(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 112785486 A
(43)申请公布日 2021.05.11

(21)申请号 201911081492.0

(22)申请日 2019.11.07

(71)申请人 英特尔公司
地址 美国加利福尼亚州

(72)发明人 姚安邦 陆鸣 王一凯 陈晓明 黄俊杰 吕涛 罗元轲 杨毅 陈峰 王志明 郑治桥 王山东

(74)专利代理机构 上海专利商标事务所有限公司 31100
代理人 何焜 黄嵩泉

(51)Int.Cl.
G06T 1/40(2006.01)
G06F 9/38(2006.01)
G06F 9/50(2006.01)
G06T 5/00(2006.01)

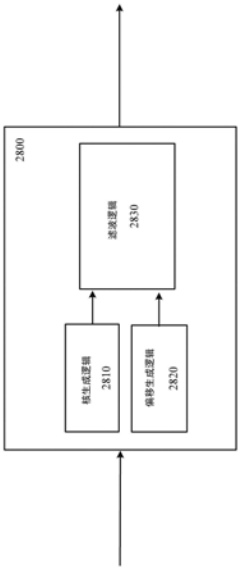
权利要求书2页 说明书60页 附图58页

(54)发明名称

用于图像去噪声的自适应可变形核预测网络

(57)摘要

本申请公开了用于图像去噪声的自适应可变形核预测网络。实施例一般地涉及用于图像去噪声的自适应可变形核预测网络。通过在计算引擎上实现的卷积神经网络对图像去噪声的方法的实施例，该图像包括多个像素，该方法包括：对于该图像的多个像素中的每一个，执行以下操作：对于该像素生成具有多个核值的卷积核；对于该像素生成多个偏移，多个偏移分别对应于多个核值，多个偏移中的每一个用于指示从该像素的像素位置的偏离量；基于该像素的像素位置和多个偏移来确定多个偏离的像素位置；以及利用该卷积核和多个偏离的像素位置的像素值来对该像素滤波，以获得去噪声的像素。



CN 112785486 A

1.一种通过在计算引擎上实现的卷积神经网络对图像去噪声的方法,所述图像包括多个像素,所述方法包括:

对于所述图像的所述多个像素中的每一个,

对于所述像素生成具有多个核值的卷积核;

对于所述像素生成多个偏移,所述多个偏移分别对应于多个核值,所述多个偏移中的每一个用于指示从所述像素的像素位置的偏离量;

基于所述像素的像素位置和所述多个偏移来确定多个偏离的像素位置;以及

利用所述卷积核和所述多个偏离的像素位置的像素值来对所述像素滤波,以获得去噪声的像素。

2.如权利要求1所述的方法,其特征在于,所述多个偏移中的每一个包括位置值以用于指示从所述像素的像素位置的偏离量。

3.如权利要求2所述的方法,其特征在于,所述位置值包括浮点值。

4.如权利要求1所述的方法,其特征在于,对于所述图像的至少两个像素的所述多个核值是不同的。

5.如权利要求1所述的方法,其特征在于,对于所述图像的至少两个像素的所述多个偏移是不同的。

6.如权利要求1所述的方法,其特征在于,所述多个偏移是在所述卷积核生成之前或与所述卷积核生成同时生成的。

7.如权利要求1所述的方法,其特征在于,利用所述卷积核和所述多个偏离的像素位置的像素值来对所述像素滤波包括:

将所述卷积核的多个核值应用于所述多个偏离的像素位置的像素值,以获得所述像素值的加权平均值。

8.如权利要求1所述的方法,其特征在于,所述偏离量的上限是预定义的。

9.如权利要求1所述的方法,其特征在于,所述卷积核的核尺寸被预分配为3×3、5×5、7×7、……和(2n-1)×(2n-1)中的一个,其中n是正整数,n>=2且n<min(floor(W/2),floor(H/2)),W和H是所述图像的宽度和高度。

10.一种用于对图像去噪声的装置,包括:

数据存储,用于存储包括图像的数据,所述图像包括多个像素;以及

计算引擎,耦合至所述数据存储,所述计算引擎用于通过卷积神经网络对所述图像去噪声,所述计算引擎用于:

对于所述图像的所述多个像素中的每一个,

对于所述像素生成具有多个核值的卷积核;

对于所述像素生成多个偏移,所述多个偏移分别对应于所述多个核值,所述多个偏移中的每一个用于指示从所述像素的像素位置的偏离量;

基于所述像素的像素位置和所述多个偏移来确定多个偏离的像素位置;以及

利用所述卷积核和所述多个偏离的像素位置的像素值来对所述像素滤波,以获得去噪声的像素。

11.如权利要求10所述的装置,其特征在于,所述多个偏移中的每一个包括位置值以用于指示从所述像素的像素位置的偏离量。

12.如权利要求11所述的装置,其特征在于,所述位置值包括浮点值。

13.如权利要求10所述的装置,其特征在于,对于所述图像的至少两个像素的所述多个核值是不同的。

14.如权利要求10所述的装置,其特征在于,对于所述图像的至少两个像素的所述多个偏移是不同的。

15.如权利要求10所述的装置,其特征在于,所述多个偏移是在所述卷积核生成之前或与所述卷积核生成同时生成的。

16.如权利要求10所述的装置,其特征在于,利用所述卷积核和所述多个偏离的像素位置的像素值来对所述像素滤波包括:

将所述卷积核的多个核值应用于所述多个偏离的像素位置的像素值,以获得所述像素值的加权平均值。

17.如权利要求10所述的装置,其特征在于,所述偏离量的上限是预定义的。

18.如权利要求10所述的装置,其特征在于,所述卷积核的核尺寸被预分配为3×3、5×5、7×7、……和(2n-1)×(2n-1)中的一个,其中n是正整数,n>=2且n<min(floor(W/2),floor(H/2)),W和H是所述图像的宽度和高度。

19.一种设备,包括用于执行如权利要求1-9中任一项所述的方法的装置。

20.一种机器可读介质,包括多条指令,所述多条指令响应于在计算设备上被执行而导致所述计算设备执行如权利要求1-9中的任一项所述的方法。

用于图像去噪声的自适应可变形核预测网络

技术领域

[0001] 实施例总体上涉及数据处理,并且更具体地涉及经由通用图形处理单元的数据处理。

背景技术

[0002] 当前的并行图形数据处理包括被开发为用于对图形数据执行特定操作的系统和方法,这些特定操作诸如例如,线性插值、曲面细分、栅格化、纹理映射、深度测试等。传统上,图形处理器使用固定功能计算单元来处理图形数据;然而,最近,图形处理器的多个部分已经变得可编程,使得此类处理器能够支持用于处理顶点和片段数据的更广泛种类的操作。

[0003] 为了进一步提高性能,图形处理器典型地实现诸如流水线操作之类的处理技术,这些处理技术尝试贯穿图形流水线的不同部分并行地处理尽可能多的图形数据。具有单指令多线程(SIMT)架构的并行图形处理器被设计成使图形流水线中的并行处理的量最大化。在SIMT架构中,成组的并行线程尝试尽可能频繁地一起同步地执行程序指令,以提高处理效率。可以在Shane Cook的CUDA编程(CUDA Programming)第三章,第37-51页(2013年)中找到对SIMT架构的软件和硬件的总体概述。

附图说明

[0004] 为了以能够详细理解本实施例的以上记载特征的方式,可通过参考实施例来对以上简要概括的实施例进行更具体的描述,这些实施例中的一些在所附附图中被图示。然而,应当注意,所附附图仅图示出典型实施例,并且因此不应被认为是对其范围的限制。

[0005] 图1是图示配置成用于实现本文中所描述的实施例的一个或多个方面的计算机系统的框图;

[0006] 图2A-图2D图示根据实施例的并行处理器组件;

[0007] 图3A-图3C是根据实施例的图形多处理器和基于多处理的GPU的框图;

[0008] 图4A-图4F图示在其中多个GPU通信地耦合至多个多核处理器的示例性架构;

[0009] 图5图示根据实施例的图形处理流水线;

[0010] 图6图示根据实施例的机器学习软件栈;

[0011] 图7图示根据实施例的通用图形处理单元;

[0012] 图8图示根据实施例的多GPU计算系统;

[0013] 图9A-图9B图示示例性深度神经网络的层;

[0014] 图10图示示例性递归神经网络;

[0015] 图11图示深度神经网络的训练和部署;

[0016] 图12是图示分布式学习的框图;

[0017] 图13图示适用于使用经训练的模型执行推断的示例性推断芯片上系统(SOC);

[0018] 图14是根据实施例的处理系统的框图;

[0019] 图15A-图15C图示由本文中描述的实施例提供的计算系统和图形处理器；

[0020] 图16A-图16C图示由本文中描述的实施例提供的附加的图形处理器和计算加速器架构的框图；

[0021] 图17是根据一些实施例的图形处理器的图形处理引擎的框图；

[0022] 图18A-图18B图示根据本文中描述的实施例的包括在图形处理器核中采用的处理元件阵列的线程执行逻辑；

[0023] 图19图示根据实施例的附加的执行单元；

[0024] 图20是图示根据一些实施例的图形处理器指令格式的框图；

[0025] 图21是根据另一个实施例的图形处理器的框图；

[0026] 图22A-图22B图示根据一些实施例的图形处理器命令格式和命令序列；

[0027] 图23图示根据一些实施例的用于数据处理系统的示例性图形软件架构；

[0028] 图24A是图示根据实施例的IP核开发系统的框图；

[0029] 图24B图示根据本文中描述的一些实施例的集成电路封装组件的横截面侧视图；

[0030] 图24C图示封装组件，该封装组件包括连接到衬底的多个单元的硬件逻辑小芯片(例如，基础管芯)；

[0031] 图24D图示根据实施例的包括可互换小芯片的封装组件；

[0032] 图25是图示根据实施例的示例性芯片上系统集成电路的框图；

[0033] 图26A-图26B是图示根据本文中所描述的实施例的用于在SoC内使用的示例性图形处理器的框图；

[0034] 图27A是示出常规的核预测网络(KPN)的示意图；

[0035] 图27B是示出通过常规的KPN对像素进行滤波的示例的示意图；

[0036] 图28A是示出根据实施例的自适应可变形核预测网络(ADKPN)的示意图；

[0037] 图28B是示出根据实施例的通过ADKPN对像素进行滤波的示例的示意图；

[0038] 图28C是示出常规的KPN与ADKPN之间的训练和/或测试性能度量的比较的曲线图；

[0039] 图29是示出根据实施例的通过ADKPN对图像进行去噪声的方法的流程图；

[0040] 图30A-30D分别示出输入图像、参考图像、通过常规的KPN产生的输出图像、以及根据实施例的通过ADKPN产生的输出图像；

[0041] 图31A-31D分别示出输入图像、参考图像、通过常规的KPN产生的输出图像、以及根据实施例的通过ADKPN产生的输出图像；

[0042] 图32A-32D分别示出输入图像、参考图像、通过常规的KPN产生的输出图像、以及根据实施例的通过ADKPN产生的输出图像；以及

[0043] 图33A-33D分别示出输入图像、参考图像、通过常规的KPN产生的输出图像、以及根据实施例的通过ADKPN产生的输出图像。

具体实施方式

[0044] 在一些实施例中，图形处理单元(GPU)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、以及各种通用GPU(GPGPU)功能。GPU可通过总线或另一互连(例如，诸如PCIe或NVLink之类的高速互连)而通信地耦合至主机处理器/核。在其他实施例中，GPU可与核集成在同一封装或芯片上，并且通过内部处理器总线/互连(即，在封装或

芯片的内部)通信地耦合至核。不论连接GPU所采用的方式如何,处理器核都可以采取工作描述符中所包含的命令/指令序列的形式将工作分配给GPU。GPU随后使用专用电路/逻辑来有效地处理这些命令/指令。

[0045] 在下列描述中,阐述了众多特定细节来提供更全面的理解。然而,对本领域技术人员将会显而易见的是,可在没有这些特定细节中的一个或多个细节的情况下实施本文中所描述的实施例。在其他实例中,未描述公知的特征以避免使当前实施例的细节变得模糊。

[0046] 系统概览

[0047] 图1是图示出被配置成用于实现本文中所描述的示例性实施例的一个或多个方面的计算系统100的框图。计算系统100包括具有一个或多个处理器102和系统存储器104的处理子系统101。这一个或多个处理器102与系统存储器104经由可包括存储器中枢105的互连路径进行通信。存储器中枢105可以是芯片组组件内的单独组件或者可被集成在一个或多个处理器102内。存储器中枢105经由通信链路106而与I/O子系统111耦合。I/O子系统111包括I/O中枢107,该I/O中枢107可以使得计算系统100能够从一个或多个输入设备108接收输入。另外,I/O中枢107可以使得显示控制器能够向一个或多个显示设备110A提供输出,该显示控制器可包括在一个或多个处理器102中。在一个实施例中,与I/O中枢107耦合的一个或多个显示设备110A可以包括本地的、内部的、或嵌入式的显示设备。

[0048] 在一个实施例中,处理子系统101包括经由总线或其他通信链路113耦合至存储器中枢105的一个或多个并行处理器112。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中的一种,诸如但不限于PCI Express,或者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,一个或多个并行处理器112形成可包括大量处理核和/或处理集群的集中于计算的并行或向量处理系统,诸如,集成众核(MIC)处理器。在一个实施例中,一个或多个并行处理器112形成可以向经由I/O中枢107耦合的一个或多个显示设备110A中的一个输出像素的图形处理子系统。一个或多个并行处理器112还可以包括显示控制器以及用于实现到一个或多个显示设备110B的直接连接的显示接口(未示出)。

[0049] 在I/O子系统111内,系统存储单元114可以连接至I/O中枢107,从而为计算系统100提供存储机制。I/O开关116可被用来提供实现I/O中枢107与其他组件之间的连接的接口机制,这些其他组件诸如,可被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设备120被添加的各种其他设备。网络适配器118可以是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Wi-Fi、蓝牙、近场通信(NFC)或包括一个或多个无线电装置的其他网络设备中的一者或多者。

[0050] 计算系统100可以包括未显式地示出的其他组件,包括USB或其他端口连接、光学存储驱动器、视频捕捉设备等等,这些组件也可连接至I/O中枢107。使图1中的各组件互连的通信路径可使用任何合适的协议来实现,这些协议诸如,基于PCI(外围组件互连)的协议(例如,PCI-Express)或任何其他总线或点对点通信接口和/或(多个)协议,诸如NV-Link高速互连或本领域已知的互连协议。

[0051] 在一个实施例中,一个或多个并行处理器112包含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)并构成图形处理单元(GPU)。在另一实施例中,本文中更详细地描述,一个或多个并行处理器112包含针对通用处理进行优化同时保留底层计算架构的电路。在又一实施例中,计算系统100的组件可与一个或多个其他系统元件集成在单个

集成电路上。例如，一个或多个并行处理器112、存储器中枢105、处理器102、以及I/O中枢107可以被集成在芯片上系统(SoC)集成电路中。替代地，计算系统100的组件可被集成到单个封装中，以形成系统级封装(SIP)配置。在一个实施例中，计算系统100的组件中的至少部分可被集成到多芯片模块(MCM)中，该多芯片模块可以与其他多芯片模块互连成为模块化计算系统。

[0052] 将会领会，本文中所示出的计算系统100是说明性的，并且变型和修改是可能的。可按需要修改连接拓扑，包括桥接器的数量和布置、(多个)处理器102的数量、以及(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直接地而不是通过桥接器连接至(多个)处理器102，而其他设备通过存储器中枢105和(多个)处理器102而与系统存储器104进行通信。在其他替代拓扑中，(多个)并行处理器112连接至I/O中枢107或直接连接至一个或多个处理器102中的一个，而不是连接至存储器中枢105。在其他实施例中，I/O中枢107和存储器中枢105可被集成到单个芯片中。一些实施例可包括经由多个插槽附连的两组或更多组(多个)处理器102，这些处理器102可以与(多个)并行处理器112的两个或更多个实例耦合。

[0053] 本文中所示出的特定组件中的一些是可选的，并且并非在计算系统100的所有实现方式中都包括这些组件。例如，可支持任何数量的插入式卡或外围设备，或者可消除一些组件。此外，一些架构可针对类似于图1中所图示的那些组件的组件使用不同的术语。例如，在一些架构中，存储器中枢105可被称为北桥，而I/O中枢107可被称为南桥。

[0054] 图2A图示出根据实施例的并行处理器200。并行处理器200的各组件可使用诸如可编程处理器、专用集成电路(ASIC)或现场可编程门阵列(FPGA)之类的一个或多个集成电路设备来实现。所图示的并行处理器200是根据实施例的图1中所示出的一个或多个并行处理器112的变型。

[0055] 在一个实施例中，并行处理器200包括并行处理单元202。并行处理单元包括实现与其他设备的通信的I/O单元204，这些其他设备包括并行处理单元202的其他实例。I/O单元204可直接连接至其他设备。在一个实施例中，I/O单元204通过使用中枢或开关接口(诸如，存储器中枢105)而与其他设备连接。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以及存储器交叉开关216连接，其中，主机接口206接收涉及执行处理操作的命令，并且存储器交叉开关216接收涉及执行存储器操作的命令。

[0056] 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引导至前端208。在一个实施例中，前端208与调度器210耦合，该调度器210被配置成用于将命令或其他工作项目分发给处理集群阵列212。在一个实施例中，调度器210确保在将任务分发给处理集群阵列212中的处理集群之前处理集群阵列212被恰当地配置并且处于有效状态。在一个实施例中，经由在微控制器上执行的固件逻辑来实现调度器210。微控制器实现的调度器210可配置成在粗粒度和细粒度下执行复杂的调度和工作分发操作，从而实现对在处理阵列212上执行的线程的快速抢占和上下文切换。在一个实施例中，主机软件可以经由多个图形处理门铃机制中的一者来证实用于在处理阵列212上调度的工作负荷。工作负荷随后可以由调度器微控制器内的调度器210逻辑跨处理阵列212自动地分发。

[0057] 处理集群阵列212可以包括高达“N”个处理集群(例如,集群214A、集群214B至集群214N)。处理集群阵列212中的每个集群214A-214N可以执行大量的并发线程。调度器210可以使用各种调度和/或工作分发算法将工作分配给处理集群阵列212中的集群214A-214N,这些调度和/或工作分发算法可取决于针对每种类型的程序或计算出现的工作负荷而变化。调度可以由调度器210动态地处置,或可以在对于被配置成供处理集群阵列212执行的程序逻辑的编译期间部分地由编译器逻辑协助。在一个实施例中,处理集群阵列212中的不同的集群214A-214N可以被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。

[0058] 处理集群阵列212可以被配置成用于执行各种类型的并行处理操作。在一个实施例中,处理集群阵列212被配置成用于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,这些处理任务包括对视频和/或音频数据的过滤、执行建模操作(包括物理操作)以及执行数据变换。

[0059] 在一个实施例中,处理集群阵列212被配置成用于执行并行图形处理操作。在其中并行处理器200被配置成用于执行图形处理操作的实施例中,处理集群阵列212可以包括用于支持此类图形处理操作的执行的附加逻辑,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可以被配置成用于执行与图形处理有关的着色器程序,诸如但不限于,顶点着色器、曲面细分着色器、几何着色器以及像素着色器。并行处理单元202可以经由I/O单元204从系统存储器传输数据以供处理。在处理期间,可在处理期间将所传输的数据存储到芯片上存储器(例如,并行处理器存储器222),随后将该数据写回到系统存储器。

[0060] 在一个实施例中,当使用并行处理单元202来执行图形处理时,调度器210可以被配置成用于将处理工作负荷分成近似相等尺寸的任务,以更好地实现图形处理操作到处理集群阵列212中的多个集群214A-214N的分发。在一些实施例中,处理集群阵列212的部分可以被配置成用于执行不同类型的处理。例如,第一部分可被配置成用于执行顶点着色和拓扑生成,第二部分可被配置成用于执行曲面细分和几何着色,并且第三部分可被配置成用于执行像素着色或其他屏幕空间操作,以产生用于显示的经渲染的图像。由集群214A-214N中的一个或多个集群产生的中间数据可被存储在缓冲器中,以允许该中间数据在集群214A-214N之间传送,以供进行进一步处理。

[0061] 在操作期间,处理集群阵列212可以经由调度器210接收将要被执行的处理任务,该调度器210从前端208接收定义处理任务的命令。对于图形处理操作,处理任务可以包括将要被处理的数据以及定义将如何处理该数据(例如,将执行什么程序)的状态参数和命令的索引,该数据例如,表面(补片(patch))数据、基元数据、顶点数据和/或像素数据。调度器210可被配置成用于取出与任务相对应的索引,或者可从前端208接收索引。前端208可以被配置成用于确保在由传入命令缓冲器(例如,批量缓冲器、推入缓冲器等)指定的工作负荷被发起之前处理集群阵列212被配置成有效状态。

[0062] 并行处理单元202的一个或多个实例中的每个实例可以与并行处理器存储器222耦合。可以经由存储器交叉开关216来访问并行处理器存储器222,该存储器交叉开关216可以接收来自处理集群阵列212以及I/O单元204的存储器请求。存储器交叉开关216可以经由存储器接口218来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至分区单元220N),这些分区单元可以各自耦合至并行处

理器存储器222的部分(例如,存储器单元)。在一种实现方式中,分区单元220A-220N的数量被配置成等于存储器单元的数量,以使得第一分区单元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设备的数量。

[0063] 在各实施例中,存储器单元224A-224N可以包括各种类型的存储器设备,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还可以包括3D堆叠式存储器,包括但不限于高带宽存储器(HBM)。本领域技术人员将会领会,存储器单元224A-224N的具体实现方式可以有所不同,并且可以从各种常规设计中的一种设计进行选择。诸如帧缓冲器或纹理映射之类的渲染目标可跨存储器单元224A-224N进行存储,从而允许分区单元220A-220N并行地写入每个渲染目标的部分,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,以有利于利用与本地高速缓存存储器结合的系统存储器的统一存储器设计。

[0064] 在一个实施例中,处理集群阵列212中的集群214A-214N中的任一者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中的任一者的数据。存储器交叉开关216可以被配置成用于将每个集群214A-214N的输出传输到任一分区单元220A-220N或传输到另一集群214A-214N,该另一集群214A-214N可以对该输出执行附加的处理操作。每个集群214A-214N可以通过存储器交叉开关216而与存储器接口218进行通信,以从各种外部存储器设备进行读取或写入到各种外部存储器设备。在一个实施例中,存储器交叉开关216具有到存储器接口218的连接以与I/O单元204进行通信,以及具有到并行处理器存储器222的本地实例的连接,从而使得不同处理集群214A-214N内的处理单元能够与系统存储器或对于并行处理单元202而言不是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可以使用虚拟通道来分离集群214A-214N与分区单元220A-220N之间的业务流(traffic stream)。

[0065] 尽管在并行处理器200内图示出并行处理单元202的单个实例,但可以包括并行处理单元202的任何数量的实例。例如,并行处理单元202的多个实例可以被设置在单个插入式卡上,或者多个插入式卡可以是互连的。并行处理单元202的不同实例可以被配置成用于互操作,即使不同的实例具有不同数量的处理核、不同的本地并行处理器存储器量、和/或其他配置差别。例如,在一个实施例中,并行处理单元202的一些实例可以包括相对于其他实例更高精度的浮点单元。包含并行处理单元202或并行处理器200的一个或多个实例的系统能以各种配置和形状因数来实现,这些配置和形状因子包括但不限于,台式计算机、膝上型计算机、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。

[0066] 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中的一个分区单元的实例。如所图示,分区单元220包括L2高速缓存221、帧缓冲器接口225、以及ROP 226(栅格操作单元)。L2高速缓存221是被配置成用于执行从存储器交叉开关216和ROP226接收的加载和存储操作的读取/写入高速缓存。读取未命中和紧迫写回请求由L2高速缓存221输出到帧缓冲器接口225以供处理。还可以经由帧缓冲器接口225将更新发送至帧缓冲器以供处理。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中的一者对接，该存储器单元诸如图2A的存储器单元224A-224N(例如，在并行处理器存储器222内)。

[0067] 在图形应用中，ROP 226是执行栅格操作(诸如，模板印制(stencil)、z测试、混合等等)的处理单元。ROP 226随后输出经处理的图形数据，经处理的图形数据被存储在图形存储器中。在一些实施例中，ROP 226包括用于压缩被写入至存储器的深度或颜色数据并解压缩从存储器读取的深度或颜色数据的压缩逻辑。压缩逻辑可以是利用多种压缩算法中的一种或多种的无损压缩逻辑。由ROP 226执行的压缩的类型可以基于将要被压缩的数据的统计特性而变化。例如，在一个实施例中，逐片(tile)地对深度和颜色数据执行Δ色彩压缩。

[0068] 在一些实施例中，ROP 226被包括在每个处理集群(例如，图2A的集群214A-214N)内而非被包括在分区单元220内。在此类实施例中，通过存储器交叉开关216来传送像素数据而非像素片段数据的读取和写入请求。经处理的图形数据可被显示在显示设备(诸如，图1的一个或多个显示设备110中的一个显示设备)上，可被路由以供(多个)处理器102进一步处理，或者可被路由以供图2A的并行处理器200内的处理实体中的一个处理实体进一步处理。

[0069] 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2A的处理集群214A-214N中的一个处理集群的实例。处理集群214可以被配置成用于并行地执行多个线程，其中，术语“线程”是指在特定的输入数据集合上执行的特定程序的实例。在一些实施例中，使用单指令多数据(SIMD)指令发布技术来支持大量线程的并行执行而无需提供多个独立的指令单元。在其他实施例中，使用单指令多线程(SIMT)技术来使用被配置成用于向处理集群中的每个处理集群内的处理引擎集合发布指令的公共指令单元来支持大量总体上同步的线程的并行执行。与其中所有处理引擎典型地执行相同指令的SIMD执行机制不同，SIMT执行允许不同的线程更容易地遵循通过给定的线程程序的发散的执行路径。本领域技术人员将理解，SIMD处理机制表示SIMT处理机制的功能子集。

[0070] 可以经由将处理任务分发给SIMT并行处理器的流水线管理器232来控制处理集群214的操作。流水线管理器232从图2A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。所图示的图形多处理器234是SIMT并行处理器的示例性实例。然而，可将不同架构的各种类型的SIMT并行处理器包括在处理集群214内。可以将图形多处理器234的一个或多个实例包括在处理集群214内。图形多处理器234可以处理数据，并且数据交叉开关240可以用于将经处理的数据分发到多个可能的目的地中的一个目的地，包括其他着色器单元。流水线管理器232可以通过为将要经由数据交叉开关240分发的经处理的数据指定目的地来促进对经处理的数据的分发。

[0071] 处理集群214内的每个图形多处理器234可以包括相同的功能执行逻辑集合(例如，算术逻辑单元、加载－存储单元等)。能以流水线化的方式配置功能执行逻辑，在该流水线化的方式中，可以在完成先前指令之前发布新的指令。功能执行逻辑支持各种操作，包括整数和浮点算术、比较操作、布尔操作、位移位、以及各种代数函数的计算。在一个实施例中，可以利用同一功能单元硬件来执行不同的操作，并且可能存在功能单元的任何组合。

[0072] 被传送至处理集群214的指令构成线程。跨并行处理引擎集合执行的线程集合是线程组。线程组对不同的输入数据执行同一程序。线程组内的每个线程可以被分派给图形

多处理器234内的不同处理引擎。线程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处理该线程组所在的周期期间可能是空闲的。线程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，可在连续的时钟周期上执行处理。在一个实施例中，可在图形多处理器234上同时执行多个线程组。

[0073] 在一个实施例中，图形多处理器234包括内部高速缓存存储器，以执行加载和存储操作。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248)。每个图形多处理器234还具有对分区单元(例如，图2A的分区单元220A-220N)内的L2高速缓存的访问，这些L2高速缓存在所有处理集群214之间共享并且可被用来在线程之间传输数据。图形多处理器234还可访问芯片外全局存储器，该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。并行处理单元202外部的任何存储器可被用作全局存储器。其中处理集群214包括图形多处理器234的多个实例的实施例可以共享公共指令和数据，这些公共指令和数据可被存储在L1高速缓存248中。

[0074] 每个处理集群214可包括被配置成用于将虚拟地址映射到物理地址的MMU 245(存储器管理单元)。在其他实施例中，MMU 245的一个或多个实例可驻留在图2A的存储器接口218内。MMU 245包括用于将虚拟地址映射到片的物理地址的页表条目(PTE)的集合并且可选地包括高速缓存行索引。MMU245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。对物理地址进行处理，以分发表面数据访问局部性，从而允许分区单元之间的高效的请求交织。可使用高速缓存行索引来确定针对高速缓存行的请求是命中还是未命中。

[0075] 在图形和计算应用中，处理集群214可被配置以使得每个图形多处理器234耦合至纹理单元236以供执行纹理映射操作，例如，确定纹理样本位置、读取纹理数据、以及过滤纹理数据。纹理数据读取自内部纹理L1高速缓存(未示出)，或者在一些实施例中，读取自图形多处理器234内的L1高速缓存，并按需要从L2高速缓存、本地并行处理器存储器或系统存储器取出。每个图形多处理器234向数据交叉开关240输出经处理的任务，以向另一处理集群214提供经处理的任务以供进一步处理，或者经由存储器交叉开关216将经处理的任务存储在L2高速缓存、本地并行处理器存储器或系统存储器中。preROP 242(预先栅格操作单元)被配置成用于从图形多处理器234接收数据、将数据引导至ROP单元，这些ROP单元可与如本文中所描述的分区单元(例如，图2A的分区单元220A-220N)一起被定位。preROP 242单元可针对颜色混合执行优化、组织像素颜色数据、并且执行地址转换。

[0076] 将会领会，本文中所描述的核架构是说明性的，并且变型和修改是可能的。可将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)包括在处理集群214内。进一步地，尽管仅示出了一个处理集群214，但是如本文中所描述的并行处理单元可包括处理集群214的任何数量的实例。在一个实施例中，每个处理集群214可以被配置成用于使用单独且不同的处理单元、L1高速缓存等来独立于其他处理集群214进行操作。

[0077] 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，该执行流

水线包括但不限于,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、以及一个或多个加载/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而与高速缓存存储器272以及共享存储器270耦合。在一个实施例中,图形多处理器234附加地包括张量和/或光线追踪核263,这些张量和/或光线追踪核263包括用于加速矩阵和/或光线追踪操作的硬件逻辑。

[0078] 在一个实施例中,指令高速缓存252从流水线管理器232接收要执行的指令流。指令被高速缓存在指令高速缓存252中并被分派以供指令单元254执行。指令单元254可以将指令作为线程组(例如,经线)进行分派,其中,线程组中的每个线程被指派给GPGPU核262内的不同执行单元。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。可以使用地址映射单元256将统一地址空间中的地址转换为可以由加载/存储单元266访问的不同的存储器地址。

[0079] 寄存器堆258为图形多处理器234的功能单元提供寄存器集合。寄存器堆258为连接至图形多处理器234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,寄存器堆258在功能单元中的每个功能单元之间划分,以使得给每个功能单元分配寄存器堆258中的专用部分。在一个实施例中,寄存器堆258在由图形多处理器234执行的不同经线之间划分。

[0080] GPGPU核262可以各自包括用于执行图形多处理器234的指令的浮点单元(FPU)和/或整数算术逻辑单元(ALU)。根据实施例,GPGPU核262在架构上可能类似,或者可能在架构上相区别。例如,并且在一个实施例中,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可以实现针对浮点算术的IEEE754-2008标准,或实现可变精度浮点算术。图形多处理器234可以附加地包括用于执行特定功能(诸如,复制矩形或像素混合操作)的一个或多个固定功能单元或专门功能单元。在一个实施例中,GPGPU核中的一个或多个还可以包括固定或专门功能逻辑。

[0081] 在一个实施例中,GPGPU核262包括能够对多个数据集合执行单个指令的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。针对GPGPU核的SIMD指令可以由着色器编译器在编译时生成,或在执行针对单程序多数据(SPMD)或SIMT架构而编写并且编译的程序时自动地生成。可以经由单个SIMD指令来执行被配置成用于SIMT执行模型的程序的多个线程。例如,并且在一个实施例中,可以经由单个SIMD8逻辑单元来并行执行八个SIMT线程,这八个SIMT线程执行相同或类似的操作。

[0082] 存储器和高速缓存互连268是将图形多处理器234的功能单元中的每个功能单元连接至寄存器堆258并连接至共享存储器270的互连网络。在一个实施例中,存储器和高速缓存互连268是允许加载/存储单元266实现共享存储器270与寄存器堆258之间的加载和存储操作的交叉开关互连。寄存器堆258能以与GPGPU核262相同的频率进行操作,因此GPGPU核262与寄存器堆258之间的数据传输是非常低等待时间的。可以使用共享存储器270来实现在图形多处理器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可以被用作数据高速缓存,例如,用于对在功能单元与纹理单元236之间传输的纹理数据进行高速缓存。共享存储器270还可以被用作所高速缓存的受管理的程序。除被存储在高速缓存存储器272内的自动高速缓存的数据之外,在GPGPU核262上执行的线程还能以编程方式将数据

存储在共享存储器内。

[0083] 图3A-图3C图示出根据实施例的附加图形多处理器。图3A-图3B图示图形多处理器325、350，它们是图2C的图形多处理器234的变型。图3C图示图形处理单元(GPU)380，该GPU380包括布置成多核组365A-365N的图形处理资源的专用集合。所图示的图形多处理器325、350和多核组365A-365N可以是能够同时执行大量执行线程的流式多处理器(SM)。

[0084] 图3A示出了根据附加实施例的图形多处理器325。相对于图2D的图形多处理器234，图形多处理器325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(多个)纹理单元344A-344B的多个实例。图形多处理器325还包括多个图形或计算执行单元集合(例如，GPGPU核336A-336B、张量核337A-337B、光线追踪核338A-338B)以及多个加载/存储单元集合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。

[0085] 各组件可以经由互连结构(interconnect fabric)327进行通信。在一个实施例中，互连结构327包括一个或多个交叉开关以实现在图形多处理器325的各组件之间的通信。在一个实施例中，互连结构327是单独的、高速网络结构层，图形多处理器325的每个组件堆叠在该网络结构层上。图形多处理器325的组件经由互连结构327与远程组件进行通信。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可以对图形多处理器325内的通信进行仲裁，以确保组件之间公平的带宽分配。

[0086] 图3B示出了根据附加实施例的图形多处理器350。图形处理器包括多个执行资源集合356A-356D，其中，如图2D和图3A中所图示，每个执行资源集合包括多个指令单元、寄存器堆、GPGPU核、以及加载存储单元。执行资源356A-356D可以与用于纹理操作的(多个)纹理单元360A-360D协同工作，同时共享指令高速缓存354和共享存储器353。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353以及纹理和/或数据高速缓存存储器358A-358B的多个实例。各组件可以经由类似于图3A的互连结构327的互连结构352进行通信。

[0087] 本领域技术人员将会理解，图1、图2A-图2D以及图3A-图3B中所描述的架构是描述性的并且不限制本实施例的范围。因此，在不背离本文中所描述的实施例的范围的情况下，本文中所描述的技术可在任何恰当配置的处理单元上实现，该处理单元包括但不限于一个或多个移动应用处理器、一个或多个台式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2A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。

[0088] 在一些实施例中，本文中所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、以及各种通用GPU(GPGPU)功能。GPU可通过总线或其他互连(例如，诸如PCIe或NVLink之类的高速互连)而通信地耦合至主机处理器/核。在其他实施例中，GPU可与核集成在同一封装或芯片上，并且通过内部处理器总线/互连(即，在封装或芯片内部)通信地耦合至核。不论连接GPU所采用的方式如何，处理器核都可以采用工作描述符中所包含的命令/指令序列的形式将工作分配给GPU。GPU随后使用专用电路/逻辑来有效地处理这些命令/指令。

[0089] 图3C图示图形处理单元(GPU)380，该GPU 380包括布置为多核组365A-365N的专用的图形处理资源的集合。虽然提供仅单个多核组365A的细节，但是将理解，其他多核组365B-365N可配备有相同或类似集合的图形处理资源。

[0090] 如所图示，多核组365A可包括图形核的集合370、张量核的集合371以及光线追踪核的集合372。调度器/分派器368调度和分派图形线程用于在各个核370、371、372上执行。寄存器堆的集合369存储在执行图形线程时由核370、371、372使用的操作数值。这些寄存器堆可包括例如用于存储整数值的整数寄存器、用于存储浮点值的浮点寄存器、用于存储紧缩数据元素(整数和/或浮点数据元素)的向量寄存器以及用于存储张量/矩阵值的操作数矩阵寄存器。在一个实施例中，操作数矩阵寄存器被实现为组合的向量寄存器的集合。

[0091] 一个或多个组合的第一级(L1)高速缓存和共享存储器单元373在本地将图形数据存储在每个多核组365A内，图形数据诸如纹理数据、顶点数据、像素数据、光线数据、包围体数据等。一个或多个纹理单元374也可用于执行纹理操作，诸如，纹理映射和采样。由所有多核组365A-365N或多核组365A-365N的子集共享的第二级(L2)高速缓存375存储用于多个并发的图形线程的图形数据和/或指令。如所图示，可跨多个多核组365A-365N共享L2高速缓存375。一个或多个存储器控制器367将GPU 380耦合至存储器366，该存储器366可以是系统存储器(例如，DRAM)和/或专用图形存储器(例如，GDDR6存储器)。

[0092] 输入/输出(I/O)电路363将GPU 380耦合至一个或多个I/O设备362，这一个或多个I/O设备362诸如数字信号处理器(DSP)、网络控制器或用户输入设备。芯片上互连可用于将I/O设备362耦合至GPU 380和存储器366。I/O电路363的一个或多个存储器管理单元(IOMMU)364直接将I/O设备362耦合至系统存储器366。在一个实施例中，IOMMU 364管理用于将虚拟地址映射到系统存储器366中的物理地址的多个集合的页表。在该实施例中，I/O设备362、(多个)CPU 361和(多个)GPU 380可共享相同的虚拟地址空间。

[0093] 在一个实现方式中，IOMMU 364支持虚拟化。在这种情况下，IOMMU364管理用于将宾客/图形虚拟地址映射到宾客/图形物理地址的第一集合的页表以及用于将宾客/图形物理地址映射到(例如，系统存储器366内的)系统/主机物理地址的第二集合的页表。第一集合的页表和第二集合的页表中的每一个的基址可被存储在控制寄存器中，并且在上下文切换时被换出(例如，使得新上下文被提供有对相关集合的页表的访问权)。虽然未在图3C中图示，但是核370、371、372和/或多核组365A-365N中的每一个可包括转换后备缓冲器(TLB)，这些TLB用于对宾客虚拟至宾客物理转换、宾客物理至主机物理转换以及宾客虚拟至主机物理转换进行高速缓存。

[0094] 在一个实施例中，CPU 361、GPU 380和I/O设备362被集成在单个半导体芯片和/或芯片封装上。所图示的存储器366可集成在同一芯片上，或者可经由片外接口被耦合至存储器控制器367。在一个实现方式中，存储器366包括与其他物理系统级存储器共享同一虚拟地址空间的GDDR6存储器，但是本发明的根本性原理不限于该特定的实现方式。

[0095] 在一个实施例中，张量核371包括专门设计成用于执行矩阵操作的多个执行单元，这些矩阵操作是用于执行深度学习操作的基本计算操作。例如，可将同步矩阵乘法操作用于神经网络训练和推断。张量核371可使用各种操作数精度来执行矩阵处理，操作数精度包括单精度浮点(例如，32位)、半精度浮点(例如，16位)、整数字(16位)、字节(8位)和半字节(4位)。在一个实施例中，神经网络实现方式提取每个经渲染场景的特征，从而潜在地组合

来自多个帧的细节,以构建高质量的最终图像。

[0096] 在深度学习实现方式中,可调度并行的矩阵乘法工作用于在张量核371上执行。神经网络的训练尤其需要大量矩阵点积操作。为了处理N x N x N矩阵乘法的内积公式化,张量核371可包括至少N个点积处理元件。在矩阵乘法开始之前,一个完整的矩阵被加载到操作数矩阵寄存器中,并且对于N个循环中的每个循环,第二矩阵的至少一列被加载。对于每个循环,存在被处理的N个点积。

[0097] 取决于特定的实现方式,能以不同精度来存储矩阵元素,包括16位的字、8位的字节(例如,INT8)以及4位的半字节(例如,INT4)。可为张量核371指定不同的精度模式以确保将最高效的精度用于不同的工作负载(例如,诸如推断工作负载,其可容忍至字节和半字节的离散化(quantization))。

[0098] 在一个实施例中,光线追踪核372加速用于实时光线追踪实现方式和非实时光线追踪实现方式两者的光线追踪操作。具体而言,光线追踪核372包括光线遍历/相交电路,该光线遍历/相交电路用于使用包围体层级结构(BVH)来执行光线遍历并识别封围在BVH容体内的光线与基元之间的相交。光线追踪核372还可包括用于执行深度测试和剔除(例如,使用Z缓冲器或类似布置)的电路。在一个实现方式中,光线追踪核372与本文中描述的图像降噪技术一致地执行遍历和相交操作,该图像降噪技术的至少部分可在张量核371上执行。例如,在一个实施例中,张量核371实现深度学习神经网络以执行对由光线追踪核372生成的帧的降噪。然而,(多个)CPU 361、图形核370和/或光线追踪核372还可实现降噪和/或深度学习算法的全部或部分。

[0099] 此外,如上文所描述,可采用对于降噪的分布式方法,在该分布式方法中,GPU 380在通过网络或高速互连而耦合至其他计算设备的计算设备中。在该实施例中,经互连的计算设备共享神经网络学习/训练数据以改善整个系统学习执行用于不同类型的图像帧和/或不同的图形应用的降噪的速度。

[0100] 在一个实施例中,光线追踪核372处理所有的BVH遍历和光线-基元相交,从而挽救图形核370免于被针对每条光线的数千条指令过载。在一个实施例中,每个光线追踪核372包括用于执行包围盒测试(例如,用于遍历操作)的第一集合的专业电路以及用于执行光线-三角形相交测试(例如,使已被遍历的光线相交)的第二集合的专业电路。由此,在一个实施例中,多核组365A可简单地发起光线探测,并且光线追踪核372独立地执行光线遍历和相交,并将命中数据(例如,命中、无命中、多个命中等)返回到线程上下文。当光线追踪核370执行遍历和相交操作时,其他核371、372被释放以执行其他图形或计算工作。

[0101] 在一个实施例中,每个光线追踪核372包括用于执行BVH测试操作的遍历单元以及执行光线-基元相交测试的相交单元。相交单元生成“命中”、“无命中”或“多个命中”响应,该相交单元将这些响应提供给适当的线程。在遍历和相交操作期间,其他核(例如,图形核370和张量核371)的执行资源被释放以执行其他形式的图形工作。

[0102] 在下文描述的一个特定实施例中,使用在其中工作被分布在图形核370与光线追踪核372之间的混合式栅格化/光线追踪方法。

[0103] 在一个实施例中,光线追踪核372(和/或其他核370、371)包括对光线追踪指令集的硬件支持,光线追踪指令集诸如:微软的DirectX光线追踪(DXR),其包括DispatchRays命令;以及光线生成着色器、最近命中着色器、任何命中着色器和未命中着色器,它们启用为

每个对象指派唯一集合的着色器和纹理。可由光线追踪核372、图形核370和张量核371支持的另一光线追踪平台是Vulkan 1.1.85。然而，要注意本发明的根本性原理不限于任何特定的光线追踪指令集架构ISA。

[0104] 一般而言，各个核372、371、370可支持包括用于以下各项的指令/函数的光线追踪指令集：光线生成、最近命中、任何命中、光线-基元相交、逐基元和层级结构包围盒构建、未命中、拜访、和异常。更具体地，一个实施例包括用于执行以下功能的光线追踪指令：

[0105] 光线生成——可为每个像素、样本或用户定义的工作分配执行光线生成指令。

[0106] 最近命中——可执行最近命中指令以对场景内光线与基元的最近交点定位。

[0107] 任何命中——任何命中指令识别场景内光线与基元的多个相交，从而潜在地识别新的最近交点。

[0108] 相交——相交指令执行光线-基元相交测试并输出结果。

[0109] 逐基元包围盒构建——该指令围绕给定的基元或基元组建立包围盒(例如，当建立新BVH或其他加速数据结构时)。

[0110] 未命中——指示光线未命中场景或场景的指定区域内的全部几何体。

[0111] 拜访——指示光线将遍历的子容体。

[0112] 异常——包括各种类型的异常处置器(例如，为各种错误条件调用)。

[0113] 用于GPU到主机处理器互连的技术

[0114] 图4A图示出其中多个GPU 410-413通过高速链路440A-440D(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的示例性架构。在一个实施例中，取决于实现方式，高速链路440A-440D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，这些互连协议包括但不限于，PCIe 4.0或5.0以及NVLink 2.0。然而，本发明的基本原理不限于任何特定的通信协议或吞吐量。

[0115] 另外，在一个实施例中，GPU 410-413中的两个或更多个通过高速链路442A-442B进行互连，这可使用与用于高速链路440A-440D的那些协议/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中的两个或更多个可通过高速链路443进行连接，该高速链路443可以是在20GB/s、30GB/s、120GB/s或更高速度下进行操作的对称多处理器(SMP)总线。替代地，图4A中示出的各种系统组件之间的所有通信可使用相同的协议/链路(例如，通过公共互连结构)来完成。如所提到，然而，本发明的基本原理不限于任何特定类型的互连技术。

[0116] 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430A-430B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450A-450D通信地耦合至GPU存储器420-423。存储器互连430A-430B和450A-450D可利用相同或不同的存储器访问技术。作为示例并且不作为限制，处理器存储器401-402和GPU存储器420-423可以是诸如动态随机存取存储器(DRAM)(包括堆叠的DRAM)、图形DDR SDRAM(GDDR)(例如，GDDR5、GDDR6)、或高带宽存储器(HBM)之类的易失性存储器，并且/或者可以是诸如3D XPoint或Nano-Ram之类的非易失性存储器。在一个实施例中，这些存储器中的某个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)层级结构)。

[0117] 如下文中所描述，尽管各处理器405-406和GPU 410-413可分别物理地耦合至特定

的存储器401-402、420-423，但是可实现其中同一虚拟系统地址空间(也被称为“有效地址”空间)在所有的各种物理存储器之间进行分布的统一存储器架构。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(在该示例中，得到总计256GB可寻址的存储器)。

[0118] 图4B图示出根据一个实施例的多核处理器407和图形加速模块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，该线卡经由高速链路440耦合至处理器407。替代地，可将图形加速模块446集成在与处理器407相同的封装或芯片上。

[0119] 所图示的处理器407包括多个核460A-460D，各自具有转换后备缓冲器461A-461D以及一个或多个高速缓存462A-462D。核可包括用于执行指令并处理数据的各种其他组件，未图示出这些组件以避免使本发明的基本原理模糊(例如，指令取出单元、分支预测单元、解码器、执行单元、重排序缓冲器等)。高速缓存462A-462D可包括第1级(L1)高速缓存和第2级(L2)高速缓存。另外，一个或多个共享高速缓存456可被包括在高速缓存层级结构中并由核集合460A-460D共享。例如，处理器407的一个实施例包括24个核，各自具有其自己的L1高速缓存、十二个共享L2高速缓存、以及十二个共享L3高速缓存。在该实施例中，L2高速缓存和L3高速缓存中的一者由两个相邻的核共享。处理器407和图形加速器集成模块446与系统存储器441连接，该系统存储器441可包括处理器存储器401-402。

[0120] 经由通过一致性总线464的核间通信为存储在各高速缓存462A-462D、456和系统存储器441中的数据和指令维持一致性。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，以响应于检测到的对特定高速缓存行的读取或写入而通过一致性总线464进行通信。在一种实现方式中，通过一致性总线464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/一致性技术为本领域技术人员很好地理解并且将不在此详细描述，以避免使本发明的基本原理模糊。

[0121] 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作为核的对等体参与高速缓存一致性协议。具体而言，接口435通过高速链路440(例如，PCIe总线、NVLink等)提供到代理电路425的连接性，并且接口437将图形加速模块446连接至高速链路440。

[0122] 在一种实现方式中，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提供高速缓存管理、存储器访问、上下文管理、以及中断管理服务。图形处理引擎431、432、N可各自包括单独的图形处理单元(GPU)。替代地，图形处理引擎431、432、N可包括GPU内的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)、采样器以及blit引擎。换言之，图形加速模块可以是具有多个图形处理引擎431-432、N的GPU，或者图形处理引擎431-432、N可以是集成在公共封装、线卡或芯片上的单独的GPU。

[0123] 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，该MMU 439用于执行诸如虚拟到物理存储器转换(也称为有效到实际存储器转换)之类的各种存储器管理功能以及用于访问系统存储器441的存储器访问协议。MMU 439还可包括用于对虚拟/有效到物理/实际地址转换进行高速缓存的转换后备缓冲器(TLB)(未示出)。在一种实现方式中，高速缓存438存储用于由图形处理引擎431-432、N高效访问的命令和数据。在一个实

施例中，使高速缓存438和图形存储器433-434、M中存储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来完成，该代理电路425代表高速缓存438和存储器433-434、M参与高速缓存一致性机制(例如，向高速缓存438发送与处理器高速缓存462A-462D、456上的高速缓存行的修改/访问相关的更新并从高速缓存438接收更新)。

[0124] 寄存器集合445存储针对由图形处理引擎431-432、N执行的线程的上下文数据，并且上下文管理电路448管理这些线程上下文。例如，上下文管理电路448可在上下文切换期间执行用于保存和恢复各线程的上下文的保存和恢复操作(例如，其中，第一线程被保存并且第二线程被存储，以使得第二线程可以由图形处理引擎执行)。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的指定区域(例如，由上下文指针标识)。当返回到该上下文时，其随后可恢复寄存器值。在一个实施例中，中断管理电路447接收并处理接收自系统设备的中断。

[0125] 在一种实现方式中，由MMU 439将来自图形处理引擎431的虚拟/有效地址转换为系统存储器411中的实际/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个、8个、16个)图形加速器模块446和/或其他加速器设备。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，或者可在多个应用之间共享。在一个实施例中，存在虚拟化图形执行环境，其中，图形处理引擎431-432、N的资源与多个应用或虚拟机(VM)共享。这些资源可被细分为“切片(slice)”，这些切片基于处理要求以及与VM和/或应用相关联的优先级而被分配给不同的VM和/或应用。

[0126] 因此，加速器集成电路充当用于图形加速模块446的到系统的桥接器，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，以管理对图形处理引擎、中断和存储器管理的虚拟化。

[0127] 因为图形处理引擎431-432、N的硬件资源被显式地映射到由主机处理器407查看的实际地址空间，所以任何主机处理器可以使用有效地址值直接对这些资源进行寻址。在一个实施例中，加速器集成电路436的一个功能是对图形处理引擎431-432、N的物理分离，以使得它们对系统看起来是独立单元。

[0128] 如所提到，在所图示的实施例中，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中的每个图形处理引擎。图形存储器433-434、M存储由图形处理引擎431-432、N中的每个图形处理引擎处理的指令和数据。图形存储器433-434、M可以是诸如DRAM(包括堆叠的DRAM)、GDDR存储器(例如，GDDR5、GDDR6)或HBM之类的易失性存储器，和/或可以是诸如3D XPoint或Nano-Ram之类的非易失性存储器。

[0129] 在一个实施例中，为了降低高速链路440上的数据通信量，使用偏置技术来确保存储在图形存储器433-434、M中的数据是将由图形处理引擎431-432、N最频繁地使用并且优选地不由核460A-460D使用(至少不频繁地使用)的数据。类似地，偏置机制尝试将核(并且优选地，不是图形处理引擎431-432、N)所需要的数据保持在这些核的高速缓存462A-462D、456和系统存储器411内。

[0130] 图4C图示出其中加速器集成电路436被集成在处理器407内的另一实施例。在该实施例中，经由接口437和接口435(其同样可利用任何形式的总线或接口协议)，图形处理引擎431-432、N通过高速链路440与加速器集成电路436直接通信。加速器集成电路436可执行

与参考图4B所描述的那些操作相同的操作，但是考虑到该加速器集成电路436与一致性总线464和高速缓存462A-462D、456紧密邻近，其潜在地可在较高的吞吐量下执行操作。

[0131] 一个实施例支持不同的编程模型，这些编程模型包括专用进程编程模型(无图形加速模块虚拟化)和共享编程模型(具有虚拟化)。后者可包括受加速器集成电路436控制的编程模型以及受图形加速模块446控制的编程模型。

[0132] 在专用进程模型的一个实施例中，图形处理引擎431-432、N在单个操作系统下专用于单个应用或进程。该单个应用可以将其他应用请求汇集至图形引擎431-432、N，从而提供VM/分区内的虚拟化。

[0133] 在专用进程编程模型中，图形处理引擎431-432、N可由多个VM/应用分区共享。共享模型要求系统管理程序使图形处理引擎431-432、N虚拟化，以允许由每个操作系统访问。对于不具有管理程序的单分区系统，图形处理引擎431-432、N由操作系统所有。在这两种情况下，操作系统可以使图形处理引擎431-432、N虚拟化，以提供对每个进程或应用的访问。

[0134] 对于共享编程模型，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要素。在一个实施例中，进程要素被存储在系统存储器411中，并且是使用本文中所描述的有效地址到实际地址转换技术可寻址的。进程句柄可以是在向图形处理引擎431-432、N注册其上下文(即，调用系统软件以将进程要素添加到进程要素链表)时提供给主机进程的实现方式特定的值。进程句柄的较低的16位可以是进程要素在进程要素链表内的偏移。

[0135] 图4D图示出示例性加速器集成切片490。如本文中所使用，“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存储进程要素483。在一个实施例中，进程要素483响应于来自在处理器407上执行的应用480的GPU调用481而被存储。进程要素483包含对应应用480的进程状态。包含在进程要素483中的工作描述符(WD)484可以是由应用请求的单个作业，或者可包含指向作业队列的指针。在后一种情况中，WD 484是指向应用的地址空间482中的作业请求队列的指针。

[0136] 图形加速模块446和/或各个图形处理引擎431-432、N可以由系统中的进程的全部或其子集共享。本发明的实施例包括用于建立进程状态并将WD 484发送至图形加速模块446以在虚拟化环境中开始作业的基础设施。

[0137] 在一种实现方式中，专用进程编程模型是实现方式特定的。在该模型中，单个进程拥有图形加速模块446或单独的图形处理引擎431。因为图形加速模块446为单个进程所拥有，所以在指派图形加速模块446时，管理程序针对拥有的分区对加速器集成电路436进行初始化，并且操作系统针对拥有的进程对加速器集成电路436进行初始化。

[0138] 在操作中，加速器集成切片490中的WD取出单元491取出包括对将要由图形加速模块446的图形处理引擎中的一个完成的工作的指示的下一WD 484。如所图示，来自WD 484的数据可被存储在寄存器445中，并且由MMU 439、中断管理电路447和/或上下文管理电路448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内的段表/页表486的段/页走查电路。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，由图形处理引擎431-432、N生成的有效地址493由MMU 439转换成实际地址。

[0139] 在一个实施例中，相同的寄存器集合445针对每个图形处理引擎431-432、N和/或图形加速模块446被复制并且可由管理程序或操作系统初始化。这些复制的寄存器中的每

一个可被包括在加速器集成切片490中。在表1中示出可由管理程序初始化的示例性寄存器。

[0140] 表1-管理程序初始化的寄存器

[0141]

1	切片控制寄存器
2	实际地址(RA)调度的进程区域指针
3	权限掩码覆盖寄存器
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	实际地址(RA)管理程序加速器利用记录指针
9	存储描述寄存器

[0142] 在表2中示出可由操作系统初始化的示例性寄存器。

[0143] 表2-操作系统初始化的寄存器

[0144]

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	权限掩码
6	工作描述符

[0145] 在一个实施例中，每个WD 484对于特定的图形加速模块446和/或图形处理引擎431-432、N是特定的。其包含图形处理引擎431-432、N进行其工作所要求的所有信息，或者其可以是指向应用已经建立将要完成的工作的命令队列所在的存储器位置的指针。

[0146] 图4E图示出共享模型的一个实施例的附加细节。该实施例包括进程要素列表499被存储在其中的管理程序实际地址空间498。管理程序实际地址空间498是经由管理程序496可访问的，该管理程序496使用于操作系统495的图形加速模块引擎虚拟化。

[0147] 共享编程模型允许来自系统中的分区的全部或其子集的进程的全部或其子集使用图形加速模块446。存在其中图形加速模块446由多个进程和分区共享的两种编程模型：时分共享和图形定向共享。

[0148] 在该模型中，系统管理程序496拥有图形加速模块446并使其功能对所有的操作系统495可用。为了使图形加速模块446支持由系统管理程序496进行的虚拟化，图形加速模块446可遵守下列要求：1)应用的作业请求必须是自主的(即，不需要在作业之间维持状态)，或者图形加速模块446必须提供上下文保存和恢复机制。2)由图形加速模块446保证在所指定的时间量内完成应用的作业请求，包括任何转换错误，或者图形加速模块446提供抢占对作业的处理的能力。3)当在定向共享编程模型下进行操作时，必须保证图形加速模块446在进程之间的公平性。

[0149] 在一个实施例中，对于共享模型，要求应用480利用图形加速模块446类型、工作描述符(WD)、权限掩码寄存器(AMR)值和上下文保存/恢复区域指针(CSRP)作出操作系统495系统调用。图形加速模块446类型描述系统调用的目标加速功能。图形加速模块446类型可以是系统特定的值。特别针对图形加速模块446对WD进行格式化，并且WD可以采用图形加速模块446命令、指向用户定义的结构的有效地址指针、指向命令队列的有效地址指针、或用于描述将要由图形加速模块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值是用于当前进程的AMR状态。传递至操作系统的值类似于设置AMR的应用。如果加速器集成电路436和图形加速模块446实现方式不支持用户权限掩码覆盖寄存器(UAMOR)，则操作系统可在管理程序调用中传递AMR之前将当前UAMOR值应用到AMR值。管理程序496可在将AMR放置到进程要素483中之前可选地应用当前权限掩码覆盖寄存器(AMOR)值。在一个实施例中，CSRP是寄存器445中的包含应用地址空间482中用于图形加速模块446保存和恢复上下文状态的区域的有效地址的一个寄存器。如果不要求在作业之间对状态进行保存或者在作业被抢占时，则该指针是可选的。上下文保存/恢复区域可以是固定的系统存储器。

[0150] 一旦接收系统调用，则操作系统495可验证应用480已注册并且已给予该应用480使用图形加速模块446的权限。操作系统495随后利用表3中示出的信息来调用管理程序496。

[0151] 表3-OS对管理程序的调用参数

[0152]

1	工作描述符(WD)
2	权限掩码寄存器(AMR)值(潜在地被掩码)。
3	有效地址(EA)上下文保存/恢复区域指针(CSRP)
4	进程ID(PID)和可选的线程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)

[0153] 一旦接收管理程序调用，则管理程序496验证操作系统495已注册并且已给予该操作系统495使用图形加速模块446的权限。管理程序496随后将进程要素483置于针对对应的图形加速模块446类型的进程要素链表中。进程要素可包括表4中示出的信息。

[0154] 表4-进程要素信息

[0155]

1	工作描述符（WD）
2	权限掩码寄存器（AMR）值（潜在地被掩码）。
3	有效地址（EA）上下文保存/恢复区域指针（CSRP）
4	进程 ID（PID）和可选的线程 ID（TID）

[0156]

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	中断向量表，从管理程序调用参数推导出。
9	状态寄存器（SR）值
10	逻辑分区 ID（LPID）
11	实际地址（RA）管理程序加速器利用记录指针
12	存储描述符寄存器（SDR）

[0157] 在一个实施例中，管理程序使多个加速器集成切片490寄存器445初始化。

[0158] 如图4F中所图示，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，该公共虚拟存储器地址空间被用来访问物理处理器存储器401-402和GPU存储器420-423。在该实现方式中，在GPU 410-413上执行的操作利用同一虚拟/有效存储器地址空间来访问处理器存储器401-402并且反之亦然，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，第三部分被分配给GPU存储器420，依此类推。整个虚拟/有效存储器空间(有时被称为有效地址空间)由此跨处理器存储器401-402和GPU存储器420-423中的每一个分布，从而允许任何处理器或GPU利用映射到该存储器的虚拟地址来访问任何物理存储器。

[0159] 在一个实施例中，MMU 439A-439E中的一个或多个内的偏置/一致性管理电路494A-494E确保主机处理器(例如，405)的高速缓存与GPU 410-413的高速缓存之间的高速缓存一致性，并且实现指示某些类型的数据应当被存储在其中的物理存储器的偏置技术。尽管在图4F中图示出偏置/一致性管理电路494A-494E的多个实例，但偏置/一致性电路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。

[0160] 一个实施例允许GPU附连的存储器420-423被映射为系统存储器的部分并使用共享虚拟存储器(SVM)技术来访问，但是不会遭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作为系统存储器被访问、并且没有繁重的高速缓存一致性开销的能力为GPU卸载提供了有益的操作环境。此种布置允许主机处理器405软件设置操作数并访问计算结果，而没有传统I/O DMA数据复制的开销。此类传统复制涉及驱动程序调用、中断和存储器映射的I/O(MMIO)访问，这些相对于简单存储器访问都是低效的。同时，访问GPU附连的存储器420-423而没有高速缓存一致性开销的能力对于被卸载的计算的执行时间可能是关键的。例如，在具有大量流式的写存储器业务的情况下，高速缓存一致性开销可以显著地降低GPU 410-413看到的有效写入带宽。操作数设置的效率、结果访问的效率以及GPU计算的效率在确定GPU卸载的有效性时都发挥作用。

[0161] 在一种实现方式中，GPU偏置与主机处理器偏置之间的选择由偏置跟踪程序数据结构驱动。例如，可使用偏置表，该偏置表可以是页粒度的结构(即，在存储器页粒度下受控制)，该页粒度的结构包括每GPU附连的存储器页的1或2个位。偏置表可在一个或多个GPU附

连的存储器420-423的偷取的存储器范围中实现，在GPU 410-413中具有或不具有偏置高速缓存(例如，用于对频繁/最近使用的偏置表条目进行高速缓存)。替代地，可将整个偏置表维持在GPU内。

[0162] 在一种实现方式中，在对GPU存储器的实际访问之前，访问与对GPU附连的存储器420-423的每次访问相关联的偏置表条目，导致下列操作。首先，来自GPU 410-413的在GPU偏置中发现它们的页的本地请求直接被转发至对应的GPU存储器420-423。来自GPU的在主机偏置中发现它们的页的本地请求被转发至处理器405(例如，如以上所讨论，通过高速链路)。在一个实施例中，来自处理器405的在主机处理器偏置中发现所请求的页的请求像正常存储器读取那样完成该请求。替代地，涉及GPU偏置的页的请求可被转发至GPU 410-413。如果GPU当前不是正在使用该页，则GPU随后可将该页转变为主机处理器偏置。

[0163] 可以通过基于软件的机制、硬件辅助的基于软件的机制、或对于有限的情况的集合而言通过基于纯硬件的机制来改变页的偏置状态。

[0164] 一种用于改变偏置状态的机制采用API调用(例如，OpenCL)，其进而调用GPU的设备驱动程序，该设备驱动程序进而向GPU发送消息(或将命令描述符入列)以指引其改变偏置状态、并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偏置到GPU偏置的转变是必需的，但对于相反的转变不是必需的。

[0165] 在一个实施例中，通过临时渲染不可由主机处理器405高速缓存的GPU偏置的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，取决于实现方式，GPU 410可以或可以不立即授权访问。因此，为了减少主机处理器405与GPU 410之间的通信，确保GPU偏置的页是GPU而不是主机处理器405要求的那些页是有益的，并且反之亦然。

[0166] 图形处理流水线

[0167] 图5图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图示的图形处理流水线500。可以将图形处理器包括在如本文中所描述的并行处理子系统内，该并行处理子系统诸如图2A的并行处理器200，在一个实施例中，该并行处理器是图1的(多个)并行处理器112的变型。各种并行处理系统可以通过本文中所描述的并行处理单元(例如，图2A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2C的图形多处理器234)可被配置成用于执行顶点处理单元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、以及片段/像素处理单元524中的一者或多者的功能。数据组装器502、基元组装器506、514、518、曲面细分单元510、栅格化器522、以及栅格操作单元526的功能也可由处理集群(例如，图2A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2A的分区单元220A-220N)执行。图形处理流水线500还可使用针对一个或多个功能的专用处理单元来实现。在一个实施例中，图形处理流水线500的一个或多个部分可由通用处理器(例如，CPU)内的并行处理逻辑执行。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2A中的并行处理器存储器222)，该存储器接口528可以是图2A的存储器接口218的实例。

[0168] 在一个实施例中，数据组装器502是收集表面和基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到顶点处理单元504。顶点处理单元504是

可编程的执行单元，该可编程的执行单元执行顶点着色器程序，从而按照顶点着色器程序所指定地来照明以及变换顶点数据。顶点处理单元504读取高速缓存、本地或系统存储器中所存储的数据以供在处理顶点数据时使用，并且可被编程为用于将顶点数据从基于对象的坐标表示变换为世界空间坐标空间或规范化设备坐标空间。

[0169] 基元组装器506的第一实例从顶点处理单元504接收顶点属性。基元组装器506按需要读取所存储的顶点属性并且构建图形基元以供曲面细分控制处理单元508进行处理。图形基元包括如由各种图形处理应用编程接口(API)支持的三角形、线段、点、补片等。

[0170] 曲面细分控制处理单元508将输入顶点视为几何补片的控制点。将控制点从来自补片的输入表示(例如，补片的基础)变换为适合于在由曲面细分评估处理单元512进行的表面评估中使用的表示。曲面细分控制处理单元508还可以计算几何补片的边缘的曲面细分因子。曲面细分因子应用于单个边缘，并量化与该边缘相关联的依赖于视图的细节等级。曲面细分单元510被配置成用于接收补片的边缘的曲面细分因子，并且用于将补片曲面细分成多个几何基元(诸如，线、三角形或四边形基元)，这些几何基元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对经细分的补片的参数化坐标进行操作，以生成与几何基元相关联的每个顶点的表面表示和顶点属性。

[0171] 基元组装器514的第二实例从曲面细分评估处理单元512接收顶点属性，根据需要读取所存储的顶点属性，并且构建图形基元以供几何处理单元516进行处理。几何处理单元516是可编程的执行单元，该可编程的执行单元执行几何着色器程序以如几何着色器程序所指定地变换从基元组装器514接收到的图形基元。在一个实施例中，几何处理单元516被编程为用于将图形基元细分为一个或多个新的图形基元并计算被用来对这些新的图形基元进行栅格化的参数。

[0172] 在一些实施例中，几何处理单元516可以在几何流中添加或删除元素。几何处理单元516向基元组装器518输出指定新的图形基元的参数和顶点。基元组装器518从几何处理单元516接收参数和顶点并构建图形基元以供视口缩放、剔除和裁剪单元520进行处理。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处理几何数据时使用。视口缩放、剔除和裁剪单元520执行裁剪、剔除和视口缩放，并且将经处理的图形基元输出到栅格化器522。

[0173] 栅格化器522可以执行深度剔除和其他基于深度的优化。栅格化器522还对新的图形基元执行扫描转换以生成片段并将那些片段及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是被配置成执行片段着色器程序或像素着色器程序的可编程的执行单元。片段/像素处理单元524按照片段或像素着色器程序所指定地变换从栅格化器522接收的片段或像素。例如，片段/像素处理单元524可被编程为用于执行包括但不限于纹理映射、着色、混合、纹理校正和透视校正的操作，以产生被输出到栅格操作单元526的经着色的片段或像素。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处理片段数据时使用。片段或像素着色器程序可被配置成用于取决于为处理单元配置的采样率而以样本、像素、片或其他粒度来进行着色。

[0174] 栅格操作单元526是处理单元，该处理单元执行栅格操作并将像素数据输出为将要被存储在图形存储器(例如，如图2A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、将要在一个或多个显示设备110上显示或用于由一个或多个处理器102或并行处

理器112中的一个进一步处理的经处理的图形数据，这些栅格操作包括但不限于模版印制、z测试、混合等等。在一些实施例中，栅格操作单元526被配置成用于压缩被写入到存储器的z或颜色数据并解压缩从存储器读取的z或颜色数据。

[0175] 机器学习概览

[0176] 可应用上文描述的架构以使用机器学习模型来执行训练和推断操作。机器学习已经在解决许多种类的任务方面取得成功。当训练和使用机器学习算法(例如，神经网络)时产生的计算自然地适合于高效的并行实现。因此，诸如通用图形处理单元(GPGPU)之类的并行处理器已在深度神经网络的实际实现中扮演了重要角色。具有单指令多线程(SIMT)架构的并行图形处理器被设计成使图形流水线中的并行处理的量最大化。在SIMT架构中，成组的并行线程尝试尽可能频繁地一起同步地执行程序指令，以提高处理效率。由并行机器学习算法实现方式提供的效率允许使用高容量网络并且使得能够对较大的数据集训练那些网络。

[0177] 机器学习算法是可以基于数据集合进行学习的算法。机器学习算法的实施例可以被设计成用于对数据集内的高级抽象进行建模。例如，可以使用图像识别算法来确定给定的输入属于若干类别中的哪个类别；给定输入，回归算法可以输出数字值；并且可以使用模式识别算法来生成经转换的文本或者执行文本到语音和/或语音识别。

[0178] 示例性类型的机器学习算法是神经网络。存在许多类型的神经网络；简单类型的神经网络是前馈网络。前馈网络可以被实现为在其中按层来布置节点的非循环图。典型地，前馈网络拓扑包括由至少一个隐藏层分离的输入层和输出层。隐藏层将由输入层接收的输入变换为对在输出层中生成输出有用的表示。网络节点经由边被完全连接到相邻层中的节点，但在每个层内的节点之间不存在边。在前馈网络的输入层的节点处接收的数据经由激活函数被传播(即，“前馈”)至输出层的节点，该激活函数基于分别与连接这些层的边中的每一条边相关联的系数(“权重”)来计算网络中每个连续层的节点的状态。取决于由正被执行的算法表示的特定模型，来自神经网络算法的输出可以采用各种形式。

[0179] 在可使用机器学习算法对特定问题建模之前，使用训练数据集来训练算法。训练神经网络涉及：选择网络拓扑；使用表示正由网络建模的问题的训练数据的集合；以及调整权重，直到网络模型针对训练数据集的所有实例都以最小误差执行。例如，在针对神经网络的有监督学习训练过程期间，由网络响应于表示训练数据集中的实例的输入而产生的输出与那个实例的“正确的”标记输出进行比较，计算表示输出与标记输出之间的差异的误差信号，并且随着误差信号通过网络的各层被向后传播，调整与连接相关联的权重以使那个误差最小化。当根据训练数据集的实例生成的输出中每个输出的误差被最小化时，网络被认为是“经训练的”。

[0180] 机器学习算法的准确度会显著地由用于训练算法的数据集的质量影响。训练过程可能是计算密集型的，并且在常规通用处理器上可能需要大量的时间。因此，使用并行处理硬件来训练许多类型的机器学习算法。这对于优化神经网络的训练是特别有用的，因为在调整神经网络中的系数时执行的计算本身自然地适于并行实现方式。具体地，许多机器学习算法和软件应用已被适配成利用通用图形处理设备内的并行处理硬件。

[0181] 图6是机器学习软件栈600的广义图。机器学习应用602可以被配置成使用训练数据集来训练神经网络或使用经训练的深度神经网络来实现机器智能。机器学习应用602可

包括用于神经网络的训练和推断功能和/或可用于在部署之前训练神经网络的专业软件。机器学习应用602可实现任何类型的机器智能，包括但不限于：图像识别、绘图和定位、自主导航、语音合成、医学成像或语言翻译。

[0182] 可以经由机器学习框架602来启用用于机器学习应用604的硬件加速。机器学习框架604可提供机器学习基元的库。机器学习基元是常由机器学习算法执行的基本操作。在没有机器学习框架604的情况下，将需要机器学习算法的开发者创建和优化与机器学习算法相关联的主计算逻辑，随后在开发新的并行处理器时重新优化计算逻辑。相反，机器学习应用可以被配置成使用由机器学习框架604提供的基元来执行必要的计算。示例性基元包括张量卷积、激励激活函数和池化，它们是在训练卷积神经网络(CNN)时执行的计算操作。机器学习框架604还可以提供基元以实现由许多机器学习算法执行的基本线性代数子程序，诸如，矩阵和向量操作。

[0183] 机器学习框架604可以处理从机器学习应用602接收的输入数据，并生成至计算框架606的适当输入。计算框架606可抽象出提供给GPGPU驱动器608的底层指令，以使得机器学习框架604能够经由GPGPU硬件610来利用硬件加速而无需机器学习框架604非常熟悉GPGPU硬件610的架构。另外，计算框架606可以跨各种类型和世代的GPGPU硬件610来启用用于机器学习框架604的硬件加速。

[0184] GPGPU机器学习加速

[0185] 图7图示根据实施例的通用图形处理单元700。在一个实施例中，通用处理单元(GPGPU)700可以被配置成在处理与训练深度神经网络相关联的类型的计算工作负荷时尤其高效。另外，GPGPU 700可以直接链接到GPGPU的其他实例以创建多GPU集群，从而改善尤其是深度神经网络的训练速度。

[0186] GPGPU 700包括用于启用与主机处理器的连接的主机接口702。在一个实施例中，主机接口702是PCI Express接口。然而，主机接口还可以是供应方专用的通信接口或通信结构。GPGPU 700从主机处理器接收命令，并且使用全局调度器704将与那些命令相关联的执行线程分发给计算集群的集合706A-706H。计算集群706A-706H共享高速缓存存储器708。高速缓存存储器708可以充当用于计算集群706A-706H内的高速缓存存储器的较高级别的高速缓存。

[0187] GPGPU 700包括经由存储器控制器的集合712A-712B与计算集群706A-706H耦合的存储器714A-714B。在各实施例中，存储器714A-714B可包括各种类型的存储器设备，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)，包括图形双倍数据速率(GDDR)存储器。在一个实施例中，存储器714A-714N还可包括3D堆叠式存储器，包括但不限于高带宽存储器(HBM)。

[0188] 在一个实施例中，计算集群706A-706H中的每个计算集群包括图形多处理器的集合，图形多处理器诸如图4A的图形多处理器400。这些计算集群的图形多处理器包括多种类型的整数逻辑单元和浮点逻辑单元，这些多种类型的整数逻辑单元和浮点逻辑单元可以在一定精度范围内执行包括适于机器学习计算的计算操作。例如，并且在一个实施例中，计算集群706A-706H中的每一个计算集群中的浮点单元的至少子集可以被配置成执行16位或32位浮点操作，而浮点单元的不同子集可以被配置成执行64位浮点操作。

[0189] GPGPU 700的多个实例可以被配置成作为计算集群进行操作。由计算集群用于同

步和数据交换的通信机制跨实施例而有所不同。在一个实施例中，GPGPU 700的多个实例通过主机接口702进行通信。在一个实施例中，GPGPU 700包括I/O中枢708，该I/O中枢709将GPGPU 700与GPU链路710耦合，该GPU链路启用至GPGPU的其他实例的直接连接。在一个实施例中，GPU链路710耦合至专用GPU-GPU桥接器，该GPU-GPU桥接器实现GPGPU 700的多个实例之间的通信和同步。在一个实施例中，GPU链路710与高速互连耦合，以将数据传输和接收至其他GPGPU或并行处理器。在一个实施例中，GPGPU 700的多个实例位于单独的数据处理系统中，并且经由网络设备进行通信，该网络设备可经由主机接口702来访问。在一个实施例中，附加于或替代于主机接口702，GPU链路710可以被配置成启用至主机处理器的连接。

[0190] 尽管GPGPU 700的所图示配置可以被配置成训练神经网络，但是一个实施例提供GPGPU 700的替代配置，该替代配置可以被配置成用于在高性能或低功率推断平台内的部署。在推断配置中，相对于训练配置，GPGPU 700包括计算集群706A-706H中的更少的计算集群。另外，与存储器714A-714B相关联的存储器技术在推断配置与训练配置之间可以不同。在一个实施例中，GPGPU 700的推断配置可以支持推断专用指令。例如，推断配置可提供对一个或多个8位整数点积指令的支持，这一个或多个8位整数点积指令通常在用于经部署的神经网络的推断操作期间使用。

[0191] 图8图示根据实施例的多GPU计算系统800。多GPU计算系统800可包括处理器802，该处理器802经由主机接口开关804耦合至多个GPGPU806A-806D。在一个实施例中，主机接口开关804是将处理器802耦合至PCI Express总线的PCI Express开关设备，处理器802可通过该PCI Express总线与GPGPU的集合806A-806D通信。多个GPGPU 806A-806D中的每一个可以是图7的GPGPU 700的实例。GPGPU 806A-806D可以经由高速点对点GPU-GPU链路的集合816进行互连。高速GPU-GPU链路可经由专用GPU链路连接至GPGPU 806A-806D中的每一个，该专用GPU链路诸如，如图7中的GPU链路710。P2P GPU链路816启用GPGPU 806A-806D中的每个GPGPU之间的直接通信，而无需通过处理器802连接到的主机接口总线进行的通信。利用定向到P2P GPU链路的GPU-GPU通信量，主机接口总线保持可用于系统存储器访问，或用于例如经由一个或多个网络设备与多GPU计算系统800的其他实例通信。虽然在所图示的实施例中，GPGPU 806A-806D经由主机接口开关804连接至处理器802，但是在一个实施例中，处理器802包括对P2P GPU链路816的直接支持并且可以直接连接至GPGPU 806A-806D。

[0192] 机器学习神经网络实现方式

[0193] 由本文中描述的实施例提供的计算架构可以被配置成执行特别适合于训练和部署用于机器学习的神经网络的这些类型的并行处理。可以将神经网络概括为具有图关系的函数网络。如本领域中所公知，存在机器学习中所使用的多种类型的神经网络实现方式。一种示例性类型的神经网络是如先前描述的前馈网络。

[0194] 第二示例性类型的神经网络是卷积神经网络(CNN)。CNN是用于处理具有已知的、网格状拓扑的数据(诸如，图像数据)的专业的前馈神经网络。因此，CNN通常用于计算视觉和图像识别应用，但是它们也可用于其他类型的模式识别，诸如，语音和语言处理。CNN输入层中的节点被组织为“过滤器”的集合(由视网膜中发现的感受野激发的特征检测器)，并且每个过滤器集合的输出被传播至网络的连续层中的节点。用于CNN的计算包括将卷积数学运算应用于每个过滤器以产生那个过滤器的输出。卷积是由两个函数执行以产生第三函数的一种专业的数学运算，该第三函数是两个原始函数中的一个的修改版本。在卷积网络术

语中，卷积的第一函数可以被称为输入，而第二个函数可以被称为卷积核。输出可被称为特征图。例如，至卷积层的输入可以是定义输入图像的各种颜色分量的多维数据数组。卷积核可以是多维参数数组，其中通过用于神经网络的训练过程来使参数适配。

[0195] 递归神经网络(RNN)是包括层之间的反馈连接的一类前馈神经网络。RNN通过跨神经网络的不同部分共享参数数据来启用对序列化数据进行建模。用于RNN的架构包括循环。这些循环表示变量的当前值在未来时刻对其自身值的影响，因为来自RNN的输出数据的至少一部分被用作反馈以用于处理序列中的后续输入。由于语言数据可被组成的可变本质，这个特征使RNN变得对语言处理特别有用。

[0196] 下文描述的图呈现了示例性前馈网络、CNN网络和RNN网络，并且描述了用于分别训练和部署那些类型的网络中的每一种的一般过程。将理解，这些描述就本文中描述的任何特定实施例而论是示例性且非限制性的，并且一般说来所图示的概念一般可应用于深度神经网络和机器学习技术。

[0197] 上文描述的示例性神经网络可以用于执行深度学习。深度学习是使用深度神经网络的机器学习。与仅包括单个隐藏层的浅层神经网络相反，深度学习中使用的深度神经网络是由多个隐藏层组成的人工神经网络。更深的神经网络通常训练起来更具计算密集性。然而，网络的附加隐藏层启用多步模式识别，该多步模式识别产生相对于浅层机器学习技术的减小的输出误差。

[0198] 深度学习中使用的深度神经网络典型地包括用于执行特征识别的前端网络，该前端网络耦合至后端网络，该后端网络表示数学模型，该数学模型可基于提供给模型的特征表示来执行操作(例如，对象分类、语音识别等)。深度学习使得在无需针对模型执行手工特征工程的情况下执行机器学习。相反，深度神经网络可以基于输入数据内的统计结构或相关性来学习特征。所学习的特征可以提供给数学模型，该数学模型可以将所检测的特征映射至输出。由网络使用的数学模型通常专用于待执行的特定任务，并且将使用不同的模型来执行不同的任务。

[0199] 一旦将神经网络结构化，就可以将学习模型应用于网络以将网络训练成执行特定任务。学习模型描述如何调整模型内的权重以减小网络的输出误差。反向传播误差是用于训练神经网络的常用方法。输入向量被呈现给网络以供处理。使用损失函数将网络的输出与期望输出进行比较，并且为输出层中的神经元中的每个神经元计算误差值。随后，向后传播误差值，直到每个神经元具有粗略地表示该神经元对原始输出的贡献的相关联的误差值。随后，网络可以使用算法(诸如，随机梯度下降算法)通过那些误差进行学习，以更新神经网络的权重。

[0200] 图9A-图9B图示示例性卷积神经网络。图9A图示CNN内的各个层。如图9A中所示，用于对图像处理进行建模的示例性CNN可接收输入902，该输入902描述输入图像的红、绿和蓝(RGB)分量。输入902可由多个卷积层(例如，卷积层904、卷积层906)处理。来自多个卷积层的输出任选地可由全连接层的集合908处理。如先前针对前馈网络所描述，全连接层中的神经元具有至前一层中的所有激活的完全连接。来自全连接层908的输出可用于从网络生成输出结果。可使用矩阵乘法而不是卷积来计算全连接层908内的激活。并非所有的CNN实现方式都利用全连接层908。例如，在一些实现方式中，卷积层906可生成CNN的输出。

[0201] 卷积层被稀疏地连接，这不同于在全连接层908中发现的传统神经网络配置。传统

神经网络层完全被连接，使得每个输出单元与每个输入单元相互作用。然而，如所图示，卷积层被稀疏地连接，因为感受野的卷积的输出(而不是感受野中的每个节点的相应状态值)被输入到后续层的节点。与卷积层相关联的核执行卷积操作，该卷积操作的输出被发送至下一层。在卷积层内执行的降维是使得CNN能够缩放以处理大图像的一个方面。

[0202] 图9B图示在CNN的卷积层内的示例性计算级。可以在卷积层914的三个级中处理至CNN的卷积层的输入912。这三个级可以包括卷积级916、检测器级918和池化级920。随后，卷积层914可将数据输出至连续的卷积层。网络的最终卷积层可以生成输出特征图数据或提供至全连接层的输入，例如以生成用于至CNN的输入的分类值。

[0203] 在卷积级916中，并行地执行若干个卷积，以产生线性激活的集合。卷积级916可以包括仿射变换，该仿射变换是可被指定为线性变换加平移的任何变换。仿射变换包括旋转、平移、缩放和这些变换的组合。卷积级计算连接至输入中的特定区域的函数(例如，神经元)的输出，这些特定区域可以被确定为与神经元相关联的局部区域。神经元计算神经元的权重与局部输入中神经元被连接到的区域的权重之间的点积。来自卷积级916的输出定义由卷积层914的连续级处理的线性激活的集合。

[0204] 线性激活可由检测器级918处理。在检测器级918中，每个线性激活由非线性激活函数处理。非线性激活函数增加整体网络的非线性性质，而不影响卷积层的感受野。可使用若干种类型的非线性激活函数。一个特定的类型是修正线性单元(ReLU)，其使用被定义为f(x)＝max(0,x)的激活函数，使得激活函数的阈值设定为零。

[0205] 池化级920使用池化函数，该池化函数用附近输出的概括统计来替换卷积层906的输出。池化函数可用于将平移不变性引入到神经网络中，使得至输入的小平移不改变经池化的输出。局部平移的不变性在其中特征在输入数据中的存在比该特征的精确位置更重要的场景下可以是有用的。可以在池化级920期间使用各种类型的池化函数，包括最大池化、平均池化和l2范数池化。另外，一些CNN实现方式不包括池化级。相反，此类实现方式代用相对于先前的卷积级具有增加的跨步的附加卷积级。

[0206] 随后，来自卷积层914的输出可由下一层922处理。下一层922可以是一个附加的卷积层或是全连接层908中的一个层。例如，图9A的第一卷积层904可以输出至第二卷积层906，而第二卷积层可以输出至全连接层908中的第一层。

[0207] 图10图示示例性递归神经网络1000。在递归神经网络(RNN)中，网络的先前状态影响网络的当前状态的输出。可以使用各种函数以各种方式来建立RNN。RNN的使用通常围绕使用数学模型以基于先前的输入序列来预测未来。例如，RNN可用于执行统计语言建模以在给定先前的字序列的情况下预测即将来临的字。可以将所图示的RNN 1000描述为具有以下各项：输入层1002，其接收输入向量；隐藏层1004，用于实现递归函数；反馈机制1005，用于启用先前状态的“记忆”；以及输出层1006，用于输出结果。RNN 1000基于时间步长来操作。RNN在给定时间步长的状态经由反馈机制1005基于先前的时间步长被影响。针对给定的时间步长，由先前状态和在当前时间步长的输入来限定隐藏层1004的状态。在第一时间步长的初始输入(x_1)可由隐藏层1004处理。第二输入(x_2)可由隐藏层1004使用在处理初始输入(x_1)期间所确定的状态信息来处理。给定的状态可被计算为$s_t＝f(Ux_t+Ws_{(t-1)})$，其中，U和W是参数矩阵。该函数f一般是非线性的，诸如，双曲正切函数(Tanh)或修正函数f(x)＝max(0,x)的变体。然而，隐藏层1004中使用的特定数学函数可以取决于RNN 1000的特定实现方

式细节而有所不同。

[0208] 除所描述的基本CNN网络和RNN网络之外，还可实现那些网络的变化。一个示例RNN变体是长短期记忆(LSTM)RNN。LSTM RNN能够学习对于处理较长的语言序列可能必要的长期依赖性。CNN的变体是卷积深度信念网络，该卷积深度信念网络具有类似于CNN的结构并且以类似于深度信念网络的方式被训练。深度信念网络(DBN)是由随机性(随机)变量的多个层组成的生成式神经网络。可以使用贪婪式无监督学习来逐层训练DBN。随后，DBN的所学习的权重可以用于通过确定用于神经网络的最佳初始权重集合来提供预训练神经网络。

[0209] 图11图示深度神经网络的训练和部署。一旦已针对任务将给定的网络结构化，就使用训练数据集1102来训练神经网络。已开发出各种训练框架1104来启用对训练过程的硬件加速。例如，图6的机器学习框架604可被配置为训练框架604。训练框架604可以跟未经训练的神经网络1106挂钩，并且使得能够使用本文中描述的并行处理资源来训练未经训练的神经网以生成经训练的神经网1108。

[0210] 为了开始训练过程，可随机地或通过使用深度信念网络进行预训练来选择初始权重。随后，以有监督或无监督方式执行训练循环。

[0211] 有监督学习是在其中将训练被执行为中介操作的学习方法，诸如，当训练数据集1102包括与输入的期望输出配对的该输入时，或在训练数据集包括具有已知输出的输入并且神经网络的输出被手动分级的情况下。网络处理输入，并且将所得的输出与预期输出或期望输出的集合进行比较。随后，通过系统反向传播误差。训练框架1104可以进行调整，以调整控制未经训练的神经网络1106的权重。训练框架1104可以提供工具以监测未经训练的神经网络1106在多好地收敛于适合基于已知的输入数据生成正确的答案的模型。随着网络的权重被调整以改良由神经网络生成的输出，训练过程反复地发生。训练过程可以继续，直到神经网络达到与经训练的神经网1108相关联的统计上期望的准确度。随后，可以部署经训练的神经网络1108以实现任何数量的机器学习操作以基于新数据1112的输入来生成推断结果1114。

[0212] 无监督学习是在其中网络试图使用未标记数据来训练其自身的学习方法。因此，针对无监督学习，训练数据集1102将包括不具有任何相关联的输出数据的输入数据。未经训练的神经网络1106可以学习未标记输入内的分组，并且可以确定单独的输入如何与整个数据集相关。无监督训练可以用于生成自组织图，该自组织图是能够执行有助于数据降维的操作的一类的经训练的神经网络1108。无监督训练还可以用于执行异常检测，该异常检测允许识别输入数据集中偏离数据正常模式的数据点。

[0213] 还可采用有监督训练和无监督训练的变体。半监督学习是在其中在训练数据集1102中包括具有相同分布的经标记数据和未标记数据的混合的技术。渐进式学习是有监督学习的变体，其中连续地使用输入数据以进一步训练模型。渐进式学习使得经训练的神经网络1108能够适配于新数据1112，而不忘记在初始训练期间根植在网络内的知识。

[0214] 无论是有监督还是无监督的，用于特别深的神经网络的训练过程对于单个计算节点可能是过于计算密集的。可以使用计算节点的分布式网络而不是使用单个计算节点来加速训练过程。

[0215] 图12是图示分布学习的框图。分布式学习是使用多个分布式计算节点来执行神经网络的有监督训练或无监督训练的训练模型。分布式计算节点可以各自包括一个或多个主

机处理器以及通用处理节点中的一个或多个通用处理节点，该通用处理节点诸如，如图7中的高度并行的通用图形处理单元700。如所图示，分布式学习可被执行模型并行性1202、数据并行性1204、或模型并行性和数据并行性1204的组合。

[0216] 在模型并行性1202中，分布式系统中的不同计算节点可以针对单个网络的不同部分执行训练计算。例如，可以由分布式系统的不同的处理节点训练神经网络的每个层。模型并行性的益处包括缩放到特别大的模型的能力。分割与神经网络的不同层相关联的计算使得能够训练非常大的神经网络，其中所有层的权重将不拟合到单个计算节点的记忆中。在一些实例中，模型并行性在执行大型神经网络的无监督训练中可以是特别有用的。

[0217] 在数据并行性1204中，分布式网络的不同节点具有模型的完整实例，并且每个节点接收数据的不同部分。随后，组合来自不同节点的结果。虽然用于数据并行性的不同方法是可能的，但是数据并行训练方法全都需要组合结果并使每个节点之间的模型参数同步的技术。用于组合数据的示例性方法包括参数求平均和基于更新的数据并行性。参数求平均对训练数据的子集训练每个节点，并且将全局参数(例如，权重、偏差)设定为来自每个节点的参数的平均值。参数求平均使用维持参数数据的中央参数服务器。基于更新的数据并行性类似于参数求平均，例外在于，传递对模型的更新而不是将来自节点的参数传送到参数服务器。另外，能以分散的方式执行基于更新的数据并行性，其中更新被压缩并且在节点之间传送。

[0218] 可以例如在其中每个计算节点包括多个GPU的分布式系统中实现组合式模型和数据并行性1206。每个节点可以具有模型的完整实例，其中每个节点内的多个单独的GPU用于训练模型的不同部分。

[0219] 分布训练相对于在单个机器上的训练具有增加的开销。然而，本文中描述的并行处理器和GPGPU可以各自实现各种技术以用于减少分布训练的开销，这些技术包括用于启用高带宽GPU-GPU数据传送和加速的远程数据同步的技术。

[0220] 示例性机器学习应用

[0221] 可以应用机器学习以解决各种技术问题，包括但不限于计算机视觉、自主驾驶和导航、语音识别以及语言处理。计算机视觉在传统上已是机器学习应用的最活跃研究领域之一。计算机视觉的应用范围为从重现人类视觉能力(诸如，识别人脸)到创建新类别的视觉能力。例如，计算机视觉应用可以被配置成从视频中可见的物体中诱发的振动来识别声波。并行处理器加速的机器学习使得能够使用比先前可行的训练数据集明显大得多的训练数据集来训练计算机视觉应用，并且使得能够使用低功率并行处理器来部署推断系统。

[0222] 并行处理器加速的机器学习具有自主驾驶应用，包括车道和道路标志识别、障碍回避、导航和驾驶控制。加速的机器学习技术可以用于基于定义对特定训练输入的适当响应的数据集来训练驾驶模型。本文中描述的并行处理器能够对用于自主驾驶解决方案的日益复杂的神经网络进行快速训练，并且能够将低功率推断处理器部署在适合于集成到自主车辆中的移动平台中。

[0223] 并行处理器加速的深度神经网络已实现用于自动语音识别(ASR)的机器学习方法。ASR包括创建在给定的输入声序列的情况下计算最可能的语言序列的函数。使用深度神经网络的加速的机器学习已实现代替先前用于ASR的隐马尔可夫模型(HMM)和高斯混合模型(GMM)。

[0224] 并行处理器加速的机器学习还可以用于加速自然语言处理。自动学习程序可以利用统计推断算法以产生对于有误差的或不熟悉的输入强健的模型。示例性自然语言处理器应用包括人类语言之间的自动机器翻译。

[0225] 可以将用于机器学习的并行处理平台划分为训练平台和部署平台。训练平台通常是高度并行的，并且包括用于加速多GPU单节点训练和多节点多GPU训练的优化。适合于训练的示例性并行处理器包括图7的通用图形处理单元700和图8的多GPU计算系统800。相反，部署的机器学习平台通常包括适合于在诸如相机、自主机器人和自主车辆的产品中使用的低功率并行处理器。

[0226] 图13图示适合于使用经训练的模型执行推断的示例性推断芯片上系统(SOC)1300。SOC 1300可以集成多个处理组件，包括媒体处理器1302、视觉处理器1304、GPGPU 1306和多核处理器1308。SOC 1300可附加地包括芯片上存储器1305，该芯片上存储器1305可启用可由处理组件中的每个处理组件访问的共享芯片上数据池。处理组件可以针对低功率操作被优化，以使得能够部署到各种机器学习平台，包括自主车辆和自主机器人。例如，可以将SOC 1300的一种实现方式用作用于自主车辆的主控制系统的一部分。在SOC 1300被配置成在自主车辆中使用的情况下，SOC被设计和配置成用于符合部署管辖权的相关功能安全标准。

[0227] 在操作期间，媒体处理器1302和视觉处理器1304可以一致地工作以加速计算机视觉操作。媒体处理器1302可以使得能够对多个高分辨率(例如，4K、8K)视频流进行低等待时间解码。可以将经解码的视频流写入到芯片上存储器1305中的缓冲器。随后，视觉处理器1304可以解析经解码的视频，并且对经解码视频的帧执行初步处理操作以准备使用经训练的图像识别模型来处理帧。例如，视觉处理器1304可以加速用于CNN的卷积操作，该CNN用于对高分辨率视频数据执行图像识别，而后端模型计算由GPGPU 1306执行。

[0228] 多核处理器1308可以包括控制逻辑，该控制逻辑用于辅助数据传送的定序和同步以及由媒体处理器1302和视觉处理器1304执行的共享存储器操作。多核处理器1308还可以充当应用处理器，以执行可利用GPGPU 1306的推断计算能力的软件应用。例如，可以于在多核处理器1308上执行的软件中实现导航和驾驶逻辑的至少一部分。此类软件可以直接将计算工作负载发布至GPGPU 1306，或者可以将计算工作负载发布至多核处理器1308，该多核处理器可以将那些操作的至少一部分转移到GPGPU 1306。

[0229] GPGPU 1306可以包括计算集群，诸如，通用图形处理单元700内的计算集群706A-706H的低功率配置。GPGPU 1306内的计算集群可支持专门经优化以对经训练的神经网络执行推断计算的指令。例如，GPGPU 1306可支持用于执行低精度计算的指令，低精度计算诸如8位和4位整数向量操作。

[0230] 系统概览

[0231] 图14是根据实施例的处理系统1400的框图。系统1400可被用在以下各项中：单处理器台式机系统、多处理器工作站系统、或具有大量处理器1402或处理器核1407的服务器系统。在一个实施例中，系统1400是被并入在芯片上系统(SoC)集成电路内的处理平台，该芯片上系统(SoC)集成电路用于在移动设备、手持式设备或嵌入式设备中使用，诸如，用于在具有至局域网或广域网的有线或无线连接性的物联网(IoT)设备内使用。

[0232] 在一个实施例中，系统1400可包括以下各项，可与以下各项耦合，或可并入在以下

各项内:基于服务器的游戏平台、包括游戏和媒体控制台的游戏控制台、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1400是移动电话、智能电话、平板计算设备或移动互联网连接的设备(诸如,具有低内部存储容量的笔记本)的部分。处理系统1400也可包括以下各项,与以下各项耦合,或被集成在以下各项内:可穿戴设备,诸如,智能手表可穿戴设备;利用增强现实(AR)或虚拟现实(VR)特征来增强以提供视觉、音频或触觉输出来补充现实世界视觉、音频或触觉体验或以其他方式提供文本、音频、图形、视频、全息图像或视频、或触觉反馈的智能眼镜或服装;其他增强现实(AR)设备;或其他虚拟现实(VR)设备。在一些实施例中,处理系统1400包括电视机或机顶盒设备,或者是电视机或机顶盒设备的部分。在一个实施例中,系统1400可包括自动驾驶运载工具,与自动驾驶运载工具耦合,或集成在自动驾驶运载工具中,该自动驾驶运载工具诸如,公共汽车、拖拉机拖车、汽车、电机或电力循环、飞机或滑翔机(或其任何组合)。自动驾驶运载工具可使用系统1400来处理在该运载工具周围感测到的环境。

[0233] 在一些实施例中,一个或多个处理器1402各自都包括用于处理器指令的一个或多个处理器核1407,这些指令当被执行时,执行用于系统或用户软件的操作。在一些实施例中,一个或多个处理器核1407中的至少一个被配置成处理特定的指令集1409。在一些实施例中,指令集1409可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。一个或多个处理器核1407可处理不同的指令集1409,不同的指令集1409可包括用于促进对其他指令集的仿真的指令。处理器核1407也可包括其他处理设备,诸如,数字信号处理器(DSP)。

[0234] 在一些实施例中,处理器1402包括高速缓存存储器1404。取决于架构,处理器1402可具有单个内部高速缓存或多级的内部高速缓存。在一些实施例中,高速缓存存储器在处理器1402的各种组件之间被共享。在一些实施例中,处理器1402也使用外部高速缓存(例如,第3级(L3)高速缓存或末级高速缓存(LLC))(未示出),可使用已知的高速缓存一致性技术在处理器核1407之间共享该外部高速缓存。寄存器堆1406可附加地被包括在处理器1402中,并且寄存器堆106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、状态寄存器以及指令指针寄存器)。一些寄存器可以是通用寄存器,而其他寄存器可专用于处理器1402的设计。

[0235] 在一些实施例中,一个或多个处理器1402与一个或多个接口总线1410耦合,以在处理器1402与系统1400中的其他组件之间传输通信信号,诸如,地址、数据、或控制信号。在一个实施例中,接口总线1410可以是处理器总线,诸如,直接媒体接口(DMI)总线的某个版本。然而,处理器总线不限于DMI总线,并且可包括一个或多个外围组件互连总线(例如,PCI、PCI express)、存储器总线或其他类型的接口总线。在一个实施例中,(多个)处理器1402包括集成存储器控制器1416和平台控制器中枢1430。存储器控制器1416促进存储器设备与系统1400的其他组件之间的通信,而平台控制器中枢(PCH)1430提供经由本地I/O总线至I/O设备的连接。

[0236] 存储器设备1420可以是动态随机存取存储器(DRAM)设备、静态随机存取存储器(SRAM)设备、闪存设备、相变存储器设备、或具有适当的性能以充当进程存储器的某个其他存储器设备。在一个实施例中,存储器设备1420可以作为用于系统1400的系统存储器来操作,以存储数据1422和指令1421供在一个或多个处理器1402执行应用或进程时使用。存储

器控制器1416也与任选的外部图形处理器1418耦合，该任选的外部图形处理器1418可与处理器1402中的一个或多个图形处理器1408通信以执行图形操作和媒体操作。在一些实施例中，可由加速器1412辅助图形操作、媒体操作或计算操作，该加速器1412是可被配置用于执行专业的图形操作、媒体操作或计算操作的集合的协处理器。例如，在一个实施例中，加速器1412是用于优化机器学习或计算操作的矩阵乘法加速器。在一个实施例中，加速器1412是光线追踪加速器，该光线追踪加速器可用于与图形处理器1408一致地执行光线追踪操作。在一个实施例中，可替代加速器1412使用外部加速器1419，或可与加速器1412一致地使用外部加速器1419。

[0237] 在一些实施例中，显示设备1411可以连接至(多个)处理器1402。显示设备1411可以是以下各项中的一项或多项：内部显示设备，如在移动电子设备或膝上型设备中；或经由显示接口(例如，显示端口等)附接的外部显示设备。在一个实施例中，显示设备1411可以是头戴式显示器(HMD)，诸如，用于在虚拟现实(VR)应用或增强现实(AR)应用中使用的立体显示设备。

[0238] 在一些实施例中，平台控制器中枢1430使外围设备能够经由高速I/O总线而连接至存储器设备1420和处理器1402。I/O外围设备包括但不限于音频控制器1446、网络控制器1434、固件接口1428、无线收发器1426、触摸传感器1425、数据存储设备1424(例如，非易失性存储器、易失性存储器、硬盘驱动器、闪存、NAND、3D NAND、3D XPoint等)。数据存储设备1424可以经由存储接口(例如，SATA)或经由如外围组件互连总线(例如，PCI、PCI express)之类的外围总线来进行连接。触摸传感器1425可以包括触摸屏传感器、压力传感器、或指纹传感器。无线收发器1426可以是Wi-Fi收发器、蓝牙收发器、或移动网络收发器，该移动网络收发器诸如3G、4G、5G或长期演进(LTE)收发器。固件接口1428使得能够与系统固件进行通信，并且可以例如是统一可扩展固件接口(UEFI)。网络控制器1434可启用到有线网络的网络连接。在一些实施例中，高性能网络控制器(未示出)与接口总线1410耦合。在一个实施例中，音频控制器1446是多声道高清音频控制器。在一个实施例中，系统1400包括用于将传统(例如，个人系统2(PS/2))设备耦合至系统的任选的传统I/O控制器1440。平台控制器中枢1430还可以连接至一个或多个通用串行总线(USB)控制器1442连接输入设备，诸如，键盘和鼠标1443组合、相机1444、或其他USB输入设备。

[0239] 将会理解，所示的系统1400是示例性的而非限制性的，因为也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，存储器控制器1416和平台控制器中枢1430的实例可以集成到分立的外部图形处理器中，该分立的外部图形处理器诸如外部图形处理器1418。在一个实施例中，平台控制器中枢1430和/或存储器控制器1416可以在一个或多个处理器1402外部。例如，系统1400可包括外部存储器控制器1416和平台控制器中枢1430，该外部存储器控制器1416和平台控制器中枢1430可以被配置为在与(多个)处理器1402通信的系统芯片组内的存储器控制器中枢和外围控制器中枢。

[0240] 例如，可使用电路板(“橇板(sled)”)，在该电路板上被放置的组件(诸如，CPU、存储器和其他组件)经设计以实现提升的热性能。在一些示例中，诸如处理器之类的处理组件位于橇板的顶侧上，而诸如DIMM之类的附近存储器位于橇板的底侧上。作为由该设计提供的增强的气流的结果，组件能以比典型系统更高的频率和功率等级来操作，由此增加性能。此外，橇板配置成盲配机架中的功率和数据通信线缆，由此增强它们被快速地移除、升级、

重新安装和/或替换的能力。类似地，位于橇板上的各个组件(诸如，处理器、加速器、存储器和数据存储设备)由于它们相对于彼此增加的间距而被配置成易于升级。在说明性实施例中，组件附加地包括用于证明它们的真实性的硬件认证特征。

[0241] 数据中心可利用支持多个其他网络架构的单个网络结构(“结构”)，多个其他网络架构包括以太网和全方位路径。橇板可经由光纤耦合至交换机，这提供比典型的双绞线布线(例如，5类、5e类、6类等)更高的带宽和更低的等待时间。由于高带宽、低等待时间的互连和网络架构，数据中心在使用中可集中在物理上解散的诸如存储器、加速器(例如，GPU、图形加速器、FPGA、ASIC、神经网络和/或人工智能加速器等)和数据存储驱动器之类的资源，并且根据需要将它们提供给计算资源(例如，处理器)，从而使计算资源能够就好像被集中的资源在本地那样访问这些被集中的资源。

[0242] 功率供应或功率源可将电压和/或电流提供给系统1400或本文中描述的任何组件或系统。在一个示例中，功率供应包括用于插入到墙壁插座中的AC-DC(交流-直流)适配器。此类AC功率可以是可再生能量(例如，太阳能)功率源。在一个示例中，功率源包括DC功率源，诸如，外部AC-DC转换器。在一个示例中，功率源或功率供应包括用于通过接近充电场来充电的无线充电硬件。在一个示例中，功率源可包括内部电池、交流供应、基于动作的功率供应、太阳能功率供应、或燃料电池源。

[0243] 图15A-图15C图示由本文中描述的实施例提供的计算系统和图形处理器。图15A-图15C的具有与本文中的任何其他附图的元件相同的附图标记(或名称)的那些元件能以类似于本文中其他地方描述的任何方式操作或起作用，但不限于此。

[0244] 图15A是处理器1500的实施例的框图，该处理器1500具有一个或多个处理器核1502A-1502N、集成存储器控制器1514以及集成图形处理器1508。处理器1500可包括附加的核，这些附加的核多至由虚线框表示的附加核1502N并包括由虚线框表示的附加核1502N。处理器核1502A-1502N中的每一个包括一个或多个内部高速缓存单元1504A-1504N。在一些实施例中，每一个处理器核也具有对一个或多个共享高速缓存单元1506的访问权。内部高速缓存单元1504A-1504N和共享高速缓存单元1506表示处理器1500内的高速缓存存储器层级结构。高速缓存存储器层级结构可包括每个处理器核内的至少一个级别的指令和数据高速缓存以及一级或多级共享的中级高速缓存，诸如，第2级(L2)、第3级(L3)、第4级(L4)、或其他级别的高速缓存，其中，在外部存储器之前的最高级别的高速缓存被分类为LLC。在一些实施例中，高速缓存一致性逻辑维持各高速缓存单元1506与1504A-1504N之间的一致性。

[0245] 在一些实施例中，处理器1500还可包括一个或多个总线控制器单元的集合1616和系统代理核1510。一个或多个总线控制器单元1616管理外围总线的集合，诸如，一个或多个PCI总线或PCI Express总线。系统代理核1510提供对各处理器组件的管理功能。在一些实施例中，系统代理核1510包括用于管理对各种外部存储器设备(未示出)的访问的一个或多个集成存储器控制器1514。

[0246] 在一些实施例中，处理器核1502A-1502N中的一个或多个处理器核包括对同步多线程的支持。在此类实施例中，系统代理核1510包括用于在多线程处理期间协调并操作核1502A-1502N的组件。系统代理核1510可附加地包括功率控制单元(PCU)，该功率控制单元包括用于调节处理器核1502A-1502N和图形处理器1508的功率状态的逻辑和组件。

[0247] 在一些实施例中，处理器1500附加地包括用于执行图形处理操作的图形处理器

1508。在一些实施例中，图形处理器1508与共享高速缓存单元的集合1506以及与系统代理核1510耦合，该系统代理核1510包括一个或多个集成存储器控制器1514。在一些实施例中，系统代理核1510还包括用于将图形处理器输出驱动到一个或多个经耦合的显示器的显示控制器1511。在一些实施例中，显示控制器1511还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1508内。

[0248] 在一些实施例中，基于环的互连单元1512用于耦合处理器1500的内部组件。然而，可以使用替代的互连单元，诸如，点到点互连、交换式互连、或其他技术，包括本领域中公知的技术。在一些实施例中，图形处理器1508经由I/O链路1513与环形互连1512耦合。

[0249] 示例性I/O链路1513表示多个各种各样的I/O互连中的至少一种，包括促进各处理器组件与高性能嵌入式存储器模块1518(诸如，eDRAM模块)之间的通信的封装上I/O互连。在一些实施例中，处理器核1502A-1502N中的每个处理器核以及图形处理器1508可将嵌入式存储器模块1518用作共享的末级高速缓存。

[0250] 在一些实施例中，处理器核1502A-1502N是执行相同指令集架构的同构核。在另一实施例中，处理器核1502A-1502N在指令集架构(ISA)方面是异构的，其中，处理器核1502A-1502N中的一个或多个执行第一指令集，而其他核中的至少一个执行第一指令集的子集或不同的指令集。在一个实施例中，处理器核1502A-1502N在微架构方面是异构的，其中，具有相对较高功耗的一个或多个核与具有较低功耗的一个或多个功率核耦合。在一个事实中，处理器核1502A-1502N在计算能力方面是异构的。此外，处理器1500可实现在一个或多个芯片上，或者除其他组件之外还被实现为具有所图示的组件的SoC集成电路。

[0251] 图15B是根据本文中所描述的一些实施例的图形处理器核1519的硬件逻辑的框图。图15B的具有与本文中的任何其他附图的元件相同的附图标记(或名称)的那些元件能以类似于本文中其他地方描述的任何方式操作或起作用，但不限于此。图形处理器核1519(有时称为核切片)可以是模块化图形处理器内的一个或多个图形核。图形处理器核1519的示例是一个图形核切片，并且基于目标功率包络和性能包络，如本文中所描述的图形处理器可以包括多个图形核切片。每个图形处理器核1519可包括固定功能块1530，该固定功能块1530与多个子核1521A-1521F(也称为子切片)耦合，多个子核1521A-1521F包括模块化的通用和固定功能逻辑的块。

[0252] 在一些实施例中，固定功能块1530包括几何/固定功能流水线1531，该几何/固定功能流水线1531例如在较低性能和/或较低功率的图形处理器实现中可由图形处理器核1519中的所有子核共享。在各实施例中，几何/固定功能流水线1531包括3D固定功能流水线(例如，如在下文描述的图16中的3D流水线1612)、视频前端单元、线程生成器和线程分派器、以及统一返回缓冲器管理器，该统一返回缓冲器管理器统一返回缓冲器(例如，如下文所描述的在图17中的统一返回缓冲器1718)。

[0253] 在一个实施例中，固定功能块1530还包括图形SoC接口1532、图形微控制器1533和媒体流水线1534。图形SoC接口1532提供图形处理器核1519与芯片上系统集成电路内的其他处理器核之间的接口。图形微控制器1533是可配置成管理图形处理器核1519的各种功能的可编程子处理器，这些功能包括线程分派、调度和抢占。媒体流水线1534(例如，图16和图17的媒体流水线1616)包括用于促进对包括图像数据和视频数据的多媒体数据进行解码、编码、预处理和/或后处理的逻辑。媒体流水线1534经由对子核1521A-1521F内的计算或采

样逻辑的请求来实现媒体操作。

[0254] 在一个实施例中,SoC接口1532使图形处理器核1519能够与通用应用处理器核(例如,CPU)和/或SoC内的其他组件进行通信,其他组件包括诸如共享的末级高速缓存存储器的存储器层级结构元件、系统RAM、和/或嵌入式芯片上或封装上DRAM。SoC接口1532还可启用与SoC内的诸如相机成像流水线的固定功能设备的通信,并且启用全局存储器原子性的使用和/或实现全局存储器原子性,该全局存储器原子性可在图形处理器核1519与SoC内的CPU之间被共享。SoC接口1532还可实现针对图形处理器核1519的功率管理控制,并且启用图形核1519的时钟域与SoC内的其他时钟域之间的接口。在一个实施例中,SoC接口1532使得能够从命令流转化器和全局线程分派器接收命令缓冲器,该命令流转化器和全局线程分派器被配置成将命令和指令提供给图形处理器内的一个或多个图形核中的每一个图形核。当媒体操作将要执行时,这些命令和指令可以被分派给媒体流水线1534,或者当图形处理操作将要执行时,这些命令和指令可以被分派给几何和固定功能流水线(例如,几何和固定功能流水线1531、几何和固定功能流水线1537)。

[0255] 图形微控制器1533可被配置成执行针对图形处理器核1519的各种调度任务和管理任务。在一个实施例中,图形微控制器1533可对子核1521A-1521F内的执行单元(EU)阵列1533A-1522F、1524A-1524F内的各个图形并行引擎执行图形和/或计算工作负载调度。在该调度模型中,在包括图形处理器核1519的SoC的CPU核上执行的主机软件可以经由多个图形处理器门铃(doorbell)中的一个图形处理器门铃来提交工作负载,这调用了对适当的图形引擎的调度操作。调度操作包括:确定接下来要运行哪个工作负载,将工作负载提交到命令流转化器,抢占在引擎上运行的现有工作负载,监测工作负载的进度,以及当工作负载完成时通知主机软件。在一个实施例中,图形微控制器1533还可促进图形处理器核1519的低功率或空闲状态,从而向图形处理器核1519提供独立于操作系统和/或系统上的图形驱动器软件跨低功率状态转变来保存和恢复图形处理器核1519内的寄存器的能力。

[0256] 图形处理器核1519可具有多于或少于所图示的子核1521A-1521F,多达N个模块化子核。对于每组N个子核,图形处理器核1519还可包括共享功能逻辑1535、共享和/或高速缓存存储器1536、几何/固定功能流水线1537、以及用于加速各种图形和计算处理操作的附加的固定功能逻辑1538。共享功能逻辑1535可以包括与可由图形处理器核1519内的每N个子核共享的、与图17的共享功能逻辑1720(例如,采样器逻辑、数学逻辑、和/或线程间通信逻辑)相关联的逻辑单元。共享和/或高速缓存存储器1536可以是用于图形处理器核1519内的N个子核的集合1521A-1521F的末级高速缓存,并且还可以充当可由多个子核访问的共享存储器。几何/固定功能流水线1537而不是几何/固定功能流水线1531可被包括在固定功能块1530内,并且几何/固定功能流水线1537可包括相同或类似的逻辑单元。

[0257] 在一个实施例中,图形处理器核1519包括附加的固定功能逻辑1538,该附加的固定功能逻辑1538可包括供由图形处理器核1519使用的各种固定功能加速逻辑。在一个实施例中,附加的固定功能逻辑1538包括供在仅位置着色中使用的附加的几何流水线。在仅位置着色中,存在两个几何流水线:几何/固定功能流水线1538、1531内的完全几何流水线;以及剔除流水线,其是可被包括在附加的固定功能逻辑1538内的附加的几何流水线。在一个实施例中,剔除流水线是完全几何流水线的精简版本。完全流水线和剔除流水线可以执行同一应用的不同实例,每个实例具有单独的上下文。仅位置着色可以隐藏被丢弃三角形的

长剔除运行，从而在一些实例中使得能够更早地完成着色。例如并且在一个实施例中，附加的固定功能逻辑1538内的剔除流水线逻辑可以与主应用并行地执行位置着色器，并且通常比完全流水线更快地生成关键结果，因为剔除流水线仅取出顶点的位置属性并对顶点的位置属性进行着色，而不向帧缓冲器执行对像素的栅格化和渲染。剔除流水线可以使用所生成的关键结果来计算所有三角形的可见性信息，而无需考虑那些三角形是否被剔除。完全流水线(其在本实例中可以被称为重放(replay)流水线)可以消耗该可见性信息以跳过被剔除的三角形，从而仅对最终被传递到栅格化阶段的可见的三角形进行着色。

[0258] 在一个实施例中，附加的固定功能逻辑1538还可包括机器学习加速逻辑，诸如，固定功能矩阵乘法逻辑，该机器学习加速逻辑用于包括针对机器学习训练或推断的优化的实现方式。

[0259] 在每个图形子核1521A-1521F内包括可用于响应于由图形流水线、媒体流水线、或着色器程序作出的请求而执行图形操作、媒体操作和计算操作的执行资源的集合。图形子核1521A-1521F包括：多个EU阵列1522A-1522F、1524A-1524F；线程分派和线程间通信(TD/IC)逻辑1523A-1523F；3D(例如，纹理)采样器1525A-15255F；媒体采样器1506A-1506F；着色器处理器1527A-1527F；以及共享的本地存储器(SLM)1528A-1528F。EU阵列1522A-1522F、1524A-1524F各自包括多个执行单元，这些执行单元是能够执行浮点和整数/定点逻辑操作以服务于图形操作、媒体操作或计算操作(包括图形程序、媒体程序或计算着色器程序)的通用图形处理单元。TD/IC逻辑1523A-1523F执行针对子核内的执行单元的本地线程分派和线程控制操作，并且促进在子核的执行单元上执行的线程之间的通信。3D采样器1525A-1525F可将纹理或其他3D图形相关的数据读取到存储器中。3D采样器可以基于所配置的样本状态以及与给定纹理相关联的纹理格式以不同方式读取纹理数据。媒体采样器1506A-1506F可基于与媒体数据相关联的类型和格式来执行类似的读取操作。在一个实施例中，每个图形子核1521A-1521F可以交替地包括统一3D和媒体采样器。在子核1521A-1521F中的每一个子核内的执行单元上执行的线程可利用每个子核内的共享的本地存储器1528A-1528F，以使在线程组内执行的线程能够使用芯片上存储器的公共池来执行。

[0260] 图15C是根据本文中描述的实施例的通用图形处理器单元(GPGPU)1570的框图，该GPGPU 1570可被配置为图形处理器和/或计算加速器。GPGPU 1570可经由一个或多个系统和/或存储器总线来与主机处理器(例如，一个或多个CPU 1546)和存储器1571、1572互连。在一个实施例中，存储器1571是可与一个或多个CPU 1546进行共享的系统存储器，而存储器1572是专用于GPGPU 1570的设备存储器。在一个实施例中，GPGPU 1570和设备存储器1572内的组件可被映射到可由一个或多个CPU 1546访问的存储器地址。可经由存储器控制器1568来促进对存储器1571和1572的访问。在一个实施例中，存储器控制器1568包括内部直接存储器访问(DMA)控制器1569，或可包括用于执行否则将由DMA控制器执行的操作的逻辑。

[0261] GPGPU 1570包括多个高速缓存存储器，包括L2高速缓存1553、L1高速缓存1554、指令高速缓存1555、以及共享存储器1556，该共享存储器1556的至少部分也可被分区为高速缓存存储器。GPGPU 1570还包括多个计算单元1560A-1560N。每个计算单元1560A-1560N包括向量寄存器的集合1561、标量寄存器的集合1562、向量逻辑单元的集合1563、以及标量逻辑单元的集合1564。计算单元1560A-1560N还可包括本地共享存储器1565和程序计数器

1566。计算单元1560A-1560N可与常量高速缓存1567耦合，该常量高速缓存1567可用于存储常量数据，常量数据是在GPGPU 1570上执行的核程序或着色器程序的运行期间将不改变的数据。在一个实施例中，常量高速缓存1567是标量数据高速缓存，并且经高速缓存的数据可被直接取出到标量寄存器1562中。

[0262] 在操作期间，一个或多个CPU 1546可将命令写入到GPGPU 1570中的寄存器中，或写入到GPGPU 1570中的、已经被映射到可访问地址空间的存储器中。命令处理器1557可从寄存器或存储器读取命令，并且确定将如何在GPGPU 1570内处理那些命令。随后可使用线程分派器1558来将线程分派到计算单元1560A-1560N以执行那些命令。每个计算单元1560A-1560N可独立于其他计算单元来执行线程。此外，每个计算单元1560A-1560N可被独立地配置成用于有条件计算，并且可有条件地将计算的结果输出到存储器。当所提交的命令完成时，命令处理器1557可中断一个或多个CPU 1546。

[0263] 图16A-图16C图示由本文中描述的实施例提供的附加的图形处理器和计算加速器架构的框图。图16A-图16C的具有与本文中的任何其他附图的元件相同的附图标记(或名称)的那些元件能以类似于本文中其他地方描述的任何方式操作或起作用，但不限于此。

[0264] 图16A是图形处理器1600的框图，该图形处理器1600可以是分立的图形处理单元，或可以是与多个处理核或其他半导体器件集成的图形处理器，其他半导体器件诸如但不限于存储器设备或网络接口。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接口并且利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1600包括用于访问存储器的存储器接口1614。存储器接口1614可以是到本地存储器、一个或多个内部高速缓存、一个或多个共享的外部高速缓存、和/或系统存储器的接口。

[0265] 在一些实施例中，图形处理器1600还包括显示控制器1602，该显示控制器1602用于将显示输出数据驱动到显示设备1618。显示控制器1602包括用于显示器的一个或多个叠加平面以及多层的视频或用户界面元素的合成的硬件。显示设备1618可以是内部或外部显示设备。在一个实施例中，显示设备1618是头戴式显示设备，诸如，虚拟现实(VR)显示设备或增强现实(AR)显示设备。在一些实施例中，图形处理器1600包括用于将媒体编码到一种或多种媒体编码格式，从一种或多种媒体编码格式对媒体解码，或在一种或多种媒体编码格式之间对媒体转码的视频编解码器引擎1606，这一种或多种媒体编码格式包括但不限于：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-4AVC、H.265/HEVC，开放媒体联盟(AOMedia)VP8、VP9)、以及电影和电视工程师协会(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)。

[0266] 在一些实施例中，图形处理器1600包括块图像传送(BLIT)引擎1604，用于执行二维(2D)栅格化器操作，包括例如，位边界块传送。然而，在一个实施例中，使用图形处理引擎(GPE)1610的一个或多个组件执行2D图形操作。在一些实施例中，GPE 1610是用于执行图形操作的计算引擎，这些图形操作包括三维(3D)图形操作和媒体操作。

[0267] 在一些实施例中，GPE 1610包括用于执行3D操作的3D流水线1612，3D操作诸如，使用作用于3D基元形状(例如，矩形、三角形等)的处理函数来渲染三维图像和场景。3D流水线1612包括可编程和固定功能元件，该可编程和固定功能元件执行到3D/媒体子系统1615的

元件和/或所生成的执行线程内的各种任务。虽然3D流水线1612可用于执行媒体操作，但是GPE 1610的实施例还包括媒体流水线1616，该媒体流水线1616专门用于执行媒体操作，诸如，视频后处理和图像增强。

[0268] 在一些实施例中，媒体流水线1616包括固定功能或可编程逻辑单元用于代替、或代表视频编解码器引擎1606来执行一个或多个专业的媒体操作，诸如，视频解码加速、视频去隔行、以及视频编码加速。在一些实施例中，媒体流水线1616附加地包括线程生成单元以生成用于在3D/媒体子系统1615上执行的线程。所生成的线程在3D/媒体子系统1615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。

[0269] 在一些实施例中，3D/媒体子系统1615包括用于执行由3D流水线1612和媒体流水线1616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1615发送线程执行请求，该3D/媒体子系统1615包括用于对于对可用的线程执行资源的各种请求进行仲裁和分派的线程分派逻辑。执行资源包括用于处理3D线程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1615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，该子系统还包括用于在线程之间共享数据并用于存储输出数据的共享存储器，其包括寄存器和可寻址存储器。

[0270] 图16B图示根据本文中描述的实施例的具有分片架构的图形处理器1620。在一个实施例中，图形处理器1620包括图形处理引擎集群1622，该图形处理引擎集群1622在图形引擎片1610A-1610D内具有图16A中的图形处理器引擎1610的多个实例。每个图形引擎片1610A-1610D可经由片互连的集合1623A-1623F被互连。每个图形引擎片1610A-1610D还可经由存储器互连1625A-1625D被连接到存储器模块或存储器设备1626A-1626D。存储器设备1626A-1626D可使用任何图形存储器技术。例如，存储器设备1626A-1626D可以是图形双倍数据速率(GDDR)存储器。在一个实施例中，存储器设备1626A-1626D是高带宽存储器(HBM)模块，这些高带宽存储器(HBM)模块可与其相应的图形引擎片1610A-1610D一起在管芯上。在一个实施例中，存储器设备1626A-1626D是堆叠式存储器设备，这些堆叠式存储器设备可被堆叠在它们相应的图形引擎片1610A-1610D的顶部上。在一个实施例中，每个图形引擎片1610A-1610D和相关联的存储器1626A-1626D驻留在分开的小芯片上，这些分开的小芯片被键合到基管芯或基衬底，如在图24B-图24D中进一步详细地所描述。

[0271] 图形处理引擎集群1622可与芯片上或封装上结构互连1624连接。结构互连1624可启用图形引擎片1610A-1610D与诸如视频编解码器1606和一个或多个副本引擎1604之类的组件之间的通信。副本引擎1604可用于将数据移出存储器设备1626A-1626D和在图形处理器1620外部的存储器(例如，系统存储器)，将数据移入存储器设备1626A-1626D和在图形处理器1620外部的存储器(例如，系统存储器)，并且在存储器设备1626A-1626D与在图形处理器1620外部的存储器(例如，系统存储器)之间移动数据。结构互连1624还可用于将图形引擎片1610A-1610D互连。图形处理器1620可任选地包括显示控制器1602，以启用与外部显示设备1618的连接。图形处理器还可被配置为图形加速器或计算加速器。在加速器配置中，显示控制器1602和显示设备1618可被省略。

[0272] 图形处理器1620可经由主机接口1628连接到主机系统。主机接口1628可启用图形处理器1620、系统存储器和/或系统组件之间的通信。主机接口1628可以是例如PCI express总线或另一类型的主机系统接口。

[0273] 图16C图示根据本文中描述的实施例的计算加速器1630。计算加速器1630可包括与图16B中的图形处理器1620的架构类似性，并且针对计算加速进行优化。计算引擎集群1632可包括计算引擎片的集合1640A-1640D，计算引擎片的集合1640A-1640D包括针对并行或基于向量的通用计算操作优化的执行逻辑。在一些实施例中，计算引擎片1640A-1640D不包括固定功能图形处理逻辑，但是在一个实施例中，计算引擎片1640A-1640D中的一个或多个可包括用于执行媒体加速的逻辑。计算引擎片1640A-1640D可经由存储器互连1625A-1625D连接到存储器1626A-1626D。存储器1626A-1626D和存储器互连1625A-1625D可以是与在图形处理器1620中类似的技术，或者可以是不同的技术。图形计算引擎片1640A-1640D还可经由片互连的集合1623A-1623F被互连，并且可与结构互连1624连接和/或由结构互连1624互连。在一个实施例中，计算加速器1630包括可被配置为设备范围的高速缓存的大型L3高速缓存1636。计算加速器1630还能以与图16B中的图形处理器1620类似的方式经由主机接口1628连接至主机处理器和存储器。

[0274] 图形处理引擎

[0275] 图17是根据一些实施例的图形处理器的图形处理引擎1710的框图。在一个实施例中，图形处理引擎(GPE)1710是图15A中示出的GPE 1510的某个版本，并且还可表示图15B中的图形引擎片1510A-1510D。图17的具有与本文中的任何其他附图的元件相同的附图标记(或名称)的那些元件能以类似于本文中其他地方描述的任何方式操作或起作用，但不限于此。例如，图示出图15A的3D流水线1612和媒体流水线1616。媒体流水线1616在GPE 1710的一些实施例中是任选的，并且可以不被显式地包括在GPE 1710内。例如并且在至少一个实施例中，单独的媒体和/或图像处理器被耦合至GPE 1710。

[0276] 在一些实施例中，GPE 1710与命令流转化器1703耦合或包括命令流转化器1703，该命令流转化器1703将命令流提供给3D流水线1612和/或媒体流水线1616。在一些实施例中，命令流转化器1703与存储器耦合，该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，命令流转化器1703从存储器接收命令，并将这些命令发送至3D流水线1612和/或媒体流水线1616。这些命令是从环形缓冲器取出的指示，该环形缓冲器存储用于3D流水线1612和媒体流水线1616的命令。在一个实施例中，环形缓冲器可附加地包括存储批量的多个命令的批量命令缓冲器。用于3D流水线1612的命令还可包括对存储在存储器中的数据的引用，这些数据诸如但不限于用于3D流水线1612的顶点数据和几何数据和/或用于媒体流水线1616的图像数据和存储器对象。3D流水线1612和媒体流水线1616通过经由各自流水线内的逻辑执行操作或者通过将一个或多个执行线程分派至图形核阵列1714来处理命令和数据。在一个实施例中，图形核阵列1714包括一个或多个图形核(例如，(多个)图形核1715A、(多个)图形核1715B)的块，每个块包括一个或多个图形核。每个图形核包括图形执行资源的集合，该图形执行资源的集合包括：用于执行图形操作和计算操作的通用执行逻辑和图形专用执行逻辑；以及固定功能纹理处理逻辑和/或机器学习和人工智能加速逻辑。

[0277] 在各实施例中，3D流水线可包括用于通过处理指令以及将执行线程分派到图形核阵列1714来处理一个或多个着色器程序的固定功能和可编程逻辑，这一个或多个着色器程序诸如，顶点着色器、几何着色器、像素着色器、片段着色器、计算着色器、或其他着色器程序。图形核阵列1714提供统一的执行资源块供在处理这些着色器程序时使用。图形核阵列

1714的(多个)图形核1715A-1714B内的多功能执行逻辑(例如,执行单元)包括对各种3D API着色器语言的支持,并且可执行与多个着色器相关联的多个同步执行线程。

[0278] 在一些实施例中,图形核阵列1714包括用于执行诸如视频和/或图像处理的媒体功能的执行逻辑。在一个实施例中,执行单元包括通用逻辑,该通用逻辑可编程以便除了执行图形处理操作之外还执行并行通用计算操作。通用逻辑可与图14的(多个)处理器核1407或图15A中的核1502A-1502N内的通用逻辑并行地或结合地执行处理操作。

[0279] 由在图形核阵列1714上执行的线程生成的输出数据可以将数据输出到统一返回缓冲器(URB)1718中的存储器。URB 1718可以存储用于多个线程的数据。在一些实施例中,URB 1718可用于在图形核阵列1714上执行的不同线程之间发送数据。在一些实施例中,URB 1718可附加地用于在图形核阵列上的线程与共享功能逻辑1720内的固定功能逻辑之间的同步。

[0280] 在一些实施例中,图形核阵列1714是可缩放的,使得阵列包括可变数量的图形核,每个图形核都具有基于GPE 1710的目标功率和性能等级的可变数量的执行单元。在一个实施例中,执行资源是动态可缩放的,使得可以根据需要启用或禁用执行资源。

[0281] 图形核阵列1714与共享功能逻辑1720耦合,该共享功能逻辑1720包括在图形核阵列中的图形核之间被共享的多个资源。共享功能逻辑1720内的共享功能是向图形核阵列1714提供专业的补充功能的硬件逻辑单元。在各实施例中,共享功能逻辑1720包括但不限于采样器逻辑1721、数学逻辑1722和线程间通信(ITC)逻辑1723。另外,一些实施例在共享功能逻辑1720内实现一个或多个高速缓存1725。

[0282] 至少在其中对于给定的专业功能的需求不足以包括在图形核阵列1714中的情况下实现共享功能。相反,那个专业功能的单个实例化被实现为共享功能逻辑1720中的独立实体,并且在图形核阵列1714内的执行资源之间被共享。在图形核阵列1714之间被共享并被包括在图形核阵列1714内的确切的功能集因实施例而异。在一些实施例中,共享功能逻辑1720内的由图形核阵列1714广泛使用的特定共享功能可被包括在图形核阵列1714内的共享功能逻辑1716内。在各实施例中,图形核阵列1714内的共享功能逻辑1716可包括共享功能逻辑1720内的一些或所有逻辑。在一个实施例中,共享功能逻辑1720内的所有逻辑元件可以在图形核阵列1714的共享功能逻辑1716内被复制。在一个实施例中,共享功能逻辑1720被排除以有利于图形核阵列1714内的共享功能逻辑1716。

[0283] 执行单元

[0284] 图18A-图18B图示根据本文中所描述的实施例的线程执行逻辑1800,该线程执行逻辑1800包括在图形处理器核中采用的处理元件的阵列。图18A-图18B的具有与本文中的任何其他附图的元件相同的附图标记(或名称)的那些元件能以类似于本文中其他地方描述的任何方式操作或起作用,但不限于此。图18A-图18A图示线程执行逻辑1800的概览,该线程执行逻辑1800可表示以图2B中的每个子核221A-221F图示的硬件逻辑。图18A表示通用图形处理器内的执行单元,而图18B表示可在计算加速器内被使用的执行单元。

[0285] 如在图18A中所图示,在一些实施例中,线程执行逻辑1800包括着色器处理器1802、线程分派器1804、指令高速缓存1806、包括多个执行单元1808A-1808N的可缩放执行单元阵列、采样器1810、共享本地存储器1811、数据高速缓存1812、以及数据端口1814。在一个实施例中,可缩放执行单元阵列可通过基于工作负载的计算要求启用或禁用一个或多个

执行单元(例如,执行单元1808A、1808B、1808C、1808D,一直到1808N-1和1808N中的任一个)来动态地缩放。在一个实施例中,所包括的组件经由互连结构而互连,该互连结构链接到组件中的每个组件。在一些实施例中,线程执行逻辑1800包括通过指令高速缓存1806、数据端口1814、采样器1810、以及执行单元1808A-1808N中的一个或多个到存储器(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,1808A)是能够执行多个同步硬件线程同时针对每个线程并行地处理多个数据元素的独立式可编程通用计算单元。在各实施例中,执行单元1808A-1808N的阵列是可缩放的以包括任何数量的单独的执行单元。

[0286] 在一些实施例中,执行单元1808A-1808N主要用于执行着色器程序。着色器处理器1802可处理各种着色器程序,并且可经由线程分派器1804分派与着色器程序相关联的执行线程。在一个实施例中,线程分派器包括用于对来自图形流水线和媒体流水线的线程发起请求进行仲裁并在执行单元1808A-1808N中的一个或多个执行单元上实例化所请求的线程的逻辑。例如,几何流水线可将顶点着色器、曲面细分着色器或几何着色器分派到线程执行逻辑以用于处理。在一些实施例中,线程分派器1804还可处理来自执行的着色器程序的运行时线程生成请求。

[0287] 在一些实施例中,执行单元1808A-1808N支持包括对许多标准3D图形着色器指令的原生支持的指令集,使得以最小的转换执行来自图形库(例如,Direct 3D和OpenGL)的着色器程序。这些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)以及通用处理(例如,计算和媒体着色器)。执行单元1808A-1808N中的每个执行单元都能够进行多发布单指令多数据(SIMD)执行,并且多线程操作在面对较高等待时间的存储器访问时启用高效的执行环境。每个执行单元内的每个硬件线程都具有专用的高带宽寄存器堆和相关的独立线程状态。对于能够进行整数操作、单精度浮点操作和双精度浮点操作、能够具有SIMD分支能力、能够进行逻辑操作、能够进行超越操作和能够进行其他混杂操作的流水线,执行针对每个时钟是多发布的。在等待来自存储器或共享功能中的一个共享功能的数据时,执行单元1808A-1808N内的依赖性逻辑使等待的线程休眠,直到所请求的数据已返回。当等待的线程正在休眠时,硬件资源可致力于处理其他线程。例如,在与顶点着色器操作相关联的延迟期间,执行单元可以执行针对像素着色器、片段着色器或包括不同顶点着色器的另一类型的着色器程序的操作。各实施例可应用以使用利用单指令多线程(SIMT)的执行,作为SIMD用例的替代,或作为SIMD用例的附加。对SIMD核或操作的引用也可应用于SIMT,或结合SIMT而应用于SIMD。

[0288] 执行单元1808A-1808N中的每个执行单元对数据元素的数组进行操作。数据元素的数量是“执行尺寸”、或用于指令的通道数量。执行通道是用于数据元素访问、掩码、和指令内的流控制的执行的逻辑单元。通道的数量可独立于用于特定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1808A-1808N支持整数和浮点数据类型。

[0289] 执行单元指令集包括SIMD指令。各种数据元素可以作为紧缩数据类型被存储在寄存器中,并且执行单元将基于元素的数据尺寸来处理各个元素。例如,当对256位宽的向量进行操作时,向量的256位被存储在寄存器中,并且执行单元将向量操作为四个单独的184位紧缩数据元素(四字(QW)尺寸数据元素)、八个单独的32位紧缩数据元素(双字(DW)尺寸

数据元素)、十六个单独的16位紧缩数据元素(字(W)尺寸的数据元素)、或三十二个单独的8位数据元素(字节(B)尺寸的数据元素)。然而,不同的向量宽度和寄存器尺寸是可能的。

[0290] 在一个实施例中,可以将一个或多个执行单元组合到融合执行单元1809A-1809N中,该融合执行单元1809A-1809N具有对于融合EU而言共同的线程控制逻辑(1807A-1807N)。可以将多个EU融合到EU组中。融合的EU组中的每个EU可以被配置成执行单独的SIMD硬件线程。融合的EU组中的EU的数量可以根据实施例而有所不同。另外,可以逐EU地执行各种SIMD宽度,包括但不限于SIMD8、SIMD16和SIMD32。每个融合图形执行单元1809A-1809N包括至少两个执行单元。例如,融合执行单元1809A包括第一EU 1808A、第二EU 1808B、以及对于第一EU 1808A和第二EU 1808B而言共同的线程控制逻辑1807A。线程控制逻辑1807A控制在融合图形执行单元1809A上执行的线程,从而允许融合执行单元1809A-1809N内的每个EU使用共同的指令指针寄存器来执行。

[0291] 一个或多个内部指令高速缓存(例如,1806)被包括在线程执行逻辑1800中以对用于执行单元的线程指令进行高速缓存。在一些实施例中,一个或多个数据高速缓存(例如,1812)被包括,以在线程执行期间对线程数据进行高速缓存。在执行逻辑1800上执行的线程还可将被显式地管理的数据存储在共享本地存储器1811中。在一些实施例中,采样器1810被包括以为3D操作提供纹理采样并且为媒体操作提供媒体采样。在一些实施例中,采样器1810包括专业的纹理或媒体采样功能,以便在向执行单元提供采样数据之前在采样过程期间处理纹理数据或媒体数据。

[0292] 在执行期间,图形流水线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到线程执行逻辑1800。一旦一组几何对象已经被处理并被栅格化为像素数据,着色器处理器1802内的像素处理器逻辑(例如,像素着色器逻辑、片段着色器逻辑等)就被调用以进一步计算输出信息,并且使得结果被写入到输出表面(例如,颜色缓冲器、深度缓冲器、模板印刷(stencil)缓冲器等)。在一些实施例中,像素着色器或片段着色器计算各顶点属性的值,各顶点属性的值将跨经栅格化的对象而被内插。在一些实施例中,着色器处理器1802内的像素处理器逻辑随后执行应用编程接口(API)供应的像素着色器程序或片段着色器程序。为了执行着色器程序,着色器处理器1802经由线程分派器1804将线程分派至执行单元(例如,1808A)。在一些实施例中,着色器处理器1802使用采样器1810中的纹理采样逻辑来访问存储在存储器中的纹理图中的纹理数据。对纹理数据和输入几何数据的算术操作计算针对每个几何片段的像素颜色数据,或丢弃一个或多个像素而不进行进一步处理。

[0293] 在一些实施例中,数据端口1814提供存储器访问机制,供线程执行逻辑1800将经处理的数据输出至存储器以便在图形处理器输出流水线上进一步处理。在一些实施例中,数据端口1814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1812),以便对数据进行高速缓存供经由数据端口进行存储器访问。

[0294] 在一个实施例中,执行逻辑1800还可包括可提供光线追踪加速功能的光线追踪器1805。光线追踪器1805可支持光线追踪指令集,该光线追踪指令集包括用于光线生成的指令/函数。光线追踪指令集可与图2C中的光线追踪核245所支持的光线追踪指令集类似或不同。

[0295] 图18B图示根据实施例的执行单元1808的示例性内部细节。图形执行单元1808可包括指令取出单元1837、通用寄存器堆阵列(GRF)1824、架构寄存器堆阵列(ARF)1826、线程

仲裁器1822、发送单元1830、分支单元1832、SIMD浮点单元(FPU)的集合1834、以及在一个实施例中的专用整数SIMD ALU的集合1835。GRF 1824和ARF 1826包括与可在图形执行单元1808中活跃的每个同步硬件线程相关联的通用寄存器堆和架构寄存器堆的集合。在一个实施例中,每线程架构状态被维持在ARF 1826中,而在线程执行期间使用的数据被存储在GRF 1824中。每个线程的执行状态,包括用于每个线程的指令指针,可以被保持在ARF 1826中的线程专用寄存器中。

[0296] 在一个实施例中,图形执行单元1808具有作为同步多线程(SMT)与细粒度交织多线程(IMT)的组合的架构。该架构具有模块化配置,该模块化配置可以基于同步线程的目标数量和每个执行单元的寄存器的数量而在设计时进行微调,其中跨用于执行多个同步线程的逻辑来划分执行单元资源。可由图形执行单元1808执行的逻辑线程的数量不限于硬件线程的数量,并且可将多个逻辑线程指派给每个硬件线程。

[0297] 在一个实施例中,图形执行单元1808可协同发布多条指令,这些指令可以各自是不同的指令。图形执行单元线程1808的线程仲裁器1822可以将指令分派给以下各项中的一项以供执行:发送单元1830、分支单元1832或(多个)SIMD FPU 1834。每个执行线程可以访问GRF 1824内的128个通用寄存器,其中,每个寄存器可以存储可作为具有32位数据元素的SIMD 8元素向量访问的32个字节。在一个实施例中,每个执行单元线程具有对GRF 1824内的4个千字节的访问权,但是实施例并不限于此,并且在其他实施例中可以提供更多或更少的寄存器资源。在一个实施例中,图形执行单元1808被分区为可独立地执行计算操作的七个硬件线程,但是每个执行单元的线程数量也可根据实施例而有所不同。例如,在一个实施例中,支持多达16个硬件线程。在其中七个线程可以访问4个千字节的实施例中,GRF 1824可以存储总共28个千字节。在16个线程可访问4个千字节的情况下,GRF 1824可存储总共64个千字节。灵活的寻址模式可以准许对多个寄存器一起进行寻址,从而建立实际上更宽的寄存器或者表示跨步式矩形块数据结构。

[0298] 在一个实施例中,经由通过消息传递发送单元1830执行的“发送”指令来分派存储器操作、采样器操作以及其他较长等待时间的系统通信。在一个实施例中,分支指令被分派给专用分支单元1832以促进SIMD发散和最终收敛。

[0299] 在一个实施例中,图形执行单元1808包括用于执行浮点操作的一个或多个SIMD浮点单元(FPU)1834。在一个实施例中,(多个)FPU 1834还支持整数计算。在一个实施例中,(多个)FPU 1834可以SIMD执行多达数量M个32位浮点(或整数)操作,或者SIMD执行多达2M个16位整数或16位浮点操作。在一个实施例中,(多个)FPU中的至少一个提供支持高吞吐量超越数学函数和双精度184位浮点的扩展数学能力。在一些实施例中,8位整数SIMD ALU的集合1835也存在,并且可专门优化成执行与机器学习计算相关联的操作。

[0300] 在一个实施例中,可以在图形子核分组(例如,子切片)中对图形执行单元1808的多个实例的阵列进行实例化。为了可缩放性,产品架构师可以选择每子核分组的执行单元的确切数量。在一个实施例中,执行单元1808可以跨多个执行通道来执行指令。在进一步的实施例中,在不同通道上执行在图形执行单元1808上执行的每个线程。

[0301] 图19图示根据实施例的附加的执行单元1900。执行单元1900可以是用于在例如图15C中的计算引擎片1540A-1540D中使用的计算优化的执行单元,但不限于此。执行单元1900的变体也可在如图15B中的图形引擎片1510A-1510D中使用。在一个实施例中,执行单

元1900包括线程控制单元1901、线程状态单元1902、指令取出/预取单元1903、以及指令解码单元1904。执行单元1900附加地包括寄存器堆1906，该寄存器堆1906存储可被指派给执行单元内的硬件线程的寄存器。执行单元1900附加地包括发送单元1907和分支单元1908。在一个实施例中，发送单元1907和分支单元1908能以与图18B中的图形执行单元1808的发送单元1830和分支单元1832类似的方式操作。

[0302] 执行单元1900还包括计算单元1910，该计算单元1910包括多个不同类型的功能单元。在一个实施例中，计算单元1910包括ALU单元1911，该ALU单元1911包括算术逻辑单元的阵列。ALU单元1911可配置成执行64位、32位和16位的整数和浮点操作。可同时执行整数和浮点操作。计算单元1910还可包括脉动阵列1912和数学单元1913。脉动阵列1912包括数据处理单元的宽W深D的网络，其可用于以脉动方式执行向量或其他数据并行操作。在一个实施例中，脉动阵列1912可配置成执行矩阵操作，诸如，矩阵点积操作。在一个实施例中，脉动阵列1912支持16位浮点操作以及8位和4位整数操作。在一个实施例中，脉动阵列1912可配置成加速机器学习操作。在此类实施例中，脉动阵列1912可配置有对bfloat 16位浮点格式的支持。在一个实施例中，数学单元1913可被包括以便以高效的且比ALU单元1911更低功率的方式执行数学操作的特定子集。数学单元1913可包括可在由其他实施例提供的图形处理引擎的共享功能逻辑(例如，图17中的共享功能逻辑1720的数学逻辑1722)中发现的数学逻辑的变体。在一个实施例中，数学单元1913可配置成执行32位和64位浮点操作。

[0303] 线程控制单元1901包括用于控制执行单元内的线程的执行的逻辑。线程控制单元1901可包括线程仲裁逻辑，该线程仲裁逻辑用于启动、停止以及抢占执行单元1900内线程的执行。线程状态单元1902可用于存储用于被指派在执行单元1900上执行的线程的线程状态。将线程状态存储在执行单元1900能使得能够在线程变得被锁定或空闲时快速抢占那些线程。指令取出/预取单元1903可从较高级别执行逻辑的指令高速缓存(例如，如图18A中的指令高速缓存1806)取出指令。指令取出/预取单元1903还基于对当前执行线程的分析来发布对要被加载到执行高速缓存中的指令的预取请求。指令解码单元1904可用于对要由计算单元执行的指令进行解码。在一个实施例中，指令解码单元1904可被用作次级解码器以将复杂指令解码为组成的微操作。

[0304] 执行单元1900附加地包括寄存器堆1906，该寄存器堆可由在执行单元1900上执行的硬件线程使用。寄存器堆1906中的寄存器可跨用于执行执行单元1900的计算单元1910内的多个同步线程的逻辑而被划分。可由图形执行单元1900执行的逻辑线程的数量不限于硬件线程的数量，并且可将多个逻辑线程指派给每个硬件线程。基于所支持的硬件线程的数量，寄存器堆1906的尺寸可因实施例而异。在一个实施例中，可使用寄存器重命名来动态地将寄存器分配给硬件线程。

[0305] 图20是图示根据一些实施例的图形处理器指令格式2000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多种格式的指令的指令集。实线框图示通常被包括在执行单元指令中的组成部分，而虚线包括任选的或仅被包括在指令的子集中的组成部分。在一些实施例中，所描述和图示的指令格式2000是宏指令，因为它们是供应至执行单元的指令，这与产生自一旦指令被处理就进行的指令解码的微指令相反。

[0306] 在一些实施例中，图形处理器执行单元原生地支持128位指令格式2010的指令。基于所选择的指令、指令选项和操作数数量，64位紧凑指令格式2030可用于一些指令。原生

128位指令格式2010提供对所有指令选项的访问，而一些选项和操作在64位格式2030中受限。64位格式2030中可用的原生指令因实施例而异。在一些实施例中，使用索引字段2013中的索引值的集合将指令部分地压缩。执行单元硬件基于索引值来引用压缩表的集合，并使用压缩表输出来重构128位指令格式2010的原生指令。可使用指令的其他尺寸和格式。

[0307] 针对每种格式，指令操作码2012限定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地执行每条指令。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执行同步加法操作。默认地，执行单元跨操作数的所有数据通道执行每条指令。在一些实施例中，指令控制字段2014启用对某些执行选项的控制，这些执行选项诸如通道选择(例如，断言)以及数据通道顺序(例如，混合)。针对128位指令格式2010的指令，执行尺寸字段2016限制将被并行地执行的数据通道的数量。在一些实施例中，执行尺寸字段2016不可用于64位紧凑指令格式2030。

[0308] 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 2020、src1 2022以及一个目的地操作数2018。在一些实施例中，执行单元支持双目的地指令，其中，双目的地中的一个目的地是隐式的。数据操纵指令可具有第三源操作数(例如，SRC2 2024)，其中，指令操作码2012确定源操作数的数量。指令的最后一个源操作数可以是与指令一起传递的立即数(例如，硬编码的)值。

[0309] 在一些实施例中，128位指令格式2010包括访问/寻址模式字段2026，该访问/寻址模式字段2026例如指定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，由指令中的位直接提供一个或多个操作数的寄存器地址。

[0310] 在一些实施例中，128位指令格式2010包括访问/寻址模式字段2026，该访问/寻址模式字段2026指定指令的寻址模式和/或访问模式。在一个实施例中，访问模式用于限定针对指令的数据访问对齐。一些实施例支持包括16字节对齐访问模式和1字节对齐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，当处于第一模式时，指令可将字节对齐寻址用于源操作数和目的地操作数，并且当处于第二模式时，指令可将16字节对齐寻址用于所有的源操作数和目的地操作数。

[0311] 在一个实施例中，访问/寻址模式字段2026的寻址模式部分确定指令要使用直接寻址还是间接寻址。当使用直接寄存器寻址模式时，指令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，可以基于指令中的地址寄存器值和地址立即数字段来计算一个或多个操作数的寄存器地址。

[0312] 在一些实施例中，基于操作码2012位字段对指令进行分组从而简化操作码解码2040。针对8位的操作码，位4、位5、和位6允许执行单元确定操作码的类型。所示出的确切的操作码分组仅是示例。在一些实施例中，移动和逻辑操作码组2042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，移动和逻辑组2042共享五个最高有效位(MSB)，其中，移动(mov)指令采用0000xxxxb的形式，而逻辑指令采用0001xxxxb的形式。流控制指令组2044(例如，调用(call)、跳转(jmp))包括0010xxxxb(例如，0x20)形式的指令。混杂指令组2046包括指令的混合，包括0011xxxxb(例如，0x30)形式的同步指令(例如，等待(wait)、发送(send))。并行数学指令组2048包括0100xxxxb(例如，0x40)形式的逐分量的算术指令(例如，加、乘(mul))。并行数学组2048跨数据通道并行地执行算术操作。向量数学组2050包括0101xxxxb(例如，0x50)形式的算术指令(例如，dp4)。向量数学组对向量操作

数执行算术，诸如点积计算。在一个实施例中，所图示的操作码解码2040可用于确定执行单元的哪个部分将用于执行经解码的指令。例如，一些指令可被指定为将由脉动阵列执行的脉动指令。其他指令，诸如，光线追踪指令(未示出)可被路由到执行逻辑的切片或分区内的光线追踪核或光线追踪逻辑。

[0313] 图形流水线

[0314] 图21是根据另一实施例的图形处理器2100的框图。图21的具有与本文中的任何其他附图的元件相同的附图标记(或名称)的那些元件能以类似于本文中其他地方描述的任何方式操作或起作用，但不限于此。

[0315] 在一些实施例中，图形处理器2100包括图形流水线2120、媒体流水线2130、显示引擎2140、线程执行逻辑2150、以及渲染输出流水线2170。在一些实施例中，图形处理器2100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。图形处理器通过至一个或多个控制寄存器(未示出)的寄存器写入、或者经由通过环形互连2102发布至图形处理器2100的命令被控制。在一些实施例中，环形互连2102将图形处理器2100耦合至其他处理组件，诸如其他图形处理器或通用处理器。来自环形互连2102的命令由命令流转化器2103解译，该命令流转化器将指令供应至几何流水线2120或媒体流水线2130的各个组件。

[0316] 在一些实施例中，命令流转化器2103引导顶点取出器2105的操作，该顶点取出器2105从存储器读取顶点数据，并执行由命令流转化器2103提供的顶点处理命令。在一些实施例中，顶点取出器2105将顶点数据提供给顶点着色器2107，该顶点着色器2107对每个顶点执行坐标空间变换和照明操作。在一些实施例中，顶点取出器2105和顶点着色器2107通过经由线程分派器2131将执行线程分派至执行单元2152A-2152B来执行顶点处理指令。

[0317] 在一些实施例中，执行单元2152A-2152B是具有用于执行图形操作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2152A-2152B具有专用于每个阵列或在阵列之间被共享的所附接的L1高速缓存2151。高速缓存可以被配置为数据高速缓存、指令高速缓存、或被分区为在不同分区中包含数据和指令的单个高速缓存。

[0318] 在一些实施例中，几何流水线2120包括用于执行3D对象的硬件加速曲面细分的曲面细分组件。在一些实施例中，可编程外壳着色器2111配置曲面细分操作。可编程域着色器2117提供对曲面细分输出的后端评估。曲面细分器2113在外壳着色器2111的指示下进行操作，并且包括用于基于粗糙的几何模型来生成详细的几何对象集合的专用逻辑，该粗糙的几何模型作为输入被提供该几何流水线2120。在一些实施例中，如果不使用曲面细分，则可以绕过曲面细分组件(例如，外壳着色器2111、曲面细分器2113和域着色器2117)。

[0319] 在一些实施例中，完整的几何对象可由几何着色器2119经由被分派至执行单元2152A-2152B的一个或多个线程来处理，或者可以直接行进至裁剪器2129。在一些实施例中，几何着色器对整个几何对象而不是对如在图形流水线的先前的级中那样对顶点或顶点补片进行操作。如果禁用曲面细分，则几何着色器2119从顶点着色器2107接收输入。在一些实施例中，几何着色器2119是可由几何着色器程序编程的以便在曲面细分单元被禁用的情况下执行几何曲面细分。

[0320] 在栅格化之前，裁剪器2129处理顶点数据。裁剪器2129可以是固定功能裁剪器或具有裁剪和几何着色器功能的可编程裁剪器。在一些实施例中，渲染输出流水线2170中的栅格化器和深度测试组件2173分派像素着色器以将几何对象转换为逐像素表示。在一些实

施例中，像素着色器逻辑被包括在线程执行逻辑2150中。在一些实施例中，应用可绕过栅格化器和深度测试组件2173，并且经由流出单元2123访问未栅格化的顶点数据。

[0321] 图形处理器2100具有互连总线、互连结构、或允许数据和消息在处理器的主要组件之中传递的某个其他互连机制。在一些实施例中，执行单元2152A-2152B和相关联的逻辑单元(例如，L1高速缓存2151、采样器2154、纹理高速缓存2158等)经由数据端口2156进行互连，以便执行存储器访问并且与处理器的渲染输出流水线组件进行通信。在一些实施例中，采样器2154、高速缓存2151、2158以及执行单元2152A-2152B各自具有单独的存储器访问路径。在一个实施例中，纹理高速缓存2158也可被配置为采样器高速缓存。

[0322] 在一些实施例中，渲染输出流水线2170包含栅格化器和深度测试组件2173，其将基于顶点的对象转换为相关联的基于像素的表示。在一些实施例中，栅格化器逻辑包括用于执行固定功能三角形和线栅格化的窗口器/掩码器单元。相关联的渲染高速缓存2178和深度高速缓存2179在一些实施例中也是可用的。像素操作组件2177对数据进行基于像素的操作，但是在一些实例中，与2D操作相关联的像素操作(例如，利用混合的位块图像传送)由2D引擎2141执行，或者在显示时由显示控制器2143使用叠加显示平面来代替。在一些实施例中，共享的L3高速缓存2175可用于所有的图形组件，从而允许在无需使用主系统存储器的情况下共享数据。

[0323] 在一些实施例中，图形处理器媒体流水线2130包括媒体引擎2137和视频前端2134。在一些实施例中，视频前端2134从命令流转化器2103接收流水线命令。在一些实施例中，媒体流水线2130包括单独的命令流转化器。在一些实施例中，视频前端2134在将媒体命令发送至媒体引擎2137之前处理该命令。在一些实施例中，媒体引擎2137包括用于生成线程以用于经由线程分派器2131分派至线程执行逻辑2150的线程生成功能。

[0324] 在一些实施例中，图形处理器2100包括显示引擎2140。在一些实施例中，显示引擎2140在处理器2100外部，并且经由环形互连2102、或某个其他互连总线或结构来与图形处理器耦合。在一些实施例中，显示引擎2140包括2D引擎2141和显示控制器2143。在一些实施例中，显示引擎2140包含能够独立于3D流水线进行操作的专用逻辑。在一些实施例中，显示控制器2143与显示设备(未示出)耦合，该显示设备可以是系统集成显示设备(如在膝上型计算机中)、或者是经由显示设备连接器附接的外部显示设备。

[0325] 在一些实施例中，几何流水线2120和媒体流水线2130可被配置成用于基于多个图形和媒体编程接口执行操作，并且并非专用于任何一种应用编程接口(API)。在一些实施例中，图形处理器的驱动器软件将专用于特定图形或媒体库的API调用转换成可由图形处理器处理的命令。在一些实施例中，为全部来自Khronos Group的开放图形库(OpenGL)、开放计算语言(OpenCL)和/或Vulkan图形和计算API提供支持。在一些实施例中，也可以为来自微软公司的Direct3D库提供支持。在一些实施例中，可以支持这些库的组合。还可以为开源计算机视觉库(OpenCV)提供支持。如果可进行从未来API的流水线到图形处理器的流水线的映射，则具有兼容3D流水线的未来API也将受到支持。

[0326] 图形流水线编程

[0327] 图22A是图示根据一些实施例的图形处理器命令格式2200的框图。图22B是图示根据实施例的图形处理器命令序列2210的框图。图22A中的实线框图示一般被包括在图形命令中的组成部分，而虚线包括任选的或仅被包括在图形命令的子集中的组成部分。图22A的

示例性图形处理器命令格式2200包括用于标识命令的客户端2202、命令操作代码(操作码)2204和数据2206的数据字段。子操作码2205和命令尺寸2208也被包括在一些命令中。

[0328] 在一些实施例中,客户端2202指定图形设备的处理命令数据的客户端单元。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适当的客户端单元。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。一旦由客户端单元接收到命令,客户端单元就读取操作码2204以及子操作码2205(如果存在)以确定要执行的操作。客户端单元使用数据字段2206内的信息来执行命令。针对一些命令,预期显式的命令尺寸2208指定命令的尺寸。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尺寸。在一些实施例中,经由双字的倍数来对齐命令。可使用其他命令格式。

[0329] 图22B中的流程图示示例性图形处理器命令序列2210。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某个版本来建立、执行并终止图形操作的集合。仅出于示例性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的命令或者该命令序列。而且,命令可以作为批量的命令以命令序列被发布,使得图形处理器将以至少部分同时的方式处理命令序列。

[0330] 在一些实施例中,图形处理器命令序列2210可开始于流水线转储清除命令2212,以便使得任何活跃的图形流水线完成流水线的当前未决命令。在一些实施例中,3D流水线2222和媒体流水线2224不并发地操作。执行流水线转储清除以使得活跃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,直到活跃的绘画引擎完成未决操作并且相关的读高速缓存被无效。任选地,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,流水线转储清除命令2212可以用于流水线同步,或者在将图形处理器置于低功率状态之前使用。

[0331] 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明确地切换时,使用流水线选择命令2213。在一些实施例中,在发布流水线命令之前在执行上下文中仅需要一次流水线选择命令2213,除非上下文将发布针对两条流水线的命令。在一些实施例中,紧接在经由流水线选择命令2213的流水线切换之前需要流水线转储清除命令2212。

[0332] 在一些实施例中,流水线控制命令2214配置用于操作的图形流水线,并且用于对3D流水线2222和媒体流水线2224进行编程。在一些实施例中,流水线控制命令2214配置活跃流水线的流水线状态。在一个实施例中,流水线控制命令2214用于流水线同步,并且用于在处理批量的命令之前清除来自活跃流水线内的一个或多个高速缓存存储器的数据。

[0333] 在一些实施例中,返回缓冲器状态命令2216用于配置用于相应流水线的返回缓冲器的集合以写入数据。一些流水线操作需要分配、选择或配置一个或多个返回缓冲器,在处理期间操作将中间数据写入这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来存储输出数据并且执行跨线程通信。在一些实施例中,返回缓冲器状态2216包括选择要用于流水线操作的集合的返回缓存器的尺寸和数量。

[0334] 命令序列中的剩余命令基于用于操作的活跃流水线而不同。基于流水线判定2220,命令序列被定制用于以3D流水线状态2230开始的3D流水线2222、或者在媒体流水线状态2240处开始的媒体流水线2224。

[0335] 用于配置3D流水线状态2230的命令包括用于顶点缓冲器状态、顶点元素状态、常量颜色状态、深度缓冲器状态、以及将在处理3D基元命令之前配置的其他状态变量的3D状态设置命令。这些命令的值至少部分地基于使用中的特定3D API来确定。在一些实施例中，如果将不使用某些流水线元件，则3D流水线状态2230命令还能够选择性地禁用或绕过那些元件。

[0336] 在一些实施例中，3D基元2232命令用于提交待由3D流水线处理的3D基元。经由3D基元2232命令传递给图形处理器的命令和相关联的参数被转发到图形流水线中的顶点取出功能。顶点取出功能使用3D基元2232命令数据来生成多个顶点数据结构。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基元2232命令用于经由顶点着色器对3D基元执行顶点操作。为了处理顶点着色器，3D流水线2222将着色器执行线程分派至图形处理器执行单元。

[0337] 在一些实施例中，经由执行2234命令或事件触发3D流水线2222。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“去往(go)”或“踢除(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，以便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基元的几何处理。一旦操作完成，就对所得到的几何对象进行栅格化，并且像素引擎对所得到的像素进行着色。对于那些操作，还可以包括用于控制像素着色和像素后端操作的附加命令。

[0338] 在一些实施例中，当执行媒体操作时，图形处理器命令序列2210遵循媒体流水线2224路径。一般地，针对媒体流水线2224进行编程的特定用途和方式取决于待执行的媒体或计算操作。在媒体解码期间，特定的媒体解码操作可以被转移到媒体流水线。在一些实施例中，还可绕过媒体流水线，并且可使用由一个或多个通用处理核提供的资源来整体地或部分地执行媒体解码。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，这些计算着色器程序并不明确地与图形基元的渲染相关。

[0339] 在一些实施例中，以与3D流水线2222类似的方式配置媒体流水线2224。将用于配置媒体流水线状态2240的命令集合分派或放置到命令队列中，在媒体对象命令2242之前。在一些实施例中，用于媒体流水线状态的命令2240包括用于配置媒体流水线元件的数据，这些媒体流水线元件将用于处理媒体对象。这包括用于在媒体流水线内配置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，用于媒体流水线状态的命令2240还支持使用指向包含批量的状态设置的“间接”状态元件的一个或多个指针。

[0340] 在一些实施例中，媒体对象命令2242供应指向用于由媒体流水线处理的媒体对象的指针。媒体对象包括存储器缓冲器，该存储器缓冲器包含待处理的视频数据。在一些实施例中，在发布媒体对象命令2242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2242被排队，就经由执行命令2244或等效的执行事件(例如，寄存器写入)来触发媒体流水线2224。随后可通过由3D流水线2222或媒体流水线2224提供的操作对来自媒体流水线2224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。

[0341] 图形软件架构

[0342] 图23图示根据一些实施例的用于数据处理系统2300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2310、操作系统2320、以及至少一个处理器2330。在一些实施例中，处理器2330包括图形处理器2332以及一个或多个通用处理器核2334。图形应用2310和操作系统2320各自在数据处理系统的系统存储器2350中执行。

[0343] 在一些实施例中，3D图形应用2310包含一个或多个着色器程序，这一个或多个着色器程序包括着色器指令2312。着色器语言指令可以采用高级着色器语言，诸如，DirectD的高级着色器语言(HLSL)、OpenGL着色器语言(GLSL)，等等。应用还包括采用适于由通用处理器核2334执行的机器语言的可执行指令2314。应用还包括由顶点数据限定的图形对象2316。

[0344] 在一些实施例中，操作系统2320是来自微软公司的Microsoft® Windows®操作系统、专属的类UNIX操作系统、或使用Linux内核的变体的开源的类UNIX操作系统。操作系统2320可支持图形API 2322，诸如Direct3D API、OpenGL API或Vulkan API。当Direct3D API正在使用时，操作系统2320使用前端着色器编译器2324以将采用HLSL的任何着色器指令2312编译成较低级的着色器语言。编译可以是即时(JIT)编译，或者应用可执行着色器预编译。在一些实施例中，在3D图形应用2310的编译期间，将高级着色器编译成低级着色器。在一些实施例中，着色器指令2312以中间形式提供，诸如由Vulkan API使用的标准便携式中间表示(SPIR)的某个版本。

[0345] 在一些实施例中，用户模式图形驱动器2326包含后端着色器编译器2327，该后端着色器编译器2327用于将着色器指令2312转换成硬件专用表示。当OpenGL API在使用中时，将采用GLSL高级语言的着色器指令2312传递至用户模式图形驱动器2326以用于编译。在一些实施例中，用户模式图形驱动器2326使用操作系统内核模式功能2328来与内核模式图形驱动器2329进行通信。在一些实施例中，内核模式图形驱动器2329与图形处理器2332通信以分派命令和指令。

[0346] IP核实施方式

[0347] 至少一个实施例的一个或多个方面可以由存储在机器可读介质上的代表性代码实现，该机器可读介质表示和/或限定集成电路(诸如，处理器)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，指令可以使机器制造用于执行本文所述的技术的逻辑。这类表示(被称为“IP核”)是集成电路的逻辑的可重复使用单元，这些可重复使用单元可以作为描述集成电路的结构的硬件模型而被存储在有形的、机器可读介质上。可以将硬件模型供应至在制造集成电路的制造机器上加载硬件模型的各消费者或制造设施。可以制造集成电路，使得电路执行与本文中描述的实施例中的任一实施例相关联地描述的操作。

[0348] 图24A是图示根据实施例的IP核开发系统2400的框图，该IP核开发系统2400可以用于制造集成电路以执行操作。IP核开发系统2400可以用于生成可并入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2430可生成采用高级编程语言(例如，C/C++)的IP核设计的软件仿真2410。软件仿真2410可用于使用仿真模型2412来设计、测试并验证IP核的行为。仿真模型2412可以包括功能仿真、行为仿真和/或时序仿真。随后可从仿真模型2412创建或合成寄存器传输级(RTL)设计2415。RTL设计2415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执行的相关联的逻辑)的行为的抽象。除了RTL设计2415之外，还可以创建、设计或合成逻

辑级或晶体管级的较低级别设计。由此，初始设计和仿真的特定细节可有所不同。

[0349] 可以由设计设施进一步将RTL设计2415或等效方案合成到硬件模型2420中，该硬件模型2420可以采用硬件描述语言(HDL)或物理设计数据的某种其他表示。可以进一步仿真或测试HDL以验证IP核设计。可使用非易失性存储器2440(例如，硬盘、闪存、或任何非易失性存储介质)来存储IP核设计以用于递送至第三方制造设施2465。替代地，可以通过有线连接2450或无线连接2460(例如，经由因特网)来传输IP核设计。制造设施2465随后可以制造至少部分地基于IP核设计的集成电路。所制造的集成电路可被配置用于执行根据本文中描述的至少一个实施例的操作。

[0350] 图24B图示根据本文中描述的一些实施例的集成电路封装组件22470的截面侧视图。集成电路封装组件2470图示如本文中所描述的一个或多个处理器或加速器设备的实现方式。封装组件2470包括连接至衬底2480的多个硬件逻辑单元2472、2474。逻辑2472、2474可以至少部分地实现在可配置逻辑或固定功能逻辑硬件中，并且可包括本文中描述的(多个)处理器核、(多个)图形处理器或其他加速器设备中的任何处理器核、图形处理器或其他加速器设备的一个或多个部分。每个逻辑单元2472、2474可以实现在半导体管芯内，并且经由互连结构2473与衬底2480耦合。互连结构2473可以被配置成在逻辑2472、2474与衬底2480之间路由电信号，并且可以包括互连，该互连诸如但不限于凸块或支柱。在一些实施例中，互连结构2473可以被配置成路由电信号，诸如例如，与逻辑2472、2474的操作相关联的输入/输出(I/O)信号和/或功率或接地信号。在一些实施例中，衬底2480是基于环氧树脂的层压衬底。在其他实施例中，封装衬底2480可以包括其他合适类型的衬底。封装组件2470可以经由封装互连2483连接至其他电气设备。封装互连2483可以耦合至衬底2480的表面以将电信号路由到其他电气设备，诸如主板、其他芯片组或多芯片模块。

[0351] 在一些实施例中，逻辑单元2472、2474与桥接器2482电耦合，该桥接器2482被配置成在逻辑2472与逻辑2474之间路由电信号。桥接器2482可以是为电信号提供路由的密集互连结构。桥接器2482可以包括由玻璃或合适的半导体材料构成的桥接器衬底。电路由特征可形成在桥接器衬底上以提供逻辑2472与逻辑2474之间的芯片到芯片连接。

[0352] 尽管图示了两个逻辑单元2472、2474和桥接器2482，但是本文中所描述的实施例可以包括在一个或多个管芯上的更多或更少的逻辑单元。这一个或多个管芯可以由零个或更多个桥接器连接，因为当逻辑被包括在单个管芯上时，可以排除桥接器2482。替代地，多个管芯或逻辑单元可以由一个或多个桥接器连接。另外，在其他可能的配置(包括三维配置)中，多个逻辑单元、管芯和桥接器可被连接在一起。

[0353] 图24C图示封装组件2490，该封装组件2490包括连接到衬底2480的多个单元的硬件逻辑小芯片(例如，基础管芯)。如本文中所描述的图形处理单元、并行处理器和/或计算加速器可由分开制造的各种硅小芯片组成。在该上下文中，小芯片是至少部分地被封装的集成电路，该至少部分地被封装的集成电路包括可与其他小芯片一起被组装到更大的封装中的不同的逻辑单元。具有不同IP核逻辑的小芯片的各种集合可被组装到单个器件中。此外，可使用有源插入器技术将小芯片集成到基础管芯或基础小芯片中。本文中描述的概念启用GPU内的不同形式的IP之间的互连和通信。IP核可通过使用不同的工艺技术来制造并在制造期间被组成，这避免了尤其是对于具有若干风格的IP的大型SoC的将多个IP聚集到同一制造工艺的复杂性。允许使用多种工艺技术改善了上市时间，并提供具有成本效益的

方法来创建多个产品SKU。此外，分解的IP更适于被独立地进行功率门控，可关闭不在给定工作负载上使用的组件，从而降低总功耗。

[0354] 硬件逻辑小芯片可包括专用硬件逻辑小芯片2472、逻辑或I/O小芯片2474、和/或存储器小芯片2475。硬件逻辑小芯片2472以及逻辑或I/O小芯片2474可以至少部分地实现在可配置逻辑或固定功能逻辑硬件中，并且可包括本文中描述的(多个)处理器核、(多个)图形处理器、并行处理器或其他加速器设备中的任何处理器核、图形处理器、并行处理器或其他加速器设备的一个或多个部分。存储器小芯片2475可以是DRAM(例如，GDDR、HBM)存储器或高速缓存(SRAM)存储器。

[0355] 每个小芯片可被制造为单独的半导体管芯，并且经由互连结构2743与衬底2480耦合。互连结构2473可配置成在衬底2480内的各种小芯片与逻辑之间路由电信号。互连结构2473可包括互连，诸如但不限于凸块或支柱。在一些实施例中，互连结构2473可以被配置成路由电信号，诸如例如，与逻辑小芯片、I/O小芯片和存储器小芯片的操作相关联的输入/输出(I/O)信号和/或功率信号或接地信号。

[0356] 在一些实施例中，衬底2480是基于环氧树脂的层压衬底。在其他实施例中，衬底2480可包括其他合适类型的衬底。封装组件2490可以经由封装互连2483连接至其他电气设备。封装互连2483可以耦合至衬底2480的表面以将电信号路由到其他电气设备，诸如主板、其他芯片组或多芯片模块。

[0357] 在一些实施例中，逻辑或I/O小芯片2474和存储器小芯片2475可经由桥接器2487被电耦合，该桥接器2487配置成在逻辑或I/O小芯片2474与存储器小芯片2475之间路由电信号。桥接器2487可以是为电信号提供路由的密集互连结构。桥接器2487可以包括由玻璃或合适的半导体材料构成的桥接器衬底。电路由特征可形成在桥接器衬底上以提供逻辑或I/O小芯片2474与存储器小芯片2475之间的芯片到芯片连接。桥接器2487还可被称为硅桥接器或互连桥接器。例如，在一些实施例中，桥接器2487是嵌入式多管芯互连桥接器(EMIB)。在一些实施例中，桥接器2487可以仅是从一个小芯片到另一小芯片的直接连接。

[0358] 衬底2480可包括用于I/O 2491、高速缓存存储器2492和其他硬件逻辑2493的硬件组件。结构2485可被嵌入在衬底2480中以启用衬底2480内的各种逻辑小芯片与逻辑2491、2493之间的通信。在一个实施例中，I/O2491、结构2485、高速缓存、桥接器和其他硬件逻辑2493可集成在层叠在衬底2480的顶部上的基础管芯中。

[0359] 在各实施例中，封装组件2490可包括由结构2485或一个或多个桥接器2487互连的更少或更多数量的组件和桥接器。封装组件2490内的小芯片能以3D布置或2.5D布置来布置。一般而言，桥接器结构2487可用于促进例如逻辑或I/O小芯片与存储器小芯片之间的点对点互连。结构2485可用于将各种逻辑和/或I/O小芯片(例如，小芯片2472、2474、2491、2493)与其他逻辑和/或I/O小芯片互连。在一个实施例中，衬底内的高速缓存存储器2492可充当用于封装组件2490的全局高速缓存，充当分布式全局高速缓存的部分，或充当用于结构2485的专用高速缓存。

[0360] 图24D图示根据实施例的包括可互换小芯片2495的封装组件2494。可互换小芯片2495可被组装到一个或多个基础小芯片2496、2498上的标准化插槽中。基础小芯片2496、2498可经由桥接器互连2497被耦合，该桥接器互连2497可与本文中描述的其他桥接器互连类似，并且可以是例如EMIB。存储器小芯片也可经由桥接器互连被连接至逻辑或I/O小芯

片。I/O和逻辑小芯片可经由互连结构进行通信。基础小芯片各自可支持按照用于逻辑或I/O或存器/高速缓存的标准化格式的一个或多个插槽。

[0361] 在一个实施例中，SRAM和功率递送电路可被制造到基础小芯片2496、2498中的一个或多个中，基础小芯片2496、2498可使用相对于可互换小芯片2495不同的工艺技术来制造，可互换小芯片2495堆叠在基础小芯片的顶部上。例如，可使用较大工艺技术来制造基础小芯片2496、2498，同时可使用较小工艺技术来制造可互换小芯片。可互换小芯片2495中的一个或多个可以是存储器(例如，DRAM)小芯片。可基于针对使用封装组件2494的产品的功率和/或新能来为封装组件2494选择不同的存储器密度。此外，可在组装时基于针对产品的功率和/或性能来选择具有不同数量的类型的功能单元的逻辑小芯片。此外，可将包含具有不同类型的IP逻辑核的小芯片插入到可互换小芯片插槽中，从而启用可混合并匹配不同技术的IP块的混合式存储器设计。

[0362] 示例性芯片上系统集成电路

[0363] 图25-图26B图示根据本文中所述的各实施例的可以使用一个或多个IP核制造的示例性集成电路和相关联的图形处理器。除了所图示的内容之外，还可以包括其他逻辑和电路，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。

[0364] 图25是图示根据实施例的可使用一个或多个IP核来制造的示例性芯片上系统集成电路2500的框图。示例性集成电路2500包括一个或多个应用处理器2505(例如，CPU)、至少一个图形处理器2510，并且可附加地包括图像处理器2515和/或视频处理器2520，其中的任一个都可以是来自相同设计设施或多个不同的设计设施的模块化IP核。集成电路2500包括外围或总线逻辑，包括USB控制器2525、UART控制器2530、SPI/SDIO控制器2535和I^2S/I^2C控制器2540。此外，集成电路可包括显示设备2545，该显示设备2548耦合至高清晰度多媒体接口(HDMI)控制器2550和移动行业处理器接口(MIPI)显示接口2555中的一个或多个。可以由闪存子系统2560(包括闪存和闪存控制器)来提供存储。可以经由存储器控制器2565来提供存储器接口以获得对SDRAM或SRAM存储器设备的访问。一些集成电路附加地包括嵌入式安全引擎2570。

[0365] 图26A-图26B是图示根据本文中所描述的实施例的用于在SoC内使用的示例性图形处理器的框图。图26A图示根据实施例的可以使用一个或多个IP核制造的芯片上系统集成电路的示例性图形处理器2610。图26B图示根据实施例的可以使用一个或多个IP核制造的芯片上系统集成电路的附加示例性图形处理器2640。图26A的图形处理器2610是低功率图形处理器核的示例。图26B的图形处理器2640是较高性能图形处理器核的示例。图形处理器2610、2640中的每一个都可以是图25的图形处理器2610的变体。

[0366] 如图26A中所示，图形处理器2610包括顶点处理器2605以及一个或多个片段处理器2615A-2615N(例如，2615A、2615B、2615C、2615D，一直到2615N-1和2615N)。图形处理器2610可以经由单独的逻辑执行不同的着色器程序，使得顶点处理器2605被优化以执行用于顶点着色器程序的操作，而一个或多个片段处理器2615A-2615N执行用于片段或像素着色器程序的片段(例如，像素)着色操作。顶点处理器2605执行3D图形流水线的顶点处理级，并生成基元数据和顶点数据。(多个)片段处理器2615A-2615N使用由顶点处理器2605生成的基元数据和顶点数据来产生被显示在显示设备上的帧缓冲器。在一个实施例中，(多个)片段处理器2615A-2615N被优化以执行如在OpenGL API中提供的片段着色器程序，这些片段

着色器程序可以用于执行与如在Direct 3D API中提供的像素着色器程序类似的操作。

[0367] 图形处理器2610附加地包括一个或多个存储器管理单元(MMU)2620A-2620B、(多个)高速缓存2625A-2625B以及(多个)电路互连2630A-2630B。这一个或多个MMU 2620A-2620B为图形处理器2610(包括为顶点处理器2605和/或(多个)片段处理器2615A-2615N)提供虚拟到物理地址映射,除了存储在一个或多个高速缓存2625A-2625B中的顶点数据或图像/纹理数据之外,该虚拟到物理地址映射还可以引用存储在存储器中的顶点数据或图像/纹理数据。在一个实施例中,一个或多个MMU 2620A-2620B可以与系统内的其他MMU同步,使得每个处理器2505-2520可以参与共享或统一的虚拟存储器系统,系统内的其他MMU包括与图25的一个或多个应用处理器2505、图像处理器2515和/或视频处理器2520相关联的一个或多个MMU。根据实施例,一个或多个电路互连2630A-2630B使得图形处理器2610能够经由SoC的内部总线或经由直接连接来与SoC内的其他IP核对接。

[0368] 如图26B中所示,图形处理器2640包括图26A的图形处理器2610的一个或多个MMU 2620A-2620B、高速缓存2625A-2625B、以及电路互连2630A-2630B。图形处理器2640包括一个或多个着色器核2655A-2655N(例如,2655A、2655B、2655C、2655D、2655E、2655F,一直到2655N-1和2655N),这一个或多个着色器核提供统一着色器核架构,在该统一着色器核架构中,单个核或类型或核可以执行所有类型的可编程着色器代码,包括用于实现顶点着色器、片段着色器和/或计算着色器的着色器程序代码。存在的着色器核的确切数量可以因实施例和实现方式而异。另外,图形处理器2640包括核间任务管理器2645,该核间任务管理器2645充当用于将执行线程分派给一个或多个着色器核2655A-2655N的线程分派器和用于加速对基于片的渲染的分片操作的分片单元2658,在基于片的渲染中,针对场景的渲染操作在图像空间中被细分,例如以利用场景内的局部空间一致性或优化内部高速缓存的使用。

[0369] 在多种多样的图像处理任务(诸如机器渲染的图像的去噪声)中已经充分利用了卷积神经网络(CNN)。核预测网络(KPN)是致力于光线跟踪去噪声的深度学习解决方案,其使用CNN来估计局部加权核以用于根据邻近的像素来计算去噪声的像素。

[0370] 图27A是示出常规的核预测网络(KPN)的示意图。核预测网络在下文被称为“KPN”。如图所示的KPN 2700可以是在计算引擎或适用于卷积神经网络或类似的深度学习神经网络的任何计算硬件上实现或部署的深度全卷积神经网络。

[0371] 在训练期间,利用事先收集的有噪声图像和有噪声图像的相应的真值图像的数据集来训练KPN 2700。可利用该数据集多次迭代地训练和调谐该KPN,以学习该KPN的网络参数,诸如批处理尺寸、节点信息、偏置等等。可在训练该KPN之前分配该KPN的卷积核的核尺寸。例如,可将该KPN的卷积核的核尺寸分配为k×k,其中k的值是从3、5、7、9、11、13、15、17、19、21或类似值之一选择的。

[0372] 在图像去噪声的运行时推断期间,KPN 2700可接收图像作为输入,并对其去噪声以生成去噪声的图像作为输出。该图像可具有多个像素。该KPN 2700可包括核生成逻辑2710和滤波逻辑2730。在接收该图像之后,该KPN 2700的核生成逻辑2710可对于该图像的多个像素中的每个像素生成卷积核。该卷积核可具有多个核值。对于该图像的每个像素,该KPN2700的滤波逻辑2730可使用由核生成逻辑2710生成的相应的卷积核的多个核值来对该像素滤波以获得去噪声的像素。在对该图像的多个像素的每一个滤波之后,KPN 2700可生成去噪声的图像作为输出。以下相对于图27B描述上述滤波的细节。

[0373] 图27B是示出通过常规的KPN对像素进行滤波的示例的示意图。如先前相对于图27A所讨论，对于该图像的每个像素，该KPN 2700的滤波逻辑2730可使用由核生成逻辑2710生成的相应的卷积核的多个核值来对该像素滤波以获得去噪声的像素。

[0374] 更具体地，在图27B的示例中，对于该图像的像素位置2760-0处的像素P0生成具有3×3的核尺寸的卷积核2770，并且卷积核2770具有九个核值W0至W8。如图27B中所示，对于像素P0，KPN 2700可以利用具有九个核值W0至W8的卷积核2770和九个像素位置2760-0、2760-1、2760-2、2760-3、2760-4、2760-5、2760-6、2760-7、2760-8的像素值来对像素P0滤波，上述九个像素位置位于以该图像的像素P0的像素位置2760-0为中心的正方形3×3窗口中。更具体地，可通过计算九个像素位置2760-0、2760-1、2760-2、2760-3、2760-4、2760-5、2760-6、2760-7、2760-8的像素值的加权平均值以作为去噪声的像素的像素值$\widehat{P0}$，来进行对像素P0的滤波。即，可通过以下公式来计算去噪声的像素的像素值$\widehat{P0}$。

[0375] $$\widehat{P0}=\frac{P0\times W0+P1\times W1+\cdots+P8\times W8}{W0+W1+\cdots+W8}$$

[0376] 通常在诸如图像去噪声的图像处理任务中，九个核值W0至W8的和可以等于一。

[0377] 虽然常规的KPN已经证实其在诸如光线跟踪去噪声的图像处理任务中的潜力，但由于采用了大尺寸的卷积核(例如，具有最常用的21×21核尺寸的卷积核)，可能不能在图像质量与计算成本之间提供良好的平衡。为了利用更少的计算成本来实现更好的或相似的图像质量，可引入偏移以自适应地确定在KPN的图像去噪声中使用的像素位置。下面将讨论细节。

[0378] 在一些实施例中，在诸如图像去噪声的图像处理任务中，作为KPN的替代，使用自适应可变形核预测网络(ADKPN)。一般而言，对于图像的每个像素，ADKPN不仅可以生成卷积核，而且可以生成偏移以用于确定在图像去噪声中使用的像素位置。该ADKPN可在对像素的像素的去噪声中使用具有偏移的像素位置的像素值，而不是使用如图27B中所示的位于以像素P0的像素位置为中心的正方形窗口中的固定像素位置的像素值。这样更智能的卷积核生成和滤波范式与常规的KPN相比可具有更好的去噪声性能。

[0379] 图28A是示出根据实施例的自适应可变形核预测网络(ADKPN)的示意图。自适应可变形核预测网络在下文被称为“ADKPN”。所示的ADKPN2800可以是在计算引擎或适用于卷积神经网络或相似的深度学习神经网络的任何计算硬件上实现或部署的卷积神经网络，上述计算硬件包括但不限于图形处理单元(GPU)、中央处理单元(CPU)、专用集成电路(ASIC)等等。

[0380] 在训练期间，利用事先收集的代表性的有噪声图像和有噪声图像的相应的真值图像的训练数据集来训练ADKPN 2800。例如，该训练数据集可包括大量代表性的图像，诸如通过蒙特卡罗渲染或任何其他图像渲染解决方案生成的机器合成的有噪声图像。作为进一步示例，该训练数据集可包括以128样本每像素(spp)渲染的测试图像，以及以8192spp或更高spp渲染的参考图像。样本每像素(spp)在本领域中已知为用于通过诸如蒙特卡罗渲染的随机采样渲染算法来渲染像素的样本的数量。Spp愈大，图像质量愈佳。

[0381] 可通过用于深度神经网络的典型训练方法来进行ADKPN 2800的训练过程，诸如随机梯度下降(SGD)、动量(Momentum)、自适应梯度(AdaGrad)、自适应矩估计(Adam)等等。可

利用该训练数据集多次迭代地训练和调谐该ADKPN 2800，以学习该ADKPN 2800的网络参数，诸如批处理尺寸、节点信息、偏置等等。

[0382] 在图像去噪声的运行时推断期间，ADKPN 2800可接收图像作为输入，并对其去噪声以生成去噪声的图像作为输出。该图像可以是任何图像，诸如通过蒙特卡罗渲染或任何其他图像渲染解决方案生成的机器合成的图像。该图像可包括多个像素。

[0383] 该ADKPN 2800可包括核生成逻辑2810、偏移生成逻辑2820和滤波逻辑2830。在接收图像之后，对于该图像的多个像素中的每一个，该ADKPN 2800的核生成逻辑2810可对于该像素生成具有多个核值的卷积核，并且该ADKPN 2800的偏移生成逻辑2820可对于该像素生成多个偏移，多个偏移分别对应于该卷积核的多个核值，其中多个偏移中的每一个用于指示从该像素的像素位置的偏离量。然后，对于该图像的多个像素中的每一个，该ADKPN 2800的滤波逻辑2830可基于该像素的像素位置和由偏移生成逻辑2820对于该像素生成的多个偏移来确定多个偏离的像素位置，并使用由核生成逻辑2810对于该像素生成的卷积核以及多个偏离的像素位置的像素值来对该像素滤波，以获得去噪声的像素。在对该图像的多个像素的每一个滤波之后，该ADKPN 2800可生成去噪声的图像作为输出。以下相对于图28B描述上述滤波的细节。

[0384] 图28B是示出根据实施例的通过ADKPN对像素进行滤波的示例的示意图。如先前相对于图28A所讨论，该ADKPN 2800的滤波逻辑2830可基于像素的像素位置和对于该像素生成的多个偏移来确定多个偏离的像素位置，并使用对于该像素生成的卷积核以及多个偏离的像素位置的像素值来对该像素滤波，以获得去噪声的像素。

[0385] 更具体地，在图28B的示例中，对于该图像的像素位置2840处的像素P0生成具有3×3的核尺寸的卷积核2870，该卷积核2870具有九个核值W0至W8。还对于该像素P0生成九个偏移2850-0、2850-1、2850-2、2850-3、2850-4、2850-5、2850-6、2850-7和2850-8，这九个偏移2850-0、2850-1、2850-2、2850-3、2850-4、2850-5、2850-6、2850-7和2850-8分别对应于九个核值W0至W8。如图28B中所示，对于像素P0，该ADKPN 2800的滤波逻辑2830可基于该像素P0的像素位置2840和九个偏移2850-0、2850-1、2850-2、2850-3、2850-4、2850-5、2850-6、2850-7和2850-8来确定该图像的九个偏离的像素位置2860-0、2860-1、2860-2、2860-3、2860-4、2860-5、2860-6、2860-7和2860-8，该九个偏离的像素位置在图28B中被标记为P0'-P8'。然后该ADKPN 2800的滤波逻辑2830可利用具有九个核值W0至W8的卷积核2870和九个偏离的像素位置2860-0、2860-1、2860-2、2860-3、2860-4、2860-5、2860-6、2860-7和2860-8的像素值来对像素P0滤波，以获得去噪声的像素的像素值$\widehat{P0}$。更具体地，可通过计算该图像的九个偏离的像素位置2860-0、2860-1、2860-2、2860-3、2860-4、2860-5、2860-6、2860-7、2860-8的像素值的加权平均值以作为去噪声的像素的像素值$\widehat{P0}$，来进行对像素P0的滤波。即，可通过以下公式来计算去噪声的像素的像素值$\widehat{P0}$。

[0386] $$\widehat{P0}=\frac{P0'\times W0+P1'\times W1+\cdots+P8'\times W8}{W0+W1+\cdots+W8}$$

[0387] 在一些实施例中，对于该图像的像素生成的卷积核的多个核值的和可以是一。对于图28B的示例，卷积核2870的九个核值W0至W8的和可以是一。

[0388] 在一些实施例中，对于图像的该像素生成的多个偏移中的每一个可包括位置值以

用于指示从该像素的像素位置的偏离量。例如，九个偏移2850-0、2850-1、2850-2、2850-3、2850-4、2850-5、2850-6、2850-7和2850-8中的每一个可包括位置值以用于指示从如图28B中所示的像素位置2840的偏离量。例如，偏移2850-0可包括位置值(u0,v0)以用于指示从像素位置2840的偏离量。令像素位置2840由位置值(x,y)表示，可基于像素位置2840和偏移2850-0通过位置值(x+u0,y+v0)来确定像素位置2860-0。

[0389] 在一些实施例中，该位置值可包括浮点值。例如，上述偏移2850-0可包括浮点值(u0,v0)。相应地，作为非限制性示例，可通过本领域已知的像素插值(诸如最近邻插值、双线性插值等等)来确定具有位置值(x+u0,y+v0)的像素位置2860-0的像素值。

[0390] 在一些实施例中，对于该图像的至少两个像素的卷积核的多个核值可以是不同的。

[0391] 在一些实施例中，对于该图像的至少两个像素的多个偏移可以是不同的。

[0392] 在一些实施例中，多个偏移可以在卷积核生成之前或与卷积核生成同时被生成。例如，在接收图像之后，可以首先对于该图像的多个像素中的每一个生成多个偏移，然后对于该图像的多个像素中的每一个生成卷积核。替代地，可以对于该图像的多个像素中的每一个同时地生成多个偏移和卷积核。

[0393] 在一些实施例中，上述偏离量的上限可以是预定义的(例如在卷积神经网络的训练之前)。偏离量的上限可以是预定义的，以用于控制ADKPN2800的训练/学习过程的计算成本。例如，对于ADKPN 2800的卷积核的3×3核尺寸，偏离量的上限可被预定义为不大于11像素。

[0394] 在一些实施例中，卷积核的核尺寸可以是预分配的(例如在卷积神经网络的训练之前)。在一些实施例中，核尺寸可被预分配为3×3、5×5、7×7、……、(2n-1)×(2n-1)中的一个，其中n是正整数，n>＝2且n<min(floor(W/2),floor(H/2))，其中W和H是在训练中使用的图像的经标准化的宽度和高度或在运行时推断中要去噪声的图像的宽度和高度。上述数学函数“floor()”指的是本领域已知的向下舍入操作。

[0395] 图28C是示出常规的KPN与ADKPN之间的训练和/或测试性能度量的比较的曲线图。具体地，如图28C中所示的曲线图描绘了常规KPN和ADKPN的相对于计算迭代(横轴)的损失(纵轴)。可以清楚看出，相对于常规KPN，ADKPN利用相似的计算迭代能产生更好的结果，即更少的损失。换言之，相对于常规KPN，ADKPN能利用更少的计算成本实现相似结果。

[0396] 图29是示出根据实施例的通过ADKPN对图像进行去噪声的方法的流程图。

[0397] 通过ADKPN对图像进行去噪声的方法2900可以实现在计算引擎或适用于卷积神经网络或相似的深度学习神经网络的任何计算硬件上，上述计算硬件包括但不限于图形处理单元(GPU)、中央处理单元(CPU)、专用集成电路(ASIC)等等。该图像可包括多个像素。在一些实施例中，该图像可以是任何图像，诸如通过蒙特卡罗渲染或任何其他图像渲染解决方案生成的机器合成的图像。

[0398] 该方法2900可包括对于该图像的多个像素中的每一个像素的操作2910-2940。

[0399] 在框2910，可对于该像素生成具有多个核值的卷积核。

[0400] 在框2920，可对于该像素生成多个偏移，多个偏移分别对应于多个核值，多个偏移中的每一个用于指示从该像素的像素位置的偏离量。

[0401] 在框2930，可基于该像素的像素位置和多个偏移来确定多个偏离的像素位置。

[0402] 在框2940,可利用该卷积核和多个偏离的像素位置的像素值来对该像素滤波,以获得去噪声的像素。

[0403] 在一些实施例中,多个偏移中的每一个可包括位置值以用于指示从该像素的像素位置的偏离量。

[0404] 在一些实施例中,该位置值可包括浮点值。

[0405] 在一些实施例中,对于该图像的至少两个像素的多个核值可以是不同的。

[0406] 在一些实施例中,对于该图像的至少两个像素的多个偏移可以是不同的。

[0407] 在一些实施例中,多个偏移可以在卷积核生成之前或与卷积核生成同时被生成。

[0408] 在一些实施例中,利用卷积核和多个偏离的像素位置的像素值对像素进行滤波可包括:将该卷积核的多个核值应用于多个偏离的像素位置的像素值,以获得多个偏离的像素位置的像素值的加权平均值。

[0409] 在一些实施例中,偏离量的上限可以是预定义的(例如在卷积神经网络的训练之前)。

[0410] 在一些实施例中,卷积核的核尺寸可例如在卷积神经网络的训练之前被预分配为3×3、5×5、7×7、……、(2n-1)×(2n-1)中的一个,其中n是正整数,n>=2且n<min(floor(W/2),floor(H/2)),W和H是图像的宽度和高度。上述数学函数“floor()”指的是本领域已知的向下舍入操作。

[0411] 图30A-30D分别示出输入图像、参考图像、通过常规的KPN产生的输出图像、以及根据实施例的通过ADKPN产生的输出图像。

[0412] 图30A示出以128样本每像素(spp)通过机器合成的输入图像。图30B示出以8192spp通过机器合成的参考图像。样本每像素(spp)在本领域中已知为用于通过诸如蒙特卡罗渲染的随机采样渲染算法来渲染像素的样本的数量。Spp愈大,图像质量愈佳。可以看出,图30A的输入图像是有噪声的,而图30B的参考图像具有高质量。

[0413] 图30C示出通过常规KPN利用9×9的核尺寸对于图30A的输入图像去噪声产生的输出图像。可以看出,在椭圆形3002内的区域中存在清楚的伪影。图30D示出根据实施例的通过ADKPN利用5×5的核尺寸对于图30A的输入图像去噪声产生的输出图像。可以看出,与图30C中的输出图像形成对比,图30D中的图像显示出与图30B中的无伪影的参考图像相似的质量。

[0414] 相似地,图31A示出以128样本每像素(spp)通过机器合成且有噪声的输入图像。图31B示出以8192spp通过机器合成且具有高质量的参考图像。图31C示出通过常规KPN利用9×9的核尺寸对于图31A的输入图像去噪声产生的输出图像。可以看出,在椭圆形3102、3104和3106内的区域中存在清楚的伪影。图31D示出根据实施例的通过ADKPN利用5×5的核尺寸对于图31A的输入图像去噪声产生的输出图像。可以看出,与图31C中的输出图像形成对比,图31D中的图像显示出与图31B中的无伪影的参考图像相似的质量。

[0415] 相似地,图32A示出以128样本每像素(spp)通过机器合成且有噪声的输入图像。图32B示出以8192spp通过机器合成且具有高质量的参考图像。图32C示出通过常规KPN利用9×9的核尺寸对于图32A的输入图像去噪声产生的输出图像。可以看出,在椭圆形3202和3204内的区域中存在清楚的伪影。图32D示出根据实施例的通过ADKPN利用5×5的核尺寸对于图32A的输入图像去噪声产生的输出图像。可以看出,与图32C中的输出图像形成对比,图

32D中的图像显示出与图32B中的无伪影的参考图像相似的质量。

[0416] 相似地，图33A示出以128样本每像素(spp)通过机器合成且有噪声的输入图像。图33B示出以8192spp通过机器合成且具有高质量的参考图像。图33C示出通过常规KPN利用9×9的核尺寸对于图33A的输入图像去噪声产生的输出图像。可以看出，在椭圆形3302内的区域中存在清楚的伪影。图33D示出根据实施例的通过ADKPN利用5×5的核尺寸对于图33A的输入图像去噪声产生的输出图像。可以看出，与图33C中的输出图像形成对比，图33D中的图像显示出与图33B中的无伪影的参考图像相似的质量。

[0417] 此外，图30C-30D、图31C-31D、图32C-32D以及图33C-33D还分别示出了输出图像的结构不相似度(DSSIM)指数。本领域已知的DSSIM是结构相似度(SSIM)的变型，其测量图像的不相似度。DSSIM指数用于比较输出图像与真值图像之间的结构差别。DSSIM指数愈小，输出图像的质量愈好。例如，图30C的通过KPN得到的输出图像的DSSIM指数可以是0.0672，而图30D的通过ADKPN得到的输出图像的DSSIM指数可以是0.0610。从图30C-30D、图31C-31D、图32C-32D以及图33C-33D可以看出，ADKPN产生的输出图像全都具有比常规KPN产生的输出图像更小的DSSIM指数。换言之，ADKPN产生的输出图像全都具有比常规KPN产生的输出图像更好的质量。

[0418] 因此，可以清楚看出利用更小核尺寸的ADKPN的性能显著好于利用更大核尺寸的KPN的性能。

[0419] 各实施例的某些部分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在其上存储了计算机程序指令的计算机可读介质，计算机程序指令可以被用来对计算机(或其他电子设备)进行编程，以由一个或多个处理器执行，以根据某些实施例执行过程。计算机可读介质可包括，但不限于磁盘、光盘、只读存储器(ROM)、随机存取存储器(RAM)、可擦除可编程只读存储器(EPROM)、电可擦除可编程只读存储器(EEPROM)、磁卡或光卡、闪存、或适于存储电子指令的其他类型的计算机可读介质。此外，实施例还可以作为计算机程序产品下载，其中，程序可以从远程计算机传递到请求计算机。在一些实施例中，非瞬态计算机可读存储介质具有存储在其上的表示指令序列的数据，所述指令序列在由处理器执行时使处理器执行某些操作。

[0420] 许多方法是以它们的最基本形式描述的，但是，可以将过程添加到方法中的任一种中或从其中删除过程，并且可以向所描述的消息中的任何一个添加信息或从其中减去信息，而不会偏离本实施例的基本范围。对所属领域的技术人员而言将显而易见的是，可以作出许多进一步的修改和改编。特定实施例不是被提供用于限制概念而是为了例示概念。实施例的范围不是由以上所提供的具体示例来确定的，而仅由所附权利要求确定。

[0421] 如果说元件“A”耦合至元件“B”或者元件“A”与元件“B”耦合，则元件A可以直接耦合至元件B，或通过例如元件C间接地耦合。当说明书或权利要求书陈述组件、特征、结构、过程、或特性A“导致”组件、特征、结构、过程、或特性B时，意味着，“A”至少是“B”的部分原因，但是还可以有有助于导致“B”的至少一个其他组件、特征、结构、过程或特性。如果说明书指示组件、特征、结构、过程或特性“可能”、“可以”或“可”被包括，则该特定组件、特征、结构、过程或特性不是必须被包括。如果说明书或权利要求书引用一个(“a”或“an”)元素，则这并不意味着只有所描述的元素中的仅一个。

[0422] 实施例是实现方式或示例。说明书中对“实施例”、“一个实施例”、“一些实施例”、

或“其他实施例”的引用意味着结合实施例所描述的特定特征、结构或特性被包括在至少一些实施例中，但不一定包括在所有实施例中。各处出现的“实施例”、“一个实施例”或“一些实施例”不一定都指相同的实施例。应该理解，在对示例性实施例的以上描述中，出于使本公开变得流畅并帮助理解各新颖方面中的一个或多个方面的目的，各个特征有时被一起编组在单个实施例、附图、或其描述中。然而，本公开的方法不应被解释成反映带有所要求保护的实施例需要比每项权利要求中所明确记载的更多特征的意图。相反，如以下权利要求反映的，新颖性方面存在于比单个前述公开的实施例的所有特征更少的特征中。因此，权利要求藉此被明确纳入该说明书中，其中每一项权利要求独自作为分开的实施例。

[0423] 以下条款和/或示例涉及进一步的实施例或示例。可在一个或多个实施例中的任何地方使用示例中的细节。能以各种方式将不同的实施例或示例的各种特征与所包括的一些特征以及被排除的其他特征组合以适应各种不同的应用。示例可以包括主题，诸如：方法；用于执行所述方法的动作的装置；至少一种包括指令的机器可读介质，所述指令当由机器执行时使所述机器执行所述方法的动作；或用于根据本文中所描述的实施例和示例促进混合通信的设备或系统。

[0424] 进一步的示例

[0425] 示例1包括通过在计算引擎上实现的卷积神经网络对图像去噪声的方法，该图像包括多个像素，该方法包括：对于该图像的多个像素中的每一个，对于该像素生成具有多个核值的卷积核；对于该像素生成多个偏移，多个偏移分别对应于多个核值，多个偏移中的每一个用于指示从该像素的像素位置的偏离量；基于该像素的像素位置和多个偏移来确定多个偏离的像素位置；以及利用该卷积核和多个偏离的像素位置的像素值来对该像素滤波，以获得去噪声的像素。

[0426] 示例2包括示例1的主题，其中多个偏移中的每一个包括位置值以用于指示从该像素的像素位置的偏离量。

[0427] 示例3包括示例2的主题，其中该位置值包括浮点值。

[0428] 示例4包括示例1的主题，其中对于该图像的至少两个像素的多个核值是不同的。

[0429] 示例5包括示例1的主题，其中对于该图像的至少两个像素的多个偏移是不同的。

[0430] 示例6包括示例1的主题，其中多个偏移是在该卷积核生成之前或与该卷积核生成同时生成的。

[0431] 示例7包括示例1的主题，其中利用该卷积核和多个偏离的像素位置的像素值来对该像素滤波包括：将该卷积核的多个核值应用于多个偏离的像素位置的像素值，以获得像素值的加权平均值。

[0432] 示例8包括示例1的主题，其中偏离量的上限是预定义的。

[0433] 示例9包括示例1的主题，其中该卷积核的核尺寸被预分配为3×3、5×5、7×7、……和(2n-1)×(2n-1)中的一个，其中n是正整数，n>＝2且n<min(floor(W/2)，floor(H/2))，W和H是该图像的宽度和高度。

[0434] 示例10包括一种用于对图像去噪声的装置，该装置包括：数据存储，用于存储包括图像的数据，该图像包括多个像素；以及计算引擎，耦合至该数据存储，该计算引擎用于通过卷积神经网络对该图像去噪声，该计算引擎用于：对于该图像的多个像素中的每一个，对于该像素生成具有多个核值的卷积核；对于该像素生成多个偏移，多个偏移分别对应于多

个核值，多个偏移中的每一个用于指示从该像素的像素位置的偏离量；基于该像素的像素位置和多个偏移来确定多个偏离的像素位置；以及利用该卷积核和多个偏离的像素位置的像素值来对该像素滤波，以获得去噪声的像素。

[0435] 示例11包括示例10的主题，其中多个偏移中的每一个包括位置值以用于指示从该像素的像素位置的偏离量。

[0436] 示例12包括示例11的主题，其中该位置值包括浮点值。

[0437] 示例13包括示例10的主题，其中对于该图像的至少两个像素的多个核值是不同的。

[0438] 示例14包括示例10的主题，其中对于该图像的至少两个像素的多个偏移是不同的。

[0439] 示例15包括示例10的主题，其中多个偏移是在该卷积核生成之前或与该卷积核生成同时生成的。

[0440] 示例16包括示例10的主题，其中利用该卷积核和多个偏离的像素位置的像素值来对该像素滤波包括：将该卷积核的多个核值应用于多个偏离的像素位置的像素值，以获得像素值的加权平均值。

[0441] 示例17包括示例10的主题，其中偏离量的上限是预定义的。

[0442] 示例18包括示例10的主题，其中该卷积核的核尺寸被预分配为3×3、5×5、7×7、……和(2n-1)×(2n-1)中的一个，其中n是正整数，n>＝2且n<min(floor(W/2),floor(H/2))，W和H是该图像的宽度和高度。

[0443] 示例19包括一种设备，该设备包括用于执行如示例1-9中任一项中所述的方法的装置。

[0444] 示例20包括一种机器可读介质，包括多条指令，多条指令响应于在计算设备上被执行而导致该计算设备执行如示例1-9中的任一项中所述的方法。

[0445] 前述描述和附图应当被认为是说明性的，而不是限制性的。本领域技术人员将理解，可对本文中所描述的实施例作出各种修改和改变，而不背离如所附权利要求中所阐述的本发明的更宽泛精神和范围。

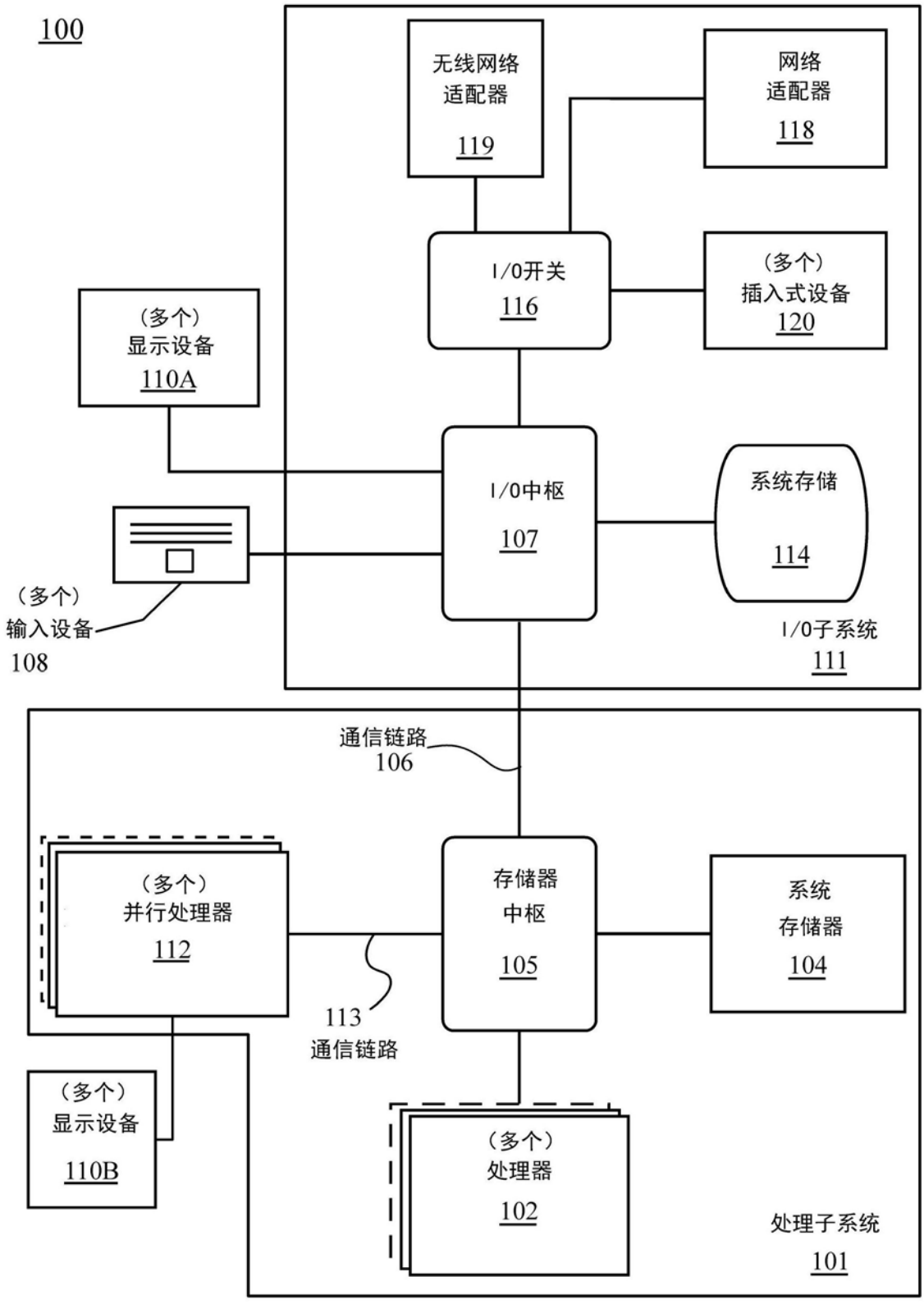

100
无线网络
适配器
119
网络
适配器
118
I/O开关
116
(多个)
插入式设备
120
(多个)
显示设备
110A
I/O中枢
107
系统存储
114
(多个)
输入设备
108
I/O子系统
111
通信链路
106
(多个)
并行处理器
112
存储器
中枢
105
系统
存储器
104
113
通信链路
(多个)
显示设备
110B
(多个)
处理器
102
处理子系统
101

图1

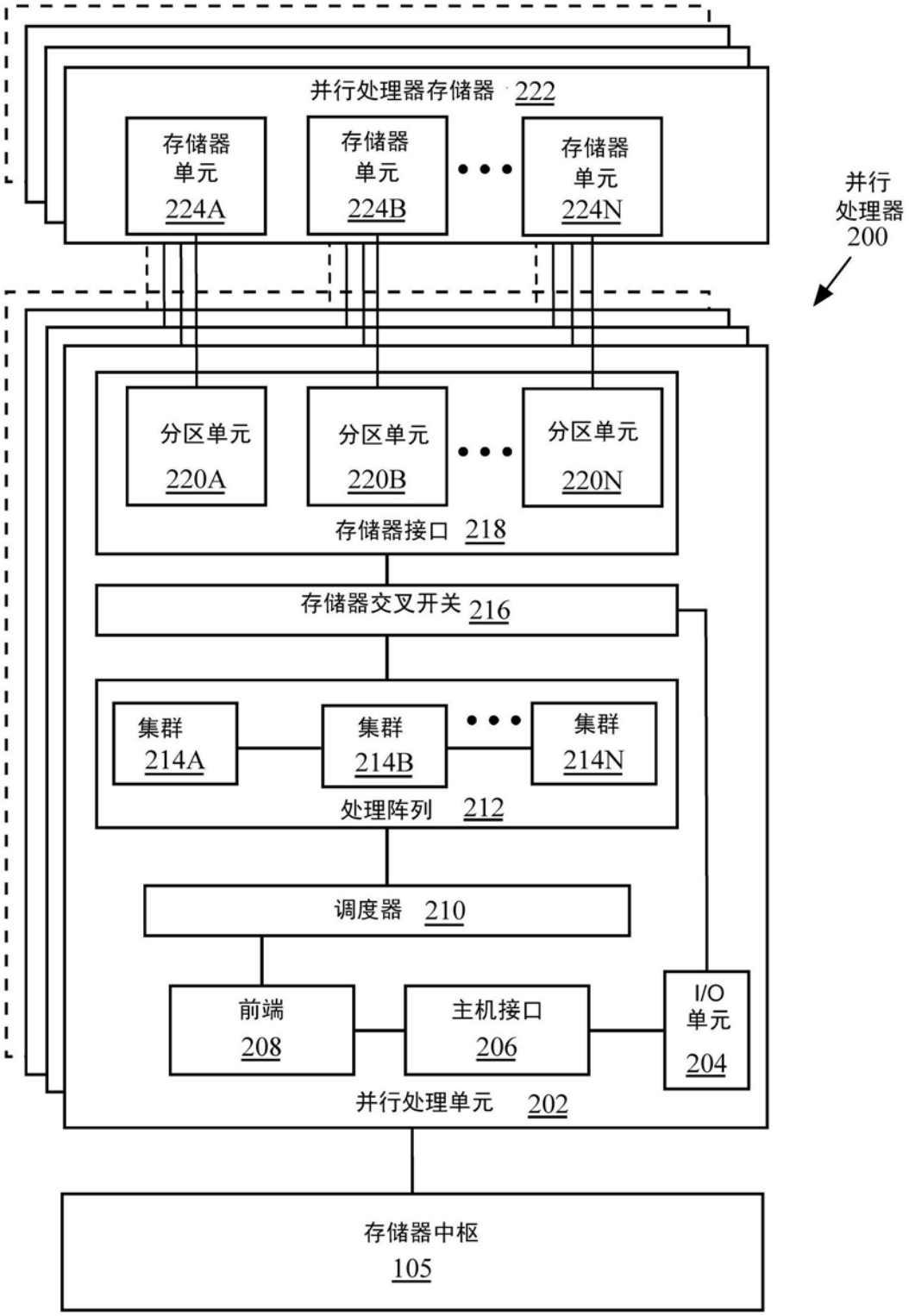

并行处理器存储器 222
存储器单元 224A
存储器单元 224B
存储器单元 224N
并行处理器 200
分区单元 220A
分区单元 220B
分区单元 220N
存储器接口 218
存储器交叉开关 216
集群 214A
集群 214B
集群 214N
处理阵列 212
调度器 210
前端 208
主机接口 206
I/O 单元 204
并行处理单元 202
存储器中枢 105

图2A

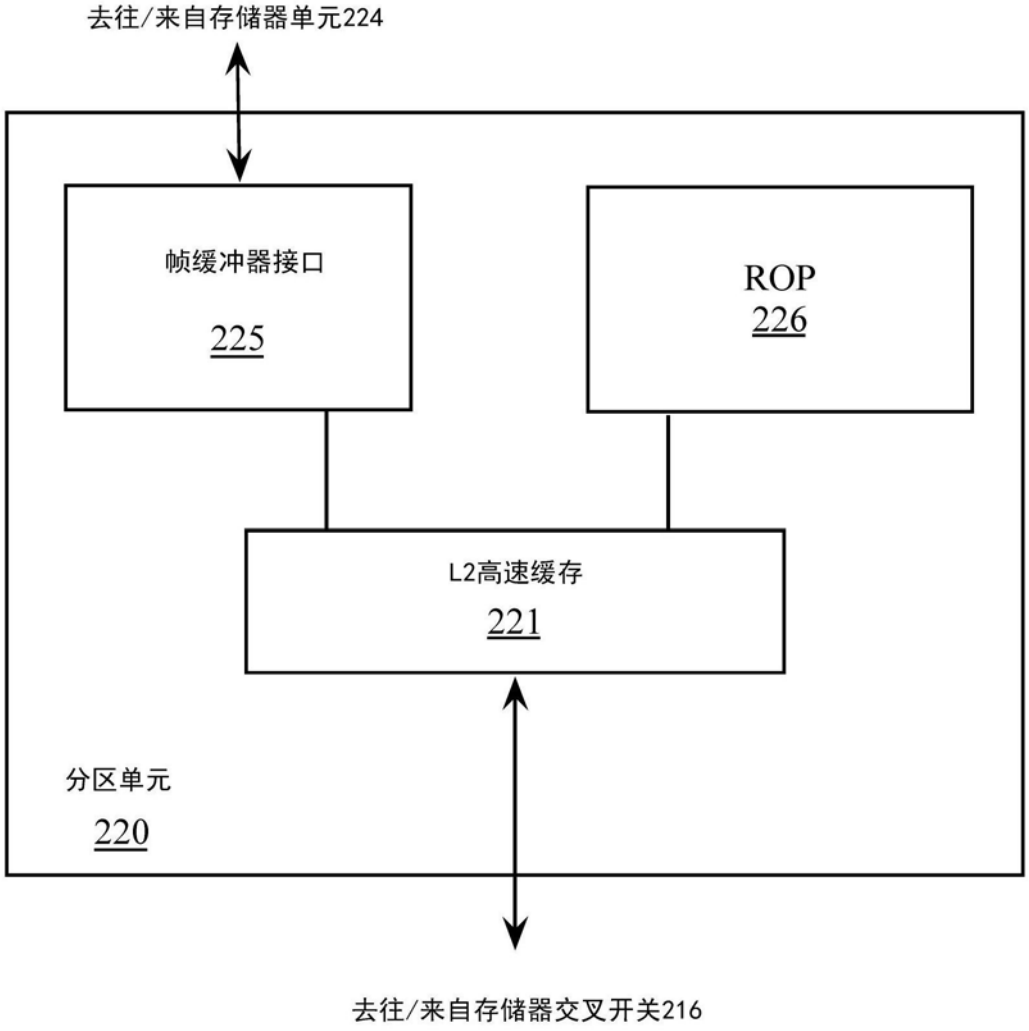

去往/来自存储器单元224
帧缓冲器接口
225
ROP
226
L2高速缓存
221
分区单元
220
去往/来自存储器交叉开关216

图2B

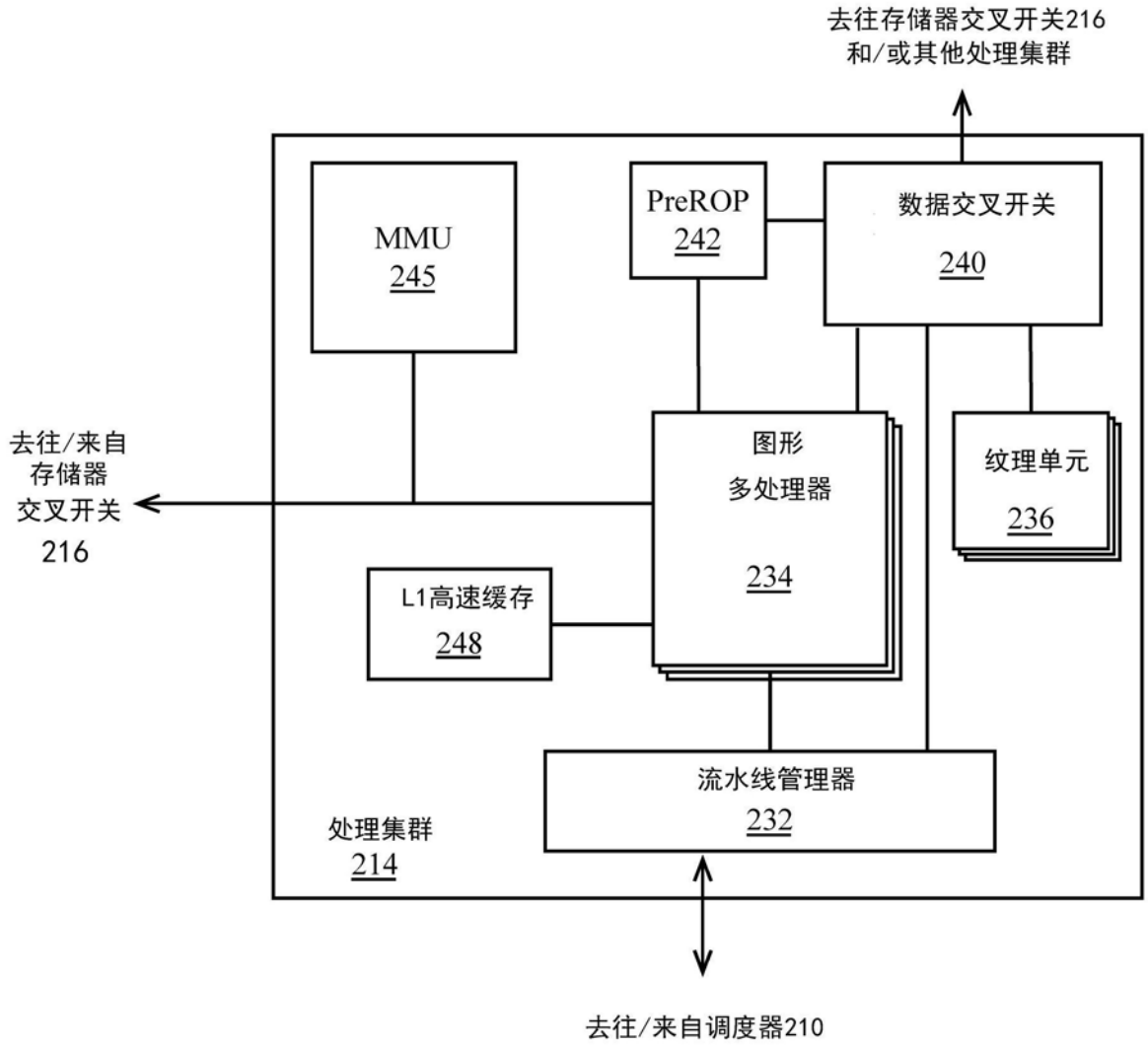

去往存储器交叉开关216
和/或其他处理集群
MMU
245
PreROP
242
数据交叉开关
240
去往/来自
存储器
交叉开关
216
图形
多处理器
234
纹理单元
236
L1高速缓存
248
流水线管理器
232
处理集群
214
去往/来自调度器210

图2C

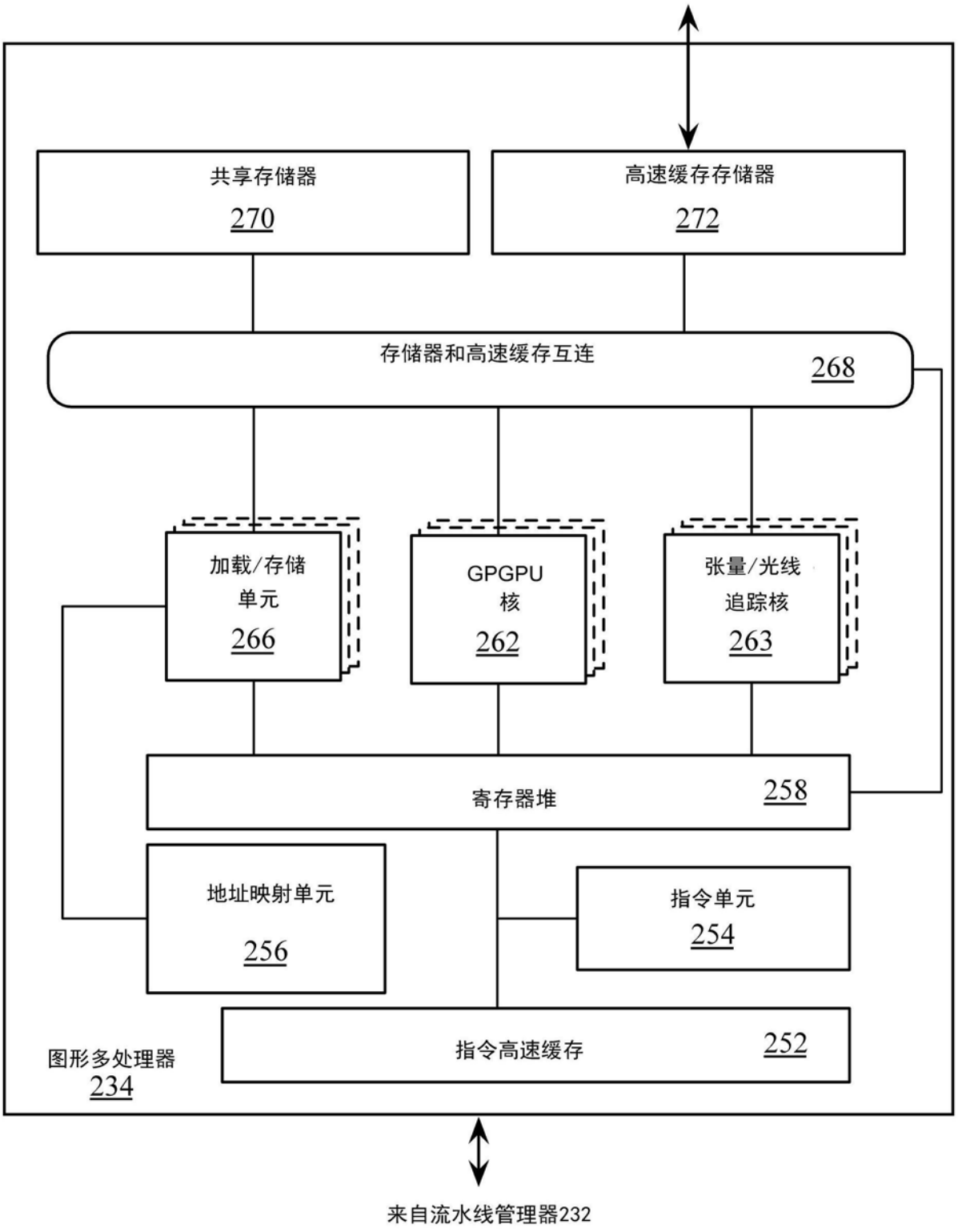

共享存储器
270
高速缓存存储器
272
存储器和高速缓存互连
268
加载/存储
单元
266
GPGPU
核
262
张量/光线
追踪核
263
寄存器堆
258
地址映射单元
256
指令单元
254
指令高速缓存
252
图形多处理器
234
来自流水线管理器232

图2D

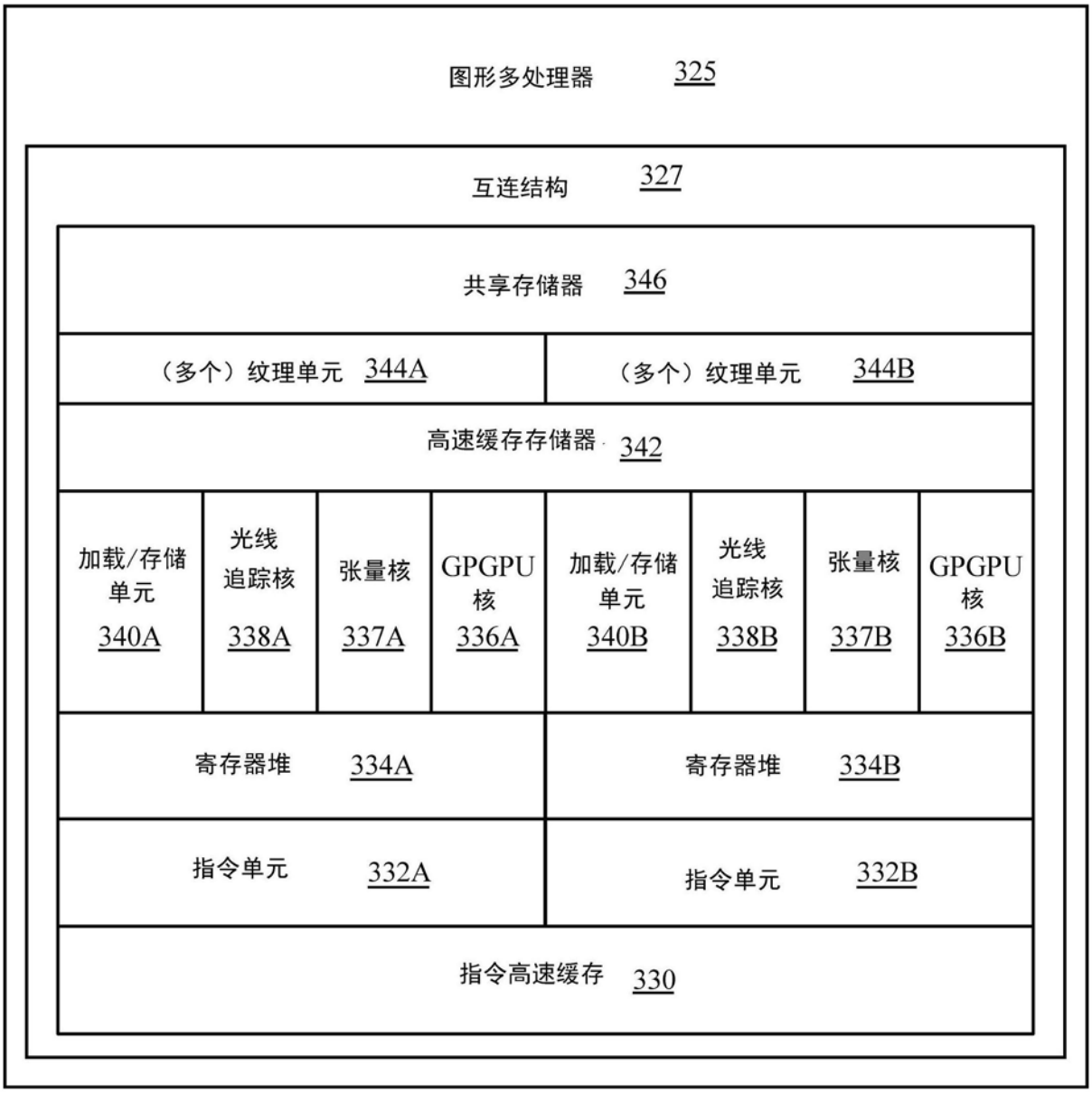

图形多处理器 325
互连结构 327
共享存储器 346
（多个）纹理单元 344A
（多个）纹理单元 344B
高速缓存存储器 342
加载/存储单元 340A
光线追踪核 338A
张量核 337A
GPGPU核 336A
加载/存储单元 340B
光线追踪核 338B
张量核 337B
GPGPU核 336B
寄存器堆 334A
寄存器堆 334B
指令单元 332A
指令单元 332B
指令高速缓存 330

图3A

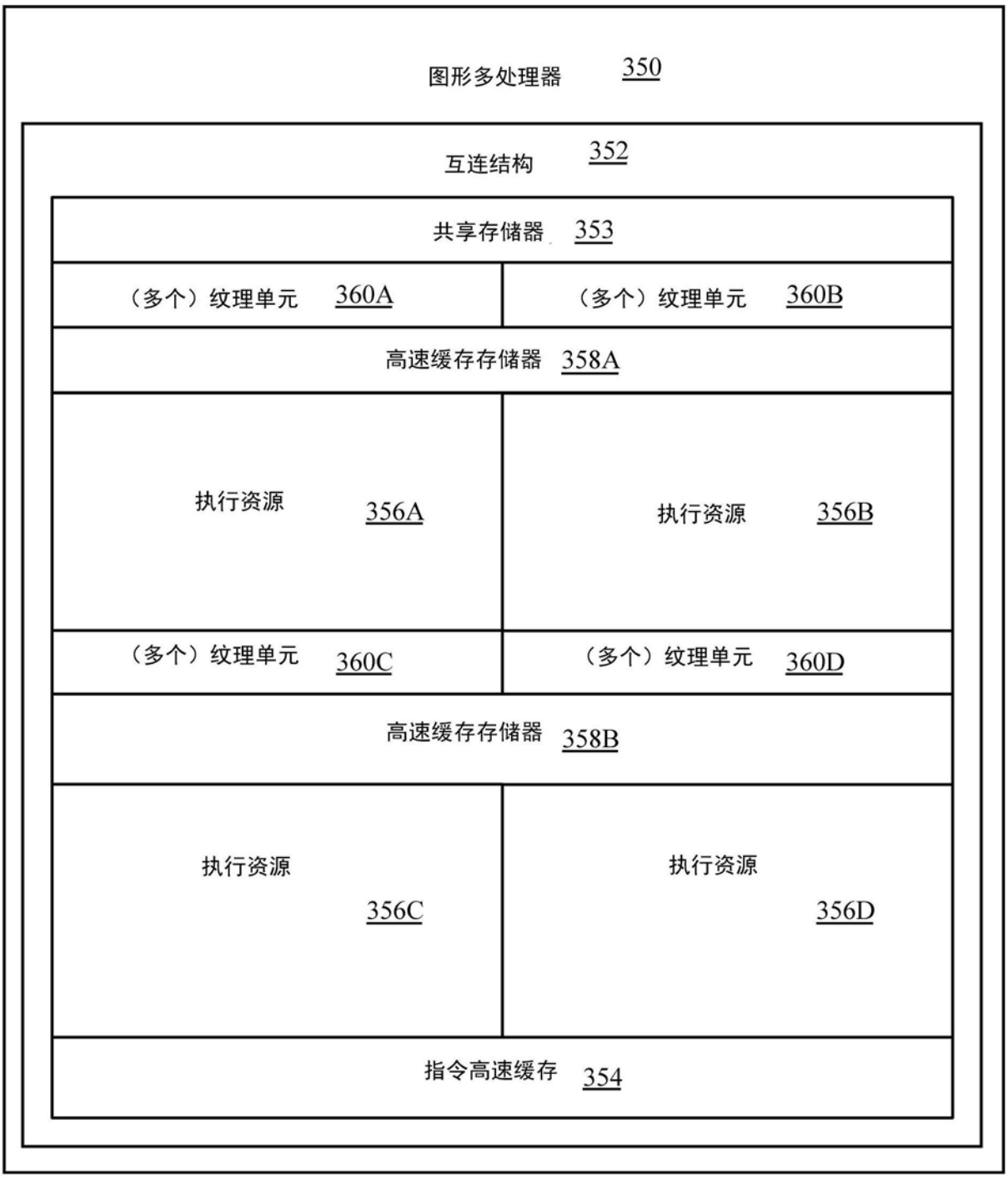

图形多处理器 350
互连结构 352
共享存储器 353
（多个）纹理单元 360A
（多个）纹理单元 360B
高速缓存存储器 358A
执行资源 356A
执行资源 356B
（多个）纹理单元 360C
（多个）纹理单元 360D
高速缓存存储器 358B
执行资源 356C
执行资源 356D
指令高速缓存 354

图3B

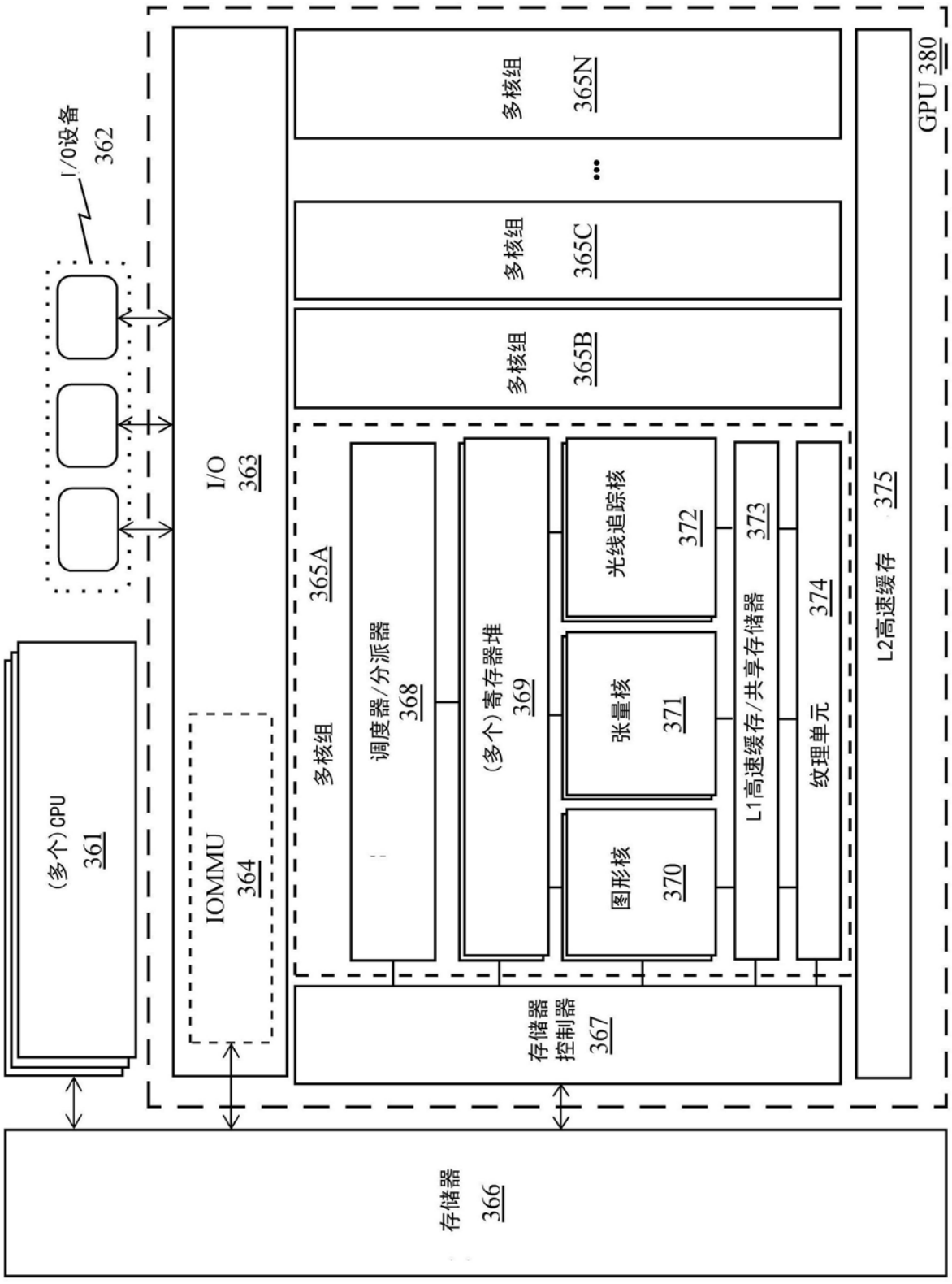

GPU 380
I/O设备 362
I/O 363
(多个)CPU 361
IOMMU 364
多核组 365A
多核组 365B
多核组 365C
多核组 365N
调度器/分派器 368
(多个)寄存器堆 369
图形核 370
张量核 371
光线追踪核 372
L1高速缓存/共享存储器 373
纹理单元 374
L2高速缓存 375
存储器控制器 367
存储器 366

图3C

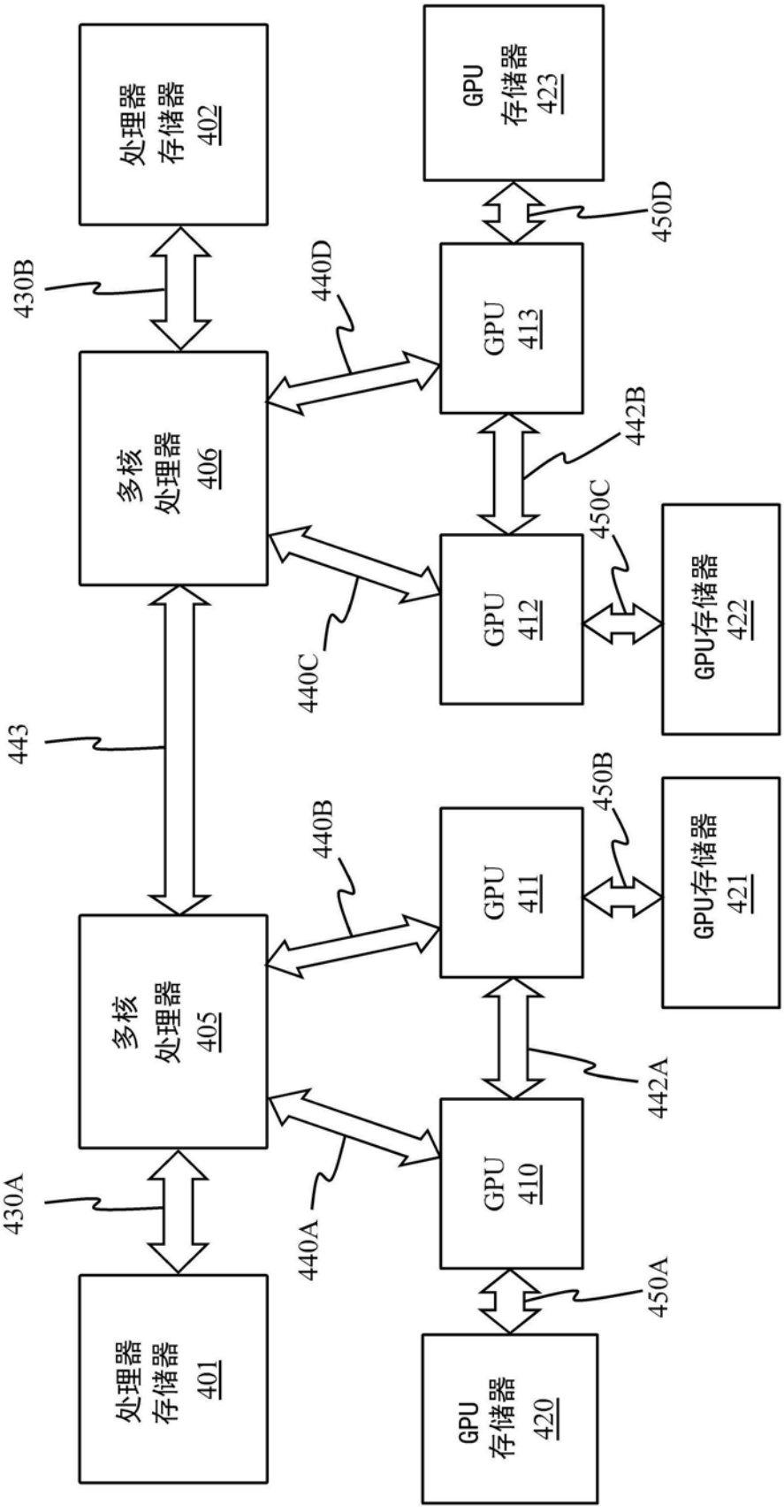

处理器存储器 402
GPU 存储器 423
450D
430B
440D
GPU 413
多核处理器 406
442B
450C
GPU 412
GPU存储器 422
440C
443
440B
450B
GPU 411
GPU存储器 421
多核处理器 405
442A
430A
440A
GPU 410
450A
处理器存储器 401
GPU 存储器 420

图4A

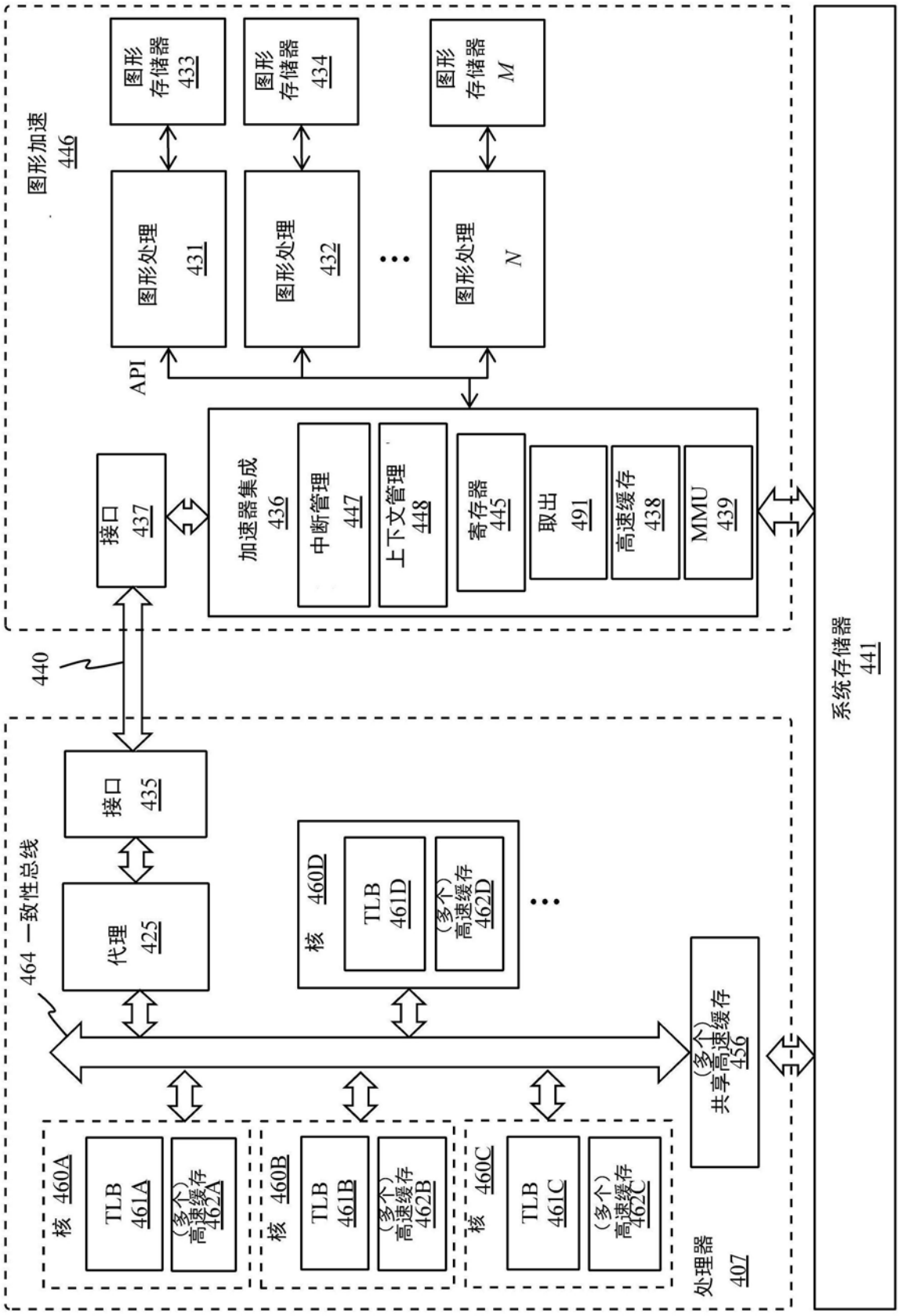

图形加速 446
图形存储器 433
图形存储器 434
图形存储器 M
图形处理 431
图形处理 432
图形处理 N
API
接口 437
加速器集成 436
中断管理 447
上下文管理 448
寄存器 445
取出 491
高速缓存 438
MMU 439
440
接口 435
代理 425
464 一致性总线
核 460D
TLB 461D
(多个)高速缓存 462D
(多个)共享高速缓存 456
核 460A
TLB 461A
(多个)高速缓存 462A
核 460B
TLB 461B
(多个)高速缓存 462B
核 460C
TLB 461C
(多个)高速缓存 462C
处理器 407
系统存储器 441

图4B

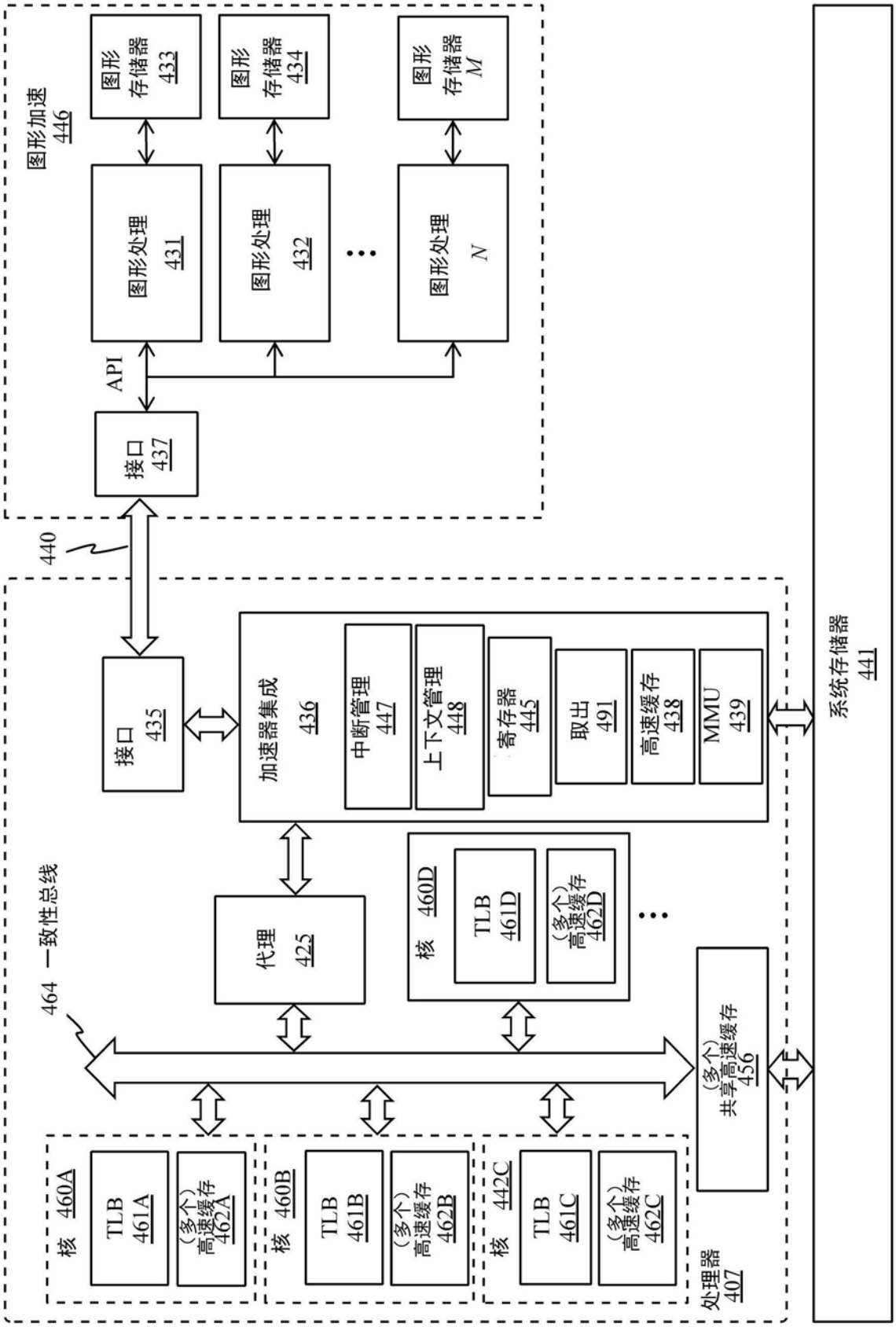

图形加速
446
图形
存储器
433
图形
存储器
434
图形
存储器
M
图形处理
431
图形处理
432
图形处理
N
API
接口
437
440
接口
435
加速器集成
436
中断管理
447
上下文管理
448
寄存器
445
取出
491
高速缓存
438
MMU
439
系统存储器
441
464 一致性总线
代理
425
核 460D
TLB
461D
(多个)
高速缓存
462D
(多个)
共享高速缓存
456
核 460A
TLB
461A
(多个)
高速缓存
462A
核 460B
TLB
461B
(多个)
高速缓存
462B
核 442C
TLB
461C
(多个)
高速缓存
462C
处理器
407

图4C

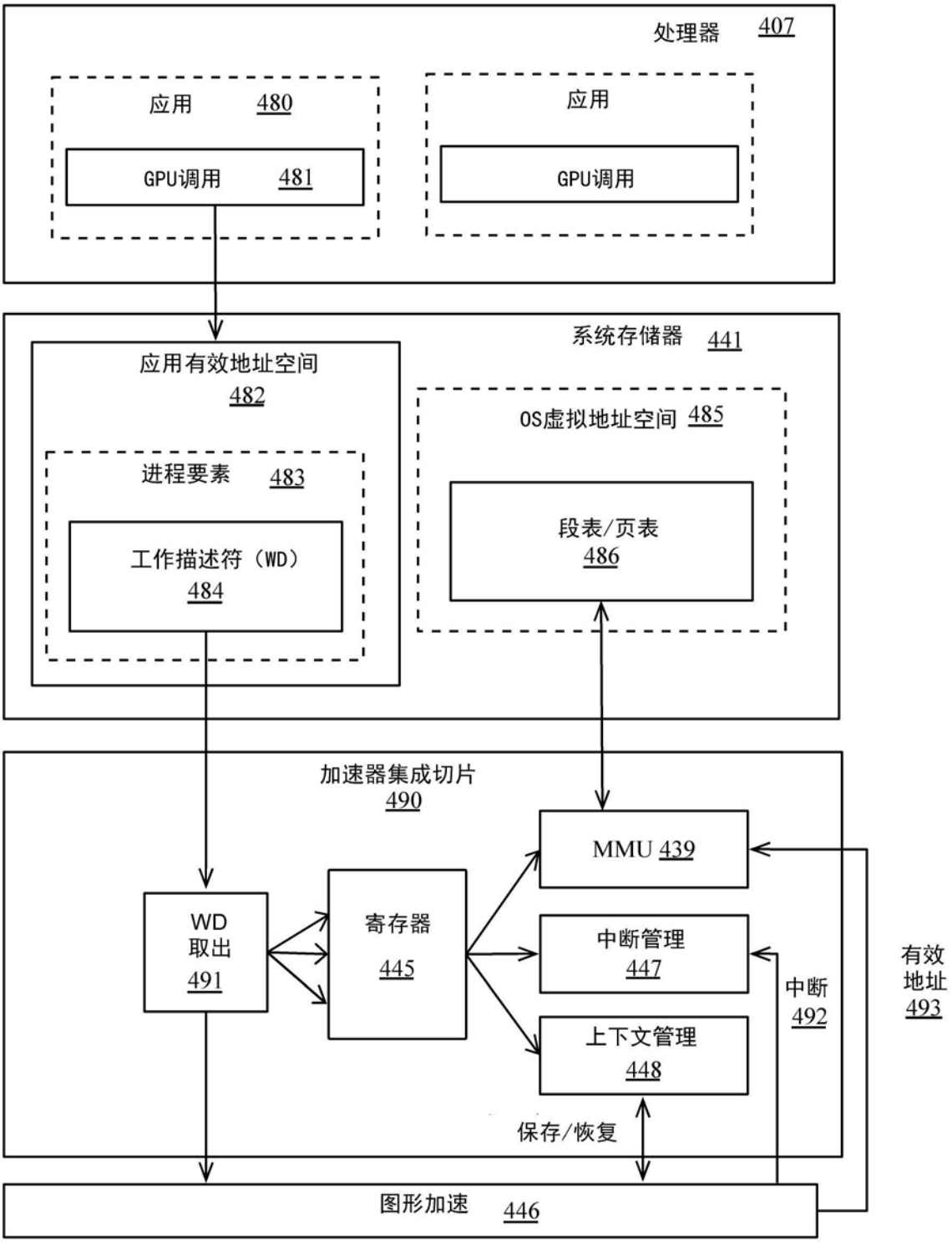

处理器 407
应用 480
GPU调用 481
应用
GPU调用
系统存储器 441
应用有效地址空间 482
进程要素 483
工作描述符（WD） 484
OS虚拟地址空间 485
段表/页表 486
加速器集成切片 490
MMU 439
WD 取出 491
寄存器 445
中断管理 447
上下文管理 448
中断 492
有效地址 493
保存/恢复
图形加速 446

图4D

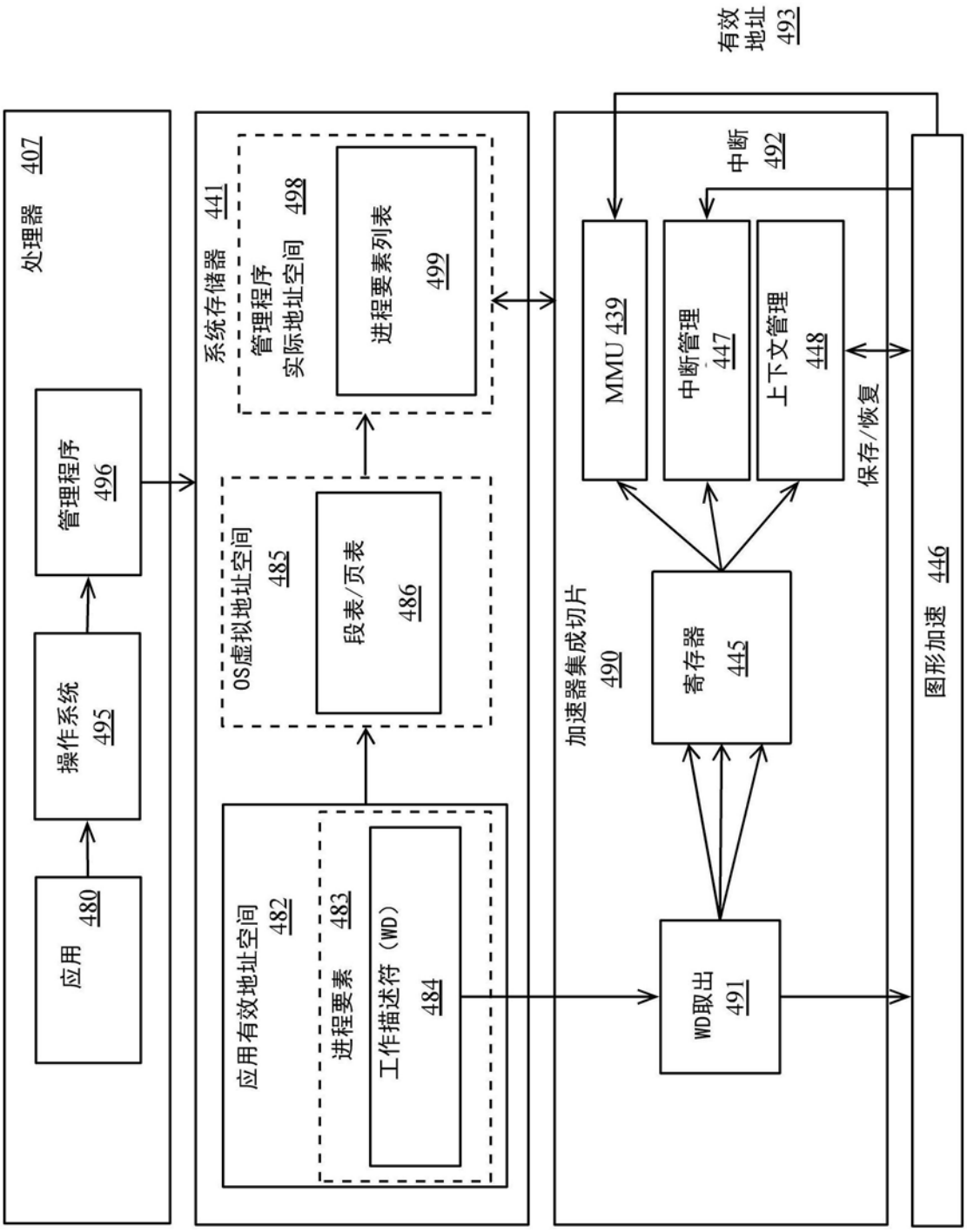

处理器 407
应用 480
操作系统 495
管理程序 496
系统存储器 441
管理程序 实际地址空间 498
进程要素列表 499
OS虚拟地址空间 485
段表/页表 486
应用有效地址空间 482
进程要素 483
工作描述符（WD） 484
有效地址 493
中断 492
MMU 439
中断管理 447
上下文管理 448
保存/恢复
加速器集成切片 490
寄存器 445
WD取出 491
图形加速 446

图4E

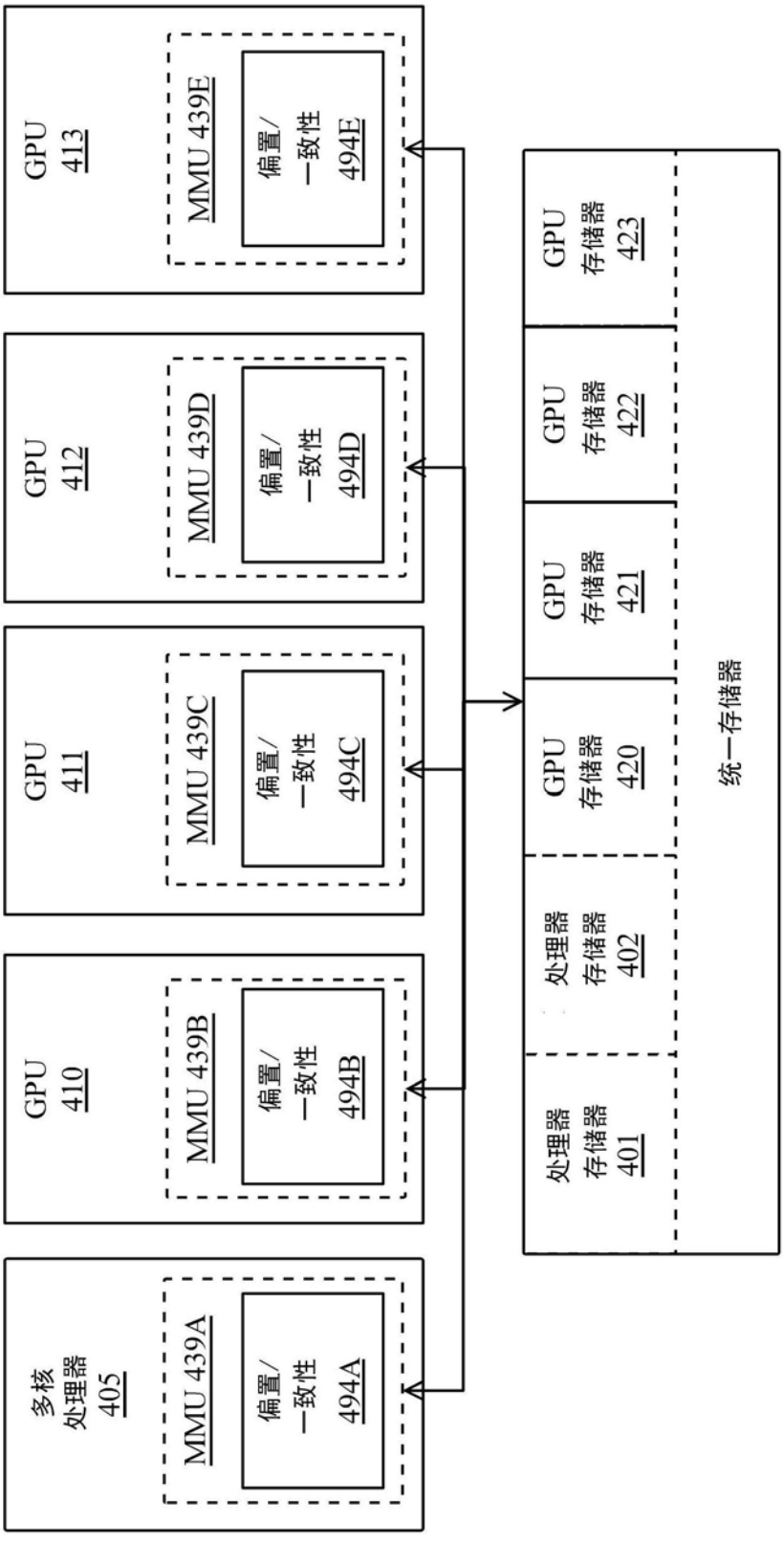

多核
处理器
405
MMU 439A
偏置/
一致性
494A
GPU
410
MMU 439B
偏置/
一致性
494B
GPU
411
MMU 439C
偏置/
一致性
494C
GPU
412
MMU 439D
偏置/
一致性
494D
GPU
413
MMU 439E
偏置/
一致性
494E
处理器
存储器
401
处理器
存储器
402
GPU
存储器
420
GPU
存储器
421
GPU
存储器
422
GPU
存储器
423
统一存储器

图4F

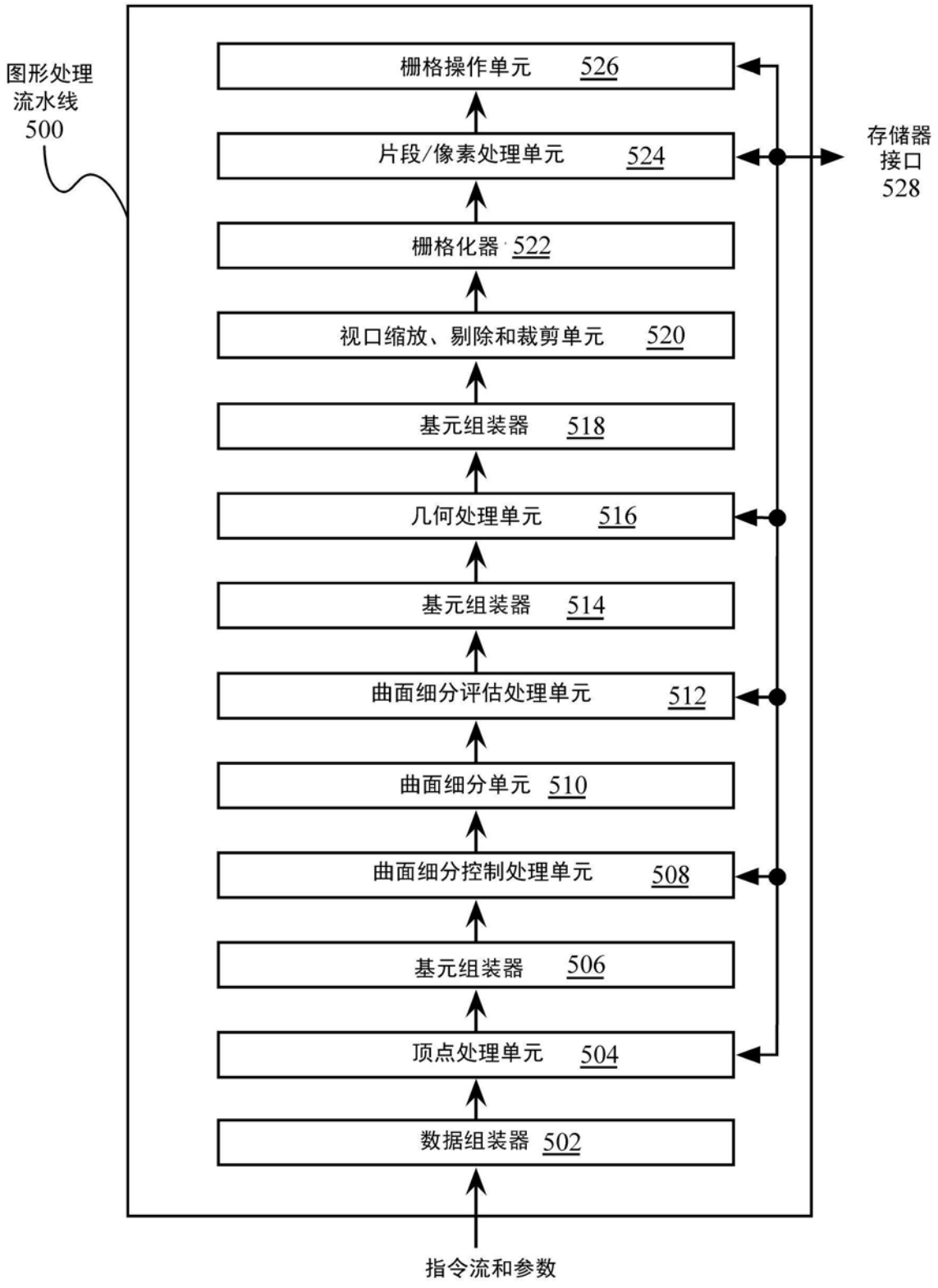

图形处理
流水线
500
栅格操作单元 526
片段/像素处理单元 524
栅格化器 522
视口缩放、剔除和裁剪单元 520
基元组装器 518
几何处理单元 516
基元组装器 514
曲面细分评估处理单元 512
曲面细分单元 510
曲面细分控制处理单元 508
基元组装器 506
顶点处理单元 504
数据组装器 502
存储器
接口
528
指令流和参数

图5

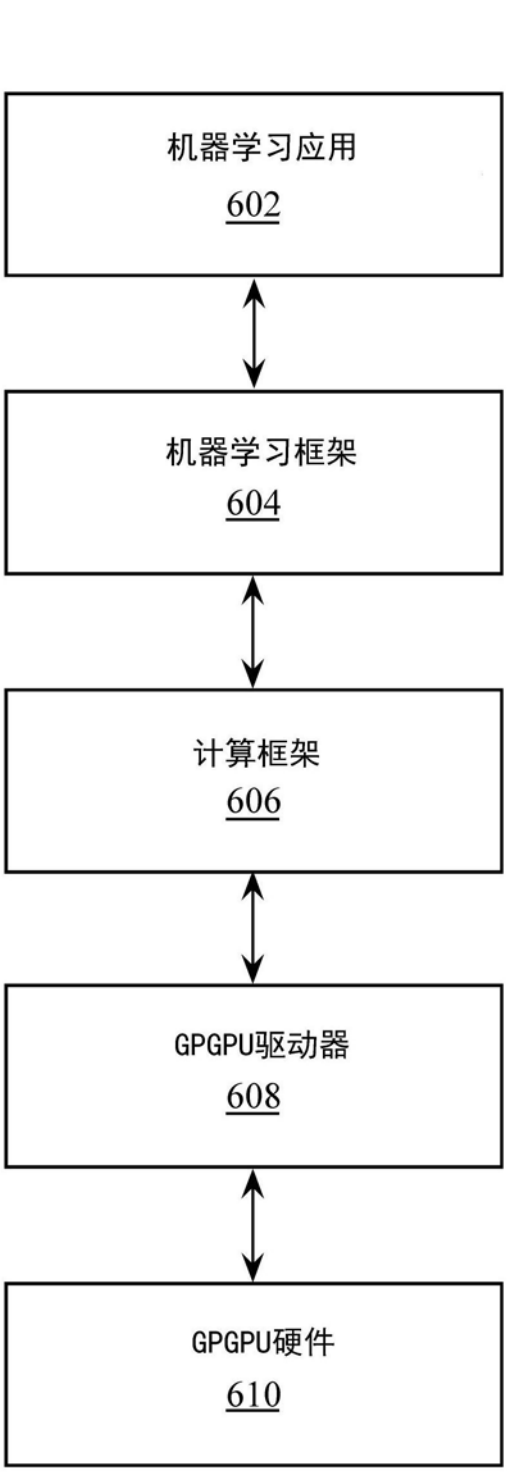


图6

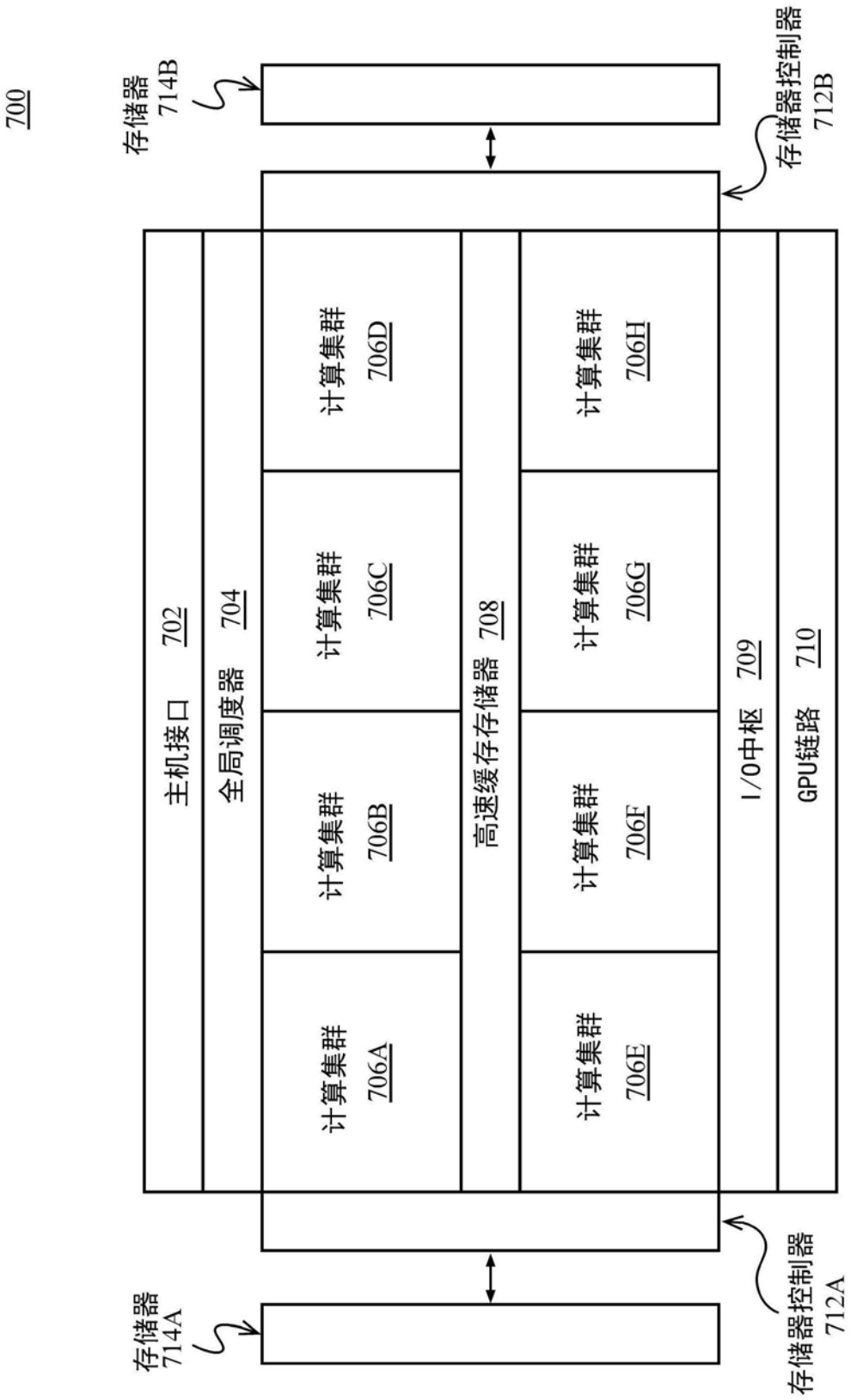

700
存储器 714B
存储器控制器 712B
主机接口 702
全局调度器 704
计算集群 706A
计算集群 706B
计算集群 706C
计算集群 706D
计算集群 706E
计算集群 706F
计算集群 706G
计算集群 706H
高速缓存存储器 708
I/O中枢 709
GPU链路 710
存储器 714A
存储器控制器 712A

图7

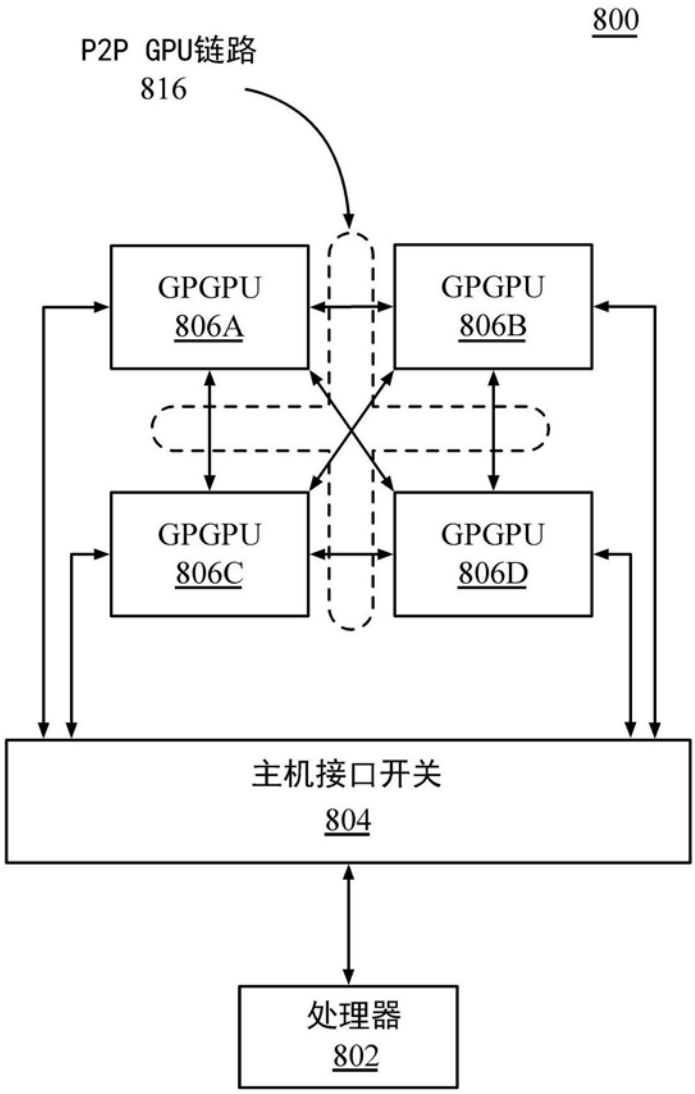

800
P2P GPU链路
816
GPGPU
806A
GPGPU
806B
GPGPU
806C
GPGPU
806D
主机接口开关
804
处理器
802

图8

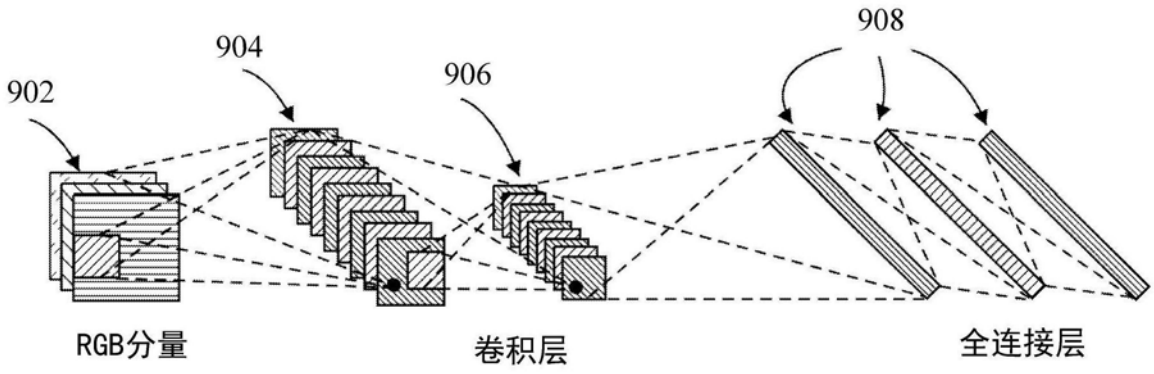

902
904
906
908
RGB分量
卷积层
全连接层

图9A

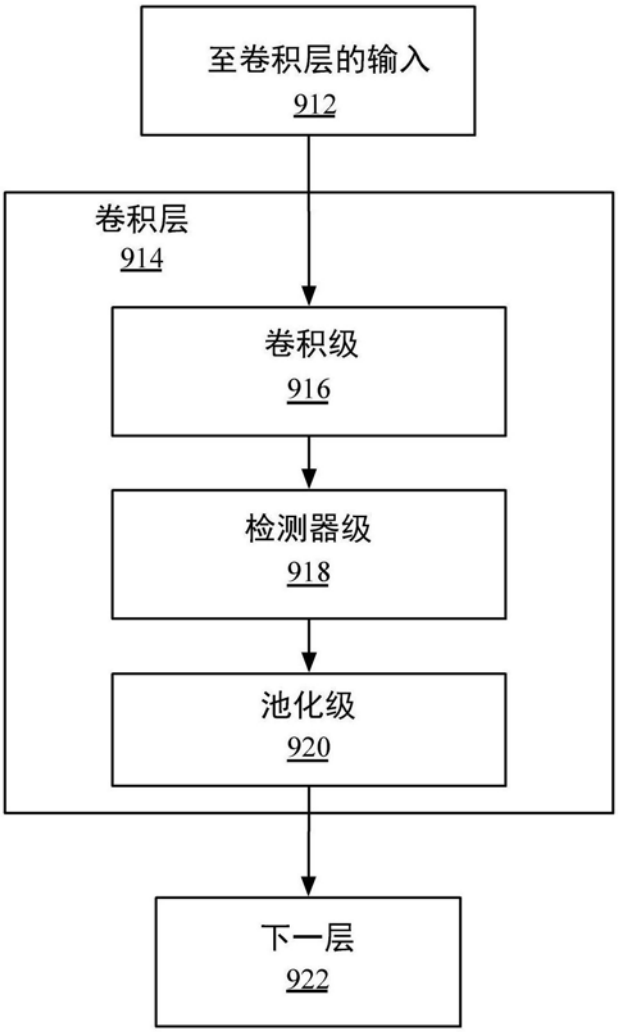


图9B

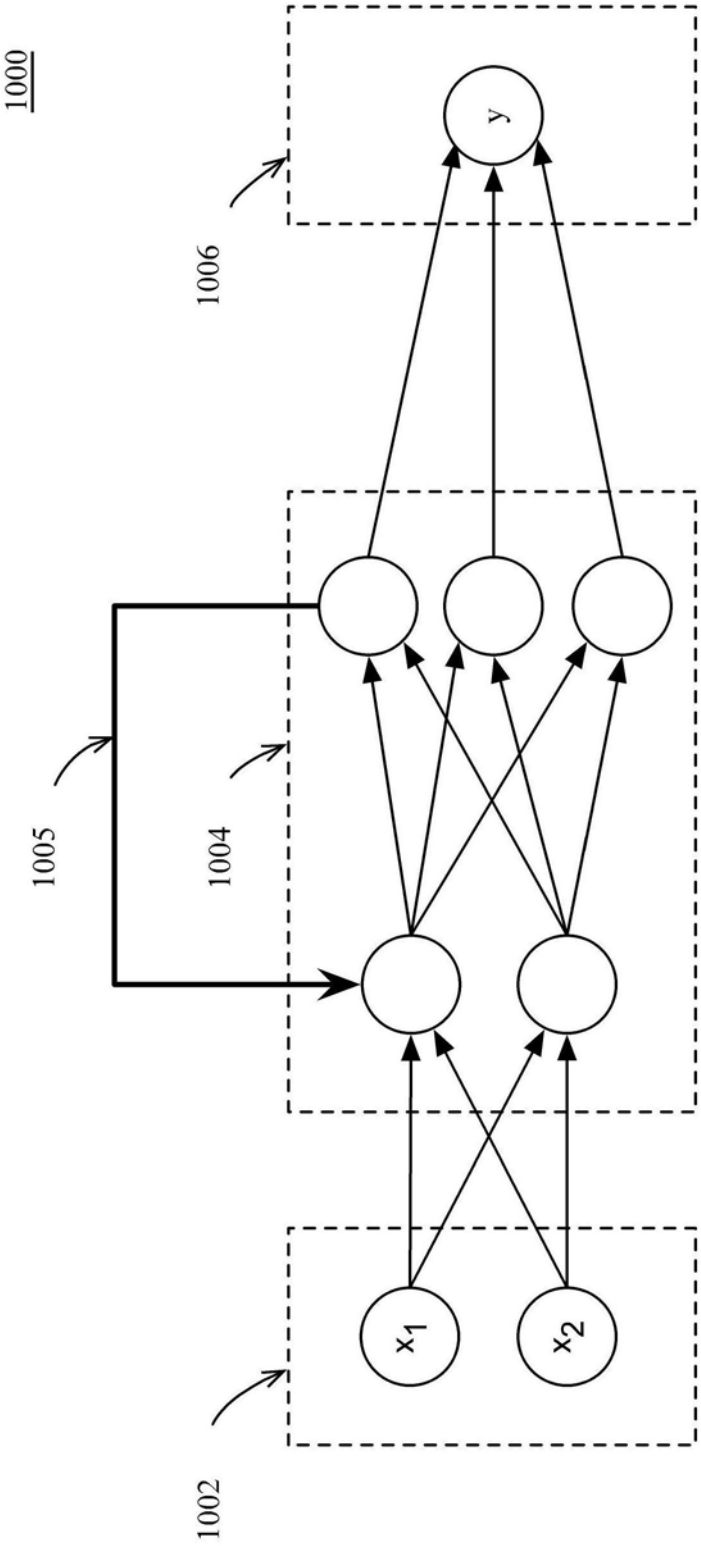


图10

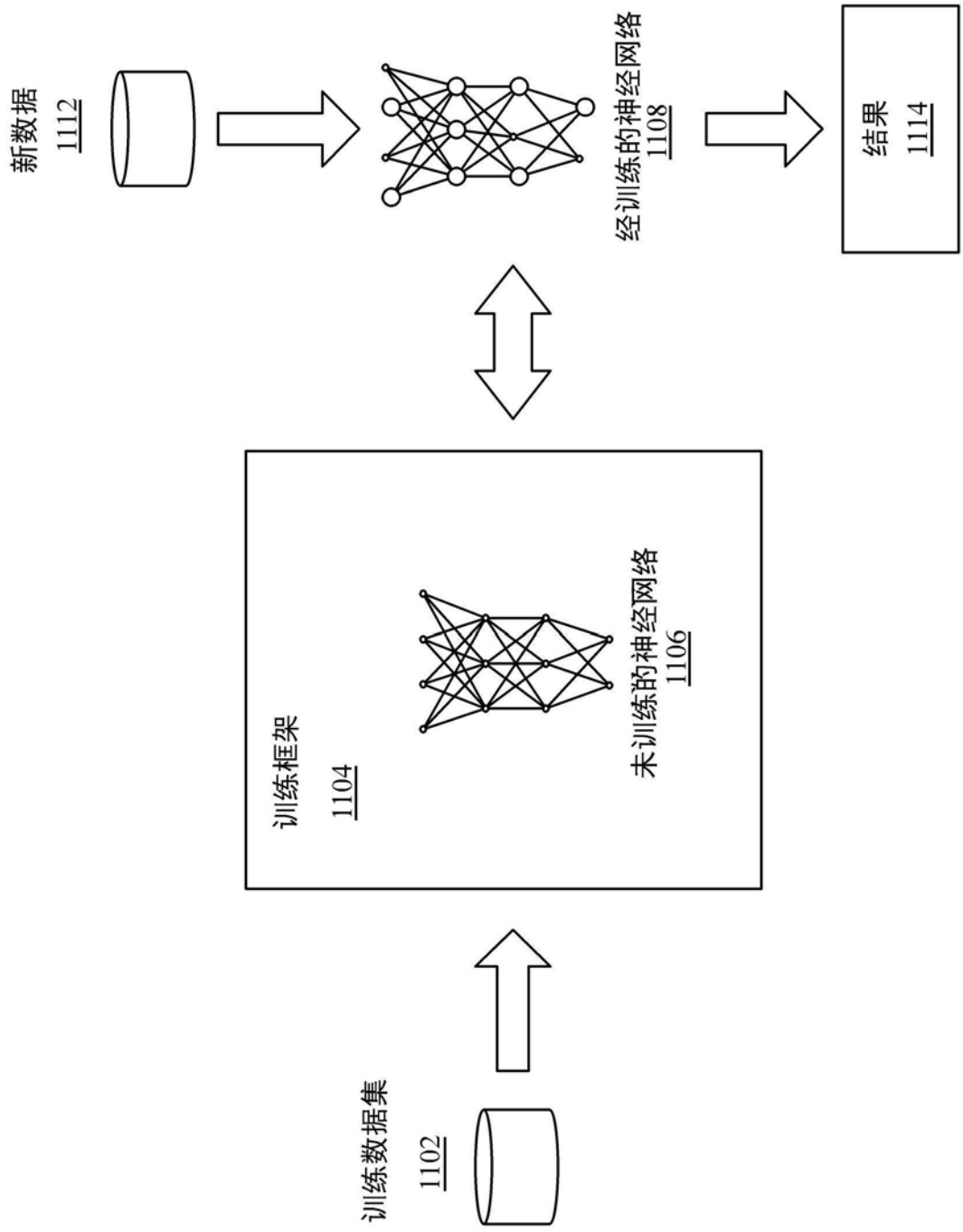

新数据
1112
经训练的神经网络
1108
结果
1114
训练框架
1104
未训练的神经网络
1106
训练数据集
1102

图11

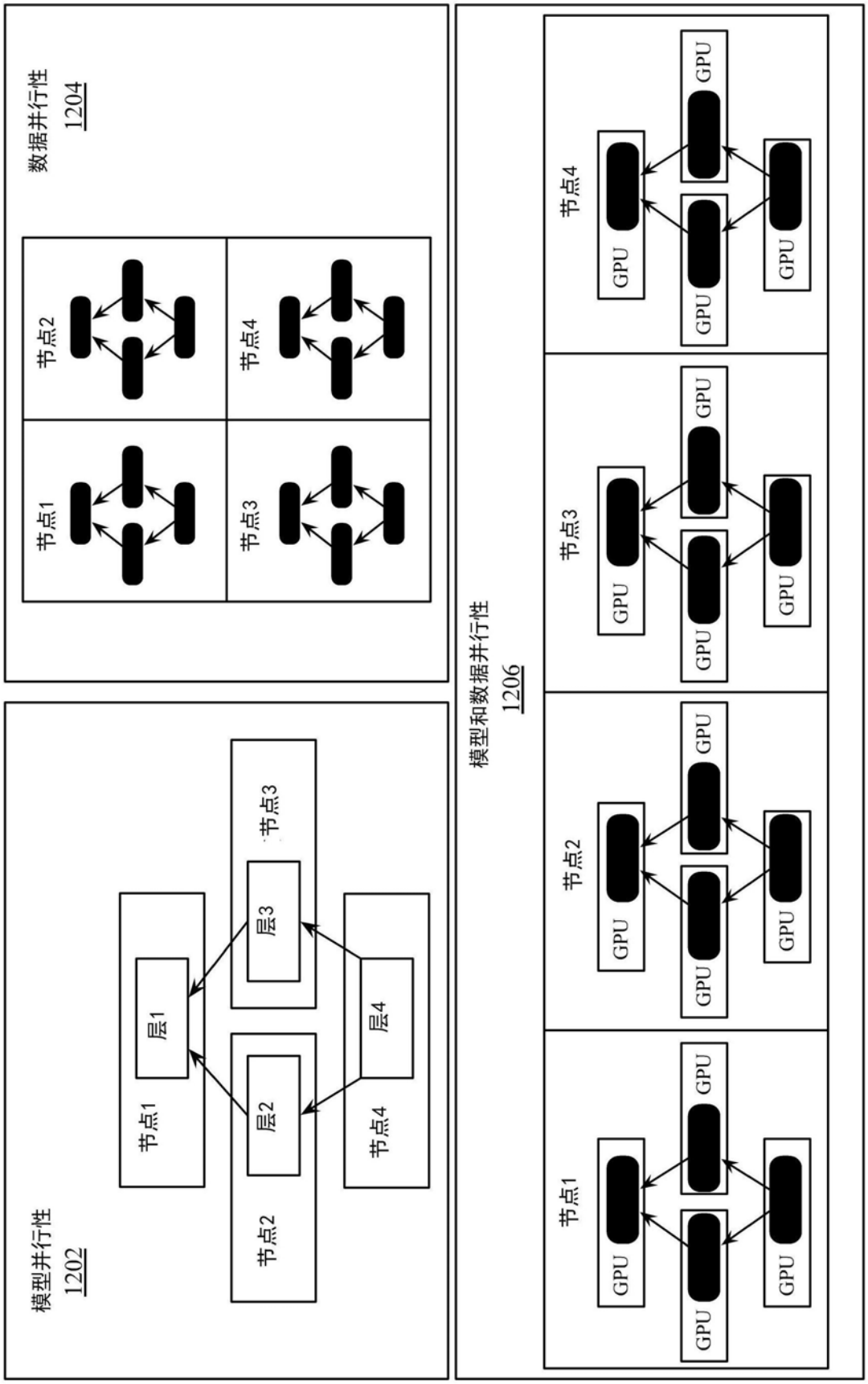

模型并行性
1202
节点1
层1
节点2
层2
节点3
层3
节点4
层4
数据并行性
1204
节点1
节点2
节点3
节点4
模型和数据并行性
1206
GPU

图12

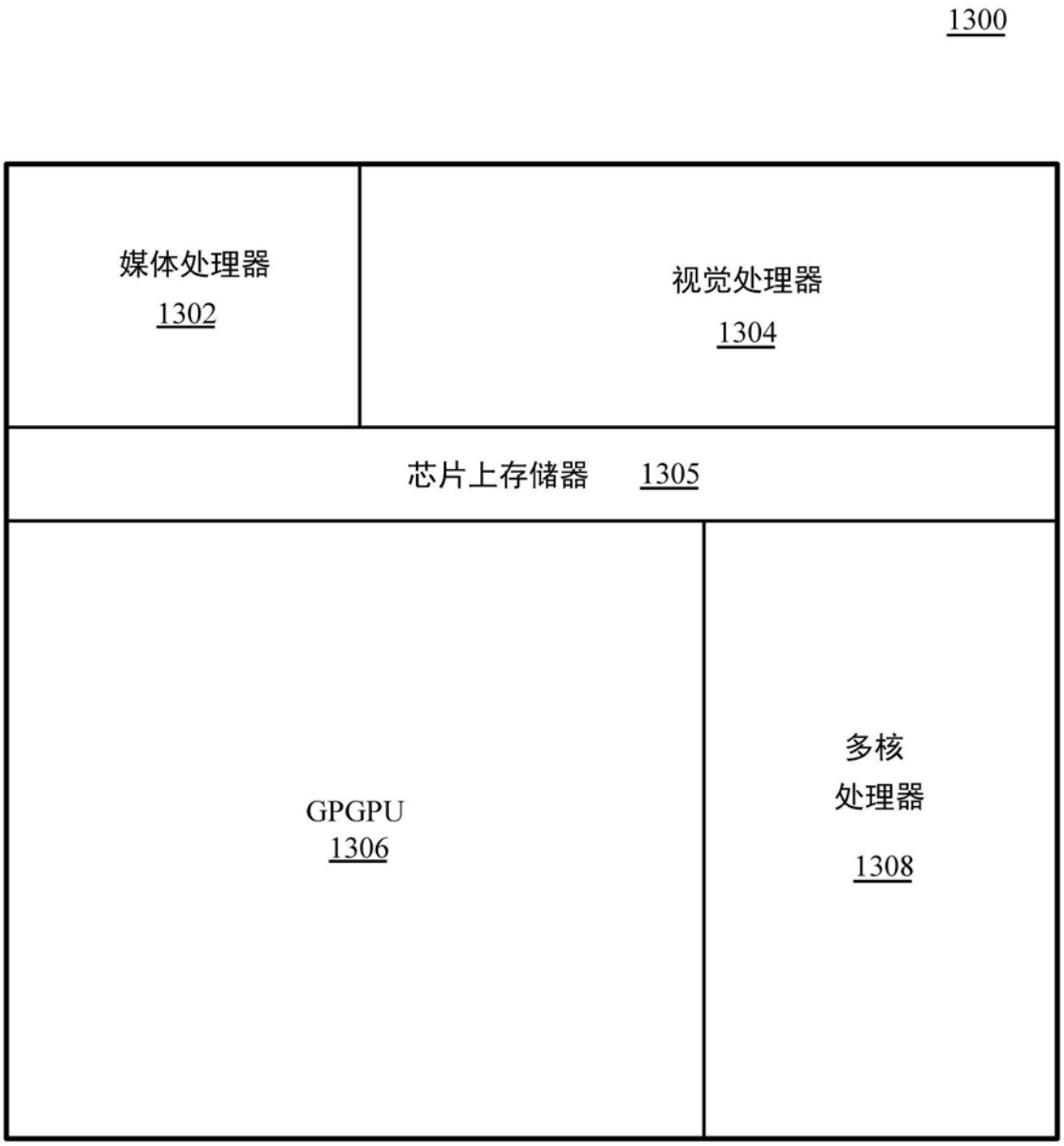

1300
媒体处理器
1302
视觉处理器
1304
芯片上存储器 1305
GPGPU
1306
多核
处理器
1308

图13

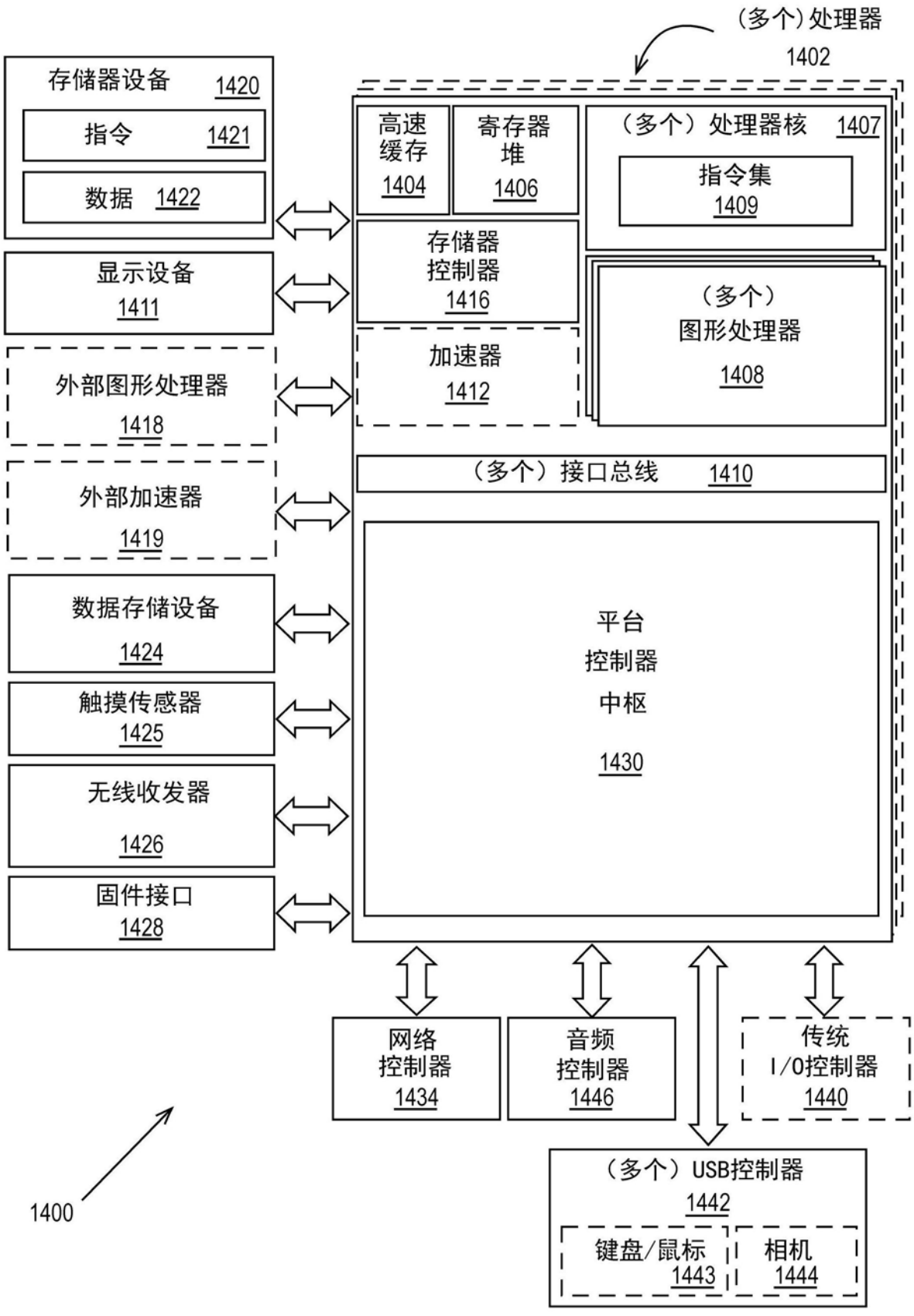

(多个)处理器
1402
存储器设备
1420
指令
1421
数据
1422
显示设备
1411
外部图形处理器
1418
外部加速器
1419
数据存储设备
1424
触摸传感器
1425
无线收发器
1426
固件接口
1428
高速缓存
1404
寄存器堆
1406
(多个)处理器核
1407
指令集
1409
存储器控制器
1416
加速器
1412
(多个)图形处理器
1408
(多个)接口总线
1410
平台控制器中枢
1430
网络控制器
1434
音频控制器
1446
传统I/O控制器
1440
(多个)USB控制器
1442
键盘/鼠标
1443
相机
1444
1400

图14

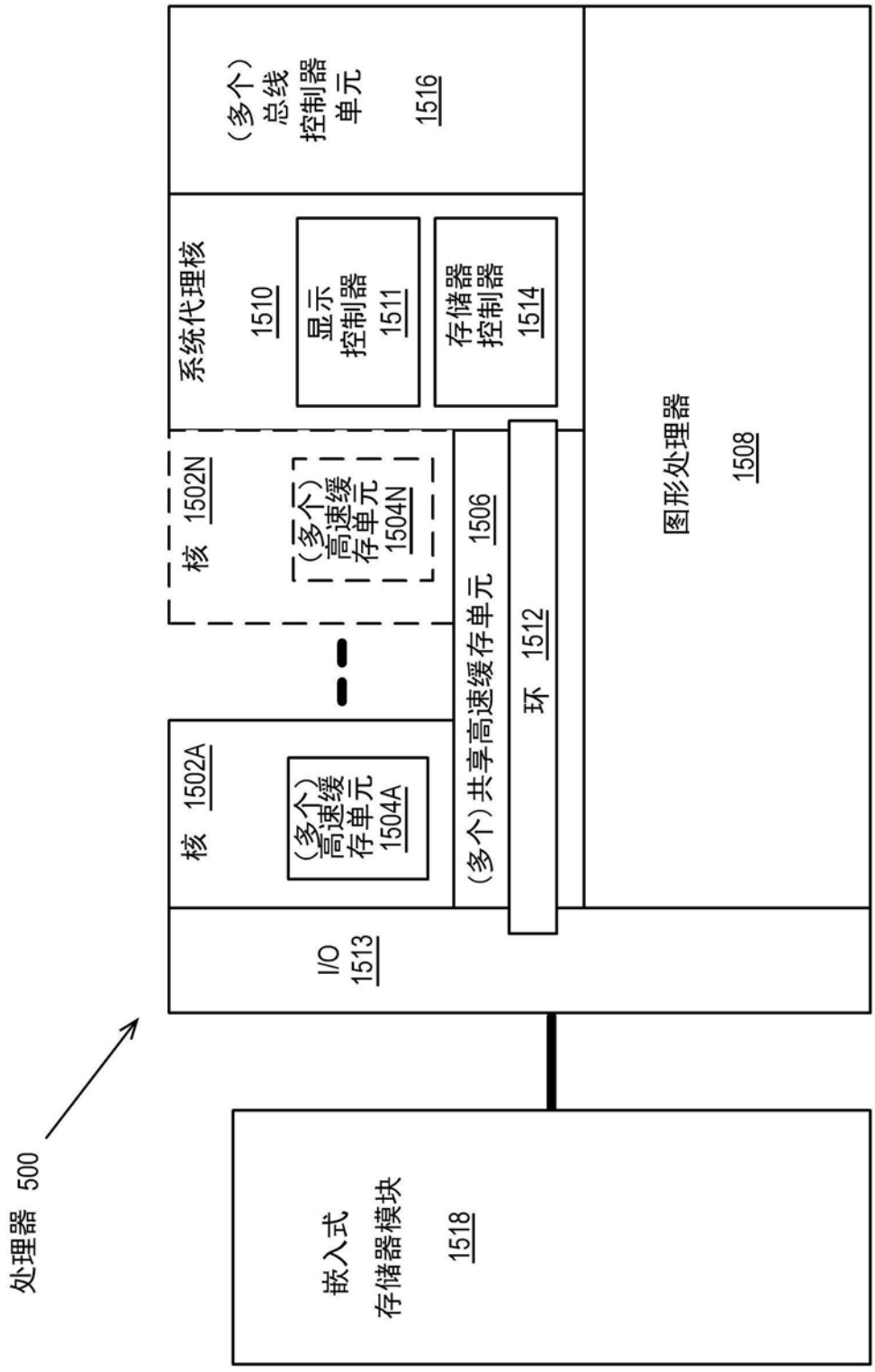

处理器 500
嵌入式
存储器模块
1518
I/O
1513
核 1502A
(多个)
高速缓
存单元
1504A
核 1502N
(多个)
高速缓
存单元
1504N
(多个)共享高速缓存单元 1506
环 1512
图形处理器
1508
系统代理核
1510
显示
控制器
1511
存储器
控制器
1514
(多个)
总线
控制器
单元
1516

图15A

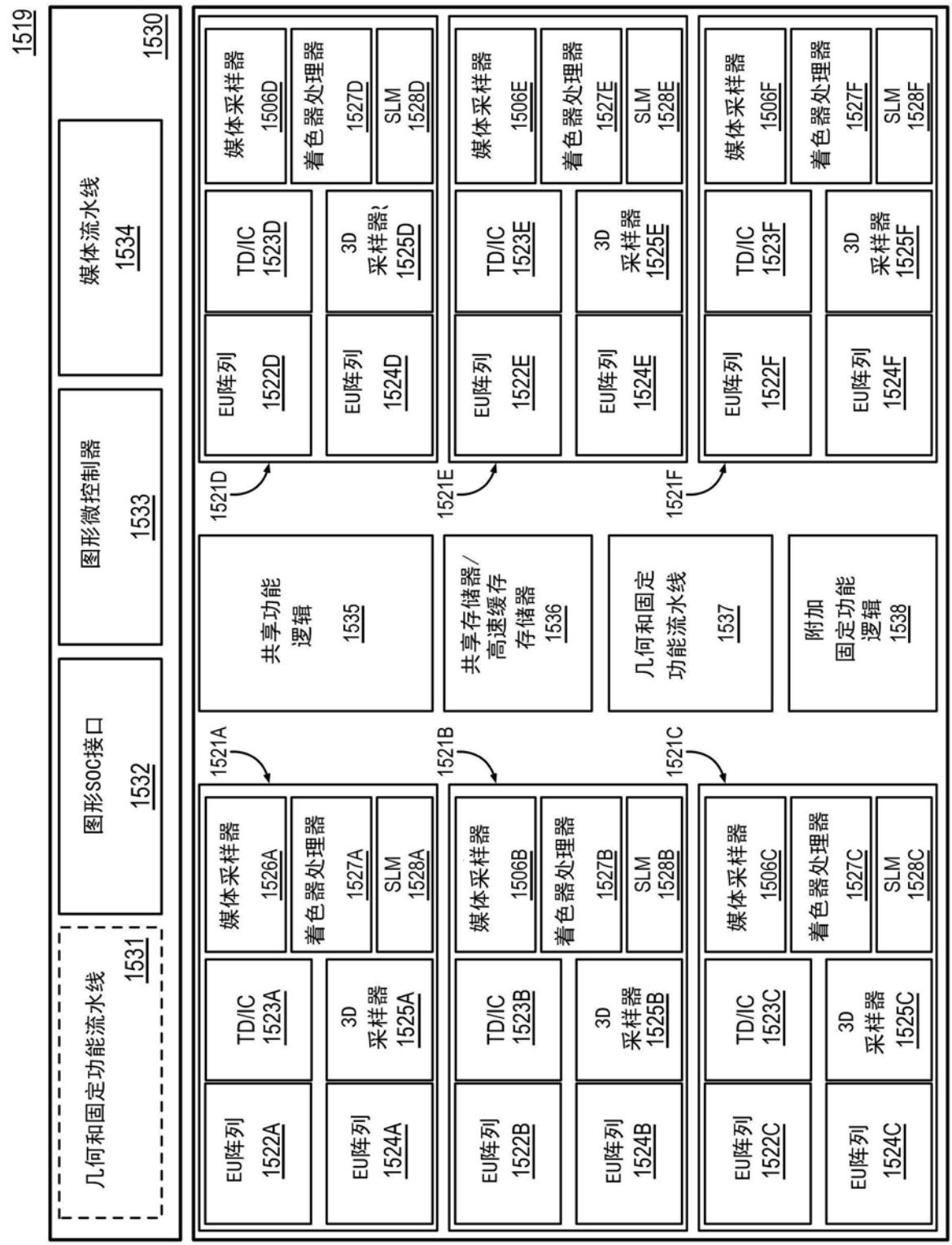

1519
几何和固定功能流水线
1531
图形SOC接口
1532
图形微控制器
1533
媒体流水线
1534
1530
共享功能逻辑
1535
共享存储器/高速缓存存储器
1536
几何和固定功能流水线
1537
附加固定功能逻辑
1538
1521A
1521B
1521C
1521D
1521E
1521F
EU阵列 1522A
TD/IC 1523A
媒体采样器 1506A
着色器处理器 1527A
SLM 1528A
EU阵列 1524A
3D 采样器 1525A
EU阵列 1522B
TD/IC 1523B
媒体采样器 1506B
着色器处理器 1527B
SLM 1528B
EU阵列 1524B
3D 采样器 1525B
EU阵列 1522C
TD/IC 1523C
媒体采样器 1506C
着色器处理器 1527C
SLM 1528C
EU阵列 1524C
3D 采样器 1525C
EU阵列 1522D
TD/IC 1523D
媒体采样器 1506D
着色器处理器 1527D
SLM 1528D
EU阵列 1524D
3D 采样器 1525D
EU阵列 1522E
TD/IC 1523E
媒体采样器 1506E
着色器处理器 1527E
SLM 1528E
EU阵列 1524E
3D 采样器 1525E
EU阵列 1522F
TD/IC 1523F
媒体采样器 1506F
着色器处理器 1527F
SLM 1528F
EU阵列 1524F
3D 采样器 1525F

图15B

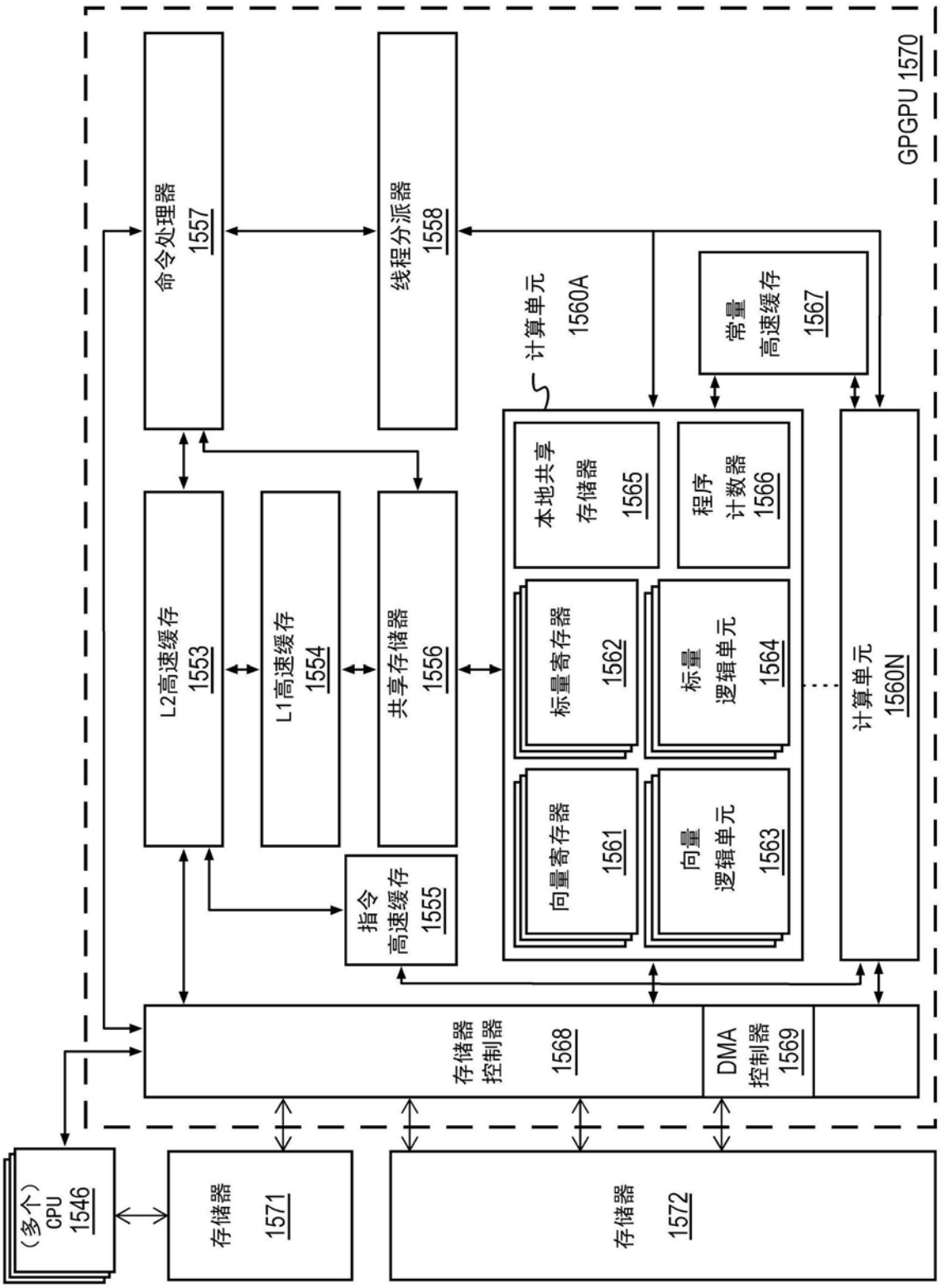

GPGPU 1570
命令处理器 1557
线程分派器 1558
计算单元 1560A
常量 高速缓存 1567
本地共享 存储器 1565
程序 计数器 1566
L2高速缓存 1553
L1高速缓存 1554
共享存储器 1556
标量寄存器 1562
标量 逻辑单元 1564
计算单元 1560N
向量寄存器 1561
向量 逻辑单元 1563
指令 高速缓存 1555
存储器 控制器 1568
DMA 控制器 1569
（多个） CPU 1546
存储器 1571
存储器 1572

图15C

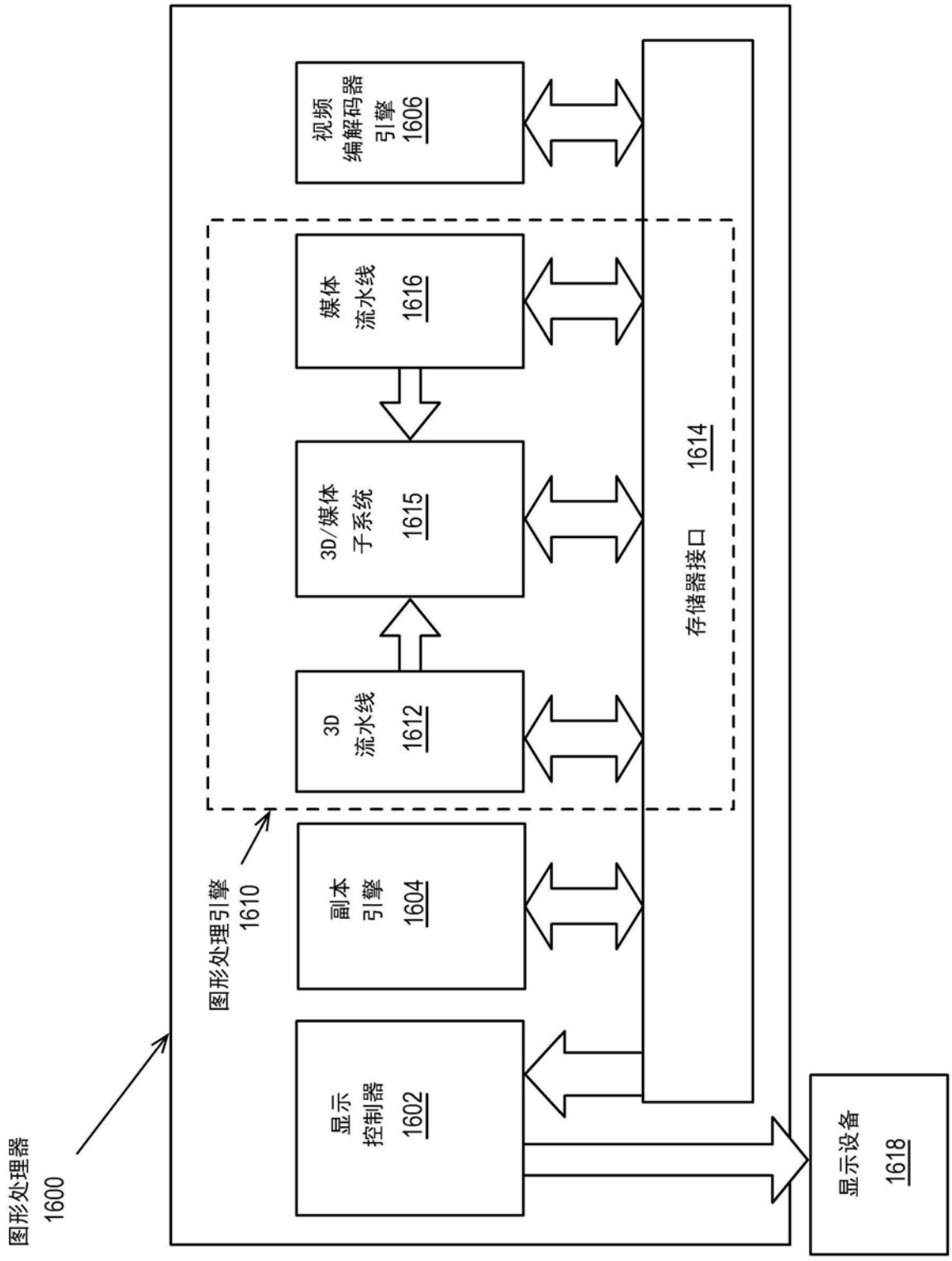

图形处理器
1600
显示
控制器
1602
副本
引擎
1604
视频
编解码器
引擎
1606
媒体
流水线
1616
3D/媒体
子系统
1615
3D
流水线
1612
图形处理引擎
1610
存储器接口
1614
显示设备
1618

图16A

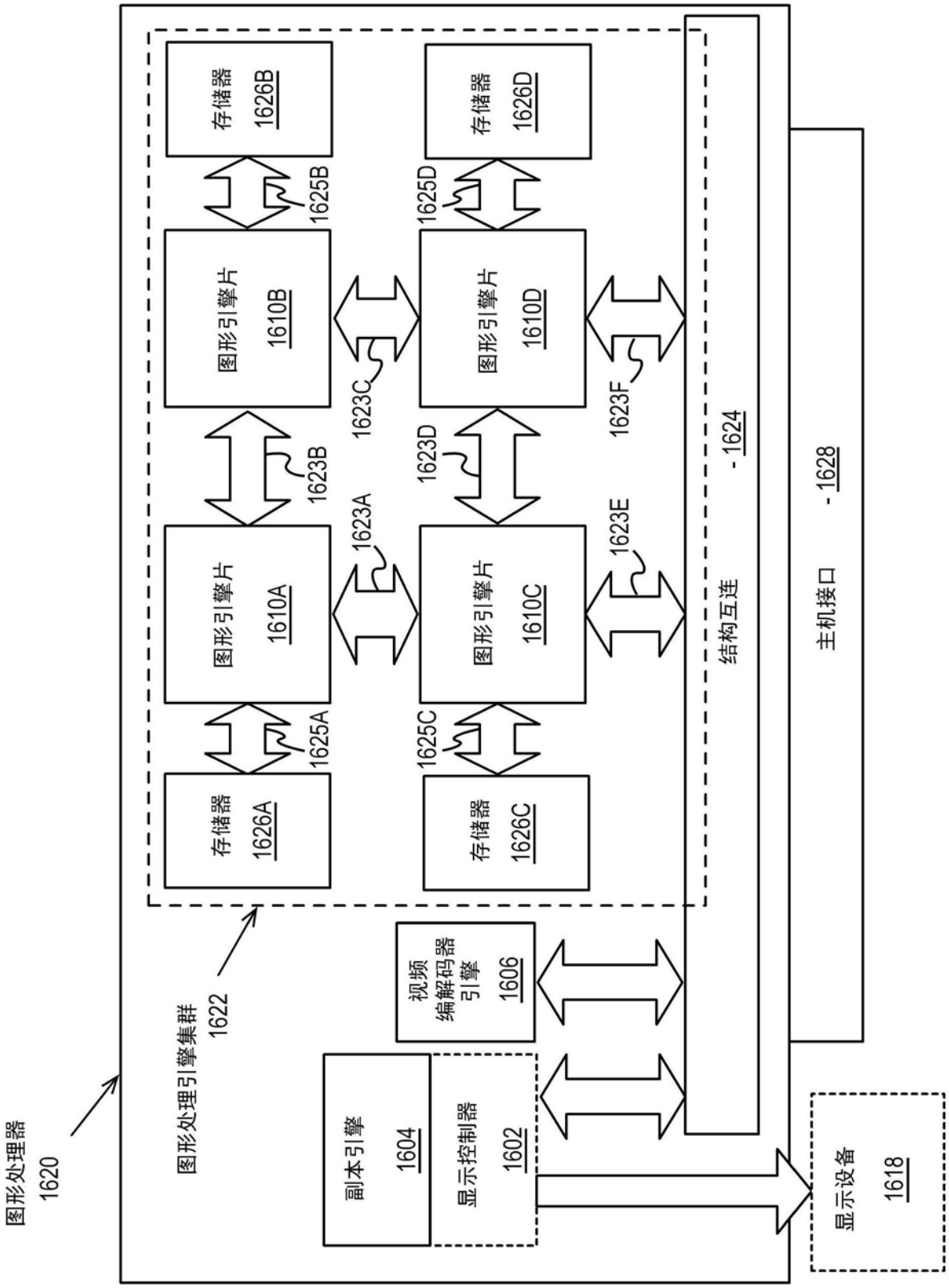

图形处理器
1620
图形处理引擎集群
1622
存储器
1626A
存储器
1626B
存储器
1626C
存储器
1626D
图形引擎片
1610A
图形引擎片
1610B
图形引擎片
1610C
图形引擎片
1610D
1625A
1625B
1625C
1625D
1623A
1623B
1623C
1623D
1623E
1623F
结构互连
1624
主机接口
1628
视频编解码器引擎
1606
副本引擎
1604
显示控制器
1602
显示设备
1618

图16B

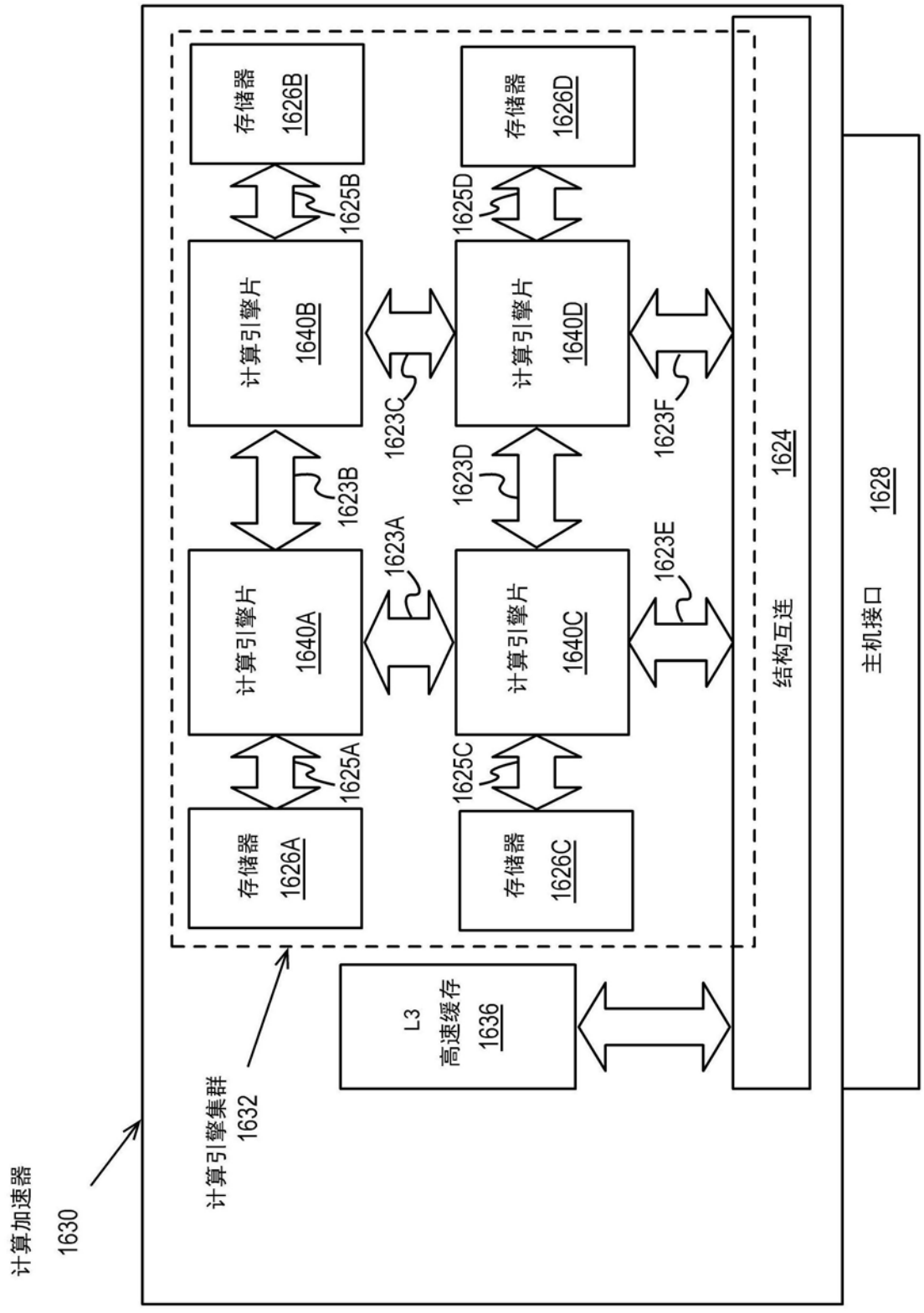

计算加速器
1630
计算引擎集群
1632
存储器
1626A
存储器
1626B
存储器
1626C
存储器
1626D
计算引擎片
1640A
计算引擎片
1640B
计算引擎片
1640C
计算引擎片
1640D
1625A
1625B
1625C
1625D
1623A
1623B
1623C
1623D
1623E
1623F
L3
高速缓存
1636
结构互连
1624
主机接口
1628

图16C

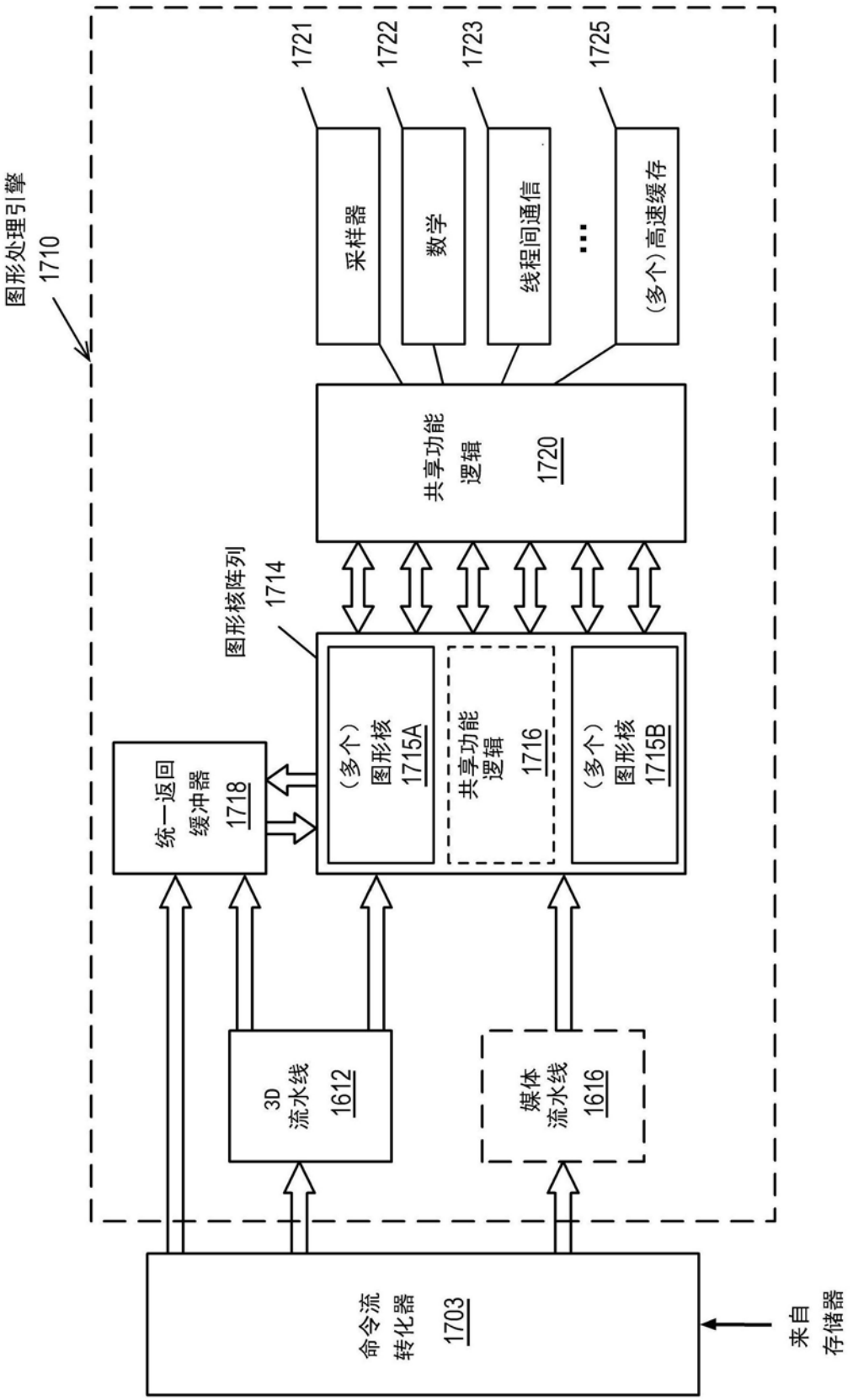

图形处理引擎
1710
1721
1722
1723
1725
采样器
数学
线程间通信
...
(多个)高速缓存
共享功能
逻辑
1720
图形核阵列
1714
统一返回
缓冲器
1718
(多个)
图形核
1715A
共享功能
逻辑
1716
(多个)
图形核
1715B
3D
流水线
1612
媒体
流水线
1616
命令流
转化器
1703
来自
存储器

图17

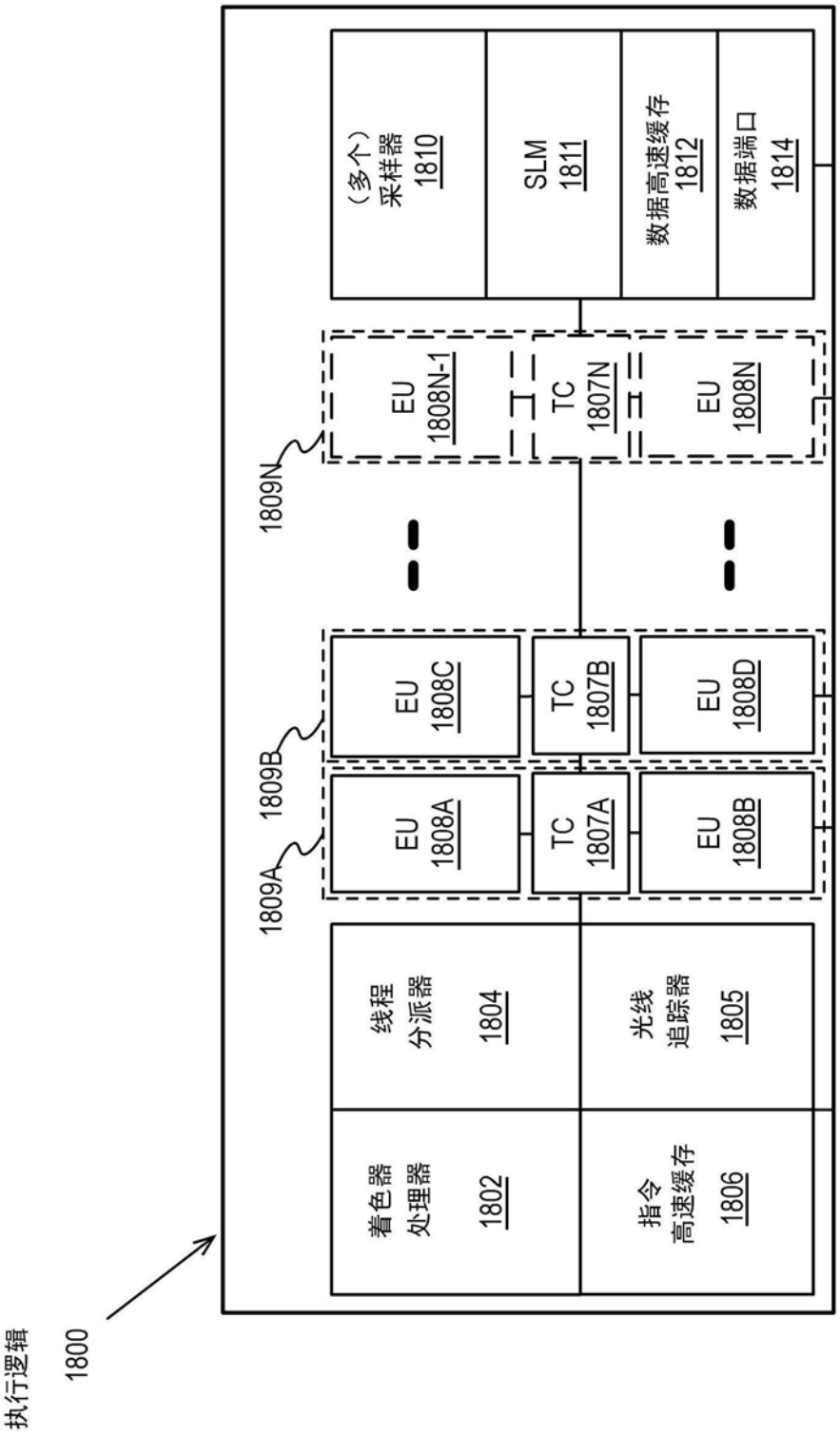

执行逻辑
1800
着色器处理器 1802
线程分派器 1804
指令高速缓存 1806
光线追踪器 1805
1809A
EU 1808A
TC 1807A
EU 1808B
1809B
EU 1808C
TC 1807B
EU 1808D
1809N
EU 1808N-1
TC 1807N
EU 1808N
（多个）采样器 1810
SLM 1811
数据高速缓存 1812
数据端口 1814

图18A

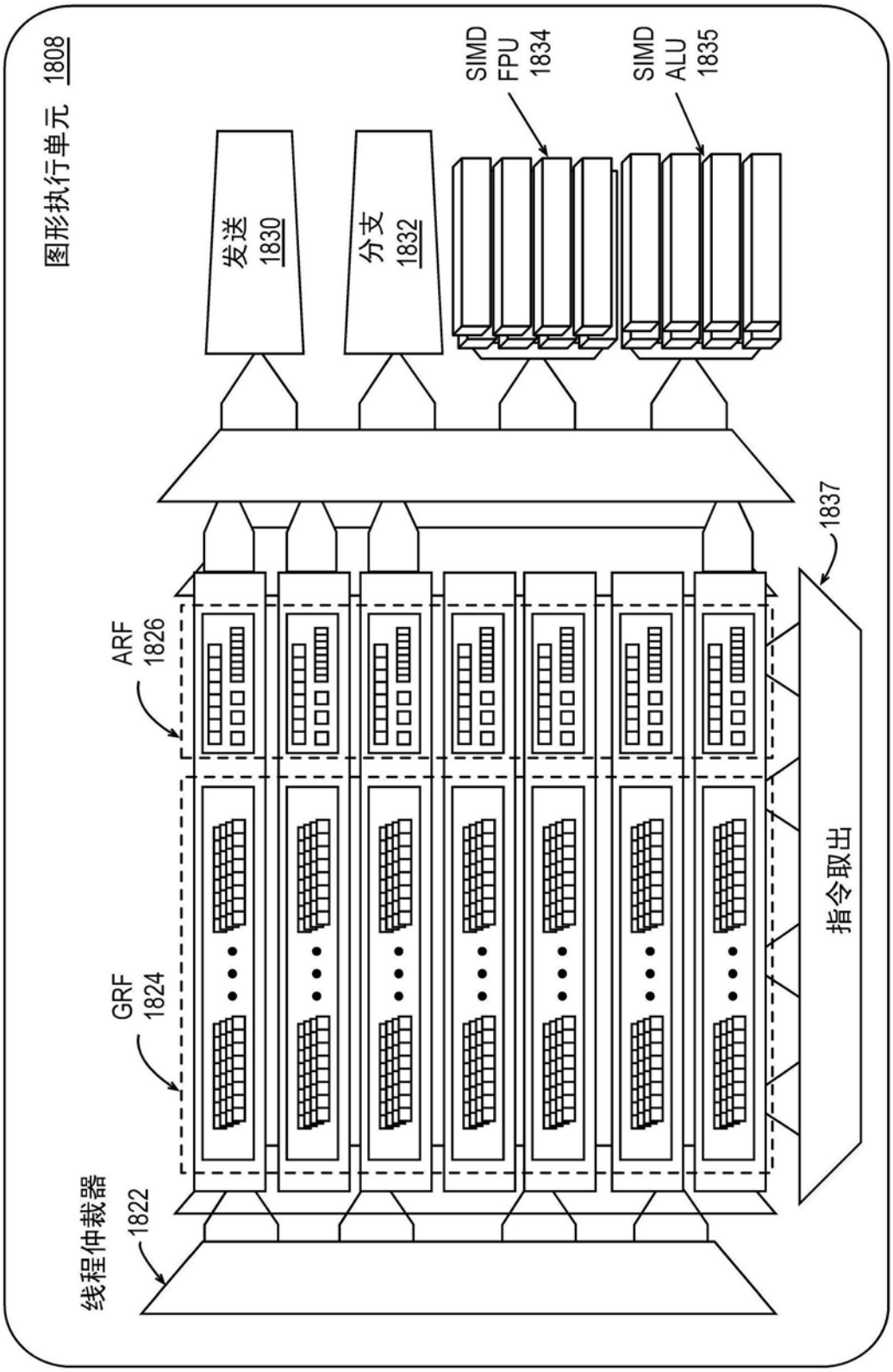

图形执行单元 1808
发送 1830
分支 1832
SIMD FPU 1834
SIMD ALU 1835
1837
ARF 1826
GRF 1824
线程仲裁器 1822
指令取出

图18B

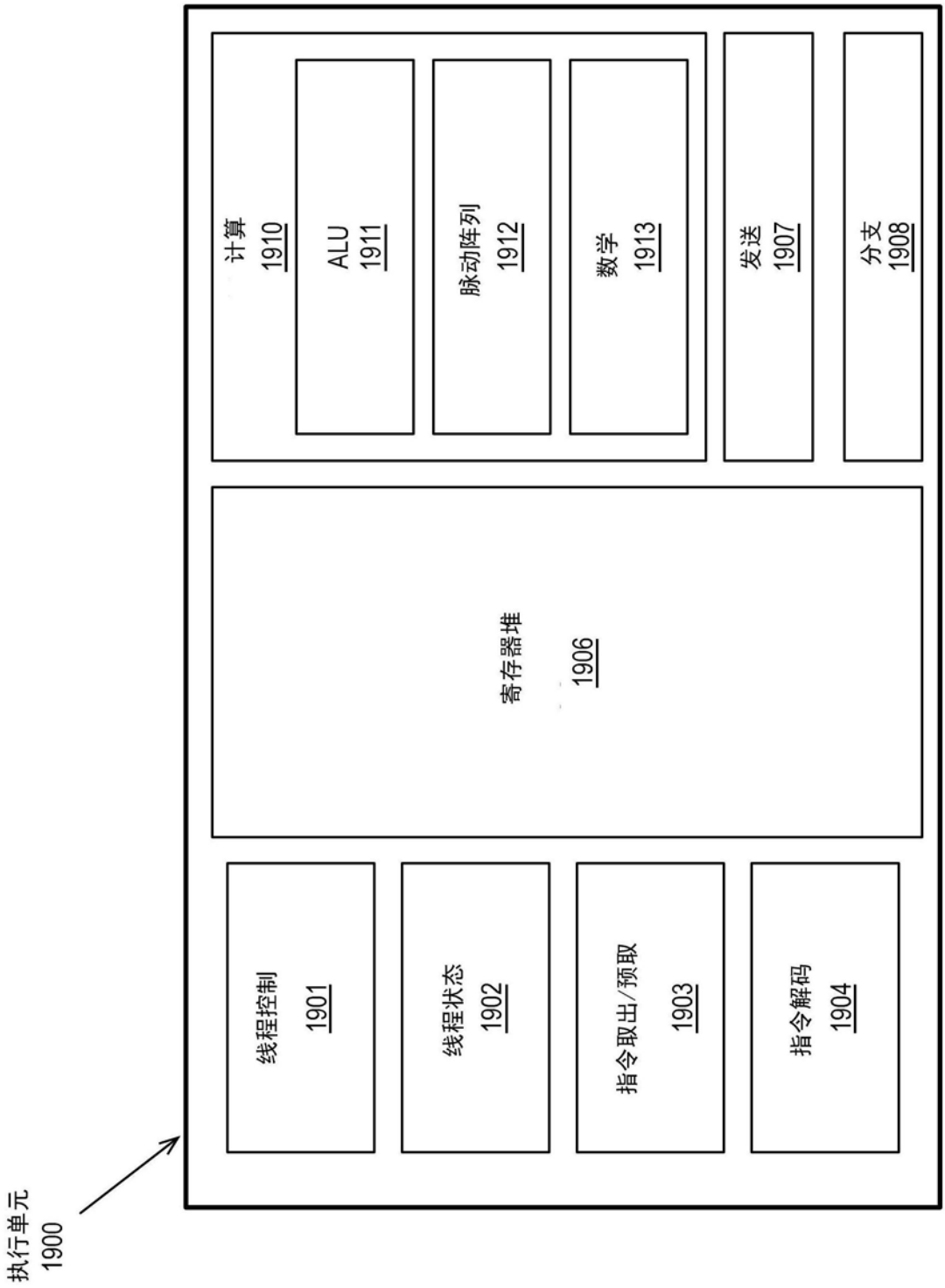

执行单元
1900
线程控制
1901
线程状态
1902
指令取出/预取
1903
指令解码
1904
寄存器堆
1906
计算
1910
ALU
1911
脉动阵列
1912
数学
1913
发送
1907
分支
1908

图19

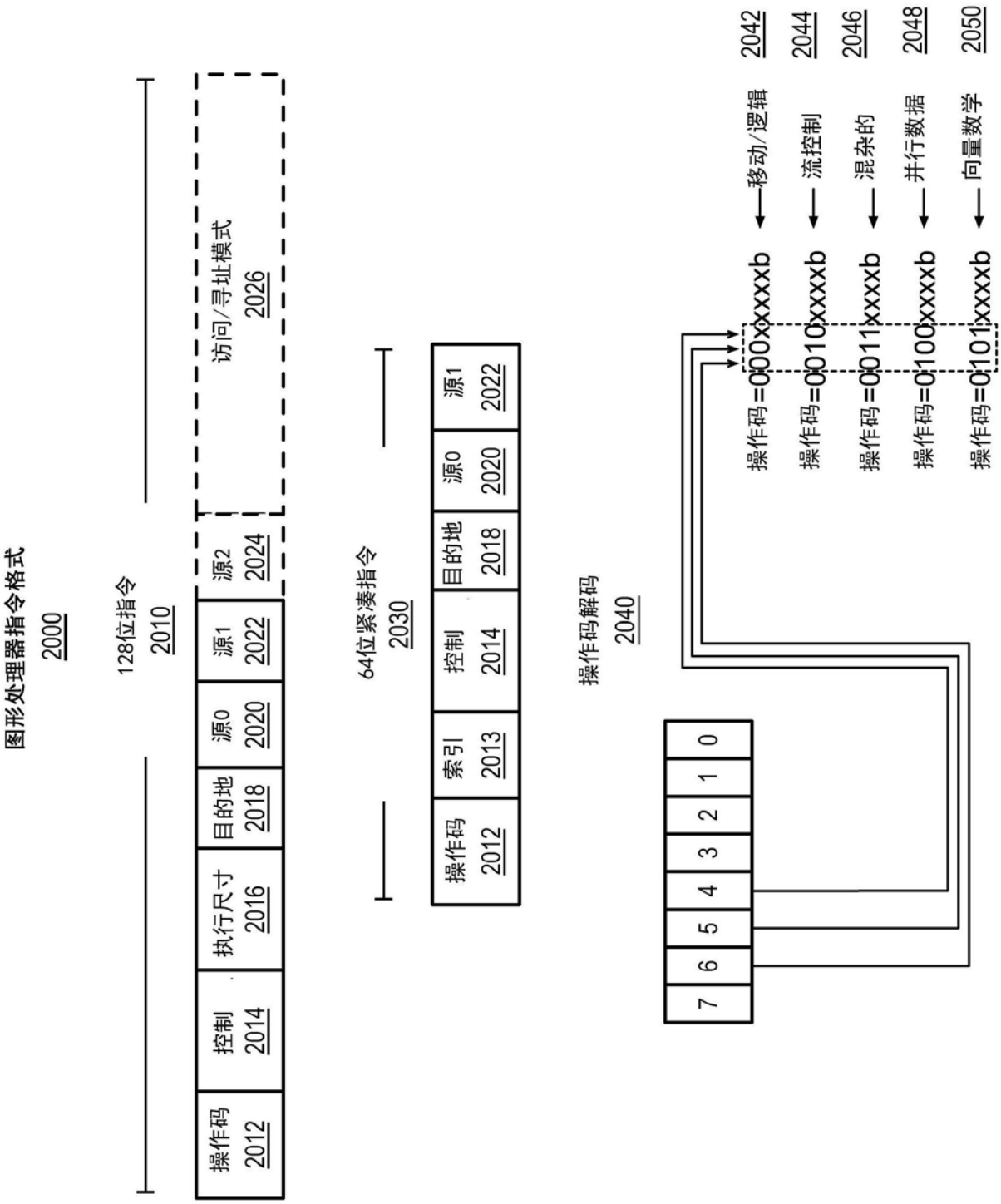

图形处理器指令格式
2000
128位指令
2010
操作码 2012
控制 2014
执行尺寸 2016
目的地 2018
源0 2020
源1 2022
源2 2024
访问/寻址模式 2026
64位紧凑指令
2030
操作码 2012
索引 2013
控制 2014
目的地 2018
源0 2020
源1 2022
操作码解码
2040
7 6 5 4 3 2 1 0
操作码=0000xxxxb
操作码=0010xxxxb
操作码=0011xxxxb
操作码=0100xxxxb
操作码=0101xxxxb
移动/逻辑 2042
流控制 2044
混杂的 2046
并行数据 2048
向量数学 2050

图20

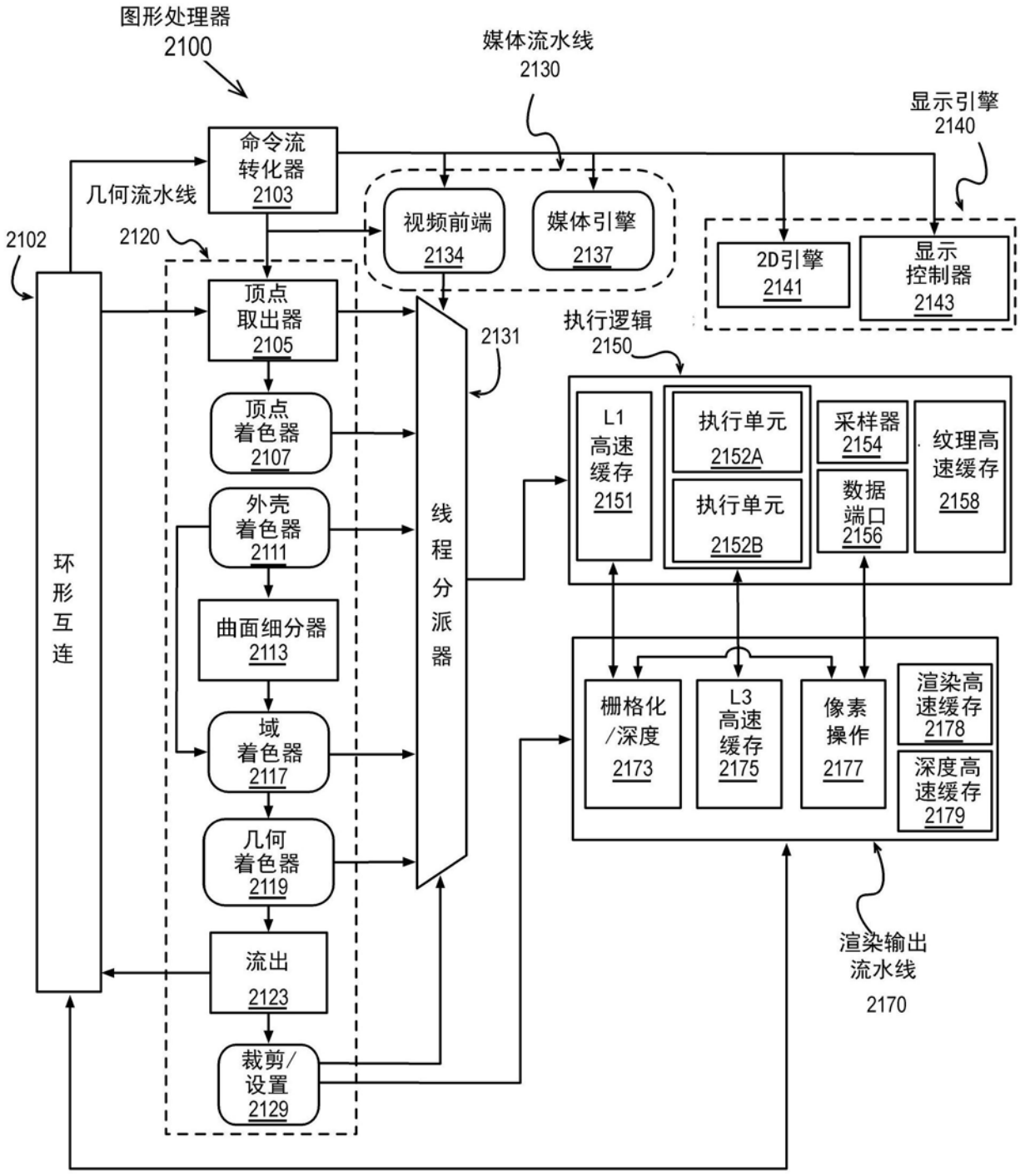

图形处理器
2100
媒体流水线
2130
显示引擎
2140
命令流
转化器
2103
几何流水线
2102
2120
视频前端
2134
媒体引擎
2137
2D引擎
2141
显示
控制器
2143
顶点
取出器
2105
2131
执行逻辑
2150
顶点
着色器
2107
外壳
着色器
2111
曲面细分器
2113
域
着色器
2117
几何
着色器
2119
流出
2123
裁剪/
设置
2129
环
形
互
连
线
程
分
派
器
L1
高速
缓存
2151
执行单元
2152A
执行单元
2152B
采样器
2154
数据
端口
2156
纹理高
速缓存
2158
栅格化
/深度
2173
L3
高速
缓存
2175
像素
操作
2177
渲染高
速缓存
2178
深度高
速缓存
2179
渲染输出
流水线
2170

图21

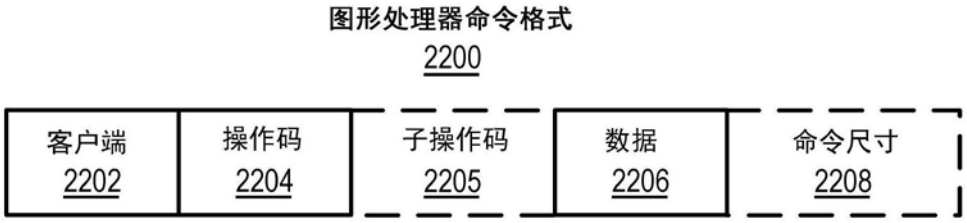

图形处理器命令格式
2200
客户端
2202
操作码
2204
子操作码
2205
数据
2206
命令尺寸
2208

图22A

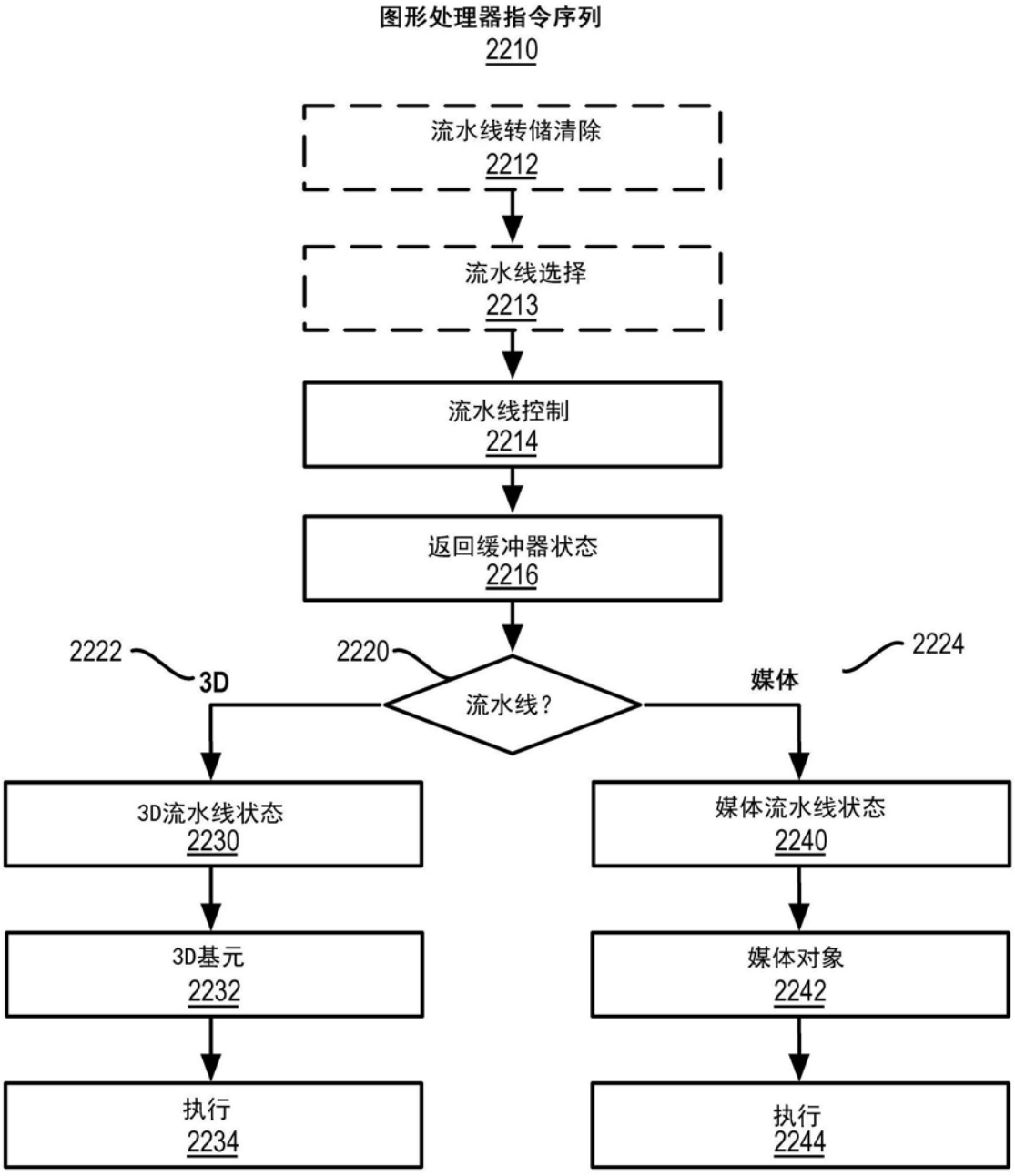

图形处理器指令序列
2210
流水线转储清除
2212
流水线选择
2213
流水线控制
2214
返回缓冲器状态
2216
2222
3D
2220
流水线?
媒体
2224
3D流水线状态
2230
3D基元
2232
执行
2234
媒体流水线状态
2240
媒体对象
2242
执行
2244

图22B

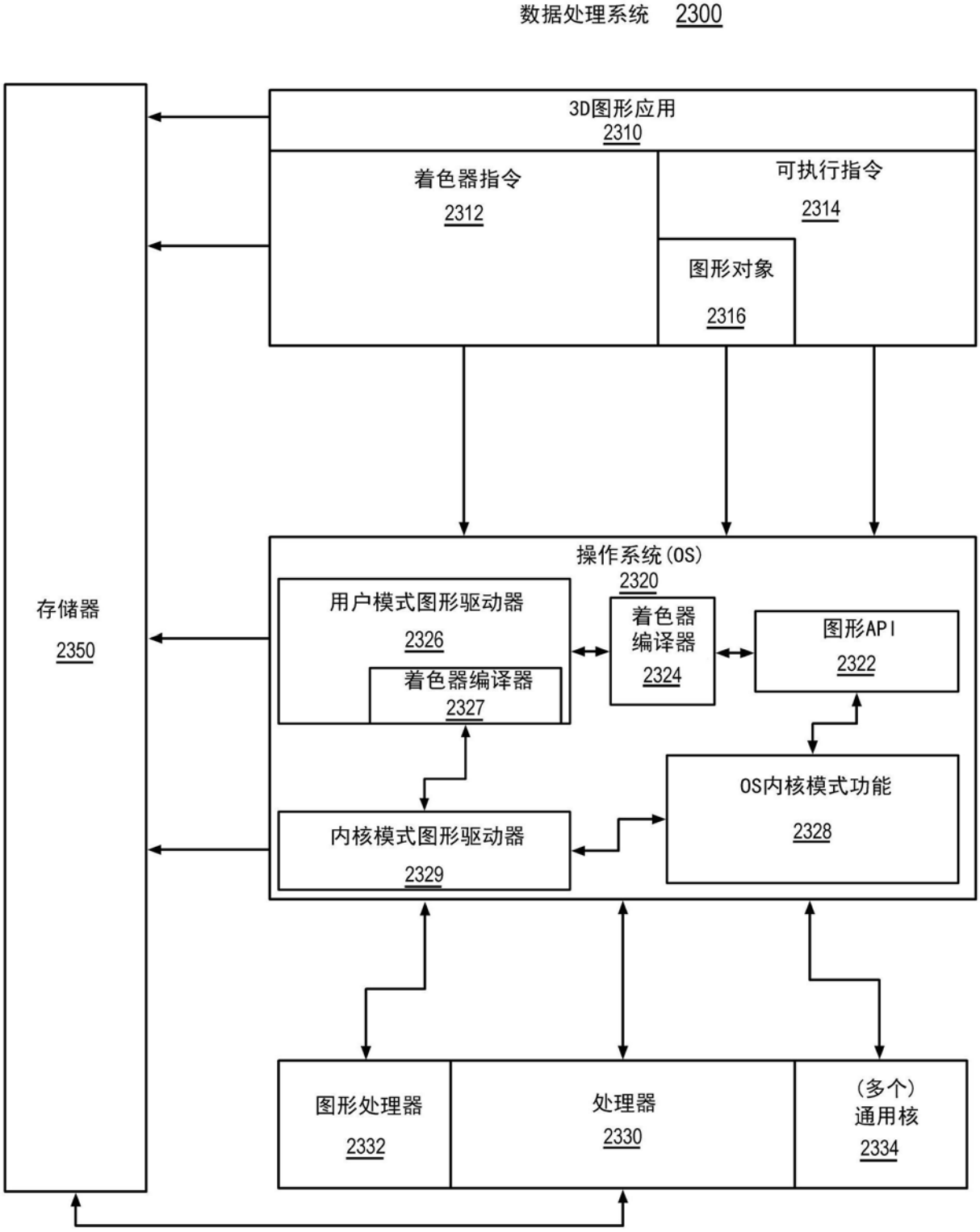

数据处理系统 2300
3D图形应用
2310
着色器指令
2312
可执行指令
2314
图形对象
2316
存储器
2350
操作系统(OS)
2320
用户模式图形驱动器
2326
着色器编译器
2327
着色器
编译器
2324
图形API
2322
OS内核模式功能
2328
内核模式图形驱动器
2329
图形处理器
2332
处理器
2330
(多个)
通用核
2334

图23

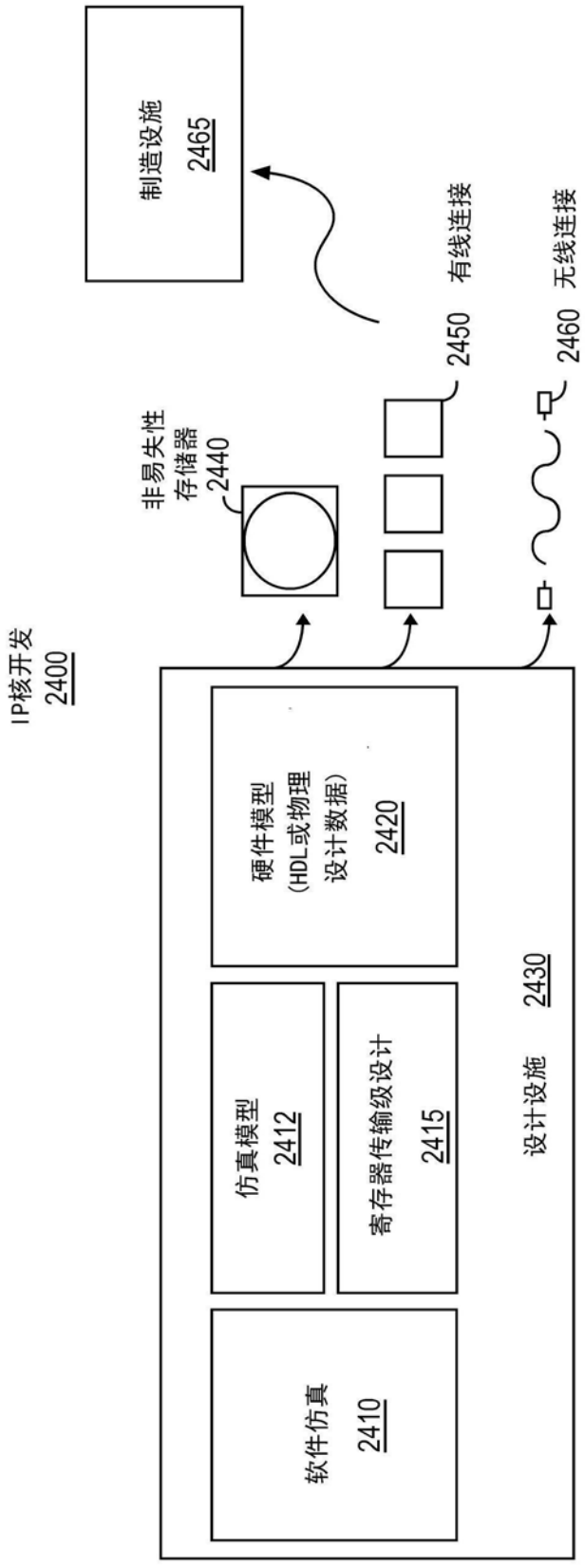

制造设施
2465
有线连接
2450
无线连接
2460
非易失性
存储器
2440
IP核开发
2400
硬件模型
(HDL或物理
设计数据)
2420
仿真模型
2412
寄存器传输级设计
2415
软件仿真
2410
设计设施
2430

图24A

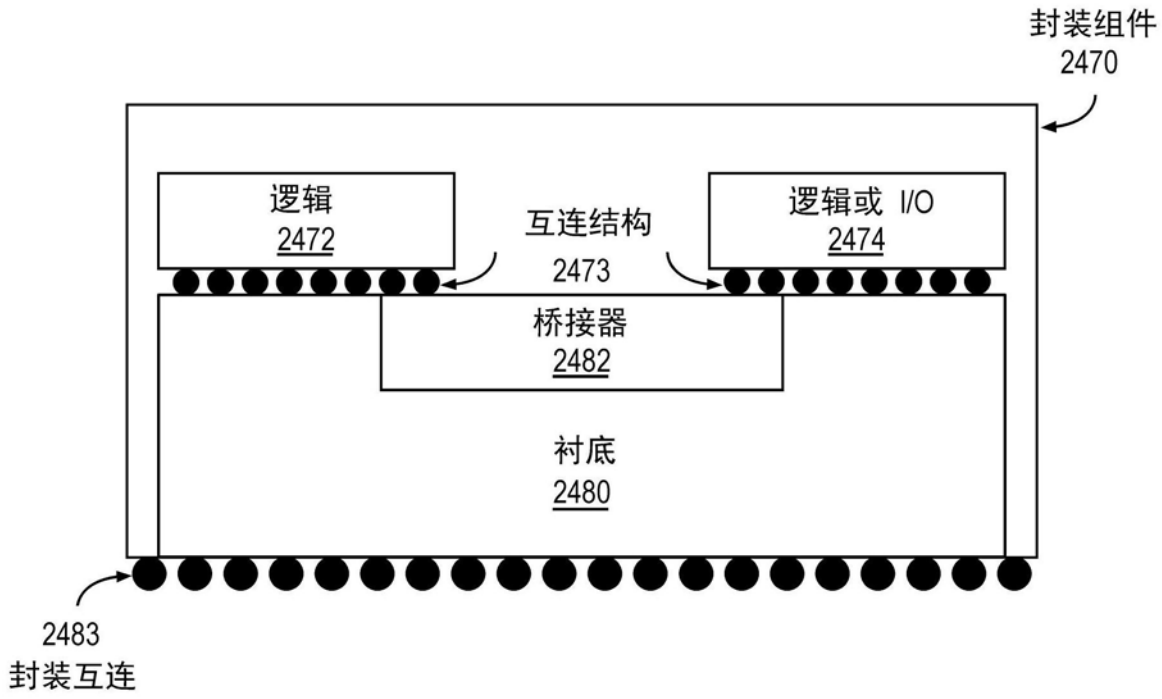

封装组件
2470
逻辑
2472
互连结构
2473
逻辑或 I/O
2474
桥接器
2482
衬底
2480
2483
封装互连

图24B

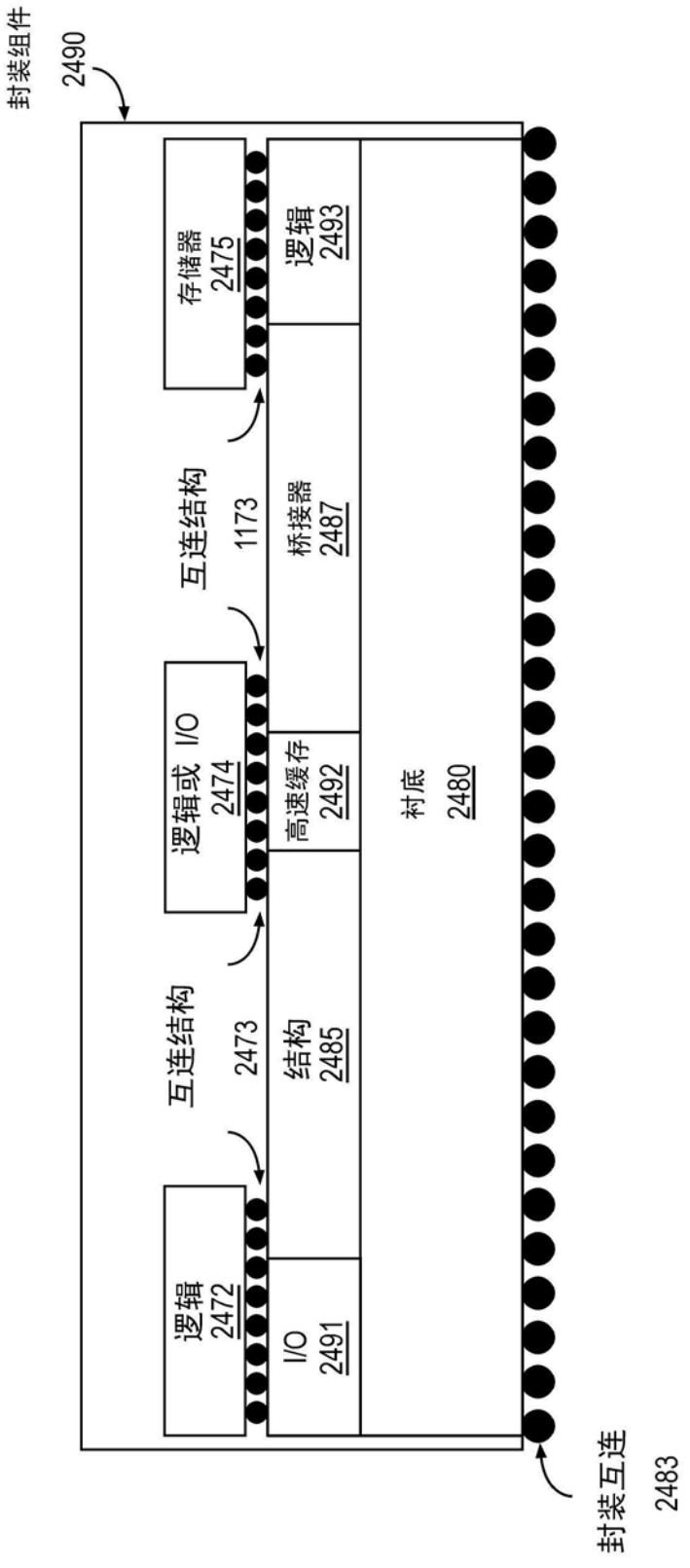

封装组件
2490
存储器
2475
逻辑
2493
互连结构
1173
桥接器
2487
逻辑或 I/O
2474
高速缓存
2492
衬底
2480
互连结构
2473
结构
2485
逻辑
2472
I/O
2491
封装互连
2483

图24C

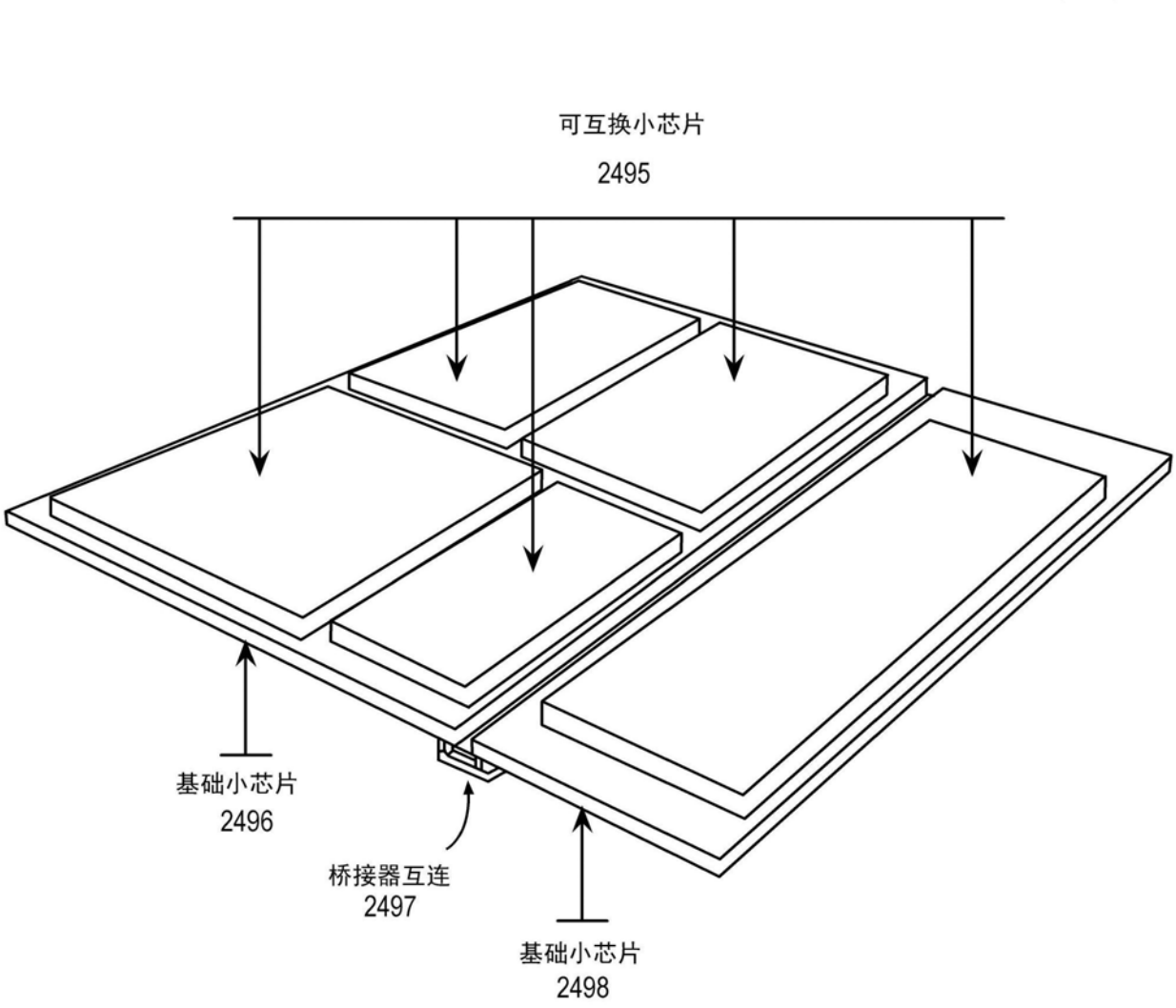


图24D

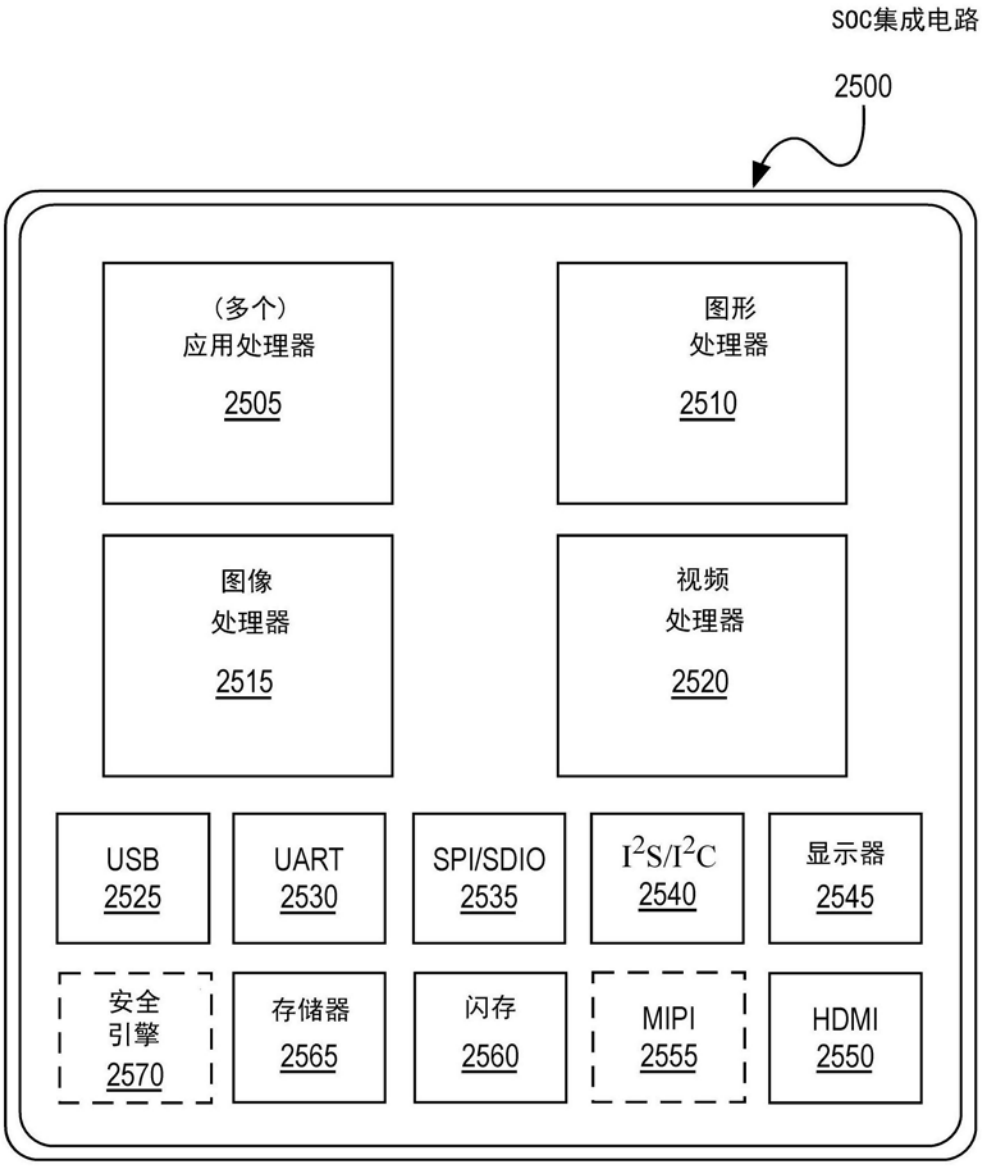

SOC集成电路
2500
(多个)
应用处理器
2505
图形
处理器
2510
图像
处理器
2515
视频
处理器
2520
USB
2525
UART
2530
SPI/SDIO
2535
I2S/I2C
2540
显示器
2545
安全
引擎
2570
存储器
2565
闪存
2560
MIPI
2555
HDMI
2550

图25

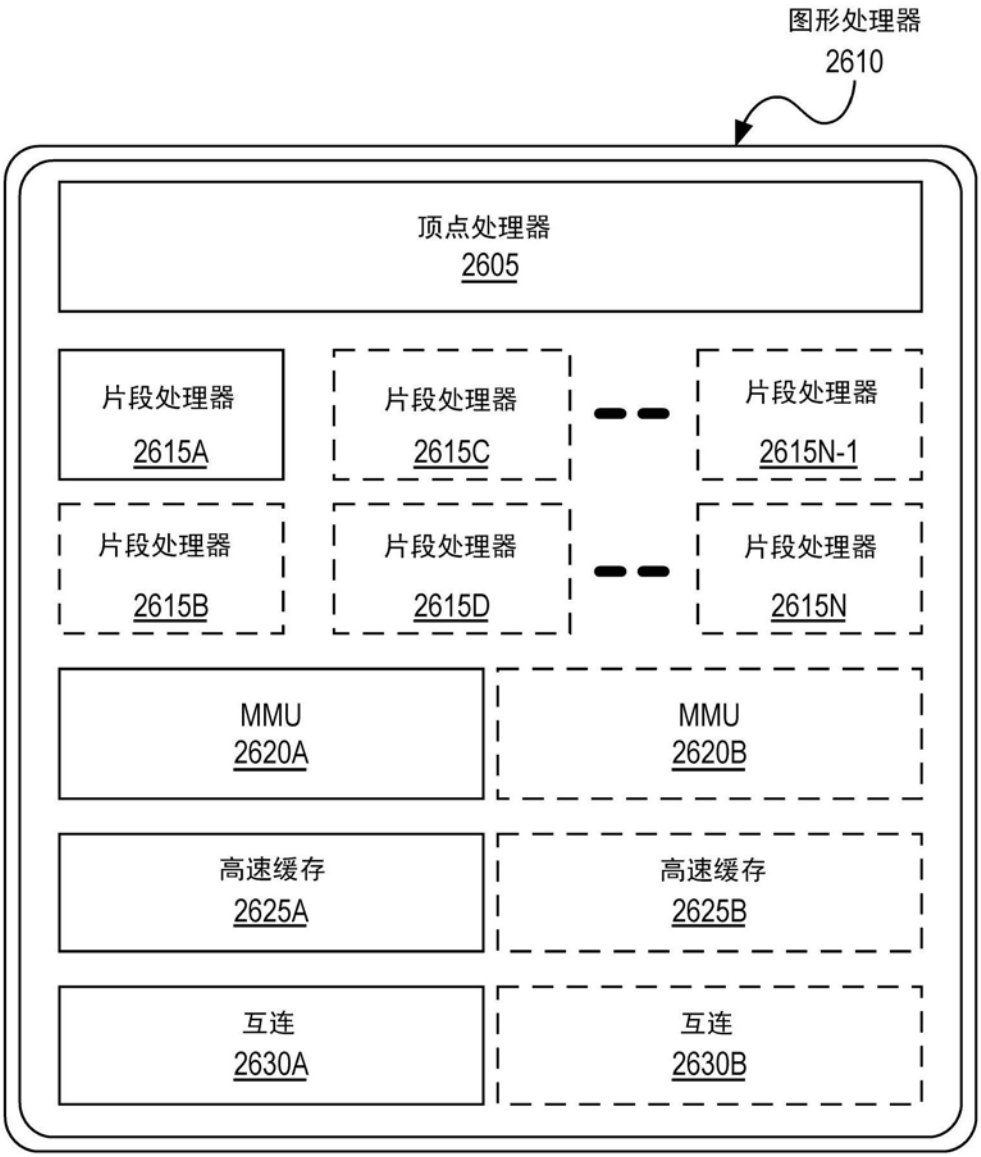

图形处理器
2610
顶点处理器
2605
片段处理器
2615A
片段处理器
2615C
片段处理器
2615N-1
片段处理器
2615B
片段处理器
2615D
片段处理器
2615N
MMU
2620A
MMU
2620B
高速缓存
2625A
高速缓存
2625B
互连
2630A
互连
2630B

图26A

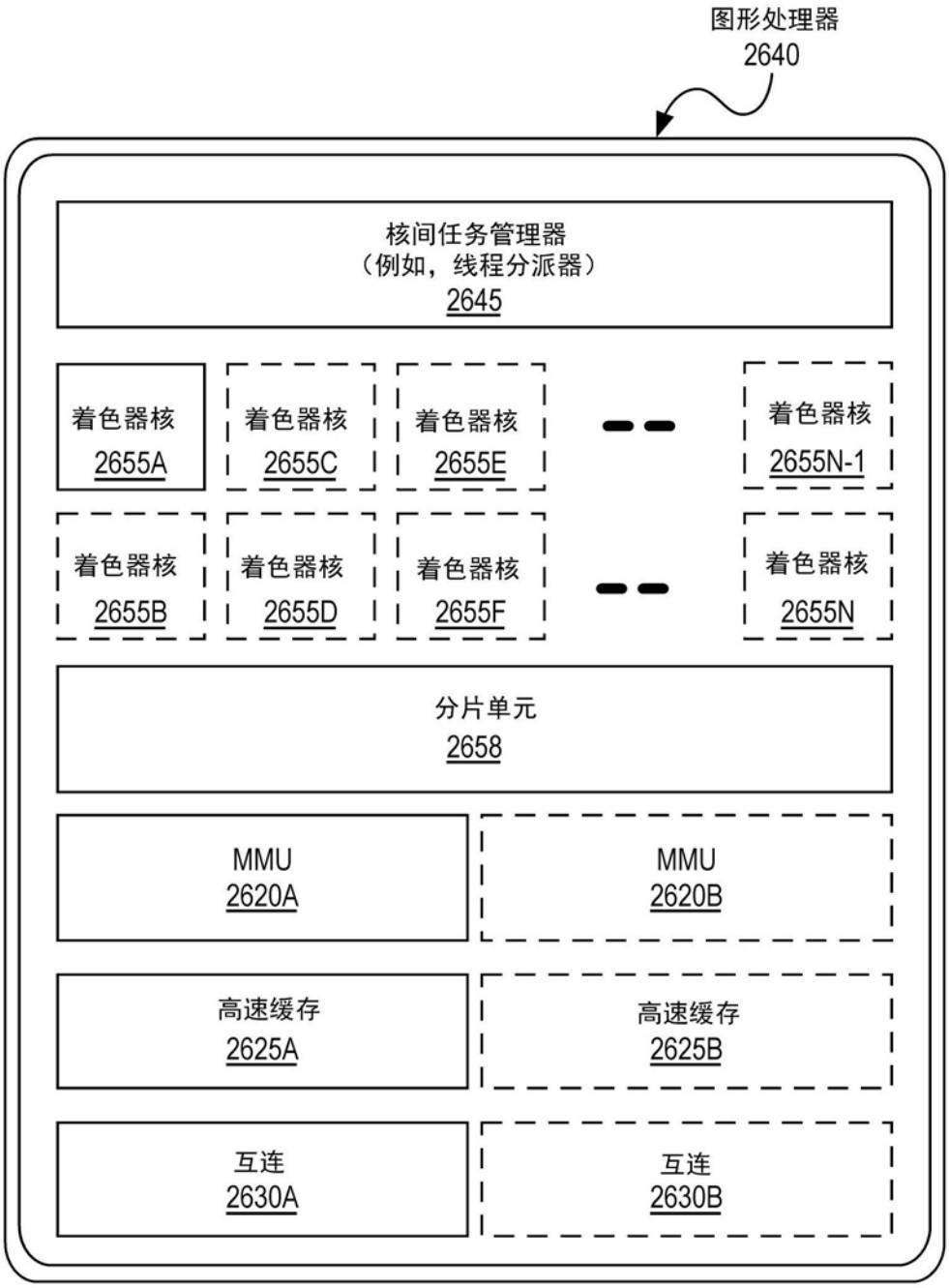

图形处理器
2640
核间任务管理器
（例如，线程分派器）
2645
着色器核
2655A
着色器核
2655C
着色器核
2655E
着色器核
2655N-1
着色器核
2655B
着色器核
2655D
着色器核
2655F
着色器核
2655N
分片单元
2658
MMU
2620A
MMU
2620B
高速缓存
2625A
高速缓存
2625B
互连
2630A
互连
2630B

图26B

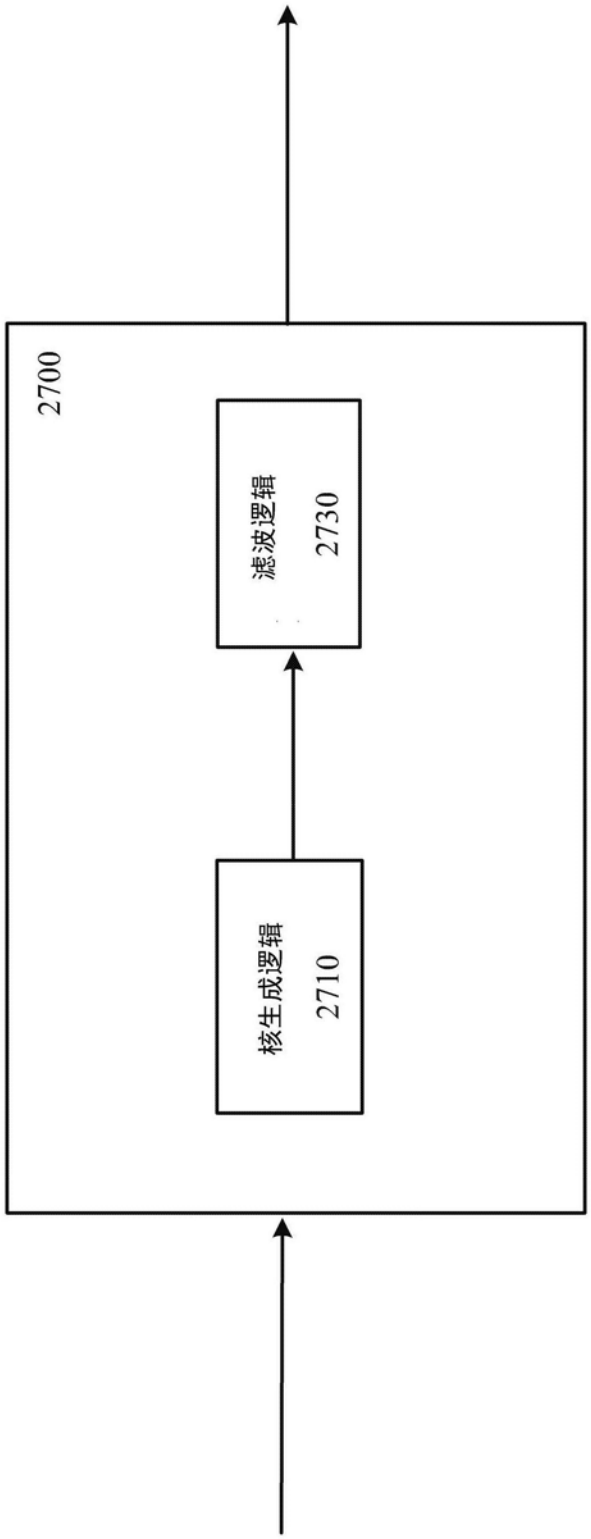


图27A

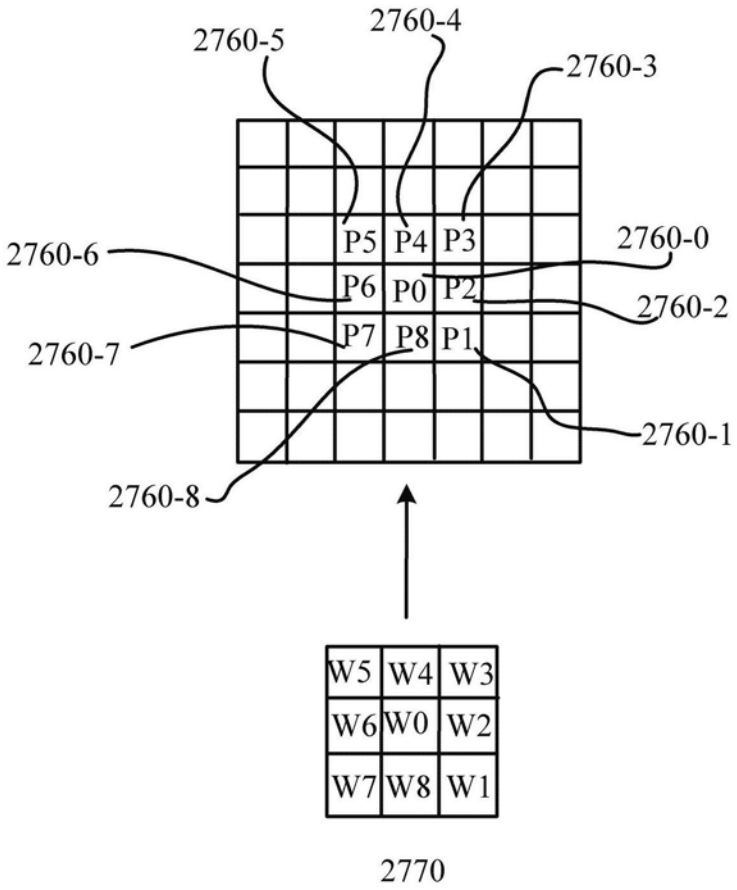

2760-5
2760-4
2760-3
2760-6
2760-0
2760-2
2760-7
2760-1
2760-8
P5
P4
P3
P6
P0
P2
P7
P8
P1
W5
W4
W3
W6
W0
W2
W7
W8
W1
2770

图27B

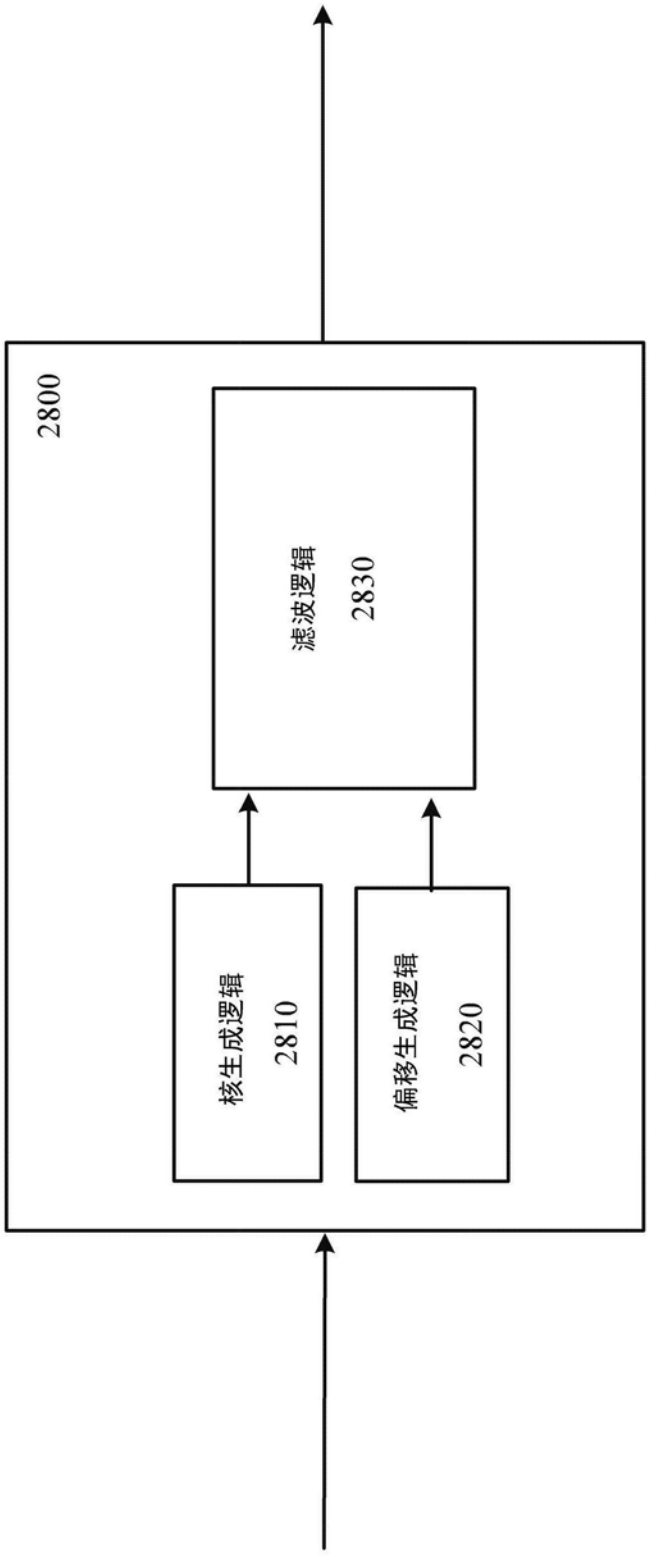


图28A

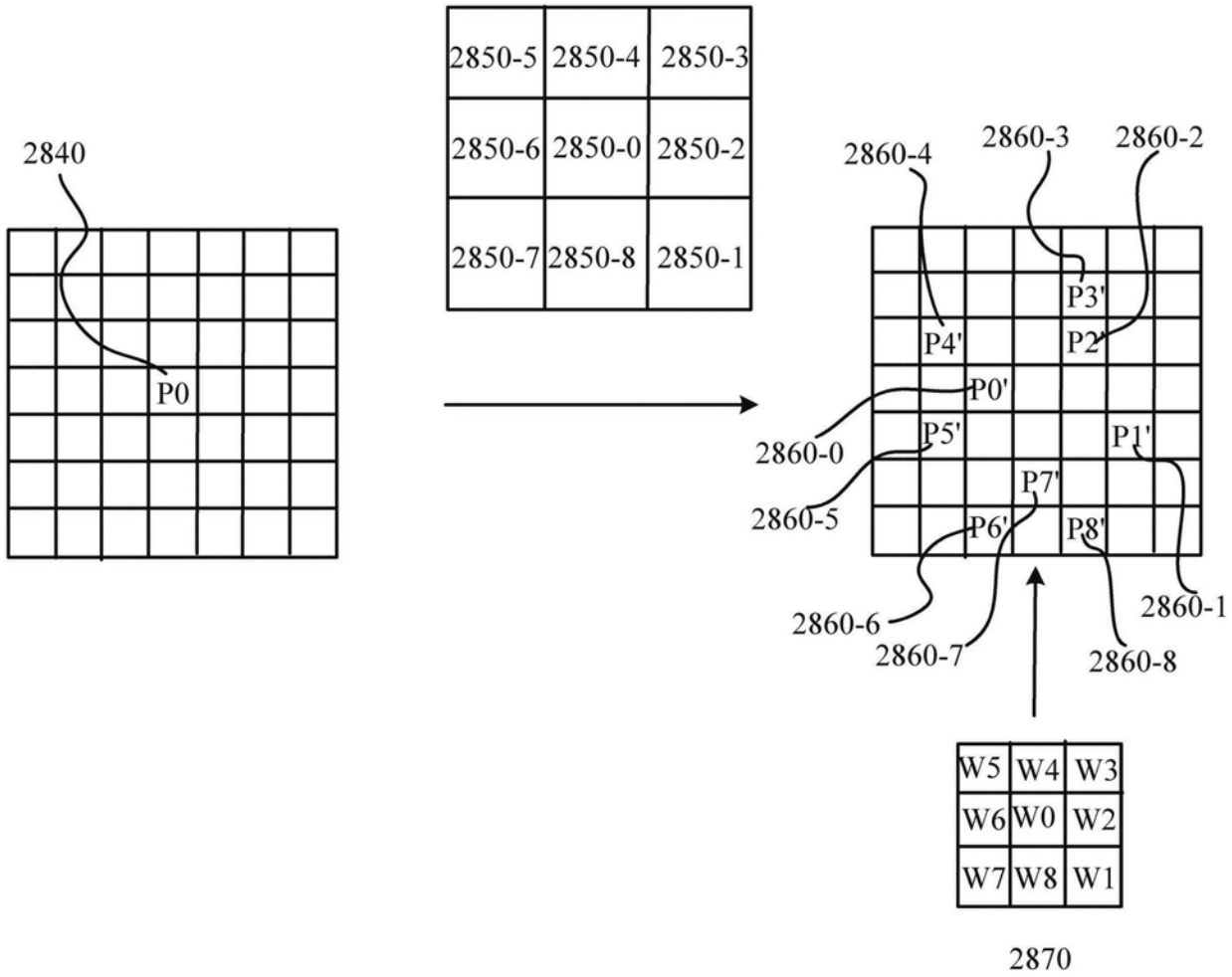

2840
P0
2850-5
2850-4
2850-3
2850-6
2850-0
2850-2
2850-7
2850-8
2850-1
2860-4
2860-3
2860-2
P3'
P4'
P2'
P0'
2860-0
P5'
P1'
2860-5
P7'
P6'
P8'
2860-6
2860-7
2860-8
2860-1
W5
W4
W3
W6
W0
W2
W7
W8
W1
2870

图28B

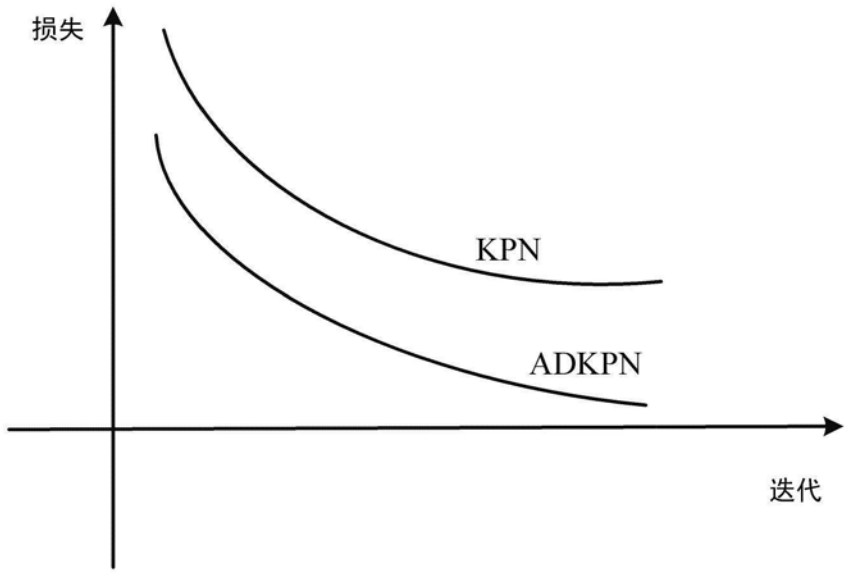

损失
KPN
ADKPN
迭代

图28C

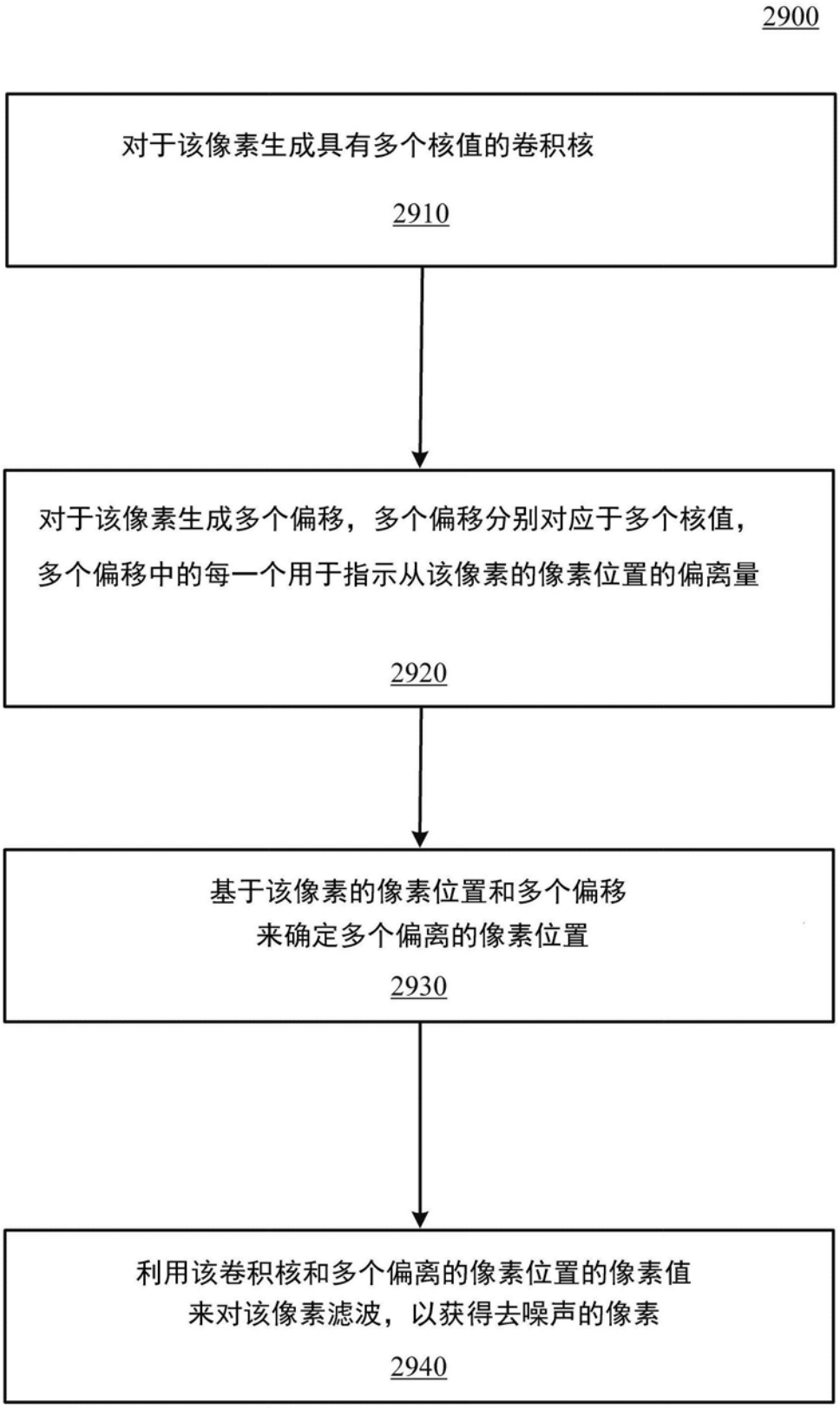

2900
对于该像素生成具有多个核值的卷积核
2910
对于该像素生成多个偏移，多个偏移分别对应于多个核值，
多个偏移中的每一个用于指示从该像素的像素位置的偏离量
2920
基于该像素的像素位置和多个偏移
来确定多个偏离的像素位置
2930
利用该卷积核和多个偏离的像素位置的像素值
来对该像素滤波，以获得去噪声的像素
2940

图29

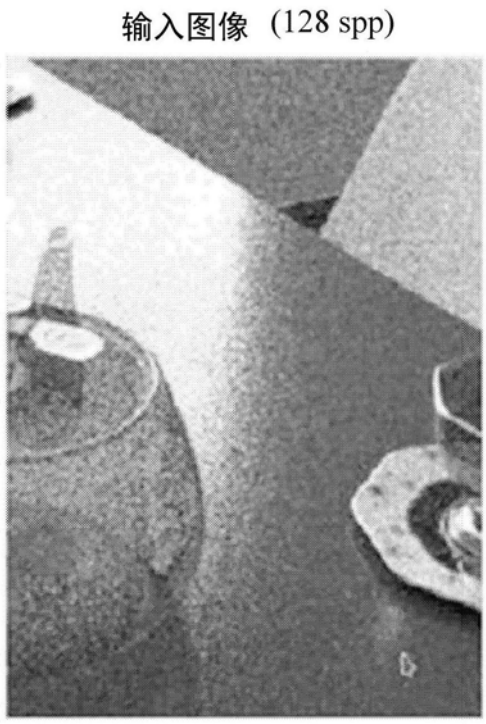


图30A

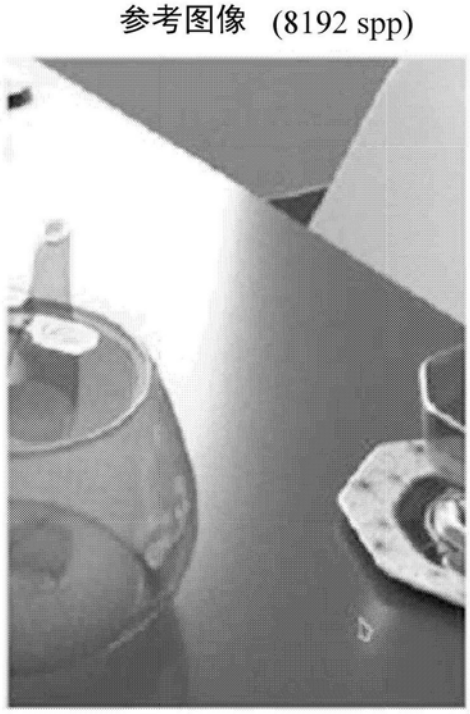


图30B

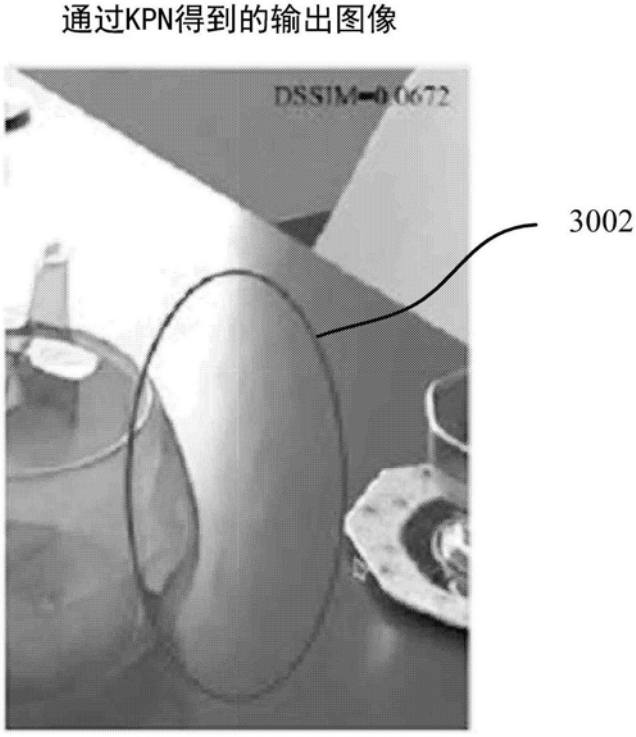

通过KPN得到的输出图像
DSSIM=0.0672
3002

图30C


通过ADKPN得到的输出图像
DSSIM=0.0610

图30D

输入图像 (128 spp)

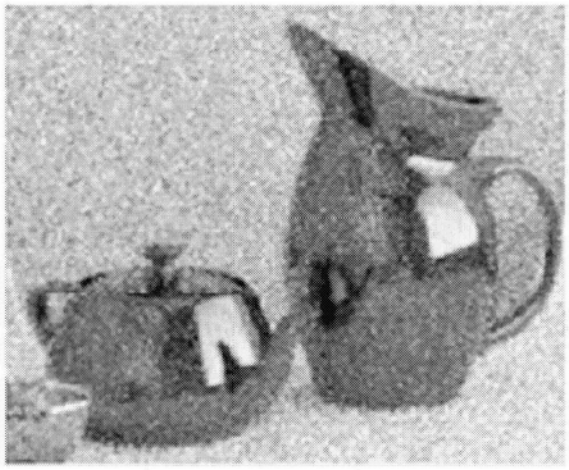


图31A

参考图像 (8192 spp)

图31B

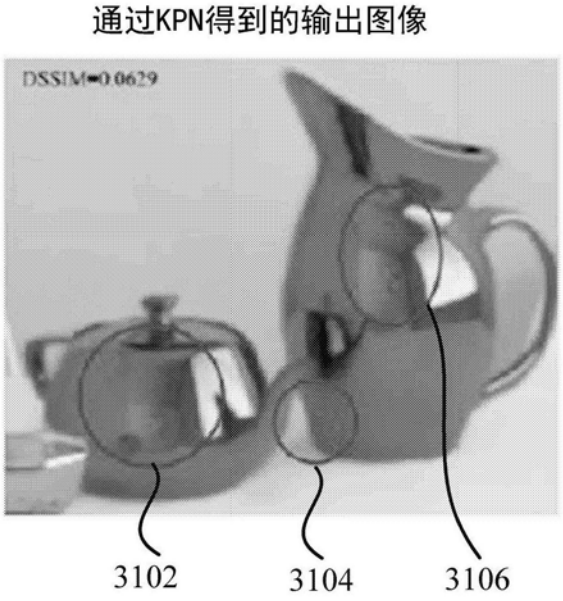

通过KPN得到的输出图像
DSSIM=0.0629
3102
3104
3106

图31C


通过ADKPN得到的输出图像
DSSIM=0.0601

图31D

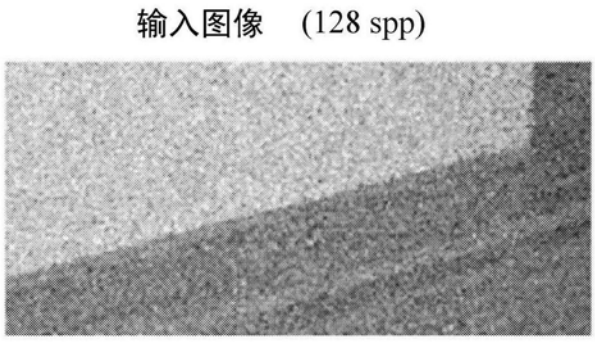


图32A

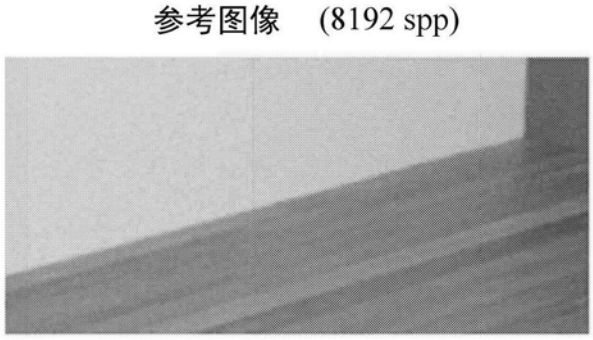


图32B

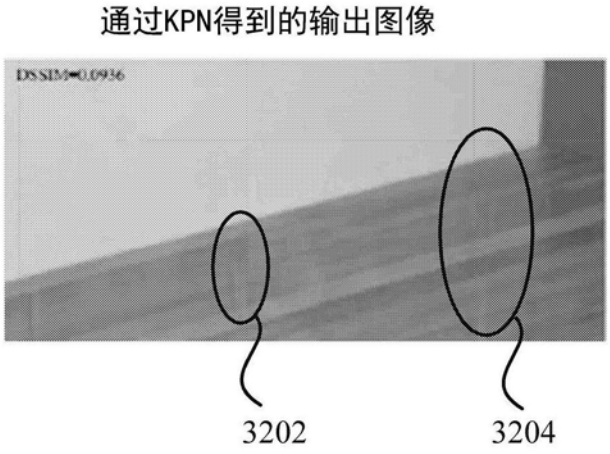


图32C

图32D

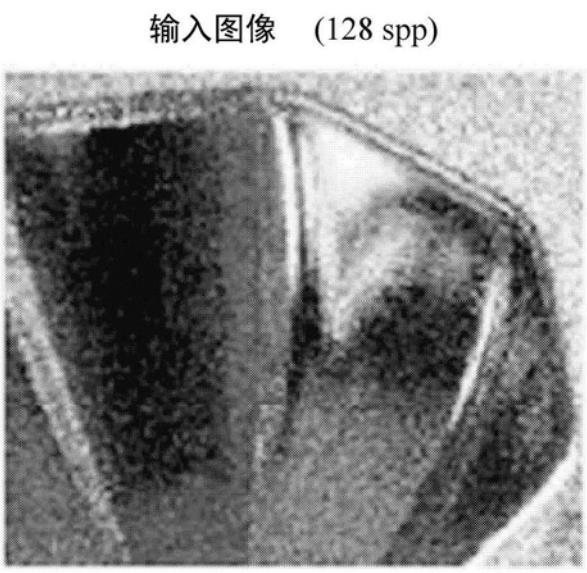


图33A

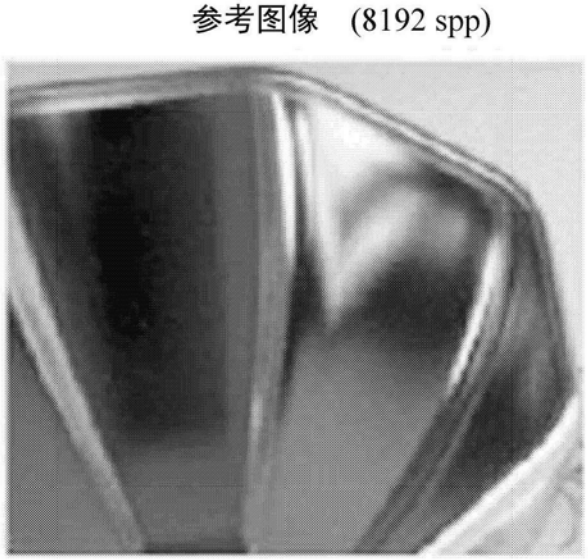


图33B

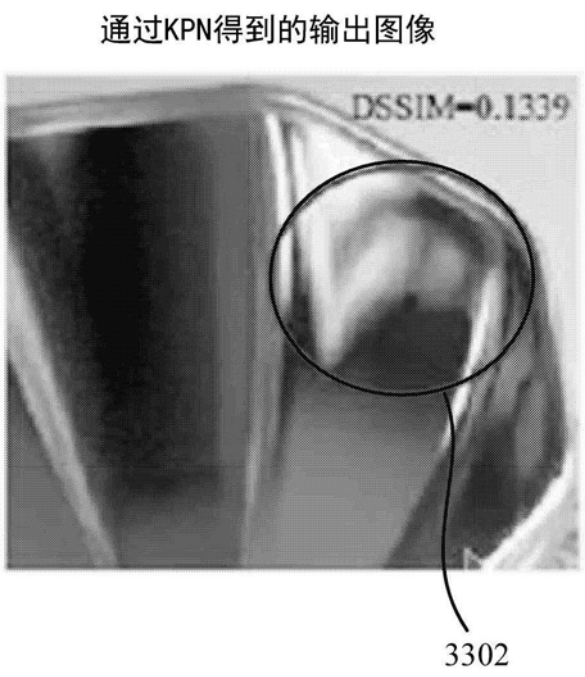


图33C

图33D