



(12) 发明专利

(10) 授权公告号 CN 112634424 B

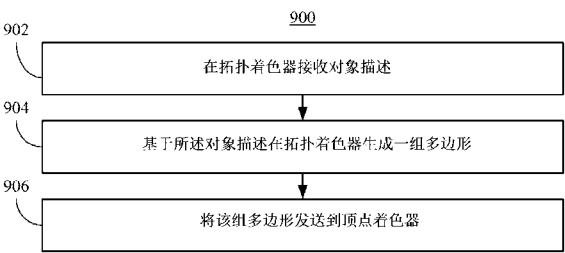
(45) 授权公告日 2025. 06. 03

(21) 申请号 202011562196.5
(22) 申请日 2018.04.09
(65) 同一申请的已公布的文献号
 申请公布号 CN 112634424 A
(43) 申请公布日 2021.04.09
(30) 优先权数据
 15/483409 2017.04.10 US
(62) 分案原申请数据
 201810310967.8 2018.04.09
(73) 专利权人 英特尔公司
 地址 美国加利福尼亚州
(72) 发明人 A.劳里岑 G.利克托尔 J.吉拉赫
 T.巴尔-安 H.拉贝
(74) 专利代理机构 中国专利代理(香港)有限公
 司 72001
 专利代理师 丁辰 姜冰

(51) Int.Cl.
 G06T 15/00 (2011.01)
 G06T 1/60 (2006.01)
 G06T 1/20 (2006.01)
(56) 对比文件
 US 2011018874 A1,2011.01.27
 CN 105513117 A,2016.04.20
 US 2014347363 A1,2014.11.27
 US 4694404 A,1987.09.15
 Chung K 等.Shader-based tessellation
to save memory bandwidth in a mobile
multimedia processor.《Computers &
Graphics》.2009,第33卷(第5期),625-637.
 Tianyun Ni 等.Efficient substitutes
for subdivision surfaces.《Acm Siggraph
Courses Siggraph》.2009,1-107.
审查员 夏冰洁
 权利要求书4页 说明书43页 附图35页

(54) 发明名称
 拓扑着色器技术

(57) 摘要
 系统、设备和方法可提供以下技术：在图形
流水线中的拓扑着色器接收对象描述、并基于对
象描述在拓扑着色器生成一组多边形。另外，将
该组多边形发送到顶点着色器。



1. 一种半导体封装设备,包括:

衬底;以及

耦合到所述衬底的逻辑,其中在以下项中的一项或多项中实现所述逻辑:可配置逻辑或固定功能性硬件逻辑,所述逻辑用于:

在图形处理器处由拓扑着色器基于对象描述生成一组多边形以创建场景的拓扑而非在主机处理器处生成拓扑图元数据,其中所述生成所述一组多边形包括以下操作:

检测一个或多个多边形的可视性的缺乏,

响应于可视性的缺乏,从所述一组多边形中排除所述一个或多个多边形,以及

基于所述对象描述来确定与所述场景中的一个或多个对象相关联的细节级别,其中所述一组多边形含有处于第一细节级别的第一数量的多边形和处于第二细节级别的、不同的第二数量的多边形;以及

基于在所述图形处理器处由所述拓扑着色器生成的所述一组多边形而提供所述场景以用于显示。

2. 根据权利要求1所述的设备,其中,可视性的所述缺乏将包括以下项中的一项或多项:遮蔽状态或离屏状态。

3. 根据权利要求1所述的设备,其中,所述生成所述一组多边形包括以下操作:

在所述图形处理器处由所述拓扑着色器从所述主机处理器接收所述对象描述,其中所述对象描述在不指定所述对象的拓扑图元数据的情况下标识要渲染的对象的类型。

4. 根据权利要求3所述的设备,其中,所述生成所述一组多边形包括以下操作:基于深度信息来确定与所述对象描述相关联的所述细节级别。

5. 根据权利要求4所述的设备,其中,所述生成所述一组多边形包括以下操作:在两个细节级别之间内插。

6. 根据权利要求4所述的设备,其中,所述生成所述一组多边形包括以下操作:标识对应于所述对象描述的子八叉树,并且其中拓扑着色器将被调用以用于所标识的子八叉树。

7. 一种系统,包括:

显示器,所述显示器用于呈现视觉内容;

存储器;以及

半导体封装设备,所述半导体封装设备耦合到所述显示器和所述存储器,其中所述半导体封装设备包括:

衬底;以及

耦合到所述衬底的图形处理器,其中所述图形处理器包括逻辑,所述逻辑用于:

由拓扑着色器基于对象描述生成一组多边形以创建场景的拓扑而非在主机处理器处生成拓扑图元数据,其中所述生成所述一组多边形包括以下操作:

检测一个或多个多边形的可视性的缺乏,

响应于可视性的缺乏,从所述一组多边形中排除所述一个或多个多边形,以及

基于所述对象描述来确定与所述场景中的一个或多个对象相关联的细节级别,其中所述一组多边形含有处于第一细节级别的第一数量的多边形和处于第二细节级别的、不同的第二数量的多边形,以及

基于在所述图形处理器处由所述拓扑着色器生成的所述一组多边形而向所述显示器

提供所述场景。

8. 根据权利要求7所述的系统,其中,可视性的所述缺乏将包括以下项中的一项或多项:遮蔽状态或离屏状态。

9. 根据权利要求7所述的系统,其中,所述生成所述一组多边形包括以下操作:

在所述图形处理器处由所述拓扑着色器从所述主机处理器接收所述对象描述,其中所述对象描述在不指定所述对象的拓扑图元数据的情况下标识要渲染的对象的类型。

10. 根据权利要求9所述的系统,其中,所述生成所述一组多边形包括以下操作:基于深度信息来确定与所述对象描述相关联的所述细节级别。

11. 根据权利要求10所述的系统,其中,所述生成所述一组多边形包括以下操作:在两个细节级别之间内插。

12. 根据权利要求10所述的系统,其中,所述生成所述一组多边形包括以下操作:标识对应于所述对象描述的子八叉树,并且其中拓扑着色器将被调用以用于所标识的子八叉树。

13. 包括指令集的至少一个计算机可读存储介质,所述指令集在由计算装置执行时使所述计算装置:

在图形处理器处由拓扑着色器基于对象描述生成一组多边形以创建场景的拓扑而非在主机处理器处生成拓扑图元数据,其中所述生成所述一组多边形包括以下操作:

检测一个或多个多边形的可视性的缺乏,

响应于可视性的缺乏,从所述一组多边形中排除所述一个或多个多边形,以及

基于所述对象描述来确定与所述场景中的一个或多个对象相关联的细节级别,其中所述一组多边形含有处于第一细节级别的第一数量的多边形和处于第二细节级别的、不同的第二数量的多边形;以及

基于在所述图形处理器处由所述拓扑着色器生成的所述一组多边形而提供所述场景以用于显示。

14. 根据权利要求13所述的至少一个计算机可读存储介质,其中,可视性的所述缺乏将包括以下项中的一项或多项:遮蔽状态或离屏状态。

15. 根据权利要求13所述的至少一个计算机可读存储介质,其中,所述生成所述一组多边形包括以下操作:

在所述图形处理器处由所述拓扑着色器从所述主机处理器接收所述对象描述,其中所述对象描述在不指定所述对象的拓扑图元数据的情况下标识要渲染的对象的类型。

16. 根据权利要求15所述的至少一个计算机可读存储介质,其中,所述生成所述一组多边形包括以下操作:基于深度信息来确定与所述对象描述相关联的所述细节级别。

17. 根据权利要求16所述的至少一个计算机可读存储介质,其中,所述生成所述一组多边形包括以下操作:在两个细节级别之间内插。

18. 根据权利要求16所述的至少一个计算机可读存储介质,其中,所述生成所述一组多边形包括以下操作:标识对应于所述对象描述的子八叉树,并且其中拓扑着色器将被调用以用于所标识的子八叉树。

19. 一种操作半导体封装设备的方法,包括:

在图形处理器处由拓扑着色器基于对象描述生成一组多边形以创建场景的拓扑而非

在主机处理器处生成拓扑图元数据,其中所述生成所述一组多边形包括以下操作:

检测一个或多个多边形的可视性的缺乏,

响应于可视性的缺乏,从所述一组多边形中排除所述一个或多个多边形,以及

基于所述对象描述来确定与所述场景中的一个或多个对象相关联的细节级别,其中所述一组多边形含有处于第一细节级别的第一数量的多边形和处于第二细节级别的、不同的第二数量的多边形;以及

基于在所述图形处理器处由所述拓扑着色器生成的所述一组多边形而提供所述场景以用于显示。

20. 根据权利要求19所述的方法,其中,可视性的所述缺乏将包括以下项中的一项或多项:遮蔽状态或离屏状态。

21. 根据权利要求19所述的方法,其中,所述生成所述一组多边形包括以下操作:

在所述图形处理器处由所述拓扑着色器从所述主机处理器接收所述对象描述,其中所述对象描述在不指定所述对象的拓扑图元数据的情况下标识要渲染的对象的类型。

22. 根据权利要求21所述的方法,其中,所述生成所述一组多边形包括以下操作:基于深度信息来确定与所述对象描述相关联的所述细节级别。

23. 根据权利要求22所述的方法,其中,所述生成所述一组多边形包括以下操作:在两个细节级别之间内插。

24. 根据权利要求22所述的方法,其中,所述生成所述一组多边形包括以下操作:标识对应于所述对象描述的子八叉树,并且其中拓扑着色器将被调用以用于所标识的子八叉树。

25. 一种半导体封装设备,包括:

用于在图形处理器处由拓扑着色器基于对象描述生成一组多边形以创建场景的拓扑而非在主机处理器处生成拓扑图元数据的部件,其中用于生成所述一组多边形的所述部件包括:

用于检测一个或多个多边形的可视性的缺乏的部件,

用于响应于可视性的缺乏,从所述一组多边形中排除所述一个或多个多边形的部件,以及

用于基于所述对象描述来确定与所述场景中的一个或多个对象相关联的细节级别的部件,其中所述一组多边形含有处于第一细节级别的第一数量的多边形和处于第二细节级别的、不同的第二数量的多边形;以及

用于基于在所述图形处理器处由所述拓扑着色器生成的所述一组多边形而提供所述场景以用于显示的部件。

26. 根据权利要求25所述的设备,其中,可视性的所述缺乏将包括以下项中的一项或多项:遮蔽状态或离屏状态。

27. 根据权利要求25所述的设备,其中,用于生成所述一组多边形的所述部件包括:

用于在所述图形处理器处由所述拓扑着色器从所述主机处理器接收所述对象描述的部件,其中所述对象描述在不指定所述对象的拓扑图元数据的情况下标识要渲染的对象的类型。

28. 根据权利要求27所述的设备,其中,用于生成所述一组多边形的所述部件包括用于

基于深度信息来确定与所述对象描述相关联的所述细节级别的部件。

29. 根据权利要求28所述的设备, 其中, 用于生成所述一组多边形的所述部件包括用于在两个细节级别之间内插的部件。

30. 根据权利要求28所述的设备, 其中, 用于生成所述一组多边形的所述部件包括用于标识对应于所述对象描述的子八叉树的部件, 并且其中拓扑着色器将被调用以用于所标识的子八叉树。

31. 一种已在其上存储有指令的计算机可读介质, 所述指令在执行时促使计算装置执行根据权利要求19-24中任一项所述的方法。

拓扑着色器技术

技术领域

[0001] 本申请的实施例一般涉及图形处理架构。更具体地,实施例涉及图形处理架构中的拓扑着色器技术。

背景技术

[0002] 在传统图形处理架构中,中央处理单元(CPU)可确定要被渲染的场景的复杂度和拓扑,并生成由图形流水线中的顶点着色器进一步处理的指令流。所述指令流可包含诸如高阶表面的顶点数据、图元等的参数,其中顶点着色器可将顶点数据从基于对象的坐标表示(对象空间)变换到诸如世界空间或归一化装置坐标(NDC)空间的备选地基于坐标系(alternatively based coordinate system)。此类架构可限于描绘场景的每个视频帧的固定拓扑,其中固定拓扑可限制质量和/或性能。此外,传统手段可使用图形流水线中稍后的曲面细分级来生成针对与几何图元关联的每个顶点的顶点属性和表面表示。只依赖于稍后级曲面细分来生成表面表示可进一步限制质量和/或性能。

附图说明

[0003] 实施例的各种优点将通过阅读以下说明书和所附权利要求以及通过参考以下附图而对本领域技术人员变得显而易见,在附图中:

[0004] 图1是框图,其示出配置成实现本文描述的实施例的一个或多个方面的计算机系统;

[0005] 图2A-2D示出了根据实施例的并行处理器部件;

[0006] 图3A-3B是根据实施例的图形多处理器的框图;

[0007] 图4A-4F示出了示例性架构,其中多个GPU通信地耦合至多个多核处理器;

[0008] 图5示出了根据实施例的图形处理流水线;

[0009] 图6是根据另一实施例的图形处理流水线的示例的框图;

[0010] 图7是根据实施例的在细节级别变化的球体的拓扑表示的示例的图示;

[0011] 图8是根据实施例的缺乏可视性条件的示例的图示;

[0012] 图9A是根据实施例的操作半导体封装设备的方法的示例的流程图;

[0013] 图9B是根据实施例的生成一组多边形的方法的示例的流程图;

[0014] 图10A是根据实施例的计算系统的示例的框图;

[0015] 图10B是根据实施例的半导体封装设备的示例的图示;

[0016] 图11是根据实施例的头戴式显示器(HMD)系统的示例的图示;

[0017] 图12是根据实施例包括于图11的HMD系统中的功能部件的示例的框图;

[0018] 图13是根据实施例包括于并行处理单元中的通用处理集群的示例的框图;

[0019] 图14是根据实施例可在并行处理单元内实现的图形处理流水线的示例的概念图示;

[0020] 图15是根据实施例的流传送多处理器的示例的框图;

- [0021] 图16-18是根据实施例的数据处理系统的概述的示例的框图；
- [0022] 图19是根据实施例的图形处理引擎的示例的框图；
- [0023] 图20-22是根据实施例的执行单元的示例的框图；
- [0024] 图23是根据实施例的图形流水线的示例的框图；
- [0025] 图24A-24B是根据实施例的图形流水线编程的示例的框图；
- [0026] 图25是根据实施例的图形软件架构的示例的框图；
- [0027] 图26是根据实施例的知识产权 (IP) 核开发系统的示例的框图；以及
- [0028] 图27是根据实施例的芯片上系统集成电路的示例的框图。

具体实施方式

[0029] 在以下描述中,阐述了许多特定细节以提供对本公开的更全面理解。然而,对本领域技术人员之一显而易见的是,在没有一个或多个这些特定细节的情况下也可实践本发明。在其他实例中,尚未描述公知特征以便避免使本发明难以理解。

[0030] 系统概述

[0031] 图1是框图,其示出配置成实现本文描述的实施例的一个或多个方面的计算系统。计算系统100包括处理子系统101,所述处理子系统101具有一个或多个处理器102和系统存储器104,所述一个或多个处理器102与所述系统存储器104经由可包括存储器中枢105的互连路径来通信。存储器中枢105可以是芯片组部件内的单独部件,或可以集成在所述一个或多个处理器102内。存储器集线器105经由通信链路106与I/O子系统111耦合。I/O子系统111包括I/O中枢107,所述I/O中枢107可以使得计算系统100能够从一个或多个输入装置108接收输入。另外,I/O中枢107可以使得显示器控制器能够将输出提供给一个或多个显示装置110A,所述显示器控制器可被包括在所述一个或多个处理器102中。在一个实施例中,与I/O中枢107耦合的所述一个或多个显示装置110A可以包括局部、内部或嵌入式显示装置。

[0032] 在一个实施例中,处理子系统101包括一个或多个并行处理器112,所述并行处理器112经由总线或其他通信链路113耦合至存储器中枢105。通信链路113可以是任何数量的基于标准的通信链路技术或协议之一(诸如但不限于,PCI快速),或可以是供应方特定的通信接口或通信结构。在一个实施例中,所述一个或多个并行处理器112形成计算上集中的并行或向量处理系统,所述系统包括大量处理核和/或处理集群(诸如,集成众核(MIC)处理器)。在一个实施例中,所述一个或多个并行处理器112形成图形处理子系统,所述图形处理子系统可以将像素输出到经由I/O中枢107耦合的所述一个或多个显示装置110A之一。所述一个或多个并行处理器112还可以包括显示器控制器和显示器接口(未示出)以能够实现到一个或多个显示装置110B的直接连接。

[0033] 在I/O子系统111内,系统存储单元114可以连接到I/O中枢107以提供用于计算系统100的存储机制。I/O开关116可以用于提供接口机制以能够实现I/O中枢107与其他部件(诸如,可集成到平台中的网络适配器118和/或无线网络适配器119,以及可以经由一个或多个插入式装置120添加的各种其他装置)之间的连接。网络适配器118可以是以以太网适配器或另一有线网络适配器。无线网络适配器119可以包括以下项中的一项或多项:Wi-Fi、蓝牙、近场通信(NFC)、或包括一个或多个无线无线电装置(wireless radio)的另一网络装置。

[0034] 计算系统100可以包括未显式示出的其他部件,包括USB或其他端口连接件、光学存储驱动、视频捕获装置、和诸如此类,它们也可连接到I/O中枢107。使图1中的各种部件互连的通信路径可使用任何合适的协议来实现,诸如基于PCI(外围部件互连)的协议(例如,PCI快速)或任何其他总线或点对点通信接口和/或协议(诸如,NV-Link高速互连、或本领域中已知的互连协议)。

[0035] 在一个实施例中,所述一个或多个并行处理器112合并针对图形和视频处理进行优化的电路(包括例如,视频输出电路),并且构成图形处理单元(GPU)。在另一个实施例中,所述一个或多个并行处理器112合并针对通用处理进行优化的电路,同时保持本文更详细地描述的底层计算架构。在又一个实施例中,计算系统100的部件可与一个或多个其他系统元件一起集成在单个集成电路上。例如,所述一个或多个并行处理器112、存储器中枢105、处理器102和I/O中枢107可以集成到芯片上系统(SoC)集成电路中。可替代地,计算系统100的部件可以集成到单个封装中以形成封装中系统(SIP)配置。在一个实施例中,计算系统100的部件的至少一部分可以集成到多芯片模块(MCM)中,所述MCM可以与其他多芯片模块一起互连到模块化计算系统中。

[0036] 将认识到的是,本文示出的计算系统100是说明性的,并且变化和修改是有可能的。可如期望的来修改连接拓扑,包括桥的数量和布置、(一个或多个)处理器102的数量和(一个或多个)并行处理器112的数量。例如,在一些实施例中,系统存储器104直接而非通过桥而被连接到(一个或多个)处理器102,而其他装置经由存储器中枢105与(一个或多个)处理器102和系统存储器104通信。在其他替代性拓扑中,(一个或多个)并行处理器112连接到I/O中枢107或直接连接到所述一个或多个处理器102之一,而非连接到存储器中枢105。在其他实施例中,I/O中枢107和存储器中枢105可集成到单个芯片中。一些实施例可包括经由多个插口附连的两组或更多组处理器102,它们可以与(一个或多个)并行处理器112的两个或更多个实例耦合。

[0037] 本文中示出的具体部件中的一些是可选的,并且可以不被包括在计算系统100的所有实现中。例如,可支持任何数量的插入式卡或外围装置,或可消除一些部件。此外,一些架构可对于与图1中示出的那些部件类似的部件使用不同的术语。例如,在一些架构中,存储器中枢105可称为北桥,而I/O中枢107可称为南桥。

[0038] 图2A示出了根据实施例的并行处理器200。并行处理器200的各种部件可使用一个或多个集成电路装置(诸如可编程处理器、专用集成电路(ASIC)或现场可编程门阵列(FPGA))来实现。根据实施例,所示出的并行处理器200是图1中所示的一个或多个并行处理器112的变体。

[0039] 在一个实施例中,并行处理器200包括并行处理单元202。所述并行处理单元包括I/O单元204,所述I/O单元204能够实现与其他装置(包括并行处理单元202的其他实例)的通信。I/O单元204可直接连接到其他装置。在一个实施例中,I/O单元204经由使用中枢或开关接口(诸如,存储器中枢105)来与其他装置连接。存储器中枢105与I/O单元204之间的连接形成通信链路113。在并行处理单元202内,I/O单元204与主机接口206和存储器交叉开关(memory crossbar)216连接,其中,主机接口206接收针对执行处理操作的命令,并且存储器交叉开关216接收针对执行存储器操作的命令。

[0040] 当主机接口206经由I/O单元204接收命令缓冲器时,主机接口206可以将用于执行

那些命令的工作操作导引至前端208。在一个实施例中,前端208与调度器210耦合,该调度器210配置成将命令或其他工作项分布至处理集群阵列212。在一个实施例中,调度器210确保在任务被分布至处理集群阵列212的处理集群之前,处理集群阵列212被适当地配置且处于有效状态。在一个实施例中,调度器210是经由在微控制器上执行的固件逻辑来实现的。微控制器实现的调度器210可配置成以粗糙粒度和精细粒度来执行复杂的调度和工作分布操作,从而能够实现在处理阵列212上执行的线程的上下文切换和快速抢占(rapid preemption)。在一个实施例中,主机软件可以经由多个图形处理门铃(graphics processing doorbell)之一来检验工作负荷在处理阵列212上调度。随后工作负荷可以由调度器微控制器内的调度器210逻辑来跨处理阵列212自动地分布。

[0041] 处理集群阵列212可以包括多达“N”个处理集群(例如,集群214A、集群214B、直到集群214N)。处理集群阵列212的每一个集群214A-214N都可以执行大量的并发线程。调度器210可以使用各种调度和/或工作分布算法来将工作分配给处理集群阵列212的集群214A-214N,这可取决于针对每一种类型的程序或计算产生的工作负荷而变化。调度可以由调度器210动态地处置,或者可以在配置用于由处理集群阵列212执行的程序逻辑的编译期间部分地由编译器逻辑进行辅助。在一个实施例中,可以将处理集群阵列212的不同集群214A-214N分配用于处理不同类型的程序,或用于执行不同类型的计算。

[0042] 可以将处理集群阵列212配置成执行各种类型的并行处理操作。在一个实施例中,将处理集群阵列212配置成执行通用并行计算操作。例如,处理集群阵列212可以包括用于执行处理任务的逻辑,所述处理任务包括过滤视频和/或音频数据、执行建模操作(包括物理操作)、以及执行数据变换。

[0043] 在一个实施例中,处理集群阵列212配置成执行并行的图形处理操作。在其中并行处理器200配置成执行图形处理操作的实施例中,处理集群阵列212可以包括用于支持执行此类图形处理操作的附加逻辑,包括但不限于用于执行纹理操作的纹理采样逻辑、以及曲面细分逻辑和其他顶点处理逻辑。另外,处理集群阵列212可以配置成执行图形处理相关着色器程序,诸如但不限于顶点着色器、曲面细分着色器、几何着色器和像素着色器。并行处理单元202可以经由I/O单元204来转移来自系统存储器的数据以供处理。在处理期间,可以将所转移的数据在处理期间存储到芯片上存储器(例如,并行处理器存储器222),然后将其写回到系统存储器。

[0044] 在一个实施例中,当并行处理单元202用于执行图形处理时,调度器210可以配置成将处理工作负荷划分成近似相等大小的任务,以更好地使得能够将图形处理操作分布到处理集群阵列212中的多个集群214A-214N。在一些实施例中,处理集群阵列212的多个部分可以配置成执行不同类型的处理。例如,第一部分可配置成执行顶点着色和拓扑生成,第二部分可配置成执行曲面细分和几何着色,并且第三部分可配置成执行像素着色或其他屏幕空间操作,以产生供显示的渲染图像。由集群214A-214N中的一者或多者所产生的中间数据可存储在缓冲器中以允许在集群214A-214N之间传送所述中间数据以供进一步处理。

[0045] 在操作期间,处理集群阵列212可以经由调度器210来接收待执行的处理任务,所述调度器从前端208接收定义处理任务的命令。针对图形处理操作,处理任务可以包括定义要如何处理数据(例如,要执行什么程序)的命令和状态参数以及待处理的数据(例如,表面(补丁(patch))数据、图元数据(primitive data)、顶点数据和/或像素数据)的索引。调度

器210可配置成获取与任务相对应的索引,或可从前端208接收索引。前端208可以配置成确保在发起由进入的命令缓冲器(例如,分批缓冲器、推动缓冲器等)所指定的工作负荷之前处理集群阵列212配置成有效状态。

[0046] 并行处理单元202的一个或多个实例中的每一个都可以与并行处理器存储器222耦合。并行处理器存储器222可以经由存储器交叉开关216来访问,存储器交叉开关216可以从处理集群阵列212以及I/O单元204接收存储器请求。存储器交叉开关216可以经由存储器接口218访问并行处理器存储器222。存储器接口218可以包括多个分区单元(例如,分区单元220A、分区单元220B、直到分区单元220N),其可以各自耦合至并行处理器存储器222的一部分(例如,存储器单元)。在一种实现中,将分区单元220A-220N的数量配置成等于存储器单元的数量,使得第一分区单元220A具有对应的第一存储器单元224A,第二分区单元220B具有对应的存储器单元224B,并且第N分区单元220N具有对应的第N存储器单元224N。在其他实施例中,分区单元220A-220N的数量可以不等于存储器装置的数量。

[0047] 在各种实施例中,存储器单元224A-224N可以包括各种类型的存储器装置,包括动态随机存取存储器(DRAM)或图形随机存取存储器,诸如,同步图形随机存取存储器(SGRAM),包括图形双数据速率(GDDR)存储器。在一个实施例中,存储器单元224A-224N还可包括3D堆叠式存储器,包括但不限于高带宽存储器(HBM)。本领域技术人员将认识到,存储器单元224A-224N的特定实现可以变化,并且可以选自各种常规设计之一。渲染目标(诸如,帧缓冲器或纹理映射)可跨存储器单元224A-224N存储,从而允许分区单元220A-220N并行写入每个渲染目标的多个部分以高效地使用并行处理器存储器222的可用带宽。在一些实施例中,可排除并行处理器存储器222的本地实例,以有利于连同本地高速缓存存储器来利用系统存储器的统一存储器设计。

[0048] 在一个实施例中,处理集群阵列212的集群214A-214N中的任何一个都可以处理将被写入到并行处理器存储器222内的存储器单元224A-224N中的任何一个的数据。可以将存储器交叉开关216配置成将每一个集群214A-214N的输出转移到任何分区单元220A-220N或另一集群214A-214N,其可以对输出执行附加处理操作。每一个集群214A-214N都可以通过存储器交叉开关216与存储器接口218通信,以便从各种外部存储器装置读取或向各种外部存储器装置写入。在一个实施例中,存储器交叉开关216具有到存储器接口218的连接以及与I/O单元204通信,以及到并行处理器存储器222的本地实例的连接,从而使不同处理集群214A-214N内的处理单元能够与系统存储器或不在并行处理单元202本地的其他存储器通信。在一个实施例中,存储器交叉开关216可以使用虚拟通道以分离集群214A-214N与分区单元220A-220N之间的业务流。

[0049] 虽然在并行处理器200内示出了并行处理单元202的单个实例,但是可以包括并行处理单元202的任何数量的实例。例如,可以在单个插入式卡上提供并行处理单元202的多个实例,或可以将多个插入式卡互连。并行处理单元202的不同实例可以配置成即使所述不同实例具有不同数量的处理核、不同量的本地并行处理器存储器和/或其他配置差异也还互操作。例如且在一个实施例中,并行处理单元202的一些实例可以相对于其他实例包括更高精度浮点单元。合并并行处理单元202或并行处理器200的一个或多个实例的系统可以以多种配置和形状因子(form factor)来实现,包括但不限于桌上型、膝上型、或手持式个人计算机、服务器、工作站、游戏控制台和/或嵌入式系统。

[0050] 图2B是根据实施例的分区单元220的框图。在一个实施例中,分区单元220是图2A的分区单元220A-220N之一的实例。如所示出,分区单元220包括L2高速缓存221、帧缓冲器接口225和ROP 226(光栅操作单元)。L2高速缓存221是读/写高速缓存,其配置成执行从存储器交叉开关216和ROP 226接收的加载和存储操作。由L2高速缓存221将读未命中(read miss)和紧急回写请求输出到帧缓冲器接口225以供处理。也可以经由帧缓冲器接口225将更新发送到帧缓冲器以供处理。在一个实施例中,帧缓冲器接口225与并行处理器存储器中的存储器单元(诸如,图2的存储器单元224A-224N(例如,在并行处理器存储器222内))之一进行接口。

[0051] 在图形应用中,ROP 226是执行诸如模板印刷(stencil)、z测试、混合、和诸如此类的光栅操作的处理单元。随后ROP 226输出存储在图形存储器中的处理的图形数据。在一些实施例中,ROP 226包括压缩逻辑,该压缩逻辑用于压缩写入到存储器的深度或颜色数据,并且解压缩从存储器读取的深度或颜色数据。压缩逻辑可以是利用多种压缩算法中的一种或多种压缩算法的无损压缩逻辑。由ROP 226所执行的压缩的类型可以基于待压缩的数据的统计特性而变化。例如,在一个实施例中,△颜色压缩逐图块地对深度和颜色数据来执行。

[0052] 在一些实施例中,ROP 226被包括在每个处理集群(例如,图2的集群214A-214N)内而非包括在分区单元220内。在此类实施例中,通过存储器交叉开关216来传送针对像素数据的读和写请求而非像素片段数据。处理的图形数据可在显示装置(诸如,图1的所述一个或多个显示装置110之一)上显示、被路由以供由(一个或多个)处理器102进一步处理、或被路由以供由图2A的并行处理器200内的处理实体之一进一步处理。

[0053] 图2C是根据实施例的并行处理单元内的处理集群214的框图。在一个实施例中,处理集群是图2的处理集群214A-214N之一的实例。处理集群214可以配置成并行执行许多线程,其中,术语“线程”是指在具体一组输入数据上执行的具体程序的实例。在一些实施例中,在不提供多个独立指令单元的情况下,使用单指令多数据(SIMD)指令发布技术以支持对大量线程的并行执行。在其他实施例中,使用配置成将指令发布到处理集群中的每一者内的一组处理引擎的公共指令单元,使用单指令多线程(SIMT)技术以支持大量一般同步的线程的并行执行。不同于SIMD执行制度(其中,所有处理引擎通常执行相同的指令),SIMT执行允许不同线程更容易沿着发散式(divergent)执行路径而通过给定线程程序。本领域技术人员将理解,SIMD处理制度表示SIMT处理制度的功能子集。

[0054] 可以经由流水线管理器232来控制处理集群214的操作,所述流水线管理器将处理任务分布到SIMT并行处理器。流水线管理器232从图2的调度器210接收指令,并且经由图形多处理器234和/或纹理单元236来管理那些指令的执行。所示出的图形多处理器234是SIMT并行处理器的示例性实例。然而,不同架构的各种类型的SIMT并行处理器可被包括在处理集群214内。图形多处理器234的一个或多个实例可以被包括在处理集群214内。图形多处理器234可以处理数据,并且数据交叉开关(data crossbar) 240可以用于将处理的数据分布到多个可能目的地(包括其他着色器单元)之一。流水线管理器232可以通过指定要经由数据交叉开关240来分布的处理的的数据的目的地来促进处理的数据的分布。

[0055] 处理集群214内的每一个图形多处理器234都可以包括相同一组功能执行逻辑(例如,算术逻辑单元、加载-存储单元等)。能以流水线方式(pipelined manner)来配置功能执

行逻辑,在流水线方式中,在先前的指令完成之前,可发布新指令。功能执行逻辑支持多种多样的操作,包括整数和浮点算术、比较操作、布尔操作、位移位(bit-shifting)和各种代数函数的计算。在一个实施例中,可以利用相同功能单元硬件来执行不同操作,并且可以存在功能单元的任何组合。

[0056] 传送至处理集群214的指令构成线程。跨一组并行处理引擎而执行的一组线程是线程群组。线程群组对不同的输入数据执行相同程序。可以将线程群组内的每一个线程分配给图形多处理器234内的不同处理引擎。线程群组可包括比图形多处理器234内的处理引擎的数量更少的线程。当线程群组包括比处理引擎的数量更少的线程时,处理引擎中的一个或多个在线程群组正在被处理的循环期间可以是空闲的。线程群组也可包括比图形多处理器234内的处理引擎的数量更多的线程。当线程群组包括比图形多处理器234内的处理引擎的数量更多的线程时,处理可以在连续时钟循环上执行。在一个实施例中,可在图形多处理器234上并发地执行多个线程群组。

[0057] 在一个实施例中,图形多处理器234包括用于执行加载和存储操作的内部高速缓存存储器。在一个实施例中,图形多处理器234可以放弃内部高速缓存,并且使用处理集群214内的高速缓存存储器(例如,L1高速缓存308)。每个图形多处理器234还有权访问在所有处理集群214中共享并且可用于在线程之间转移数据的分区单元(例如,图2的分区单元220A-220N)内的L2高速缓存。图形多处理器234还可访问芯片外全局存储器,所述芯片外全局存储器可以包括本地并行处理器存储器和/或系统存储器中的一者或多者。可将在并行处理单元202外部的任何存储器用作全局存储器。实施例(其中处理集群214包括图形多处理器234的多个实例)可以共享公共指令和数据,所述公共指令和数据可存储在L1高速缓存308中。

[0058] 每个处理集群214可包括MMU 245(存储器管理单元),所述MMU配置成将虚拟地址映射到物理地址中。在其他实施例中,MMU 245的一个或多个实例可驻留在图2的存储器接口218内。MMU 245包括:一组页表条目(PTE),用于将图块(更多地讨论分图块(tiling))的虚拟地址映射到物理地址;以及可选地高速缓存行索引。MMU 245可包括可驻留在图形多处理器234或L1高速缓存或处理集群214内的地址转换后备缓冲器(TLB)或高速缓存。物理地址经处理以分布表面数据访问局域性(locality),从而允许在分区单元中高效的请求交织。高速缓存行索引可用于确定针对高速缓存行的请求是命中还是未命中。

[0059] 在图形和计算应用中,处理集群214可被配置使得每个图形多处理器234耦合至纹理单元236以用于执行纹理映射操作,例如确定纹理样本位置、读取纹理数据和过滤纹理数据。根据需要,从内部纹理L1高速缓存(未示出)或在一些实施例中从图形多处理器234内的L1高速缓存读取纹理数据,并且从L2高速缓存、本地并行处理器存储器或系统存储器获取所述纹理数据。每个图形多处理器234将处理的任务输出到数据交叉开关240以将处理的任务提供给另一个处理集群214,以供进一步处理或以经由存储器交叉开关216将处理的任务存储在L2高速缓存、本地并行处理器存储器或系统存储器中。preROP 242(例如,预光栅操作单元)配置成从图形多处理器234接收数据、将数据导引到ROP单元,所述ROP单元可与如本文描述的分区单元(例如,图2的分区单元220A-220N)位于一起。preROP 242单元可以执行针对颜色混合的优化、组织像素颜色数据和执行地址转换。

[0060] 将认识到的是,本文描述的核架构是示出性的,并且变化和修改是有可能的。任何

数量的处理单元(例如,图形多处理器234、纹理单元236、preROP 242等)可被包括在处理集群214内。此外,虽然仅示出了一个处理集群214,但是如本文描述的并行处理单元可以包括处理集群214的任何数量的实例。在一个实施例中,每个处理集群214可以配置成使用单独且截然不同的处理单元、L1高速缓存等独立于其他处理集群214来操作。

[0061] 图2D示出了根据一个实施例的图形多处理器234。在此类实施例中,图形多处理器234与处理集群214的流水线管理器232耦合。图形多处理器234具有执行流水线,其包括但不限于:指令高速缓存252、指令单元254、地址映射单元256、寄存器堆258、一个或多个通用图形处理单元(GPGPU)核262和一个或多个加载/存储单元266。GPGPU核262和加载/存储单元266经由存储器和高速缓存互连268与高速缓存存储器272和共享存储器270耦合。

[0062] 在一个实施例中,指令高速缓存252从流水线管理器232接收待执行的指令流。所述指令被高速缓存在指令高速缓存252中,并且由指令单元254分派以供执行。指令单元254可以将指令分派为线程群组(例如,线程束(warp)),其中线程群组的每个线程被指派给GPGPU核262内的一不同执行单元。指令可以通过指定统一地址空间内的地址来访问本地、共享或全局地址空间中的任一者。地址映射单元256可以用于将统一地址空间中的地址转换成可以由加载/存储单元266访问的截然不同的存储器地址。

[0063] 寄存器堆258为图形多处理器324的功能单元提供一组寄存器。寄存器堆258为连接到图形多处理器324的功能单元(例如,GPGPU核262、加载/存储单元266)的数据路径的操作数提供临时存储。在一个实施例中,在所述功能单元中的每一者之间划分寄存器堆258,使得每个功能单元分配有寄存器堆258的专用部分。在一个实施例中,在由图形多处理器324执行的不同线程束之间划分寄存器堆258。

[0064] GPGPU核262可以各自包括浮点单元(FPU)和/或整数算术逻辑单元(ALU),所述FPU和整数ALU用于执行图形多处理器324的指令。根据实施例,GPGPU核262可以在架构上是类似的,或可以在架构上是不同的。例如且在一个实施例中,GPGPU核262的第一部分包括单精度FPU和整数ALU,而GPGPU核的第二部分包括双精度FPU。在一个实施例中,FPU可以针对浮点算术来实现IEEE 754-2008标准,或能够实现可变精度浮点算术。图形多处理器324可以另外包括一个或多个固定功能或特殊功能单元以执行特定功能(诸如,复制矩形或像素混合操作)。在一个实施例中,GPGPU核中的一者或多者也可以包括固定或特殊功能逻辑。

[0065] 在一个实施例中,GPGPU核262包括能够对多组数据执行单指令的SIMD逻辑。在一个实施例中,GPGPU核262可以物理地执行SIMD4、SIMD8和SIMD16指令,并且逻辑地执行SIMD1、SIMD2和SIMD32指令。用于GPGPU核的SIMD指令可以由着色器编译器在编译时间生成,或者可以在执行为单程序多数据(SPMD)或SIMT架构编写和编译的程序时自动生成。为SIMT执行模型而配置的程序的多个线程可以经由单SIMD指令而执行。例如且在一个实施例中,执行相同或类似操作的八个SIMT线程可以经由单个SIMD8逻辑单元并行地执行。

[0066] 存储器和高速缓存互连268是互连网络,其将图形多处理器234的功能单元中的每一个连接到寄存器堆258并连接到共享存储器270。在一个实施例中,存储器和高速缓存互连268是交叉开关互连,其允许加载/存储单元266在共享存储器270与寄存器堆258之间实现加载和存储操作。寄存器堆258能与GPGPU核262以相同频率来操作,由此在GPGPU核262与寄存器堆258之间的数据转移是非常低等待时间的。共享存储器270可以用来启用在图形多处理器234内的功能单元上执行的线程之间的通信。高速缓存存储器272可以用作例如数据

高速缓存,以便对功能单元与纹理单元236之间通信的纹理数据进行高速缓存。共享存储器270也可以用作程序管理的高速缓存。在GPGPU核262上执行的线程能以程序方式将除了存储在高速缓存存储器272内的经自动高速缓存的数据之外的数据存储在共享存储器内。

[0067] 图3A-图3B示出了根据实施例的附加图形多处理器。所示出的图形多处理器325、350是图2C的图形多处理器234的变体。所示出的图形多处理器325、350可以被配置为能够同时执行大量执行线程的流传送多处理器(SM)。

[0068] 图3A示出了根据附加实施例的图形多处理器325。图形多处理器325相对于图2D的图形多处理器234包括执行资源单元的多个附加实例。例如,图形多处理器325可以包括指令单元332A-332B、寄存器堆334A-334B和纹理单元344A-344B的多个实例。图形多处理器325还包括多组图形或计算执行单元(例如,GPGPU核336A-336B、GPGPU核337A-337B、GPGPU核338A-338B)和多组加载/存储单元340A-340B。在一个实施例中,执行资源单元具有公共指令高速缓存330、纹理和/或数据高速缓存存储器342以及共享存储器346。

[0069] 各种部件可以经由互连结构327通信。在一个实施例中,互连结构327包括一个或多个交叉开关(crossbar switch)以启用图形多处理器325的各种部件之间的通信。在一个实施例中,互连结构327是分开的高速网络结构层,图形多处理器325的每一个部件堆叠在该高速网络结构层上。图形多处理器325的部件经由互连结构327与远程部件通信。例如,GPGPU核336A-336B、337A-337B以及3378A-338B可以各自经由互连结构327与共享存储器346通信。互连结构327可以仲裁图形多处理器325内的通信以确保部件之间的公平带宽分配。

[0070] 图3B示出了根据附加实施例的图形多处理器350。图形处理器包括多组执行资源356A-356D,其中,每一组执行资源包括多个指令单元、寄存器堆、GPGPU核和加载存储单元,如图2D和图3A中所示出。执行资源356A-356D可以与纹理单元360A-360D一致地工作以用于纹理操作,同时共享指令高速缓存354和共享存储器362。在一个实施例中,执行资源356A-356D可以共享指令高速缓存354和共享存储器362以及纹理和/或数据高速缓存存储器358A-358B的多个实例。各种部件可以经由类似于图3A的互连结构327的互连结构352来通信。

[0071] 本领域技术人员将理解,图1、2A-2D以及3A-3B中所描述的架构就本实施例的范畴而言是描述性的而非限制性的。因此,在不背离本文描述的实施例的范畴的情况下,本文描述的技术可在任何正确配置的处理单元上实现,所述处理单元包括但不限于一个或多个移动应用处理器、一个或多个桌上型计算机或服务器中央处理单元(CPU)(包括多核CPU)、一个或多个并行处理单元(诸如,图2的并行处理单元202)以及一个或多个图形处理器或专用处理单元。

[0072] 在一些实施例中,如本文描述的并行处理器或GPGPU通信地耦合至主机/处理器核以加速图形操作、机器学习操作、模式分析操作和各种通用GPU(GPGPU)功能。GPU可通过总线或另一互连(例如,诸如PCIe或NVLink的高速互连)通信地耦合至主机处理器/核。在其他实施例中,GPU可集成在与所述核相同的封装或芯片上,并且通过内部处理器总线/互连(即,在所述封装或芯片的内部)通信地耦合至所述核。不管连接GPU的方式,处理器核都可用工作描述符中所包含的命令/指令的序列的形式将工作分配给GPU。GPU然后使用专用的电路/逻辑来高效地处理这些命令/指令。

[0073] 用于GPU至主机处理器互连的技术

[0074] 图4A示出了示例性架构,其中多个GPU 410-413通过高速链路440-443(例如,总线、点对点互连等)通信地耦合至多个多核处理器405-406。在一个实施例中,取决于实现,高速链路440-443支持4GB/s、30GB/s、80GB/s或更高的通信吞吐量。可使用各种互连协议,包括但不限于PCIe 4.0或5.0以及NVLink2.0。然而,本发明的根本原理不限于任何具体通信协议或吞吐量。

[0075] 另外,在一个实施例中,GPU 410-413中的两者或更多者通过高速链路444-445互连,所述高速链路可使用与用于高速链路440-443的协议/链路相同或不同的协议/链路来实现。类似地,多核处理器405-406中的两者或更多者可通过高速链路433相连接,所述高速链路433可以是以20GB/s、30GB/s、120GB/s或更高来操作的对称多处理器(SMP)总线。可替代地,图4A中所示的各种系统部件之间的所有通信可使用相同协议/链路(例如,通过公共互连结构)来取得。然而,如所提到,本发明的根本原理不限于任何具体类型的互连技术。

[0076] 在一个实施例中,每个多核处理器405-406分别经由存储器互连430-431通信地耦合至处理器存储器401-402,并且每个GPU 410-413分别通过GPU存储器互连450-453通信地耦合至GPU存储器420-423。存储器互连430-431以及450-453可利用相同或不同的存储器访问技术。通过示例且非限制的方式,处理器存储器401-402和GPU存储器420-423可以是易失性存储器,诸如动态随机存取存储器(DRAM)(包括堆叠式DRAM)、图形DDR SDRAM(GDDR)(例如,GDDR5、GDDR6)或高带宽存储器(HBM),和/或可以是非易失性存储器,诸如3D XPoint或Nano-Ram。在一个实施例中,存储器的某一部分可以是易失性存储器,并且另一部分可以是非易失性存储器(例如,使用两级存储器(2LM)分层)。

[0077] 如下文所描述,虽然各种处理器405-406和GPU 410-413可以分别物理地耦合至具体存储器401-402、420-423,但是可实现统一存储器架构,其中相同虚拟系统地址空间(也称为“有效地址”空间)被分布在所有各个物理存储器中。例如,处理器存储器401-402可各自包括64GB的系统存储器地址空间,并且GPU存储器420-423可各自包括32GB的系统存储器地址空间(在本示例中产生总共256GB的可寻址存储器)。

[0078] 图4B示出了依照一个实施例的针对多核处理器407与图形加速模块446之间的互连的附加细节。图形加速模块446可包括集成在线卡上的一个或多个GPU芯片,所述线卡经由高速链路440耦合至处理器407。可替代地,图形加速模块446可集成在与处理器407相同的封装或芯片上。

[0079] 所示出的处理器407包括多个核460A-460D,其各自具有转换后备缓冲器(translation lookaside buffer)461A-461D和一个或多个高速缓存462A-462D。所述核可包括用于执行指令和处理数据的各种其他部件(例如,指令获取单元、分支预测单元、解码器、执行单元、重排序缓冲器等),未示出所述其他部件以避免使本发明的根本原理难以理解。高速缓存462A-462D可包括1级(L1)和2级(L2)高速缓存。另外,一个或多个共享的高速缓存426可被包括在高速缓存分层中并且由多组核460A-460D共享。例如,处理器407的一个实施例包括24个核,其各自具有其自身的L1高速缓存、十二个共享的L2高速缓存和十二个共享的L3高速缓存。在本实施例中,L2和L3高速缓存中的一者由两个邻近的核共享。处理器407和图形加速器集成模块446与系统存储器441连接,所述系统存储器441可包括处理器存储器401-402。

[0080] 通过一致性总线464经由核间通信来维持存储在各种高速缓存462A-462D、456和系统存储器441中的数据和指令的一致性。例如,每个高速缓存可具有与其相关联的高速缓存一致性逻辑/电路以响应于到具体高速缓存行的检测到的读或写来通过一致性总线464进行通信。在一个实现中,通过一致性总线464来实现高速缓存监听协议,以监听高速缓存访问。高速缓存监听/一致性技术是被本领域技术人员良好理解的,并且此处将不详细描述以避免使本发明的根本原理难以理解。

[0081] 在一个实施例中,代理电路425将图形加速模块446通信地耦合至一致性总线464,从而允许图形加速模块446作为核的对等物来参与高速缓存一致性协议。具体而言,接口435提供通过高速链路440(例如,PCIe总线、NVLink等)至代理电路425的连接性,并且接口437将图形加速模块446连接到链路440。

[0082] 在一个实现中,加速器集成电路436代表图形加速模块446的多个图形处理引擎431、432、N来提供高速缓存管理、存储器访问、上下文管理和中断管理服务。图形处理引擎431、432、N可各自包括单独的图形处理单元(GPU)。可替代地,图形处理引擎431、432、N可包括GPU内不同类型的图形处理引擎,诸如图形执行单元、媒体处理引擎(例如,视频编码器/解码器)、采样器和块图转移(blit)引擎。换句话说,图形加速模块可以是具有多个图形处理引擎431-432、N的GPU,或图形处理引擎431-432、N可以是集成在公共封装、线卡或芯片上的个别GPU。

[0083] 在一个实施例中,加速器集成电路436包括存储器管理单元(MMU)439以用于执行各种存储器管理功能,诸如虚拟至物理存储器转换(也称为有效至真实存储器转换)和用于访问系统存储器441的存储器访问协议。MMU 439还可包括转换后备缓冲器(TLB)(未示出)以用于高速缓存虚拟/有效至物理/真实地址转换。在一个实现中,高速缓存438存储命令和数据以供由图形处理引擎431-432、N进行高效访问。在一个实施例中,存储在高速缓存438和图形存储器433-434、N中的数据与核高速缓存462A-462D、456以及系统存储器411保持一致。如所提到,这可经由代理电路425来取得,所述代理电路425代表高速缓存438和存储器433-434、N来参与高速缓存一致性机制(例如,将与处理器高速缓存462A-462D、456上的高速缓存行的修改/访问有关的更新发送到高速缓存438,以及从高速缓存438接收更新)。

[0084] 一组寄存器445存储用于由图形处理引擎431-432、N执行的线程的上下文数据,并且上下文管理电路448管理线程上下文。例如,上下文管理电路448可在上下文切换期间执行保存和恢复操作以保存和恢复各种线程的上下文(例如,其中,第一线程被保存并且第二线程被存储,使得可以由图形处理引擎来执行第二线程)。例如,在上下文切换时,上下文管理电路448可将当前寄存器值存储到存储器中指派的区域(例如,由上下文指针标识的)。然后,其可在返回到上下文时恢复所述寄存器值。在一个实施例中,中断管理电路447接收并处理从系统装置接收的中断。

[0085] 在一个实现中,由MMU 439将来自图形处理引擎431的虚拟/有效地址转换成系统存储器411中的真实/物理地址。加速器集成电路436的一个实施例支持多个(例如,4、8、16个)图形加速器模块446和/或其他加速器装置。图形加速器模块446可专用于在处理器407上执行的单个应用,或可在多个应用之间共享。在一个实施例中,呈现虚拟化的图形执行环境,其中与多个应用或虚拟机(VM)共享图形处理引擎431-432、N的资源。所述资源可被再分成“切片”,其基于与不同的VM和/或应用相关联的处理要求和优先级来被分配给所述VM和/

或应用。

[0086] 因此,加速器集成电路充当到图形加速模块446的系统的桥,并且提供地址转换和系统存储器高速缓存服务。另外,加速器集成电路436可为主机处理器提供虚拟化设施,以管理图形处理引擎的虚拟化、中断和存储器管理。

[0087] 由于图形处理引擎431-432、N的硬件资源被显式映射到由主机处理器407所见的真实地址空间,所以任何主机处理器都可以使用有效地址值来直接寻址这些资源。在一个实施例中,加速器集成电路436的一个功能是图形处理引擎431-432、N的物理分离,使得它们对系统呈现为独立单元。

[0088] 如所提到,在所示出的实施例中,一个或多个图形存储器433-434、M分别耦合至图形处理引擎431-432、N中的每一者。图形存储器433-434、M存储正由图形处理引擎431-432、N中的每一者处理的指令和数据。图形存储器433-434、M可以是易失性存储器,诸如DRAM(包括堆叠式DRAM)、GDDR存储器(例如,GDDR5、GDDR6)或HBM,和/或可以是非易失性存储器,诸如3D XPoint或Nano-Ram。

[0089] 在一个实施例中,为减少链路440上的数据业务,使用偏置技术以确保存储在图形存储器433-434、M中的数据是将被图形处理引擎431-432、N使用最频繁并且优选地不被核460A-460D使用(至少不是频繁地)的数据。类似地,偏置机制试图将由核(且优选地不是图形处理引擎431-432、N)所需的数据保存在系统存储器411和所述核的高速缓存462A-462D、456内。

[0090] 图4C示出了另一个实施例,其中加速器集成电路436被集成在处理器407内。在本实施例中,图形处理引擎431-432、N经由接口437和接口435(再次,其可利用任何形式的总线或接口协议)通过高速链路440来直接通信到加速器集成电路436。加速器集成电路436可执行与关于图4B所描述的那些操作相同的操作,但考虑到其与一致性总线462和高速缓存462A-462D、426的紧密接近性而潜在地处于更高的吞吐量。

[0091] 一个实施例支持不同的编程模型,包括专用进程编程模型(没有图形加速模块虚拟化)和共享的编程模型(有虚拟化)。后者可包括由加速器集成电路436控制的编程模型和由图形加速模块446控制的编程模型。

[0092] 在专用进程模型的一个实施例中,图形处理引擎431-432、N在单一操作系统下专用于单个应用或进程。单个应用可以将另一应用请求汇集(funnel)到图形引擎431-432、N,从而在VM/分区内提供虚拟化。

[0093] 在专用进程编程模型中,可由多个VM/应用分区来共享图形处理引擎431-432、N。共享的模型需要系统管理程序来虚拟化图形处理引擎431-432、N,以允许由每个操作系统进行访问。针对无管理程序的单一分区系统,图形处理引擎431-432、N由操作系统所拥有。在两种情况下,操作系统可以虚拟化图形处理引擎431-432、N以提供对每个进程或应用的访问。

[0094] 针对共享的编程模型,图形加速模块446或各个图形处理引擎431-432、N使用进程句柄来选择进程元素(process element)。在一个实施例中,进程元素存储在系统存储器411中,并且使用本文描述的有效地址至真实地址转换技术是可寻址的。进程句柄可以是在向图形处理引擎431-432、N登记其上下文(即,调用系统软件以将进程元素添加到进程元素链表)时被提供给主机进程的特定值。进程句柄的较低的16位可以是进程元素链表

内的进程元素的偏移。

[0095] 图4D示出了示例性加速器集成切片490。如本文所使用,“切片”包括加速器集成电路436的处理资源的指定部分。系统存储器411内的应用有效地址空间482存储进程元素483。在一个实施例中,响应于来自处理器407上执行的应用480的GPU调用481来存储进程元素483。进程元素483包含对应的应用480的进程状态。包含在进程元素483中的工作描述符(WD) 484可以是由应用请求的单个作业,或可包含指向作业队列的指针。在后一种情况下,WD 484是指向应用的地址空间482中的作业请求队列的指针。

[0096] 图形加速模块446和/或个别图形处理引擎431-432、N可以由系统中的所有进程或进程子集共享。本发明的实施例包括用于设置进程状态并将WD 484发送到图形加速模块446以在虚拟化环境中开始作业的基础设施。

[0097] 在一个实现中,专用进程编程模型是实现特定的。在这个模型中,单个进程拥有图形加速模块446或个别图形处理引擎431。由于图形加速模块446由单个进程所拥有,所以在指派图形加速模块446的时候,管理程序针对拥有的分区来初始化加速器集成电路436并且操作系统针对拥有的进程来初始化加速器集成电路436。

[0098] 在操作中,加速器集成切片490中的WD获取单元491获取下一个WD 484,所述下一个WD 484包括待由图形加速模块446的图形处理引擎之一来完成的工作的指示。来自WD 484的数据可存储在寄存器445中,并且由如所示出的MMU 439、中断管理电路447和/或上下文管理电路446使用。例如,MMU 439的一个实施例包括用于访问OS虚拟地址空间485内的段/页表486的段/页行走电路(walk circuitry)。中断管理电路447可处理从图形加速模块446接收的中断事件492。当执行图形操作时,由MMU 439将由图形处理引擎431-432、N所生成的有效地址493转换为真实地址。

[0099] 在一个实施例中,针对每个图形处理引擎431-432、N和/或图形加速模块446复制相同一组寄存器445,并且其可由管理程序或操作系统来初始化。这些复制的寄存器中的每一者可被包括在加速器集成切片490中。表1中示出了可由管理程序来初始化的示例性寄存器。

[0100] 表1-管理程序初始化的寄存器

[0101]	1	切片控制寄存器
	2	真实地址 (RA) 调度的进程区域指针
	3	权限掩蔽覆盖寄存器
	4	中断向量表条目偏移
	5	中断向量表条目限制
	6	状态寄存器
	7	逻辑分区ID
	8	真实地址 (RA) 管理程序加速器利用记录指针
	9	存储描述寄存器

[0102] 表2中示出了可由操作系统来初始化的示例性寄存器。

[0103] 表2-操作系统初始化的寄存器

[0104]	1	进程和线程标识
	2	有效地址 (EA) 上下文保存/恢复指针

3	虚拟地址 (VA) 加速器利用记录指针
4	虚拟地址 (VA) 存储段表指针
5	权限掩蔽
6	工作描述符

[0105] 在一个实施例中,每个WD 484是特定于具体图形加速模块446和/或图形处理引擎431-432、N的。它包含图形处理引擎431-432、N完成其工作所需的全部信息,或它可以是对其中应用已设立待完成的工作的命令队列的存储器位置的指针。

[0106] 图4E示出了共享模型的一个实施例的附加细节。本实施例包括其中存储有进程元素列表499的管理程序真实地址空间498。管理程序真实地址空间498可经由管理程序496来访问,所述管理程序496虚拟化用于操作系统495的图形加速模块引擎。

[0107] 共享的编程模型允许来自系统中所有分区或分区子集的所有进程或进程子集使用图形加速模块446。存在两个编程模型,其中,图形加速模块446由多个进程和分区共享:时间切片共享和图形定向共享(graphics directed shared)。

[0108] 在这个模型中,系统管理程序496拥有图形加速模块446,并且使其功能可用于所有操作系统495。为使图形加速模块446支持由系统管理程序496进行的虚拟化,图形加速模块446可遵守以下要求:1) 应用的作业请求必须是自主的(即,无需在作业之间保持状态),或图形加速模块446必须提供上下文保存和恢复机制。2) 由图形加速模块446保证在指定时间量内完成应用的作业请求(包括任何转换故障),或图形加速模块446提供抢占作业的处理的能力。3) 当在定向共享编程模型中操作时,必须保证图形加速模块446在进程之间的公平性。

[0109] 在一个实施例中,针对共享模型,需要应用480用图形加速模块446类型、工作描述符(WD)、权限掩蔽寄存器(AMR)值和上下文保存/恢复区域指针(CSRP)来进行操作系统495系统调用。图形加速模块446类型描述了用于系统调用的针对的加速函数。图形加速模块446类型可以是系统特定的值。WD专门针对图形加速模块446被格式化,并且可以呈图形加速模块446命令、对用户定义的结构的有效地址指针、对命令队列的有效地址指针或用于描述待由图形加速模块446完成的工作的任何其他数据结构的形式。在一个实施例中,AMR值是待用于当前进程的AMR状态。被传递到操作系统的值类似于设定AMR的应用。如果加速器集成电路436和图形加速模块446实现不支持用户权限掩蔽覆盖寄存器(UAMOR),那么操作系统可将当前UAMOR值应用于AMR值,之后在管理程序调用中传递AMR。可选地,管理程序496可应用当前权限掩蔽覆盖寄存器(AMOR)值,之后将AMR放到进程元素483中。在一个实施例中,CSR是寄存器445之一,其包含应用的地址空间482中的区域的有效地址以用于使图形加速模块446保存和恢复上下文状态。如果不需要在作业之间保存状态或当作业被抢占时,这个指针是可选的。上下文保存/恢复区域可以是固定的(pinned)系统存储器。

[0110] 在接收到系统调用时,操作系统495可验证应用480已注册并且已被给予使用图形加速模块446的权限。然后,操作系统495用表3中所示的信息来调用管理程序496。

[0111] 表3-OS至管理程序调用参数

[0112]	1	工作描述符(WD)
	2	权限掩蔽寄存器(AMR)值(潜在地被掩蔽)
	3	有效地址(EA)上下文保存/恢复区域指针(CSRP)

4	进程ID (PID) 和可选线程ID (TID)
5	虚拟地址 (VA) 加速器利用记录指针 (AURP)
6	存储段表指针 (SSTP) 的虚拟地址
7	逻辑中断服务号 (LISN)

[0113] 在接收到管理程序调用时,管理程序496验证操作系统495已注册并且已被给予使用图形加速模块446的权限。然后,管理程序496将进程元素483放入对应的图形加速模块446类型的进程元素链表中。进程元素可包括表4中所示的信息。

[0114] 表4-进程元素信息

[0115]	1	工作描述符 (WD)
	2	权限掩蔽寄存器 (AMR) 值 (潜在地被掩蔽)
	3	有效地址 (EA) 上下文保存/恢复区域指针 (CSRP)
	4	进程ID (PID) 和可选线程ID (TID)
	5	虚拟地址 (VA) 加速器利用记录指针 (AURP)
	6	存储段表指针 (SSTP) 的虚拟地址
	7	逻辑中断服务号 (LISN)
	8	从管理程序调用参数导出的中断向量表
	9	状态寄存器 (SR) 值
	10	逻辑分区ID (LPID)
	11	真实地址 (RA) 管理程序加速器利用记录指针
	12	存储描述符寄存器 (SDR)

[0116] 在一个实施例中,管理程序初始化多个加速器集成切片490寄存器445。

[0117] 如图4F中所示出,本发明的一个实施例采用经由公共虚拟存储器地址空间可寻址的统一存储器,所述公共虚拟存储器地址用于访问物理处理器存储器401-402和GPU存储器420-423。在这种实现中,在GPU 410-413上执行的操作利用相同的虚拟/有效存储器地址空间来访问处理器存储器401-402且反之亦然,由此简化可编程性。在一个实施例中,虚拟/有效地址空间的第一部分被分配给处理器存储器401,第二部分被分配给第二处理器存储器402,第三部分被分配GPU存储器420,等等。由此跨处理器存储器401-402和GPU存储器420-423中的每一者来分布整个虚拟/有效存储器空间(有时称为有效地址空间),从而允许任何处理器或GPU来访问任一物理存储器(用被映射到该存储器的虚拟地址)。

[0118] 在一个实施例中,在MMU 439A-439E中的一者或多者内的偏置/一致性管理电路494A-494E确保主机处理器(例如,405)与GPU 410-413的高速缓存之间的高速缓存一致性,并且实现指示其中应存储有某些类型的数据的物理存储器的偏置技术。虽然图4F中示出了偏置/一致性管理电路494A-494E的多个实例,但是可在一个或多个主机处理器405的MMU内和/或在加速器集成电路436内实现偏置/一致性电路。

[0119] 一个实施例允许使用共享虚拟存储器 (SVM) 技术来访问GPU附连的存储器420-423并将其映射为系统存储器的一部分,而不用经受与完全系统高速缓存一致性相关联的典型性能缺陷。GPU附连的存储器420-423作为系统存储器来访问而无繁重的高速缓存一致性开销 (overhead) 的能力为GPU卸载提供了有益的操作环境。这种安排允许主机处理器405软件来设置操作数和访问计算结果,而没有传统I/O DMA数据拷贝的开销。此类传统拷贝涉及驱

动器调用、中断和存储器映射I/O (MMIO) 访问,其相对于简单的存储器访问来说全部都是低效的。同时,访问GPU附连的存储器420-423而无高速缓存一致性开销的能力对于卸载的计算的运行时间来说可以是关键的。在具有实质流传送写存储器业务的情况下,例如,高速缓存一致性开销可以显著减少由GPU 410-413所见的有效写带宽。操作数设置的效率、结果访问的效率和GPU计算的效率在确定GPU卸载的有效性中全部都起到一定的作用。

[0120] 在一个实现中,由偏置跟踪器数据结构来驱动在GPU偏置与主机处理器偏置之间的选择。可使用偏置表,例如,其可以是每GPU附连的存储器页包括1或2位的页粒度结构(即,被控制在存储器页的粒度)。可以在一个或多个GPU附连的存储器420-423的被偷的(stolen)存储器范围中实现偏置表,其中在GPU 410-413中具有或不具有偏置高速缓存(例如,用于高速缓存偏置表的频繁/最近使用的条目)。可替代地,可将整个偏置表维持在GPU内。

[0121] 在一个实现中,在实际访问GPU存储器之前访问与每一次访问GPU附连的存储器420-423相关联的偏置表条目,从而引起以下操作。首先,来自GPU 410-413的在主机偏置中寻找其页的本地请求被直接转发到对应的GPU存储器420-423。来自GPU的本地请求被转发到处理器405(例如,通过如上文所讨论的高速链路)。在一个实施例中,来自处理器405的请求(所述请求在主机处理器偏置中寻找请求的页)完成类似于正常存储器读取的请求。可替代地,可将针对GPU偏置的页的请求转发到GPU 410-413。然后,如果GPU当前未正使用该页,那么GPU可将该页转变到主机处理器偏置。

[0122] 可以由基于软件的机制、硬件辅助的基于软件的机制抑或针对有限一组情况由纯粹基于硬件的机制来改变页的偏置状态。

[0123] 用于改变偏置状态的一个机制采用API调用(例如,OpenCL),其进而调用GPU的装置驱动器,所述装置驱动器进而发送消息(或入队命令描述符)到GPU,从而指导其改变偏置状态并且针对一些转变在主机中执行高速缓存转储清除操作。高速缓存转储清除操作对于从主机处理器405偏置转变到GPU偏置来说是需要的,但对于反向转变来说是不需要的。

[0124] 在一个实施例中,通过暂时渲染由主机处理器405不可高速缓存的GPU偏置页来维持高速缓存一致性。为了访问这些页,处理器405可请求来自GPU 410的访问,这取决于实现而可或可不立即授予访问权。因此,为减少处理器405与GPU 410之间的通信,确保GPU偏置页是被GPU所需但非被主机处理器405所需且反之亦然的那些页。

[0125] 图形处理流水线

[0126] 图5示出根据实施例的图形处理流水线500。在一个实施例中,图形处理器可以实现所示出的图形处理流水线500。所述图形处理器可以被包括在如本文描述的并行处理系统(诸如图2的并行处理器200)内,在一个实施例中,其是图1的(一个或多个)并行处理器112的变体。各种并行处理系统可以经由如本文描述的并行处理单元(例如,图2的并行处理单元202)的一个或多个实例来实现图形处理流水线500。例如,着色器单元(例如,图3的图形多处理器234)可配置成执行顶点处理单元504、曲面细分控制处理单元508、曲面细分评估处理单元512、几何处理单元516和片段/像素处理单元524中的一者或多者的功能。数据组装器502,图元组装器506、514、518,曲面细分单元510,光栅化器522和光栅操作单元526的功能也可由处理集群(例如,图3的处理集群214)内的其他处理引擎和对应的分区单元(例如,图2的分区单元220A-220N)来执行。还可使用用于一个或多个功能的专用处理单元

来实现图形处理流水线500。在一个实施例中,可以由通用处理器(例如,CPU)内的并行处理逻辑来执行图形处理流水线500的一个或多个部分。在一个实施例中,图形处理流水线500的一个或多个部分可以经由存储器接口528来访问芯片上存储器(例如,如图2中的并行处理器存储器222),所述存储器接口528可以是图2的存储器接口218的实例。

[0127] 在一个实施例中,数据组装器502是收集表面和图元的顶点数据的处理单元。数据组装器502随后将包括顶点属性的顶点数据输出至顶点处理单元504。顶点处理单元504是可编程执行单元,其执行顶点着色器程序,如由顶点着色器程序所指定那样来调亮(lighting)并变换顶点数据。顶点处理单元504读取存储在高速缓存、本地或系统存储器中的数据以供在处理顶点数据时使用,并且顶点处理单元504可被编程为将顶点数据从基于对象的坐标表示变换到世界空间坐标空间或归一化装置坐标空间。

[0128] 图元组装器506的第一实例从顶点处理单元504接收顶点属性。图元组装器506按需读取存储的顶点属性,并且构建图形图元以用于由曲面细分控制处理单元508来处理。图形图元包括如由各种图形处理应用编程接口(API)所支持的三角形、线段、点、补片等等。

[0129] 曲面细分控制处理单元508将输入顶点视为用于几何补片的控制点。控制点是来自补片(例如,补片的基)的输入表示变换到适合于由曲面细分评估处理单元512在表面评估中使用的表示。曲面细分控制处理单元508也可以计算几何补片的边缘的曲面细分因数。曲面细分因数应用于单个边缘,并且对与该边缘相关联的依赖于视图的细节级别进行量化。将曲面细分单元510配置成接收补片的边缘的曲面细分因数,并且将补片曲面细分成诸如线、三角形或四边形图元的多个几何图元,其被传送到曲面细分评估处理单元512。曲面细分评估处理单元512对再分的补片的参数化坐标进行操作以生成与几何图元相关联的每一个顶点的表面表示和顶点属性。

[0130] 图元组装器514的第二实例从曲面细分评估处理单元512接收顶点属性,按需读取存储的顶点属性,并且构建图形图元以用于由几何处理单元516来处理。几何处理单元516是可编程执行单元,其执行几何着色器程序以按几何着色器程序所指定的来变换从图元组装器514所接收的图形图元。在一个实施例中,将几何处理单元516编程为将图形图元再分成一个或多个新图形图元,并且运算用于对新图形图元进行光栅化的参数。

[0131] 在一些实施例中,几何处理单元516可在几何流中增加或删除元素。几何处理单元516将指定新图形图元的参数和顶点输出到图元组装器518。图元组装器518从几何处理单元516接收参数和顶点,并且构建图形图元以用于由视口缩放、拣选和剪辑单元520来处理。几何处理单元516读取存储在并行处理器存储器或系统存储器中的数据以供在处理几何数据时使用。视口缩放、拣选和剪辑单元520执行剪辑、拣选和视口缩放,并将处理的图形图元输出到光栅化器522。

[0132] 光栅化器522可以执行深度拣选和其他基于深度的优化。光栅化器522还对新图形图元执行扫描转换以生成片段,并且将那些片段和相关联的覆盖数据输出到片段/像素处理单元524。片段/像素处理单元524是配置成执行片段着色器程序或像素着色器程序的可编程执行单元。片段/像素处理单元524按由片段或像素着色器程序所指定的来变换从光栅化器522所接收的片段或像素。例如,可将片段/像素处理单元524编程为执行以下操作以产生输出到光栅操作单元526的着色的片段或像素,所述操作包括但不限于纹理映射、着色、混合、纹理校正和透视校正。片段/像素处理单元524可以读取存储在并行处理器存储器或

系统存储器中的数据以供在处理片段数据时使用。可将片段或像素着色器程序配置成在样本、像素、图块或其他粒度(取决于为处理单元所配置的采样率)着色。

[0133] 光栅操作单元526是处理单元,其执行包括但不限于模板印刷、z测试、混合和诸如此类的光栅操作,并将像素数据作为处理的图形数据输出,以存储在图形存储器(例如,如图2中的并行处理器存储器222和/或如图1中的系统存储器104)中,以显示在所述一个或多个显示装置110上,或者供由(一个或多个)并行处理器112或所述一个或多个处理器102中的一个来进一步处理。在一些实施例中,将光栅操作单元526配置成压缩被写入到存储器的z或颜色数据,并且解压缩从存储器读取的z或颜色数据。

[0134] 拓扑着色器技术

[0135] 现在转到图6,示出图形流水线600,其中拓扑着色器602接收例如来自主机处理器(例如,中央处理单元/CPU)、应用编程接口(API)、驱动器等的对象描述604。对象描述604可在不指定要显示的对象的拓扑的情况下包含关于要渲染的对象的类型的一般信息(例如,带有确定中心和半径的球体、带有确定骨架形状的人、等)。拓扑着色器602可接收对象描述604,基于对象描述来生成一组多边形606(例如,三角形、图元),并将该组多边形606发送到顶点着色器608(“VS”)。因此,拓扑着色器602可在仅给出球体的中心坐标和半径的情况下确定球体的外表面的恰当拓扑。类似地,拓扑着色器602可在仅给出场景中人/角色的骨架形状和深度的情况下确定人/角色的皮肤的恰当拓扑。

[0136] 如将更详细讨论的,该组多边形606的生成可涉及相对于一个或多个多边形检测可视性条件的缺乏,以及排除从该组多边形606发送到VS 608的多边形。可视性条件的缺乏可例如包含:遮蔽状态(例如,在三维/3D场景中,多边形位于另一多边形之后),离屏状态(例如,多边形在视场外)等。生成该组多边形606可还包含基于深度信息(例如,球体中心z坐标)来确定与对象描述关联的细节级别(LOD)。深度信息可从与多边形关联的一个或多个可视性样本和/或对象描述中获得。

[0137] 关于LOD,场景的拓扑数据可被再分成八叉树数据结构,其中每个内部节点具有八个子节点(例如,通过将3D空间/模型递归地再分成八个卦限来分区3D空间/模型)。在此类情况中,拓扑着色器602可被调用以用于标识的子八叉树(例如,LOD)。使得图形流水线600能够确定正渲染的对象的拓扑可以通过减少与将拓扑数据从主机处理器转移到图形流水线600关联的存储器事物的数量来增加灵活性(例如,通过可变拓扑)、改善效率、提高性能并减小功率耗用。

[0138] 顶点着色器608可通过执行操作(诸如变换、蒙皮(skinning)、和调亮)来处理该组多边形606的顶点。顶点着色器608可产生其采用为输入的相同数量的顶点。另外,外壳着色器610(“HS”)可处理定义低阶表面(诸如线、三角形、或四边形)的控制点。作为输出,HS 610可产生高阶几何“补片”和补片常数,其被传递到固定功能曲面细分器612。曲面细分器612可预处理由HS 610的输出所表示的域。作为输出,曲面细分器612可创建域的采样模式和连接样本的一组小图元(例如,点、线、三角形)。所图示的流水线600还包含域着色器614(“DS”),其处理来自HS 610的高阶几何补片连同来自曲面细分器612的曲面细分因数。曲面细分因数可包含曲面细分器输入因数以及输出因数。作为输出,DS 614可运算符合曲面细分因数的输出补片上的点的顶点位置。

[0139] 另外,几何着色器616(“GS”)可处理整个图元(例如,点、线、或三角形)连同边缘相

邻的图元的可选顶点数据。不像VS 608,GS 616可产生比其采用为输入的更多或更少的图元。流水线600可还包含像素着色器618(“PS”),其处理光栅化的图元连同内插的顶点数据,以生成每像素的值(诸如颜色和深度)。所图示的流水线600因此不只依赖于曲面细分器612来生成要渲染的场景的表面表示。

[0140] 图7图示了可由诸如已经讨论的拓扑着色器602(图6)的拓扑着色器来处理的LOD分析。在所图示的示例中,在相对高的LOD(例如,包括相对大量的多边形的高保真度拓扑)、在八叉树的第一级别700(“级别_0”)表示球体。如果球体例如具有到摄像机和/或用户的短距离(例如,前景对象),则可选择第一级别700。相反,八叉树的第二级别702(“级别_1”)可在中间LOD(例如,包括中间数量的多边形)表示球体。因此,如果球体例如具有到摄像机和/或用户的中间距离,则可选择第二级别702。另外,可在相对低的LOD(例如,包括相对小量的多边形的低保真度拓扑)、在第三级别704(“级别_2”)表示球体。如果球体例如具有到摄像机和/或用户的长距离(例如,背景对象),则可选择第三级别704。拓扑着色器可因此在所选择的子八叉树主要基于深度信息来调用。如已经注释的,拓扑着色器可基于对象描述中限制的/基础的信息来确定适当的LOD。

[0141] 确定LOD可还涉及在两个LOD之间的内插。例如,对象描述可指示对三个图示级别700、702、704的支持,其中拓扑着色器基于球体与摄像机/用户的距离来选择第一级别700和第二级别702之间的级别。在此类情况中,所选择的LOD可以与实际距离成比例地相比第二级别702包括更多多边形和相比第一级别700包括更少多边形。

[0142] 图8示出相对于要渲染的场景的视场800。在所图示的示例中,场景中的对象802(例如,球体)仅部分在视场内。在此类情况中,由于对象802的一个或多个多边形804在视场800之外(例如,在“离屏状态”中),所以对象802的一个或多个多边形804可考虑成缺乏可视性条件。类似地,即使一个或多个其它多边形806在视场800内,但它们被更靠近位于摄像机/用户的多边形遮蔽,所以一个或多个其它多边形806可考虑成缺乏可视性条件。再次,拓扑着色器可基于例如从主机处理器获得的对象描述中的限制的/基础的信息来检测可视性条件的缺乏。

[0143] 图9A示出操作半导体封装设备的方法900。方法900可被实现为一组逻辑指令中的一个或多个模块,所述一组逻辑指令存储在非暂时性机器或计算机可读的存储介质中,诸如随机存取存储器(RAM)、只读存储器(ROM)、可编程ROM(PROM)、固件、闪速存储器等;存储在可配置逻辑中,诸如例如可编程逻辑阵列(PLA)、现场可编程门阵列(FPGA)、复杂可编程逻辑装置(CPLD);存储在使用电路技术的固定功能性硬件逻辑中,所述电路技术是诸如例如专用集成电路(ASIC)、互补金属氧化物半导体(CMOS)或晶体管-晶体管逻辑(TTL)技术,或者其任何组合。

[0144] 例如,要实行方法900中示出的操作的计算机程序代码可用一个或多个编程语言的任何组合编写,所述一个或多个编程语言包含面向对象的编程语言,诸如JAVA、SMALLTALK、C++或诸如此类,以及常规的程序性编程语言,诸如“C”编程语言或类似的编程语言。此外,逻辑指令可以包含组装机指令、指令集架构(ISA)指令、机器指令、机器依赖指令、微代码、状态设置数据、用于集成电路的配置数据,个人化电子电路和/或原生于硬件(例如主机处理器、中央处理单元/CPU、微控制器等)的其他结构部件的状态信息。

[0145] 图示的处理框902在拓扑着色器接收对象描述。所述对象描述可在不描述或指定

对象的拓扑的情况下提供对象的一般描述。框904可在拓扑着色器基于所述对象描述来生成一组多边形,其中该组多边形可在框906发送到顶点着色器。

[0146] 图9B示出生成一组多边形的方法910。方法910可因此通常被结合到已经讨论的框904(图9A)中。更具体地,方法910可被实现为一组逻辑指令中的一个或多个模块,所述一组逻辑指令存储在非暂时性机器或计算机可读的存储介质中,诸如RAM、ROM、PROM、固件、闪速存储器等;存储在可配置逻辑中,诸如例如PLA、FPGA、CPLD;存储在使用电路技术的固定功能性硬件逻辑中,所述电路技术是诸如例如ASIC、CMOS或TTL技术,或者其任何组合。

[0147] 图示的处理框912基于深度信息来确定与对象描述关联的细节级别。框912可例如包含标识对应于对象描述的子八叉树,其中拓扑着色器被调用以用于所标识的子八叉树。如已经讨论的,确定细节级别可还涉及在两个细节级别之间内插。可视性条件的缺乏可在框914相对于一个或多个多边形被检测,其中(一个或多个)多边形可在框916响应于可视性条件的缺乏而被排除出该组多边形。可视性条件的缺乏可包含例如遮蔽状态、离屏状态等、或其任何组合。

[0148] 图10A示出性能提高的计算系统100。在所示出的示例中,主机处理器1002包含集成存储器控制器(IMC)1004,其与系统存储器1006(例如DRAM)通信。主机处理器1002可耦合到可包含图形流水线1016的图形处理器1008和输入/输出(I/O)模块1010。所图示的图形处理器1008还耦合到专用图形存储器1022。I/O模块1010可耦合到网络控制器1012(例如无线的和/或有线的)、显示器1014(例如固定的或头戴式液晶显示器/LCD、发光二极管/LED显示器等,用于视觉上呈现3D场景的视频)、和海量存储装置1018(例如闪速存储器、光盘、固态驱动/SSD)。

[0149] 系统存储器1006和/或海量存储装置1018可包含指令1020,其当由主机处理器1002来执行时,促使主机处理器1002生成与要由显示器1014呈现的视觉内容关联的对象描述。另外,图形处理器1008的图形流水线1016可包含拓扑着色器,其执行方法900(图9A)和/或方法910(图9B)的一个或更多方面。因此,图形流水线1016可配置成在拓扑着色器接收对象描述,基于对象描述在拓扑着色器生成一组多边形,并将该组多边形发送到顶点着色器。所图示的解决方案因此减少系统存储器1006/海量存储装置1018和图形存储器1022之间的存储器转移的数量。因此,可取得更高的效率和性能,同时在系统1000中改善质量。

[0150] 图10B示出包含衬底1032(例如硅、蓝宝石、砷化镓)以及耦合到衬底1032的逻辑1034(1034a-1034c,例如晶体管阵列和其他集成电路/IC部件)的半导体封装设备1030(例如芯片)。可被实现在可配置逻辑和/或固定功能性逻辑硬件中的逻辑1034包含图形处理器1034a、主机处理器1034b和I/O模块1034c。逻辑1034可一般地实现方法900(图9A)和/或方法910(图9B)中的一个或多个方面。因而,逻辑1034可在(例如,图形流水线的)拓扑着色器接收对象描述,基于对象描述在拓扑着色器生成一组多边形,并将该组多边形发送到顶点着色器。

[0151] 在一个示例中,逻辑1034相对于一个或更多多边形检测可视性条件的缺乏(例如,屏蔽状态、离屏状态),并响应于可视性条件的缺乏将所述一个或更多多边形排除出该组多边形。另外,逻辑1034可配置成例如通过标识对应于对象描述的子八叉树来确定与对象描述关联的细节级别。逻辑1034可还在两个细节级别之间内插。

[0152] 头戴式显示器系统概述

[0153] 图11示出了正由用户在体验沉浸式环境时穿戴的头戴式显示器 (HMD) 系统1100, 沉浸式环境诸如例如虚拟现实 (VR) 环境、增强现实 (AR) 环境、多玩家三维 (3D) 游戏等。在所示出的示例中, 一个或多个带子1120将HMD系统1100的框架1102保持于用户的眼睛前面。相应地, 放置左眼显示器1104以被用户的左眼查看并且放置右眼显示器1106以被用户的右眼查看。在诸如例如由用户穿戴的智能电话的某些示例中, 可择一地将左眼显示器1104和右眼显示器1106集成进单个显示器中。在AR的情况中, 显示器1104、1106可以是透视显示器, 所述透视显示器允许用户查看实体环境而其他经渲染的内容 (例如, 虚拟角色、信息化注释、平视显示器/HUD) 呈现于实体环境的实时馈送的上方。

[0154] 在一个示例中, 框架1102包括左下看摄像机1108以从一般位于用户前面和左眼下方的区域捕捉图像 (例如, 左手姿势)。此外, 右下看摄像机1110可从一般位于用户前面和右眼下方的区域捕捉图像 (例如, 右手姿势)。所示出的框架1102还包括左前看摄像机1112和右前看摄像机1114以分别捕捉用户的左和右眼前面的图像。框架1102还可包括左侧看摄像机1116以从用户左侧的区域捕捉图像, 并包括右侧看摄像机1118以从用户右侧的区域捕捉图像。

[0155] 由可能有交叠视场的摄像机1108、1110、1112、1114、1116、1118捕捉的图像可用来检测由用户做出的姿势以及分析和/或在显示器1104、1106上再现外部环境。在一个示例中, 所检测的姿势由 (例如, 内部的和/或外部的) 图形处理架构用来渲染和/或控制3D游戏中用户的虚拟表示。事实上, 交叠的视场可启用对由其他个体 (例如, 在多玩家游戏中) 做出的姿势的捕捉, 其中其他个体的姿势还可用来渲染/控制沉浸式体验。交叠的视场还可使HMD系统1100能够自动地检测用户附近的阻挡或其他危险。此类方法在高级驾驶员辅助系统 (ADAS) 应用中尤其有利。

[0156] 在一个示例中, 提供有交叠视场的左下看摄像机1108和右下看摄像机1110提供了具有增加的分辨率的立体视图。增加的分辨率可进而使非常相似的用户移动能够互相区分开来 (例如, 以亚-毫米的准确性)。结果可以是HMD系统1100关于可靠性的提高的性能。事实上, 所示出的解决方案在各种各样的应用中是有用的, 所述应用是诸如例如给AR设置中的信息上色、在多用户环境中的用户之间交换虚拟工具/装置、渲染虚拟项 (例如, 武器、剑、人员) 等。其他对象、肢体和/或身体部分的姿势也可被检测并用于渲染/控制虚拟环境。例如, 无论是从穿戴者还是从共享环境中的另一个体, 都可以实时地跟踪脊髓造影信号、脑电波仪信号、眼睛跟踪、呼吸或喘气、手部运动等。由摄像机1108、1110、1112、1114、1116、1118捕捉的图像也可用作上下文输入。例如, 可能确定用户正在指示文字处理应用中待编辑的具体词或待按的具体键, 游戏中待部署的具体武器或行进方向等等。

[0157] 此外, 由摄像机1108、1110、1112、1114、1116、1118捕捉的图像可用来在装备操作、医疗训练和/或远程/远端操作指导应用中实施共享的通信或网络化的互动。任务特定的姿势库或神经网络机器学习可以启用工具标识和对任务的反馈。例如, 可启用转换成远程、真实动作的虚拟工具。在又一个示例中, HMD系统1100将虚拟场景内的虚拟钻头的操纵转换成搜索倒塌的建筑物而部署的机器人装置上的钻头的远程操作。而且, HMD系统1100可以在包括了例如使用户能够向与用户动作相关联的可标识姿势列表中添加新姿势的协议的程度是可编程的。

[0158] 另外, HMD 1100中的各种摄像机可以是可配置的, 以在光谱的可见波长之外检测

光谱频率。输入摄像机中的多谱成像能力通过消除非必要的图像特征(例如,背景噪音)来允许对用户和/或对象的位置跟踪。例如,在诸如手术的增强现实(AR)应用中,仪器和装备通过它们的红外反射性而可被跟踪,而无需额外的跟踪辅助装备。而且,HMD 1100可以在低可见度的情形中被采用,其中来自各种摄像机的“实时馈送”可以通过计算机分析来提高或增强并作为视觉或音频线索显示给用户。

[0159] HMD系统1100还可放弃执行与远程计算系统或者需要功率线(例如,独立操作模式)的任何类型的数据通信。就此而言,HMD系统1100可以是具有使HMD系统1100独立于外部功率系统而操作的功率单元的“无绳”装置。相应地,用户可以玩全特性游戏而无需栓系到另一个装置(例如,游戏控制台)或功率供应。在文字处理示例中,HMD系统1100在显示器1104和1106上呈现虚拟键盘和/或虚拟鼠标以提供虚拟桌面或文字处理场景。因此,由摄像机中的一个或多个捕捉的姿势识别数据可表示虚拟键盘上的用户键入活动或虚拟鼠标的移动。优点包括但不限于:便携性和虚拟桌面隔绝附近个体的隐私性的自在。底层图形处理架构可支持视频和音频信号的压缩和/或解压缩。而且,对用户的左眼和右眼提供分开的图像可促进3D场景的渲染、生成和/或感知。左眼显示器1104和右眼显示器1106的相对位置也可以是可调整的,以匹配不同用户之间眼间隔的变化。

[0160] 图11中所示出的摄像机的数量只是为了促进讨论。事实上,取决于环境,HMD系统1100可包括少于六个或多于六个摄像机。

[0161] HMD系统的功能部件

[0162] 图12更详细地示出HMD系统。在所示出的示例中,框架1102包括应用向HMD系统提供功率的功率单元1200(例如,电池功率、适配器)。所示出的框架1102还包括运动跟踪模块1220(例如,加速度计、陀螺仪),其中运动跟踪模块1220向处理器系统1204提供运动跟踪数据、取向数据和/或位置数据。处理器系统1204可包括耦合至I/O桥1206的网络适配器1224。I/O桥1206可启用网络适配器1224和诸如例如音频输入模块1210、音频输出模块1208、显示装置1207、输入摄像机1202等等的各种部件之间的通信。

[0163] 在所示出的示例中,音频输入模块1210包括右音频输入1218和左音频输入1216,该两个输入检测为了识别用户以及附近个体的语音命令而可被处理的声音。在所捕捉的音频信号中识别的语音命令可在模态切换和其他应用期间增强姿势识别。而且,所捕捉的音频信号可提供用来提高沉浸式体验的3D信息。

[0164] 音频输出模块1208可包括右音频输出1214和左音频输出1212。音频输出模块1208可将声音递送给用户和/或其他附近个体的耳朵。音频输出模块1208可以是耳塞、贴耳式扬声器、盖耳式扬声器、扩音器等或其任何组合的形式,音频输出模块1208可将立体和/或3D音频内容递送给用户(例如,空间定位)。所示出的框架1102还包括无线模块1222,该无线模块1222可促进HMD系统和各种其他系统(例如,计算机、可穿戴式装置、游戏控制台)之间的通信。在一个示例中,无线模块1222经由网络适配器1224与处理器系统1204通信。

[0165] 所示出的显示装置1207包括左眼显示器1104和右眼显示器1106,其中呈现在显示器1104、1106上的视觉内容可经由I/O桥1206从处理器系统1204获得。输入摄像机1202可包括已经讨论过的左侧看摄像机1116、右侧看摄像机1118、左下看摄像机1108、左前看摄像机1112、右前看摄像机1114和右下看摄像机1110。

[0166] 现在转向图13,示出了通用处理集群(GPC)1300。所示出的GPC 1300可合并入诸如

例如已经讨论过的处理器系统1204(图12)的处理系统中。GPC 1300可包括与调度器通信的流水线管理器1302。在一个示例中,流水线管理器1302从调度器接收任务并将任务分布给一个或多个流传送多处理器(SM) 1304。每一个SM 1304可配置成处理线程群组,其中线程群组可以被看作是对不同输入数据执行相同或相似操作的多个相关线程。因此,可将线程群组中的每一个线程分配给具体SM 1304。在另一个示例中,线程的数量可大于SM 1304中执行单元的数量。就此而言,线程群组中的线程可并行操作。流水线管理器1302还可将经处理的数据目的地指定给工作分布交叉开关1308,该工作分布交叉开关1308与存储器交叉开关通信。

[0167] 因此,在每一个SM 1304将经处理的任务传送给工作分布交叉开关1308时,经处理的任務可被提供给另一个GPC 1300供进一步处理。还可将SM 1304的输出发送到预光栅操作(preROP)单元1314,该预光栅操作单元1314进而将数据导引至一个或多个光栅操作单元,或者执行其他操作(例如,执行地址转换、组织图片颜色数据、混合颜色等等)。SM 1304可包括内部一级(L1)高速缓存(未示出),SM 1304可存储数据于其。SM 1304还可具有到一点五级(L1.5)高速缓存1306以及经由存储器管理单元(MMU) 1310到二级(L2)高速缓存(未示出)的访问。MMU 1310可将虚拟地址映射到物理地址。就此而言,MMU 1310可包括页表条目(PTE),页表条目用来将虚拟地址映射到图块、存储器页和/或高速缓存线索引的物理地址。所示出的GPU 1300还包括纹理单元1312。

[0168] 图形流水线架构

[0169] 现在转向图14,示出了图形流水线1400。在所示出的示例中,世界空间流水线1420包括图元分布器(PD) 1402。PD 1402可收集与高阶服务、图形图元、三角形等相关联的顶点数据,并将顶点数据传送给顶点属性获取单元(VAF) 1404。VAF 1404可从共享存储器检索与每一个进入顶点相关联的顶点属性并将顶点数据和相关联的顶点属性一起存储到共享存储器中。

[0170] 所示出的世界空间流水线1420还包括顶点、曲面细分、几何处理单元(VTG) 1406。VTG 1406可包括例如顶点处理单元、曲面细分初始化处理单元、任务分布器、任务生成单元、拓扑生成单元、几何处理单元、曲面细分处理单元等或其任何组合。在一个示例中,VTG 1406是配置成执行几何程序、曲面细分程序和顶点着色器程序的可编程执行单元。由VTG 1406执行的程序可处理从VAF 1404接收到的顶点数据和顶点属性。而且,由VTG 1406执行的程序可产生图形图元、颜色值、表面归一因子和图形图元的每个顶点处的透明度值以供在图形处理流水线1400内进一步处理。

[0171] VTG 1406的顶点处理单元可以是执行顶点着色器程序的可编程执行单元,按顶点着色器程序所规定的调亮和变换顶点数据。例如,可将顶点处理单元编程为将顶点数据从基于对象的坐标表示(例如,对象空间)变换到诸如世界空间或归一化装置坐标(NDC)空间的备选地基于坐标系。此外,顶点处理单元可读取由VAF 1404存储在共享存储器中的顶点数据和顶点属性,并且处理顶点数据和顶点属性。在一个示例中,顶点处理单元将经处理的顶点存储在共享存储器中。

[0172] 曲面细分初始化处理单元(例如,外壳着色器、曲面细分控制着色器)可执行曲面细分初始化着色器程序。在一个示例中,曲面细分初始化处理单元处理由顶点处理单元产生的顶点并生成有时称为“补片”的图形图元。曲面细分初始化处理单元还可生成各种补片

属性,其中补片数据和补片属性被存储于共享存储器。VTG 1406的任务生成单元可从共享存储器检索顶点和补片的数据和属性。在一个示例中,任务生成单元生成用于处理顶点和补片的任务,以用于由图形处理流水线1400中的稍后级进行处理。

[0173] 由任务生成单元产生的任务可由VTG 1406的任务分布器重分布。例如,由顶点着色器程序和曲面细分初始化程序的各种实例产生的任务可在一个图形处理流水线1400和另一个图形处理流水线之间显著变化。相应地,任务分布器可重分布这些任务使得每一个图形处理流水线1400在稍后流水线级期间有几乎相同的工作负荷。

[0174] 如已经注释的,VTG 1406还可包括拓扑生成单元。在一个示例中,拓扑生成单元检索由任务分布器分布的任务,对包括与补片相关联的顶点在内的顶点进行索引,并计算曲面细分顶点以及连接曲面细分顶点以形成图形图元的索引的坐标(UV)。经索引的顶点可由拓扑生成单元存储在共享存储器中。VTG 1406的曲面细分处理单元可配置成执行曲面细分着色器程序(例如,域着色器、曲面细分评估着色器)。曲面细分处理单元可从共享存储器读取输入数据并将输出数据写到共享存储器。输出数据可从共享存储器传递到几何处理单元(例如下一着色器级)作为输入数据。

[0175] VTG 1406的几何处理单元可执行几何着色器程序以变换图形图元(例如,三角形、线段、点等)。在一个示例中,将顶点分组以构建图形图元,其中几何处理单元将图形图元再分成一个或多个新图形图元。几何处理单元还可运算可用来光栅化新图形图元的参数,诸如例如平面方程系数(plain equation coefficient)。

[0176] 所示出的世界空间流水线1420还包括从VTG 1406接收规定新图形图元的参数和顶点的视口缩放、拣选和剪辑单元(VPC) 1408。在一个示例中,VPC 1408执行剪辑、翻边、透视校正和视口变换以标识在最终经渲染的图像中潜在可查看的图形图元。VPC 1408还可标识可能不可查看的图形图元。

[0177] 图形处理流水线1400还可包括耦合至世界空间流水线1420的分图块单元1410。分图块单元1410可以是图形图元分类引擎,其中图形图元在世界空间流水线1420中被处理并且随后被传送给分图块单元1410。就此而言,图形处理流水线1400还可包括屏幕空间流水线1422,其中屏幕空间可被划分成高速缓存图块。每一个高速缓存图块可因此与屏幕空间的一部分相关联。对于每一个图形图元,分图块单元1410可标识与图形图元相交(例如,“分图块”)的一组高速缓存图块。在将多个图形图元分图块以后,分图块单元1410可逐高速缓存图块地处理图形图元。在一个示例中,与具体高速缓存图块相关联的图形图元每次一个图块地被传送给屏幕空间流水线1422中的设置单元1412。与多个高速缓存图块相交的图形图元可在世界空间流水线1420中被处理一次,同时被传送到屏幕空间流水线1422多次。

[0178] 在一个示例中,设置单元1412经由分图块单元1410从VPC 1408接收顶点数据并且运算与图形图元相关联的参数。参数可以包括例如边方程、偏平面方程和深度平面方程。屏幕空间流水线1422还可包括耦合至设置单元1412的光栅化器1414。光栅化器可扫描转化新图形图元并将片段和覆盖数据传送到像素着色单元(PS) 1416。光栅化器1414还可执行Z拣选和其他基于Z的优化。

[0179] 可访问共享存储器的PS 1416可执行变换从光栅化器1414接收的片段的片段着色器程序。更具体地,片段着色器程序可以像素级粒度对片段着色(例如作为像素着色器程序工作)。在另一个示例中,片段着色器程序以样本级粒度对片段着色,其中每一个像素包括

多个样本,且每一个样本表示像素的一部分。而且,取决于环境(例如采样率),片段着色器程序可以任何其他粒度对片段着色。PS 1416可执行混合、着色、透视校正、纹理映射等以生成经着色的片段。

[0180] 所示出的屏幕空间流水线1422还包括光栅操作单元(ROP) 1418,其可执行诸如例如模板印刷、Z测试、混合等等。ROP 1418随后可将像素数据作为经处理的图形数据传送给一个或多个经渲染的目标(例如,图形存储器)。ROP 1418可配置成压缩写到存储器的Z或颜色数据并且解压缩从存储器读取的Z或颜色数据。ROP 1418的位置可取决于环境而变化。

[0181] 图形处理流水线1400可由一个或多个处理元件实现。例如,VTG 1406和/或PS 1416可被实现于一个或多个SM中,PD 1402、VAF 1404、VPC 1408、分图块单元1410、设置单元1412、光栅化器1414和/或ROP 1418可在连同对应的分区单元的具体GPC的处理元件中被实现。图形处理流水线1400还可在固定功能性硬件逻辑中被实现。事实上,图形处理流水线1400可在PPU中被实现。

[0182] 因此,所示出的世界空间流水线1420处理3D空间中的图形对象,其中已知每一个图形对象相对于其他图形对象以及相对于3D坐标系的位置。相反,屏幕空间流水线1422可处理已经从3D坐标系投射到表示显示装置的表面的2D平面表面上的图形对象。此外,可将世界空间流水线1420划分成阿尔法阶段流水线和贝塔阶段流水线,其中阿尔法阶段流水线包括从PD 1402直到任务生成单元的流水线级。贝塔阶段流水线可包括从拓扑生成单元直到VPC 1408的流水线级。在此类情况中,图形处理流水线1400可在阿尔法阶段流水线中执行第一组操作(例如,单个线程、线程群组、一致行动的多个线程群组)并在贝塔阶段流水线中执行第二组操作(例如,单个线程、线程群组、一致行动的多个线程群组)。

[0183] 如果多个图形处理流水线1400在使用中,则可将与一组图形对象相关联的顶点数据和顶点属性划分使得每一个图形处理流水线1400贯穿阿尔法阶段有相似的工作负荷。相应地,阿尔法阶段处理可以实质地扩展顶点数据和顶点属性的数量,使得由任务生成单元产生的顶点数据和顶点属性的数量显著大于由PD 1402和VAF 1404处理的顶点数据和顶点属性的数量。而且,即使当用相同数量的属性开始阿尔法阶段时,与不同的图形处理流水线1400相关联的任务生成单元也可产生具有不同质量级别的顶点数据和顶点属性。在此类情况中,任务分布器可重分布由阿尔法阶段流水线产生的属性,使得每一个图形处理流水线1400在贝塔阶段流水线的开始有大致相同的工作负荷。

[0184] 现在转向图15,示出了流传送多处理器(SM) 1500。所示出的SM 1500包括耦合至指令高速缓存1502的K个调度器单元1504,其中每一个调度器单元1504从流水线管理器(未示出)接收线程块阵列,并管理每一个活动线程块阵列中的一个或多个线程块的指令调度。调度器单元1504可调度线程以用于在并行线程群组中执行,其中每一个群组可称为“线程束”。因此,每一个线程束可能包括例如六十四个线程。此外,调度器单元1504可管理多个不同的线程块,将线程块分配给线程束以用于执行。调度器单元可随后在每一个时钟周期期间在各种功能单元上调度来自多个不同线程束的指令。每一个调度器单元1504可包括一个或多个指令分派单元1522,其中每一个分派单元1522向功能单元中的一个或多个传送指令。分派单元1522的数量可取决于环境而变化。在所示出的示例中,调度器单元1504包括使来自同一线程束的两个不同指令在每个时钟周期期间能被分派的两个分派单元1522。

[0185] SM 1500还可以包括寄存器堆1506。寄存器堆1506可包括一组寄存器,该组寄存器

在功能单元之间进行划分使得每一个功能单元被分配到寄存器堆1506的专用部分。寄存器堆1506还可在正由SM 1500执行的不同线程束之间进行划分。在一个示例中,寄存器堆1506为连接到功能单元的数据路径的操作数提供临时存储。所示出的SM 1500还包括L个处理核1508,其中L可以是相对较大的数(例如192)。每一个核1508都可以是包括浮点算术逻辑单元(例如,IEEE 754-2008)和整数算术逻辑单元的流水线的单精度处理单元。

[0186] 所示出的SM 1500还包括M个双精度单元(DPU) 1510、N个特殊功能单元(SFU) 1512和P个加载/存储单元(LSU) 1514。每一个DPU 1510都可实现双精度浮点算术且每一个SFU 1512都可执行诸如例如矩形复制像素混合等的特殊功能。此外,每一个LSU 1514都可在共享存储器1518与寄存器堆1506之间实施加载和存储操作。在一个示例中,加载和存储操作是通过J个纹理单元/L1高速缓存1520和互连网络1516来实施的。在一个示例中,J个纹理单元/L1高速缓存1520还耦合至交叉开关(未示出)。因此,互连网络1516可将功能单元的每一个连接到寄存器堆1506和共享存储器1518。在一个示例中,互连网络1516担当将功能单元中的任何一个连接到寄存器堆1506中的任何寄存器的交叉开关。

[0187] SM 1500可在图形处理器(例如,图形处理单元/GPU)内实现,其中纹理单元/L1高速缓存1520可访问来自存储器的纹理映射并对纹理映射采样以产生经采样的纹理值以用于着色器程序中。由纹理单元/L1高速缓存1520执行的纹理操作包括但不限于基于mip图(mipmap)的抗混叠。

[0188] 附加系统概述示例

[0189] 图16是根据实施例的处理系统1600的框图。在各种实施例中,系统1600包括一个或多个处理器1602和一个或多个图形处理器1608,并且可以是具有大数量的处理器1602或处理器核1607的服务器系统、单处理器桌上型系统或多处理器工作站系统。在一个实施例中,系统1600是合并芯片上系统(SoC)集成电路内以供在移动、手持式或嵌入式装置中使用的处理平台。

[0190] 系统1600的实施例可以包括以下各项或可以合并以下各项内:基于服务器的游戏平台、游戏控制台(包括游戏和媒体控制台、移动游戏控制台、手持式游戏控制台或在线游戏控制台)。在一些实施例中,系统1600是移动电话、智能电话、平板计算装置或移动互联网装置。数据处理系统1600还可以包括以下各项、与以下各项耦合或被集成在以下各项内:可穿戴式装置,诸如智能手表可穿戴式装置、智能眼部穿戴式(eyewear)装置、增强现实装置或虚拟现实装置。在一些实施例中,数据处理系统1600是电视或机顶盒装置,其具有一个或多个处理器1602和由一个或多个图形处理器1608产生的图形界面。

[0191] 在一些实施例中,所述一个或多个处理器1602各自包括用于处理指令的一个或多个处理器核1607,所述指令在被执行时执行系统和用户软件的操作。在一些实施例中,所述一个或多个处理器核1607中的每个配置成处理特定指令集1609。在一些实施例中,指令集1609可以促进复杂指令集计算(CISC)、精简指令集计算(RISC)、或经由超长指令字(VLIW)的计算。多个处理器核1607可以各自处理不同的指令集1609,所述指令集可以包括用于促进对其他指令集进行仿真的指令。处理器核1607还可包括其他处理装置,诸如数字信号处理器(DSP)。

[0192] 在一些实施例中,处理器1602包括高速缓存存储器1604。取决于架构,处理器1602可以具有单个内部高速缓存或多个级别的内部高速缓存。在一些实施例中,在处理器1602

的各种部件中共享高速缓存存储器。在一些实施例中,处理器1602还使用外部高速缓存(例如,3级(L3)高速缓存或未级高速缓存(LLC))(未示出),其可使用已知的高速缓存一致性技术在处理器核1607中被共享。寄存器堆1606被另外包括在处理器1602中,所述寄存器堆可包括用于存储不同类型的数据的不同类型的寄存器(例如,整数寄存器、浮点寄存器、状态寄存器和指令指针寄存器)。一些寄存器可以是通用寄存器,而其他寄存器可以特定于处理器1602的设计。

[0193] 在一些实施例中,处理器1602耦合至用于在处理器1602与系统1600中的其他部件之间传送通信信号(诸如,地址、数据或控制信号)的处理器总线1610部件。在一个实施例中,系统1600使用示例性‘中枢’系统架构,包括存储器控制器中枢1616和输入输出(I/O)控制器中枢1630。存储器控制器中枢1616促进存储器装置与系统1600的其他部件之间的通信,而I/O控制器中枢(ICH) 1630提供经由本地I/O总线到I/O装置的连接。在一个实施例中,存储器控制器中枢1616的逻辑被集成在处理器内。

[0194] 存储器装置1620可以是动态随机存取存储器(DRAM)装置、静态随机存取存储器(SRAM)装置、闪速存储器装置、相变存储器装置或具有合适的性能以充当进程存储器的某一其他存储器装置。在一个实施例中,存储器装置1620可以作为系统1600的系统存储器来操作,以存储数据1622和指令1621,以供当所述一个或多个处理器1602执行应用或进程时使用。存储器控制器中枢1616也与可选的外部图形处理器1612耦合,所述外部图形处理器1612可与处理器1602中的一个或多个图形处理器1608通信,以执行图形和媒体操作。

[0195] 在一些实施例中,ICH 1630使外围装置能够经由高速I/O总线连接到存储器装置1620和处理器1602。I/O外围装置包含但不限于:音频控制器1646、固件接口1628、无线收发器1626(例如,Wi-Fi、蓝牙)、数据存储装置1624(例如,硬盘驱动、闪速存储器等)和用于将传统(例如,个人系统2(PS/2))装置耦合至系统的传统I/O控制器1640。一个或多个通用串行总线(USB)控制器1642连接输入装置(诸如,键盘和鼠标1644组合)。网络控制器1634还可以耦合至ICH 1630。在一些实施例中,高性能网络控制器(未示出)耦合至处理器总线1610。将认识到的是,所示出的系统1600是示例性的而非限制性的,因为还可以使用以不同方式配置的其他类型的数据处理系统。例如,I/O控制器中枢1630可以集成在所述一个或多个处理器1602内,或者存储器控制器中枢1616和I/O控制器中枢1630可以集成到谨慎(discreet)外部图形处理器(诸如外部图形处理器1612)中。

[0196] 图17是处理器1700的实施例的框图,所述处理器1700具有一个或多个处理器核1702A-1702N、集成式存储器控制器1714和集成式图形处理器1708。图17中具有与本文任何其他图的元件相同的参考数字(或名称)的那些元件可以以与本文别处描述的方式类似的任何方式来操作或运作,但并不限于此。处理器1700可包括多达且包括由虚线框表示的附加核1702N的附加核。处理器核1702A-1702N中的每个都包含一个或多个内部高速缓存单元1704A-1704N。在一些实施例中,每个处理器核还能够访问一个或多个共享高速缓存单元1706。

[0197] 内部高速缓存单元1704A-1704N和共享高速缓存单元1706表示处理器1700内的高速缓存存储器分层。高速缓存存储器分层可包括每个处理器核内的指令和数据高速缓存的至少一个级别以及共享中间级高速缓存的一个或多个级别(诸如,2级(L2)、3级(L3)、4级(L4)或高速缓存的其他级别),其中,在外部存储器前面的最高级别高速缓存被归类为LLC。

在一些实施例中,高速缓存一致性逻辑保持各种高速缓存单元1706和1704A-1704N之间的一致性。

[0198] 在一些实施例中,处理器1700还可包括一组一个或多个总线控制器单元1716以及系统代理核1710。所述一个或多个总线控制器单元1716管理一组外围总线,诸如一个或多个外围部件互连总线(例如,PCI、PCI快速)。系统代理核1710提供对各种处理器部件的管理功能性。在一些实施例中,系统代理核1710包括一个或多个集成式存储器控制器1714以管理对各种外部存储器装置(未示出)的访问。

[0199] 在一些实施例中,处理器核1702A-1702N中的一个或多个包括对同时多线程的支持。在此类实施例中,系统代理核1710包括用于在多线程处理期间协调和操作核1702A-1702N的部件。系统代理核1710可另外包括功率控制单元(PCU),所述PCU包括用于调节处理器核1702A-1702N和图形处理器1708的功率状态的逻辑和部件。

[0200] 在一些实施例中,处理器1700另外包括用于执行图形处理操作的图形处理器1708。在一些实施例中,图形处理器1708与一组共享高速缓存单元1706和系统代理核1710(包括所述一个或多个集成式存储器控制器1714)耦合。在一些实施例中,显示器控制器1711与图形处理器1708耦合以将图形处理器输出驱动到一个或多个耦合的显示器。在一些实施例中,显示器控制器1711可以是经由至少一个互连与图形处理器耦合的单独模块,或者可以集成在图形处理器1708或系统代理核1710内。

[0201] 在一些实施例中,使用基于环的互连单元1712来耦合处理器1700的内部部件。然而,可以使用备选互连单元,诸如点到点互连、切换式互连、或其他技术,包括本领域众所周知的技术。在一些实施例中,图形处理器1708经由I/O链路1713与环形互连1712耦合。

[0202] 示例性I/O链路1713表示多种I/O互连中的至少一种,包括促进各种处理器部件与高性能嵌入式存储器模块1718(诸如eDRAM模块)之间的通信的封装上(on package) I/O互连。在一些实施例中,处理器核1702-1702N中的每个和图形处理器1708将嵌入式存储器模块1718用作共享末级高速缓存。

[0203] 在一些实施例中,处理器核1702A-1702N是执行相同的指令集架构的同质核。在另一个实施例中,处理器核1702A-1702N在指令集架构(ISA)方面是异质的,其中,处理器核1702A-N中的一个或多个执行第一指令集,而另一些核中的至少一个执行第一指令集的子集或不同指令集。在一个实施例中,处理器核1702A-1702N在微架构方面是异质的,其中,具有相对更高功率耗用的一个或多个核与具有更低功率耗用的一个或多个功率核耦合。另外,处理器1700可以被实现在一个或多个芯片上或者被实现为具有除其他部件之外的所示出的部件的SoC集成电路。

[0204] 图18是图形处理器1800的框图,所述图形处理器1800可以是分立式图形处理单元,或可以是与多个处理核集成的图形处理器。在一些实施例中,图形处理器经由到图形处理器上的寄存器的存储器映射的I/O接口并且用被放置到处理器存储器中的命令进行通信。在一些实施例中,图形处理器1800包括用于访问存储器的存储器接口1814。存储器接口1814可以是到本地存储器、一个或多个内部高速缓存、一个或多个共享外部高速缓存、和/或到系统存储器的接口。

[0205] 在一些实施例中,图形处理器1800还包括用于将显示器输出数据驱动到显示装置1820的显示器控制器1802。显示器控制器1802包括用于显示器的一个或多个重叠平面的硬

件以及多层视频或用户接口元件的组成。在一些实施例中,图形处理器1800包括用于向、从一个或多个媒体编码格式或在所述格式之间对媒体进行编码、解码、或者转码的视频编解码器引擎1806,所述格式包括但不限于:移动图像专家组(MPEG)格式(诸如MPEG-2)、高级视频译码(AVC)格式(诸如H.264/MPEG-4 AVC)、以及运动图像&电视工程师协会(SMPTE) 421M/VC-1、和联合图像专家组(JPEG)格式(诸如JPEG、以及运动JPEG(MJPEG)格式)。

[0206] 在一些实施例中,图形处理器1800包括用于执行二维(2D)光栅化器操作的块图转移(BLIT)引擎1804,所述2D光栅化器操作包括(例如)位边界块转移。然而,在一个实施例中,使用图形处理引擎(GPE) 1810的一个或多个部件来执行2D图形操作。在一些实施例中,图形处理引擎1810是用于执行图形操作的计算引擎,所述图形操作包括三维(3D)图形操作和媒体操作。

[0207] 在一些实施例中,GPE 1810包括用于执行3D操作的3D流水线1812,诸如使用作用于3D图元图元形状(例如矩形、三角形等)的处理功能来渲染三维图像和场景。3D流水线1812包含可编程和固定功能元件,所述元件执行元件内的各种任务和/或对3D/媒体子系统1815生成执行线程。虽然3D流水线1812可以用于执行媒体操作,但是GPE 1810的实施例还包括媒体流水线1816,所述媒体流水线1816特定地用于执行媒体操作,诸如视频后处理(video post-processing)和图像提高。

[0208] 在一些实施例中,媒体流水线1816包括用于代替或代表视频编解码器引擎1806执行一个或多个专门的媒体操作(诸如视频解码加速、视频去交织和视频编码加速)的固定功能或可编程逻辑单元。在一些实施例中,媒体流水线1816另外包括线程生成单元以便生成用于在3D/媒体子系统1815上执行的线程。所生成的线程在3D/媒体子系统1815中所包括的一个或多个图形执行单元上执行对媒体操作的计算。

[0209] 在一些实施例中,3D/媒体子系统1815包括用于执行由3D流水线1812和媒体流水线1816生成的线程的逻辑。在一个实施例中,流水线向3D/媒体子系统1815发送线程执行请求,所述3D/媒体子系统1815包括用于仲裁并分派对可用的线程执行资源的各种请求的线程分派逻辑。执行资源包括用于处理3D和媒体线程的图形执行单元的阵列。在一些实施例中,3D/媒体子系统1815包括用于线程指令和数据的一个或多个内部高速缓存。在一些实施例中,所述子系统还包括共享存储器(包括寄存器和可寻址存储器),以在线程之间共享数据并存储输出数据。

[0210] 3D/媒体处理

[0211] 图19是依照一些实施例的图形处理器的图形处理引擎1910的框图。在一个实施例中,GPE 1910是图18中所示的GPE 1810的一个版本。图19中具有与本文任何其他图的元件相同的参考数字(或名称)的元件可以以与本文别处描述的方式类似的任何方式来操作或运作,但并不限于此。

[0212] 在一些实施例中,GPE 1910与命令流传送器1903耦合,所述命令流传送器将命令流提供给GPE 3D和媒体流水线1912、1916。在一些实施例中,命令流传送器1903耦合至存储器,所述存储器可以是系统存储器、或内部高速缓存存储器和共享高速缓存存储器中的一个或多个。在一些实施例中,命令流传送器1903从存储器接收命令,并且将命令发送给3D流水线1912和/或媒体流水线1916。所述命令是从存储3D和媒体流水线1912、1916的命令的环形缓冲器获取的指示。在一个实施例中,所述环形缓冲器可另外包括存储多批多个命令的

批命令缓冲器。3D和媒体流水线1912、1916通过经由相应流水线内的逻辑来执行操作或者通过将一个或多个执行线程分派至执行单元阵列1914来处理所述命令。在一些实施例中,执行单元阵列1914是可缩放的,使得所述阵列基于GPE 1910的目标功率和性能级别而包括可变数量的执行单元。

[0213] 在一些实施例中,采样引擎1930与存储器(例如,高速缓存存储器或系统存储器)以及执行单元阵列1914耦合。在一些实施例中,采样引擎1930为允许执行阵列1914从存储器读取图形和媒体数据的执行单元阵列1914提供了存储器存取机制。在一些实施例中,采样引擎1930包括用于执行针对媒体的专门图像采样操作的逻辑。

[0214] 在一些实施例中,采样引擎1930中的专门的媒体采样逻辑包括去噪/去交织模块1932、运动估计模块1934以及图像缩放和过滤模块1936。在一些实施例中,去噪/去交织模块1932包括用于对经解码的视频数据执行去噪或去交织算法中的一个或多个的逻辑。去交织逻辑将经交织的视频内容的交替场(alternating field)组合到视频的单个帧中。去噪逻辑从视频和图像数据减少或去除数据噪声。在一些实施例中,所述去噪逻辑和去交织逻辑是运动自适应的并且基于在视频数据中检测到的运动量使用空间或时间过滤。在一些实施例中,去噪/去交织模块1932包括专用的运动检测逻辑(例如,在运动估计引擎1934内)。

[0215] 在一些实施例中,运动估计引擎1934通过对视频数据执行视频加速功能(诸如,运动向量估计和预测)来提供对视频操作的硬件加速。运动估计引擎确定描述连续视频帧之间的图像数据的变换的运动向量。在一些实施例中,图形处理器媒体编解码器使用视频运动估计引擎1934在宏块级别对视频执行操作,否则对于其用通用处理器来执行可以是过于计算密集的。在一些实施例中,运动估计引擎1934一般可用于图形处理器部件以便辅助视频解码和处理功能,所述视频解码和处理功能对于视频数据内的运动的方向或幅度是敏感或自适应的。

[0216] 在一些实施例中,图像缩放和过滤模块1936执行图像处理操作,以提高所产生的图像和视频的视觉质量。在一些实施例中,缩放和过滤模块1936在向执行单元阵列1914提供数据之前在采样操作期间处理图像和视频数据。

[0217] 在一些实施例中,GPE 1910包括数据端口1944,所述数据端口1944为了图形子系统访问存储器提供附加机制。在一些实施例中,数据端口1944针对操作促进存储器访问,所述操作包括渲染目标写入、恒定缓冲器读取、暂存存储器空间读取/写入、和媒体表面访问。在一些实施例中,数据端口1944包括用于高速缓存对存储器的访问的高速缓存存储器空间。高速缓存存储器可以是单个数据高速缓存,或被分离成经由数据端口来访问存储器的多个子系统的多个高速缓存(例如,渲染缓冲器高速缓存、恒定缓冲器高速缓存等)。在一些实施例中,执行在执行单元阵列1914中的执行单元上的线程通过经由数据分布互连来交换消息而与数据端口通信,所述数据分布互连耦合GPE 1910的每个子系统。

[0218] 执行单元

[0219] 图20是图形处理器2000的另一实施例的框图。图20中具有与本文任何其他图的元件相同的参考数字(或名称)的元件可以以与本文别处描述的方式类似的任何方式来操作或运作,但并不限于此。

[0220] 在一些实施例中,图形处理器2000包括环形互连2002、流水线前端2004、媒体引擎2037和图形核2080A-2080N。在一些实施例中,环形互连2002将图形处理器耦合至其他处理

单元,包括其他图形处理器或者一个或多个通用处理器核。在一些实施例中,图形处理器是集成在多核处理系统内的许多处理器之一。

[0221] 在一些实施例中,图形处理器2000经由环形互连2002接收多批命令。由流水线前端2004中的命令流传送器2003解译进入的命令。在一些实施例中,图形处理器2000包括用于经由图形核2080A-2080N来执行3D几何处理和媒体处理的可缩放执行逻辑。针对3D几何处理命令,命令流传送器2003将命令供应给几何流水线2036。针对至少一些媒体处理命令,命令流传送器2003将命令供应给视频前端2034,所述视频前端2034与媒体引擎2037耦合。在一些实施例中,媒体引擎2037包括用于视频和图像后处理的视频质量引擎(VQE) 2030以及用于提供硬件加速的媒体数据编码和解码的多格式编码/解码(MFX) 2033引擎。在一些实施例中,几何流水线2036和媒体引擎2037各自生成对于由至少一个图形核2080A所提供的线程执行资源的执行线程。

[0222] 在一些实施例中,图形处理器2000包括以模块化核2080A-2080N(有时称为核切片)为特征的可缩放线程执行资源,每个模块化核具有多个子核2050A-2050N、2060A-2060N(有时称为核子切片)。在一些实施例中,图形处理器2000可以具有任意数量的图形核2080A至2080N。在一些实施例中,图形处理器2000包括图形核2080A,所述图形核2080A至少具有第一子核2050A和第二核子核2060A。在其他实施例中,图形处理器是具有单个子核(例如,2050A)的低功率处理器。在一些实施例中,图形处理器2000包括多个图形核2080A-2080N,每个图形核包括一组第一子核2050A-2050N和一组第二子核2060A-2060N。该组第一子核2050A-2050N中的每个子核至少包括第一组执行单元2052A-2052N和媒体/纹理采样器2054A-2054N。该组第二子核2060A-2060N中的每个子核至少包括第二组执行单元2062A-2062N和采样器2064A-2064N。在一些实施例中,每个子核2050A-2050N、2060A-2060N共享一组共享资源2070A-2070N。在一些实施例中,所述共享资源包括共享高速缓存存储器和像素操作逻辑。其他共享资源也可被包括在图形处理器的各种实施例中。

[0223] 图21示出了线程执行逻辑2100,包括在GPE的一些实施例中所采用的处理元件的阵列。图21中具有与本文任何其他图的元件相同的参考数字(或名称)的那些元件可以以与本文别处描述的方式类似的任何方式来操作或运作,但并不限于此。

[0224] 在一些实施例中,线程执行逻辑2100包括像素着色器2102、线程分派器2104、指令高速缓存2106、可缩放执行单元阵列(包括多个执行单元2108A-2108N)、采样器2110、数据高速缓存2112和数据端口2114。在一个实施例中,所包括的部件经由互连结构而互连,所述互连结构链接到所述部件中的每个。在一些实施例中,通过指令高速缓存2106、数据端口2114、采样器2110和执行单元阵列2108A-2108N中的一个或多个,线程执行逻辑2100包括到存储器(诸如,系统存储器或高速缓存存储器)的一个或多个连接。在一些实施例中,每个执行单元(例如,2108A)是单独向量处理器,其能够执行多个同时的线程并且针对每个线程来并行处理多个数据元素。在一些实施例中,执行单元阵列2108A-2108N包括任何数量的单独执行单元。

[0225] 在一些实施例中,执行单元阵列2108A-2108N主要用于执行“着色器”程序。在一些实施例中,阵列2108A-2108N中的执行单元执行包括对许多标准3D图形着色器指令的原生支持的指令集,使得用最小转换来执行来自图形库(例如,Direct 3D和OpenGL)的着色器程序。执行单元支持顶点和几何处理(例如,顶点程序、几何程序、顶点着色器)、像素处理(例

如,像素着色器、片段着色器)和通用处理(例如,计算和媒体着色器)。

[0226] 执行单元阵列2108A-2108N中的每个执行单元对数据元素的阵列进行操作。数据元素的数量是“执行大小”或用于指令的通道的数量。执行通道是用于指令内的流控制、数据元素访问、和掩蔽的执行的逻辑单元。通道的数量可以独立于具体图形处理器的物理算术逻辑单元(ALU)或浮点单元(FPU)的数量。在一些实施例中,执行单元2108A-2108N支持整数和浮点数据类型。

[0227] 执行单元指令集包括单指令多数据(SIMD)指令。可以将各种数据元素作为紧缩数据类型存储在寄存器中,并且执行单元将基于元素的数据大小来处理各种元素。例如,当在256位宽的向量上进行操作时,所述256位的向量存储在寄存器中,并且所述执行单元作为四个单独64位紧缩数据元素(四倍字长(QW)大小的数据元素)、八个单独32位紧缩数据元素(双倍字长(DW)大小的数据元素)、十六个单独16位紧缩数据元素(字长(W)大小的数据元素)、或三十二个单独8位数据元素(字节(B)大小的数据元素)在所述向量上进行操作。然而,不同的向量宽度和寄存器大小是可能的。

[0228] 一个或多个内部指令高速缓存(例如,2106)被包括在线程执行逻辑2100中,以高速缓存执行单元的线程指令。在一些实施例中,一个或多个数据高速缓存(例如,2112)被包括以高速缓存在线程执行期间的线程数据。在一些实施例中,采样器2110被包括以为了3D操作提供纹理采样并且为了媒体操作提供媒体采样。在一些实施例中,采样器2110包括专门的纹理或媒体采样功能性,以在向执行单元提供所采样的数据之前在采样过程期间处理纹理或媒体数据。

[0229] 在执行期间,图形和媒体流水线经由线程生成和分派逻辑将线程发起请求发送给线程执行逻辑2100。在一些实施例中,线程执行逻辑2100包括本地线程分派器2104,所述本地线程分派器2104仲裁来自图形和媒体流水线的线程发起请求并在一个或多个执行单元2108A-2108N上例示所请求的线程。例如,几何流水线(例如,图20的2036)将顶点处理、曲面细分或几何处理线程分派给线程执行逻辑2100(图21)。在一些实施例中,线程分派器2104还可以处理来自执行着色器程序的执行时间线程生成请求。

[0230] 一旦几何对象的群组已被处理并被光栅化到像素数据中,调用像素着色器2102以进一步计算输出信息并促使结果被写到输出表面(例如,颜色缓冲器、深度缓冲器、模板印刷缓冲器等)。在一些实施例中,像素着色器2102运算要跨栅格化的对象被内插的各种顶点属性的值。在一些实施例中,像素着色器2102然后执行应用编程接口(API)供应的像素着色器程序。为了执行所述像素着色器程序,像素着色器2102经由线程分派器2104将线程分派给执行单元(例如,2108A)。在一些实施例中,像素着色器2102使用采样器2110中的纹理采样逻辑来访问存储器中所存储的纹理映射中的纹理数据。对纹理数据和输入几何数据进行算术操作运算针对每个几何片段的像素颜色数据,或从进一步处理中丢弃一个或多个像素。

[0231] 在一些实施例中,数据端口2114为了线程执行逻辑2100将处理的数据输出到存储器以便在图形处理器输出流水线上处理而提供存储器访问机制。在一些实施例中,数据端口2114包括或耦合至一个或多个高速缓存存储器(例如,数据高速缓存2112)以经由数据端口高速缓存用于存储器访问的数据。

[0232] 图22是根据一些实施例图示了图形处理器指令格式2200的框图。在一个或多个实

施例中,图形处理器执行单元支持具有采用多个格式的指令的指令集。实线框示出通常包括在执行单元指令中的分量,而虚线包括可选的或仅包括在指令的子集中的分量。在一些实施例中,所描述和示出的指令格式2200是宏指令,因为它们是供应至执行单元的指令,这与从指令解码得出的微操作相反(一旦所述指令被处理)。

[0233] 在一些实施例中,图形处理器执行单元原生支持采用128位格式2210的指令。64位紧凑指令格式2230对于基于所选的指令、指令选项和操作数的数量的一些指令是可用的。原生128位格式2210提供对所有指令选项的访问,而一些选项和操作被限定于64位格式2230中。在64位格式2230中可用的原生指令随实施例而变化。在一些实施例中,使用索引字段2213中的一组索引值将指令部分地紧凑。执行单元硬件引用基于所述索引值的一组紧凑表,并且使用紧凑表输出来以128位格式2210重构原生指令。

[0234] 对每个格式,指令操作码2212定义执行单元要执行的操作。执行单元跨每个操作数的多个数据元素并行执行每个指令。例如,响应于加法指令,执行单元跨表示纹理元素或图片元素的每个颜色通道来执行同时加法运算。默认情况下,执行单元跨操作数的所有数据通道执行每个指令。在一些实施例中,指令控制字段2214能够实现对某些执行选项(诸如通道选择(例如,预测)以及数据通道排序(例如,搅和)的控制。针对128位指令2210,执行大小字段2216限制将被并行执行的数据通道的数量。在一些实施例中,执行大小字段2216对于在64位紧凑指令格式2230中使用是不可用的。

[0235] 一些执行单元指令具有多达三个操作数,包括两个源操作数src0 2220、src1 2222和一个目的地2218。在一些实施例中,执行单元支持双目的地指令,其中,所述目的地之一是隐式的。数据操纵指令可以具有第三源操作数(例如, SRC2 2224),其中,指令操作码2212确定源操作数的数量。指令的最后一个源操作数可以通过所述指令传递的立即(例如,硬编码的)值。

[0236] 在一些实施例中,128位指令格式2210包括访问/地址模式信息2226,所述访问/地址模式信息2226指定(例如)使用直接寄存器寻址模式还是间接寄存器寻址模式。当使用直接寄存器寻址模式时,直接通过指令2210中的位来提供一个或多个操作数的寄存器地址。

[0237] 在一些实施例中,128位指令格式2210包括访问/地址模式字段2226,所述访问/地址模式字段2226指定针对所述指令的地址模式和/或访问模式。在一个实施例中,访问模式用于定义针对所述指令的数据访问对齐。一些实施例支持包括16字节对齐的访问模式和1字节对齐的访问模式的访问模式,其中,访问模式的字节对齐确定指令操作数的访问对齐。例如,当处于第一模式中时,指令2210可针对源和目的地操作数使用字节对齐的寻址,并且当处于第二模式中时,指令2210可针对所有的源和目的地操作数使用16字节对齐的寻址。

[0238] 在一个实施例中,访问/地址模式字段2226的地址模式部分确定指令将使用直接还是间接寻址。当使用直接寄存器寻址模式时,指令2210中的位直接提供一个或多个操作数的寄存器地址。当使用间接寄存器寻址模式时,可基于所述指令中的地址立即字段和地址寄存器值来计算一个或多个操作数的寄存器地址。

[0239] 在一些实施例中,基于操作码2212位字段对指令进行分组,以简化操作码解码2240。针对8位操作码,位4、5和6允许执行单元确定操作码的类型。所示出的精确操作码分组仅仅是示例。在一些实施例中,移动和逻辑操作码群组2242包括数据移动和逻辑指令(例如,移动(mov)、比较(cmp))。在一些实施例中,移动和逻辑群组2242共享五个最高有效位

(MSB), 其中, 移动(mov)指令采用0000xxxxb的形式, 以及逻辑指令采用0001xxxxb的形式。流控制指令群组2244(例如, 调用、跳(jmp))包括采用0010xxxxb形式(例如, 0x20)的指令。混杂指令群组2246包括指令的混合, 其包括采用0011xxxxb形式(例如, 0x30)的同步指令(例如, 等待、发送)。并行数学指令群组2248包括采用0100xxxxb形式(例如, 0x40)的分量式算术指令(例如, 加、乘(mul))。并行数学群组2248跨数据通道并行执行算术操作。向量数学群组2250包括采用0101xxxxb形式(例如, 0x50)的算术指令(例如, dp4)。向量数学群组执行诸如对向量操作数的点积运算的算术。

[0240] 图形流水线

[0241] 图23是图形处理器2300的另一实施例的框图。图23中具有与本文任何其他图的元件相同的参考数字(或名称)的元件可以以与本文别处描述的方式类似的任何方式来操作或运作, 但并不限于此。

[0242] 在一些实施例中, 图形处理器2300包括图形流水线2320、媒体流水线2330、显示器引擎2340、线程执行逻辑2350和渲染输出流水线2370。在一些实施例中, 图形处理器2300是包括一个或多个通用处理核的多核处理系统内的图形处理器。图形处理器由到一个或多个控制寄存器(未示出)的寄存器写入控制或者经由借助环形互连2302发布至图形处理器2300的命令控制。在一些实施例中, 环形互连2302将图形处理器2300耦合至其他处理部件, 诸如其他图形处理器或通用处理器。来自环形互连2302的命令由命令流传送器2303解译, 所述命令流传送器2303将指令供应给图形流水线2320或媒体流水线2330的单独部件。

[0243] 在一些实施例中, 命令流传送器2303指导顶点获取器2305的操作, 所述顶点获取器2305从存储器读取顶点数据并执行由命令流传送器2303提供的顶点处理命令。在一些实施例中, 顶点获取器2305将顶点数据提供给顶点着色器2307, 所述顶点着色器2307对每个顶点执行坐标空间变换和调亮操作。在一些实施例中, 顶点获取器2305和顶点着色器2307通过经由线程分派器2331将执行线程分派给执行单元2352A、2352B来执行顶点处理指令。

[0244] 在一些实施例中, 执行单元2352A、2352B是具有用于执行图形和媒体操作的指令集的向量处理器的阵列。在一些实施例中, 执行单元2352A、2352B具有特定用于每个阵列或在阵列之间共享的附连的L1高速缓存2351。所述高速缓存可以被配置为数据高速缓存、指令高速缓存或单个高速缓存, 其被分区以将数据和指令包含在不同的分区中。

[0245] 在一些实施例中, 图形流水线2320包括用于执行3D对象的硬件加速曲面细分的曲面细分部件。在一些实施例中, 可编程的外壳着色器2311配置曲面细分操作。可编程的域着色器2317提供曲面细分输出的后端评估。曲面细分器2313在外壳着色器2311的方向进行操作并且包含专用逻辑, 所述专用逻辑用于基于粗糙几何模型来生成一组详细的几何对象, 所述粗糙几何模型作为输入被提供至图形流水线2320。在一些实施例中, 如果未使用曲面细分, 则可以对曲面细分部件2311、2313、2317进行旁路。

[0246] 在一些实施例中, 完整的几何对象可以由几何着色器2319经由被分派给执行单元2352A、2352B的一个或多个线程来处理, 或可以直接继续进行至剪辑器2329。在一些实施例中, 几何着色器在整个几何对象(而非如图形流水线的先前阶段中的顶点的补片或者顶点)上进行操作。如果曲面细分被禁用, 则几何着色器2319从顶点着色器2307接收输入。在一些实施例中, 如果曲面细分单元被禁用, 则几何着色器2319可由几何着色器程序编程以执行几何曲面细分。

[0247] 在光栅化之前,剪辑器2329处理顶点数据。剪辑器2329可以是具有剪辑和几何着色器功能的可编程剪辑器或者固定功能剪辑器。在一些实施例中,渲染输出流水线2370中的光栅化器2373(例如,深度测试部件)分派像素着色器以将几何对象转化成它们的按像素表示。在一些实施例中,像素着色器逻辑被包括在线程执行逻辑2350中。在一些实施例中,应用可以对光栅化器2373进行旁路,并且经由流出单元(stream out unit)2323来访问未光栅化的顶点数据。

[0248] 图形处理器2300具有互连总线、互连结构或某种其他互连机制,其允许在处理器的主要部件中传递数据和消息。在一些实施例中,执行单元2352A、2352B和(一个或多个)关联的高速缓存2351、纹理和媒体采样器2354以及纹理/采样器高速缓存2358经由数据端口2356互连,以执行存储器访问并与处理器的渲染输出流水线部件进行通信。在一些实施例中,采样器2354、高速缓存2351、2358以及执行单元2352A、2352B各自具有单独的存储器访问路径。

[0249] 在一些实施例中,渲染输出流水线2370包含光栅化器2373,所述光栅化器2373将基于顶点的对象转化成关联的基于像素的表示。在一些实施例中,光栅化器逻辑包括用于执行固定功能三角形和线光栅化的窗口器(windower)/掩蔽器单元。相关联的渲染高速缓存2378和深度高速缓存2379在一些实施例中也是可用的。像素操作部件2377对数据执行基于像素的操作,虽然在一些示例中,与2D操作相关联的像素操作(例如,位块图采用混合来转移)由2D引擎2341执行,或在显示时间由显示器控制器2343使用重叠显示平面来代替。在一些实施例中,共享的L3高速缓存2375可用于所有的图形部件,从而允许在无需使用主系统存储器的情况下共享数据。

[0250] 在一些实施例中,图形处理器媒体流水线2330包括媒体引擎2337和视频前端2334。在一些实施例中,视频前端2334从命令流传送器2303接收流水线命令。在一些实施例中,媒体流水线2330包括单独的命令流传送器。在一些实施例中,视频前端2334在将媒体命令发送给媒体引擎2337之前处理所述命令。在一些实施例中,媒体引擎2337包括用于生成线程以用于经由线程分派器2331分派给线程执行逻辑2350的线程生成功能性。

[0251] 在一些实施例中,图形处理器2300包括显示器引擎2340。在一些实施例中,显示器引擎2340在处理器2300外部并且经由环形互连2302、或某个其他互连总线或结构与图形处理器耦合。在一些实施例中,显示器引擎2340包括2D引擎2341和显示器控制器2343。在一些实施例中,显示器引擎2340包含能够独立于3D流水线而操作的专用逻辑。在一些实施例中,显示器控制器2343与显示装置(未示出)耦合,所述显示装置可以是系统集成式显示装置(如在膝上型计算机中),或可以是经由显示装置连接器所附连的外部显示装置。

[0252] 在一些实施例中,图形流水线2320和媒体流水线2330可配置成基于多个图形和媒体编程接口来执行操作,并且不特定于任何一个应用编程接口(API)。在一些实施例中,用于图形处理器的驱动器软件将特定于具体图形或媒体库的API调用转换成可由图形处理器来处理的命令。在一些实施例中,为来自Khronos Group的开放图形库(OpenGL)和开放计算语言(OpenCL)、来自Microsoft Corporation的Direct3D库提供支持、或者可以向OpenGL和D3D两者提供支持。还可以为开源计算机视觉库(OpenCV)提供支持。如果可以进行从未来API的流水线至图形处理器的流水线的映射,则还将支持具有可兼容3D流水线的未来API。

[0253] 图形流水线编程

[0254] 图24A是根据一些实施例的图形处理器命令格式2400的框图。图24B是根据实施例的图形处理器命令序列2410的框图。图24A中的实线框示出一般被包括在图形命令中的分量,而虚线框包括可选的或仅包括在图形命令的子集中的分量。图24A的示例性图形处理器命令格式2400包括用于标识命令的目标客户端2402、命令操作代码(操作码)2404和用于命令的相关数据2406的数据字段。在一些命令中还包括子操作码2405和命令大小2408。

[0255] 在一些实施例中,客户端2402指定处理命令数据的图形装置的客户端单元。在一些实施例中,图形处理器命令解析器检查每个命令的客户端字段以调整对命令的进一步处理并将命令数据路由至合适的客户端单元。在一些实施例中,图形处理器客户端单元包括存储器接口单元、渲染单元、2D单元、3D单元、和媒体单元。每个客户端单元具有处理命令的对应的处理流水线。一旦命令被客户端单元接收,则客户端单元读取操作码2404以及(如果存在)子操作码2405以确定要执行的操作。客户端单元使用数据字段2406中的信息来执行命令。针对一些命令,期待显式命令大小2408以指定命令的大小。在一些实施例中,命令解析器基于命令操作码自动地确定命令中的至少一些命令的大小。在一些实施例中,经由双倍字长的倍数对命令进行对齐。

[0256] 图24B中的流程图示出了示例性图形处理器命令序列2410。在一些实施例中,以图形处理器的实施例为特征的数据处理系统的软件或固件使用所示出的命令序列的一个版本来设立、执行并终止图形操作的集合。仅出于示例目的示出并描述了样本命令序列,因为实施例不限于这些特定命令或者此命令序列。此外,所述命令可以作为一批命令在命令序列中被发布,使得图形处理器将至少部分同时处理命令的序列。

[0257] 在一些实施例中,图形处理器命令序列2410可以以流水线转储清除命令2412开始,以促使任何活动的图形流水线完成所述流水线的当前未决命令。在一些实施例中,3D流水线2422和媒体流水线2424不同时进行操作。执行流水线转储清除以促使活动的图形流水线完成任何未决命令。响应于流水线转储清除,用于图形处理器的命令解析器将暂停命令处理,直到活动的绘图引擎完成未决操作且相关的读取高速缓存无效。可选地,渲染高速缓存中被标记为‘脏’的任何数据可以被转储清除到存储器。在一些实施例中,可以针对流水线同步或在将图形处理器置于低功率状态中之前使用流水线转储清除命令2412。

[0258] 在一些实施例中,当命令序列要求图形处理器在流水线之间进行显式切换时,使用流水线选择命令2413。在一些实施例中,在发布流水线命令之前在执行上下文内仅要求一次流水线选择命令2413,除非所述上下文要发布针对两条流水线的命令。在一些实施例中,在经由流水线选择命令2413的流水线切换之前立即要求流水线转储清除命令2412。

[0259] 在一些实施例中,流水线控制命令2414配置用于操作的图形流水线,并用于对3D流水线2422和媒体流水线2424编程。在一些实施例中,流水线控制命令2414配置针对活动流水线的流水线状态。在一个实施例中,流水线控制命令2414被用于流水线同步,以及用于在处理一批命令之前将数据从活动的流水线内的一个或多个高速缓存存储器中清除。

[0260] 在一些实施例中,使用返回缓冲器状态命令2416来配置用于使相应的流水线写入数据的一组返回缓冲器。一些流水线操作要求分配、选择或配置一个或多个返回缓冲器,所述操作在处理期间将中间数据写入到所述一个或多个返回缓冲器中。在一些实施例中,图形处理器还使用一个或多个返回缓冲器以存储输出数据并且执行横跨线程通信。在一些实施例中,返回缓冲器状态2416包括选择返回缓冲器的大小和数量以用于流水线操作的集

合。

[0261] 命令序列中的剩余命令基于用于操作的活动流水线而不同。基于流水线确定2420,针对3D流水线2422或媒体流水线2424来定制命令序列,所述3D流水线2422以3D流水线状态2430开始,所述媒体流水线2424始于媒体流水线状态2440。

[0262] 用于3D流水线状态2430的命令包括用于以下各项的3D状态设置命令:顶点缓冲器状态、顶点元素状态、恒定颜色状态、深度缓冲器状态和将在处理3D图元图元命令之前配置的其他状态变量。至少部分地基于使用中的具体3D API来确定这些命令的值。在一些实施例中,3D流水线状态2430命令还能够选择性地禁用或旁路某些流水线元件(如果将不使用那些元件)。

[0263] 在一些实施例中,3D图元图元2432命令用于提交要由3D流水线处理的3D图元图元。经由3D图元图元2432传递到图形处理器的命令和关联的参数被转发到图形流水线中的顶点获取功能。顶点获取功能使用3D图元图元2432命令数据来产生顶点数据结构。顶点数据结构被存储在一个或多个返回缓冲器中。在一些实施例中,3D图元图元2432命令用于经由顶点着色器对3D图元图元执行顶点操作。为了处理顶点着色器,3D流水线2422将着色器执行线程分派给图形处理器执行单元。

[0264] 在一些实施例中,经由执行2434命令或事件来触发3D流水线2422。在一些实施例中,寄存器写入触发命令执行。在一些实施例中,经由命令序列中的‘走(go)’或‘踢(kick)’命令来触发执行。在一个实施例中,使用流水线同步命令来触发命令执行,以通过图形流水线来转储清除命令序列。3D流水线将执行针对3D图元图元的几何处理。一旦操作完成,对所得几何对象进行光栅化,并且像素引擎给所得像素上色。针对那些操作还可包括用于控制像素着色和像素后端操作的附加命令。

[0265] 在一些实施例中,当执行媒体操作时,图形处理器命令序列2410沿媒体流水线2424路径。一般地,针对媒体流水线2424的编程的特定使用和方式取决于要执行的媒体或计算操作。在媒体解码期间,可将特定的媒体解码操作卸载到媒体流水线。在一些实施例中,还可以对媒体流水线进行旁路并且可以整体地或部分地执行媒体解码(使用由一个或多个通用处理核提供的资源)。在一个实施例中,媒体流水线还包括用于通用图形处理器单元(GPGPU)操作的元件,其中,图形处理器用于使用计算着色器程序来执行SIMD向量操作,所述计算着色器程序不显式涉及图形图元图元的渲染。

[0266] 在一些实施例中,以与3D流水线2422类似的方式配置媒体流水线2424。在媒体对象命令2442之前将一组媒体流水线状态命令2440分派到或放置到命令队列中。在一些实施例中,媒体流水线状态命令2440包括用于配置媒体流水线元件的数据,所述媒体流水线元件将用于处理媒体对象。这包括用于配置媒体流水线内的视频解码和视频编码逻辑的数据(诸如编码或解码格式)。在一些实施例中,媒体流水线状态命令2440还支持将一个或多个指针用于包含一批状态设置的“间接”状态元件。

[0267] 在一些实施例中,媒体对象命令2442将指针供应给用于由媒体流水线处理的媒体对象。媒体对象包括包含要处理的视频数据的存储器缓冲器。在一些实施例中,在发布媒体对象命令2442之前,所有的媒体流水线状态必须是有效的。一旦流水线状态被配置并且媒体对象命令2442被排队,则经由执行命令2444或等效的执行事件(例如,寄存器写入)来触发媒体流水线2424。然后可以通过由3D流水线2422或媒体流水线2424提供的操作对来自媒

体流水线2424的输出进行后处理。在一些实施例中,以与媒体操作类似的方式来配置和执行GPGPU操作。

[0268] 图形软件架构

[0269] 图25示出根据一些实施例的用于数据处理系统2500的示例性图形软件架构。在一些实施例中,软件架构包括3D图形应用2510、操作系统2520、以及至少一个处理器2530。在一些实施例中,处理器2530包括图形处理器2532以及一个或多个通用处理器核2534。图形应用2510和操作系统2520各自在数据处理系统的系统存储器2550中执行。

[0270] 在一些实施例中,3D图形应用2510包含一个或多个着色器程序,所述着色器程序包括着色器指令2512。着色器语言指令可以采用高级着色器语言,诸如高级着色器语言(HLSL)或OpenGL着色器语言(GLSL)。所述应用还包括采用适合于由通用处理器核2534执行的机器语言的可执行指令2514。所述应用还包括由顶点数据定义的图形对象2516。

[0271] 在一些实施例中,操作系统2520是来自Microsoft Corporation的Microsoft® Windows® 操作系统、专有类UNIX操作系统或开源类UNIX操作系统(使用Linux内核的变形)。当Direct3D API在使用时,操作系统2520使用前端着色器编译器2524以将采用HLSL的任何着色器指令2512编译成较低级着色器语言。所述编译可以是即时制(JIT)编译,或者所述应用可执行着色器预编译。在一些实施例中,在对3D图形应用2510进行编译期间,将高级着色器编译成低级着色器。

[0272] 在一些实施例中,用户模式图形驱动器2526包含后端着色器编译器2527,所述后端着色器编译器2527用于将着色器指令2512转化成硬件特定表示。当OpenGL API在使用中时,采用GLSL高级语言的着色器指令2512被传递到用户模式图形驱动器2526以用于编译。在一些实施例中,用户模式图形驱动器2526使用操作系统内核模式功能2528来与内核模式图形驱动器2529进行通信。在一些实施例中,内核模式图形驱动器2529与图形处理器2532进行通信以分派命令和指令。

[0273] IP核实现

[0274] 至少一个实施例的一个或多个方面可由存储在机器可读介质上的表示性代码来实现,所述机器可读介质表示和/或定义诸如处理器的集成电路内的逻辑。例如,机器可读介质可以包括表示处理器内的各种逻辑的指令。当由机器读取时,所述指令可以促使所述机器制作用于执行本文描述的技术的逻辑。此类表示(称为“IP核”)是用于集成电路的逻辑的可重复使用单元,其可以作为描述集成电路的结构硬件模型而存储在有形的机器可读介质上。可以将硬件模型供应至各种消费者或制造设施,其在制造所述集成电路的制作机器上加载硬件模型。可以制作所述集成电路,使得所述电路执行与本文描述的任何实施例相关联地描述的操作。

[0275] 图26是根据实施例的示出IP核开发系统2600的框图,所述IP核开发系统2600可用于制造集成电路以执行操作。IP核开发系统2600可以用于生成可合并到更大的设计中或用于构建整个集成电路(例如,SOC集成电路)的模块化、可重复使用设计。设计设施2630可采用高级编程语言(例如,C/C++)生成IP核设计的软件仿真2610。软件仿真2610可用于设计、测试并验证IP核的行为。然后可从仿真模型2600创建或合成寄存器转移级别(RTL)设计。RTL设计2615是对硬件寄存器之间的数字信号的流进行建模的集成电路(包括使用建模的数字信号来执行的相关联逻辑)的行为的提取(abstraction)。除了RTL设计2615之外,还可

以创建、设计或合成在逻辑级别或晶体管级别的较低级设计。因此,初始设计和仿真的具体细节可以变化。

[0276] 可由设计设施进一步将RTL设计2615或等效物合成到硬件模型2620中,所述硬件模型2620可以采用硬件描述语言(HDL)或物理设计数据的某一其他表示。可以进一步仿真或测试HDL以验证IP核设计。可以使用非易失性存储器2640(例如,硬盘、闪存存储器或任何非易失性存储介质)来存储IP核设计以便递送到第3方制作设施2665。备选地,可以通过有线连接2650或无线连接2660来传送(例如,经由互联网)IP核设计。制作设施2665然后可以制作至少部分地基于IP核设计的集成电路。所制作的集成电路可被配置成执行依照本文描述的至少一个实施例的操作。

[0277] 图27是根据实施例的示出示例性芯片上系统集成电路2700的框图,可使用一个或多个IP核来制作所述芯片上系统集成电路2700。示例性集成电路包括一个或多个应用处理器2705(例如,CPU)、至少一个图形处理器2710,并且可以另外包括图像处理器2715和/或视频处理器2720,其中的任何可以是来自相同或多个不同设计设施的模块化IP核。集成电路包括外围或总线逻辑,其包括USB控制器2725、UART控制器2730、SPI/SDIO控制器2735、I²S/I²C控制器2740。另外,集成电路可以包括显示装置2745,所述显示装置2745耦合至高清晰度多媒体接口(HDMI)控制器2750和移动行业处理器接口(MIPI)显示器接口2755中一个或多个。可以由闪存存储器子系统2760(包括闪存存储器和闪存存储器控制器)来提供存储。可经由存储器控制器2765来提供存储器接口以用于访问SDRAM或SRAM存储器装置。一些集成电路另外包括嵌入式安全引擎2770。

[0278] 另外,其他逻辑和电路可被包括在集成电路2700的处理器中,包括附加的图形处理器/核、外围接口控制器或通用处理器核。

[0279] 在一个示例中,如相对于图6-10B和在以下示例中所描述的,图形处理流水线1400(图14)包含拓扑着色器。

[0280] 本公开/申请提供了如下的技术方案:

[0281] 1. 一种系统,包括:

[0282] 显示器,用于呈现视觉内容;

[0283] 存储器,用于存储一组指令;以及

[0284] 耦合于所述显示器和所述存储器的半导体封装设备,所述半导体封装设备包含:

[0285] 衬底;

[0286] 耦合于所述衬底的主机处理器,其中当由所述主机处理器执行时,该组指令促使所述主机处理器生成与所述视觉内容关联的对象描述,以及

[0287] 耦合于所述衬底的图形处理器,其中所述图形处理器包含进行以下操作的逻辑:

[0288] 在拓扑着色器接收所述对象描述,

[0289] 基于所述对象描述在所述拓扑着色器生成一组多边形,以及

[0290] 将该组多边形发送到顶点着色器。

[0291] 2. 如技术方案1所述的系统,其中所述逻辑将:

[0292] 相对于一个或多个多边形来检测可视性条件的缺乏,以及

[0293] 响应于可视性条件的所述缺乏将所述一个或多个多边形排除出该组多边形。

[0294] 3. 如技术方案2所述的系统,其中可视性条件的所述缺乏将包含遮蔽状态或离屏

状态中的一个或多个。

[0295] 4. 如技术方案1所述的系统,其中所述逻辑将基于深度信息来确定与所述对象描述关联的细节级别。

[0296] 5. 如技术方案4所述的系统,其中所述逻辑将标识对应于所述对象描述的子八叉树,并且其中所述拓扑着色器将被调用以用于所标识的子八叉树。

[0297] 6. 如技术方案4所述的系统,其中所述逻辑将在两个细节级别之间内插。

[0298] 7. 一种设备,包括:

[0299] 衬底;以及

[0300] 耦合到所述衬底的逻辑,其中所述逻辑在可配置逻辑或固定功能性硬件逻辑中的一个或多个中实现,所述逻辑用于:

[0301] 在拓扑着色器接收对象描述,

[0302] 基于所述对象描述在所述拓扑着色器生成一组多边形,以及

[0303] 将该组多边形发送到顶点着色器。

[0304] 8. 如技术方案7所述的设备,其中所述逻辑将:

[0305] 相对于一个或多个多边形来检测可视性条件的缺乏,以及

[0306] 响应于可视性条件的所述缺乏将所述一个或多个多边形排除出该组多边形。

[0307] 9. 如技术方案8所述的设备,其中可视性条件的所述缺乏将包含遮蔽状态或离屏状态中的一个或多个。

[0308] 10. 如技术方案7所述的设备,其中所述逻辑将基于深度信息来确定与所述对象描述关联的细节级别。

[0309] 11. 如技术方案10所述的设备,其中所述逻辑将标识对应于所述对象描述的子八叉树,并且其中所述拓扑着色器将被调用以用于所标识的子八叉树。

[0310] 12. 如技术方案10所述的设备,其中所述逻辑将在两个细节级别之间内插。

[0311] 13. 一种方法,包括:

[0312] 在拓扑着色器接收对象描述;

[0313] 基于所述对象描述在所述拓扑着色器生成一组多边形;以及

[0314] 将该组多边形发送到顶点着色器。

[0315] 14. 如技术方案13所述的方法,其中生成该组多边形包含:

[0316] 相对于一个或多个多边形来检测可视性条件的缺乏;以及

[0317] 响应于可视性条件的所述缺乏将所述一个或多个多边形排除出该组多边形。

[0318] 15. 如技术方案14所述的方法,其中可视性条件的所述缺乏将包含遮蔽状态或离屏状态中的一个或多个。

[0319] 16. 如技术方案13所述的方法,其中生成该组多边形包含基于深度信息来确定与所述对象描述关联的细节级别。

[0320] 17. 如技术方案16所述的方法,其中确定所述细节级别包含标识对应于所述对象描述的子八叉树,并且其中所述拓扑着色器将被调用以用于所标识的子八叉树。

[0321] 18. 如技术方案16所述的方法,其中确定所述细节级别包含在两个细节级别之间内插。

[0322] 19. 至少一种计算机可读存储介质,包括一组指令,所述指令在由计算系统执行

时促使所述计算系统:

[0323] 在拓扑着色器接收对象描述;

[0324] 基于所述对象描述在所述拓扑着色器生成一组多边形;以及

[0325] 将该组多边形发送到顶点着色器。

[0326] 20. 如技术方案19所述的至少一种计算机可读存储介质,其中所述指令在被执行时促使所述计算系统:

[0327] 相对于一个或多个多边形来检测可视性条件的缺乏;以及

[0328] 响应于可视性条件的所述缺乏将所述一个或多个多边形排除出该组多边形。

[0329] 21. 如技术方案20所述的至少一种计算机可读存储介质,其中可视性条件的所述缺乏将包含遮蔽状态或离屏状态中的一个或多个。

[0330] 22. 如技术方案19所述的至少一种计算机可读存储介质,其中所述指令在被执行时促使所述计算系统基于深度信息来确定与所述对象描述关联的细节级别。

[0331] 23. 如技术方案22所述的至少一种计算机可读存储介质,其中所述指令在被执行时促使所述计算系统标识对应于所述对象描述的子八叉树,并且其中所述拓扑着色器将被调用以用于所标识的子八叉树。

[0332] 24. 如技术方案22所述的至少一种计算机可读存储介质,其中所述指令在被执行时促使所述计算系统在两个细节级别之间内插。

[0333] 附加注释和示例

[0334] 示例1可包含性能提高的计算系统,包括:显示器,用于呈现视觉内容;存储器,用于存储一组指令;以及耦合于所述显示器和所述存储器的半导体封装设备,所述半导体封装设备包含:衬底;耦合于所述衬底的主机处理器,其中当由所述主机处理器执行时,该组指令促使所述主机处理器生成与所述视觉内容关联的对象描述;以及耦合于所述衬底的图形处理器,其中所述图形处理器包含进行以下操作的逻辑:在拓扑着色器接收所述对象描述、基于所述对象描述在所述拓扑着色器生成一组多边形、并将该组多边形发送到顶点着色器。

[0335] 示例2可包含示例1所述的系统,其中所述逻辑将相对于一个或多个多边形来检测可视性条件的缺乏,并响应于可视性条件的所述缺乏将所述一个或多个多边形排除出该组多边形。

[0336] 示例3可包含示例2所述的系统,其中可视性条件的所述缺乏将包含遮蔽状态或离屏状态中的一个或多个。

[0337] 示例4可包含示例1到3中任一项所述的系统,其中所述逻辑将基于深度信息来确定与所述对象描述关联的细节级别。

[0338] 示例5可包含示例4所述的系统,其中所述逻辑将标识对应于所述对象描述的子八叉树,并且其中所述拓扑着色器将被调用以用于所标识的子八叉树。

[0339] 示例6可包含示例4所述的系统,其中所述逻辑将在两个细节级别之间内插。

[0340] 示例7可包含半导体封装设备,包括:衬底;以及耦合到所述衬底的逻辑,其中所述逻辑在可配置逻辑或固定功能性硬件逻辑中的一个或多个中实现,所述逻辑用于:在拓扑着色器接收对象描述、基于所述对象描述在所述拓扑着色器生成一组多边形、并将该组多边形发送到顶点着色器。

[0341] 示例8可包含示例7所述的设备,其中所述逻辑将相对于一个或多个多边形来检测可视性条件的缺乏,并响应于可视性条件的所述缺乏将所述一个或多个多边形排除出该组多边形。

[0342] 示例9可包含示例8所述的设备,其中可视性条件的所述缺乏将包含遮蔽状态或离屏状态中的一个或多个。

[0343] 示例10可包含如示例7到9中任一项所述的设备,其中所述逻辑将基于深度信息来确定与所述对象描述关联的细节级别。

[0344] 示例11可包含示例10所述的设备,其中所述逻辑将标识对应于所述对象描述的子八叉树,并且其中所述拓扑着色器将被调用以用于所标识的子八叉树。

[0345] 示例12可包含示例10所述的设备,其中所述逻辑将在两个细节级别之间内插。

[0346] 示例13可包含操作半导体封装设备的方法,包括:在拓扑着色器接收对象描述、基于所述对象描述在所述拓扑着色器生成一组多边形、并将该组多边形发送到顶点着色器。

[0347] 示例14可包含示例13所述的方法,其中生成该组多边形包含相对于一个或多个多边形来检测可视性条件的缺乏,并响应于可视性条件的所述缺乏将所述一个或多个多边形排除出该组多边形。

[0348] 示例15可包含示例14所述的方法,其中可视性条件的所述缺乏将包含遮蔽状态或离屏状态中的一个或多个。

[0349] 示例16可包含示例13到15中任一项所述的方法,其中生成该组多边形包含基于深度信息来确定与所述对象描述关联的细节级别。

[0350] 示例17可包含示例16所述的方法,其中确定所述细节级别包含标识对应于所述对象描述的子八叉树,并且其中所述拓扑着色器将被调用以用于所标识的子八叉树。

[0351] 示例18可包含示例16所述的方法,其中确定所述细节级别包含在两个细节级别之间内插。

[0352] 示例19可包含至少一种计算机可读存储介质,包括一组指令,所述指令在由计算系统执行时促使所述计算系统:在拓扑着色器接收对象描述、基于所述对象描述在所述拓扑着色器生成一组多边形、并将该组多边形发送到顶点着色器。

[0353] 示例20可包含示例19所述的至少一种计算机可读存储介质,其中所述指令在被执行时促使所述计算系统相对于一个或多个多边形来检测可视性条件的缺乏,并响应于可视性条件的所述缺乏将所述一个或多个多边形排除出该组多边形。

[0354] 示例21可包含示例20所述的至少一种计算机可读存储介质,其中可视性条件的所述缺乏将包含遮蔽状态或离屏状态中的一个或多个。

[0355] 示例22可包含示例19到21中任一项所述的至少一种计算机可读存储介质,其中所述指令在被执行时促使所述计算系统基于深度信息来确定与所述对象描述关联的细节级别。

[0356] 示例23可包含示例22所述的至少一种计算机可读存储介质,其中所述指令在被执行时促使所述计算系统标识对应于所述对象描述的子八叉树,并且其中所述拓扑着色器将被调用以用于所标识的子八叉树。

[0357] 示例24可包含示例22所述的至少一种计算机可读存储介质,其中所述指令在被执行时促使所述计算系统在两个细节级别之间内插。

[0358] 示例25可包含半导体封装设备,所述半导体封装设备包括:用于在拓扑着色器接收对象描述的组件;用于基于所述对象描述在所述拓扑着色器生成一组多边形的组件;以及用于将该组多边形发送到顶点着色器的组件。

[0359] 示例26可包含示例25所述的设备,其中用于生成该组多边形的组件包含用于相对于一个或多个多边形来检测可视性条件的缺乏的组件,以及用于响应于可视性条件的所述缺乏将所述一个或多个多边形排除出该组多边形的组件。

[0360] 示例27可包含示例26所述的设备,其中可视性条件的所述缺乏将包含遮蔽状态或离屏状态中的一个或多个。

[0361] 示例28可包含如示例25到27中任一项所述的设备,其中用于生成该组多边形的组件包含用于基于深度信息来确定与所述对象描述关联的细节级别的组件。

[0362] 示例29可包含示例28所述的设备,其中用于确定所述细节级别的组件包含用于标识对应于所述对象描述的子八叉树的组件,并且其中所述拓扑着色器将被调用以用于所标识的子八叉树。

[0363] 示例30可包含示例28所述的设备,其中用于确定所述细节级别的组件包含在两个细节级别之间内插。

[0364] 本文描述的技术可因此在调用顶点着色器之间使用图形流水线中新的级在运行于图形处理器上时确定要渲染的拓扑的复杂度/质量。因此,图形处理器可在不用将非必要数据从主机处理器存储器转移到专用图形存储器的情况下生成拓扑。可程序地生成几何,同时最小化在生成多边形时调用的数据的量。

[0365] 术语“耦合”此处可以用来指所讨论的部件之间任何类型的直接或间接的关系,并可以应用于电学、机械、流体、光学、电磁、电机或其他连接。此外,术语“第一”、“第二”等此处可以仅用来促进讨论,并不带有任何具体的时间或时间顺序的显著性意义,除非另外指明。而且,应理解不定冠词“一”或“一个”带有“一个或多个”或“至少一个”的意思。

[0366] 如在本申请和权利要求书中使用的,由术语“一个或多个”联接的项的列表可意指所列项的任何组合。例如,短语“A、B或C中的一个或多个”可意指A、B、C;A和B;A和C;B和C;或A、B和C。

[0367] 上面已经参考特定实施例描述了实施例。然而,本领域内技术人员将理解,可以对其进行各种修改和改变而不偏离如在所附的权利要求书中所阐述的实施例的较宽精神和范围。因此认为前述描述和附图是说明性的而不是限制性的。

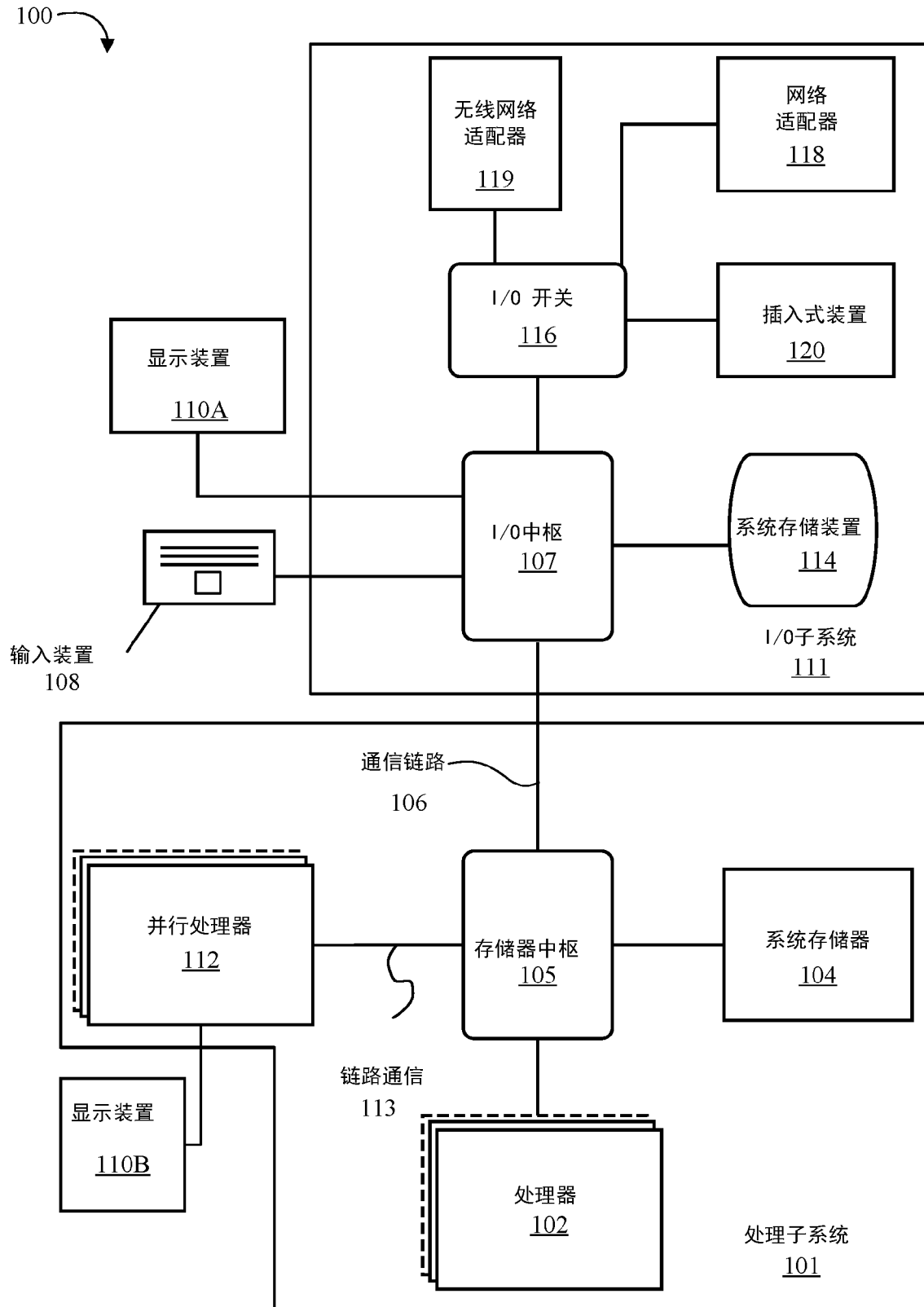


图 1

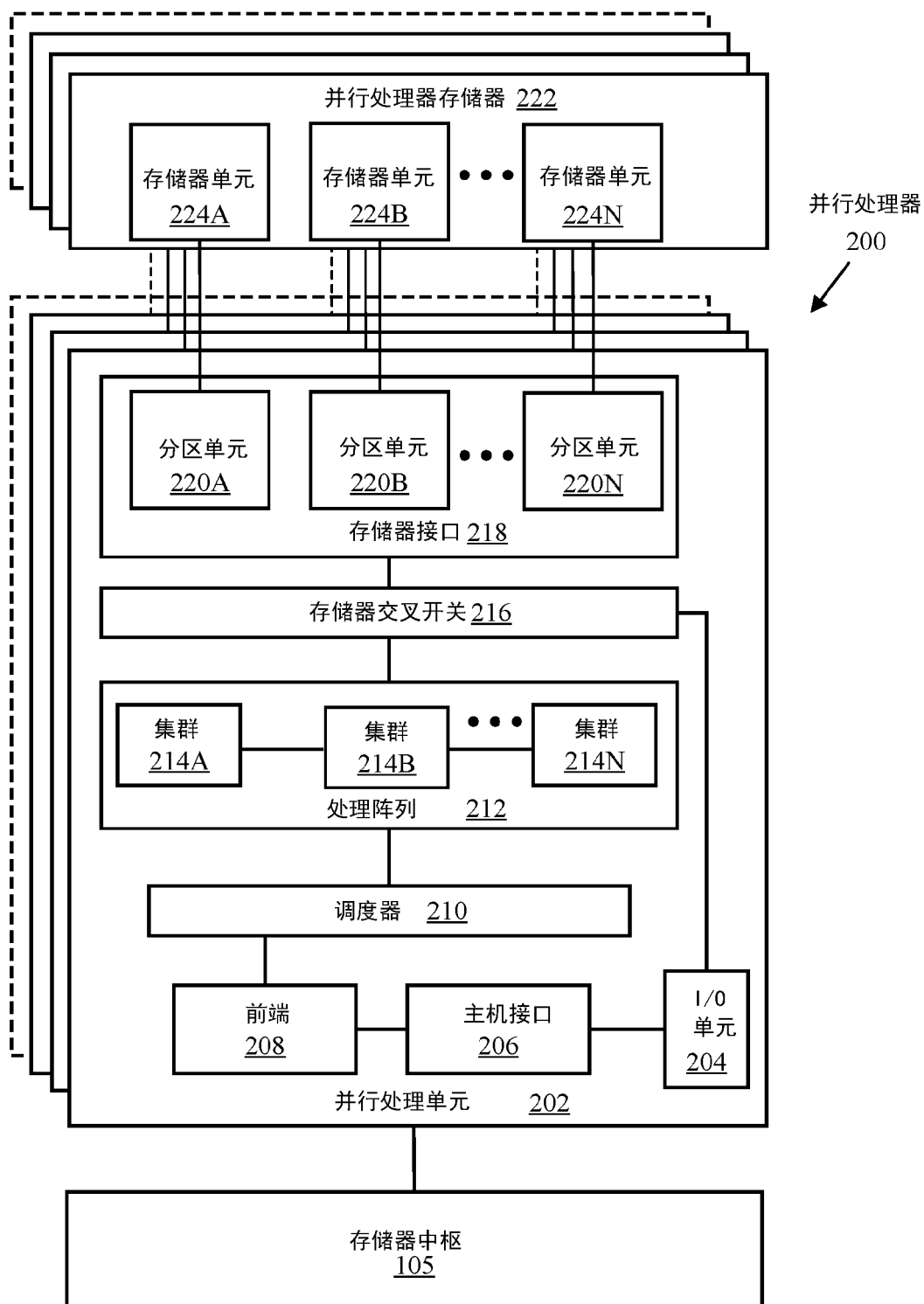


图 2A

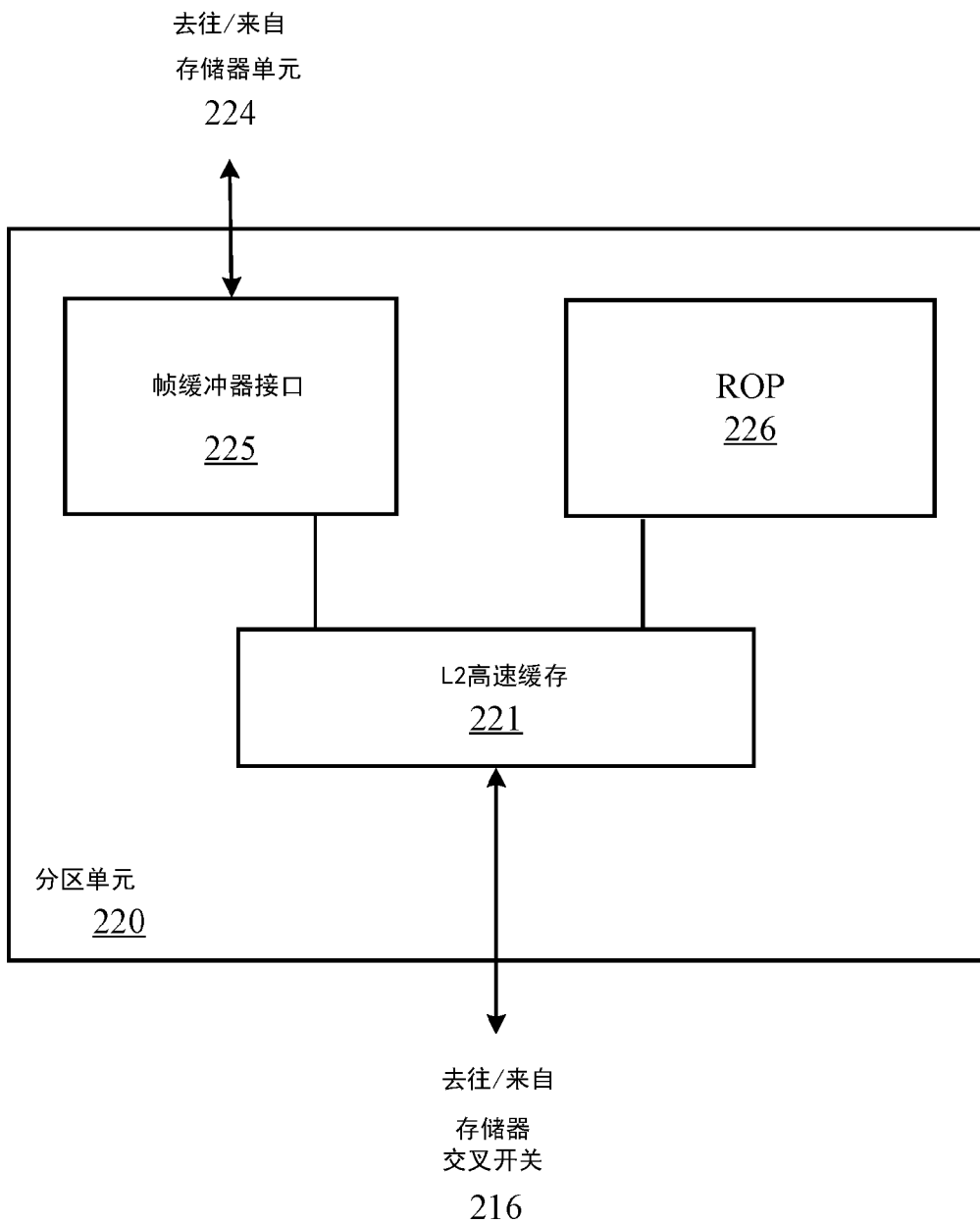


图 2B

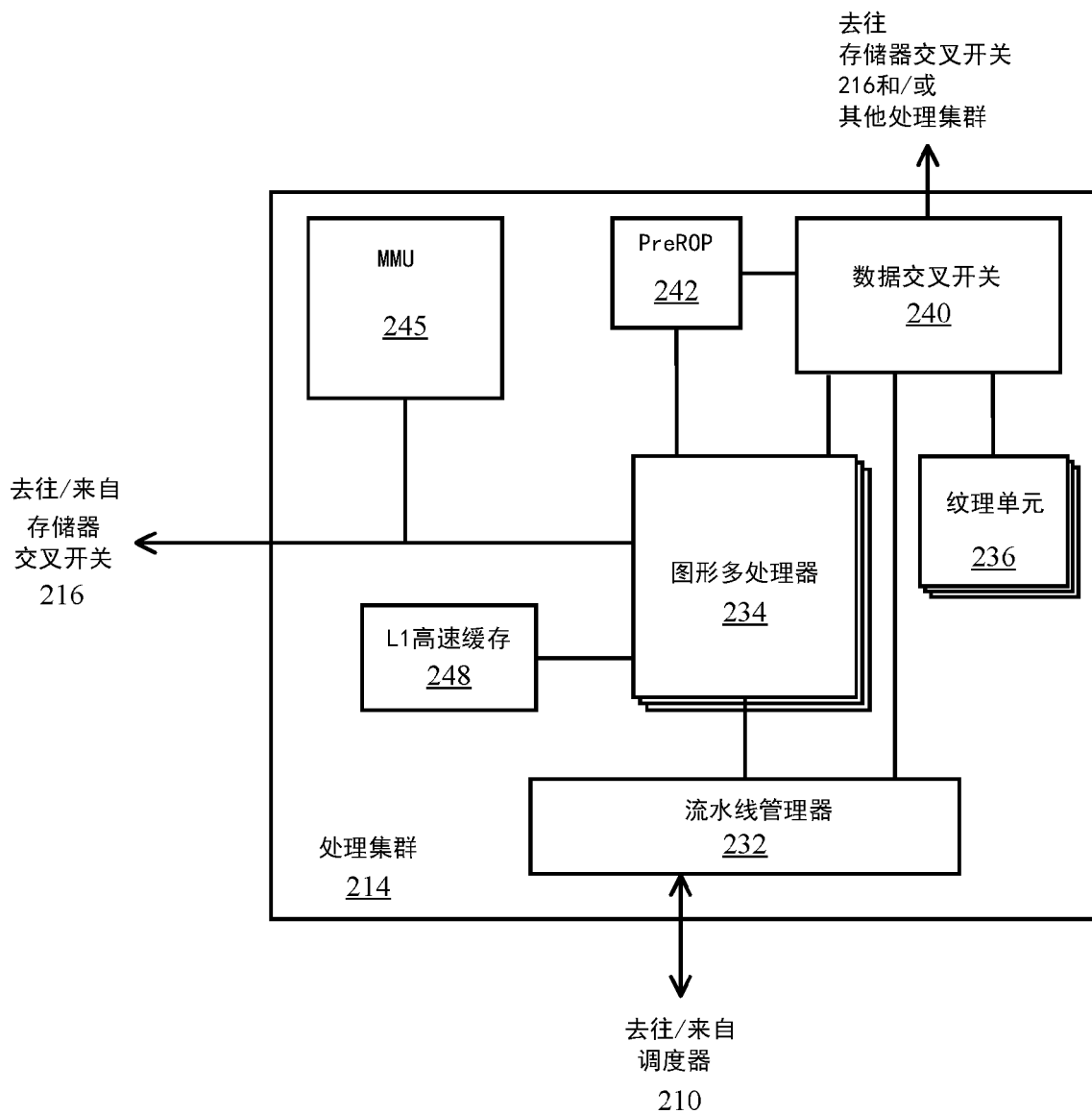


图 2C

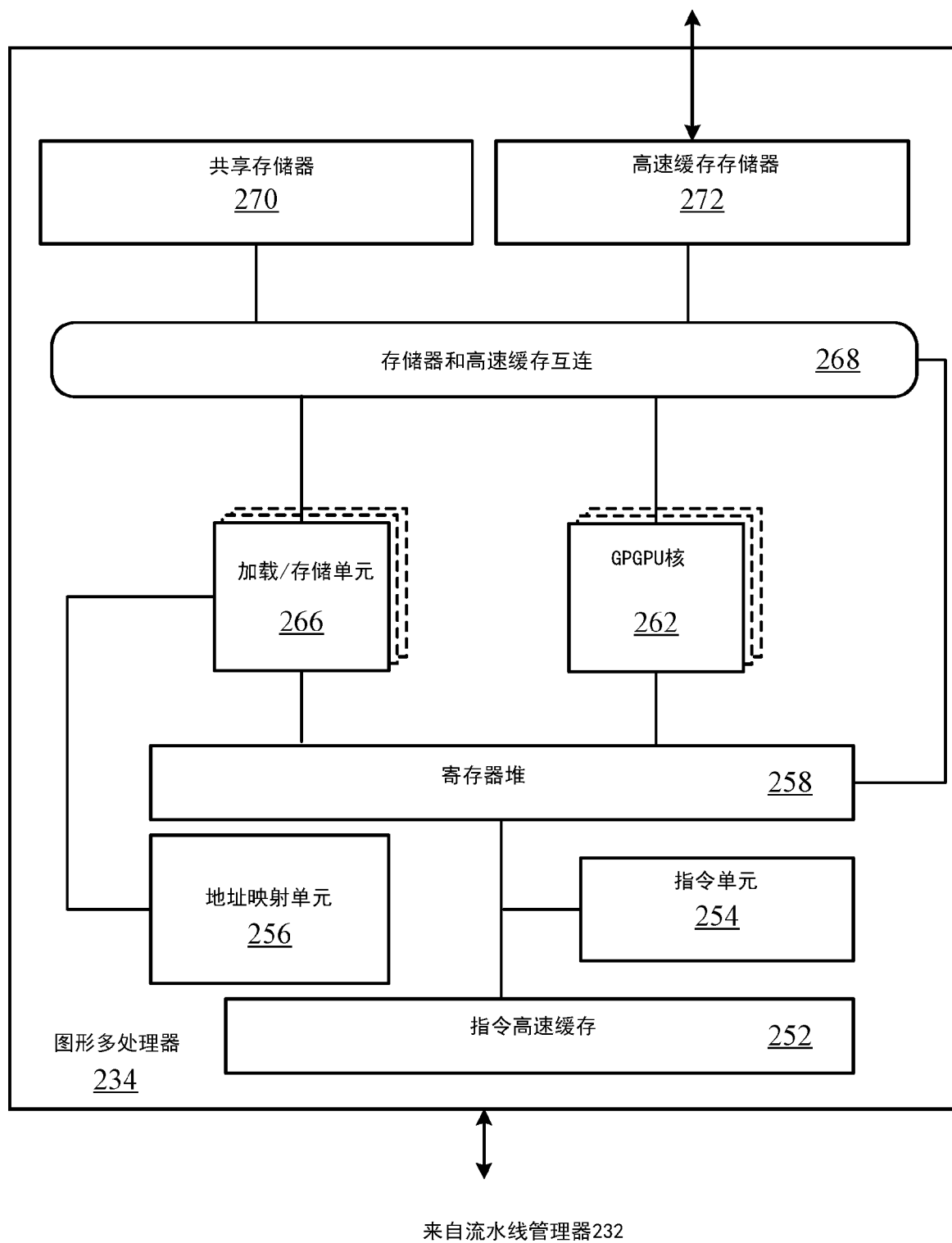


图 2D

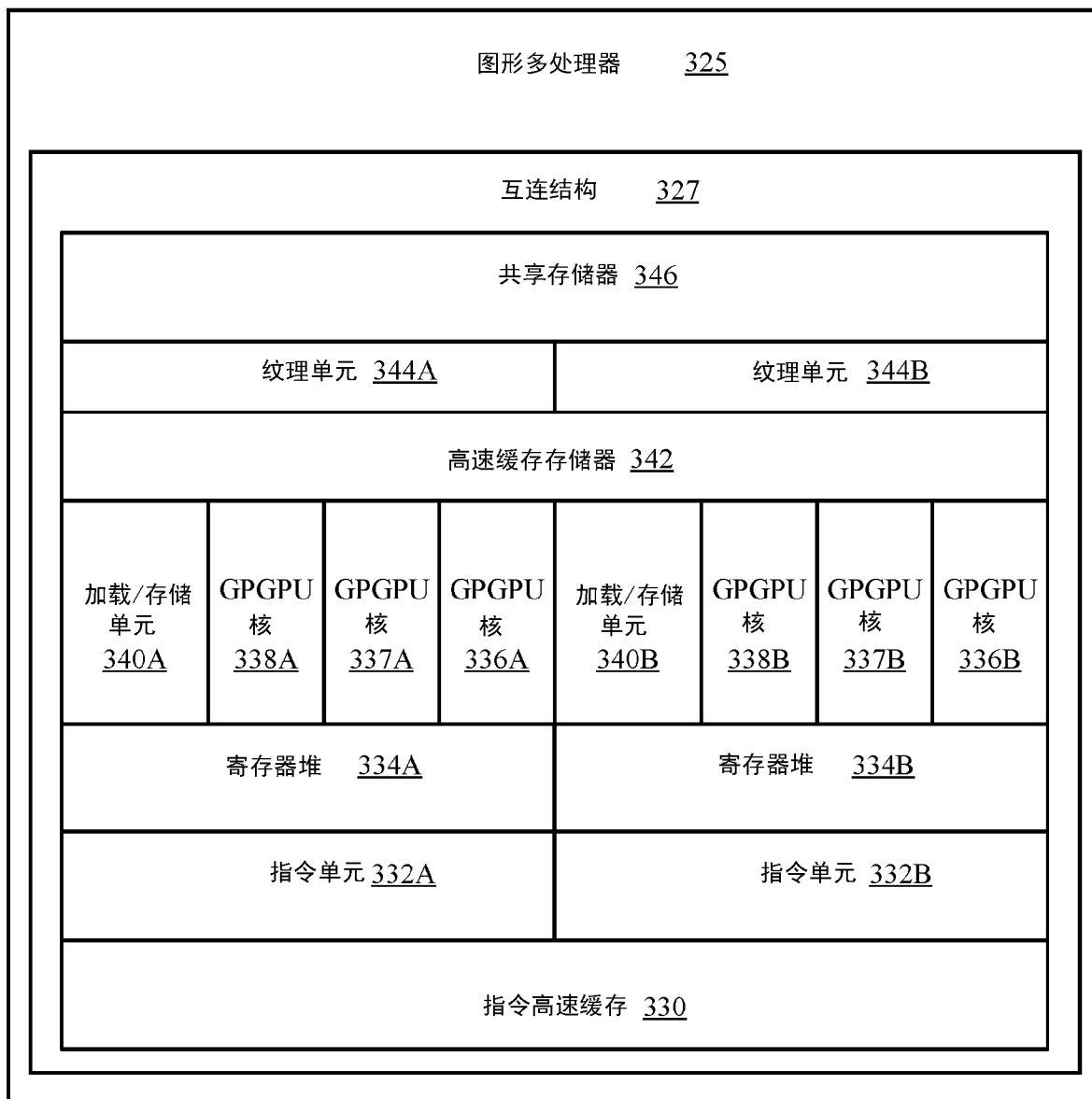


图 3A

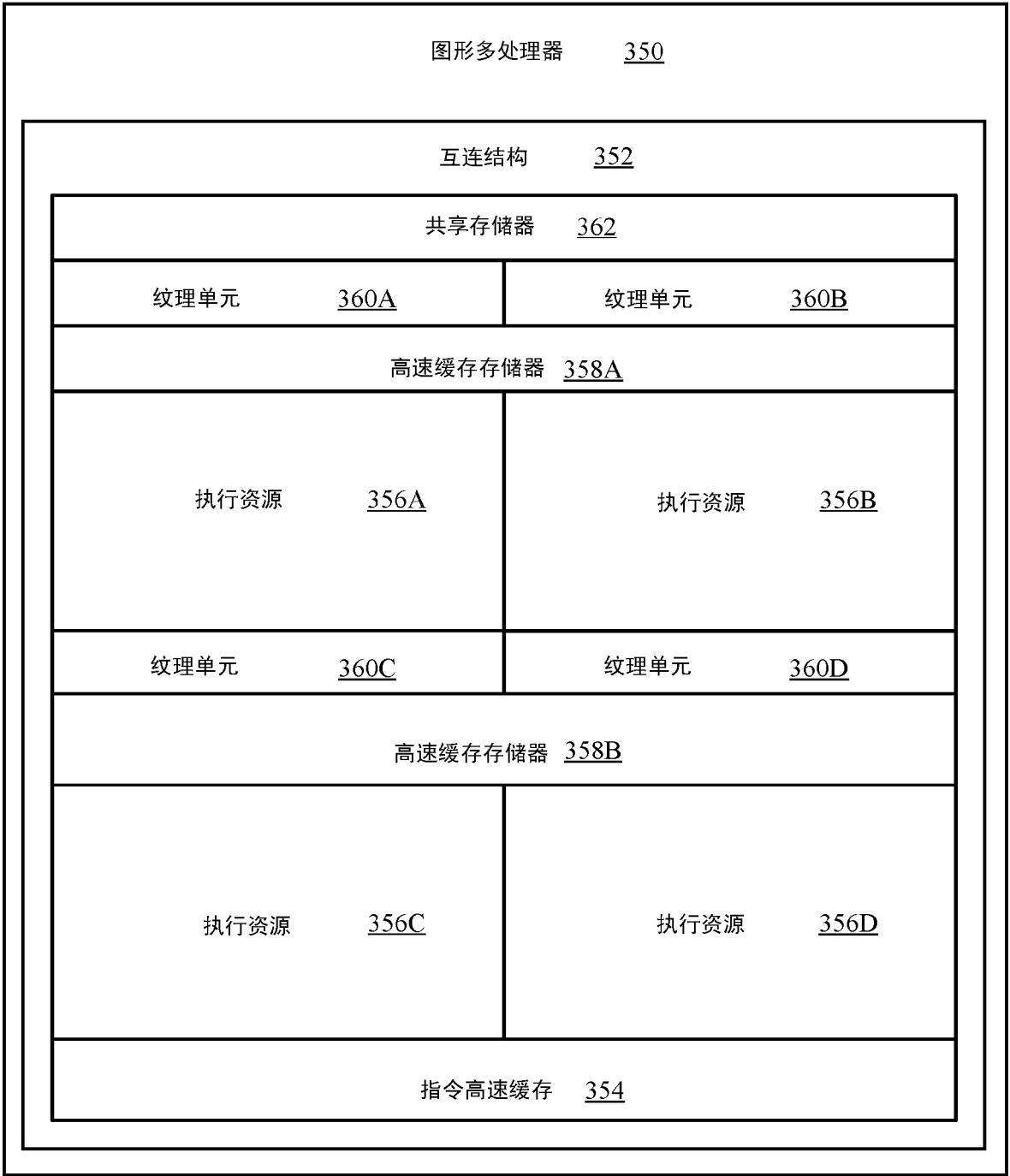


图 3B

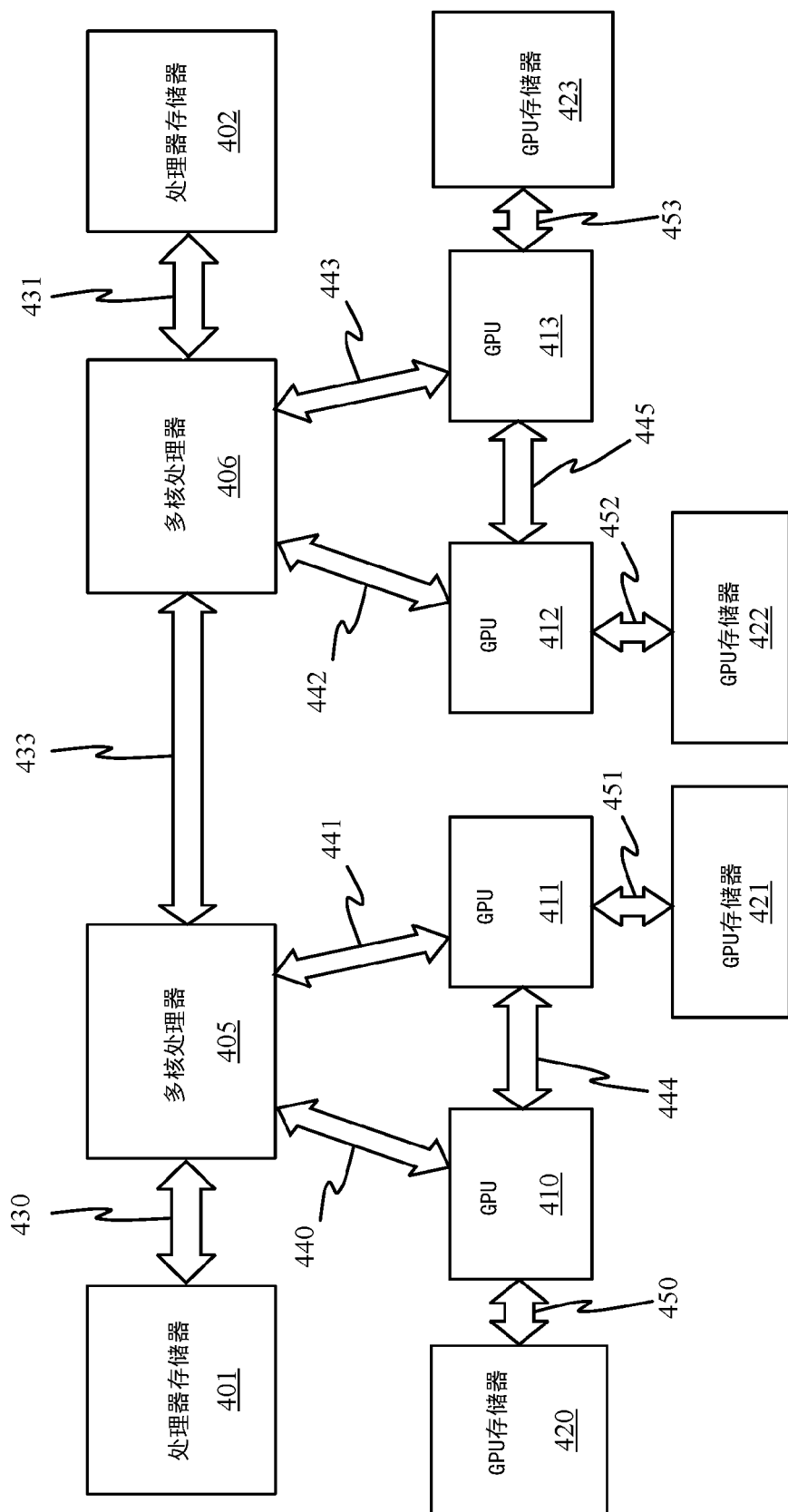


图 4A

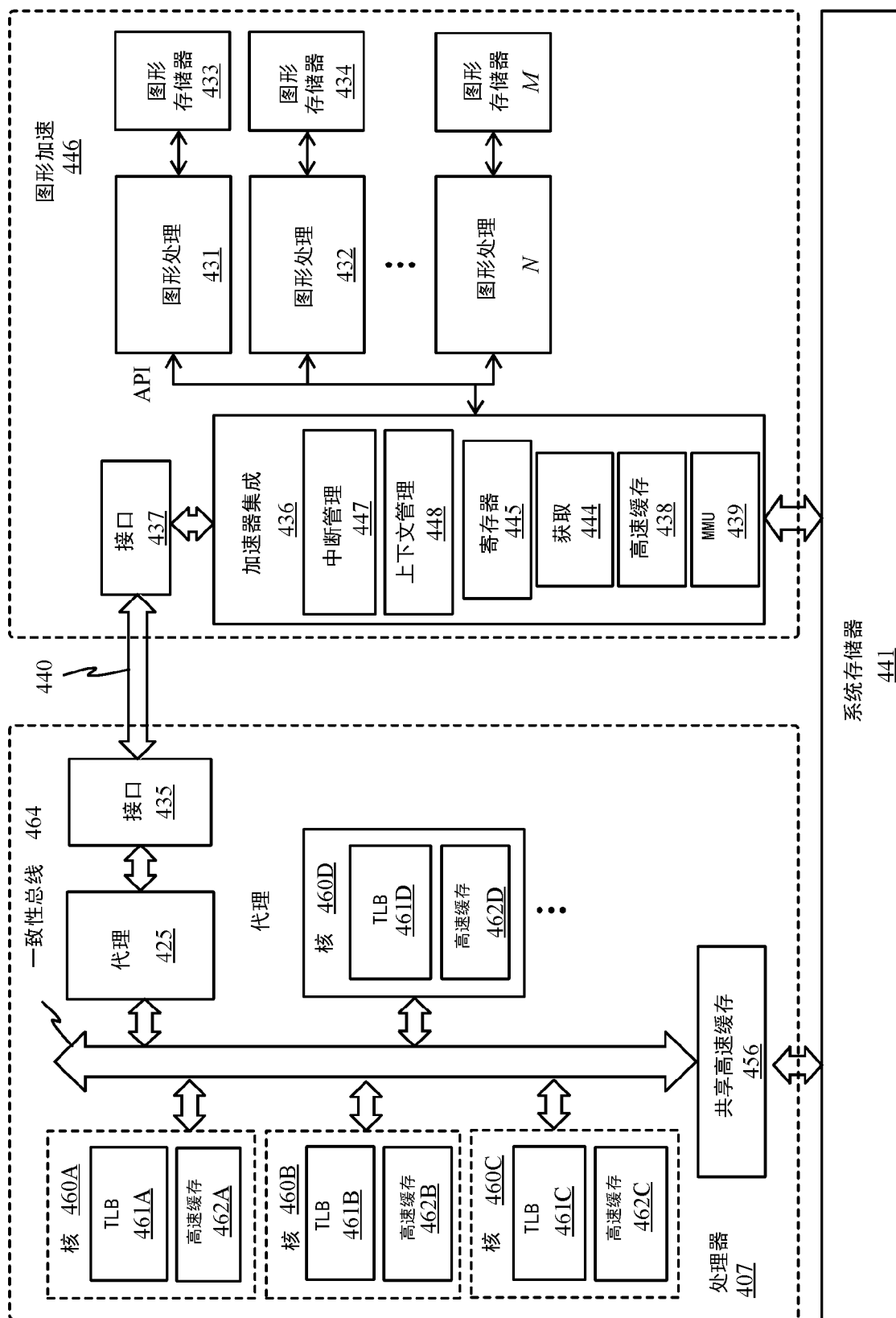


图 4B

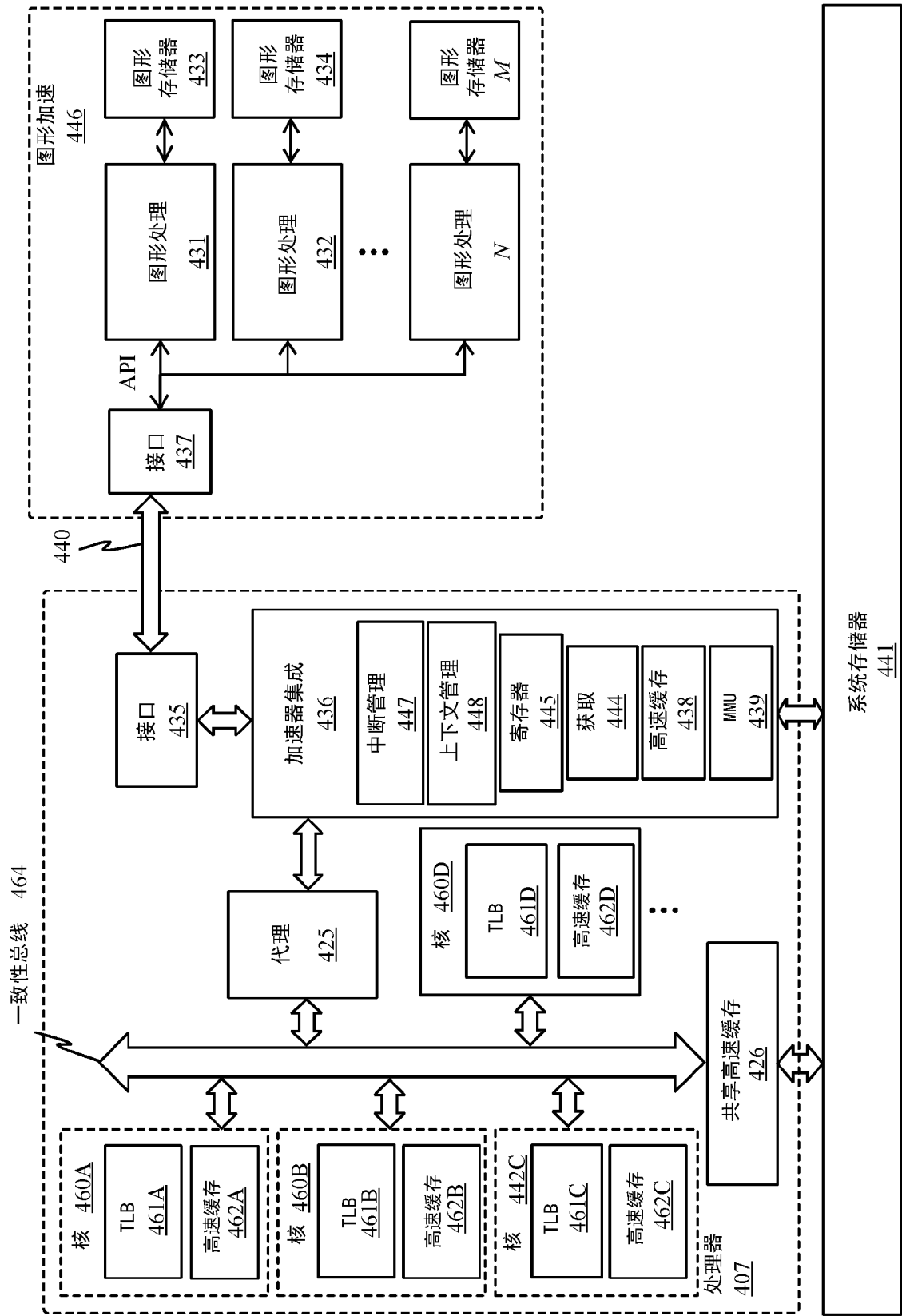


图 4C

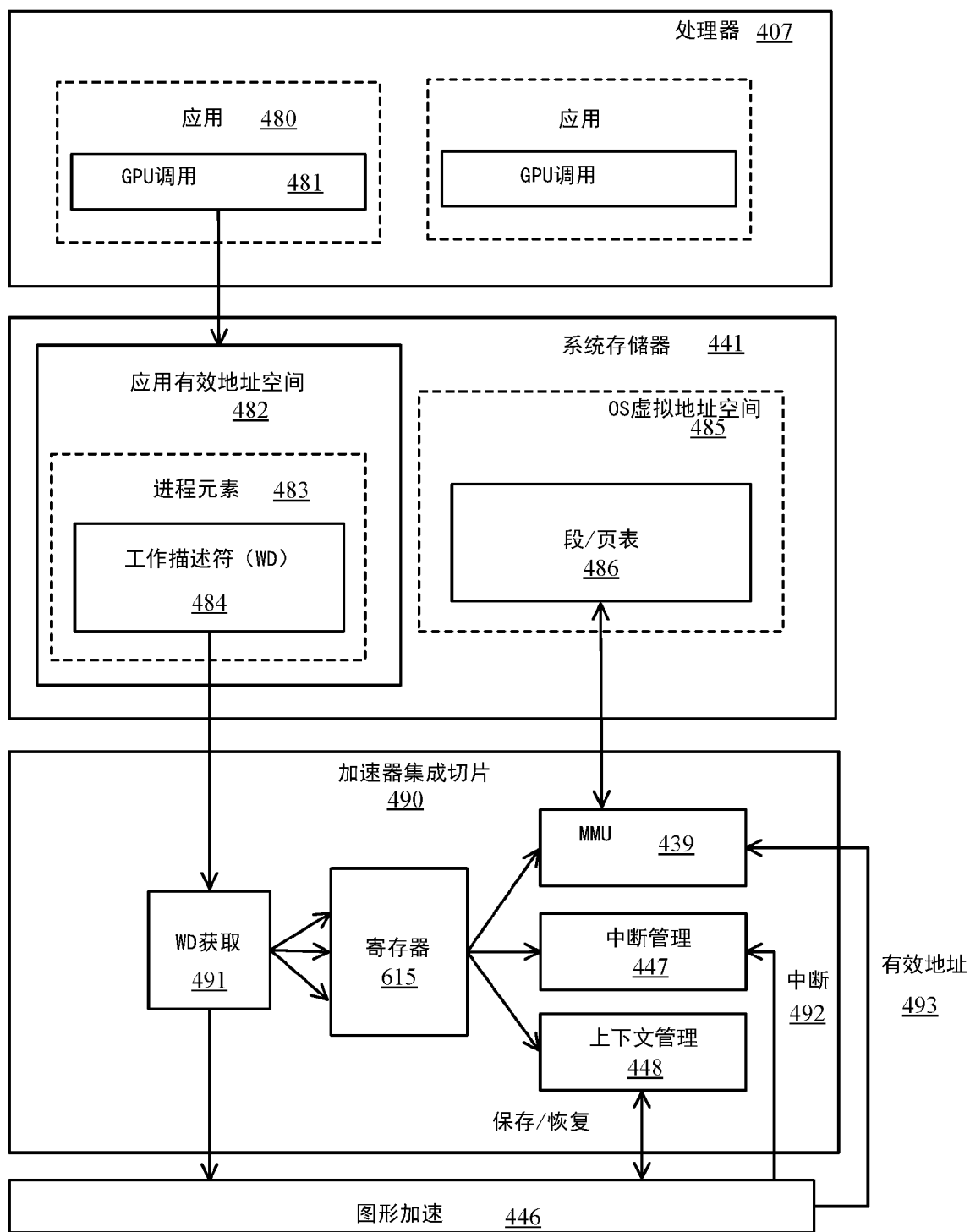


图 4D

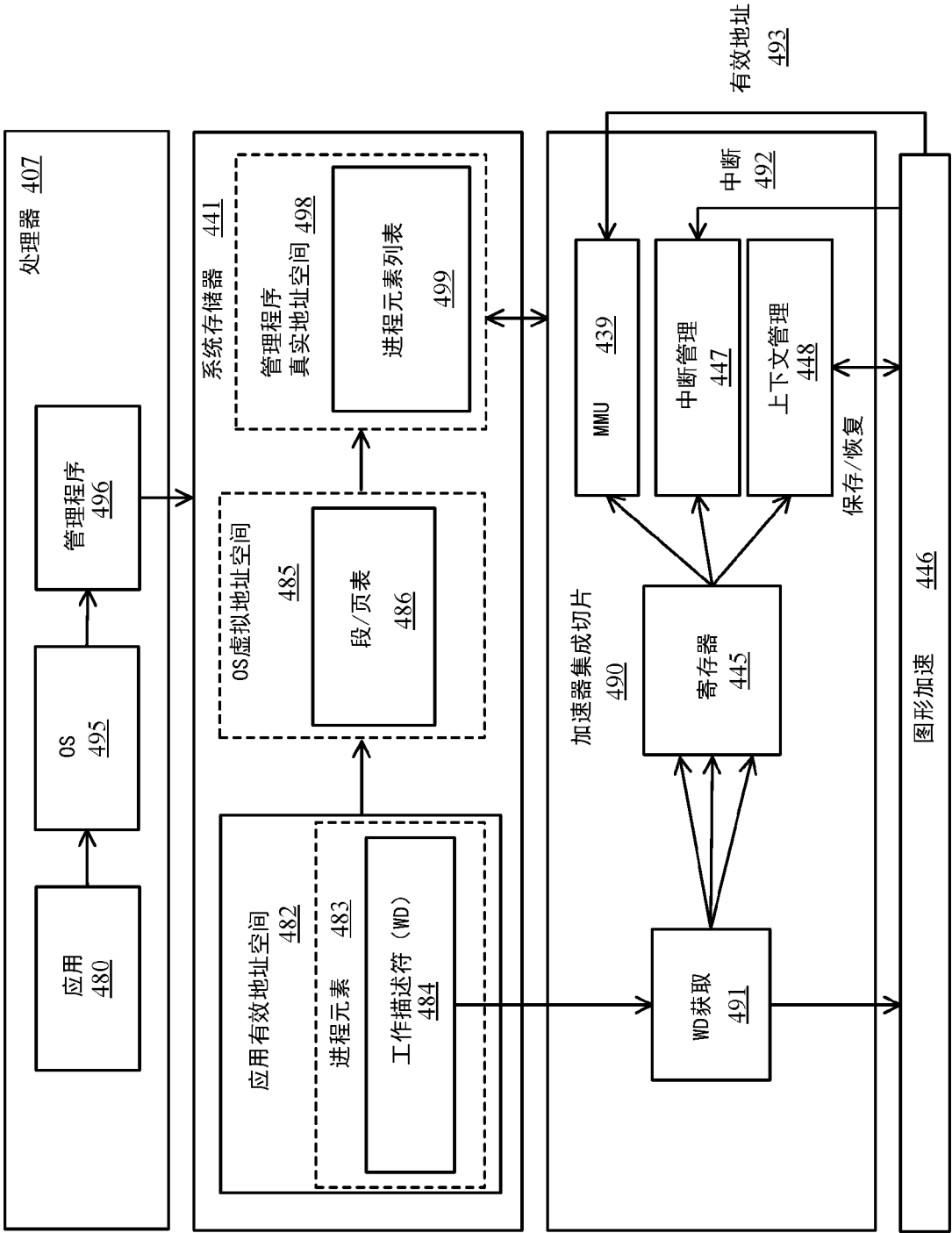


图 4E

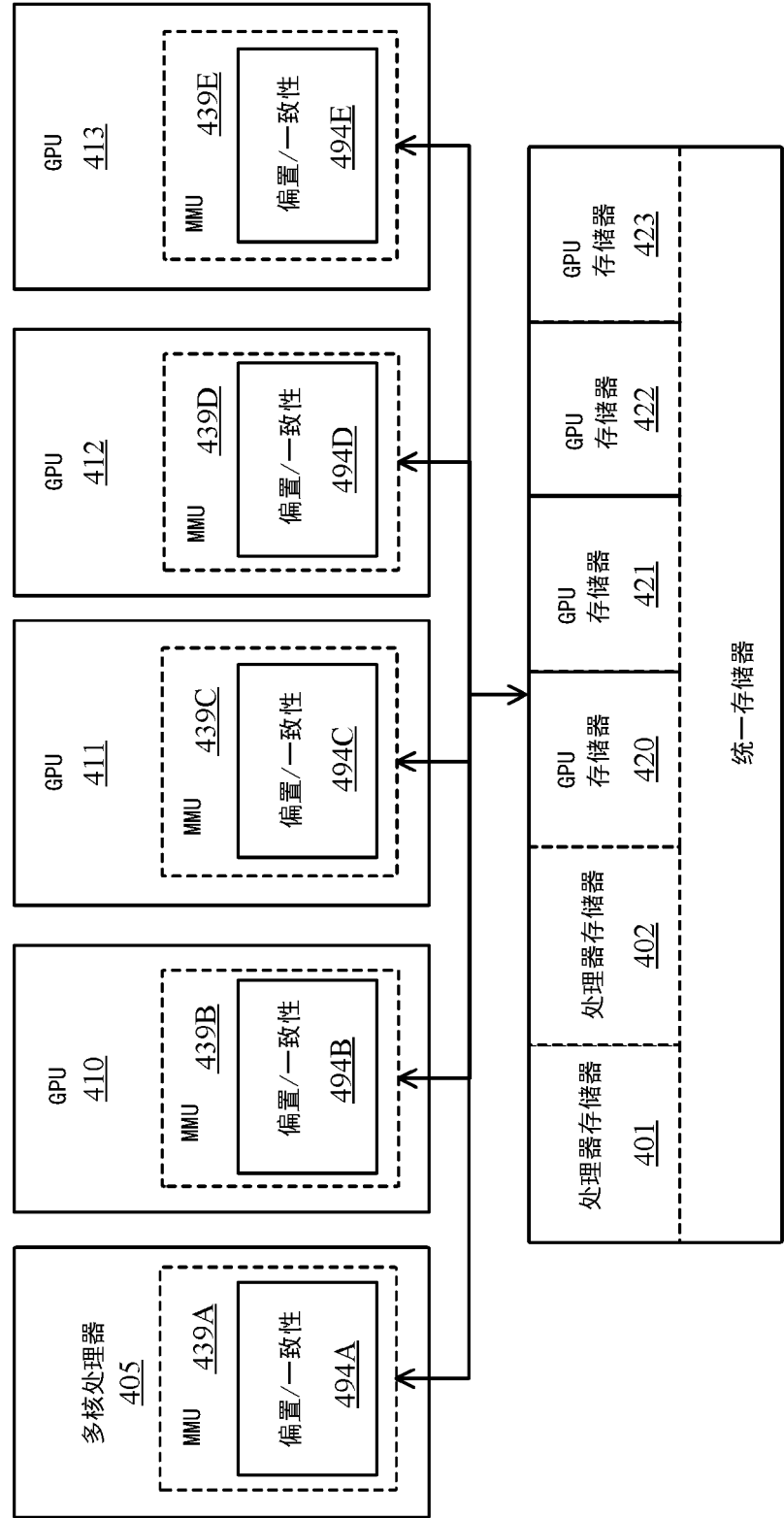


图 4F

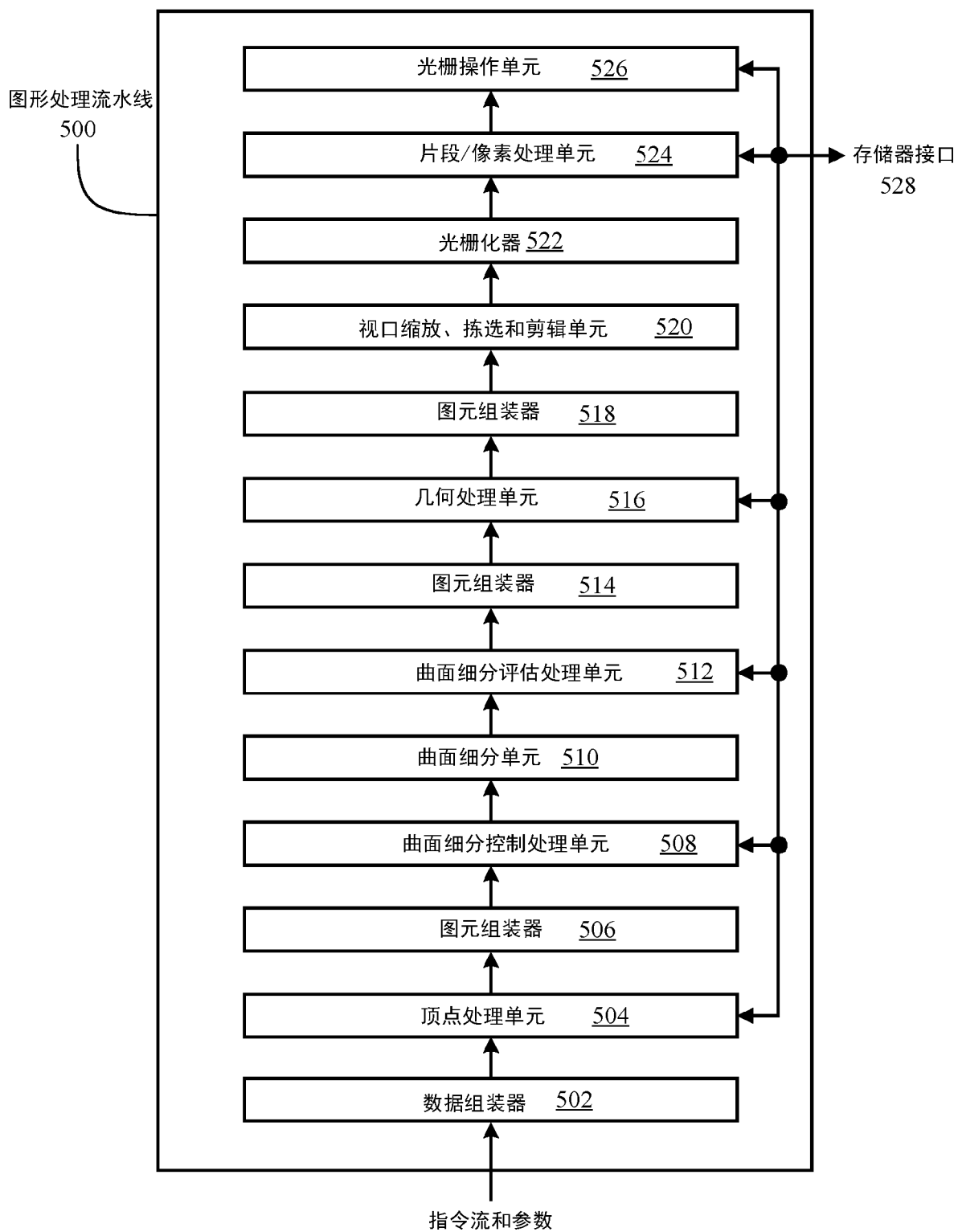


图 5

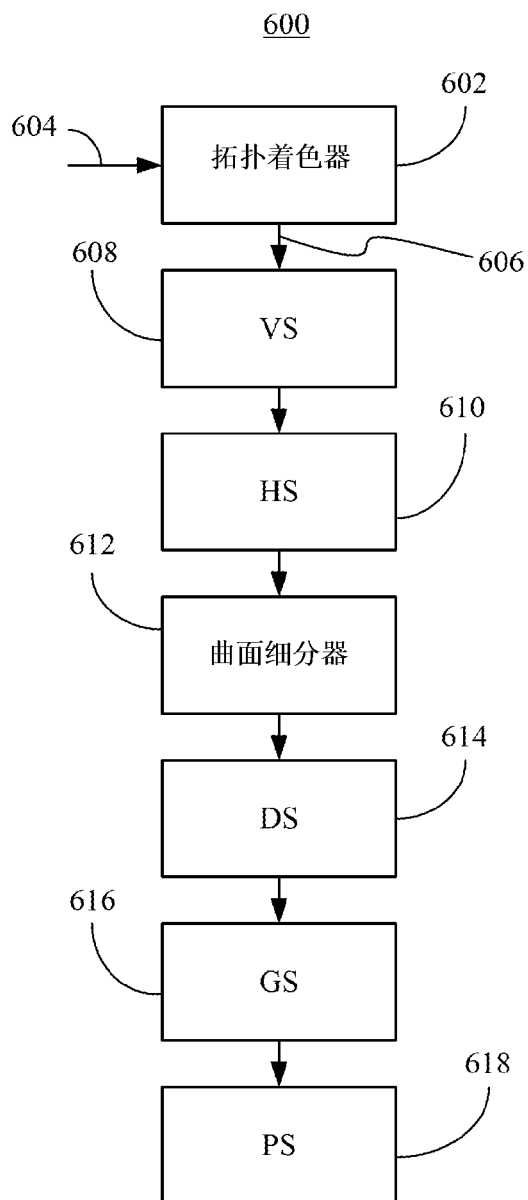


图 6

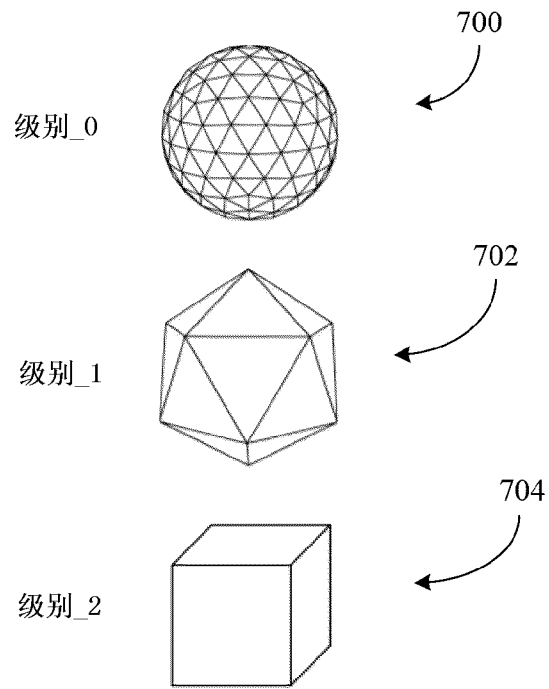


图 7

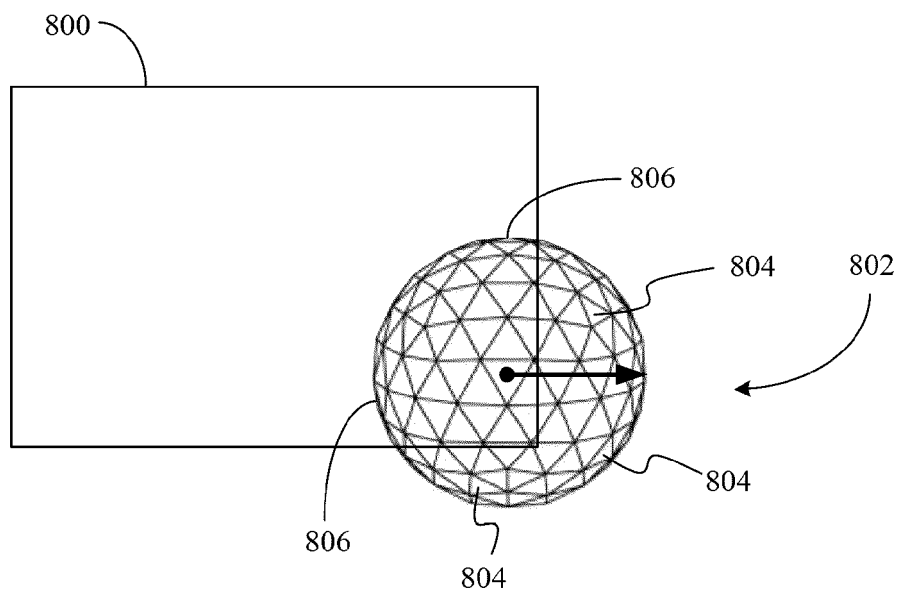


图 8

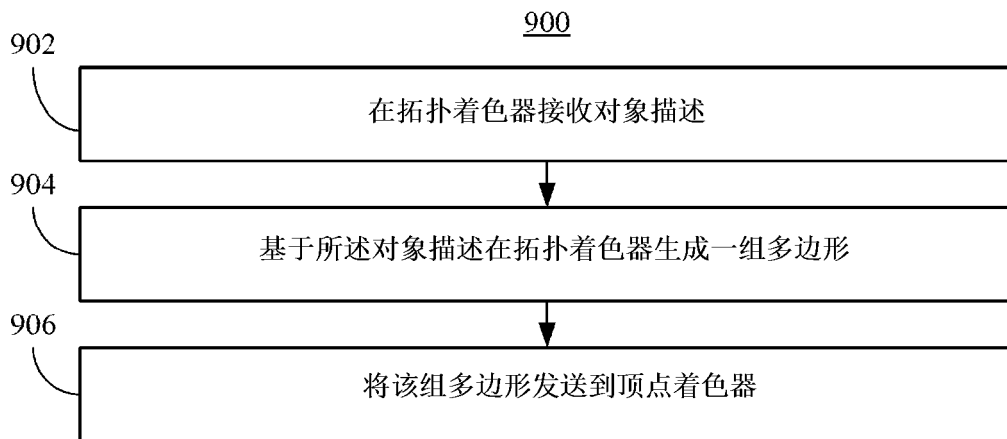


图 9A

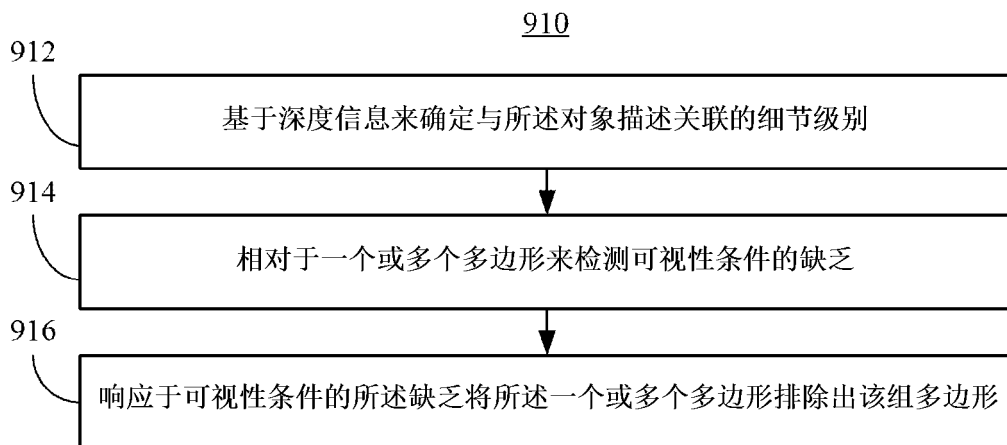


图 9B

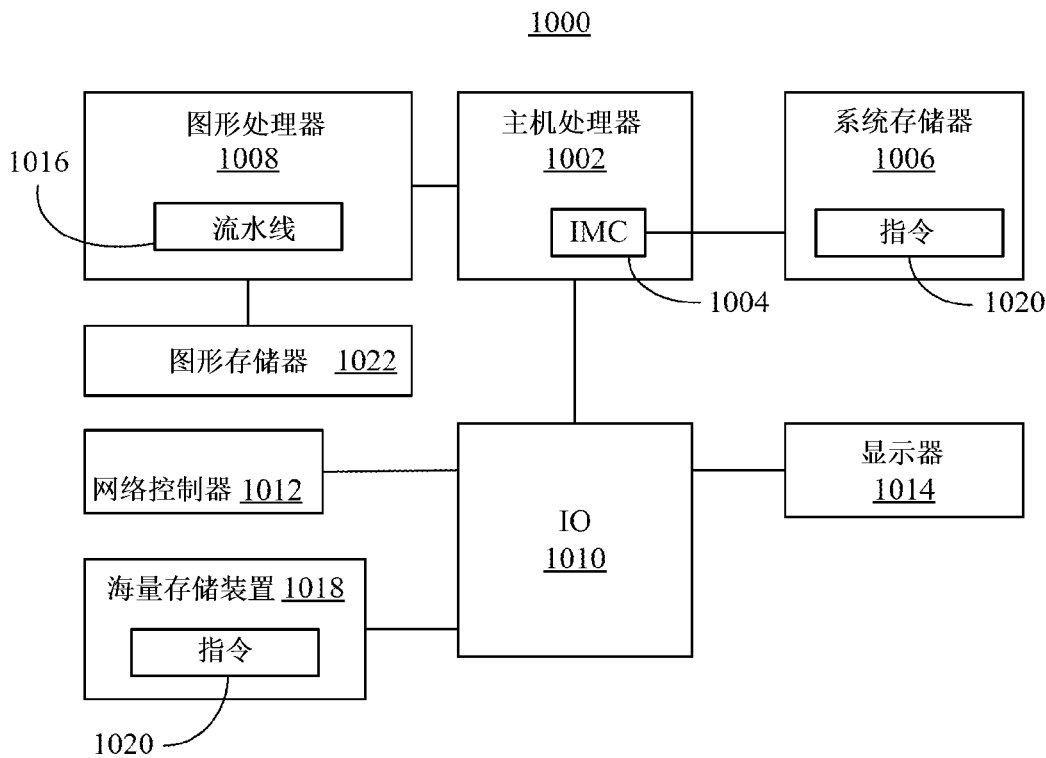


图 10A

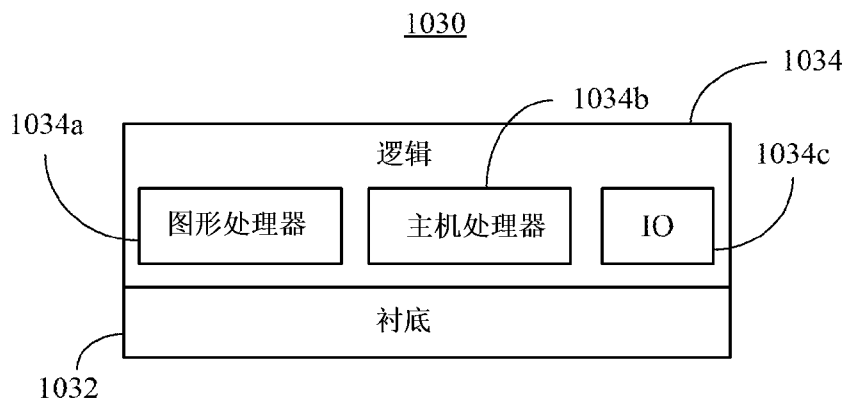


图 10B

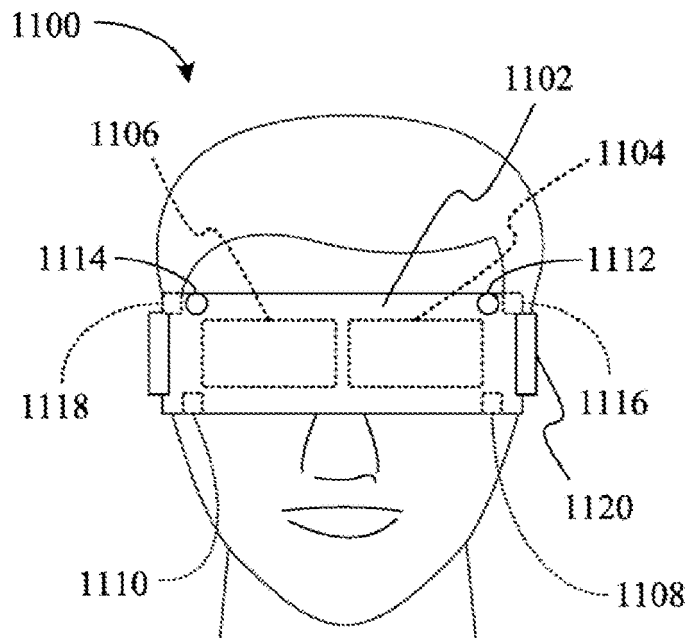


图 11

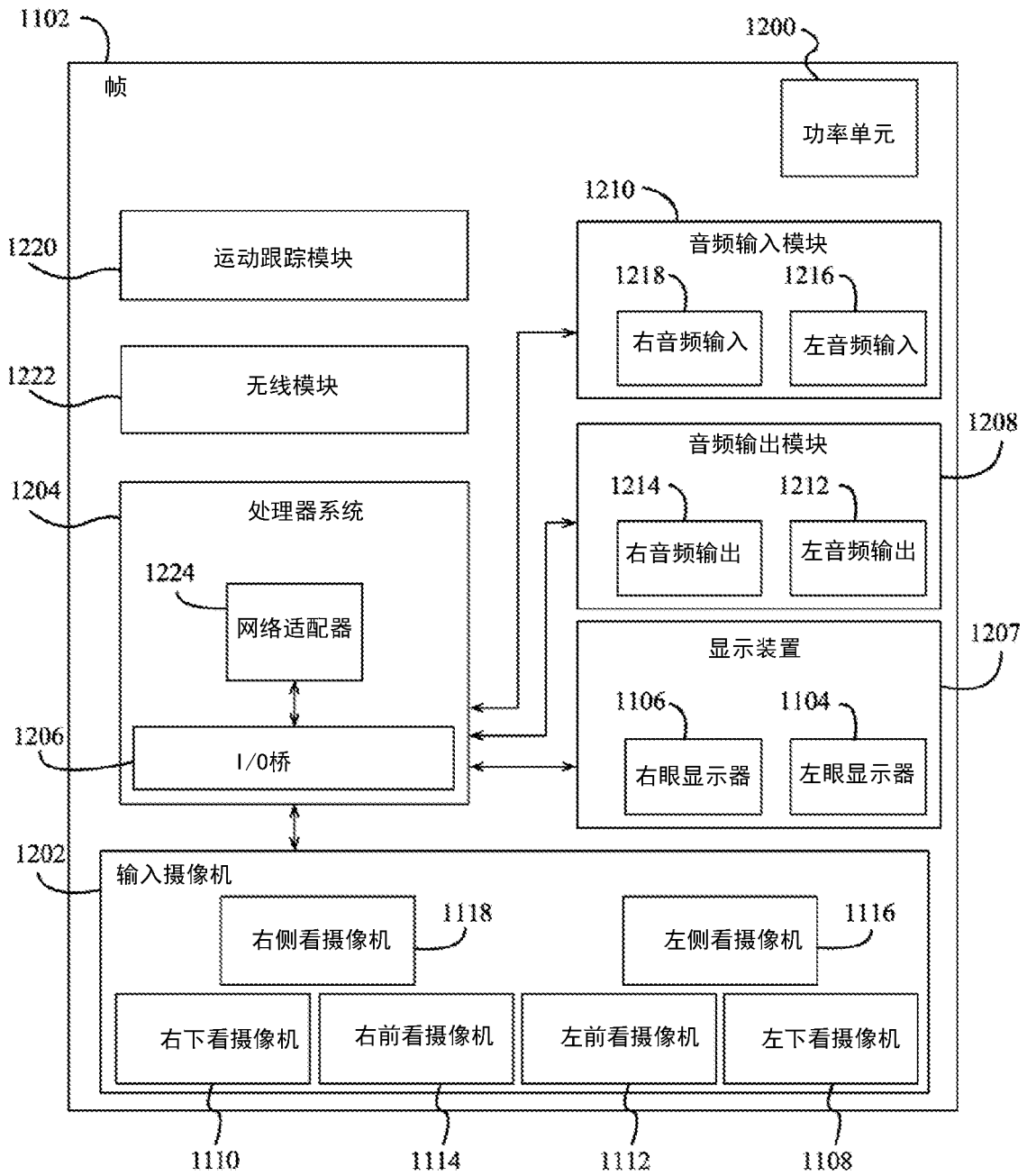
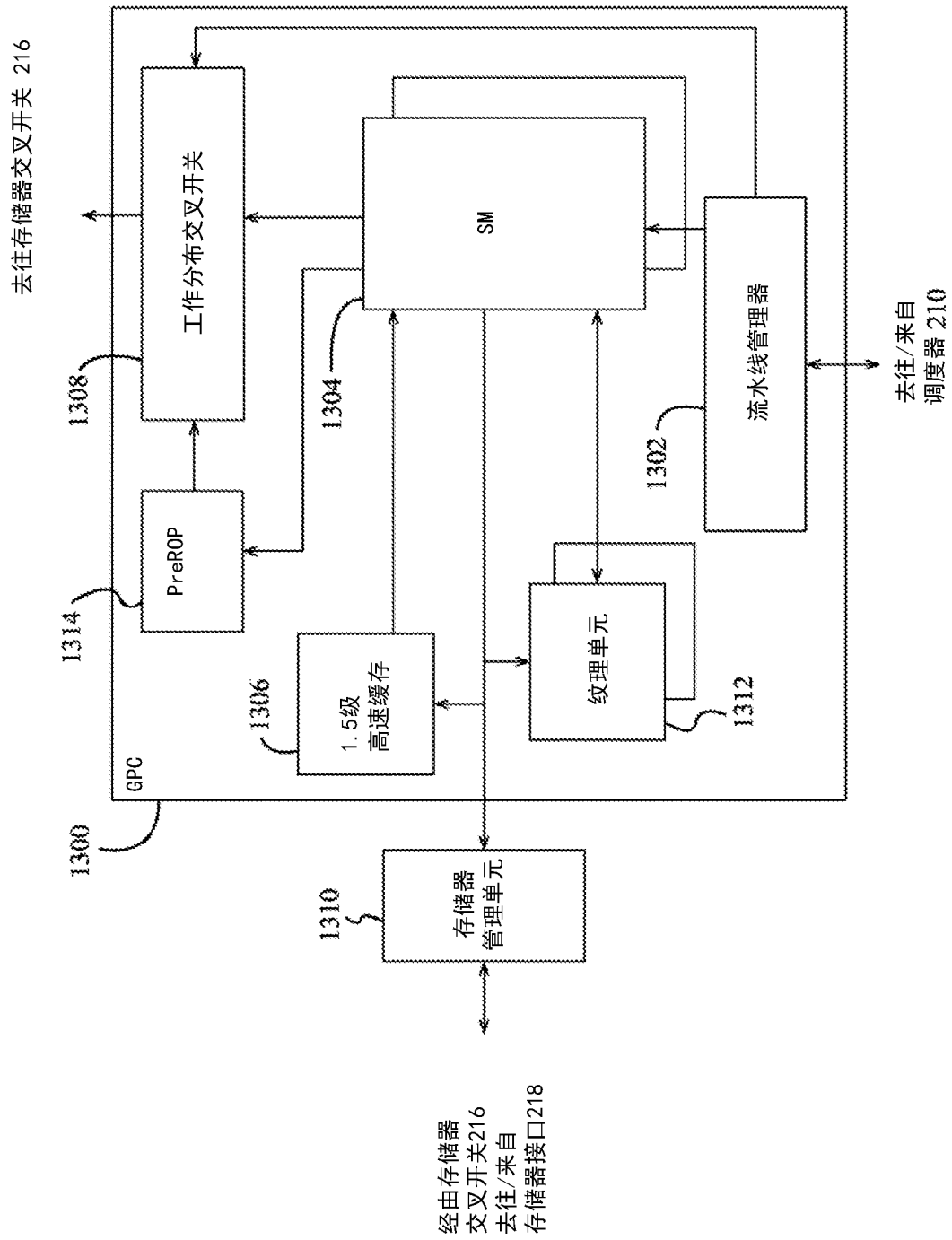


图 12



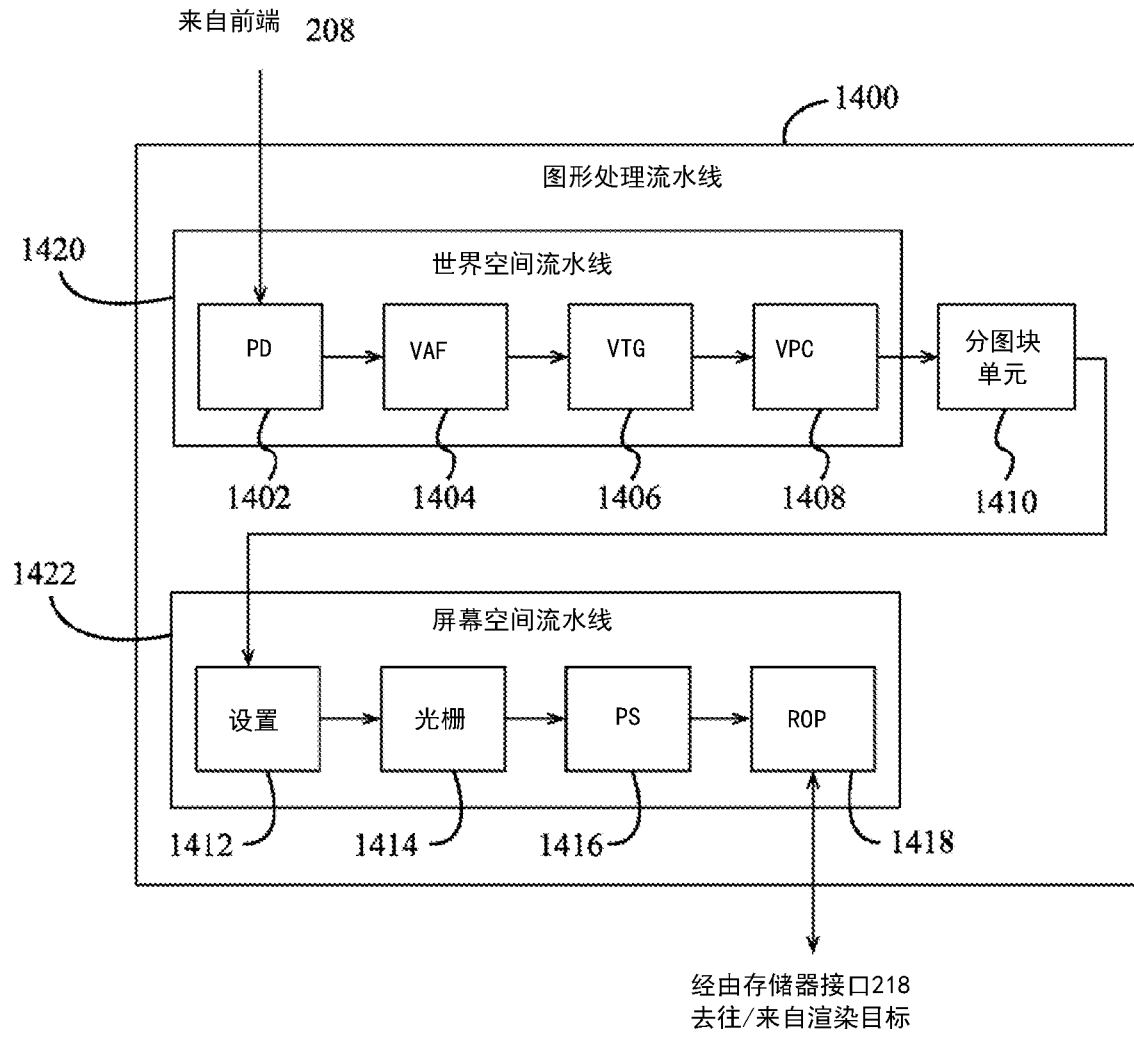


图 14

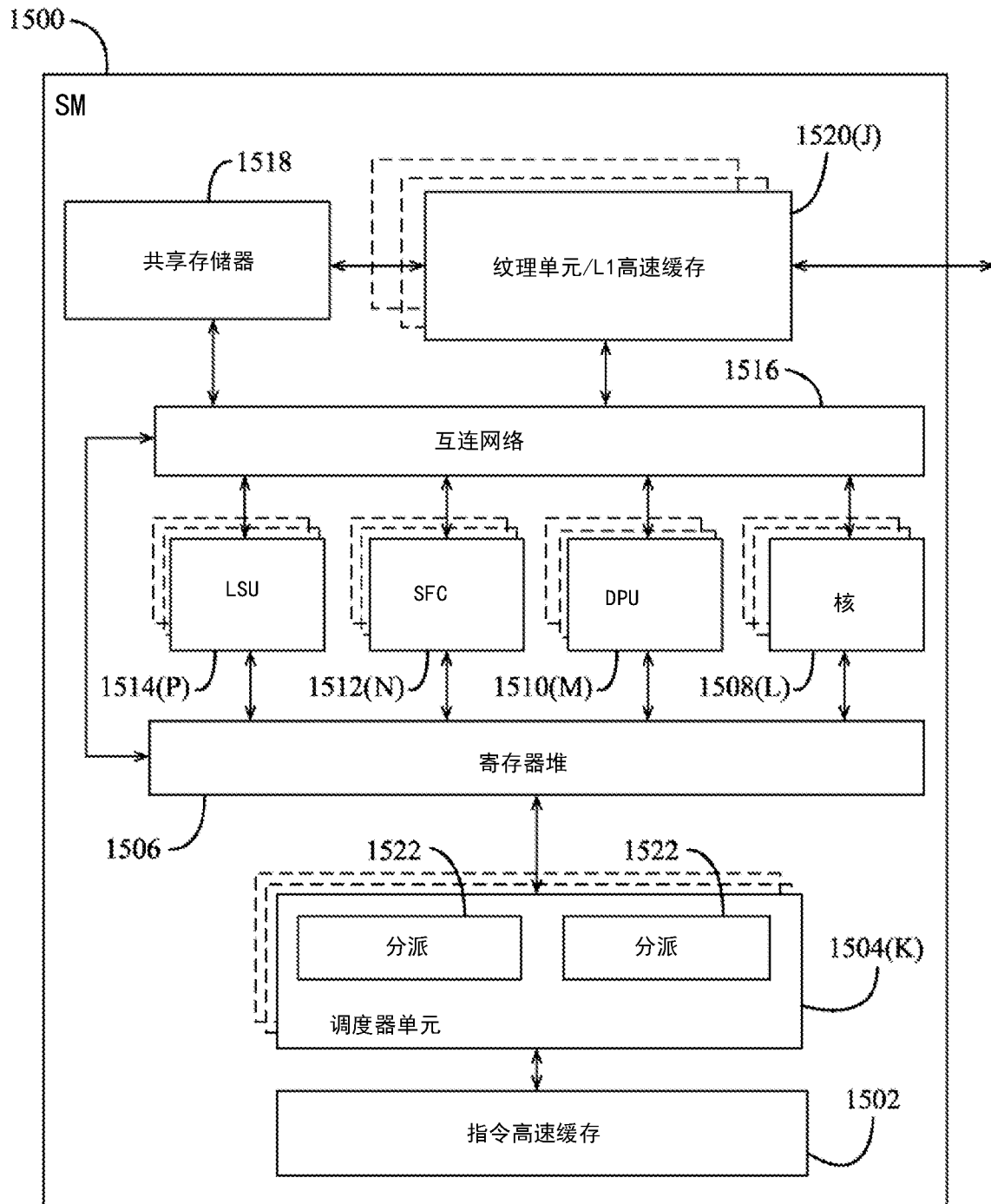


图 15

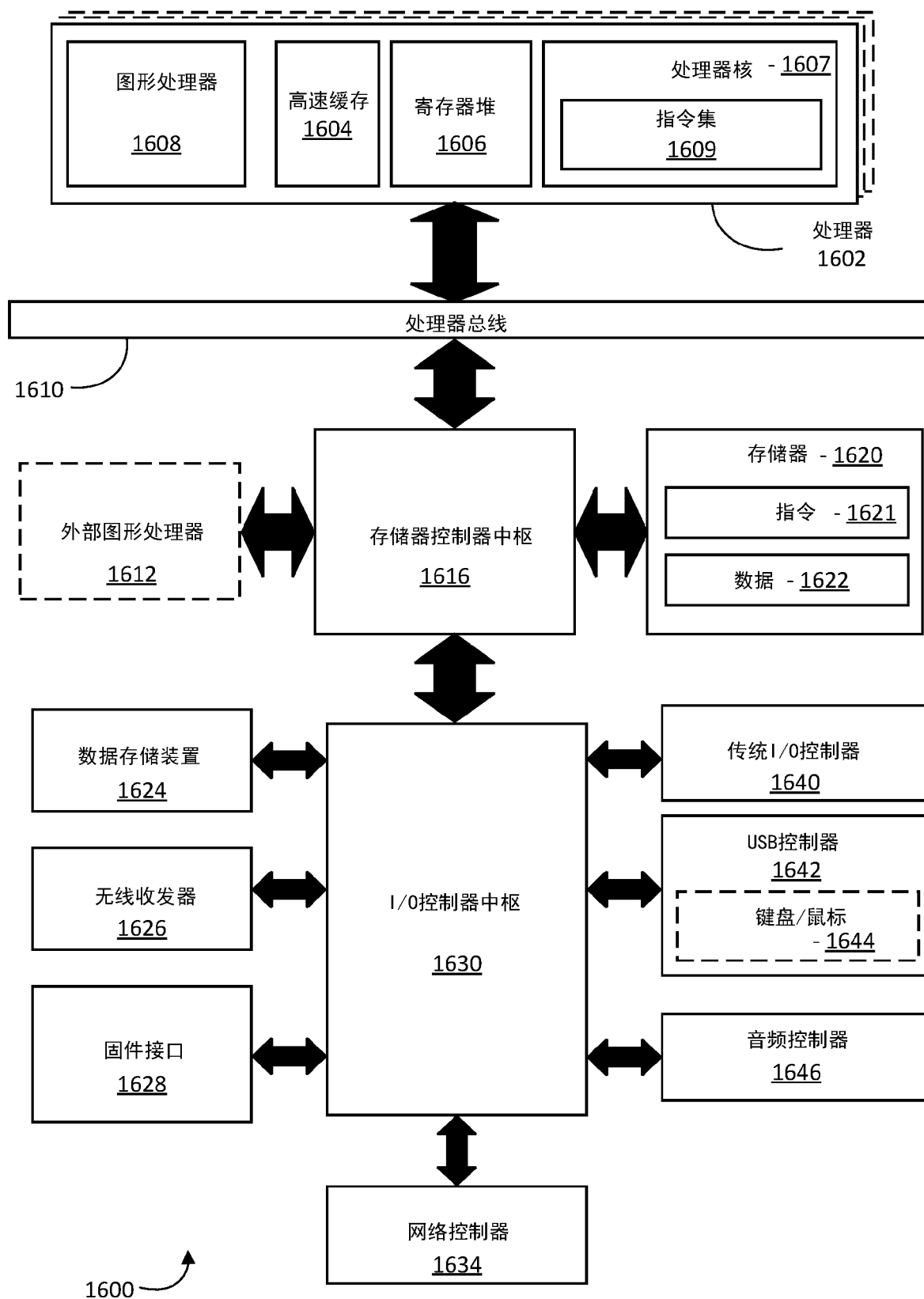


图 16

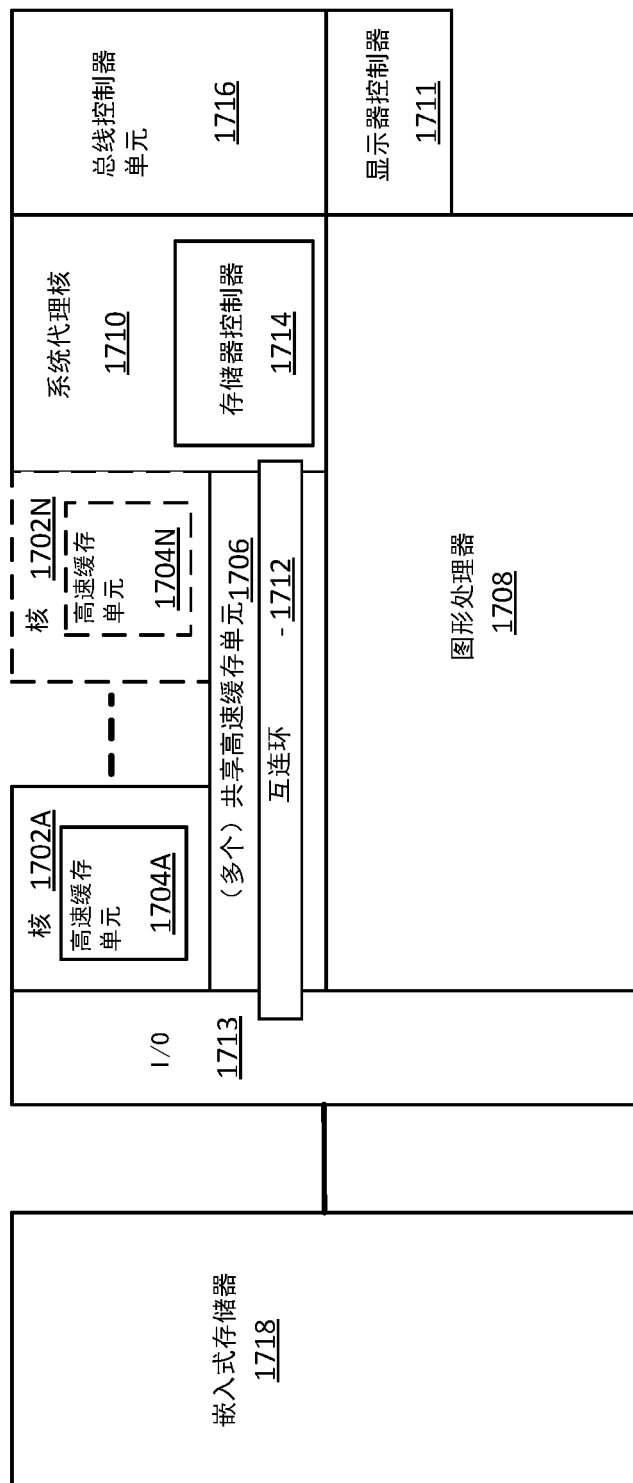


图 17

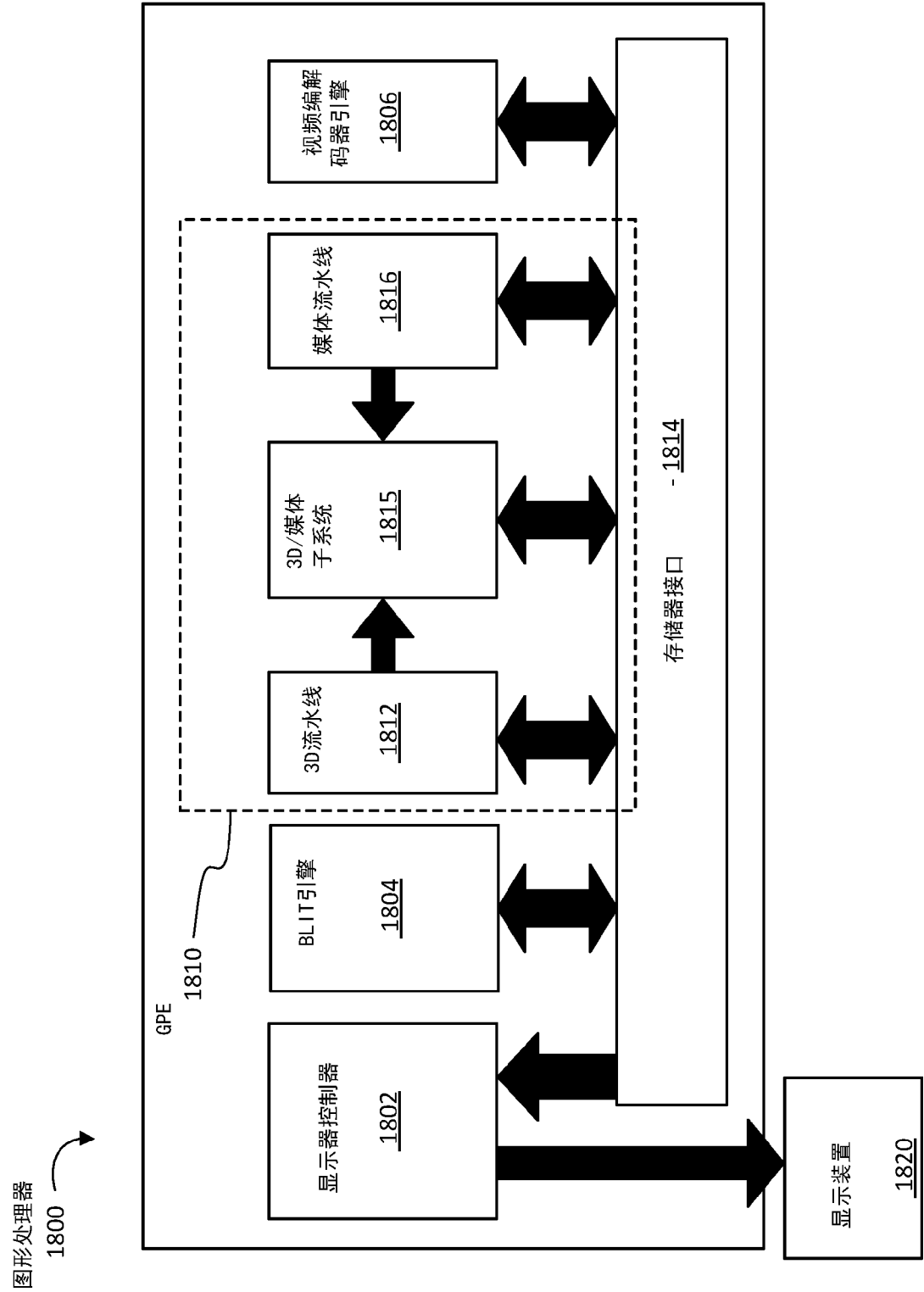


图 18

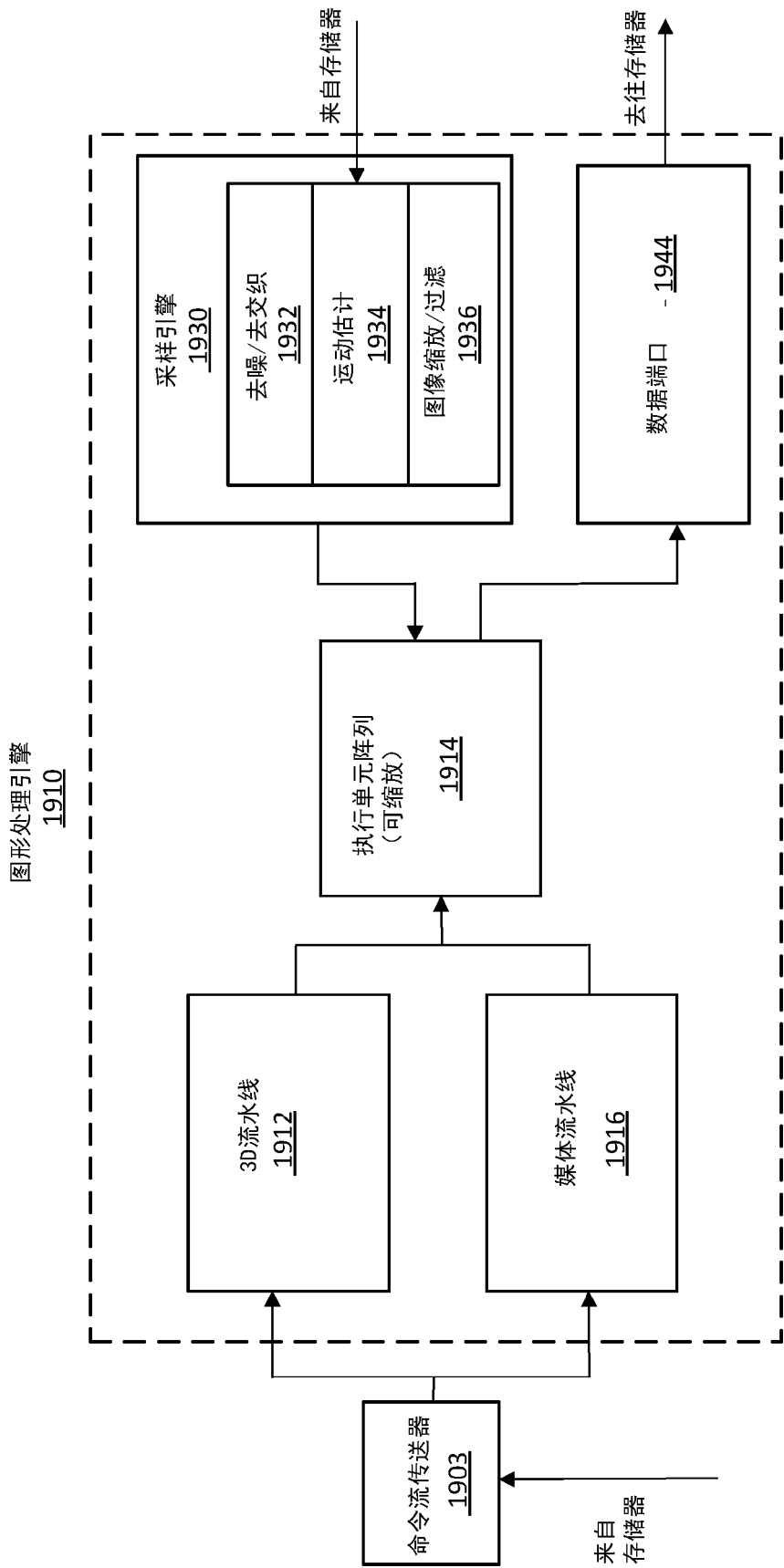


图 19

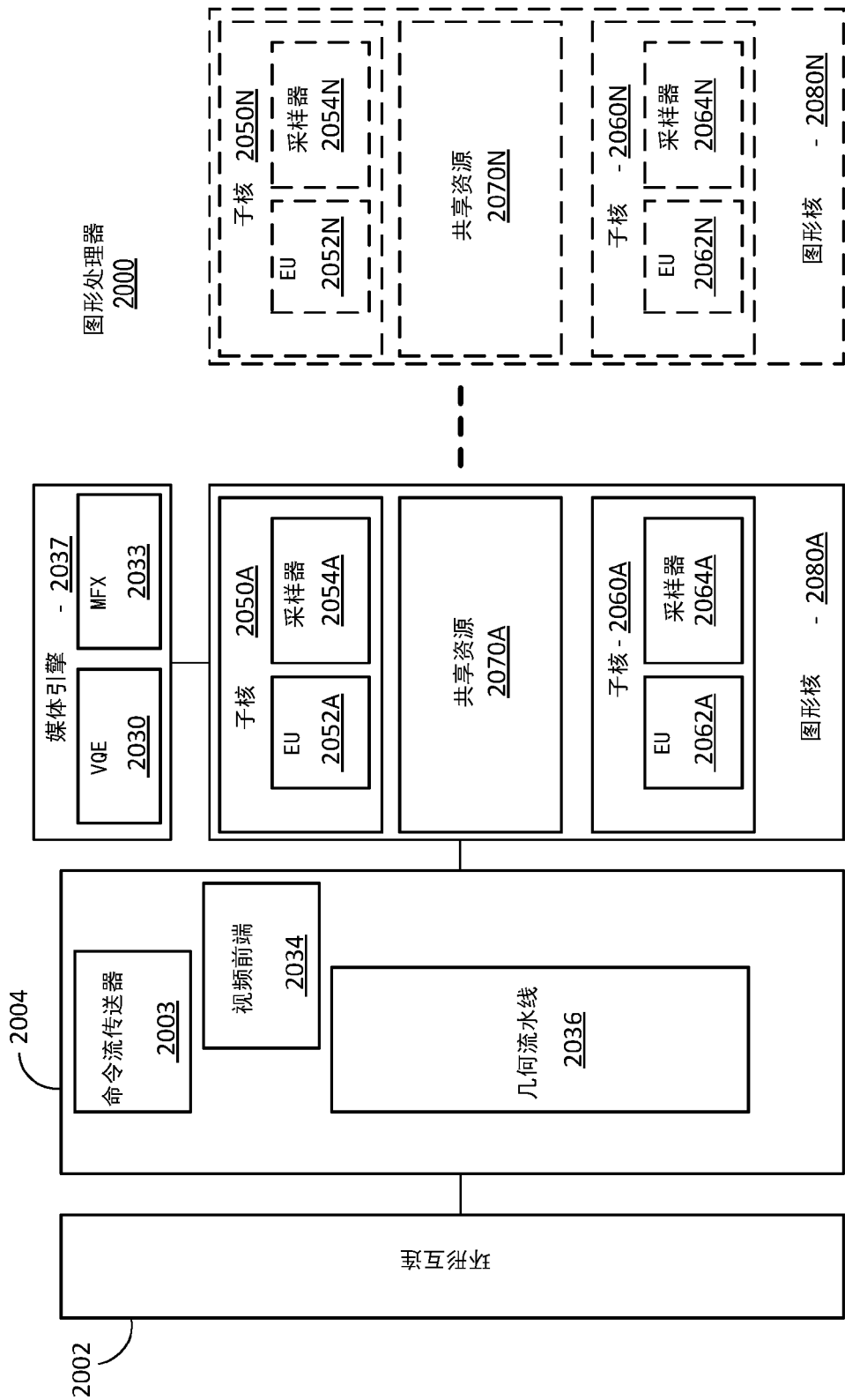


图 20

2100

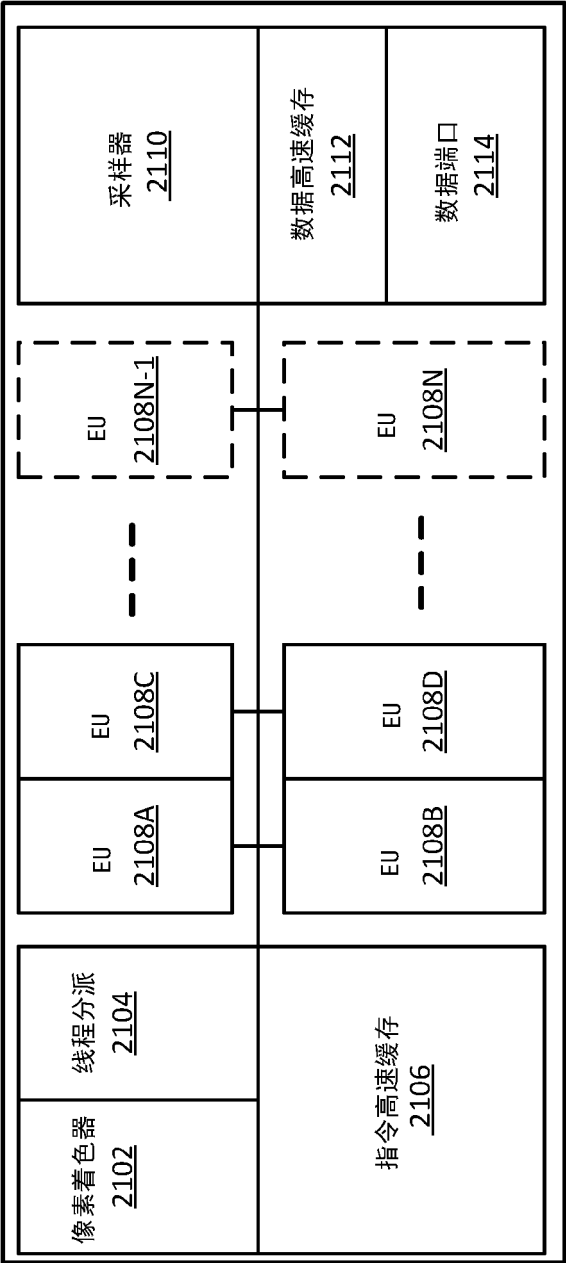


图 21

图形核指令格式

2200

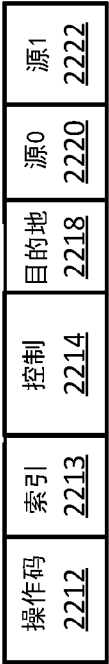
128-位指令

2210



64-位紧凑指令

2230



操作码解码

2240

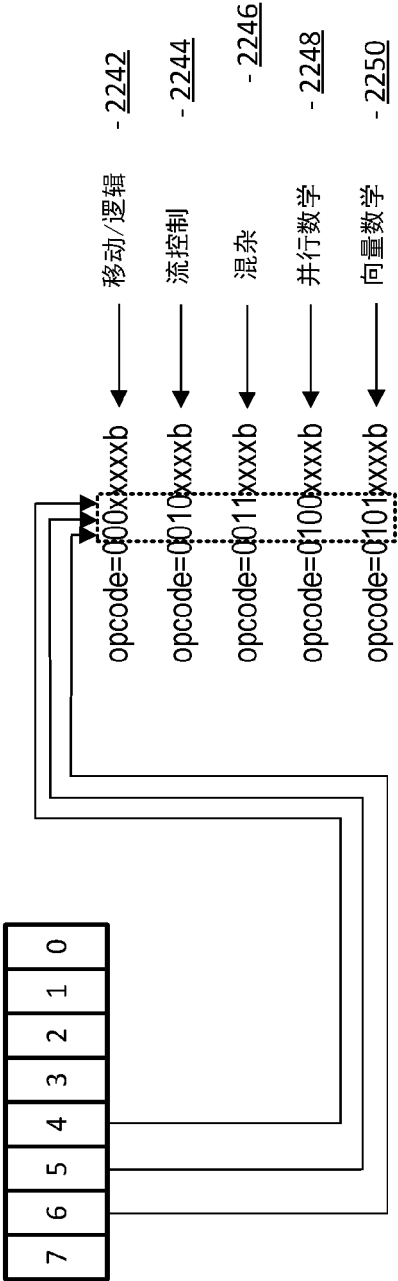


图 22

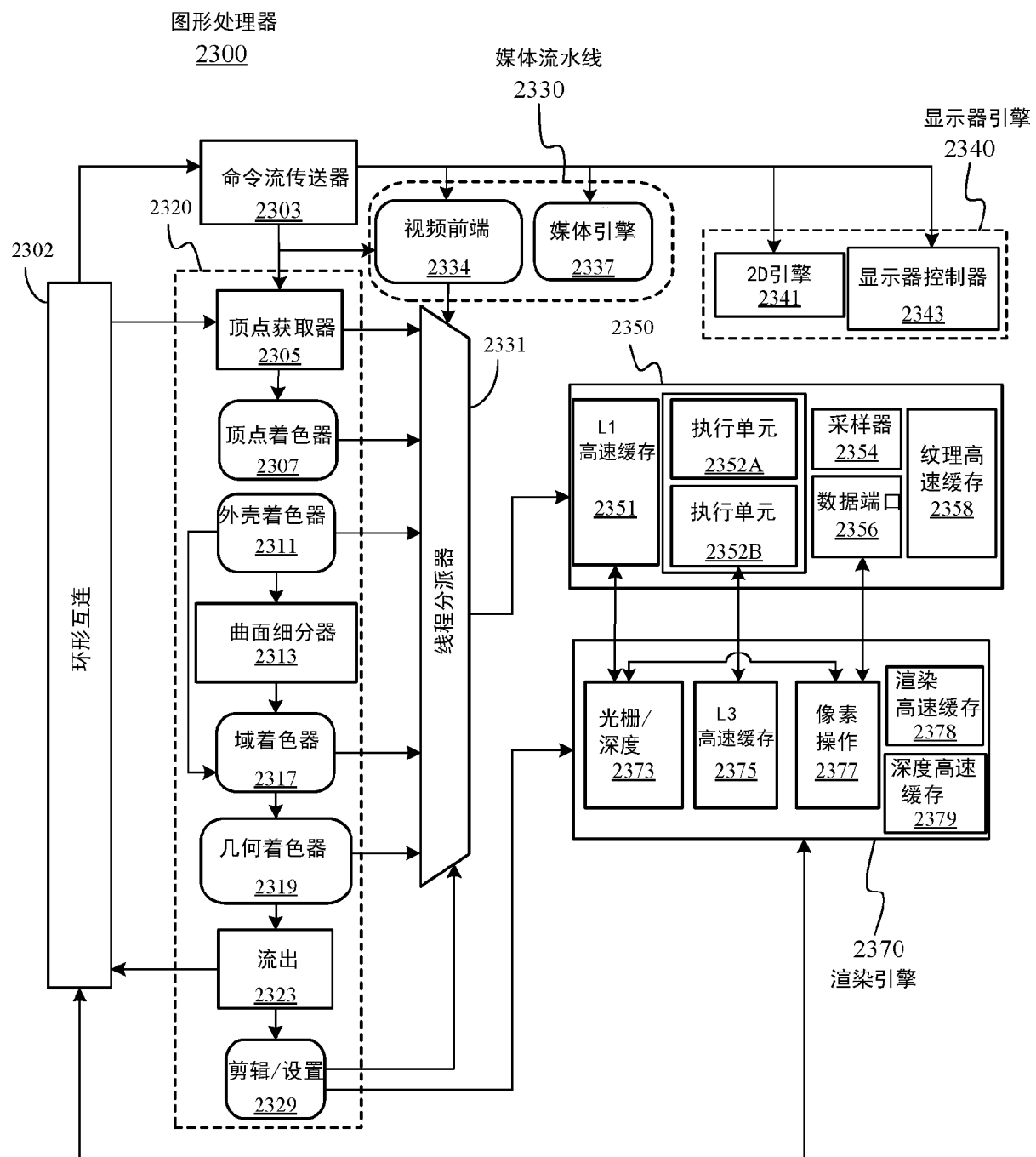


图 23

图形处理器命令格式
2400

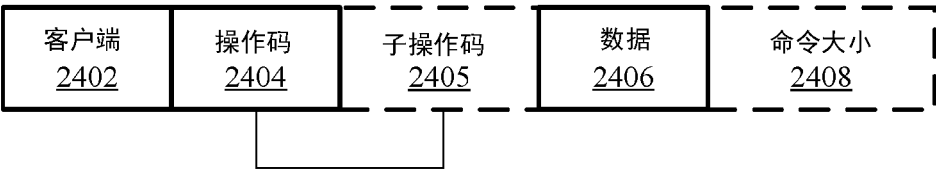


图 24A

图形处理器命令序列
2410

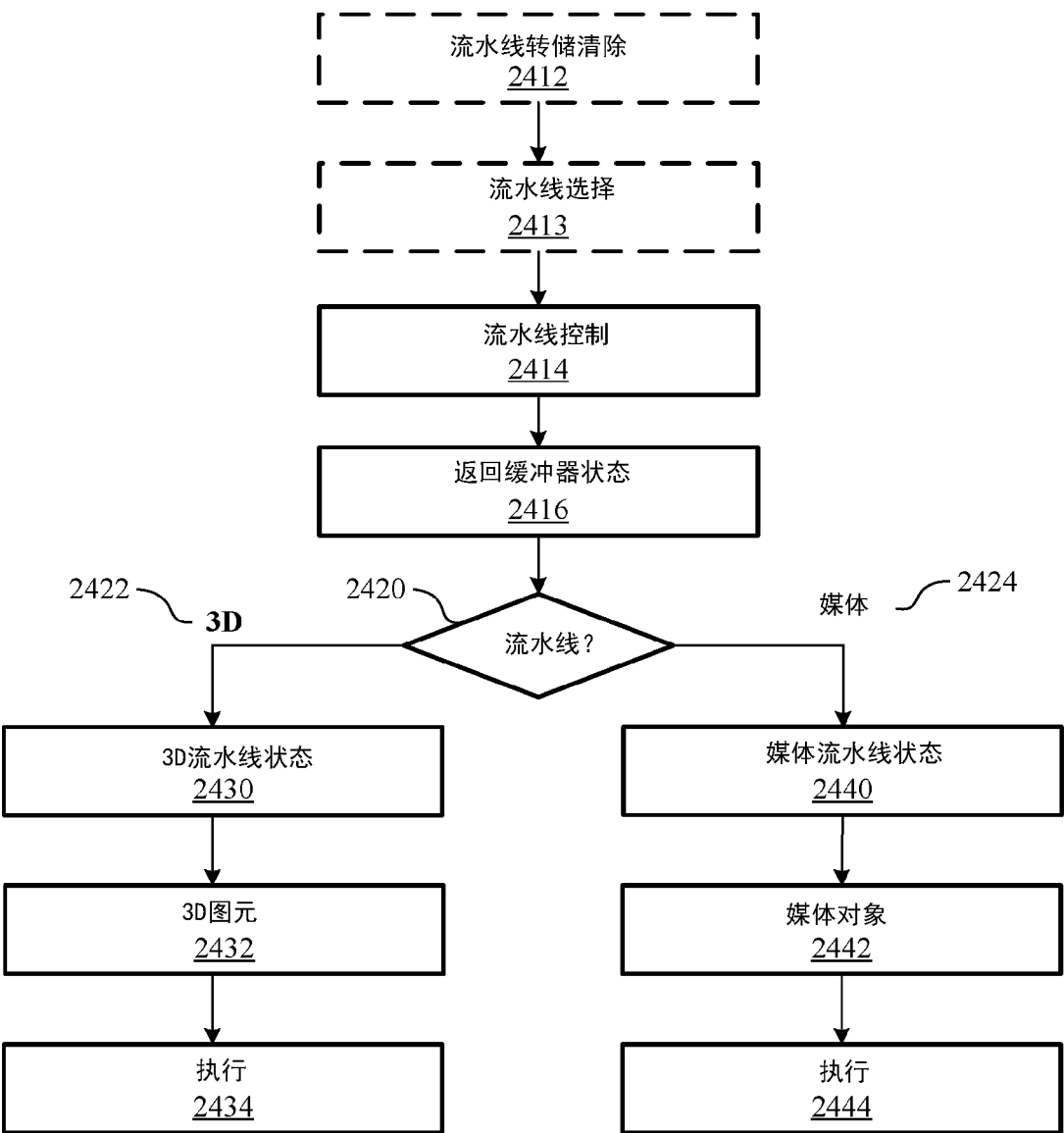


图 24B

数据处理系统 2500

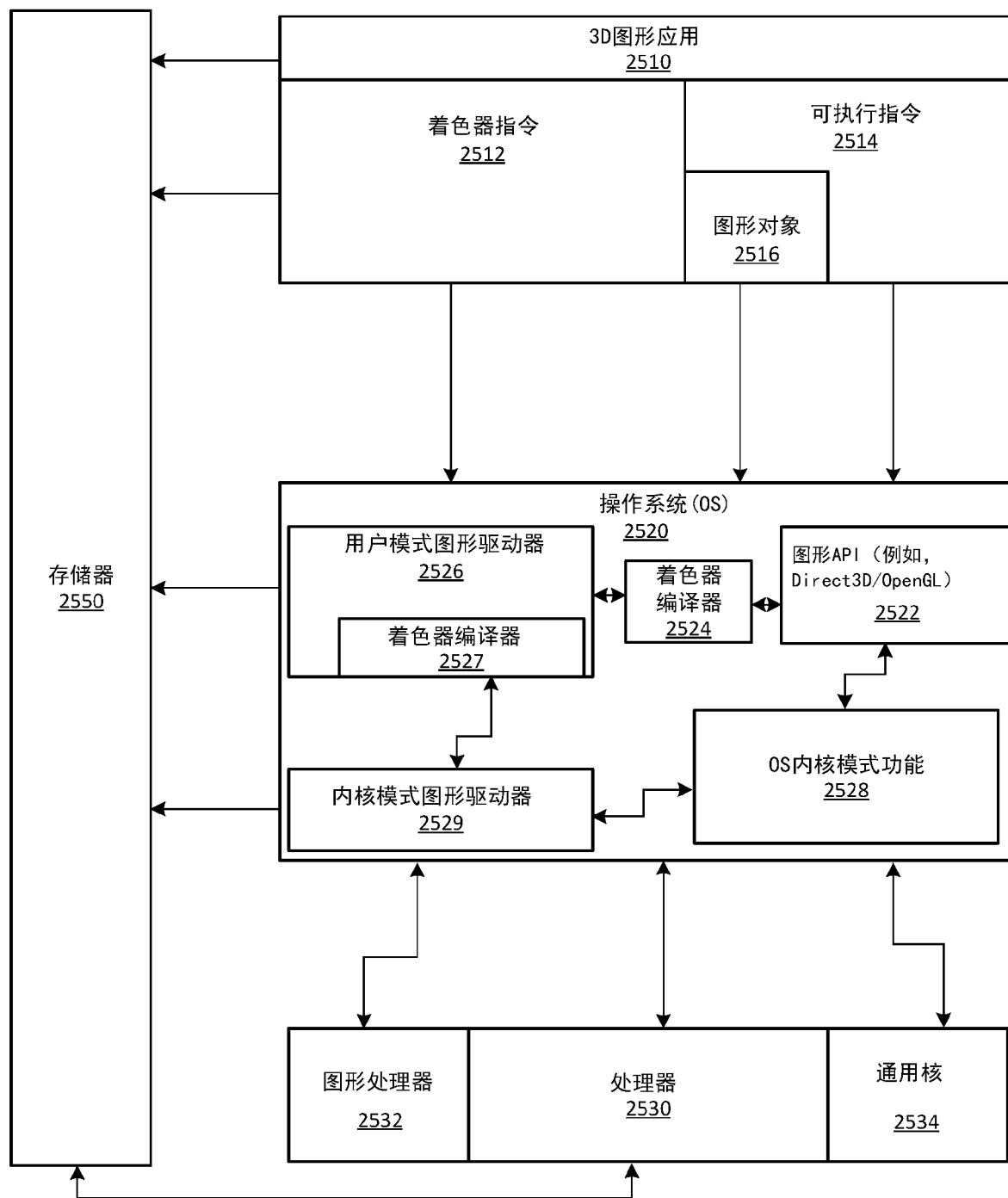


图 25

IP核开发 - 2600

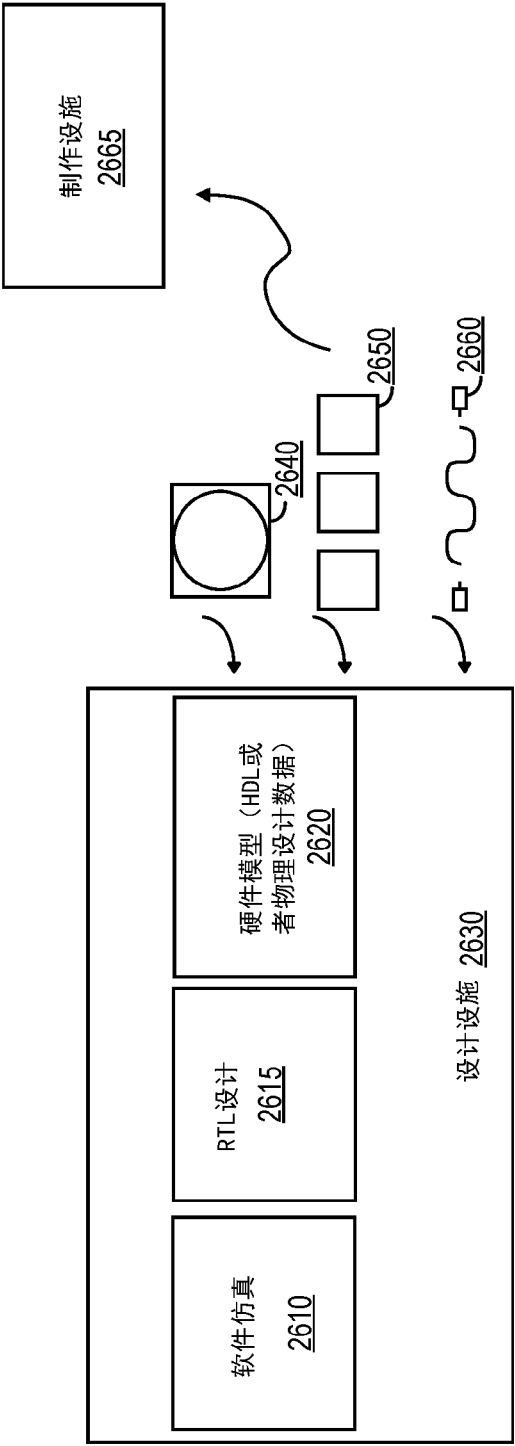


图 26

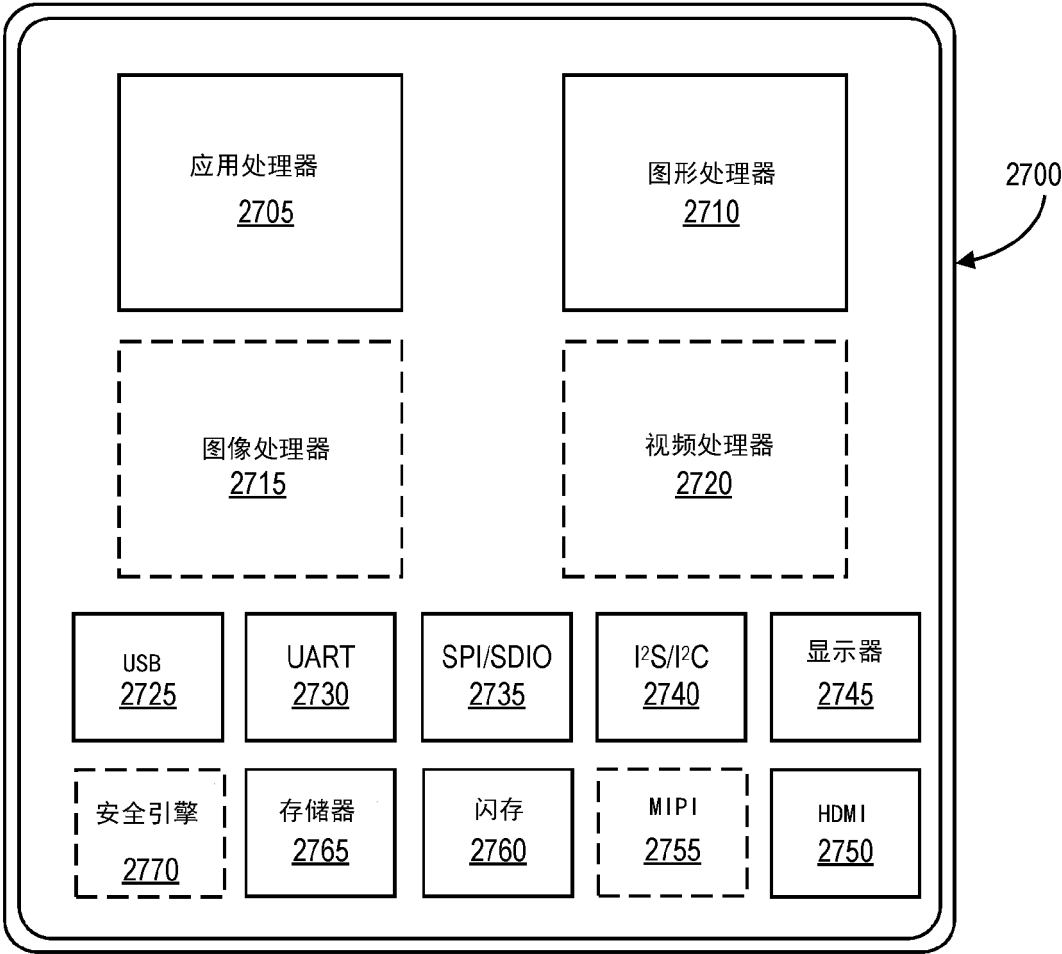


图 27