



(12) 发明专利

(10) 授权公告号 CN 101484883 B

(45) 授权公告日 2013. 01. 30

(21) 申请号 200780024810. 9

G06T 1/00 (2006. 01)

(22) 申请日 2007. 06. 22

(56) 对比文件

(30) 优先权数据

11/480, 663 2006. 06. 30 US

US 6895491 B2, 2005. 05. 17, 全文.

US 6907600 B2, 2005. 06. 14, 全文.

US 5088031 A, 1992. 02. 11, 全文.

US 5077654, 1991. 12. 31, 全文.

CN 1687900 A, 2005. 10. 26, 全文.

CN 1512354 A, 2004. 07. 14, 全文.

(85) PCT申请进入国家阶段日

2008. 12. 30

(86) PCT申请的申请数据

PCT/US2007/071953 2007. 06. 22

(87) PCT申请的公布数据

W02008/005722 EN 2008. 01. 10

(73) 专利权人 英特尔公司

地址 美国加利福尼亚州

(72) 发明人 B·韦布 A·纳瓦尔 W·A·甘地

(74) 专利代理机构 中国专利代理(香港)有限公

司 72001

代理人 柯广华 王丹昕

审查员 蓝娟

(51) Int. Cl.

G06F 12/02 (2006. 01)

G06F 13/28 (2006. 01)

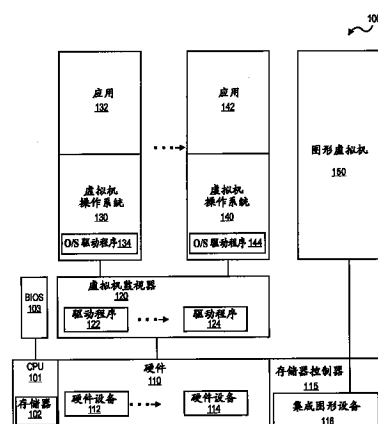
权利要求书 2 页 说明书 20 页 附图 16 页

(54) 发明名称

用于重新映射图形数据的存储器地址的装置和方法

(57) 摘要

一种方法和装置用于创建访客物理地址 (GPA)- 主机物理地址 (HPA) 映像转换表、更新这些转换表和使用这些转换表来转换实现虚拟机监视器以支持虚拟机的计算环境的图形数据直接存储器访问 (DMA) 请求的 GPA。可相对于虚拟机监视器透明地通过计算环境的渲染或显示路径从一个或多个虚拟机发送这些请求。通过存储器控制器检测发送给现有的全局和页目录表的项, 从所检测的项生成映像表项, 并对于映像表项将 GPA 转换为 HPA, 执行上述复制、更新和使用。



1. 一种用于重新映射图形数据的存储器地址的装置,包括:

在具有虚拟机监视器 VMM 和图形虚拟机的虚拟化环境中的存储器控制器电路,用于创建将图形数据的访客物理地址 GPA 转换为直接存储器访问 DMA 请求的主机物理地址 HPA 的单级查询表,其中所述单级查询表对于所述存储器控制器电路可见,而对于图形虚拟机(150)不可见;并且其中所述存储器控制器电路包括用于将所创建的项的数据或偏移量与来自将 GPA 重新映射到 HPA 的单级查询表或三级遍历的数据或偏移量组合以便创建所述单级查询表的项的电路。

2. 如权利要求 1 所述的装置,其中所述单级查询表包括用于将计算系统的渲染路径中的图形数据重新映射的多个项,所述计算系统在所述 VMM 的第一虚拟机(130/140)上执行应用以便向所述渲染路径发送直接存储器访问(DMA)请求。

3. 如权利要求 1 所述的装置,其中所述单级查询表包括用于将计算系统的显示路径中的图形数据重新映射的多个项,所述计算系统在所述 VMM 的第一虚拟机(130/140)上执行应用以便向所述显示路径发送直接存储器访问(DMA)请求。

4. 如权利要求 1 所述的装置,其中所述单级查询表对于所述存储器控制器电路可见而对于所述 VMM 不可见,所述单级查询表是显示路径的全局访客转换表(GTT)的映像表和渲染路径的每个进程页显示(PPPD)转换表的映像表之一,并且其中所述存储器控制器电路还包括用于在存储器中写所述单级查询表的电路。

5. 如权利要求 1 所述的装置,其中所述存储器控制器电路包括用于创建所述单级查询表、将所述单级查询表存储在连续存储器中并使用所述单级查询表的图形电路、图形芯片组或存储器控制器(MC)。

6. 如权利要求 5 所述的装置,所述装置还包括用于预先分配所述连续存储器而无需向所述 VMM 报告所述连续存储器的基本输入输出系统(BIOS)。

7. 如权利要求 2 所述的装置,其中所述单级查询表是在引导所述计算系统期间、或在启动在所述 VMM 上的第一虚拟机(130/140)中的应用期间创建的。

8. 如权利要求 1 所述的装置,其中所述存储器控制器电路还包括通过将第一存储器地址转换与第二存储器地址转换组合来创建所述单级查询表的项的电路。

9. 如权利要求 8 所述的装置,其中所述第一存储器地址转换用于将要由在所述 VMM 上的第一虚拟机(130/140)上启动的应用发送的第一虚拟存储器地址转换为供所述 VMM 使用的 GPA,以及所述第二存储器地址转换用于将供所述 VMM 使用的 GPA 转换为所述 HPA。

10. 如权利要求 9 所述的装置,其中所述第一虚拟存储器地址包括要由在所述 VMM 上的第二虚拟机中启动的应用创建而用于访问存储在主存储器中的图形数据的地址。

11. 如权利要求 1 所述的装置,还包括用于检测要由所述 VMM 上的第二图形虚拟机中启动的应用所创建的项的检测电路。

12. 如权利要求 11 所述的装置,其中要创建的项包括用于创建或更新显示路径的全局访客转换表(GTT)、渲染路径的每个页目录(PPD)转换表和所述渲染路径的每个进程全局转换表(PPGTT)之一的项。

13. 如权利要求 1 所述的装置,其中所述存储器控制器电路还包括用于将所述单级查询表存储在连续存储器中的电路;并且还包括:

图形芯片组,用于使用所述单级查询表来将在所述 VMM 的第一虚拟机中运行的应用的

图形数据的 GPA 地址转换为 HPA 地址。

14. 一种计算机服务器,包括:

检测电路,用于检测要由在虚拟机监视器 VMM 上的第一虚拟机中启动的应用创建的项;

查询电路,用于将来自要创建的项的数据或偏移量与来自要由所述 VMM 创建的项的数据或偏移量组合,以便创建将访客地址转换为主机地址的单级查询表的项;

在具有所述 VMM 和图形虚拟机的虚拟化环境中的存储器控制器电路,用于将要由应用在第一存储器地址转换表中创建的项存储在主存储器中,将要由所述 VMM 创建的存储在第二存储器地址转换表中的项存储在所述主存储器中,并将所述单级查询表的项存储在所述主存储器中。

15. 如权利要求 14 所述的计算机服务器,其中所述存储器控制器电路包括图形设备电路或图形控制器电路,用于使用所述单级查询表来将图形显示引擎和图形渲染引擎之一中的 DMA 请求的访客地址转换为主机物理地址 (HPA),而无需向所述 VMM 发送所述请求。

16. 如权利要求 14 所述的计算机服务器,其中所述存储器控制器电路适于使所述单级查询表对于图形虚拟机、所述 VMM 或在所述第一虚拟机中启动的所述应用不可见,并使所述单级查询表对于所述存储器控制器电路中的分离图形卡和集成图形芯片组之一可见。

17. 一种计算系统,包括:

耦合到主存储器的处理器;

在具有图形虚拟机 (150) 和虚拟机监视器 VMM 的虚拟化环境中的耦合到所述处理器和所述主存储器的图形芯片组,所述芯片组包括用于创建将图形数据的访客物理地址 GPA 转换为直接存储器访问 DMA 请求的主机物理地址 HPA 的单级查询表的存储器控制器电路,其中所述图形虚拟机对于所述存储器控制器电路可见,而对于所述 VMM 不可见,并且所述单级查询表对于所述存储器控制器电路可见,而对于所述图形虚拟机或所述 VMM 不可见;并且其中所述存储器控制器电路包括用于将所创建的项的数据或偏移量与来自将 GPA 重新映射到 HPA 的单级查询表或三级遍历的数据或偏移量组合以便创建所述单级查询表的项的电路。

18. 如权利要求 17 所述的计算系统,其中所述图形芯片组是集成图形设备和分离图形控制器之一,用于与在所述 VMM 上运行的图形虚拟机通信而无需访问所述 VMM。

19. 如权利要求 18 所述的计算系统,其中所述图形虚拟机是软件应用,并且所述集成图形设备或所述分离图形控制器在图形虚拟机执行期间不访问所述 VMM。

20. 如权利要求 19 所述的计算系统,其中所述图形虚拟机从虚拟应用 (132/142) 接收所述 DMA 请求,并将所述请求发送给所述芯片组以便使用所述表从 GPA 转换为 HPA。

用于重新映射图形数据的存储器地址的装置和方法

背景技术

[0001] 虚拟机体系结构在逻辑上将物理机分区,以使得机器的底层硬件进行时间共享并且看起来像是一个或多个独立操作的虚拟机 (VM)。虚拟机监视器 (VMM) 在计算机上运行以便于对其它软件进行一个或多个 VM 的抽象。每个 VM 可用作独立平台,以便运行它自己的操作系统 (OS) 和应用软件。在 VM 中运行的软件在本文统称为“访客软件”。

[0002] VMM 环境提供了使得在实际系统上运行的软件能够为访客软件提供近乎完全的虚拟系统或 VM 模仿的硬件和系统控制指令。此类环境的益处包括例如:能够在单个物理机上运行多个操作系统;提高了 CPU 和硬件资源的使用率。

[0003] 虚拟化技术使得平台支持在单个机器或计算系统(或环境)上运行多个分区。这些分区相互隔离,从而具有增加的鲁棒性的优点。这些分区在 VMM 之上运行,VMM 可描述为具有诸如存储器的平台资源的“真实视图”的软件虚拟化层。因此,该存储器真实视图可描述为“主机物理地址”或 HPA(如主机地址)。每个分区或 VM 具有存储器的“虚拟化视图”,它可描述为“访客物理地址”或 GPA。

[0004] 访客软件期望会像它在专用计算机上而不是在 VM 中运行那样操作。即,访客软件期望可控制各种事件并且具有对计算机(如物理机器)上的硬件资源的访问权。物理机器的硬件资源可包括一个或多个处理器、驻留在处理器上的资源(如控制寄存器、高速缓存等)、存储器(例如驻留在存储器中多个地址处的指令和/或数据,如图形指令和/或数据)、图形设备和/或控制器(如图形电路、图形芯片组、图形卡等)以及驻留在物理机器中的其它资源(如输入/输出设备)。事件可包括渲染和显示图形数据以在 VMM 环境中显示图形图像。这些图像可包括像素 图像、编码图像、视频图像或帧、静态图像、图片图像、动画图像、电影等。

[0005] 因此,VMM 向其它软件(“访客软件”、“访客”、或单个“访客”)提供一个或多个 VM 的抽象。VMM 可向各种访客提供相同或不同的抽象。每个访客期望在 VM 中提供的硬件平台的全部设施都可供它使用。例如,根据在 VM 中提供的处理器和平台的体系结构,访客期望具有对所有寄存器、高速缓存、结构、I/O 设备、存储器、图形设备/控制器等的访问权。此外,每个访客期望 VMM 可例如通过处理在显示器或监视器上显示图形图像的访客的图形指令(例如,包括图形地址)和/或数据来处理各种事件。

[0006] 例如,在一些情况下,VMM 可依靠设备的虚拟化来实现输入/输出 (IO) 设备的支持。通常,IO 设备由 VMM 虚拟化,并且 VMM 直接控制平台上的实际硬件。在这些情况下,VMM 模仿显露给 VMM 的 IO 设备。因为 VMM 与硬件直接通信,所以 VMM 携带所支持的所有设备的驱动程序。携带所有驱动程序使得 VMM 代码或软件充斥或具有非期望的巨大代码量,从而导致复杂度增加。

附图说明

[0007] 附图的各图举例而非限制性地示出本发明的各个实施例,其中:

[0008] 图 1A 是示出根据一个实施例的虚拟化环境中的集成图形渲染路径的框图。

- [0009] 图 1B 是示出根据一个实施例的虚拟化环境中的分离图形渲染路径的框图。
- [0010] 图 2 是示出根据集成图形设备或分离图形控制器实施例的、存储在存储器中多个地址处并用于在显示器或监视器上显示的图形数据的框图。
- [0011] 图 3 是示出根据一个实施例的高级调度器模型中的图形数据路径的框图。
- [0012] 图 4 是示出根据一个实施例的用于重新映射三级遍历的直接存储器访问 (DMA) 的框图。
- [0013] 图 5 是示出根据一个实施例的 VMM 中的具有 3 级遍历的图形数据路径的框图。
- [0014] 图 6A 是示出根据一个实施例的虚拟机监视器 (VMM) 环境中的集成图形设备的框图。
- [0015] 图 6B 是示出根据一个实施例的虚拟机监视器 (VMM) 环境中的分离图形设备的框图。
- [0016] 图 7 是示出根据一个实施例用于创建映像全局访客转换表 (GTT) 的环境的框图。
- [0017] 图 8 是用于创建映像全局 GTT 的过程的流程图。
- [0018] 图 9 是示出根据一个实施例用于创建映像访客每个进程页显示 (PPPD) 的环境的框图。
- [0019] 图 10 是根据一个实施例用于创建映像访客 PPPD 的过程的流程图。
- [0020] 图 11 是用于维持映像表的环境的框图。
- [0021] 图 12 是示出根据一个实施例的图形渲染和显示数据路径的框图。
- [0022] 图 13 是示出根据一个实施例的图形显示路径更新和显示路径的框图。
- [0023] 图 14 是示出根据一个实施例的图形渲染路径更新和渲染路径的框图。

具体实施方式

[0024] 本发明的一个或多个实施例一般涉及计算机系统设计。更具体地说, 其中一个或多个实施例涉及用于渲染和显示图形数据以便在虚拟化环境中显示图形图像的方法和装置。例如, 如上所述, 在 VMM 模拟显露给 VM 的 I/O 设备并且 VMM 携带所支持的所有设备的驱动程序的情况下, 携带所有驱动程序会使 VMM 代码或软件充斥或具有非期望的巨大代码量, 从而导致复杂性增加。位于美国加利福尼亚州圣克拉拉的 Intel 公司 (Intel Corporation of Santa Clara, California) 开发了用于定向输入 / 输出 (I/O) 体系结构的虚拟化技术, 该虚拟化技术提供对允许访客 VM 直接驱动 I/O 设备的支持, 从而从 VMM 中免去了驱动程序 (例如, 参见由 Intel 公司在 2006 年 2 月发布的序号为 D51397-001 的 Intel's VT-d 技术 - **Intel® Virtualization Technology for Directed I/O Architecture Specification**)。

[0025] 因此, 本文的描述包括用于根据每个访客应用的图形指令 (例如, 包括图形地址) 和数据来显示在多个虚拟机上运行的多个访客应用的图形图像的方法和装置。例如, 可使用图形虚拟机 (VM) (它可包括 VM 操作系统和其它应用, 如在图形 VM 中运行的其它图形应用)、图形控制器和 / 或图形设备例如通过处理访客的图形指令和 / 或数据来处理各种应用程序图形需要, 以便在显示器上显示图形图像。处理访客的图形指令和 / 或数据可包括例如通过将图形指令和 / 或数据写入到存储器中的物理地址 (例如, 物理地址对应于来自访客应用的图形地址) 以及从存储器的物理地址读取图形指令和 / 或数据来访问图形指令和 / 或数据。

[0026] 在以下描述中,阐述了众多具体细节,例如虚拟机监视器(VMM)环境实现、信号和总线的大小和名称、系统组件、硬件、处理器、电路、设备、图形设备、图形控制器、存储器、操作系统、VM的类型和相互关系,以及逻辑分区/集成选择,以便更充分地理解本发明。但是,本领域的技术人员将明白,在没有这些具体细节的情况下,也可实现本发明。在其它情况下,并未详细示出控制结构和门级电路(如电路系统),以免使本发明晦涩难懂。使用所包含的描述,本领域的技术人员将能够实现合适的逻辑电路,而无需进行过多的实验。

[0027] 在以下描述中,使用某些术语来描述本发明的特征。例如,术语“设备”、“芯片”、“芯片组”和/或“电路”可表示硬件、待由处理器执行的应用程序、计算机操作系统、包含待由处理器执行的机器可读介质、和/或配置成执行一个或多个功能的软件。例如,“设备”、“控制器”、“芯片”、“芯片组”和/或“电路”的示例可包括但不限于或局限于电子硬件、逻辑电路、处理器、存储器、集成电路、芯片、芯片组、“卡”、其上具有电子硬件和/或软件的印刷电路板、固件、有限状态机或甚至组合逻辑。集成电路可采取诸如微处理器的处理器、专用集成电路、数字信号处理器、微控制器等形式。在一些情况中,本文描述的电路、图形设备、图形控制器和/或芯片组可以是或可以包括图形电路、图形芯片组、图形卡、图形存储器、图形VM、图形VM操作系统、集成图形设备、分离图形控制器、渲染引擎、显示引擎等。而且,在本文中的一些情况中,术语“图形数据”可用于描述图形指令和数据。

[0028] 此外,访问、接收、检测、感测或以其它方式知道表中的输入可包括在试图在表中输入之前、期间或之后(包括在输入实际没有存储在表中时)硬件和/或软件感测该输入。在一些情况中,在考虑另一个地址而转换、变换、建立或创建一个地址(例如,将图形地址(例如,来自访客应用的指令或请求中所包含的虚拟地址)转换或变换为GPA或主机物理地址(HPA))可表示相同或相关的进程、装置和/或系统。例如,可根据GPA或HPA与图形地址之间的关系(例如,存储器地址变换)从图形地址创建、转换、变换或建立GPA或HPA。

[0029] 图1A是示出根据一个实施例的虚拟化环境中的集成图形渲染路径的框图。图1A示出的VMM环境100包括耦合到虚拟机操作系统130、虚拟机操作系统140与硬件110或耦合在它们之间的虚拟机监视器(VMM)120。图1A还示出耦合到硬件110的存储器控制器115的图形设备116的图形VM150(它可包括VM操作系统和其它应用,例如在图形VM中运行的其它图形应用)。图形(或“Gfx”)VM150可使用由Intel的定向IO的虚拟化技术(例如,使用Intel的VT-d技术)提供的机制来直接驱动Gfx设备116。设备116可执行图形控制器(例如,参见图1B和6B中的控制器118)的功能。可以在例如桌面型计算机、膝上型计算机、手持式计算设备、个人计算机(PC)、工作站、计算机服务器、联网计算机、电子设备、计算设备等中实施环境100。而且,本文描述的存储器控制器115或其它存储器控制器(MC)可以位于与环境或计算机的处理器(如CPU101)相同或不同的芯片中(或之上)。在一些情况下,存储器控制器可以位于不包括处理器的芯片或芯片组中(或之上)。

[0030] 集成图形路径可以是能够通过发出具有访客物理地址(GPA)的直接存储器访问(DMA)请求并使存储器控制器将GPA转换为主机物理地址(HPA)或主存储器的物理存储器地址而无需访问VMM来访问图形数据的路径。例如,设备116和/或存储器控制器115可以将来自操作系统130和/或140的直接存储器访问(DMA)请求的GPA“独立地”转换为用于从存储器的物理地址直接访问图形数据的HPA(例如,存储器360的HPA362,如针对图2所述),而无需使用或访问图形VM150和/或无需访问VMM120(例如,在系统130和

140 通过 VMM 120 访问图形控制器 116 时,无需发送对 VMM 中的图形驱动程序的请求或以其它方式访问 VMM 120 来获取图形驱动程序功能性)。因此,图形 VM150 可与图形控制器 116 通信以便满足系统 130 和 140 的图形需求,而不需要 VMM 120 中的图形驱动程序。集成的部件可包括一个或多个集成图形设备,例如与存储器控制器 115 集成或作为存储器控制器 115 的一部分的集成图形设备 116。MC 115 可包括存储器控制器设备电路或被描述为存储器控制器设备电路,并且下文还将在图 6A 中进一步描述。

[0031] 硬件 110 包括硬件设备 112 和一个或多个额外的硬件设备,如硬件设备 114。硬件 110 还包括中央处理单元 (CPU) 101 和存储器 102。CPU 101 耦合到基本输入 / 输出系统 (BIOS) 103。

[0032] VMM 120 包括驱动程序 122 和一个或多个其他驱动程序 (例如用于它直接驱动和 / 或控制的设备的驱动程序 124)。系统 130 包括操作系统驱动程序 134,并且可以运行一个或多个应用,如应用 132。类似地,系统 140 包括操作系统驱动程序 144,并且可以运行一个或多个应用,如应用 142。

[0033] CPU 101 可以是处理器,例如能够执行支持 VMM 120 和 VMM120 上运行的各种虚拟机以及图形 VM 150 的所需处理的处理器。CPU101 可以是环境 100 的中央处理单元 (CPU),并因此控制并协调该环境的整体操作。在某些实施例中,CPU 101 通过执行例如存储器 102 中存储的操作系统软件来实现此目的。CPU 101 的处理器可以是或可以包括一个或多个可编程通用或专用微处理器、数字信号处理器 (DSP)、可编程控制器、专用集成电路 (ASIC)、可编程逻辑设备 (PLD) 等或此类设备的组合。类似地,存储器 102 可以存储要被 CPU 101、图形 VM 150、设备 116 和 / 或控制器 115 操作、转换、检测、感测、接收和 / 或访问的指令和 / 或数据,例如通过将这些指令和 / 或数据存储存储在存储器 102 的地址 (例如使用 HPA 寻址的地址) 处来存储它们。存储器 102 可以是或包括环境 100 的“主存储器”。存储器 102 表示任何形式的随机存取存储器 (RAM)、只读存储器 (ROM)、闪存存储器等或此类设备的组合。存储器 102 可以包括多种类型的存储器,例如 RAM、动态 RAM (DRAM)、标准动态 RAM (SDRAM)、双数据速率 RAM “DDR RAM” 和双数据速率版本 2 RAM “DDR2 RAM”。例如,正如本领域中公知的,存储器 102 可以是一组或多组存储器芯片、集成电路、RAM 类型等。CPU 101 能够经由地址和数据总线对主存储器 102 直接访问。硬件 110 和设备 112-114 可以表示可安装在环境 100 中的物理设备,例如键盘、海量存储控制器、网络接口、鼠标、声卡等。BIOS 103 可以表示可存储在例如存储器 102 中或存储在单独的非易失性存储器 (未示出) 中的软件指令。

[0034] 图形设备 116 可以是集成图形设备 (例如与例如 MC 115 的存储器控制器集成并执行图形显示功能的设备)。而且,集成图形可以描述没有或不包括“胶连逻辑 (glue logic)”、输入 / 输出接口和 / 或 MC 115 和装置 116 之间的总线的系统或设备。在一些情况下,装置 116 可以比分离图形控制器 (例如与图 1B 和 6B 的控制器 118 相比) 成本低和 / 或包括更少的电子电路或硬件。

[0035] 在大多数实施例中,当使用直接存储器访问 (DMA) 时,需要使用主机硬件物理存储器地址或存储器中的物理地址 (例如参见图 2 的地址 362) 来访问主存储器中的图形数据。这种读和写可以描述为访问存储器和 / 或检测要对存储器设置的项 (例如要向存储器的物理地址写数据和从存储器的物理地址读数据),并且可以根据应用 132 之一 (例如虚拟

机中的应用)来进行这种读或写或因应用 132 之一(例如虚拟机中的应用)而导致这种读或写。在大多数虚拟化环境中,设备发出的 DMA 请求中的地址将是通过 Gfx 地址重新映射表(例如通过 GTT 或图形地址重新映射表(GART))转换为 GPA 的图形地址,但是对于要由设备 116(或图 1B 和 6B 的控制器 118)满足的 DMA 请求,必须将这些 GPA 转换为 HPA。

[0036] 在计算机系统(例如环境 100)开始被加电时(或复位操作之后),例如从关机状态起的系统引导或初始化过程期间,测试和配置了易失性存储器结构之前,BIOS 软件(例如,BIOS 103)(例如经由 CPU 101)控制计算机系统(例如环境 100)。因此,BIOS 可以在引导序列期间预留存储器 102 中的“隐藏”或“窃取”部分以用于存储本文所描述的转换表(例如映像表)。

[0037] BIOS 软件可以存储在非易失性存储器中或作为固件。固件是指存储在非易失性存储器结构(例如只读存储器(ROM)、闪速存储器等)中的处理器例行程序(例如软件)。这些存储器结构即使在断开电源时仍保留存储在其中的代码。尽管固件存储在非易失性存储器中,但仍可以将固件复制到易失性存储器中或在易失性存储器中建立映像。

[0038] 应用 132 表示可以在系统 130 的控制下被加载和/或执行的程序。OS 和驱动程序 134 与 VMM 提供的虚拟平台接口。在一些情况中,驱动程序 134 可以是操作系统图形驱动程序,VMM 120 向其表示或提供虚拟机,例如包括 CPU、存储器和含图形设备和硬件的硬件的机器,虽然虚拟允许系统 130 看上去像是真实的非虚拟系统(例如例如物理系统)一样工作。

[0039] 类似地,应用 142 表示可以在系统 140 的控制下被加载和/或执行的程序。应用 132 和 142 的指令和数据可以存储在存储器 102 中(例如存储在存储器 102 的物理地址处)。

[0040] 操作系统 130 和/或 140 可以包括或可以是本领域中公知的常规 OS,例如磁盘操作系统(DOS)、Windows、Unix、Linux、OS/2、OS/9、Xenx 等。例如,操作系统可以包括诸如美国华盛顿州雷德蒙德的 Microsoft 公司(Redmond, Washington)的 WINDOWS XP[®]或另一种 WINDOWS[®]操作系统。操作系统 130 和/或 140 还可以包括美国加利福尼亚州库珀蒂诺 Apple 计算机公司(Cupertino, California)的 MACINTOSH[®]操作系统。

[0041] 而且,图形 VM、操作系统(例如图形 VM 150)和/或图形设备均可包括或可以是一组例如通过控制资源的操作和分配来控制图 1-14 的设备和进程的一个或多个软件程序。应用程序 132 或 142 可以是一组具有要显示给用户的图形、图形对象、图像、图形帧和/或视频的一个或多个软件程序。

[0042] VMM 120 是模仿计算机系统的虚拟机监视器;如图所示,它向在其控制下运行的软件提供与真实系统硬件相似的接口。VMM 120 可以在逻辑上对物理机器分区,以使得该机器的底层硬件 110 进行时间共享并看上去像是一个或多个独立操作的虚拟机(例如系统 130 和 140)。VMM 120 可以创建 VM 环境并运行在计算机上,以便于为其他软件进行一个或多个 VM 的抽象。每个 VM 可用作独立平台,以便运行它自己的操作系统(OS)和应用软件。VM 中运行的软件在本文统称为“访客软件”。例如,VMM 120 可以提供接口,从而表示 CPU 101、存储器 102、硬件 110 或与此类似的设备中的一个或多个。VM 内运行的软件常常称为“访客”软件。

[0043] 根据一些实施例,可以将图形设备 116、存储器控制器 215 和图形控制器 118 其

中一个或多个的任何组合视为图形路径。如本文定义的,图形“芯片组”可以包括 MC 115、设备 116 和 / 或控制器 118。例如,对于集成的实施例(例如图 1A 和图 6A),图形芯片组可以是 MC 115 和设备 116。或者,对于非集成的实施例(例如图 1B 和图 6B),图形芯片组可以是 MC 115 和设备 118。一些情况(例如将图形设备(例如图形设备 116)作为存储器控制器(例如 MC 115)的一部分集成的情况)可以描述为“集成图形”(例如图形设备是集成图形设备(例如图 1A 和 6A 中所示的设备 116)的情况)。或者,一些情况(例如并不将图形设备(例如图形设备 118)作为存储器控制器(例如 MC 215)的一部分集成的情况)可以描述为“分离图形”(例如图形设备不是存储器控制器的一部分或不与存储器控制器集成(例如图 1B 和 6B 中所示的设备 118)的情况)。

[0044] 图 1B 是示出根据一个实施例的虚拟化环境中分离图形渲染路径的框图。例如,图 1B 示出的 VMM 环境 200 包括耦合到虚拟机操作系统 130、虚拟机操作系统 140 和硬件 110 或耦合在它们之间的虚拟机 VMM 120。图 1B 还示出虚拟机图形 VM 150、硬件 110 的耦合的存储器控制器 215,存储器控制器 215 又耦合到分离的图形控制器 118。控制器 118 可以执行本领域中公知的图形设备或控制器的功能。可以例如在桌面型计算机、膝上型计算机、手持计算设备、个人计算机(PC)、工作站、电子设备、计算设备等中实施环境 200。

[0045] 图 1B 示出耦合在图形控制器 118、CPU 101 和图形 VM 150 之间的存储器控制器 215。MC 215 可以包括或描述为存储器控制器设备电路,并且下文在图 6B 中对其进行进一步的描述。分离图形设备 118 可以“独立地”将来自操作系统 130、140 或 150 的直接存储器访问(DMA)请求的 GPA 转换成用于从存储器的物理地址直接访问图形数据的 HPA(例如,存储器 360 的 HPA 362,如针对图 2 所述),而无需使用 VMM 120(例如,在系统 130 和 140 通过 VMM 120 访问图形控制器 118 时,不发送对 VMM 中的图形驱动程序请求或以其他方式访问 VMM 120 以获取图形驱动程序功能性)。因此,图形 VM 150 可以与图形控制器 118 通信,例如发送使用图形 VM 150 和控制器 118 来满足系统 130 和系统 140 的图形需求的命令,而无需 VMM 120 中的图形驱动程序。分离部件可以包括一个或多个分离图形设备,例如与存储器控制器 215 分离或不是其一部分的分离图形控制器 118。

[0046] 在一些实施例中,直接将图形设备(例如,图形设备 116 和 / 或控制器 118)指定给图形 VM(例如,图形 VM 150)。此类直接指定(例如,不发送通过 VMM 120 在图形 VM 150 与设备 116 之间传送的图形数据)可以描述为指定的图形。而且,指定的图形或直接指定可以包括图形 VM 可以直接驱动图形设备或控制器的情况,和 / 或图形 VM 中的驱动程序可以对图形设备或控制器的硬件或电路编程而不经使用 VMM 的情况。

[0047] 芯片组或控制器 118 可以包括在分离的图形卡中和 / 或可以从存储器读取图形数据并提供图形数据以便在屏幕、显示器或监视器上显示为图像(例如视频)。图形控制器 118 可以是分离图形设备(例如不与存储器控制器(例如控制器 MC 215)集成且执行图形显示功能的设备)。而且,分离图形可以描述具有或包括“胶连逻辑”、输入 / 输出接口和 / 或 MC 215 和装置 118 之间的总线的系统或设备。控制器 118 可以执行与本文针对设备 116 所描述的功能相同的功能,所例外的是控制器 118 是环境 200 中的分离控制器并作为分离图形控制器与 MC 215 交互或通信。相应地,MC 215 可以执行与本文针对 MC 115 所描述的功能相同的功能,所例外的是 MC 215 使用环境 200 中的分离控制器 118,并且不包括集成图形设备。

[0048] 图 1B 的其他组件可通过上文参考图 1A 和图 6A 来描述,所例外的是它们处在环境 200 中,并且(例如经由 MC 215)与控制器 118 交互或通信,而非与设备 116 交互或通信。例如,存储器 102 可以存储要由 CPU 101、图形 VM 150、MC 215 和 / 或控制器 118 操作、转换、检测和 / 或访问的指令和 / 或数据。

[0049] 当前图形设备(例如设备 116 或 118)使用图形地址转换表来将图形地址(例如 DMA 的地址)变换为物理地址。当将图形设备指定给 VM 时,则可以由图形 VM 中的驱动程序构建转换表。因为图形驱动程序仅具有存储器的访客(GPA)视图,所以通过查询该表获得的地址得到 GPA 地址,因此在(例如要渲染或显示的图形数据的)存储器访问之前需要转换成 HPA。

[0050] 例如,图 2 是示出根据一个实施例的图形数据路径的框图。图 2 示出的环境 300 包括耦合到图形设备 / 控制器 316 的存储器 360,图形设备 / 控制器 316 又耦合到监视器 310。监视器 310 可以是显示器或监视器,例如本领域中公知的计算机或视频监视器。设备 / 控制器 316 可以表示设备 116 或控制器 118。渲染引擎 317 可以用于渲染图形数据,而显示引擎 319 可以用于显示(已经由渲染引擎渲染的)数据。而且,引擎 317 和 319 可以使用渲染数据路径和显示数据路径,其已经包含用于将图形地址变换为物理地址的转换表。存储器 360 使用主机物理地址 (HPA) 362 来(例如通过将图形数据存储在 HPA 地址处来)存储图形数据(例如图形数据 370、372、374、376 和 378)。存储器 360 可以是或包括存储器,并且可以包含上文针对存储器 102 描述的数据。而且,图形数据 370、372、374、376 和 378 可以是图形指令和 / 或数据。例如,数据 370-378 中任何一个数据可以是用于应用 132 和 / 或 142 中任何一个应用的图形数据。因此,地址 362 的相邻位置或地址中的数据可以并不来自相同应用。数据 370-378 可以是数据的“块”,例如 4,096 (4K) 字节或其他数量字节的图形数据的块。

[0051] 如图所示,渲染引擎 317 可以(例如通过 MC 115 或 215 来)访问存储在存储器 360 中的图形数据,例如通过使用 HPA 362 来将渲染的图形数据写入到存储器 360 的 HPA 362 以及从存储器 360 的 HPA 362 读取渲染的图形数据。显示引擎 319 也可以(例如通过 MC 115 或 215 来)访问存储在存储器 360 中的图形数据,例如通过使用 HPA 362 从存储器 360 的 HPA 362 读取渲染的图形数据。引擎 319 可以将渲染的图形数据显示在监视器 310 上,例如通过(例如在从存储器 360 读取渲染的图形数据之后)显示视频馈送、图像、像素化的图像或帧等。例如,在一些情况中,应用 132 可以包括要由渲染引擎 317 渲染并存储在地址 362 处的图形数据。此后可以由显示引擎 319 访问并处理此渲染的数据,并对其处理以显示在显示器 310 上。

[0052] 在一些实施例中,数据块 370-378 的地址是被直接存储器访问 (DMA) 设备用来访问主存储器中的图形数据的主机硬件物理存储器地址或存储器中的物理地址。例如,DMA 请求可以是 MC 115 和 / 或设备 116 不使用 CPU 101 而直接从设备(例如硬件设备 112)接收的有关从存储器 102 读取数据或将数据写入存储器 102 的请求。这种读或写可以描述为访问存储器,并且可以是根据应用 132 中的一个应用来进行的或由此应用所产生的。在将图形设备指定给 VM 的虚拟化环境中,DMA 请求中的地址可以是 GPA,但是对于要由设备 116 满足的 DMA 请求,必须将这些 GPA 转换为 HPA。与针对图 2 描述的概念相似的概念适用于图 1B 和 6B 的 MC 215 和控制器 118(例如环境 200)。

[0053] 虽然可考虑其他实施例,但是图形设备 116 可包括两个主数据路径,例如渲染数据路径和显示数据路径。例如,图 3 是示出根据一个实施例的高级调度器模型中的图形数据路径的框图。本文描述的高级调度器可以包括如美国华盛顿州雷德蒙德的 Microsoft 公司 (Redmond, Washington) 的 Microsoft 高级调度器的功能性的功能性。高级调度器可以是或包括 MC 115 支持的功能性。而且,高级调度器可以是或包括图形存储器管理器操作系统软件或应用。具体地说,高级调度器可以是操作系统或图形 VM(例如图形 VM 150)的一部分。图 3 示出的环境 400 包括非虚拟化环境中的显示路径 410 和渲染路径 420。显示引擎 219 使用图形地址来访问数据。通过全局 GTT 264 将图形地址转换为 HPA。渲染引擎 217 使用两级分层页表遍历来将图形地址转换为 HPA。表 264、266 和 268 是必需的,因为图形 VM 150 使用虚拟地址来对设备 116 编程,而存储在存储器 360 中的图形数据被存储在物理地址 362 中(例如,参见图 2)。因此,数据路径 410 和 420 结合从图形地址到物理地址(例如,HPA)的转换表。因为在将图形设备直接指定给 VM 的虚拟化环境中,GTT 264 和页目录 266 由图形 VM(例如图形 VM 150)来构建或创建(例如通过在引导期间创建它们的项),所以它们包含 GPA 中的项。因此,必须将这些 GPA 地址转换为 HPA 地址以便正确地存储器 360 访问图形数据。与针对图 3 描述的概念相似的概念适用于图 1B 和 6B 的 MC 215 和控制器 118(例如环境 200)。

[0054] 虽然图 3 和图 6 示出可满足高级调度器功能性(例如 Microsoft 高级调度器的功能性)的转换表 264、266、268、272、274 和 278,但是本文针对这些表描述的概念也可以应用于多种其他转换表功能性或标准。

[0055] 根据实施例,DMA 重新映射规范可以使用多级页表遍历(walk)以将 GPA 地址转换为 HPA。遍历所需的级数取决于需要支持的地址空间的大小。例如,对于 32 位,需要 3 级页表遍历。对于支持 39+ 位地址空间的较大系统,需要 4 级遍历。例如,图 4 是示出根据一个实施例的直接存储器访问(DMA)重新映射 3 级遍历的框图。图 4 示出包括图形设备 116 的环境 500,图形设备 116 发送要通过 3 级遍历 580(可称为表 580)转换或重新映射的 DMA 请求 510。在此示例中,请求 510 包括 GPA 512 和请求方 ID 514(例如总线、设备、功能)。GPA512 可以表示虚拟存储器地址,例如从应用 132 接收的地址。上下文项将请求映射到要使用的适合页表或转换。

[0056] GPA 512 具有 32 位,可分成 2,+9,+9,+12 位查询。第一次查询,查询 532 使用 GPA 512 的前 2 位在页 542 中查询。第二次查询,查询 534 使用随后 9 位在页 544 中查询。第三次查询,查询 536 在页 546 中查询随后 9 位。余下的查询,查询 538 在 4 千字节的页 550 中查询余下的 12 位。可以认识到,位于每个中间级的查询 538 的 12 位使得页 550 具有 4 千字节或以下的大小。3 级遍历 580 可以设计成以 4 千字节页的粒度来提供重新映射,同时使得多个页表本身能够在 4 千字节的页内。例如,遍历 580 可以转换数据 370-378 或它的块的 32 位虚拟地址或 GPA,正如参考上文描述的。

[0057] 在一些实施例中,3 级遍历 580 可以在系统初始化或引导或加载应用时由 VMM 120(例如通过构建页表 542、544、546 和 550 来)创建或构建。也可考虑,可以在例如实现环境 100 的计算系统(例如计算设备)和/或 CPU 101(例如根据 BIOS 103)的引导或初始化期间由 MC 115 和/或 VMM 创建表 580。与针对图 4 描述的概念相似的概念适用于图 1B 和 6B 的 MC 215 和控制器 118(例如,环境 200)。

[0058] 可以认识到,除了当前图形和物理地址变换(例如表 264、266 和 268)之外,执行 3 级 DMA 重新映射遍历(在路径 410 和 420 中是必需的情况中)可能在这些路径中带来严重的延迟问题。例如,图 5 是示出根据一个实施例的在应用 3 级遍历的本发明实施例之前虚拟化环境中的图形数据路径的框图。图 5 示出包括显示引擎 219 的环境 600,在路径 410 中,显示引擎 219 将显示请求 602 传送到全局 GTT 264,全局 GTT 264 则将 GPA 地址 612 输出到 3 级遍历 580 中,3 级遍历 580 将 HPA 转换(translation)614 传送到存储器 360。图 5 还示出在路径 420 中,渲染引擎 217 将渲染请求 604 发送到页目录 266,页目录 266 则将 GPA 地址 622 输出到 3 级遍历 580 中,3 级遍历 580 将 HPA 转换 624 输出到每个进程 GTT 268,每个进程 GTT 268 将 GPA 地址 626 输出到 3 级遍历 580,3 级遍历 580 将 HPA 转换 628 发送到存储器 360。

[0059] 又如图 2 所示,设备/控制器 316 没有存储器 360 的物理地址的真实视图。因此,必须将图形 VM150 编程的 GPA 请求转换或变换为 HPA 以便正确地访问位于存储器 360 的 HPA 362 处的图形数据。

[0060] 更具体地说,在图 5 中,显示请求 602 可以是使用从应用 132 发送到全局 GTT 264 或在 GTT 264 中查询的一组图形地址的 DMA 请求。然后, GTT 264 输出或传送 GPA 转换 612(此处,因为图形 VM 150 没有存储器的真实视图,所以认为 GPA 是存储器的 HPA)。接下来,3 级遍历 580 将转换 612 变换成 HPA 614,并将转换 614 发送或传送到存储器 360。注意除了表 264 处所需的转换外,路径 410 还在遍历 580 处包括附加的 3 级遍历。

[0061] 而且,渲染请求 604 可以是使用从应用 132 发送到页目录 266 或在页目录 266 中查询的图形地址的 DMA 请求。然后,目录 266 发送或传送 GPA 转换 622 以指向下一级表(图形 VM 150 也创建具有 GPA 项的表)。但是,这是 GPA,所以可以将其转换为 HPA。接下来,3 级遍历 580 将转换 622 变换成 HPA 转换 624,并将转换 624 发送或传送到每个进程 GTT 268。每个进程 268 传送或发送 GPA 转换 626。接下来,3 级遍历 580 将转换 626 变换成 HPA 转换 628,并将转换 628 发送或传送到存储器 360。注意除了表 266 和 268 处所需的转换外,路径 420 还需要 6 个附加级的遍历。

[0062] 与针对图 5 描述的概念相似的概念适用于图 1B 和 6B 的 MC 215 和控制器 118(例如,环境 200)。例如,可以由引擎 229 和 227 分别生成请求 602 和 604。

[0063] 当前实现是按 Intel 公司的 VT-d 技术来定义的。一些实施例使用多级表来将地址从 GPA 转换为 HPA(例如为了转换 32 位地址,可以使用 3 级表)。可以认识到,图形设备或控制器可以是具有严格延迟要求的高性能设备。未能满足延迟要求可能导致显示撕裂和失帧或渲染丧失,从而在监视器或显示器上显示的图像中造成用户可见的缺陷。因此,映像表可以提供将 GPA 转换为 HPA 的更有效率的方式。

[0064] 根据一些实施例,可以组合图 5 的表和遍历以消除或减少多级遍历,并提供将产生高性能的实现。例如,可以用单级重新映射表来替代每个进程 GTT 268 与存储器 360 之间的 3 级遍历。而且,可以创建一个映像表,以在该表与存储器 360 之间将全局 GTT 264 与 3 级遍历 580 组合。类似地,可以创建另一个映像表,以在页目录 266 与表 268 之间将页目录 266 与 3 级遍历组合。因此,这些映像表可以将路径 410 和路径 420 中分别从引擎 219 和 217 接收的虚拟地址(图形地址)直接变换到 HPA。

[0065] 根据实施例,环境 100 可以用于创建转换表的项以便将来自应用 132 和/或 142

的用于访问图形数据的请求（例如 DMA 请求）的虚拟存储器地址或访客物理地址（GPA）转换为存储器 102 中存储或具有该图形数据的主机硬件物理存储器地址（例如，HPA）。例如，图 6A 是示出根据一个实施例的 VMM 环境中的集成图形设备的框图。图 6A 示出耦合在 CPU 101、图形控制器 116、输入 / 输出控制器 211（例如 IO 存储器控制器）和存储器 260 之间的存储器控制器 115。存储器 260 可以是与存储器 102 和 / 或存储器 360 相同的存储器，其可以是主存储器，可以是 RAM 存储器或可以是其他存储器。

[0066] 图 6A 示出包括图形设备 116 的 MC 115，图形设备 116 具有渲染引擎 217 和显示引擎 219。渲染引擎 217 可以用于渲染图形数据，而显示引擎 219 可以用于显示（已经由渲染引擎渲染的）数据。而且，引擎 217 和 219 可以使用渲染数据路径和显示数据路径，其已经包含用于将图形地址变换为物理地址的转换表。在一些情况中，引擎 217 和 219 可以使用或包括存储在存储器 260 中的一个或多个转换表。引擎 217 和 219 也可以表示引擎 317 和 319。

[0067] 例如，存储器 260 可以包括图形 VM 分配的存储器 262 和 / 或预先分配的映像存储器 270。存储器 262 包括诸如访客全局访客转换表（GTT）264（可以称为表 264）、访客每个进程页目录（PPPD）266（可以称为表 266）和每个进程 GTT 268（可以称为表 268 和 / 或 PPGTT）的表。表 264、266 和 268 可以用于将图形地址转换为访客物理地址。存储器 262 中存储的表可以例如在实现环境 100 的计算系统和 / 或 CPU 101（例如，根据 BIOS 103）的引导或初始化期间由图形驱动程序来创建或构建。存储器 262 中存储的表也可以在加载或执行应用 132 或 142 中的一个应用期间由图形驱动程序创建。在一些情况中，“加载”和“启动”可以描述相同的功能或发生的情况，例如当计算机程序被处理器启动并加载到主存储器中时。类似地，有时，“执行”和“运行”可以描述相同的功能或发生的情况，例如执行被处理器加载到存储器中的计算机程序以便执行根据该程序的功能（例如包括显示和渲染图形数据）时。此类驱动程序（driver）可以是图形 VM 150 的一部分，例如可以是软件、应用程序、应用代码。而且，此类驱动程序可以是存储在机器可访问介质中的指令，这些指令可以被机器（例如处理器）执行以使得机器执行本文针对这些驱动程序描述的功能，和 / 或创建和 / 或使用本文描述的表（例如映像表）。此类驱动程序还可以包括来自应用要求硬件执行数据的处理和 / 或显示的命令。例如，这些驱动程序可以使 MC 115、MC 215、设备 116、控制器 118 和 / 或存储器 260 执行本文描述的功能。

[0068] 例如，存储器 270 可以包括诸如映像全局 GTT 272（可以称为表 272）、映像访客 PPPD 274（可以称为表 274）和 / 或单级直接存储器访问（DMA）重新映射 278（可以称为表 278）的多种映像转换表。在（例如根据 DMA 请求）渲染和 / 或显示图形数据期间，表 272、274 和 278 可以分别替代表 264、266 和 268 或替代表 264、266 和 268 的功能。在一些情况中，表 272、274 和 278 中的仅一个或两个表可以存在和 / 或用于显示图形。根据一些实施例，存储器 260 可以包括仅存储器 262、仅存储器 270 或存储器 262 和存储器 270。

[0069] 用于存储器 270 的表（例如无论是否连续的表 272、274 和 / 或 278）的存储器存储位置、地址和 / 或空间可以在例如实现环境 100 的系统和 / 或 CPU 101 的引导或初始化期间在存储器 260 中予以预留或预先分配（例如根据 BIOS 103 而不向 VMM 报告所分配的存储器）。具体地说，可以在实现环境 100 的系统和 / 或 CPU 101 的引导或初始化期间由 BIOS 103 和 / 或 VMM 120 在存储器 260 中预留或预先分配用于存储器 270 的表的空间。

[0070] 根据一些实施例,在引导期间,BIOS 可以预留足够的空间来存储图形地址至 HPA 转换表,例如映像全局 GTT 272,并且预留足够的存储量来存储用于将 PPPD 项预先转换为 HPA 的转换表(例如表 274)。预先转换的表 274 确保在查询 PPPD 项(例如映像访客 PPPD 274)之后 无需联机重新映射遍历。可以约束表 274 的大小以支持特定数量的应用(例如,多达 100、200、300、400、1000 或其任意组合个应用)。下文参考图 9 进一步描述创建表 278。类似地,在引导或初始化 VM 应用(例如应用 132)的图形程序期间,BIOS 可以包括用于预留足够空间以存储单级 DMA 重新映射表(例如重新映射 278)的项,其中单级 DMA 重新映射表(例如重新映射 278)用于将数据从 GPA 转换为 HPA。

[0071] 例如,表 278 可以执行与将 GPA 转换为 HPA 的表的操作或功能相似的操作或功能,例如与图 4 的 3 级遍历 580 所执行的功能相同的功能。因此,重新映射 278 可以是单级重新映射表(例如,无法包含在 4 千字节页内的表),并且可以位于连续的存储器中。具有连续表(例如单级查询表)避免了遍历多级表的各个级所引起的延迟,从而在渲染和 / 或显示图形数据期间提供更快且更有效率的 GPA 至 HPA 的转换或变换的益处。连续存储器可以描述为存储器中有共同的边缘或边界、接触、相邻、邻接、无间断地连接和 / 或直接相邻的地址或位置。例如,存储器中的连续地址可以是一个接一个的地址。或者,存储在具有不连续次序的位置中的地址或位置,或不包含表数据的部分的位置可以描述为非连续存储器。可设想,表 272、274 可以位于或存储在连续存储器中(例如,可将这些表的其中一个或多个都存储在存储器 260 的连续存储器位置中)。表 278 也可以位于与表 272 或 274 分离或不连续的存储器位置或地址中。在一些情况中,存储表 272 的存储器地址与存储表 274 的存储器地址相邻,但是用于表 272 以及表 274 的存储器地址都不与用于表 278 的存储器地址相邻。表 278 的大小将取决于图形 VM(例如,设备 116)的图形地址空间。例如,表 278 可以对应于每个 4 千字节页包含 8 千字节的项,并且因此表 278 可能需要存储器 270 中 8 兆字节连续表以提供 4 吉字节图形地址空间的转换。用于存储器 270 的所有表的空间可以由系统 BIOS 来分配,而不向 VMM 或 VM 报告,正如上文提到的。

[0072] 存储器 270 还可以包括表 278 表示的 GPA 至 HPA 转换表。表 278 可以是单级、2 级、3 级或 4 级表。表 278(例如表 278 表示图 4 的 3 级遍历 580 的情况)可以在实现环境 100 的计算系统和 / 或 CPU 101(例如根据 BIOS 103)的引导或初始化期间由 VMM 来创建(例如,无需处理或访问 MC 115 或 215)。

[0073] 表 278 可以例如在实现环境 100 的计算系统和 / 或 CPU 101(例如,根据 BIOS 103)的引导或初始化期间由 VMM 来创建或构建。表 272 和 274 也可以由硬件(例如 MC 115 或 215)来创建。在一些情况中,在 MC 115 或 MC 215 的电路或硬件(例如检测电路)检测到或感测到由图形 VM 150 创建的对应表 264 或 266(例如非映像表)的项被创建时创建表 272 或 274(例如映像表)的项。根据一些实施例,MC 115 或 MC 215 的电路或硬件感测到表 264 的非映像项被创建,则创建表 272 的对应项或关联项(例如在表 272 中与表 264 中的项或对表 264 的更新相同的位置或地址设置转换的 HPA 项)。同样地,在一些情况中,MC 115 或 MC 215 的电路或硬件感测到表 266 的非映像项被创建,则创建表 274 的对应项或关联项(例如在表 274 中与表 266 中的项或对表 266 的更新相同的位置或地址设置转换的 HPA 项)。

[0074] 还可以在实现环境 100 的计算系统的操作期间(例如在诸如 VM 应用或应用 132

之一的应用（例如，图形应用或包括将图像显示在显示器或监视器上的图形指令的应用）启动、加载、初始化和 / 或执行期间）更新表 264、266、268、272、274 和 / 或 278。例如，图形 VM 150 和 / 或图形 VM 150 的驱动程序可以在应用 132 之一被启动时在存储器 270 中更新表 272、274 和 / 或 278。

[0075] 在一些情况中，在 MC 115 或 MC 215 的电路或硬件感测到图形 VM 150 更新对应表 264 或 266 的项（例如，创建了新的项或盖写表 264 或 266 非映像表的现有项）时，表 272 或 274（例如映像表）的项将被更新。根据一些实施例，MC 115 或 MC 215 的电路或硬件感测到表 264 中现有的非映像项被更新，则更新表 272 的对应项或关联项（例如在表 272 中与对表 264 的更新相同的位置设置转换的 HPA 项）。例如，可以在存储器中从基地址（例如，1000）将表 264 中的项偏移相同的地址量（例如，+5）（例如，将处于 1005），而在存储器中从不同的基地址（例如，2000）将表 272 中的项偏移相同的地址量（例如，+5）（例如将处于 2005）。同样地，在一些情况中，MC 115 或 MC 215 的电路或硬件感测到表 266 中现有的非映像项被更新，则更新表 274 的对应项或关联项（例如在表 274 中与对表 266 的更新相同的位置设置转换的 HPA 项）。因此，表 274 和 266 可以采用相同索引号编索引。

[0076] 在一些情况中，图形 VM 150 可以包括高级调度器，例如包括美国华盛顿州雷德蒙德的 Microsoft 公司（Redmond, Washington）的 Microsoft 高级调度器的功能性或部分功能性的高级调度器。例如，当应用被启动时，利用高级调度器体系结构编译的图形 VM 可以创建、存储和 / 或更新表 264、266、268。转换表 264、266、268、272、274 和 / 或 278 可以满足高级调度器的功能性，例如 Microsoft 高级调度器功能性。还可以设想，转换表 264、266、268、272、274 和 / 或 278 可以满足不同于高级调度器的功能性的本领域中公知的另一种标准。

[0077] 根据一些实施例，图形 VM 150 可以不使用或不访问 VMM 120 来访问图形设备 116，而系统 130 和 140 通过 VMM 120 来访问图形设备 116。确切地说，MC 115 可以使用表 272 的项将应用所使用的虚拟地址（图形地址）转换为图形数据在主存储器中的主机硬件物理存储器地址（例如，HPA）。类似地，MC 115 可以使用表 274 创建表 274 的项，以便从每个进程页目录的虚拟地址转换为图形数据在主存储器中的主机硬件物理存储器地址（例如，HPA）。

[0078] 虽然图 1A 和 6A 参考集成图形设备来描述实施例，但是本文描述的这些概念也适用于各种其他图形设备、芯片组、电路、硬件、渲染路径和 / 或显示路径。例如，分离图形路径可以使用与环境 100 的集成路径相似的机制。分离图形路径可以是能够通过发出具有访客物理地址（GPA）的 DMA 请求且无需访问 VMM 和 / 或无需访问图形 VM 而使图形控制器和 / 或存储器控制器将 GPA 转换为主存储器的主机物理地址（HPA）或物理存储器地址来访问图形数据的路径。

[0079] 根据实施例，环境 200 可以用于创建转换表的项以便将来自应用 132 和 / 或 142 的用于访问图形数据的请求（例如 DMA 请求）的虚拟存储器地址或访客物理地址转换为该图形数据存储或位于存储器 102 中的主机硬件物理存储器地址（例如，HPA）。例如，图 6B 是示出根据一个实施例的虚拟机监视器（VMM）环境中的分离图形设备的框图。图 6B 示出耦合在图形控制器 118、CPU 101、输入 / 输出控制器 211（例如 IO 存储器控制器）和存储器 260 之间的存储器控制器 215。存储器 260 可以是与存储器 102 相同的存储器和 / 或执

行与存储器 102 相同的功能。存储器 260 也可以是主存储器,可以是 RAM 存储器或其他存储器。

[0080] 图 6B 示出耦合到图形控制器 118 的 MC 215,图形控制器 118 具有渲染引擎 227 和显示引擎 229。渲染引擎 227 可以用于渲染图形数据,而显示引擎 229 可以用于显示(已经由渲染引擎渲染的)数据。而且,引擎 227 和 229 可以使用渲染数据路径和显示数据路径,其已经包含用于将图形地址变换为物理地址的转换表。在一些情况中,引擎 227 和 229 可以使用或包括存储在存储器 260 中的一个或多个转换表。渲染引擎 227 和显示引擎 229 可以执行与本文针对渲染引擎 217 和显示引擎 219 所描述的功能相同的功能,所例外的是渲染引擎 227 和显示引擎 229 是环境 200 中的分离控制器的一部分,并作为分离的渲染引擎和分离的显示引擎与 MC 215 交互或通信。引擎 227 和 229 也可以表示引擎 317 和 319。

[0081] 图 6B 的其他组件可在本文中参考图 1A 和图 6A 来描述,所例外的是它们处在环境 200 中,并且(例如经由 MC 215)与控制器 118、渲染引擎 227 和显示引擎 229 交互或通信,而非与设备 116、渲染引擎 217 和显示引擎 219 交互或通信。

[0082] 根据实施例,环境 200、图形 VM 150、VMM 120、硬件 110、CPU 101、存储器 102、BIOS 103、系统 130 和 140 以及 MC 215 可以预留存储器空间用于创建、更新和使用表 264、表 266、表 268 映像全局 GTT 272、映像访客 PPPD 274 和 / 或单级直接存储器访问 (DMA) 重新映射 278,正如本文参考环境 200 和图 1A 和图 1B 的那些表的创建所描述的。

[0083] 根据一些实施例,图形 VM 150 可以不使用或不访问 VMM 120 来访问图形设备 118,而系统 130 和 140 通过 VMM 120 来访问图形设备 118。一般来说,图形控制器期望能够(例如,这对于系统和 / 或图形路径更具效率)接收来自图形 VM 150(例如包含 GPA 的指令)的命令指令(例如图形指令),以及访问存储器中的数据(例如图形数据),而无需 VMM120 来提供 GPA 至 HPA 的转换。例如,设备 118 可以是分离图形控制器,例如无需访问 VMM 120 而与操作系统 130 和 / 或 140 通信的图形控制器。为了实现此目的,控制器 118 和 / 或 MC 215 可以包括用于使用转换表的项来从 GPA 转换为 HPA 的电路、硬件、软件和 / 或逻辑。确切地说,控制器 118 和 / 或 MC215 可以使用表 272 的项以在图形转换表中将应用所使用的虚拟地址(图形地址)转换为图形数据在主存储器中的主机硬件物理存储器地址。类似地,控制器 118 和 / 或 MC215 可以使用表 274 创建表 274 的项,以便从每个进程页目录的虚拟地址(例如 GPA)转换为图形数据在主存储器中的主机硬件物理存储器地址。

[0084] 如图 1A、1B、6A 和 6B 所示,环境 100 或 200 允许在 IO 设备直接被指定给 VM 时将输入 / 输出设备(例如设备 112、114 或 116)直接指定给虚拟机(例如,系统 130、140 或 150),这样可以将该设备的设备驱动程序移出 VMM 120 而移入到控制 VM 中。例如,通过将图形设备 116 或控制器 118 直接指定给图形 VM 150,可以将通常位于 VMM120 中的所有图形驱动程序(例如驱动程序 122 的驱动程序)移入到图形 VM150 中。但是,因为 VM 具有存储器的虚拟化视图(例如图形 VM 150 具有存储器 102 的虚拟化视图),所以这些设备驱动程序或 VM(例如图形 VM 150)中的驱动程序使用 GPA 来对 IO 设备(例如,设备 116 和 / 或控制器 118)编程。为了获取对正确数据的访问权,可能需要 MC 115 或 215(例如,图形芯片组的一部分)将 IO 设备发出的地址从 GPA 转换为 HPA。更具体来说,由于存储器的“虚拟化视图”,所以每个部分或 VM 可以发送含有 GPA 的 DMA 请求。但是,因为图形设备(例如,设备 116 和 / 或图形控制器 118)具有“虚拟化视图”(例如,物理存储器地址的视图),为

了对于图形设备来说能够正确满足请求,所以必须将 GPA 转换为 HPA。

[0085] 可以在设备 116 或控制器 118 经由具有 GPA 的图形 VM 150 从系统 130 接收到 DMA 请求时由 MC 115 或 215 来执行此转换。这些 DMA 请求的地址被 MC 115 或 215 转换以指向表 272 或 274 中的存储器地址,而该存储器地址与 DMA 请求在表 264 或 266 中所指向的地址相对应。因此,并不是使用表 264 或 266 和 3 级遍历(或其他 GPA 至 HPA 转换表)来将 GPA 变换或转换为 HPA,而是使用表 272 或 274 将 DMAGPA 转换为 HPA。这样通过渲染和/或显示引擎可以更快速且更有效率地在存储器中访问图形数据。

[0086] 具体地说,因为图形 VM150 通过 MC 115 或 215 来访问存储器 260,所以图形 VM 150 具有存储器 260 的物理地址的访客视图。因此,仍必须由存储器控制器、图形设备和/或二者的组合将 DMA 请求的 GPA 转换或变换为 HPA,以便正确地从存储器 260 访问图形数据。

[0087] 无论何时只要创建或更新访客表(264、266),就对映像表的项进行 DMA 重新映射(例如从 GPA 映射到 HPA),因此映像表的项包含 HPA 项。例如,图 7 是示出根据一个实施例的用于创建映像全局 GTT 的环境的框图。图 7 示出的环境 700 具有图形 VM 150,该图形 VM 150 尝试将 GPA GTT 项安装到访客 GTT 中。例如,图形 VM 150 可以安装项 710(例如,包含作为来自应用 132 的 DMA 请求要使用的数据的偏移量和图形地址)。使用在(偏移量、GPA 数据)712 中提供的 GPA 数据,在(偏移量、GPA 数据)712 所指定的位置处创建、存储、安装或更新表 264。

[0088] 因为图形 VM 150 没有存储器 260 的物理地址的真实视图,所以每次当来自应用 132 的 DMA 请求要访问该地址处(例如在 GPA 712 处)的数据时需要将表 264 的(偏移量、GPA 数据)712 的数据部分(例如,(偏移量、GPA 数据)712 包含偏移量和 GPA 作为数据)转换或变换为 HPA,以便正确地从存储器访问图形数据。并不是一一执行所述无数种变换,而是可以通过将(偏移量、GPA 数据)712 转换或变换为适合的 HPA,将该转换存储在映像表中的表内相同的对应位置处从而建立更有效率的系统。每次由路径 410 处理来自应用 132 的 DMA 请求时,并不是在表 264 中执行查询(例如变换或转换)然后在表 580 中执行另一次查询,而是执行映像表中的单级查询以便更快速且更有效率地将 DMA 请求变换为 HPA 地址。

[0089] 因此,环境 700 示出由 MC 115 转换或变换为更新的(偏移量、HPA 数据)714 数据部分的(偏移量、GPA 数据)712 被转换,然后存储在映像全局 GTT 表 272 中的表内相同的对应位置处。因此,从图看出(偏移量、GPA 数据)712 发送到两个位置或被 MC 115 “复制(forked)”以更新表 264 和表 272。MC 115 可以包括或作为查询电路,例如执行此转换或变换的电路(还参见图 8)。现在,在运行时(例如 VM 上执行图形应用期间),每次由路径 410 处理来自应用 132 的 DMA 请求时,并不是在表 264 中执行查询(例如变换或转换),然后在表 580 中执行另一次查询,而是执行映像表 272 中的单级查询以便更快速且更有效率地将 DMA 请求变换为 HPA 地址。确切地说,映像表 GTT 272 现在可以用于执行单级查询来将 DMA 请求的图形地址变换或转换为(HPA 数据)714(例如通过查询映像表中的对应地址以获取(HPA 数据)714)。有关更多解释,参见图 12。

[0090] 此外,图 8 是示出根据一个实施例的用于创建映像全局 GTT 的过程的流程图。图 8 示出过程 800,例如可以对应于环境 700 和/或以其他方式创建映像全局 GTT 的项的过程。图 8 示出过程 800,包括框 810,其中 Gfx VM 尝试将 GPA GTT 项安装到访客 GTT 中。例如,

图形 VM 150 可以（例如从图形应用或设备驱动程序）接收或访问图形地址项并将其变换为全局 GTT 的 GPA 项。框 810 可以包括通过图形 VM 或操作系统接收或访问项 710，如上文对于图 7 描述的那样。框 810 还可以包括上文参考图形 VM 150 由项 710 创建、转换、变换或构建全局 GTT 264 的（偏移量、GPA 数据）712 的描述。在框 820 处，将 GfxVM 提供（例如访问和变换）的 GPA 项存储在访客 GTT 中，例如上文对于（偏移量、GPA 数据）712 和表 264 描述的那样。

[0091] 接下来，在框 830 处，检测到（例如，访问或复制）Gfx VM 安装的 GPA 项，并将其转换为 HPA 项（例如，将 GPA 数据部分转换为 HPA，但是（偏移量、GPA 数据）712 的索引、偏移量或位置不变）。框 830 可以包括 MC115 检测到由 Gfx VM 150 或其他部件（例如，由在 GTT 的主存储器中存储项的其他实体）向全局 GTT 264 添加了项。框 830 还可以包括 MC 115 使用 3 级遍历 580（或另一种 GPA 至 HPA 转换，例如单级重新映射 278）将（偏移量、GPA 数据）712 转换为（HPA 数据）714，正如上文对于图 7 描述的那样。在框 830 处的转换可以描述为（例如使用重新映射表 278）将数据从 GPA 转换为 HPA 的 DMA 重新映射。

[0092] 可以在环境 100 引导或初始化期间或在应用 132 或 142 之一加载或执行期间安装项 710 和 / 或（偏移量、GPA 数据）712。类似地，MC115 可例如使用软件、固件、逻辑、电路、诱捕、停止和 / 或适合的设备或进程，检测或感测到在全局 GTT 264 的存储器中安装、创建或存储了（偏移量、GPA 数据）712。虽然框 830 可以包括通过 3 级遍历的转换，并且图 7 示出 3 级遍历 580，但是还可以使用其他适合的 GPA 至 HPA 转换（例如单级遍历，如表 278）。

[0093] 在框 840 处，将 HPA 项存储在映像全局 GTT 中。框 840 可以包括将（HPA 数据）714 存储在映像全局 GTT 272 中的与访客 GTT 相同的对应位置中，正如上文对于图 7 所描述的那样。框 840 和 / 或图 7 可以包括存储例如创建或构建映像表的项，正如上文对图 6A 或 6B 的表 272 所描述的那样。

[0094] 与图 7-8 描述的概念相似的概念适用于图 1B 和 6B 的 MC 215 和控制器 118（例如，环境 200）。图 9 是示出根据一个实施例的用于创建每个进程页显示（PPPD）映像访客的环境的框图。图 9 示出的环境 900 具有图形 VM 150，该图形 VM 150 尝试将 GPA PPPD 项安装到访客 PPPD 中。例如，图形 VM 150 可以安装项 910（例如，包含偏移量和图形地址，其作为数据用于来自应用 132 的 DMA 请求）。可以将项 910 的偏移量和图形地址数据包含在请求 604 中。图形 VM 150 从项 910 创建、转换、变换或构建页目录 266 的（偏移量、GPA 数据）912。使用在（偏移量、GPA 数据）912 中提供的 GPA 数据，在（偏移量、GPA 数据）912 所指定的位置处创建、存储、安装或更新表 266。

[0095] 与上文对图 7 的描述相似，因为图形 VM 150 没有存储器 260 的物理地址的真实视图，所以每次当来自应用 132 的 DMA 请求要访问该地址处（例如在 GPA 712 处）的数据时，需要将（偏移量、GPA 数据）912 的数据部分转换或变换为 HPA，以便正确地访问图形数据。因此，同样地，在运行时并不是一一执行所述无数种变换，而是可以通过将（偏移量、GPA 数据）912 转换或变换为适合的 HPA，并将该转换存储在映像表中的表内相同的对应位置处从而建立更有效率的系统。每次由路径 420 处理来自应用 132 的 DMA 请求时，并不是在表 266 中执行查询然后在表 580 中执行另一次查询，而是使用映像表 274 执行单级查询以将图形地址变换或转换为 HPA。

[0096] 相应地，环境 900 示出（偏移量、GPA 数据）912 由 MC 115 转换或变换为（HPA 数

据)924,然后在表 274 中与表 266 中相同的对应位置处存储(例如,存储、安装或更新)(HPA 数据)924。从图看出(偏移量、GPA 数据)912 发送到两个位置或通过发送到表 266 来“复制”(偏移量、GPA 数据)912,以及检测、访问和/或将(偏移量、GPA 数据)912 发送 MC 115。MC 115 可以包括或作为查询电路,例如包括执行此转换或变换的电路(还参见图 10)。每次由路径 420 处理 DMA 请求时,并不是在表 266 中执行查询然后在表 580 中执行另一次查询,而是在映像表 274 中执行单级查询以更快速且更有效率地将图形地址变换或转换为 HPA。有关更多解释,参见图 12。

[0097] 此外,图 10 是示出根据一个实施例的用于创建映像访客 PPPD 的过程的流程图。图 10 示出过程 1000,例如可以或可以不对应于环境 900 和/或以其他方式创建映像访客 PPPD 的项的过程。图 10 示出过程 1000 包括框 1010,其中 Gfx VM 尝试将 GPA PPPD 项安装到访客 PPPD 中。例如,图形 VM150 可以(例如从图形应用或设备驱动程序)访问图形地址项并将其变换为访客 PPPD 的 GPA 项。框 1010 可以包括图形 VM 或操作系统接收或访问项 910,如上文对于框 810 和/或图 9 描述的那样。框 1010 还可以包括 VM 150 由项 910 创建、转换、变换或构建表 266 的(偏移量、GPA 数据)912,如上文所描述。在框 1020 处,将 Gfx VM 提供(例如访问和变换)的 GPA 项存储在访客页目录中,例如上文对于(偏移量、GPA 数据)912 和表 266 描述的那样。

[0098] 接下来,在框 1030 处,检测到(例如,访问或复制)Gfx VM 安装的 GPA 项,并将其转换为 HPA 项(例如,将 GPA 数据部分转换为 HPA,但是(偏移量、GPA 数据)912 的索引、偏移量或位置不变)。框 1030 可以包括 MC 115 检测到向页目录 266 添加了项和/或检测到由 GfxVM 150 或其他部件(例如,由在页目录的主存储器中存储项的其他实体)向存储器写入或输入了项。框 1030 还可以包括 MC 115 使用 3 级遍历 580(或另一种 GPA 至 HPA 转换,例如单级重新映射 278)将(偏移量、GPA 数据)912 转换为(HPA 数据)924,正如上文对于图 9 描述的那样。在框 1030 处的转换可以描述为(例如使用重新映射表 278)将数据从 GPA 转换为 HPA 的 DMA 重新映射。

[0099] 可以在环境 100 引导或初始化期间或在应用 132 或 142 之一加载或执行期间由图形 VM 150 安装(例如访问)项 910 和/或(偏移量、GPA 数据)912。类似地,MC 115 可例如使用软件、固件、逻辑、电路、诱捕、停止和/或适合的设备或进程,检测或感测到在页目录 266 的存储器中插入、创建或存储了(偏移量、GPA 数据)912。虽然框 1030 可以包括使用 3 级遍历的转换,并且图 9 示出 3 级遍历 580,但是还可以使用其他适合的 GPA 至 HPA 转换(例如单级遍历,如表 278)。

[0100] 在框 1040 处,将 HPA 项存储在映像访客 PPPD 中。框 1040 可以包括将(HPA 数据)924 存储在映像访客 PPPD 274 中的映像表内相同的对应位置中,正如上文对于图 9 所描述的那样。框 1040 和/或图 9 可以包括存储例如创建或构建映像表的项,正如上文参考图 6A 或 6B 的表 274 所描述的那样。

[0101] 根据一些实施例,环境 900 和/或过程 1000 可以包括用于创建映像访客 PPPD 的框,而非用于创建单级 DMA 重新映射的框。或者,一些实施例可以包括创建单级 DMA 重新映射,但不创建映像访客 PPPD(例如,图 6A、6B 和/或图 4 中所描述的,而非图 9-10 中所示那样)。在一些实施例中,在框 820 和 1020 处由 Gfx VM 将 GPA 项转换为映像全局 GTT 和/或映像访客 PPPD 也是可选的。此外,在一些情况中,不存在非映像表。MC 感测到对表进行写

入的尝试,可以由 MC 和 / 或以虚拟方式(例如非映像表是指示成功写入而不存储数据的虚拟表)来提供成功写入的指示。在一些情况中,对不存在的表写入可以是对 MC 写入。

[0102] 与图 9-10 描述的概念相似的概念适用于图 1B 和 6B 的 MC 215 和控制器 118(例如,环境 200)。在此类情况中,MC 215 可以包括或作为查询电路,例如执行此转换或变换的电路。

[0103] 图 11 是用于维护或更新映像表的环境的框图。图 11 示出的环境 1100 包括图形 VM 150、存储器 260 和映像存储器 270。图形 VM 150 可以创建或产生全局 GTT 更新 1110 和 / 或进程页目录更新 1120。更新 1110 和 / 或 1120 可以分别与(偏移量、GPA 数据)712 和 / 或(偏移量、GPA 数据)912 相似,不同的是更新 1110 和 1120 可以在环境 100 的计算系统引导或初始化之后发生。更新 1110 和 1120 还可以描述在任何时间更新表中已经存在的项。更新 1110 和 1120 可以例如在将它们发送、写入或以其他方式用于更新存储器 260 中的表 264 或 266 时被 MC 115 感测或检测到。

[0104] 根据实施例,MC 115 使用单级重新映射来创建要写入或输入到存储器 270 的表 272 和 / 或 274 中的映像表更新或项。例如,单级重新映射可以使用例如单级 DMA 重新映射 278 的表将 GPA 地址转换为 HPA 地址。或者,在其他实施例中,响应到检测后,MC 115 使用多级遍历 580 来创建要写入或输入到存储器 270 的表 272 和 / 或 274 中的映像表更新或项(根据一些实施例)。

[0105] 类似地,VMM 120 可以提供要发送到或写入存储器 260 中的表 268 中的项 1125。根据实施例,硬件可以不对每个进程 GTT 表建立映像,从而这些表可以保留在 GPA 中。因此,在一些情况中,在访问存储器之前,需要使用 3 级遍历 580 转换项 1125 以创建 HPA 地址。此过程可以与上文对图 6A、图 6B 和 / 或图 4 的每个 GPA 至 HPA 表的创建所描述的过程相似。

[0106] 与图 11 描述的概念相似的概念适用于图 1B 和 6B 的 MC 215 和控制器 118(例如,环境 200)。

[0107] 图 12 示出根据一个实施例的图形渲染和显示数据路径的框图。图 12 示出的环境 1200 包括路径 1210(例如显示路径)和路径 1220(例如渲染路径)。路径 1210 可以是与图 5 的路径 410 相似的路径,不同的是路径 1210 使用表 272 来替代表 264 和遍历 580。同样地,路径 1220 可以是与图 5 的路径 420 相似的渲染路径,不同的是路径 1220 使用表 274 来替代目录 266 和遍历 580,并且可以使用重新映射 278 替代遍历 580。环境 1200 包括(例如图形设备 116 的)渲染引擎,该渲染引擎将渲染的请求 604 发送到映像访客 PPPD 274。通过表 274 将请求 604 转换为转换 1224。

[0108] 可以理解到,使用表 274 来执行单级转换,远比使用图 3、图 6 或图 9 的目录 266 和遍历 580 更有效率。1224 处的转换可以对应于图 5 的 HPA 624 或使用图 9 的(HPA 数据)924 的转换的请求或表 274 的项。转换 1224 还可以对应于图 7 的转换 714。可以由 MC 115 使用存储器 270 来执行使用表 274 的转换,正如图 6A 或 6B 所示那样。这样 1224 处的转换可以是发送到每个进程 GTT 268 的查询。在表 268 中,所述查询提供 GPA 1226。GPA 1226 可以对应于图 5 的 GPA 626 或图 9 的 924 或表 268 的项。然后将转换 1226 发送到单级 DMA 重新映射 278。由重新映射 278 将 GPA 1226 转换为转换 1228。转换 1228 可以是或对应于图 5 的 HPA 628 或通过表 268 或 278 转换的 HPA。可以由 MC 115 使用存储在存储器 270 中的单级 DMA 重新映射(例如重新映射 278)来执行 GPA 1226 转换为转换 1228 的转换,如图

6A 或 6B 所示那样。可以将转换 1228 发送到存储器 360 以用于例如图 2 的数据访问。

[0109] 可以理解到,使用重新映射 278 将 GPA 1226 转换为转换 1228 比上文使用图 5 的查询 580 将 GPA 626 转换为 HPA 628 的过程更有效率(例如,重新映射 278 是单级的或单次查询转换,而查询 580 是 3 级转换)。或者,在一些实施例中,可以使用除重新映射 278 之外的重新映射或转换来将 GPA 1226 转换为存储器 360 的适合转换。例如,可以使用 3 级遍历 580 来替代重新映射 278。

[0110] 然后,例如可以通过寻址图 2 描述的 HPA 362 处的图形数据 370 或 374 的方式,使用转换 1228 来访问存储器 360 中的数据(例如读取或写入图形数据)。

[0111] 在环境 1200 中,显示引擎(例如图形设备 116)也可以将显示请求 602 发送到映像全局 GTT 272。表 272 可以将请求 602 转换为转换 1214,例如对于图 3、图 5 或图 9 中使用表 272 所描述的那样。转换 1214 可以对应于图 5 的 HPA 614 或图 9 的(HPA 数据)924。转换 1214 还可以对应于使用图 7 的(HPA 数据)714 的转换。可以理解到,使用表 272 将请求 602 转换为转换 1214 可比图 5 中使用表 264 和查询 580 更有效率且更快速。可以由 MC 115 使用存储在存储器 270 中的表 272 来从请求 602 转换出转换 1214,正如图 6A 或 6B 所示那样。可以将转换 1214 发送到存储器 360 以用于例如图 2 的数据访问。

[0112] 表 272 和 274 可以在体系结构上对于 VM(例如,系统 130 或 140 或应用 132 或 142)和/或 VMM(例如,VMM 120)不可见(例如,不可见于或不显露给 VM 或 VMM)。根据实施例,表 278 可以在体系结构上总是对于 VMM(例如,VMM 120)可见(例如显露给 VMM)。表 272、274 和 278 可以对于诸如 MC 115 或 215、设备 116 和/或控制器 118 的硬件或芯片组可见。存储器中的表、项、数据或位置可视为对于 VM、VMM、电路、OS 和/或其他实体可见,如果存在下列情况:存储器中的表、项、数据或位置被报告给 VM、VMM、电路、OS 和/或其他实体、VM、VMM、电路、OS 和/或其他实体知道其存在、其被显露给 VM、VMM、电路、OS 和/或其他实体、VM、VMM、电路、OS 和/或其他实体可对其访问、其可写入到 VM、VMM、电路、OS 和/或其他实体、其可从 VM、VMM、电路、OS 和/或其他实体读取或可以其他方式对其操作。存储器中的表、项、数据或位置可视为对于 VM、VMM、电路、OS 和/或其他实体不可见,如果存在下列情况:存储器中的表、项、数据或位置未被报告给 VM、VMM、电路、OS 和/或其他实体、未使 VM、VMM、电路、OS 和/或其他实体知道其存在、未显露给 VM、VMM、电路、OS 和/或其他实体、VM、VMM、电路、OS 和/或其他实体不可对其访问、其不可写入到 VM、VMM、电路、OS 和/或其他实体、其不可从 VM、VMM、电路、OS 和/或其他实体读取或不可以其他方式对其操作。

[0113] MC 115 可以对于 VM 或 VMM 透明地使用表 272、274 和/或重新映射 278 来产生转换。MC 115 或 215 可以对于 VM 或 VMM 透明地创建表 272、274 和/或 278 的项或对表 272、274 和/或 278 更新项。可以由 MC 115 或 215 检测或感测对 GTT ADDR(例如全局转换表地址)窗口(aperture)的写入,并复制映像表项、更新或使用来透明地执行表 272、274 的创建、更新和使用。复制使用可以包括使用表 272、274 和/或 278。复制项或更新可以包括将项发送到表 272、274 和/或 278 或将项写入到表 272、274 和/或 278 或对表 272、274 和/或 278 更新。此外,转换项或数据、查询项或数据或以其他方式使用表 272、274 和/或 278 或显示或渲染图形数据可以包括,(例如使用基地址和/或偏移量地址)对表 272、274 和/或 278 中位于对应项在对应表 264、266 和/或 268 中所在地址或位置相同的位置或地址处的项或数据建立索引。

[0114] 与针对图 12 描述的概念相似的概念适用于图 1B 和 6B 的 MC 215 和控制器 118(例如,环境 200)。

[0115] 图 13 是示出根据一个实施例的图形显示路径更新和显示路径的框图。图 13 示出环境 1300,环境 1300 包括显示更新路径 1310 和显示路径 1320。路径 1310 可以对应于参考图 11 描述的环境 1100 的局部。其中示出路径 1310 包括全局 GTT 更新 1110,全局 GTT 更新 1110 被 MC 115 复制以向访客全局 GTT 264 发送一个项并向单级重新映射 278 发送另一个项。从单级重新映射 278,将更新发送到映像访客全局 GTT 272,例如以便更新表 272 的项。

[0116] 路径 1320 可以对应于图 12 的路径 1210 的描述,并且可以提供与图 3 的路径 410 相同的功能性。其中示出路径 1320 包括 MC 115 的显示引擎 219 将请求 602 发送到表 272。表 272 转换该请求,并将其作为转换 1214 发送到存储器 360。

[0117] 与针对图 13 描述的概念相似的概念适用于图 1B 和 6B 的 MC 215 和控制器 118(例如,环境 200)。

[0118] 图 14 是示出根据一个实施例的图形渲染路径更新和渲染路径的框图。图 14 示出环境 1400,环境 1400 包括渲染更新路径 1410 和渲染路径 1420。路径 1410 可以对应于图 11 的环境 1100 的局部。路径 1410 包括每个进程页目录更新 1120 进入 MC 115,MC 115 将此更新复制成用于访客每个进程页目录 266 的一个更新和发送到单级重新映射 278 的另一个更新。从单级重新映射 278,将更新的转换发送到映像每个进程页目录 274。更新 1120 可以对应于图 9 的更新(偏移量、GPA 数据)912。

[0119] 路径 1410 还包括 MC 115 接收并转发到每个进程 GTT 268 的每个进程 GTT 更新 1125。更新 1125 可以是与针对图 6A、图 6B 和图 4 描述的更新表 278 相似的更新。

[0120] 其中示出渲染路径 1420 包括,MC 115 的渲染引擎 217 将渲染请求 604 发送到表 274,表 274 将转换 1224 发送到表 268,表 268 将转换 1226 发送到表 278,表 278 将转换 1228 发送到存储器 360。路径 1428 可以对应于图 12 的路径 1220。

[0121] 与针对图 14 描述的概念相似的概念适用于图 1B 和 6B 的 MC 215 和控制器 118(例如,环境 200)。

[0122] 图形 VM(例如,图形 VM 150)、VMM 120、MC 115 和 / 或图形设备(例如,设备 116 和 / 或控制器 118)可以包括用于执行上文参考图 1-14 描述的功能的适合电子装置或计算机硬件和软件。还应该认识到,在不背离本发明范围的前提下,可将更多或更少的设备和 / 或过程结合到图 1-14 所示的设备和方法中,并且所示出和本文所描述的框的安排不暗示着有特定的次序。还将认识到,结合图 1-14 描述的方法和设备可以在机器可执行指令(例如软件)中实施。这些指令可以用于使得使用这些指令编程的通用或专用处理器执行所描述的操作和过程。或者,这些操作和过程可以由包含用于执行这些操作的硬连线逻辑的特定硬件组件来执行,或由编程的计算机组件与定制的硬件组件的任何组合来执行。这些方法可以作为计算机程序产品来提供,该计算机程序产品可以包括其上存储有可用于将计算机(或其他电子设备)编程来执行这些方法的指令的机器可读介质。为了解释本说明书的目的,术语“机器可读介质”还应该视为包括能够存储用于由机器执行并使该机器执行本发明任一方法的指令序列或将其编码的任何介质。因此术语“机器可读介质”应该视为包括但不限于固态存储器、光盘和磁盘以及载波信号。而且,在本领域中通常将软件以一种形式

或另一种形式（例如，程序、过程、进程、应用、模块、逻辑...）论述为执行动作或促成结果。此类表述仅是描述通过计算机执行软件使该计算机的处理 器执行动作或产生结果的便捷方式。在一些情况中，本文描述的环境可以在 PC 上实现，或者在计算机服务器（例如用作例如具有要显示的数据的一个或多个客户机计算机的服务器）上实现。

[0123] 在说明书前文中，描述了特定实施例。但是，在不背离所附权利要求中提出的本发明的更宽泛精神和范围的前提下可以对其进行多种修改和更改。因此，本说明书和附图应视为说明性意义的而非限定性意义的。

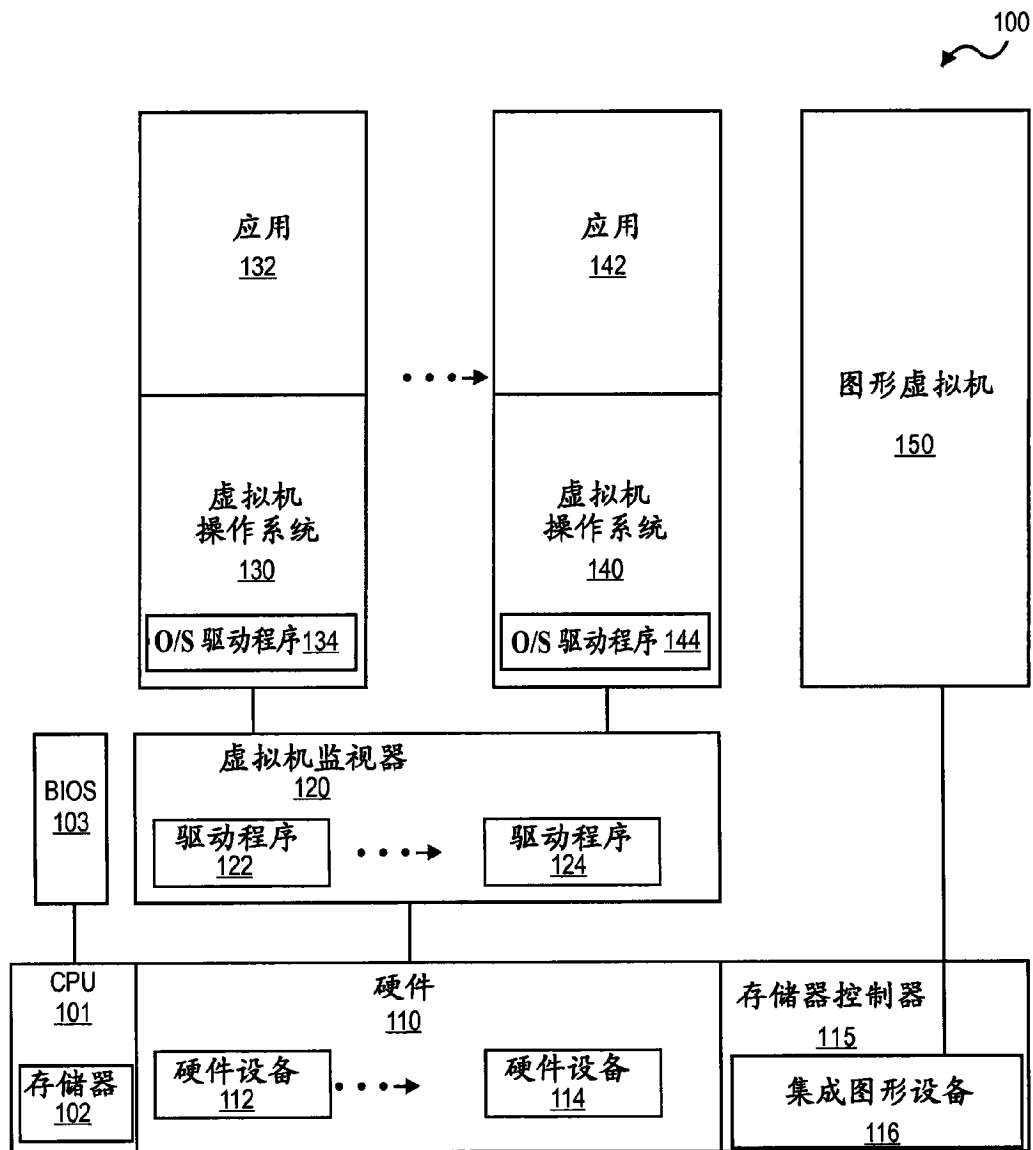


图 1A

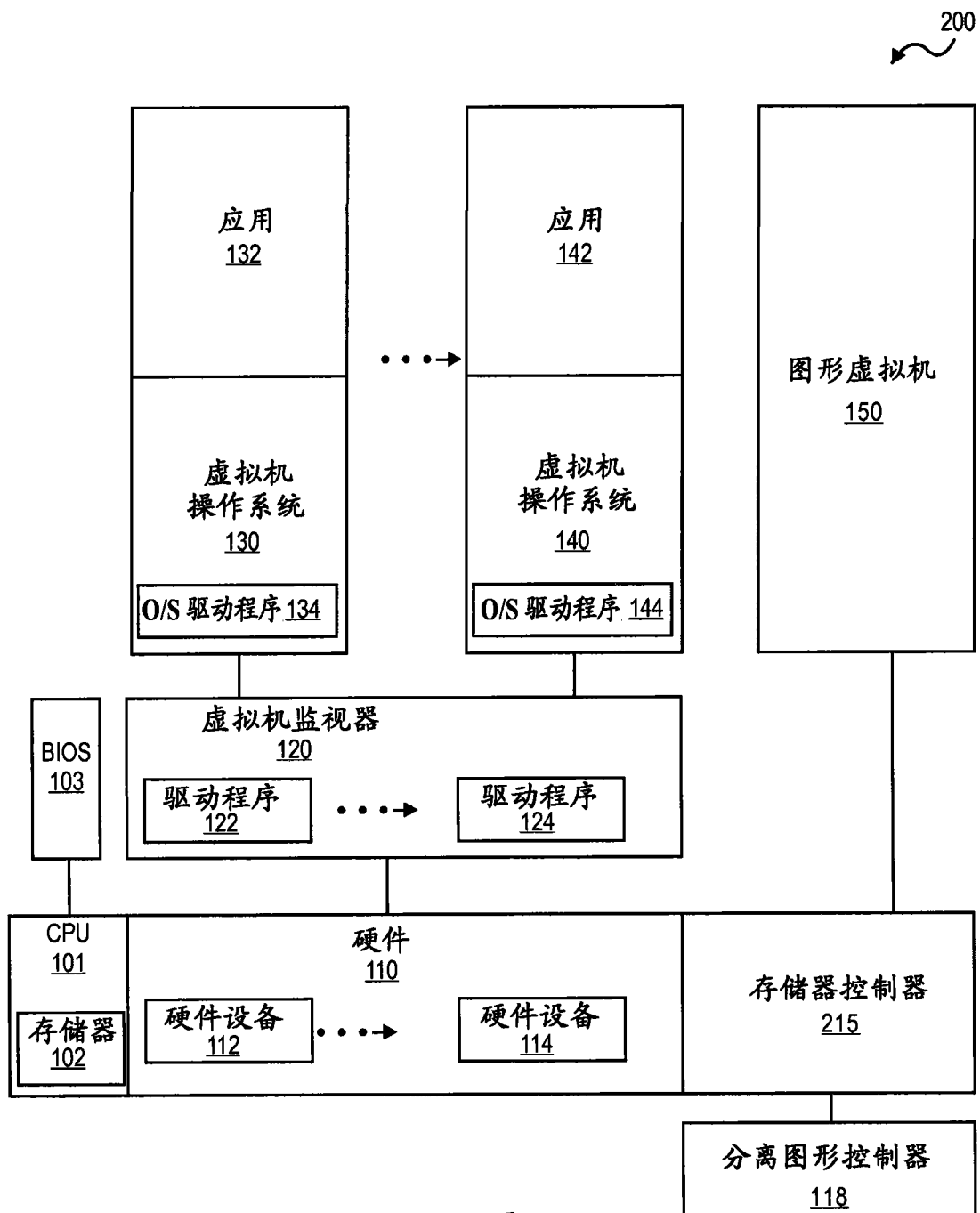


图 1B

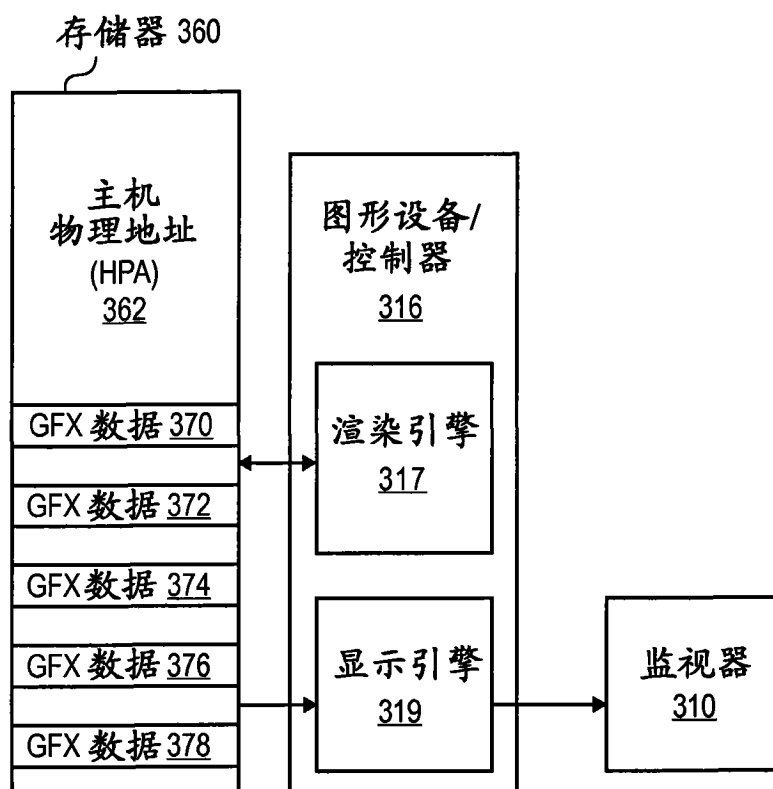


图 2

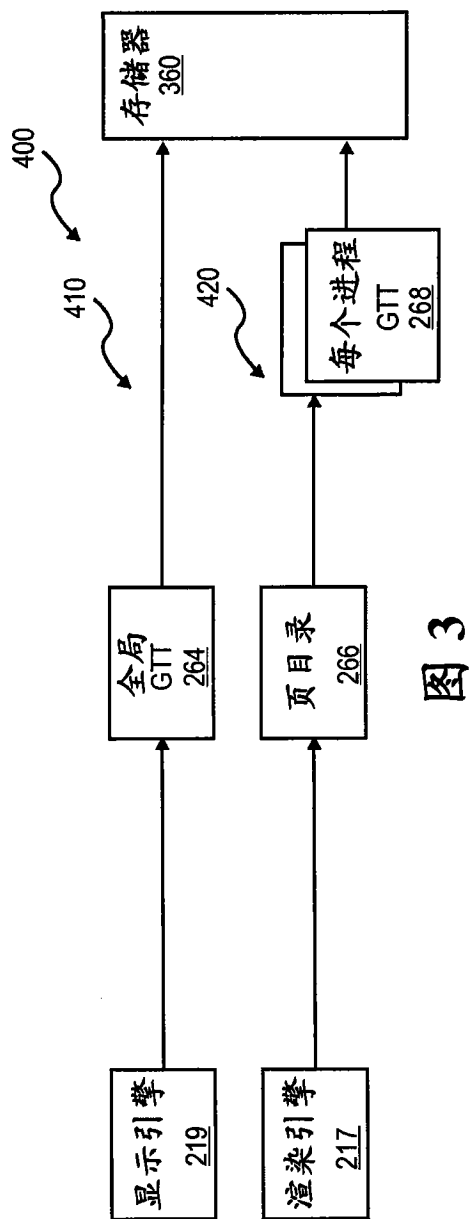


图 3

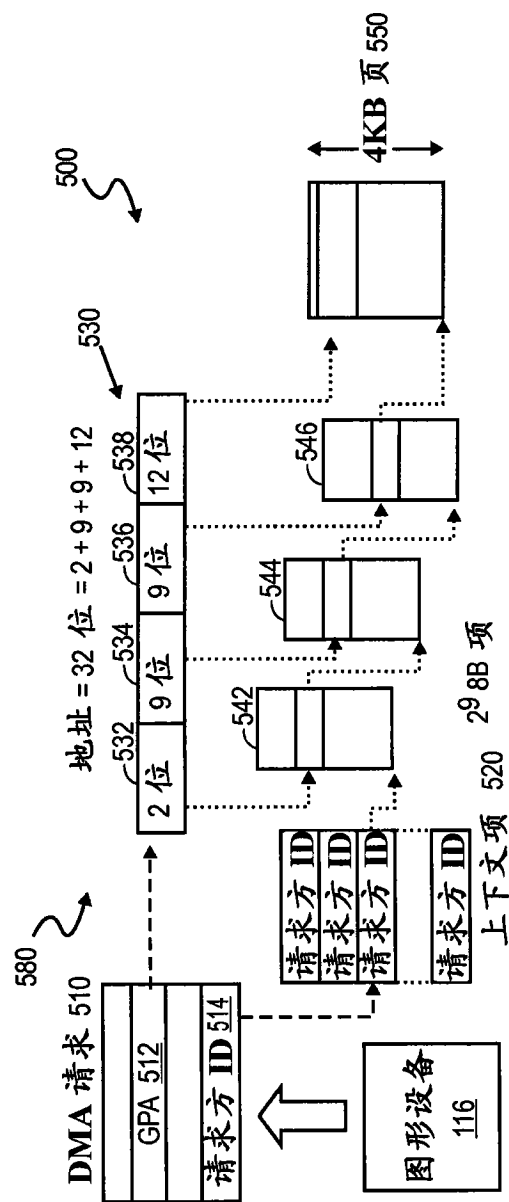


图 4

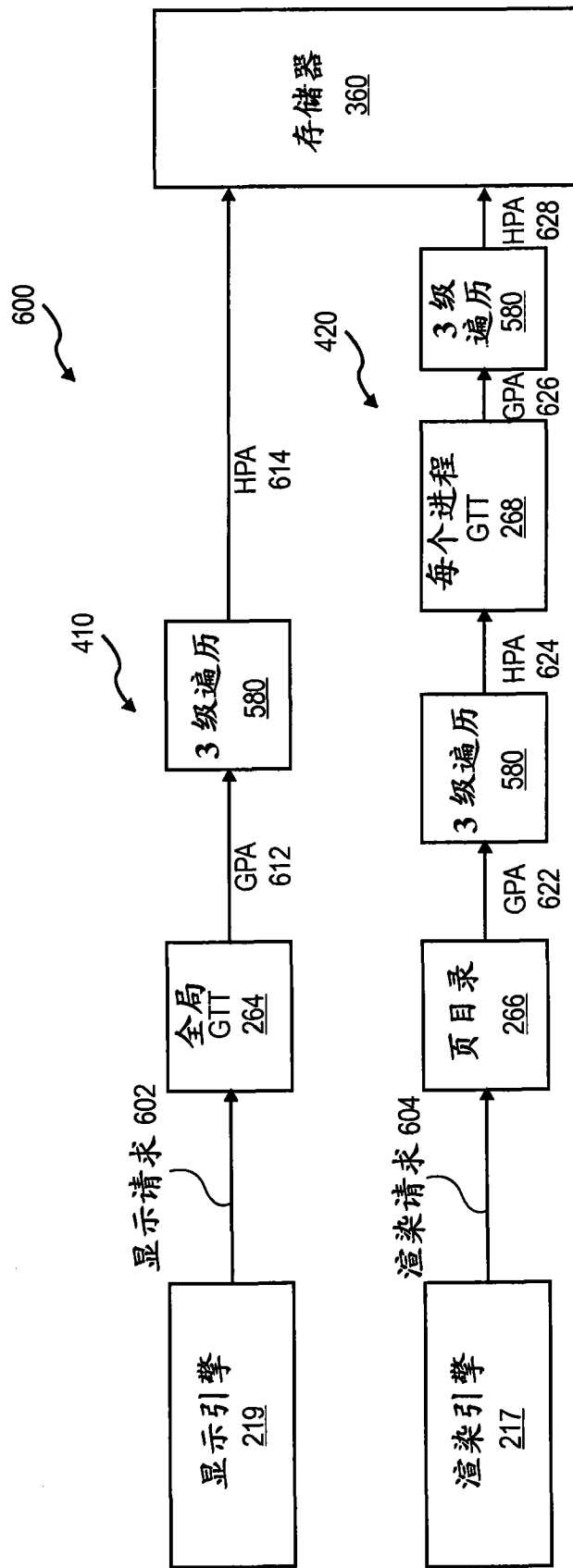


图 5

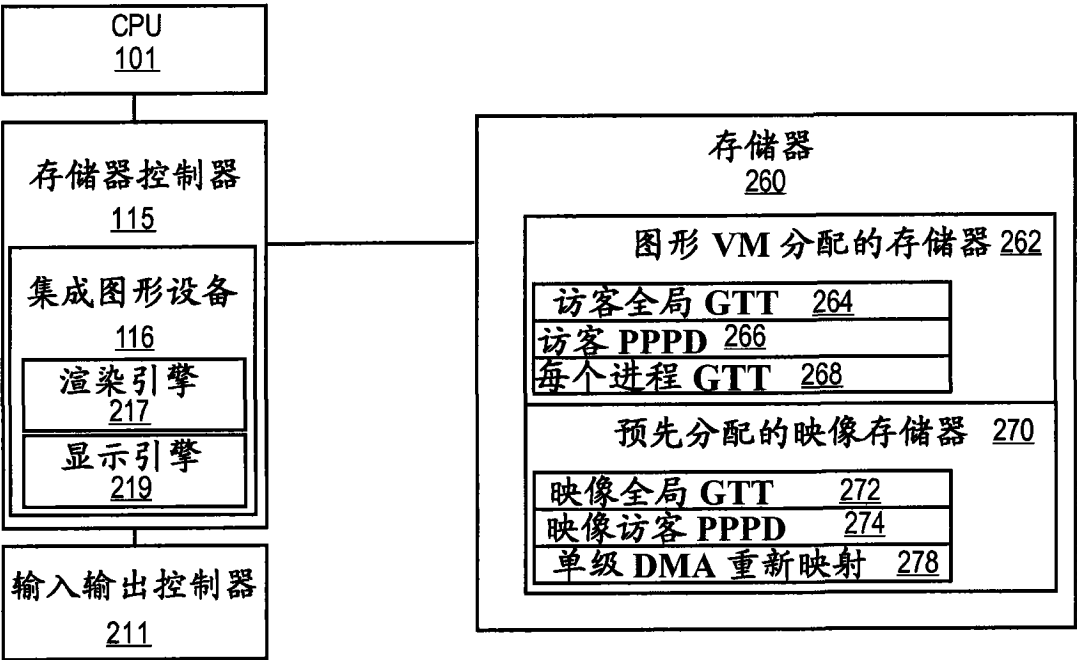


图 6A

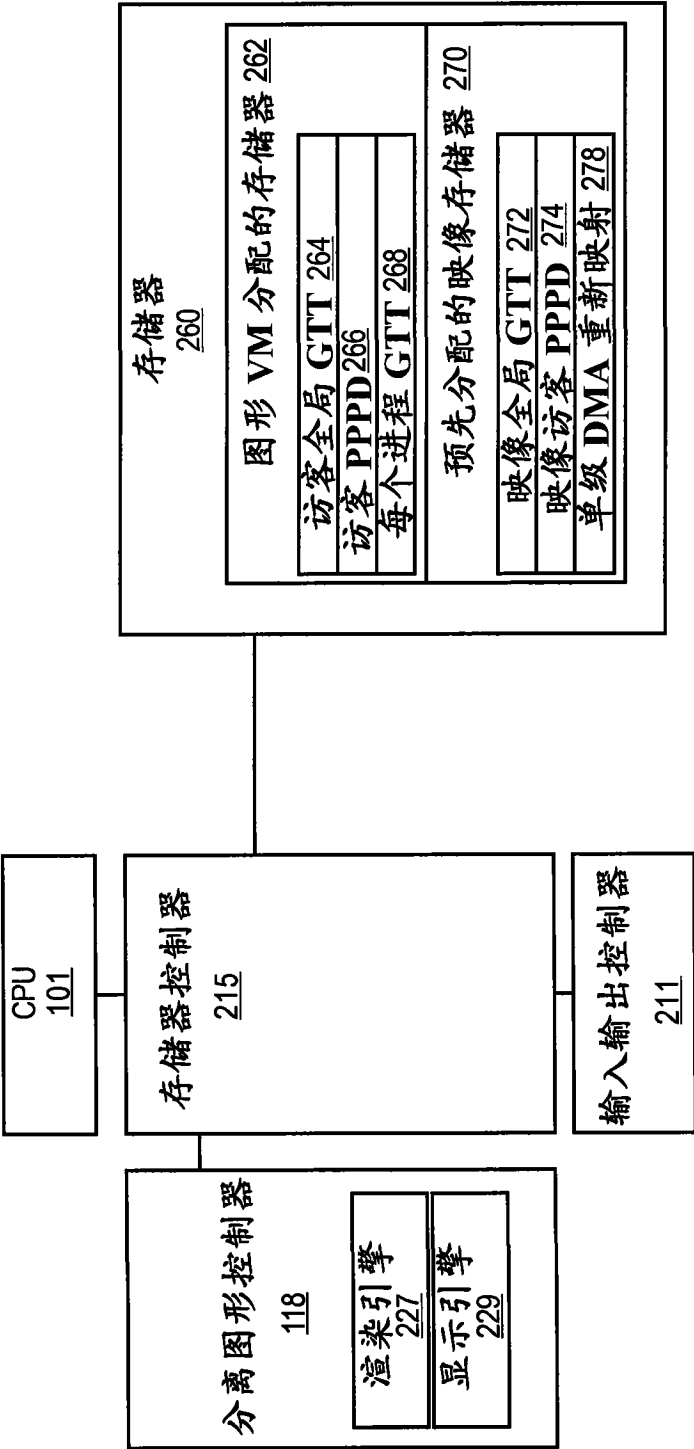


图 6B

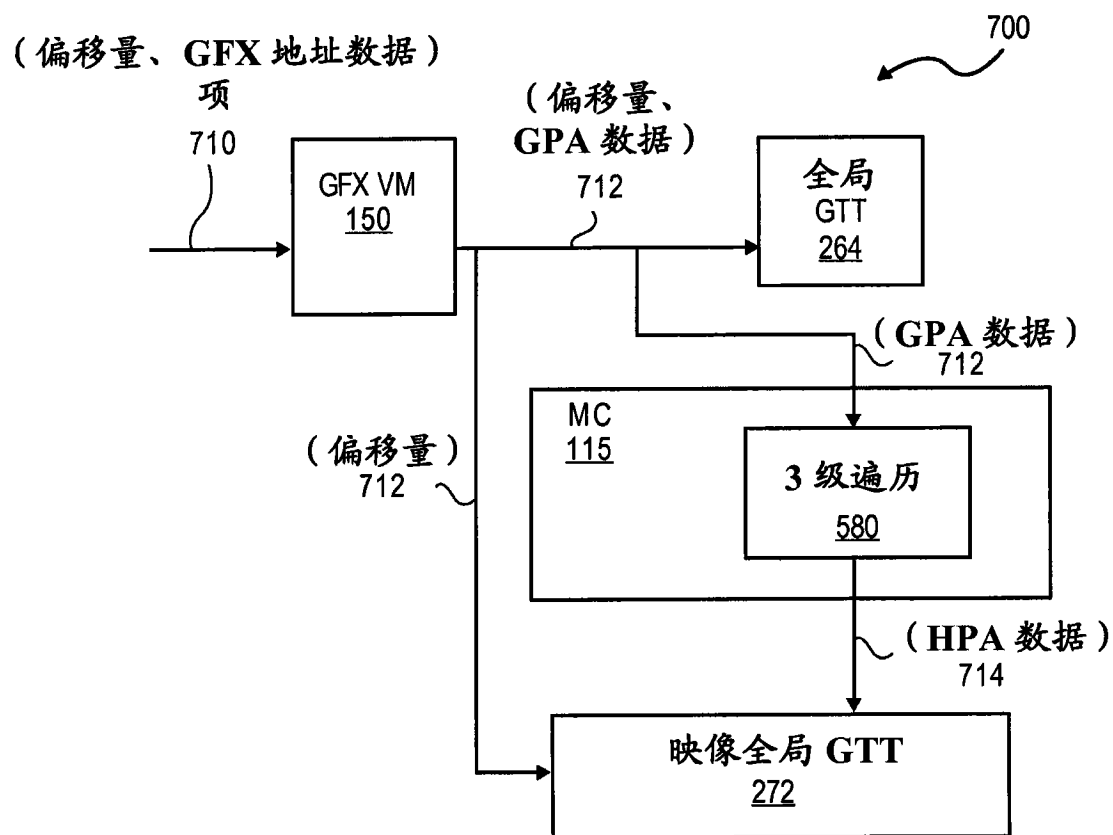


图 7

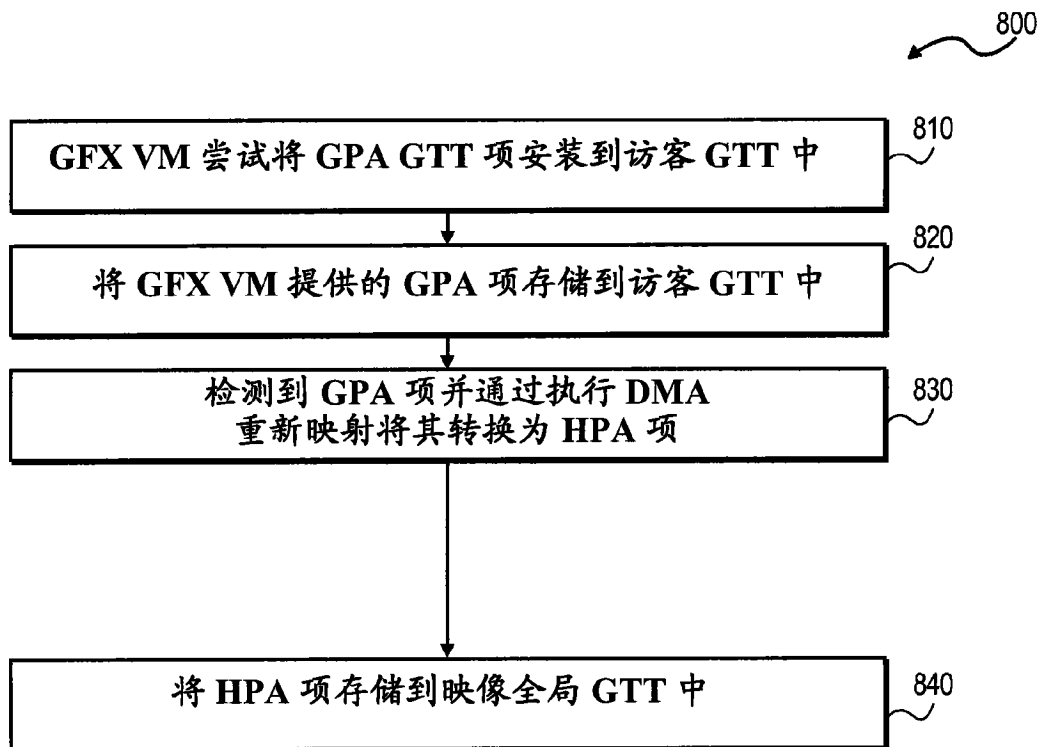


图 8

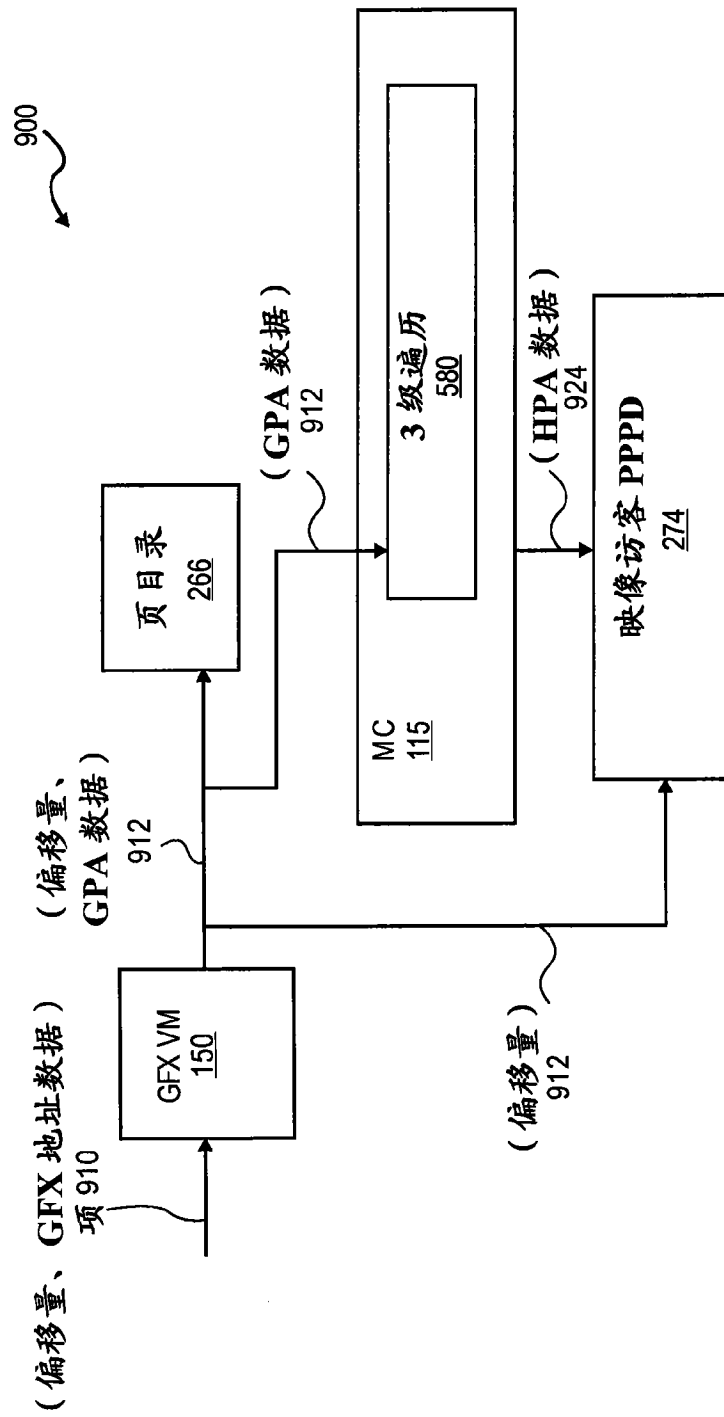


图 9

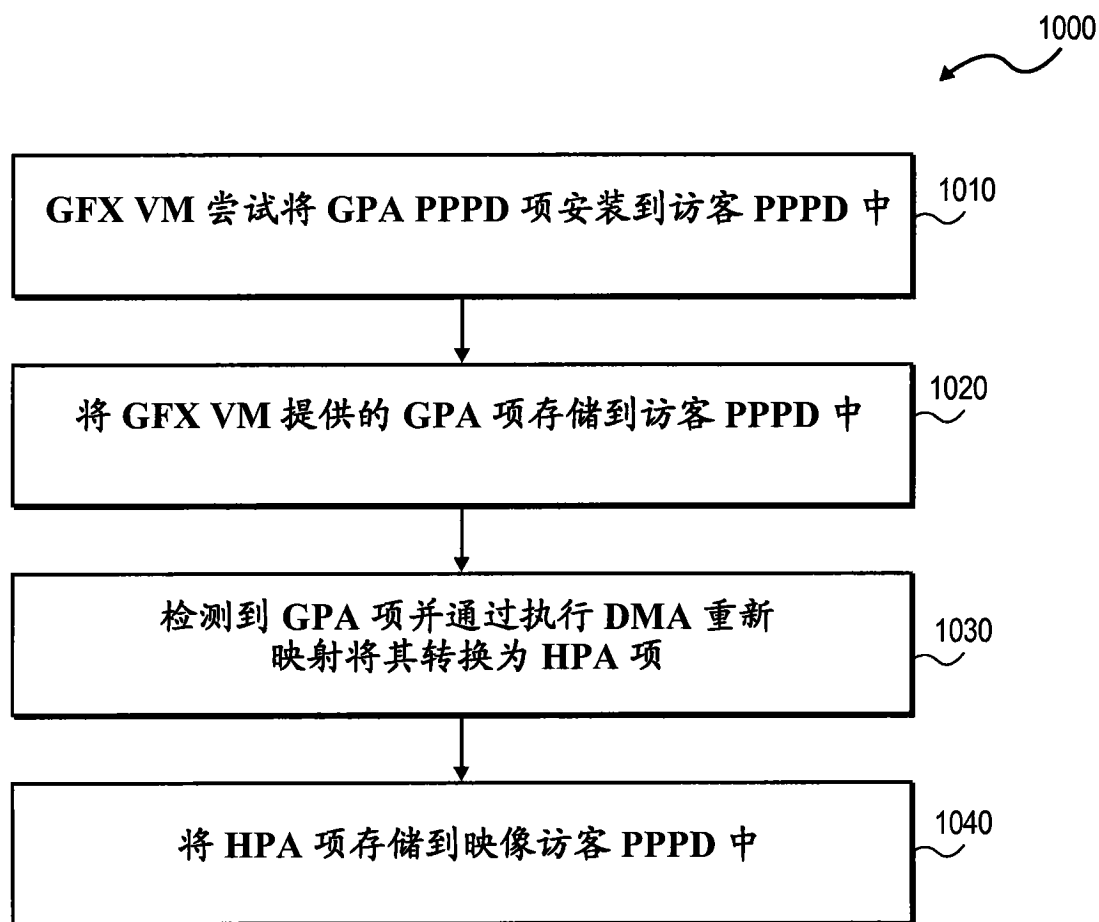
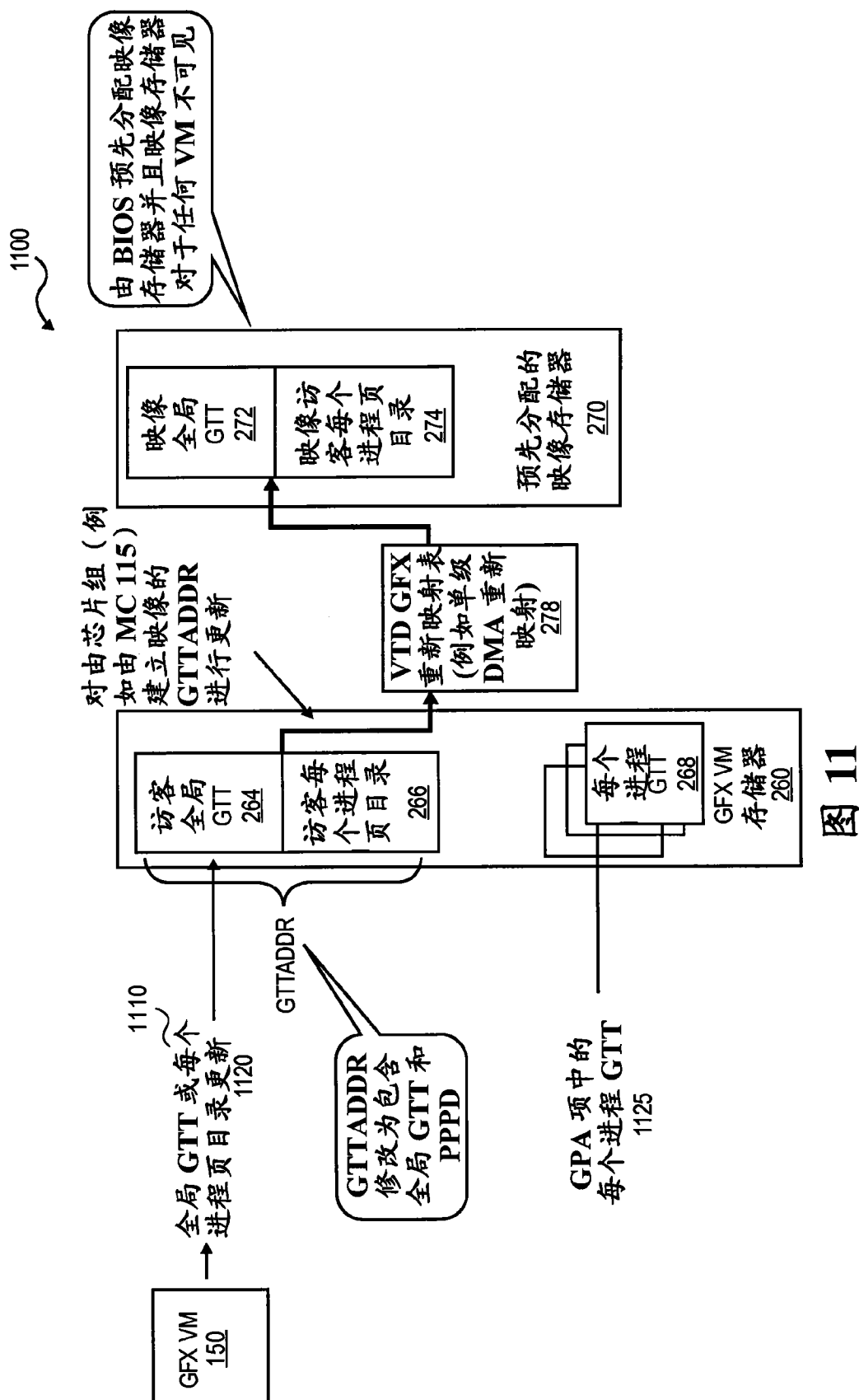


图 10



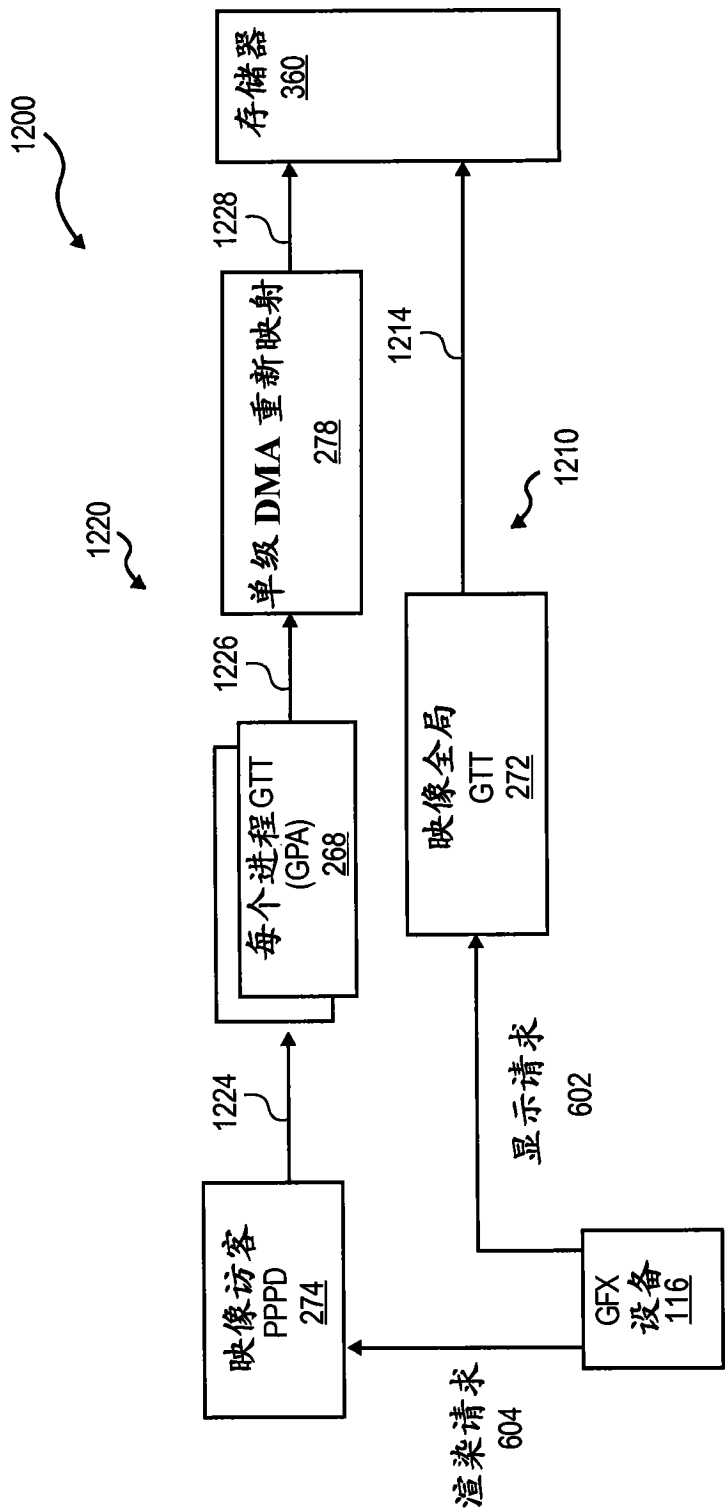
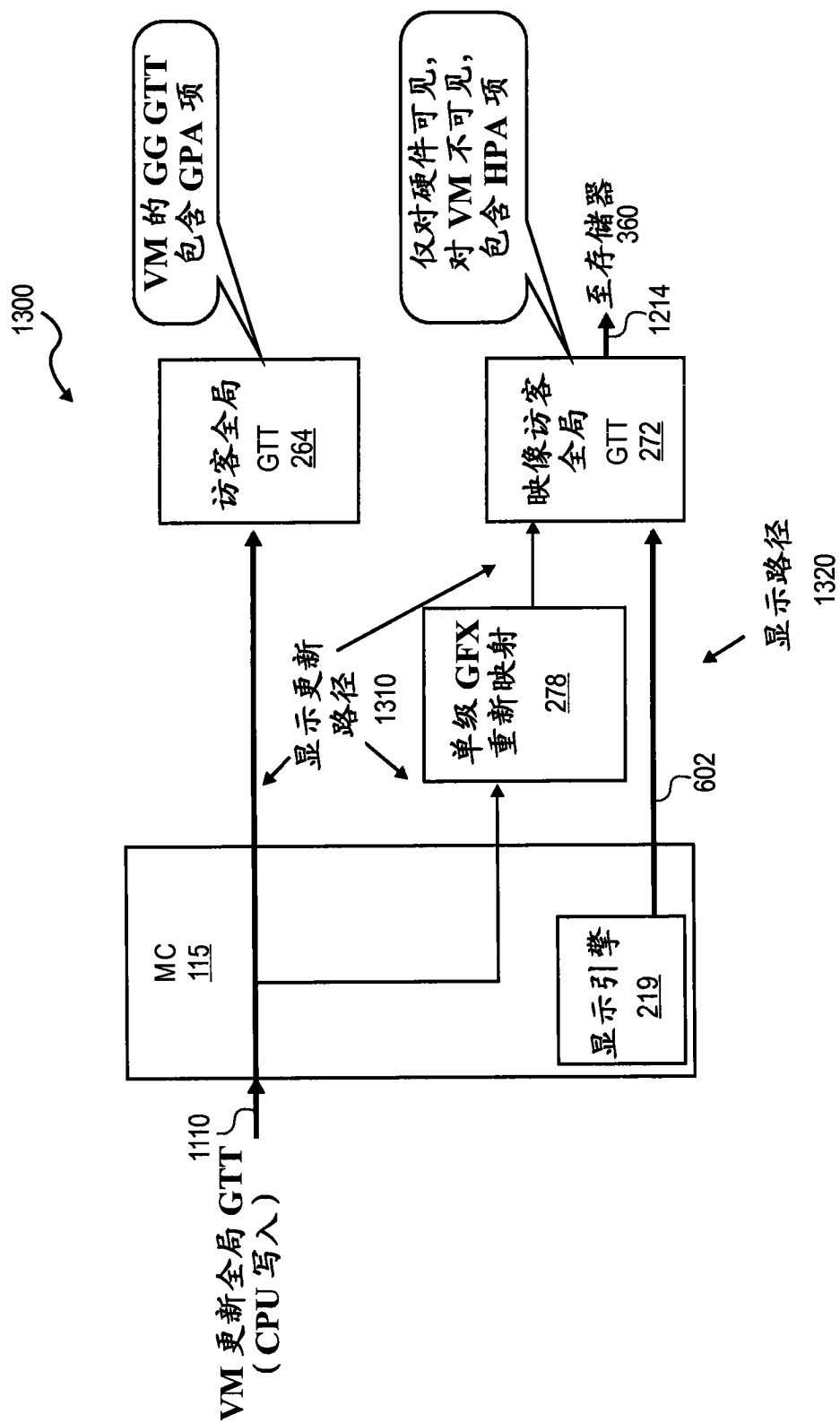


图 12



13

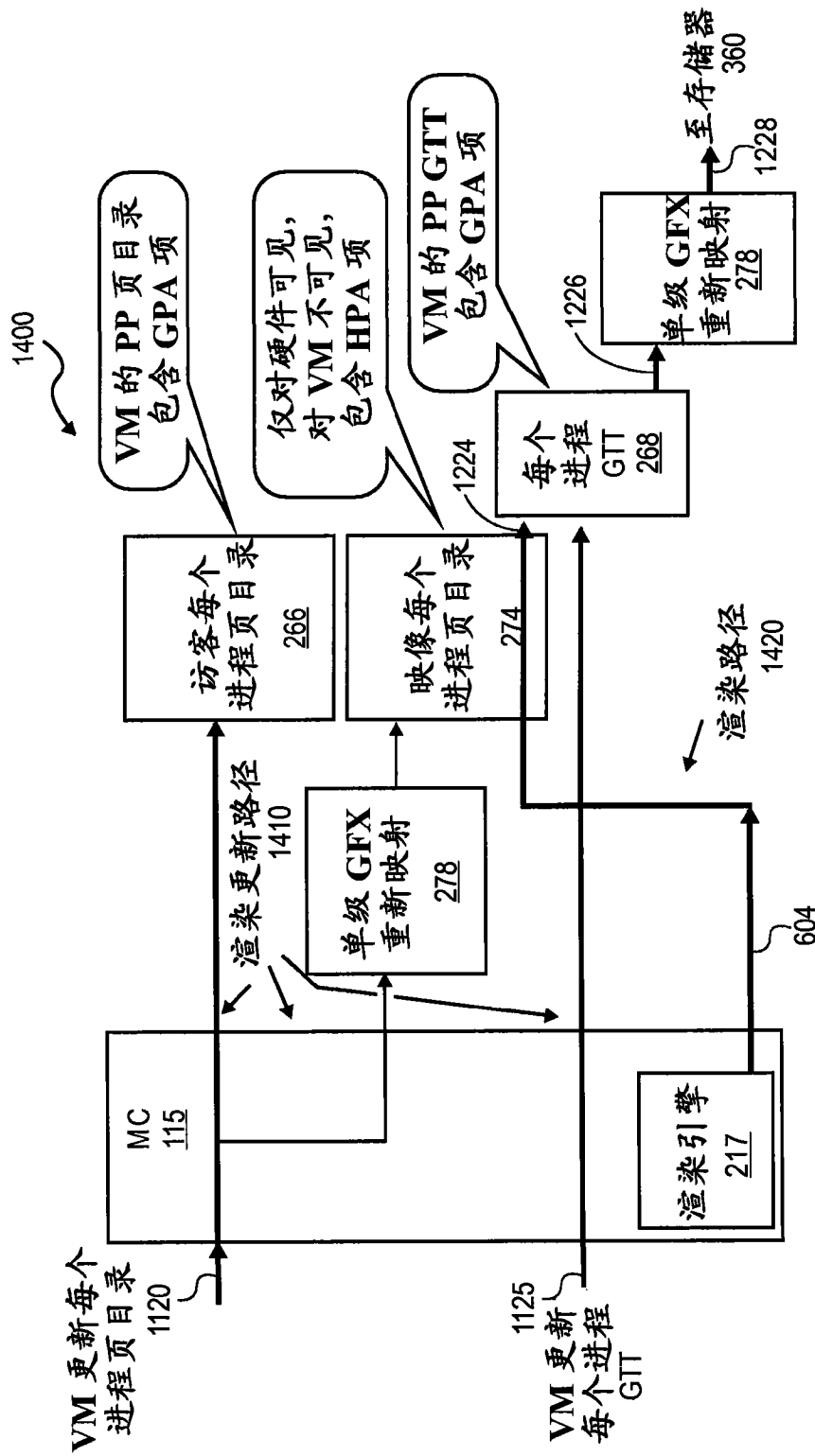


图 14