(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 111708718 A
(43)申请公布日 2020.09.25

(21)申请号 202010098690.4

(22)申请日 2020.02.18

(30)优先权数据
16/356,455 2019.03.18 US

(71)申请人 英特尔公司
地址 美国加利福尼亚州

(72)发明人 A·R·阿普 A·考克 J·雷 N·库雷 P·萨蒂 S·卡玛 V·兰甘纳坦

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

(51)Int.Cl.
G06F 12/084(2016.01)
G06T 1/20(2006.01)
G06T 1/60(2006.01)

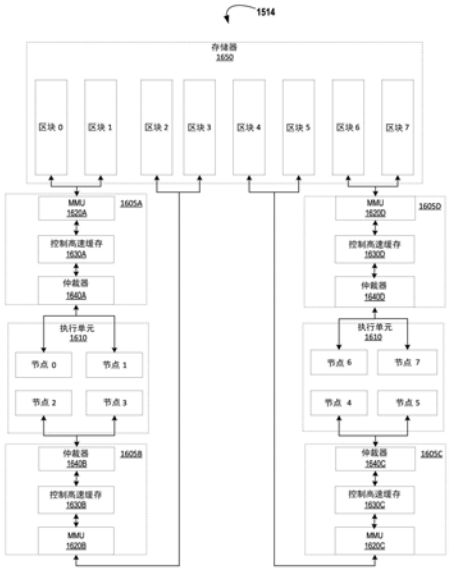
权利要求书2页 说明书28页 附图23页

(54)发明名称

存储器压缩散列机制

(57)摘要

本申请公开了存储器压缩散列机制。公开了用于促进存储器数据压缩的装置。装置包括:存储器,具有多个区块,用于存储主数据和与主数据相关联的元数据;以及存储器管理单元(MMU),耦合至多个区块,用于执行散列函数来为主数据和元数据计算到存储器中的虚拟地址位置中的索引,并且调节元数据虚拟地址位置以将每个经调节的元数据虚拟地址位置存储在存储相关联的主数据的区块中。



CN 111708718 A

1.一种用于促进存储器数据压缩的装置,包括:

存储器,具有多个区块,用于存储主数据和与所述主数据相关联的元数据;以及

存储器管理单元MMU,耦合至所述多个区块,用于执行散列函数来为所述主数据和所述元数据计算到存储器中的虚拟地址位置中的索引,并且调节元数据虚拟地址位置以将每个经调节的元数据虚拟地址位置存储在存储相关联的主数据的区块中。

2.如权利要求1所述的装置,其中,所述MMU调节所述元数据的地址位置包括:将要存储在所述区块中的所述元数据组合以生成元数据块。

3.如权利要求2所述的装置,其中,所述MMU调节所述元数据的地址位置进一步包括:执行一个或多个移位操作。

4.如权利要求3所述的装置,其中,所述MMU包括多个MMU,每个MMU耦合至所述多个区块中的一个或多个。

5.如权利要求4所述的装置,其中,所述多个MMU中的每一个包括实现为执行所述散列函数的散列表。

6.如权利要求5所述的装置,其中,所述多个MMU中的每一个进一步执行线性映射以将主数据地址映射到元数据地址。

7.如权利要求4所述的装置,进一步包括:

第一MMU,耦合至第一区块,所述第一区块用于存储第一集合的主数据和与所述第一集合的主数据相关联的第一元数据块的元数据;以及

第二MMU,耦合至第二区块,所述第二区块用于存储第二集合的主数据和与所述第二集合的主数据相关联的第二元数据块的元数据。

8.一种用于促进存储器数据压缩的方法,包括:

执行散列函数来为主数据和与所述主数据相关联的元数据计算到存储器中的虚拟地址位置中的索引;

调节元数据虚拟地址位置;以及

将所述元数据存储在经调节的元数据虚拟地址位置处,其中每个经调节的元数据虚拟地址位置位于存储相关联的主数据的区块中。

9.如权利要求8所述的方法,进一步包括:

接收主数据地址;以及

将所述主数据地址映射到元数据地址。

10.如权利要求9所述的方法,进一步包括:调节所述元数据的地址位置包括执行一个或多个移位操作。

11.如权利要求10所述的方法,进一步包括:将要存储在所述区块中的所述元数据组合以生成元数据块。

12.如权利要求8所述的方法,进一步包括:

将第一集合的主数据和与所述第一集合的主数据相关联的第一元数据块的元数据存储在第一区块处;以及

将第二集合的主数据和与所述第二集合的主数据相关联的第二元数据块的元数据存储在第二区块处。

13.一种图形处理单元GPU,包括:

存储器,具有多个区块,用于存储主数据和与所述主数据相关联的元数据;以及

多个结构元件,耦合至所述多个区块,每个结构元件包括存储器管理单元MMU,所述MMU耦合至所述多个区块中的一个或多个,所述MMU用于执行散列函数来为所述主数据和所述元数据计算到存储器中的虚拟地址位置中的索引,并且调节元数据虚拟地址位置以将每个经调节的元数据虚拟地址位置存储在存储相关联的主数据的区块中。

14.如权利要求13所述的GPU,其中,所述MMU调节所述元数据的地址位置包括:将要存储在所述区块中的所述元数据组合以生成元数据块。

15.如权利要求14所述的GPU,其中,所述MMU调节所述元数据的地址位置进一步包括:执行一个或多个移位操作。

16.如权利要求15所述的GPU,其中,所述MMU包括实现为执行所述散列函数的散列表。

17.如权利要求16所述的GPU,其中,所述MMU进一步执行线性映射以将主数据地址映射到元数据地址。

18.如权利要求13所述的GPU,进一步包括:

第一结构元件,具有耦合至第一区块的第一MMU,所述第一区块用于存储第一集合的主数据和与所述第一集合的主数据相关联的第一元数据块的元数据;以及

第二结构元件,具有耦合至第二区块的第二MMU,所述第二区块用于存储第二集合的主数据和与所述第二集合的主数据相关联的第二元数据块的元数据。

19.如权利要求18所述的GPU,进一步包括:

第一集合的一个或多个处理节点,耦合至所述第一结构元件;以及

第二集合的一个或多个处理节点,耦合至所述第二结构元件。

20.如权利要求19所述的GPU,其中,所述第一结构元件包括耦合在所述第一集合的一个或多个处理节点与所述第一MMU之间的第一控制高速缓存,所述第一控制高速缓存用于执行数据压缩和解压缩,并且所述第二结构元件包括耦合在所述第二集合的一个或多个处理节点与所述第二MMU之间的第二控制高速缓存,所述第二控制高速缓存用于执行数据压缩和解压缩。

存储器压缩散列机制

技术领域

[0001] 本发明总体上涉及图形处理，并且更具体地涉及存储器数据压缩。

背景技术

[0002] 图形处理单元(GPU)是高度线程化机器，其中并行地执行程序的数百个线程以实现高吞吐量。GPU线程组被实现在网格着色应用中以执行三维(3D)渲染。随着越来越复杂的GPU需要大量计算，保持存储器带宽要求是有挑战的。因此，带宽压缩已经变得关键以确保硬件/存储器子系统能支持所需的带宽。

附图说明

[0003] 为了以能够详细理解本发明的以上记载特征的方式，可通过参考实施例来对以上简要概括的本发明进行更具体的描述，这些实施例中的一些在所附附图中被图示。然而，应注意的是，附图仅展示本发明的典型的实施例，且因此将不被视为限制其范围，因为本发明可以承认其他等效实施例。

[0004] 图1是根据实施例的处理系统的框图；

[0005] 图2是根据实施例的处理器的框图；

[0006] 图3是根据实施例的图形处理器的框图；

[0007] 图4是根据一些实施例的图形处理器的图形处理引擎的框图；

[0008] 图5是由附加实施例提供的图形处理器的框图；

[0009] 图6A和图6B图示线程执行逻辑，该线程执行逻辑包括在一些实施例中采用的处理元件阵列；

[0010] 图7是图示根据一些实施例的图形处理器指令格式的框图；

[0011] 图8是根据另一个实施例的图形处理器的框图；

[0012] 图9A和图9B图示根据一些实施例的图形处理器命令格式和命令序列；

[0013] 图10图示根据一些实施例的用于数据处理系统的示例性图形软件架构；

[0014] 图11A和图11B是图示根据实施例的IP核开发系统的框图；

[0015] 图12是图示根据实施例的示例性芯片上系统集成电路的框图；

[0016] 图13A和图13B是图示附加的示例性图形处理器的框图；

[0017] 图14A和图14B是图示根据实施例的芯片上系统集成电路的附加示例性图形处理器的框图；

[0018] 图15图示计算设备的一个实施例；

[0019] 图16图示图形处理单元的一个实施例；

[0020] 图17图示存储器空间的一个实施例；

[0021] 图18图示重新打包的存储器空间的一个实施例；

[0022] 图19图示存储器管理单元的一个实施例；以及

[0023] 图20是图示用于执行压缩散列的过程的一个实施例的流程图。

具体实施方式

[0024] 在以下描述中,陈述许多具体细节以提供对本发明的更透彻理解。然而,将对本领域技术人员显而易见的是,可在没有这些特定细节中的一个或多个细节的情况下实施本发明。在其他实例中,未描述公知的特征以避免使本发明模糊。

[0025] 在实施例中,存储器管理单元执行散列函数来为主数据和元数据计算到存储器中的物理地址位置中的索引,并且调节元数据物理地址位置以将每个经调节的元数据物理地址位置存储在存储相关联的主数据的区块中。

[0026] 图1是根据实施例的处理系统100的框图。在各实施例中,系统100包括一个或多个处理器102以及一个或多个图形处理器108,并且可以是单处理器台式机系统、多处理器工作站系统或具有大量处理器102或处理器核107的服务器系统。在一个实施例中,系统100是被并入在用于在移动设备、手持式设备或嵌入式设备中使用的芯片上系统(SoC)集成电路内的处理平台。

[0027] 在一个实施例中,系统100可包括以下各项或可并入在以下各项内:基于服务器的游戏平台、包括游戏和媒体控制台的游戏控制台、移动游戏控制台、手持式游戏控制台或在线游戏控制台。在一些实施例中,系统100是移动电话、智能电话、平板计算设备或移动互联网设备。处理系统100也可包括可穿戴设备,可与可穿戴设备耦合或可集成在可穿戴设备内,该可穿戴设备诸如,智能手表可穿戴设备、智能眼镜设备、增强现实设备或虚拟现实设备。在一些实施例中,处理系统100是电视机或机顶盒设备,该电视机或机顶盒设备具有一个或多个处理器102以及由一个或多个图形处理器108生成的图形界面。

[0028] 在一些实施例中,该一个或多个处理器102各自都包括一个或多个处理器核107,该一个或多个处理器核107用于处理指令,这些指令当被执行时,执行用于系统和用户软件的操作。在一些实施例中,一个或多个处理器核107中的每一个处理器核都被配置成处理特定的指令集109。在一些实施例中,指令集109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。多个处理器核107各自都可处理不同的指令集109,不同的指令集109可包括用于促进对其他指令集的仿真的指令。处理器核107也可包括其他处理设备,诸如数字信号处理器(DSP)。

[0029] 在一些实施例中,处理器102包括高速缓存存储器104。取决于架构,处理器102可具有单个内部高速缓存或多级的内部高速缓存。在一些实施例中,高速缓存存储器在处理器102的各种组件之间被共享。在一些实施例中,处理器102也使用外部高速缓存(例如,第3级(L3)高速缓存或末级高速缓存(LLC))(未示出),可使用已知的高速缓存一致性技术在处理器核107之间共享该外部高速缓存。寄存器堆106附加地被包括在处理器102中,寄存器堆106可包括用于存储不同类型数据的不同类型的寄存器(例如,整数寄存器、浮点寄存器、状态寄存器以及指令指针寄存器)。一些寄存器可以是通用寄存器,而其他寄存器可专用于处理器102的设计。

[0030] 在一些实施例中,一个或多个处理器102与一个或多个接口总线110耦合,以在处理器102与系统100中的其他组件之间传输通信信号,诸如,地址、数据、或控制信号。在一个实施例中,接口总线110可以是处理器总线,诸如,直接媒体接口(DMI)总线的某个版本。然而,处理器总线不限于DMI总线,并且可包括一个或多个外围组件互连总线(例如,PCI、PCI Express)、存储器总线或其他类型的接口总线。在一个实施例中,(多个)处理器102包括集

成存储器控制器116和平台控制器中枢130。存储器控制器116促进存储器设备与系统100的其他组件之间的通信，而平台控制器中枢(PCH)130提供经由本地I/O总线至I/O设备的连接。

[0031] 存储器设备120可以是动态随机存取存储器(DRAM)设备、静态随机存取存储器(SRAM)设备、闪存设备、相变存储器设备、或具有适当的性能以充当进程存储器的某个其他存储器设备。在一个实施例中，存储器设备120可以作为用于系统100的系统存储器来操作，以存储数据122和指令121供在一个或多个处理器102执行应用或进程时使用。存储器控制器116也与任选的外部图形处理器112耦合，该任选的外部图形处理器112可与处理器102中的一个或多个图形处理器108通信以执行图形操作和媒体操作。在一些实施例中，显示设备111可以连接至(多个)处理器102。显示设备111可以是以下各项中的一项或多项：内部显示设备，如在移动电子设备或膝上型设备中；或经由显示接口(例如，显示端口等)附接的外部显示设备。在一个实施例中，显示设备111可以是头戴式显示器(HMD)，诸如，用于在虚拟现实(VR)应用或增强现实(AR)应用中使用的立体显示设备。

[0032] 在一些实施例中，平台控制器中枢130使外围设备能够经由高速I/O总线而连接至存储器设备120和处理器102。I/O外围设备包括但不限于音频控制器146、网络控制器134、固件接口128、无线收发器126、触摸传感器125、数据存储设备124(例如，硬盘驱动器、闪存等)。数据存储设备124可以经由存储接口(例如，SATA)或经由如外围组件互连总线(例如，PCI、PCI Express)等外围总线来进行连接。触摸传感器125可以包括触摸屏传感器、压力传感器、或指纹传感器。无线收发器126可以是Wi-Fi收发器、蓝牙收发器、或移动网络收发器，该移动网络收发器诸如3G、4G或长期演进(LTE)收发器。固件接口128使得能够与系统固件进行通信，并且可以例如是统一可扩展固件接口(UEFI)。网络控制器134可启用到有线网络的网络连接。在一些实施例中，高性能网络控制器(未示出)与接口总线110耦合。在一个实施例中，音频控制器146是多声道高清音频控制器。在一个实施例中，系统100包括用于将传统(例如，个人系统2(PS/2))设备耦合至系统的任选的传统I/O控制器140。平台控制器中枢130还可以连接至一个或多个通用串行总线(USB)控制器142连接输入设备，诸如，键盘和鼠标143组合、相机144、或其他USB输入设备。

[0033] 将会理解，所示的系统100是示例性的而非限制性的，因为也可以使用以不同方式配置的其他类型的数据处理系统。例如，存储器控制器116和平台控制器中枢130的实例可以集成到分立的外部图形处理器中，该分立的外部图形处理器诸如外部图形处理器112。在一个实施例中，平台控制器中枢130和/或存储器控制器116可以在一个或多个处理器102外部。例如，系统100可包括外部存储器控制器116和平台控制器中枢130，该外部存储器控制器116和平台控制器中枢130可以被配置为在与(多个)处理器102通信的系统芯片组内的存储器控制器中枢和外围控制器中枢。

[0034] 图2是处理器200的实施例的框图，该处理器200具有一个或多个处理器核202A-202N、集成存储器控制器214以及集成图形处理器208。图2的具有与本文中的任何其他附图的元件相同的附图标记(或名称)的那些元件能以类似于本文中其他地方描述的任何方式操作或起作用，但不限于此。处理器200可包括附加的核，这些附加的核多至由虚线框表示的附加核202N并包括由虚线框表示的附加核202N。处理器核202A-202N中的每一个包括一个或多个内部高速缓存单元204A-204N。在一些实施例中，每一个处理器核也具有对一个或

多个共享高速缓存单元206的访问权。

[0035] 内部高速缓存单元204A-204N和共享高速缓存单元206表示处理器200内的高速缓存存储器层级结构。高速缓存存储器层级结构可包括每个处理器核内的至少一个级别的指令和数据高速缓存以及一级或多级共享的中级高速缓存，诸如，第2级(L2)、第3级(L3)、第4级(L4)、或其他级别的高速缓存，其中，在外部存储器之前的最高级别的高速缓存被分类为LLC。在一些实施例中，高速缓存一致性逻辑维持各高速缓存单元206与204A-204N之间的一致性。

[0036] 在一些实施例中，处理器200还可包括一个或多个总线控制器单元的集合216和系统代理核210。一个或多个总线控制器单元216管理外围总线的集合，诸如一个或多个PCI总线或PCI Express总线。系统代理核210提供对各处理器组件的管理功能。在一些实施例中，系统代理核210包括用于管理对各种外部存储器设备(未示出)的访问的一个或多个集成存储器控制器214。

[0037] 在一些实施例中，处理器核202A-202N中的一个或多个处理器核包括对同步多线程的支持。在此类实施例中，系统代理核210包括用于在多线程处理期间协调并操作核202A-202N的组件。系统代理核210可附加地包括功率控制单元(PCU)，该功率控制单元包括用于调节处理器核202A-202N和图形处理器208的功率状态的逻辑和组件。

[0038] 在一些实施例中，处理器200附加地包括用于执行图形处理操作的图形处理器208。在一些实施例中，图形处理器208耦合至共享高速缓存单元的集合206以及系统代理核210，该系统代理核210包括一个或多个集成存储器控制器214。在一些实施例中，系统代理核210还包括用于将图形处理器输出驱动到一个或多个经耦合的显示器的显示控制器211。在一些实施例中，显示控制器211还可以是经由至少一个互连与图形处理器耦合的单独模块，或者可以集成在图形处理器208内。

[0039] 在一些实施例中，基于环的互连单元212用于耦合处理器200的内部组件。然而，可以使用替代的互连单元，诸如，点到点互连、交换式互连、或其他技术，包括本领域中公知的技术。在一些实施例中，图形处理器208经由I/O链路213与环形互连212耦合。

[0040] 示例性I/O链路213表示多个各种各样的I/O互连中的至少一种，包括促进各处理器组件与高性能嵌入式存储器模块218(诸如，eDRAM模块)之间的通信的封装上I/O互连。在一些实施例中，处理器核202A-202N中的每个处理器核以及图形处理器208将嵌入式存储器模块218用作共享的末级高速缓存。

[0041] 在一些实施例中，处理器核202A至202N是执行相同指令集架构的同构核。在另一实施例中，处理器核202A-202N在指令集架构(ISA)方面是异构的，其中，处理器核202A-202N中的一个或多个执行第一指令集，而其他核中的至少一个执行第一指令集的子集或不同的指令集。在一个实施例中，处理器核202A-202N在微架构方面是异构的，其中，具有相对较高功耗的一个或多个核与具有较低功耗的一个或多个功率核耦合。此外，处理器200可实现在一个或多个芯片上，或者除其他组件之外也被实现为具有所图示的组件的SoC集成电路。

[0042] 图3是图形处理器300的框图，该图形处理器300可以是分立式图形处理单元，或者可以是集成有多个处理核的图形处理器。在一些实施例中，图形处理器经由到图形处理器上的寄存器的存储器映射的I/O接口并且利用被放置到处理器存储器中的命令进行通信。

在一些实施例中,图形处理器300包括用于访问存储器的存储器接口314。存储器接口314可以是到本地存储器、一个或多个内部高速缓存、一个或多个共享的外部高速缓存、和/或到系统存储器的接口。

[0043] 在一些实施例中,图形处理器300还包括显示控制器302,该显示控制器302用于将显示输出数据驱动到显示设备320。显示控制器302包括用于显示器的一个或多个叠加平面以及多层的视频或用户界面元素的合成的硬件。显示设备320可以是内部或外部显示设备。在一个实施例中,显示设备320是头戴式显示设备,诸如,虚拟现实(VR)显示设备或增强现实(AR)显示设备。在一些实施例中,图形处理器300包括用于将媒体编码到一种或多种媒体编码格式,从一种或多种媒体编码格式对媒体解码,或在一种或多种媒体编码格式之间对媒体转码的视频编解码器引擎306,这一种或多种媒体编码格式包括但不限于:移动图像专家组(MPEG)格式(诸如,MPEG-2)、高级视频译码(AVC)格式(诸如,H.264/MPEG-4AVC)、以及电影和电视工程师协会(SMPTE)421M/VC-1、和联合图像专家组(JPEG)格式(诸如,JPEG、以及运动JPEG(MJPEG)格式)。

[0044] 在一些实施例中,图形处理器300包括块图像传送(BLIT)引擎304,用于执行二维(2D)栅格化器操作,包括例如,位边界块传送。然而,在一个实施例中,使用图形处理引擎(GPE)310的一个或多个组件执行2D图形操作。在一些实施例中,GPE 310是用于执行图形操作的计算引擎,这些图形操作包括三维(3D)图形操作和媒体操作。

[0045] 在一些实施例中,GPE 310包括用于执行3D操作的3D流水线312,3D操作诸如,使用作用于3D基元形状(例如,矩形、三角形等)的处理函数来渲染三维图像和场景。3D流水线312包括可编程和固定功能元件,该可编程和固定功能元件执行到3D/媒体子系统315的元件和/或所生成的执行线程内的各种任务。虽然3D流水线312可用于执行媒体操作,但是GPE 310的实施例还包括媒体流水线316,该媒体流水线316专门用于执行媒体操作,诸如,视频后处理和图像增强。

[0046] 在一些实施例中,媒体流水线316包括固定功能或可编程逻辑单元用于代替、或代表视频编解码器引擎306来执行一个或多个专业的媒体操作,诸如,视频解码加速、视频去隔行、以及视频编码加速。在一些实施例中,媒体流水线316附加地包括线程生成单元以生成用于在3D/媒体子系统315上执行的线程。所生成的线程在3D/媒体子系统315中所包括的一个或多个图形执行单元上执行对媒体操作的计算。

[0047] 在一些实施例中,3D/媒体子系统315包括用于执行由3D流水线312和媒体流水线316生成的线程的逻辑。在一个实施例中,流水线向3D/媒体子系统315发送线程执行请求,该3D/媒体子系统315包括用于对于对可用的线程执行资源的各种请求进行仲裁和分派的线程分派逻辑。执行资源包括用于处理3D线程和媒体线程的图形执行单元的阵列。在一些实施例中,3D/媒体子系统315包括用于线程指令和数据的一个或多个内部高速缓存。在一些实施例中,该子系统还包括用于在线程之间共享数据并用于存储输出数据的共享存储器,其包括寄存器和可寻址存储器。

图形处理引擎

[0048] 图4是根据一些实施例的图形处理器的图形处理引擎410的框图。在一个实施例中,图形处理引擎(GPE)410是图3中示出的GPE 310的某个版本。图4的具有与本文中的任何其他附图的元件相同的附图标记(或名称)的那些元件能以类似于本文中其他地方描述的

任何方式操作或起作用,但不限于此。例如,图示出图3的3D流水线312和媒体流水线316。媒体流水线316在GPE 410的一些实施例中是任选的,并且可以不显式地被包括在GPE 410内。例如并且在至少一个实施例中,单独的媒体和/或图像处理器被耦合至GPE410。

[0049] 在一些实施例中,GPE 410与命令流转化器403耦合或包括命令流转化器403,该命令流转化器403将命令流提供给3D流水线312和/或媒体流水线316。在一些实施例中,命令流转化器403与存储器耦合,该存储器可以是系统存储器、或内部高速缓存存储器和共享高速缓存存储器中的一个或多个。在一些实施例中,命令流转化器403从存储器接收命令,并将这些命令发送至3D流水线312和/或媒体流水线316。这些命令是从环形缓冲器取出的指示,该环形缓冲器存储用于3D流水线312和媒体流水线316的命令。在一个实施例中,环形缓冲器可附加地包括存储批量的多个命令的批量命令缓冲器。用于3D流水线312的命令还可包括对存储在存储器中的数据的引用,这些数据诸如但不限于用于3D流水线312的顶点数据和几何数据和/或用于媒体流水线316的图像数据和存储器对象。3D流水线312和媒体流水线316通过经由各自流水线内的逻辑执行操作或者通过将一个或多个执行线程分派至图形核阵列414来处理命令和数据。在一个实施例中,图形核阵列414包括一个或多个图形核(例如,(多个)图形核415A、(多个)图形核415B)的块,每个块包括一个或多个图形核。每个图形核包括图形执行资源的集合,该图形执行资源的集合包括:用于执行图形操作和计算操作的通用执行逻辑和图形专用执行逻辑;以及固定功能纹理处理逻辑和/或机器学习和人工智能加速逻辑。

[0050] 在各实施例中,3D流水线312包括固定功能逻辑和可编程逻辑,用于通过处理指令并将执行线程分派给图形核阵列414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、计算着色器或其他着色器程序。图形核阵列414提供统一的执行资源块供在处理这些着色器程序时使用。图形核阵列414的(多个)图形核415A-415B内的多功能执行逻辑(例如,执行单元)包括对各种3D API着色器语言的支持,并且可执行与多个着色器相关联的多个同步执行线程。

[0051] 在一些实施例中,图形核阵列414还包括用于执行诸如视频和/或图像处理的媒体功能的执行逻辑。在一个实施例中,执行单元附加地包括通用逻辑,该通用逻辑可编程以便除了执行图形处理操作之外还执行并行通用计算操作。通用逻辑可与图1的(多个)处理器核107或图2中的核202A-202N内的通用逻辑并行地或结合地执行处理操作。

[0052] 由在图形核阵列414上执行的线程生成的输出数据可以将数据输出到统一返回缓冲器(URB)418中的存储器。URB 418可以存储用于多个线程的数据。在一些实施例中,URB 418可用于在图形核阵列414上执行的不同线程之间发送数据。在一些实施例中,URB 418可附加地用于在图形核阵列上的线程与共享功能逻辑420内的固定功能逻辑之间的同步。

[0053] 在一些实施例中,图形核阵列414是可缩放的,使得阵列包括可变数量的图形核,每个图形核都具有基于GPE 410的目标功率和性能等级的可变数量的执行单元。在一个实施例中,执行资源是动态可缩放的,从而可以根据需要启用或禁用执行资源。

[0054] 图形核阵列414与共享功能逻辑420耦合,该共享功能逻辑420包括在图形核阵列中的图形核之间被共享的多个资源。共享功能逻辑420内的共享功能是向图形核阵列414提供专业的补充功能的硬件逻辑单元。在各实施例中,共享功能逻辑420包括但不限于采样器421逻辑、数学422逻辑和线程间通信(ITC)423逻辑。另外,一些实施例在共享功能逻辑420

内实现一个或多个高速缓存425。

[0055] 在对于给定的专业功能的需求不足以包括在图形核阵列414中的情况下实现共享功能。相反，那个专业功能的单个实例化被实现为共享功能逻辑420中的独立实体，并且在图形核阵列414内的执行资源之间被共享。在图形核阵列414之间被共享并被包括在图形核阵列414内的确切的功能集因实施例而异。在一些实施例中，共享功能逻辑420内的由图形核阵列414广泛使用的特定共享功能可被包括在图形核阵列414内的共享功能逻辑416内。在各实施例中，图形核阵列414内的共享功能逻辑416可包括共享功能逻辑420内的一些或所有逻辑。在一个实施例中，共享功能逻辑420内的所有逻辑元件可以在图形核阵列414的共享功能逻辑416内被复制。在一个实施例中，共享功能逻辑420被排除以有利于图形核阵列414内的共享功能逻辑416。

[0056] 图5是根据本文中所描述的一些实施例的图形处理器核500的硬件逻辑的框图。图5的具有与本文中的任何其他附图的元件相同的附图标记(或名称)的那些元件能以类似于本文中其他地方描述的任何方式操作或起作用，但不限于此。在一些实施例中，所图示的图形处理器核500被包括在图4的图形核阵列414内。图形处理器核500(有时称为核切片)可以是模块化图形处理器内的一个或多个图形核。图形处理器核500的示例是一个图形核切片，并且基于目标功率包络和性能包络，如本文中所描述的图形处理器可以包括多个图形核切片。每个图形核500可包括固定功能块530，该固定功能块530与多个子核501A-501F(也称为子切片)耦合，多个子核501A-501F包括模块化的通用和固定功能逻辑的块。

[0057] 在一些实施例中，固定功能块530包括几何/固定功能流水线536，该几何/固定功能流水线536例如在较低性能和/或较低功率的图形处理器实现中可由图形处理器500中的所有子核共享。在各实施例中，几何/固定功能流水线536包括3D固定功能流水线(例如，如在图3和图4中的3D流水线312)、视频前端单元、线程生成器和线程分派器、以及统一返回缓冲器管理器，该统一返回缓冲器管理器管理诸如图4的统一返回缓冲器418的统一返回缓冲器。

[0058] 在一个实施例中，固定功能块530还包括图形SoC接口537、图形微控制器538和媒体流水线539。图形SoC接口537提供图形核500与芯片上系统集成电路内的其他处理器核之间的接口。图形微控制器538是可配置成管理图形处理器500的各种功能的可编程子处理器，这些功能包括线程分派、调度和抢占。媒体流水线539(例如，图3和图4的媒体流水线316)包括用于促进对包括图像数据和视频数据的多媒体数据进行解码、编码、预处理和/或后处理的逻辑。媒体流水线539经由对子核501A-501F内的计算或采样逻辑的请求来实现媒体操作。

[0059] 在一个实施例中，SoC接口537使图形核500能够与通用应用处理器核(例如，CPU)和/或SoC内的其他组件进行通信，其他组件包括诸如共享的末级高速缓存存储器的存储器层级结构元件、系统RAM、和/或嵌入式芯片上或封装上DRAM。SoC接口537还可启用与SoC内的诸如相机成像流水线的固定功能设备的通信，并且启用全局存储器原子性的使用和/或实现全局存储器原子性，该全局存储器原子性可在图形核500与SoC内的CPU之间被共享。SoC接口537还可实现针对图形核500的功率管理控制，并且启用图形核500的时钟域与SoC内的其他时钟域之间的接口。在一个实施例中，SoC接口537使得能够从命令流转化器和全局线程分派器接收命令缓冲器，该命令流转化器和全局线程分派器被配置成将命令和指令

提供给图形处理器内的一个或多个图形核中的每一个图形核。当媒体操作将要执行时，这些命令和指令可以被分派给媒体流水线539，或者当图形处理操作将要执行时，这些命令和指令可以被分派给几何和固定功能流水线(例如，几何和固定功能流水线536、几何和固定功能流水线514)。

[0060] 图形微控制器538可被配置成执行针对图形核500的各种调度任务和管理任务。在一个实施例中，图形微控制器538可对子核501A-501F内的执行单元(EU)阵列502A-502F、504A-504F内的各个图形并行引擎执行图形和/或计算工作负载调度。在该调度模型中，在包括图形核500的SoC的CPU核上执行的主机软件可以经由多个图形处理器门铃(doorbell)中的一个图形处理器门铃来提交工作负载，这调用了对适当的图形引擎的调度操作。调度操作包括：确定接下来要运行哪个工作负载，将工作负载提交到命令流转化器，抢占在引擎上运行的现有工作负载，监测工作负载的进度，以及当工作负载完成时通知主机软件。在一个实施例中，图形微控制器538还可促进图形核500的低功率或空闲状态，从而向图形核500提供独立于操作系统和/或系统上的图形驱动器软件跨低功率状态转变来保存和恢复图形核500内的寄存器的能力。

[0061] 图形核500可具有多于或少于所图示的子核501A-501F，多达N个模块化子核。对于每组N个子核，图形核500还可包括共享功能逻辑510、共享和/或高速缓存存储器512、几何/固定功能流水线514、以及用于加速各种图形和计算处理操作的附加的固定功能逻辑516。共享功能逻辑510可以包括与可由图形核500内的每N个子核共享的、与图4的共享功能逻辑420(例如，采样器逻辑、数学逻辑、和/或线程间通信逻辑)相关联的逻辑单元。共享和/或高速缓存存储器512可以是用于图形核500内的N个子核的集合501A-501F的末级高速缓存，并且还可以充当可由多个子核访问的共享存储器。几何/固定功能流水线514而不是几何/固定功能流水线536可被包括在固定功能块530内，并且几何/固定功能流水线514可包括相同或类似的逻辑单元。

[0062] 在一个实施例中，图形核500包括附加的固定功能逻辑516，该附加的固定功能逻辑516可包括供由图形核500使用的各种固定功能加速逻辑。在一个实施例中，附加的固定功能逻辑516包括供在仅位置着色中使用的附加的几何流水线。在仅位置着色中，存在两个几何流水线：几何/固定功能流水线516、536内的完全几何流水线；以及剔除流水线，其是可被包括在附加的固定功能逻辑516内的附加的几何流水线。在一个实施例中，剔除流水线是完全几何流水线的精简版本。完全流水线和剔除流水线可以执行同一应用的不同实例，每个实例具有单独的上下文。仅位置着色可以隐藏被丢弃三角形的长剔除运行，从而在一些实例中使得能够更早地完成着色。例如并且在一个实施例中，附加的固定功能逻辑516内的剔除流水线逻辑可以与主应用并行地执行位置着色器，并且通常比完全流水线更快地生成关键结果，因为剔除流水线仅取出顶点的位置属性并对顶点的位置属性进行着色，而不向帧缓冲器执行对像素的栅格化和渲染。剔除流水线可以使用所生成的关键结果来计算所有三角形的可见性信息，而无需考虑那些三角形是否被剔除。完全流水线(其在本实例中可以被称为重放(replay)流水线)可以消耗该可见性信息以跳过被剔除的三角形，从而仅对最终被传递到栅格化阶段的可见的三角形进行着色。

[0063] 在一个实施例中，附加的固定功能逻辑516还可包括机器学习加速逻辑，诸如，固定功能矩阵乘法逻辑，该机器学习加速逻辑用于包括针对机器学习训练或推断的实现方

式。

[0064] 在每个图形子核501A-501F内包括可用于响应于由图形流水线、媒体流水线、或着色器程序作出的请求而执行图形操作、媒体操作和计算操作的执行资源的集合。图形子核501A-501F包括：多个EU阵列502A-502F、504A-504F；线程分派和线程间通信(TD/IC)逻辑503A-503F；3D(例如，纹理)采样器505A-505F；媒体采样器506A-506F；着色器处理器507A-507F；以及共享的本地存储器(SLM)508A-508F。EU阵列502A-502F、504A-504F各自包括多个执行单元，这些执行单元为能够执行浮点和整数/定点逻辑操作以服务于图形操作、媒体操作或计算操作(包括图形程序、媒体程序或计算着色器程序)的通用图形处理单元。TD/IC逻辑503A-503F执行针对子核内的执行单元的本地线程分派和线程控制操作，并且促进在子核的执行单元上执行的线程之间的通信。3D采样器505A-505F可将纹理或其他3D图形相关的数据读取到存储器中。3D采样器可以基于所配置的样本状态以及与给定纹理相关联的纹理格式以不同方式读取纹理数据。媒体采样器506A-506F可基于与媒体数据相关联的类型和格式来执行类似的读取操作。在一个实施例中，每个图形子核501A-501F可以交替地包括统一3D和媒体采样器。在子核501A-501F中的每一个子核内的执行单元上执行的线程可利用每个子核内的共享的本地存储器508A-508F，以使在线程组内执行的线程能够使用芯片上存储器的公共池来执行。

执行单元

[0065] 图6A-图6B图示根据本文中所描述的实施例的线程执行逻辑600，该线程执行逻辑600包括在图形处理器核中采用的处理元件阵列。图6A-图6B的具有与本文中的任何其他附图的元件相同的附图标记(或名称)的那些元件能以类似于本文中其他地方描述的任何方式操作或起作用，但不限于此。图6A图示线程执行逻辑600的概览，该线程执行逻辑600可包括对于图5的每个子核501A-501F图示的硬件逻辑的变体。图6B图示执行单元的示例性内部细节。

[0066] 如在图6A中所图示，在一些实施例中，线程执行逻辑600包括着色器处理器602、线程分派器604、指令高速缓存606、包括多个执行单元608A-608N的可缩放执行单元阵列、采样器610、数据高速缓存612、以及数据端口614。在一个实施例中，可缩放执行单元阵列可通过基于工作负载的计算要求启用或禁用一个或多个执行单元(例如，执行单元608A、608B、608C、608D，一直到608N-1和608N中的任一个)来动态地缩放。在一个实施例中，所包括的组件经由互连结构而互连，该互连结构链接到组件中的每个组件。在一些实施例中，线程执行逻辑600包括通过指令高速缓存606、数据端口614、采样器610、以及执行单元608A-608N中的一个或多个到存储器(诸如，系统存储器或高速缓存存储器)的一个或多个连接。在一些实施例中，每个执行单元(例如，608A)是能够执行多个同步硬件线程同时针对每个线程并行地处理多个数据元素的独立式可编程通用计算单元。在各实施例中，执行单元608A-608N的阵列是可缩放的以包括任何数量的单独执行单元。

[0067] 在一些实施例中，执行单元608A-608N主要用于执行着色器程序。着色器处理器602可处理各种着色器程序，并且可经由线程分派器604分派与着色器程序相关联的执行线程。在一个实施例中，线程分派器包括用于对来自图形流水线和媒体流水线的线程发起请求进行仲裁并在执行单元608A-608N中的一个或多个执行单元上实例化所请求的线程的逻辑。例如，几何流水线可将顶点着色器、曲面细分着色器或几何着色器分派到现场执行逻辑

以用于处理。在一些实施例中，线程分派器604还可处理来自执行的着色器程序的运行时间线程生成请求。

[0068] 在一些实施例中，执行单元608A-608N支持包括对许多标准3D图形着色器指令的原生支持的指令集，使得以最小的转换执行来自图形库(例如，Direct 3D和OpenGL)的着色器程序。这些执行单元支持顶点和几何处理(例如，顶点程序、几何程序、顶点着色器)、像素处理(例如，像素着色器、片段着色器)以及通用处理(例如，计算和媒体着色器)。执行单元608A-608N中的每个执行单元都能够进行多发布单指令多数据(SIMD)执行，并且多线程操作在面对较高等待时间的存储器访问时启用高效的执行环境。每个执行单元内的每个硬件线程都具有专用的高带宽寄存器堆和相关的独立线程状态。对于能够进行整数操作、单精度浮点操作和双精度浮点操作、能够具有SIMD分支能力、能够进行逻辑操作、能够进行超越操作和能够进行其他混杂操作的流水线，执行是针对每个时钟多发布的。在等待来自存储器或共享功能之一的数据时，执行单元608A-608N内的依赖性逻辑使等待的线程休眠，直到所请求的数据已返回。当等待的线程正在休眠时，硬件资源可致力于处理其他线程。例如，在与顶点着色器操作相关联的延迟期间，执行单元可以执行针对像素着色器、片段着色器或包括不同顶点着色器的另一类型的着色器程序的操作。

[0069] 执行单元608A-608N中的每个执行单元对数据元素的数组进行操作。数据元素的数量是“执行尺寸”、或用于指令的通道数量。执行通道是用于数据元素访问、掩码、和指令内的流控制的执行的逻辑单元。通道的数量可独立于用于特定图形处理器的物理算术逻辑单元(ALU)或浮点单元(FPU)的数量。在一些实施例中，执行单元608A-608N支持整数和浮点数据类型。

[0070] 执行单元指令集包括SIMD指令。各种数据元素可以作为紧缩数据类型存储在寄存器中，并且执行单元将基于元素的数据尺寸来处理各个元素。例如，当对256位宽的向量进行操作时，向量的256位被存储在寄存器中，并且执行单元将向量操作为四个单独的64位紧缩数据元素(四字(QW)尺寸数据元素)、八个单独的32位紧缩数据元素(双字(DW)尺寸数据元素)、十六个单独的16位紧缩数据元素(字(W)尺寸的数据元素)、或三十二个单独的8位数据元素(字节(B)尺寸的数据元素)。然而，不同的向量宽度和寄存器尺寸是可能的。

[0071] 在一个实施例中，可以将一个或多个执行单元组合到融合执行单元609A-609N中，该融合执行单元609A-609N具有对于融合EU而言共同的线程控制逻辑(607A-607N)。可以将多个EU融合到一EU组中。融合的EU组中的每个EU可以被配置成执行单独的SIMD硬件线程。融合的EU组中的EU的数量可以根据实施例而有所不同。另外，可以逐EU地执行各种SIMD宽度，包括但不限于SIMD8、SIMD16和SIMD32。每个融合图形执行单元609A-609N包括至少两个执行单元。例如，融合执行单元609A包括第一EU 608A、第二EU 608B、以及对于第一EU 608A和第二EU 608B而言共同的线程控制逻辑607A。线程控制逻辑607A控制在融合图形执行单元609A上执行的线程，从而允许融合执行单元609A-609N内的每个EU使用共同的指令指针寄存器来执行。

[0072] 一个或多个内部指令高速缓存(例如，606)被包括在线程执行逻辑600中，以对用于执行单元的线程指令进行高速缓存。在一些实施例中，一个或多个数据高速缓存(例如，612)被包括，以在线程执行期间对线程数据进行高速缓存。在一些实施例中，采样器610被包括以为3D操作提供纹理采样并且为媒体操作提供媒体采样。在一些实施例中，采样器610

包括专业的纹理或媒体采样功能,以便在向执行单元提供采样数据之前在采样过程期间处理纹理数据或媒体数据。

[0073] 在执行期间,图形流水线和媒体流水线经由线程生成和分派逻辑将线程发起请求发送到线程执行逻辑600。一旦一组几何对象已经被处理并被栅格化为像素数据,着色器处理器602内的像素处理器逻辑(例如,像素着色器逻辑、片段着色器逻辑等)就被调用以进一步计算输出信息,并且使得结果被写入到输出表面(例如,颜色缓冲器、深度缓冲器、模板印刷(stencil)缓冲器等)。在一些实施例中,像素着色器或片段着色器计算各顶点属性的值,各顶点属性的值将跨经栅格化的对象而被内插。在一些实施例中,着色器处理器602内的像素处理器逻辑随后执行应用编程接口(API)供应的像素着色器程序或片段着色器程序。为了执行着色器程序,着色器处理器602经由线程分派器604将线程分派至执行单元(例如,608A)。在一些实施例中,着色器处理器602使用采样器610中的纹理采样逻辑来访问存储在存储器中的纹理图中的纹理数据。对纹理数据和输入几何数据的算术操作计算针对每个几何片段的像素颜色数据,或丢弃一个或多个像素而不进行进一步处理。

[0074] 在一些实施例中,数据端口614提供存储器访问机制,供线程执行逻辑600将经处理的数据输出至存储器以便在图形处理器输出流水线上进一步处理。在一些实施例中,数据端口614包括或耦合至一个或多个高速缓存存储器(例如,数据高速缓存612),以便对数据进行高速缓存供经由数据端口进行存储器访问。

[0075] 如图6B中所图示,图形执行单元608可包括指令取出单元637、通用寄存器堆阵列(GRF)624、架构寄存器堆阵列(ARF)626、线程仲裁器622、发送单元630、分支单元632、SIMD浮点单元的集合(FPU)634、以及在一个实施例中的专用整数SIMD ALU的集合635。GRF 624和ARF 626包括与可在图形执行单元608中活跃的每个同步硬件线程相关联的通用寄存器堆和架构寄存器堆的集合。在一个实施例中,每线程架构状态被维持在ARF 626中,而在线程执行期间使用的数据被存储在GRF 624中。每个线程的执行状态,包括用于每个线程的指令指针,可以被保持在ARF 626中的线程专用寄存器中。

[0076] 在一个实施例中,图形执行单元608具有作为同步多线程(SMT)与细粒度交织多线程(IMT)的组合的架构。该架构具有模块化配置,该模块化配置可以基于同步线程的目标数量和每个执行单元的寄存器的数量而在设计时进行微调,其中跨用于执行多个同步线程的逻辑来划分执行单元资源。

[0077] 在一个实施例中,图形执行单元608可协同发布多条指令,这些指令可以各自是不同的指令。图形执行单元线程608的线程仲裁器622可以将指令分派给以下各项中的一项以供执行:发送单元630、分支单元632或(多个)SIMD FPU 634。每个执行线程可以访问GRF 624内的128个通用寄存器,其中,每个寄存器可以存储可作为具有32位数据元素的SIMD 8元素向量访问的32个字节。在一个实施例中,每个执行单元线程具有对GRF 624内的4个千字节的访问权,但是实施例并不限于此,并且在其他实施例中可以提供更多或更少的寄存器资源。在一个实施例中,多达七个线程可以同步执行,但是每执行单元的线程数量还可以根据实施例而有所不同。在其中七个线程可以访问4个千字节的实施例中,GRF 624可以存储总共28个千字节。灵活的寻址模式可以准许对多个寄存器一起进行寻址,从而建立实际上更宽的寄存器或者表示跨步式矩形块数据结构。

[0078] 在一个实施例中,经由通过消息传递发送单元630执行的“发送”指令来分派存储

器操作、采样器操作以及其他较长等待时间的系统通信。在一个实施例中，分支指令被分派给专用分支单元632以促进SIMD发散和最终收敛。

[0079] 在一个实施例中，图形执行单元608包括用于执行浮点操作的一个或多个SIMD浮点单元(FPU)634。在一个实施例中，(多个)FPU 634还支持整数计算。在一个实施例中，(多个)FPU 634可以SIMD执行多达数量M个32位浮点(或整数)操作，或者SIMD执行多达2M个16位整数或16位浮点操作。在一个实施例中，(多个)FPU中的至少一个提供支持高吞吐量超越数学函数和双精度64位浮点的扩展数学能力。在一些实施例中，8位整数SIMD ALU的集合635也存在，并且可专门优化成执行与机器学习计算相关联的操作。

[0080] 在一个实施例中，可以在图形子核分组(例如，子切片)中对图形执行单元608的多个实例的阵列进行实例化。为了可缩放性，产品架构师可以选择每子核分组的执行单元的确切数量。在一个实施例中，执行单元608可以跨多个执行通道来执行指令。在进一步的实施例中，在不同通道上执行在图形执行单元608上执行的每个线程。

[0081] 图7是图示根据一些实施例的图形处理器指令格式700的框图。在一个或多个实施例中，图形处理器执行单元支持具有多种格式的指令的指令集。实线框图示通常被包括在执行单元指令中的组成部分，而虚线包括任选的或仅被包括在指令的子集中的组成部分。在一些实施例中，所描述和图示的指令格式700是宏指令，因为它们是供应至执行单元的指令，这与产生自一旦指令被处理就进行的指令解码的微指令相反。

[0082] 在一些实施例中，图形处理器执行单元原生地支持128位指令格式710的指令。基于所选择的指令、指令选项和操作数数量，64位紧凑指令格式730可用于一些指令。原生128位指令格式710提供对所有指令选项的访问，而一些选项和操作在64位格式730中受限。64位格式730中可用的原生指令因实施例而异。在一些实施例中，使用索引字段713中的索引值的集合将指令部分地压缩。执行单元硬件基于索引值来引用压缩表的集合，并使用压缩表输出来重构128位指令格式710的原生指令。

[0083] 针对每种格式，指令操作码712限定执行单元要执行的操作。执行单元跨每个操作数的多个数据元素并行地执行每条指令。例如，响应于加法指令，执行单元跨表示纹理元素或图片元素的每个颜色通道执行同步加法操作。默认地，执行单元跨操作数的所有数据通道执行每条指令。在一些实施例中，指令控制字段714启用对某些执行选项的控制，这些执行选项诸如通道选择(例如，断言)以及数据通道顺序(例如，混合)。针对128位指令格式710的指令，执行尺寸字段716限制将被并行地执行的数据通道的数量。在一些实施例中，执行尺寸字段716不可用于64位紧凑指令格式730。

[0084] 一些执行单元指令具有多达三个操作数，包括两个源操作数src0720、src1 722以及一个目的地操作数718。在一些实施例中，执行单元支持双目的地指令，其中，双目的地中的一个目的地是隐式的。数据操纵指令可具有第三源操作数(例如，SRC2 724)，其中，指令操作码712确定源操作数的数量。指令的最后一个源操作数可以是与指令一起传递的立即数(例如，硬编码的)值。

[0085] 在一些实施例中，128位指令格式710包括访问/寻址模式字段726，该访问/寻址模式字段726例如指定使用直接寄存器寻址模式还是间接寄存器寻址模式。当使用直接寄存器寻址模式时，由指令中的位直接提供一个或多个操作数的寄存器地址。

[0086] 在一些实施例中，128位指令格式710包括访问/寻址模式字段726，该访问/寻址模

式字段726指定指令的寻址模式和/或访问模式。在一个实施例中，访问模式用于限定针对指令的数据访问对齐。一些实施例支持包括16字节对齐访问模式和1字节对齐访问模式的访问模式，其中，访问模式的字节对齐确定指令操作数的访问对齐。例如，当处于第一模式时，指令可将字节对齐寻址用于源操作数和目的地操作数，并且当处于第二模式时，指令可将16字节对齐寻址用于所有的源操作数和目的地操作数。

[0087] 在一个实施例中，访问/寻址模式字段726的寻址模式部分确定指令要使用直接寻址还是间接寻址。当使用直接寄存器寻址模式时，指令中的位直接提供一个或多个操作数的寄存器地址。当使用间接寄存器寻址模式时，可以基于指令中的地址寄存器值和地址立即数字段来计算一个或多个操作数的寄存器地址。

[0088] 在一些实施例中，基于操作码712位字段对指令进行分组从而简化操作码解码740。针对8位的操作码，位4、位5、和位6允许执行单元确定操作码的类型。所示出的确切的操作码分组仅是示例。在一些实施例中，移动和逻辑操作码组742包括数据移动和逻辑指令(例如，移动(mov)、比较(cmp))。在一些实施例中，移动和逻辑组742共享五个最高有效位(MSB)，其中，移动(mov)指令采用0000xxxxb的形式，而逻辑指令采用0001xxxxb的形式。流控制指令组744(例如，调用(call)、跳转(jmp))包括0010xxxxb(例如，0x20)形式的指令。混杂指令组746包括指令的混合，包括0011xxxxb(例如，0x30)形式的同步指令(例如，等待(wait)、发送(send))。并行数学指令组748包括0100xxxxb(例如，0x40)形式的逐分量的算术指令(例如，加、乘(mul))。并行数学组748跨数据通道并行地执行算术操作。向量数学组750包括0101xxxxb(例如，0x50)形式的算术指令(例如，dp4)。向量数学组对向量操作数执行算术，诸如点积计算。

图形流水线

[0089] 图8是图形处理器800的另一实施例的框图。图8的具有与本文中的任何其他附图的元件相同的附图标记(或名称)的那些元件能以类似于本文中其他地方描述的任何方式操作或起作用，但不限于此。

[0090] 在一些实施例中，图形处理器800包括图形流水线820、媒体流水线830、显示引擎840、线程执行逻辑850、以及渲染输出流水线870。在一些实施例中，图形处理器800是包括一个或多个通用处理核的多核处理系统内的图形处理器。图形处理器通过至一个或多个控制寄存器(未示出)的寄存器写入、或者经由通过环形互连802发布至图形处理器800的命令被控制。在一些实施例中，环形互连802将图形处理器800耦合至其他处理组件，诸如其他图形处理器或通用处理器。来自环形互连802的命令由命令流转化器803解译，该命令流转化器将指令供应至几何流水线820或媒体流水线830的各个组件。

[0091] 在一些实施例中，命令流转化器803引导顶点取出器805的操作，该顶点取出器805从存储器读取顶点数据，并执行由命令流转化器803提供的顶点处理命令。在一些实施例中，顶点取出器805将顶点数据提供给顶点着色器807，该顶点着色器807对每个顶点执行坐标空间变换和照明操作。在一些实施例中，顶点取出器805和顶点着色器807通过经由线程分派器831将执行线程分派至执行单元852A-852B来执行顶点处理指令。

[0092] 在一些实施例中，执行单元852A-852B是具有用于执行图形操作和媒体操作的指令集的向量处理器的阵列。在一些实施例中，执行单元852A-852B具有专用于每个阵列或在阵列之间被共享的所附接的L1高速缓存851。高速缓存可以被配置为数据高速缓存、指令高

速缓存、或被分区为在不同分区中包含数据和指令的单个高速缓存。

[0093] 在一些实施例中，几何流水线820包括用于执行3D对象的硬件加速曲面细分的曲面细分组件。在一些实施例中，可编程外壳着色器811配置曲面细分操作。可编程域着色器817提供对曲面细分输出的后端评估。曲面细分器813在外壳着色器811的指示下进行操作，并且包括用于基于粗糙的几何模型来生成详细的几何对象集合的专用逻辑，该粗糙的几何模型作为输入被提供该几何流水线820。在一些实施例中，如果不使用曲面细分，则可以绕过曲面细分组件(例如，外壳着色器811、曲面细分器813和域着色器817)。

[0094] 在一些实施例中，完整的几何对象可由几何着色器819经由被分派至执行单元852A-852B的一个或多个线程来处理，或者可以直接行进至裁剪器829。在一些实施例中，几何着色器对整个几何对象而不是对如在图形流水线的先前的级中那样对顶点或顶点补片进行操作。如果禁用曲面细分，则几何着色器819从顶点着色器807接收输入。在一些实施例中，几何着色器819是可由几何着色器程序编程的以便在曲面细分单元被禁用的情况下执行几何曲面细分。

[0095] 在栅格化之前，裁剪器829处理顶点数据。裁剪器829可以是固定功能裁剪器或具有裁剪和几何着色器功能的可编程裁剪器。在一些实施例中，渲染输出流水线870中的栅格化器和深度测试组件873分派像素着色器以将几何对象转换为逐像素表示。在一些实施例中，像素着色器逻辑被包括在线程执行逻辑850中。在一些实施例中，应用可绕过栅格化器和深度测试组件873，并且经由流出单元823访问未栅格化的顶点数据。

[0096] 图形处理器800具有互连总线、互连结构、或允许数据和消息在处理器的主要组件之中传递的某个其他互连机制。在一些实施例中，执行单元852A-852B和相关联的逻辑单元(例如，L1高速缓存851、采样器854、纹理高速缓存858等)经由数据端口856进行互连，以便执行存储器访问并且与处理器的渲染输出流水线组件进行通信。在一些实施例中，采样器854、高速缓存851、858以及执行单元852A-852B各自具有单独的存储器访问路径。在一个实施例中，纹理高速缓存858也可被配置为采样器高速缓存。

[0097] 在一些实施例中，渲染输出流水线870包含栅格化器和深度测试组件873，其将基于顶点的对象转换为相关联的基于像素的表示。在一些实施例中，栅格化器逻辑包括用于执行固定功能三角形和线栅格化的窗口器/掩码器单元。相关联的渲染高速缓存878和深度高速缓存879在一些实施例中也是可用的。像素操作组件877对数据进行基于像素的操作，但是在一些实例中，与2D操作相关联的像素操作(例如，利用混合的位块图像传送)由2D引擎841执行，或者在显示时由显示控制器843使用叠加显示平面来代替。在一些实施例中，共享的L3高速缓存875可用于所有的图形组件，从而允许在无需使用主系统存储器的情况下共享数据。

[0098] 在一些实施例中，图形处理器媒体流水线830包括媒体引擎837和视频前端834。在一些实施例中，视频前端834从命令流转化器803接收流水线命令。在一些实施例中，媒体流水线830包括单独的命令流转化器。在一些实施例中，视频前端834在将媒体命令发送至媒体引擎837之前处理该命令。在一些实施例中，媒体引擎837包括用于生成线程以用于经由线程分派器831分派至线程执行逻辑850的线程生成功能。

[0099] 在一些实施例中，图形处理器800包括显示引擎840。在一些实施例中，显示引擎840在处理器800外部，并且经由环形互连802、或某个其他互连总线或结构来与图形处理器

耦合。在一些实施例中，显示引擎840包括2D引擎841和显示控制器843。在一些实施例中，显示引擎840包含能够独立于3D流水线进行操作的专用逻辑。在一些实施例中，显示控制器843与显示设备(未示出)耦合，该显示设备可以是系统集成显示设备(如在膝上型计算机中)、或者是经由显示设备连接器附接的外部显示设备。

[0100] 在一些实施例中，几何流水线820和媒体流水线830可被配置成用于基于多个图形和媒体编程接口执行操作，并且并非专用于任何一种应用编程接口(API)。在一些实施例中，图形处理器的驱动器软件将专用于特定图形或媒体库的API调用转换成可由图形处理器处理的命令。在一些实施例中，为全部来自Khronos Group的开放图形库(OpenGL)、开放计算语言(OpenCL)和/或Vulkan图形和计算API提供支持。在一些实施例中，也可以为来自微软公司的Direct3D库提供支持。在一些实施例中，可以支持这些库的组合。还可以为开源计算机视觉库(OpenCV)提供支持。如果可进行从未来API的流水线到图形处理器的流水线的映射，则具有兼容3D流水线的未来API也将受到支持。

图形流水线编程

[0101] 图9A是图示根据一些实施例的图形处理器命令格式900的框图。图9B是图示根据实施例的图形处理器命令序列910的框图。图9A中的实线框图示一般被包括在图形命令中的组成部分，而虚线包括任选的或仅被包括在图形命令的子集中的组成部分。图9A的示例性图形处理器命令格式900包括用于标识命令的客户端902、命令操作代码(操作码)904和数据906的数据字段。子操作码905和命令尺寸908也被包括在一些命令中。

[0102] 在一些实施例中，客户端902指定图形设备的处理命令数据的客户端单元。在一些实施例中，图形处理器命令解析器检查每个命令的客户端字段，以调整对命令的进一步处理并将命令数据路由至适当的客户端单元。在一些实施例中，图形处理器客户端单元包括存储器接口单元、渲染单元、2D单元、3D单元、和媒体单元。每个客户端单元具有处理命令的对应的处理流水线。一旦由客户端单元接收到命令，客户端单元就读取操作码904以及子操作码905(如果存在)以确定要执行的操作。客户端单元使用数据字段906内的信息来执行命令。针对一些命令，预期显式的命令尺寸908指定命令的尺寸。在一些实施例中，命令解析器基于命令操作码自动地确定命令中的至少一些命令的尺寸。在一些实施例中，经由双字的倍数来对齐命令。

[0103] 图9B中的流程图示示例性图形处理器命令序列910。在一些实施例中，以图形处理器的实施例为特征的数据处理系统的软件或固件使用所示出的命令序列的某个版本来建立、执行并终止图形操作的集合。仅出于示例性目的示出并描述了样本命令序列，因为实施例不限于这些特定的命令或者该命令序列。而且，命令可以作为批量的命令以命令序列被发布，使得图形处理器将以至少部分同时的方式处理命令序列。

[0104] 在一些实施例中，图形处理器命令序列910可开始于流水线转储清除命令912，以便使得任何活跃的图形流水线完成流水线的当前未决命令。在一些实施例中，3D流水线922和媒体流水线924不并发地操作。执行流水线转储清除以使得活跃的图形流水线完成任何未决命令。响应于流水线转储清除，用于图形处理器的命令解析器将暂停命令处理，直到活跃的绘画引擎完成未决操作并且相关的读高速缓存被无效。任选地，渲染高速缓存中被标记为“脏”的任何数据可以被转储清除到存储器。在一些实施例中，流水线转储清除命令912可以用于流水线同步，或者在将图形处理器置于低功率状态之前使用。

[0105] 在一些实施例中，当命令序列要求图形处理器在流水线之间明确地切换时，使用流水线选择命令913。在一些实施例中，在发布流水线命令之前在执行上下文中仅需要一次流水线选择命令913，除非上下文将发布针对两条流水线的命令。在一些实施例中，紧接在经由流水线选择命令913的流水线切换之前需要流水线转储清除命令912。

[0106] 在一些实施例中，流水线控制命令914配置用于操作的图形流水线，并且用于对3D流水线922和媒体流水线924进行编程。在一些实施例中，流水线控制命令914配置活跃流水线的流水线状态。在一个实施例中，流水线控制命令914用于流水线同步，并且用于在处理批量的命令之前清除来自活跃流水线内的一个或多个高速缓存存储器的数据。

[0107] 在一些实施例中，返回缓冲器状态命令916用于配置用于相应流水线的返回缓冲器的集合以写入数据。一些流水线操作需要分配、选择或配置一个或多个返回缓冲器，在处理期间操作将中间数据写入这一个或多个返回缓冲器中。在一些实施例中，图形处理器还使用一个或多个返回缓冲器来存储输出数据并且执行跨线程通信。在一些实施例中，返回缓冲器状态916包括选择要用于流水线操作的集合的返回缓存器的尺寸和数量。

[0108] 命令序列中的剩余命令基于用于操作的活跃流水线而不同。基于流水线判定920，命令序列被定制用于以3D流水线状态930开始的3D流水线922、或者在媒体流水线状态940处开始的媒体流水线924。

[0109] 用于配置3D流水线状态930的命令包括用于顶点缓冲器状态、顶点元素状态、常量颜色状态、深度缓冲器状态、以及将在处理3D基元命令之前配置的其他状态变量的3D状态设置命令。这些命令的值至少部分地基于使用中的特定3D API来确定。在一些实施例中，如果将不使用某些流水线元件，则3D流水线状态930命令还能够选择性地禁用或绕过那些元件。

[0110] 在一些实施例中，3D基元932命令用于提交待由3D流水线处理的3D基元。经由3D基元932命令传递给图形处理器的命令和相关联的参数被转发到图形流水线中的顶点取出功能。顶点取出功能使用3D基元932命令数据来生成多个顶点数据结构。顶点数据结构被存储在一个或多个返回缓冲器中。在一些实施例中，3D基元932命令用于经由顶点着色器对3D基元执行顶点操作。为了处理顶点着色器，3D流水线922将着色器执行线程分派至图形处理器执行单元。

[0111] 在一些实施例中，经由执行934命令或事件触发3D流水线922。在一些实施例中，寄存器写入触发命令执行。在一些实施例中，经由命令序列中的“去往(go)”或“踢除(kick)”命令来触发执行。在一个实施例中，使用流水线同步命令来触发命令执行，以便通过图形流水线来转储清除命令序列。3D流水线将执行针对3D基元的几何处理。一旦操作完成，就对所得到的几何对象进行栅格化，并且像素引擎对所得到的像素进行着色。对于那些操作，还可以包括用于控制像素着色和像素后端操作的附加命令。

[0112] 在一些实施例中，当执行媒体操作时，图形处理器命令序列910遵循媒体流水线924路径。一般地，针对媒体流水线924进行编程的特定用途和方式取决于待执行的媒体或计算操作。在媒体解码期间，特定的媒体解码操作可以被转移到媒体流水线。在一些实施例中，还可绕过媒体流水线，并且可使用由一个或多个通用处理核提供的资源来整体地或部分地执行媒体解码。在一个实施例中，媒体流水线还包括用于通用图形处理器单元(GPGPU)操作的元件，其中，图形处理器用于使用计算着色器程序来执行SIMD向量操作，这些计算着

色器程序并不明确地与图形基元的渲染相关。

[0113] 在一些实施例中，以与3D流水线922类似的方式配置媒体流水线924。将用于配置媒体流水线状态940的命令集合分派或放置到命令队列中，在媒体对象命令942之前。在一些实施例中，用于媒体流水线状态的命令940包括用于配置媒体流水线元件的数据，这些媒体流水线元件将用于处理媒体对象。这包括用于在媒体流水线内配置视频解码和视频编码逻辑的数据，诸如编码或解码格式。在一些实施例中，用于媒体流水线状态的命令940还支持使用指向包含批量的状态设置的“间接”状态元件的一个或多个指针。

[0114] 在一些实施例中，媒体对象命令942供应指向用于由媒体流水线处理的媒体对象的指针。媒体对象包括存储器缓冲器，该存储器缓冲器包含待处理的视频数据。在一些实施例中，在发布媒体对象命令942之前，所有的媒体流水线状态必须是有效的。一旦流水线状态被配置并且媒体对象命令942被排队，就经由执行命令944或等效的执行事件(例如，寄存器写入)来触发媒体流水线924。随后可通过由3D流水线922或媒体流水线924提供的操作对来自媒体流水线924的输出进行后处理。在一些实施例中，以与媒体操作类似的方式来配置和执行GPGPU操作。

图形软件架构

[0115] 图10图示根据一些实施例的用于数据处理系统1000的示例性图形软件架构。在一些实施例中，软件架构包括3D图形应用1010、操作系统1020、以及至少一个处理器1030。在一些实施例中，处理器1030包括图形处理器1032以及一个或多个通用处理器核1034。图形应用1010和操作系统1020各自在数据处理系统的系统存储器1050中执行。

[0116] 在一些实施例中，3D图形应用1010包含一个或多个着色器程序，这一个或多个着色器程序包括着色器指令1012。着色器语言指令可以采用高级着色器语言，诸如高级着色器语言(HLSL)或OpenGL着色器语言(GLSL)。应用还包括采用适于由通用处理器核1034执行的机器语言的可执行指令1014。应用还包括由顶点数据限定的图形对象1016。

[0117] 在一些实施例中，操作系统1020是来自微软公司的Microsoft® Windows®操作系统、专属的类UNIX操作系统、或使用Linux内核的变体的开源的类UNIX操作系统。操作系统1020可支持图形API 1022，诸如Direct3D API、OpenGL API或Vulkan API。当Direct3D API正在使用时，操作系统1020使用前端着色器编译器1024以将采用HLSL的任何着色器指令1012编译成较低级的着色器语言。编译可以是即时(JIT)编译，或者应用可执行着色器预编译。在一些实施例中，在3D图形应用1010的编译期间，将高级着色器编译成低级着色器。在一些实施例中，着色器指令1012以中间形式提供，诸如由Vulkan API使用的标准便携式中间表示(SPIR)的某个版本。

[0118] 在一些实施例中，用户模式图形驱动器1026包含后端着色器编译器1027，该后端着色器编译器1027用于将着色器指令1012转换成硬件专用表示。当OpenGL API在使用中时，将采用GLSL高级语言的着色器指令1012传递至用户模式图形驱动器1026以用于编译。在一些实施例中，用户模式图形驱动器1026使用操作系统内核模式功能1028来与内核模式图形驱动器1029进行通信。在一些实施例中，内核模式图形驱动器1029与图形处理器1032通信以分派命令和指令。

IP核实施方式

[0119] 至少一个实施例的一个或多个方面可以由存储在机器可读介质上的代表性代码

实现，该机器可读介质表示和/或限定集成电路(诸如，处理器)内的逻辑。例如，机器可读介质可以包括表示处理器内的各种逻辑的指令。当由机器读取时，指令可以使机器制造用于执行本文所述的技术的逻辑。这类表示(被称为“IP核”)是集成电路的逻辑的可重复使用单元，这些可重复使用单元可以作为描述集成电路的结构的硬件模型而被存储在有形的、机器可读介质上。可以将硬件模型供应至在制造集成电路的制造机器上加载硬件模型的各消费者或制造设施。可以制造集成电路，使得电路执行与本文中描述的实施例中的任一实施例相关联地描述的操作。

[0120] 图11A是图示根据实施例的IP核开发系统1100的框图，该IP核开发系统1100可以用于制造集成电路以执行操作。IP核开发系统1100可以用于生成可并入到更大的设计中或用于构建整个集成电路(例如，SOC集成电路)的模块化、可重复使用设计。设计设施1130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1110。软件仿真1110可用于使用仿真模型1112来设计、测试并验证IP核的行为。仿真模型1112可以包括功能仿真、行为仿真和/或时序仿真。随后可从仿真模型1112创建或合成寄存器传输级(RTL)设计1115。RTL设计1115是对硬件寄存器之间的数字信号的流进行建模的集成电路(包括使用建模的数字信号执行的相关联的逻辑)的行为的抽象。除了RTL设计1115之外，还可以创建、设计或合成逻辑级或晶体管级的较低级别设计。由此，初始设计和仿真的特定细节可有所不同。

[0121] 可以由设计设施进一步将RTL设计1115或等效方案合成到硬件模型1120中，该硬件模型1120可以采用硬件描述语言(HDL)或物理设计数据的某种其他表示。可以进一步仿真或测试HDL以验证IP核设计。可使用非易失性存储器1140(例如，硬盘、闪存、或任何非易失性存储介质)来存储IP核设计以用于递送至第三方制造设施1165。替代地，可以通过有线连接1150或无线连接1160(例如，经由因特网)来传输IP核设计。制造设施1165随后可以制造至少部分地基于IP核设计的集成电路。所制造的集成电路可被配置用于执行根据本文中描述的至少一个实施例的操作。

[0122] 图11B图示根据本文中描述的一些实施例的集成电路封装组件1170的截面侧视图。集成电路封装组件1170图示如本文中所描述的一个或多个处理器或加速器设备的实现方式。封装组件1170包括连接至衬底1180的多个硬件逻辑单元1172、1174。逻辑1172、1174可以至少部分地实现在可配置逻辑或固定功能逻辑硬件中，并且可包括本文中描述的(多个)处理器核、(多个)图形处理器或其他加速器设备中的任何处理器核、图形处理器或其他加速器设备的一个或多个部分。每个逻辑单元1172、1174可以实现在半导体管芯内，并且经由互连结构1173与衬底1180耦合。互连结构1173可以被配置成在逻辑1172、1174与衬底1180之间路由电信号，并且可以包括互连，该互连诸如但不限于凸块或支柱。在一些实施例中，互连结构1173可以被配置成路由电信号，诸如例如，与逻辑1172、1174的操作相关联的输入/输出(I/O)信号和/或功率或接地信号。在一些实施例中，衬底1180是基于环氧树脂的层压衬底。在其他实施例中，封装组件1170可以包括其他合适类型的衬底。封装组件1170可以经由封装互连1183连接至其他电气设备。封装互连1183可以耦合至衬底1180的表面以将电信号路由到其他电气设备，诸如主板、其他芯片组或多芯片模块。

[0123] 在一些实施例中，逻辑单元1172、1174与桥接器1182电耦合，该桥接器1182被配置成在逻辑1172与逻辑1174之间路由电信号。桥接器1182可以是为电信号提供路由的密集互连结构。桥接器1182可以包括由玻璃或合适的半导体材料构成的桥接器衬底。电路由特征

可形成在桥接器衬底上以提供逻辑1172与逻辑1174之间的芯片到芯片连接。

[0124] 尽管图示了两个逻辑单元1172、1174和桥接器1182，但是本文中所描述的实施例可以包括在一个或多个管芯上的更多或更少的逻辑单元。这一个或多个管芯可以由零个或更多个桥接器连接，因为当逻辑被包括在单个管芯上时，可以排除桥接器1182。替代地，多个管芯或逻辑单元可以由一个或多个桥接器连接。另外，在其他可能的配置(包括三维配置)中，多个逻辑单元、管芯和桥接器可被连接在一起。

示例性芯片上系统集成电路

[0125] 图12-图14图示根据本文中所述的各实施例的可以使用一个或多个IP核制造的示例性集成电路和相关联的图形处理器。除了所图示的内容之外，还可以包括其他逻辑和电路，包括附加的图形处理器/核、外围接口控制器或通用处理器核。

[0126] 图12是图示根据实施例的可以使用一个或多个IP核制造的示例性芯片上系统集成电路1200的框图。示例性集成电路1200包括一个或多个应用处理器1205(例如，CPU)、至少一个图形处理器1210，并且可附加地包括图像处理器1215和/或视频处理器1220，其中的任一个都可以是来自相同设计设施或多个不同设计设施的模块化IP核。集成电路1200包括外围或总线逻辑，包括USB控制器1225、UART控制器1230、SPI/SDIO控制器1235和I2S/I2C控制器1240。另外，集成电路可包括显示设备1245，该显示设备1245耦合至高清晰度多媒体接口(HDMI)控制器1250和移动行业处理器接口(MIPI)显示接口1255中的一个或多个。可以由闪存子系统1260(包括闪存和闪存控制器)来提供存储。可以经由存储器控制器1265来提供存储器接口以获得对SDRAM或SRAM存储器设备的访问。一些集成电路附加地包括嵌入式安全引擎1270。

[0127] 图13A-图13B是图示根据本文中所描述的实施例的用于在SoC内使用的示例性图形处理器的框图。图13A图示根据实施例的可以使用一个或多个IP核制造的芯片上系统集成电路的示例性图形处理器1310。图13B图示根据实施例的可以使用一个或多个IP核制造的芯片上系统集成电路的附加示例性图形处理器1340。图13A的图形处理器1310是低功率图形处理器核的示例。图13B的图形处理器1340是较高性能图形处理器核的示例。图形处理器1310、1340中的每一个都可以是图12的图形处理器1210的变体。

[0128] 如图13A中所示，图形处理器1310包括顶点处理器1305以及一个或多个片段处理器1315A-1315N(例如，1315A、1315B、1315C、1315D，一直到1315N-1和1315N)。图形处理器1310可以经由单独的逻辑执行不同的着色器程序，使得顶点处理器1305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1315A-1315N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1305执行3D图形流水线的顶点处理级，并生成基元数据和顶点数据。(多个)片段处理器1315A-1315N使用由顶点处理器1305生成的基元数据和顶点数据来产生被显示在显示设备上的帧缓冲器。在一个实施例中，(多个)片段处理器1315A-1315N被优化以执行如在OpenGL API中提供的片段着色器程序，这些片段着色器程序可以用于执行与如在Direct 3D API中提供的像素着色器程序类似的操作。

[0129] 图形处理器1310附加地包括一个或多个存储器管理单元(MMU)1320A-1320B、(多个)高速缓存1325A-1325B以及(多个)电路互连1330A-1330B。这一个或多个MMU 1320A-1320B为图形处理器1310(包括为顶点处理器1305和/或(多个)片段处理器1315A-1315N)提供虚拟到物理地址映射，除了存储在一个或多个高速缓存1325A-1325B中的顶点数据或图

像/纹理数据之外，该虚拟到物理地址映射还可以引用存储在存储器中的顶点数据或图像/纹理数据。在一个实施例中，一个或多个MMU 1320A-1320B可以与系统内的其他MMU同步，使得每个处理器1205-1220可以参与共享或统一的虚拟存储器系统，系统内的其他MMU包括与图12的一个或多个应用处理器1205、图像处理器1215和/或视频处理器1220相关联的一个或多个MMU。根据实施例，一个或多个电路互连1330A-1330B使得图形处理器1310能够经由SoC的内部总线或经由直接连接来与SoC内的其他IP核对接。

[0130] 如图13B中所示，图形处理器1340包括图13A的图形处理器1310的一个或多个MMU 1320A-1320B、高速缓存1325A-1325B、以及电路互连1330A-1330B。图形处理器1340包括一个或多个着色器核1355A-1355N(例如，1355A、1355B、1355C、1355D、1355E、1355F，一直到1355N-1和1355N)，这一个或多个着色器核提供统一着色器核架构，在该统一着色器核架构中，单个核或类型或核可以执行所有类型的可编程着色器代码，包括用于实现顶点着色器、片段着色器和/或计算着色器的着色器程序代码。存在的着色器核的确切数量可以因实施例和实现方式而异。另外，图形处理器1340包括核间任务管理器1345，该核间任务管理器1345充当用于将执行线程分派给一个或多个着色器核1355A-1355N的线程分派器和用于加速对基于片的渲染的分片操作的分片单元1358，在基于片的渲染中，针对场景的渲染操作在图像空间中被细分，例如以利用场景内的局部空间一致性或优化内部高速缓存的使用。

[0131] 图14A-图14B图示根据本文中所描述的实施例的附加示例性图形处理器逻辑。图14A图示图形核1400，该图形核1400可被包括在图12的图形处理器1210内，并且可以是如图13B中的统一着色器核1355A-1355N。图14B图示适合部署在多芯片模块上的高度并行的通用图形处理单元1430。

[0132] 如在图14A中所示，图形核1400包括对于图形核1400内的执行资源而言共同的共享指令高速缓存1402、纹理单元1418和高速缓存存储器/共享存储器1420。图形核1400可包括用于每个核的多个切片1401A-1401N或分区，并且图形处理器可包括图形核1400的多个实例。切片1401A-1401N可以包括支持逻辑，该支持逻辑包括本地指令高速缓存1404A-1404N、线程调度器1406A-1406N、线程分派器1408A-1408N、以及寄存器的集合1410A-1410N。为了执行逻辑操作，切片1401A-1401N可包括一组附加的功能单元(AFU1412A-1412N)、浮点单元(FPU 1414A-1414N)、整数算术逻辑单元(ALU1416-1416N)、地址计算单元(ACU 1413A-1413N)、双精度浮点单元(DPFPU1415A-1415N)、以及矩阵处理单元(MPU 1417A-1417N)。

[0133] 这些计算单元中的一些以特定精度进行操作。例如，FPU1414A-1414N可执行单精度(32位)和半精度(16位)浮点操作，而DPFPU1415A-1415N执行双精度(64位)浮点操作。ALU 1416A-1416N能以8位精度、16位精度和32位精度执行可变精度整数操作，并且可以被配置用于混合精度操作。MPU 1417A-1417N还可以被配置用于混合精度矩阵操作，包括半精度浮点操作和8位整数操作。MPU 1417A-1417N可执行各种各样的矩阵操作以加速机器学习应用框架，包括启用对加速的通用矩阵到矩阵乘法(GEMM)的支持。AFU 1412A-1412N可执行不被浮点单元或整数单元支持的附加逻辑操作，包括三角函数操作(例如，正弦、余弦等)。

[0134] 如图14B中所示，通用处理单元(GPGPU)1430可以被配置成使得能够由图形处理单元的阵列执行高度并行的计算操作。另外，GPGPU 1430可以直接链接到GPGPU的其他实例以创建多GPU集群，从而改善尤其是深度神经网络的训练速度。GPGPU 1430包括用于启用与主

机处理器的连接的主机接口1432。在一个实施例中，主机接口1432是PCI Express接口。然而，主机接口还可以是供应方专用的通信接口或通信结构。GPGPU 1430从主机处理器接收命令，并且使用全局调度器1434将与那些命令相关联的执行线程分发给计算集群的集合1436A-1436H。计算集群1436A-1436H共享高速缓存存储器1438。高速缓存存储器1438可以充当用于计算集群1436A-1436H内的高速缓存存储器的较高级别的高速缓存。

[0135] GPGPU 1430包括经由存储器控制器的集合1442A-1442B与计算集群1436A-1436H耦合的存储器1434A-1434B。在各实施例中，存储器1434A-1434B可包括各种类型的存储器设备，包括动态随机存取存储器(DRAM)或图形随机存取存储器，诸如，同步图形随机存取存储器(SGRAM)，包括图形双倍数据速率(GDDR)存储器。

[0136] 在一个实施例中，计算集群1436A-1436H各自包括图形核的集合，图形核诸如图14A的图形核1400，这些图形核可包括多种类型的整数逻辑单元和浮点逻辑单元，所述多种类型的整数逻辑单元和浮点逻辑单元可以在一定精度范围内执行包括适于机器学习计算的计算操作。例如并且在一个实施例中，计算集群1436A-1436H中的每一个计算集群中的浮点单元的至少一个子集可以被配置成执行16位或32位浮点操作，而浮点单元的不同子集可以被配置成执行64位浮点操作。

[0137] GPGPU 1430的多个实例可以被配置成作为计算集群进行操作。由计算集群用于同步和数据交换的通信机制跨实施例而有所不同。在一个实施例中，GPGPU 1430的多个实例通过主机接口1432进行通信。在一个实施例中，GPGPU 1430包括I/O中枢1439，该I/O中枢1439将GPGPU 1430与GPU链路1440耦合，该GPU链路1440启用至GPGPU的其他实例的直接连接。在一个实施例中，GPU链路1440耦合至专用GPU-GPU桥接器，该GPU-GPU桥接器实现GPGPU 1430的多个实例之间的通信和同步。在一个实施例中，GPU链路1440与高速互连耦合，以将数据传输和接收至其他GPGPU或并行处理器。在一个实施例中，GPGPU 1430的多个实例位于单独的数据处理系统中，并且经由网络设备进行通信，该网络设备可经由主机接口1432来访问。在一个实施例中，附加于或替代于主机接口1432，GPU链路1440可以被配置成启用至主机处理器的连接。

[0138] 尽管GPGPU 1430的所图示配置可以被配置成训练神经网络，但是一个实施例提供GPGPU 1430的替代配置，该替代配置可以被配置成用于在高性能或低功率推断平台内的部署。在推断配置中，相对于训练配置，GPGPU1430包括计算集群1436A-1436H中的更少的计算集群。另外，与存储器1434A-1434B相关联的存储器技术在推断配置与训练配置之间可以不同，其中，更高带宽的存储器技术专用于训练配置。在一个实施例中，GPGPU 1430的推断配置可以支持推断专用指令。例如，推断配置可提供对一个或多个8位整数点积指令的支持，这一个或多个8位整数点积指令通常在用于经部署的神经网络的推断操作期间使用。

[0139] 图15图示计算设备1500的一个实施例。计算设备1500(例如，智能可穿戴设备、虚拟现实(VR)设备、头戴式显示器(HMD)、移动计算机、物联网(IoT)设备、膝上型计算机、台式计算机、服务器计算机等)可与图1的处理系统100相同，并且相应地，为了简明、清楚和易于理解，以上参考图1-图14所阐述的细节中的许多细节此后不作进一步讨论或重复。

[0140] 计算设备1500可以包括任何数量和类型的通信设备，诸如，大型计算系统，诸如服务器计算机、台式计算机等，并且可进一步包括机顶盒(例如，基于互联网的有线电视机顶盒等)、基于全球定位系统(GPS)的设备等。计算设备1500可以包括用作通信设备的移动计

算设备，诸如包括智能手机的蜂窝电话、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、可穿戴设备(例如，眼镜、手表、手环、智能卡、珠宝、衣物等)、媒体播放器等。例如，在一个实施例中，计算设备1500可以包括采用计算机平台的移动计算设备，该计算机平台主控将计算设备1500的各种硬件和/或软件组件集成在单个芯片上的诸如芯片上系统(“SoC”或“SOC”)之类的集成电路(“IC”)。

[0141] 如所示，在一个实施例中，计算设备1500可以包括任何数量和类型的硬件和/或软件组件，诸如(但不限于)GPU 1514、图形驱动器(也称为“GPU驱动器”、“图形驱动器逻辑”、“驱动器逻辑”、用户模式驱动器(UMD)、UMD、用户模式驱动器框架(UMDF)、UMDF或简称为“驱动器”)1516、CPU 1512、存储器1508、网络设备、驱动器等，以及诸如触摸屏、触摸面板、触摸板、虚拟或常规键盘、虚拟或常规鼠标、端口、连接器等之类的输入/输出(I/O)源1504。

[0142] 计算设备1500可以包括用作计算机设备1500的硬件和/或物理资源与用户之间的接口的操作系统(OS)1506。构想CPU 1512可以包括一个或多个处理器，而GPU 1514可以包括一个或多个图形处理器。

[0143] 应当注意的是，贯穿本文档，可互换地使用如“节点”、“计算节点”、“服务器”、“服务器设备”、“云计算机”、“云服务器”、“云服务器计算机”、“机器”、“主机”、“设备”、“计算设备”、“计算机”、“计算系统”等术语。应当进一步注意，贯穿本文档，可以互换地使用如“应用”、“软件应用”、“程序”、“软件程序”、“包”、“软件包”等术语。并且，贯穿本文档，可以互换地使用如“作业”、“输入”、“请求”、“消息”等术语。

[0144] 可以构想，并且如参考图1-14进一步描述的，如上所述的图形流水线的一些进程以软件实现，而其余的进程以硬件实现。图形流水线可以以图形协处理器设计来实现，其中，CPU 1512被设计为用于与GPU1514一起工作，GPU 1514可以被包括在CPU 1512中或与其共同定位。在一个实施例中，GPU 1514可采用用于执行与图形渲染相关的常规功能的任何数量与类型的常规软件与硬件逻辑，以及用于执行任何数量与类型的指令的新型软件与硬件逻辑。

[0145] 如上所述，存储器1508可以包括随机存取存储器(RAM)，该RAM包括具有对象信息的应用数据库。存储器控制器中枢可以访问RAM中的数据并且将其转发至GPU 1514以供图形流水线处理。RAM可以包括双数据速率RAM(DDR RAM)、扩展数据输出RAM(EDO RAM)等。CPU 1512与硬件图形流水线交互以共享图形流水线功能。

[0146] 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1508中。所得的图像随后被传递到I/O源1504，诸如，用于显示图像的显示组件。可以构想，显示设备可以是用于向用户显示信息的各种类型的显示设备，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。

[0147] 存储器1508可以包括缓冲器(例如，帧缓冲器)的预分配区域；然而，本领域普通技术人员应当理解，各实施例不限于此，并且可以使用较低级图形流水线可访问的任何存储器。计算设备1500可进一步包括如图1所引用的平台控制器中枢(PCH)130、一个或多个I/O源1504等。

[0148] CPU 1512可以包括用于执行指令的一个或多个处理器，以便执行计算系统实现的任何软件例程。指令经常涉及对数据执行的某种操作。数据和指令两者都可以存储在系统存储器1508和任何相关联的高速缓存中。高速缓存通常被设计成具有比系统存储器1508短

的等待时间;例如,高速缓存可以被集成到与(多个)处理器相同的(多个)硅芯片上和/或用较快的静态RAM(SRAM)单元进行构造,而系统存储器1508可以用较慢的动态RAM(DRAM)单元进行构造。与系统存储器1508相对照,通过倾向于将更频繁使用的指令和数据存储在高速缓存中,改善了计算设备1500的整体性能效率。可以构想,在一些实施例中,GPU 1514可以作为CPU 1512的一部分(诸如物理CPU封装的一部分)存在,在这种情况下,存储器1508可以由CPU 1512和GPU 1514共享或保持分开。

[0149] 系统存储器1508可以对计算设备1500内的其他组件可用。例如,在软件程序的实现中,从至计算设备1500的各种接口(例如键盘和鼠标、打印机端口、局域网(LAN)端口、调制解调器端口等)接收到的或从计算机设备1500的内部存储元件(例如,硬盘驱动器)检取到的任何数据(例如,输入图形数据)通常在由一个或多个处理器操作之前临时排队进入系统存储器1508。类似地,软件程序确定应通过计算系统接口中的一个从计算设备1500发送到外部实体或存储到内部存储元件中的数据在其被传输或存储之前经常在系统存储器1508中临时排队。

[0150] 进一步地,例如,PCH可用于确保此类数据在系统存储器1508与其适当的对应计算系统接口(以及内部存储设备,如果计算系统是如此设计的话)之间正确地传递并可在其自身与所示I/O源/设备1504之间具有双向点对点链路。类似地,MCH可用于管理CPU 1512与GPU 1514、接口与内部存储元件之间对于系统存储器1508访问的多种竞争请求,这些请求可能在时间上彼此紧接地出现。

[0151] I/O源1504可以包括一个或多个I/O设备(例如,网络适配器),一个或多个I/O设备实现为用于向计算设备1500传送数据和/或传送来自计算设备1500的数据;或者用于计算设备1500内的大规模非易失性存储(例如,硬盘驱动器)。包括字母数字及其他键的用户输入设备可用于将信息和命令选择传递至GPU 1514。另一类型的用户输入设备是用于将方向信息和命令选择传递至GPU 1514并控制显示设备上的光标移动的光标控制,诸如鼠标、轨迹球、触摸屏、触摸板或光标方向键。可以采用计算机设备1500的相机和麦克风阵列来观察手势、记录音频和视频并接收和发射视觉命令和音频命令。

[0152] 计算设备1500可进一步包括(多个)网络接口以提供对网络的访问,网络诸如,LAN、广域网(WAN)、城域网(MAN)、个域网(PAN)、蓝牙、云网络、移动网络(例如,第3代(3G)、第4代(4G)等)、内联网、互联网等。(多个)网络接口可以包括例如具有天线的无线网络接口,天线可以表示一个或多个天线。(多个)网络接口还可包括例如经由网络电缆与远程设备通信的有线网络接口,网络电缆可以是例如以太网电缆、同轴电缆、光纤电缆、串行电缆或并行电缆。

[0153] (多个)网络接口可例如通过符合IEEE 802.11b和/或IEEE802.11g标准来提供对LAN的访问,并且/或者无线网络接口可以例如通过符合蓝牙标准来提供对个域网的访问。还可支持其他无线网络接口和/或协议,包括,先前的以及后续的版本的标准。除了经由无线LAN标准的通信或代替经由无线LAN标准的通信,(多个)网络接口可使用例如以下协议来提供无线通信: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/或任何其他类型的无线通信协议。

[0154] (多个)网络接口可以包括一个或多个通信接口,诸如,调制解调器、网络接口卡或其他众所周知的接口设备,诸如,用于为了提供通信链路以支持例如LAN或WAN而耦合至以

太网、令牌环或其他类型的物理有线或无线附连的那些通信接口。以此方式，计算机系统还可以经由常规的网络基础设施(例如，包括内联网或互联网)耦合至多个外围设备、客户端、控制表面、控制台或服务器。

[0155] 应当理解，对于某些实现方式，比在上文中所描述的示例更少或更多地配备的系统可以是优选的。因此，取决于众多因素，诸如价格约束、性能要求、技术改进或其他情况，计算设备1500的配置可以随着实现方式而改变。电子设备或计算机系统1500的示例可以包括(但不限于)：移动设备、个人数字助理、移动计算设备、智能电话、蜂窝电话、手持设备、单向寻呼机、双向寻呼机、消息收发设备、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、手持式计算机、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、网络装置、web装置、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、用户站、移动用户中心、无线电网络控制器、路由器、集线器、网关、桥接器、交换机、机器或上述各项的组合。

[0156] 实施例可以被实现为以下各项中的任何一项或其组合：使用主板互连的一个或多个微芯片或集成电路、硬连线逻辑、由存储器设备存储且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)。作为示例，术语“逻辑”可以包括软件或硬件和/或软件和硬件的组合。

[0157] 实施例可以被提供为例如计算机程序产品，该计算机程序产品可包括一种或多种机器可读介质，这一种或多种机器可读介质具有存储于其上的机器可执行指令，这些机器可执行指令在由一个或多个机器(诸如，计算机、计算机网络或其他电子设备)执行时可导致这一个或多个机器执行根据在本文中所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(紧凑盘只读存储器)以及磁光盘、ROM、RAM、EPROM(可擦除可编程只读存储器)、EEPROM(电可擦除可编程只读存储器)、磁卡或光卡、闪存、或者适合于存储机器可执行指令的其他类型的介质/机器可读介质。

[0158] 此外，实施例可作为计算机程序产品被下载，其中，经由通信链路(例如，调制解调器和/或网络连接)，借助于在载波或其他传播介质中具体化和/或由载波或其他传播介质调制的一个或多个数据信号，可将程序从远程计算机(例如，服务器)传输至请求计算机(例如，客户端)。

[0159] 图16图示GPU 1514的一个实施例。如图16所示，GPU 1514包括执行单元1610，执行单元1610具有经由结构架构耦合的多个节点(例如，节点0-节点7)。在一个实施例中，每个节点包括经由结构元件1605耦合至存储器1650的多个处理元件。在此类实施例中，每个结构元件1605耦合至两个节点和存储器1650中的两个区块。因此，结构元件1605A将节点0和节点1耦合至区块0和区块1，结构元件1605B将节点2和节点3耦合至区块2和区块3，结构元件1605C将节点4和节点5耦合至区块4和区块5，并且结构元件1605D将节点6和节点7耦合至区块6和区块7。

[0160] 根据一个实施例，每个结构元件1605包括MMU 1620、控制高速缓存1630和仲裁器1640。MMU 1620执行存储器管理以管理存储器区块0-7之间的虚拟地址空间。在一个实施例中，每个MMU 1620管理数据到存储器1650中的相关联的存储器区块的传输以及数据从存储器1650中的相关联的存储器区块的传输。仲裁器1640在每个相关联的节点之间针对对存储

器1650的访问进行仲裁。例如，仲裁器1640A在处理节点0和1之间针对对区块0和1的访问进行仲裁。

[0161] 控制高速缓存(CC)1630执行对存储器数据的压缩/解压缩。例如，CC 1630对接收自处理节点的要写入存储器1650的数据(例如，主表面数据)进行压缩，并且在传输至处理节点之前对读取自存储器1650的数据进行解压缩。根据一个实施例，存储在存储器1650中的每个地址处的经压缩的数据包括相关联的元数据，相关联的元数据指示数据的压缩状态(例如，主表面数据将如何被压缩/解压缩)。在此类实施例中，MMU 1620基于主表面数据的物理地址直接计算元数据存储器位置。

[0162] 在进一步的实施例中，基于存储器的尺寸拆分存储器的部分。例如，在1字节的元数据表示256字节的主表面数据的压缩方案中，拆分存储器的1/256用于元数据。因此，具有8GB的本地存储器的实施例实现在存储器1650中分配32MB的元数据空间。在又一实施例中，考虑到散列含义，MMU 1620基于物理地址计算元数据地址。结果，最终内容被转发至CC 1630。

[0163] 当产生元数据的散列变得破碎(例如，贯穿存储器空间散布)时出现问题。图17图示每节点间隔为1KB的存储器空间的一个实施例。如图17所示，每1KB的数据(例如，主表面数据)与4B的元数据相关联。在一个实施例中，通过1KB的粒度对数据进行散列。因此散列导致数据不会被连续地存储在线性空间中。例如，第一个8KB的数据中的每个KB在区块0-7之间散布。因此，第一个、第九个、第十六个等主表面数据地址驻留在同一区块中(如图17中的框的顶行所示)。然而，对于相关联的4B的元数据(在框的底行中示出)，1KB的散列粒度导致元数据跨8个区块散布。因此，元数据通常可以在与其相关联的数据到存储器的路径的不同路径(例如，不同区块)中结束。

[0164] 根据一个实施例，在已经对主数据和元数据执行散列函数之后，MMU 1620调节(或重新打包)元数据地址位置。在此类实施例中，调节导致元数据的对齐，使得每4B的元数据存储在与其相关联的1KB的数据所存储的相同的区块中。在进一步的实施例中，经散列的元数据的地址位置被调节，使得每4B的相关联的元数据被组合为区块中的32B元数据块的地址(如图18所示)。例如，区块0可以存储在第一、第九、第十六等地址处的主表面数据并且存储32B元数据块的相关联的元数据。类似地，区块1存储在第二、第十、第十七等地址处的主表面数据并且存储具有相关联的元数据的第二32B元数据块。

[0165] 图19图示MMU 1620的一个实施例。MMU 1620包括散列引擎1920，散列引擎1920实现散列表1921以执行散列函数来计算到存储器1650中的物理地址位置中的索引。在一个实施例中，散列引擎1920根据原始节点散列函数执行散列。在此类实施例中，散列函数考虑地址的某些位并且基于这些位导出特定的区块地址。散列函数使存储器访问跨所有物理存储器区块散布，从而实现对存储器区块的并行访问。与所有访问去往单个存储器区块并且导致该存储器区块处的串行化相比，使存储器访问跨独立的存储器区块并行化有助于提高存储器性能。作为示例，利用跨存储器区块的1KB的散列粒度，表1图示可以使用的散列的一个实施例。

表1

	a[15]	a[14]	a[13]	a[12]	a[11]
原始_节点[0]	x		x		x
原始_节点[1]	x	x		x	
原始_节点[2]		x	x		

节点ID然后可以被认为是{原始_节点[0],原始_节点[2],原始_节点[1]}

对于该特定实施例(例如,1KB粒度的散列并且使用以上散列函数),存储器访问的基本位变为{a[12:10]}。

[0166] 散列引擎1920还包括映射逻辑1922和打包逻辑1924以促进导致所有元数据(例如,32B)被放置在与相关联的主数据相同的区块中的散列。映射逻辑1922执行线性映射(例如,256-1映射)以将主数据映射到元数据。打包逻辑1924执行移位操作以将元数据组合到区块中的元数据块中。继续以跨存储器区块进行散列的1KB粒度作为示例,元数据地址[39:0]然后被形成为(假设40位的地址,用于总体上访问存储器):{8'b0,主_表面_地址[39:21],基本_位[2:0],主_表面_地址[20:17],主_表面_地址[13:9,7]},其中基本_位是来自散列函数的a[12:10]。元数据高速缓存行内的字节偏移由元数据_地址[5:0]=主_表面_地址[13:9,7]给出。

[0167] 图20是图示用于执行压缩散列的过程的一个实施例的流程图。在处理框2010处,接收主数据地址。在处理框2020处,执行线性映射以将主数据地址映射到相关联的元数据地址。在处理框2030处,对主数据地址和元数据地址执行函数。在处理框2040处,执行移位操作以将元数据打包在区块内。在一个实施例中,移位操作取决于所实现的散列函数的类型。

[0168] 以下条款和/或示例涉及进一步的实施例或示例。可在一个或多个实施例中的任何地方使用示例中的细节。能以各种方式将不同的实施例或示例的各种特征与所包括的一些特征以及被排除的其他特征组合以适应各种不同的应用。示例可以包括主题,诸如:方法;用于执行所述方法的动作的装置;至少一种包括指令的机器可读介质,所述指令当由机器执行时使所述机器执行所述方法的动作;或用于根据本文中所描述的实施例和示例促进混合通信的设备或系统。

[0169] 一些实施例涉及示例1,示例1包括一种用于促进存储器数据压缩的装置,包括:存储器,具有多个区块,用于存储主数据和与主数据相关联的元数据;以及存储器管理单元(MMU),耦合至多个区块,用于执行散列函数来为主数据和元数据计算到存储器中的虚拟地址位置中的索引,并且调节元数据虚拟地址位置以将每个经调节的元数据虚拟地址位置存储在存储相关联的主数据的区块中。

[0170] 示例2包括示例1的主题,其中MMU调节元数据的地址位置包括:将要存储在区块中的元数据组合以生成元数据块。

[0171] 示例3包括示例1和2的主题，其中MMU调节元数据的地址位置进一步包括：执行一个或多个移位操作。

[0172] 示例4包括示例1-3的主题，其中MMU包括多个MMU，每个MMU耦合至多个区块中的一个或多个。

[0173] 示例5包括示例1-4的主题，其中多个MMU中的每一个包括实现为执行散列函数的散列表。

[0174] 示例6包括示例1-5的主题，其中多个MMU中的每一个进一步执行线性映射以将主数据地址映射到元数据地址。

[0175] 示例7包括示例1-6的主题，进一步包括：第一MMU，耦合至第一区块，第一区块用于存储第一集合的主数据和与第一集合的主数据相关联的第一元数据块的元数据；以及第二MMU，耦合至第二区块，第二区块用于存储第二集合的主数据和与第二集合的主数据相关联的第二元数据块的元数据。

[0176] 一些实施例涉及示例8，示例8包括一种用于促进存储器数据压缩的方法，包括：执行散列函数来为主数据和与主数据相关联的元数据计算到存储器中的虚拟地址位置中的索引，调节元数据虚拟地址位置并且将元数据存储在经调节的元数据虚拟地址位置处，其中每个经调节的元数据虚拟地址位置位于存储相关联的主数据的区块中。

[0177] 示例9包括示例8的主题，进一步包括：接收主数据地址；以及将主数据地址映射到元数据地址。

[0178] 示例10包括示例8和9的主题，进一步包括：调节元数据的地址位置包括执行一个或多个移位操作。

[0179] 示例11包括示例8-10的主题，进一步包括：将要存储在区块中的元数据组合以生成元数据块。

[0180] 示例12包括示例8-11的主题，进一步包括：将第一集合的主数据和与第一集合的主数据相关联的第一元数据块的元数据存储在第一区块处；以及将第二集合的主数据和与第二集合的主数据相关联的第二元数据块的元数据存储在第二区块处。

[0181] 一些实施例涉及示例13，示例13包括一种图形处理单元(GPU)，包括：存储器，具有多个区块，用于存储主数据和与主数据相关联的元数据；以及多个结构元件，耦合至多个区块，每个结构元件包括存储器管理单元(MMU)，MMU耦合至多个区块中的一个或多个，MMU用于执行散列函数来为主数据和元数据计算到存储器中的虚拟地址位置中的索引，并且调节元数据虚拟地址位置以将每个经调节的元数据虚拟地址位置存储在存储相关联的主数据的区块中。

[0182] 示例14包括示例13的主题，其中MMU调节元数据的地址位置包括：将要存储在区块中的元数据组合以生成元数据块。

[0183] 示例15包括示例13和14的主题，其中MMU调节元数据的地址位置进一步包括：执行一个或多个移位操作。

[0184] 示例16包括示例13-15的主题，其中MMU包括实现为执行散列函数的散列表。

[0185] 示例17包括示例13-16的主题，其中MMU进一步执行线性映射以将主数据地址映射到元数据地址。

[0186] 示例18包括示例13-17的主题，进一步包括：第一结构元件，具有耦合至第一区块

的第一MMU,第一区块用于存储第一集合的主数据和与第一集合的主数据相关联的第一元数据块的元数据;以及第二结构元件,具有耦合至第二区块的第二MMU,第二区块用于存储第二集合的主数据和与第二集合的主数据相关联的第二元数据块的元数据。

[0187] 示例19包括示例13-18的主题,进一步包括:第一集合的一个或多个处理节点,耦合至第一结构元件;以及第二集合的一个或多个处理节点,耦合至第二结构元件。

[0188] 示例20包括示例13-19的主题,其中第一结构元件包括耦合在第一集合的一个或多个处理节点与第一MMU之间的第一控制高速缓存,第一控制高速缓存用于执行数据压缩和解压缩,并且第二结构元件包括耦合在第二集合的一个或多个处理节点与第二MMU之间的第二控制高速缓存,第二控制高速缓存用于执行数据压缩和解压缩。

[0189] 上文已经参考特定实施例描述本发明。然而,本领域技术人员将理解,可对实施例作出各种修改和改变,而不背离如所附权利要求所述的本发明的更宽泛的精神和范围。因此,应当以说明性而非限制性的意义看待前述说明书和附图。

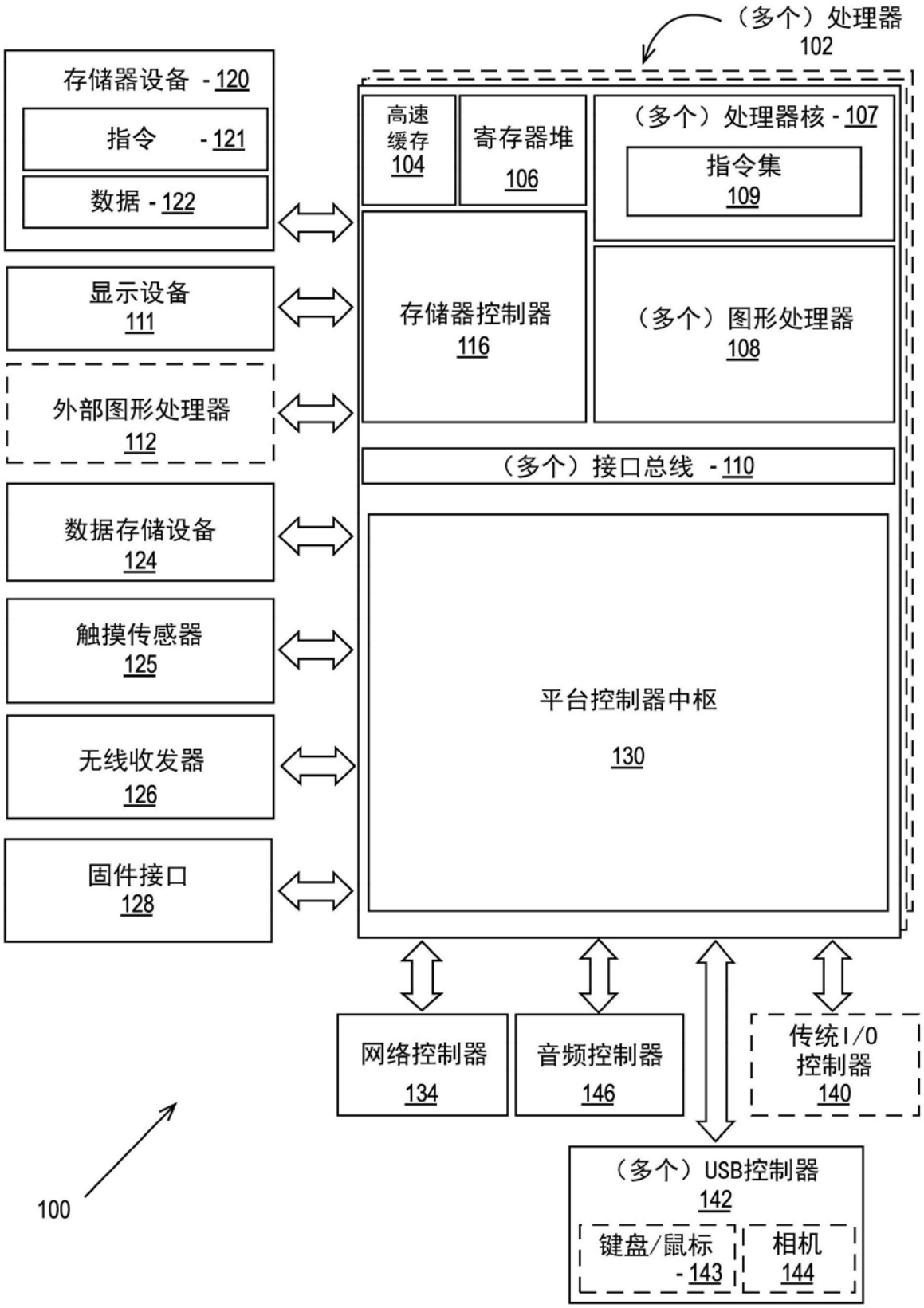

（多个）处理器
102
存储器设备 - 120
指令 - 121
数据 - 122
高速缓存
104
寄存器堆
106
（多个）处理器核 - 107
指令集
109
显示设备
111
存储器控制器
116
（多个）图形处理器
108
外部图形处理器
112
（多个）接口总线 - 110
数据存储设备
124
触摸传感器
125
平台控制器中枢
130
无线收发器
126
固件接口
128
网络控制器
134
音频控制器
146
传统I/O
控制器
140
（多个）USB控制器
142
键盘/鼠标
- 143
相机
144
100

图1

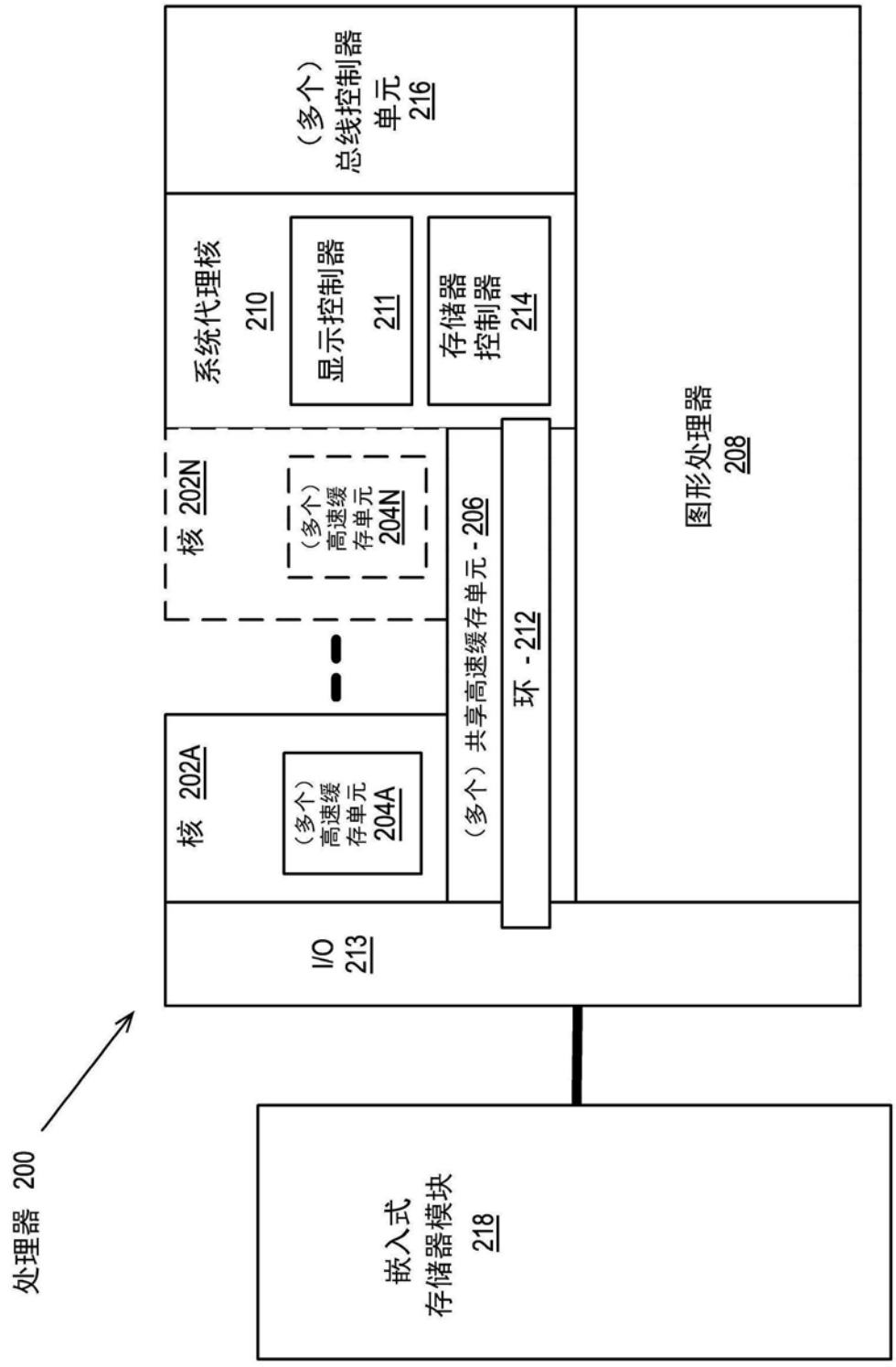

处理器 200
嵌入式
存储器模块
218
I/O
213
核 202A
（多个）
高速缓
存单元
204A
核 202N
（多个）
高速缓
存单元
204N
（多个）共享高速缓存单元 - 206
环 - 212
系统代理核
210
显示控制器
211
存储器
控制器
214
（多个）
总线控制器
单元
216
图形处理器
208

图2

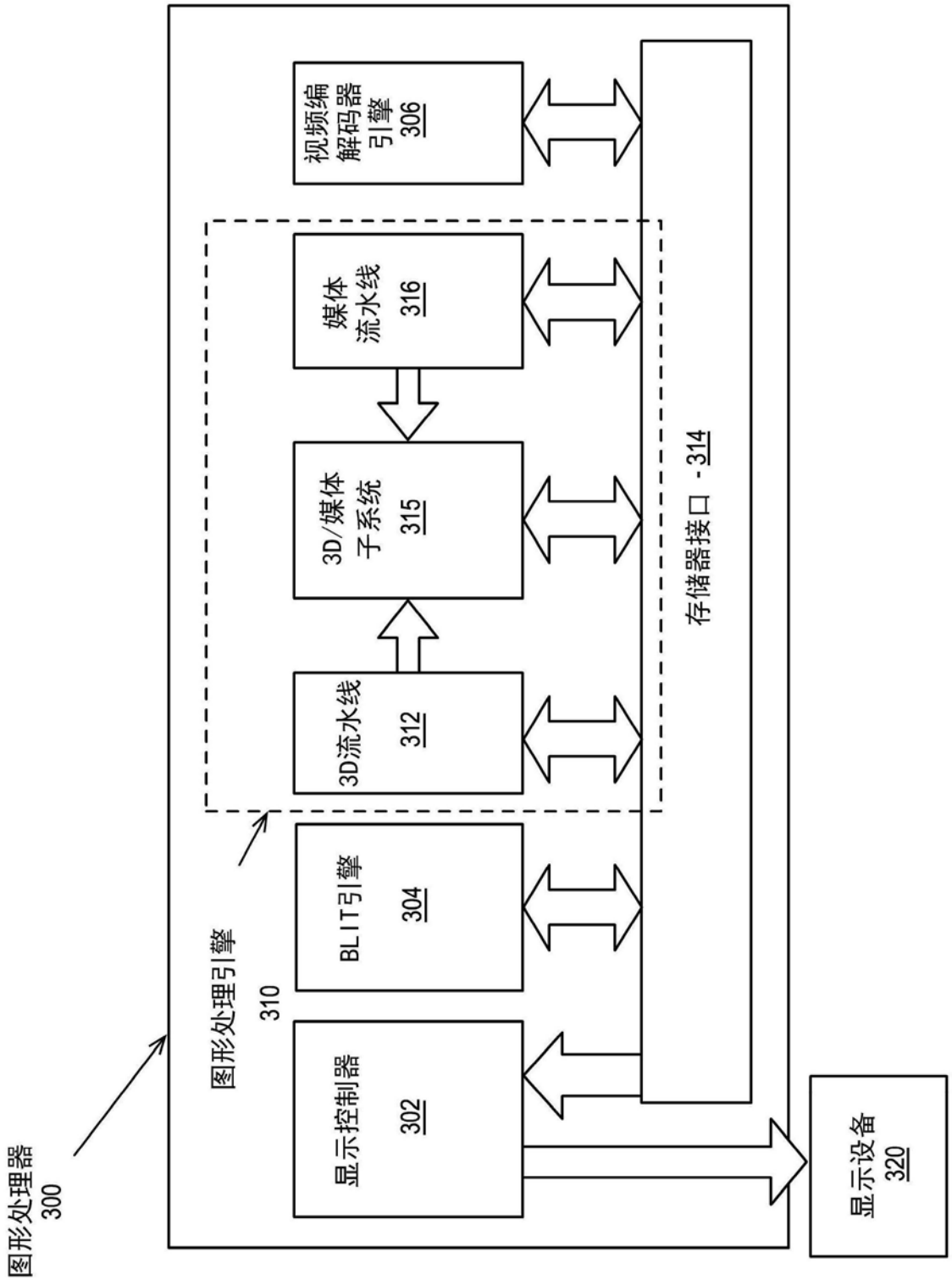

图形处理器
300
显示控制器
302
BLIT引擎
304
图形处理引擎
310
3D流水线
312
3D/媒体
子系统
315
媒体
流水线
316
视频编
解码器
引擎
306
存储器接口 - 314
显示设备
320

图3

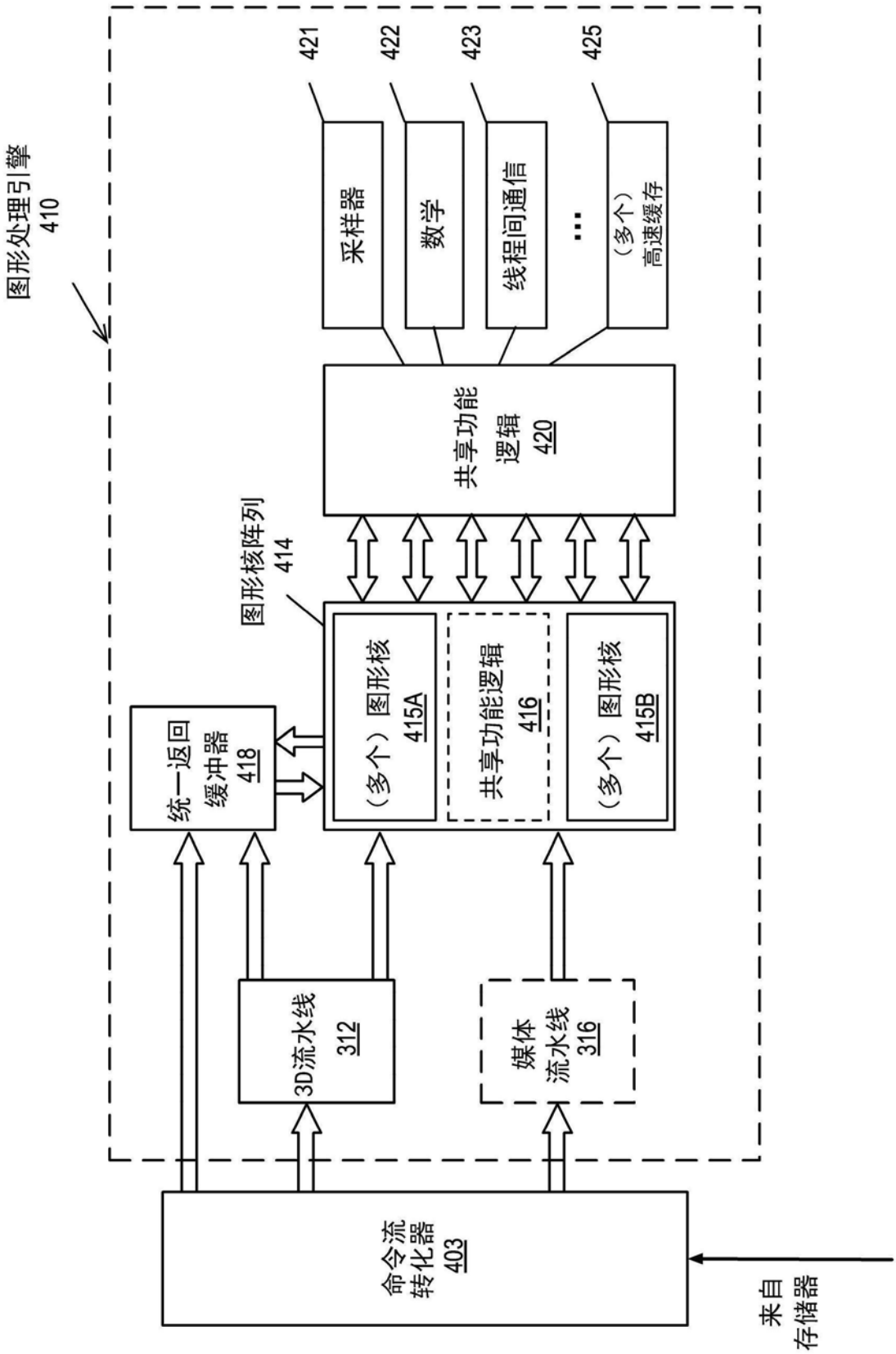

图形处理引擎
410
421
422
423
425
采样器
数学
线程间通信
...
（多个）
高速缓存
共享功能
逻辑
420
图形核阵列
414
统一返回
缓冲器
418
（多个）图形核
415A
共享功能逻辑
416
（多个）图形核
415B
3D流水线
312
媒体
流水线
316
命令流
转化器
403
来自
存储器

图4

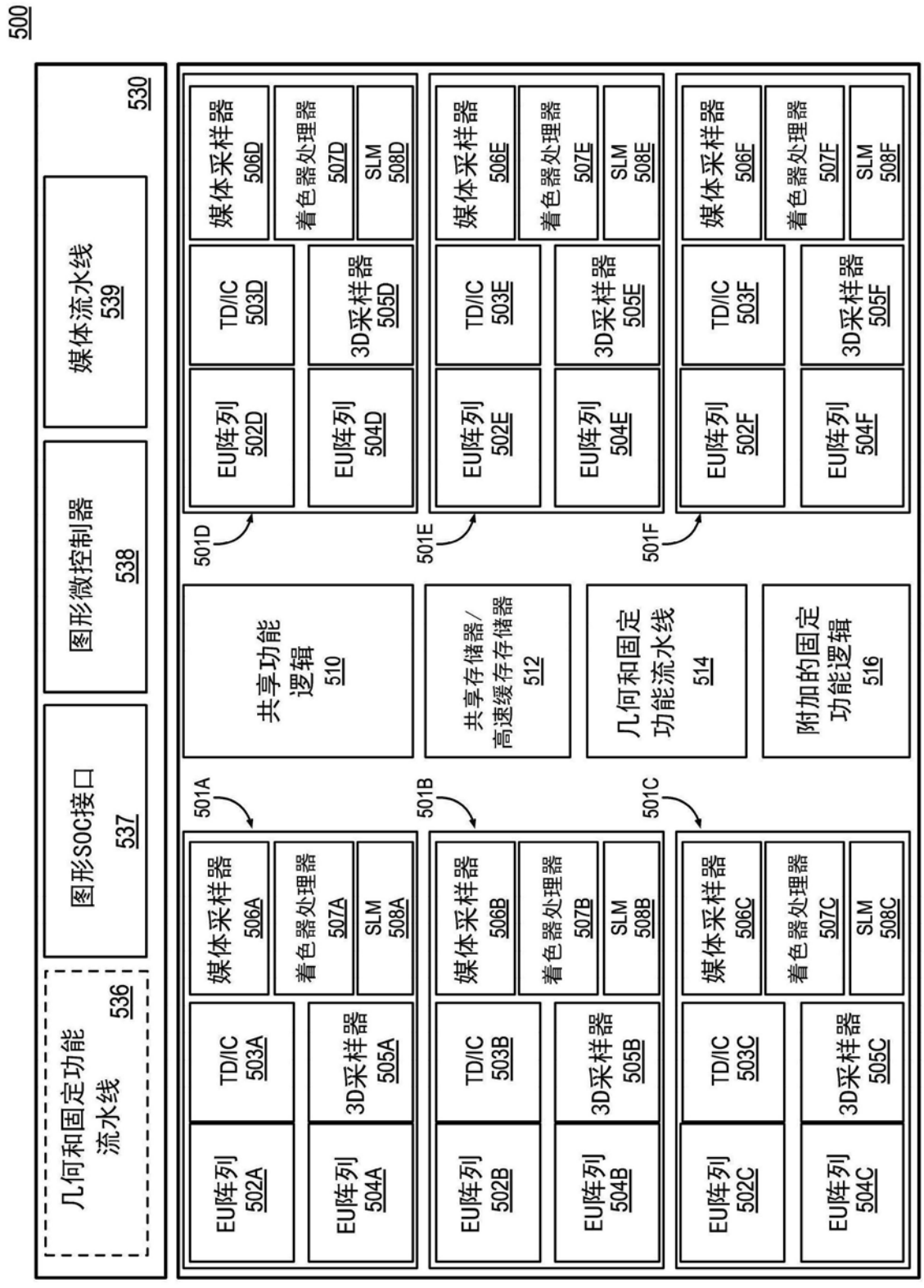

500
几何和固定功能流水线 536
图形SOC接口 537
图形微控制器 538
媒体流水线 539
530
501A
501B
501C
501D
501E
501F
EU阵列 502A
TD/IC 503A
媒体采样器 506A
着色器处理器 507A
SLM 508A
EU阵列 504A
3D采样器 505A
EU阵列 502B
TD/IC 503B
媒体采样器 506B
着色器处理器 507B
SLM 508B
EU阵列 504B
3D采样器 505B
EU阵列 502C
TD/IC 503C
媒体采样器 506C
着色器处理器 507C
SLM 508C
EU阵列 504C
3D采样器 505C
EU阵列 502D
TD/IC 503D
媒体采样器 506D
着色器处理器 507D
SLM 508D
EU阵列 504D
3D采样器 505D
EU阵列 502E
TD/IC 503E
媒体采样器 506E
着色器处理器 507E
SLM 508E
EU阵列 504E
3D采样器 505E
EU阵列 502F
TD/IC 503F
媒体采样器 506F
着色器处理器 507F
SLM 508F
EU阵列 504F
3D采样器 505F
共享功能逻辑 510
共享存储器/高速缓存存储器 512
几何和固定功能流水线 514
附加的固定功能逻辑 516

图5

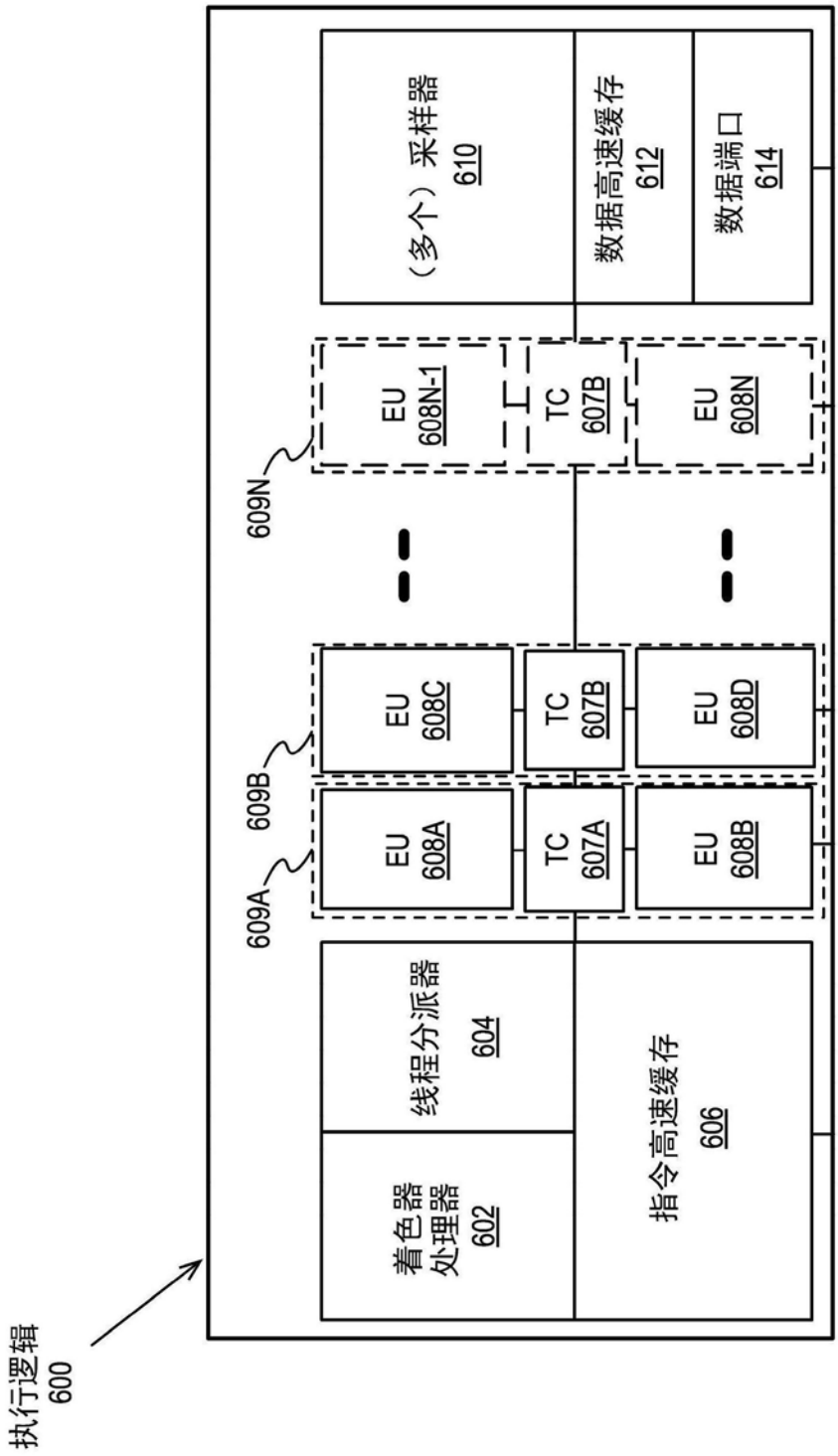

执行逻辑
600
着色器
处理器
602
线程分派器
604
指令高速缓存
606
数据端口
614
数据高速缓存
612
（多个）采样器
610
609A
609B
609N
EU
608A
TC
607A
EU
608B
EU
608C
TC
607B
EU
608D
EU
608N-1
TC
607B
EU
608N

图6A

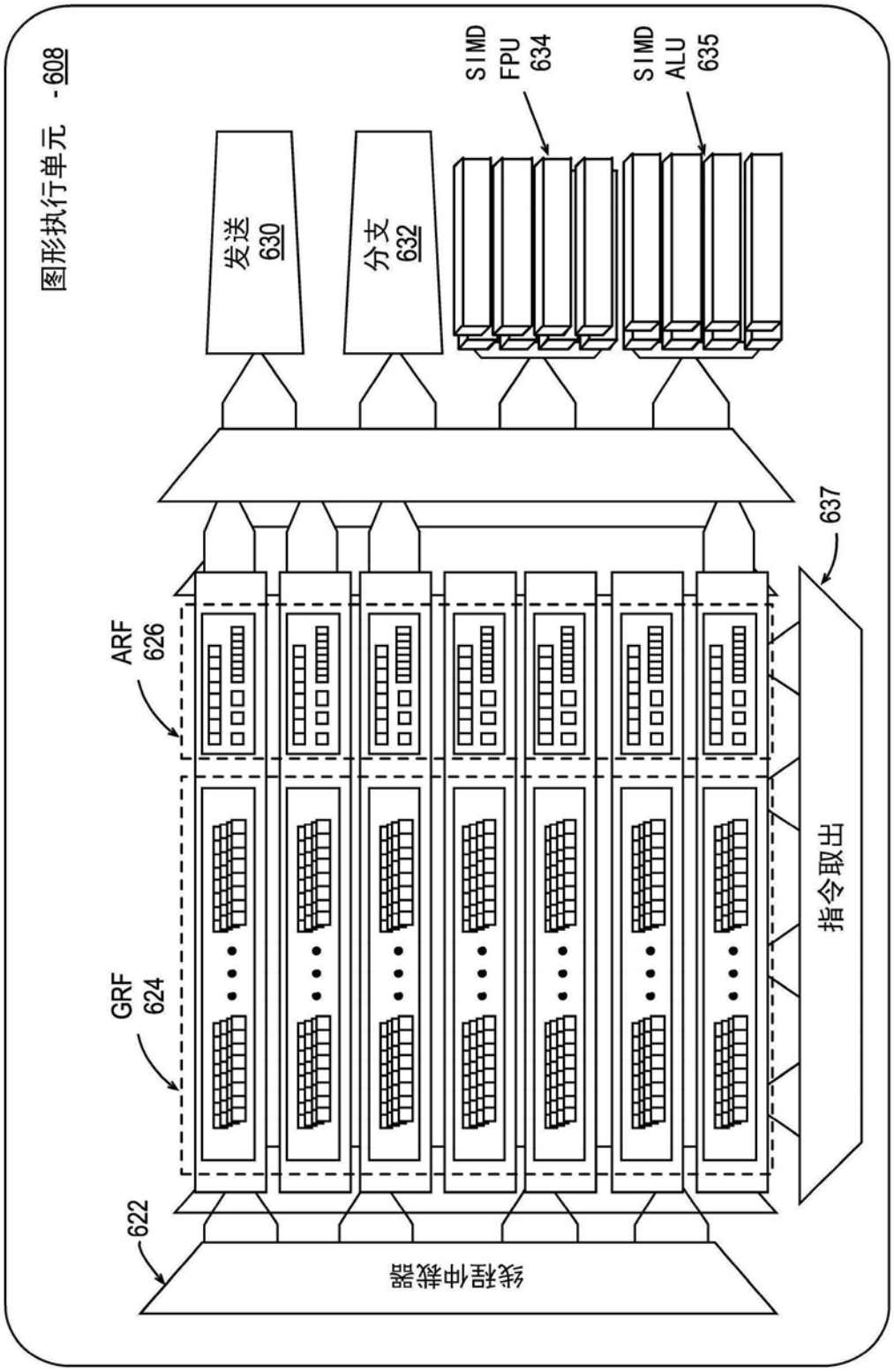

图形执行单元 -608
SIMD FPU 634
SIMD ALU 635
发送 630
分支 632
637
ARF 626
GRF 624
622
指令取出
线程仲裁器

图6B

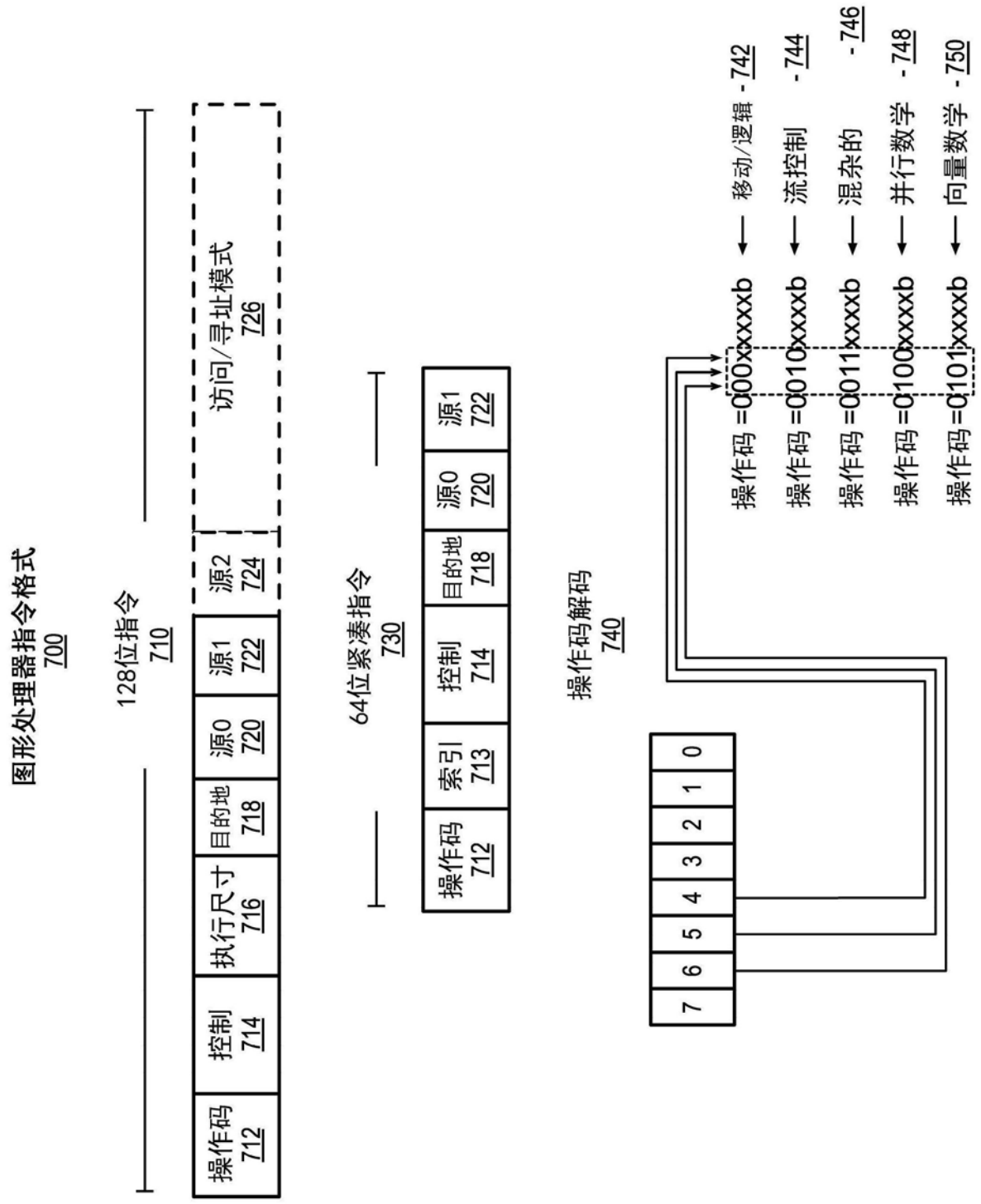

图形处理器指令格式
700
128位指令
710
操作码 712
控制 714
执行尺寸 716
目的地 718
源0 720
源1 722
源2 724
访问/寻址模式 726
64位紧凑指令
730
操作码 712
索引 713
控制 714
目的地 718
源0 720
源1 722
操作码解码
740
7 6 5 4 3 2 1 0
操作码 =00xxxxxb
移动/逻辑 - 742
操作码 =010xxxxb
流控制 - 744
操作码 =011xxxxb
混杂的 - 746
操作码 =100xxxxb
并行数学 - 748
操作码 =101xxxxb
向量数学 - 750

图7

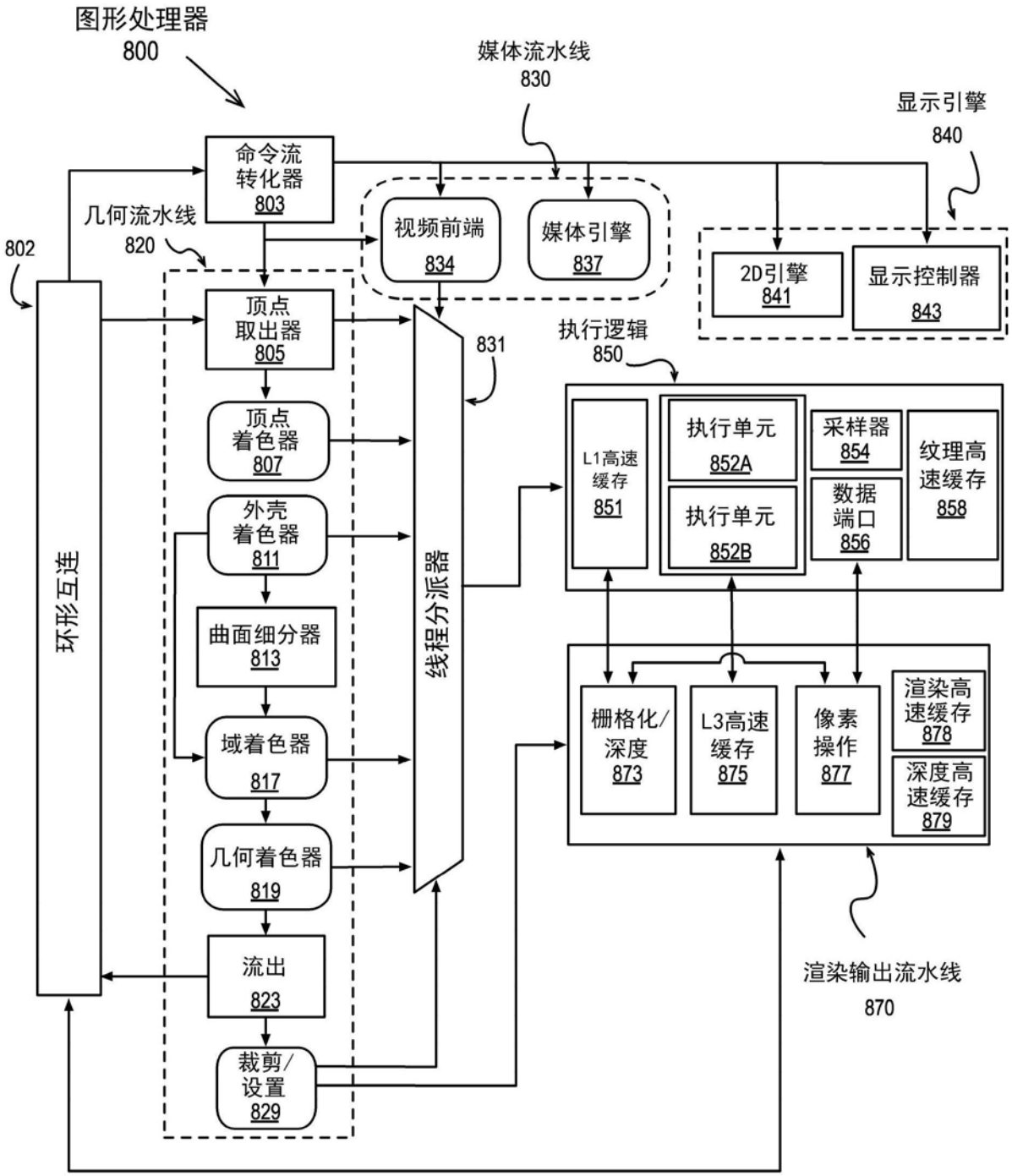

图形处理器
800
媒体流水线
830
显示引擎
840
命令流转化器
803
几何流水线
820
802
视频前端
834
媒体引擎
837
2D引擎
841
显示控制器
843
顶点取出器
805
执行逻辑
850
831
顶点着色器
807
外壳着色器
811
曲面细分器
813
域着色器
817
几何着色器
819
流出
823
裁剪/设置
829
环形互连
线程分派器
L1高速缓存
851
执行单元
852A
执行单元
852B
采样器
854
数据端口
856
纹理高速缓存
858
栅格化/深度
873
L3高速缓存
875
像素操作
877
渲染高速缓存
878
深度高速缓存
879
渲染输出流水线
870

图8

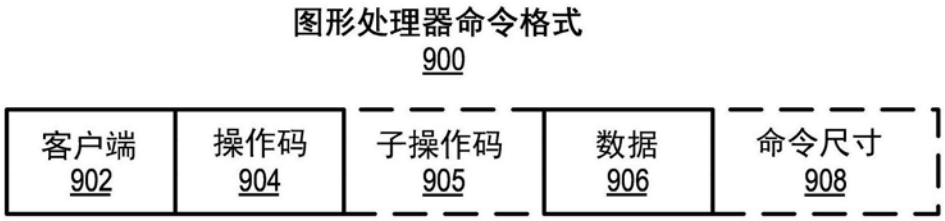

图形处理器命令格式
900
客户端
902
操作码
904
子操作码
905
数据
906
命令尺寸
908

图9A

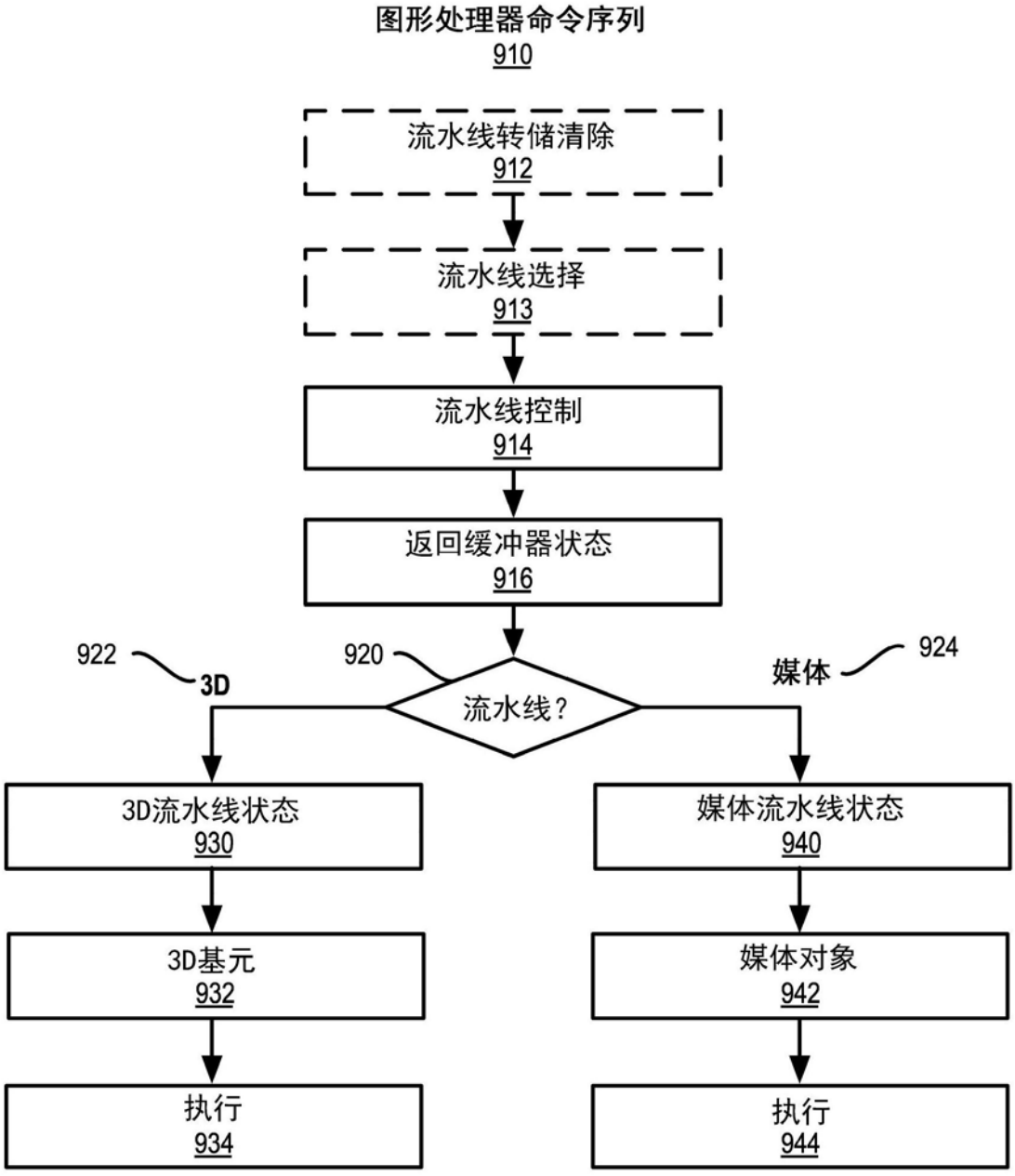

图形处理器命令序列
910
流水线转储清除
912
流水线选择
913
流水线控制
914
返回缓冲器状态
916
922
3D
920
流水线?
媒体
924
3D流水线状态
930
3D基元
932
执行
934
媒体流水线状态
940
媒体对象
942
执行
944

图9B

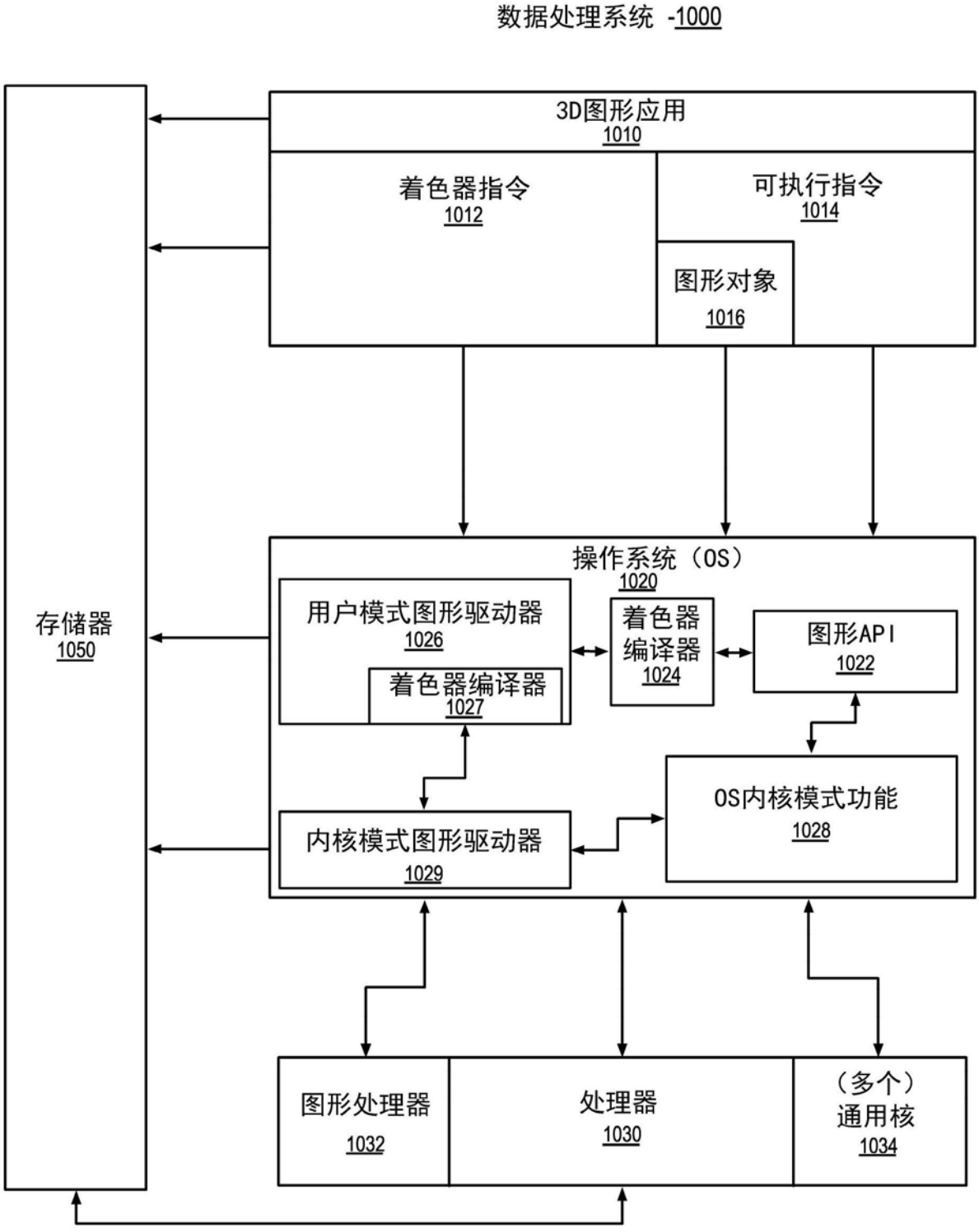

数据处理系统 -1000
3D图形应用
1010
着色器指令
1012
可执行指令
1014
图形对象
1016
操作系统（OS）
1020
存储器
1050
用户模式图形驱动器
1026
着色器编译器
1027
着色器编译器
1024
图形API
1022
OS内核模式功能
1028
内核模式图形驱动器
1029
图形处理器
1032
处理器
1030
（多个）通用核
1034

图10

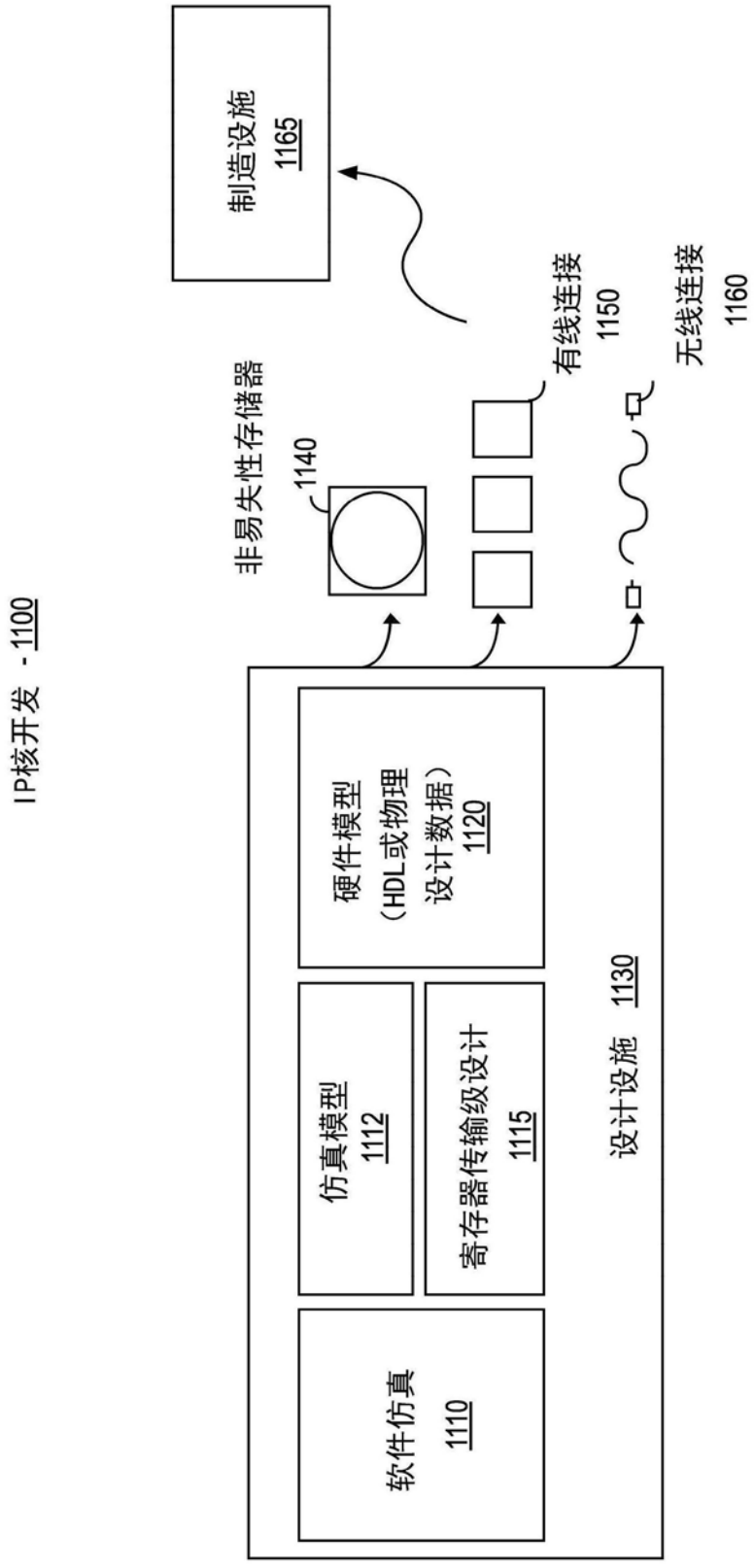

IP核开发 - 1100
制造设施 1165
非易失性存储器 1140
有线连接 1150
无线连接 1160
硬件模型 (HDL或物理 设计数据) 1120
仿真模型 1112
寄存器传输级设计 1115
软件仿真 1110
设计设施 1130

图11A

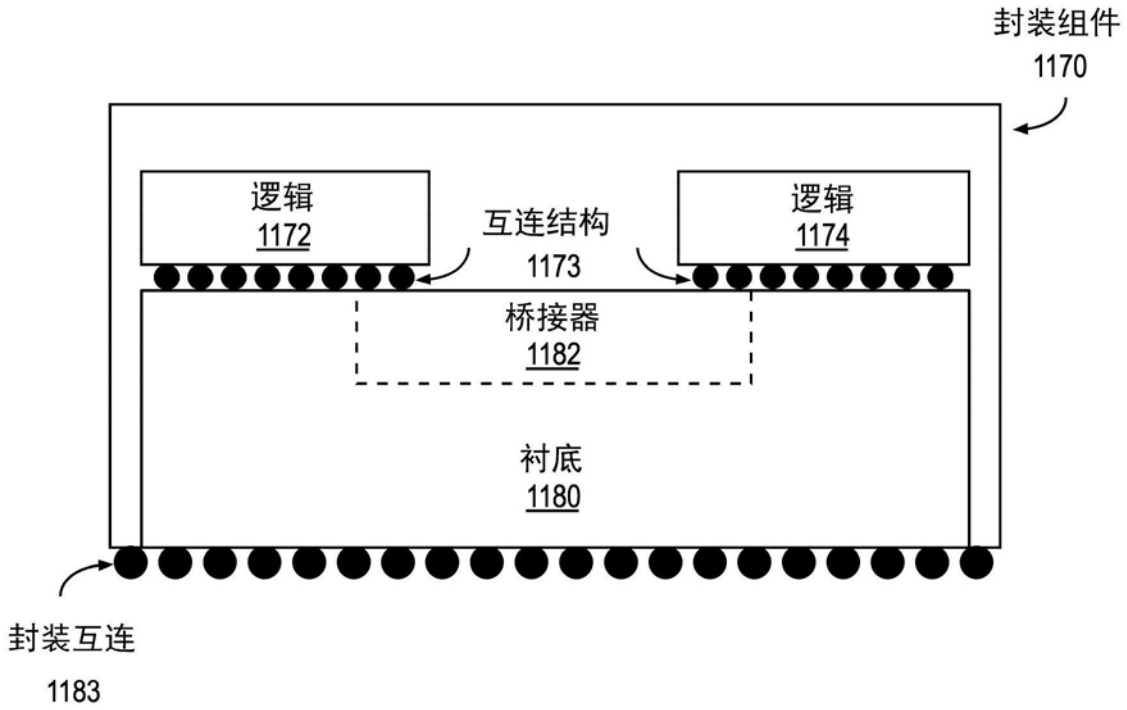


图11B

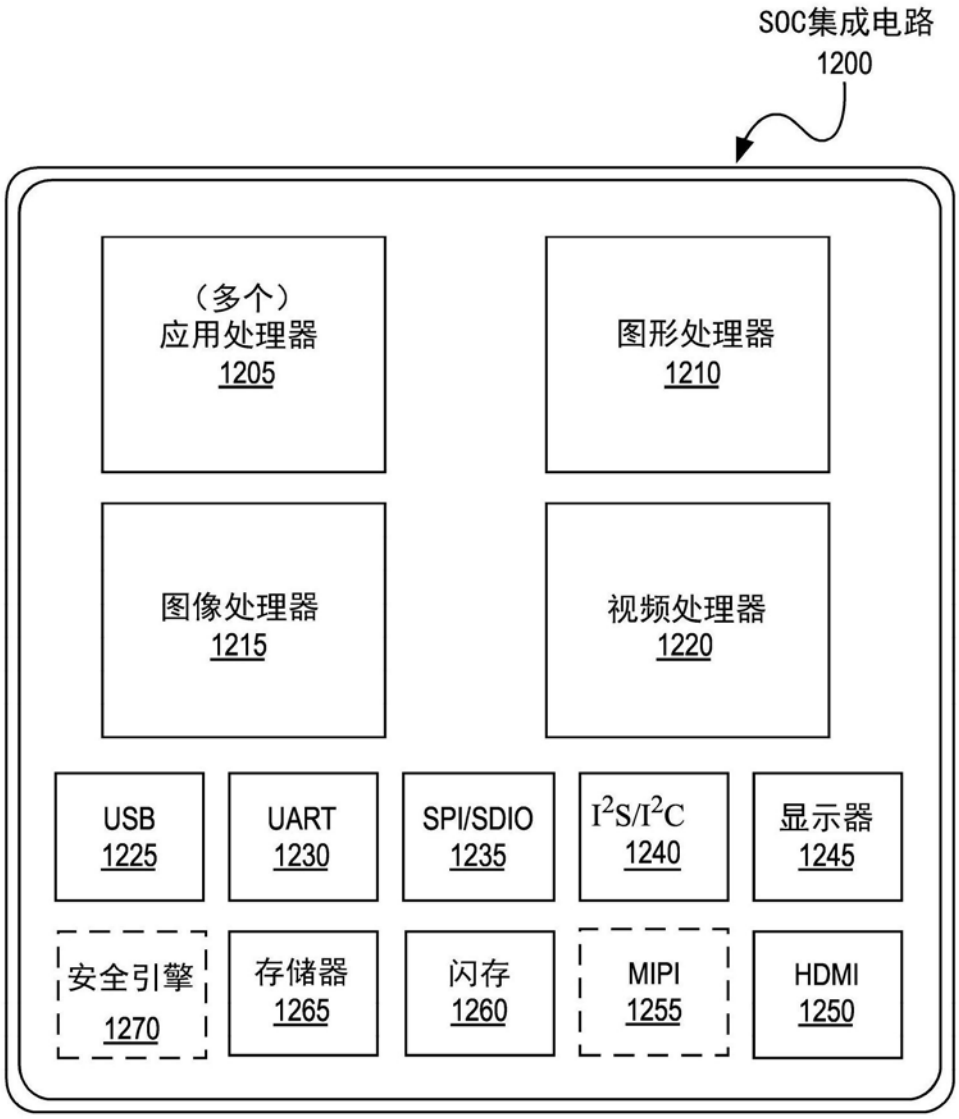

SOC集成电路
1200
（多个）
应用处理器
1205
图形处理器
1210
图像处理器
1215
视频处理器
1220
USB
1225
UART
1230
SPI/SDIO
1235
I2S/I2C
1240
显示器
1245
安全引擎
1270
存储器
1265
闪存
1260
MIPI
1255
HDMI
1250

图12

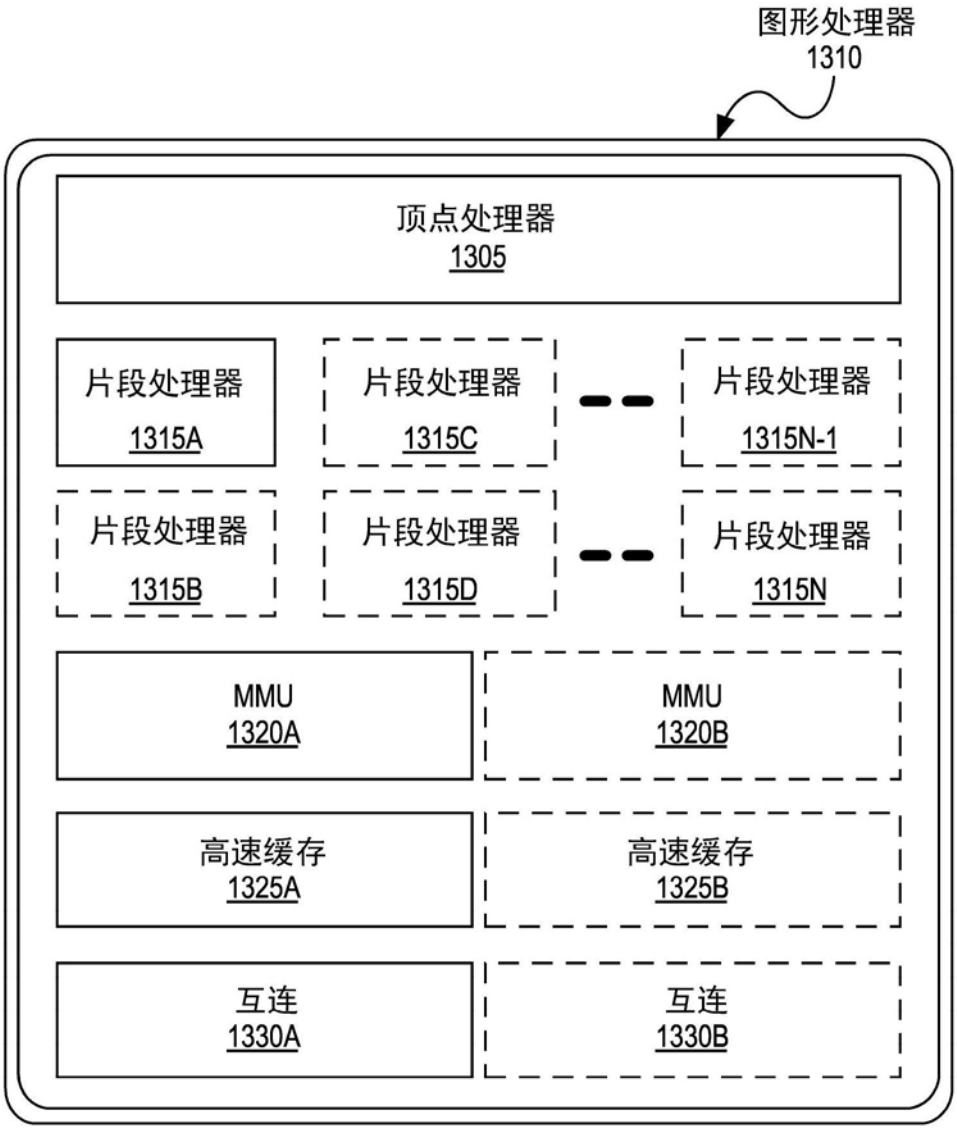

图形处理器
1310
顶点处理器
1305
片段处理器
1315A
片段处理器
1315C
片段处理器
1315N-1
片段处理器
1315B
片段处理器
1315D
片段处理器
1315N
MMU
1320A
MMU
1320B
高速缓存
1325A
高速缓存
1325B
互连
1330A
互连
1330B

图13A

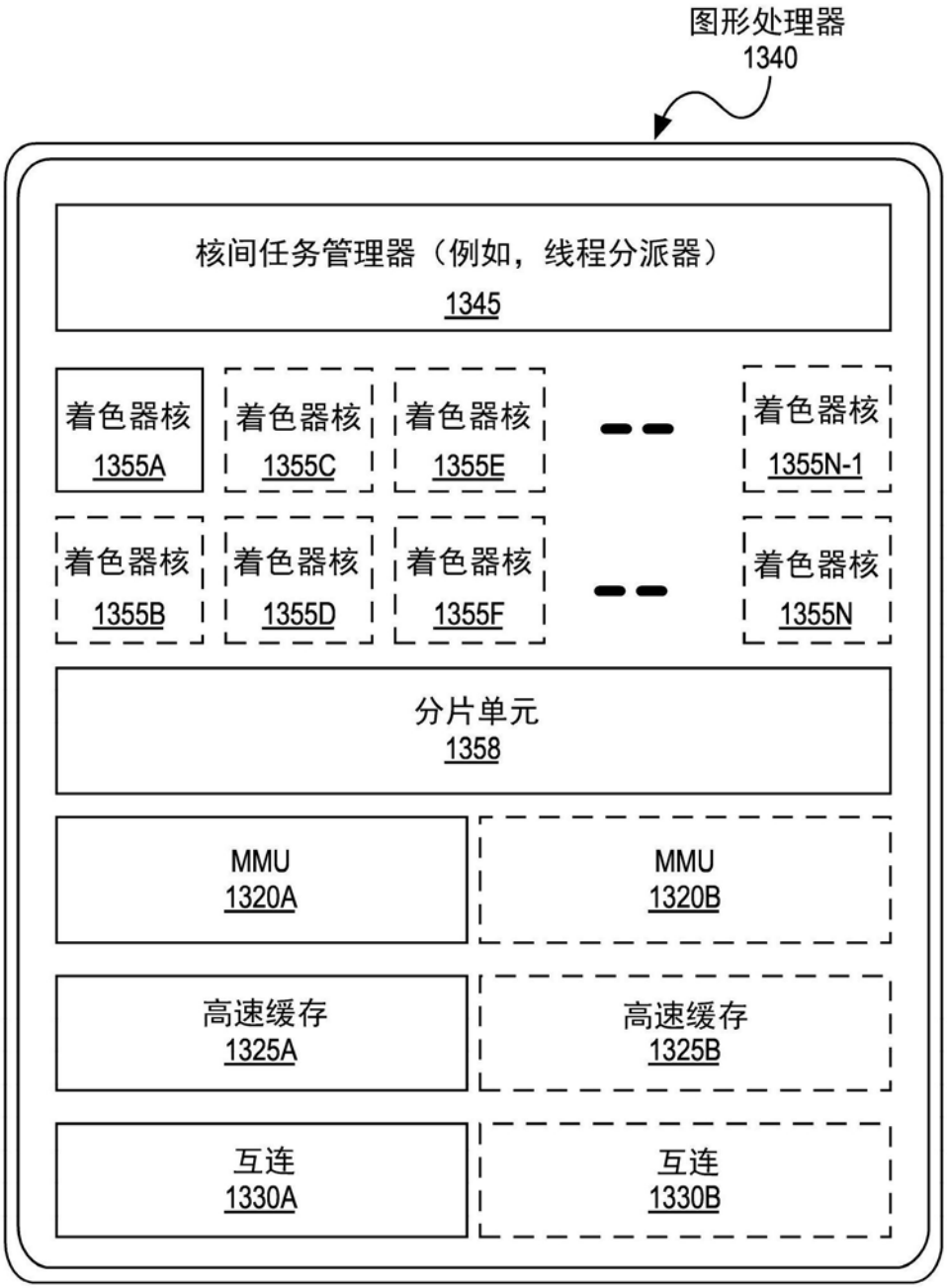

图形处理器
1340
核间任务管理器（例如，线程分派器）
1345
着色器核
1355A
着色器核
1355C
着色器核
1355E
着色器核
1355N-1
着色器核
1355B
着色器核
1355D
着色器核
1355F
着色器核
1355N
分片单元
1358
MMU
1320A
MMU
1320B
高速缓存
1325A
高速缓存
1325B
互连
1330A
互连
1330B

图13B

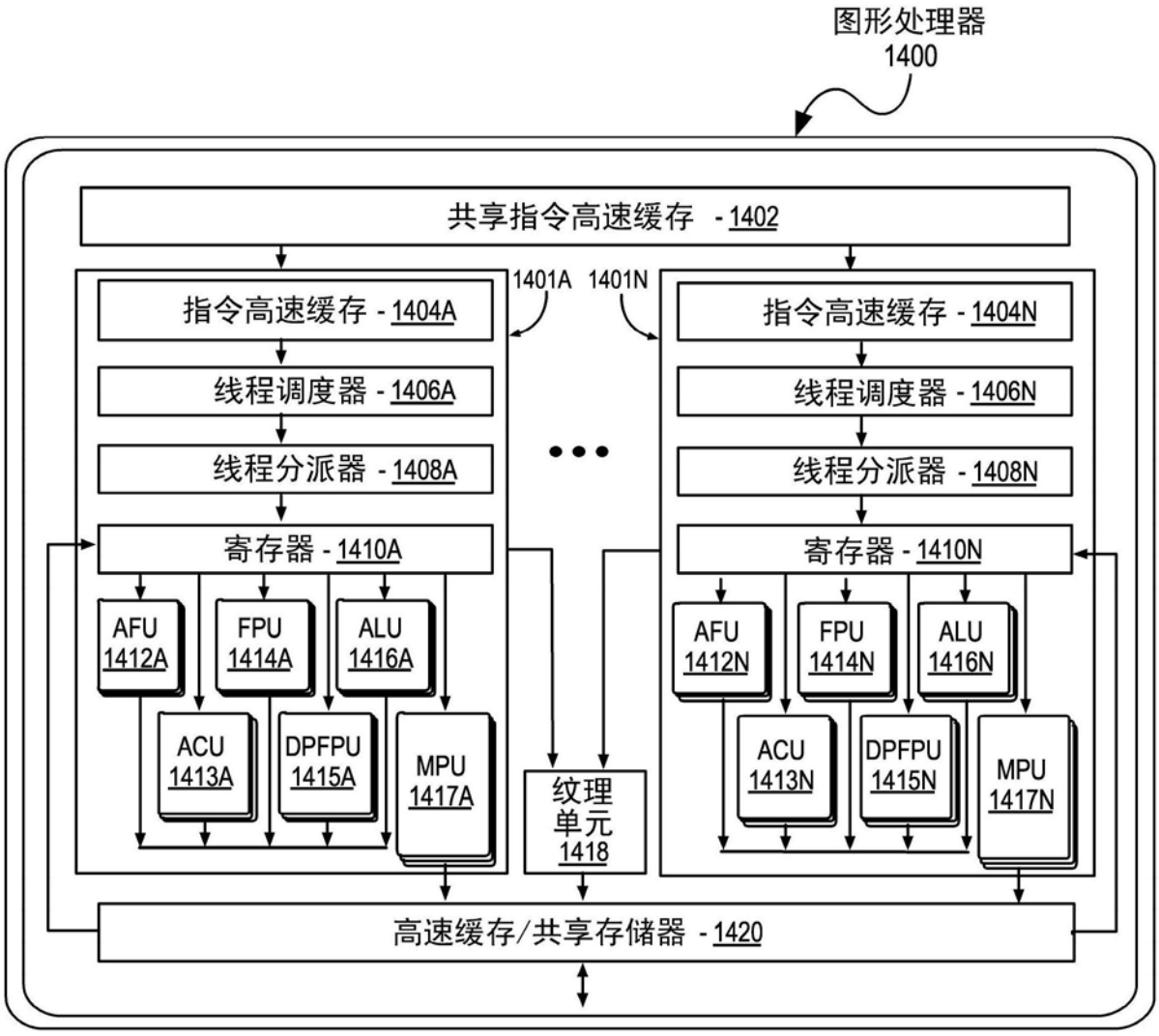

图形处理器
1400
共享指令高速缓存 - 1402
1401A
1401N
指令高速缓存 - 1404A
线程调度器 - 1406A
线程分派器 - 1408A
寄存器 - 1410A
AFU 1412A
FPU 1414A
ALU 1416A
ACU 1413A
DPFPU 1415A
MPU 1417A
纹理单元 1418
指令高速缓存 - 1404N
线程调度器 - 1406N
线程分派器 - 1408N
寄存器 - 1410N
AFU 1412N
FPU 1414N
ALU 1416N
ACU 1413N
DPFPU 1415N
MPU 1417N
高速缓存/共享存储器 - 1420

图14A

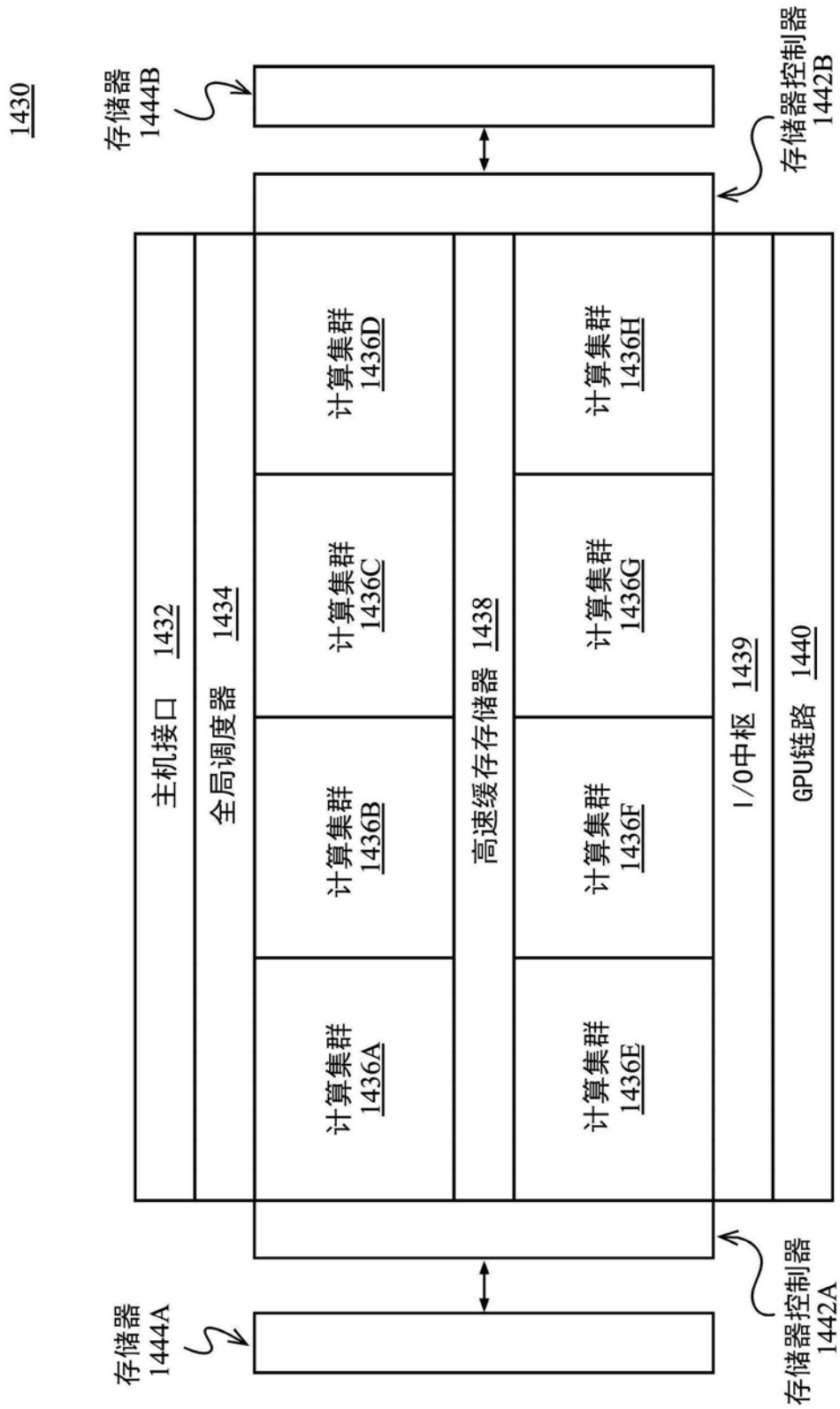

1430
存储器 1444B
存储器控制器 1442B
主机接口 1432
全局调度器 1434
计算集群 1436A
计算集群 1436B
计算集群 1436C
计算集群 1436D
计算集群 1436E
计算集群 1436F
计算集群 1436G
计算集群 1436H
高速缓存存储器 1438
I/O中枢 1439
GPU链路 1440
存储器 1444A
存储器控制器 1442A

图14B


计算设备（例如，主机）
1500
操作系统（OS）
1506
图形驱动器
1516
图形处理单元（GPU）
1514
中央处理单元（CPU）
1512
存储器
1508
（多个）输入输出（I/O）源（例如，（多个）相机、
（多个）微处理器、（多个）扬声器、（多个）传感器、
（多个）显示屏、（多个）媒体播放器等等）
1504

图15

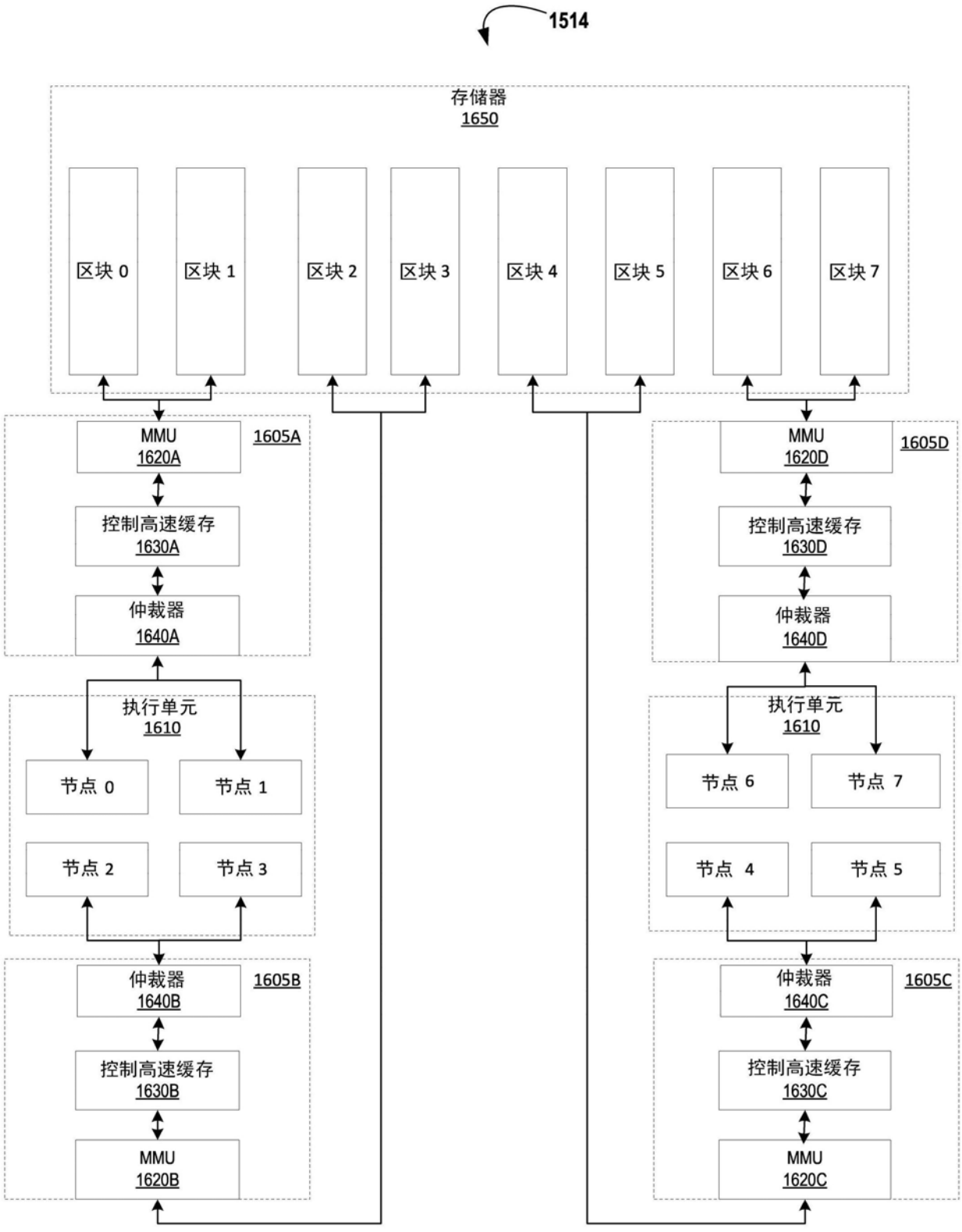

1514
存储器
1650
区块 0
区块 1
区块 2
区块 3
区块 4
区块 5
区块 6
区块 7
MMU
1620A
1605A
控制高速缓存
1630A
仲裁器
1640A
执行单元
1610
节点 0
节点 1
节点 2
节点 3
仲裁器
1640B
1605B
控制高速缓存
1630B
MMU
1620B
MMU
1620D
1605D
控制高速缓存
1630D
仲裁器
1640D
执行单元
1610
节点 6
节点 7
节点 4
节点 5
仲裁器
1640C
1605C
控制高速缓存
1630C
MMU
1620C

图16

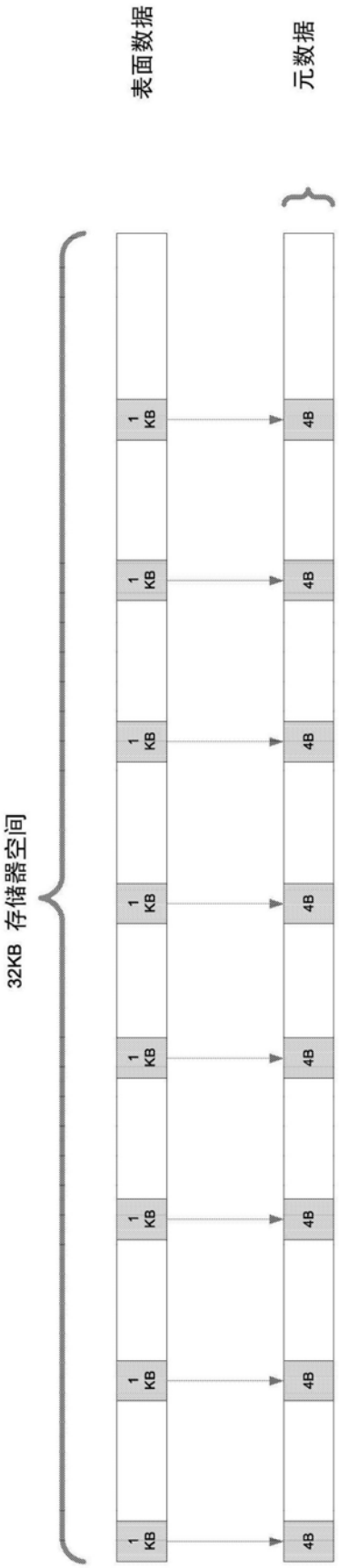


图17

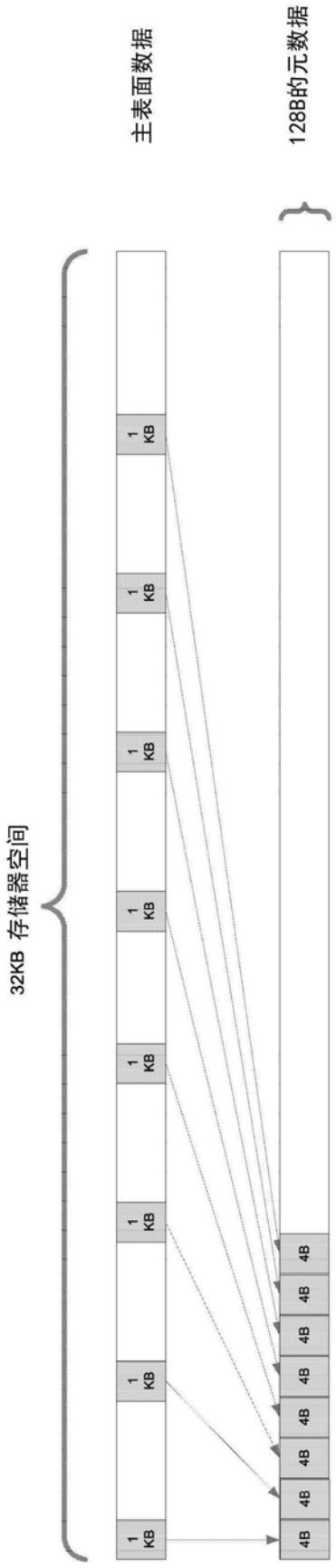


图18

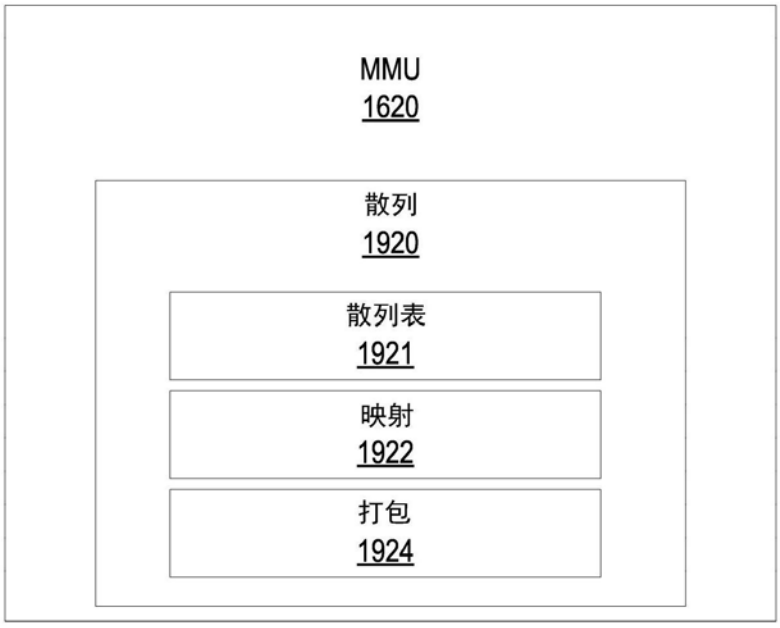

MMU
1620
散列
1920
散列表
1921
映射
1922
打包
1924

图19

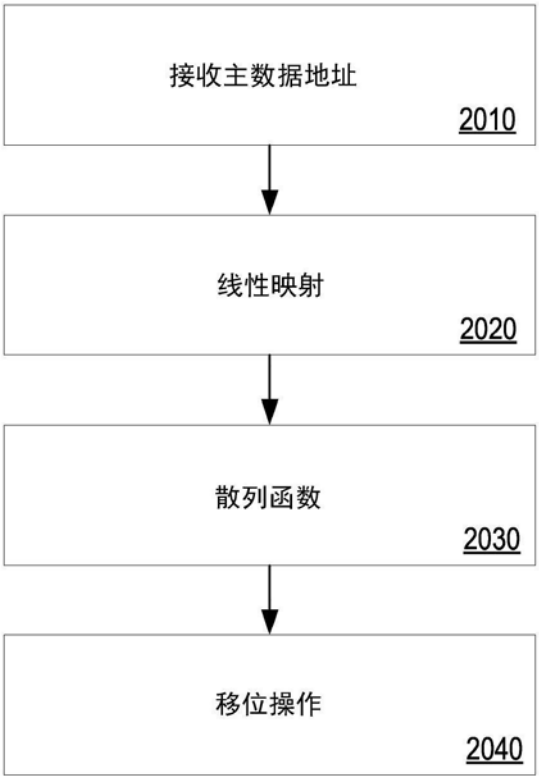

接收主数据地址
2010
线性映射
2020
散列函数
2030
移位操作
2040

图20