(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 113383310 A
(43)申请公布日 2021.09.10

(21)申请号 202080014231.1

(22)申请日 2020.03.14

(30)优先权数据
62/819337 2019.03.15 US
62/819361 2019.03.15 US
62/819435 2019.03.15 US
62/935670 2019.11.15 US

(85)PCT国际申请进入国家阶段日
2021.08.13

(86)PCT国际申请的申请数据
PCT/US2020/022845 2020.03.14

(87)PCT国际申请的公布数据
WO2020/190807 EN 2020.09.24

(71)申请人 英特尔公司
地址 美国加利福尼亚州

(72)发明人 P·苏尔蒂 S·迈于兰
V·安德烈 A·阿普 V·乔治
A·科克 M·麦克费森
E·乌尔-艾哈迈德-瓦尔
V·兰加纳坦 J·雷
L·斯特里拉马萨玛 S·金

(74)专利代理机构 中国专利代理(香港)有限公司 72001
代理人 叶晓勇 姜冰

(51)Int.Cl.
G06F 9/30(2006.01)

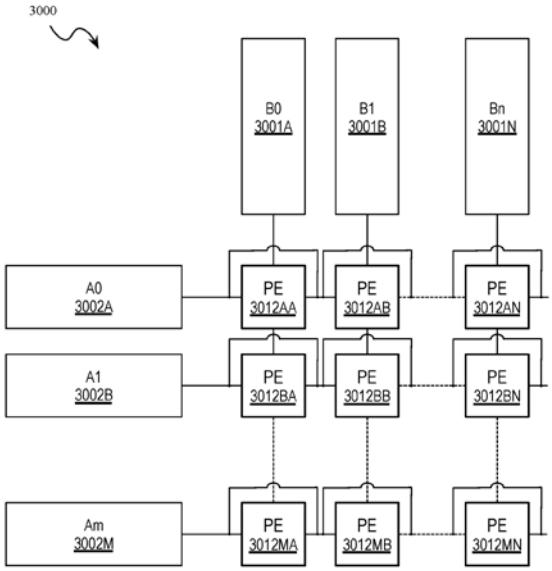
权利要求书2页 说明书69页 附图62页

(54)发明名称

矩阵加速器架构内的脉动分解

(57)摘要

本文描述的实施例包括提供经由脉动处理单元对稀疏数据执行算术的技术的软件、固件和硬件逻辑。一个实施例提供了在使用稀疏数据时优化对脉动阵列的训练和推理的技术。一个实施例提供了在执行稀疏计算操作时使用解压缩信息的技术。一个实施例能够实现经由共享寄存器堆的特殊功能计算阵列的分解。一个实施例能够实现GPGPU上的打包数据压缩和扩展操作。一个实施例提供了利用GPGPU的高速缓存层级内的块稀疏性的技术。



CN 113383310 A

1.一种通用图形处理单元,包括:

矩阵加速器,所述矩阵加速器包括用于旁路具有零值输入的矩阵相乘运算的逻辑,基于与所述输入相关联的元数据执行所述旁路。

2.如权利要求1所述的通用图形处理单元,其中,所述矩阵加速器要接收所述元数据作为与指定所述零值输入的位置的操作数相关联的输入。

3.如权利要求1所述的通用图形处理单元,其中,要针对输入数据的整个集合来预生成所述元数据。

4.如权利要求1所述的通用图形处理单元,其中,要基于到所述矩阵加速器的输入的第一矩阵的行或到所述矩阵加速器的输入的第二矩阵的列来预生成所述元数据。

5.如权利要求1所述的通用图形处理单元,其中,所述矩阵加速器要基于由输入操作数所引用的数据来生成所述元数据,所述数据包括所述零值输入。

6.如权利要求5所述的通用图形处理单元,其中,所述矩阵加速器要基于到所述矩阵加速器的输入的第一矩阵的行和到所述矩阵加速器的输入的第二矩阵的列来生成所述元数据。

7.如权利要求1所述的通用图形处理单元,其中,所述矩阵加速器包括多个处理元件。

8.如权利要求7所述的通用图形处理单元,其中,所述多个处理元件被配置为处理元件的脉动阵列。

9.如权利要求7所述的通用图形处理单元,其中,在所述输入被加载到所述多个处理元件之前,关于所述输入的子矩阵来分析或生成所述元数据。

10.如权利要求7所述的通用图形处理单元,其中,每个处理元件包括用于检测零值输入并基于所述零值输入旁路所述矩阵相乘运算的硬件逻辑。

11.如权利要求10所述的通用图形处理单元,其中,处理元件要在单个时钟循环内旁路具有零值输入的第一矩阵相乘运算和加载第二矩阵相乘运算的输入。

12.一种方法,包括:

在具有矩阵加速器的通用图形处理器上:

分析到要由所述矩阵加速器执行的矩阵相乘运算的输入的元数据,到所述矩阵相乘运算的所述输入包括多个输入矩阵的一个或多个元素;

基于所述元数据确定到所述矩阵相乘运算的所述输入是否包括零值输入;以及

响应于确定所述矩阵相乘运算包括零值输入而旁路所述矩阵相乘运算的至少第一部分。

13.如权利要求12所述的方法,其中,旁路所述矩阵相乘运算的至少一部分包括:

确定到所述矩阵相乘运算的所述第一部分的多个输入中的至少一个为零值输入;

旁路所述多个输入到与所述矩阵相乘运算的所述第一部分相关联的处理元件中的加载;以及

确定到所述矩阵相乘运算的第二部分的多个输入中的每个是非零值输入;

将到所述矩阵相乘运算的所述第二部分的所述多个输入加载到所述处理元件中;以及

经由所述处理元件执行所述矩阵相乘运算的所述第二部分。

14.如权利要求13所述的方法,所述方法还包括:

在第一时钟循环期间,旁路所述多个输入到与所述矩阵相乘运算的所述第一部分相关

联的所述处理元件中的加载;以及

在所述第一时钟循环期间,将到所述矩阵相乘运算的所述第二部分的所述多个输入加载到所述处理元件中。

15.一种数据处理系统,包括:

存储器装置;以及

通用图形处理单元,所述通用图形处理单元与所述存储器装置耦合,其中所述通用图形处理单元包括:

矩阵加速器,所述矩阵加速器包括用于旁路具有零值输入的矩阵相乘运算的逻辑,基于与所述输入相关联的元数据执行所述旁路,其中所述矩阵加速器包括多个处理元件,并且用于接收所述元数据作为与指定所述零值输入的位置的操作数相关联的输入,或者基于由输入操作数所引用的数据而生成所述元数据,所述数据包括所述零值输入。

16.如权利要求15所述的数据处理系统,其中,所述多个处理元件被配置为处理元件的脉动阵列。

17.如权利要求15所述的数据处理系统,其中,在所述输入被加载到所述多个处理元件中之前,关于输入的子矩阵来分析或生成所述元数据。

18.如权利要求15所述的数据处理系统,其中,所述矩阵加速器要基于由所述输入操作数所引用的所述数据来生成所述元数据,所述矩阵加速器要基于到所述矩阵加速器的输入的第一矩阵的行和到所述矩阵加速器的输入的第二矩阵的列来生成所述元数据。

19.如权利要求15所述的数据处理系统,其中,每个处理元件包括用于检测零值输入并基于所述零值输入旁路所述矩阵相乘运算的硬件逻辑。

20.如权利要求19所述的数据处理系统,其中,处理元件要在单个时钟循环内旁路具有零值输入的第一矩阵相乘运算并且加载第二矩阵相乘运算的输入。

矩阵加速器架构内的脉动分解

[0001] 相关申请的交叉引用

本申请涉及并根据35 U.S.C.119(e)要求由Abhishek Appu等人于2019年3月15日提交的题为GRAPHICS PROCESSING的美国临时申请62/819337(代理人案号AC0271-Z)、由Lakshminarayanan Striramassarma等人于2019年3月15日提交的题为GRAPHICS DATA PROCESSING的美国临时申请62/819435(代理人案号AC0285-Z)、由Subramaniam Maiyuran等人于2019年3月15日提交的题为SYSTEMS AND METHODS FOR PARTITIONING CACHE TO REDUCE CACHE ACCESS LATENCY的美国临时申请62/819361(代理人案号AC0286-Z)以及由Abhishek Appu等人于2019年11月15日提交的题为SYSTOLIC ARITHMETIC ON SPARSE DATA的美国临时申请62/935670(代理人案号AC5197-Z)的权益和优先权,所有内容通过引用并入本文中。

技术领域

[0002] 本公开一般涉及数据处理,并且更特定地涉及经由通用图形处理单元的加速矩阵运算。

背景技术

[0003] 当前并行图形数据处理包括被开发成对图形数据执行特定操作的系统和方法,所述特定操作诸如例如线性内插(linear interpolation)、曲面细分(tessellation)、栅格化(rasterization)、纹理映射(texture mapping)、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,已使图形处理器的部分可编程,从而使此类处理器能够支持用于处理顶点数据和片段数据的更广泛种类的操作。

[0004] 为了进一步提高性能,图形处理器通常实现诸如流水线化(pipelining)的处理技术,其试图遍及图形流水线的不同部分并行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最大化图形流水线中的并行处理量。在SIMT架构中,并行线程的群组试图尽可能经常地一起同步执行程序指令以提高处理效率。对于SIMT架构的软件和硬件的总体概述可在Shane Cook,*CUDA Programming*的第三章第37-51页(2013)中找到。

附图说明

[0005] 为了可详细地理解本实施例的上述特征所采用的方式,可通过参考实施例来得到对上文简要概述的实施例的更特定描述,所述实施例中的一些在附图中示出。然而,要注意,附图仅示出了典型的实施例,并且因此不应被认为是对其范围的限制。

[0006] 图1是示出被配置成实现本文描述的实施例的一个或多个方面的计算机系统的框图;

图2A-2D示出并行处理器组件;

图3A-3C是图形多处理器和基于多处理器的GPU的框图;

图4A-4F示出了示例性架构,其中多个GPU通信地耦合到多个多核处理器;

图5示出了图形处理流水线;

图6示出了机器学习软件堆栈;

图7示出了通用图形处理单元;

图8示出了多GPU计算系统;

图9A-9B示出了示例性深度神经网络的层;

图10示出了示例性递归神经网络;

图11示出了深度神经网络的训练和部署;

图12是示出分布式学习的框图;

图13示出了适合于使用经训练的模型来执行推理的示例性推理片上系统(SOC);

图14是处理系统的框图;

图15A-15C示出了计算系统和图形处理器;

图16A-16C示出了另外的图形处理器和计算加速器架构的框图;

图17是图形处理器的图形处理引擎的框图;

图18A-18B示出了包括在图形处理器核中采用的处理元件的阵列的线程执行逻辑;

图19示出了另外的执行单元;

图20是示出图形处理器指令格式的框图;

图21是另外的图形处理器架构的框图;

图22A-22B示出了图形处理器命令格式和命令序列;

图23示出了用于数据处理系统的示例性图形软件架构;

图24A是示出IP核开发系统的框图;

图24B示出了集成电路封装组装件的截面侧视图;

图24C示出了包括连接到衬底(例如,基础管芯)的硬件逻辑小芯片的多个单元的封装组装件;

图24D示出了包括可互换小芯片的封装组装件;

图25是示出示例性片上系统集成电路的框图;

图26A-26B是示出供在SoC内使用的示例性图形处理器的框图;

图27示出了根据实施例的附加执行单元;

图28示出了根据实施例的由指令流水线执行的矩阵运算;

图29A-29B示出了根据一些实施例的基于硬件的脉动阵列的细节;

图30示出了包括部分总和回送(loopback)和用于加速稀疏矩阵相乘的电路系统的脉动阵列;

图31A-31C示出了通过跳过零值输入的运算来加速稀疏矩阵相乘的技术;

图32示出了包括分解的脉动阵列的处理资源;

图33示出了用于128位宽操作数的打包字节、打包字和打包双字(dword)的数据类型;

图34示出了包括用于执行GPGPU的打包数据压缩和扩展操作的逻辑的处理系统;

图35A-35B示出了GPGPU打包数据压缩和扩展操作;

图36A-36B示出了神经网络的训练数据内的非结构化稀疏性和块稀疏性之间的比较；

图37示出了包括其中可旁路稀疏数据访问的高速缓存的处理系统；

图38是包括GPGPU数据压缩流水线的图形处理系统的框图；以及

图39是根据实施例的包括图形处理器的计算装置的框图。

具体实施方式

[0007] 图形处理单元(GPU)通信地耦合到主机/处理器核以加速例如图形操作、机器学习操作、模式分析操作和/或各种通用GPU(GPGPU)功能。GPU可通过总线或另一互连(例如，高速互连，诸如PCIe或NVLink)通信地耦合到主机处理器/核。备选地，GPU可与核集成在相同封装或芯片上，并且通过内部处理器总线/互连(即，在封装或芯片内部)通信地耦合到核。不管连接GPU所采用的方式如何，处理器核可以以工作描述符中所包含的命令/指令的序列的形式将工作分配给GPU。GPU接着将专用电路系统/逻辑用于高效地处理这些命令/指令。

[0008] 本文描述的实施例包括提供经由脉动处理单元对稀疏数据执行算术的技术的软件、固件和硬件逻辑。一个实施例提供了在使用稀疏数据时优化对脉动阵列的训练和推理的技术。一个实施例提供了在执行稀疏计算操作时使用解压缩信息的技术。一个实施例能够实现经由共享寄存器堆的特殊功能计算阵列的分解。一个实施例能够实现GPGPU上的打包数据压缩和扩展操作。一个实施例提供了利用GPGPU的高速缓存层级内的块稀疏性的技术。

[0009] 在以下描述中，阐述了许多特定细节以提供更透彻的理解。然而，对于本领域技术人员来说将清楚的是，可在没有这些特定细节中的一个或多个的情况下实践本文描述的实施例。在其它情况下，尚未描述公知的特征，以免模糊本实施例的细节。

[0010] 系统概述

图1是示出被配置成实现本文描述的实施例的一个或多个方面的计算机系统100的框图。计算系统100包括处理子系统101，所述处理子系统101具有一个或多个处理器102和系统存储器104，所述一个或多个处理器102和系统存储器104经由可包括存储器集线器(hub)105的互连路径来通信。存储器集线器105可以是芯片集组件内的单独组件，或可集成在一个或多个处理器102内。存储器集线器105经由通信链路106与I/O子系统111耦合。I/O子系统111包括I/O集线器107，所述I/O集线器107可使得计算系统100能够从一个或多个输入装置108接收输入。另外，I/O集线器107可使得显示控制器能够将输出提供给一个或多个显示装置110A，所述显示控制器可被包括在一个或多个处理器102中。在一个实施例中，与I/O集线器107耦合的一个或多个显示装置110A可包括局部、内部或嵌入式显示装置。

[0011] 处理子系统101例如包括一个或多个并行处理器112，所述一个或多个并行处理器112经由总线或其它通信链路113耦合到存储器集线器105。通信链路113可以是任何数量的基于标准的通信链路技术或协议之一(诸如但不限于，PCI Express)，或可以是供应商特定的通信接口或通信组构。一个或多个并行处理器112可形成计算上集中的并行或向量处理系统，所述系统可包括大量处理核和/或处理集群(诸如，集成众核(MIC)处理器)。例如，一个或多个并行处理器112形成图形处理子系统，所述图形处理子系统可将像素输出到经由I/O集线器107耦合的一个或多个显示装置110A之一。一个或多个并行处理器112还可包括

显示控制器和显示器接口(未示出)以能够实现与一个或多个显示装置110B的直接连接。

[0012] 在I/O子系统111内,系统存储单元114可连接到I/O集线器107以提供用于计算系统100的存储机制。I/O开关116可用于提供接口机制以能够实现I/O集线器107与其它组件之间的连接,所述其它组件诸如可集成到平台中的网络适配器118和/或无线网络适配器119,以及可经由一个或多个附加(add-in)装置120添加的各种其它装置。(一个或多个)附加装置120还可包括例如一个或多个外部图形处理器装置和/或计算加速器。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可包括以下中的一个或多个:Wi-Fi、蓝牙、近场通信(NFC)、或包括一个或多个无线无线电装置(wireless radio)的其它网络装置。

[0013] 计算系统100可包括未明确示出的其它组件,其包括USB或其它端口连接件、光学存储驱动器、视频捕获装置等,所述其他组件也可连接到I/O集线器107。将图1中的各种组件互连的通信路径可使用任何合适的协议来实现,诸如基于PCI(外设组件互连)的协议(例如,PCI-Express)或任何其它总线或点对点通信接口和/或(一个或多个)协议,诸如NVLink高速互连或本领域中已知的互连协议。

[0014] 一个或多个并行处理器112可合并针对图形和视频处理进行优化的电路系统,其包括例如视频输出电路系统),并且构成图形处理单元(GPU)。备选地或附加地,一个或多个并行处理器112可合并针对通用处理进行优化的电路系统,同时保持本文中更详细描述的底层计算架构。计算系统100的组件可与一个或多个其它系统元件一起集成在单个集成电路上。例如,一个或多个并行处理器112、存储器集线器105、(一个或多个)处理器102和I/O集线器107可集成到片上系统(SoC)集成电路中。备选地,计算系统100的组件可集成到单个封装中以形成封装中系统(SIP)配置。在一个实施例中,计算系统100的组件的至少一部分可集成到多芯片模块(MCM)中,所述多芯片模块(MCM)可与其它多芯片模块一起互连到模块化计算系统中。

[0015] 将认识到的是,本文中示出的计算系统100是说明性的,并且变化和修改是有可能的。可按期望修改连接拓扑,其包括桥接器的数量和布置、(一个或多个)处理器102的数量和(一个或多个)并行处理器112的数量。例如,系统存储器104可直接而非通过桥接器而被连接到(一个或多个)处理器102,而其它装置经由存储器集线器105与系统存储器104和(一个或多个)处理器102通信。在其它备选拓扑中,(一个或多个)并行处理器112连接到I/O集线器107或直接连接到一个或多个处理器102之一,而不是连接到存储器集线器105。在其它实施例中,I/O集线器107和存储器集线器105可集成到单个芯片中。还可能的是经由多个插口(socket)附连的两组或更多组处理器102,它们可与(一个或多个)并行处理器112的两个或更多个实例耦合。

[0016] 本文中示出的特定组件中的一些是可选的,并且可不被包括在计算系统100的所有实现中。例如,可支持任何数量的附加卡或外设,或可消除一些组件。此外,针对与图1中示出的那些组件类似的组件,一些架构可使用不同的术语。例如,在一些架构中,存储器集线器105可称为北桥(Northbridge),而I/O集线器107可称为南桥(Southbridge)。

[0017] 图2A示出了并行处理器200。并行处理器200可以是如本文描述的GPU、GPGPU等。并行处理器200的各种组件可使用一个或多个集成电路装置(诸如可编程处理器、专用集成电路(ASIC)或现场可编程门阵列(FPGA))来实现。所示出的并行处理器200可以是图1中所示

的(一个或多个)并行处理器112或图1中所示的(一个或多个)并行处理器112中的一个。

[0018] 并行处理器200包括并行处理单元202。并行处理单元包括I/O单元204,所述I/O单元204能够实现与其它装置(包括并行处理单元202的其它实例)的通信。I/O单元204可直接连接到其它装置。例如,I/O单元204经由使用集线器或开关接口(诸如,存储器集线器105)来与其它装置连接。存储器集线器105与I/O单元204之间的连接形成通信链路113。在并行处理单元202内,I/O单元204与主机接口206和存储器交叉开关(memory crossbar)216连接,其中主机接口206接收针对执行处理操作的命令,并且存储器交叉开关216接收针对执行存储器操作的命令。

[0019] 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可将用于执行那些命令的工作操作导引至前端208。在一个实施例中,前端208与调度器210耦合,该调度器210配置成将命令或其它工作项分布至处理集群阵列212。调度器210确保在任务被分布至处理集群阵列212的处理集群之前,处理集群阵列212被适当地配置且处于有效状态。调度器210可经由微控制器上执行的固件逻辑来实现。微控制器实现的调度器210可配置成以粗糙粒度和精细粒度来执行复杂的调度和工作分布操作,从而能够实现处理阵列212上执行的线程的上下文切换和快速抢占(rapid preemption)。优选地,主机软件可经由多个图形处理门铃(doorbell)之一来检验工作负荷在处理阵列212上调度。随后工作负荷可由调度器微控制器内的调度器210逻辑来跨处理阵列212自动地分布。

[0020] 处理集群阵列212可包括多达“N”个处理集群(例如,集群214A、集群214B、直到集群214N)。处理集群阵列212的每个集群214A-214N都可执行大数量的并发线程。调度器210可使用各种调度和/或工作分布算法将工作分配给处理集群阵列212的集群214A-214N,这可取决于针对每种类型的程序或计算而产生的工作负荷而变化。调度可由调度器210动态地处置,或者可在配置用于由处理集群阵列212执行的程序逻辑的编译期间部分地由编译器逻辑进行辅助。可选地,可将处理集群阵列212的不同集群214A-214N分配用于处理不同类型的程序,或用于执行不同类型的计算。

[0021] 可将处理集群阵列212配置成执行各种类型的并行处理操作。例如,将集群阵列212配置成执行通用并行计算操作。例如,处理集群阵列212可包括用于执行处理任务的逻辑,所述处理任务包括过滤视频和/或音频数据、执行建模操作(包括物理操作)以及执行数据变换。

[0022] 处理集群阵列212配置成执行并行图形处理操作。在其中并行处理器200配置成执行图形处理操作的这样的实施例中,处理集群阵列212可包括用于支持执行此类图形处理操作的附加逻辑,其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它顶点处理逻辑。另外,处理集群阵列212可配置成执行图形处理相关的着色器(shader)程序,诸如但不限于顶点着色器、曲面细分着色器、几何着色器和像素着色器。并行处理单元202可经由I/O单元204来转移来自系统存储器的数据以便处理。在处理期间,可将所转移的数据在处理期间存储到芯片上存储器(例如,并行处理器存储器222),然后将其写回到系统存储器。

[0023] 在其中并行处理单元202用于执行图形处理的实施例中,调度器210可配置成将处理工作负荷划分成近似相等大小的任务,以更好地能够实现将图形处理操作分布到处理集群阵列212的多个集群214A-214N。在这些实施例中的一些中,处理集群阵列212的部分可配

置成执行不同类型的处理。例如，第一部分可配置成执行顶点着色和拓扑生成，第二部分可配置成执行曲面细分和几何着色，并且第三部分可配置成执行像素着色或其它屏幕空间操作，以产生用于显示的渲染图像。由集群214A-214N中的一个或多个所产生的中间数据可存储在缓冲器中以允许中间数据在集群214A-214N之间传送以便进一步处理。

[0024] 在操作期间，处理集群阵列212可经由调度器210来接收要执行的处理任务，所述调度器210从前端208接收定义处理任务的命令。针对图形处理操作，处理任务可包括定义要如何处理数据(例如，要执行什么程序)的命令和状态参数以及要处理的数据的索引，所述数据例如表面(补片(patch))数据、图元(primitive)数据、顶点数据和/或像素数据。调度器210可配置成提取(fetch)与任务相对应的索引，或可从前端208接收索引。前端208可配置成确保在发起由传入命令缓冲器(例如，分批缓冲器、推动缓冲器等)所指定的工作负荷之前处理集群阵列212配置成有效状态。

[0025] 并行处理单元202的一个或多个实例中的每个都可与并行处理器存储器222耦合。并行处理器存储器222可经由存储器交叉开关216来访问，所述存储器交叉开关216可从处理集群阵列212以及I/O单元204接收存储器请求。存储器交叉开关216可经由存储器接口218访问并行处理器存储器222。存储器接口218可包括多个分区单元(例如，分区单元220A、分区单元220B、直到分区单元220N)，其可各自耦合到并行处理器存储器222的一部分(例如，存储器单元)。可将分区单元220A-220N的数量配置成等于存储器单元的数量，使得第一分区单元220A具有对应的第一存储器单元224A，第二分区单元220B具有对应的存储器单元224B，并且第N分区单元220N具有对应的第N存储器单元224N。在其它实施例中，分区单元220A-220N的数量可不等于存储器装置的数量。

[0026] 存储器单元224A-224N可包括各种类型的存储器装置，其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)，其包括图形双数据速率(GDDR)存储器。可选地，存储器单元224A-224N还可包括3D堆叠式存储器，其包括但不限于高带宽存储器(HBM)。本领域技术人员将认识到，存储器单元224A-224N的特定实现可变化，并且可选自各种常规设计之一。渲染目标，诸如帧缓冲器或纹理(texture)映射可跨存储器单元224A-224N存储，从而允许分区单元220A-220N并行写入每个渲染目标的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，以有利于利用系统存储器连同本地高速缓冲存储器的统一存储器设计。

[0027] 可选地，处理集群阵列212的集群214A-214N中的任何一个都具有处理将被写入到并行处理器存储器222内的存储器单元224A-224N中的任何的数据的能力。可将存储器交叉开关216配置成将每个集群214A-214N的输出转移到任何分区单元220A-220N或另一集群214A-214N，其可对输出执行附加处理操作。每个集群214A-214N都可通过存储器交叉开关216与存储器接口218通信，以从各种外部存储器装置读取或写入到各种外部存储器装置。在具有存储器交叉开关216的实施例中的一个中，存储器交叉开关216具有与存储器接口218的连接以与I/O单元204通信，以及与并行处理器存储器222的本地实例的连接，从而使不同处理集群214A-214N内的处理单元能够与系统存储器或对于并行处理单元202不是本地的其它存储器通信。通常，存储器交叉开关216可例如能够使用虚拟通道以分离集群214A-214N与分区单元220A-220N之间的业务流。

[0028] 虽然在并行处理器200内示出了并行处理单元202的单个实例，但是可包括并行处

理单元202的任何数量的实例。例如，可在单个附加卡上提供并行处理单元202的多个实例，或可将多个附加卡互连。并行处理单元202的不同实例可配置成互操作，即使不同实例具有不同数量的处理核、不同量的本地并行处理器存储器和/或其它配置差异也是如此。可选地，并行处理单元202的一些实例相对于其它实例可包括更高精度浮点单元。合并并行处理单元202或并行处理器200的一个或多个实例的系统可采用各种配置和形状因数(form factor)来实现，包括但不限于台式计算机、膝上型计算机、或手持个人计算机、服务器、工作站、游戏控制台和/或嵌入式系统。

[0029] 图2B是分区单元220的框图。分区单元220可以是图2A的分区单元220A-220N之一的实例。如所示出，分区单元220包括L2高速缓存221、帧缓冲器接口225和ROP 226(栅格操作单元)。L2高速缓存221是读/写高速缓存，其配置成执行从存储器交叉开关216和ROP 226接收的加载和存储操作。由L2高速缓存221将读未命中(read miss)和紧急回写请求输出到帧缓冲器接口225以便处理。也可经由帧缓冲器接口225将更新发送到帧缓冲器以便处理。在一个实施例中，帧缓冲器接口225与并行处理器存储器中的存储器单元(诸如，图2A的存储器单元224A-224N(例如，在并行处理器存储器222内))中的一个通过接口连接(interface)。分区单元220还可附加地或备选地经由存储器控制器(未示出)与并行处理器存储器中的存储器单元中的一个通过接口连接。

[0030] 在图形应用中，ROP 226是执行诸如模板印刷(stencil)、z测试、混合之类的栅格操作的处理单元。随后ROP 226输出存储在图形存储器中的经处理的图形数据。在一些实施例中，ROP 226包括压缩逻辑，用于压缩写入到存储器的深度或颜色数据，并且解压缩从存储器读取的深度或颜色数据。压缩逻辑可以是利用多种压缩算法中的一种或多种压缩算法的无损压缩逻辑。由ROP 226所执行的压缩的类型可基于要压缩的数据的统计特性而变化。例如，在一个实施例中，变量颜色压缩(delta color compression)在逐贴片(per-tile)的基础上对深度和颜色数据执行。

[0031] 在每个处理集群(例如，图2A的集群214A-214N)而不是分区单元220内可包括ROP 226。在这样的实施例中，通过存储器交叉开关216传送对于像素数据而不是像素片段数据的读和写请求。经处理的图形数据可在显示装置(例如图1的一个或多个显示装置110)上显示、被路由以供(一个或多个)处理器102进一步处理、或被路由以供图2A的并行处理器200内的处理实体中的一个来进一步处理。

[0032] 图2C是并行处理单元内的处理集群214的框图。例如，处理集群是图2A的处理集群214A-214N之一的实例。处理集群214可配置成并行执行许多线程，其中术语“线程”是指在特定的一组输入数据上执行的特定程序的实例。可选地，在不提供多个独立指令单元的情况下，可使用单指令多数据(SIMD)指令发布技术以支持对大数量线程的并行执行。备选地，使用配置成将指令发布到处理集群中的每一个内的一组处理引擎的公共指令单元，单指令多线程(SIMT)技术可被用于支持大量的一般同步的线程的并行执行。不像SIMD执行制度，其中所有处理引擎通常执行相同的指令，SIMT执行允许不同线程通过给定线程程序更容易地遵循分歧的执行路径。本领域技术人员将理解，SIMD处理制度表示SIMT处理制度的功能子集。

[0033] 可经由流水线管理器232来控制处理集群214的操作，所述流水线管理器232将处理任务分布到SIMT并行处理器。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。所示出的图形多处理器234是SIMT并行处理器的示例性实例。然而，不同架构的各种类型的SIMT并行处理器可被包括在处理集群214内。图形多处理器234的一个或多个实例可被包括在处理集群214内。图形多处理器234可处理数据，并且数据交叉开关240可用于将所处理的数据分布到多个可能目的地(包括其它着色器单元)之一。流水线管理器232可通过指定针对要经由数据交叉开关240来分布的经处理的数据的目的地来促进分布经处理的数据。

[0034] 处理集群214内的每个图形多处理器234可包括相同一组功能执行逻辑(例如，算术逻辑单元、加载-存储单元等)。能以流水线方式来配置功能执行逻辑，采用该流水线方式，在先前的指令完成之前可发布新指令。功能执行逻辑支持各种操作，其包括整数和浮点算术、比较操作、布尔操作、位移位和各种代数函数的计算。可利用相同功能单元硬件来执行不同操作，并且可存在功能单元的任何组合。

[0035] 传送至处理集群214的指令构成线程。跨一组并行处理引擎而执行的一组线程是线程群组。线程群组对不同的输入数据执行相同程序。可将线程群组内的每个线程分配给图形多处理器234内的不同处理引擎。线程群组可包括比图形多处理器234内的处理引擎数量更少的线程。当线程群组包括比处理引擎的数量更少的线程时，处理引擎中的一个或多个在线程群组正在被处理的循环期间可以是空闲的。线程群组也可包括比图形多处理器234内的处理引擎数量更多的线程。当线程群组包括比图形多处理器234内的处理引擎数量更多的线程时，可通过连续时钟循环执行处理。可选地，可在图形多处理器234上并发地执行多个线程群组。

[0036] 图形多处理器234可包括用于执行加载和存储操作的内部高速缓冲存储器。可选地，图形多处理器234可放弃内部高速缓存，并且使用处理集群214内的高速缓冲存储器(例如，L1高速缓存248)。每个图形多处理器234还有权访问可用于在线程之间转移数据并且在所有处理集群214当中共享的分区单元(例如，图2A的分区单元220A-220N)内的L2高速缓存。图形多处理器234还可访问芯片外全局存储器，所述芯片外全局存储器可包括本地并行处理器存储器和/或系统存储器中的一个或多个。可将并行处理单元202外部的任何存储器用作全局存储器。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共享公共指令和数据，所述公共指令和数据可存储在L1高速缓存248中。

[0037] 每个处理集群214可包括MMU 245(存储器管理单元)，所述MMU 245(存储器管理单元)配置成将虚拟地址映射到物理地址中。在其它实施例中，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包括一组页表条目(PTE)，其用于将贴片的虚拟地址映射到物理地址；以及可选地高速缓存行索引。MMU 245可包括可驻留在图形多处理器234或L1高速缓存或处理集群214内的地址转化后备缓冲器(address translation lookaside buffer)(TLB)或高速缓存。物理地址经处理以分布表面数据访问局域性，从而允许在分区单元当中高效的请求交织。高速缓存行索引可用于确定针对高速缓存行的请求是命中还是未命中。

[0038] 在图形和计算应用中，处理集群214可被配置使得每个图形多处理器234耦合到纹理单元236以用于执行纹理映射操作，例如确定纹理样本位置、读取纹理数据和过滤纹理数据。根据需要，从内部纹理L1高速缓存(未示出)或在一些实施例中从图形多处理器234内的L1高速缓存读取纹理数据，并且从L2高速缓存、本地并行处理器存储器或系统存储器提取

所述纹理数据。每个图形多处理器234将经处理的任务输出到数据交叉开关240以将经处理的任务提供给另一处理集群214，以供进一步处理或以经由存储器交叉开关216将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(预栅格操作单元)配置成从图形多处理器234接收数据、将数据引导到ROP单元，所述ROP单元可与如本文描述的分区单元(例如，图2A的分区单元220A-220N)位于一起。preROP 242单元可执行针对颜色混合的优化、组织像素颜色数据和执行地址转化。

[0039] 将认识到，本文描述的核架构是说明性的，并且变形和修改是有可能的。任何数量的处理单元(例如，图形多处理器234、纹理单元236、preROP 242等)可被包括在处理集群214内。此外，虽然仅示出了一个处理集群214，但是如本文描述的并行处理单元可包括处理集群214的任何数量的实例。可选地，每个处理集群214可配置成使用单独且不同的处理单元、L1高速缓存等独立于其它处理集群214来操作。

[0040] 图2D示出了图形多处理器234的示例，图2D中图形多处理器234与处理集群214的流水线管理器232耦合。图形多处理器234具有执行流水线，其包括但不限于：指令高速缓存252、指令单元254、地址映射单元256、寄存器堆(file)258、一个或多个通用图形处理单元(GPGPU)核262和一个或多个加载/存储单元266。GPGPU核262和加载/存储单元266经由存储器和高速缓存互连268与高速缓冲存储器272和共享存储器270耦合。图形多处理器234可另外包括张量和/或光线追踪核263，其包括用于加速矩阵和/或光线追踪操作的硬件逻辑。

[0041] 指令高速缓存252可从流水线管理器232接收待执行的指令流。指令被高速缓存在指令高速缓存252中，并且由指令单元254分派以便执行。指令单元254可将指令分派为线程群组(例如，线程束(warp))，其中线程群组的每个线程被指派给GPGPU核262内的不同执行单元。指令可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何。地址映射单元256可用于将统一地址空间中的地址转化成可由加载/存储单元266访问的不同的存储器地址。

[0042] 寄存器堆258为图形多处理器234的功能单元提供一组寄存器。寄存器堆258为连接到图形多处理器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。在功能单元中的每个之间可划分寄存器堆258，使得每个功能单元分配寄存器堆258的专用部分。例如，在由图形多处理器234执行的不同线程束之间可划分寄存器堆258。

[0043] GPGPU核262可各自包括浮点单元(FPU)和/或整数算术逻辑单元(ALU)，它们用于执行图形多处理器234的指令。在一些实现中，GPGPU核262可包括硬件逻辑，其可以以其它方式驻留在张量和/或光线追踪核263内。GPGPU核262可在架构上是类似的，或可在架构上是不同的。例如且在一个实施例中，GPGPU核262的第一部分包括单精度FPU和整数ALU，而GPGPU核的第二部分包括双精度FPU。可选地，FPU可针对浮点算术来实现IEEE 754-2008标准，或能够实现可变精度浮点算术。图形多处理器234可另外包括一个或多个固定功能或特殊功能单元以执行特定功能(诸如，复制矩形或像素混合操作)。GPGPU核中的一个或多个也可包括固定或特殊功能逻辑。

[0044] GPGPU核262可包括能够对多组数据执行单指令的SIMD逻辑。可选地，GPGPU核262可物理地执行SIMD4、SIMD8和SIMD16指令，并且逻辑上执行SIMD1、SIMD2和SIMD32指令。用于GPGPU核的SIMD指令可由着色器编译器在编译时间生成，或者可在执行针对单程序多数

据(SPMD)或SIMT架构而编写和编译的程序时自动生成。针对SIMT执行模型而配置的程序的多个线程可经由单个SIMD指令而执行。例如且在一个实施例中，执行相同或类似操作的八个SIMT线程可经由单个SIMD8逻辑单元并行地执行。

[0045] 存储器和高速缓存互连268是互连网络，其将图形多处理器234的功能单元中的每个连接到寄存器堆258并连接到共享存储器270。例如，存储器和高速缓存互连268是交叉开关互连，其允许加载/存储单元266在共享存储器270与寄存器堆258之间实现加载和存储操作。寄存器堆258能与GPGPU核262以相同频率操作，由此在GPGPU核262与寄存器堆258之间的数据转移是非常低的时延。共享存储器270可用于能够实现图形多处理器234内的功能单元上执行的线程之间的通信。高速缓冲存储器272可用作例如数据高速缓存，以对功能单元与纹理单元236之间传递的纹理数据进行高速缓存。共享存储器270也可用作程序管理的高速缓存(cached)。在GPGPU核262上执行的线程能以程序方式将除了存储在高速缓冲存储器272内的自动高速缓存的数据之外的数据存储在共享存储器内。

[0046] 图3A-3C示出了根据实施例的另外的图形多处理器。图3A-3B示出了图形多处理器325、350，所述图形多处理器325、350与图2C的图形多处理器234相关，并且可代替这些中的一个使用。因此，本文中任何特征与图形多处理器234的组合的公开也公开了与(一个或多个)图形多处理器325、350的对应组合，但不限于此。图3C示出了图形处理单元(GPU)380，其包括布置成多核群组365A-365N的图形处理资源的专用集合，所述多核群组365A-365N对应于图形多处理器325、350。示出的图形多处理器325、350和多核群组365A-365N可以是能够同时执行大量执行线程的流播多处理器(streaming multiprocessor)(SM)。

[0047] 图3A的图形多处理器325包括涉及图2D的图形多处理器234的执行资源单元的多个附加实例。例如，图形多处理器325可包括指令单元332A-332B、寄存器堆334A-334B和(一个或多个)纹理单元344A-344B的多个实例。图形多处理器325还包括多组图形或计算执行单元(例如，GPGPU核336A-336B、张量核337A-337B、光线追踪核338A-338B)和多组加载/存储单元340A-340B。执行资源单元具有公共指令高速缓存330、纹理和/或数据高速缓冲存储器342以及共享存储器346。

[0048] 各种组件可经由互连组构327通信。互连组构327可包括一个或多个交叉开关(crossbar switch)以能够实现图形多处理器325的各种组件之间的通信。互连组构327可以是分开的、高速网络组构层，图形多处理器325的每个组件堆叠在该分开的高速网络组构层上。图形多处理器325的组件经由互连组构327与远程组件通信。例如，GPGPU核336A-336B、337A-337B以及338A-338B可各自经由互连组构327与共享存储器346通信。互连组构327可仲裁图形多处理器325内的通信以确保组件之间的公平带宽分配。

[0049] 图3B的图形处理器350包括多组执行资源356A-356D，其中每组执行资源包括多个指令单元、寄存器堆、GPGPU核和加载存储单元，如图2D和图3A中所示出。执行资源356A-356D可与(一个或多个)纹理单元360A-360D一致地工作以用于纹理操作，同时共享指令高速缓存354和共享存储器353。例如，执行资源356A-356D可共享指令高速缓存354和共享存储器353，以及纹理和/或数据高速缓冲存储器358A-358B的多个实例。各种组件可经由类似于图3A的互连组构327的互连组构352来通信。

[0050] 本领域技术人员将理解，图1、图2A-2D以及图3A-3B中所描述的架构就本实施例的范畴而言是描述性的而非限制性的。因此，在不背离本文描述的实施例的范畴的情况下，本

文描述的技术可在任何正确配置的处理单元上实现，所述处理单元包括但不限于一个或多个移动应用处理器、一个或多个台式计算机或服务器中央处理单元(CPU)(包括多核CPU)、一个或多个并行处理单元(诸如，图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。

[0051] 如本文描述的并行处理器或GPGPU可通信地耦合到主机/处理器核以加速图形操作、机器学习操作、模式分析操作和各种通用GPU(GPGPU)功能。GPU可通过总线或其它互连(例如，诸如PCIe或NVLink的高速互连)通信地耦合到主机处理器/核。在其它实施例中，GPU可与所述核集成在相同封装或芯片上，并且通过内部处理器总线/互连(即，在所述封装或芯片的内部)通信地耦合到所述核。不管连接GPU所采用的方式如何，处理器核都可采取以工作描述符中所包含的命令/指令的序列的形式将工作分配给GPU。GPU然后使用专用的电路系统/逻辑以用于高效地处理这些命令/指令。

[0052] 图3C示出了包括布置到多核群组365A-365N中的图形处理资源的专用集合的图形处理单元(GPU)380。尽管提供了仅单个多核群组365A的细节，但将领会的是，其它多核群组365A-365N可被配备有图形处理资源的相同或类似集合。关于多核群组365A-365N描述的细节也可适用于本文描述的任何图形多处理器234、325、350。

[0053] 如所示出的，多核群组365A可包括一组图形核370、一组张量核371和一组光线追踪核372。调度器/分派器368调度和分派图形线程以供在各种核370、371、372上执行。一组寄存器堆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、用于存储打包数据元素(整数和/或浮点数据元素)的向量寄存器及用于存储张量/矩阵值的贴片寄存器。贴片寄存器可被实现为向量寄存器的组合集合。

[0054] 一个或多个组合的1级(L1)高速缓存和共享存储器单元373在每个多核群组365A内本地地存储图形数据，诸如纹理数据、顶点数据、像素数据、光线数据、包围体积数据等。一个或多个纹理单元374还可被用于执行纹理操作，诸如纹理映射和采样。由多核群组365A-365N的全部或其子集共享的2级(L2)高速缓存375存储用于多个并发图形线程的图形数据和/或指令。如所示出的，L2高速缓存375可跨多个多核群组365A-365N被共享。一个或多个存储器控制器367将GPU 380耦合到存储器366，所述存储器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)。

[0055] 输入/输出(I/O)电路系统363将GPU 380耦合到一个或多个I/O装置362，诸如数字信号处理器(DSP)、网络控制器或用户输入装置。片上互连可被用于将I/O装置362耦合到GPU 380和存储器366。I/O电路系统363的一个或多个I/O存储器管理单元(IOMMU)364将I/O装置362直接耦合到系统存储器366。可选地，IOMMU 364管理页表的多个集合，以将虚拟地址映射到系统存储器366中的物理地址。I/O装置362、(一个或多个)CPU 361和(一个或多个)GPU 380然后可共享相同虚拟地址空间。

[0056] 在IOMMU 364的一个实现中，IOMMU 364支持虚拟化。在此情况下，它可管理页表的第一集合以将客户/图形虚拟地址映射到客户/图形物理地址，并且管理页表的第二集合以将客户/图形物理地址映射到系统/主机物理地址(例如，在系统存储器366内)。页表的第一和第二集合中的每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，使得新的上下文被提供有对页表的相关集合的访问)。尽管在图3C中未被示出，但多核群组

365A-365N和/或核370、371、372中的每个可包括转化后备缓冲器(TLB),以对客户虚拟到客户物理转化、客户物理到主机物理转化以及客户虚拟到主机物理转化进行高速缓存。

[0057] CPU 361、GPU 380和I/O装置362可被集成在单个半导体芯片和/或芯片封装上。示出的存储器366可被集成在相同芯片上,或者可经由片外(off-chip)接口被耦合到存储器控制器367。在一个实现中,存储器366包括GDDR6存储器,所述GDDR6存储器共享与其它物理系统级存储器相同虚拟地址空间,但是本文描述的基础原理不限于此特定实现。

[0058] 张量核371可包括特别设计成执行矩阵运算的多个执行单元,所述矩阵运算是用于执行深度学习操作的基础计算操作。例如,同时矩阵乘法运算可被用于神经网络训练和推理。张量核371可使用各种操作数精度来执行矩阵处理,所述各种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)。例如,神经网络实现取出每个经渲染的场景的特征,潜在地组合来自多个帧的细节,以构造高质量的最终图像。

[0059] 在深度学习实现中,可调度并行矩阵乘法工作以供在张量核371上执行。神经网络的训练特别要求大量的矩阵点积运算。为了处理N x N x N矩阵相乘的内积公式,张量核371可包括至少N个点积处理元素。在矩阵相乘开始前,一个完整矩阵被加载到贴片寄存器,并且在N个周期的每个周期,第二矩阵的至少一列被加载。每个周期,有被处理的N个点积。

[0060] 取决于特定实现,可以以不同精度存储矩阵元素,所述不同精度包括16位字、8位字节(例如,INT8)和4位半字节(例如,INT4)。可为张量核371指定不同精度模式以确保最高效的精度被用于不同工作负载(例如,诸如可容许量化到字节和半字节的推理工作负载)。

[0061] 光线追踪核372对于实时光线追踪和非实时光线追踪实现二者均可使光线追踪操作加速。特别地,光线追踪核372可包括光线遍历(ray traversal)/交叉电路系统,以用于使用包围体积层级(bounding volume hierarchy)(BVH)来执行光线遍历并且标识封闭在BVH体积之内的图元与光线之间的交叉。光线追踪核372还可包括用于执行深度测试和拣选(culling)(例如,使用Z缓冲器或类似布置)的电路系统。在一个实现中,光线追踪核372与本文描述的图像去噪技术协同执行遍历和交叉操作,其至少一部分可在张量核371上被执行。例如,张量核371可实现深度学习神经网络以执行由光线追踪核372生成的帧的去噪。然而,(一个或多个)CPU 361、图形核370和/或光线追踪核372还可实现去噪和/或深度学习算法的全部或一部分。

[0062] 另外,如上所述,可采用去噪的分布式方法,其中GPU 380在通过网络或高速互连耦合到其它计算装置的计算装置中。在该分布式方法中,互连的计算装置可共享神经网络学习/训练数据来改进速度,利用该速度整个系统学习对不同类型的图像帧和/或不同的图形应用执行去噪。

[0063] 光线追踪核372可处理所有BVH遍历和/或光线-图元交叉,从而避免图形核370以每光线数千个指令而过载。例如,每个光线追踪核372包括用于执行包围盒测试(例如,对于遍历操作)的专用电路系统的第一集合和/或用于执行光线-三角形交叉测试(例如,交叉已被遍历的光线)的专用电路系统的第二集合。因此,例如,多核群组365A可仅仅启动光线探头,并且光线追踪核372独立执行光线遍历和交叉并且将命中(hit)数据(例如,命中、无命中(no hit)、多次命中等)返回到线程上下文。在光线追踪核372执行遍历和交叉操作的同时,其它核370、371被释放以执行其它图形或计算工作。

[0064] 可选地，每个光线追踪核372可包括用于执行BVH测试操作的遍历单元和/或执行光线-图元交叉测试的交叉单元。交叉单元生成“命中”、“无命中”或“多次命中”响应，交叉单元将该响应提供到适当的线程。在遍历和交叉操作期间，其它核(例如，图形核370和张量核371)的执行资源被释放以执行其它形式的图形工作。

[0065] 在下述的一个可选实施例中，使用了混合栅格化/光线追踪方法，其中在图形核370与光线追踪核372之间分布工作。

[0066] 光线追踪核372(和/或其它核370、371)可包括对诸如Microsoft的DirectX光线追踪(DXR)之类的光线追踪指令集的硬件支持，所述DXR包括DispatchRays命令以及光线-生成、最接近-命中、任何-命中和未命中(miss)着色器，这些能够实现对每个对象指派纹理和着色器的独特集合。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，注意本文描述的基础原理不限于任何特定光线追踪ISA。

[0067] 一般而言，各种核372、371、370可支持光线追踪指令集，所述光线追踪指令集包括用于光线生成、最接近命中、任何命中、光线-图元交叉、每图元和分层包围盒构造、未命中、访问及异常(exception)中的一个或多个的指令/功能。更特定地，优选的实施例包括光线追踪指令以执行以下功能中的一个或多个：

光线生成 - 可为每个像素、样本或其它用户定义的工作指派执行光线生成指令。

[0068] 最接近命中 - 可执行最接近命中指令，以用场景内的图元来定位光线的最接近交叉点。

[0069] 任何命中 - 任何命中指令标识场景内的图元与光线之间的多个交叉，潜在地标识新的最接近交叉点。

[0070] 交叉 - 交叉指令执行光线-图元交叉测试并且输出结果。

[0071] 每图元包围盒构造 - 此指令围绕给定图元或图元的群组构建包围盒(例如，在构建新的BVH或其它加速数据结构时)。

[0072] 未命中 - 指示光线未命中场景的指定区域或场景内的所有几何。

[0073] 访问 - 指示光线将遍历的子代体积(children volume)。

[0074] 异常 - 包括各种类型的异常处理程序(例如，针对各种误差状况而被调用)。

[0075] 用于GPU与主机处理器互连的技术

图4A示出了示例性架构，其中多个GPU 410-413(例如，诸如图2A中所示的并行处理器200)通过高速链路440A-440D(例如，总线、点对点互连等)通信地耦合到多个多核处理器405-406。取决于实现，高速链路440A-440D可支持4GB/s、30GB/s、80GB/s或更高的通信吞吐量。可使用各种互连协议，包括但不限于PCIe 4.0或5.0以及NVLink 2.0。然而，本文描述的根本原理不限于任何特定通信协议或吞吐量。

[0076] GPU 410-413中的两个或更多个可通过高速链路442A-442B互连，所述高速链路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连接，所述高速链路443可以是以20GB/s、30GB/s、120GB/s或更高来操作的对称多处理器(SMP)总线。备选地，图4A中所示的各种系统组件之间的所有通信可使用相同协议/链路(例如，通过公共互连组构(fabric))来实现。然而，如所提到的，本文描述的根本原理不限于任何特定类型的互连技术。

[0077] 每个多核处理器405-406可分别经由存储器互连430A-430B通信地耦合到处理器存储器401-402，并且每个GPU 410-413分别通过GPU存储器互连450A-450D通信地耦合到GPU存储器420-423。存储器互连430A-430B以及450A-450D可利用相同或不同的存储器访问技术。通过示例且非限制的方式，处理器存储器401-402和GPU存储器420-423可以是易失性存储器，诸如动态随机存取存储器(DRAM)(包括堆叠式DRAM)、图形DDR SDRAM(GDDR)(例如，GDDR5、GDDR6)或高带宽存储器(HBM)，和/或可以是非易失性存储器，诸如3D XPoint/Optane或Nano-Ram。例如，存储器的某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)。

[0078] 如下文所述，虽然各种处理器405-406和GPU 410-413可分别物理地耦合到特定存储器401-402、420-423，但是可实现统一存储器架构，其中相同虚拟系统地址空间(也称为“有效地址”空间)被分布在所有各个物理存储器当中。例如，处理器存储器401-402可各自包括64GB的系统存储器地址空间，并且GPU存储器420-423可各自包括32GB的系统存储器地址空间(在本示例中产生总共256GB的可寻址存储器)。

[0079] 图4B示出多核处理器407与图形加速模块446之间的互连的额外可选细节。该图形加速模块446可包括集成在线卡上的一个或多个GPU芯片，所述线卡经由高速链路440耦合到处理器407。备选地，可将图形加速模块446与处理器407集成在相同的封装或芯片上。

[0080] 所示出的处理器407包括多个核460A-460D，其各自具有转化后备缓冲器461A-461D和一个或多个高速缓存462A-462D。所述核可包括用于执行指令和处理数据的各种其它组件(例如，指令提取单元、分支预测单元、解码器、执行单元、重排序缓冲器等)，未示出所述其它组件以免模糊本文描述的组件的根本原理。高速缓存462A-462D可包括1级(L1)和2级(L2)高速缓存。另外，一个或多个共享的高速缓存456可被包括在高速缓存层级中并且由多组核460A-460D共享。例如，处理器407的一个实施例包括24个核，其各自具有其自身的L1高速缓存、十二个共享的L2高速缓存和十二个共享的L3高速缓存。在本实施例中，L2和L3高速缓存中的一个由两个邻近的核共享。处理器407和图形加速器集成模块446与系统存储器441连接，所述系统存储器441可包括处理器存储器401-402。

[0081] 通过一致性总线464经由核间通信来针对存储在各种高速缓存462A-462D、456和系统存储器441中的数据和指令而维持一致性。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/电路系统以响应于对于特定高速缓存行的检测到的读或写来通过一致性总线464进行通信。在一个实现中，通过一致性总线464来实现高速缓存监听协议，以监听高速缓存访问。高速缓存监听/一致性技术被本领域技术人员良好地理解，并且此处将不详细描述以免模糊使本文描述的根本原理。

[0082] 可提供代理电路425，所述代理电路425将图形加速模块446通信地耦合到一致性总线464，从而允许图形加速模块446作为核的对等物来加入高速缓存一致性协议。具体而言，接口435提供通过高速链路440(例如，PCIe总线、NVLink等)至代理电路425的连接性，并且接口437将图形加速模块446连接到高速链路440。

[0083] 在一个实现中，加速器集成电路436代表图形加速模块446的多个图形处理引擎431、432、N来提供高速缓存管理、存储器访问、上下文管理和中断管理服务。图形处理引擎431、432、N可各自包括单独的图形处理单元(GPU)。备选地，图形处理引擎431、432、N可包括GPU内不同类型的图形处理引擎，诸如图形执行单元、媒体处理引擎(例如，视频编码器/解

码器)、采样器和blit引擎。换句话说,图形加速模块可以是具有多个图形处理引擎431-432、N的GPU,或图形处理引擎431-432、N可以是集成在公共封装、线卡或芯片上的个体GPU。

[0084] 加速器集成电路436可包括存储器管理单元(MMU)439以用于执行各种存储器管理功能,诸如虚拟至物理存储器转化(也称为有效至真实存储器转化)和用于访问系统存储器441的存储器访问协议。MMU 439还可包括转化后备缓冲器(TLB)(未示出)以用于高速缓存虚拟/有效至物理/真实地址转化。在一个实现中,高速缓存438存储命令和数据以供图形处理引擎431-432、N进行高效访问。存储在高速缓存438和图形存储器433-434、M中的数据可与核高速缓存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来实现,所述代理电路425代表高速缓存438和存储器433-434、M来参与高速缓存一致性机制(例如,将更新发送到高速缓存438(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),以及从高速缓存438接收更新)。

[0085] 一组寄存器445存储用于由图形处理引擎431-432、N执行的线程的上下文数据,并且上下文管理电路448管理线程上下文。例如,上下文管理电路448可在上下文切换期间执行保存和恢复操作以保存和恢复各种线程的上下文(例如,其中,第一线程被保存并且第二线程被存储,使得可由图形处理引擎来执行第二线程)。例如,在上下文切换时,上下文管理电路448可将当前寄存器值存储到存储器中指派的(例如,由上下文指针标识的)区域。然后,其可在返回到上下文时恢复所述寄存器值。中断管理电路447例如可接收并处理从系统装置接收的中断。

[0086] 在一个实现中,由MMU 439将来自图形处理引擎431的虚拟/有效地址转化成系统存储器411中的真实/物理地址。可选地,加速器集成电路436支持多个(例如,4、8、16个)图形加速器模块446和/或其它加速器装置。图形加速器模块446可专用于在处理器407上执行的单个应用,或可在多个应用之间共享。可选地,提供虚拟化的图形执行环境,其中与多个应用或虚拟机(VM)共享图形处理引擎431-432、N的资源。所述资源可被细划分成“切片”,其被分配给不同的VM和/或应用,这基于与所述VM和/或应用相关联的处理要求和优先级来进行。

[0087] 因此,加速器集成电路436充当到对于图形加速模块446的系统的桥接器,并且提供地址转化和系统存储器高速缓存服务。在一个实施例中,为了促进桥接功能性,加速器集成电路436还可包括共享I/O 497(例如,PCIe、USB)和硬件以能够实现电压、计时、性能、热气和安全的系统控制。共享I/O 497可利用单独的物理连接或可穿过高速链路440。另外,加速器集成电路436可为主机处理器提供虚拟化设施,以管理中断、存储器管理和图形处理引擎的虚拟化。

[0088] 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所见的真实地址空间,所以任何主机处理器都可使用有效地址值来直接寻址这些资源。加速器集成电路436的一个可选功能是图形处理引擎431-432、N的物理分离,使得它们对系统呈现为独立的单元。

[0089] 一个或多个图形存储器433-434、M可分别耦合到图形处理引擎431-432、N中的每个。图形存储器433-434、M存储正在由图形处理引擎431-432、N中的每个处理的指令和数据。图形存储器433-434、M可以是易失性存储器,诸如DRAM(包括堆叠式DRAM)、GDDR存储器(例如,GDDR5、GDDR6)或HBM,和/或可以是非易失性存储器,诸如3D XPoint/Optane或Nano-

Ram。

[0090] 为了减少高速链路440上的数据业务，可使用偏置技术以确保存储在图形存储器433-434、M中的数据是将被图形处理引擎431-432、N最频繁地使用的并且优选地不被核460A-460D(至少不是频繁地)使用的数据。类似地，偏置机制试图将由核(且优选地不是图形处理引擎431-432、N)所需的数据保存在系统存储器411和所述核的高速缓存462A-462D、456内。

[0091] 根据图4C中所示出的变型，加速器集成电路436被集成在处理器407内。图形处理引擎431-432、N经由接口437和接口435(其再次可利用任何形式的总线或接口协议)通过高速链路440来直接与加速器集成电路436通信。加速器集成电路436可执行与针对于图4B所描述的那些操作相同的操作，但考虑到其与一致性总线464和高速缓存462A-462D、456的紧密接近性而潜在地处于更高的吞吐量。

[0092] 实施例可支持不同的编程模型，包括专用进程编程模型(没有图形加速模块虚拟化)和共享的编程模型(有虚拟化)。后者可包括由加速器集成电路436控制的编程模型和由图形加速模块446控制的编程模型。

[0093] 在专用进程模型的实施例中，图形处理引擎431-432、N可在单一操作系统下专用于单个应用或进程。单个应用可将其它应用请求汇集(funnel)到图形引擎431-432、N，从而在VM/分区内提供虚拟化。

[0094] 在专用进程编程模型中，可由多个VM/应用分区来共享图形处理引擎431-432、N。共享的模型要求系统管理程序来虚拟化图形处理引擎431-432、N，以允许由每个操作系统进行访问。针对无管理程序的单分区系统，图形处理引擎431-432、N由操作系统所拥有。在两种情况下，操作系统可虚拟化图形处理引擎431-432、N以提供对每个进程或应用的访问。

[0095] 针对共享的编程模型，图形加速模块446或个体图形处理引擎431-432、N使用进程句柄(process handle)来选择进程元素(process element)。进程元素可存储在系统存储器441中，并且使用本文描述的有效地址至真实地址转化技术是可寻址的。进程句柄可以是在向图形处理引擎431-432、N来登记其上下文(那就是说，调用系统软件以将进程元素添加到进程元素链表)时被提供给主机进程的实现特定的值。进程句柄的较低16位可以是进程元素链表内的进程元素的偏移。

[0096] 图4D示出了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441内的应用有效地址空间482存储进程元素483。响应于来自处理器407上执行的应用480的GPU调用(invocation)481可存储进程元素483。进程元素483包含对应的应用480的进程状态。包含在进程元素483中的工作描述符(WD)484可以是由应用请求的单个作业，或可包含指向作业队列的指针。在后一种情况下，WD 484是指向应用的地址空间482中的作业请求队列的指针。

[0097] 图形加速模块446和/或个体图形处理引擎431-432、N可被系统中的进程中的全部或子集共享。例如，本文描述的技术可包括用于设置进程状态并且向图形加速模块446发送WD 484以在虚拟化环境中开始作业的基础设施。

[0098] 在一个实现中，专用进程编程模型是实现特定的。在这个模型中，单个进程拥有图形加速模块446或个体图形处理引擎431。由于图形加速模块446由单个进程所拥有，在指派图形加速模块446之时，管理程序为拥有的分区初始化加速器集成电路436，并且操作系统

为拥有的进程初始化加速器集成电路436。

[0099] 在操作中，加速器集成切片490中的WD提取单元491提取下一个WD 484，所述下一个WD 484包括待由图形加速模块446的图形处理引擎之一来完成的工作的指示。来自WD 484的数据可存储在寄存器445中，并且由如所示出的MMU 439、中断管理电路447和/或上下文管理电路448使用。例如，MMU 439可包括用于访问OS虚拟地址空间485内的段/页表486的段/页行走电路系统(walk circuitry)。中断管理电路447可处理从图形加速模块446接收的中断事件492。当执行图形操作时，由MMU 439将由图形处理引擎431-432、N所生成的有效地址493转化为真实地址。

[0100] 可为每个图形处理引擎431-432、N和/或图形加速模块446复制相同一组寄存器445，并且这组寄存器445可由管理程序或操作系统来初始化。这些复制的寄存器中的每个可被包括在加速器集成切片490中。表1中示出了可由管理程序来初始化的示例性寄存器。

[0101] 表1-管理程序初始化的寄存器

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

表2中示出了可由操作系统来初始化的示例性寄存器。

[0102] 表2-操作系统初始化的寄存器

1	进程和线程标识
2	有效地址(EA)上下文保存/恢复指针
3	虚拟地址(VA)加速器利用记录指针
4	虚拟地址(VA)存储段表指针
5	权限掩蔽
6	工作描述符

每个WD 484可以是特定于特定图形加速模块446和/或图形处理引擎431-432、N的。它包含图形处理引擎431-432、N要完成其工作所要求的全部信息，或者它可以是对其中应用已设立待完成的工作的命令队列的存储器位置的指针。

[0103] 图4E示出了共享模型的附加可选细节。它包括其中存储有进程元素列表499的管理程序真实地址空间498。管理程序真实地址空间498经由管理程序496是可访问的，所述管理程序496虚拟化用于操作系统495的图形加速模块引擎。

[0104] 共享的编程模型允许来自系统中所有分区或分区子集的所有进程或进程子集使用图形加速模块446。存在两个编程模型，其中，图形加速模块446由多个进程和分区共享：时间切片共享和图形定向共享(graphics directed shared)。

[0105] 在这个模型中，系统管理程序496拥有图形加速模块446，并且使其功能可用于所

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

[0106] 针对共享模型，要求应用480可用图形加速模块446类型、工作描述符(WD)、权限掩蔽寄存器(AMR)值和上下文保存/恢复区域指针(CSRP)来进行操作系统495系统调用。图形加速模块446类型描述了用于系统调用的靶向加速功能。图形加速模块446类型可以是系统特定的值。WD专门针对图形加速模块446被格式化，并且可采用如下形式：图形加速模块446命令、对用户定义的结构的有效地址指针、对命令队列的有效地址指针或用于描述待由图形加速模块446完成的工作的任何其它数据结构。在一个实施例中，AMR值是待用于当前进程的AMR状态。被传递到操作系统的值类似于设定AMR的应用。如果加速器集成电路436和图形加速模块446实现不支持用户权限掩蔽覆盖寄存器(UAMOR)，则在管理程序调用中传递AMR之前操作系统可将当前UAMOR值应用于AMR值。可选地，在将AMR放置到进程元素483中之前管理程序496可应用当前权限掩蔽覆盖寄存器(AMOR)值。CSRP可以是寄存器445之一，其包含应用的地址空间482中的区域的有效地址以用于使图形加速模块446保存和恢复上下文状态。如果不要求在作业之间保存状态或当作业被抢占时，这个指针是可选的。上下文保存/恢复区域可以是固定的(pinned)系统存储器。

[0107] 在接收到系统调用时，操作系统495可验证应用480已注册并且已被给予使用图形加速模块446的权限。然后，操作系统495用表3中所示的信息来调用管理程序496。

[0108] 表3-OS至管理程序调用参数

1	工作描述符(WD)
2	权限掩蔽寄存器(AMR)值(潜在地被掩蔽)
3	有效地址(EA)上下文保存/恢复区域指针(CSRP)
4	进程ID(PID)和可选线程ID(TID)
5	虚拟地址(VA)加速器利用记录指针(AURP)
6	存储段表指针(SSTP)的虚拟地址
7	逻辑中断服务号(LISN)

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

[0109] 表4-进程元素信息

1	工作描述符(WD)
2	权限掩蔽寄存器(AMR)值(潜在地被掩蔽)
3	有效地址(EA)上下文保存/恢复区域指针(CSRP)
4	进程ID(PID)和可选线程ID(TID)
5	虚拟地址(VA)加速器利用记录指针(AURP)
6	存储段表指针(SSTP)的虚拟地址

7	逻辑中断服务号(LISN)
8	从管理程序调用参数导出的中断向量表
9	状态寄存器(SR)值
10	逻辑分区ID(LPID)
11	真实地址(RA)管理程序加速器利用记录指针
12	存储装置描述符寄存器(SDR)

管理程序可初始化多个加速器集成切片490寄存器445。

[0110] 如图4F中所示出,在一个可选实现中,采用经由公共虚拟存储器地址空间可寻址的统一存储器,所述公共虚拟存储器地址空间用于访问物理处理器存储器401-402和GPU存储器420-423。在这种实现中,在GPU 410-413上执行的操作利用相同的虚拟/有效存储器地址空间来访问处理器存储器401-402且反之亦然,由此简化可编程性。虚拟/有效地址空间的第一部分可被分配给处理器存储器401,第二部分被分配给第二处理器存储器402,第三部分被分配GPU存储器420,等等。由此跨处理器存储器401-402和GPU存储器420-423中的每个可分布整个虚拟/有效存储器空间(有时称为有效地址空间),从而允许任何处理器或GPU访问任何物理存储器(采用被映射到该存储器的虚拟地址)。

[0111] 可提供MMU 439A-439E中的一个或多个内的偏置/一致性管理电路系统494A-494E,所述偏置/一致性管理电路系统494A-494E确保主机处理器(例如,405)与GPU 410-413的高速缓存之间的高速缓存一致性,并且实现指示其中应存储有某些类型的数据的物理存储器的偏置技术。虽然图4F中示出了偏置/一致性管理电路系统494A-494E的多个实例,但是可在一个或多个主机处理器405的MMU内和/或在加速器集成电路436内实现偏置/一致性电路系统。

[0112] 可使用共享虚拟存储器(SVM)技术来访问GPU附连的存储器420-423并可将其映射为系统存储器的一部分,而无需经受与完全系统高速缓存一致性相关联的典型性能缺陷。GPU附连的存储器420-423作为系统存储器被访问而无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。这种布置允许主机处理器405软件设置操作数和访问计算结果,而没有传统I/O DMA数据拷贝的开销。此类传统拷贝涉及驱动器调用、中断和存储器映射I/O(MMIO)访问,其相对于简单的存储器访问全部都是低效的。同时,访问GPU附连的存储器420-423而无高速缓存一致性开销的能力对于被卸载的计算的执行时间可以是关键的。在具有实质流播写存储器业务的情况下,例如,高速缓存一致性开销可显著减少由GPU 410-413所见的有效写带宽。操作数设置的效率、结果访问的效率和GPU计算的效率在确定GPU卸载的有效性中全部都起到一定作用。

[0113] 可由偏置跟踪器数据结构来驱动GPU偏置与主机处理器偏置之间的选择。例如,可使用偏置表,其可以是每GPU附连的存储器页包括1或2个位的页粒度结构(即,以存储器页的粒度来控制)。可在一个或多个GPU附连的存储器420-423的被偷的(stolen)存储器范围中实现偏置表,其中在GPU 410-413中具有或不具有偏置高速缓存(例如,用于高速缓存偏置表的频繁/最近使用的条目)。备选地,可将整个偏置表维持在GPU内。

[0114] 在一个实现中,在实际访问GPU存储器之前访问与每一次访问GPU附连的存储器420-423相关联的偏置表条目,从而促使以下操作。首先,来自GPU 410-413的在GPU偏置中寻找其页的本地请求被直接转发到对应的GPU存储器420-423。来自GPU的在主机偏置中寻

找其页的本地请求被转发到处理器405(例如,通过如上文所讨论的高速链路)。可选地,来自处理器405的在主机处理器偏置中寻找所请求的页的请求完成像正常存储器读取的请求。备选地,可将针对GPU偏置的页的请求转发到GPU 410-413。然后,如果GPU当前未在使用该页,则GPU可将该页转变到主机处理器偏置。

[0115] 可由基于软件的机制、硬件辅助的基于软件的机制抑或针对有限的一组情况由纯粹基于硬件的机制来改变页的偏置状态。

[0116] 用于改变偏置状态的一个机制采用API调用(例如,OpenCL),其进而调用GPU的装置驱动器,所述装置驱动器进而发送消息(或入队命令描述符)到GPU,从而指导它改变偏置状态并且针对一些转变在主机中执行高速缓存转储清除(flushing)操作。高速缓存转储清除操作对于从主机处理器405偏置转变到GPU偏置来说是需要的,但对于反向转变来说是不需要的。

[0117] 通过暂时渲染由主机处理器405不可高速缓存的GPU偏置页可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,其可或可不立即授予访问(取决于实现)。因此,为减少主机处理器405与GPU 410之间的通信,对于确保GPU偏置页是由GPU所要求但非被主机处理器405所要求(且反之亦然)的那些页是有利的。

[0118] 图形处理流水线

图5示出图形处理流水线500。图形多处理器(诸如,如图2D中的图形多处理器234、图3A的图形多处理器325、图3B的图形多处理器350)可实现所示出的图形处理流水线500。图形多处理器可被包括在如本文描述的并行处理子系统(诸如图2A的并行处理器200)内,其可与图1的(一个或多个)并行处理器112有关并且可代替那些中的一个使用。各种并行处理系统可经由如本文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)可配置成执行顶点处理单元504、曲面细分控制处理单元508、曲面细分评估处理单元512、几何处理单元516和片段/像素处理单元524中的一个或多个的功能。数据组装器502、图元组装器506、514、518、曲面细分单元510、栅格化器522和栅格操作单元526的功能也可由处理集群(例如,图2A的处理集群214)内的其它处理引擎和对应的分区单元(例如,图2A的分区单元220A-220N)来执行。还可使用针对一个或多个功能的专用处理单元来实现图形处理流水线500。还可能的是,由通用处理器(例如,CPU)内的并行处理逻辑来执行图形处理流水线500的一个或多个部分。可选地,图形处理流水线500的一个或多个部分可经由存储器接口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),所述存储器接口528可以是图2A的存储器接口218的实例。图形处理器流水线500也可经由如图3C中的多核群组365A来实现。

[0119] 数据组装器502是可收集对于表面和图元的顶点数据的处理单元。数据组装器502随后将包括顶点属性的顶点数据输出至顶点处理单元504。顶点处理单元504是可编程执行单元,其执行顶点着色器程序,如由顶点着色器程序所指定那样照亮(lighting)并变换顶点数据。顶点处理单元504读取存储在高速缓存、本地或系统存储器中的数据以供在处理顶点数据时使用,并且顶点处理单元504可被编程为将顶点数据从基于对象的坐标表示变换到世界空间坐标空间或归一化装置坐标空间。

[0120] 图元组装器506的第一实例从顶点处理单元504接收顶点属性。图元组装器506按

需读取存储的顶点属性，并且构建图形图元以用于由曲面细分控制处理单元508处理。图形图元包括如由各种图形处理应用编程接口(API)所支持的三角、线段、点、补片(patch)等等。

[0121] 曲面细分控制处理单元508将输入顶点视为用于几何补片的控制点。控制点是从来自补片的输入表示(例如，补片的基础)变换到适合于供曲面细分评估处理单元512在表面评估中使用的表示。曲面细分控制处理单元508也可计算对于几何补片的边缘的曲面细分因数。曲面细分因数应用于单个边缘，并且对与该边缘相关联的依赖视图的细节等级进行量化。曲面细分单元510配置成接收对于补片的边缘的曲面细分因数，并且将补片曲面细分成诸如线、三角或四边形图元的多个几何图元，其被传送到曲面细分评估处理单元512。曲面细分评估处理单元512对细划分的补片的参数化坐标进行操作，以生成对于与几何图元相关联的每个顶点的表面表示和顶点属性。

[0122] 图元组装器514的第二实例从曲面细分评估处理单元512接收顶点属性，所述曲面细分评估处理单元512按需读取存储的顶点属性，并且构建图形图元以便由几何处理单元516处理。几何处理单元516是可编程执行单元，其执行几何着色器程序以按由几何着色器程序所指定那样变换从图元组装器514所接收的图形图元。几何处理单元516可被编程为将图形图元细划分成一个或多个新图形图元，并且运算用于对新图形图元进行栅格化的参数。

[0123] 几何处理单元516可以能够在几何流中增加或删除元素。几何处理单元516将指定新图形图元的参数和顶点输出到图元组装器518。图元组装器518从几何处理单元516接收参数和顶点，并且构建图形图元以便由视口缩放、拣选和裁剪单元(clip unit)520来处理。几何处理单元516读取存储在并行处理器存储器或系统存储器中的数据以供处理几何数据时使用。视口缩放、拣选和裁剪单元520执行裁剪、拣选和视口缩放，并将经处理的图形图元输出到栅格化器522。

[0124] 栅格化器522可执行深度拣选和其它基于深度的优化。栅格化器522还对新图形图元执行扫描转换以生成片段，并且将那些片段和相关联的覆盖数据输出到片段/像素处理单元524。片段/像素处理单元524是配置成执行片段着色器程序或像素着色器程序的可编程执行单元。片段/像素处理单元524按由片段或像素着色器程序所指定那样变换从栅格化器522接收的片段或像素。例如，可将片段/像素处理单元524编程为执行以下操作以产生输出到栅格操作单元526的着色的片段或像素，所述操作包括但不限于纹理映射、着色、混合、纹理校正和透视校正。片段/像素处理单元524可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。可将片段或像素着色器程序配置成以样本、像素、贴片或其它粒度来着色，这取决于针对处理单元配置的采样率。

[0125] 栅格操作单元526是处理单元，其执行包括但不限于模板印刷、z测试、混合之类的栅格操作，并将像素数据作为经处理的图形数据输出以便存储在图形存储器(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，从而显示在所述一个或多个显示装置110上，或者供(一个或多个)并行处理器112或一个或多个处理器102中的一个做进一步处理。可将栅格操作单元526配置成压缩被写入到存储器的z或颜色数据，并且解压缩从存储器读取的z或颜色数据。

[0126] 机器学习概述

上述架构可应用于使用机器学习模型来执行训练和推理操作。机器学习在解决多种任务方面已经成功了。当训练和使用机器学习算法(例如,神经网络)时出现的计算自然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经在深度神经网络的实际实现中起到重要作用。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量最大化。在SIMT架构中,并行线程的群组试图尽可能经常地一起同步执行程序指令,以提高处理效率。由并行机器学习算法实现提供的效率允许使用高容量网络,并使那些网络能够在更大的数据集上得到训练。

[0127] 机器学习算法是可基于数据集学习的算法。例如,机器学习算法可设计成对数据集内的高级抽象建模。例如,图像识别算法可用于确定给定输入属于若干类别中的哪一个;给定输入的话,回归(regression)算法可输出数值;并且模式识别算法可用于生成经转化的文本或执行文本到语音和/或语音识别。

[0128] 示例性类型的机器学习算法是神经网络。有许多类型的神经网络;简单类型的神经网络是前馈网络。前馈网络可实现为其中节点用层布置的非循环图(acyclic graph)。通常,前馈网络拓扑包括被至少一个隐藏层分开的输入层和输出层。隐藏层将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。网络节点经由到相邻层中的节点的边缘而完全连接,但在每个层内的节点之间没有边缘。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(fed forward)”)到输出层的节点,该激活函数基于分别与连接层的边缘中的每个相关联的系数(“权重”)计算网络中每个连续层的节点的状态。根据由被执行的算法所表示的特定模型,来自神经网络算法的输出可采取各种形式。

[0129] 在机器学习算法可用于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、使用表示正被网络建模的问题的训练数据的集以及调整权重直到网络模型以最小误差对训练数据集的所有实例执行。例如,在对于神经网络的受监督学习训练过程期间,网络响应于表示训练数据集中的实例而产生的输出与对该实例的“正确的”经标记输出比较,计算表示输出与经标记输出之间的差异的误差信号,并且调整与连接相关联的权重以在误差信号通过网络层向后传播时使该误差最小化。在对于从训练数据集的实例生成的输出中的每个的误差被最小化时,网络被认为是“经训练的(trained)”。

[0130] 机器学习算法的精确度可明显受到用于训练算法的数据集的质量的影响。训练过程可以是计算密集的并且在常规的通用处理器上可能需要大量时间。因此,并行处理硬件用于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特别有用,这是因为在调整神经网络中的系数中执行的计算使得它们很自然地有助于并行实现。具体地,许多机器学习算法和软件应用已适合于利用通用图形处理装置内的并行处理硬件。

[0131] 图6是机器学习软件堆栈600的通用图。机器学习应用602可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理功能性和/或可用于在部署之前训练神经网络的专用软件。机器学习应用602可实现任何类型的机器智能,其包括但不限于图像识别、测绘(mapping)和定位(localization)、自主导航、语音合成、医学成像或语言翻译。

[0132] 用于机器学习应用602的硬件加速可经由机器学习框架604而启用。机器学习框架604可提供机器学习图元库。机器学习图元是机器学习算法普遍执行的基本操作。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人员创建和优化与机器学习算法相

关联的主计算逻辑，然后在开发新的并行处理器时重新优化计算逻辑。取而代之，机器学习应用可配置成使用机器学习框架604提供的图元来执行必要的计算。示例性图元包括张量卷积、激活函数和池化(pooling)，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还可提供图元来实现由许多机器学习算法执行的基本线性代数子程序，例如矩阵和向量运算。

[0133] 机器学习框架604可处理从机器学习应用602接收的输入数据并且生成到计算框架606的合适的输入。计算框架606可抽取提供给GPGPU驱动器608的底层指令以使机器学习框架604能够经由GPGPU硬件610利用硬件加速而不需要机器学习框架604深入了解GPGPU硬件610的架构。另外，计算框架606可对机器学习框架604实现跨各种类型的和各代GPGPU硬件610的硬件加速。

[0134] GPGPU机器学习加速

图7示出通用图形处理单元700，其可以是图2A的并行处理器200或图1的(一个或多个)并行处理器112。通用处理单元(GPGPU)700可配置成在处理与训练深度神经网络相关联的那类计算工作负荷时特别高效。另外，GPGPU 700可直接链接到GPGPU的其它实例来创建多GPU集群以针对特定深度神经网络提高训练速度。

[0135] GPGPU 700包括主机接口702，用于实现与主机处理器的连接。主机接口702可以是PCI Express接口。然而，主机接口还可以是供应商特定通信接口或通信组构。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布到一组处理集群706A-706H。处理集群706A-706H共享高速缓冲存储器708。高速缓冲存储器708可对于处理集群706A-706H内的高速缓冲存储器充当较高级高速缓存。所示的处理集群706A-706H可对应于如图2A中的处理集群214A-214N。

[0136] GPGPU 700包括经由一组存储器控制器712A-712B而与处理集群706A-H耦合的存储器714A-714B。存储器714A-714B可包括各种类型的存储器装置，其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)，其包括图形双数据速率(GDDR)存储器。存储器714A-714B还可包括3D堆叠存储器，其包括但不限于高带宽存储器(HBM)。

[0137] 处理集群706A-706H中的每个可包括一组图形多处理器，诸如图2D的图形多处理器234、图3A的图形多处理器325、图3B的图形多处理器350、或者可包括如图3C中的多核群组365A-365N。计算集群的图形多处理器包括多种类型的整数和浮点逻辑单元，其可在包括适合于机器学习计算的一定精度范围执行计算操作。例如，至少处理集群706A-706H中的每个中的浮点单元的子集可配置成执行16位或32位浮点运算，而浮点单元的不同子集可配置成执行64位浮点运算。

[0138] GPGPU 700的多个实例可配置成作为计算集群操作。该计算集群用于同步和数据交换的通信机制因实施例而变化。例如，GPGPU 700的多个实例通过主机接口702通信。在一个实施例中，GPGPU 700包括I/O集线器709，其将GPGPU 700与GPU链路710耦合，该GPU链路710能够实现到GPGPU的其它实例的直接连接。GPU链路710可耦合到专用GPU到GPU网桥，其能够实现GPGPU 700的多个实例之间的通信和同步。可选地，GPU链路710与高速互连耦合以向其它GPGPU或并行处理器传送数据和接收数据。GPGPU 700的多个实例可位于独立数据处理系统中并且经由网络装置通信，该网络装置经由主机接口702而可访问。除主机接口702

外或作为主机接口702的备选，GPU链路710可配置成能够实现与主机处理器的连接。

[0139] 尽管示出的GPGPU 700的配置可配置成训练神经网络，但GPGPU 700的备选配置可被配置用于部署在高性能或低功率推理平台内。在推理配置中，相对于训练配置，GPGPU 700包括更少的处理集群706A-706H。另外，与存储器714A-714B相关联的存储器技术在推理和训练配置之间可不同。在一个实施例中，GPGPU 700的推理配置可支持推理特定的指令。例如，推理配置可为通常在对于部署的神经网络的推理操作期间使用的一个或多个8位整数点积指令提供支持。

[0140] 图8示出多GPU计算系统800。该多GPU计算系统800可包括处理器802，其经由主机接口开关804耦合到多个GPGPU 806A-806D。主机接口开关804可以是PCI Express开关装置，其将处理器802耦合到PCI Express总线，处理器802通过该PCI Express总线可与一组GPGPU 806A-806D通信。多个GPGPU 806A-806D中的每个可以是图7的GPGPU 700的实例。GPGPU 806A-806D可经由一组高速点到点GPU到GPU链路816互连。高速GPU到GPU链路可经由专用GPU链路(诸如图7中的GPU链路710)而连接到GPGPU 806A-806D中的每个。P2P GPU链路816在GPGPU 806A-806D中的每个之间实现直接通信而不需要在处理器802被连接到其的主机接口总线上通信。利用被引导到P2P GPU链路的GPU到GPU业务，主机接口总线仍然可用于系统存储器访问或者例如经由一个或多个网络装置与多GPU计算系统800的其它实例通信。虽然在图8中，GPGPU 806A-806D经由主机接口开关804连接到处理器802，但处理器802可备选地包括对P2P GPU链路816的直接支持并且可直接连接到GPGPU 806A-806D。

[0141] 机器学习神经网络实现

本文描述的计算架构可配置成执行特别适合于训练和部署神经网络以用于机器学习的并行处理的类型。神经网络可归纳为具有图关系(graph relationship)的功能的网络。如在本领域内众所周知的，有多种类型的神经网络实现在机器学习中使用。一个示例性类型的神经网络是前馈网络，如之前描述的那样。

[0142] 第二个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)的专门前馈神经网络。因此，CNN通常用于计算视觉和图像识别应用，但它们也可用于其它类型的模式识别，例如语音和语言处理。CNN输入层中的节点被组织成一组“过滤器”(受在视网膜中发现的接受域启发的特征检测器)，并且每组过滤器的输出被传播到网络的连续层中的节点。对于CNN的计算包括对每个过滤器应用卷积数学运算来产生该过滤器的输出。卷积是由两个函数执行以产生第三函数的一种专门的数学运算，该第三函数是该两个原始函数中的一个的修改后的版本。在卷积网络术语中，到卷积的第一函数可称为输入，而第二函数可称为卷积内核。输出可称为特征图。例如，到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阵列的数据。卷积内核可以是多维阵列的参数，其中这些参数由训练过程为神经网络而适配。

[0143] 递归神经网络(RNN)是前馈神经网络系列，其包括层之间的反馈连接。RNN通过跨神经网络的不同部分共享参数数据而启用对顺序(sequential)数据建模。RNN的架构包括循环。循环代表变量的当前值在未来时间对它自身的值的影响，这是因为来自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。该特征由于可在其中组成语言数据的变量性质而使得RNN对于语言处理特别有用。

[0144] 下文描述的图呈现示例性前馈、CNN和RNN网络，以及描述用于分别训练和部署那

些类型的网络中的每个的通用过程。将理解这些描述关于本文描述的任何特定实施例是示例性且非限制性的并且所示出的概念一般可应用于深度神经网络和一般的机器学习技术。

[0145] 上文描述的示例性神经网络可用于执行深度学习。深度学习是使用深度神经网络的机器学习。在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，这与只包括单个隐藏层的浅层神经网络相对。更深层神经网络通常在计算上更密集来训练。然而，网络的额外隐藏层实现多步模式识别，其相对于浅机器学习技术导致输出误差减少。

[0146] 在深度学习中使用的深度神经网络通常包括耦合到后端网络的前端网络，用于执行特征识别，该后端网络表示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)的数学模型。深度学习使机器学习能够被执行而不需要对模型执行手工特征工程化。取而代之，深度神经网络可基于输入数据内的统计结构或相关性来学习特征。习得的特征可提供给数学模型，其可将所检测的特征映射到输出。网络使用的数学模型一般专门针对待执行的特定任务，并且不同的模型将用于执行不同任务。

[0147] 一旦神经网络被构造，则学习模型可应用于网络来训练网络以执行特定任务。学习模型描述了如何调整模型内的权重来减少网络的输出误差。误差后向传播是用于训练神经网络的常见方法。向网络呈现输入向量以用于处理。使用损耗函数将网络的输出与期望输出比较并且对输出层中的神经元中的每个计算误差值。然后，误差值被向后传播直到每个神经元具有相关联的误差值，其大致表示它对原始输出的贡献。然后，网络可使用算法(诸如随机梯度下降算法)从那些误差中学习，以更新神经网络的权重。

[0148] 图9A-9B示出示例性卷积神经网络。图9A示出CNN内的各种层。如在图9A中示出的，用于对图像处理建模的示例性CNN可接收输入908，其描述输入图像的红、绿和蓝(RGB)分量。输入902可被多个卷积层(例如，卷积层904、卷积层906)处理。来自多个卷积层的输出可以可选地被一组完全连接层908处理。完全连接层中的神经元具有到之前的层中的所有激活的完全连接，如之前针对前馈网络描述的那样。来自完全连接层908的输出可用于从网络生成输出结果。完全连接层908内的激活可使用矩阵乘法而不是卷积来计算。不是所有的CNN实现都利用完全连接层908。例如，在一些实现中，卷积层906能够为CNN生成输出。

[0149] 卷积层稀疏连接，这与在完全连接层908中发现的传统神经网络配置不同。传统神经网络层完全连接，使得每个输出单元与每个输入单元交互。然而，如示出的，因为场的卷积的输出是到后续层的节点的输入(而不是场中的节点中的每个节点的相应状态值)，所以卷积层稀疏连接。与卷积层相关联的内核执行卷积运算，其输出被发送给下一个层。在卷积层内执行的降维是使CNN能够放缩以处理大的图像的一个方面。

[0150] 图9B示出CNN的卷积层内的示例性计算级。到CNN的卷积层912的输入可在卷积层914的三个级中被处理。这三个级可包括卷积级916、检测器级918和池化级920。然后，卷积层914可向连续卷积层输出数据。网络的最后的卷积层可生成输出特征图数据或向完全连接层提供输入，例如以对到CNN的输入生成分类值。

[0151] 在卷积级916中并行执行若干卷积以产生一组线性激活。卷积级916可包括仿射变换，其是可规定作为线性变换加平移的任何变换。仿射变换包括这些变换的旋转、平移、缩放和组合。卷积级计算连接到输入中的特定区域的功能(例如，神经元)的输出，该特定区域可被确定作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神经元连接到其的局部输入中的区域之间的点积。来自卷积级916的输出定义被卷积层914的连续级所处理的

一组线性激活。

[0152] 线性激活可被检测器级918处理。在检测器级918中，每个线性激活被非线性激活函数处理。该非线性激活函数使整体网络的非线性性质增加而不影响卷积层的相应场。可使用若干类型的非线性激活函数。一个特定类型是整流线性单元(ReLU)，其使用定义为$f(x) = \max(0, x)$的激活函数，使得激活被阈值化在零。

[0153] 池化级920使用池化函数，其用附近输出的汇总统计来代替卷积层906的输出。池化函数可用于将平移不变性引入神经网络，使得对输入的小的平移不改变池化输出。局部平移的不变性在输入数据中特征的存在比特征的精确位置更重要的情景中可以是有用的。在池化段920期间可使用各种类型的池化函数，其包括最大池化、平均池化和12范数池化。另外，一些CNN实现不包括池化级。取而代之，这样的实现替代相对于之前的卷积级具有增加步幅的额外卷积级。

[0154] 然后，来自卷积层914的输出可被下一个层922处理。下一个层922可以是额外卷积层或完全连接层908中的一个。例如，图9A的第一卷积层904可向第二卷积层906输出，而第二卷积层可向完全连接层908的第一层输出。

[0155] 图10示出示例性递归神经网络1000。在递归神经网络(RNN)中，网络的之前的状态影响网络的当前状态的输出。RNN可使用各种函数用各种方式来构建。RNN的使用一般围绕着使用数学模型以基于输入的先验序列预测未来。例如，给定之前的词序列，RNN可用于执行统计语言建模来预测即将到来的词。所示出的RNN 1000可描述为具有接收输入向量的输入层1002、实现递归功能的隐藏层1004、启用之前状态的‘记忆’的反馈机制1005和输出结果的输出层1006。RNN 1000基于时间步骤操作。RNN在给定时间步骤的状态基于之前的时间步骤经由反馈机制1005而受影响。对于给定时间步骤，隐藏层1004的状态由之前的状态和当前时间步骤的输入来定义。在第一时间步骤的初始输入(x_1)可被隐藏层1004处理。第二输入(x_2)可被隐藏层1004使用在初始输入(x_1)的处理期间所确定的状态信息处理。给定状态可计算为$s_t = f(Ux_t + Ws_{t-1})$，其中U和W是参数矩阵。函数f一般是非线性的，例如双曲正切函数(Tanh)或整流函数$f(x) = \max(0, x)$的变型。然而，在隐藏层1004中使用的特定数学函数可根据RNN 1000的特定实现细节而变化。

[0156] 除所描述的基本CNN和RNN网络外，可启用这些网络上的变化。一个示例RNN变型是长短期记忆(LSTM)RNN。LSTM RNN能够学习长期依赖性，其对于处理较长语言序列是必要的。关于CNN的变型是卷积深度信任网络，其具有与CNN相似的结构并且采用与深度信任网络相似的方式来训练。深度信任网络(DBN)是生成式神经网络，其由多层的概率性(随机)变量组成。DBN可使用贪婪的无监督学习而逐层训练。然后，DBN的习得权重可用于通过对神经网络确定权重的最佳初始集而提供预先训练神经网络。

[0157] 图11示出深度神经网络的训练和部署。一旦已经为任务构造给定网络，就使用训练数据集1102来训练神经网络。已开发各种训练框架1104来启用训练过程的硬件加速。例如，图6的机器学习框架604可配置为训练框架604。训练框架604可钩入(hook into)未经训练的神经网络1106并且使该未经训练的神经网能够使用本文描述的并行处理资源来训练以生成经训练的神经网1108。

[0158] 为了开始训练过程，可随机或通过使用深度信任网络的预先训练来选择初始权

重。然后，采用受监督或无监督方式执行训练循环。

[0159] 受监督学习是其中训练作为介导(mediated)操作而执行的学习方法，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人工分级的情况下。网络处理输入并且将所得的输出与一组预期或期望输出相比较。然后，误差通过系统被向后传播。训练框架1104可调整成调整控制未经训练的神经网络1106的权重。训练框架1104可提供工具来监测未经训练的神经网络1106在多大程度上向适合于基于已知输入数据生成正确应答的模型收敛。随着调整网络的权重来改善神经网络生成的输出，训练过程反复出现。训练过程可持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精确度。然后，可部署经训练的神经网络1108来实现任何数量的机器学习操作，以基于新数据1112的输入生成推理结果1114。

[0160] 无监督学习是其中网络试图使用未标记的数据来训练它自己的学习方法。从而，对于无监督学习，训练数据集1102将包括输入数据而没有任何相关联的输出数据。未经训练的神经网络1106可学习未标记的输入内的分组并且可确定个体输入如何与整体数据集相关。无监督训练可用于生成自组织图，其是能够执行在数据的降维方面有用的操作的一类经训练的神经网络1108。无监督训练还可用于执行异常检测，其允许标识输入数据集中偏离数据的正常模式的数据点。

[0161] 还可采用关于受监督和无监督训练的变化。半监督学习是其中训练数据集1102包括相同分布的标记和无标记数据的混合的技术。递增式学习是其中输入数据被持续用于进一步训练模型的受监督学习的变型。递增式学习使经训练的神经网络1108能够适合于新的数据1112而没有忘记初始训练期间网络内灌输的知识。

[0162] 无论是受监督还是无监督，对于特别是深度神经网络的训练过程对于单个计算节点在计算上可能太密集。代替使用单个计算节点，计算节点的分布式网络可用于加速训练过程。

[0163] 图12是示出分布式学习的框图。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受监督或无监督训练的训练模型。分布式计算节点可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，诸如，如图7中的高度并行通用图形处理单元700。如示出的，分布式学习可以是所执行的模型并行结构1202、数据并行结构1204或模型和数据并行结构1206的组合。

[0164] 在模型并行结构1202中，分布式系统中的不同计算节点可对单个网络的不同部分执行训练计算。例如，神经网络的每个层可由分布式系统的不同处理节点训练。模型并行结构的益处包括放缩到尤其大模型的能力。拆分与神经网络的不同层相关联的计算启用了其中所有层的权重将不适合单个计算节点的存储器的这一非常大的神经网络的训练。在一些实例中，模型并行在执行大的神经网络的无监督训练中可特别有用。

[0165] 在数据并行结构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。然后，来自不同节点的结果组合。虽然不同的方法对于数据并行结构是可能的，但数据并行训练方法全部需要将结果组合并且使每个节点之间的模型参数同步的技术。组合数据的示例性方法包括参数平均和基于更新的数据并行结构。参数平均训练了训练数据子集上的每个节点并且将全局参数(例如，权重、偏置)设置成来自每个节点的参数的平均值。参数平均使用中央参数服务器，其维持参数数据。基于更新的数据并行结

构与参数平均相似，不同之处在于对模型的更新被传输，而不是将来自节点的参数传输到参数服务器。另外，基于更新的数据并行结构可采用分散方式执行，其中更新被压缩并且在节点之间传输。

[0166] 组合模型和数据并行结构1206可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。每个节点可具有模型的完整实例，其中每个节点内的独立GPU用于训练模型的不同部分。

[0167] 分布式训练相对于在单个机器上的训练具有增加的开销。然而，本文描述的并行处理器和GPGPU可各自实现各种技术来减少分布式训练的开销，这些技术包括实现高带宽GPU到GPU数据传输和加速远程数据同步的技术。

[0168] 示例性机器学习应用

机器学习可应用于解决各种技术问题，其包括但不限于计算机视觉、自主驾驶和导航、语音识别和语言处理。在传统上，计算机视觉已经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从再现人类视觉能力(例如识别面部)到创建视觉能力的新类别而变化。例如，计算机视觉应用可配置成从视频中可见的对象中引发的振动中识别声波。并行处理器加速机器学习使计算机视觉应用能够使用比之前可行的明显更大的训练数据集来训练并且使推理系统能够使用低功率并行处理器来部署。

[0169] 并行处理器加速机器学习具有自主驾驶应用，其包括车道和道路标志识别、障碍避免、导航和驾驶控制。加速机器学习技术可用于基于定义对特定训练输入的合适的响应的数据集来训练驾驶模型。本文描述的并行处理器可对用于自主驾驶技术方案的日益复杂的神经网络实现快速训练并且在适合于集成到自主车辆内的移动平台中启用低功率推理处理器的部署。

[0170] 并行处理器加速深度神经网络对自动语音识别(ASR)启用机器学习方法。ASR包括创建这样的函数，即：给定输入声序列，计算最可能的语言序列。使用深度神经网络的加速机器学习已经实现了对之前用于ASR的隐藏马尔可夫模型(HMM)和高斯混合模型(GMM)的替换。

[0171] 并行处理器加速机器学习还可用于加速自然语言处理。自动学习规程可利用统计推理算法来产生对错误或不熟悉输入具有鲁棒性的模型。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。

[0172] 用于机器学习的并行处理平台可分成训练平台和部署平台。训练平台一般是高度并行的并且包括优化，以加速多GPU单节点训练和多节点多GPU训练。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，所部署的机器学习平台一般包括适合于用在诸如摄像机、自主机器人和自主车辆之类的产品的较低功率并行处理器。

[0173] 图13示出适合于使用经训练的模型来执行推理的示例性推理片上系统(SOC)1300。SOC 1300可集成处理组件，其包括媒体处理器1302、视觉处理器1304、GPGPU 1306和多核处理器1308。GPGPU 1306可以是如本文描述的GPGPU(诸如，GPGPU 700)并且多核处理器1308可以是本文描述的多核处理器(诸如，多核处理器405-406)。SOC 1300可另外包括片上存储器1305，其可实现处理组件中的每个可访问的共享片上数据池。处理组件可对低功率操作优化以实现到各种机器学习平台的部署，其包括自主车辆和自主机器人。例如，SOC

1300的一个实现可用作自主车辆的主控制系统的一部分。在SOC 1300配置成供自主车辆使用的情况下,SOC设计且配置成遵从部署管辖的相关功能安全标准。

[0174] 在操作期间,媒体处理器1302和视觉处理器1304可一起(in concert)工作来加速计算机视觉操作。媒体处理器1302可实现多个高分辨率(例如,4K、8K)视频流的低时延解码。经解码的视频流可写入片上存储器1305中的缓冲器。然后,视觉处理器1304可对经解码的视频解析并且在准备使用经训练的图像识别模型准备处理帧时对经解码的视频的帧执行初步处理操作。例如,视觉处理器1304可针对用于在高分辨率视频数据上执行图像识别的CNN加速卷积运算,而后端模型计算由GPGPU 1306执行。

[0175] 多核处理器1308可包括控制逻辑来帮助媒体处理器1302和视觉处理器1304所执行的数据传输和共享存储器操作的定序和同步。多核处理器1308还可充当应用处理器来执行可利用GPGPU 1306的推理计算能力的软件应用。例如,导航和驾驶逻辑的至少一部分可在多核处理器1308上执行的软件中实现。这样的软件可直接向GPGPU 1306发出计算工作负荷或可将计算工作负荷发出到多核处理器1308,该多核处理器1308可向GPGPU 1306卸载那些操作的至少一部分。

[0176] GPGPU 1306可包括计算集群,诸如通用图形处理单元700内的处理集群706A-706H的低功率配置。GPGPU 1306内的计算集群可支持被专门优化以在经训练的神经网络上执行推理计算的指令。例如,GPGPU 1306可支持执行诸如8位和4位整数向量运算之类的低精度计算的指令。

[0177] 附加系统概述

图14是处理系统1400的框图。图14的与本文中任何其它图的元件具有相同或类似的名称的元件描述了与其它图中的元件相同的元件,可以以与其类似的方式操作或起作用,可包括相同的组件,并且可链接到其它实体,如本文中其它地方所述的那些那样,但不限于此。系统1400可用于单处理器台式计算机系统、多处理器工作站系统或具有大量处理器1402或处理器核1407的服务器系统中。系统1400可以是并入在供移动、手持式或嵌入式装置中(诸如在带有到局域或广域网的有线或无线连接性的物联网(IoT)装置内)使用的片上系统(SoC)集成电路内的处理平台。

[0178] 系统1400可以是具有与图1的那些组件对应的组件的处理系统。例如,在不同的配置中,(一个或多个)处理器1402或(一个或多个)处理器核1407可与图1的(一个或多个)处理器102对应。(一个或多个)图形处理器1408可与图1的(一个或多个)并行处理器112对应。外部图形处理器1418可以是图1的(一个或多个)附加装置120之一。

[0179] 系统1400可包括以下各项、与以下各项耦合或者被集成在以下各项内:基于服务器的游戏平台;游戏控制台,包括游戏和媒体控制台、移动游戏控制台、手持式游戏控制台或在线游戏控制台。系统1400可以是以下各项的一部分:移动电话、智能电话、平板计算装置或诸如带有低内部存储容量的膝上型计算机之类的移动因特网连接的装置。处理系统1400还可包括以下各项、与以下各项耦合或者被集成在以下各项内:可穿戴装置,诸如智能手表可穿戴装置;智能眼镜(smart eyewear)或服装,其用增强现实(AR)或虚拟现实(VR)特征来被增强以提供视觉、音频或触觉输出,以补充现实世界视觉、音频或触觉体验或者以其它方式提供文本、音频、图形、视频、全息图像或视频、或者触觉反馈;其它增强现实(AR)装置;或者其它虚拟现实(VR)装置。处理系统1400可包括电视或机顶盒装置,或者是电视或机

顶盒装置的一部分。系统1400可包括以下各项、与以下各项耦合或者被集成在以下各项内：自动驾驶交通工具，诸如公共汽车、牵引车拖车、汽车、摩托车或电动自行车、飞机或滑翔机(或其任何组合)。自动驾驶交通工具可使用系统1400来处理在交通工具周围感测到的环境。

[0180] 一个或多个处理器1402可包括一个或多个处理器核1407以处理指令，所述指令在被执行时，执行用于系统或用户软件的操作。一个或多个处理器核1407中的至少一个处理器核可被配置成处理特定指令集1409。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可处理不同指令集1409，所述指令集1409可包括用于促进对其它指令集的仿真的指令。处理器核1407还可包括其它处理装置，诸如数字信号处理器(DSP)。

[0181] 处理器1402可包括高速缓冲存储器1404。取决于架构，处理器1402可具有单个内部高速缓存或多个级别的内部高速缓存。在一些实施例中，在处理器1402的各种组件之间共享高速缓冲存储器。在一些实施例中，处理器1402还使用外部高速缓存(例如，3级(L3)高速缓存或末级高速缓存(LLC))(未示出)，其可在使用已知高速缓存一致性技术的处理器核1407之间被共享。寄存器堆1406可另外被包括在处理器1402中，并且可包括用于存储不同类型的数据的不同类型的寄存器(例如，整数寄存器、浮点寄存器、状态寄存器和指令指针寄存器)。一些寄存器可以是通用寄存器，而其它寄存器可特定于处理器1402的设计。

[0182] 一个或多个处理器1402可与一个或多个接口总线1410耦合，以在处理器1402与系统1400中的其它组件之间传送通信信号，诸如地址、数据或控制信号。在这些实施例中的一个中，接口总线1410可以是处理器总线，诸如某一版本的直接媒体接口(DMI)总线。然而，处理器总线不限于DMI总线，并且可包括一个或多个外设组件互连总线(Peripheral Component Interconnect bus)(例如，PCI、PCI express)、存储器总线或其它类型的接口总线。例如，(一个或多个)处理器1402可包括集成的存储器控制器1416和平台控制器集线器1430。存储器控制器1416促进存储器装置与系统1400的其它组件之间的通信，而平台控制器集线器(PCH)1430经由本地I/O总线提供到I/O装置的连接。

[0183] 存储器装置1420可以是动态随机存取存储器(DRAM)装置、静态随机存取存储器(SRAM)装置、闪速存储器装置、相变存储器装置或具有适合性能以充当进程存储器的某一其它存储器装置。存储器装置1420可例如作为用于系统1400的系统存储器进行操作，以存储数据1422和指令1421以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还与可选的外部图形处理器1418耦合，所述外部图形处理器1418可与处理器1402中的一个或多个图形处理器1408进行通信以执行图形和媒体操作。在一些实施例中，可由加速器1412协助图形、媒体和/或计算操作，所述加速器1412是可被配置成执行图形、媒体或计算操作的专门集合的协处理器。例如，加速器1412可以是用于优化机器学习或计算操作的矩阵乘法加速器。加速器1412可以是光线追踪加速器，其可被用于与图形处理器1408协同执行光线追踪操作。在一个实施例中，可替代加速器1412或与加速器1412协同使用外部加速器1419。

[0184] 可提供显示装置1411，所述显示装置1411可连接到(一个或多个)处理器1402。显示装置1411可以是如在移动电子装置或膝上型装置中的内部显示装置或者经由显示接口(例如，DisplayPort等)附连的外部显示装置中的一个或多个。显示装置1411可以是头戴式

显示器(HMD),诸如供在虚拟现实(VR)应用或增强现实(AR)应用中使用的立体显示装置。

[0185] 平台控制器集线器1430可使得外设能经由高速I/O总线连接到存储器装置1420和处理器1402。I/O外设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、数据存储装置1424(例如,非易失性存储器、易失性存储器、硬盘驱动器、闪速存储器、NAND、3D NAND、3D XPoint/Optane等)。数据存储装置1424可经由存储接口(例如,SATA)或经由诸如外设组件互连总线(例如,PCI、PCI express)之类的外设总线进行连接。触摸传感器1425可包括触摸屏传感器、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、蓝牙收发器或诸如3G、4G、5G或长期演进(LTE)收发器之类的移动网络收发器。固件接口1428能够实现与系统固件通信,并且可以是例如统一可扩展固件接口(UEFI)。网络控制器1434可能够实现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。音频控制器1446可以是多通道高清晰度音频控制器。在这些实施例中的一些中,系统1400包括用于将传统(legacy)(例如,个人系统2(PS/2))装置耦合到系统的可选的传统I/O控制器1440。平台控制器集线器1430还可连接到一个或多个通用串行总线(USB)控制器1442连接输入装置,诸如键盘和鼠标1443组合、相机1444或其它USB输入装置。

[0186] 将领会的是,示出的系统1400是示例性的而非限制性的,因为以不同方式配置的其它类型的数据处理系统也可被使用。例如,存储器控制器1416和平台控制器集线器1430的实例可被集成到分立的外部图形处理器(诸如,外部图形处理器1418)中。平台控制器集线器1430和/或存储器控制器1416可在一个或多个处理器1402的外部。例如,系统1400可包括外部存储器控制器1416和平台控制器集线器1430,其可被配置为与(一个或多个)处理器1402通信的系统芯片组内的存储器控制器集线器和外设控制器集线器。

[0187] 例如,可使用电路板(“滑板(sled)”),将诸如CPU、存储器和其它组件之类的组件放置在所述滑板上,其被设计用于增加的热性能。诸如处理器之类的处理组件可位于滑板的顶侧上,而诸如DIMM之类的近存储器位于滑板的底侧上。作为通过此设计提供的增强气流的结果,组件可比在典型系统中更高的频率和功率水平操作,由此增加性能。此外,滑板被配置成与机架中的功率和数据通信缆线盲配对,从而增强它们被快速移除、升级、重新安装和/或替换的能力。类似地,位于滑板上的各个组件(诸如处理器、加速器、存储器和数据存储驱动器)被配置成由于它们与彼此增加的间距而容易被升级。在说明性实施例中,组件另外包括硬件证明特征以证实其确实性(authenticity)。

[0188] 数据中心可利用单个网络架构(“组构”),所述单个网络架构支持包括以太网和全路径(Omni-Path)的多个其它网络架构。滑板可经由光纤被耦合到交换机,所述光纤提供比典型双绞线缆线(例如,类别5、类别5e、类别6等)更高的带宽和更低的时延。由于高带宽、低时延互连和网络架构,数据中心可使用在物理上解聚的池资源(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)以及数据存储驱动器),并且在按照需要的基础上将它们提供到计算资源(例如,处理器),使得计算资源能访问池化资源(pooled resource),如同池化资源是本地的那样。

[0189] 电力供应或电源可将电压和/或电流提供到本文描述的系统1400或任何组件或系统。在一个示例中,电力供应包括用于插入到壁装电源插座的AC到DC(交流到直流)适配器。此类AC电力可以是可再生能源(例如,太阳能)电源。在一个示例中,电源包括DC电源,诸如

外部AC到DC转换器。电源或电力供应还可包括无线充电硬件以经由接近充电场进行充电。电源可包括内部电池、交流供应、基于运动的电力供应、太阳能供应或燃料电池源。

[0190] 图15A-15C示出了计算系统和图形处理器。图15A-15C的与本文中任何其它图的元件具有相同或类似的名称的元件描述了与其它图中的元件相同的元件，可以以与其类似的方式操作或起作用，可包括相同的组件，并且可链接到其它实体，如本文中其它地方所述的那些那样，但不限于此。

[0191] 图15A是处理器1500的框图，其可以是处理器1402中的一个的变型并且可代替这些处理器之一使用。因此，本文中任何特征与处理器1500的组合的公开也公开了与(一个或多个)处理器1402的对应组合，但不限于此。处理器1500可具有一个或多个处理器核1502A-1502N、集成存储器控制器1514和集成图形处理器1508。在排除集成图形处理器1508的情况下，包括该处理器的系统将包括系统芯片集内或经由系统总线耦合的图形处理器装置。处理器1500可包括另外的核，所述另外的核多达并且包括由虚线框表示的另外的核1502N。处理器核1502A-1502N中的每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，每个处理器核1502A-1502N还可访问一个或多个共享高速缓存单元1506。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内的高速缓冲存储器层级。高速缓冲存储器层级可包括每个处理器核内的至少一级的指令和数据高速缓存，以及一级或多级的共享中间级高速缓存，诸如2级(L2)、3级(L3)、4级(L4)或其它级的高速缓存，其中在外部存储器前的最高级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种高速缓存单元1506与1504A-1504N之间的一致性。

[0192] 处理器1500还可包括一组一个或多个总线控制器单元1516和系统代理核1510。一个或多个总线控制器单元1516管理一组外设总线，诸如一个或多个PCI或PCI express总线。系统代理核1510提供用于各种处理器组件的管理功能性。系统代理核1510可包括一个或多个集成存储器控制器1514以管理对各种外部存储器装置(未示出)的访问。

[0193] 例如，处理器核1502A-1502N中的一个或多个处理器核可包括对同时多线程的支持。系统代理核1510包括用于在多线程的处理期间协调和操作核1502A-1502N的组件。系统代理核1510可另外包括功率控制单元(PCU)，所述功率控制单元(PCU)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。

[0194] 处理器1500可另外包括用于执行图形处理操作的图形处理器1508。在这些实施例中的一些中，图形处理器1508与一组共享高速缓存单元1506和系统代理核1510耦合，所述系统代理核1510包括一个或多个集成存储器控制器1514。系统代理核1510还可包括用于将图形处理器输出驱动到一个或多个耦合的显示器的显示控制器1511。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，或者可被集成在图形处理器1508内。

[0195] 基于环的互连单元1512可被用于耦合处理器1500的内部组件。然而，可使用备选互连单元，诸如点对点互连、交换互连或其它技术，其包括本领域中公知的技术。在具有基于环的互连1512的这些实施例中的一些中，图形处理器1508经由I/O链路1513与基于环的互连1512耦合。

[0196] 示例性I/O链路1513表示多个种类的I/O互连中的至少一个，其包括促进各种处理器组件与诸如eDRAM模块之类的高性能嵌入式存储器模块1518之间的通信的封装上I/O互

连。可选地,处理器核1502A-1502N中的每个和图形处理器1508可使用嵌入式存储器模块1518作为共享末级高速缓存。

[0197] 处理器核1502A-1502N可例如是执行相同指令集架构的同质核(homogenous core)。备选地,处理器核1502A-1502N在指令集架构(ISA)方面是异质的(heterogeneous),其中处理器核1502A-1502N中的一个或多个处理器核执行第一指令集,而其它核中的至少一个核执行第一指令集的子集或不同指令集。处理器核1502A-1502N可在微架构方面是异质的,其中具有相对更高功耗的一个或多个核与具有更低功耗的一个或多个功率核耦合。作为另一示例,处理器核1502A-1502N在计算能力方面是异质的。另外,处理器1500可在一个或多个芯片上被实现,或者被实现为除其它组件外还具有示出的组件的SoC集成电路。

[0198] 图15B是根据本文描述的一些实施例的图形处理器核1519的硬件逻辑的框图。有时被称为核切片(core slice)的图形处理器核1519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是一个图形核切片的示例,并且如本文描述的图形处理器可包括基于目标功率和性能包络(performance envelope)的多个图形核切片。每个图形处理器核1519可包括与也称为子切片的多个子核1521A-1521F耦合的固定功能块1530,所述多个子核1521A-1521F包括通用和固定功能逻辑的模块化块。

[0199] 固定功能块1530可包括几何/固定功能流水线1531,所述几何/固定功能流水线1531可例如在更低性能/或更低功率图形处理器实现中由图形处理器核1519中的所有子核共享。几何/固定功能流水线1531可包括3D固定功能流水线(例如,如下述图16A中的3D流水线1612)、视频前端单元、线程派生器(thread spawner)和线程分派器(thread dispatcher)以及管理统一返回缓冲器(unified return buffer)(例如,如下所述的在图17中的统一返回缓冲器1718)的统一返回缓冲器管理器。

[0200] 固定功能块1530还可包括图形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与片上系统集成电路内的其它处理器核之间的接口。图形微控制器1533是可编程子处理器,其可被配置成管理图形处理器核1519的各种功能,其包括线程分派、调度和抢占(pre-emption)。媒体流水线1534(例如,图16A和图17的媒体流水线1616)包括用于促进包括图像和视频数据的多媒体数据的解码、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。

[0201] SoC接口1532可使得图形处理器核1519能够与通用应用处理器核(例如,CPU)和/或SoC内的其它组件通信,所述SoC内的其它组件包括诸如共享末级高速缓冲存储器、系统RAM和/或嵌入式片上或封装上DRAM之类的存储器层级元件。SoC接口1532还可能够实现与SoC内的固定功能装置(诸如,相机成像流水线)的通信,并且能够实现全局存储器原子的使用和/或实现全局存储器原子,所述全局存储器原子可在图形处理器核1519与SoC内的CPU之间被共享。SoC接口1532还可用于图形处理器核1519的功率管理控制,并且能够实现图形核1519的时钟域与SoC内的其它时钟域之间的接口。可选地,SoC接口1532能够实现接收来自命令流播器和全局线程分派器的命令缓冲器(command buffer),所述命令流播器和全局线程分派器被配置成向图形处理器内的一个或多个图形核中的每个提供命令和指令。命令和指令可在要执行媒体操作时被分派到媒体流水线1534,或者在要执行图形处理操作时被分派到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)。

[0202] 图形微控制器1533可被配置成执行用于图形处理器核1519的各种调度和管理任务。在一个配置中,图形微控制器1533可例如在子核1521A-1521F内的执行单元(EU)阵列1522A-1522F、1524A-1524F内的各种图形并行引擎上执行图形和/或计算工作负载调度。在此工作负载调度中,在包括图形处理器核1519的SoC的CPU核上执行的主机软件可将工作负载提交给多个图形处理器门铃(graphic processor doorbell)中的一个,这调用在适当图形引擎上的调度操作。调度操作包括确定接下来要运行哪个工作负载、向命令流播器提交工作负载、对在引擎上运行的现有工作负载进行抢占、监测工作负载的进展、以及在工作负载完成时通知主机软件。可选地,图形微控制器1533还可促进用于图形处理器核1519的低功率或空闲状态,从而为图形处理器核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对图形处理器核1519内的寄存器进行保存和恢复的能力。

[0203] 图形处理器核1519可具有多于或少于示出的子核1521A-1521F,多达N个的模块化子核。对于N个子核的每个集合,图形处理器核1519还可包括共享功能逻辑1535、共享和/或高速缓冲存储器1536、几何/固定功能流水线1537以及另外的固定功能逻辑1538以加速各种图形和计算处理操作。共享功能逻辑1535可包括与图17的共享功能逻辑1720相关联的逻辑单元(例如,采样器、数学和/或线程间通信逻辑),所述逻辑单元可由图形处理器核1519内的每N个子核共享。共享和/或高速缓冲存储器1536可以是用于图形处理器核1519内的N个子核1521A-1521F的集合的末级高速缓存,并且还可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可代替固定功能块1530内的几何/固定功能流水线1531而被包括并且可包括相同或类似的逻辑单元。

[0204] 图形处理器核1519可包括另外的固定功能逻辑1538,其可包括供图形处理器核1519使用的各种固定功能加速逻辑。可选地,另外的固定功能逻辑1538包括供在仅位置着色中使用的另外的几何流水线。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;以及拣选流水线(cull pipeline),其是可被包括在另外的固定功能逻辑1538内的另外的几何流水线。例如,拣选流水线可以是完全几何流水线的裁减版本(trimmed down version)。完全流水线和拣选流水线可执行相同应用的不同实例,每个实例具有单独的上下文。仅位置着色可隐藏被丢弃三角形的长拣选运行,使得在一些实例中能更早完成着色。例如,另外的固定功能逻辑1538内的拣选流水线逻辑可执行与主应用并行的位置着色器,并且一般比完全流水线更快生成关键结果,因为拣选流水线仅对顶点的位置属性进行提取并着色,而不向帧缓冲器执行像素的栅格化和渲染。拣选流水线可使用生成的关键结果来计算用于所有三角形的可见性信息,而不考虑那些三角形是否被拣选。完全流水线(其在此实例中可被称为重放流水线)可消耗可见性信息以跳过被拣选的三角形,以仅对最终被传递到栅格化阶段的可见三角形进行着色。

[0205] 可选地,另外的固定功能逻辑1538还可包括诸如固定功能矩阵乘法逻辑之类的机器学习加速逻辑,以用于包括针对机器学习训练或推理的优化的实现。

[0206] 在每个图形子核1521A-1521F内包括一组执行资源,其可被用于响应于图形流水线、媒体流水线或着色器程序的请求而执行图形、媒体和计算操作。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-1525F、媒体采样器1506A-1506F、着色器处理器1527A-

1527F及共享本地存储器(SLM)1528A-1528F。EU阵列1522A-1522F、1524A-1524F各自包括多个执行单元，所述多个执行单元是能够为图形、媒体或计算操作(包括图形、媒体或计算着色器程序)服务而执行浮点和整数/固定点逻辑运算的通用图形处理单元。TD/IC逻辑1523A-1523F执行用于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可将纹理或其它3D图形有关数据读取到存储器中。3D采样器可基于配置的样本状态和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。例如，每个图形子核1521A-1521F可交替包括统一3D和媒体采样器。在子核1521A-1521F中的每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本地存储器1528A-1528F，以使得在线程群组内执行的线程能使用片上存储器的公共池来执行。

[0207] 图15C是根据本文描述的实施例的可被配置为图形处理器(例如，图形处理器1508)和/或计算加速器的通用图形处理单元(GPGPU)1570的框图。GPGPU 1570可经由一个或多个系统和/或存储器总线与主机处理器(例如，一个或多个CPU 1546)和存储器1571、1572互连。存储器1571可以是可与一个或多个CPU 1546共享的系统存储器，而存储器1572是专用于GPGPU 1570的装置存储器。例如，装置存储器1572和GPGPU 1570内的组件可被映射到一个或多个CPU 1546可访问的存储器地址中。可经由存储器控制器1568来促进对存储器1571和1572的访问。存储器控制器1568可包括内部直接存储器存取(DMA)控制器1569，或者可包括逻辑以执行在其它情况下将由DMA控制器执行的操作。

[0208] GPGPU 1570包括多个高速缓冲存储器，其包括L2高速缓存1553、L1高速缓存1554、指令高速缓存1555和共享存储器1556，该共享存储器1556的至少一部分也可被分区为高速缓冲存储器。GPGPU 1570还包括多个计算单元1560A-1560N。每个计算单元1560A-1560N包括一组向量寄存器1561、标量寄存器1562、向量逻辑单元1563和标量逻辑单元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可被用于存储常量数据，所述常量数据是在GPGPU 1570上执行的内核或着色器程序的运行期间将不改变的数据。常量高速缓存1567可以是标量数据高速缓存，并且经高速缓存的(cached)数据可被直接提取到标量寄存器1562中。

[0209] 在操作期间，一个或多个CPU 1546可将命令写入已被映射到可访问地址空间中的GPGPU 1570中的寄存器或存储器中。命令处理器1557可从寄存器或存储器读取命令，并且确定将在GPGPU 1570内如何处理那些命令。然后线程分派器1558可被用于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可独立于其它计算单元执行线程。另外，每个计算单元1560A-1560N可被独立配置用于有条件的计算，并且可有条件地将计算的结果输出到存储器。在提交的命令完成时，命令处理器1557可中断一个或多个CPU 1546。

[0210] 图16A-16C例如根据图15A-15C示出了由本文描述的实施例提供的另外的图形处理器和计算加速器架构的框图。图16A-16C的与本文中任何其它图的元件具有相同或类似的名称的元件描述与其它图中的元件相同的元件，可以以与其类似的方式操作或起作用，可包括相同组件，并且可链接到其它实体，如本文中其它地方所描述的那些那样，但不限于

此。

[0211] 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，或者可以是与多个处理核集成的图形处理器，或诸如但不限于存储器装置或网络接口的其它半导体装置。图形处理器1600可以是图形处理器1508的变型并且可代替图形处理器1508使用。因此，本文中任何特征与图形处理器1508的组合的公开也公开了与图形处理器1600的对应组合，但不限于此。图形处理器可经由到图形处理器上的寄存器的存储器映射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。图形处理器1600可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、一个或多个共享外部高速缓存和/或到系统存储器的接口。

[0212] 可选地，图形处理器1600还包括用于将显示输出数据驱动到显示装置1618的显示控制器1602。显示控制器1602包括用于一个或多个覆盖平面的硬件，以用于显示和组合用户接口元素或视频的多个层。显示装置1618可以是内部或外部显示装置。在一个实施例中，显示装置1618是头戴式显示装置，诸如虚拟现实(VR)显示装置或增强现实(AR)显示装置。图形处理器1600可包括视频编解码器引擎1606以将媒体编码成一个或多个媒体编码格式、从一个或多个媒体编码格式将媒体解码、或者在一个或多个媒体编码格式之间对媒体进行转码，所述编码格式包括但不限于运动图像专家组(MPEG)格式(诸如，MPEG-2)、高级视频编码(AVC)格式(诸如，H.264/MPEG-4 AVC、H.265/HEVC)、开放媒体联盟(AOMedia)VP8、VP9以及电影与电视工程师协会(SMPTE)421M/VC-1和联合图像专家组(JPEG)格式，诸如，JPEG、和运动JPEG(MJPEG)格式。

[0213] 图形处理器1600可包括用于执行二维(2D)栅格化器操作(包括例如位边界块传送)的块图像传送(BLIT)引擎1604。然而，备选地，可使用图形处理引擎(GPE)1610的一个或多个组件来执行2D图形操作。在一些实施例中，GPE 1610是用于执行包括三维(3D)图形操作和媒体操作的图形操作的计算引擎。

[0214] GPE 1610可包括用于执行3D操作的3D流水线1612，所述3D操作诸如使用作用于3D图元形状(例如，矩形、三角形等)的处理功能来渲染三维图像和场景。3D流水线1612包括可编程和固定功能元件，所述可编程和固定功能元件在元件内执行各种任务和/或派生(spawn)到3D/媒体子系统1615的执行线程。虽然3D流水线1612可被用于执行媒体操作，但GPE 1610的实施例还包括特别用于执行媒体操作(诸如，视频后处理和图像增强)的媒体流水线1616。

[0215] 媒体流水线1616可包括固定功能或可编程逻辑单元以代替或者代表视频编解码器引擎1606来执行一个或多个专用媒体操作，诸如视频解码加速、视频去交织和视频编码加速。媒体流水线1616另外可包括线程派生单元以派生用于在3D/媒体子系统1615上执行的线程。派生的线程在3D/媒体子系统1615中包括的一个或多个图形执行单元上执行用于媒体操作的计算。

[0216] 3D/媒体子系统1615可包括用于执行由3D流水线1612和媒体流水线1616派生的线程的逻辑。流水线可将线程执行请求发送到3D/媒体子系统1615，所述3D/媒体子系统1615包括线程分派逻辑，所述线程分派逻辑可用于将各种请求仲裁(arbitrate)并分派到可用线程执行资源。执行资源包括用于处理3D和媒体线程的图形执行单元的阵列。3D/媒体子系统1615可包括用于线程指令和数据的一个或多个内部高速缓存。附加地，3D/媒体子系统

1615还可包括共享存储器，所述共享存储器包括寄存器和可寻址存储器，以在线程之间共享数据并且存储输出数据。

[0217] 图16B示出了图形处理器1620，其是图形处理器1600的变型并且可代替图形处理器1600使用，并且反之亦然。因此，本文中任何特征与图形处理器1600的组合的公开也公开了与图形处理器1620的对应组合，但不限于此。图形处理器1620根据本文描述的实施例具有拼贴(tiled)架构。图形处理器1620可包括图形处理引擎集群1622，所述图形处理引擎集群1622在图形引擎贴片1610A-1610D内具有图16A的图形处理引擎1610的多个实例。每个图形引擎贴片1610A-1610D可经由一组贴片互连1623A-1623F而被互连。每个图形引擎贴片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装置1626A-1626D。存储器装置1626A-1626D可使用任何图形存储器技术。例如，存储器装置1626A-1626D可以是图形双倍数据率(GDDR)存储器。存储器装置1626A-1626D可以是高带宽存储器(HBM)模块，其可与其相应图形引擎贴片1610A-1610D一起在管芯上。存储器装置1626A-1626D可以是是堆叠存储器装置，其可被堆叠在其相应图形引擎贴片1610A-1610D之上。如图24B-24D中进一步详细描述的，每个图形引擎贴片1610A-1610D和相关联存储器1626A-1626D可驻留在单独的小芯片上，所述小芯片被接合到基础管芯或基础衬底。

[0218] 图形处理器1620可配置有非统一存储器访问(NUMA)系统，其中存储器装置1626A-1626D与相关联的图形引擎贴片1610A-1610D耦合。给定的存储器装置可能会被图形引擎贴片而不是直接与其连接的贴片访问。然而，当访问本地贴片时，对存储器装置1626A-1626D的访问时延可以是最低的。在一个实施例中，启用了高速缓存一致性NUMA(ccNUMA)系统，该系统使用贴片互连1623A-1623F来启用图形引擎贴片1610A-1610D内的高速缓存控制器之间的通信，以在多于一个高速缓存存储相同存储器位置时保持一致的存储器图像。

[0219] 图形处理引擎集群1622可与片上或封装上组构互连1624连接。组构互连1624可能够实现在图形引擎贴片1610A-1610D与诸如视频编解码器引擎1606和一个或多个复制引擎1604的组件之间的通信。复制引擎1604可被用于将数据移出以下各项、将数据移入以下各项以及在以下各项之间移动数据：存储器装置1626A-1626D和在图形处理器1620外的存储器(例如，系统存储器)。组构互连1624还可被用于互连图形引擎贴片1610A-1610D。图形处理器1620可以可选地包括用于能够实现与外部显示装置1618的连接的显示控制器1602。图形处理器还可被配置为图形或计算加速器。在加速器配置中，可省略显示控制器1602和显示装置1618。

[0220] 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可能够实现在图形处理器1620、系统存储器和/或其它系统组件之间的通信。主机接口1628可例如是PCI express总线或另一类型的主机系统接口。

[0221] 图16C示出了根据本文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的图形处理器1620的架构类似性并且被优化用于计算加速。计算引擎集群1632可包括计算引擎贴片1640A-1640D的集合，其包括被优化用于并行或基于向量的通用计算操作的执行逻辑。计算引擎贴片1640A-1640D可不包括固定功能图形处理逻辑，虽然在一些实施例中，计算引擎贴片1640A-1640D中的一个或多个计算引擎贴片可包括用于执行媒体加速的逻辑。计算引擎贴片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。图形计算引擎贴片1640A-1640D还可经由贴片互连1623A-1623F的集合被互连，并且可与组构互连1624连接和/或通过组构互连1624被互连。在一个实施例中，计算加速器1630包括可被配置为装置范围高速缓存的大的L3高速缓存1636。计算加速器1630还可以以与图16B的图形处理器1620类似的方式经由主机接口1628连接到主机处理器和存储器。

[0222] 图形处理引擎

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。图形处理引擎(GPE)1710可以是图16A中示出的GPE 1610的某一版本，并且还可表示图16B的图形引擎贴片1610A-1610D。图17的与本文中任何其它图的元件具有相同或类似的名称的元件描述了与其它图中的元件相同的元件，可以以与其类似的方式操作或起作用，可包括相同的组件，并且可链接到其它实体，如本文中其它地方所述的那些那样，但不限于此。例如，在图17中还示出了图16A的3D流水线1612和媒体流水线1616。媒体流水线1616在GPE 1710的一些实施例中是可选的，并且可未被显式地包括在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到GPE 1710。

[0223] GPE 1710可与命令流播器1703耦合或者包括该命令流播器1703，该命令流播器1703将命令流提供到3D流水线1612和/或媒体流水线1616。备选地或附加地，命令流播器1703可直接耦合到统一返回缓冲器1718。统一返回缓冲器1718可通信地耦合到图形核阵列1714。可选地，命令流播器1703与存储器耦合，所述存储器可以是系统存储器，或内部高速缓冲存储器和共享高速缓冲存储器中的一个或多个。命令流播器1703可接收来自存储器的命令，并且将命令发送到3D流水线1612和/或媒体流水线1616。命令是从存储用于3D流水线1612和媒体流水线1616的命令的环形缓冲器提取的指令(directive)。环形缓冲器可另外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的命令还可包括对在存储器中存储的数据(诸如但不限于用于3D流水线1612的顶点和几何数据和/或用于媒体流水线1616的图像数据和存储器对象)的引用。3D流水线1612和媒体流水线1616通过经由相应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到图形核阵列1714来处理命令和数据。图形核阵列1714可包括图形核(例如，(一个或多个)图形核1715A、(一个或多个)图形核1715B)的一个或多个块，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，其包括用于执行图形和计算操作的通用和图形特定执行逻辑；以及固定功能纹理处理和/或机器学习和人工智能加速逻辑。

[0224] 在各种实施例中，3D流水线1612可包括用于通过处理指令并且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(诸如，顶点着色器、几何着色器、像素着色器、片段着色器、计算着色器或其它着色器程序)的固定功能和可编程逻辑。图形核阵列1714提供执行资源的统一块以供在处理这些着色器程序中使用。图形核阵列1714的(一个或多个)图形核1715A-1715B内的多用途执行逻辑(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时执行线程。

[0225] 图形核阵列1714可包括用于执行媒体功能(诸如，视频和/或图像处理)的执行逻辑。执行单元可包括通用逻辑，该通用逻辑可编程以除图形处理操作外还执行并行通用计算操作。该通用逻辑可与在图14的(一个或多个)处理器核1407或如图15A中的核1502A-1502N内的通用逻辑并行或结合地执行处理操作。

[0226] 由在图形核阵列1714上执行的线程生成的输出数据可将数据输出到在统一返回缓冲器(URB)1718中的存储器。URB 1718可存储用于多个线程的数据。URB 1718可被用于在图形核阵列1714上执行的不同线程之间发送数据。URB 1718可另外被用于在图形核阵列上的线程与在共享功能逻辑1720内的固定功能逻辑之间的同步。

[0227] 可选地,图形核阵列1714可以是可缩放的,使得该阵列包括可变数量的图形核,这些图形核各自基于GPE 1710的目标功率和性能水平而具有可变数量的执行单元。执行资源可以是动态可缩放的,使得可按照需要启用或禁用执行资源。

[0228] 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共享的多个资源。在共享功能逻辑1720内的共享功能是硬件逻辑单元,所述硬件逻辑单元将专用补充功能性提供到图形核阵列1714。在各种实施例中,共享功能逻辑1720包括但不限于采样器1721、数学1722和线程间通信(ITC)1723逻辑。另外,可实现在共享功能逻辑1720内的一个或多个高速缓存1725。

[0229] 至少在其中对给定专用功能的需求不足以包括在图形核阵列1714内的情况下,实现共享功能。相反,该专用功能的单个实例化被实现为在共享功能逻辑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之间被共享并且包括在图形核阵列1714内的功能的精确集合跨实施例而变化。由图形核阵列1714广泛使用的共享功能逻辑1720内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。可选地,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内的一些或全部逻辑。可在图形核阵列1714的共享功能逻辑1716内重复共享功能逻辑1720内的全部逻辑元件。备选地,排除了共享功能逻辑1720以有利于图形核阵列1714内的共享功能逻辑1716。

[0230] 执行单元

图18A-18B示出了根据本文描述的实施例的线程执行逻辑1800,其包括在图形处理器核中采用的处理元件的阵列。图18A-18B的与本文中任何其它图的元件具有相同或类似的名称的元件描述了与其它图中的元件相同的元件,可以以与其类似的方式操作或起作用,可包括相同的组件,并且可链接到其它实体,如本文中其它地方所述的那些那样,但不限于此。图18A-18B示出了线程执行逻辑1800的概述,该线程执行逻辑1800可代表利用图15B的每个子核1521A-1521F示出的硬件逻辑。图18A代表通用图形处理器内的执行单元,而图18B代表可在计算加速器内使用的执行单元。

[0231] 如在图18A中所示出的,线程执行逻辑1800可包括着色器处理器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和数据端口1814。可选地,可缩放执行单元阵列可通过基于工作负载的计算要求来启用或禁用一个或多个执行单元(例如,执行单元1808A、1808B、1808C、1808D到1808N-1和1808N中的任何执行单元)来动态地进行缩放。被包括的组件可经由链接到组件中的每个组件的互连组构被互连。线程执行逻辑1800可包括通过指令高速缓存1806、数据端口1814、采样器1810和执行单元1808A-1808N中的一个或多个到存储器(诸如,系统存储器或高速缓冲存储器)的一个或多个连接。每个执行单元(例如,1808A)可以是独立可编程通用计算单元,其能执行多个同时硬件线程,同时为每个线程并行处理多个数据元素。在各种实施例中,执行单元1808A-1808N的阵列是可缩放的,以包括任何数量的各个执行单元。

[0232] 执行单元1808A-1808N可主要被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，并且经由线程分派器1804分派与着色器程序相关联的执行线程。线程分派器可包括用于对来自图形和媒体流水线的线程发起请求进行仲裁并且在一个或多个执行单元1808A-1808N上实例化所请求的线程的逻辑。例如，几何流水线可将顶点、曲面细分或几何着色器分派到线程执行逻辑以用于处理。可选地，线程分派器1804还可处理来自在执行的着色器程序的运行时间线程派生请求。

[0233] 执行单元1808A-1808N可支持指令集，所述指令集包括对许多标准3D图形着色器指令的本机支持，使得来自图形库(例如，Direct 3D和OpenGL)的着色器程序以最小的转化被执行。执行单元支持顶点和几何处理(例如，顶点程序、几何程序、顶点着色器)、像素处理(例如，像素着色器、片段着色器)和通用处理(例如，计算和媒体着色器)。执行单元1808A-1808N中的每个执行单元能进行多发布(multi-issue)单指令多数据(SIMD)执行，并且多线程操作在面临更高时延存储器访问时能够实现高效的执行环境。每个执行单元内的每个硬件线程具有专用高带宽寄存器堆和相关联的独立线程状态。执行是对能进行整数、单精度和双精度浮点运算、SIMD分支能力、逻辑运算、超越运算和其它杂项运算的流水线的每时钟多发布。在等待来自存储器或共享功能之一的数据时，执行单元1808A-1808N内的依赖性逻辑促使在等待的线程进行休眠，直到所请求的数据已被返回为止。当在等待的线程正在休眠时，硬件资源可专用于处理其它线程。例如，在与顶点着色器操作相关联的延迟期间，执行单元可执行用于像素着色器、片段着色器或另一类型的着色器程序(包括不同顶点着色器，诸如图21中所示的顶点着色器2107)的操作。各种实施例可应用于：通过作为使用SIMD的备选方案或者除使用SIMD之外还使用单指令多线程(SIMT)来使用执行。对SIMD核或操作的引用也可应用于SIMT或者应用于与SIMT组合的SIMD。

[0234] 执行单元1808A-1808N中的每个执行单元对数据元素的阵列进行操作。数据元素的数量是“执行大小”或指令的通道的数量。执行通道是用于指令内的数据元素访问、掩蔽(masking)和流控制的执行的逻辑单元。通道的数量可独立于特定图形处理器的物理算术逻辑单元(ALU)、浮点单元(FPU)或其它逻辑单元(例如，张量核、光线追踪核等)的数量。附加地，执行单元1808A-1808N可支持整数和浮点数据类型。

[0235] 执行单元指令集包括SIMD指令。各种数据元素可作为打包数据类型存储在寄存器中，并且执行单元将基于元素的数据大小来处理各种元素。例如，在对256位宽向量进行操作时，向量的256位被存储在寄存器中，并且执行单元对作为四个单独的184位打包数据元素(四字(QW)大小数据元素)、八个单独的32位打包数据元素(双字(DW)大小数据元素)、十六个单独的16位打包数据元素(字(W)大小数据元素)或三十二个单独的8位数据元素(字节(B)大小数据元素)的向量进行操作。然而，不同向量宽度和寄存器大小是可能的。

[0236] 可选地，一个或多个执行单元可被组合成具有线程控制逻辑(1807A-1807N)的融合的执行单元1809A-1809N，所述线程控制逻辑(1807A-1807N)对于融合的EU是公共的。多个EU可被融合成EU群组。融合的EU群组中的每个EU可被配置成执行单独的SIMD硬件线程。融合的EU群组中EU的数量可根据实施例而变化。另外，可每EU执行各种SIMD宽度，包括但不限于SIMD8、SIMD16和SIMD32。每个融合的图形执行单元1809A-1809N包括至少两个执行单元。例如，融合的执行单元1809A包括第一EU 1808A、第二EU 1808B以及线程控制逻辑1807A，该线程控制逻辑1807A对第一EU 1808A和第二EU 1808B是公共的。线程控制逻辑

1807A控制在融合的图形执行单元1809A上执行的线程，允许融合的执行单元1809A-1809N内的每个EU使用公共指令指针寄存器来执行。

[0237] 线程执行逻辑1800中包括一个或多个内部指令高速缓存(例如，1806)以对用于执行单元的线程指令进行高速缓存。在线程执行逻辑1800中可包括一个或多个数据高速缓存(例如，1812)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上执行的线程还可在共享本地存储器1811中存储显式管理的数据。可包括采样器1810以提供3D操作的纹理采样和媒体操作的媒体采样。采样器1810可包括专用纹理或媒体采样功能性，以在向执行单元提供采样的数据前在采样过程期间处理纹理或媒体数据。

[0238] 在执行期间，图形和媒体流水线经由线程派生和分派逻辑向线程执行逻辑1800发送线程发起请求。一旦几何对象的群组已被处理并且栅格化成像素数据，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)便被调用于进一步计算输出信息，并且使结果被写入到输出表面(例如，颜色缓冲器、深度缓冲器、模板缓冲器等)。像素着色器或片段着色器可计算要跨栅格化对象内插的各种顶点属性的值。着色器处理器1802内的像素处理器逻辑然后可执行应用编程接口(API)供应的像素或片段着色器程序。为了执行着色器程序，着色器处理器1802经由线程分派器1804向执行单元(例如，1808A)分派线程。着色器处理器1802可使用采样器1810中的纹理采样逻辑来访问存储在存储器中的纹理映射中的纹理数据。对纹理数据和输入几何数据的算术运算为每个几何片段计算像素颜色数据，或者丢弃一个或多个像素而不进行进一步处理。

[0239] 此外，数据端口1814可提供存储器访问机制，以便线程执行逻辑1800将经处理的数据输出至存储器以用于在图形处理器输出流水线上进一步处理。数据端口1814可包括或者耦合到一个或多个高速缓冲存储器(例如，数据高速缓存1812)来对数据进行高速缓存以用于经由数据端口1814的存储器访问。

[0240] 可选地，执行逻辑1800还可包括光线追踪器1805，所述光线追踪器1805可提供光线追踪加速功能性。光线追踪器1805可支持包括用于光线生成的指令/功能的光线追踪指令集。该光线追踪指令集可与由图3C中的光线追踪核372支持的光线追踪指令集类似或不同。

[0241] 图18B示出了执行单元1808的示例性内部细节。图形执行单元1808可包括指令提取单元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)1834的集合以及可选地包括专用整数SIMD ALU 1835的集合。GRF 1824和ARF 1826包括与可在图形执行单元1808中是活动的每个同时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ARF 1826中可维持每线程架构状态，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)可被保持在ARF 1826中的线程特定寄存器中。

[0242] 图形执行单元1808可具有这样的架构，该架构是同时多线程(SMT)和细粒度交错多线程(IMT)的组合。该架构可具有模块化配置，可在设计时基于每执行单元的寄存器的数量和同时线程的目标数量来对所述模块化配置进行微调，其中执行单元资源跨用于执行多个同时线程的逻辑被划分。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，并且多个逻辑线程可被指派到每个硬件线程。

[0243] 可选地，图形执行单元1808可共同发布多个指令，所述多个指令各自可以是不同

的指令。图形执行单元线程1808的线程仲裁器1822可将指令分派给发送单元1830、分支单元1832或(一个或多个)SIMD FPU 1834中的一个以用于执行。每个执行线程可访问GRF 1824内的128个通用寄存器,其中每个寄存器可存储32个字节,其可作为32位数据元素的SIMD 8元素向量访问。每个执行单元线程可访问GRF 1824内的4千字节,虽然实施例不限于此,并且在其它实施例中可提供更多或更少的寄存器资源。图形执行单元1808可被分区成可独立执行计算操作的七个硬件线程,虽然每执行单元的线程的数量还可根据实施例而变化。例如,可支持多达16个硬件线程。在示例性实施例中,其中七个线程可访问4千字节,GRF 1824可存储总共28千字节。在另一示例性实施例中,在16个线程可访问4千字节的情况下,GRF 1824可存储总共64千字节。然而,每执行单元的线程的数量不限于这些示例并且可多于或少于给定的数量。灵活的寻址模式可允许对寄存器一起进行寻址以有效地构建更宽的寄存器或者表示跨步矩形块数据结构(strided rectangular block data structure)。

[0244] 附加地或备选地,可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、采样器操作和其它较长时延系统通信。可将分支指令分派给专用分支单元1832以促进SIMD发散和最终收敛。

[0245] 图形执行单元1808可包括一个或多个SIMD浮点单元((一个或多个)FPU)1834以执行浮点运算。(一个或多个)FPU 1834还可支持整数计算。在一些实例中,(一个或多个)FPU 1834可SIMD执行多达M个数量的32位浮点(或整数)运算,或者SIMD执行多达2M个16位整数或16位浮点运算。可选地,(一个或多个)FPU中的至少一个提供扩展的数学能力以支持高吞吐量超越数学函数和双精度184位浮点。还可存在8位整数SIMD ALU 1835的集合,并且8位整数SIMD ALU 1835的集合可被特别地优化以执行与机器学习计算相关联的操作。

[0246] 可选地,图形执行单元1808的多个实例的阵列可在图形子核分组(例如,子切片)中被实例化。为了可缩放性,产品架构师可选定每子核分组的执行单元的确切数量。执行单元1808可跨多个执行通道执行指令。此外,在图形执行单元1808上执行的每个线程可以是在不同的通道上被执行的。

[0247] 图19示出了另外的示例性执行单元1900。图19的与本文中任何其它图的元件具有相同或类似的名称的元件描述了与其它图中的元件相同的元件,可以以与其类似的方式操作或起作用,可包括相同的组件,并且可链接到其它实体,如本文中其它地方所述的那些那样,但不限于此。执行单元1900可以是供在例如如图16C中的计算引擎贴片1640A-1640D中使用的计算优化的执行单元,但不限于此。执行单元1900还可用于如图16B中的图形引擎贴片1610A-1610D中。执行单元1900可包括线程控制单元1901、线程状态单元1902、指令提取/预提取单元1903和指令解码单元1904。执行单元1900另外包括寄存器堆1906,所述寄存器堆1906存储可被指派到执行单元内的硬件线程的寄存器。执行单元1900另外包括发送单元1907和分支单元1908。发送单元1907和分支单元1908可与图18B的图形执行单元1808的发送单元1830和分支单元1832类似地操作。

[0248] 执行单元1900还包括计算单元1910,所述计算单元1910包括多个不同类型的功能单元。计算单元1910还可包括ALU单元1911,所述ALU单元1911包括算术逻辑单元的阵列。ALU单元1911可被配置成执行64位、32位和16位整数和浮点运算。整数和浮点运算可同时被执行。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包括可用于以脉动方式执行向量或其它数据并行操作的数据处理单元的宽度W和深度D的网络。脉动阵列

1912可被配置成执行矩阵运算，诸如矩阵点积运算。脉动阵列1912可支持16位浮点运算及8位和4位整数运算。脉动阵列1912可被配置成加速机器学习操作。脉动阵列1912可被配置有对bfloat16(16位浮点格式)的支持。可包括数学单元1913来以高效和比ALU单元1911更低功率的方式执行数学运算的特定子集。数学单元1913可包括在由描述的其它实施例提供的图形处理引擎的共享功能逻辑中找到的数学逻辑(例如，图17的共享功能逻辑1720的数学逻辑1722)。数学单元1913可被配置成执行32位和64位浮点运算。

[0249] 线程控制单元1901包括用于控制执行单元内线程的执行的逻辑。线程控制单元1901可包括用于开始、停止和抢占执行单元1900内线程的执行的线程仲裁逻辑。线程状态单元1902可用于为被指派在执行单元1900上执行的线程存储线程状态。在执行单元1900内存储线程状态能够实现在线程变为阻塞或空闲时那些线程的快速抢占。指令提取/预提取单元1903可从更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)提取指令。指令提取/预提取单元1903还可基于当前在执行的线程的分析来发布对要被加载到指令高速缓存中的指令的预提取请求。指令解码单元1904可被用于解码要由计算单元执行的指令。指令解码单元1904可被用作次级解码器以将复杂指令解码成组成的微操作(constituent micro-operation)。

[0250] 执行单元1900另外包括可由在执行单元1900上执行的硬件线程使用的寄存器堆1906。可跨用于执行在执行单元1900的计算单元1910内的多个同时线程的逻辑来划分寄存器堆1906中的寄存器。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，并且多个逻辑线程可被指派到每个硬件线程。寄存器堆1906的大小可基于支持的硬件线程的数量跨实施例而变化。寄存器重命名可被用于动态地将寄存器分配到硬件线程。

[0251] 图20是示出图形处理器指令格式2000的框图。图形处理器执行单元支持具有采用多种格式的指令的指令集。实线框示出了一般被包括在执行单元指令中的组成部分，而虚线包括可选的或者仅被包括在指令的子集中的组成部分。描述和示出的指令格式2000是宏指令，因为它们是供应给执行单元的指令，而与一旦指令被处理由指令解码产生的微操作相反。

[0252] 如本文描述的图形处理器执行单元可本机地支持采用128位指令格式2010的指令。基于所选择的指令、指令选项和操作数的数量，64位压缩指令格式2030可用于一些指令。本机128位指令格式2010提供对所有指令选项的访问，而采用64位格式2030，一些选项和操作被限制。采用64位格式2030的可用的本机指令随实施例而变化。使用索引字段2013中索引值的集合来部分地压缩指令。执行单元硬件基于索引值来引用压缩表的集合，并且使用压缩表输出来重构采用128位指令格式2010的本机指令。可使用指令的其它大小和格式。

[0253] 对于每种格式，指令操作码2012定义执行单元要执行的操作。执行单元跨每个操作数的多个数据元素并行执行每个指令。例如，响应于相加指令，执行单元跨表示纹理元素或图片元素的每个颜色通道来执行同时加法运算。默认情况下，执行单元跨操作数的所有数据通道来执行每个指令。指令控制字段2014可能够实现对诸如通道选择(例如，断定(predication))和数据通道次序(例如，搅混(swizzle))之类的某些执行选项的控制。对于采用128位指令格式2010的指令，执行大小字段2016限制将被并行执行的数据通道的数量。执行大小字段2016可能不可供64位压缩指令格式2030中使用。

[0254] 一些执行单元指令具有多达三个操作数，所述三个操作数包括两个源操作数src0 2020、src1 2022和一个目的地2018。执行单元可支持双目的地指令，其中目的地中的一个是隐含的。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。指令的最后源操作数可以是利用指令传递的立即(例如，硬编码的)值。

[0255] 128位指令格式2010可包括访问/地址模式字段2026，该访问/地址模式字段2026指定例如使用直接寄存器寻址模式还是间接寄存器寻址模式。在使用直接寄存器寻址模式时，由指令中的位来直接提供一个或多个操作数的寄存器地址。

[0256] 128位指令格式2010还可包括访问/地址模式字段2026，该访问/地址模式字段2026指定指令的地址模式和/或访问模式。访问模式可被用于定义指令的数据访问对齐。可支持包括16字节对齐的访问模式和1字节对齐的访问模式的访问模式，其中访问模式的字节对齐确定指令操作数的访问对齐。例如，当处于第一模式中时，指令可将字节对齐的寻址用于源操作数和目的地操作数，并且当处于第二模式中时，指令可将16字节对齐的寻址用于所有源操作数和目的地操作数。

[0257] 访问/地址模式字段2026的地址模式部分可确定指令要使用直接寻址还是间接寻址。在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。在使用间接寄存器寻址模式时，可基于指令中的地址立即字段和地址寄存器值来计算一个或多个操作数的寄存器地址。

[0258] 基于操作码2012位字段可对指令进行分组，以简化操作码解码2040。对于8位操作码，4、5和6位允许执行单元确定操作码的类型。所示出的精确操作码分组仅是示例。移动和逻辑操作码群组2042可包括数据移动和逻辑指令(例如，移动(mov)、比较(cmp))。移动和逻辑群组2042可共享五个最高有效位(MSB)，其中移动(mov)指令采用0000xxxxb的形式，并且逻辑指令采用0001xxxxb的形式。流控制指令群组2044(例如调用、跳(jmp))包括采用0010xxxxb(例如，0x20)形式的指令。杂项指令群组2046包括指令的混合，包括采用0011xxxxb(例如，0x30)形式的同步指令(例如等待、发送)。并行数学指令群组2048包括采用0100xxxxb(例如，0x40)的形式的逐组成部分的算术指令(例如，加、乘(mul))。并行数学群组2048跨数据通道并行执行算术运算。向量数学群组2050包括采用0101xxxxb(例如，0x50)形式的算术指令(例如，dp4)。向量数学群组对向量操作数执行诸如点积计算的算术。示出的操作码解码2040在一个实施例中可用于确定执行单元的哪个部分将被用于执行解码的指令。例如，一些指令可被指定为将由脉动阵列执行的脉动指令。诸如光线追踪指令(未示出)的其它指令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。

[0259] 图形流水线

图21是根据另一实施例的图形处理器2100的框图。图21的与本文中任何其它图的元件具有相同或类似的名称的元件描述了与其它图中的元件相同的元件，可以以与其类似的方式操作或起作用，可包括相同的组件，并且可链接到其它实体，如本文中其它地方所述的那些那样，但不限于此。

[0260] 图形处理器2100可包括不同类型的图形处理流水线，诸如几何流水线2120、媒体流水线2130、显示引擎2140、线程执行逻辑2150和渲染输出流水线2170。图形处理器2100可以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。可通过对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

令来控制图形处理器。环形互连2102可将图形处理器2100耦合到其它处理组件，诸如其它图形处理器或通用处理器。来自环形互连2102的命令由命令流播器2103解译，该命令流播器2103将指令供应至几何流水线2120或媒体流水线2130的各个组件。

[0261] 命令流播器2103可指导顶点提取器2105的操作，该顶点提取器2105从存储器读取顶点数据，并执行由命令流播器2103提供的顶点处理命令。顶点提取器2105可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。顶点提取器2105和顶点着色器2107可通过经由线程分派器2131向执行单元2152A-2152B分派执行线程来执行顶点处理指令。

[0262] 执行单元2152A-2152B可以是具有用于执行图形和媒体操作的指令集的向量处理器的阵列。执行单元2152A-2152B可具有附连的L1高速缓存2151，所述L1高速缓存2151对于每个阵列是特定的，或者在阵列之间被共享。高速缓存能被配置为数据高速缓存、指令高速缓存或被分区以在不同分区中包含数据和指令的单个高速缓存。

[0263] 几何流水线2120可包括曲面细分组件以执行3D对象的硬件加速的曲面细分。可编程外壳着色器(programmable hull shader)2111可配置曲面细分操作。可编程域着色器2117可提供曲面细分输出的后端评估。曲面细分器2113可在外壳着色器2111的指导下进行操作，并且包含专用逻辑以基于作为到几何流水线2120的输入而提供的粗略几何模型来生成详细的几何对象的集合。此外，如果未使用曲面细分，则可绕过曲面细分组件(例如，外壳着色器2111、曲面细分器2113和域着色器2117)。

[0264] 完整几何对象可由几何着色器2119经由分派给执行单元2152A-2152B的一个或多个线程来处理，或者可直接行进至裁剪器(clipper)2129。几何着色器可对整个几何对象进行操作，而不是如在图形流水线的先前阶段中对顶点或顶点的补片(patch)进行操作。如果曲面细分被禁用，则几何着色器2119接收来自顶点着色器2107的输入。几何着色器2119可由几何着色器程序可编程以在曲面细分单元被禁用时执行几何曲面细分。

[0265] 在栅格化前，裁剪器2129处理顶点数据。裁剪器2129可以是具有裁剪和几何着色器功能的可编程裁剪器或固定功能裁剪器。渲染输出流水线2170中的栅格化器和深度测试组件2173可分派像素着色器以将几何对象转换成逐像素表示。像素着色器逻辑可被包括在线程执行逻辑2150中。可选地，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。

[0266] 图形处理器2100具有允许数据和消息在处理器的主要组件之间传递的互连总线、互连组构或某种其它互连机构。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互连，以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。采样器2154、高速缓存2151、2158和执行单元2152A-2152B可各自具有单独的存储器访问路径。可选地，纹理高速缓存2158还可被配置为采样器高速缓存。

[0267] 渲染输出流水线2170可包含栅格化器和深度测试组件2173，该栅格化器和深度测试组件2173将基于顶点的对象转换成相关联的基于像素的表示。栅格化器逻辑可包括用于执行固定功能三角形和线栅格化的窗口化器(windower)/掩蔽器单元。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对数据执行基于像素的操作，尽管在一些实例中，与2D操作相关联的像素操作(例如，带有混合

(blending)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使用覆盖显示平面代替。共享L3高速缓存2175可能对于全部图形组件是可用的,从而允许在不使用主系统存储器的情况下共享数据。

[0268] 图形处理器媒体流水线2130可包括媒体引擎2137和视频前端2134。视频前端2134可接收来自命令流播器2103的流水线命令。媒体流水线2130可包括单独的命令流播器。视频前端2134可在将命令发送至媒体引擎2137之前处理媒体命令。媒体引擎2137可包括线程派生功能性来派生线程,以便经由线程分派器2131分派到线程执行逻辑2150。

[0269] 图形处理器2100可包括显示引擎2140。显示引擎2140可在处理器2100的外部,并且可经由环形互连2102或某一其它互连总线或组构与图形处理器耦合。显示引擎2140可包括2D引擎2141和显示控制器2143。显示引擎2140可包含能独立于3D流水线操作的专用逻辑。显示控制器2143可与显示装置(未示出)耦合,该显示装置可以是系统集成的显示装置(如在膝上型计算机中),或者可以是经由显示装置连接器附连的外部显示装置。

[0270] 几何流水线2120和媒体流水线2130可能可配置成基于多个图形和媒体编程接口执行操作,并且不特定于任何一个应用编程接口(API)。用于图形处理器的驱动器软件可将特定于特定图形或媒体库的API调用转化成可由图形处理器处理的命令。可为全部来自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。还可为来自微软公司的Direct3D库提供支持。可支持这些库的组合。还可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也将被支持。

[0271] 图形流水线编程

图22A是示出用于对图形处理流水线进行编程的图形处理器命令格式2200的框图,所述图形处理流水线诸如例如本文中结合图16A、17、21描述的流水线。图22B是示出根据实施例的图形处理器命令序列2210的框图。图22A中的实线框示出了一般被包括在图形命令中的组成部分,而虚线包括可选的或者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。一些命令中还包括子操作码2205和命令大小2208。

[0272] 客户端2202可指定处理命令数据的图形装置的客户端单元。图形处理器命令解析器可检查每个命令的客户端字段以调节命令的进一步处理,并且将命令数据路由到适当的客户端单元。图形处理器客户端单元可包括存储器接口单元、渲染单元、2D单元、3D单元和媒体单元。每个客户端单元可具有处理命令的对应处理流水线。一旦由客户端单元接收到命令,客户端单元便读取操作码2204和子操作码2205(如果子操作码2205存在的话),以确定要执行的操作。客户端单元使用数据字段2206中的信息来执行命令。对于一些命令,预期显式命令大小2208来指定命令的大小。命令解析器可基于命令操作码来自动确定至少一些命令的大小。可经由双字的倍数来对齐命令。还可使用其它命令格式。

[0273] 图22B中的流程示出了示例性图形处理器命令序列2210。以示例性图形处理器为特征的数据处理系统的软件或固件可使用所示出的命令序列的版本来设定、执行和终止图形操作的集合。仅出于示例的目的示出并描述了样本命令序列,并且样本命令序列不限于这些特定命令或此命令序列。另外,命令可作为命令序列中的批量命令被发布,使得图形处理器将至少部分并发地处理命令的序列。

[0274] 图形处理器命令序列2210可以以流水线转储清除命令2212开始，以促使任何活动的图形流水线完成该流水线的当前未决命令。可选地，3D流水线2222和媒体流水线2224可不并发地操作。执行流水线转储清除以促使活动的图形流水线完成任何未决命令。响应于流水线转储清除，图形处理器的命令解析器将暂停命令处理，直至活动的绘图引擎完成未决操作并且相关的读取高速缓存失效。可选地，可将渲染高速缓存中标记为“脏”的任何数据转储清除到存储器。流水线转储清除命令2212可被用于流水线同步，或者在将图形处理器置于低功率状态前被使用。

[0275] 在命令序列要求图形处理器在流水线之间显式地切换时，可使用流水线选择命令2213。除非上下文要为两个流水线发布命令，否则在发布流水线命令前，可在执行上下文内仅要求一次流水线选择命令2213。紧接经由流水线选择命令2213的流水线切换之前，可要求流水线转储清除命令2212。

[0276] 流水线控制命令2214可配置图形流水线以用于操作，并且可被用于对3D流水线2222和媒体流水线2224进行编程。流水线控制命令2214可配置活动的流水线的流水线状态。流水线控制命令2214可被用于流水线同步，并且在处理一批命令前从活动的流水线内的一个或多个高速缓冲存储器中清除数据。

[0277] 返回缓冲器状态命令2216可被用于为相应流水线配置返回缓冲器的集合以写入数据。一些流水线操作要求分配、选择或配置一个或多个返回缓冲器，在处理期间这些操作将中间数据写入到所述一个或多个返回缓冲器。图形处理器还可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。返回缓冲器状态2216可包括选择要用于流水线操作的集合的返回缓冲器的大小和数量。

[0278] 命令序列中的剩余命令基于用于操作的活动流水线而不同。基于流水线确定2220，将命令序列定制到以3D流水线状态2230开始的3D流水线2222或者从媒体流水线状态2240开始的媒体流水线2224。

[0279] 用于配置3D流水线状态2230的命令包括3D状态设置命令，所述3D状态设置命令用于在处理3D图元命令之前要配置的顶点缓冲器状态、顶点元素状态、恒定颜色状态、深度缓冲器状态以及其它状态变量。至少部分基于使用中的特定3D API来确定这些命令的值。如果将不使用某些流水线元件，则3D流水线状态2230命令还可能能够选择性地禁用或绕过那些元件。

[0280] 3D图元2232命令可被用于提交要由3D流水线处理的3D图元。经由3D图元2232命令传递到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提取功能。顶点提取功能使用3D图元2232命令数据来生成顶点数据结构。顶点数据结构被存储在一个或多个返回缓冲器中。3D图元2232命令可被用于经由顶点着色器对3D图元执行顶点操作。为处理顶点着色器，3D流水线2222将着色器执行线程分派到图形处理器执行单元。

[0281] 可经由执行2234命令或事件来触发3D流水线2222。寄存器可写入触发命令执行。可经由命令序列中的“go”或“kick”命令来触发执行。可使用流水线同步命令来触发命令执行以转储清除通过图形流水线的命令序列。3D流水线将执行3D图元的几何处理。一旦操作完成，所得到的几何对象便被栅格化，并且像素引擎对所得到的像素进行上色。对于那些操作，还可包括用于控制像素着色和像素后端操作的另外的命令。

[0282] 在执行媒体操作时，图形处理器命令序列2210可沿着媒体流水线2224路径。一般

而言，用于媒体流水线2224的编程的特定使用和方式取决于要执行的媒体或计算操作。可在媒体解码期间将特定媒体解码操作卸载到媒体流水线。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全部或部分地执行媒体解码。媒体流水线还可包括用于通用图形处理器单元(GPGPU)操作的元件，其中图形处理器被用于使用计算着色器程序来执行SIMD向量运算，所述计算着色器程序与图形图元的渲染不是显式相关的。

[0283] 以与3D流水线2222类似的方式可对媒体流水线2224进行配置。将用于配置媒体流水线状态2240的命令的集合分派或放置到在媒体对象命令2242之前的命令队列中。用于媒体流水线状态2240的命令可包括用于配置媒体流水线元件的数据，所述媒体流水线元件将被用于处理媒体对象。这包括用于配置媒体流水线内的视频解码和视频编码逻辑的数据，诸如编码和解码格式。用于媒体流水线状态2240的命令还可支持使用到包含一批状态设置的“间接”状态元素的一个或多个指针。

[0284] 媒体对象命令2242可将指针供应到媒体对象以便由媒体流水线处理。媒体对象包括存储器缓冲器，所述存储器缓冲器包含要处理的视频数据。可选地，在发布媒体对象命令2242之前，所有媒体流水线状态必须是有效的。一旦配置了流水线状态，并且将媒体对象命令2242排队，便经由执行命令2244或等效执行事件(例如，寄存器写入)来触发媒体流水线2224。然后可通过由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。可以以与媒体操作类似的方式配置和执行GPGPU操作。

[0285] 图形软件架构

图23示出了用于数据处理系统2300的示例性图形软件架构。这样的软件架构可包括3D图形应用2310、操作系统2320和至少一个处理器2330。处理器2330可包括图形处理器2332和一个或多个通用处理器核2334。处理器2330可以是处理器1402或本文中所述处理器中的任何其它处理器的变型。处理器2330可代替处理器1402或本文中所述处理器中的任何其它处理器使用。因此，任何特征与处理器1402或本文中所述处理器中的任何其它处理器的组合的公开也公开了与图形处理器2330的对应组合，但不限于此。此外，图23的与本文中任何其它图的元件具有相同或类似的名称的元件描述了与其它图中的元件相同的元件，可以以与其类似的方式操作或起作用，可包括相同的组件，并且可链接到其它实体，如本文中其它地方所述的那些那样，但不限于此。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。

[0286] 3D图形应用2310可包含一个或多个着色器程序，该一个或多个着色器程序包括着色器指令2312。着色器语言指令可采用高级着色器语言，诸如Direct3D的高级着色器语言(HLSL)或OpenGL着色器语言(GLSL)等等。应用还可包括采用适合由通用处理器核2334执行的机器语言的可执行指令2314。应用还可包括由顶点数据定义的图形对象2316。

[0287] 操作系统2320可以是来自微软公司的Microsoft® Windows®操作系统、专有的类UNIX操作系统或使用Linux内核的变型的开源类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。Direct3D API在使用中时，操作系统2320使用前端着色器编译器2324来将采用HLSL的任何着色器指令2312编译成更低级着色器语言。编译可以是即时(JIT)编译或者应用可执行着色器预编译。在3D图形应用2310的编译期间可将高级着色器编译成低级着色器。可以以中间形式(诸如，由Vulkan API使用的标准可移植中间表示(SPIR)的版本)提供着色器指令2312。

[0288] 用户模式图形驱动器2326可包含用于将着色器指令2312转换成硬件特定表示的后端着色器编译器2327。OpenGL API在使用中时，将采用GLSL高级语言的着色器指令2312传递到用户模式图形驱动器2326以用于编译。用户模式图形驱动器2326可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通信。内核模式图形驱动器2329可与图形处理器2332通信以分派命令和指令。

[0289] IP核实现

一个或多个方面可由存储在机器可读介质上的代表性代码来实现，该代表性代码表示和/或定义诸如处理器的集成电路内的逻辑。例如，机器可读介质可包括表示处理器内的各种逻辑的指令。在由机器读取时，指令可促使机器制作逻辑以执行本文描述的技术。称为“IP核”的此类表示是用于集成电路的逻辑的可重复使用单元，该可重复使用单元可作为对集成电路的结构进行描述的硬件模型而被存储在有形机器可读介质上。可将硬件模型供应至各种客户或制造设施，所述客户或制造设施将硬件模型加载在制造集成电路的制作机器上。可制作集成电路，使得电路执行与本文描述的实施例中的任何实施例相关联的所描述的操作。

[0290] 图24A是示出根据实施例的可被用于制造集成电路以执行操作的IP核开发系统2400的框图。IP核开发系统2400可被用于生成可被并入到更大的设计中或被用于构造完整集成电路(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可被用于使用仿真模型2412来设计、测试和验证IP核的行为。仿真模型2412可包括功能、行为和/或时序仿真。然后可从仿真模型2412创建或合成寄存器传送级(RTL)设计2415。RTL设计2415是对硬件寄存器之间的数字信号流进行建模的集成电路的行为的抽象，包括使用建模的数字信号执行的相关联的逻辑。除RTL设计2415外，还可创建、设计或合成处于逻辑级或晶体管级的较低级设计。因此，初始设计和仿真的特定细节可变化。

[0291] 可由设计设施将RTL设计2415或等效物进一步合成为硬件模型2420，该硬件模型2420可采用硬件描述语言(HDL)或物理设计数据的某种其它表示。可进一步对HDL进行仿真或测试以验证IP核设计。可使用非易失性存储器2440(例如，硬盘、闪速存储器或任何非易失性存储介质)来存储IP核设计以用于递送到第三方制作设施2465。备选的是，可通过有线连接2450或无线连接2460(例如，经由因特网)来传送IP核设计。制作设施2465然后可制作至少部分基于IP核设计的集成电路。制作的集成电路可被配置成执行根据本文描述的至少一个实施例的操作。

[0292] 图24B示出了集成电路封装组装件2470的截面侧视图。集成电路封装组装件2470示出了如本文描述的一个或多个处理器或加速器装置的实现。封装组装件2470包括连接到衬底2480的硬件逻辑2472、2474的多个单元。逻辑2472、2474可至少部分地以可配置逻辑或固定功能性逻辑硬件实现，并且可包括本文描述的(一个或多个)处理器核、(一个或多个)图形处理器或其它加速器装置中的任何装置的一个或多个部分。逻辑2472、2474的每个单元可在半导体管芯内被实现，并且经由互连结构2473与衬底2480耦合。互连结构2473可被配置成在逻辑2472、2474与衬底2480之间路由电信号，并且可包括互连，该互连诸如但不限于凸块(bump)或柱。互连结构2473可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。可选地，衬底2480可以是环氧基

层压衬底(epoxy-based laminate substrate)。衬底2480还可包括其它合适类型的衬底。封装组装件2470可经由封装互连2483被连接到其它电气装置。封装互连2483可被耦合到衬底2480的表面,以将电信号路由到其它电气装置,诸如主板、其它芯片组或多芯片模块。

[0293] 逻辑2472、2474的单元可与桥2482电耦合,该桥2482被配置成在逻辑2472、2474之间路由电信号。桥2482可以是为电信号提供路线(route)的密集互连结构。桥2482可包括由玻璃或合适的半导体材料构成的桥衬底。可在桥衬底上形成电路由部件(electrical routing feature),以在逻辑2472、2474之间提供芯片到芯片连接。

[0294] 虽然示出了逻辑2472、2474的两个单元和桥2482,但是本文描述的实施例可包括在一个或多个管芯上的更多或更少逻辑单元。由于当逻辑被包括在单个管芯上时可排除桥2482,因此可通过零个或多于零个桥来连接一个或多个管芯。备选的是,可通过一个或多个桥来连接多个管芯或逻辑单元。另外,在其它可能配置(包括三维配置)中可将多个逻辑单元、管芯和桥连接在一起。

[0295] 图24C示出了包括连接到衬底2480(例如,基础管芯)的硬件逻辑小芯片的多个单元的封装组装件2490。如本文描述的图形处理单元、并行处理器和/或计算加速器可由单独制造的多样化的硅小芯片构成。在此上下文中,小芯片是至少部分封装的集成电路,其包括可与其它小芯片被组装到更大封装中的逻辑的不同单元。带有不同IP核逻辑的小芯片的多样化集合可被组装到单个装置中。另外,可使用有源中介层(interposer)技术将小芯片集成到基础管芯或基础小芯片中。本文描述的概念能够实现GPU内的不同形式的IP之间的互连和通信。可使用不同工艺技术来制造并且在制造期间构成IP核,这避免了将多个IP(特别是在带有若干特点(flavors)IP的大的SoC上)汇聚到相同制造工艺的复杂性。能够实现多个工艺技术的使用改进了推向市场的时间,并且提供了创建多个产品SKU的有成本效益的方式。另外,解聚的IP更易于独立地被功率选通,在给定工作负载上不在使用中的组件可被断电,从而降低总体功率消耗。

[0296] 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和逻辑或I/O小芯片2474可至少部分地用可配置逻辑或固定功能性逻辑硬件实现,并且可包括(一个或多个)处理器核、(一个或多个)图形处理器、并行处理器或本文描述的其它加速器装置中的任何的一个或多个部分。存储器小芯片2475可以是DRAM(例如,GDDR、HBM)存储器或高速缓冲(SRAM)存储器。

[0297] 每个小芯片可被制作为单独的半导体管芯,并且经由互连结构2473与衬底2480耦合。互连结构2473可被配置成在各种小芯片与衬底2480内的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柱。在一些实施例中,互连结构2473可被配置成路由电信号,诸如,例如与逻辑、I/O和存储器小芯片的操作相关联的输入/输出(I/O)信号和/或功率或接地信号。

[0298] 衬底2480可以是环氧基层压衬底,然而它不限于此,并且衬底2480还可包括其它合适类型的衬底。封装组装件2490可经由封装互连2483被连接到其它电气装置。封装互连2483可被耦合到衬底2480的表面,以将电信号路由到其它电气装置,诸如主板、其它芯片组或多芯片模块。

[0299] 逻辑或I/O小芯片2474和存储器小芯片2475可经由桥2487被电耦合,该桥2487被配置成在逻辑或I/O小芯片2474与存储器小芯片2475之间路由电信号。桥2487可以是为电

信号提供路由的密集互连结构。桥2487可包括由玻璃或合适的半导体材料构成的桥衬底。可在桥衬底上形成电路由部件，以在逻辑或I/O小芯片2474与存储器小芯片2475之间提供芯片到芯片连接。桥2487还可被称为硅桥或互连桥。例如，桥2487是嵌入式多管芯互连桥(EMIB)。备选地，桥2487可只是从一个小芯片到另一小芯片的直接连接。

[0300] 衬底2480可包括用于I/O 2491、高速缓冲存储器2492和其它硬件逻辑2493的硬件组件。组构2485可被嵌入在衬底2480中以能够实现在各种逻辑小芯片与衬底2480内的逻辑2491、2493之间的通信。可选地，I/O 2491、组构2485、高速缓存、桥和其它硬件逻辑2493可被集成到基础管芯中，该基础管芯被层叠在衬底2480之上。组构2485可以是片上网络互连，或者是在封装组装件的组件之间交换数据分组的另一形式的分组交换组构。

[0301] 此外，封装组装件2490还可包括由组构2485或一个或多个桥2487互连的更少或更多数量的组件和小芯片。封装组装件2490内的小芯片可按在3D或2.5D布置来进行布置。一般而言，桥结构2487可被用于促进在例如逻辑或I/O小芯片与存储器小芯片之间的点到点互连。组构2485可被用于将各种逻辑和/或I/O小芯片(例如，小芯片2472、2474、2491、2493)与其它逻辑和/或I/O小芯片互连。衬底内的高速缓冲存储器2492可充当用于封装组装件2490的全局高速缓存、分布式全局高速缓存的一部分或者充当用于组构2485的专用高速缓存。

[0302] 图24D示出了根据实施例的包括可互换小芯片2495的封装组装件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上的标准化槽中。基础小芯片2496、2498可经由桥互连2497被耦合，该桥互连2497可类似于本文描述的其它桥互连，并且可例如是EMIB。存储器小芯片还可经由桥互连被连接到逻辑或I/O小芯片。I/O和逻辑小芯片可经由互连组构通信。基础小芯片可各自支持采用标准化格式的一个或多个槽以用于逻辑或I/O或存储器/高速缓存中的一个。

[0303] 可将SRAM和功率递送电路制作到基础小芯片2496、2498中的一个或多个基础小芯片中，所述基础小芯片2496、2498可使用相对于堆叠在基础小芯片之上的可互换小芯片2495不同的工艺技术来被制作。例如，可使用更大的工艺技术来制作基础小芯片2496、2498，而可使用更小的工艺技术来制作可互换小芯片。可互换小芯片2495中的一个或多个可互换小芯片可以是存储器(例如，DRAM)小芯片。可基于针对使用封装组装件2494的产品的功率和/或性能，为封装组装件2494选择不同存储器密度。另外，可基于针对产品的功率和/或性能，在组装时选择带有不同数量的类型的功能单元的逻辑小芯片。另外，可将包含不同类型的IP逻辑核的小芯片插入到可互换小芯片槽中，能够实现可混合并匹配不同技术IP块的混合处理器设计。

[0304] 示例性片上系统集成电路

图25-26示出了可使用一个或多个IP核来制作的示例性集成电路以及相关联的图形处理器。除了所示出的内容外，还可包括其它逻辑和电路，包括另外的图形处理器/核、外设接口控制器或通用处理器核。图25-26的与本文中的任何其它图的元件具有相同或类似的名称的元件描述了与其它图中的元件相同的元件，可以以与其类似的方式操作或起作用，可包括相同的组件，并且可链接到其它实体，如本文中其它地方所述的那些那样，但不限于此。

[0305] 图25是示出了可使用一个或多个IP核来制作的示例性片上系统集成电路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,所述图形处理器2510可以是图形处理器1408、1508、2510或本文描述的任何图形处理器的变型,并且可用于代替所描述的任何图形处理器。因此,本文中任何特征与图形处理器的组合的公开也公开了与图形处理器2510的对应组合,但不限于此。集成电路2500可另外包括图像处理器2515和/或视频处理器2520,以上处理器中的任何处理器可以是来自相同或多个不同设计设施的模块化IP核。集成电路2500可包括外设或总线逻辑,所述外设或总线逻辑包括USB控制器2525、UART控制器2530、SPI/SDIO控制器2535和I2S/I2C控制器2540。另外,集成电路可包括耦合到高清晰度多媒体接口(HDMI)控制器2550和移动工业处理器接口(MIPI)显示接口2555中的一个或多个的显示装置2545。可通过包括闪速存储器和闪速存储器控制器的闪速存储器子系统2560来提供存储。可经由存储器控制器2565提供存储器接口以便访问SDRAM或SRAM存储器装置。一些集成电路另外包括嵌入式安全引擎2570。

[0306] 图26A-26B是示出了根据本文描述的实施例的供SoC内使用的示例性图形处理器的框图。图形处理器可以是图形处理器1408、1508、2510或本文描述的任何其它图形处理器的变型。图形处理器可代替图形处理器1408、1508、2510或本文描述的形处理器中的任何其它图形处理器使用。因此,任何特征与图形处理器1408、1508、2510或本文描述的图形处理器中的任何其它图形处理器的组合的公开也公开了与图26A-26B的图形处理器的对应组合,但不限于此。图26A示出了根据实施例的可使用一个或多个IP核来制作的片上系统集成电路的示例性图形处理器2610。图26B示出了根据实施例的可使用一个或多个IP核来制作的片上系统集成电路的另外的示例性图形处理器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是更高性能图形处理器核的示例。例如,图形处理器2610、2640中的每个图形处理器可以是图25的图形处理器2510的变型,如本段开头所述的那样。

[0307] 如图26A中所示出的,图形处理器2610包括顶点处理器2605和一个或多个片段处理器2615A-2615N(例如,2615A、2615B、2615C、2615D到2615N-1和2615N)。图形处理器2610可经由单独的逻辑执行不同着色器程序,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阶段,并且生成图元和顶点数据。(一个或多个)片段处理器2615A-2615N使用由顶点处理器2605生成的图元和顶点数据来产生在显示装置上显示的帧缓冲(framebuffer)。(一个或多个)片段处理器2615A-2615N可被优化以执行如OpenGL API中提供的片段着色器程序,所述片段着色器程序可被用于执行与如针对在Direct 3D API中提供的像素着色器程序类似的操作。

[0308] 图形处理器2610另外包括一个或多个存储器管理单元(MMU)2620A-2620B、(一个或多个)高速缓存2625A-2625B和(一个或多个)电路互连2630A-2630B。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(一个或多个)片段处理器2615A-2615N)提供虚拟地址到物理地址映射,这些处理器除了引用在一个或多个高速缓存2625A-2625B中存储的顶点或图像/纹理数据之外还可引用在存储器中存储的顶点或图像/纹理数据。一个或多个MMU 2620A-2620B可与系统内的其它MMU同步,所述其它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,使得每个处理器2505-2520可参与到共享或统一的虚拟存储器系统中。图形处理

器2610的组件可与本文描述的其它图形处理器的组件对应。一个或多个MMU 2620A-2620B可与图2C的MMU 245对应。顶点处理器2605和片段处理器2615A-2615N可与图形多处理器234对应。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的内部总线或者经由直接连接来与SoC内的其它IP核通过接口连接。一个或多个电路互连2630A-2630B可与图2C的数据交叉开关240对应。可在图形处理器2610的类似组件和本文描述的各种图形处理器架构之间找到进一步的对应关系。

[0309] 如图26B所示出的，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、(一个或多个)高速缓存2625A-2625B和(一个或多个)电路互连2630A-2630B。图形处理器2640包括提供统一着色器核架构的一个或多个着色器核2655A-2655N(例如，2655A、2655B、2655C、2655D、2655E、2655F直到2655N-1和2655N)，在该统一着色器核架构中单个核或类型或核可执行全部类型的可编程着色器代码，其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在实施例和实现之间变化。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；以及用于为基于贴片的渲染加速拼贴操作(tiling operation)的拼贴单元2658，在该基于贴片的渲染中，用于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相干性或优化内部高速缓存的使用。着色器核2655A-2655N可与例如如图2D中的图形多处理器234、或分别是图3A和图3B的图形多处理器325、350、或图3C的多核群组365A对应。

[0310] 本文描述的实施例包括提供经由脉动处理单元对稀疏数据执行算术的技术的软件、固件和硬件逻辑。一个实施例提供了在使用稀疏数据时优化对脉动阵列的训练和推理的技术。一个实施例提供了在执行稀疏计算操作时使用解压缩信息的技术。提供了一种架构，其能够实现与其它处理资源独立缩放的矩阵和/或张量处理逻辑。一个实施例提供了GPGPU上的打包数据压缩和扩展操作。

[0311] 具有张量加速逻辑和统一存储器的GPGPU

图27是根据实施例的数据处理系统2700的框图。数据处理系统2700是具有处理器2702、统一存储器2710和包括机器学习加速逻辑的GPGPU 2720的异质处理系统。处理器2702和GPGPU 2720可以是如本文描述的处理器和GPGPU/并行处理器中的任何。处理器2702可执行存储在系统存储器2712中的编译器2715的指令。编译器2715在处理器2702上执行以将源代码2714A编译成编译的代码2714B。编译的代码2714B可包括可由处理器2702执行的指令和/或可由GPGPU 2720执行的指令。在编译期间，编译器2715可执行操作以插入元数据，其包括关于编译的代码2714B中存在的数据并行性的级别的提示和/或关于与基于编译的代码2714B要分派的线程相关联的数据局域性的提示。编译器2715可包括执行这样的操作所必需的信息，或者操作可在运行时库2716的帮助下执行。运行时库2716还可辅助编译器2715编译源代码2714A，并且还可包括在运行时与编译的代码2714B链接以促进编译的指令在GPGPU 2720上的执行的指令。

[0312] 统一存储器2710表示可由处理器2702和GPGPU 2720访问的统一地址空间。统一存储器可包括系统存储器2712以及GPGPU存储器2718。GPGPU存储器2718是GPGPU 2720的地址空间内的存储器，并且可包括系统存储器2712的一些或全部。在一个实施例中，GPGPU存储器2718还可包括专供GPGPU 2720排他地使用的任何存储器的至少一部分。在一个实施例

中，储存在系统存储器2712中的编译的代码2714B可被映射到GPGPU存储器2718中以供GPGPU 2720访问。

[0313] GPGPU 2720包括多个计算块2724A-2724N，其可包括本文描述的各种处理资源中的一个或多个。处理资源可以是或包括各种不同的计算资源，诸如例如执行单元、计算单元、流播多处理器、图形多处理器或多核群组。在一个实施例中，GPGPU 2720另外包括张量(例如，矩阵)加速器2723，其可包括被设计成加速矩阵运算(例如，点积等)的子集的一个或多个特殊功能计算单元。张量加速器2723也可称为张量加速器或张量核。在一个实施例中，张量加速器2723内的逻辑组件可跨多个计算块2724A-2724N的处理资源分布。

[0314] GPGPU 2720还可包括可由计算块2724A-2724N和张量加速器2723共享的一组资源，包括但不限于一组寄存器2725、功率和性能模块2726以及高速缓存2727。在一个实施例中，寄存器2725包括直接和间接可访问的寄存器，其中间接可访问的寄存器被优化以供张量加速器2723使用。功率和性能模块2726可被配置成调整计算块2724A-2724N的功率递送和时钟频率，以对计算块2724A-2724N内的空闲组件进行功率门控。在各种实施例中，高速缓存2727可包括指令高速缓存和/或较低级数据高速缓存。

[0315] GPGPU 2720可另外包括L3数据高速缓存2730，其可用于高速缓存由张量加速器2723和/或计算块2724A-2724N内的计算元件从统一存储器2710访问的数据。在一个实施例中，L3数据高速缓存2730包括共享本地存储器2732，其可由计算块2724A-2724N内的计算元件和张量加速器2723共享。

[0316] 在一个实施例中，GPGPU 2720包括指令处置逻辑，诸如提取和解码单元2721和调度器控制器2722。提取和解码单元2721包括提取单元和解码单元以提取和解码指令，以便由张量加速器2723或计算块2724A-2724N中的一个或多个执行。指令可经由调度器控制器2722被调度到计算块2724A-2724N内的适当的功能单元或张量加速器。在一个实施例中，调度器控制器2722为可配置以执行高级调度操作的ASIC。在一个实施例中，调度器控制器2722是能够执行从固件模块加载的调度器指令的微控制器或低每指令能量处理核。

[0317] 在一个实施例中，要由计算模块2724A-2724N执行的一些功能可被直接调度到或卸载到张量加速器2723。在各种实施例中，张量加速器2723包括被配置成高效地执行矩阵计算操作的处理元件逻辑，所述矩阵计算操作诸如由3D图形或计算着色器程序使用的乘法和加法运算以及点积运算。在一个实施例中，张量加速器2723能够被配置成加速由机器学习框架所使用的操作。在一个实施例中，张量加速器2723是专用集成电路，其被明确地配置成执行一组特定的并行矩阵相乘和/或加法运算。在一个实施例中，张量加速器2723是提供可在工作负载之间更新的固定功能逻辑的现场可编程门阵列(FPGA)。可由张量加速器2723执行的一组矩阵运算可相对于可由计算块2724A-2724N执行的运算而受到限制。然而，张量加速器2723能够以相对于计算块2724A-2724N显著更高的吞吐量来执行那些操作。

[0318] 图28示出了根据实施例的由指令流水线2800执行的矩阵运算2805。指令流水线2800可被配置成执行矩阵运算2805，诸如但不限于点积运算。两个向量的点积是标量值，它等于向量的相应分量的乘积之和。可如下面的等式(1)中所示来计算点积。

$$\vec{a} \cdot \vec{b} = \sum_{i=1}^{n} a_i b_i = a_1 b_1 + \ldots + a_n b_n \qquad (1)$$

[0319] 点积可用于卷积神经网络(CNN)的卷积运算中。图28示出了使用包括点积运算的矩阵运算2805的二维(2D)卷积。虽然示出了2D卷积,但是可使用N维过滤器对N维体积执行N维卷积。感受野贴片2802高亮输入体积缓冲器2804中的输入体积的一部分。输入体积缓冲器可存储在存储器2830中。可在感受野贴片2802内的数据和卷积过滤器之间执行点积矩阵运算2805,以在输出缓冲器2806内生成数据点,该数据点也可存储在存储器2830中。存储器2830可以是本文描述的存储器中的任何存储器,其包括系统存储器2712、GPGPU存储器2718或一个或多个高速缓冲存储器2727、2730,如图27中的那样。

[0320] 输出缓冲器2806内的数据点的组合表示由卷积运算生成的激活图。由跨输入体积缓冲器2804的滑动感受野贴片生成激活图内的每个点。激活图数据可被输入到激活函数以确定输出激活值。在一个实施例中,输入体积缓冲器2804的卷积可在框架内被定义为高级矩阵运算2905。高级矩阵运算可经由图元运算来执行,诸如基本线性代数子程序(BLAS)运算。图元运算可经由由指令流水线2800执行的硬件指令来加速。

[0321] 用于加速硬件指令的指令流水线2800可包括可提取和解码硬件指令的指令提取和解码单元2721,以及可将解码的指令调度到张量加速器2723和/或计算块2724A-2724N内的一个或多个处理资源的调度器控制器2722。在一个实施例中,可将硬件指令调度到计算块2724A-2724N,并且将其卸载到张量加速器2723。执行矩阵运算2805的一个或多个硬件指令和相关联数据可存储于存储器2830中。硬件指令的输出也可存储在存储器2830中。

[0322] 在一个实施例中,张量加速器2723能够执行一个或多个硬件指令以使用集成脉动阵列2808(DP逻辑)来执行矩阵运算2805。脉动阵列2808可包括可配置成执行点积运算的可编程和固定功能硬件的组合。虽然计算块2724A-2724N内的功能单元也可被配置成执行点积运算,但是脉动阵列2808可被配置成以相对于计算块2724A-2724N显著更高的吞吐量执行点积运算的有限子集。

[0323] 图29A-29B示出了根据一些实施例的基于硬件的脉动阵列2808的细节。图29A示出了可配置成在单个时钟循环内执行多个点积运算的多个功能单元的网格。图29B示出了单个示例性功能单元。这在脉动矩阵计算期间。

[0324] 如图29A中所示,在一个实施例中,脉动阵列2808可配置成使用各种功能单元来执行一组并行点积运算。点积可以以“脉动”方式执行,其中SIMD数据跨多层功能单元被泵送(pump)。如图29A中所示,在一个实施例中,脉动阵列2808可配置成使用各种功能单元来执行一组并行点积运算。点积可以以“脉动”方式执行,其中SIMD数据跨多层功能单元被泵送。脉动阵列2808是布置在网格中的功能单元的集合。功能单元的网格以锁步方式工作,并且被优化以执行相乘累加运算。要由脉动阵列2808运算的矩阵被划分成子矩阵,所述子矩阵跨功能单元的网格被泵送。

[0325] 在一个实施例中,脉动阵列2808可使用可配置的脉动深度处理可配置数量的SIMD数据通道。对于给定指令,可选择SIMD宽度和脉动深度以处理一组源数据。脉动深度定义将用于处理指令的硬件逻辑的脉动层的数量。脉动层是具有可变SIMD宽度的乘法器和加法器

逻辑单元的群组，其中脉动层可接收初始累加器值作为输入，并生成用于输出到后续脉动层或输出到寄存器的点积值。

[0326] 在一些实施例中，可处理三个源，其中每个源可以是向量寄存器或立即数。在一个实施例中，源2900(SRC0)可以是一个或多个初始累加器值，其可以是单个值或累加器值的向量。初始累加器值将被加到由第一脉动层内的每个功能单元计算的第一组点积。由功能单元计算的点积可被提供到给定SIMD通道的下一脉动层。点积可基于源2901(SRC1)和源2902(SRC2)来计算，它们是可包含打包数据的一个或多个通道的向量寄存器，每个通道包含四元素向量。在一个实施例中，每个通道是32位宽，并且提供四个8位向量元素。一些实施例可配置成从具有8位元素、4位元素和/或2位元素的输入向量计算点积。在一个实施例中，可使用所支持的元素大小(例如，8位x2位、8位x4位、4位x4位等)的任何组合来执行混合精度运算。在一个实施例中，脉动阵列2808被配置用于整数计算，尽管在一些实施例中可配置自动固定点操作。尽管本文描述的指令是四元素点积，但是在一些实施例中，脉动阵列2808还可被配置成支持对每向量的不同数量的元素的浮点点积计算。

[0327] 在一个实施例中，四元素向量的多个通道可被打包到各种宽度(例如，64位、128位、256位、512位等)的单个向量寄存器中。可经由脉动阵列2808为经由源2901和源2902提供的向量元素的多个通道计算同时点积。可基于用于点积计算的所选择的执行大小和脉动深度来配置要处理的向量元素的通道的数量。在一个实施例中，可使用脉动阵列2808的多个循环来计算比指定执行大小和/或脉动深度更宽的源向量。

[0328] 在给定时钟循环内可执行的计算的数量可基于SIMD通路(lane)和脉动层的数量而变化。如图所示，脉动阵列2808可使用脉动深度为四来对吞吐量的每SIMD通路执行十六个点积。如果被配置用于八个SIMD通路，则逻辑可在给定循环内执行128个八位整数(INT8)点积。如果被配置用于八个SIMD通路和脉动深度为八，则每个通路可执行32个八位整数(INT8)点积和总共256个点积。这些特定数量的操作是一个实施例的示例，并且其它实施例在吞吐量上变化。此外，如果数据类型不同，则将基于不同的数据类型来缩放操作的数量。

[0329] 在每个功能单元处，经由乘法器和加法器逻辑来计算点积，并且点积被加到累加器值。所得到的数据可被输出到目的地寄存器或提供给下一脉动层的累加器。功能单元2912的细节在图29B中示出。

[0330] 如图29B中所示，功能单元2912可包括一组输入数据缓冲器2904、2906和累加器2922，它们各自可接受输入数据。在一个实施例中，数据缓冲器2906可接受源2902(SRC2)，其可以是输入数据的打包向量。输入数据缓冲器2904可接受源2901(SRC1)，其也可为输入数据的打包向量。累加器2922可接受为功能单元2912提供初始累加器值的源2900(SRC0)。初始累加器值被加到从源2901和源2902的元素计算的点积。使用一组乘法器2923A-2923D和加法器2924经由对源向量的逐元素乘法来计算点积。乘法器2923A-2923D用于计算一组乘积。由加法器2924计算该组乘积的总和。该总和可与经由源2900提供的任何初始值进行累加(例如，相加)。在一个实施例中，该累加值可作为输入值2926提供给下一累加器，其可驻留在随后的脉动层中。在一个实施例中，源2901可包括输入数据的多个通道。源2901的附加通道可作为SRC1输入被中继到附加SIMD通路2928。在一个实施例中，源2902可包括输入数据的多个通道。源2902的附加通道可用作到附加脉动深度内的逻辑单元的SRC2输入数据。在一个实施例中，源2900可以可选地包括多个通道，其中附加通道作为输入被提供到附

加功能单元内的累加器。在一个实施例中，源2900可以是被加到初始脉动层的每个功能单元中的每个累加器的单个值。

[0331] 图30示出了脉动阵列3000，其包括部分总和回送和用于加速稀疏矩阵相乘的电路系统。在上述脉动阵列2808中，包括权重数据的操作数可在阵列内为固定的，并且部分总和遍及阵列结构传播。虽然关于脉动阵列2808的其它细节可以是适用的，但是在脉动阵列3000中，部分总和被再循环而不是被传播到下一脉动层。在一个实施例中，脉动阵列3000可被配置有M行和N列的处理元件(PE 3012AA-PE 3012MN)。处理元件可访问以输入矩阵的行和列数据的形式存储输入数据的寄存器。寄存器可存储在脉动阵列3000本地的寄存器堆中，或者存储在与脉动阵列3000耦合或包括脉动阵列3000的处理资源的寄存器堆中。寄存器可存储矩阵A 3002A-3002M的行元素，其要与矩阵B 3001A-3002N的列元素相乘。

[0332] 在一个实施例中，可在每个时钟循环每个处理元件PE 3012AA-PE 3012MN处执行融合乘加(FMA)。矩阵A的元素乘以矩阵B的对应元素，并且然后加到累加器值，或者对于第一循环，加到可选的初始输入值(例如，SRC0)。可在每个处理元件处配置部分总和回送。在每个循环之后，累加器值可在处理元件内被回送，并且被用作下一循环的输入。一旦对整行执行了操作，就可将结果存储到寄存器堆。在一组计算循环之后，处理元件PE 3012AA-PE 3012MN之间的数据移动可基于正在执行的指令或宏操作而变化。

[0333] 加速稀疏矩阵相乘

当对稀疏输入数据执行矩阵乘法运算时，如果输入元素A或输入元素B中的任一者为零，那么相乘运算的结果将为零。因此，如果任一输入为零，则可旁路该运算，并且累加器值可作为FMA运算的结果输出而不执行相乘，从而导致稀疏矩阵运算的性能和效率的提高。

[0334] 图31A-31C示出了通过跳过零值输入的运算来加速稀疏矩阵相乘的技术。脉动阵列2808和脉动阵列3000可追踪用于输入元素的元数据，并且旁路具有一个或多个零输入的乘法。图31A示出了其中脉动阵列2808包括跳过稀疏输入加载的加载过滤器的系统3100。图31B示出了其中脉动阵列3000包括跳过包括零值操作数的矩阵相乘运算的逻辑的系统3150。图31C示出了加速稀疏矩阵相乘的方法3180。图31A-31C的技术的方面可交叉应用于脉动阵列2808和脉动阵列3000两者。虽然这些技术被应用于关于脉动阵列描述的加速稀疏矩阵相乘，但是这些技术可应用于其它类型的矩阵或张量加速器单元。

[0335] 如图31A中所示，存储器3120可存储矩阵3102和矩阵3104。脉动阵列2808可包括矩阵A负载单元3126、矩阵B负载单元3122、矩阵A馈送单元3128和矩阵B馈送单元3124。矩阵3102可作为矩阵B被加载和馈送，而矩阵3104可作为矩阵A被加载和馈送。矩阵A和矩阵B的子矩阵可通过作为脉动阵列2808的处理元件操作的功能单元3130被加载和馈送。

[0336] 在一个实施例中，负载B过滤器3127和负载A过滤器2127可包括缓冲器，以存储矩阵3102的稀疏性图3112和矩阵3104的稀疏性图3114。负载B过滤器3121可通过矩阵B负载单元3122旁路零值元素的加载。负载A过滤器3127可由矩阵A负载单元3126旁路零值元素的加载。当矩阵3102或矩阵3104中的一个的元素发生旁路时，对应元素的旁路也发生。未被旁路的元素可由功能单元3130处理。

[0337] 如图31B中所示，系统3150包括脉动阵列3000，其具有跳过包括零值操作数的矩阵相乘运算的逻辑。如上所述，每个PE 3012AA-3013MN包括用于执行矩阵运算的计算的硬件逻辑。A(A0，A1，到A_M)和B(B0，B1，到B_N)是与点积、矩阵相乘、相乘/加法或相乘累加运算相关

联的输入矩阵的元素。在一个实施例中，每个PE 3012AA-3013MN与用于与要执行的操作相关联的输入操作数的元数据(3151a、3151b、……、3151m;3152a、3152b、……、3152n;3660a、3660b、……、3160n;3170a、3170b、……、3170n)相关联。元数据可包括哪些输入操作数具有零值的指示。

[0338] 对于例如要由PE 3012AA对元素A0和元素B0执行的操作，可检查元数据3050a以确定输入A0是否为零值输入。可检查元数据3152a以确定输入B0是否为零。当PE 3012AA被配置成执行矩阵相乘运算，并且输入中的任一个为零时，可跳过PE 3012AA处的运算。

[0339] 在一个实施例中，除了跳过要执行的运算之外，还基于元数据(3150a、3150b、……、3150n;3152a、3152b、……、3152n;3660a、3660b、……、3160n;3170a、3170b、……、3170n)旁路操作数的加载。在一个实施例中，如果输入A0为零，则输入A0和输入元素B0两者的加载可被旁路。如果输入B0为零，则输入B0和输入元素A0两者的加载可被旁路。当跳过加载的旁路时，可在跳过加载的相同循环中加载紧接着要处理的元素。在这样的配置中，仅非零值被输入到脉动阵列3000。在整个矩阵A的行或矩阵B的列为零的情况下，可旁路对整个行或列的运算。

[0340] 如图31C中所示，与如本文描述的矩阵加速器、张量加速器、张量处理器、张量核等相关联的逻辑(诸如但不限于脉动阵列2808和/或脉动阵列3000)可经由方法3180加速矩阵相乘运算。方法3180可由加速器内部的控制逻辑或加速器内部的逻辑执行，并且使得加速器能够跳过针对零值输入的操作。

[0341] 方法3180包括用于与矩阵/张量加速器相关联的逻辑来加载标识输入矩阵的零和/或非零元素的元数据(3182)。元数据可采用具有每个非零元素的有效位的位字段的形式，其中位字段指示应当加载哪些元素。元数据可采用具有每个零元素的有效位的位字段的形式，其中位字段指示哪些元素应当被跳过。元数据可由矩阵或张量加速逻辑生成，或者可与要处理的输入矩阵一起作为输入被供应。

[0342] 然后，该逻辑可分析到要执行的相乘运算的输入元素的元数据(3184)。如果检测到零值输入(3185，是)，则逻辑将选择下一组输入元素(3186)，然后分析那些输入元素的元数据(3184)。如果没有检测到零值输入(3185，否)，则逻辑可将输入元素加载到处理元件中以用于处理(3186)。一旦输入元素被加载，则逻辑可对输入元素执行矩阵相乘运算(3186)。然后，逻辑可将相乘运算的结果加到累加器值。累加器值可以是一系列相乘累加运算的累加结果或相乘加法运算提供的输入值。如果有另外的元素要处理，则逻辑可然后选择下一组输入元素(3186)。如果没有额外的元素要处理，则可输出结果。

[0343] 使用上面的技术，本领域技术人员可实现例如包括张量加速器的通用图形处理单元，所述张量加速器包括旁路零值输入的矩阵相乘运算的逻辑。可基于输入的元数据来执行旁路，其中元数据可在张量加速器内或张量加速器外生成。元数据可以是为整个输入集合预生成的，或者可在对输入矩阵数据的行和列执行矩阵相乘运算的同时由张量加速器在行/列的基础上生成。张量加速器可包括多个处理元件，其可以是处理元件的脉动阵列。在一个实施例中，在输入被加载到处理元件中之前，可关于输入的子矩阵来分析或生成元数据。在一个实施例中，每个处理元件包括检测零值输入或分析标识零值输入的元数据的逻辑。在一个实施例中，当由于零值输入而跳过运算时，可在跳过操作数加载的相同时钟循环内加载下一组输入操作数。在这种配置中，处理元件可仅在所有输入操作数是非零的时加

载用于相乘运算的操作数，并且在处理元件上加载将那些操作数之前可跳过其中任何输入为零的所有相乘运算。

[0344] 经由共享寄存器堆的特殊功能计算阵列的分解

在不同的抽象程度下，本文描述的处理资源可表示与本文描述的GPU中的图形处理器或图形处理器结构(例如，并行处理单元、图形处理引擎、多核群组、计算单元、接下来的图形核的计算单元)相关联的处理元件(例如，GPGPU核、光线追踪核、张量核、执行资源、执行单元(EU)、流处理器、流播多处理器(SM)、图形多处理器)。例如，处理资源可以是图形多处理器234的GPGPU核262或张量/光线追踪核263中的一个；图形多处理器324的光线追踪核338A-338B、张量核337A-337B或GPGPU核336A-336B；图形多处理器350的执行资源356A-356D；多核群组365A-365N的GFX核370、张量核371或光线追踪核372中的一个；计算单元1506A-1506N的向量逻辑单元1563或标量逻辑单元1564中的一个；具有EU阵列1522A-1522F或EU阵列1524A-1524F的执行单元；执行逻辑1800的执行单元1808A-1808N；和/或执行单元1900。处理资源还可以是例如图形处理引擎431-432、GPGPU硬件610、GPGPU 700、处理集群706A-706H、GPGPU 806A-806D、GPGPU 1306、图形处理引擎1610、图形处理引擎集群1622和/或图形处理引擎1710内的执行资源。处理资源还可以是图形处理器2510、图形处理器2610和/或图形处理器2640内的处理资源。

[0345] 当考虑采用例如图形多处理器324形式的处理资源时，图形多处理器324可包括张量和/或光线追踪核263。图形多处理器325可包括张量核337A-337B和/或光线追踪核338A-338B。多核群组365A可包括张量核371和光线追踪核372。执行逻辑1800可包括光线追踪器1805。执行逻辑1900可包括脉动阵列1912。将这种特殊功能逻辑与ALU和/或FPU中发现的更通用的逻辑紧密耦合存在优点。然而，这样的架构可能要求特殊功能逻辑与更通用的逻辑成正比地缩放。

[0346] 本文描述的实施例提供了一种架构，其中特殊功能逻辑与包含通用执行逻辑的处理资源分离。此类特殊功能逻辑可包括用于矩阵/张量加速的脉动阵列或用于加速光线追踪的光线追踪核。此类架构可提供包括独立于特殊功能逻辑而缩放通用处理单元的数量的能力的益处。虽然下文关于脉动阵列给出实例，但本文描述的分解技术一般可应用于GPGPU内的特殊功能逻辑。

[0347] 图32示出了根据实施例的包括分解的脉动阵列3212A-3212B的计算块3200。取代如图28中在单独的张量加速器2723中包括脉动阵列2808，或者如图19中在每个执行单元1900中包括脉动阵列1912，可在与图27的计算块2724A-2724N中的一个类似的计算块3200中包括脉动阵列3212A-3212B的分解集合。计算块3200还可包括图18A的执行逻辑1800的组件，其包括可与EU 1808A-1808N类似的多个互连的处理资源(PR 3208A-3208O)或者如本文描述的任何其它处理资源。

[0348] 脉动阵列3212A-3212B包括数据处理单元的W宽和D深的网络，其可用于以脉动方式执行向量或其它数据并行操作(类似于本文描述的其它脉动阵列)。在一个实施例中，脉动阵列3212A-3212B可被配置成执行矩阵运算，诸如矩阵点积运算。在一个实施例中，脉动阵列3212A-3212B支持16位浮点运算以及8位和4位整数运算。在一个实施例中，脉动阵列3212可被配置成加速机器学习操作。在这样的实施例中，脉动阵列3212可被配置有支持bfloat 16位浮点格式。通过在计算块3200内但在PR 3208A-3208O之外包括脉动阵列

3212A-3212B,可独立于PR 3208A-32080的数量来缩放脉动阵列3212A-3212B的大小和数量。另外,可保留PR内的通信带宽,否则该通信带宽将被脉动阵列活动消耗。此外,当不执行矩阵工作负载时,脉动阵列3212A-3212B可被时钟/功率门控。

[0349] 脉动阵列3212A-3212和PR 3208A-32080之间的通信可经由高速缓存或共享本地存储器(高速缓存/SLM 3210)和/或共享寄存器堆3214来执行。在一个实施例中,代替不同的共享寄存器堆3214,高速缓存/SLM 3210可被分区以便用作共享寄存器堆。共享寄存器堆3214可与其它GPGPU寄存器堆(诸如图19中的寄存器堆1906)类似地构造。共享寄存器堆还可包括用于配置脉动阵列3212A-3212B和PR 3208A-32080之间的交互的一组专用寄存器。高速缓存/SLM 3210可以是L1高速缓存、L2高速缓存和/或显式可寻址的管芯上存储器的块。

[0350] 用于由脉动阵列3212A-3212B处理的矩阵数据可存储在高速缓存/SLM 3210中。处理命令或指令可经由共享寄存器堆3214被提供给脉动阵列3212A-3212B。处理结果可由PR 3208A-32080从高速缓存/SLM 3210中读取,或者从共享寄存器堆内的目的地/输出寄存器中读取。在操作期间,代替消耗PR 3208A-32080内的总线/组构带宽,通信业务可被本地化到脉动阵列3212A-3212B、高速缓存/SLM 3210和/或共享寄存器堆3214。计算块3200内的PR 3208A-32080中的任何可将矩阵工作负载卸载到脉动阵列3212A-3212B中的一个或两个。可从PR向脉动阵列发送消息,所述消息具有指定要执行的运算和所述运算的操作数的命令。脉动阵列3212A-3212B可执行所请求的运算(相乘/加法、融合相乘/加法、相乘/累加、点积等),并将结果输出到共享寄存器堆3214。用于所请求的运算的输入、中间和/或输出数据可存储在高速缓存/SLM 3210中,并且多个相关操作可被链接(chain)。在一个实施例中,当执行用于神经网络的训练或推理的处理操作时,脉动阵列3128A-3128B还可执行包括但不限于S形(sigmoid)、ReLU和双曲正切(TanH)激活的激活函数。在这样的实施例中,可以以粗粒度将用于神经网络的操作卸载到脉动阵列3212A-3212B。

[0351] 使用上面的技术,本领域技术人员可实现例如包括计算块的通用图形处理单元,该计算块包括第一组通用处理资源和第一矩阵加速器,该第一组通用处理资源经由共享寄存器堆与第一矩阵加速器耦合。在一个实施例中,计算块包括与第一矩阵加速器耦合的第二组通用处理资源。在一个实施例中,计算块包括与第一组通用处理资源和第二组通用处理资源耦合的第二矩阵加速器。第一组通用处理资源或第二组通用处理资源内的处理资源可接收与要执行的矩阵运算相关联的解码的指令,并且将矩阵运算卸载到第一矩阵加速器或第二矩阵加速器。

[0352] 矩阵运算被卸载到的矩阵加速器可执行矩阵运算,而卸载通用处理资源可并行地执行其它操作。例如,在卸载通用处理资源上执行的线程可卸载操作并产生执行。卸载通用处理资源然后可上下文切换到不同的线程并且继续线程执行直到接收到完成的所卸载的操作的通知为止。卸载通用处理资源然后可上下文切换回到卸载线程,所述卸载线程然后经由在通用处理资源与矩阵加速器之间共享的共享寄存器堆和/或存储器来读取所卸载的操作的结果。所卸载的操作可包括用于矩阵加速器将神经网络激活函数应用于经处理的神经网络数据。

[0353] GPGPU上的打包数据压缩和扩展操作

可压缩要由矩阵加速器处理的稀疏数据,以用于功能单元之间的传输或以用于存

储在存储器中。然后可在源处扩展稀疏数据。启用打包数据压缩和扩展操作可导致在传送稀疏数据时装置组构和/或存储器总线上所消耗的带宽量的减少。

[0354] 图33示出了用于128位宽操作数的打包字节3310、打包字3320和打包双字(dword)3330的数据类型。该示例的打包字节格式3310是128位长，并且包含十六个打包字节数据元素。字节在这里被定义为8位的数据。每个字节数据元素的信息被存储在字节0的位7至位0、字节1的位15至位8、字节2的位23至位16、以及最后字节15的位120至位127中。因此，所有可用的位都被用在寄存器中。这种存储布置提高了处理器的存储效率。同样，在访问十六个数据元素的情况下，现在可并行地对十六个数据元素执行一个操作。

[0355] 通常，数据元素是与相同长度的其它数据元素一起存储在单个寄存器或存储器位置中的个别数据片段。在打包数据序列中，存储在寄存器中的数据元素的数量可以是128位除以个别数据元素的位长度。类似地，在其它打包数据序列中，存储在寄存器中的数据元素的数量可以是64位除以个别数据元素的位长度。尽管所说明的数据类型为128位长，但本文描述的处理器可被配置用于64位操作数、32位操作数或其它大小的操作数。该示例的打包字格式3320是128位长，并且包含八个打包字数据元素。每个打包字包含十六位信息。打包双字格式3330是128位长，并且包含四个打包双字数据元素。每个打包双字数据元素包含三十二位信息。打包的四字(quadword)是128位长，并且包含两个打包的四字数据元素。在一个实施例中，两个打包的128位操作数可存储在如图18B中所示的256位GRF寄存器1824中。

[0356] 图34示出了包括用于执行GPGPU的打包数据压缩和扩展操作的逻辑的处理系统3400。处理系统3400包括存储器3402、3432和寄存器，以存储操作码3404、源操作数3410、控制操作数3420和目的地操作3430。处理系统3400配置成执行一个或多个GPGPU指令以执行向量和/或打包数据压缩，并且执行一个或多个GPGPU指令以执行向量和/或打包数据扩展。GPGPU指令可采用如图20中的128位指令格式2010的形式，或可采用64位格式2030的形式。

[0357] 用于压缩和扩展的GPGPU指令可为源3419和/或目的地3430提供寄存器操作数，或者可引用存储器地址，从而导致来自存储器3402的隐式加载或存储到存储器3432。在从存储器3402加载源3410或者目的地3430被存储到存储器3432的情况下，加载或者存储可遍历一个或者多个高速缓冲存储器。

[0358] 在所示实施例中，控制操作数3420作为寄存器提供，尽管其它实施例可变化，使得源和/或目的地操作数中的每个引用存储器位置。在一个实施例中，控制操作数3420可以是立即值。操作码3404可从存储器位置提取并且存储在指令缓冲器中以便由解码器3406解码。然后，可将解码的操作码3404提供给处理资源3408以便执行。

[0359] 处理资源3408表示可找到的与如本文描述的GPU中的图形处理器或图形处理器结构(例如，并行处理单元、图形处理引擎、多核群组、计算单元、接下来的图形核的计算单元)相关联的GPGPU核、光线追踪核、张量核、执行资源、执行单元(EU)、流处理器、流播多处理器(SM)或图形多处理器中的任何一个或多个的处理元件或功能单元。例如，处理资源3408可以是图形多处理器234的GPGPU核262或张量/光线追踪核263中的一个的执行资源(例如，ALU或FPU)；图形多处理器325的光线追踪核338A-338B、张量核337A-337B或GPGPU核336A-336B；图形多处理器350的执行资源356A-356D；多核群组365A-365N的GFX核370、张量核371或光线追踪核372中的一个；计算单元1506A-1506N的向量逻辑单元1563或标量逻辑单元1564中的一个；具有EU阵列1522A-1522F或EU阵列1524A-1524F的执行单元；执行逻辑1800

的执行单元1808A-1808N;和/或执行单元1900。处理资源3408还可以是例如图形处理引擎431-432、GPGPU硬件610、GPGPU 700、处理集群706A-706H、GPGPU 806A-806D、GPGPU 1306、图形处理引擎1610、图形处理引擎集群1622和/或图形处理引擎1710内的执行资源。处理资源3408还可以是图形处理器2510、图形处理器2610和/或图形处理器2640内的执行资源。

[0360] 图35A-35B示出了GPGPU打包数据压缩和扩展操作。图35A示出GPGPU打包数据压缩3500。图35B示出了GPGPU打包数据扩展3550。GPGPU打包数据压缩3500和扩展3550可由图34的处理系统3400执行。GPGPU打包数据压缩3500和扩展3550在处置稀疏数据集时可具有特别用途,尽管潜在的用途不限于此。为了压缩稀疏数据集,可对源数据执行比较操作以生成标识打包数据的非零元素的控制向量。压缩操作可用于为操作打包稀疏数据。然后,扩展操作可用于扩展压缩的操作结果。

[0361] 如图35A中所示,可被指定为寄存器或存储器地址的源3410可以以元素H 3514至元素A 3521的形式存储打包数据。使用GPGPU打包数据压缩指令的源3410可基于控制操作数3420被压缩到目的地3430中。控制操作数3420选择源3410的要复制到目的地3430的元素。基于控制操作数3420的对应元素3524至3531来选择源3410的元素。将所选择的元素复制到目的地3430。如图所示,源的元素B 3520、E 3517、F 3516和H 3514由元素3530、元素3527、元素3526和元素3525来选择。所选择的元素(B 3531、E 3530、F 3529、H 2038)被压缩到目的地3430中,其中目的地3430的其余元素(元素3527、元素3526、元素2025、元素3524)不受影响,并且如果目的地3430被初始化为零则可保持零值,或者可保持用于初始化目的地3430的任何一个或多个值。源3410的未选择的元素(G 3515、D 3518、C 3519、A 3521)不被复制到目的地。在稀疏数据的情况下,可对源3410执行比较以生成非零值的掩码,并且非零值的掩码可用于生成控制操作数3420的掩码,使得压缩操作可用于从稀疏数据集收集非零值。控制操作数数据可被保存并用于在处理之后扩展数据。

[0362] 源和目的地可以是字节、16位字、32位双字或64位四字。在一个实施例中,控制操作数3420每向量元素使用一个字节。向量可包括大量的值(例如,16、128、256等)。GPGPU打包数据压缩指令可如下面的表5中所示的那样执行,其中A是控制数据,B是目的地,并且C是源数据。NUM_ELEMENTS是向量寄存器或存储器中要压缩的元素的数量。

[0363] 表5:GPGPU打包数据压缩

01	j = 0;
02	for (i=0; i<NUM_ELEMENTS; i++)
03	{
04	If (A[i] == 1)
05	B[j++] = C[i]
06	}

如图35B中所示,GPGPU打包数据扩展3550操作可用于基于控制操作数3420将数据从源3410扩展到目的地3430中。源3410和/或目的地3430可指定寄存器或存储器。控制操作数3420提供了源3410的元素要被写入到的目的地3430中的元素的映射。目的地3430的未选中元素不被写入并且可保留其初始化值。被写入目的地的源3410的元素的数量取决于被选择用于输出的目的地元素的数量。例如,所示的源3410存储元件W 3554至B 3561。控制操作数3420包括元素3564、元素3566、元素3567和元素3570中的选择值(例如,“1”),其分别对应

于目的地3430的元素3574、元素3576、元素3577和元素3580。从向量寄存器或存储器的最低地址朝向较高地址选择源3410的元素。当由控制操作数3420选择四个元素时，源的前四个元素(B 3561、E 3560、F 3559、H 3558)被写入到指示的位置处的目的地3430。元素Z 3557、Y 3556、X 3555和W 3554不被复制到目的地。向量扩展操作可如下面的表6中所示的那样来表示。

[0364] 表6:GPGPU打包数据扩展

01	j = 0;
02	for (i=0; i<NUM_ELEMENTS; i++)
03	{
04	If (A[i] == 1)
05	B[i] = C[j++]
06	}

使用上面的技术，本领域技术人员可实现例如图形多处理器，其包括图形处理器核和用于解码由图形处理器核执行的指令的指令解码器。指令可指定源操作数、目的地操作数和控制操作数。该指令当由图形处理器核执行时要使图形处理器核从由源操作数标识的位置读取一组打包数据的一个或多个元素，并将所述一个或多个元素压缩到由目的地操作数标识的位置中。所述一组打包数据的一个或多个元素由与控制操作数相关联的控制向量的对应元素选择。在一个实施例中，由源操作数和/或目的操作数标识的位置是图形多处理器的寄存器堆中的寄存器。在一个实施例中，由源操作数和/或目的操作数标识的位置是寄存器堆中在图形多处理器的处理元件和图形多处理器内或与图形多处理器耦合的矩阵加速器之间共享的寄存器。由源操作数标识的位置可以是图形多处理器可访问的存储器的存储器地址。由目的操作数标识的位置也可以是图形多处理器可访问的存储器的存储器地址。

[0365] 使用上面的技术，本领域技术人员还可实现例如图形多处理器，其包括图形处理器核和用于解码由图形处理器核执行的指令的指令解码器。指令可指定源操作数、目的地操作数和控制操作数，其中指令在被执行时使图形处理器核从由源操作数标识的位置读取一组打包数据的一个或多个元素，并且将所述一个或多个元素扩展到由目的地操作数标识的位置中，所述一组打包数据的一个或多个元素通过与控制操作数相关联的控制向量的对应元素被写入到由目的地操作数标识的位置的元素。在一个实施例中，由源操作数和/或目的操作数标识的位置是图形多处理器的寄存器堆中的寄存器。在一个实施例中，由源操作数和/或目的操作数标识的位置是寄存器堆中在图形多处理器的处理元件和图形多处理器内或与图形多处理器耦合的矩阵加速器之间共享的寄存器。由源操作数标识的位置可以是图形多处理器可访问的存储器的存储器地址。由目的操作数标识的位置也可以是图形多处理器可访问的存储器的存储器地址。

[0366] 利用GPGPU的高速缓存层级内的块稀疏性

在深度神经网络(DNN)领域中应用各种技术来改进计算速度和/或效率，同时维持结果的可接受准确度。已经确定，对于一些类型的DNN模型，可降低操作的计算精度，而不会显著降低模型的准确度。除了降低计算精度之外，例如，通过使用16位浮点值而不是36位浮点值进行训练，或者使用整数值执行推理操作，训练技术可适于将稀疏性引入到与DNN模型

相关联的权重值中。稀疏性是指具有零值的DNN的权重的比例。较高稀疏性对应于较少权重，这导致较小计算和存储要求。利用零值权重，可旁路使用这些权重执行的任何乘法，并且可使用稀疏矩阵格式来紧凑地存储和传送模型。经验结果显示DNN可容许高水平的稀疏性最小或可接受的准确度降低。鉴于这种趋势，采用GPGPU和用于执行DNN模型的训练和推理操作的并行处理器硬件将是有益的。

[0367] 图36A-36B示出了神经网络的训练数据内的非结构化稀疏性和块稀疏性之间的比较。图36A示出了非结构化稀疏性和块稀疏性之间的比较。图36B示出了神经网络的权重的块修剪(block pruning)。

[0368] 如图36A中所示，DNN模型中的稀疏性可采用如在矩阵3602中的不规则稀疏性的形式，或者采取如在矩阵3604中的块稀疏性的形式。如矩阵3602中的不规则稀疏性可在训练期间自然地发生，或者修剪技术可被应用于低于最小阈值的置零(zero-out)权重值。可通过在逐块基础上进行修剪来引起如矩阵3604中的块稀疏。可定义块大小并且可修剪低于最小绝对量的块。

[0369] 如图36B中所示，在训练DNN模型之后可通过对块内的权重值求和并且如果权重的绝对幅度低于阈值则将块中的所有权重设置为零来执行权重修剪。可通过使用例如0.4的权重阈值和4×4块来将矩阵3612修剪成块稀疏矩阵3614。对于每个4×4块，可对块中的权重求和，并且具有低于阈值的绝对量的任何块可使其权重设置为零。在推理期间可旁路涉及那些块的矩阵乘法运算。

[0370] 对GPGPU的高速缓存层级的各种适配可被配置成利用块稀疏性。机器学习框架可适于使得机器学习训练图元能够训练具有硬件加速的块稀疏性的机器学习模型。可提供稀疏权重修剪图元，其能够实现将训练后权重(post-training weight)修剪到与由GPGPU提供的高速缓存写入和读取抑制技术协作的特定块稀疏性大小。代替在GPGPU操作期间将为零的块写入高速缓存行或从高速缓存行读取为零的块，高速缓存控制元数据可用于表示零高速缓存行的一个或多个块。抑制稀疏数据的高速缓存读取和写入保留了高速缓存和存储器带宽以供非稀疏数据操作使用。

[0371] 图37示出包括其中可旁路稀疏数据访问的高速缓存的处理系统3700。处理系统3700包括处理资源3702、加载/存储单元3704、高速缓存控制器3706、高速缓存3708和存储器3720。处理系统3700利用包括表面状态元数据3722和/或高速缓存行元数据3707的元数据来抑制对稀疏数据的高速缓存访问。

[0372] 在一个实施例中，存储在存储器3720中的表面状态元数据可结合数据压缩技术使用，以能够实现对存储在存储器3720中的稀疏数据的基于零的压缩。响应于由处理资源3702经由加载/存储单元3704执行的存储器访问，高速缓存控制器可将元数据从与存储器访问相关联的表面状态元数据3722加载到高速缓存行元数据3707中。高速缓存行元数据3707可存储这样的元数据：所述元数据指示一组高速缓存行3728中的高速缓存行是否仅存储零数据和/或这样的信息是否可被写入到与所述一组高速缓存行3728相关联的一组标签3718。

[0373] 通过跳过将零值数据写入到高速缓存而实现的带宽节省可经由数据压缩在整个存储器系统中实现。

[0374] 图38是包括GPGPU数据压缩流水线的图形处理系统3800的框图。图形处理系统

3800的数据压缩流水线被配置成能够实现3D图形和媒体数据以及非类型化的计算数据的压缩。图形处理系统3800包括处理资源3805，其包括一组着色器核3810、3812、3814、数据端口3820和表面状态高速缓存3822。着色器核3810、3812、3814将存储器读取/写入消息发送到数据端口3820以访问图形处理系统3800的存储器子系统3834。对存储器子系统3834的访问由高速缓存层级3830高速缓存。在将高速缓存层级3830中的高速缓存数据写入存储器子系统3834之前，可经由GPGPU编解码器单元3832压缩该数据。GPGPU编解码器单元3832还可在将从存储器子系统3834读取的数据写入到高速缓存层级3830之前对该数据进行解压缩。

[0375] 着色器存储器表面与可存储在表面状态高速缓存3822中的表面状态相关联。表面状态具有关于表面的信息，诸如每像素的位、拼贴模式、清除像素状态、压缩状态等。由GPGPU编解码器单元使用该信息来在将数据发送到存储器子系统之前压缩该数据。对于在处理资源3805上运行的GPGPU程序，诸如深度学习和/或机器学习工作负载，数据通常是“非类型化的”(即，没有硬件数据格式被使用)且不拼贴的(即，在存储器中线性地布局)。在由软件进行的存储器分配期间，GPU驱动器将应用试探法来确定是否将针对缓冲器启用数据压缩。如果启用数据压缩，则驱动器将分配辅助缓冲器以存储压缩元数据，并且还将分配表面状态。在一些实施例中，编译器要确保从内核对缓冲器的所有访问都利用适当的表面状态指针来完成。在一些实施例中，对于GPGPU应用，表面状态将指示缓冲器存储器布局(即，未拼贴或结构化为2D/3D)。这些表面的数据格式将取决于表面的数据类型。例如，对于深度学习推理，数据类型可以是8位整数数据类型(例如，INT8)。对于深度学习训练，格式可以是16位浮动端口格式(例如，FP16、bfloat16)。

[0376] 在一些实施例中，即使计算数据表面具有指定格式，数据端口3820在访问期间也将不执行任何格式转换，因为这不是由这些应用所要求的。相反，格式信息仅用于驱动压缩/解压缩算法试探法。压缩数据“块”是由压缩单元一起压缩的高速缓存行(通常为2或4个高速缓存行)的块。

[0377] 使用上面的技术，本领域技术人员还可实现例如通用图形处理器，其包括处理资源、高速缓存控制器和高速缓冲存储器。高速缓存控制器可在高速缓冲存储器内存储用于高速缓存行的高速缓存行元数据，其中高速缓存行元数据与用于与高速缓存行相关联的存储器分配的表面状态元数据相关联。当高速缓存行元数据指示要写入高速缓存行的数据包括零值的块时，可旁路对高速缓存行的写入。在一个实施例中，表面状态元数据包括用于包含块稀疏数据的存储器分配的块稀疏元数据。可在训练期间使用块权重修剪来引起块稀疏数据的稀疏性。在块权重修剪期间，对数据的预定块内的值求和，并且当总和的绝对幅度值低于阈值时，将块内的所有值设置为零。用于块稀疏性的块大小可被选择成与高速缓存的高速缓存行粒度相对应。用于块稀疏性的块大小也可选择成与高速缓存行的压缩块(例如，两个到四个高速缓存行)对应，使得在训练期间使用的块稀疏性与由GPGPU提供的压缩和高速缓存行优化对准。

[0378] 附加的示例性计算装置

图39是根据实施例的包括图形处理器3904的计算装置3900的框图。计算装置3900可以是包括上述实施例中每个实施例的功能性的计算装置。计算装置3900可以是通信装置或被包括在通信装置内，所述通信装置诸如机顶盒(例如，基于因特网的有线电视机顶盒等)、基于全球定位系统(GPS)的装置等。计算装置3900还可以是移动计算装置或者被包括

在移动计算装置内，所述移动计算装置诸如蜂窝电话、智能电话、个人数字助理(PDA)、平板计算机、膝上型计算机、电子阅读器、智能电视、电视平台、可穿戴装置(例如，眼镜、手表、手环(bracelet)、智能卡、珠宝、服装物品等)、媒体播放器等。例如，在一个实施例中，计算装置3900包括移动计算装置，所述移动计算装置采用在单个芯片上集成计算装置3900的各种硬件和/或软件组件的集成电路(“IC”)(诸如片上系统(“SoC”或“SOC”))。

[0379] 计算装置3900包括图形处理器3904。图形处理器3904表示本文描述的任何图形处理器。图形处理器包括一个或多个图形引擎、图形处理器核以及其它图形执行资源，如本文描述的那样。这样的图形执行资源可以以包括但不限于执行单元、着色器引擎、片段处理器、顶点处理器、流播多处理器、图形处理器集群、或者适合于图形资源或图像资源的处理或者在异质处理器中执行通用计算操作的计算资源的任何集合的形式来呈现。

[0380] 在一个实施例中，图形处理器3904包括高速缓存3914，其可以是单个高速缓存或者被划分成多段高速缓冲存储器，其包括但不限于任何数量的L1、L2、L3或L4高速缓存、渲染高速缓存、深度高速缓存、采样器高速缓存和/或着色器单元高速缓存。高速缓存3914可具有如本文描述的近区和远区。高速缓存3914还可包括支持存储体散列算法的动态重新配置的动态散列逻辑。在一些实施例中，图形处理器3904包括GPGPU引擎3944，其包括共享本地存储器(SLM 3934)以及包括包括供GPGPU引擎3944使用的寄存器的寄存器堆3924。寄存器堆3924可包括通用寄存器、架构寄存器、配置寄存器和其它类型的寄存器。通用寄存器堆(GRF)和/或架构寄存器堆(ARF)也可驻留在GPGPU引擎3944内的计算单元(例如，计算3950、计算3955)的一个或多个块内的处理资源内。还可存在共享组构3942，其能够实现GPGPU引擎3944的各个组件之间的快速通信。

[0381] 如图所示，在一个实施例中，并且除了图形处理器3904之外，计算装置3900还可包括任何数量和类型的硬件组件和/或软件组件，其包括但不限于应用处理器3906、存储器3908和输入/输出(I/O)源3910。应用处理器3906可与硬件图形流水线交互以共享图形流水线功能性。经处理的数据被存储在硬件图形流水线中的缓冲器中，并且状态信息被存储在存储器3908中。所得到的数据可被转移到显示控制器以便经由如本文描述的显示装置输出。显示装置可具有各种类型，诸如阴极射线管(CRT)、薄膜晶体管(TFT)、液晶显示器(LCD)、有机发光二极管(OLED)阵列等，并且可被配置成经由图形用户界面向用户显示信息。

[0382] 应用处理器3906可包括一个或多个处理器，并且可以是至少部分地用于执行计算装置3900的操作系统(OS)3902的中央处理单元(CPU)。OS 3902可充当计算装置3900的硬件和/或物理资源与一个或多个用户之间的接口。OS 3902可包括用于计算装置3900中的各种硬件装置的驱动器逻辑，其包括图形驱动器逻辑3922，诸如图23的用户模式图形驱动器2326和/或内核模式图形驱动2329。

[0383] 要设想，在一些实施例中，图形处理器3904可作为应用处理器3906的部分(诸如，物理CPU封装的部分)存在，在这种情况下，存储器3908的至少一部分可由应用处理器3906和图形处理器3904共享，尽管存储器3908的至少一部分可专用于图形处理器3904，或者图形处理器3904可具有存储器的单独存储。存储器3908可包括缓冲器(例如，帧缓冲器)的预分配的区域；然而，本领域的普通技术人员应理解，实施例不限于此，并且可使用下部图形流水线可访问的任何存储器。存储器3908可包括各种形式的随机存取存储器(RAM)(例如，

SDRAM、SRAM等)，其包括利用图形处理器3904来渲染桌面或3D图形场景的应用。存储器控制器集线器可访问存储器3908中的数据并且将其转发到图形处理器3904以用于图形流水线处理。可使存储器3908对计算装置3900内的其它组件可用。例如，在软件程序或应用的实现中，从计算装置3900的各种I/O源3910接收的任何数据(例如，输入图形数据)可在它们被一个或多个处理器(例如，应用处理器3906)操作之前被临时排队到存储器3908中。类似地，软件程序确定应该通过计算系统接口之一从计算装置3900发送到外部实体或者存储到内部存储元件中的数据在其被传送或存储之前通常在存储器3908中临时排队。

[0384] I/O源可包括诸如触摸屏、触摸面板、触摸板、虚拟或常规键盘、虚拟或常规鼠标、端口、连接器、网络装置等的装置，并且可经由如图1中的I/O集线器107、如图3中的输入/输出(I/O)电路系统363、如图14中的平台控制器集线器1430等附连。另外，I/O源3910可包括被实现用于向计算装置3900(例如，联网适配器)和/或从计算装置3900(例如，联网适配器)转移数据的一个或多个I/O装置；或者用于计算装置3900内的大规模非易失性存储装置(例如，硬盘驱动器)。包括字母数字键和其它键的用户输入装置可用于将信息和命令选择传递到图形处理器3904。另一种类型的用户输入装置是光标控件，诸如鼠标、轨迹球、触摸屏、触摸板或光标方向键，以将方向信息和命令选择传递到GPU并控制显示装置上的光标移动。计算装置3900的相机和麦克风阵列可被用于观察姿势、记录音频和视频以及接收和传送视觉和音频命令。

[0385] 被配置为网络接口的I/O源3910可提供对网络的接入，所述网络诸如LAN、广域网(WAN)、城域网(MAN)、个域网(PAN)、蓝牙、云网络、蜂窝或移动网络(例如，第3代(3G)、第4代(4G)、第5代(5G)等)、卫星网络、内联网、因特网等。(一个或多个)网络接口可包括例如具有一个或多个天线(e)的无线网络接口。(一个或多个)网络接口还可包括例如有线网络接口，以经由网络缆线与远程装置通信，所述网络缆线可以是例如以太网缆线、同轴缆线、光纤缆线、串行缆线或并行缆线。

[0386] (一个或多个)网络接口可例如通过符合IEEE 802.11标准提供对LAN的接入，和/或无线网络接口可例如通过符合蓝牙标准提供对个域网的接入。也可支持其它无线网络接口和/或协议，其包括标准的先前和后续版本。除了经由无线LAN标准的通信之外，或者代替经由无线LAN标准的通信，(一个或多个)网络接口可使用例如时分多址(TDMA)协议、全球移动通信系统(GSM)协议、码分多址(CDMA)协议和/或任何其它类型的无线通信协议来提供无线通信。

[0387] 要领会的是，对于某些实现，与上述示例相比，更少或更多装备的系统可能是优选的。因此，计算装置3900的配置可取决于许多因素(诸如价格限制、性能要求、技术改进或其它情况)从实现到实现变化。示例包括(但不限于)移动装置、个人数字助理、移动计算装置、智能电话、蜂窝电话、手机、单向寻呼机、双向寻呼机、消息收发(messaging)装置、计算机、个人计算机(PC)、台式计算机、膝上型计算机、笔记本计算机、手持计算机、平板计算机、服务器、服务器阵列或服务器场、web服务器、网络服务器、因特网服务器、工作站、小型计算机、大型计算机、超级计算机、网络设备、web设备、分布式计算系统、多处理器系统、基于处理器的系统、消费电子产品、可编程消费电子产品、电视、数字电视、机顶盒、无线接入点、基站、订户站、移动订户中心、无线电网络控制器、路由器、集线器、网关、桥、交换机、机器或它们的组合。

[0388] 实施例可被实现为使用母板互连的一个或多个微芯片或集成电路、硬连线逻辑、由存储器装置存储并由微处理器执行的软件、固件、专用集成电路(ASIC)和/或现场可编程门阵列(FPGA)中的任何一个或其组合。作为示例,术语“逻辑”可包括软件或硬件和/或软件和硬件的组合。

[0389] 实施例可被提供为例如计算机程序产品,其可包括其上存储有机器可执行指令的一个或多个机器可读介质,所述机器可执行指令当由诸如计算机、计算机网络或者其它电子装置之类的一个或多个机器执行时,可导致一个或多个机器运行根据本文描述的实施例的操作。机器可读介质可包括但不限于软盘、光盘、CD-ROM(致密盘只读存储器)和磁光盘、ROM、RAM、EPROM(可擦除可编程只读存储器)、EEPROM(电可擦可编程只读存储器)、磁卡或光卡、闪速存储器或适于存储机器可执行指令的其它类型的非暂时性机器可读介质。

[0390] 此外,实施例可作为计算机程序产品被下载,其中程序可经由通信链路(例如,调制解调器和/或网络连接)通过体现在载波或其它传播介质中或由载波或其它传播介质调制的一个或多个数据信号从远程计算机(例如,服务器)转移到请求计算机(例如,客户端)。

[0391] 本文中对“一个实施例”或“一实施例”的引用意味着结合实施例描述的特定特征、结构或特性可包括在本发明的至少一个实施例中。说明书中各个地方出现的短语“在一个实施例中”不一定全部指相同实施例。下面的图中描绘的过程可由包括硬件(例如电路系统、专用逻辑等)、软件(作为非暂时性机器可读存储介质上的指令)或硬件和软件两者的组合的处理逻辑来执行。将对各种实施例进行详细参考,其示例在附图中示出。在以下详细描述中,阐述了许多特定细节以便提供对本发明的透彻理解。然而,对本领域普通技术人员来说将清楚的是,可在没有这些特定细节的情况下实践本发明。在其它情况下,尚未详细描述公知的方法、过程、组件、电路和网络,以免不必要地模糊实施例的方面。

[0392] 还将理解的是,尽管术语第一、第二等在本文中可用于描述各种元素,但是这些元素不应受这些术语限制。这些术语仅用于区分一种元素与另一种元素。例如,在不脱离本发明的范围的情况下,第一接触件可被术语化为第二接触件,并且类似地,第二接触件可被术语化为第一接触件。第一接触件和第二接触件两者都是接触件,但不是相同接触件。

[0393] 本文中使用的术语仅出于描述特定实施例而不旨在限制所有实施例的目的。如在本发明的描述和所附权利要求书中使用的,除非上下文另有明确指示,否则单数形式“一”“一个””和“该”也旨在包括复数形式。还将理解的是,如本文中所使用的术语“和/或”是指并涵盖相关联的所列项目中的一个或多个的任何和所有可能的组合。将进一步理解,术语“包括(comprise和/或comprising)”,当在本说明书中使用时,指定所述特征、整体、步骤、操作、元件和/或组件的存在,但不排除一个或多个其它特征、整体、步骤、操作、元件、组件和/或它们的群组的存在或添加。

[0394] 如本文中所使用的,取决于上下文,术语“如果”可被解释成意味着“当……时”或“在……时”或“响应于确定”或“响应于检测”。类似地,取决于上下文,短语“如果要确定”或“如果检测到[陈述的条件或事件]”可解释成“在确定……时”或“响应于确定”或“在检测到[陈述的条件或事件]时”或“响应于检测到[所述条件或事件]”。

[0395] 本文描述的实施例包括提供经由脉动处理单元对稀疏数据执行算术的技术的软件、固件和硬件逻辑。一个实施例提供了在使用稀疏数据时优化对脉动阵列的训练和推理的技术。一个实施例提供了在执行稀疏计算操作时使用解压缩信息的技术。一个实施例能

够实现经由共享寄存器堆的特殊功能计算阵列的分解。一个实施例能够实现GPGPU上的打包数据压缩和扩展操作。一个实施例提供了利用GPGPU的高速缓存层级内的块稀疏性的技术。

[0396] 一个实施例提供了一种通用图形处理单元，其包括矩阵加速器，该矩阵加速器包括用于旁路具有零值输入的矩阵相乘运算的逻辑，该旁路基于与输入相关联的元数据来执行。

[0397] 一个实施例提供了一种方法，该方法包括在具有矩阵加速器的通用图形处理器上分析到要由矩阵加速器执行的矩阵相乘运算的输入的元数据，到矩阵相乘运算的输入包括多个输入矩阵的一个或多个元素，基于元数据确定到矩阵相乘运算的输入是否包括零值输入，并且响应于确定矩阵相乘运算包括零值输入，旁路矩阵相乘运算的至少第一部分。

[0398] 一个实施例提供了一种数据处理系统，该数据处理系统包括存储器装置和与该存储器装置耦合的通用图形处理单元，其中该通用图形处理单元包括矩阵加速器，该矩阵加速器包括用于旁路具有零值输入的矩阵相乘运算的逻辑，基于与输入相关联的元数据执行旁路。矩阵加速器包括多个处理元件并且被配置成接收元数据作为与指定零值输入的位置的操作数相关联的输入或基于输入操作数引用的数据生成元数据，该数据包括零值输入。

[0399] 要在说明性而不是限制性的意义上看待前述描述和附图。本领域技术人员将理解，在不脱离所附权利要求书中阐述的特征的更广泛的精神和范围的情况下，可对本文描述的实施例进行各种修改和改变。

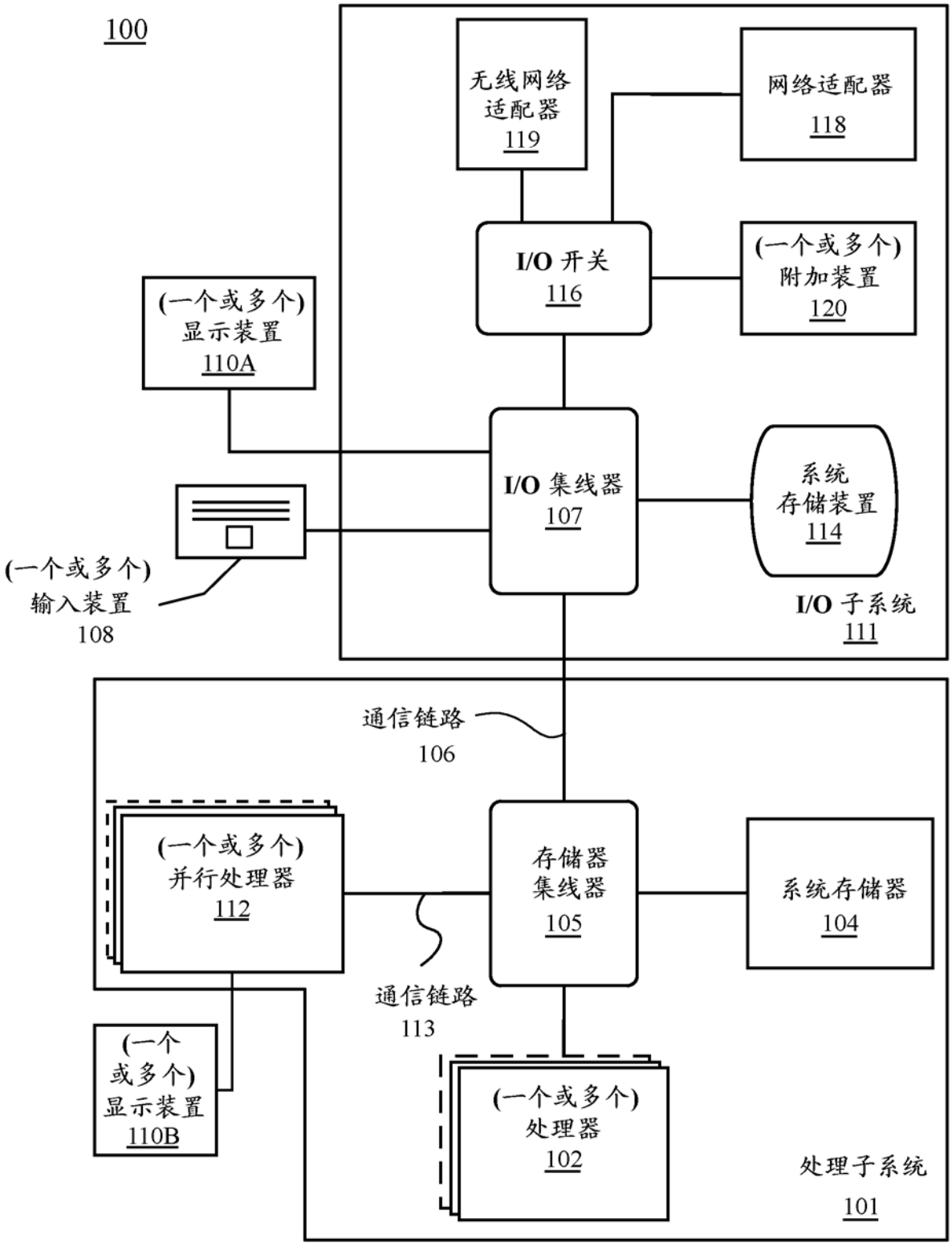

100
无线网络适配器 119
网络适配器 118
I/O 开关 116
(一个或多个)附加装置 120
(一个或多个)显示装置 110A
I/O 集线器 107
系统存储装置 114
(一个或多个)输入装置 108
I/O 子系统 111
通信链路 106
(一个或多个)并行处理器 112
存储器集线器 105
系统存储器 104
通信链路 113
(一个或多个)显示装置 110B
(一个或多个)处理器 102
处理子系统 101

图 1

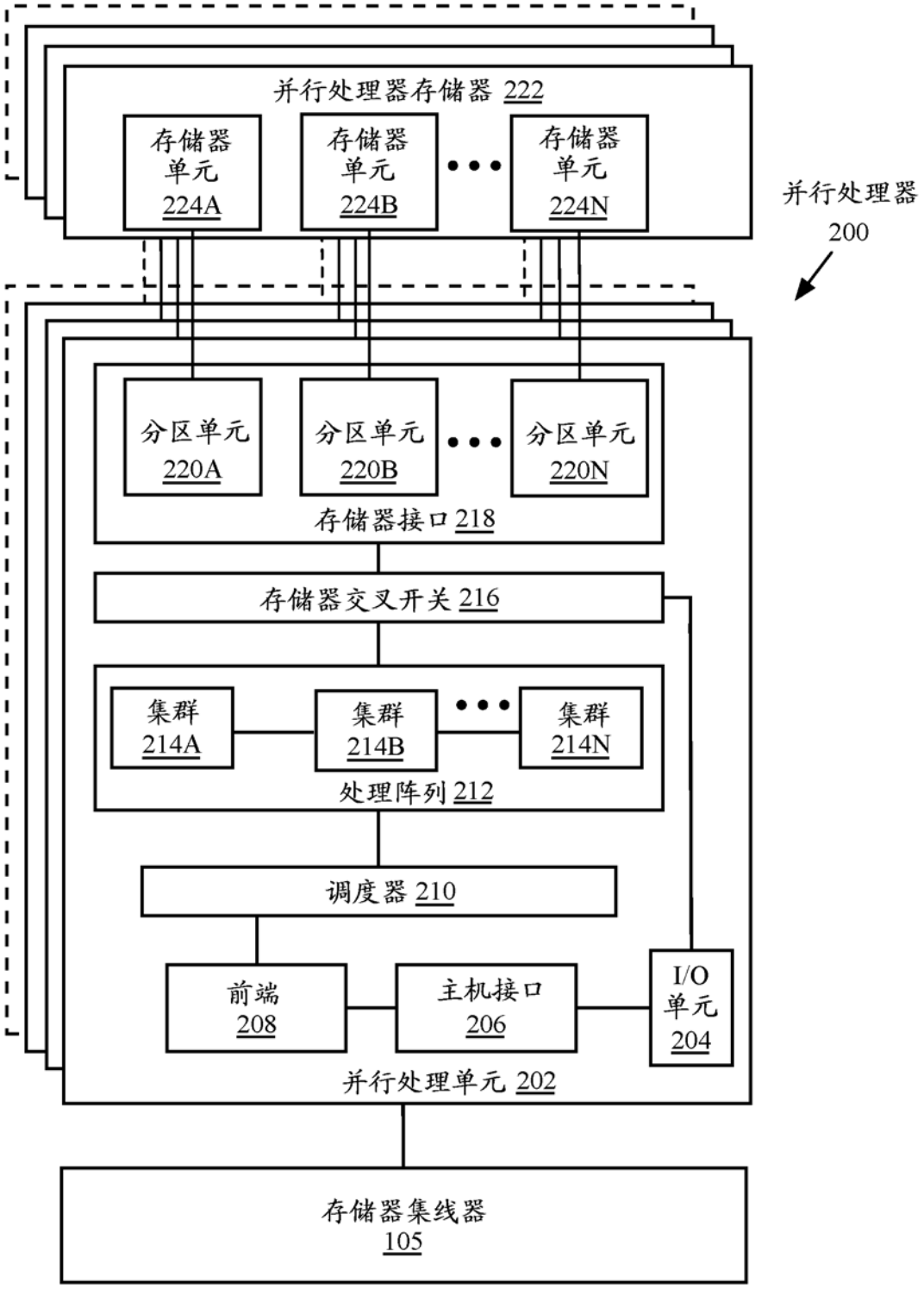

并行处理器存储器 222
存储器单元 224A
存储器单元 224B
存储器单元 224N
并行处理器 200
分区单元 220A
分区单元 220B
分区单元 220N
存储器接口 218
存储器交叉开关 216
集群 214A
集群 214B
集群 214N
处理阵列 212
调度器 210
前端 208
主机接口 206
I/O 单元 204
并行处理单元 202
存储器集线器 105

图 2A

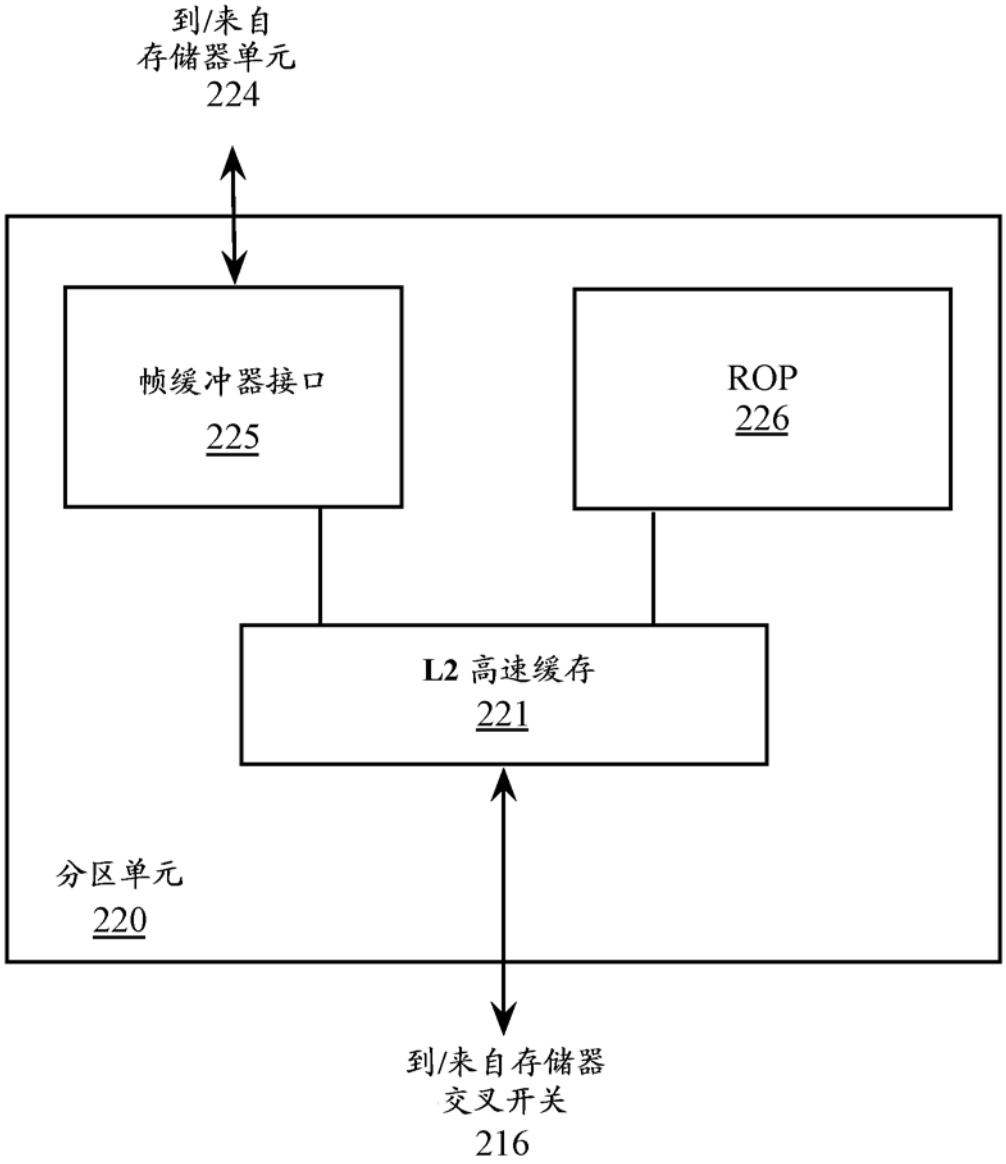

到/来自
存储器单元
224
帧缓冲器接口
225
ROP
226
L2 高速缓存
221
分区单元
220
到/来自存储器
交叉开关
216

图 2B

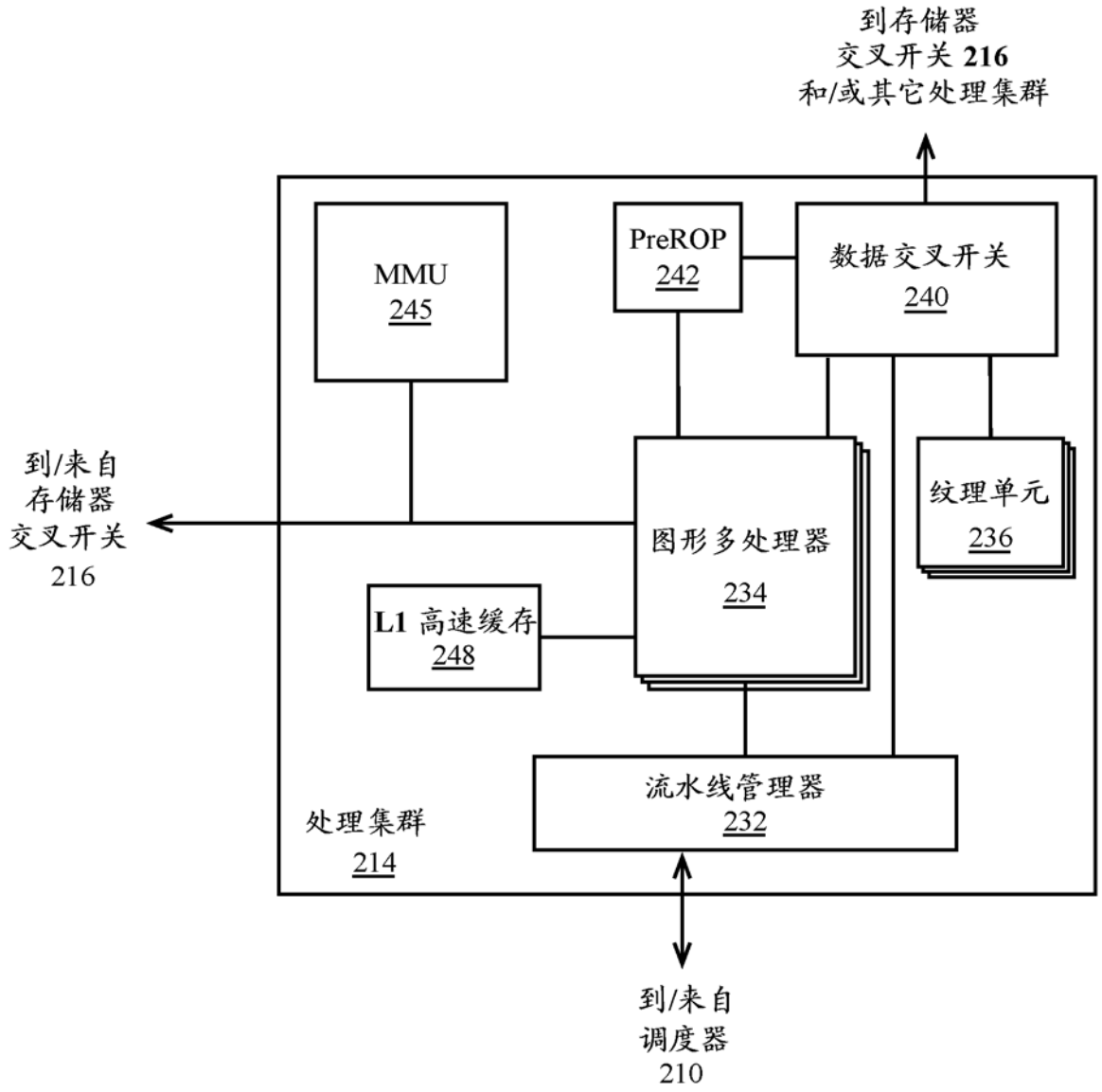

到存储器
交叉开关 216
和/或其它处理集群
MMU
245
PreROP
242
数据交叉开关
240
到/来自
存储器
交叉开关
216
图形多处理器
234
纹理单元
236
L1 高速缓存
248
流水线管理器
232
处理集群
214
到/来自
调度器
210

图 2C

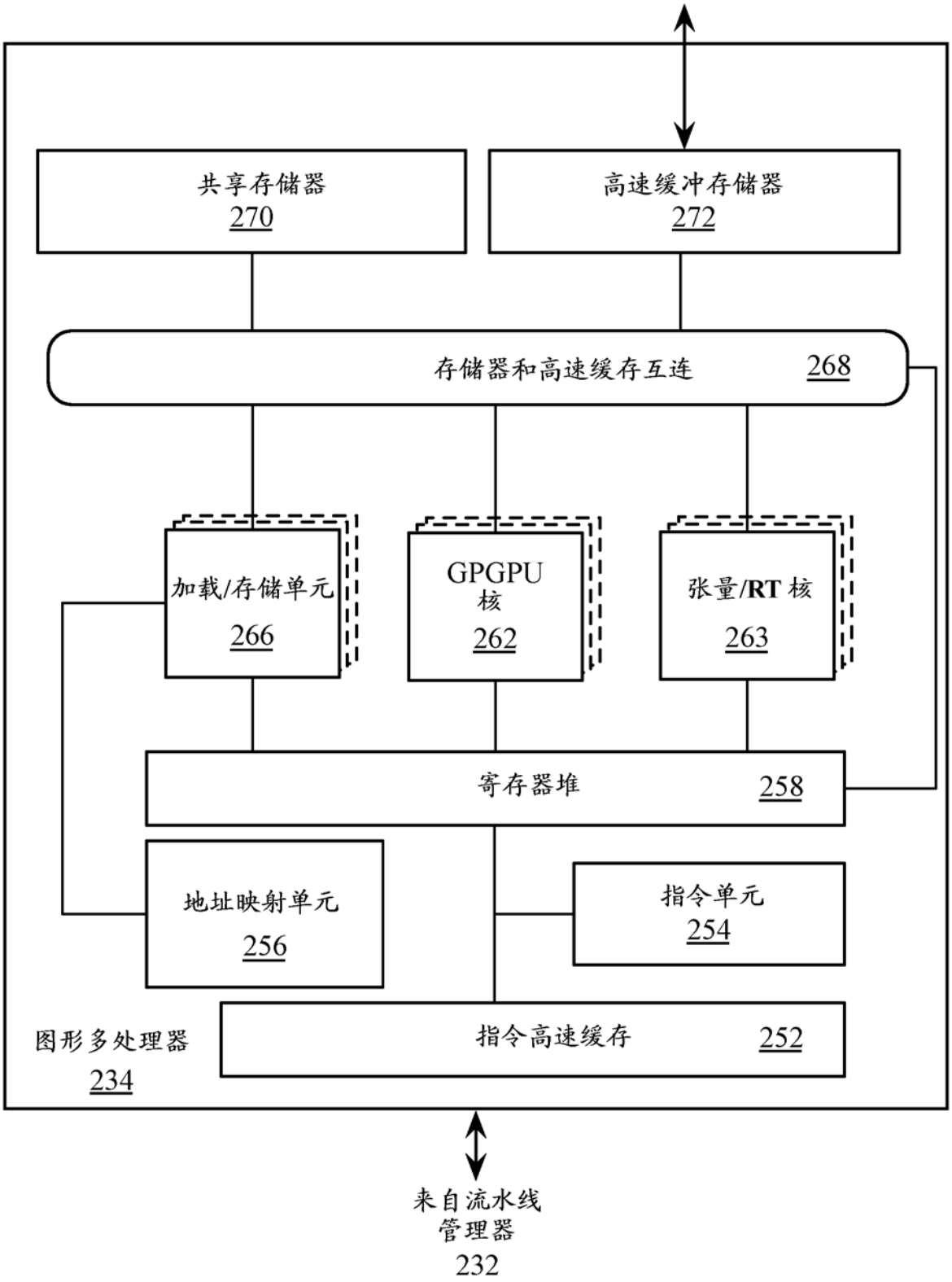

共享存储器
270
高速缓冲存储器
272
存储器和高速缓存互连
268
加载/存储单元
266
GPGPU
核
262
张量/RT 核
263
寄存器堆
258
地址映射单元
256
指令单元
254
图形多处理器
234
指令高速缓存
252
来自流水线
管理器
232

图 2D

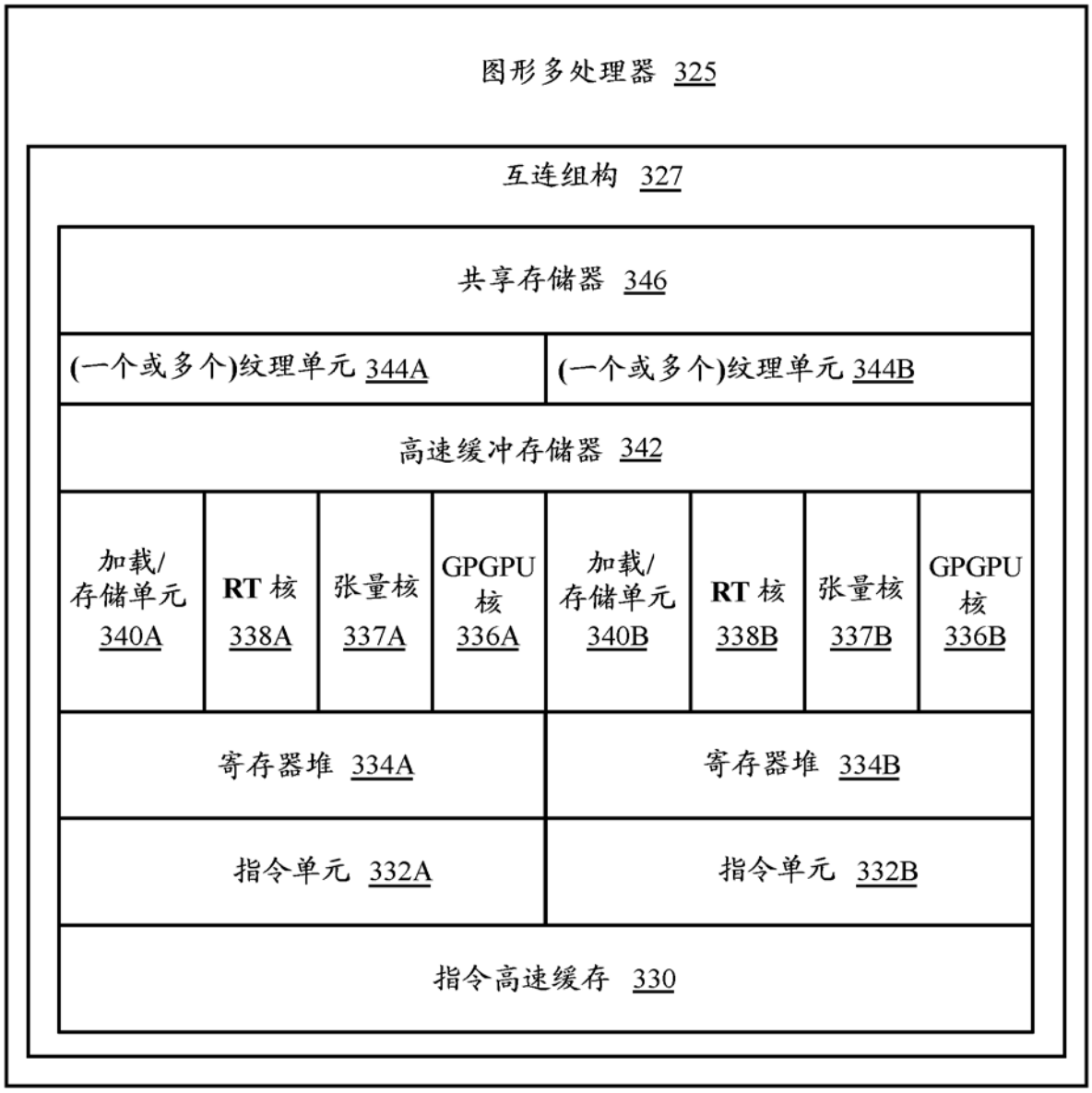

图形多处理器 325
互连组构 327
共享存储器 346
(一个或多个)纹理单元 344A
(一个或多个)纹理单元 344B
高速缓冲存储器 342
加载/存储单元 340A
RT 核 338A
张量核 337A
GPGPU 核 336A
加载/存储单元 340B
RT 核 338B
张量核 337B
GPGPU 核 336B
寄存器堆 334A
寄存器堆 334B
指令单元 332A
指令单元 332B
指令高速缓存 330

图 3A

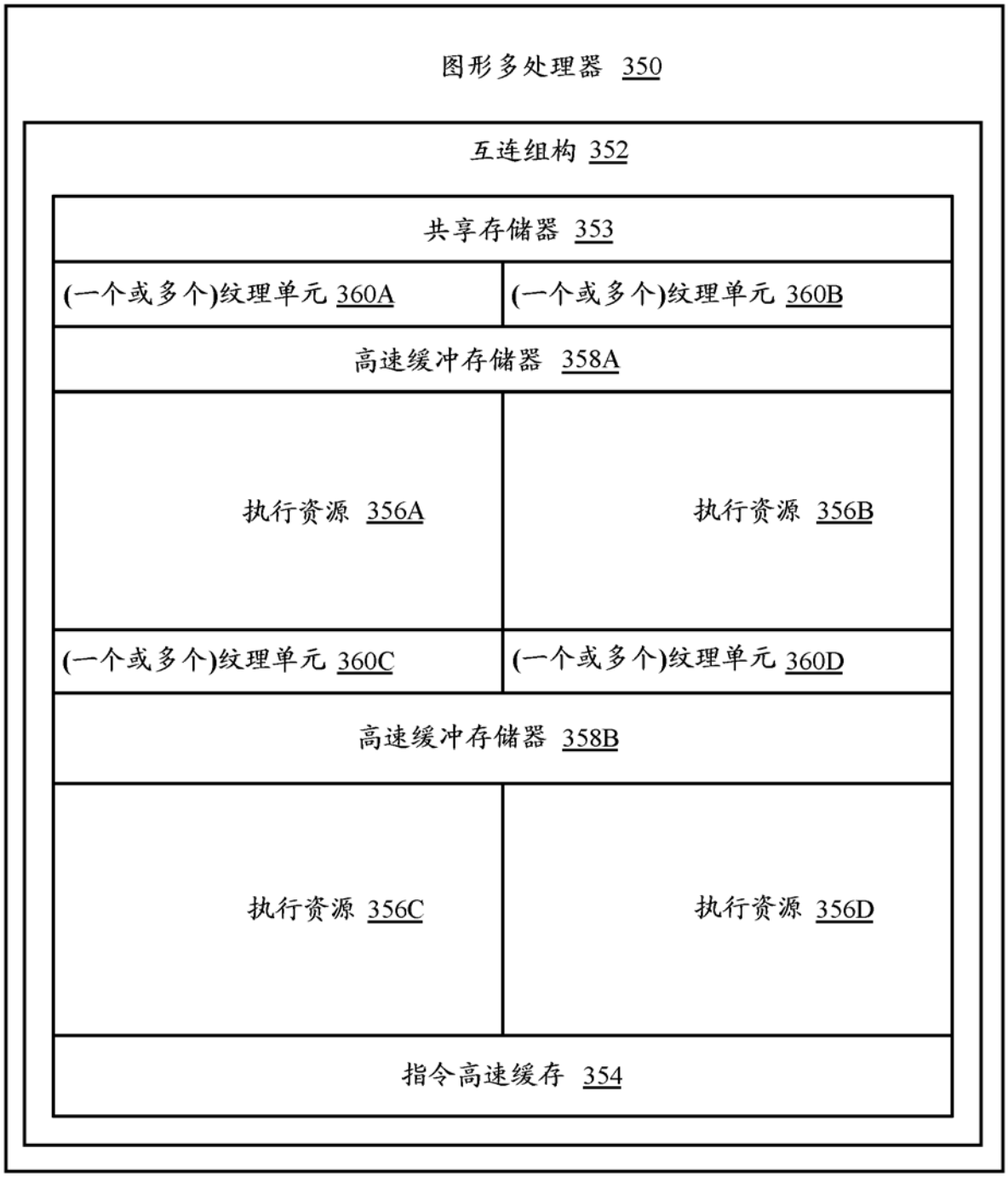

图形多处理器 350
互连组构 352
共享存储器 353
(一个或多个)纹理单元 360A
(一个或多个)纹理单元 360B
高速缓冲存储器 358A
执行资源 356A
执行资源 356B
(一个或多个)纹理单元 360C
(一个或多个)纹理单元 360D
高速缓冲存储器 358B
执行资源 356C
执行资源 356D
指令高速缓存 354

图 3B

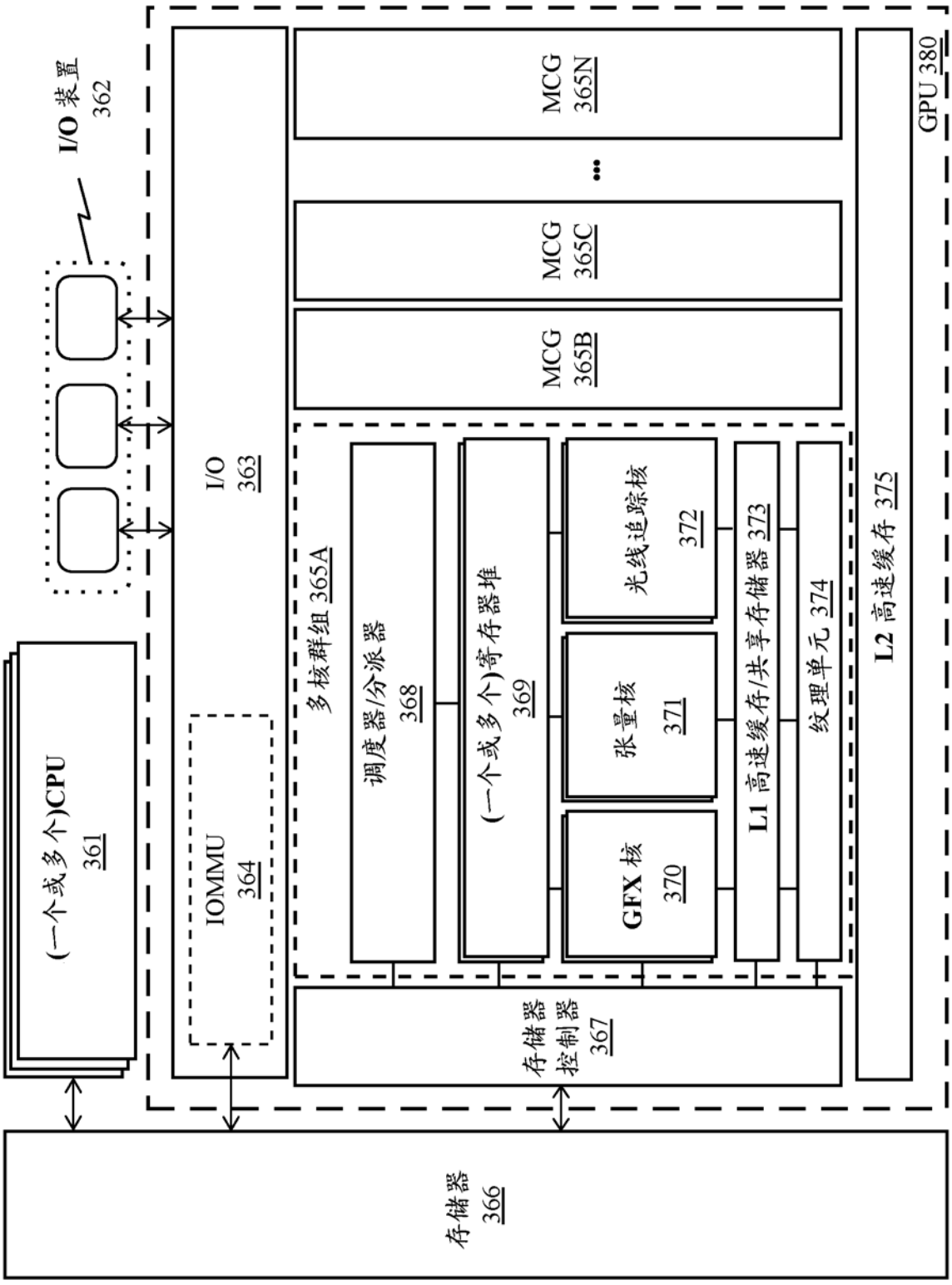

(一个或多个)CPU
361
I/O 装置
362
I/O
363
IOMMU
364
多核群组 365A
调度器/分派器
368
(一个或多个)寄存器堆
369
GFX 核
370
张量核
371
光线追踪核
372
L1 高速缓存/共享存储器 373
纹理单元 374
MCG
365B
MCG
365C
MCG
365N
L2 高速缓存 375
GPU 380
存储器
控制器
367
存储器
366

图 3C

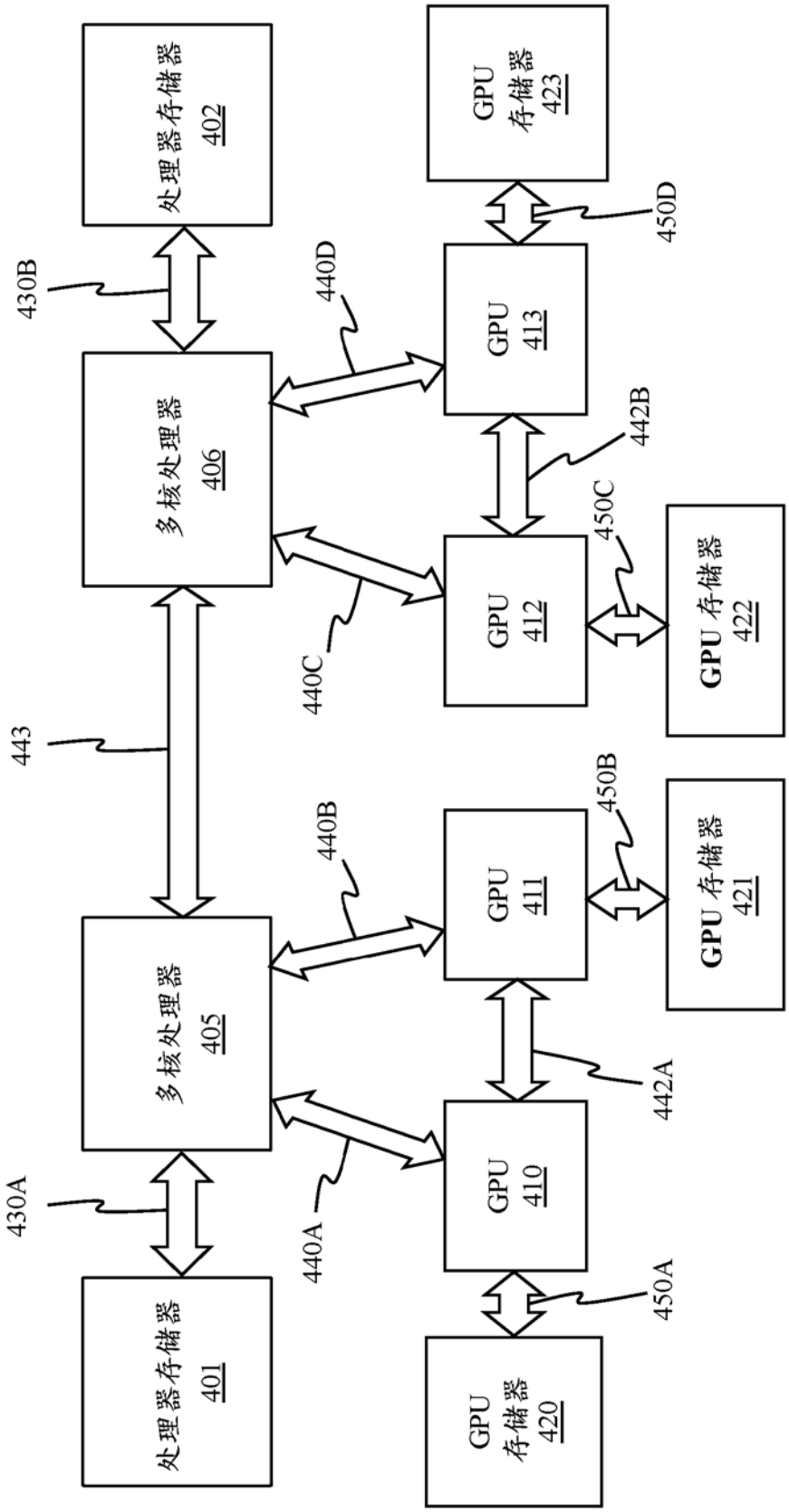

处理器存储器 401
处理器存储器 402
多核处理器 405
多核处理器 406
GPU 410
GPU 411
GPU 412
GPU 413
GPU 存储器 420
GPU 存储器 421
GPU 存储器 422
GPU 存储器 423
430A
430B
440A
440B
440C
440D
442A
442B
443
450A
450B
450C
450D

图 4A

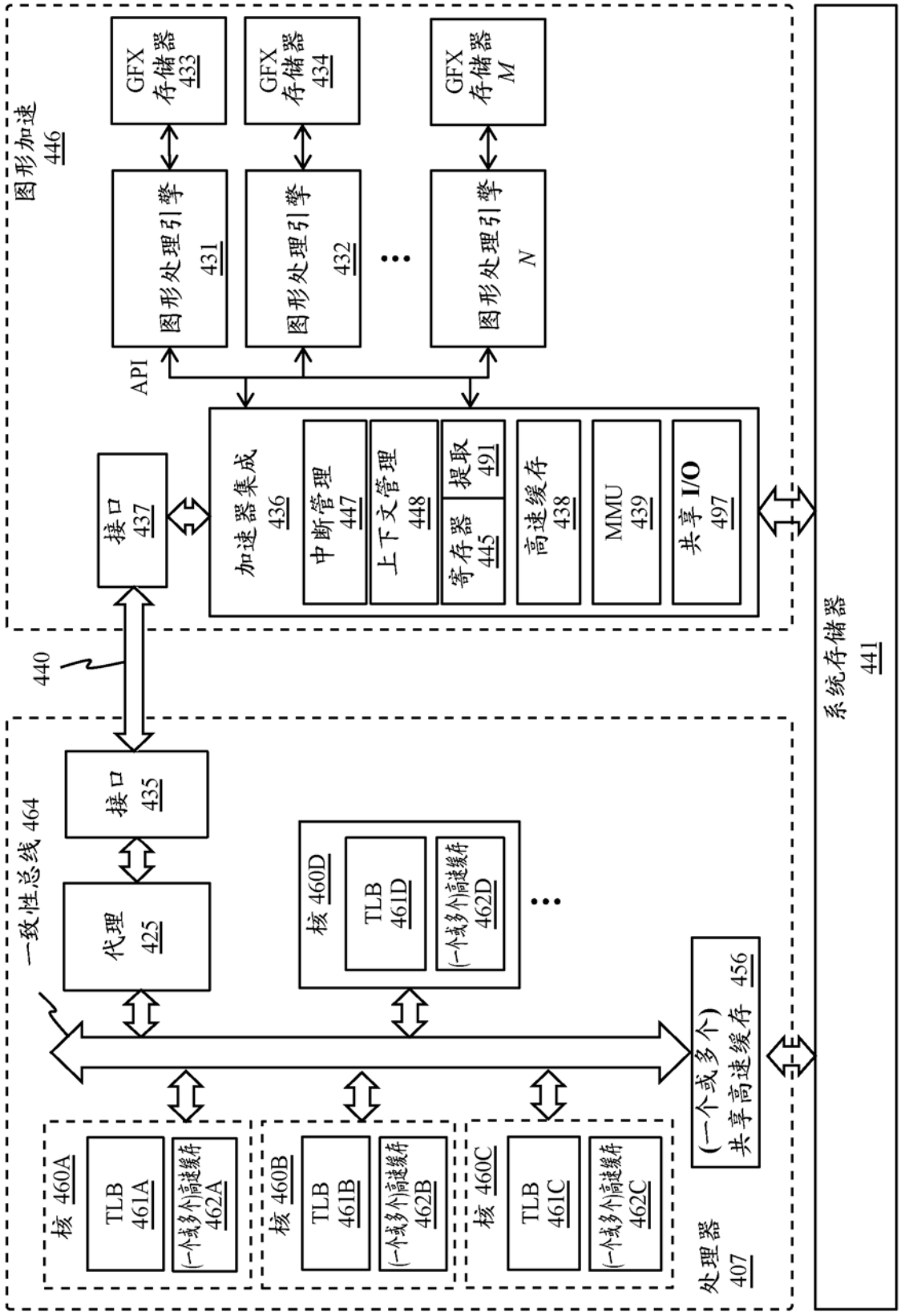

图形加速
446
GFX
存储器
433
GFX
存储器
434
GFX
存储器
M
图形处理引擎
431
图形处理引擎
432
图形处理引擎
N
API
接口
437
加速器集成
436
中断管理
447
上下文管理
448
提取
491
寄存器
445
高速缓存
438
MMU
439
共享 I/O
497
440
一致性总线 464
接口
435
代理
425
核 460D
TLB
461D
(一个或多个)高速缓存
462D
(一个或多个)
共享高速缓存 456
核 460A
TLB
461A
(一个或多个)高速缓存
462A
核 460B
TLB
461B
(一个或多个)高速缓存
462B
核 460C
TLB
461C
(一个或多个)高速缓存
462C
处理器
407
系统存储器
441

图 4B

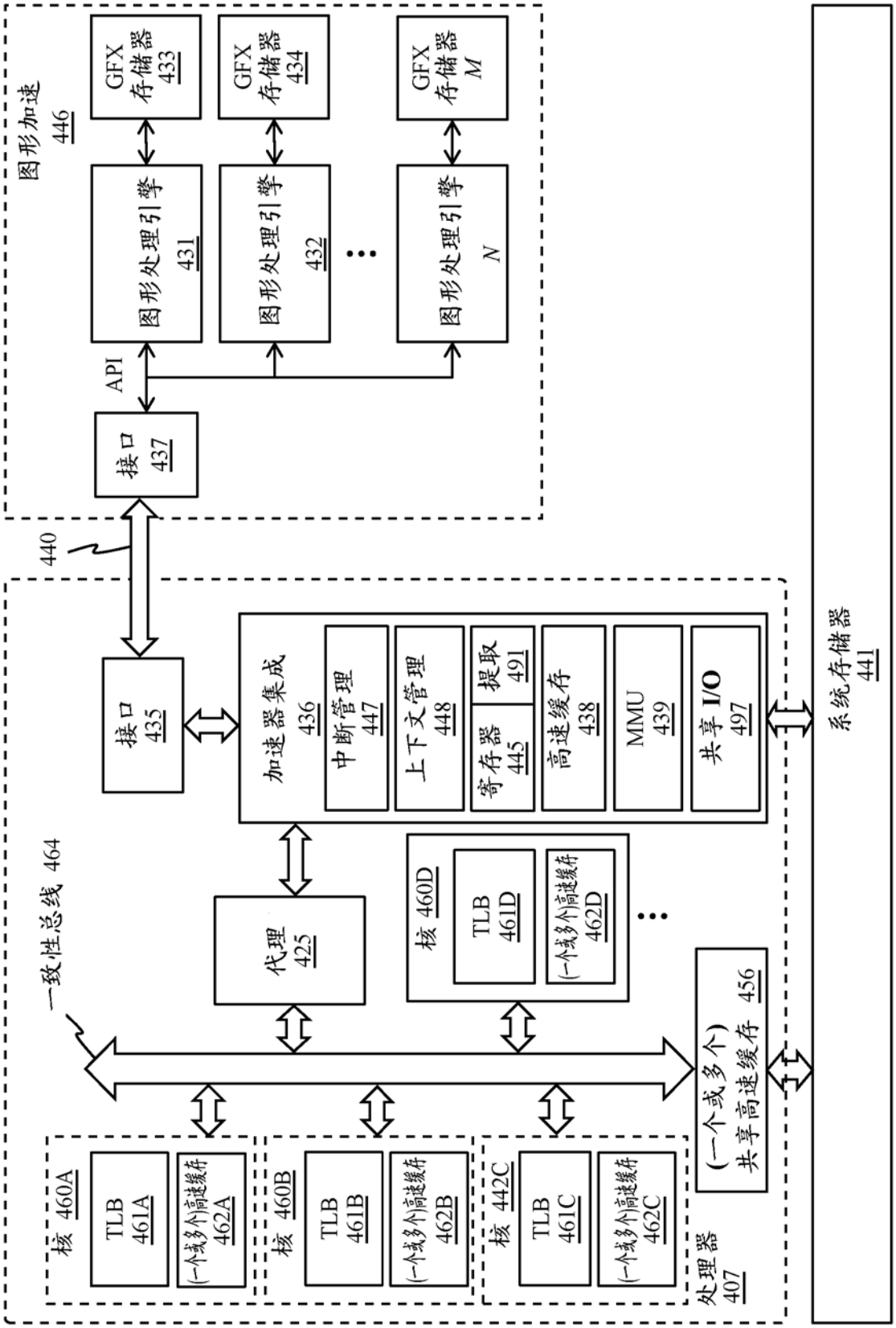

图形加速 446
GFX 存储器 433
GFX 存储器 434
GFX 存储器 M
图形处理引擎 431
图形处理引擎 432
图形处理引擎 N
API
接口 437
440
接口 435
加速器集成 436
中断管理 447
上下文管理 448
寄存器 445
提取 491
高速缓存 438
MMU 439
共享 I/O 497
系统存储器 441
一致性总线 464
代理 425
核 460D
TLB 461D
(一个或多个)高速缓存 462D
(一个或多个)共享高速缓存 456
核 460A
TLB 461A
(一个或多个)高速缓存 462A
核 460B
TLB 461B
(一个或多个)高速缓存 462B
核 442C
TLB 461C
(一个或多个)高速缓存 462C
处理器 407

图 4C

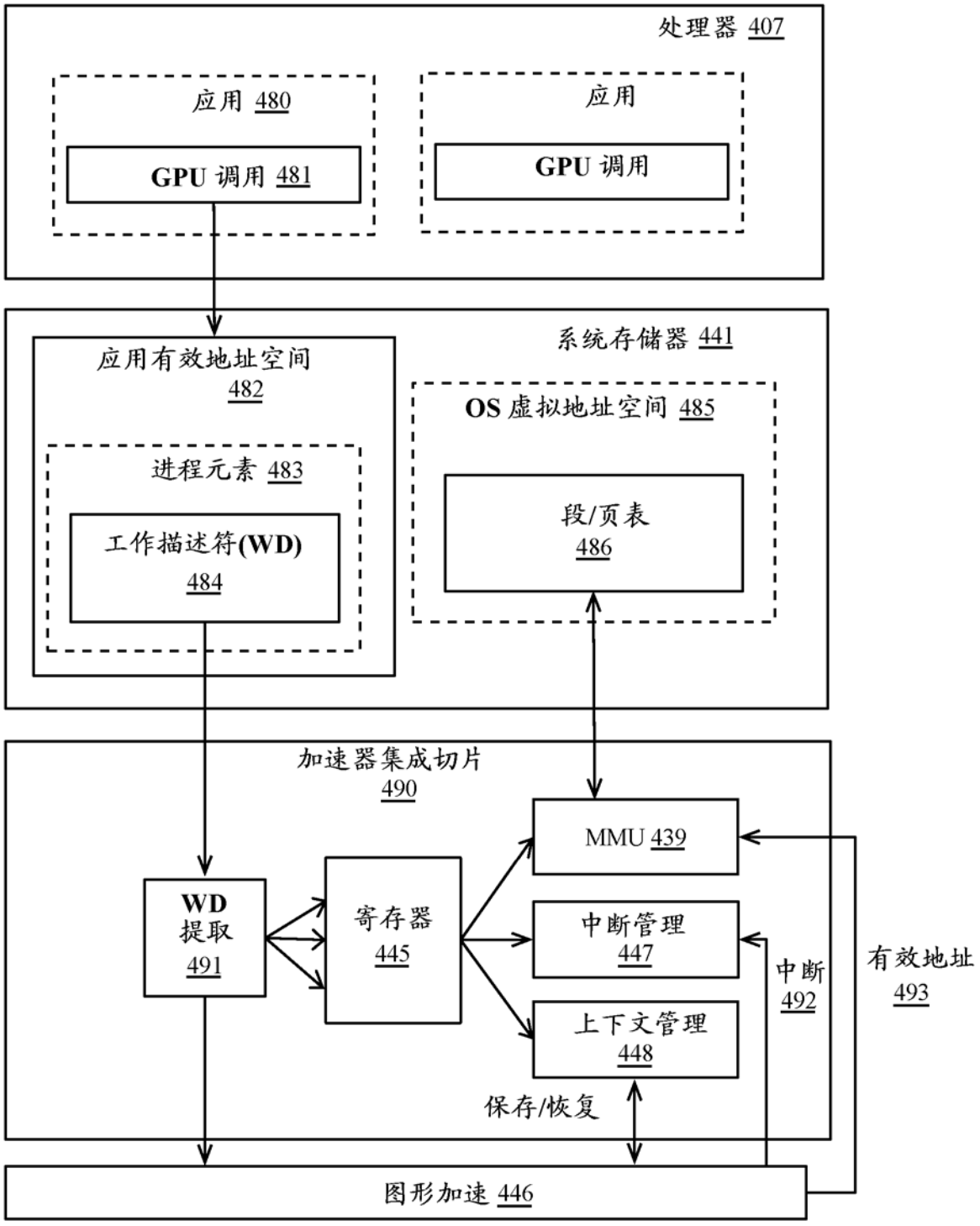

处理器 407
应用 480
GPU 调用 481
应用
GPU 调用
系统存储器 441
应用有效地址空间
482
进程元素 483
工作描述符(WD)
484
OS 虚拟地址空间 485
段/页表
486
加速器集成切片
490
MMU 439
WD
提取
491
寄存器
445
中断管理
447
上下文管理
448
中断
492
有效地址
493
保存/恢复
图形加速 446

图 4D

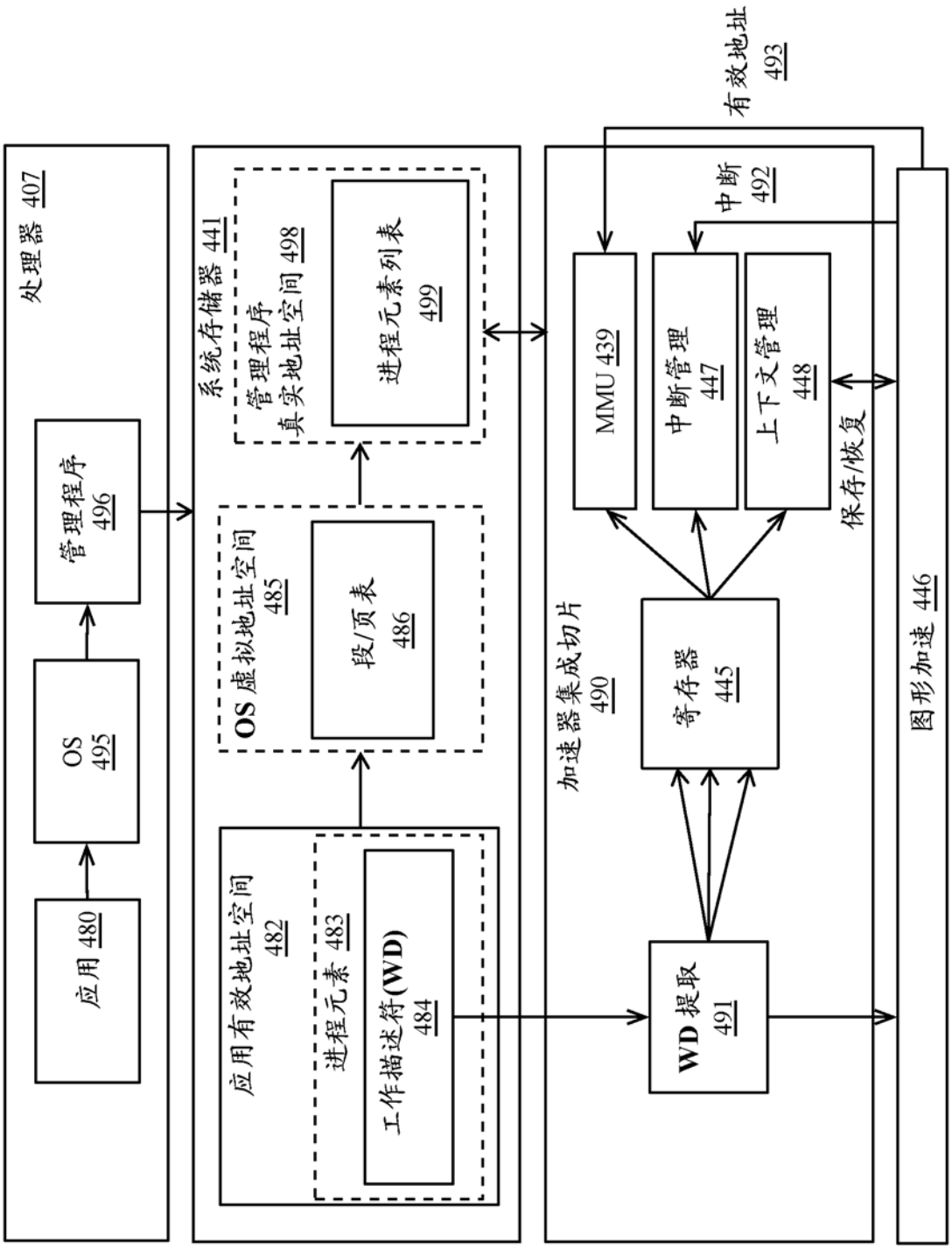

处理器 407
应用 480
OS 495
管理程序 496
系统存储器 441
管理程序 真实地址空间 498
进程元素列表 499
OS 虚拟地址空间 485
段/页表 486
应用有效地址空间 482
进程元素 483
工作描述符(WD) 484
加速器集成切片 490
MMU 439
中断管理 447
上下文管理 448
寄存器 445
WD 提取 491
保存/恢复
中断 492
有效地址 493
图形加速 446

图 4E

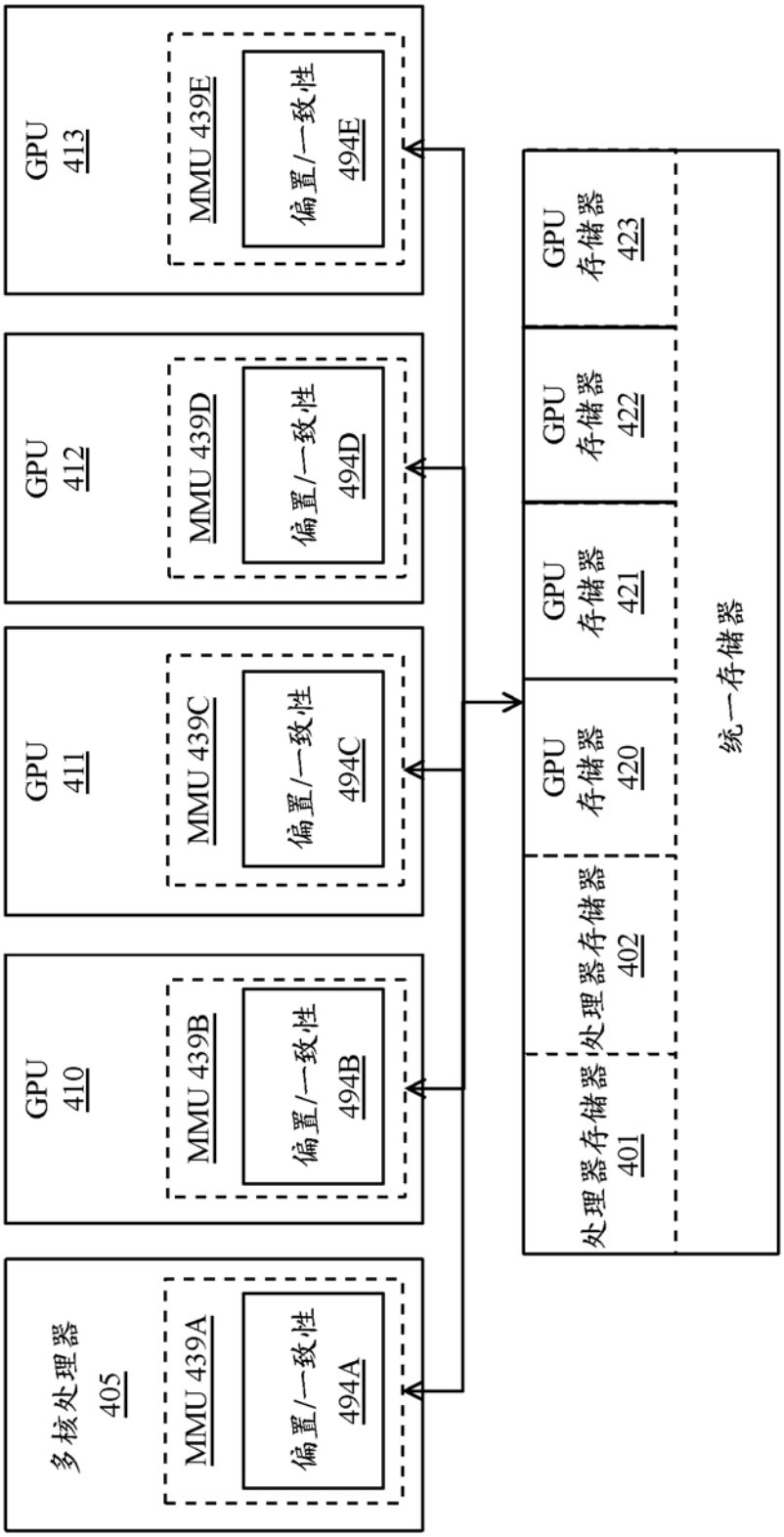

多核处理器
405
MMU 439A
偏置/一致性
494A
GPU
410
MMU 439B
偏置/一致性
494B
GPU
411
MMU 439C
偏置/一致性
494C
GPU
412
MMU 439D
偏置/一致性
494D
GPU
413
MMU 439E
偏置/一致性
494E
处理器存储器
401
处理器存储器
402
GPU
存储器
420
GPU
存储器
421
GPU
存储器
422
GPU
存储器
423
统一存储器

图 4F

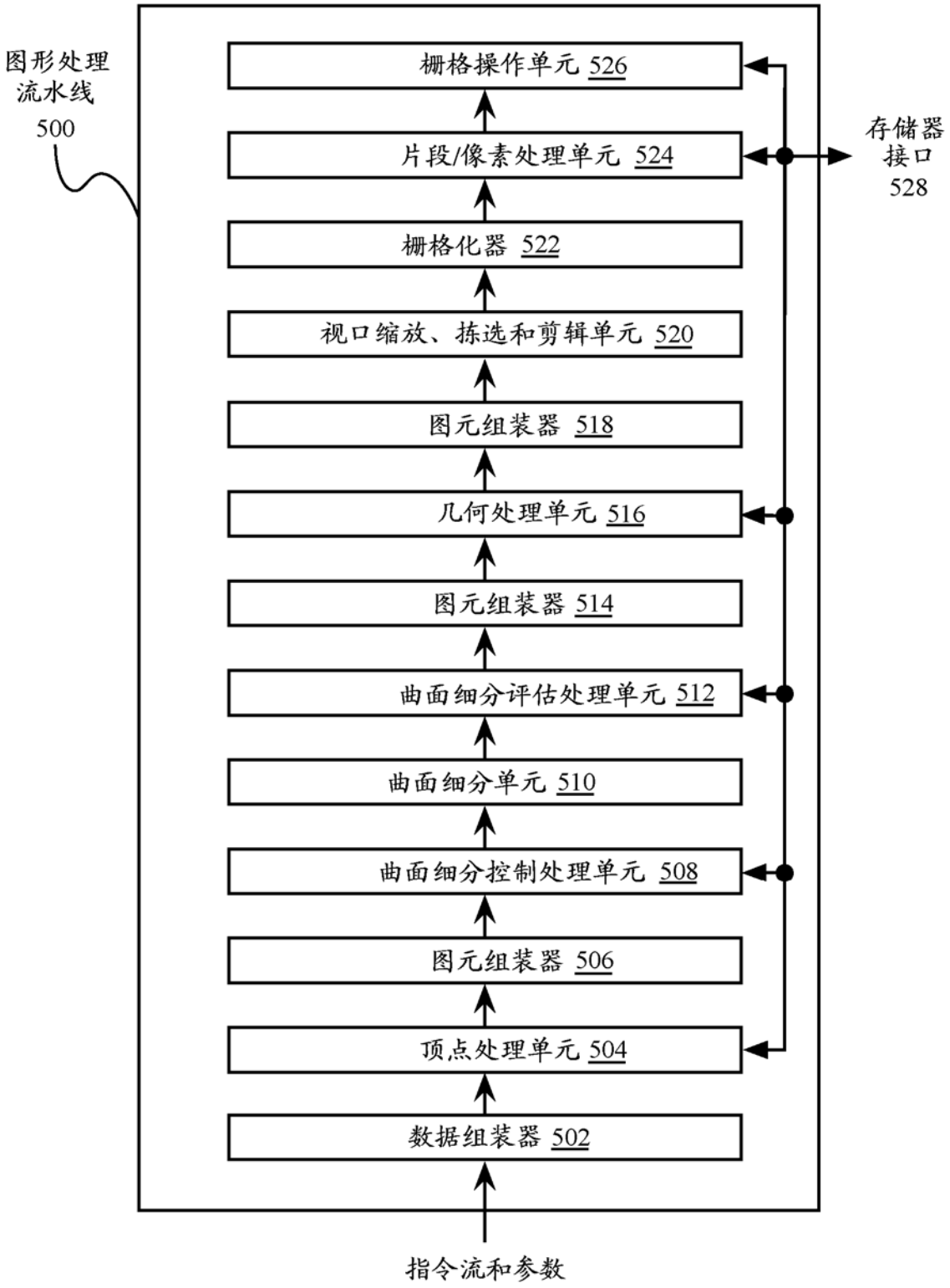

图形处理流水线
500
栅格操作单元 526
片段/像素处理单元 524
栅格化器 522
视口缩放、拣选和剪辑单元 520
图元组装器 518
几何处理单元 516
图元组装器 514
曲面细分评估处理单元 512
曲面细分单元 510
曲面细分控制处理单元 508
图元组装器 506
顶点处理单元 504
数据组装器 502
存储器接口
528
指令流和参数

图 5

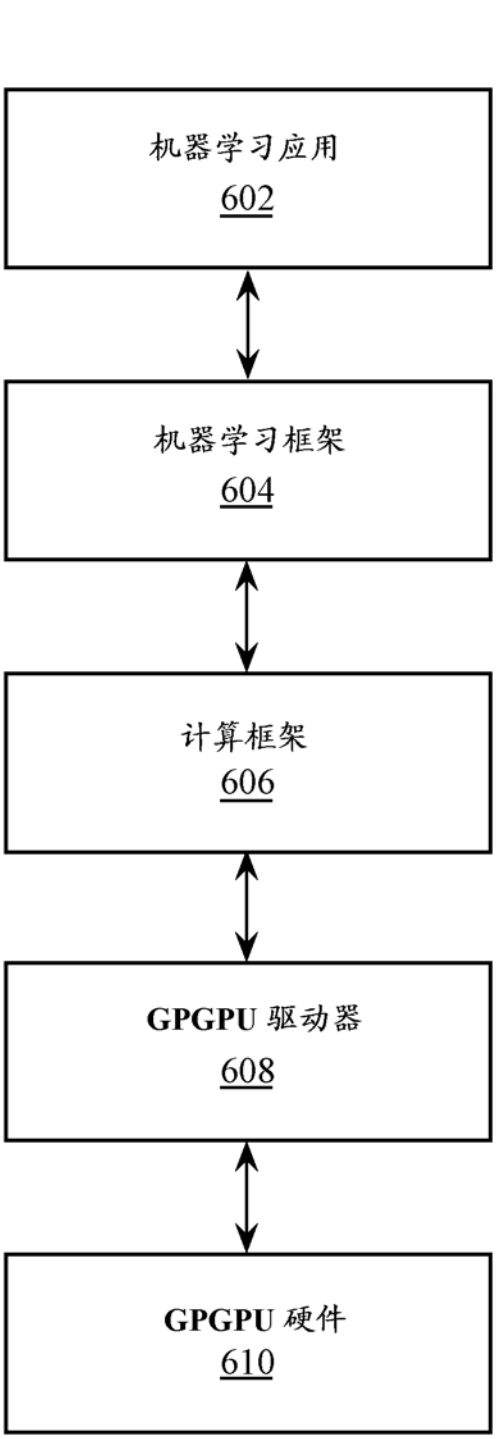


图 6

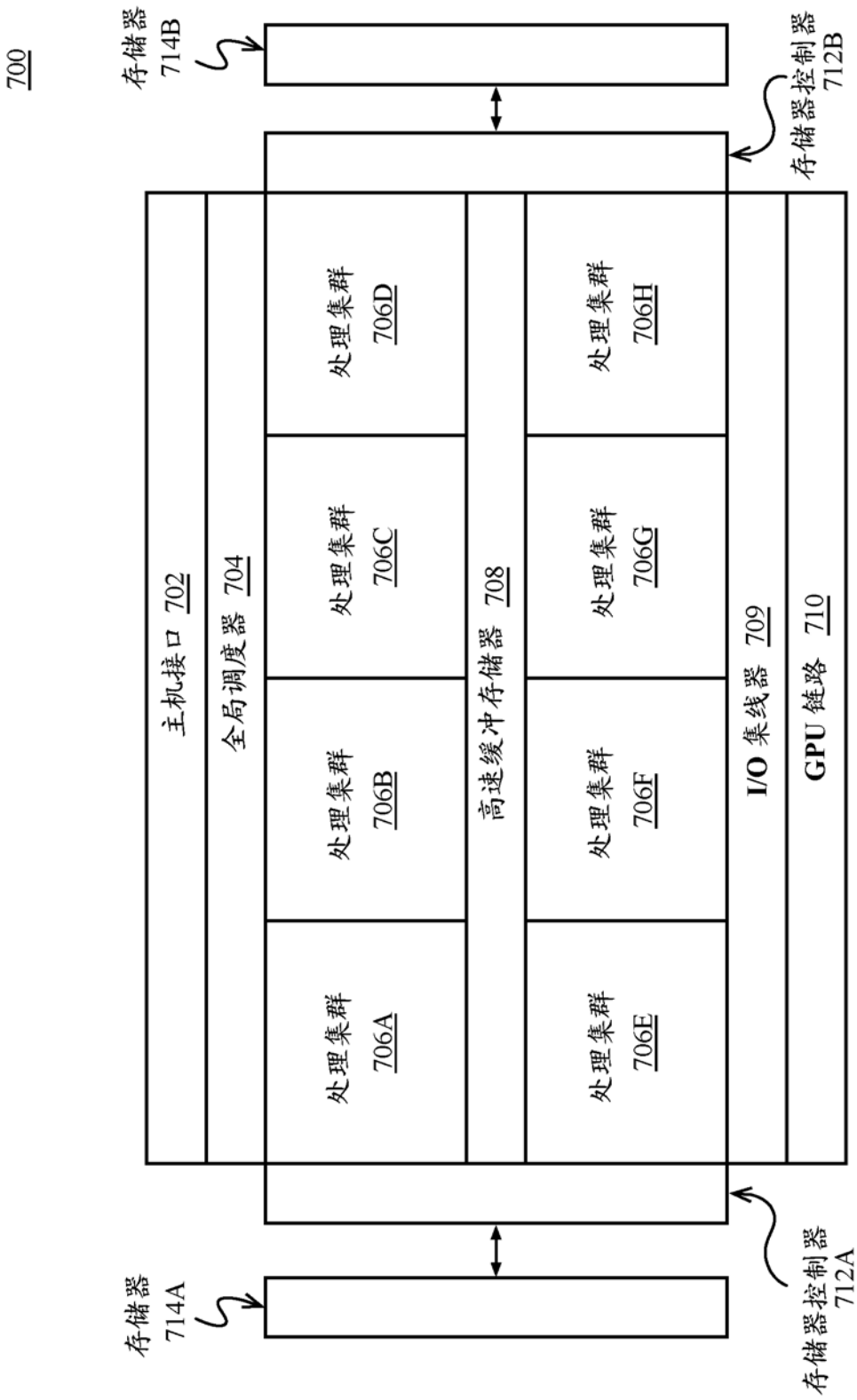

700
存储器 714B
存储器控制器 712B
主机接口 702
全局调度器 704
处理集群 706A
处理集群 706B
处理集群 706C
处理集群 706D
高速缓冲存储器 708
处理集群 706E
处理集群 706F
处理集群 706G
处理集群 706H
I/O 集线器 709
GPU 链路 710
存储器 714A
存储器控制器 712A

图 7

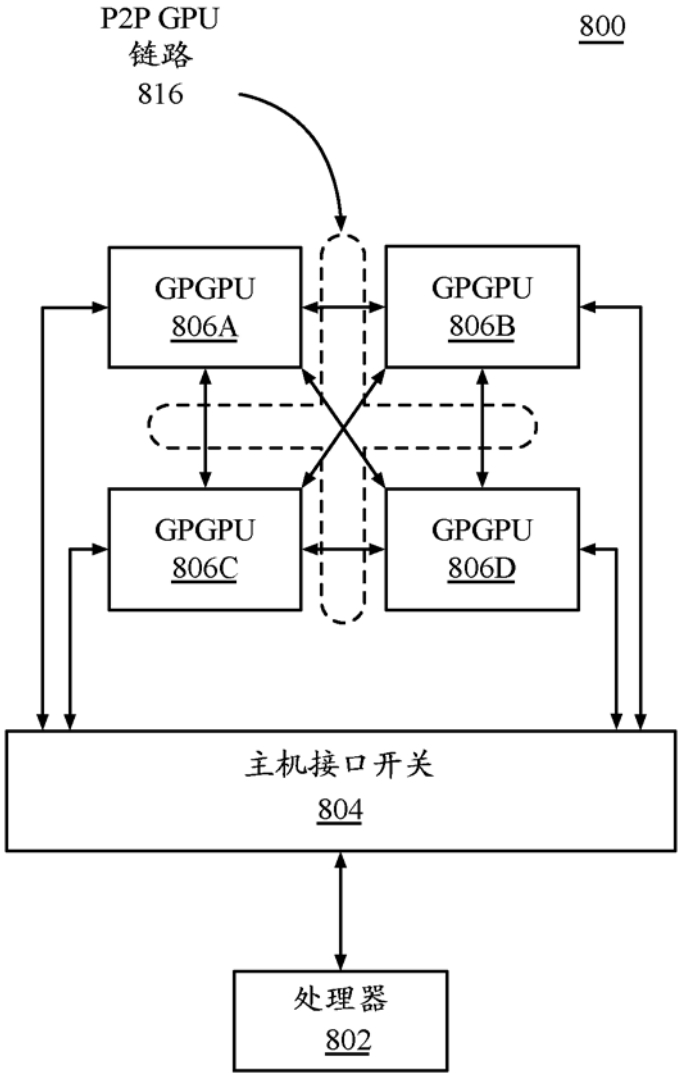

P2P GPU
链路
816
800
GPGPU
806A
GPGPU
806B
GPGPU
806C
GPGPU
806D
主机接口开关
804
处理器
802

图 8

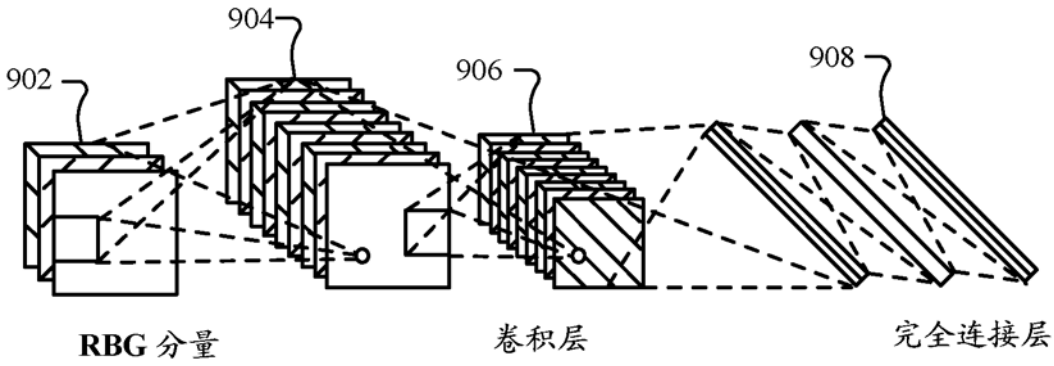

902
904
906
908
RBG 分量
卷积层
完全连接层

图 9A

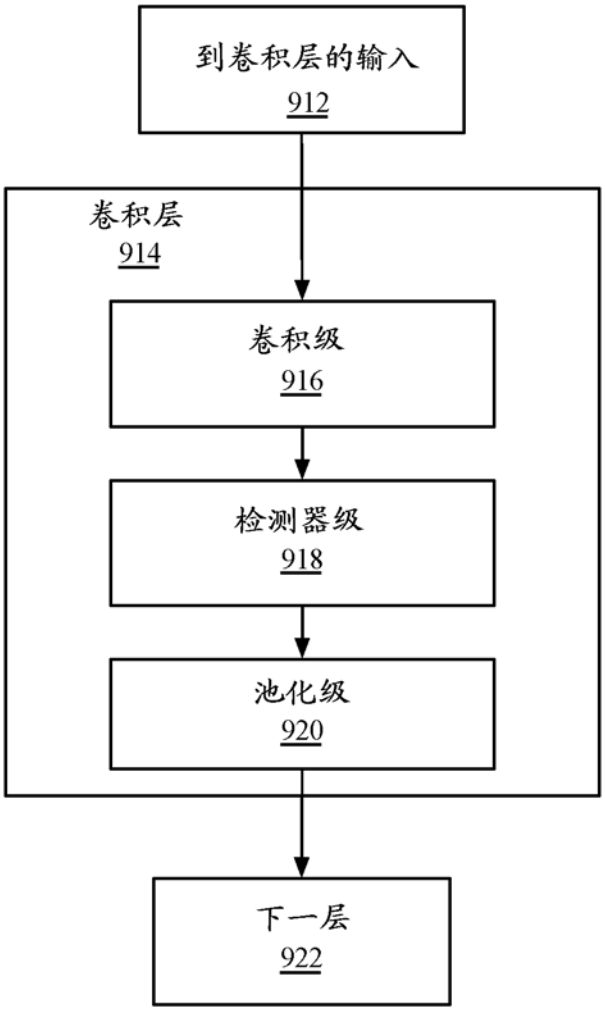


图 9B

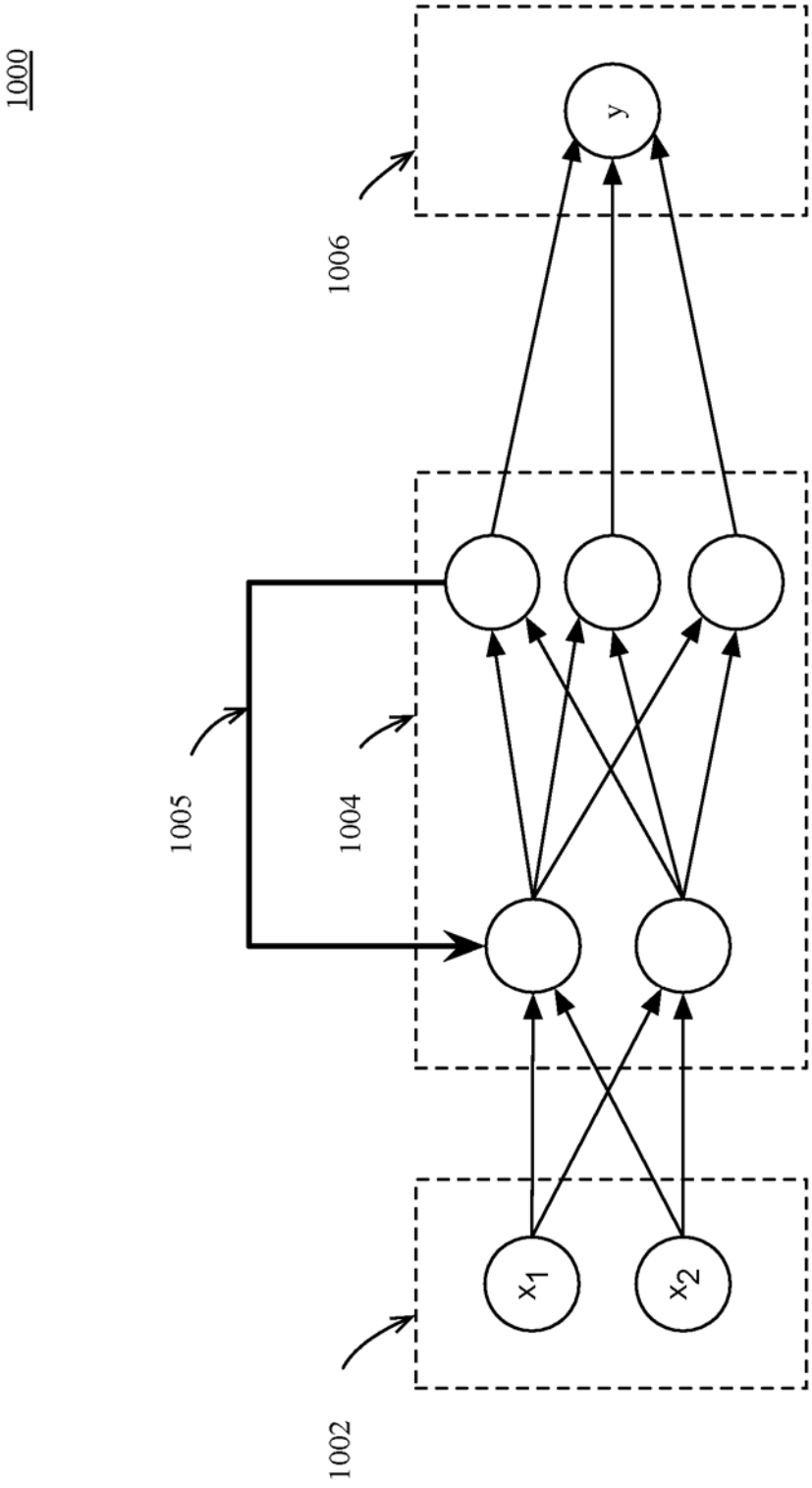


图 10

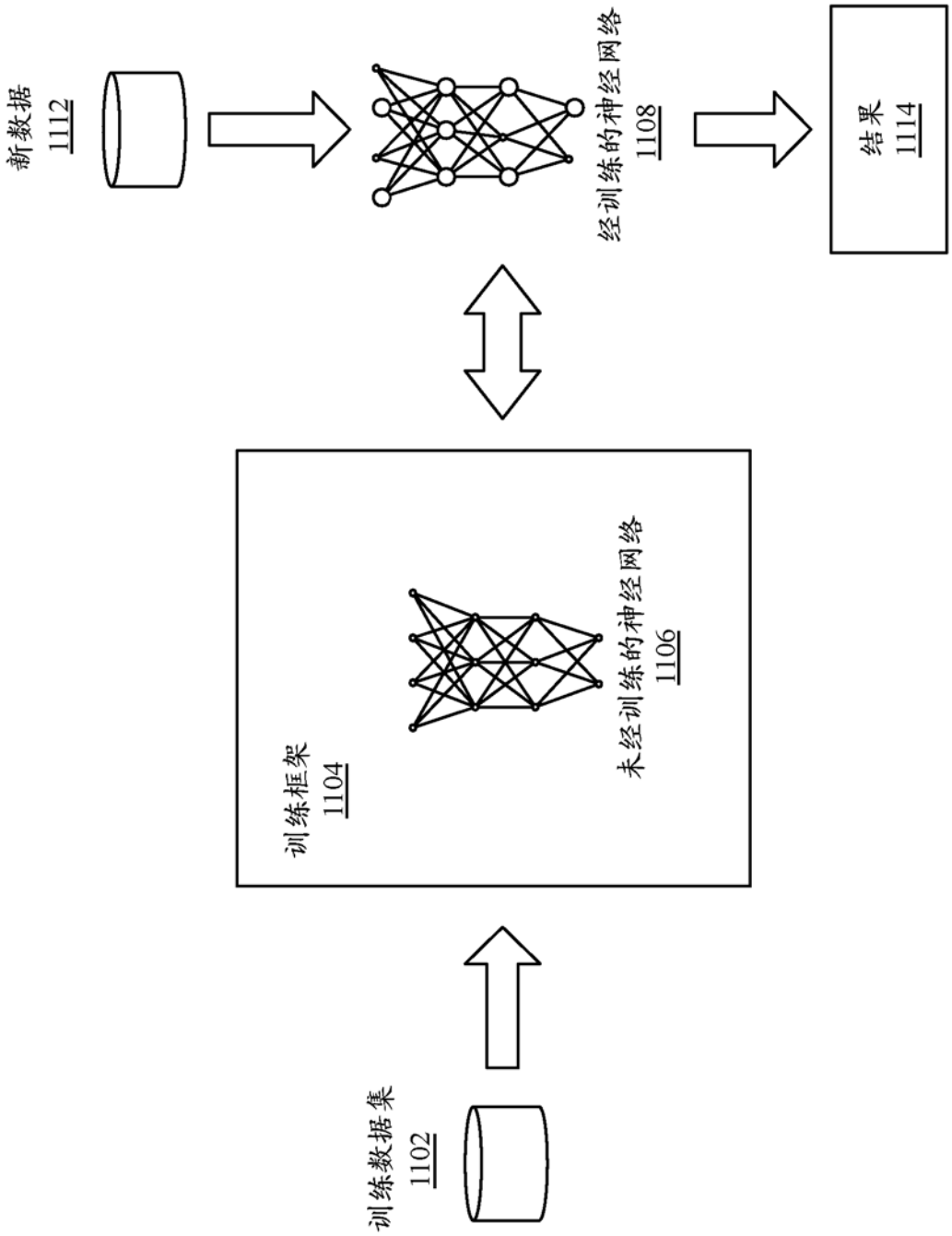

新数据
1112
经训练的神经网络
1108
结果
1114
训练框架
1104
未经训练的神经网络
1106
训练数据集
1102

图 11

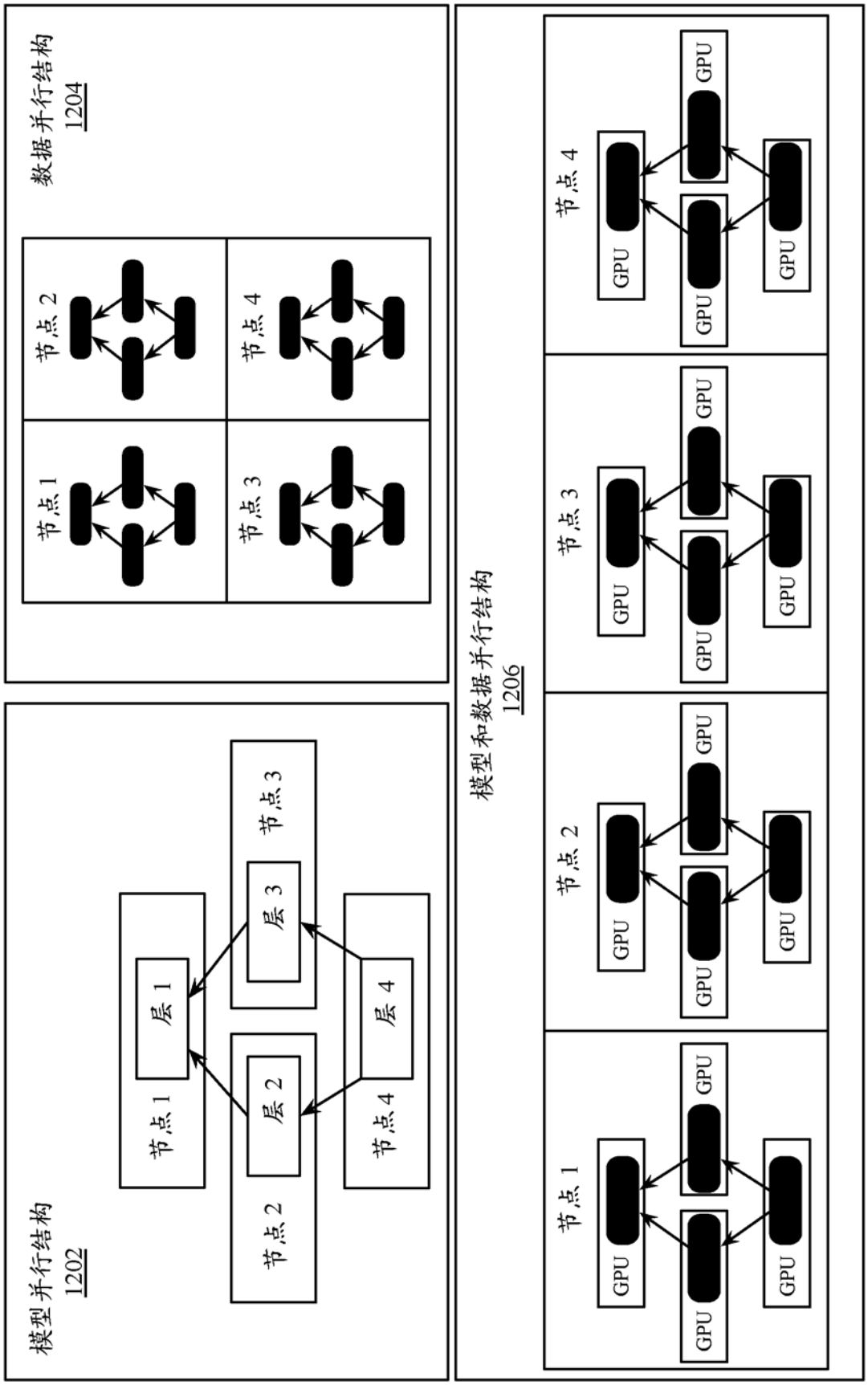

模型并行结构
1202
节点 1
层 1
节点 2
层 2
节点 3
层 3
节点 4
层 4
数据并行结构
1204
节点 1
节点 2
节点 3
节点 4
模型和数据并行结构
1206
GPU

图 12

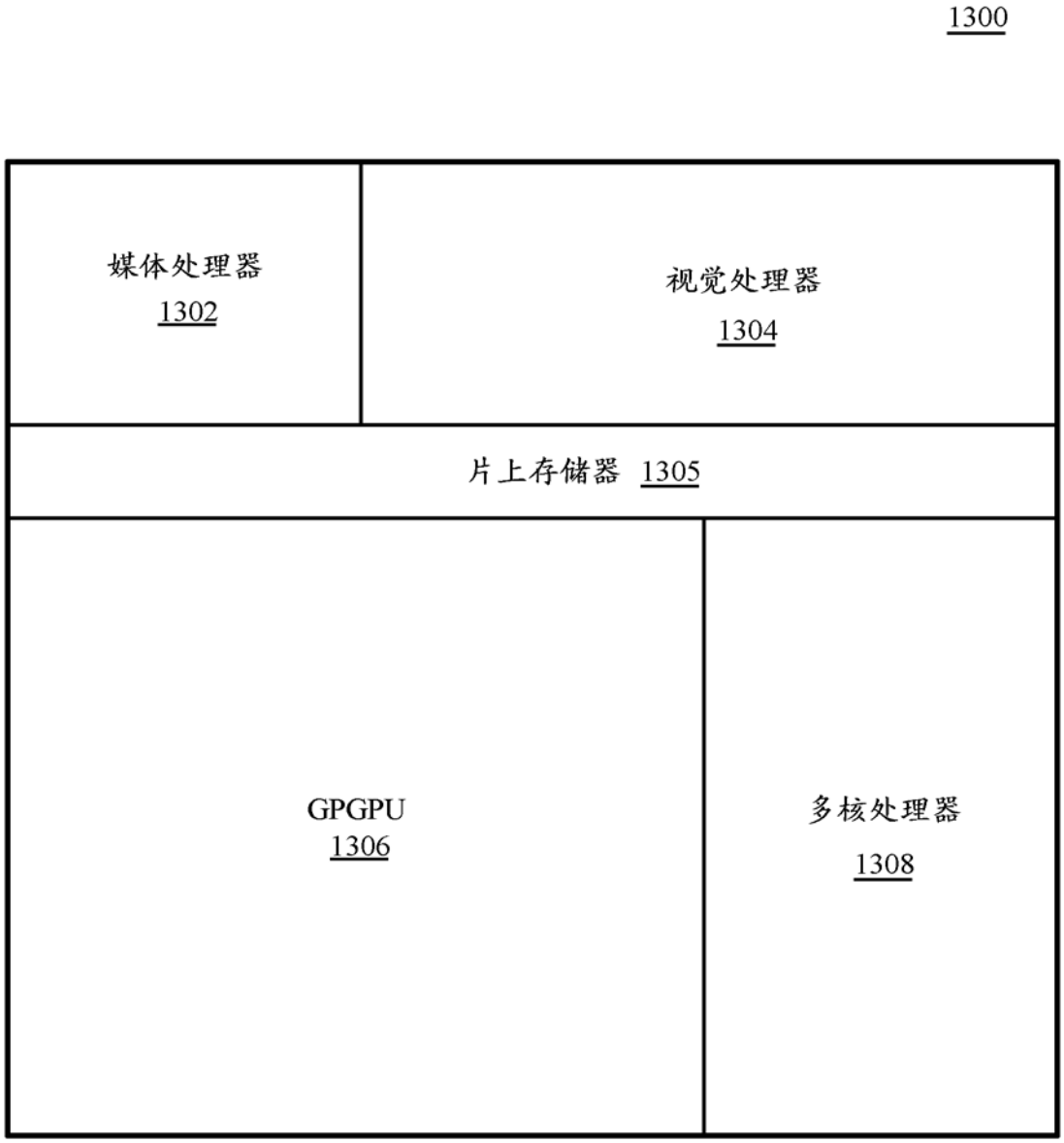


图 13

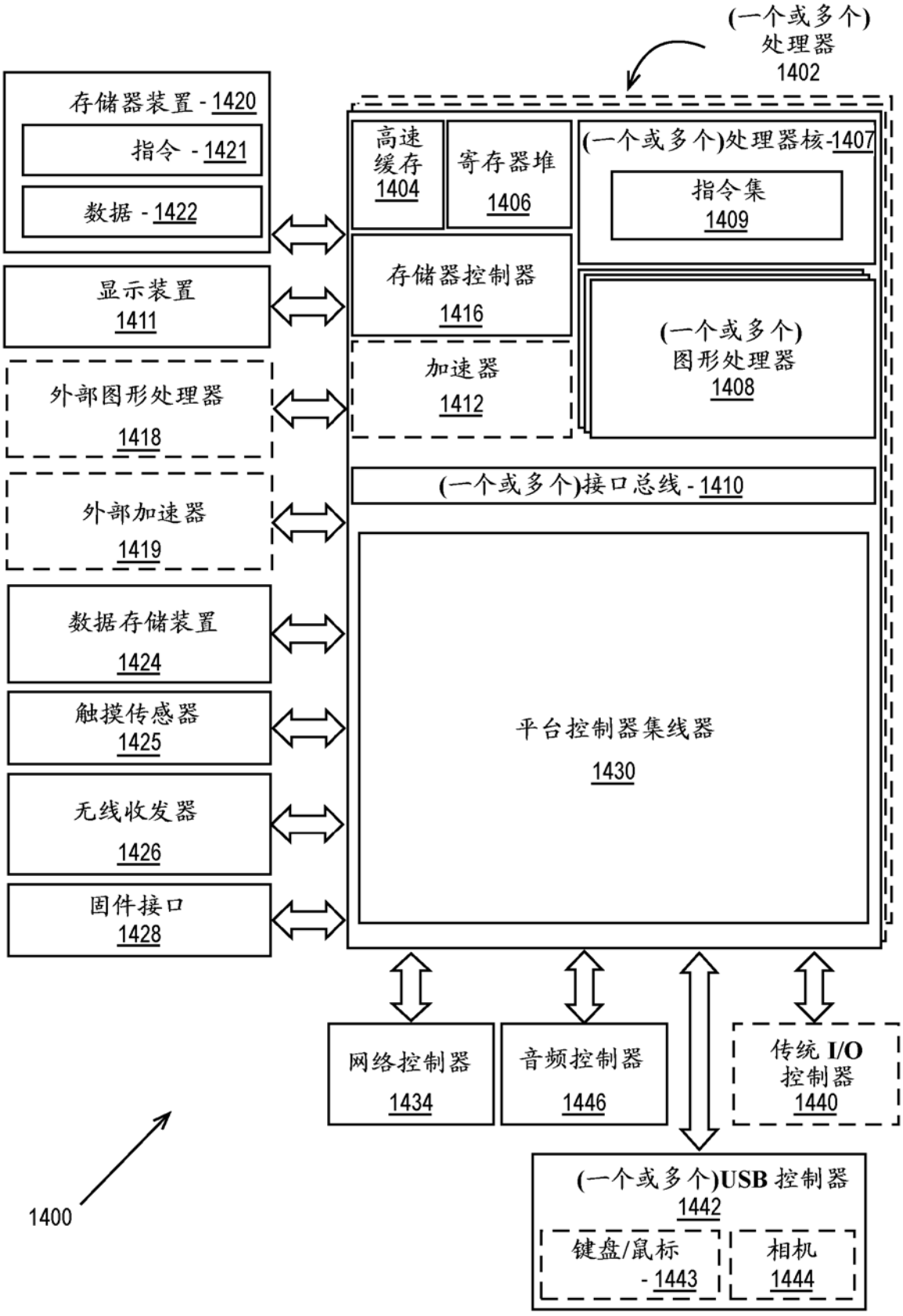

(一个或多个)
处理器
1402
存储器装置 - 1420
指令 - 1421
数据 - 1422
高速
缓存
1404
寄存器堆
1406
(一个或多个)处理器核-1407
指令集
1409
显示装置
1411
存储器控制器
1416
(一个或多个)
图形处理器
1408
加速器
1412
外部图形处理器
1418
(一个或多个)接口总线 - 1410
外部加速器
1419
数据存储装置
1424
平台控制器集线器
1430
触摸传感器
1425
无线收发器
1426
固件接口
1428
网络控制器
1434
音频控制器
1446
传统 I/O
控制器
1440
(一个或多个)USB 控制器
1442
键盘/鼠标
- 1443
相机
1444
1400

图 14

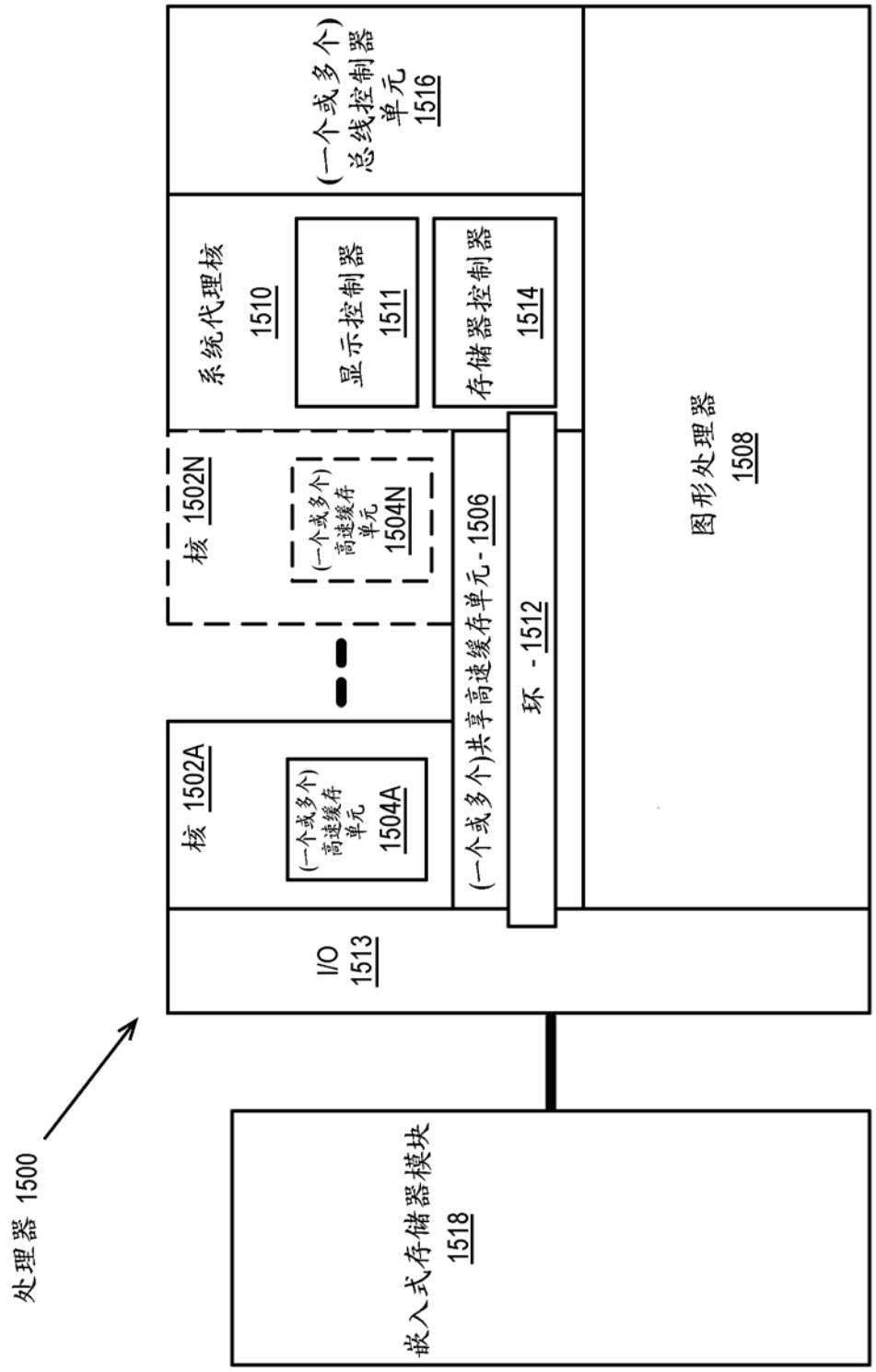

处理器 1500
嵌入式存储器模块 1518
I/O 1513
核 1502A
(一个或多个)高速缓存单元 1504A
核 1502N
(一个或多个)高速缓存单元 1504N
(一个或多个)共享高速缓存单元 - 1506
环 - 1512
系统代理核 1510
显示控制器 1511
存储器控制器 1514
(一个或多个)总线控制器单元 1516
图形处理器 1508

图 15A

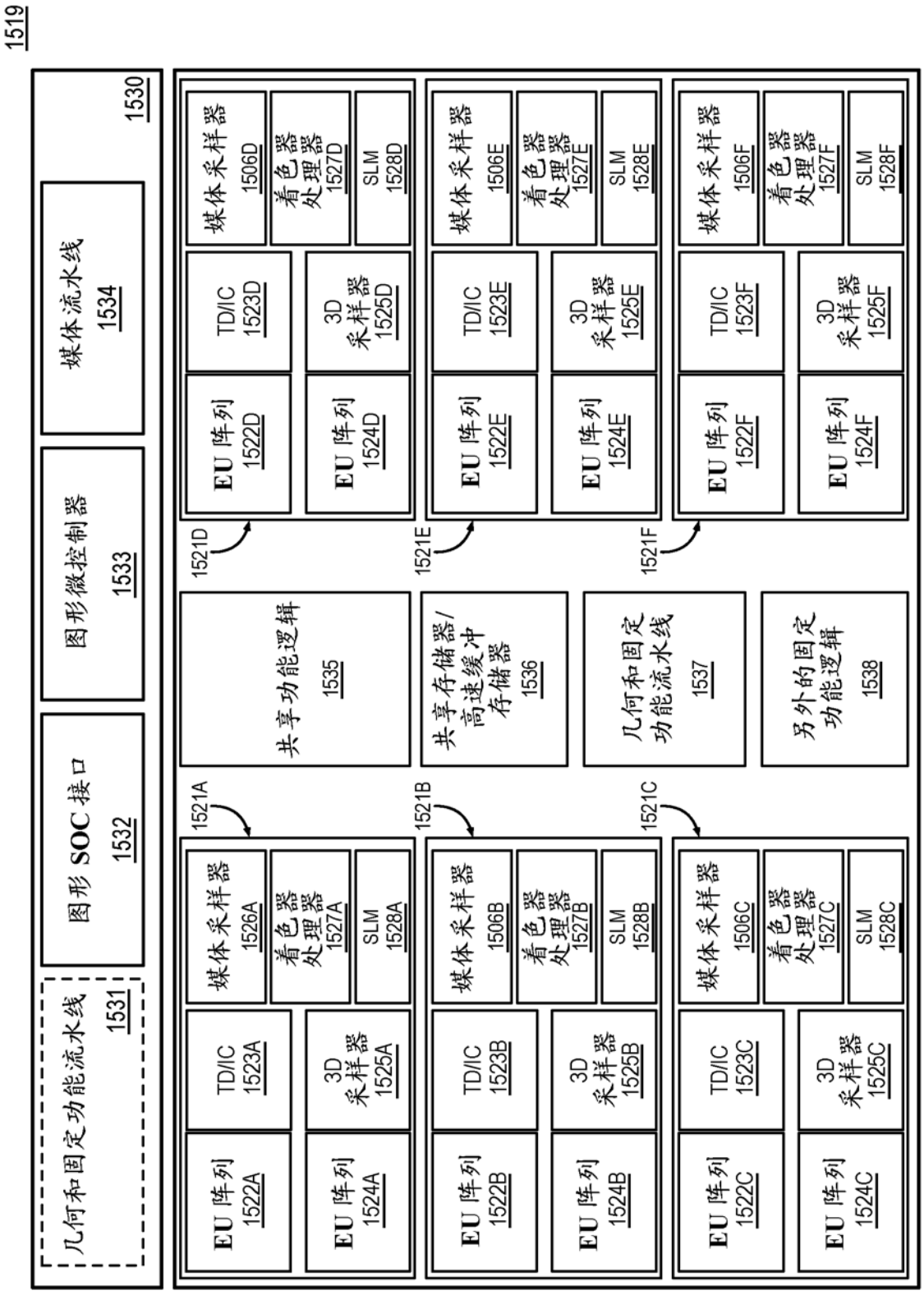

1519
几何和固定功能流水线 1531
图形 SOC 接口 1532
图形微控制器 1533
媒体流水线 1534
1530
共享功能逻辑 1535
共享存储器/高速缓冲存储器 1536
几何和固定功能流水线 1537
另外的固定功能逻辑 1538
1521A
1521B
1521C
1521D
1521E
1521F
EU 阵列 1522A
TD/IC 1523A
3D 采样器 1525A
EU 阵列 1524A
媒体采样器 1526A
着色器处理器 1527A
SLM 1528A
EU 阵列 1522B
TD/IC 1523B
3D 采样器 1525B
EU 阵列 1524B
媒体采样器 1506B
着色器处理器 1527B
SLM 1528B
EU 阵列 1522C
TD/IC 1523C
3D 采样器 1525C
EU 阵列 1524C
媒体采样器 1506C
着色器处理器 1527C
SLM 1528C
EU 阵列 1522D
TD/IC 1523D
3D 采样器 1525D
EU 阵列 1524D
媒体采样器 1506D
着色器处理器 1527D
SLM 1528D
EU 阵列 1522E
TD/IC 1523E
3D 采样器 1525E
EU 阵列 1524E
媒体采样器 1506E
着色器处理器 1527E
SLM 1528E
EU 阵列 1522F
TD/IC 1523F
3D 采样器 1525F
EU 阵列 1524F
媒体采样器 1506F
着色器处理器 1527F
SLM 1528F

图 15B

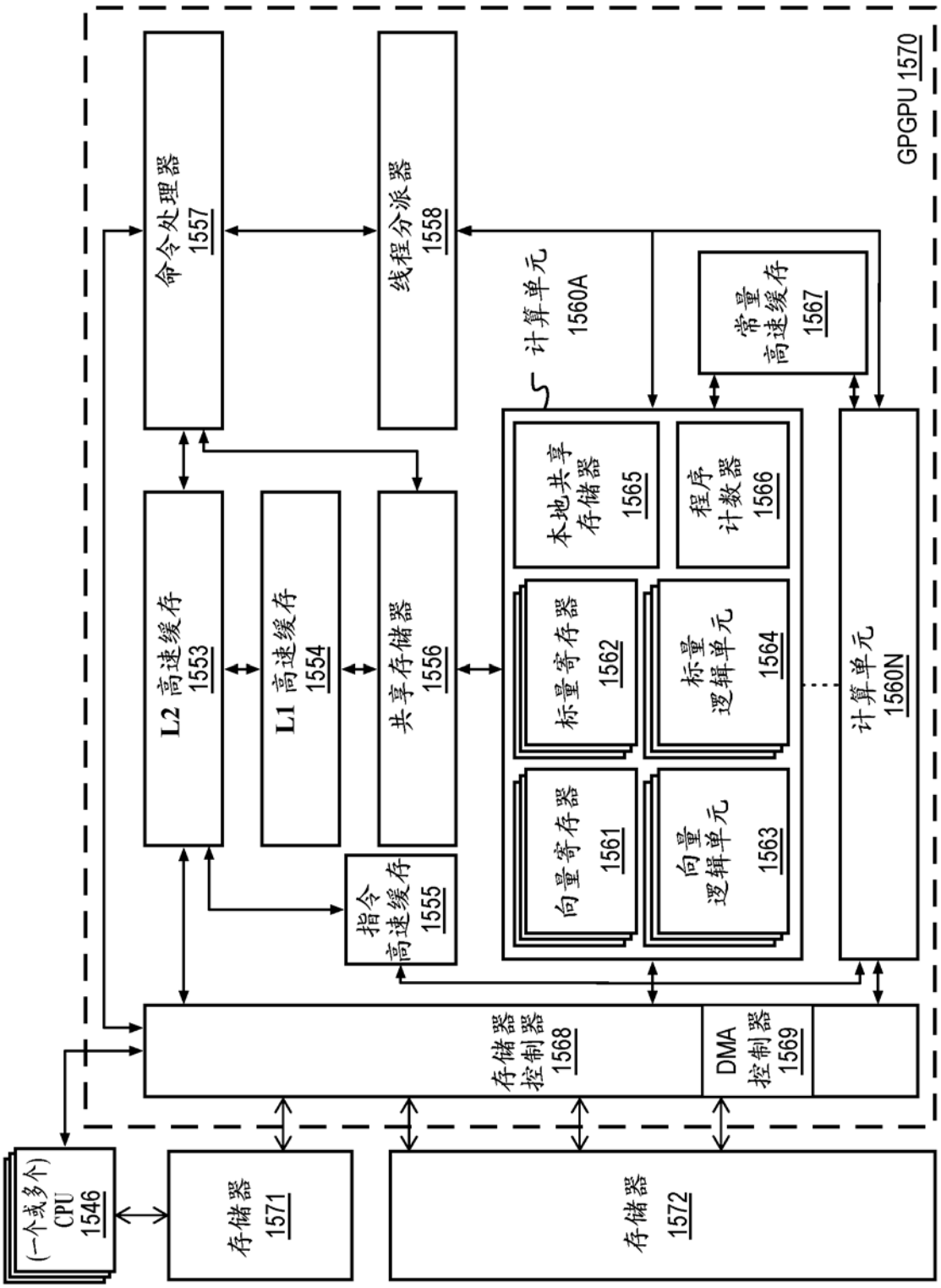

GPGPU 1570
命令处理器 1557
线程分派器 1558
计算单元 1560A
常量高速缓存 1567
本地共享存储器 1565
程序计数器 1566
L2 高速缓存 1553
L1 高速缓存 1554
共享存储器 1556
标量寄存器 1562
标量逻辑单元 1564
计算单元 1560N
向量寄存器 1561
向量逻辑单元 1563
指令高速缓存 1555
存储器控制器 1568
DMA 控制器 1569
(一个或多个) CPU 1546
存储器 1571
存储器 1572

图 15C

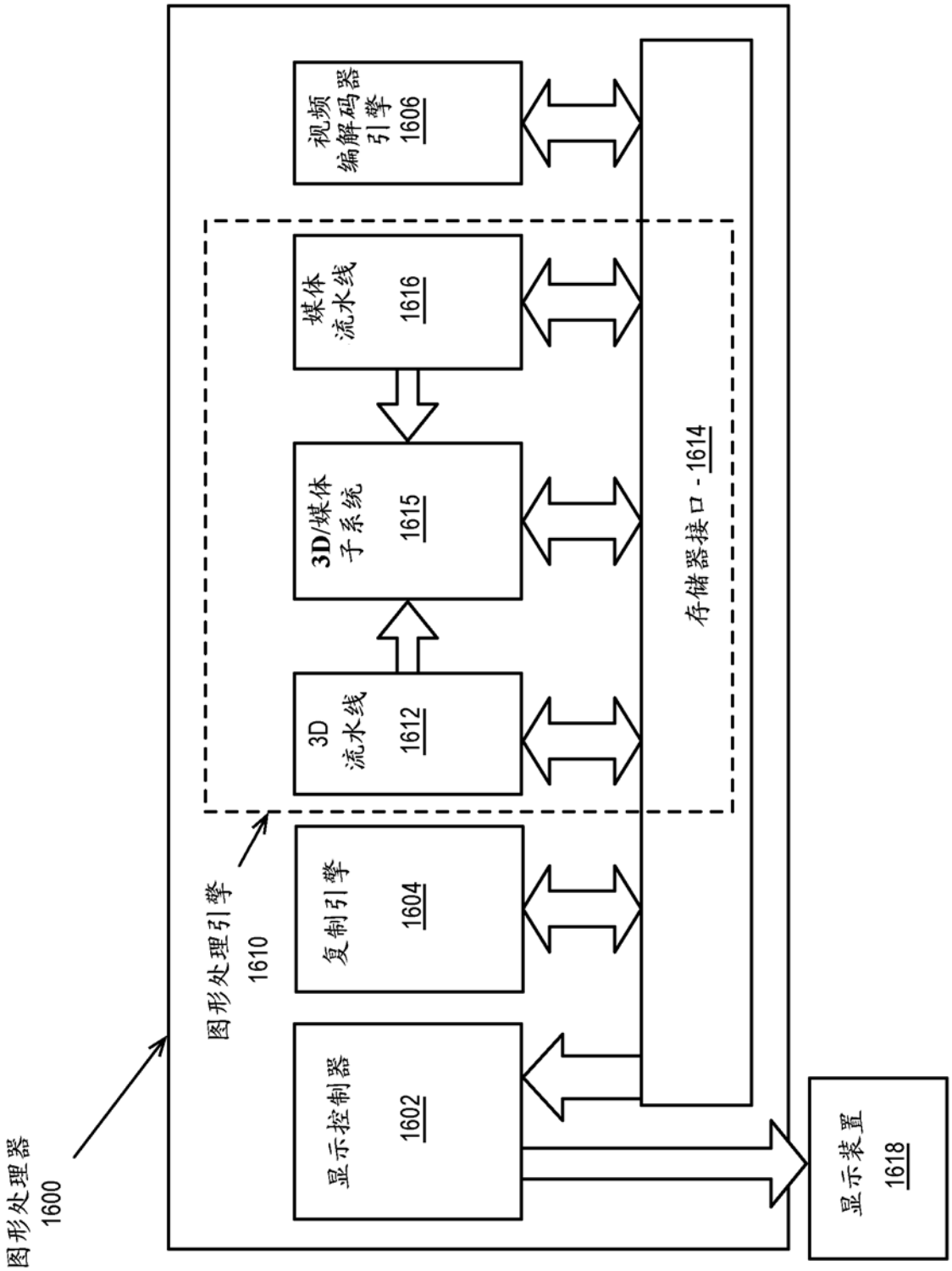

图形处理器
1600
显示控制器
1602
图形处理引擎
1610
复制引擎
1604
3D
流水线
1612
3D/媒体
子系统
1615
媒体
流水线
1616
视频
编解码器
引擎
1606
存储器接口 - 1614
显示装置
1618

图 16A

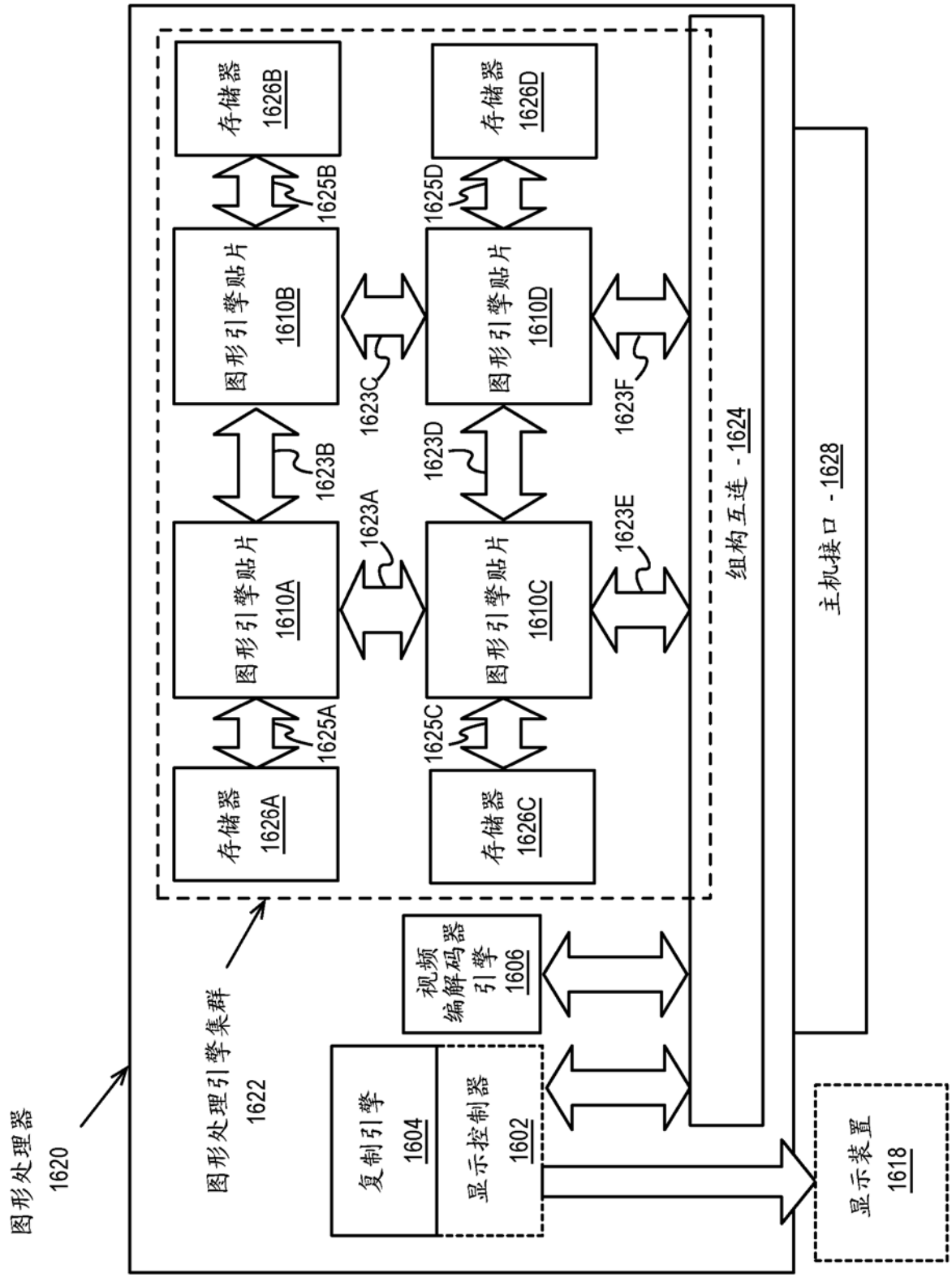

图形处理器
1620
图形处理引擎集群
1622
复制引擎
1604
显示控制器
1602
视频编解码器引擎
1606
图形引擎贴片
1610A
图形引擎贴片
1610B
图形引擎贴片
1610C
图形引擎贴片
1610D
存储器
1626A
存储器
1626B
存储器
1626C
存储器
1626D
1625A
1625B
1625C
1625D
1623A
1623B
1623C
1623D
1623E
1623F
组构互连 - 1624
主机接口 - 1628
显示装置
1618

图 16B

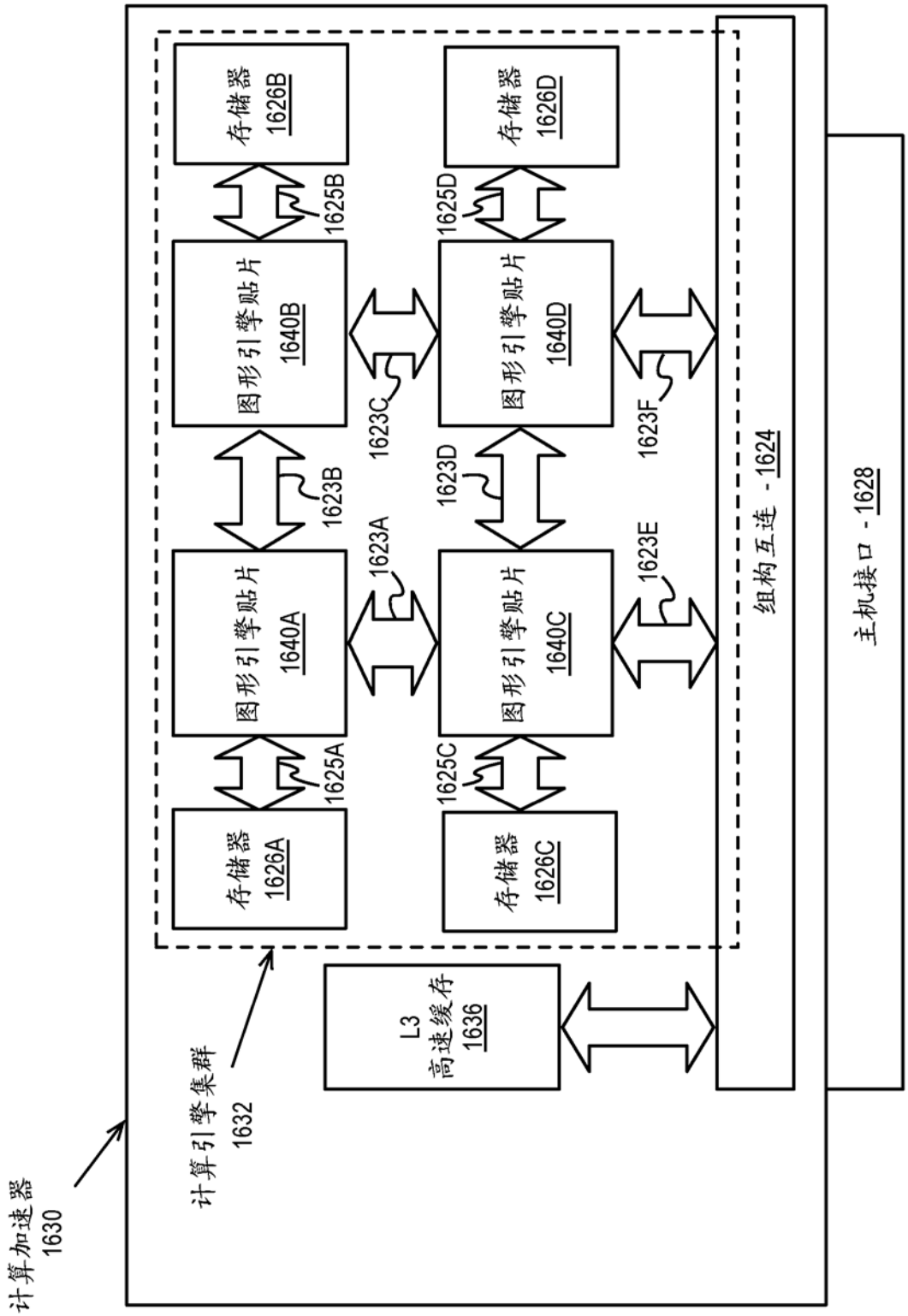

计算加速器
1630
计算引擎集群
1632
L3
高速缓存
1636
组构互连 - 1624
主机接口 - 1628
图形引擎贴片
1640A
图形引擎贴片
1640B
图形引擎贴片
1640C
图形引擎贴片
1640D
存储器
1626A
存储器
1626B
存储器
1626C
存储器
1626D
1623A
1623B
1623C
1623D
1623E
1623F
1625A
1625B
1625C
1625D

图 16C

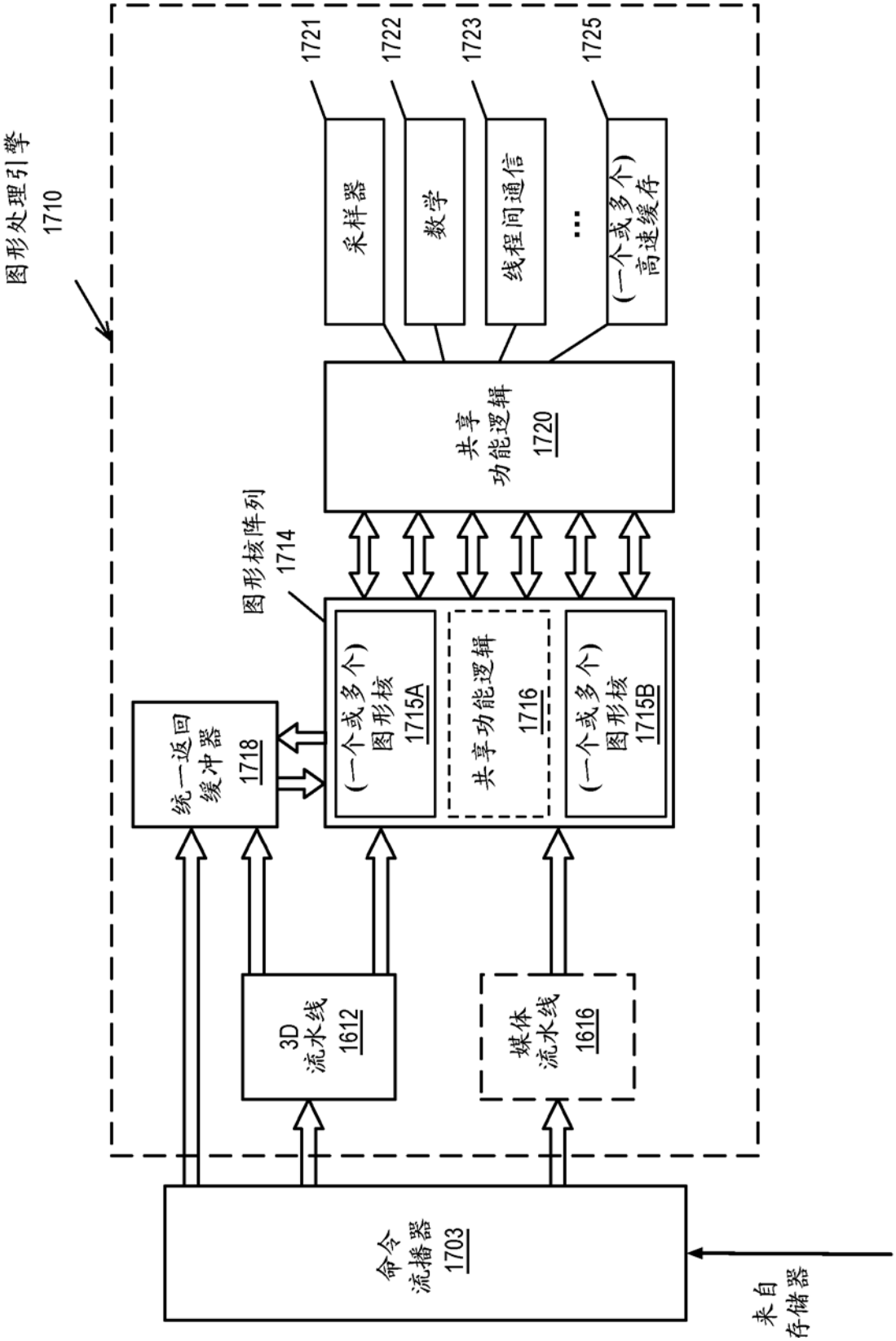

图形处理引擎
1710
1721
1722
1723
1725
采样器
数学
线程间通信
...
(一个或多个)
高速缓存
共享
功能逻辑
1720
图形核阵列
1714
统一返回
缓冲器
1718
(一个或多个)
图形核
1715A
共享功能逻辑
1716
(一个或多个)
图形核
1715B
3D
流水线
1612
媒体
流水线
1616
命令
流播器
1703
来自
存储器

图 17

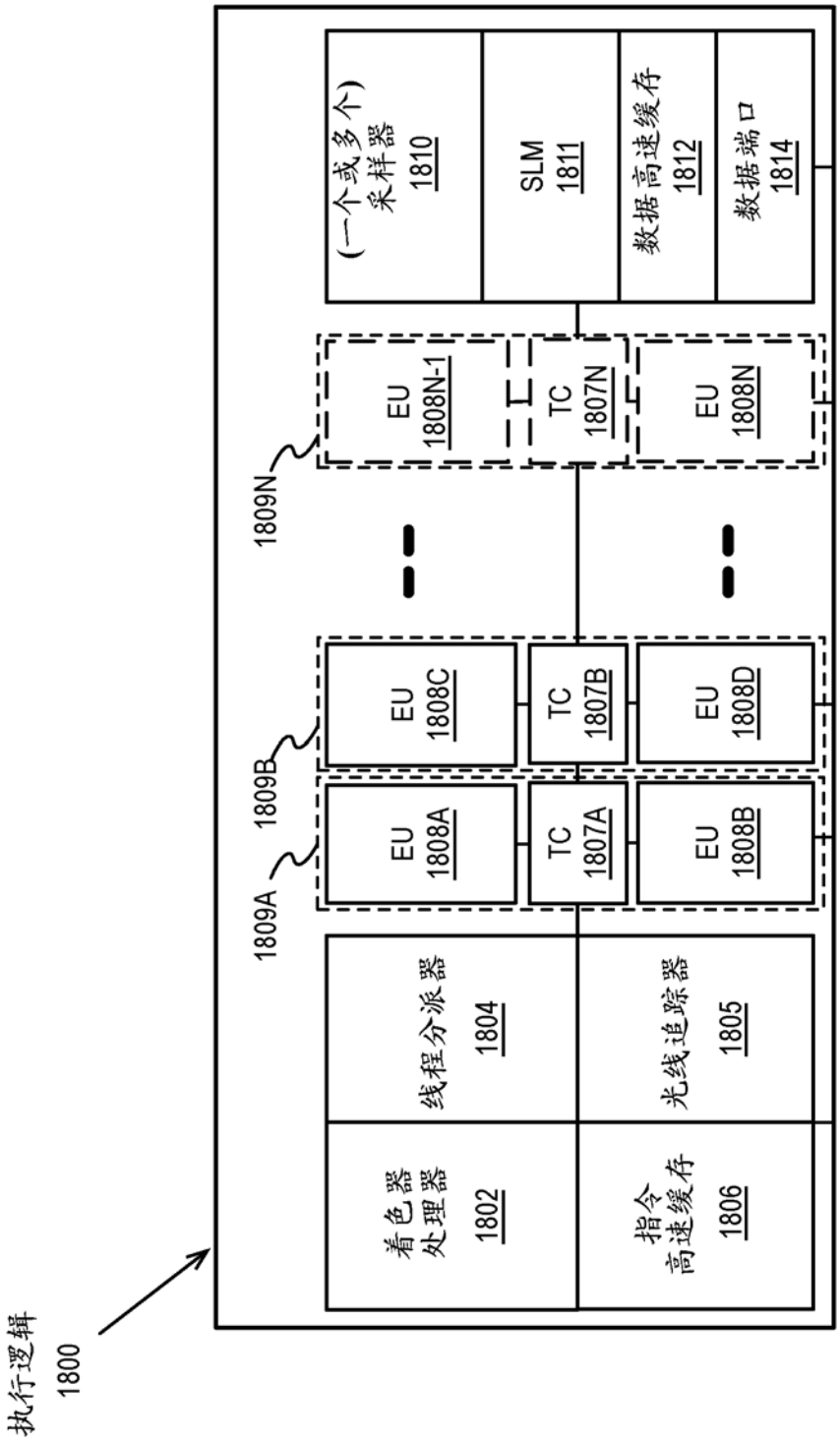

执行逻辑
1800
着色器处理器
1802
线程分派器
1804
光线追踪器
1805
指令高速缓存
1806
(一个或多个)采样器
1810
SLM
1811
数据高速缓存
1812
数据端口
1814
1809A
EU 1808A
TC 1807A
EU 1808B
1809B
EU 1808C
TC 1807B
EU 1808D
1809N
EU 1808N-1
TC 1807N
EU 1808N

图 18A

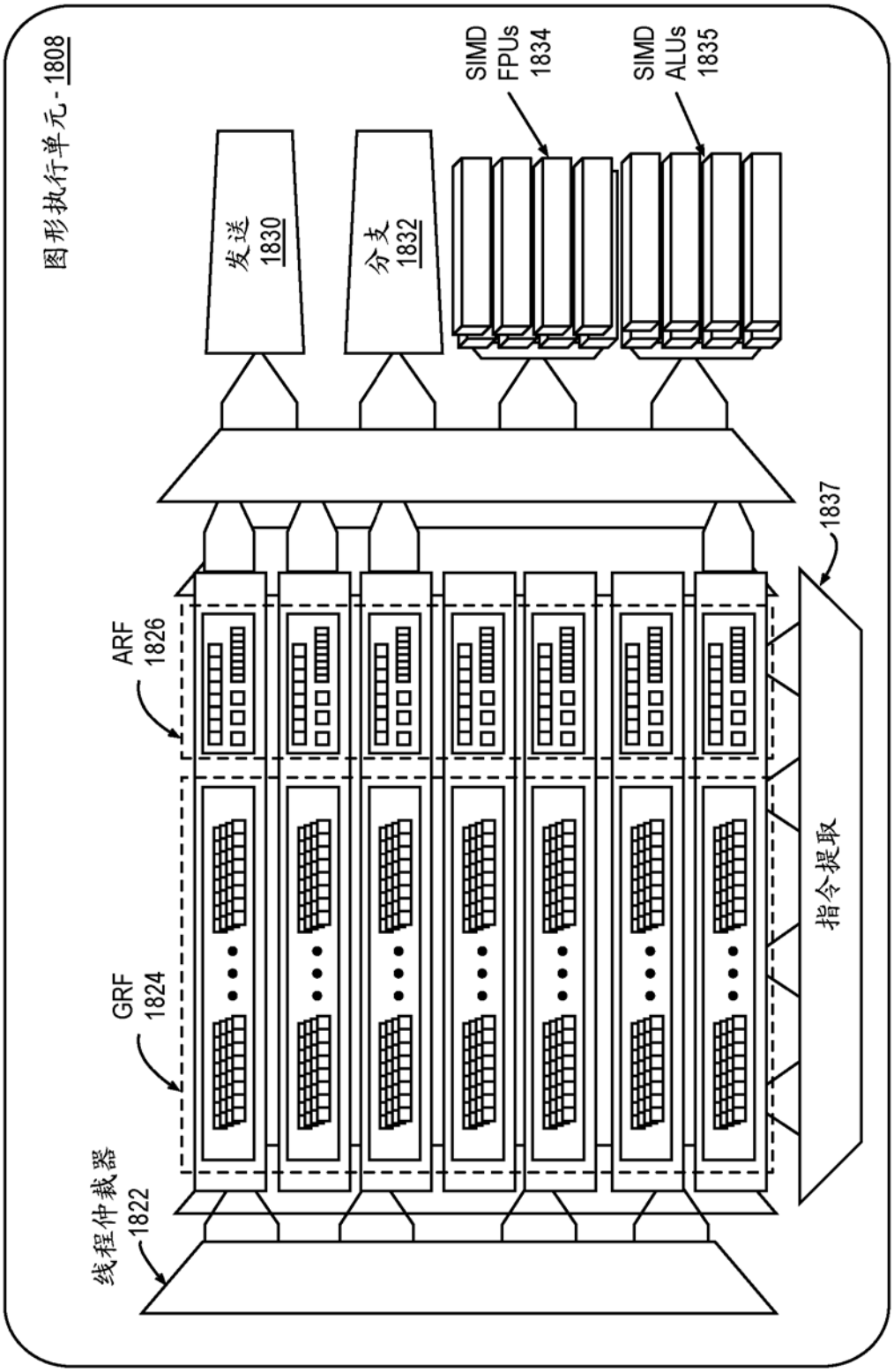

图形执行单元-1808
发送 1830
分支 1832
SIMD FPUs 1834
SIMD ALUs 1835
1837
ARF 1826
GRF 1824
线程仲裁器 1822
指令提取

图 18B

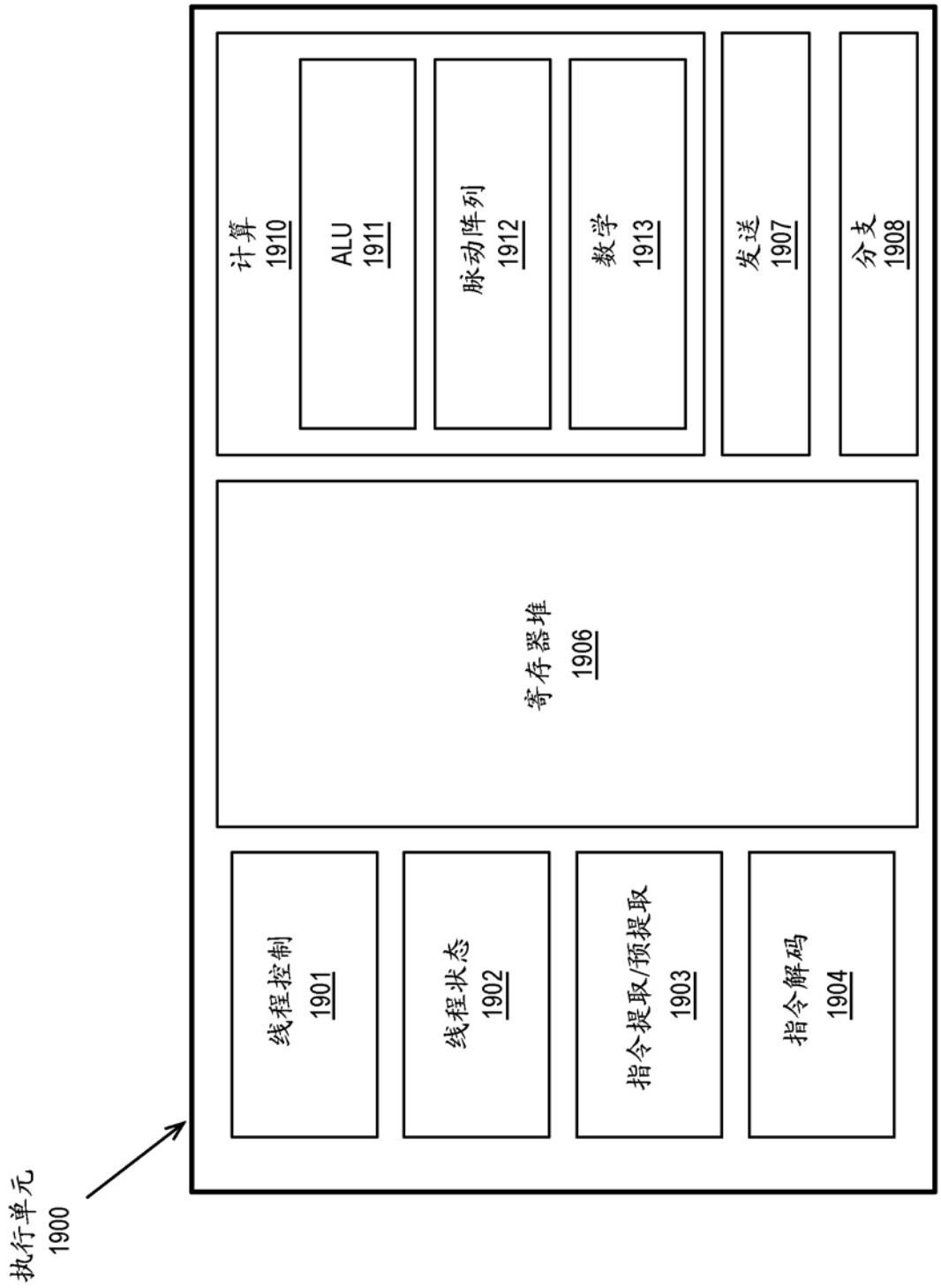

执行单元
1900
线程控制
1901
线程状态
1902
指令提取/预提取
1903
指令解码
1904
寄存器堆
1906
计算
1910
ALU
1911
脉动阵列
1912
数学
1913
发送
1907
分支
1908

图 19

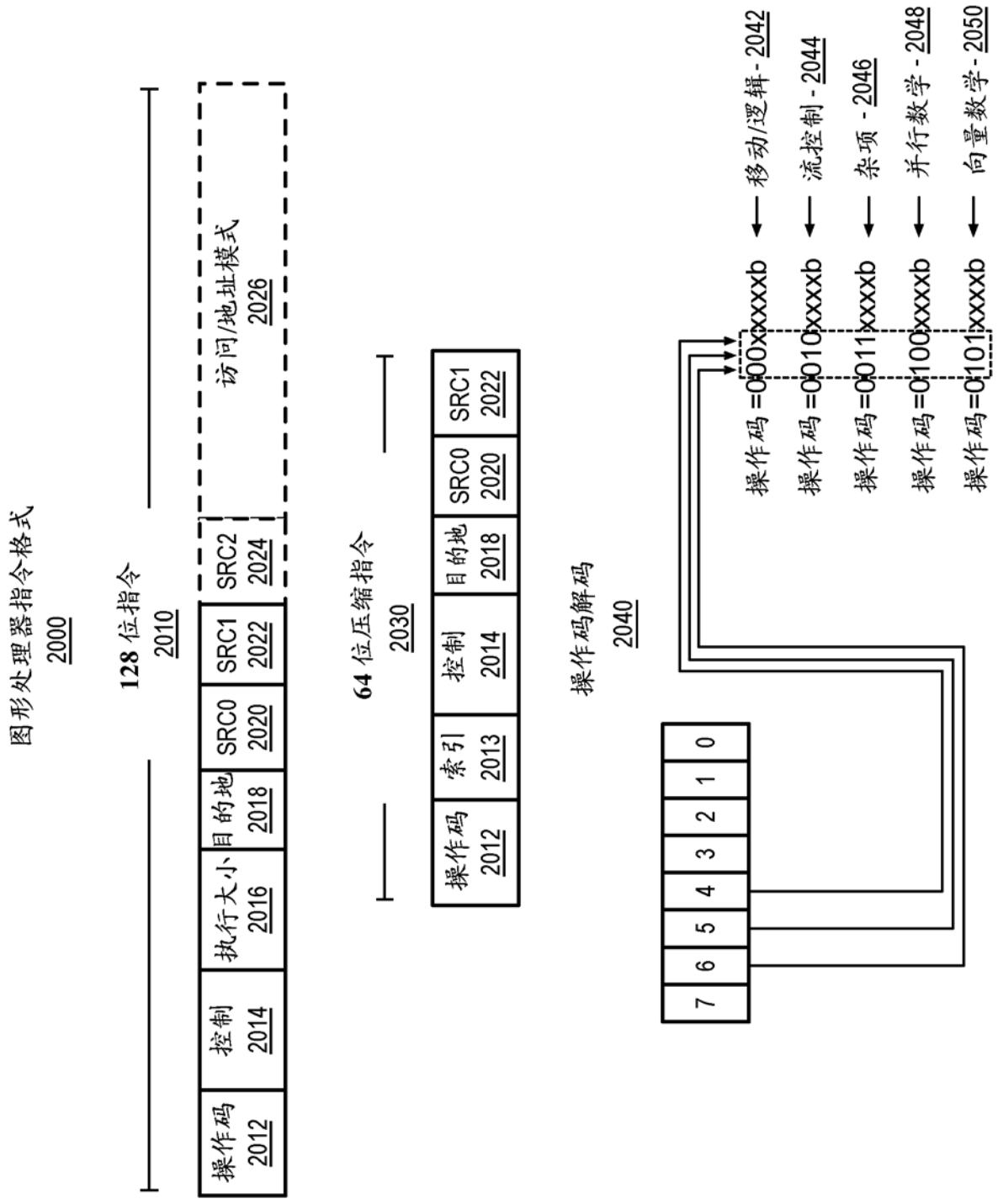

图形处理器指令格式
2000
128 位指令
2010
操作码 2012
控制 2014
执行大小 2016
目的地 2018
SRC0 2020
SRC1 2022
SRC2 2024
访问/地址模式 2026
64 位压缩指令
2030
操作码 2012
索引 2013
控制 2014
目的地 2018
SRC0 2020
SRC1 2022
操作码解码
2040
7 6 5 4 3 2 1 0
操作码 =0000xxxxb
操作码 =0010xxxxb
操作码 =0011xxxxb
操作码 =0100xxxxb
操作码 =0101xxxxb
移动/逻辑 - 2042
流控制 - 2044
杂项 - 2046
并行数学 - 2048
向量数学 - 2050

图 20

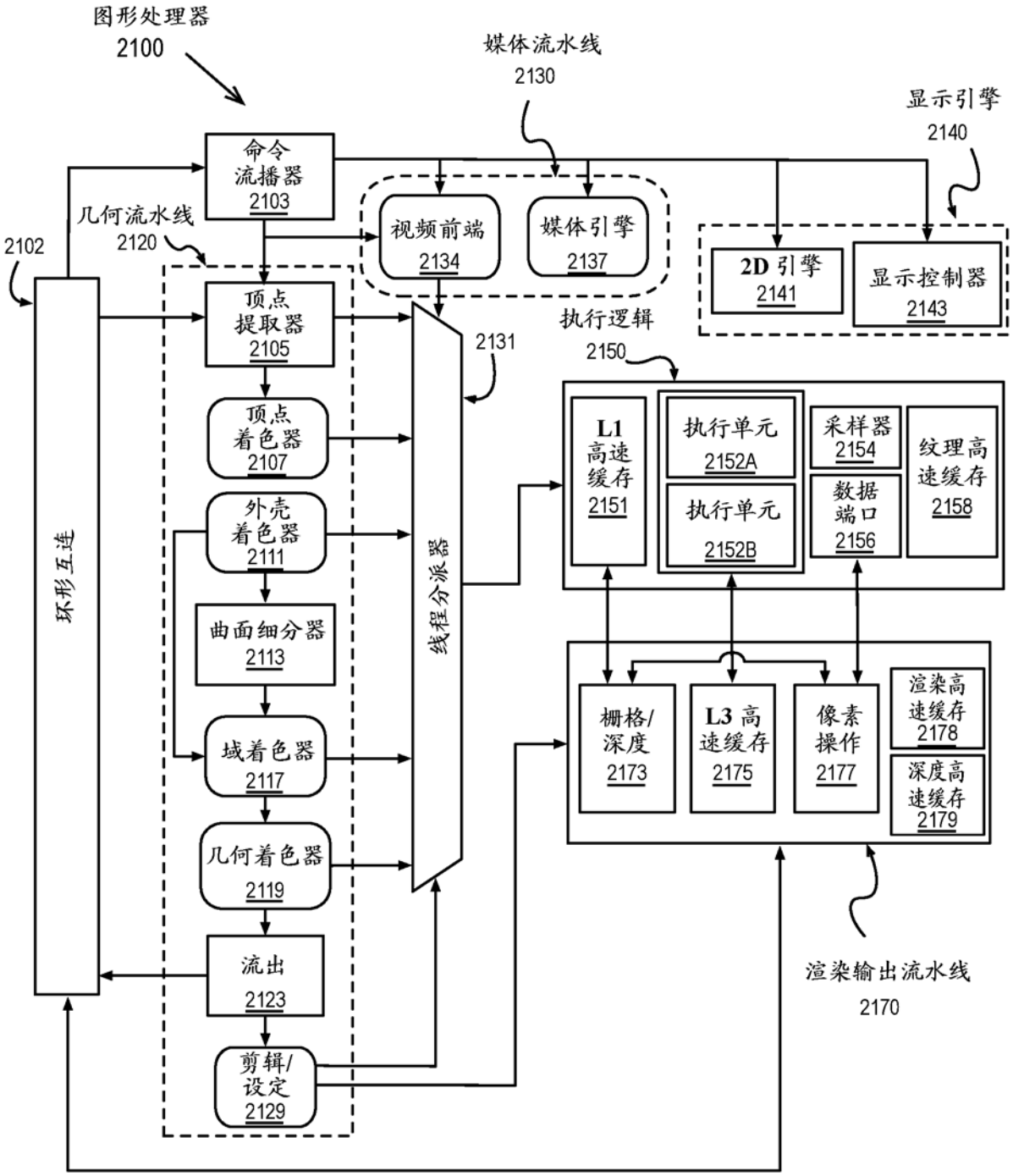

图形处理器
2100
媒体流水线
2130
显示引擎
2140
命令流播器
2103
几何流水线
2120
2102
视频前端
2134
媒体引擎
2137
2D 引擎
2141
显示控制器
2143
执行逻辑
2150
顶点提取器
2105
2131
顶点着色器
2107
外壳着色器
2111
曲面细分器
2113
域着色器
2117
几何着色器
2119
流出
2123
剪辑/设定
2129
环形互连
线程分派器
L1高速缓存
2151
执行单元
2152A
执行单元
2152B
采样器
2154
数据端口
2156
纹理高速缓存
2158
栅格/深度
2173
L3 高速缓存
2175
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

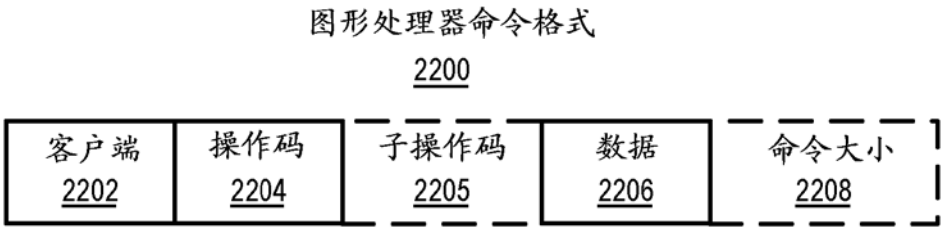

图形处理器命令格式
2200
客户端
2202
操作码
2204
子操作码
2205
数据
2206
命令大小
2208

图 22A

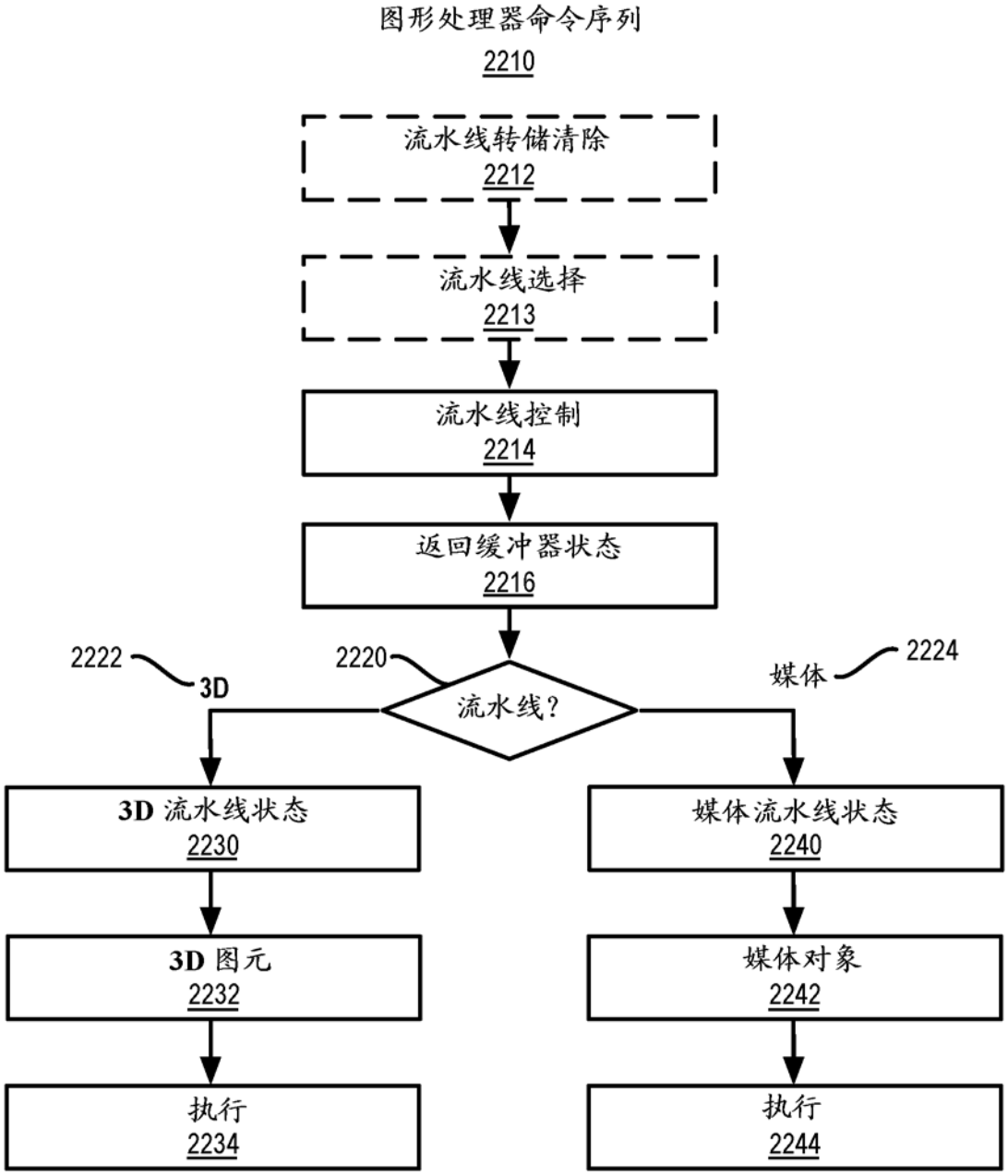

图形处理器命令序列
2210
流水线转储清除
2212
流水线选择
2213
流水线控制
2214
返回缓冲器状态
2216
2222
3D
2220
流水线?
媒体
2224
3D 流水线状态
2230
3D 图元
2232
执行
2234
媒体流水线状态
2240
媒体对象
2242
执行
2244

图 22B

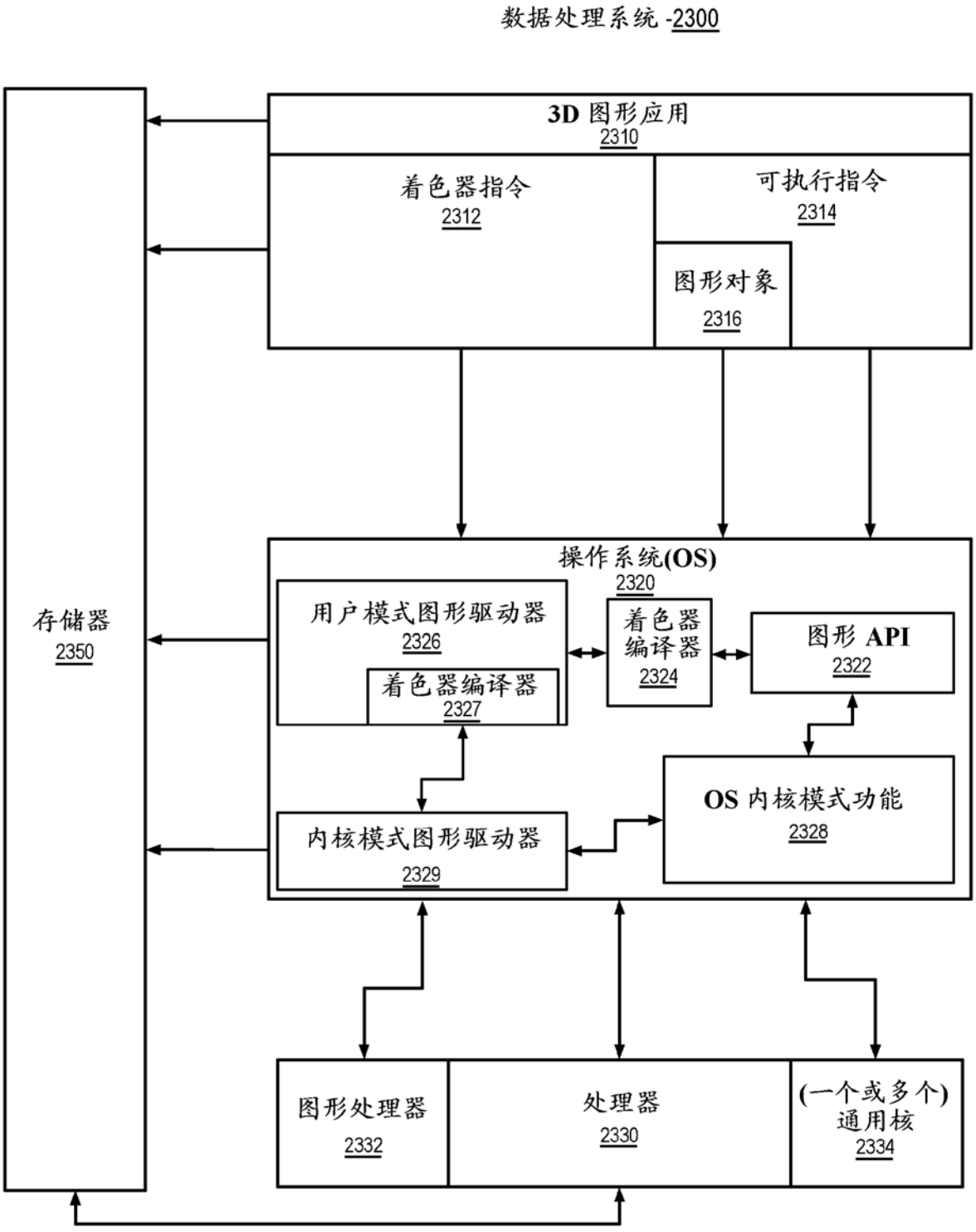

数据处理系统 -2300
3D 图形应用
2310
着色器指令
2312
可执行指令
2314
图形对象
2316
操作系统(OS)
2320
存储器
2350
用户模式图形驱动器
2326
着色器编译器
2327
着色器编译器
2324
图形 API
2322
OS 内核模式功能
2328
内核模式图形驱动器
2329
图形处理器
2332
处理器
2330
(一个或多个)通用核
2334

图 23

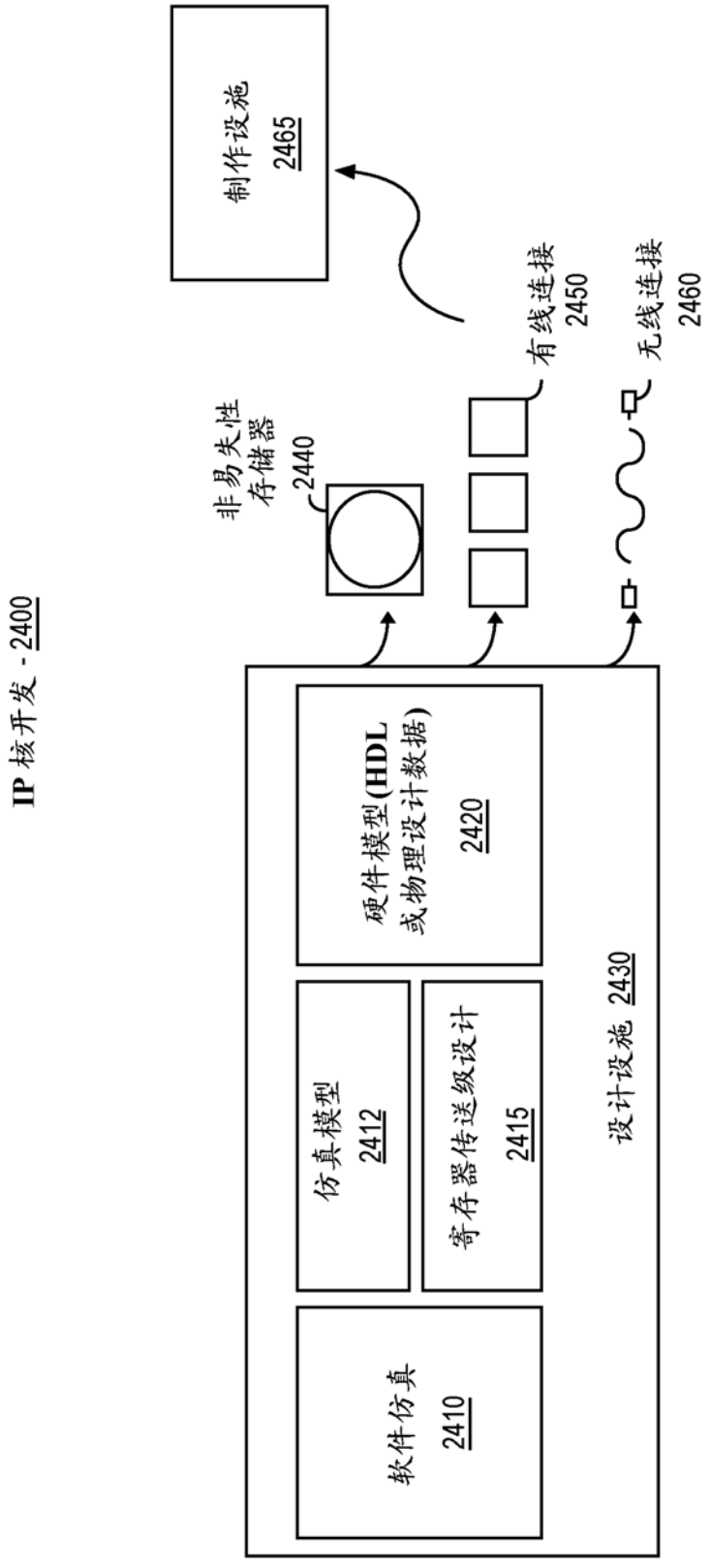

IP 核开发 - 2400
制作设施 2465
有线连接 2450
无线连接 2460
非易失性存储器 2440
硬件模型(HDL或物理设计数据) 2420
仿真模型 2412
寄存器传送级设计 2415
软件仿真 2410
设计设施 2430

图 24A

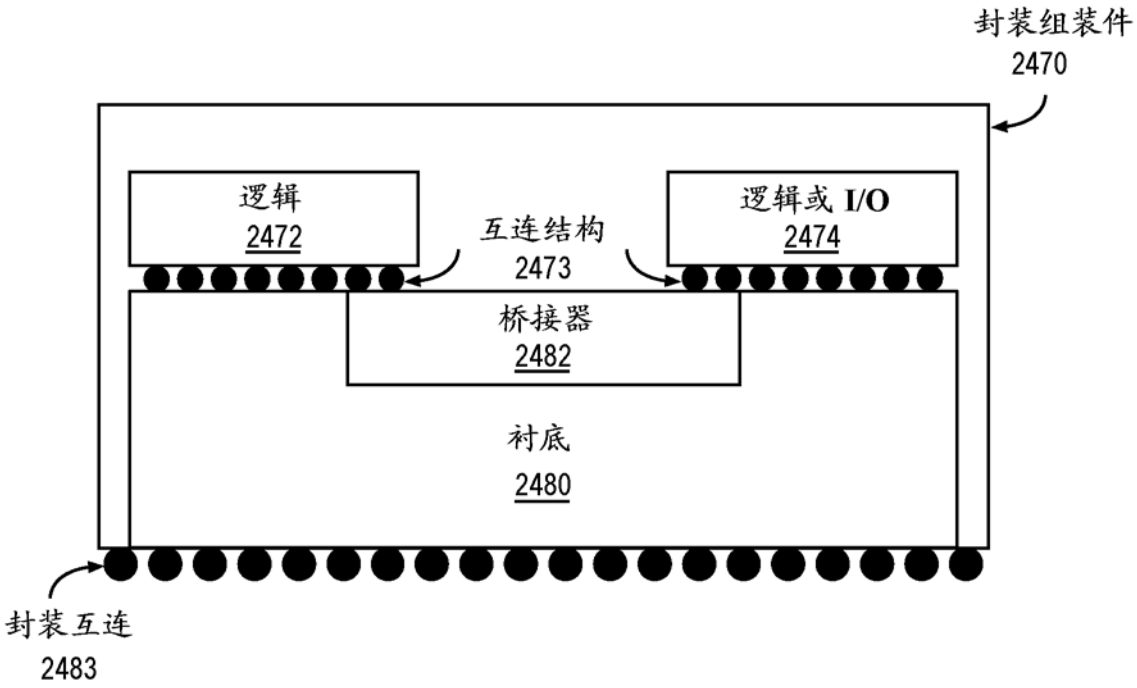

封装组装件
2470
逻辑
2472
互连结构
2473
逻辑或 I/O
2474
桥接器
2482
衬底
2480
封装互连
2483

图 24B

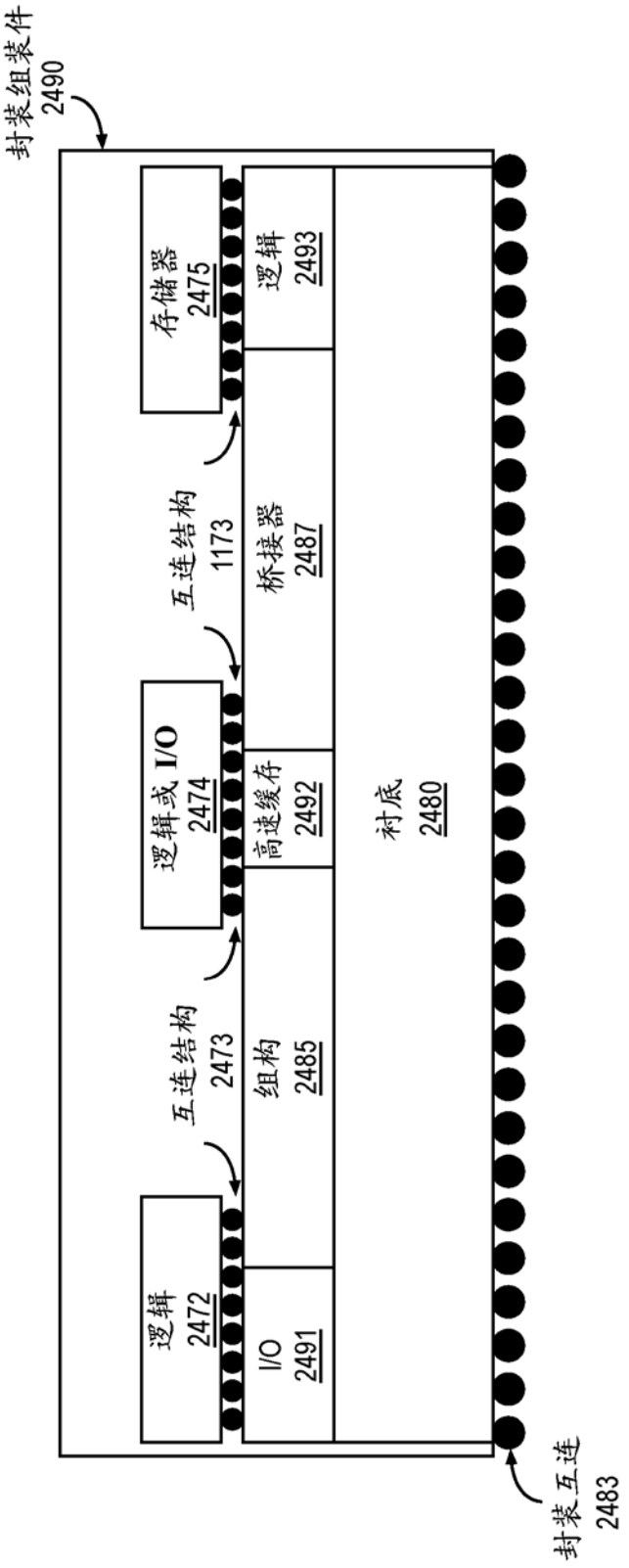

封装组装件
2490
存储器
2475
逻辑
2493
互连结构
1173
桥接器
2487
逻辑或 I/O
2474
高速缓存
2492
衬底
2480
互连结构
2473
组构
2485
逻辑
2472
I/O
2491
封装互连
2483

图 24C

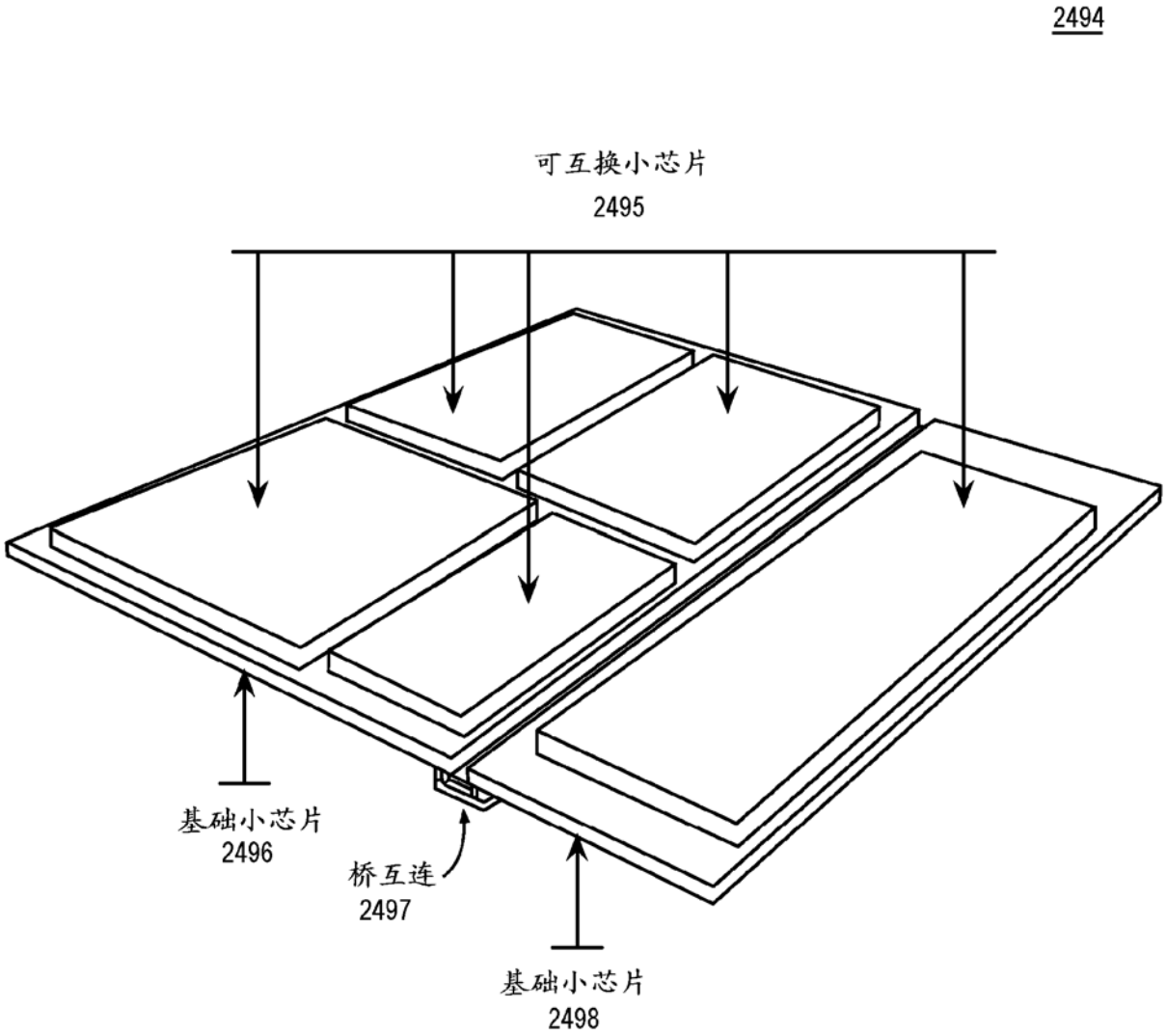


图 24D

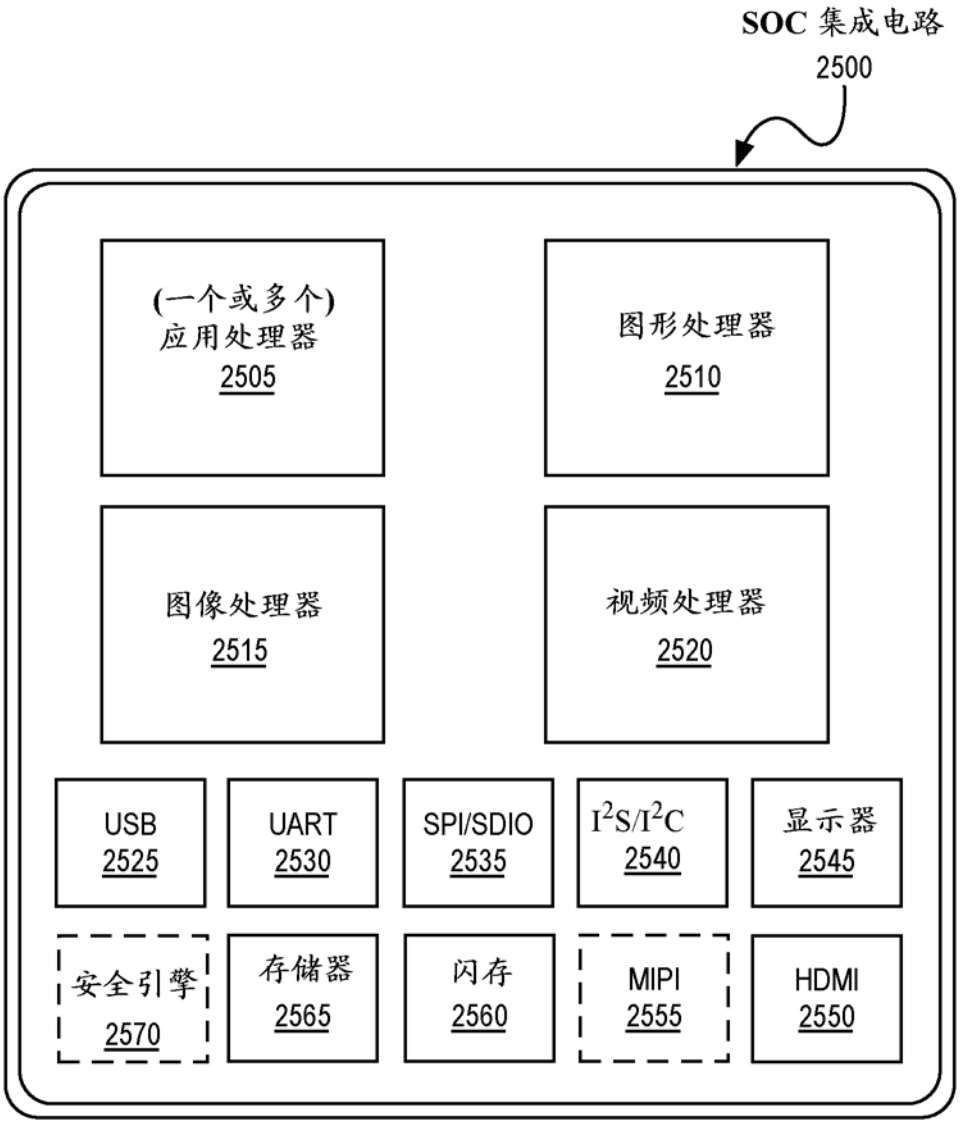

SOC 集成电路
2500
(一个或多个)
应用处理器
2505
图形处理器
2510
图像处理器
2515
视频处理器
2520
USB
2525
UART
2530
SPI/SDIO
2535
I2S/I2C
2540
显示器
2545
安全引擎
2570
存储器
2565
闪存
2560
MIPI
2555
HDMI
2550

图 25

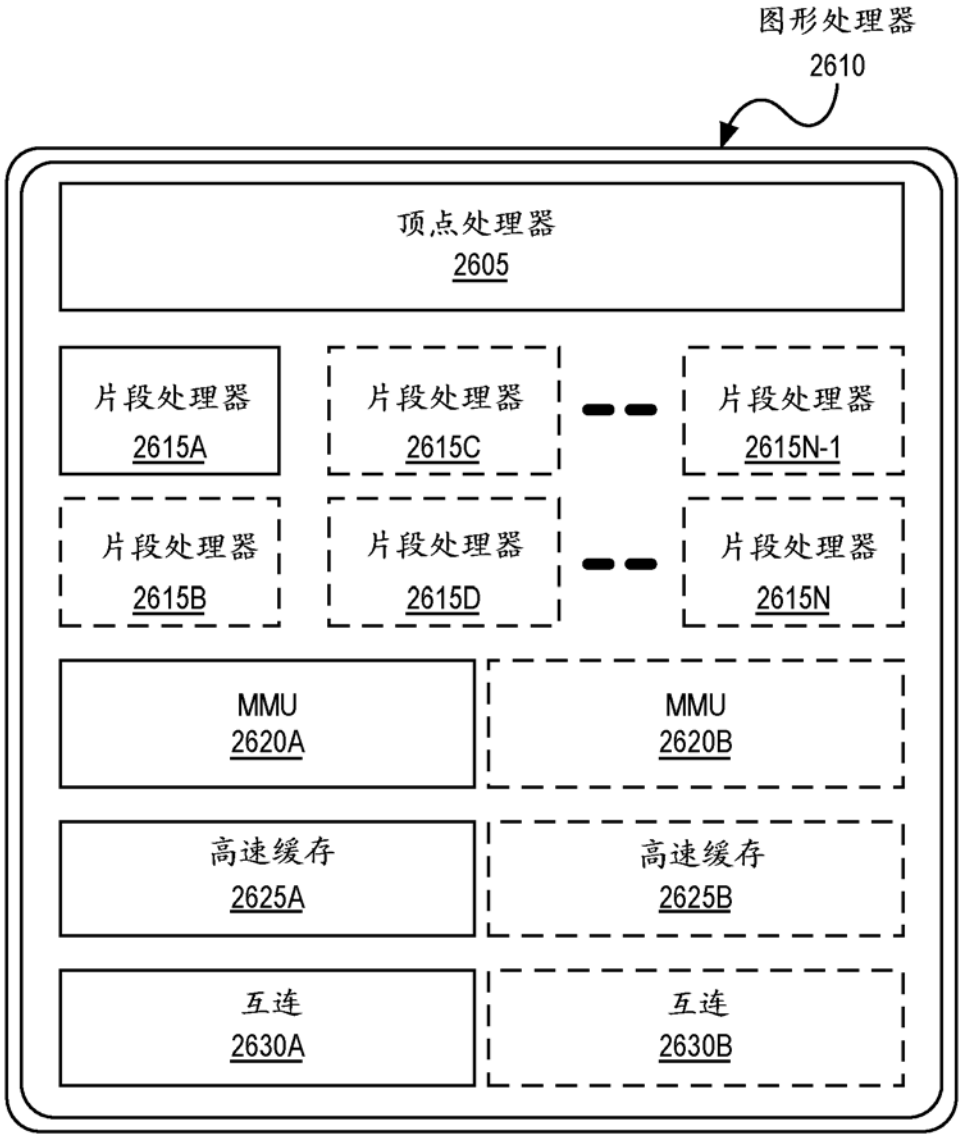

图形处理器
2610
顶点处理器
2605
片段处理器
2615A
片段处理器
2615C
片段处理器
2615N-1
片段处理器
2615B
片段处理器
2615D
片段处理器
2615N
MMU
2620A
MMU
2620B
高速缓存
2625A
高速缓存
2625B
互连
2630A
互连
2630B

图 26A

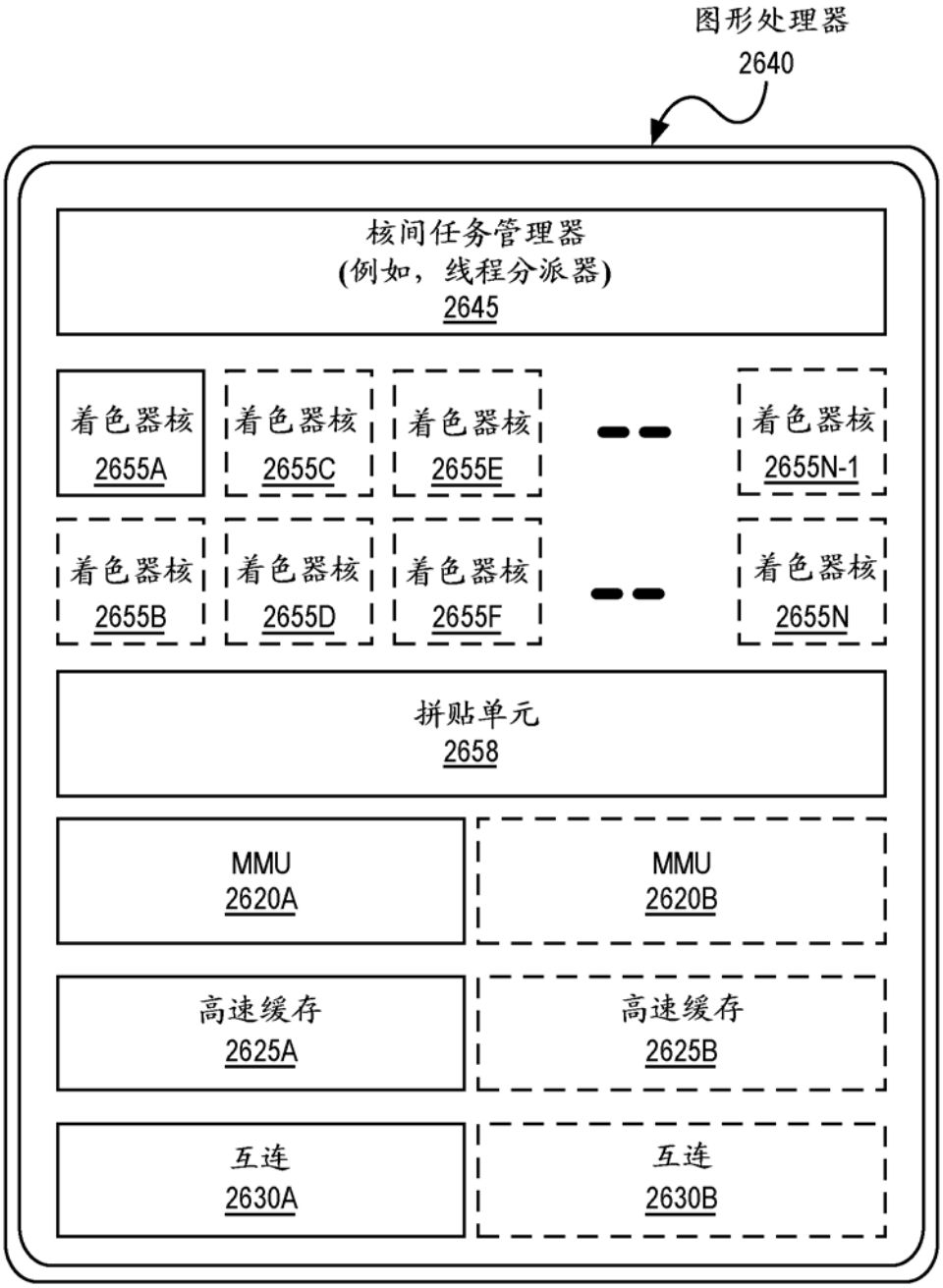

图形处理器
2640
核间任务管理器
(例如，线程分派器)
2645
着色器核
2655A
着色器核
2655C
着色器核
2655E
着色器核
2655N-1
着色器核
2655B
着色器核
2655D
着色器核
2655F
着色器核
2655N
拼贴单元
2658
MMU
2620A
MMU
2620B
高速缓存
2625A
高速缓存
2625B
互连
2630A
互连
2630B

图 26B

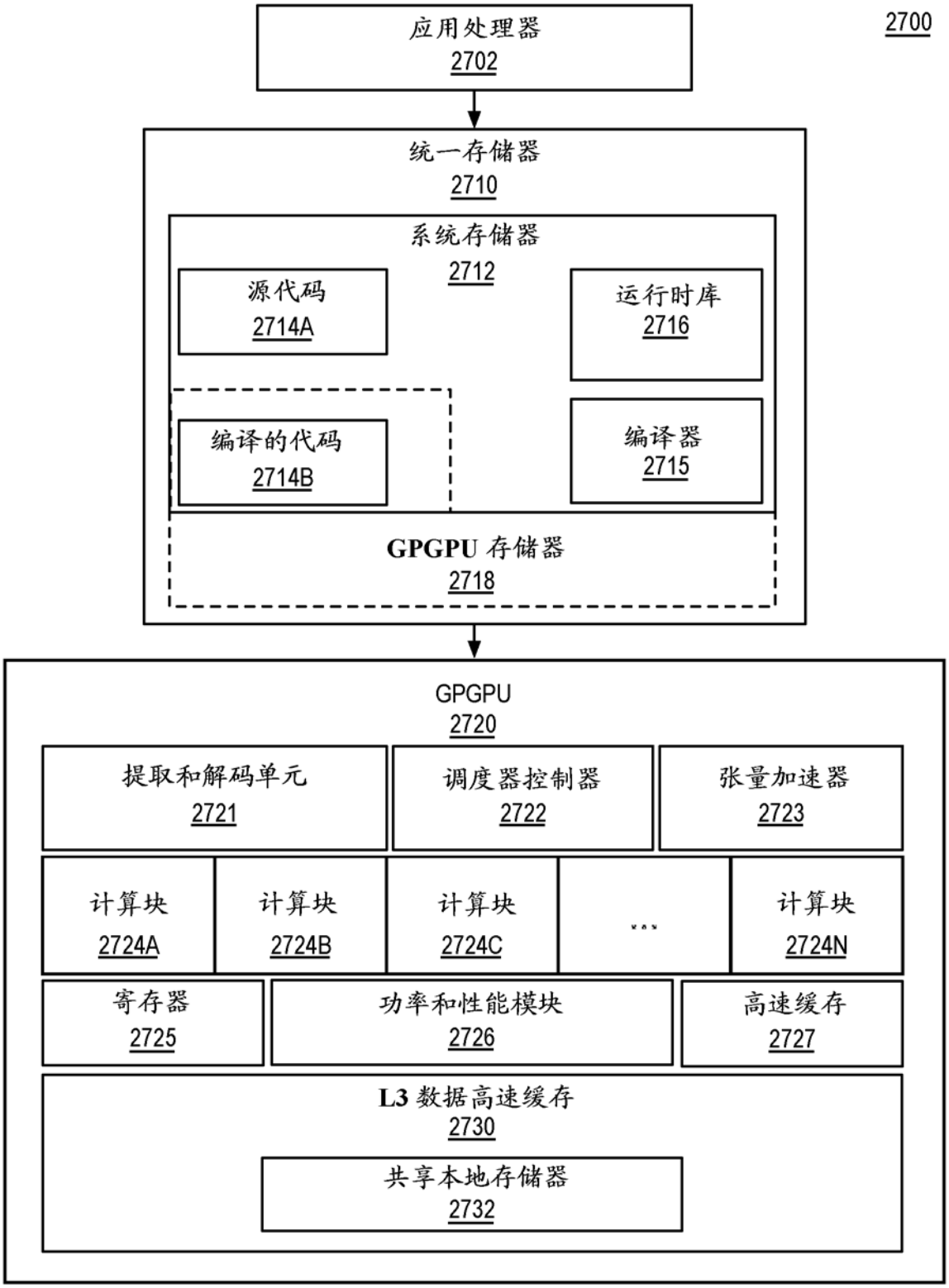

2700
应用处理器
2702
统一存储器
2710
系统存储器
2712
源代码
2714A
运行时库
2716
编译的代码
2714B
编译器
2715
GPGPU 存储器
2718
GPGPU
2720
提取和解码单元
2721
调度器控制器
2722
张量加速器
2723
计算块
2724A
计算块
2724B
计算块
2724C
...
计算块
2724N
寄存器
2725
功率和性能模块
2726
高速缓存
2727
L3 数据高速缓存
2730
共享本地存储器
2732

图 27

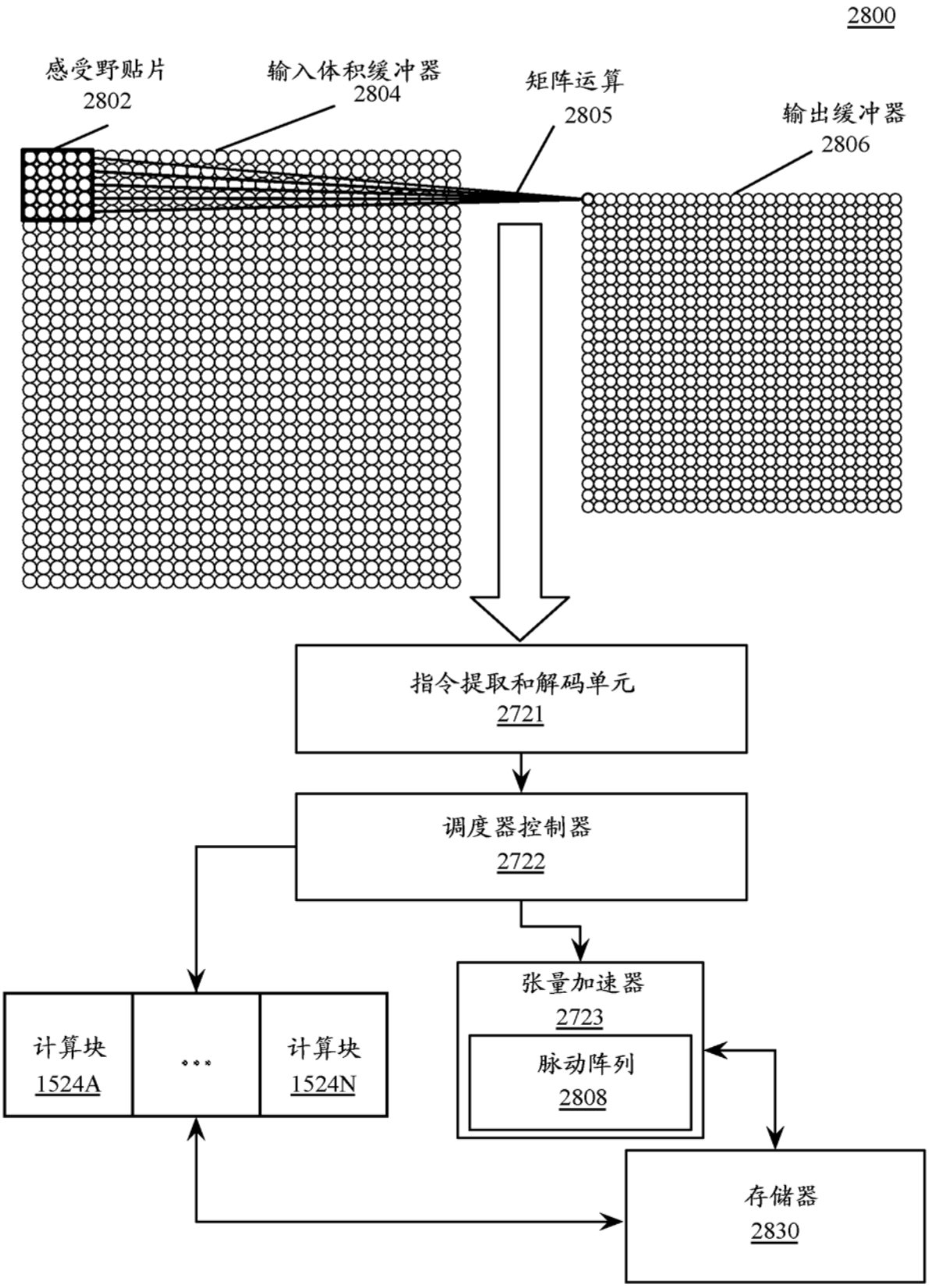

2800
感受野贴片
2802
输入体积缓冲器
2804
矩阵运算
2805
输出缓冲器
2806
指令提取和解码单元
2721
调度器控制器
2722
计算块
1524A
...
计算块
1524N
张量加速器
2723
脉动阵列
2808
存储器
2830

图 28

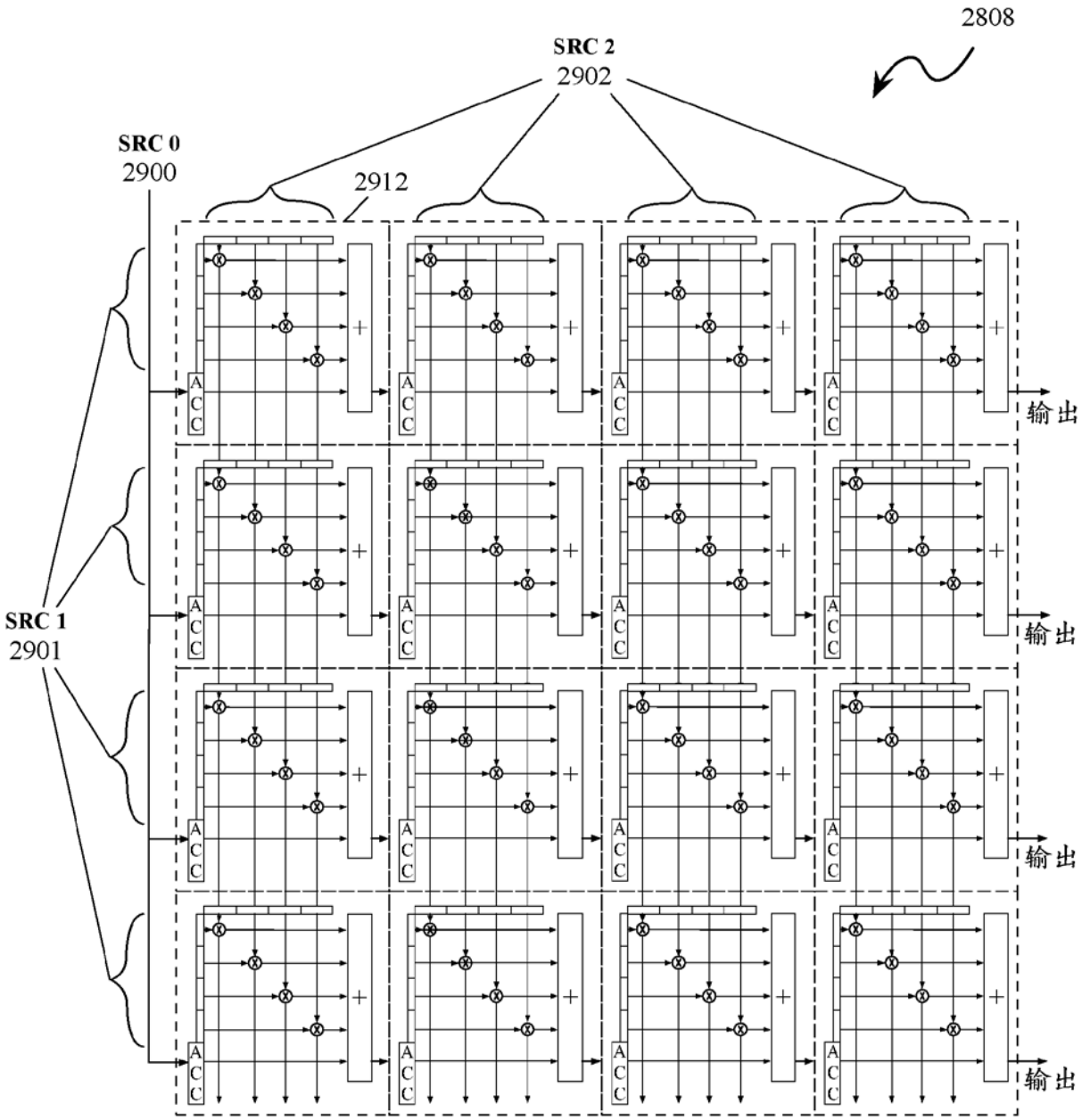


图 29A

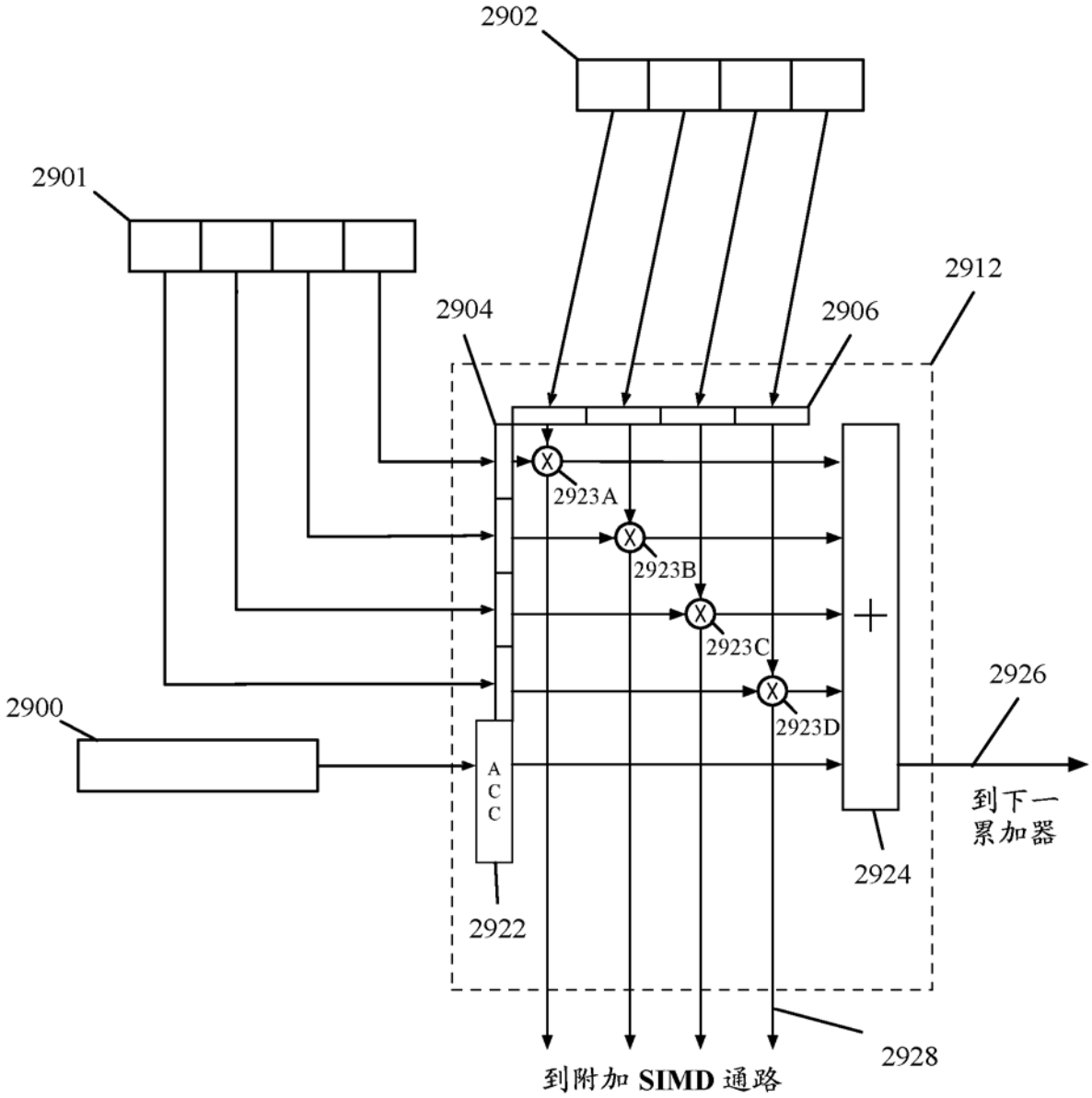

2902
2901
2912
2904
2906
2923A
2923B
2923C
2923D
2926
2900
ACC
到下一
累加器
2924
2922
2928
到附加 SIMD 通路

图 29B

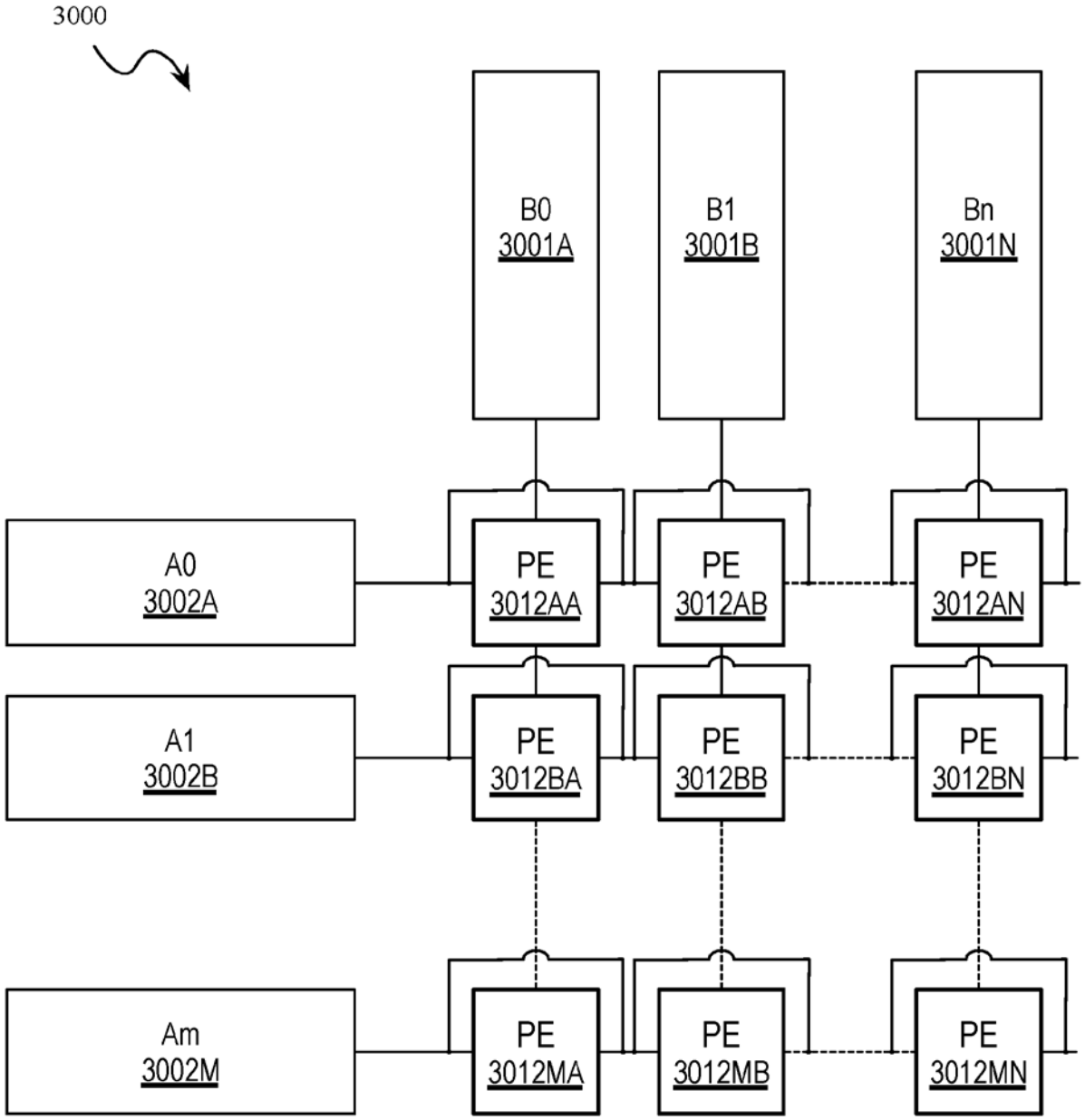

3000
B0
3001A
B1
3001B
Bn
3001N
A0
3002A
A1
3002B
Am
3002M
PE
3012AA
PE
3012AB
PE
3012AN
PE
3012BA
PE
3012BB
PE
3012BN
PE
3012MA
PE
3012MB
PE
3012MN

图 30

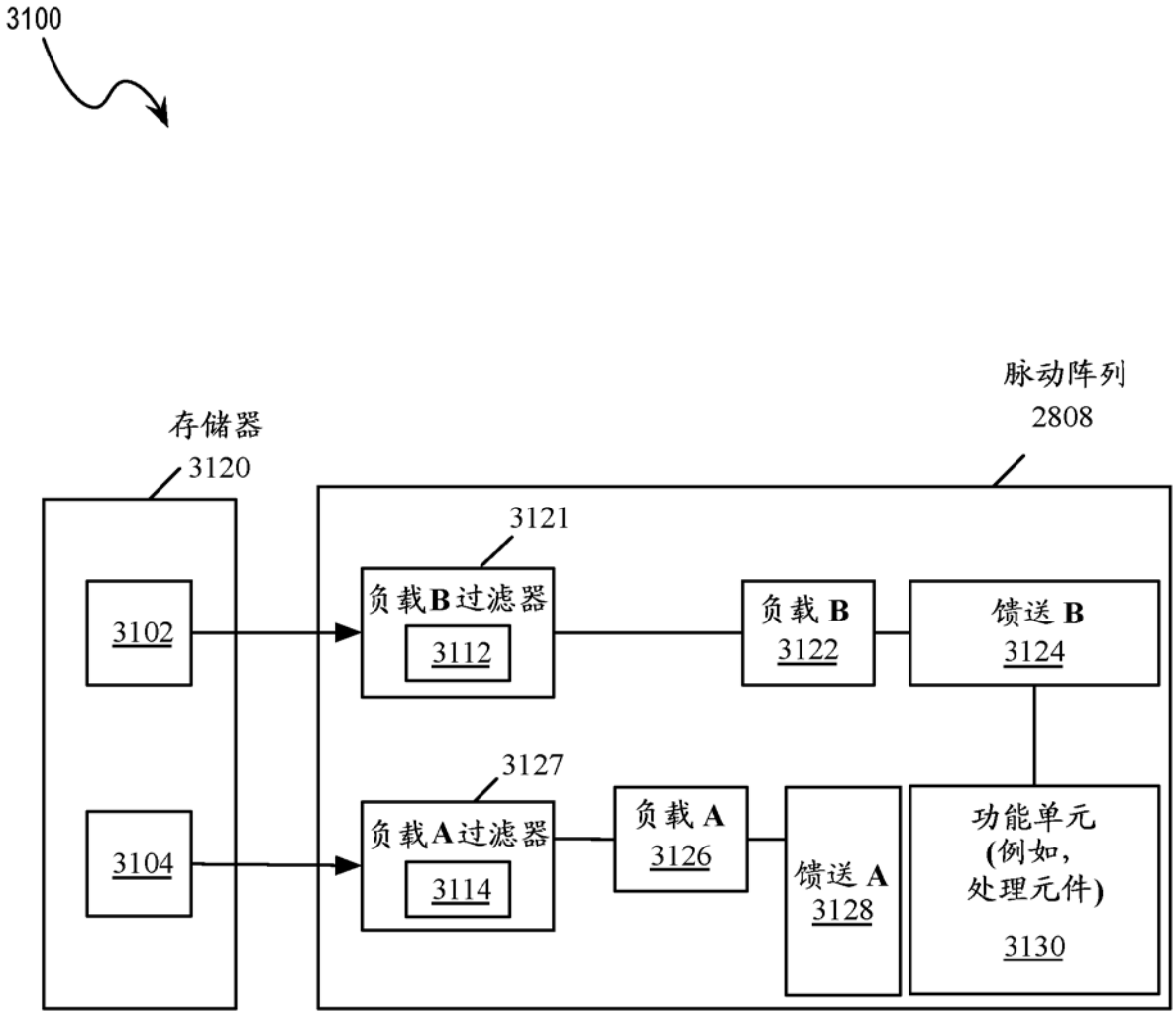

3100
脉动阵列
2808
存储器
3120
3121
3102
负载B过滤器
3112
负载 B
3122
馈送 B
3124
3127
3104
负载A过滤器
3114
负载 A
3126
馈送 A
3128
功能单元
(例如,
处理元件)
3130

图 31A

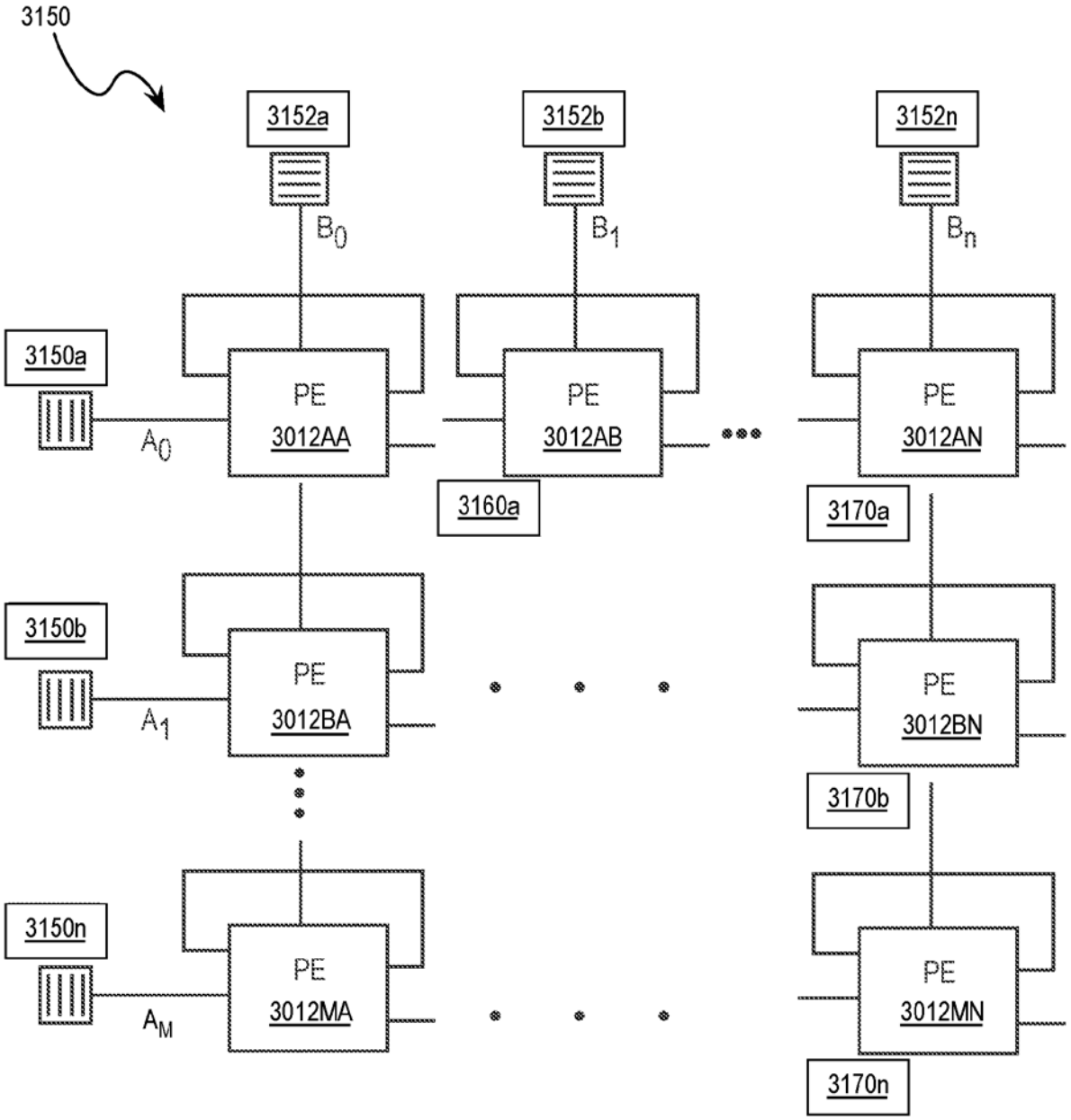

3150
3152a
3152b
3152n
B0
B1
Bn
3150a
A0
PE
3012AA
PE
3012AB
PE
3012AN
3160a
3170a
3150b
A1
PE
3012BA
PE
3012BN
3170b
3150n
AM
PE
3012MA
PE
3012MN
3170n

图 31B

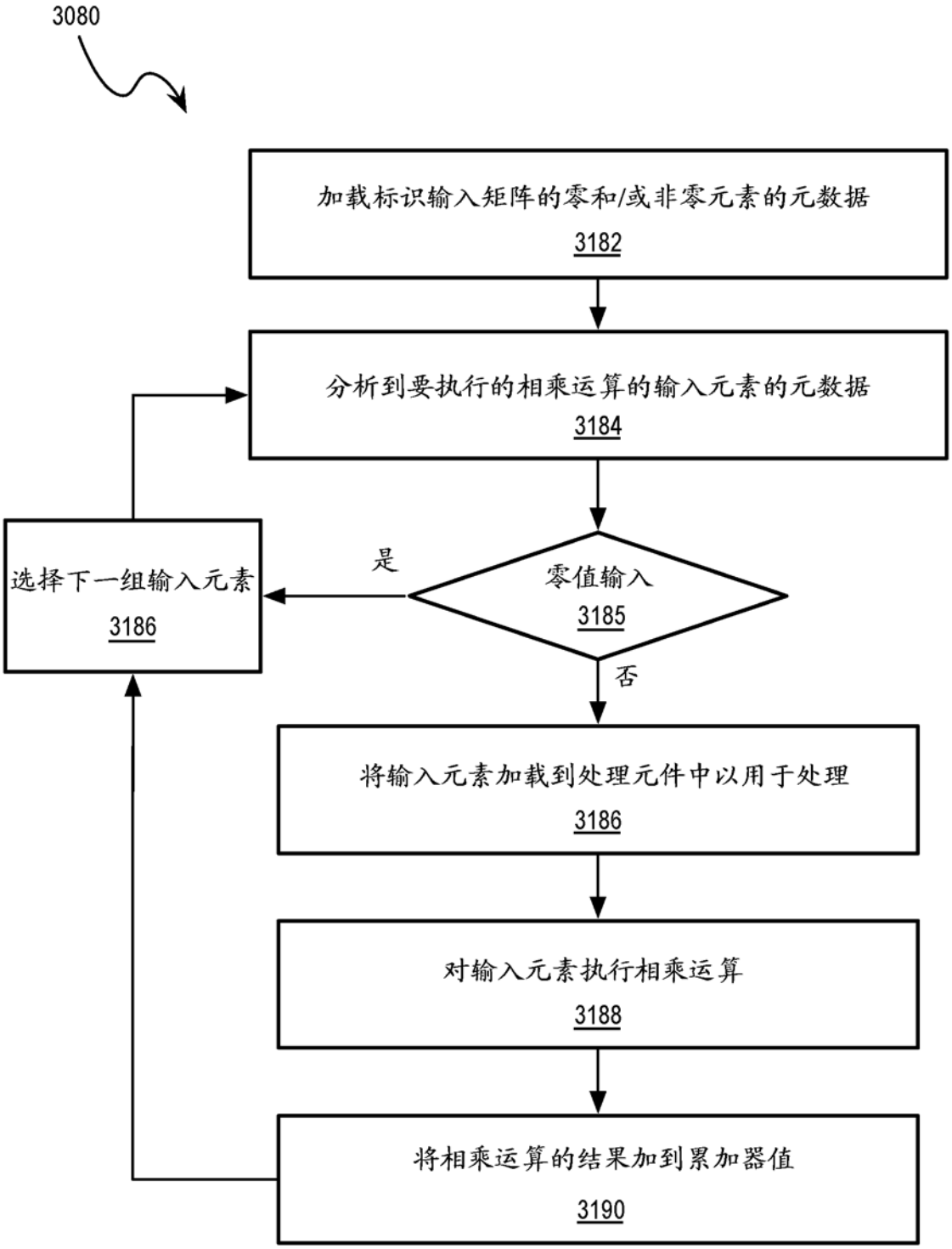

3080
加载标识输入矩阵的零和/或非零元素的元数据
3182
分析到要执行的相乘运算的输入元素的元数据
3184
零值输入
3185
是
选择下一组输入元素
3186
否
将输入元素加载到处理元件中以用于处理
3186
对输入元素执行相乘运算
3188
将相乘运算的结果加到累加器值
3190

图 31C

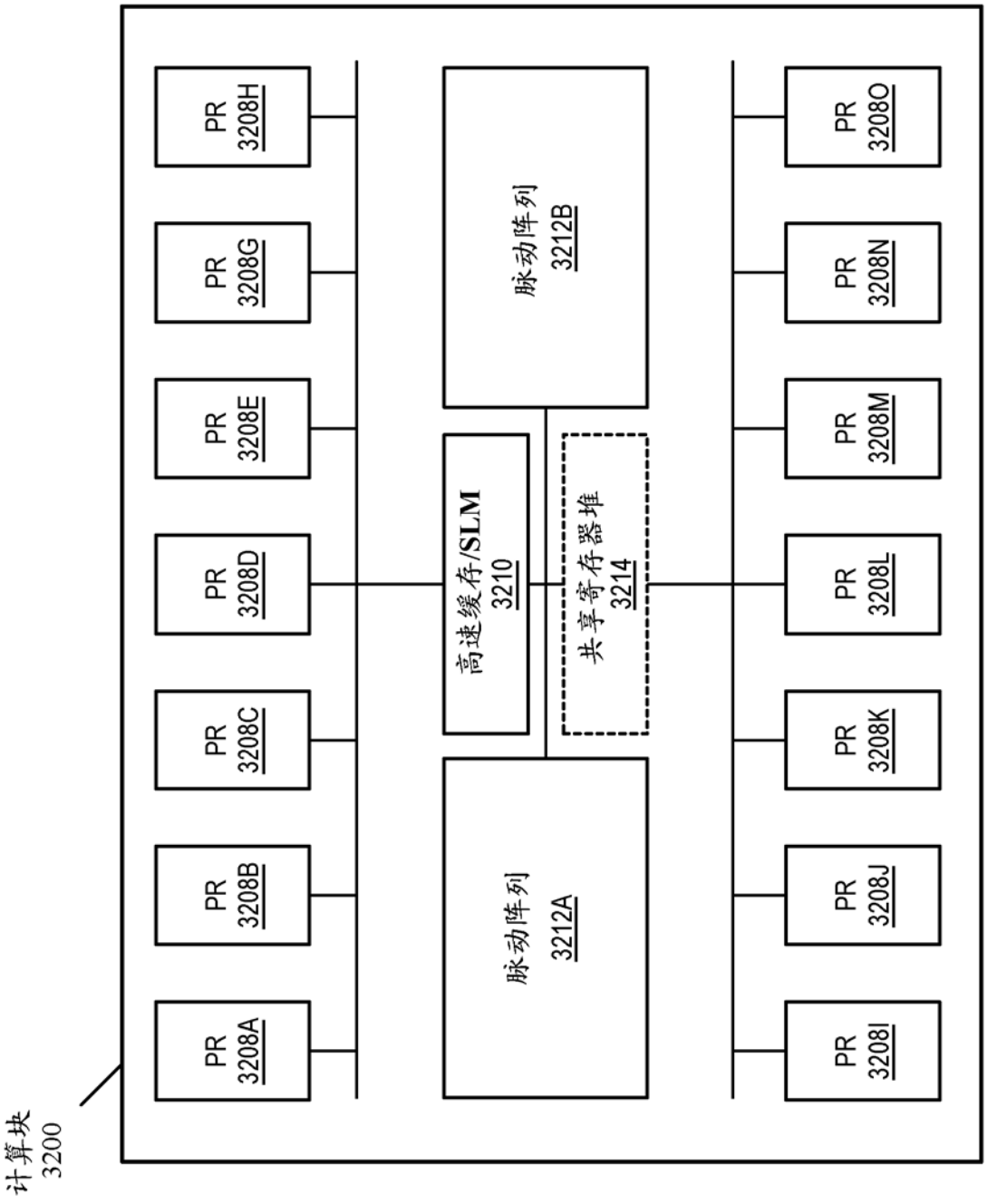

计算块
3200
PR
3208A
PR
3208B
PR
3208C
PR
3208D
PR
3208E
PR
3208G
PR
3208H
脉动阵列
3212A
高速缓存/SLM
3210
共享寄存器堆
3214
脉动阵列
3212B
PR
3208I
PR
3208J
PR
3208K
PR
3208L
PR
3208M
PR
3208N
PR
3208O

图 32

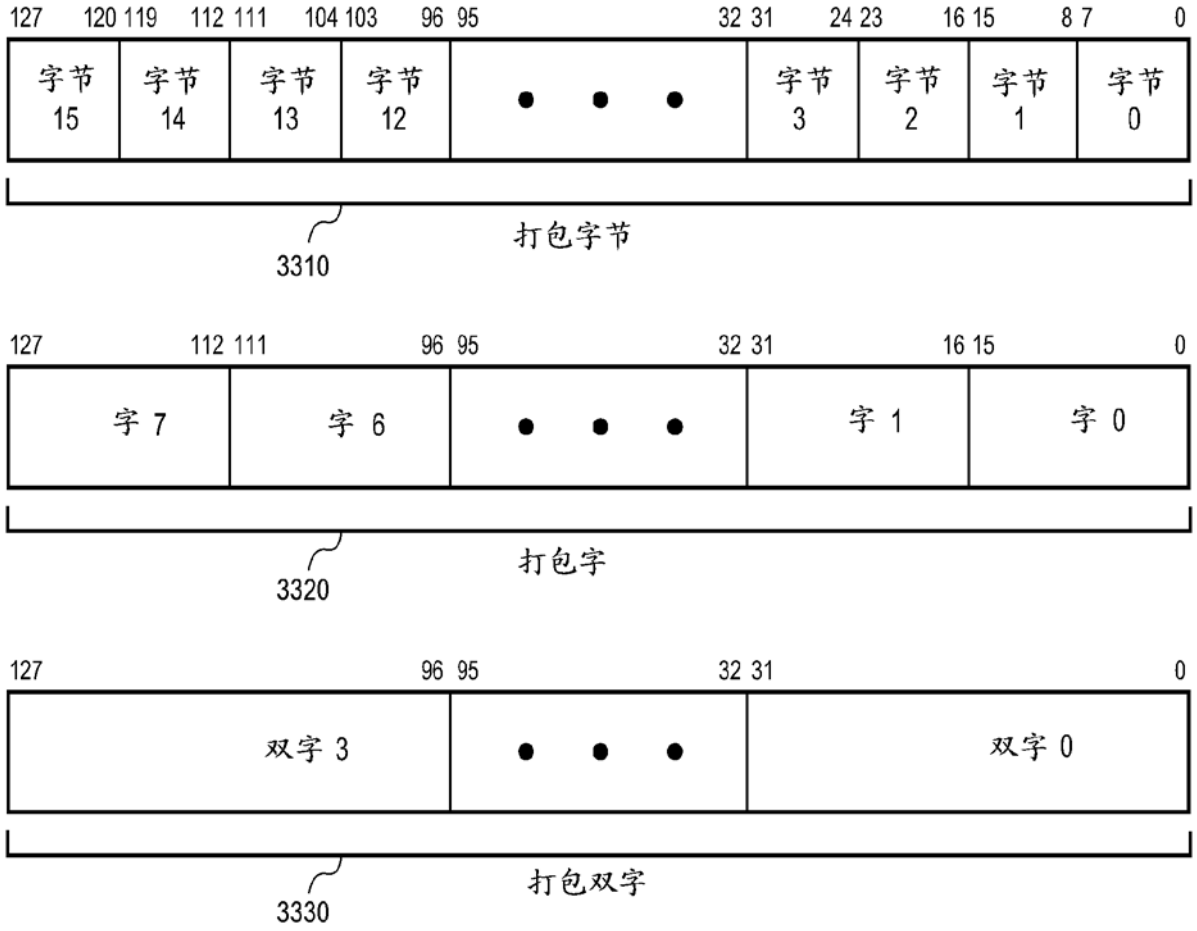

127 120 119 112 111 104 103 96 95 32 31 24 23 16 15 8 7 0
字节 15
字节 14
字节 13
字节 12
字节 3
字节 2
字节 1
字节 0
打包字节
3310
127 112 111 96 95 32 31 16 15 0
字 7
字 6
字 1
字 0
打包字
3320
127 96 95 32 31 0
双字 3
双字 0
打包双字
3330

图 33

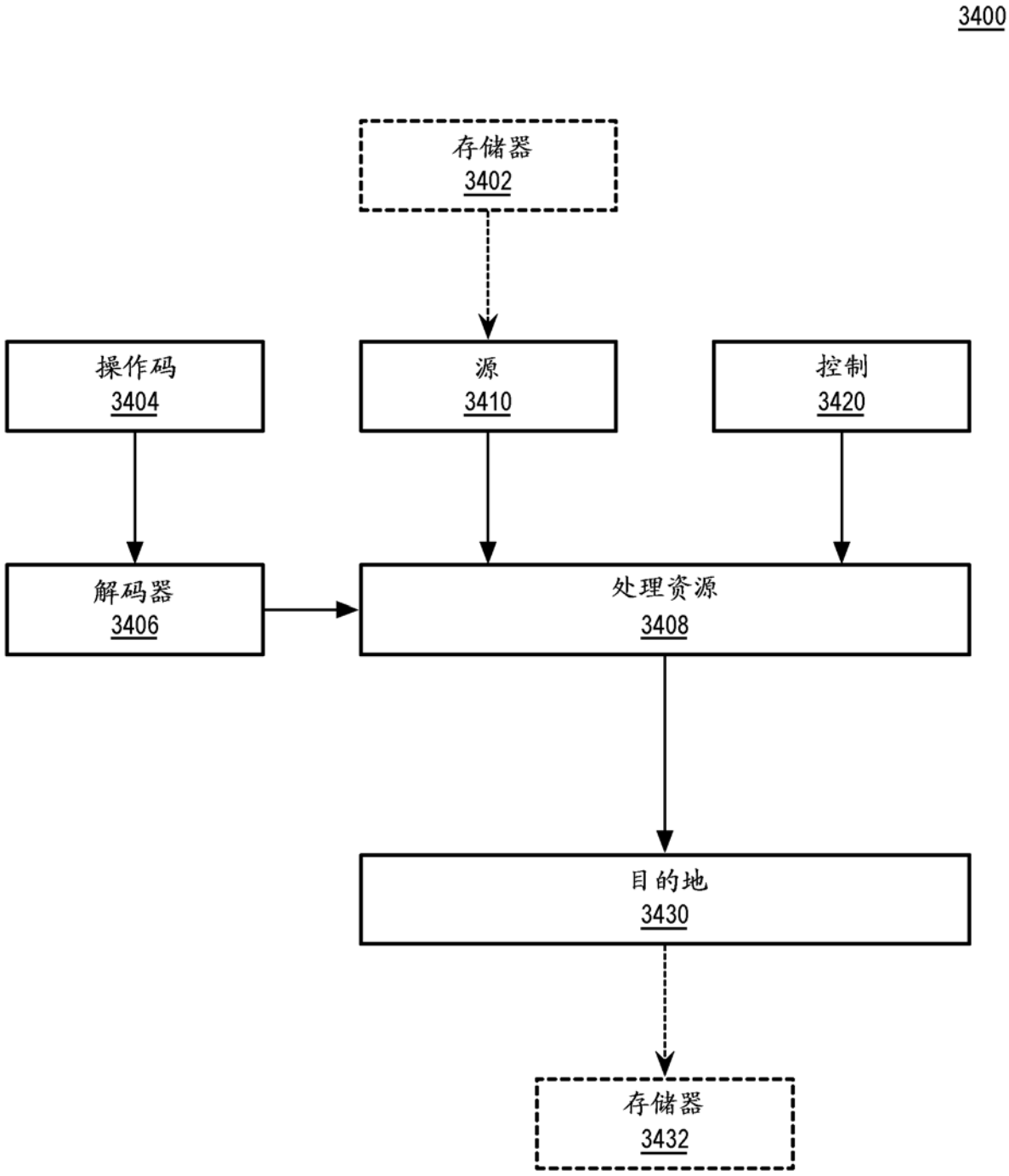

3400
存储器
3402
操作码
3404
源
3410
控制
3420
解码器
3406
处理资源
3408
目的地
3430
存储器
3432

图 34

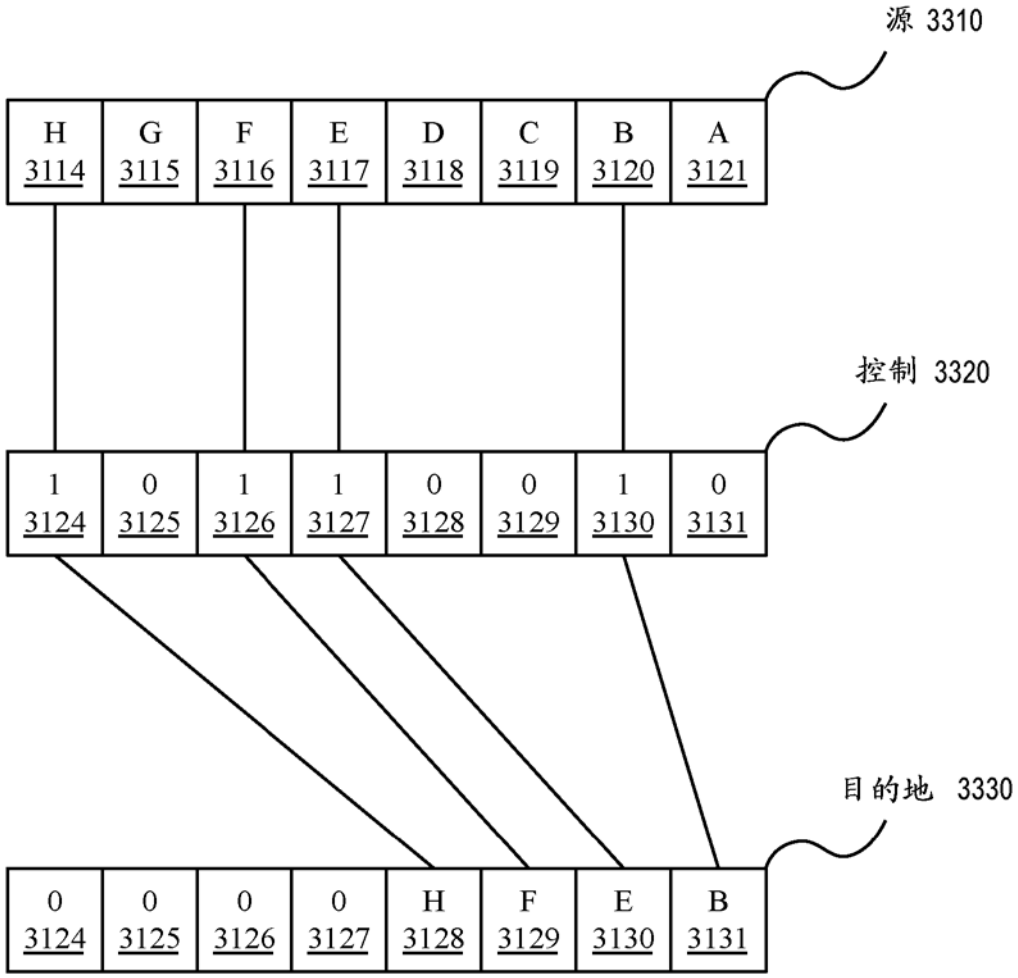

GPGPU 打包数据压缩 - 3400
源 3310
H 3114
G 3115
F 3116
E 3117
D 3118
C 3119
B 3120
A 3121
控制 3320
1 3124
0 3125
1 3126
1 3127
0 3128
0 3129
1 3130
0 3131
目的地 3330
0 3124
0 3125
0 3126
0 3127
H 3128
F 3129
E 3130
B 3131

图 35A

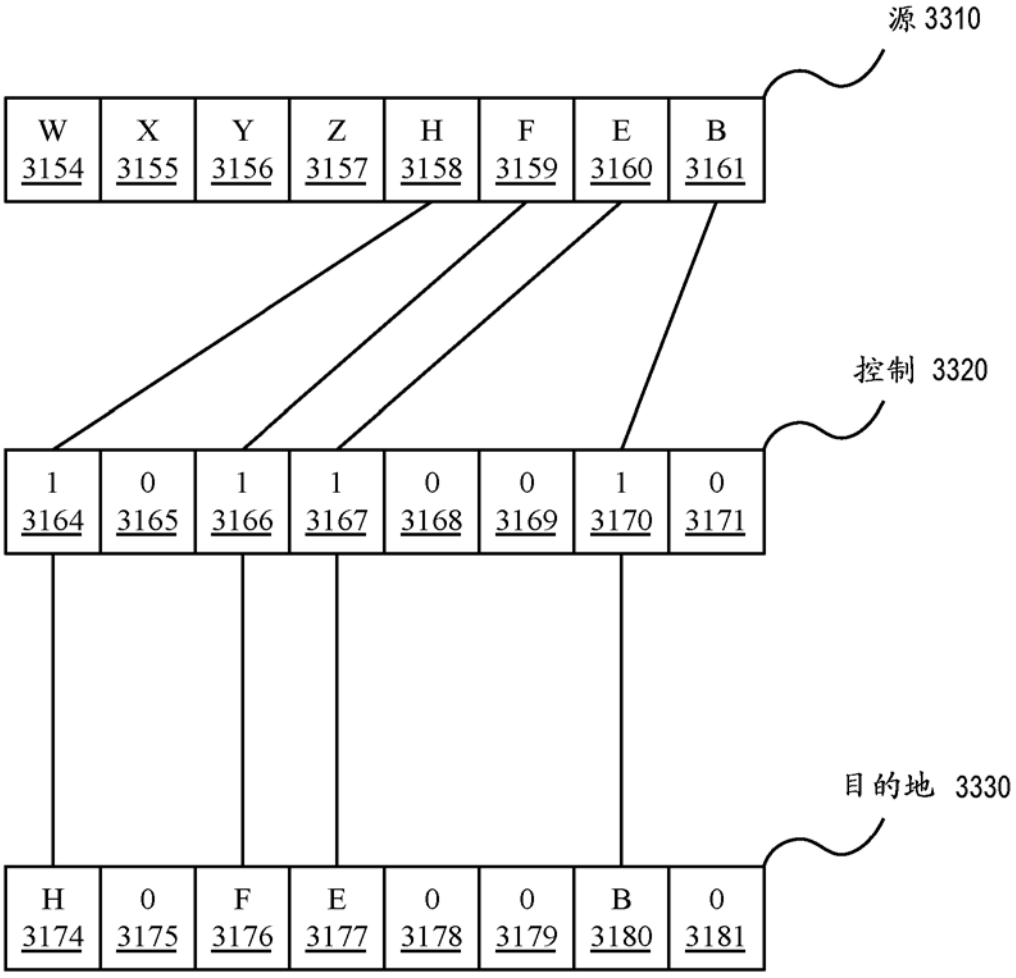

GPGPU 打包数据扩展 - 3450
源 3310
W 3154
X 3155
Y 3156
Z 3157
H 3158
F 3159
E 3160
B 3161
控制 3320
1 3164
0 3165
1 3166
1 3167
0 3168
0 3169
1 3170
0 3171
目的地 3330
H 3174
0 3175
F 3176
E 3177
0 3178
0 3179
B 3180
0 3181

图 35B

3602

0	-0.2	-0.5	0
0	0	0.1	0.2
0.2	0	-0.5	0
-0.3	0	-0.3	-0.5

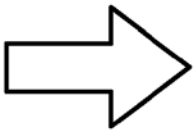
3604

0	0	-0.3	0
0	0	0.5	0.8
0	0.1	0	0
0.3	0.7	0	0

图 36A

3612

0	-0.2	-0.2	0.3
0	0.3	0.2	0.3
0.2	0.1	0.1	-0.2
0.5	0.2	-0.1	-0.1



3614

0	0	-0.2	0.3
0	0	0.2	0.3
0.2	0.1	0	0
0.5	0.2	0	0

图 36B

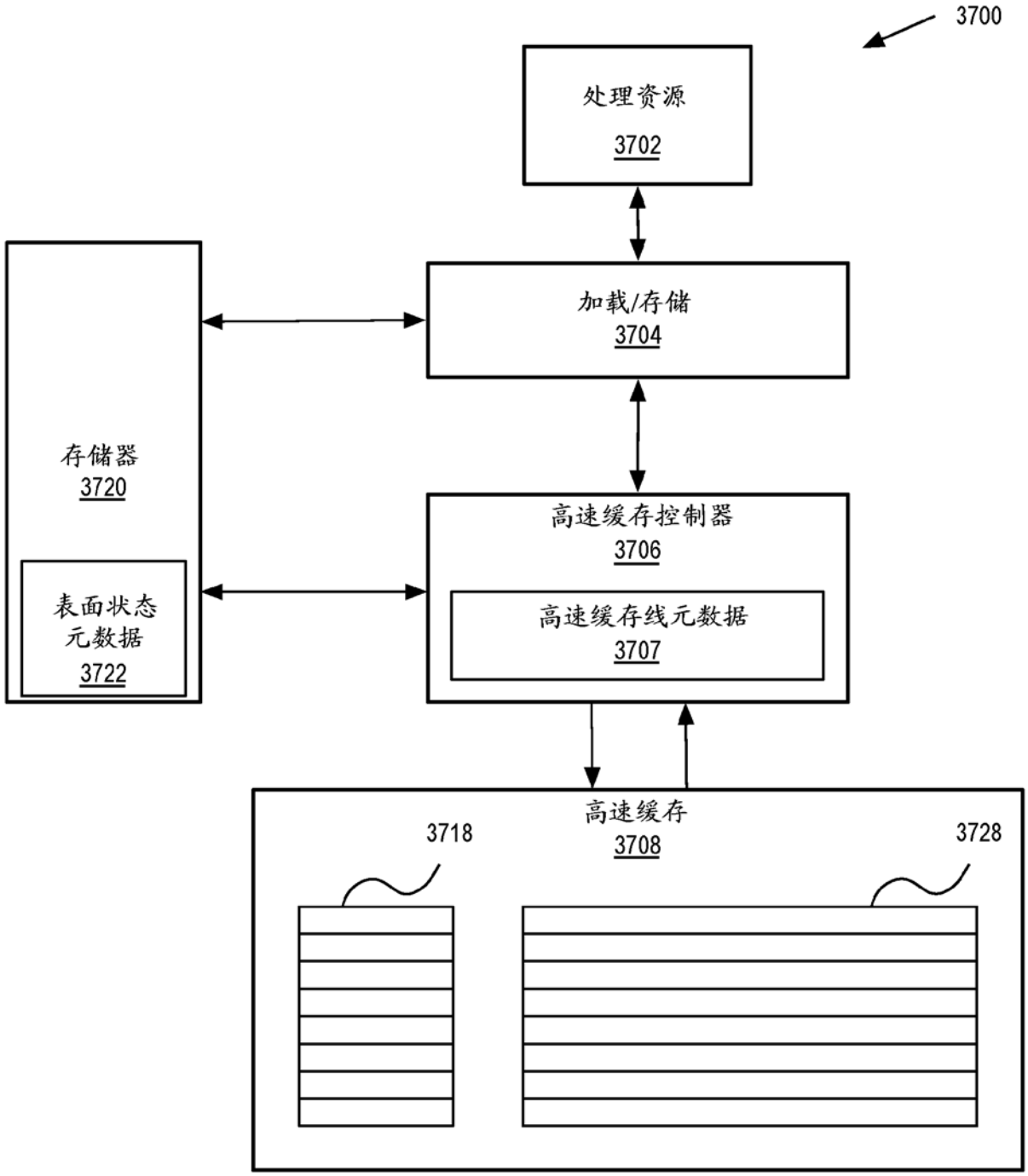

3700
处理资源
3702
加载/存储
3704
存储器
3720
表面状态
元数据
3722
高速缓存控制器
3706
高速缓存线元数据
3707
高速缓存
3708
3718
3728

图 37

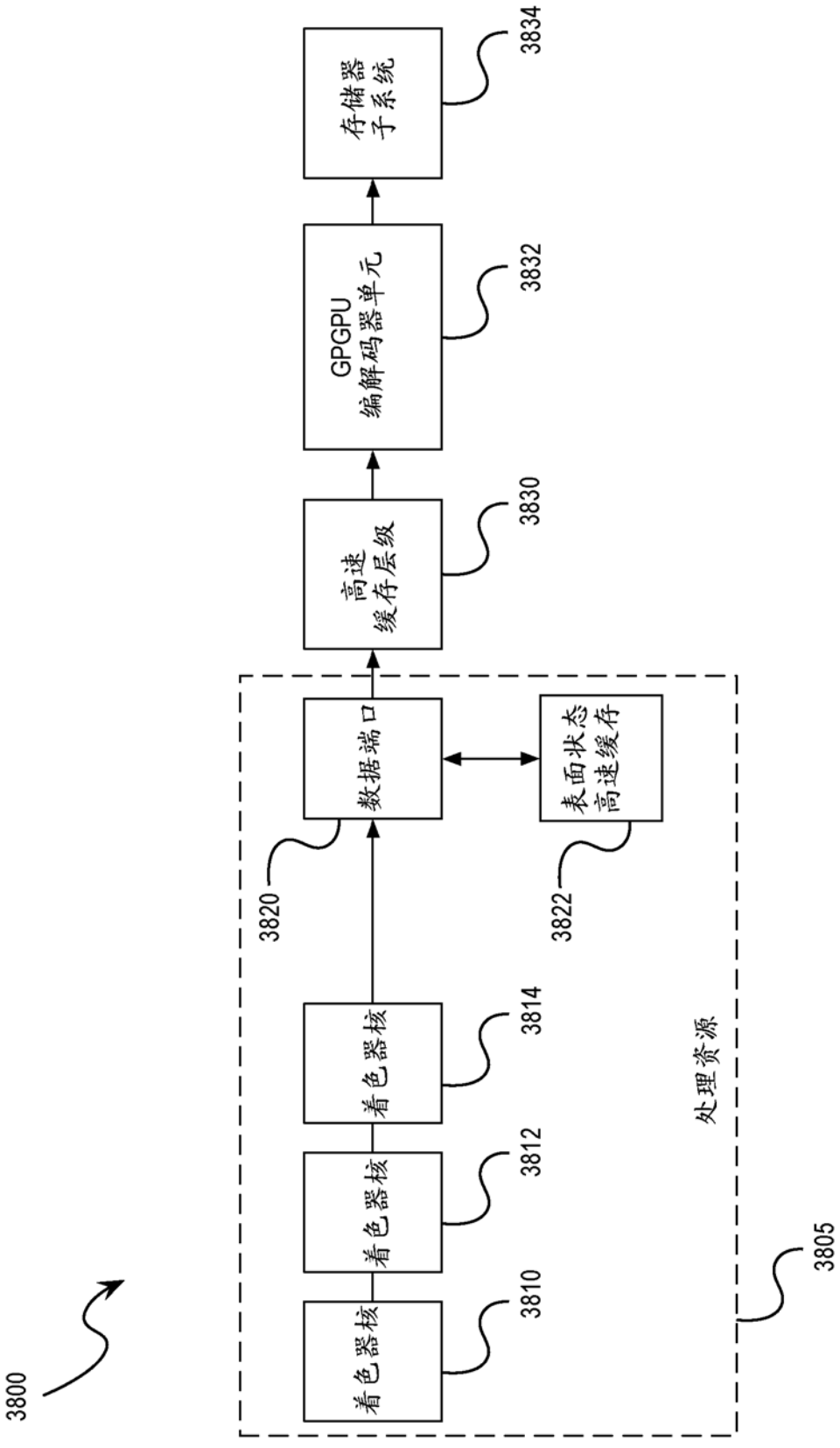

3800
着色器核
3810
着色器核
3812
着色器核
3814
数据端口
3820
表面状态高速缓存
3822
处理资源
3805
高速缓存层级
3830
GPGPU 编解码器单元
3832
存储器子系统
3834

图 38

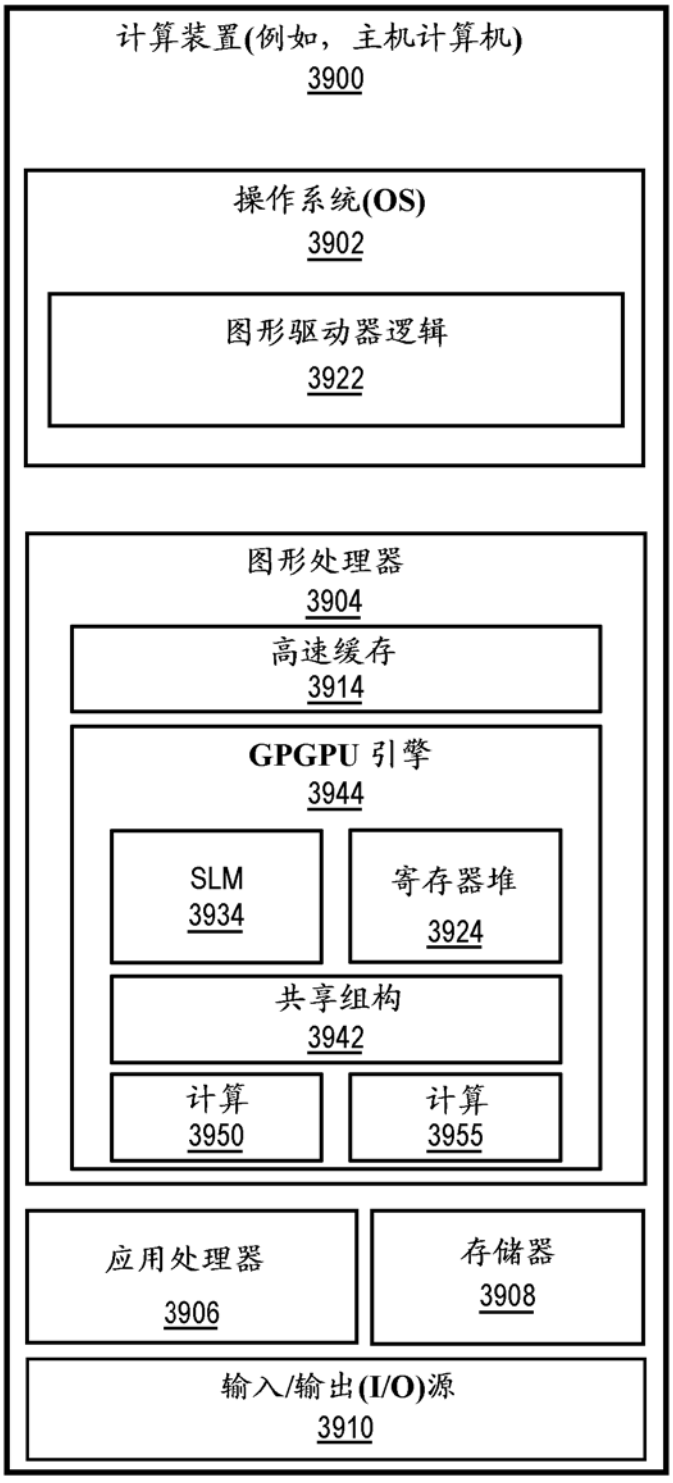

计算装置(例如，主机计算机)
3900
操作系统(OS)
3902
图形驱动器逻辑
3922
图形处理器
3904
高速缓存
3914
GPGPU 引擎
3944
SLM
3934
寄存器堆
3924
共享组构
3942
计算
3950
计算
3955
应用处理器
3906
存储器
3908
输入/输出(I/O)源
3910

图 39