荷时，则使用其他路径（例如利用路径 305a 和 305b）。不过，考虑以下情形。假设一个“写”数据序列（“W”表示“写”，WA 表

示“写入存储地址 / 位置 A”）：

[0019] 1、WA 至 GPU0，经由 305 ；（如图 3 所示该数据＝ 4）

[0020] 2、WB 至 GPU0 ；

[0021] 3、WC 至 GPU0 ；

[0022] 4、WA 至 GPU0，经由 307 ；（如图 3 所示该数据＝ 6）

[0023] 这里欲写入的数据为“6”。如上所述，首先发生写入“4”，但经由主桥总线 305a 和 305b。写入“6”在后来发生，但经由直接总线 307（较快的途径）。“6”先写入，但“4”随后到达，覆盖了“6”。如果预料到这样的问题，则一种避免该问题的模式是基于地址的路由。可将地址进行划分，以便将针对相同地址的数据发送在相同路径上。例如，偶双字地址可经由直接路径 307 发送，而奇双字地址可经由主桥路径 305a 和 305b 发送。或者，八个地址中的七个（不以二进制 000 结尾的地址）可经由直接路径 307 发送，而其他则可经由主桥路径 305a 和 305b 发送。

[0024] 依据另一种模式，全部数据都经由一条路径或另一条路径发送。由于有一条路径未被使用，该模式并不是最理想的。不过可允许客户端 311a 和 311b 中的每一个都具有自己的规则。这样就可具有多个图形硬件功能，其区别使用所述不同路径，并实现路径的充分利用。

[0025] 在下面详细描述的实施例中讨论“写”操作。亦即，GPU0 304a 向 GPU1 304b 传送数据，但不会从 GPU1 304b 获取数据。但是，在其他实施例中，可依据这里所述的相同方法在任一方向上实施读操作和写操作。

[0026] 图 4 显示依据一实施例的两 GPU 系统 400 的元件方块示意图。图 4 包括依据一实施例的总体图形子系统互联（图中未显示用于将可显示像素传送至主板或显示设备的显示连接 (display link)）。系统 400 包括 GPU0 404a 和 GPU1 404b。GPU404 通过直接总线 407 和各自的侧端口彼此直接耦接。将标记为“RC”的方块设为根联合体核心 (root complex core)。将标记为“EP”的方块设为端点。需要注意的是，RC 和 EP 是 PCI Express(PCIe) 端口的常用技术。为了正常通信，RC 必须与 EP 对话而不是另一个 RC。在一个利用普通端口设计实现点对点的实施例中，包括有绑定逻辑 (strapping logic)，以使 404a 中的“RC”或 404b 中的“RC”作为真正的 RC，而另一个作为 EP。各 GPU404 包括存储控制器 (MC) 和主机数据路径方块 (HDP)。如图所示，通信路径 421 为读 / 写 (R/W)+d 请求路径，这里的“W+d”表示“写请求加数据”。“CPL 数据”箭头 413 显示数据读取完成的路径。箭头 423 表示 W+d 请求路径。图中未显示主机 PCIe 与其他 GPU 内部方块，例如内部寄存器、只读存储器 (ROM)、音频和视频图形阵列 / 适配器 (VGA) 的连接。

[0027] GPU404 进一步经由各自的主桥总线 405a 和 405b 通过交叉矩阵 (crossbar)417 耦接至桥 406（本实施例中为北桥）。CPU408 耦接至桥 406。

[0028] 图 5 显示依据一实施例的 4GPU 系统的方块示意图。在一实施例中，两个图形处理单元 GPUA404a 和 GPUB404b 占据一块板（板 0），另外两个图形处理单元 GPUC504c 和 GPUD504d 占据另一块板（板 1）。GPU504a 和 GPU504b 通过直接总线 507a 和显示连接 (display link)511a 耦接。GPU504a 和 GPU504b 分别通过各自的 PICe 总线耦接至分立桥 509a。

[0029] GPU504c 和 GPU504d 通过直接总线 507b 和显示连接 511c 耦接。GPU504c 和 GPU504d

分别通过各自的 PICe 总线耦接至分立桥 509b。

[0030] 板 0 和板 1 由显示连接 511b 耦接。板 0 和板 1 进一步经由自分立桥 509a 至桥 506 的 PCIe 总线以及自分立桥 509b 至桥 506 的另一条 PCIe 总线通过桥 506 耦接。CPU508 耦接至桥 506。可在系统 500 中的任意一对 GPU 之间同时进行点对点传输。一实施例中，如果 GPU 对分别为 AB 和 CD，则各传输的传输速度最高约为 12.2GB/s。如果 GPU 对分别为 AC 和 BD 或者是 AD 和 BC 并且传输方向相反，则各传输的传输速度约为 6.1GB/s。如果 GPU 对分别为 AC 和 BD 或者是 AD 和 BC 并且传输方向相同，则各传输的传输速度约为 3GB/s。

[0031] 图 6 显示依据一实施例的侧端口子系统的方块示意图。下面的标记部分参照图 6 提供不同实施例的实施细节，但本发明并不受限于所述特定细节。

[0032] 主机 PCIe 端点

[0033] 一实施例中，主机 PCIe 接口由总线接口核心逻辑 (bus interface core logic ; BIF)、模拟物理层 (analog PHY) 和封装器 (wrapper) 组成。该接口被设为端点 (EP)。该总线以每引脚 5Gbps 的标准 PCI Express2.0 规范 (PCIe2) 速率运行，或者以 16 通道接口的理论最大速率 8GB/s 运行。该总线具有 128 位 credit/debit 主接口 (master interface) 用于出去的存储器写入 (至主机或对等体)，以及 128 位 credit/debit 从接口 (slave interface)，用于进来的存储器写入 (来自主机或对等体)。该总线具有连接只读存储器 (ROM)、内部寄存器总线、音频和视频图形阵列 (VGA) 的 32 位接口。该总线将主机请求通过其从接口 (标记为 S) 路由至所述多个 GPU 目标。

[0034] 请求接口具有相伴的完成接口。所述 BIF 利用一组地址孔径 (address aperture) 引导流量，该组地址孔径分别是 : 帧缓冲区的地址孔径 ; 寄存器地址孔径 ; 以及只读存储器地址孔径。它针对配置寄存器和绑定 (strap) 使用 32 位接口。它还具有与时钟发生器 (clock generator ;CG) 的接口，以处理高级配置与电源接口 (advanced configuration and power interface ;ACPI) 电源状态管理、时钟、重置和扫描输入，以及调试总线菊花链接口 (debug bus daisy chain interface)。它还具有与所述主机数据路径方块 (HDP) 的接口，用以帧缓冲区读写。

[0035] 侧端口 RC

[0036] 用于侧端口 (XSP) 的 PCI Express 逻辑被配置为 RC。RC 和 EP 操作模式在配置和功能上差别很小。XSP 总线以每个引脚 0.5Gbps 的标准 PCIe2 规范速率运行，或者以 16 通道接口的理论最大速率 8GB/s 运行。它具有 128 数据位 credit/debit 主接口，用于出去的至对等 GPU 的存储控制器 (MC) 存储空间写入请求，以及 128 位 credit/debit 从接口，用于进入的来自对等 GPU 的存储器写入。这些接口是在主机 BIF 的例子中由 HDP 使用的接口。它针对配置寄存器和绑定使用 32 位接口。它独立于所述主机接口而保留自己的链接电源状态。与所述主机接口相同，通道数目可由软件编程为 8 或 16。由于在专用集成电路 (application specific integrated circuit ;ASIC) 上的两 PCIe 物理层以相同的速率运行，因此可使用一对参考时钟 (REFCLK) 输入引脚，缓冲 REFCLK 在芯片内分布至所述两物理层。

[0037] 所述 XSP 例不需要任何其他读 / 写接口或由主机 BIF 支持的中断处理。该侧端口不需要由 ASIC 上电激活，因此大多数情况下它不使用基于引脚的绑定。可利用 SRBM 总线通过软件寄存器写入而独立于所述主机接口编程该侧端口的配置。

[0038] 所述 XSP 不需要支持寄存器读或写、I/O 空间或配置空间周期。在该设备上仅需可操作的单个 PCIe 功能。在仅发出缓写（根本没有读）并使用全局同步方案 (global synchronization scheme) 的实施例中，不需要完成信息 (completion message)。

[0039] 存储控制器中心 (MC hub) 将写请求路由至所述 XSP。与主机 BIF 请求一样，XSP 请求使用相同的地址。

[0040] 主机数据路径方块

[0041] 一实施例中，该主机数据路径方块 (HDP) 为标准方块，其为 BusIP 的一部分。该 HDP 处理由所述主机接口执行的帧缓冲区读和写。在多 GPU 系统中，帧缓冲区读和写可以来自主机或者来自另一 GPU 发起的点对点写入。主机访问可使用多个 HDP 拼接孔径 (tiling aperture) 中的其中一个，而点对点访问则可使用或不使用（对等体可由请求者 ID 识别）。该 HDP 与该主机 BIF 的从端口连接并作为读写客户端与 MC 连接。该 BIF 和 HDP 相互配合以保持读至写的一致性；在写入之前无法读取。对于点对点写入同步，向保留存储器地址的特定写入由该 HDP 识别，并针对向本地存储器的特定“写入相位”触发 MC 读取以及在前写入冲洗，接着写回至发送客户端的寄存器空间中的相位专用邮箱。

[0042] 自总线进来的读和写由该 BIF 中的专门孔径路由至该 HDP。该孔径对应 32MB 或更多本地内存。可选择性地将该 HDP 处理的地址当作虚拟地址并在 HDP 专门的页表中查找。

[0043] 为最佳利用各 GPU 的可用 PCIe 孔径并支持写相位，点对点写入地址由发送 GPU 的 MC 编码为相位和偏移，并可穿插自动每相基地址寄存器 (base address register；BAR) 更新。该 HDP 使用来自该 BIF 的请求者 ID 加上传输位的结合以决定使用哪个写入相位和动态地址孔径。

[0044] 存储控制器 (MC) 方块

[0045] 该 MC 具有两个连接所述侧端口的接口：针对来自 MC 客户端的写请求的 128 位 BIF 主机接口；以及针对自其他 GPU 进来的向本地存储器的写入的 128 位从接口。该 MC 还包括寻址逻辑，以将主请求通过所述 XSP 路由至对等存储器。XDP 方块处理接口功能。它察看从请求并进行地址偏移或视频存储器 (video memory；VM) 查找，它维持写完成计数器，并提供混合写入 (write combining)(32 字节 BIF 请求至 64 字节 MC 请求）。

[0046] 来自或到达所述侧端口的请求可为 16 字节（具有掩蔽 (masking)）至 64 字节。为达最佳性能，请求应当为 64 字节。

[0047] 所述侧端口 BIF 接口传递 GART 翻译的 40 位系统地址，不论全图形虚拟内存激活与否。所述 GART 与系统背景域 (context domain) 相同，执行 36 位 GPU 设备地址的翻译。

[0048] 所述 MC 维持写完成状态，以允许多 GPU 保证的复制操作的冲洗。类似于所述 HDP，由 XDP 识别的专门地址用于触发特定写相位的写完成的核查，并生成至所述发送客户端的寄存器空间中的邮箱的写回。

[0049] 直接存储器访问引擎 (DRMDMA)

[0050] 所述具有侧端口的 GPU 包括直接存储器访问 (direct memory access；DMA) 引擎，其可执行有效的存储器主机块传输 (block transfer；BLT)。可对所述 DRMDMA 进行编程，以在本地存储器、主机存储器以及另一 GPU 的本地存储器中任意两者之间传输。随着地址处理（如后文所述）的增强，所述 DRMDMA 还可利用所述 XSP 和 PCIe 桥移动存储块。一实施例的标准传输能力包括：

[0051] . 主机至本地（主机 BLT 或复制）- 每时钟峰值速率 16 字节

[0052] . 以常量填充本地 - 每时钟峰值速率 32 字节

[0053] . 本地至主机（复制）- 每时钟峰值速率 16 字节

[0054] . 本地至远端 GPU（复制）- 每时钟峰值速率 32 字节至相邻 GPU，每时钟峰值速率 16 字节至不相邻 GPU

[0055] 所述 DRMDMA 可支持每时钟 32 字节的读和 32 字节的写，其与两个 16 字节 PCIe 写接口相匹配。

[0056] 主机端口相较侧端口负载平衡

[0057] 添加侧端口允许至相邻 GPU 的较大带宽，这是 2GPU 和 4GPU 使用模型需要大部分带宽的地方。所述主机端口也具有许多可用带宽并且仍然可用于与相邻 GPU 通信。

[0058] 负载平衡最简单的方法包括基于不同客户端的固定分配静态分配用于传输数据的端口 - 例如，DRMDMA 分配给侧端口，颜色缓冲块分配给主机端口。

[0059] 第二种负载平衡方法包括基于侧端口的繁忙情况将任意客户端的流量动态路由至所述两个端口中的任意一个。见下文的“任意”负载平衡描述。

[0060] 使用上述第二种方法，通过两条不同的路径向相同的像素地址写入有可能出现竞争情况。当这些写入属于两个不同的基元 (primitive) 时，则会违背像素顺序规则。较早基元的写入可在较后基元写入之后到达。选择基于地址的路由可解决此问题。一种后备方法是针对相关客户端使用单个路径。

[0061] 如果对等写入请求正在通过主机端口和侧端口向邻居发送，则可能会由于通过该两端口的不同延迟而丢失请求顺序。在某些情况下，主机端口甚至会停滞，因而需要采取措施以确保必要时保留请求顺序。

[0062] 所述 MC 的唯一顺序保证是自相同客户端向相同地址的写入不会重新排序。因此，这也是对等写入唯一的写后写 (write after write ;WAW) 要求。依据实施例有数种方式来确保该要求。所述 MC 处理针对本地存储器以及主机和对等体的读 / 写。缺省的 MC 顺序规则如这里所述。额外的路由规则和总线本身提供额外的顺序保证。全负载平衡仅适用于进入相邻 GPU 的请求。这由孔径核查决定，非相邻请求将全部去往所述主机端口。

[0063] 负载平衡 :固定映射至端口（“固定”）

[0064] 确保向相同地址的两个写入不会失序的方式是通过地址核查负载平衡。执行地址核查以确保给定地址总是使用相同的端口。例如，对于特定粒度 (granularity)，“偶”地址总是使用主机端口，而“奇”地址总是使用所述侧端口。这样，针对相同地址的两个请求不会在通过所述端口时失序。

[0065] 所使用的地址核查映射可就查看哪些地址位以及使用什么阈值进行平衡进行编程。由于较大的粒度可能意味着写突发集中通过一个端口或另一个端口，而太大的粒度可能导致端口不足而大突发被送往一个端口或另一个端口，因此所使用的地址位设定了平衡的粒度。所使用的阈值允许主机端口和侧端口之间较好的平衡比例。例如，如果使用三个地址位，主侧端口的比例可为 :0 ： 8,1 ： 7,2 ： 6,...,8 ： 0,这取决于所使用的阈值。将全部请求送往所述主机端口或送往所述侧端口是该固定映射的退化情形。

[0066] 通过此类固定负载平衡，如果一个端口（主机端口）繁忙或停滞，后备可调节发送至另一端口（侧端口）的带宽。此外，特定的写模式容易造成理想的平衡比例被破坏（例

如跨越写(striding write)总是映射至一个端口,而空置另一个端口)。因此,由于该负载平衡方法为静态、软件编程而不是针对可用带宽作出反应,因而存在缺陷。下文讨论动态负载平衡。

[0067] 负载平衡:无约束或任意顺序,不保证写后写(“任意”)

[0068] “任意”负载平衡基于充分或几乎充分的条件在端口之间切换。由于预期的较高可用性,该侧端口应当最终被优先。应当提供各端口的剩余信用数(outstanding credit)的控制,以调节平衡。一些客户端不需要WAW一致性保证。例如,由于不会在同一复制中两次传输相同的地址,因此可通过复制之间的冲洗保证一致性。这样将允许源和目标全速进行,只需要在潜在重叠传输之间进行正常的写完成同步。将该流量与“任意”类型的流量混合还可缓和它们各自的弱点,潜在地使这些写入填满任意间隙。所述侧端口将尽可能多地使用,主机端口仅在侧端口满负荷时使用。

[0069] 负载平衡的任意和/或全部实现方法可并行使用于独立的传输流。软件可配置各相位使用的负载平衡类型。

[0070] 如果“任意”流量与“固定”流量混合,则该“任意”流量应当有助于弥补“固定”选择的弱点所造成的任何缺陷。硬件允许“任意”类型的请求绕过停滞或等待的“固定”请求。

[0071] “固定”和“任意”仅是两种负载平衡模式,还可有其他多种实施例。

[0072] 负载平衡:可能的使用情景

[0073] 下面为可能的使用模型,但实施例不限于此。

[0074] 1)对等写入为DMA:DRMDMA使用1-2阶段将数据复制到对等GPU;

[0075] 1a)可使用“任意”类型的负载平衡进行传输,但其可能要求有重叠危险的任意DMA之间的额外同步。软件可核查决定是否发生重叠。

[0076] 1b)两DMA引擎的替代负载平衡选择是使一引擎仅使用主机端口,而另一引擎仅使用侧端口。或者,仅使用主机端口可包含向不相邻对等体的写入。由于不保证主机端口的可用带宽,因此软件难以最佳地划分复制工作,但如果两相位隐藏同步延迟,则可保持两端口满负荷而不需要任意潜在重叠复制之间的任何额外同步。如果相关资源不重叠,则硬件可允许仅使用主机端口的相位和仅使用侧端口的相位使用相同的相位号。

[0077] 2)WAW敏感客户端写入(使用或不使用复制)

[0078] 2a)类似上述1b,如果正在发送一个或多个对等写入,分别给这些对等写入分配专用路径“仅使用主机端口”或“仅使用对等”,以充分使用两个端口,并且在端口之间没有顺序依赖或交叉的背压路径(backpressure path)。

[0079] 2b)如果WAW敏感客户端是进行传输的唯一客户端,则只需使用“固定”类型,以保持端口充分利用,如果其他有问题,则“仅使用侧端口”为安全备份(和/或另一客户端可在部分时间使用“仅使用主机端口”)。

[0080] 寻址

[0081] GPU的主机接口接收针对寄存器(映射在存储空间中)和本地存储器(帧缓冲区)以及其他块的总线请求。BIF中的一组孔径(或地址范围)将PCIe空间地址(64位)映射到该GPU中的特定目标。本地存储器读和写通过所述HDP到达所述MC。

[0082] GPU发出的存储器请求使用地址,该地址是或者不是虚拟地址。对于Windows XP或者Vista图形驱动器模型,GPU“设备”地址形成孔径区域化平面地址空间,其包含系统存

储器（可进一步通过 GART 映射）或本地存储器。对于未来的驱动器模式，GPU 地址被处理为页面映射虚拟地址，并且对于系统存储器和本地存储器，所查找的物理地址是独立的。

[0083] 系统存储空间

[0084] 从 GPU 角度看，系统存储器通常指附着至 CPU 的动态随机访问存储器 (DRAM)，但实际中系统存储器包括属于任意可访问的存储映射设备的任意存储空间，尤其是其它 GPU 存储器。取决于系统结构，分立桥和芯片组桥（例如北桥）联合处理系统存储器访问的映射和路由。

[0085] 从 CPU 角度看，物理地址空间 (32 位或 64 位）划分成附着至该 CPU 的真实系统存储器以及“其他”存储器，后者为存储映射设备上（例如 PCIe 总线上）的任意存储器。此划分通过输入输出 (I/O) 系统的最顶端节点中的一个基址 - 限制配置寄存器对 (base-limit configuration register pair) 完成。还有另一对用于寄存器空间。在下方的二进制树的每一层，总线地址范围被分为两部分，每个下游分支都具有一个基址 - 限制对。GPU 的孔径尺寸由板绑定 (board strap) 设置。现在，该尺寸支持 32 位和 64 位操作系统，并且不论多少 GPU 试图插入该系统都支持。

[0086] 从 GPU 角度看，系统地址通过 GART（系统页表）或直接映射至总线地址空间，将未生成的最重要位 (most significant bit ;MSB) 设为 0。各上游 PCIe 桥结点查对基址 - 限制对，以路由至更上游或对等连接。

[0087] 图 7 显示具有两 GPU 的系统存储空间的高阶示意图，显示从各实体角度所看到的内容。图 7A 为 CPU 视图，图 7B 为 GPU0 视图，图 7C 为 GPU1 视图。

[0088] 视频存储器禁用

[0089] GPU 提供给 MC 的 36 位地址经历如下操作序列：

[0090] 1、查对本地存储器的图圈。如果该地址落在该范围内，则减去基址，并将基址用作物理本地存储器地址。

[0091] 2、如果在本地存储器的图圈之外，则查对侧端口存储器的图圈。这通常对应相邻 GPU 所使用的地址范围。如果该地址落在该范围内，则将该地址在不作修改的情况下依据负载用在所述侧端口或主机接口上。如果该地址用于侧端口，则远端 GPU 客户端查对自身的孔径，并且此查对应当在本地存储器的图圈中通过。如果该地址用于主机接口，则所述桥和北桥中的孔径查对将把该请求路由至所述远端 GPU。该远端主机接口将这些请求传递至其 HDP 或 SRBM。

[0092] 3、如果该地址落在上述两孔径之外，则其为系统地址。如果该地址落在非探测 (non-snooped) 孔径内，则将其在不作修改的情况下传递至所述主机接口。否则，在所述系统 GART（例如系统页表）中查找该地址，然后将其传递至所述主机接口。

[0093] 视频存储器启用

[0094] 将 GPU 地址作为虚拟地址并由页表在 VM 中译为系统物理地址或本地视频存储器地址。传统上，系统物理地址为 40 位，请求的目标总是主机接口（通过 GART 之后 - 实际上是系统页表）。不过，为支持针对点对电传输的主机接口和侧端口的使用，实施例也允许系统物理地址通过该侧端口的路由。下面为依据一实施例的序列。

[0095] 1、在适当的页表中查找 GPU 客户端（例如 DMA 引擎）所作的请求。

[0096] 2、如果页表项 (page table entry ;PTE) 中的“S”位（针对系统空间）未设，则访

问该请求 GPU 的本地存储器。

[0097] 3、如果“S”位已设，则执行侧端口孔径 (side port aperture ;SPA) 地址范围查对。

[0098] 4、如果该地址在 SPA 之外，则该请求去往该主机接口 (GART 查找后)。

[0099] 5、如果该地址在 SPA 以内，则做出负载平衡决定，并将该请求路由至所述侧端口或主机接口。

[0100] 6、如果该请求通过该主机接口路由，则其为标准点对点请求。

[0101] 7、如果该请求通过该侧端口路由，则将该系统地址传递至远端 GPU，在该远端 GPU 的 HDP 空间页表中查找该地址，将其 (几乎总是) 映射至本地物理存储地址。

[0102] 同步

[0103] 使用多 GPU 渲染要求绘制与数据移动步骤之间的同步，以确保数据一致性。独立的命令流执行所述绘制和复制。交替帧渲染 (alternate frame rendering ;AFR) 的典型序列为

[0104] 1、GPU0 上的 3D 引擎在动态或持续的纹理表面绘制

[0105] 2、3D 引擎冲洗并发出完成信号 (发出旗语信号指令)

[0106] 3、在 GPU0 上的复制引擎等待旗语，接着将结果复制到 GPU1

[0107] 4、复制引擎冲洗并发出完成信号

[0108] 5、GPU1 上的 3D 引擎等待旗语，然后使用已渲染纹理绘制

[0109] 所述 BIF 和 MC(以及北桥和 CPU 存储器系统) 对于读操作和写操作的顺序分别具有自身的一致性规则。在无软件干预的情况下在多 GPU 之间保持端到端请求的目标一致性，需要同步机制，该机制考虑了所有这些规则 (或者不取决于这些规则)。

[0110] 多 GPU 的同步需求类似于单个 GPU 需要在两 DMA 之间进行表面数据同步的额外步骤。在此种 DMA 情况下以及广播写功能的情况下，需要向一 GPU 上的客户端保证另一 GPU 上已完成写。该 MC 和旗语块有助于解决此任务。

[0111] 图圈的压缩

[0112] 设计有效 GPU 系统的一个挑战是随着 GPU 的数目和各 GPU 的本地内存量的增加，将所有内容都适配至所述系统存储空间的能力被折衷，并且设备图圈不得不缩小尺寸。如果可映射至系统空间的 GPU 本地内存量有限，未映射的内存的使用则被限制在本地 GPU 上。关于可映射 CPU 的 GPU 空间的限制范围没有过多关注，后备方法是将表面复制到系统存储器而不是对本地视频存储器的锁定访问。

[0113] 依据一实施例的解决方案是通过加入偏移寄存器 (P2P BAR)，使对等体可访问本地内存的其余部分。当目标 BIF 看到来自源端的写请求时，在应用存储器的图圈为 HDP 产生 36 位设备地址后，加入偏移 (以 4KB 页粒度)。源端能看到的毗连本地虚拟内存空间数量不大于其先前的情况，但现在这是一进入全部本地内存的窗口，其可通过重写 P2P BAR 而改变。理论上来说，可随时移动 BAR，只要写入流内的变化是有序的。值得注意的是，多个偏移只要可由源端辨识 (例如写相位)，并不都需要同步或空间分离。

[0114] 图 8 显示具有偏移的存储空间示意图。图 8A 显示 GPU0 角度的系统内存空间视图，图 8B 是没有偏移的 GPU1LVM 映射，以及图 8C 是具有偏移的 GPU1 映射。

[0115] 写相位

[0116] 当单个 GPU 发送多组写入时，或是同时进行多组写入（例如 3D 管路和 DRMDMA 复制）或是来自相同客户端的背对背写入，最好保证写入完成。在后一情况中的目标是隐藏冲洗 / 同步操作的延迟。其次，最好允许多个源 GPU 写入单个目标 GPU 并针对各个源独立跟踪写入完成。

[0117] 依据一提供理想结果的实施例，（目标）BIF 具有 8 组基址 / 限制寄存器（相位范围），每组具有 3 位相位 ID。可由源 GPU 以各种方式写入该目标。例如：

[0118] . 多达 8 个 GPU 中的每一个可能保留一个相位范围专用。使单个目标 GPU 上的范围重叠将变得模糊（“交叉流”）。较好的方案是让相位由 PCIe 源 ID 决定，其消除了任何重叠限制。

[0119] . 多达 4 个 GPU 中的每一个可保留两个不重叠的相位范围。

[0120] . 两个 GPU 可使用全部 8 个相位范围（但实际上认为每个 GPU 源只有 2 个相位范围有用）

[0121] 一实施例中，单个 GPU 中有 7 个流量源，其可分配给不同相位

[0122] . 颜色缓冲块或深度缓冲块

[0123] . 着色输出块

[0124] . 指令处理器

[0125] .DRMDMA(2 个交替相位）

[0126] 因此，最终相位值应当为源 ID 和范围查对结果的结合，取决于系统中 GPU 的数目。

[0127] 写相位与小孔径的结合

[0128] 可同时允许这些特征（写相位和小孔径）存在。亦即，允许源 GPU 看到全部目标表面并且每个 GPU 具有多个源。一实施例中，每个相位使用 P2P BAR，从而在目标地址离开太远以至于不适合单个孔径的情况下，单个源 GPU 能够使用多个相位。不过，由于 PCIe 空间孔径不够大，不足以进行目标处的范围查对，因此选择 P2P BAR 可能存在问题。

[0129] 一个解决方案是复制发送侧的相位范围寄存器和比较，并使用比较结果，a) 以传输结果相位信息以及每相位 P2P Bar 的偏移；以及 b) 当请求超出该相位的当前范围时，更新 BAR。该 P2P 孔径尺寸预先通过独立的全局寄存器固定。BAR 移动的方式包括一些迟滞量，以最小化频率更新。

[0130] 解决方案的第二部分包括使用扩展 (36 位）虚拟空间，从而有可能映射全部远端 GPU 内存，并使 PCIe 空间的分解视图进入源 BIF。不过，由于需要传送相位位，减少了不改变 P2P BAR 的情况下可寻址的表面最大尺寸。值得注意的是，GPU 真实的 PCIe 孔径无需任何特别的对准；通过使可用地址范围的 1/n 分割达合理粒度来决定相位（代价是目标侧的额外比较器）。值得注意的是假定任何物理表面页的分散 - 集合在目标 HDP 页表或 GART 中进行，以使 PCIe 空间表面地址毗连。

[0131] 图 9 显示相位压缩如何工作。图中仅显示两个相位。假定本地内存为 256MB，每个 GPU 仅有 128MB PCIe 孔径。图 9A 为 GPU0 角度的系统存储空间视图 (GPU 虚拟存储空间）。图 9B 为 PCIe 角度的系统存储空间视图（结构物理内存空间）。图 9C 为 GPU1 角度的 LVM 重构视图 (HDP 存储空间）。

[0132] 为充分利用有限的 PCIe 孔径尺寸，可关闭来自相同源 GPU 的多相位的使用。值得注意的是，一个典型大表面为每部分 2Kx2Kx32b 或 64MB。这将适合 512MB/8GPU 孔径。一实

施例中，孔径编码的应用是在发送 BIF 之前。

[0133] 这里所述的结构和方法包括系统，该系统包括：多个专用处理器，耦接至主桥总线；直接总线，将所述各专用处理器直接耦接至其余专用处理器中的至少一个；以及存储控制器，耦接至至少一个所述专用处理器。其中，所述至少一存储控制器决定是否通过该主机总线或直接总线传输数据以及是否通过该主机总线或直接总线接收数据。

[0134] 实施例中，该存储控制器驻留于至少一个所述专用处理器上。

[0135] 实施例中，所述多个专用处理器包括图形处理单元 (GPU)。

[0136] 实施例中，所述主机总线包括高速周边元件接口 (PCIe) 总线。

[0137] 实施例中，所述直接总线包括 PCIe 总线。

[0138] 实施例中，所述直接总线包括专有总线。

[0139] 实施例中，该系统还包括至少一本地存储器，至少一系统存储器，以及至少一对等存储器，其中所述至少一对等存储器驻留于专用处理器上，所述存储控制器经配置而基于所述各存储器的地址范围决定访问哪个存储器。

[0140] 实施例中，当地址落在该对等存储器范围内时，该存储控制器基于所述主机总线和直接总线的流量决定使用哪个端口。

[0141] 实施例中，当地址落在该对等存储器范围内时，该存储控制器基于所述主机总线和直接总线目前是否满负荷而决定使用哪个端口。

[0142] 实施例中，当地址落在该对等存储器范围内时，该存储控制器的决定基于向所述主机总线或直接总线分配专门地址。

[0143] 实施例中，该系统还包括至少一本地存储器，至少一系统存储器，以及至少一对等存储器，其中，所述至少一对等存储器驻留于专用处理器上，并且所述存储控制器经配置而基于所述各存储器的地址范围决定访问哪个存储器，并基于所述主机总线和直接总线的流量执行动态负载平衡。

[0144] 实施例中，在多处理器系统中传输数据的方法包括：决定是否通过主机处理器与多个专用处理器共享的主机总线或者通过该多个专用处理器之间的直接总线传输或接收数据，其中，所做决定取决于传输的存储地址范围。

[0145] 实施例中，所述地址范围指定驻留于专用处理器上的对等存储器、所述主机总线上的系统存储器以及所述专用处理器之间耦接的本地存储器中其中一者。

[0146] 实施例中，落在指定所述对等存储器的地址范围内的专门地址被分配至所述主机总线或直接总线。

[0147] 实施例中，当地址落在指定该对等存储器的范围内时，所做决定基于所述各主机总线和直接总线的流量。

[0148] 一实施例中，该方法还包括执行动态负载平衡，以基于历史流量数据调节所述各总机主线和直接总线上的百分比流量。

[0149] 一实施例中，该方法还包括向所述主机处理器分配存储器的图圈并分配所述各专用处理器上的各自流量，其中，该存储器的图圈应用于该对等存储器、系统存储器和本地存储器，并且该图圈决定自各处理器角度所看到的内存。

[0150] 一实施例中，该方法还包括执行图圈的压缩，该执行图圈的压缩包括使用偏移寄存器通过改变偏移而使对等存储器全部空间都是可见的。

[0151] 一实施例中，指令包括硬件描述语言指令，其可用于创建专用集成电路 (ASIC) 以执行所述方法。

[0152] 这里所述的结构和方法还包括存储指令的计算机可读媒体，当执行所述指令时，使多处理器系统中的数据传输方法得以执行。所述方法包括：决定是否通过主机处理器与多个专用处理器共享的主机总线或者通过该多个专用处理器之间的直接总线传输或接收数据，其中，所做决定取决于传输的存储地址范围。

[0153] 一实施例中，所述地址范围指定驻留于专用处理器上的对等储存器，该主机总线上的系统存储器以及耦接至该专用处理器的本地存储器中的其中一者。

[0154] 一实施例中，在该范围内指定地址包括指定将该对等存储器分配给所述主机总线或直接总线。

[0155] 一实施例中，当地址落在指定该对等存储器的范围内时，所做决定基于所述各主机总线和直接总线的流量。

[0156] 一实施例中，该方法还包括执行动态负载平衡以基于历史流量数据调节所述总机主线和直接总线上的百分比流量。

[0157] 一实施例中，该方法还包括向所述主机处理器分配存储器的图圈并分配所述各专用处理器上的各自流量，其中，该存储器的图圈应用于该对等存储器、系统存储器和本地存储器，并且该图圈决定自各处理器角度所看到的内存。

[0158] 一实施例中，该方法还包括执行图圈的压缩，该执行图圈的压缩包括使用偏移寄存器通过改变偏移而使对等存储器全部空间都是可见的。

[0159] 一实施例中，所述指令包括硬件描述语言指令，其可用于创建 ASIC 以执行所述方法。

[0160] 这里所述的结构和方法还包括在多处理器系统中传输数据的方法，该方法包括：决定是否通过主机处理器与多个专用处理器共享的主机总线或者通过所述多个专用处理器之间的直接总线传输或接收数据，其中所做决定取决于传输的存储地址范围；通过该直接总线在该多个专用处理器之间直接传输数据，其中，该专用处理器包括具有框型缓冲图圈的图形处理单元，并且直接传输状态是包括提供经由该直接总线在框型缓冲图圈之外的访问。

[0161] 可将上述实施例的各种态样实现为功能，将其编程到任意类型电路中，包括但不限于可编程逻辑电路 (programmable logic device；PLD)，例如现场可编程门阵列 (filed programmable gate array；FPGA)、可编程阵列逻辑 (programmable array logic；PAL) 设备、电性可编程逻辑和存储器设备、标准单元设备、以及 ASIC 和全定制集成电路。实现实施例态样的其他可能性包括具有存储器的微控制器（例如 EEPROM、快闪存储器等）、嵌入式微处理器、固件、软件等。此外，实施例态样可包括于具有基于软件电路仿真的微处理器、离散逻辑（时序和组合）、自定义设备、模糊（神经）逻辑、量子设备和任意上述设备类型的组合中。当然，基础的设备技术可提供于各种元件类型中，例如 MOSFET 技术，例如 CMOS、bipolar 技术例如射极耦合逻辑 (emitter-coupled logic；ECL)、高分子技术（例如硅共轭聚合物和金属共轭聚合物 - 金属结构）、模拟和数字混合等。

[0162] 说明书和权利要求书中所用的术语“处理器”包括处理器核心或处理器的一部分。另外，尽管这里通常提到独立的一个或多个 GPU 和一个或多个 CPU，实施例中，GPU 和 CPU 被

包括于单个集成电路封装或单片晶粒上。因此，在这些实施例中是单个设备执行所请求保护的方法。

[0163] 除非特别指出，说明书和权利要求书中的“包括”等词语都解释为包含的含义，不具有排他或穷举的含义。也就是“包括，但不限于”的意思。单数或复数形式的词语还分别包括复数形式和单数形式。此外，“这里”、“以下”、“上面”、“下面”等词语指应用整体而非特定部分。当“或者”用于两个或多个条目时，该术语覆盖下列所有解释：任意所列条目、全部所列条目，以及所列条目的任意组合。

[0164] 上面对方法和系统实施例的描述并非意图穷举或限制本发明。尽管这里出于说明目的对方法和系统的特定实施例和实施方式作了描述，本领域的技术人员应了解，可在本发明范围内进行各种等同修改。这里所揭露的教导可用于其他系统，而不只是上面所述的包括图形处理或视频处理的系统。这里所述的各种操作可在各种结构中执行，以不同的方式分布。此外，尽管这里描述多种配置，但这些配置并非意图限制或排他。

[0165] 其他实施例中，这里所述的某些或全部硬件和软件功能可存在于打印机、照相机、电视、DVD 播放器、数字录像机 (digital video recorder ;DVR) 或个人摄像机 (personal video recorder ;PVR)、手持设备、手机或其他设备中。可通过组合上述各种实施例的元素和动作而提供另外的实施例。可对所述方法和系统作出变更。

[0166] 一般而言，在下面的权利要求书中，所使用的术语不应当解释为将所述方法和系统限制于说明书和权利要求书的特定实施例，而是应当解释为包括依据权利要求操作的任意处理系统和方法。因此所述方法和系统不受揭露限制，而是由权利要求整体决定。

[0167] 尽管所述方法和系统的特定态样以特定权利要求形式呈现，发明人考虑了所述方法和系统以任意数目权利要求形式的各种态样。例如，尽管所述方法和系统仅有一个态样举例实施为计算机可读媒体，其他态样同样可实施为计算机可读媒体。这样的计算机可读媒体可储存计算机设备（例如个人电脑、个人数字助理、PVR、移动设备等）要执行的指令或是在执行时用来创建执行上述态样的设备 (GPU、ASIC 等）或软件应用的指令（例如 Verilog 或硬件描述语言）。所述要求保护的发明可体现为计算机代码（例如 HDL、Verilog 等），创建、储存、合成并使用这些计算机代码来生成 GDSII 数据（或等同物）。接着，可基于该数据生产 ASIC。

[0168] 因此，提出申请后，发明人保留添加额外权利要求的权利，以获得所述方法和系统的其他态样的额外权利要求形式。

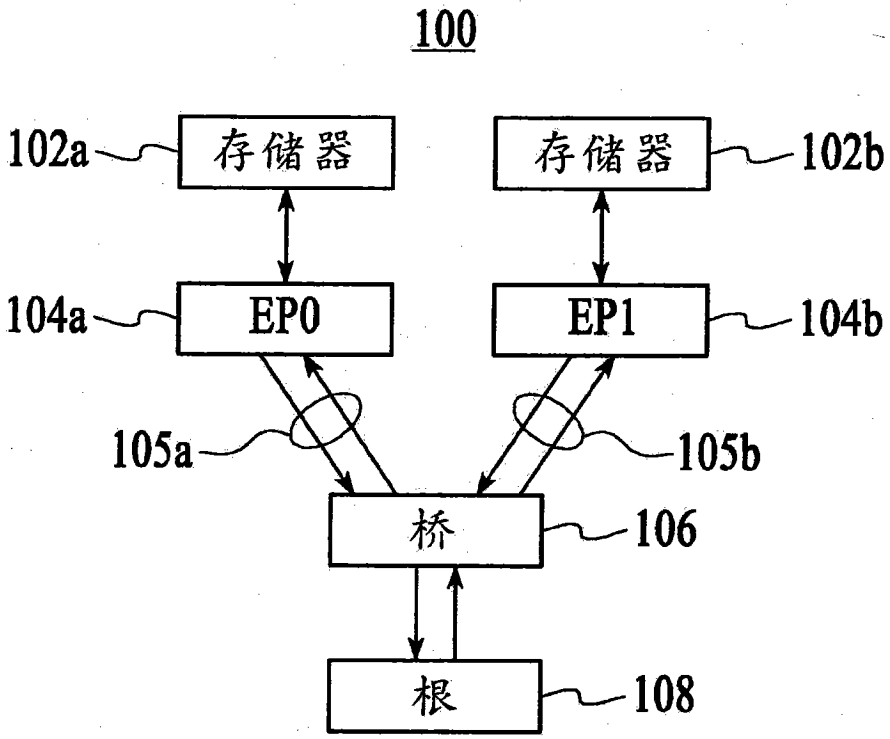

100
102a
存储器
存储器
102b
104a
EP0
EP1
104b
105a
105b
桥
106
根
108

图 1（现有技术）

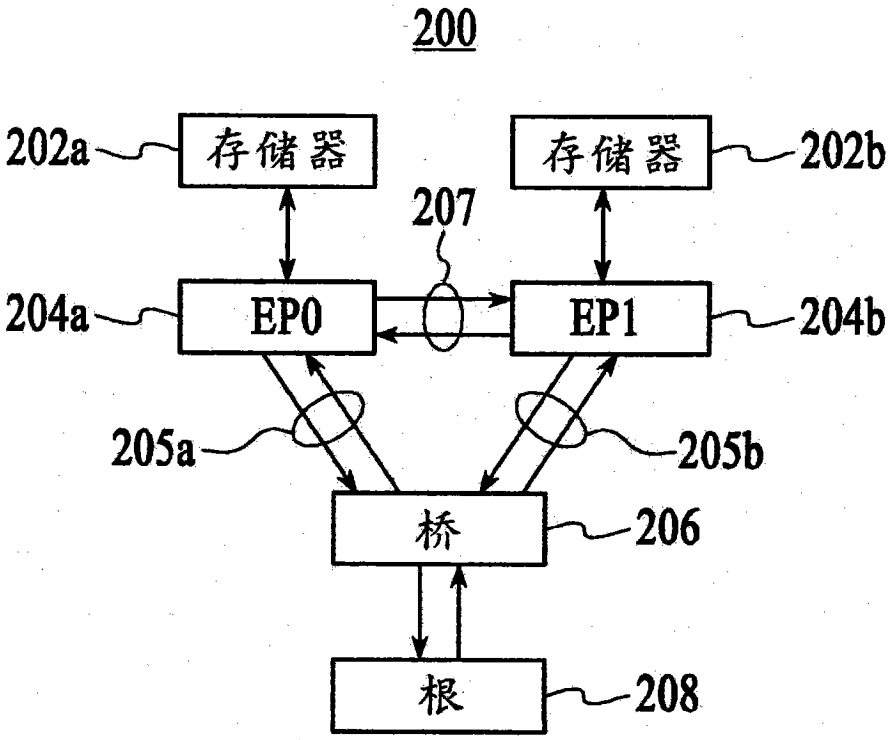

200
202a
存储器
存储器
202b
207
204a
EP0
EP1
204b
205a
205b
桥
206
根
208

图 2

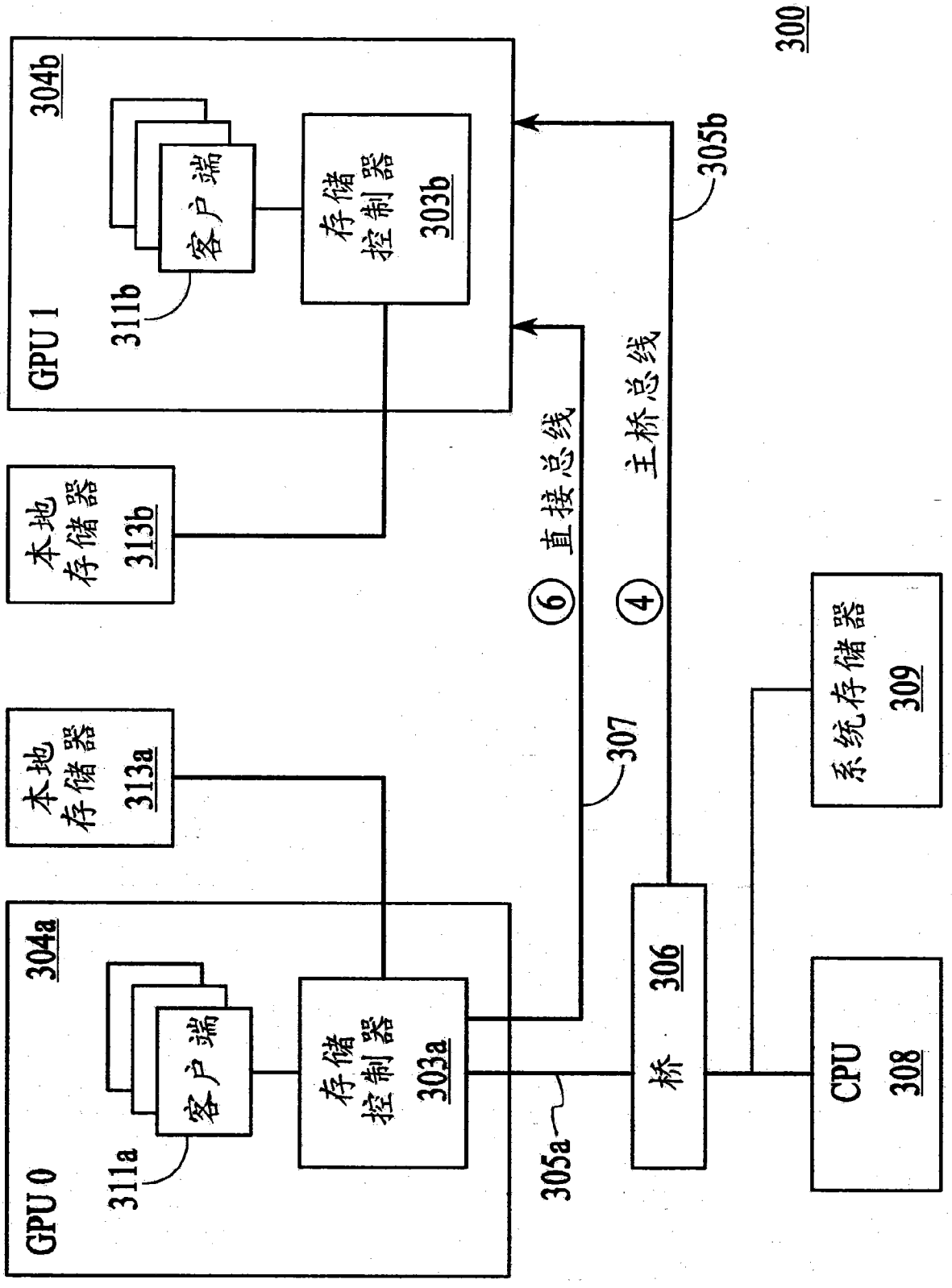

300
GPU 1
304b
客户端
311b
存储控制器
303b
本地存储器
313b
⑥ 直接总线
④ 主桥总线
305b
本地存储器
313a
307
系统存储器
309
GPU 0
304a
客户端
311a
存储控制器
303a
305a
桥
306
CPU
308

图 3

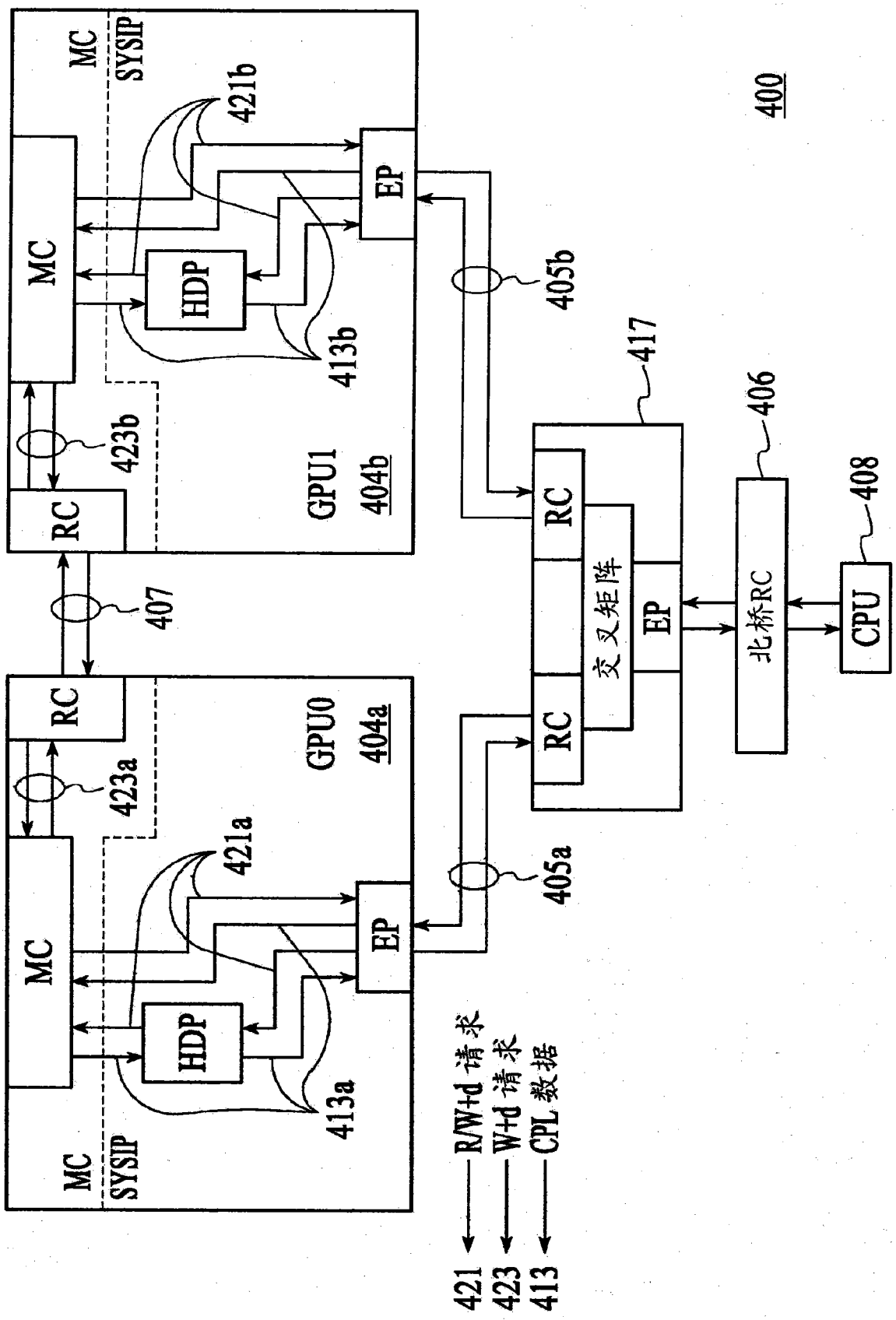

MC
SYSIP
421b
MC
HDP
EP
413b
423b
RC
GPU1
404b
407
RC
GPU0
404a
423a
421a
MC
HDP
EP
413a
MC
SYSIP
405b
405a
417
RC
交叉矩阵
EP
RC
406
北桥RC
408
CPU
400
R/W+d 请求
W+d 请求
CPL 数据
421
423
413

图 4

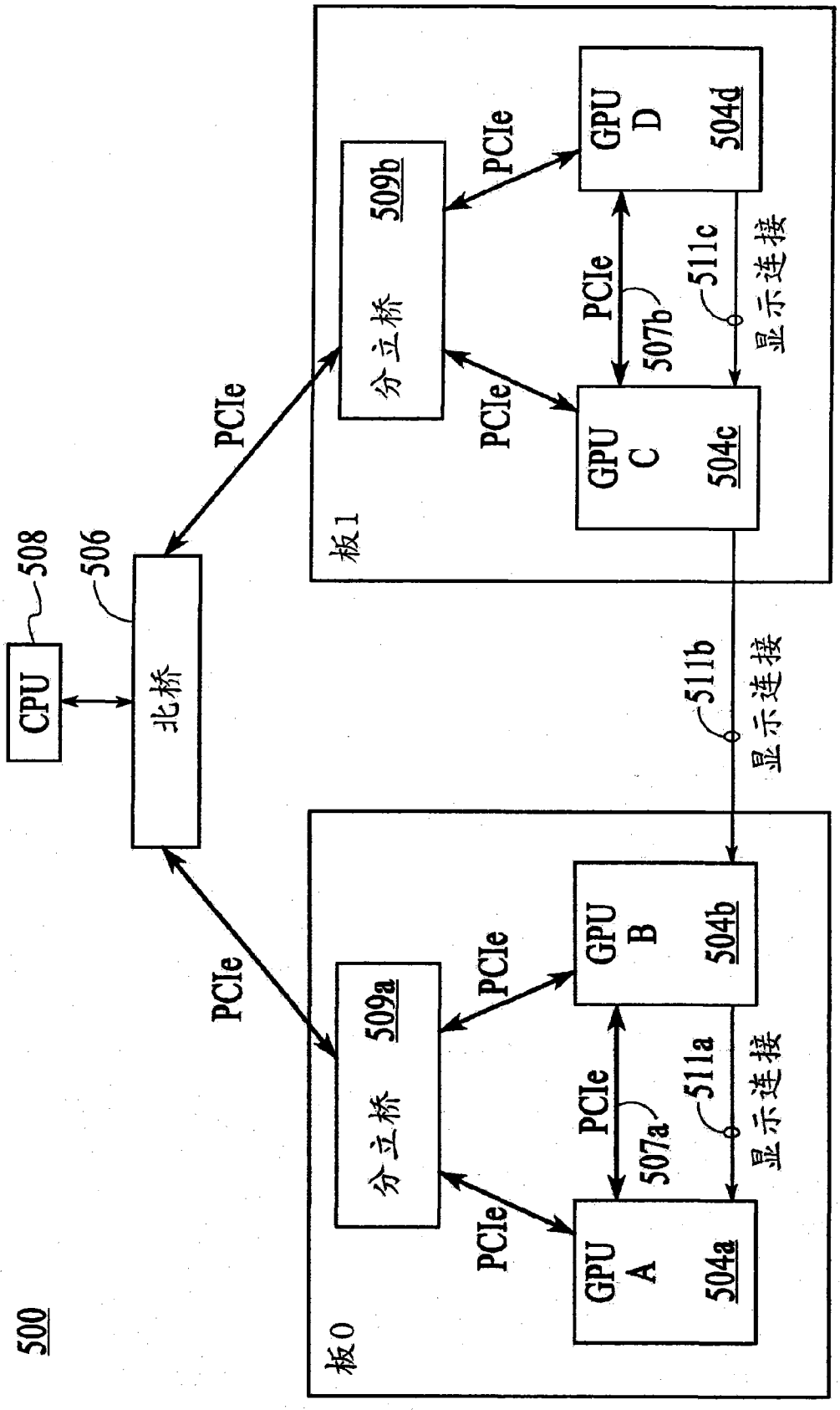

500
板0
板1
CPU
508
北桥
506
PCIe
分立桥
509a
509b
GPU A
504a
GPU B
504b
GPU C
504c
GPU D
504d
507a
507b
511a
511b
511c
显示连接

图 5

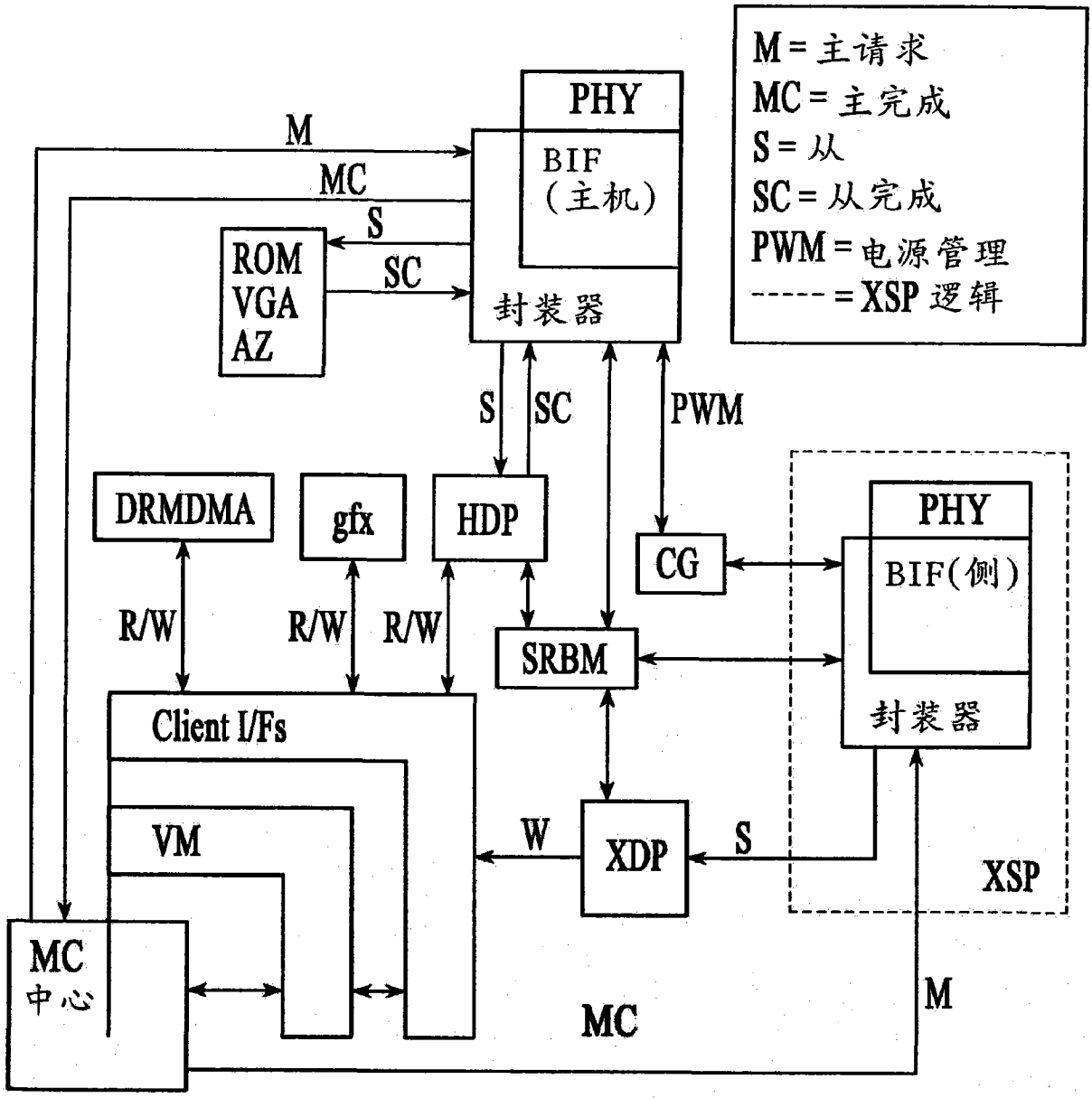

M = 主请求
MC = 主完成
S = 从
SC = 从完成
PWM = 电源管理
----- = XSP 逻辑
PHY
BIF
(主机)
封装器
M
MC
S
SC
ROM
VGA
AZ
S
SC
PWM
DRMDMA
gfx
HDP
CG
PHY
BIF(侧)
R/W
R/W
R/W
SRBM
封装器
Client I/Fs
VM
W
XDP
S
XSP
MC
中心
MC
M

图 6

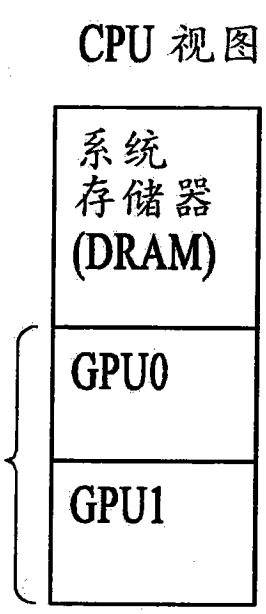

CPU 视图
系统
存储器
(DRAM)
GPU0
GPU1

图7A

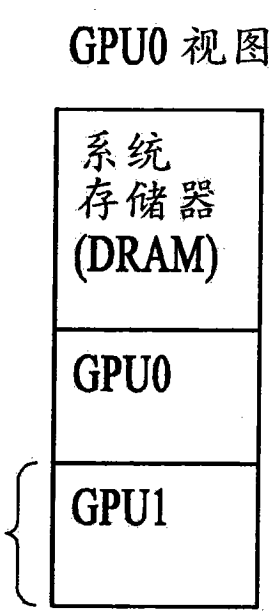

GPU0 视图
系统
存储器
(DRAM)
GPU0
GPU1

图7B

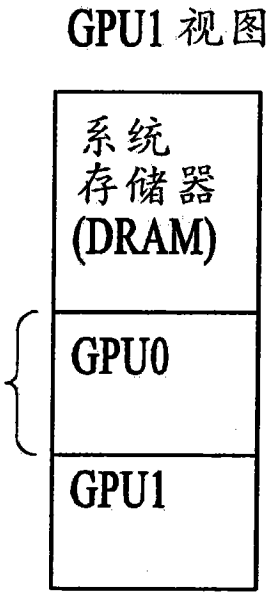
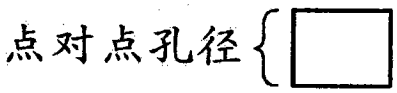

GPU1 视图
系统
存储器
(DRAM)
GPU0
GPU1

图 7C


点对点孔径

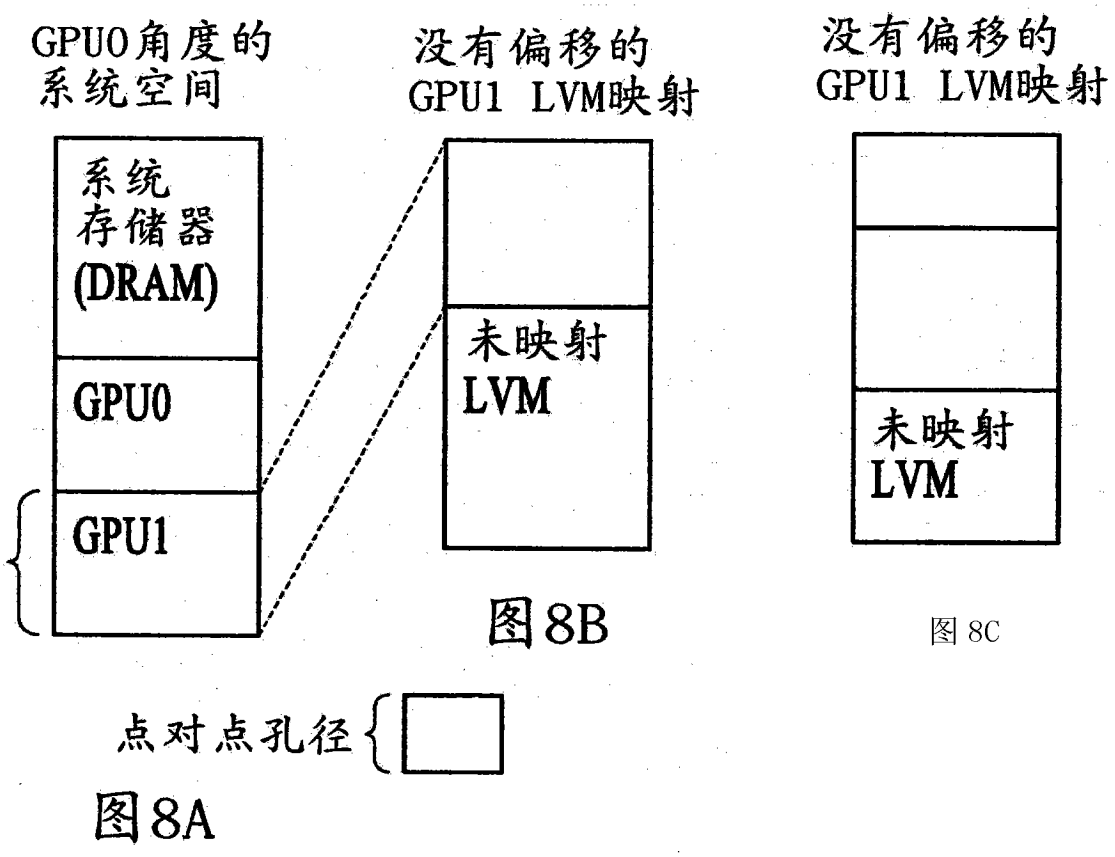

GPU0角度的
系统空间
系统
存储器
(DRAM)
GPU0
GPU1
没有偏移的
GPU1 LVM映射
未映射
LVM
没有偏移的
GPU1 LVM映射
未映射
LVM
点对点孔径

图8B

图8C

图8A

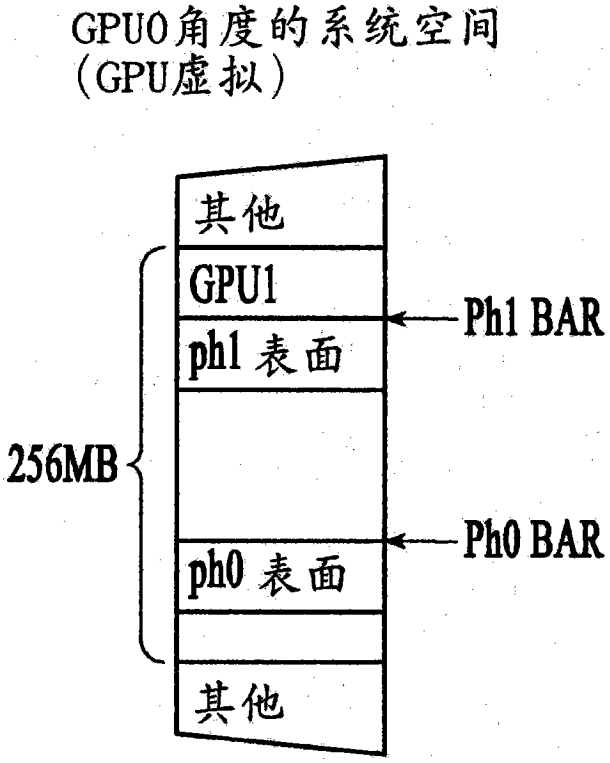

GPU0角度的系统空间
(GPU虚拟)
其他
GPU1
ph1 表面
256MB
ph0 表面
其他
Ph1 BAR
Ph0 BAR

图 9A

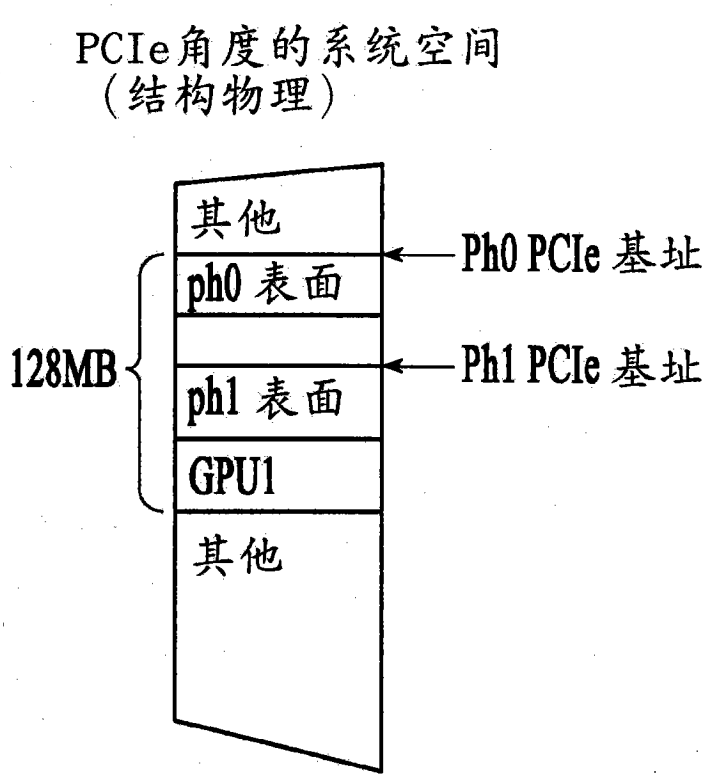

PCIe角度的系统空间
(结构物理)
其他
ph0 表面
128MB
ph1 表面
GPU1
其他
Ph0 PCIe 基址
Ph1 PCIe 基址

图 9B

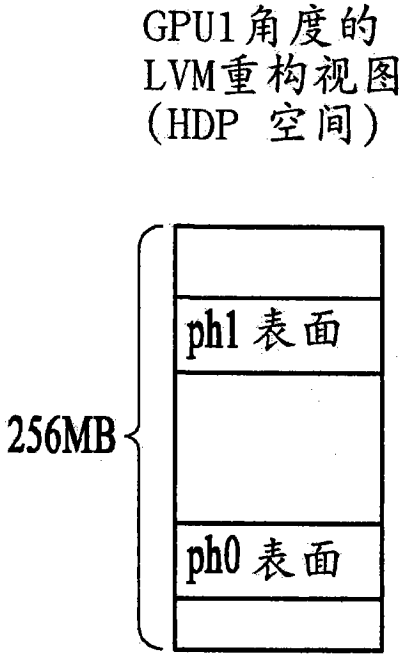

GPU1角度的
LVM重构视图
(HDP 空间)
ph1 表面
256MB
ph0 表面

图 9C