



(12) 发明专利申请

(10) 申请公布号 CN 114830189 A

(43) 申请公布日 2022. 07. 29

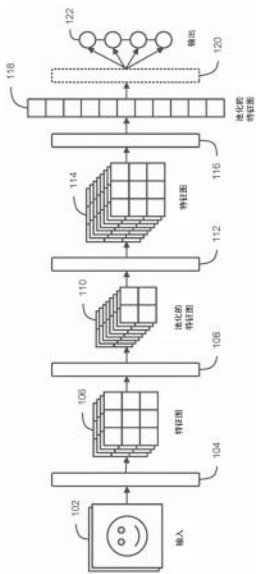
(21) 申请号 201980102206.6
(22) 申请日 2019.11.20
(85) PCT国际申请进入国家阶段日
2022.05.11
(86) PCT国际申请的申请数据
PCT/CN2019/119792 2019.11.20
(87) PCT国际申请的公布数据
W02021/097728 EN 2021.05.27
(71) 申请人 辉达公司
地址 美国加利福尼亚州
(72) 发明人 赵基宇 高爽 刘昕
R·B·特瓦尔
(74) 专利代理机构 北京市磐华律师事务所
11336
专利代理师 高伟

(51) Int.Cl.
G06V 20/56 (2022.01)
G06V 10/44 (2022.01)
G06V 10/82 (2022.01)
G06V 10/94 (2022.01)
G06V 10/96 (2022.01)

权利要求书3页 说明书85页 附图52页

(54) 发明名称
利用神经网络识别多尺度特征

(57) 摘要
用于识别一个或多个图像内的特征的装置、系统和技术。使用一个或多个神经网络识别一个或多个图像中的特征,所述神经网络包含具有可由一个或多个并行处理单元执行的多个滤波器的卷积层。



1. 一种处理器,包括:

一个或多个电路,用于使用一个或多个神经网络来至少部分地基于图像内的一个或多个特征的位置识别所述图像中的所述一个或多个特征。

2. 如权利要求1所述的处理器,其中所述一个或多个电路用于至少通过以下方式识别所述图像中的所述一个或多个特征:

计算用于所述位置的第一滤波器的第一值;

计算用于所述位置的第二滤波器的第二值;以及

至少部分地基于所述第一值和所述第二值计算特征图的第三值。

3. 如权利要求2所述的处理器,其中所述第三值至少部分地通过聚合所述第一值和所述第二值来计算。

4. 如权利要求2所述的处理器,其中所述第一值和所述第二值指示所述第一滤波器和所述第二滤波器检测到所述位置的所述一个或多个特征的概率。

5. 如权利要求1所述的处理器,其中所述一个或多个神经网络是卷积神经网络。

6. 如权利要求1所述的处理器,其中所述一个或多个电路用于至少部分地基于两个或多个滤波器来计算特征图,其中计算所述特征图包括:不同地组合来自不同位置的所述两个或多个滤波器的值。

7. 如权利要求1所述的处理器,其中所述一个或多个神经网络在所述一个或多个神经网络中的卷积神经网络的卷积层中识别一个或多个特征。

8. 如权利要求7所述的处理器,其中所述卷积层包括逐深度卷积和逐点卷积。

9. 一种机器可读介质,具有存储于其上的一组指令,所述指令如果由一个或多个处理器执行,则使所述一个或多个处理器至少:

使用一个或多个神经网络至少部分地基于图像内的一个或多个特征的位置来识别所述图像中的所述一个或多个特征。

10. 如权利要求9所述的机器可读介质,其中所述指令在被执行时,使所述一个或多个处理器至少通过以下方式识别所述图像内的所述一个或多个特征:

计算用于所述位置的第一滤波器的第一值;

计算用于所述位置的第二滤波器的第二值;以及

至少部分地基于所述第一值和所述第二值计算特征图的第三值。

11. 如权利要求10所述的机器可读介质,其中所述指令在被执行时,进一步使所述一个或多个处理器至少部分地通过聚合所述第一值和所述第二值来计算所述第三值。

12. 如权利要求10所述的机器可读介质,其中所述第一值和所述第二值指示所述第一滤波器或所述第二滤波器检测到所述位置的所述一个或多个特征的权重。

13. 如权利要求12所述的机器可读介质,其中所述指令在被执行时,使所述一个或多个处理器使用softmax函数计算所述权重。

14. 如权利要求9所述的机器可读介质,其中所述一个或多个神经网络是卷积神经网络。

15. 如权利要求9所述的机器可读介质,其中所述指令在被执行时,使所述一个或多个处理器至少部分地基于两个或多个滤波器来计算特征图,其中计算所述特征图包括:不同地组合来自不同位置的所述两个或多个滤波器的值。

16. 如权利要求9所述的机器可读介质,其中所述指令在被执行时,使所述一个或更多个处理器在所述一个或更多个神经网络中的卷积神经网络的卷积层中识别一个或更多个特征,所述卷积层包括逐深度卷积和逐点卷积。

17. 一种处理器,包括:

一个或更多个电路,用于帮助训练一个或更多个神经网络,以至少部分地基于图像内的一个或更多个特征的位置识别所述图像中的所述一个或更多个特征。

18. 如权利要求17所述的处理器,其中所述一个或更多个电路用于至少通过以下方式识别所述一个或更多个特征:

计算用于所述位置的第一滤波器的第一值;

计算用于所述位置的第二滤波器的第二值;以及

至少部分地基于所述第一值和所述第二值计算特征图的第三值。

19. 如权利要求18所述的处理器,其中至少部分地通过将softmax函数应用于所述第一值和所述第二值且将所述softmax函数应用于所述第一值和所述第二值的结果进行聚合,来计算所述第三值。

20. 如权利要求18所述的处理器,其中所述第一值和所述第二值指示所述第一滤波器或所述第二滤波器检测到所述位置的所述一个或更多个特征的概率。

21. 如权利要求17所述的处理器,其中要被训练的所述一个或更多个神经网络是卷积神经网络。

22. 如权利要求17所述的处理器,其中所述一个或更多个电路用于至少部分地基于两个或更多个滤波器来计算特征图,其中计算所述特征图包括:不同地组合来自不同位置的所述两个或更多个滤波器的值。

23. 如权利要求17所述的处理器,其中所述一个或更多个神经网络在所述一个或更多个神经网络中的卷积神经网络的卷积层中识别一个或更多个特征。

24. 如权利要求23所述的处理器,其中所述卷积层包括逐深度卷积和逐点卷积。

25. 一种方法,包括:

训练一个或更多个神经网络,以至少部分地基于图像内的一个或更多个特征的位置识别所述图像中的所述一个或更多个特征。

26. 如权利要求25所述的方法,其中所述一个或更多个神经网络包括至少一个卷积层,所述卷积层:

将一个或更多个滤波器应用于所述图像;

计算与所述一个或更多个滤波器的输出相关联的权重;以及

将所述权重聚合成所述图像的一个或更多个特征图。

27. 如权利要求26所述的方法,其中计算所述图像中的不同位置处的权重。

28. 如权利要求26所述的方法,其中所述一个或更多个滤波器中的每一个识别所述图像中的所述一个或更多个特征中的每一个。

29. 如权利要求25所述的方法,其中要被训练的所述一个或更多个神经网络是卷积神经网络。

30. 如权利要求25所述的方法,其中所述一个或更多个特征图包括对应于所述一个或更多个特征的值,其中所述一个或更多个特征中的每一个在所述图像中的不同位置处。

31. 如权利要求26所述的方法,其中所述卷积层包括逐深度卷积和逐点卷积。

利用神经网络识别多尺度特征

技术领域

[0001] 至少一个实施例涉及用于促进和执行人工智能的卷积神经网络中的改进的卷积层。例如，至少一个实施例涉及用于根据本文中描述的各种新颖技术在卷积层中并行地识别特征的处理器和计算系统。

背景技术

[0002] 执行卷积层步骤以在卷积神经网络中产生特征图会基于在一个或多个输入数据项中被识别的特征而增加复杂度。识别数据中特定特征所需要的内存、时间以及计算资源的量随着要被识别的每个特征而增加。通常，资源和计算复杂度会由于卷积层中用于识别每个个体特征所需的每个滤波器而增加。这些资源和计算复杂度在输入数据包括多个个体信息层时进一步增加。尤其是，对于输入数据项集合中的每一个信息层，针对要被识别的每一个特征，卷积神经网络中的卷积层将需要对输入数据项集合中的每一层应用特定尺寸的个体滤波器。由于对输入数据项集合中的每一层应用滤波器的操作会针对被应用的每一个滤波器（例如，在可分离的卷积层中）产生个体输出，因此必须执行逐点操作以便将不同的滤波器信息聚合到每个输入数据项的输出特征图中。随着更多更复杂的特征在数据集中被识别，滤波器计数和相关资源需求大大增加。

附图说明

- [0003] 图1示出了根据至少一个实施例的示例卷积神经网络；
- [0004] 图2A示出了根据至少一个实施例的输入数据项中的竖直线特征；
- [0005] 图2B示出了根据至少一个实施例的输入数据项中的水平线特征；
- [0006] 图2C示出了根据至少一个实施例的输入数据项中的多尺度特征；
- [0007] 图3示出了根据至少一个实施例的卷积神经网络的卷积层中的逐深度卷积；
- [0008] 图4示出了根据至少一个实施例的卷积神经网络的卷积层中的逐点卷积；
- [0009] 图5示出了根据至少一个实施例的卷积神经网络中的可分离卷积层；
- [0010] 图6示出了根据至少一个实施例的在卷积神经网络的空间自适应可分离卷积层中执行逐深度卷积的架构；
- [0011] 图7示出了根据至少一个实施例的卷积神经网络中的空间自适应可分离卷积层；
- [0012] 图8示出了根据至少一个实施例的用于在空间自适应可分离卷积层中生成特征图的过程；
- [0013] 图9A示出了根据至少一个实施例的推理和/或训练逻辑；
- [0014] 图9B示出了根据至少一个实施例的推理和/或训练逻辑；
- [0015] 图10示出了根据至少一个实施例的神经网络的训练和部署；
- [0016] 图11示出了根据至少一个实施例的示例数据中心系统；
- [0017] 图12A示出了根据至少一个实施例的自主车辆的示例；
- [0018] 图12B示出了根据至少一个实施例的图12A的自主车辆的相机位置和视野的示例；

- [0019] 图12C是根据至少一个实施例的示出图12A的自主车辆的示例系统架构的框图；
- [0020] 图12D是根据至少一个实施例的示出用于一个或多个基于云的服务器与图12A的自主车辆之间进行通信的系统的图；
- [0021] 图13是根据至少一个实施例的示出计算机系统的框图；
- [0022] 图14是根据至少一个实施例的示出计算机系统的框图；
- [0023] 图15示出了根据至少一个实施例的计算机系统；
- [0024] 图16示出了根据至少一个实施例的计算机系统；
- [0025] 图17A示出了根据至少一个实施例的计算机系统；
- [0026] 图17B示出了根据至少一个实施例的计算机系统；
- [0027] 图17C示出了根据至少一个实施例的计算机系统；
- [0028] 图17D示出了根据至少一个实施例的计算机系统；
- [0029] 图17E和图17F示出了根据至少一个实施例的共享编程模型；
- [0030] 图18示出了根据至少一个实施例的示例性集成电路和相关的图形处理器；
- [0031] 图19A-图19B示出了根据至少一个实施例的示例性集成电路和相关联的图形处理器；
- [0032] 图20A-图20B示出了根据至少一个实施例的附加的示例性图形处理器逻辑；
- [0033] 图21示出了根据至少一个实施例的计算机系统；
- [0034] 图22A示出了根据至少一个实施例的并行处理器；
- [0035] 图22B示出了根据至少一个实施例的分区单元；
- [0036] 图22C示出了根据至少一个实施例的处理集群；
- [0037] 图22D示出了根据至少一个实施例的图形多处理器；
- [0038] 图23示出了根据至少一个实施例的多图形处理单元 (GPU) 系统；
- [0039] 图24示出了根据至少一个实施例的图形处理器；
- [0040] 图25是根据至少一个实施例的示出用于处理器的处理器微架构的框图；
- [0041] 图26示出了根据至少一个实施例的深度学习应用程序处理器；
- [0042] 图27是根据至少一个实施例的示出了示例神经形态处理器的框图；
- [0043] 图28示出了根据一个或多个实施例的图形处理器的至少部分；
- [0044] 图29示出了根据一个或多个实施例的图形处理器的至少部分；
- [0045] 图30示出了根据一个或多个实施例的图形处理器的至少部分；
- [0046] 图31是根据至少一个实施例的图形处理器的图形处理引擎3110的框图；
- [0047] 图32是根据至少一个实施例的图形处理器核心的至少部分的框图；
- [0048] 图33A-图33B示出了根据至少一个实施例的线程执行逻辑3300,其包括图形处理器核心的处理元件的阵列。
- [0049] 图34示出了根据至少一个实施例的并行处理单元 (“PPU”)；
- [0050] 图35示出了根据至少一个实施例的通用处理集群 (“GPC”)；
- [0051] 图36示出了根据至少一个实施例的并行处理单元 (“PPU”) 的存储器分区单元；以及
- [0052] 图37示出了根据至少一个实施例的流式多处理器。

具体实施方式

[0053] 图1示出了根据至少一个实施例的包含若干卷积层104、112的卷积神经网络,所述卷积层104、112负责识别在一个或多个输入数据项102中包含或显示的特征。在至少一个实施例中,该卷积神经网络使用一个或多个输入数据项102辅助计算机视觉应用,所述输入数据项例如包含表示颜色通道的多个层的输入图像,以及可用时的其他信息。

[0054] 在至少一个实施例中,卷积神经网络包含多个层,包括用于特征学习104、106、108、110、112、114、116、118的那些层,以及用于分类120、122的那些层。在至少一个实施例中,卷积神经网络将获取包含多个层的输入图像,并在特征学习阶段104、106、108、110、112、114、116、118期间为每个输入图像中的方面、特征或对象分配重要性。在至少一个实施例中,分类阶段将获取关于一个或多个输入图像中的方面、特征或对象的信息,并使用经训练的神经网络对所述一个或多个输入图像中的那些方面、特征或对象进行分类。

[0055] 在至少一个实施例中,系统训练卷积神经网络,以在特征学习阶段104、106、108、110、112、114、116、118期间生成包含关于输入图像中的方面、特征或对象的信息的特征图106、110、114、118。在至少一个实施例中,特征学习阶段104、106、108、110、112、114、116、118包含多个步骤,其中卷积层104、112被应用于二维数据矩阵,例如特征图。

[0056] 在至少一个实施例中,一个或多个卷积层形成特征学习阶段104、106、108、110、112、114、116、118。在至少一个实施例中,一个或多个卷积层将包含一组滤波器(也称为内核、特征提取器和矩阵),其中每个滤波器应用于图像中的数据,以获得图像中的每一层的图像的宽度和高度。在至少一个实施例中,卷积层104、112将表示数据层的多个特征图106、114作为输入,例如RGB图像中的红色层、绿色层和蓝色层。在至少一个实施例中,特征图106、114是与图像相关联的值的二维矩阵,其中每个值可以表示一个或多个输入图像102的方面。在至少一个实施例中,在特征图中表示的一个或多个输入图像102的方面可以是与存在于图像102中的位置或图像的层中的方面的可能性相关联的概率。

[0057] 在至少一个实施例中,滤波器是小的矩阵。在至少一个实施例中,滤波器可用于识别图像的特征,以及执行模糊、锐化、压印、边缘检测或其他与图像相关的过滤操作。在至少一个实施例中,滤波器或内核将通过卷积运算被应用于图像,所述卷积运算可以包括在输入图像的所有维度上执行点积运算。在至少一个实施例中,卷积层104、112可以包含一个或多个滤波器。

[0058] 在至少一个实施例中,卷积层104、112可以包含逐深度卷积阶段和逐点卷积阶段。在至少一个实施例中,逐深度卷积将在输入图像102或特征图106、114的宽度和高度上应用一个或多个滤波器。在至少一个实施例中,逐深度卷积将在输入图像102或特征图106、114的宽度和高度的子集上应用一个或多个滤波器或内核。在至少一个实施例中,一个或多个滤波器或内核的应用可用于识别图像102或特征图106、114中的特征。在至少一个实施例中,从逐深度卷积输出的一组特征图将包含关于图像102或特征图106、114的输入集中的方面、特征或对象的信息。在至少一个实施例中,卷积层中的逐点卷积运算将在每像素的基础上把卷积运算应用于通过逐深度卷积生成的每个特征图。在至少一个实施例中,逐点卷积将组合包含在由逐深度卷积生成的每个特征图中的信息,如下文进一步描述的。

[0059] 在至少一个实施例中,卷积层104、112将生成一个或多个特征图106、114。在至少一个实施例中,一个或多个池化层108、116将在卷积层104、112之后减小特征图106、

114的大小。在至少一个实施例中,一个或更多个池化层108、116减小特征图大小,以便降低后续卷积层104、112或卷积神经网络中的其他操作120中的计算功率要求。

[0060] 在至少一个实施例中,一个或更多个池化层108、116可以是最大池化层或平均池化层。在至少一个实施例中,最大池化层从已经由卷积层104、112中的内核处理的输入图像或特征图的一部分返回最大值。在至少一个实施例中,平均池化层从已经由卷积层104、112中的内核处理的输入图像或特征图的一部分返回所有值的平均值。

[0061] 在至少一个实施例中,在预定数量的卷积层104、112和随后的池化层108、116已经从输入图像或特征图中提取了方面、特征或对象之后,分类阶段120被执行以生成包含每个分类的相应概率的输出122。在至少一个实施例中,全连接层120可用于学习由特征学习阶段104、106、108、110、112、114、116、118识别的高级特征的非线性组合。至少在一个实施例中,全连接层120可以包含几个步骤。在至少一个实施例中,全连接层120可以将特征图的输入集平坦化为单维向量。在至少一个实施例中,附加步骤可以在120处被执行。在至少一个实施例中,数据的平坦化表示被馈送到前馈神经网络中,如下所述,并且反向传播被用于通过迭代训练来训练所述神经网络。在至少一个实施例中,在预定数量的迭代或时期之后,前馈神经网络能够区分在早期卷积层104、112期间在滤波器或内核中最初指定的特征。在至少一个实施例中,使用Softmax分类技术将输出进行分类。

[0062] 图2A示出了输入数据项(例如图像)中的竖直线特征@102@02。在至少一个实施例中,卷积层可以将一个或更多个图像作为输入。在至少一个实施例中,一个或更多个图像可以包含要由卷积层识别的方面、特征或对象@102@02。在至少一个实施例中,竖直线@102@02可以是可以由卷积层中的滤波器或内核识别的方面、特征或对象。在至少一个实施例中,卷积层中的一个或更多个滤波器可以负责识别单个特征,例如输入图像中的竖直线@102@02。

[0063] 图2B示出了输入数据项(例如图像)中的水平线特征@102@04。在至少一个实施例中,卷积层可以将一个或更多个图像作为输入,并且一个或更多个图像可以包含将由卷积层在特征图中识别或强调的方面、特征或对象@102@04。在至少一个实施例中,水平线@102@04可以是可以由卷积层中的滤波器或内核在特征图中识别、提取或强调的方面、特征或对象。在至少一个实施例中,卷积层中的一个或更多个滤波器可以负责识别单个特征,例如输入图像中的水平线@102@04。

[0064] 图2C示出了输入数据项中的多尺度特征@102@06,所述输入数据项例如可变大小的对象。在至少一个实施例中,卷积层可以将一个或更多个图像作为输入,并且一个或更多个图像可以包含将由包含一个或更多个滤波器或内核的卷积层在特征图中识别或强调的可变大小的一个或更多个方面、特征或对象@102@06。在至少一个实施例中,多尺度特征@102@06可以包括可以由卷积层中的滤波器或内核在特征图中识别、提取或强调的一个或更多个方面、特征或对象。在至少一个实施例中,卷积层中的一个或更多个滤波器可以负责识别输入图像中多个尺度@102@06的单个特征。在至少一个实施例中,可以针对特征图或其他输入数据项中的每个多尺度特征@102@06的每个位置对齐多个滤波器。在至少一个实施例中,滤波器可以通过填充或线性缩放各种尺寸的滤波器来对齐。

[0065] 图3示出了卷积神经网络的卷积层中的逐深度卷积。在至少一个实施例中,输入数据项302(例如图像)可以包含多个层。在至少一个实施例中,输入数据图像302中的层可以包含颜色信息,例如RGB输入图像302中的红色层、绿色层和蓝色层。在至少一个实施例中,

卷积神经网络的卷积层中的逐深度卷积可以首先将每个层304分离成各个层306、308、310，其表示为二维矩阵或如上所述的特征图。在至少一个实施例中，每个分离的层或特征图306、308、310可以表示关于输入数据项或图像302的信息的子集。

[0066] 在至少一个实施例中，每个分离的层或特征图306、308、310接下来将具有通过卷积应用的滤波器或内核。在至少一个实施例中，分离的层或特征图306、308、310可以是维度为 $K \times K$ 的二维矩阵。在至少一个实施例中，输入数据项或图像302可以具有 C_{in} 层或特征图306、308、310。在至少一个实施例中，卷积神经网络的卷积层中的逐深度卷积可以输出 C_{out} 层或特征图322。

[0067] 在至少一个实施例中，传统卷积层或可分离卷积层将在卷积步骤312期间跨所有层或特征图306、308、310共享单个 $K \times K$ 滤波器。在至少一个实施例中，空间自适应可分离卷积层可以在卷积步骤312期间向不同的层或特征图306、308、310应用不同大小的多个滤波器或内核，不限于 $K \times K$ ，如下所述。在至少一个实施例中，空间自适应可分离卷积层可以在卷积步骤312期间将大小为 $K \times K$ 的多个滤波器或内核应用于不同的层或特征图306、308、310，其中每个滤波器对应于不同的特征，如下面所描述的。

[0068] 在至少一个实施例中，每个 $K \times K$ 滤波器或每个较小维度的滤波器在卷积步骤312被应用于每个输入层或特征图306、308、310。在至少一个实施例中，卷积步骤312可包含计算输入矩阵和滤波器之间的点积。在至少一个实施例中，输出特征图314、316、318可以是维度为 $K \times K$ 的二维矩阵，其表示滤波器或内核与输入层或特征图306、308、310之间的点积。至少在一个实施例中，输出特征图314、316、318可以是可变维度的二维矩阵，其表示尺寸小于 $K \times K$ 的滤波器与二维矩阵中的值的子集（其表示输入层或特征图306、308、310）之间的点积。在至少一个实施例中，输出特征图314、316、318可以被组合320成多层输出322，从而可以应用逐点卷积，如下所述。

[0069] 图4示出了卷积神经网络的卷积层中的逐点卷积。在至少一个实施例中，输入特征图402可以被生成为来自卷积层中的逐深度卷积运算的输出，如上所述。在至少一个实施例中，对于每一层，输入特征图402可以是二维矩阵，其包含关于在卷积层中的逐深度卷积运算期间已经应用滤波器或内核之后图像中的层的信息。在至少一个实施例中，输入特征图402在可分离卷积层中可以具有 $K \times K$ 维度。在至少一个实施例中，输入特征图402在空间自适应可分离卷积层中可以是可变维度的。

[0070] 在至少一个实施例中，卷积层中的逐点卷积运算404将在每像素或每矩阵元素的基础上在由逐深度卷积生成的输入特征图402上应用卷积运算。在至少一个实施例中，逐点卷积404将组合包含在由逐深度卷积生成的输入特征图402内的空间信息，并产生输出特征图406。

[0071] 图5示出了卷积神经网络中的可分离卷积层。在至少一个实施例中，可分离卷积层是分解卷积层的一种形式，其由如上所述的逐深度卷积508和同样如上所述的 1×1 逐点卷积522组成。在至少一个实施例中，分解卷积层同时跨输入通道502、504、506执行卷积。

[0072] 在至少一个实施例中，逐深度卷积508对每输入特征图应用一个或更多个 $K \times K$ 滤波器或内核以获得 $K \times K$ 特征图 $U_{p,q}$ 中的空间信息，其中 $p, q \in [1, \dots, C_{in}]$ 510、512、514、516、518、520。在至少一个实施例中， 1×1 逐点卷积522组合 $U_{p,q}$ 510、512、514、516、518、520中的空间信息以生成输出特征图524、526、528。在至少一个实施例中，可以在逐深度卷积508期

间对每个输入图像层或特征图502、504、506应用多个滤波器。图5说明了逐深度卷积508,其中 $K \times K$ 滤波器或内核(其中,要应用的滤波器或内核的数量 M 等于2)。

[0073] 在至少一个实施例中,如图5所示的可分离卷积层接收输入图像层或特征图502、504、506作为输入 C_{in} ,其中每个输入图像层或特征图502、504、506具有宽度 W 和高度 H 。在至少一个实施例中,激活大小可以描述要在可分离卷积层中执行的大量计算。在至少一个实施例中,可分离卷积层的激活大小或计算是:

$$[0074] \quad W \times H \times K \times K \times C_{in} + W \times H \times C_{in} \times C_{out}$$

[0075] 在至少一个实施例中,如本文所述的卷积神经网络可以尝试使用反向传播来学习应用于卷积层中的输入数据502、504、506的滤波器或内核的值。在至少一个实施例中,包含描述权重510、512、514、516、518、520、524、526、528的矩阵(例如特征图)的卷积神经网络中的每一层或卷积层可以被认为是可学习层,或包含可学习元素的层。在至少一个实施例中,可以学习的层中的滤波器或内核的元素计数被认为是层中的所述滤波器或内核的参数。在至少一个实施例中,对于可分离卷积层,参数计数或可以学习的元素是:

$$[0076] \quad K \times K \times C_{in} + C_{in} \times C_{out}$$

[0077] 在至少一个实施例中,可分离卷积层可以包含多于一个 $K \times K$ 滤波器或内核,以被应用于每个输入特征图或图像层502、504、506。在至少一个实施例中,应用于每个输入层或特征图502、504、506的滤波器的总数是 M 。图5示出了根据至少一个实施例的具有 $M=2$ 的可分离卷积层。在至少一个实施例中,包含多于一个 $K \times K$ 滤波器或内核的可分离卷积层的激活大小是:

$$[0078] \quad W \times H \times K \times K \times M \times C_{in} + W \times H \times C_{in} \times M \times C_{out}$$

[0079] 在至少一个实施例中,包含多于一个 $K \times K$ 滤波器或内核的可分离卷积层的激活大小为:

$$[0080] \quad K \times K \times M \times C_{in} + C_{in} \times M \times C_{out}$$

[0081] 图6示出了用于在卷积神经网络的空间自适应可分离卷积层中执行逐深度卷积的架构。在至少一个实施例中,如本文所述,空间自适应可分离卷积层使卷积神经网络中的神经元能够在输入数据项或图像602中的不同位置处自适应地进行调整。在至少一个实施例中,输入值602可以是图像层或特征图,如上文结合示例卷积神经网络所述的。

[0082] 在至少一个实施例中,“拆分”操作从输入图像层或特征图602生成多个路径或矩阵604、606,其中用于创建中间特征图或矩阵604、606的不同空间滤波器可具有可变的内核大小,或者可具有统一的内核大小,但被构造为识别不同的特征。在至少一个实施例中,要应用的滤波器的总数是 M 。在至少一个实施例中,逐深度卷积层将对每输入图像层或特征图602应用多个滤波器或空间变换604、606。在至少一个实施例中,多个滤波器或空间变换604、606可以是可变尺寸或统一尺寸的。在至少一个实施例中, M 个不同的空间变换 $F_{mc}: X_c \rightarrow U_{mc} \in \mathbb{R}^{W \times H}$ 604、606,其中 $c \in [1, \dots, C_{in}]$ 被应用于输入特征图 $X \in \mathbb{R}^{W \times H \times C_{in}}$ 602。在至少一个实施例中, F_{mc} 604、606是应用于第 c 个输入特征图 X_c 的第 m 个滤波器。

[0083] 在至少一个实施例中,“竞争”操作将生成紧凑特征描述符624,其用关于输入602中的多尺度特征的更多信息进行编码。在至少一个实施例中,“竞争”操作将在由各个滤波器 F_{mc} 识别的每个空间位置处跨不同分支或滤波器通道604、606应用软注意力。在至少一个实施例中,对每个像素处的特定于每个滤波器通道的特征图应用Softmax运算608,以获得

选择权重图 $S_m \in \mathbb{R}^{W \times H \times C_{in}}$ 610、612, 其中 $m=1, \dots, M$ 。在至少一个实施例中, 在每个选择权重图 S_m 610、612和特征图 U_{mc} 604、606之间执行点积以生成所选信息 D_{mc} 618、620。

[0084] 在至少一个实施例中, 逐元素加法运算跨滤波器通道 F_{mc} 聚合所选信息 D_{mc} 以生成紧凑特征描述符 $V_c \in \mathbb{R}^{W \times H}$ 624, 其中 $c \in [1, \dots, C_{in}]$ 。在至少一个实施例中, 选择权重图值 S_{mc}^{ij} 610、612可以从像素值 U_{mc}^{ij} 604、606 (其中 i 是特定行, j 是特定列, c 是特定输入 602, 而 m 是特定的滤波器通道) 被计算。在至少一个实施例中, Softmax可被用于确定选择权重图值 610、612, 其可被计算为:

$$[0085] \quad S_{mc}^{ij} = \frac{e^{U_{mc}^{ij}}}{\sum_{m=1}^M e^{U_{mc}^{ij}}}$$

[0086] 在至少一个实施例中, 紧凑特征描述符或最终的空间自适应特征图 $V_c \in \mathbb{R}^{W \times H}$ 624, 其中 $c \in [1, \dots, C_{in}]$, 可以计算为:

$$[0087] \quad V_c = \sum_{m=1}^M D_m$$

[0088] 或

$$[0089] \quad V_c = \sum_{m=1}^M U_m \cdot S_m$$

[0090] 图7示出了卷积神经网络中的空间自适应可分离卷积层。在至少一个实施例中, 在每个输入图像层或特征图 X_c 702、704、706上, 其中 $c \in [1, \dots, C_{in}]$, 逐深度卷积 728 如结合图6所述地被执行。在至少一个实施例中, 滤波器或内核 $F_{mc}: X_c \rightarrow U_{mc} \in \mathbb{R}^{W \times H}$ 被应用于输入特征图 $X_c \in \mathbb{R}^{W \times H}$ 702、704、706, 其中 $c \in [1, \dots, C_{in}]$ 。在至少一个实施例中, F_{mc} 是应用于第 c 个输入特征图 X_c 的第 m 个滤波器, 以便生成 U_{mc} 708、710、712、714、716、718。

[0091] 在至少一个实施例中, 对于每个滤波器通道 U_{mc} 708、710、712、714、716、718, 在逐深度卷积之后应用softmax函数 720, 如上所述。在至少一个实施例中, 从每个softmax 720 输出和每个滤波器通道 U_{mc} 708、710、712、714、716、718之间的点积 722、724 收集软信息。在至少一个实施例中, 针对每个滤波器通道聚合 726 来自点积 722、724 的软信息输出, 如上所述, 以便生成紧凑的特征描述符或最终的空间自适应特征图 $V_c \in \mathbb{R}^{W \times H}$ 730、732、734, 其中 $c \in [1, \dots, C_{in}]$ 。

[0092] 在至少一个实施例中, 如上所述, 逐点卷积 736 在每个紧凑特征描述符 V_c 730、732、734 上执行以生成输出特征图 738、740、742。在至少一个实施例中, 当在逐深度卷积 728 期间应用的“拆分”数量或滤波器的数量增加时, 逐点卷积 736 输入值或紧凑特征描述符 730、732、734 不改变。在至少一个实施例中, 因为在逐深度卷积 728 中执行的每个“拆分”和“竞争”操作中的每个像素独立于每个其他像素, 所以每个输入特征图 702、704、706 的每个逐深度卷积 728 可以并行地被执行。在至少一个实施例中, 每个输入特征图 702、704、706 的每个并行逐深度卷积操作 728 可以如本文所述地在图形处理单元 (GPU) 或任何其他并行处理单元 (PPU) 上执行。

[0093] 在至少一个实施例中, 如图7所示的空间自适应可分离卷积层接收图像层或特征图 702、704、706 作为输入 C_{in} , 其中每个输入图像层或特征图 702、704、706 具有宽度 W 和高度 H 。在至少一个实施例中, 激活尺寸可以描述要在可分离卷积层中执行的计算的数目。在至

少一个实施例中,空间自适应可分离卷积层的激活尺寸或计算是:

$$[0094] \quad W \times H \times C_{in} \times (\sum_{m=1}^M K_m \times K_m) + W \times H \times C_{in} \times M \times C_{out}$$

[0095] 在至少一个实施例中,如本文所述,卷积神经网络可以尝试使用反向传播来学习如应用于空间自适应可分离卷积层中的输入数据702、704、706的滤波器或内核的值。在至少一个实施例中,卷积神经网络中的每一层或卷积层,其包含描述权重702、704、706的矩阵(例如特征图),可以被认为是可学习层或包含可学习元素的层。在至少一个实施例中,可以学习的层中的滤波器或内核的元素的数目被认为是层中的所述滤波器或内核的参数。在至少一个实施例中,对于本文所述的空间自适应可分离卷积层,参数计数或可学习的元素描述如下,其中 $\sum_{m=1}^M K_m \times K_m \ll C_{out} \times M$:

$$[0096] \quad (\sum_{m=1}^M K_m \times K_m) \times C_{in} + C_{in} \times M \times C_{out}$$

[0097] 图8示出了由卷积神经网络中的空间自适应可分离卷积层生成输出特征图的过程。在至少一个实施例中,空间自适应可分离卷积层通过执行逐深度卷积816开始802。在至少一个实施例中,逐深度卷积816中的步骤可以在图形处理单元(GPU)或任何其他并行处理单元(PPU)上并行执行,如本文所述。

[0098] 在至少一个实施例中,如上所述的空间滤波器被应用804于输入特征图。在至少一个实施例中,Softmax函数被应用806于通过将空间滤波器应用于输入特征图804而生成的空间信息。在至少一个实施例中,基于来自Softmax函数806的输出和通过将空间滤波器应用于输入特征图804而生成的空间信息来跨滤波器通道生成808所选信息。在至少一个实施例中,跨滤波器通道生成808的所选信息被聚合810成紧凑特征描述符。在至少一个实施例中,从跨滤波器通道生成808的所选信息聚合810的紧凑特征信息通过逐点卷积812进行组合,以便生成输出特征图,从而完成814通过空间自适应可分离卷积层生成输出特征图的过程。

[0099] 推理和训练逻辑

[0100] 图9A示出了用于执行与一个或更多个实施例相关联的推理和/或训练操作的推理和/或训练逻辑915。下面结合图9A和/或图9B提供关于推理和/或训练逻辑915的细节。

[0101] 在至少一个实施例中,推理和/或训练逻辑915可以包括但不限于代码和/或数据存储901,用于存储前向和/或输出权重和/或输入/输出数据,和/或在一个或更多个实施例的方面中配置被训练为和/或用于推理的神经网络的神经元或层的其他参数。在至少一个实施例中,训练逻辑915可以包括或耦合到用于存储图形代码或其他软件以控制时序和/或顺序的代码和/或数据存储901,其中权重和/或其他参数信息被加载以配置逻辑,包括整数和/或浮点单元(统称为算术逻辑单元(ALU))。在至少一个实施例中,代码(诸如图代码)基于所述代码所对应的神经网络的架构将权重或其他参数信息加载到处理器ALU中。在至少一个实施例中,代码和/或数据存储901存储在使用一个或更多个实施例的方面训练和/或推理期间的输入/输出数据和/或权重参数的前向传播期间结合一个或更多个实施例训练或使用的神经网络的每个层的权重参数和/或输入/输出数据。在至少一个实施例中,代码和/或数据存储901的任何部分都可以包括在其他片上或片外数据存储内,包括处理器的L1、L2或L3高速缓存或系统存储器。

[0102] 在至少一个实施例中,代码和/或数据存储901的任何部分可以在一个或更多个处

理器或其他硬件逻辑设备或电路的内部或外部。在至少一个实施例中,代码和/或数据存储901可以是高速缓存存储器、动态随机可寻址存储器(“DRAM”)、静态随机可寻址存储器(“SRAM”)、非易失性存储器(例如闪存)或其他存储。在至少一个实施例中,对代码和/或数据存储901是在处理器的内部还是外部的选择,例如,或者由DRAM、SRAM、闪存或某种其他存储类型组成,可以取决于存储片上或片外的可用存储空间,正在执行训练和/或推理功能的延迟要求,在神经网络的推理和/或训练中使用的数据的批大小或这些因素的某种组合。

[0103] 在至少一个实施例中,推理和/或训练逻辑915可以包括但不限于代码和/或数据存储905,以存储与在一个或多个实施例的方面中被训练为和/或用于推理的神经网络的神经元或层相对应的反向和/或输出权重和/或输入/输出数据神经网络。在至少一个实施例中,在使用一个或多个实施例的方面训练和/或推理期间,代码和/或数据存储905存储在输入/输出数据和/或权重参数的反向传播期间结合一个或多个实施例训练或使用的神经网络的每个层的权重参数和/或输入/输出数据。在至少一个实施例中,训练逻辑915可以包括或耦合到用于存储图代码或其他软件以控制时序和/或顺序的代码和/或数据存储905,其中权重和/或其他参数信息被加载以配置逻辑,该逻辑包括整数和/或浮点单元(统称为算术逻辑单元(ALU))。在至少一个实施例中,代码(诸如图代码)基于所述代码所对应的神经网络的架构将权重或其他参数信息加载到处理器ALU中。在至少一个实施例中,代码和/或数据存储905的任何部分可以与其他片上或片外数据存储一起包括,包括处理器的L1、L2或L3高速缓存或系统存储器。在至少一个实施例中,代码和/或数据存储905的任何部分可以在一个或多个处理器或其他硬件逻辑设备或电路上的内部或外部。在至少一个实施例中,代码和/或数据存储905可以是高速缓存存储器、DRAM、SRAM、非易失性存储器(例如闪存)或其他存储。在至少一个实施例中,代码和/或数据存储905是在处理器的内部还是外部的选择,例如,是由DRAM、SRAM、闪存还是其他某种存储类型组成,取决于可用存储是片上还是片外,正在执行的训练和/或推理功能的延迟要求,在神经网络的推理和/或训练中使用的数据批大小或这些因素的某种组合。

[0104] 在至少一个实施例中,代码和/或数据存储901以及代码和/或数据存储905可以是分开的存储结构。在至少一个实施例中,代码和/或数据存储901以及代码和/或数据存储905可以是相同的存储结构。在至少一个实施例中,代码和/或数据存储901以及代码和/或数据存储905可以是部分相同的存储结构和部分不同的存储结构。在至少一个实施例中,代码和/或数据存储901以及代码和/或数据存储905的任何部分可以与其他片上或片外数据存储包括在一起,包括处理器的L1、L2或L3高速缓存或系统存储器。

[0105] 在至少一个实施例中,推理和/或训练逻辑915可以包括但不限于一个或多个算术逻辑单元(“ALU”)910(包括整数和/或浮点单元),用于至少部分地基于训练和/或推理代码(例如,图代码)或由其指示来执行逻辑和/或数学运算,其结果可能会产生存储在激活存储920中的激活(例如,来自神经网络内部的层或神经元的输出值),其是存储在代码和/或数据存储901和/或代码和/或数据存储905中的输入/输出和/或权重参数数据的函数。在至少一个实施例中,激活响应于执行指令或其他代码,由ALU 910执行的线性代数和/或基于矩阵的数学生成在激活存储920中存储的激活,其中存储在代码和/或数据存储905中和/或代码和/或数据存储901中的权重值用作具有其他值的操作数,例如偏置值、梯度信息、动量值或其他参数或超参数,可以将任何或所有这些存储在代码和/或数据存储905或代码和/

或数据存储901或其他片上或片外存储中。

[0106] 在至少一个实施例中,一个或多个处理器或其他硬件逻辑设备或电路中包括一个或多个ALU 910,而在另一实施例中,一个或多个ALU 910可以在处理器或其他硬件逻辑设备或使用它们(例如协处理器)的电路外。在至少一个实施例中,可以将一个或多个ALU 910包括在处理器的执行单元之内,或者以其他方式包括在由处理器的执行单元可访问的ALU组中,该处理器的执行单元可以在同一处理器内或者分布在不同类型的不同处理器之间(例如,中央处理单元、图形处理单元、固定功能单元等)。在至少一个实施例中,数据存储901、代码和/或数据存储905以及激活存储920可以在相同的处理器或其他硬件逻辑设备或电路上,而在另一实施例中,它们可以在不同的处理器或其他硬件逻辑设备或电路或相同和不同处理器或其他硬件逻辑设备或电路的某种组合中。在至少一个实施例中,激活存储920的任何部分可以与其他片上或片外数据存储包括在一起,包括处理器的L1、L2或L3高速缓存或系统存储器。此外,推理和/或训练代码可以与处理器或其他硬件逻辑或电路可访问的其他代码一起存储,并可以使用处理器的提取、解码、调度、执行、退出和/或其他逻辑电路来提取和/或处理。

[0107] 在至少一个实施例中,激活存储920可以是高速缓存存储器、DRAM、SRAM、非易失性存储器(例如,闪存)或其他存储。在至少一个实施例中,激活存储920可以完全地或部分地在一个或多个处理器或其他逻辑电路内部或外部。在至少一个实施例中,可以取决于片上或片外可用的存储,进行训练和/或推理功能的延迟要求,在推理和/或训练神经网络中使用的数据的批量大小或这些因素的某种组合,选择激活存储920是处理器的内部还是外部,例如,或者包含DRAM、SRAM、闪存或其他存储类型。在至少一个实施例中,图9A中所示的推理和/或训练逻辑915可以与专用集成电路(“ASIC”)结合使用,例如来自Google的TensorFlow®处理单元、来自Graphcore™的推理处理单元(IPU)或来自Intel公司的Nervana®(例如“Lake Crest”)处理器。在至少一个实施例中,图9A所示的推理和/或训练逻辑915可与中央处理单元(“CPU”)硬件,图形处理单元(“GPU”)硬件或其他硬件(例如现场可编程门阵列(“FPGA”))结合使用。

[0108] 图9B示出了根据至少一个实施例的推理和/或训练逻辑915。在至少一个实施例中,推理和/或训练逻辑915可以包括但不限于硬件逻辑,其中计算资源被专用或以其他方式唯一地连同对应于神经网络内的一层或更多层神经元的权重值或其他信息一起使用。在至少一个实施例中,图9B中所示的推理和/或训练逻辑915可以与专用集成电路(ASIC)结合使用,例如来自Google的TensorFlow®处理单元,来自Graphcore™的推理处理单元(IPU)或来自Intel Corp的Nervana®(例如“Lake Crest”)处理器。在至少一个实施例中,图9B中所示的推理和/或训练逻辑915可以与中央处理单元(CPU)硬件、图形处理单元(GPU)硬件或其他硬件(例如现场可编程门阵列(FPGA))结合使用。在至少一个实施例中,推理和/或训练逻辑915包括但不限于代码和/或数据存储901以及代码和/或数据存储905,其可以用于存储代码(例如,图代码)、权重值和/或其他信息,包括偏置值、梯度信息、动量值和/或其他参数或超参数信息。在图9B中所示的至少一个实施例中,代码和/或数据存储901以及代码和/或数据存储905中的每一个都分别与专用计算资源(例如计算硬件902和计算硬件906)相关联。在至少一个实施例中,计算硬件902和计算硬件906中的每一个包括一个或多个

ALU,这些ALU仅分别对存储在代码和/或数据存储901和代码和/或数据存储905中的信息执行数学函数(例如线性代数函数),执行函数的结果被存储在激活存储920中。

[0109] 在至少一个实施例中,代码和/或数据存储901和905以及相应的计算硬件902和906中的每一个分别对应于神经网络的不同层,使得从代码和/或数据存储901和计算硬件902的一个“存储/计算对901/902”得到的激活提供作为代码和/或数据存储905和计算硬件906的下一个“存储/计算对905/906”的输入,以便反映神经网络的概念组织。在至少一个实施例中,每个存储/计算对901/902和905/906可以对应于一个以上的神经网络层。在至少一个实施例中,在推理和/或训练逻辑915中可以包括在存储计算对901/902和905/906之后或与之并行的附加存储/计算对(未示出)。

[0110] 神经网络训练和部署

[0111] 图10示出了根据至少一个实施例的深度神经网络的训练和部署。在至少一个实施例中,使用训练数据集1002来训练未经训练的神经网络1006。在至少一个实施例中,训练框架1004是PyTorch框架,而在其他实施例中,训练框架1004是TensorFlow,Boost,Caffe,Microsoft Cognitive Toolkit/CNTK,MXNet,Chainer,Keras,Deeplearning4j或其他训练框架。在至少一个实施例中,训练框架1004训练未经训练的神经网络1006,并使它能够使用本文所述的处理资源来训练,以生成经训练的神经网络1008。在至少一个实施例中,权重可以被随机选择或通过使用深度信念网络预训练。在至少一个实施例中,可以有监督、部分有监督或无监督的方式执行训练。

[0112] 在至少一个实施例中,使用有监督学习来训练未经训练的神经网络1006,其中训练数据集1002包括与用于输入的期望输出配对的输入,或者其中训练数据集1002包括具有已知输出的输入和神经网络1006是手动层次的输出。在至少一个实施例中,以有监督的方式来训练未经训练的神经网络1006,并且处理来自训练数据集1002的输入,并将结果输出与一组期望或想要的输出进行比较。在至少一个实施例中,然后通过未经训练的神经网络1006将误差传播回去。在至少一个实施例中,训练框架1004调整控制未经训练的神经网络1006的权重。在至少一个实施例中,训练框架1004包括用于监视未经训练的神经网络1006向模型(例如,经训练的神经网络1008)收敛的程度的工具,适于基于已知输入数据(例如新数据1012)生成正确答案(例如结果1014)的模型。在至少一个实施例中,训练框架1004反复训练未经训练的神经网络1006,同时调整权重以使用损失函数和调整算法(例如随机梯度下降)来改善未经训练的神经网络1006的输出。在至少一个实施例中,训练框架1004训练未经训练的神经网络1006,直到未经训练的神经网络1006达到期望的精度为止。在至少一个实施例中,然后可以部署经训练的神经网络1008以实现任何数量的机器学习操作。

[0113] 在至少一个实施例中,使用无监督学习来训练未经训练的神经网络1006,其中未经训练的神经网络1006尝试使用未标记的数据来训练自己。在至少一个实施例中,无监督学习训练数据集1002将包括输入数据,而没有任何关联的输出数据或“地面实况”数据。在至少一个实施例中,未经训练的神经网络1006可以学习训练数据集1002内的分组,并且可以确定各个输入如何与未经训练的数据集1002相关。在至少一个实施例中,可以使用无监督训练来生成自组织图,其是能够执行对减少新数据1012的维度有用的操作的一种类型的经训练的神经网络1008。在至少一个实施例中,无监督训练也可以用于执行异常检测,这允许识别新数据集1012中偏离新数据集1012的正常模式的数据点。

[0114] 在至少一个实施例中,可以使用半监督学习,这是一种技术,其中在训练数据集1002中包括标记数据和未标记数据的混合。在至少一个实施例中,训练框架1004可以用于例如通过转移的学习技术来执行递增学习。在至少一个实施例中,递增学习使得经训练的神经网络1008能够适应新数据1012,而不会忘记在初始训练期间注入到网络内的知识。

[0115] 数据中心

[0116] 图11示出了可以使用至少一个实施例的示例数据中心1100。在至少一个实施例中,数据中心1100包括数据中心基础设施层1110、框架层1120、软件层1130和应用程序层1140。

[0117] 在至少一个实施例中,如图11所示,数据中心基础设施层1110可以包括资源协调器1112、分组计算资源1114和节点计算资源(“节点C.R.”) 1116(1)-1116(N),其中“N”代表任意整个正整数。在至少一个实施例中,节点C.R. 1116(1)-1116(N)可以包括但不限于任何数量的中央处理单元(“CPU”)或其他处理器(包括加速器、现场可编程门阵列(FPGA)、图形处理器等),存储器设备(例如动态只读存储器),存储设备(例如固态或磁盘驱动器),网络输入/输出(“NW I/O”)设备,网络交换机,虚拟机(“VM”),电源模块和冷却模块等。在至少一个实施例中,节点C.R. 1116(1)-1116(N)中的一个或多个节点C.R.可以是具有一个或多个上述计算资源的服务器。

[0118] 在至少一个实施例中,分组计算资源1114可以包括容纳在一个或多个机架内的节点C.R.的单独分组(未示出),或者容纳在各个地理位置的数据中心内的许多机架(也未示出)。分组的计算资源1114内的节点C.R.的单独分组可以包括可以被配置或分配为支持一个或多个工作负载的分组的计算、网络、存储器或存储资源。在至少一个实施例中,可以将包括CPU或处理器的几个节点C.R.分组在一个或多个机架内,以提供计算资源来支持一个或多个工作负载。在至少一个实施例中,一个或多个机架还可以包括任何数量的电源模块、冷却模块和网络交换机,以任意组合。

[0119] 在至少一个实施例中,资源协调器1112可以配置或以其他方式控制一个或多个节点C.R. 1116(1)-1116(N)和/或分组的计算资源1114。在至少一个实施例中,资源协调器1112可以包括用于数据中心1100的软件设计基础结构(“SDI”)管理实体。在至少一个实施例中,资源协调器可以包括硬件、软件或其某种组合。

[0120] 在至少一个实施例中,如图11所示,框架层1120包括作业调度器1132、配置管理器1134、资源管理器1136和分布式文件系统1138。在至少一个实施例中,框架层1120可以包括支持软件层1130的软件1132和/或应用程序层1140的一个或多个应用程序1142的框架。在至少一个实施例中,软件1132或应用程序1142可以分别包括基于Web的服务软件或应用程序,例如由Amazon Web Services, Google Cloud和Microsoft Azure提供的服务或应用程序。在至少一个实施例中,框架层1120可以是但不限于一种免费和开放源软件网络应用程序框架,例如可以利用分布式文件系统1138来进行大范围数据处理(例如“大数据”)的Apache Spark™(以下称为“Spark”)。在至少一个实施例中,作业调度器1132可以包括Spark驱动器,以促进对数据中心1100的各个层所支持的工作负载进行调度。在至少一个实施例中,配置管理器1134可以能够配置不同的层,例如软件层1130和包括Spark和用于支持大规模数据处理的分布式文件系统1138的框架层1120。在至少一个实施例中,资源管理器1136能够管理映射到或分配用于支持分布式文件系统1138和作业调度器1132的集群或分组计

算资源。在至少一个实施例中,集群或分组计算资源可以包括数据中心基础设施层1110上的分组计算资源1114。在至少一个实施例中,资源管理器1136可以与资源协调器1112协调以管理这些映射的或分配的计算资源。

[0121] 在至少一个实施例中,包括在软件层1130中的软件1132可以包括由节点C.R.1116(1)-1116(N)的至少一部分,分组的计算资源1114和/或框架层1120的分布式文件系统1138使用的软件。一种或更多种类型的软件可以包括但不限于Internet网页搜索软件、电子邮件病毒扫描软件、数据库软件和流视频内容软件。

[0122] 在至少一个实施例中,应用程序层1140中包括的一个或更多个应用程序1142可以包括由节点C.R.1116(1)-1116(N)的至少一部分、分组计算资源1114和/或框架层1120的分布式文件系统1138使用的一种或更多种类型的应用程序。一种或更多种类型的应用程序可以包括但不限于任何数量的基因组学应用程序、认知计算和机器学习应用程序,包括训练或推理软件,机器学习框架软件(例如PyTorch、TensorFlow、Caffe等)或其他与一个或更多个实施例结合使用的机器学习应用程序。

[0123] 在至少一个实施例中,配置管理器1134、资源管理器1136和资源协调器1112中的任何一个可以基于以任何技术上可行的方式获取的任何数量和类型的数据来实现任何数量和类型的自我修改动作。在至少一个实施例中,自我修改动作可以减轻数据中心1100的数据中心操作员做出可能不好的配置决定并且可以避免数据中心的未充分利用和/或执行差的部分。

[0124] 在至少一个实施例中,数据中心1100可以包括工具、服务、软件或其他资源,以根据本文所述的一个或更多个实施例来训练一个或更多个机器学习模型或者使用一个或更多个机器学习模型来预测或推理信息。例如,在至少一个实施例中,可以通过使用上文关于数据中心1100描述的软件和计算资源,根据神经网络架构通过计算权重参数来训练机器学习模型。在至少一个实施例中,通过使用通过本文所述的一种或更多种训练技术计算出的权重参数,可以使用上面与关于数据中心1100所描述的资源,使用对应于一个或更多个神经网络的经训练的机器学习模型来推理或预测信息。

[0125] 在至少一个实施例中,数据中心可以使用CPU、专用集成电路(ASIC)、GPU、FPGA或其他硬件来使用上述资源来执行训练和/或推理。此外,上述的一个或更多个软件和/或硬件资源可以配置成一种服务,以允许用户训练或执行信息推理,例如图像识别、语音识别或其他人工智能服务。

[0126] 推理和/或训练逻辑915用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图9A和/或图9B提供关于推理和/或训练逻辑915的细节。在至少一个实施例中,推理和/或训练逻辑915可以在系统图11中使用,用于至少部分地基于使用神经网络训练操作、神经网络函数和/或体系架构,或者本文所述的神经网络用例计算的权重参数推理或预测操作。

[0127] 在至少一个实施例中,空间自适应可分离卷积层7可以和系统图11一起用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构、或本文所述的神经网络用例计算的权重参数来推理和预测操作。

[0128] 自主车辆

[0129] 图12A示出了根据至少一个实施例的自动驾驶车辆1200的示例。在至少一个实施

例中,自动驾驶车辆1200(在本文中可替代地称为“车辆1200”)可以是但不限于客运车辆,例如汽车、卡车、公共汽车和/或可容纳一个或更多个乘客的另一种类型的车辆。在至少一个实施例中,车辆1200可以是用于拖运货物的半牵引车-拖车。在至少一个实施例中,车辆1200可以是飞机、机器人车辆或其他类型的车辆。

[0130] 可以根据由美国运输部下属的国家公路交通安全管理局(“NHTSA”)和汽车工程师学会(“SAE”)“与用于道路机动车辆的驾驶自动化系统有关的术语(Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles)”(例如,于2018年6月15日发布的标准号J3016-201806,于2016年9月30日发布的标准号J3016-201609,以及该版本的以前和将来的版本此标准)定义的自动化级别来描述自动驾驶汽车。在一个或更多个实施例中,车辆1200可能能够根据自动驾驶级别的级别1至级别5中的一个或更多个来进行功能。例如,在至少一个实施例中,根据实施例,车辆1200可能能够进行条件自动化(级别3)、高度自动化(级别4)和/或全自动(级别5)。

[0131] 在至少一个实施例中,车辆1200可以包括但不限于组件,诸如底盘、车身、车轮(例如2、4、6、8、18等)、轮胎、车轴和车辆的其他组件。在至少一个实施例中,车辆1200可以包括但不限于推进系统1250,例如内燃机、混合动力装置、全电动发动机和/或另一种推进系统类型。在至少一个实施例中,推进系统1250可以连接至车辆1200的传动系,其可以包括但不限于变速器,以使得能够对车辆1200进行推进。在至少一个实施例中,可以响应于从油门/加速器1252接收信号以控制推进系统1250。

[0132] 在至少一个实施例中,当推进系统1250正在运行时(例如,当车辆行驶时),转向系统1254(其可以包括但不限于方向盘)用于使车辆1200转向(例如,沿着期望的路径或路线)。在至少一个实施例中,转向系统1254可以从转向致动器1256接收信号。方向盘对于全自动化(级别5)功能可以是可选的。在至少一个实施例中,制动传感器系统1246可以用于响应于从制动致动器1248和/或制动传感器接收到的信号来操作车辆制动器。

[0133] 在至少一个实施例中,控制器1236可以包括但不限于一个或更多个片上系统(“SoC”) (图12A中未示出)和/或图形处理单元(“GPU”)向车辆1200的一个或更多个组件和/或系统提供信号(例如,代表命令)。例如,在至少一个实施例中,控制器1236可以发送信号以通过制动致动器1248操作车辆制动,通过一个或更多个转向致动器1256操作转向系统1254,通过一个或更多个油门(throttle)/加速器1252操作推进系统1250。一个或更多个控制器1236可以包括一个或更多个机载(例如,集成)计算设备(例如,超级计算机),其处理传感器信号并输出操作命令(例如,表示命令的信号)以实现自动驾驶和/或协助驾驶员驾驶车辆1200。在至少一个实施例中,一个或更多个控制器1236可以包括用于自动驾驶功能的第一控制器1236,用于功能安全功能的第二控制器1236,用于人工智能功能(例如计算机视觉)的第三控制器1236,用于信息娱乐功能的第四控制器1236,用于紧急情况下的冗余的第五控制器1236和/或其他控制器。在至少一个实施例中,单个控制器1236可以处理上述功能中的两个或更多个,两个或更多控制器1236可以处理单个功能和/或其任何组合。

[0134] 在至少一个实施例中,一个或更多个控制器1236响应于从一个或更多个传感器(例如,传感器输入)接收到的传感器数据,提供用于控制车辆1200的一个或更多个组件和/或系统的信号。在至少一个实施例中,传感器数据可以从传感器接收,传感器类型例如但不限于一个或更多个全球导航卫星系统(“GNSS”)传感器1258(例如,一个或更多个全球定位

系统传感器)、一个或更多个RADAR传感器1260、一个或更多个超声波传感器1262、一个或更多个LIDAR传感器1264、一个或更多个惯性测量单元(IMU)传感器1266(例如,一个或更多个加速度计、一个或更多个陀螺仪、一个或更多个磁罗盘、一个或更多个磁力计等)、一个或更多个麦克风1296、一个或更多个立体声相机1268、一个或更多个广角相机1270(例如鱼眼相机)、一个或更多个红外相机1272、一个或更多个环绕相机1274(例如,360度相机)、远程相机(图12A中未示出)、中程相机(图12A中未示出)、一个或更多个速度传感器1244(例如,用于测量车辆1200的速度)、一个或更多个振动传感器1242、一个或更多个转向传感器1240、一个或更多个制动传感器(例如,作为制动传感器系统1246的一部分)和/或其他传感器类型接收。

[0135] 在至少一个实施例中,一个或更多个控制器1236可以从车辆1200的仪表板1232接收输入(例如,由输入数据表示)并通过人机接口(“HMI”)显示器1234、声音信号器、扬声器和/或车辆1200的其他组件提供输出(例如,由输出数据、显示数据等表示)。在至少一个实施例中,输出可包括信息,诸如车速、速度、时间、地图数据(例如,高清晰度地图(图12A中未显示)、位置数据(例如,车辆1200的位置,例如在地图上)、方向、其他车辆的位置(例如,占用光栅)、关于对象的信息以及由一个或更多个控制器1236感知到的对象的状态等。例如,在至少一个实施例中,HMI显示器1234可以显示关于一个或更多个对象的存在的信息(例如,路牌、警告标志、交通信号灯变更等)和/或有关驾驶操作车辆已经、正在或将要制造的信息(例如,现在改变车道、在两英里内驶出34B出口等)。

[0136] 在至少一个实施例中,车辆1200进一步包括网络接口1224,其可以使用一个或更多个无线天线1226和/或一个或更多个调制解调器通过一个或更多个网络进行通信。例如,在至少一个实施例中,网络接口1224可能能够通过长期演进(“LTE”)、宽带码分多址(“WCDMA”)、通用移动通信系统(“UMTS”)、全球移动通信系统(“GSM”)、IMT-CDMA多载波(“CDMA2000”)等进行通信。在至少一个实施例中,一个或更多个无线天线1226还可以使用一个或更多个局域网(例如Bluetooth、Bluetooth Low Energy (LE)、Z-Wave、ZigBee等)和/或一个或更多个低功耗广域网(以下简称“LPWAN”) (例如LoRaWAN、SigFox等),使环境中的对象(例如,车辆、移动设备)之间进行通信。

[0137] 推理和/或训练逻辑915用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图9A和/或图9B提供关于推理和/或训练逻辑915的细节。在至少一个实施例中,推理和/或训练逻辑915可以在系统图12A中用于至少部分地基于使用神经网络训练操作\神经网络功能和/或架构或本文所述的神经网络用例计算出的权重参数来推理或预测操作。

[0138] 在至少一个实施例中,空间自适应可分离卷积层7可以和系统图12A一起用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构、或本文所述的神经网络用例计算的权重参数来推理和预测操作。

[0139] 图12B示出了根据至少一个实施例的图12A的自动驾驶车辆1200的相机位置和视野的示例。在至少一个实施例中,相机和各自的视野是一个示例实施例,并且不旨在进行限制。例如,在至少一个实施例中,可以包括附加的和/或替代的相机和/或相机可以位于车辆1200上的不同位置。

[0140] 在至少一个实施例中,用于相机的相机类型可以包括但不限于可以适于与车辆

1200的组件和/或系统一起使用的数字相机。一个或多个相机可以以汽车安全完整性级别(“ASIL”)B和/或其他ASIL进行操作。在至少一个实施例中,根据实施例,相机类型可以具有任何图像捕获速率,例如60帧每秒(fps)、1220fps、240fps等。在至少一个实施例中,相机可以能够使用滚动快门、全局快门、另一种类型的快门或其组合。在至少一个实施例中,滤色器阵列可以包括红色透明透明(“RCCC”)滤色器阵列、红色透明透明蓝色(“RCCB”)滤色器阵列、红色蓝色绿色透明(“RBGC”)滤色器阵列、Foveon X3滤色器阵列、拜耳(Bayer)传感器(“RGGB”)滤色器阵列、单色传感器滤色器阵列和/或其他类型的滤色器阵列。在至少一个实施例中,可以使用透明像素相机,例如具有RCCC、RCCB和/或RBGC滤色器阵列的相机,以努力提高光敏性。

[0141] 在至少一个实施例中,一个或多个相机可以用于执行先进驾驶员辅助系统(“ADAS”)功能(例如,作为冗余或故障安全设计的一部分)。例如,在至少一个实施例中,可以安装多功能单声道相机以提供包括车道偏离警告、交通标志辅助和智能大灯控制的功能。在至少一个实施例中,一个或多个相机(例如,所有相机)可以同时记录并提供图像数据(例如,视频)。

[0142] 在至少一个实施例中,可以将相机中的一个或多个安装在安装组件中,例如定制设计的(三维(“3D”)打印的)组件,以便切出杂散光和来自在车内的反光(例如,仪表板的反射在挡风玻璃镜中反光),其可能会干扰相机的图像数据捕获能力。关于后视镜安装组件,在至少一个实施例中,后视镜组件可以是3D打印定制的,使得相机安装板匹配后视镜的形状。在至少一个实施例中,一个或多个相机可以被集成到后视镜中。在至少一个实施例中,对于侧视相机,一个或多个相机也可以集成在舱室的每个角落的四个支柱内。

[0143] 在至少一个实施例中,具有包括车辆1200前面的环境的部分的视野的相机(例如,前向相机)可以用于环视,以及在一个或多个控制器1236和/或控制SoC的帮助下帮助识别向前的路径和障碍物,从而提供对于生成占用网格和/或确定优选的车辆路径至关重要的信息。在至少一个实施例中,前向相机可以用于执行许多与LIDAR相同的ADAS功能,包括但不限于紧急制动、行人检测和避免碰撞。在至少一个实施例中,前向相机也可以用于ADAS功能和系统,包括但不限于车道偏离警告(“LDW”)、自动巡航控制(“ACC”)和/或其他功能(例如交通标志识别)。

[0144] 在至少一个实施例中,各种相机可以用于前向配置,包括例如包括CMOS(“互补金属氧化物半导体”)彩色成像器的单目相机平台。在至少一个实施例中,广角相机1270可以用于感知从外围进入的对象(例如,行人、过马路或自行车)。尽管在图12B中仅示出了一个广角相机1270,但是,在其他实施例中,车辆1200上可以有任何数量(包括零)的广角相机1270。在至少一个实施例中,任何数量的远程相机1298(例如,远程立体相机对)可用于基于深度的对象检测,尤其是对于尚未训练神经网络的对象。在至少一个实施例中,远程相机1298也可以用于对象检测和分类以及基本对象跟踪。

[0145] 在至少一个实施例中,任何数量的立体声相机1268也可以包括在前向配置中。在至少一个实施例中,一个或多个立体声相机1268可以包括集成控制单元,该集成控制单元包括可缩放处理单元,该可缩放处理单元可以提供可编程逻辑(“FPGA”)和具有单个芯片上集成的控制器局域网(“CAN”)或以太网接口的多核心微处理器。在至少一个实施例中,这样的单元可以用于生成车辆1200的环境的3D地图,包括对图像中所有点的距离估计。在至

少一个实施例中,一个或更多个立体相机1268可以包括但不限于紧凑型立体视觉传感器,其可以包括但不限于两个相机样本(左右分别一个)和一个图像处理芯片,其可以测量从车辆1200到目标对象的距离并使用所生成的信息(例如,元数据)来激活自主紧急制动和车道偏离警告功能。在至少一个实施例中,除了本文所述的那些之外,还可以使用其他类型的立体相机1268。

[0146] 在至少一个实施例中,具有包括车辆1200侧面的环境的一部分的视野的相机(例如,侧视相机)可以用于环绕查看,从而提供用于创建和更新占据网格的信息,以及产生侧面碰撞警告。例如,在至少一个实施例中,环绕相机1274(例如,如图12B所示的四个环绕相机1274)可以定位在车辆1200上。一个或更多个环绕相机1274可以包括但不限于,任意数量和组合的广角相机1270、一个或更多个鱼目样本、一个或更多个360度相机和/或类似的相机。例如,在至少一个实施例中,四个鱼目样本相机可以位于车辆1200的前、后和侧面。在至少一个实施例中,车辆1200可以使用三个环绕相机1274(例如,左、右和后面),并且可以利用一个或更多个其他相机(例如,前向相机)作为第四个环视相机。

[0147] 在至少一个实施例中,具有包括车辆1200后方的环境的一部分的视野的相机(例如,后视相机)可以用于停车辅助、环视、后方碰撞警告、以及创建和更新占用光栅。在至少一个实施例中,可以使用各种各样的相机,包括但不限于还适合作为一个或更多个前向相机的相机(例如,远程相机1298和/或一个或更多个中程相机1276、一个或更多个立体相机1268、一个或更多个红外相机1272等),如本文所述。

[0148] 推理和/或训练逻辑915用于执行与一个或更多个实施例相关联的推理和/或训练操作。结合图9A和/或图9B,在本文中提供了关于推理和/或训练逻辑915的细节。在至少一个实施例中,推理和/或训练逻辑915可以在图12B的系统中使用,用于至少部分地基于使用神经网络训练操作计算出的权重参数、神经网络函数和/或架构、或本文所述的神经网络用例来推理或预测操作。

[0149] 在至少一个实施例中,空间自适应可分离卷积层7可以和系统图12B一起用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构、或本文所述的神经网络用例计算的权重参数来推理和预测操作。

[0150] 图12C示出了根据至少一个实施例的图12A的自动驾驶车辆1200的示例系统架构的框图。在至少一个实施例中,图12C中的车辆1200的一个或更多个组件、一个或更多个特征和一个或更多个系统中的每一个都示出为经由总线1202连接。在至少一个实施例中,总线1202可包括但不限于CAN数据接口(在本文中可替代地称为“CAN总线”)。在至少一个实施例中,CAN可以是车辆1200内部的网络,用于帮助控制车辆1200的各种特征和功能,例如制动器的致动、加速、制动、转向、雨刷等。在一个实施例中,总线1202可以配置成具有数十个甚至数百个节点,每个节点具有其自己的唯一识别符(例如,CAN ID)。在至少一个实施例中,可以读取总线1202以找到方向盘角度、地面速度、发动机每分钟转动次数(“RPM”)、按钮位置和/或其他车辆状态指示器。在至少一个实施例中,总线1202可以是符合ASIL B的CAN总线。

[0151] 在至少一个实施例中,除了CAN之外或可从FlexRay和/或以太网(Ethernet)中选择其一使用。在至少一个实施例中,可以有任意数量的总线1202,其可以包括但不限于零或更多的CAN总线,零或更多的FlexRay总线,零或更多的以太网总线,和/或零或更多的使

用其他协议的其他类型的总线。在至少一个实施例中,两个或更多个总线1202可以用于执行不同的功能,和/或可以用于冗余。例如,第一总线1202可以用于碰撞避免功能,并且第二总线1202可以用于致动控制。在至少一个实施例中,每个总线1202可以与车辆1200的任何组件通信,并且两个或更多个总线1202可以与相同的组件通信。在至少一个实施例中,任何数量的片上系统(“SoC”)1204中的每一个,一个或更多个控制器1236中的每一个和/或车辆内的每个计算机都可以访问相同的输入数据(例如,来自车辆1200的传感器的输入),并且可以连接到公共总线,例如CAN总线。

[0152] 在至少一个实施例中,车辆1200可以包括一个或更多个控制器1236,诸如本文关于图12A所描述的那些。控制器1236可以用于多种功能。在至少一个实施例中,控制器1236可以耦合到车辆1200的各种其他组件和系统中的任何一个,并且可以用于控制车辆1200、车辆1200的人工智能、车辆1200的信息娱乐和/或相似功能。

[0153] 在至少一个实施例中,车辆1200可以包括任何数量的SoC 1204。SoC1204中的每一个可以包括但不限于中央处理单元(“一个或更多个CPU”)1206、图形处理单元(“一个或更多个GPU”)1208、一个或更多个处理器1210、一个或更多个高速缓存1212、一个或更多个加速器1214、一个或更多个数据存储1216和/或其他未显示的组件和特征。在至少一个实施例中,一个或更多个SoC 1204可以用于在各种平台和系统中控制车辆1200。例如,在至少一个实施例中,一个或更多个SoC 1204可以与高清晰度(“HD”)地图1222在系统(例如,车辆1200的系统)中组合,该高清晰度地图1222可以经由网络接口1224从一个或更多个服务器(图12C中未示出)获得地图刷新和/或更新。

[0154] 在至少一个实施例中,一个或更多个CPU 1206可以包括CPU集群或CPU复合体(在本文中可替代地称为“CCPLEX”)。在至少一个实施例中,一个或更多个CPU 1206可以包括多个核心和/或二级(“L2”)高速缓存。例如,在至少一个实施例中,一个或更多个CPU 1206可以在相互耦合的多处理器配置中包括八个核心。在至少一个实施例中,一个或更多CPU 1206可以包括四个双核心集群,其中每个集群具有专用的L2高速缓存(例如,2MB L2高速缓存)。在至少一个实施例中,一个或更多CPU 1206(例如,CCPLEX)可以配置成支持同时的集群操作,使得一个或更多CPU 1206的集群的任何组合在任何给定的时间都可以是活跃的。

[0155] 在至少一个实施例中,一个或更多个CPU 1206可以实现电源管理功能,这些功能包括但不限于以下特征中的一个或更多个:空闲时可以自动对各个硬件模块进行时钟门控以节省动态功率;当核心由于执行等待中断(“WFI”)/事件等待(“WFE”)指令而未主动执行指令时,可以对每个核心时钟进行门控;每个核心都可以独立供电;当所有核心都被时钟门控或功率门控时,每个核心集群可以被独立地时钟门控;以及/或当所有核心都被功率门控时,每个核心集群可以被独立地功率门控。在至少一个实施例中,一个或更多CPU 1206可以进一步实现用于管理功率状态的增强算法,其中指定了允许的功率状态和预期的唤醒时间,并且硬件/微码确定了针对核心、集群和CCPLEX输入的最佳功率状态。在至少一个实施例中,处理核心可以在软件中支持简化的功率状态输入序列,其中工作被分担给微码。

[0156] 在至少一个实施例中,一个或更多个GPU 1208可以包括集成的GPU(在本文中或者称为“iGPU”)。在至少一个实施例中,一个或更多个GPU 1208可以是可编程的,并且对于并行工作负载可以是有效的。在至少一个实施例中,一个或更多个GPU 1208在至少一个实施例中可以使用增强的张量指令集。在一个实施例中,一个或更多个GPU 1208可以包括一个

或更多个流式微处理器,其中每个流式微处理器可以包括一级(“L1”)高速缓存(例如,具有至少96KB的存储容量的L1高速缓存),以及流式微处理器中的两个或更多个可以共享L2高速缓存(例如,具有512KB存储容量的L2高速缓存)。在至少一个实施例中,一个或更多个GPU 1208可以包括至少八个流式微处理器。在至少一个实施例中,一个或更多个GPU 1208可以使用计算应用程序编程接口(API)。在至少一个实施例中,一个或更多GPU1208可以使用一个或更多个并行计算平台和/或编程模型(例如,NVIDIA的CUDA)。

[0157] 在至少一个实施例中,一个或更多个GPU 1208可以经功耗优化以在汽车和嵌入式用例中获得最佳性能。例如,在一个实施例中,可以在鳍式场效应晶体管(“FinFET”)上制造一个或更多个GPU 1208。在至少一个实施例中,每个流式微处理器可以包含多个划分为多个块的混合精度处理核心,。例如但不限于,可以将64个PF32核心和32个PF64核心划分为四个处理块。在至少一个实施例中,可以为每个处理块分配16个FP32核心、8个FP64核心、16个INT32核心、两个用于深度学习矩阵算术的混合精度NVIDIA张量核心、零级(“L0”)指令缓存、线程束调度器、分派单元和/或64KB寄存器文件。在至少一个实施例中,流式微处理器可以包括独立的并行整数和浮点数据路径来提供混合了计算和寻址运算的工作量的有效执行。在至少一个实施例中,流式微处理器可以包括独立的线程调度能力,以实现更细粒度的同步和并行线程之间的协作。在至少一个实施例中,流式微处理器可以包括组合的L1数据高速缓存和共享存储器单元,以便在简化编程的同时提高性能。

[0158] 在至少一个实施例中,一个或更多个GPU 1208可以包括高带宽存储器(“HBM”)和/或16GB HBM2存储器子系统,以在一些示例中提供约900GB/秒的峰值存储带宽。在至少一个实施例中,除了或替代于HBM存储器,可以使用同步图形随机存取存储器(“SGRAM”),例如图形双倍数据速率类型的五同步随机存取存储器(“GDDR5”)。

[0159] 在至少一个实施例中,一个或更多个GPU 1208可以包括统一存储器技术。在至少一个实施例中,地址转换服务(“ATS”)支持可以用于允许一个或更多个GPU 1208直接访问一个或更多个CPU 1206页表。在至少一个实施例中,当一个或更多个GPU 1208存储器管理单元(“MMU”)经历未命中时,可以将地址转换请求发送到一个或更多个CPU 1206。作为响应,在至少一个实施例中,一个或更多个CPU 1206可以在其页面表中查找地址的虚拟-物理的映射并将转换传送回一个或更多个GPU 1208。在至少一个实施例中,统一存储器技术可以允许单个统一虚拟地址空间用于一个或更多个CPU 1206和一个或更多个GPU 1208两者的存储器,从而简化了一个或更多个GPU 1208的编程以及将应用程序移植到一个或更多个GPU 1208。

[0160] 在至少一个实施例中,一个或更多个GPU 1208可以包括任意数量的访问计数器,其可以跟踪一个或更多个GPU 1208对其他处理器的存储器的访问频率。在至少一个实施例中,一个或更多个访问计数器可以帮助确保将存储器页移动到最频繁访问页面的处理器的物理存储器中,从而提高处理器之间共享的存储器范围的效率。

[0161] 在至少一个实施例中,一个或更多个SoC 1204可以包括任何数量的高速缓存1212,包括本文所述的那些。例如,在至少一个实施例中,一个或更多个高速缓存1212可以包括可用于一个或更多个CPU 1206和一个或更多个GPU 1208(例如,连接到CPU 1206和GPU 1208两者)的三级(“L3”)高速缓存。在至少一个实施例中,一个或更多个高速缓存1212可以包括回写式高速缓存,该回写式高速缓存可以例如通过使用高速缓存相干协议(例如,MEI、

MESI、MSI等)来跟踪线的状态。在至少一个实施例中,尽管可以使用较小的高速缓存大小,根据实施例,L3高速缓存可以包括4MB或更多。

[0162] 在至少一个实施例中,一个或更多个SoC 1204可以包括一个或更多个加速器1214(例如,硬件加速器、软件加速器或其组合)。在至少一个实施例中,一个或更多个SoC 1204可以包括硬件加速集群,其可以包括优化的硬件加速器和/或大的片上存储器。在至少一个实施例中,大的片上存储器(例如4MB的SRAM)可以使硬件加速集群能够加速神经网络和其他计算。在至少一个实施例中,硬件加速集群可以用于补充一个或更多个GPU 1208并且卸载一个或更多个GPU 1208的一些任务(例如,释放一个或更多个GPU 1208的更多周期以执行其他任务)。在至少一个实施例中,一个或更多个加速器1214可以用于足够稳定以经得起加速检验的目标工作负载(例如,感知、卷积神经网络(“CNN”)、递归神经网络(“RNN”)等)。在至少一个实施例中,CNN可以包括基于区域或区域卷积神经网络(“RCNN”)和快速RCNN(例如,如用于对象检测)或其他类型的CNN。

[0163] 在至少一个实施例中,一个或更多个加速器1214(例如,硬件加速集群)可以包括一个或更多个深度学习加速器(“DLA”)。一个或更多个DLA可以包括但不限于一个或更多个Tensor处理单元(“TPU”),其可以配置成每秒提供额外的10万亿次操作用于深度学习应用程序和推理。在至少一个实施例中,TPU可以是配置成并被优化用于执行图像处理功能(例如,用于CNN、RCNN等)的加速器。可以针对神经网络类型和浮点运算以及推理的特定集合进一步优化一个或更多个DLA。在至少一个实施例中,一个或更多个DLA的设计可以提供比典型的通用GPU更高的每毫米性能,并且通常大大超过CPU的性能。在至少一个实施例中,一个或更多个TPU可执行若干功能,包括支持例如INT8、INT16和FP16数据类型以用于特征和权重的单实例卷积功能以及后处理器功能的。在至少一个实施例中,一个或更多个DLA可以针对各种功能中的任何功能,在处理或未处理的数据上快速且有效地执行神经网络,尤其是CNN,包括例如但不限于:用于使用来自相机传感器的数据进行对象识别和检测的CNN;用于使用来自相机传感器的数据进行距离估算的CNN;用于使用来自麦克风1296的数据进行紧急车辆检测以及识别和检测的CNN;用于使用来自相机传感器的数据进行人脸识别和车主识别的CNN;以及/或用于安全和/或安全相关事件的CNN。

[0164] 在至少一个实施例中,DLA可以执行一个或更多个GPU 1208的任何功能,并且通过使用推理加速器,例如,设计者可以将一个或更多个DLA或一个或更多个GPU 1208作为目标用于任何功能。例如,在至少一个实施例中,设计者可以将CNN的处理和浮点运算集中在一个或更多个DLA上,并将其他功能留给一个或更多个GPU 1208和/或其他一个或更多个加速器1214。

[0165] 在至少一个实施例中,一个或更多个加速器1214(例如硬件加速集群)可以包括可编程视觉加速器(“PVA”),其在本文中可替代地称为计算机视觉加速器。在至少一个实施例中,一个或更多个PVA可以设计和配置为加速用于高级驾驶员辅助系统(“ADAS”)1238、自动驾驶、增强现实(“AR”)应用程序和/或虚拟现实(“VR”)应用程序的计算机视觉算法。一个或更多个PVA可以在性能和灵活性之间取得平衡。例如,在至少一个实施例中,一个或更多个PVA中的每一个可以包括例如但不限于任何数量的精简指令集计算机(“RISC”)核心、直接存储器访问(“DMA”)和/或任意数量的向量处理器。

[0166] 在至少一个实施例中,RISC核心可以与图像传感器(例如,本文描述的任意相机的

图像传感器)、图像信号处理器等交互。在至少一个实施例中,每个RISC核心可以包括任意数量的存储器。在至少一个实施例中,根据实施例,RISC核心可以使用多种协议中的任意一种。在至少一个实施例中,RISC核心可以执行实时操作系统(“RTOS”)。在至少一个实施例中,可以使用一个或多个集成电路设备、专用集成电路(“ASIC”)和/或存储设备来实现RISC核心。例如,在至少一个实施例中,RISC核心可以包括指令高速缓存和/或紧密耦合的RAM。

[0167] 在至少一个实施例中,DMA可以使一个或多个PVA的组件能够独立于一个或多个CPU 1206访问系统存储器。在至少一个实施例中,DMA可以支持用于向PVA提供优化的任何数量的特征,包括但不限于,支持多维寻址和/或循环寻址。在至少一个实施例中,DMA可以支持多达六个或多个寻址的维度,其可以包括但不限于块宽度、块高度、块深度、水平块步进、垂直块步进和/或深度步进。

[0168] 在至少一个实施例中,向量处理器可以是可编程处理器,其可以设计为有效且灵活地执行用于计算机视觉算法并提供信号处理能力的编程。在至少一个实施例中,PVA可以包括PVA核心和两个向量处理子系统分区。在至少一个实施例中,PVA核心可以包括处理器子系统、DMA引擎(例如,两个DMA引擎)和/或其他外围设备。在至少一个实施例中,向量处理子系统可以用作PVA的主要处理引擎,并且可以包括向量处理单元(“VPU”)、指令高速缓存和/或向量存储器(例如“VMEM”)。在至少一个实施例中,VPU核心可以包括数字信号处理器,例如,单指令多数据(“SIMD”)、超长指令字(“VLIW”)数字信号处理器。在至少一个实施例中,SIMD和VLIW的组合可以提高吞吐量和速度。

[0169] 在至少一个实施例中,每个向量处理器可以包括指令高速缓存并且可以耦合到专用存储器。结果,在至少一个实施例中,每个向量处理器可以配置为独立于其他向量处理器执行。在至少一个实施例中,特定PVA中包括的向量处理器可以配置为采用数据并行性。例如,在至少一个实施例中,单个PVA中包括的多个向量处理器可以执行相同的计算机视觉算法,除了在图像的不同区域上之外。在至少一个实施例中,包括在特定PVA中的向量处理器可以在相同图像上同时执行不同的计算机视觉算法,或者甚至在序列图像或部分图像上执行不同的算法。在至少一个实施例中,除其他外,在硬件加速集群中可以包括任何数量的PVA,并且在每个PVA中可以包括任何数量的向量处理器。在至少一个实施例中,PVA可以包括附加的纠错码(“ECC”)存储器,以增强整体系统安全性。

[0170] 在至少一个实施例中,一个或多个加速器1214(例如硬件加速集群)可以包括片上计算机视觉网络和静态随机存取存储器(“SRAM”),用于为一个或多个加速器1214提供高带宽,低延迟SRAM。在至少一个实施例中,片上存储器可以包括至少4MB SRAM,其包含但不限于八个现场可配置的内存块,PVA和DLA均可以对其进行访问。在至少一个实施例中,每对存储块可以包括高级外围总线(“APB”)接口、配置电路、控制器和多路复用器。在至少一个实施例中,可以使用任何类型的存储器。在至少一个实施例中,PVA和DLA可以经由为PVA和DLA提供对存储器的高速访问的主干网来访问存储器。在至少一个实施例中,主干网可以包括片上计算机视觉网络,其将PVA和DLA互连到存储器(例如,使用APB)。

[0171] 在至少一个实施例中,片上计算机视觉网络可以包括接口,该接口在传输任何控制信号/地址/数据之前确定PVA和DLA均提供就绪和有效信号。在至少一个实施例中,接口可以提供用于发送控制信号/地址/数据的单独的相位和单独的信道,以及用于连续数据传

输的突发型通信。在至少一个实施例中,尽管可以使用其他标准和协议,但是接口可以符合国际标准化组织(“ISO”)26262或国际电工委员会(“IEC”)61508标准。

[0172] 在至少一个实施例中,一个或更多个SoC 1204可以包括实时视线追踪硬件加速器。在至少一个实施例中,实时视线追踪硬件加速器可以用于快速且有效地确定对象的位置和范围(例如,在世界模型内),以生成实时可视化模拟,以用于RADAR信号解释,用于声音传播合成和/或分析,用于SONAR系统的模拟,用于一般的波传播模拟,与用于定位和/或其他功能的LIDAR数据进行比较,和/或用于其他用途。

[0173] 在至少一个实施例中,一个或更多个加速器1214(例如硬件加速集群)具有用于自动驾驶的广泛用途。在至少一个实施例中,PVA可以是可用于ADAS和自动驾驶汽车中的关键处理阶段的可编程视觉加速器。在至少一个实施例中,在低功耗和低延迟下PVA的能力与需要可预测的处理的算法域良好匹配。换句话说,PVA在半密集或密集的常规计算中表现出色,即使在小型数据集上也是如此,这些数据集需要具有低延迟和低功耗的可预测的运行时间。在至少一个实施例中,自主车辆,诸如车辆1200,PVA被设计为运行经典的计算机视觉算法,因为它们在对对象检测和整数数学运算方面是有效的。

[0174] 例如,根据技术的至少一个实施例,PVA被用于执行计算机立体视觉。在至少一个实施例中,可以在一些示例中使用基于半全局匹配的算法,尽管这并不意味着限制性。在至少一个实施例中,用于3-5级自动驾驶的应用程序在运行中使用动态的估计/立体匹配(例如,从运动中恢复结构、行人识别、车道检测等)。在至少一个实施例中,PVA可以对来自两个单目相机的输入执行计算机立体视觉功能。

[0175] 在至少一个实施例中,PVA可以用于执行密集的光流。例如,在至少一个实施例中,PVA可以处理原始RADAR数据(例如,使用4D快速傅立叶变换)以提供处理后的RADAR数据。在至少一个实施例中,例如,通过处理原始飞行时间数据以提供处理后的飞行时间数据,将PVA用于飞行时间深度处理。

[0176] 在至少一个实施例中,DLA可用于运行任何类型的网络以增强控制和驾驶安全性,包括例如但不限于神经网络,其输出用于每个对象检测的置信度。在至少一个实施例中,可以将置信度表示或解释为概率,或者表示为提供每个检测相对于其他检测的相对“权重”。在至少一个实施例中,置信度使系统能够做出进一步的决定,即关于哪些检测应当被认为是真正的阳性检测而不是假阳性检测。例如,在至少一个实施例中,系统可以为置信度设置阈值,并且仅将超过阈值的检测视为真阳性检测。在使用自动紧急制动(“AEB”)系统的实施例中,假阳性检测将导致车辆自动执行紧急制动,这显然是不希望的。在至少一个实施例中,高度自信的检测可以被认为是AEB的触发。在至少一个实施例中,DLA可以运行用于回归置信度值的神经网络。在至少一个实施例中,神经网络可以将参数的至少一些子集作为其输入,例如包围盒尺寸,获得的地平面估计(例如,从另一子系统),与从神经网络和/或其他传感器(例如,一个或更多个LIDAR传感器1264或一个或更多个RADAR传感器1260)等获得的对象的车辆1200方向、距离、3D位置估计相关的一个或更多个IMU传感器1266的输出。

[0177] 在至少一个实施例中,一个或更多个SoC 1204可以包括一个或更多个数据存储装置1216(例如,存储器)。在至少一个实施例中,一个或更多个数据存储1216可以是一个或更多个SoC 1204的片上存储器,其可以存储要在一个或更多个GPU 1208和/或DLA上执行的神经网络。在至少一个实施例中,一个或更多个数据存储1216可以具有足够大的容量以存储

神经网络的多个实例以用于冗余和安全。在至少一个实施例中，一个或更多个数据存储1212可以包括L2或L3高速缓存。

[0178] 在至少一个实施例中，一个或更多个SoC 1204可以包括任何数量的处理器1210（例如，嵌入式处理器）。一个或更多个处理器1210可以包括启动和电源管理处理器，该启动和电源管理处理器可以是专用处理器和子系统，以处理启动电源和管理功能以及相关的安全实施。在至少一个实施例中，启动和电源管理处理器可以是一个或更多个SoC 1204启动序列的一部分，并且可以提供运行时电源管理服务。在至少一个实施例中，启动功率和管理处理器可以提供时钟和电压编程，辅助系统低功率状态转换，一个或更多个SoC 1204热和温度传感器管理和/或一个或更多个SoC 1204功率状态管理。在至少一个实施例中，每个温度传感器可以实现为其输出频率与温度成比例的环形振荡器，并且一个或更多个SoC 1204可以使用环形振荡器来检测一个或更多个CPU 1206，一个或更多个GPU 1208和/或一个或更多个加速器1214的温度。在至少一个实施例中，如果确定温度超过阈值，则启动和电源管理处理器可以进入温度故障例程，并将一个或更多个SoC 1204置于较低功耗状态和/或将车辆1200置于司机的安全停车图案（例如，使车辆1200安全停车）。

[0179] 在至少一个实施例中，一个或更多个处理器1210可以进一步包括一组嵌入式处理器，其可以用作音频处理引擎。在至少一个实施例中，音频处理引擎可以是音频子系统，其能够通过多个接口以及广泛且灵活范围的音频I/O接口为硬件提供对多通道音频的完全硬件支持。在至少一个实施例中，音频处理引擎是专用处理器核心，其具有带专用RAM的数字信号处理器。

[0180] 在至少一个实施例中，一个或更多个处理器1210可以进一步包括始终在线的处理器引擎，该引擎可以提供必要的硬件特征以支持低功率传感器管理和唤醒用例。在至少一个实施例中，始终在线的处理器引擎上的处理器可以包括但不限于处理器核心、紧密耦合的RAM、支持外围设备（例如，定时器和中断控制器）、各种I/O控制器外围设备以及路由逻辑。

[0181] 在至少一个实施例中，一个或更多个处理器1210可以进一步包括安全集群引擎，该安全集群引擎包括但不限于用于处理汽车应用程序的安全管理的专用处理器子系统。在至少一个实施例中，安全集群引擎可以包括但不限于两个或更多个处理器核心、紧密耦合的RAM、支持外围设备（例如，定时器、中断控制器等）和/或路由逻辑。在安全模式下，在至少一个实施例中，两个或更多个核心可以以锁步模式操作，并且可以用作具有用以检测其操作之间的任何差异的比较逻辑的单个核心。在至少一个实施例中，一个或更多个处理器1210可以进一步包括实时相机引擎，该实时相机引擎可以包括但不限于用于处理实时相机管理的专用处理器子系统。在至少一个实施例中，一个或更多个处理器1210可以进一步包括高动态范围信号处理器，该高动态范围信号处理器可以包括但不限于图像信号处理器，该图像信号处理器是作为相机处理管线的一部分的硬件引擎。

[0182] 在至少一个实施例中，一个或更多个处理器1210可以包括视频图像合成器，该视频图像合成器可以是处理块（例如，在微处理器上实现），该处理块实现视频回放应用程序产生最终的视频所需要的视频后处理功能，以产生用于播放器窗口的最终图像。在至少一个实施例中，视频图像合成器可以在一个或更多个广角相机1270、一个或更多个环绕相机1274和/或一个或更多个舱内监控相机传感器上执行透镜畸变校正。在至少一个实施例中，

优选地,由在SoC 1204的另一实例上运行的神经网络来监控舱室内监控相机传感器,该神经网络被配置为识别舱室事件并相应地做出响应。在至少一个实施例中,舱室内系统可以执行但不限于唇读以激活蜂窝服务和拨打电话、指示电子邮件、改变车辆的目的地、激活或改变车辆的信息娱乐系统和设置、或者提供语音激活的网上冲浪。在至少一个实施例中,当车辆以自主模式运行时,某些功能对于驾驶员是可用的,否则将其禁用。

[0183] 在至少一个实施例中,视频图像合成器可以包括用于同时空间和时间降噪的增强的时间降噪。例如,在至少一个实施例中,在运动发生在视频中的情况下,降噪适当地对空间信息加权,从而减小由相邻帧提供的信息的权重。在至少一个实施例中,在图像或图像的一部分不包括运动的情况下,由视频图像合成器执行的时间降噪可以使用来自先前图像的信息来降低当前图像中的噪声。

[0184] 在至少一个实施例中,视频图像合成器还可以配置为对输入的立体透镜帧执行立体校正。在至少一个实施例中,当使用操作系统桌面时,视频图像合成器还可以用于用户接口合成,并且不需要一个或更多个GPU 1208来连续渲染新表面。在至少一个实施例中,当对一个或更多个GPU 1208供电并使其活跃地进行3D渲染时,视频图像合成器可以被用于卸载一个或更多个GPU 1208以改善性能和响应性。

[0185] 在至少一个实施例中,一个或更多个SoC 1204可以进一步包括用于从相机接收视频和输入的移动工业处理器接口(“MIPI”)相机串行接口、高速接口和/或可用于相机和相关像素输入功能的视频输入块。在至少一个实施例中,一个或更多个SoC 1204可以进一步包括输入/输出控制器,该输入/输出控制器可以由软件控制并且可以被用于接收未提交给特定角色的I/O信号。

[0186] 在至少一个实施例中,一个或更多个SoC 1204可以进一步包括广泛的外围接口,以使得能够与外围设备、音频编码器/解码器(“编解码器”),电源管理和/或其他设备通信。一个或更多个SoC 1204可用于处理来自(例如,通过千兆位多媒体串行链路和以太网连接)相机、传感器(例如,一个或更多个LIDAR传感器1264,一个或更多个RADAR传感器1260等,其可以通过以太网连接)的数据,来自总线1202的数据(例如,车辆1200的速度、方向盘位置等),来自一个或更多个GNSS传感器1258的数据(例如,通过以太网或CAN总线连接)等。在至少一个实施例中,一个或更多个SoC 1204可以进一步包括专用高性能海量存储控制器,其可以包括它们自己的DMA引擎,并且可以用于使一个或更多个CPU 1206摆脱常规数据管理任务。

[0187] 在至少一个实施例中,一个或更多个SoC 1204可以是具有灵活架构的端到端平台,其跨越自动化级别3-5级,从而提供利用并有效使用计算机视觉和ADAS技术来实现多样性和冗余的综合的功能安全架构,其提供了可提供灵活、可靠的驾驶软件堆栈以及深度学习工具的平台。在至少一个实施例中,一个或更多个SoC 1204可以比常规系统更快、更可靠,并且甚至在能量效率和空间效率上也更高。例如,在至少一个实施例中,一个或更多个加速器1214当与一个或更多个CPU 1206、一个或更多个GPU 1208以及一个或更多个数据存储装置1216结合时,可以提供用于3-5级自动驾驶车辆的快速、有效的平台。

[0188] 在至少一个实施例中,计算机视觉算法可以在CPU上执行,CPU可以使用高级编程语言(例如C程序设计语言)配置为在多种视觉数据上执行多种处理算法。然而,在至少一个实施例中,CPU通常不能满足许多计算机视觉应用程序的性能要求,例如与执行时间和功耗

有关的性能要求。在至少一个实施例中,许多CPU不能实时执行复杂的对象检测算法,该算法被用于车载ADAS应用程序和实际3-5级自动驾驶车辆中。

[0189] 本文所述的实施例允许同时和/或序列地执行多个神经网络,并且允许将结果结合在一起以实现3-5级自动驾驶功能。例如,在至少一个实施例中,在DLA或离散GPU(例如,一个或更多个GPU 1220)上执行的CNN可包括文本和单词识别,从而允许超级计算机读取和理解交通标志,包括神经网络尚未被专门训练的标志。在至少一个实施例中,DLA还可包括神经网络,该神经网络能够识别、解释并提供符号的语义理解,并将该语义理解传递给在CPU Complex上运行的路径规划模块。

[0190] 在至少一个实施例中,对于3、4或5级的驱动,可以同时运行多个神经网络。例如,在至少一个实施例中,由“警告:闪烁的灯指示结冰状况(Caution:flashing lights indicate icy conditions)”和电灯一起组成的警告标志可以由多个神经网络独立地或共同地解释。在至少一个实施例中,可以通过第一部署的神经网络(例如,已经训练的神经网络)将标志本身识别为交通标志,可以通过第二部署的神经网络来解释文本“闪烁的灯指示结冰状况(flashes indicate icy conditions)”,其通知车辆的路径规划软件(最好在CPU Complex上执行):当检测到闪烁的灯光时,就会存在结冰状况。在至少一个实施例中,可以通过在多个帧上操作第三部署的神经网络来识别闪烁的灯,向车辆的路径规划软件通知存在(或不存在)闪烁的灯。在至少一个实施例中,所有三个神经网络可以同时运行,例如在DLA内和/或在一个或更多个GPU 1208上。

[0191] 在至少一个实施例中,用于面部识别和车辆所有者识别的CNN可以使用来自相机传感器的数据来识别授权驾驶员和/或车辆1200的所有者的存在。在至少一个实施例中,当所有者接近驾驶员门并打开灯时,常开传感器处理器引擎可用于解锁车辆,并且,在安全模式下,当所有者离开车辆时,可用于禁用车辆。以此方式,一个或更多个SoC 1204提供防止盗窃和/或劫车的保障。

[0192] 在至少一个实施例中,用于紧急车辆检测和识别的CNN可以使用来自麦克风1296的数据来检测和识别紧急车辆警报器。在至少一个实施例中,一个或更多个SoC 1204使用CNN来对环境和城市声音进行分类,以及对视觉数据进行分类。在至少一个实施例中,训练在DLA上运行的CNN以识别紧急车辆的相对接近速度(例如,通过使用多普勒效应)。在至少一个实施例中,还可以训练CNN来识别针对车辆正在运行的区域的紧急车辆,如一个或更多个GNSS传感器1258所识别。在至少一个实施例中,当在欧洲运行时,CNN将寻求检测欧洲警报器,而在美国时,CNN将寻求仅识别北美警报器。在至少一个实施例中,一旦检测到紧急车辆,就可以在一个或更多个超声波传感器1262的辅助下使用控制程序来执行紧急车辆安全例程、减速车辆、将车辆驶至路边、停车、和/或使车辆闲置,直到紧急车辆通过。

[0193] 在至少一个实施例中,车辆1200可以包括一个或更多个CPU 1218(例如,一个或更多个离散CPU或一个或更多个dCPU),其可以经由高速互连(例如PCIe)耦合到一个或更多个SoC 1204。在至少一个实施例中,一个或更多个CPU 1218可以包括X86处理器,例如一个或更多个CPU 1218可用于执行各种功能中的任何功能,例如包括在ADAS传感器和一个或更多个SoC 1204之间潜在的仲裁不一致的结果,和/或一个或更多个监控控制器1236的状态和健康和/或片上信息系统(“信息SoC”)1230。

[0194] 在至少一个实施例中,车辆1200可以包括一个或更多个GPU 1220(例如,一个或更

多个离散GPU或一个或更多个dGPU),其可以经由高速互连(例如NVIDIA的NVLINK)耦合到一个或更多个SoC 1204。在至少一个实施例中,一个或更多个GPU 1220可以提供附加的人工智能功能,例如通过执行冗余和/或不同的神经网络,并且可以至少部分地基于来自车辆1200的传感器的输入(例如,传感器数据)来用于训练和/或更新神经网络。

[0195] 在至少一个实施例中,车辆1200可以进一步包括网络接口1224,其可以包括但不限于一个或更多个无线天线1226(例如,用于不同通信协议的一个或更多个无线天线1226,诸如蜂窝天线、蓝牙天线等)。在至少一个实施例中,网络接口1224可以用于使能通过云上互联网(例如,采用服务器和/或其他网络设备)与其他车辆和/或计算设备(例如乘客的客户端设备)的无线连接。在至少一个实施例中,为了与其他车辆通信,可以在车辆1200和其他车辆之间建立直接链路和/或可以建立间接链路(例如,通过网络和互联网)。在至少一个实施例中,可以使用车辆到车辆的通信链路来提供直接链路。车辆到车辆的通信链路可以向车辆1200提供关于车辆1200附近的车辆的信息(例如,车辆1200前面、侧面和/或后面的车辆)。在至少一个实施例中,该前述功能可以是车辆1200的协作自适应巡航控制功能的一部分。

[0196] 在至少一个实施例中,网络接口1224可以包括SoC,其提供调制和解调功能并使一个或更多个控制器1236能够通过无线网络进行通信。在至少一个实施例中,网络接口1224可以包括射频前端,用于从基带到射频的上转换以及从射频到基带的下转换。在至少一个实施例中,可以以任何技术上可行的方式执行频率转换。例如,可以通过公知的过程和/或使用超外差过程来执行频率转换。在至少一个实施例中,射频前端功能可以由单独的芯片提供。在至少一个实施例中,网络接口可以包括用于通过LTE、WCDMA、UMTS、GSM、CDMA2000、蓝牙、蓝牙LE、Wi-Fi、Z-Wave、ZigBee、LoRaWAN和/或其他无线协议进行通信的无线功能。

[0197] 在至少一个实施例中,车辆1200可以进一步包括一个或更多个数据存储1228,其可以包括但不限于片外(例如,一个或更多个SoC 1204)存储。在至少一个实施例中,一个或更多个数据存储1228可以包括但不限于一个或更多个存储元件,包括RAM、SRAM、动态随机存取存储器(“DRAM”)、视频随机存取存储器(“VRAM”)、闪存、硬盘和/或其他组件和/或可以存储至少一位数据的设备。

[0198] 在至少一个实施例中,车辆1200可以进一步包括一个或更多个GNSS传感器1258(例如,GPS和/或辅助GPS传感器),以辅助地图绘制、感知、占用光栅生成和/或路径规划功能。在至少一个实施例中,可以使用任何数量的GNSS传感器1258,包括例如但不限于使用具有以太网的USB连接器连接到串行接口(例如RS-232)桥的GPS。

[0199] 在至少一个实施例中,车辆1200可以进一步包括一个或更多个RADAR传感器1260。一个或更多个RADAR传感器1260可以由车辆1200用于远程车辆检测,即使在黑暗和/或恶劣天气条件下。在至少一个实施例中,RADAR功能安全级别可以是ASIL B。一个或更多个RADAR传感器1260可以使用CAN和/或总线1202(例如,以传输由一个或更多个RADAR传感器1260生成的数据)来进行控制和访问对象跟踪数据,在某些示例中可以访问以太网以访问原始数据。在至少一个实施例中,可以使用各种各样的RADAR传感器类型。例如但不限于,RADAR传感器1260中的一个或更多个传感器可适合于前、后和侧面RADAR使用。在至少一个实施例中,一个或更多个RADAR传感器1260是脉冲多普勒RADAR传感器。

[0200] 在至少一个实施例中,一个或更多个RADAR传感器1260可以包括不同的配置,例如

具有窄视野的远程、具有宽事业的近程、近程侧面覆盖等。在至少一个实施例中，远程RADAR可以用于自适应巡航控制功能。在至少一个实施例中，远程RADAR系统可以提供通过两次或更多次独立扫描（例如在250m范围内）实现的宽广的视野。在至少一个实施例中，一个或更多个RADAR传感器1260可以帮助在静态对象和运动对象之间区分，并且可以被ADAS系统1238用于紧急制动辅助和向前碰撞警告。包括在远程RADAR系统中的一个或更多个传感器1260可以包括但不限于具有多个（例如六个或更多个）固定RADAR天线以及高速CAN和FlexRay接口的单基地多模式RADAR。在至少一个实施例中，具有六个天线、中央四个天线可以创建聚焦的波束图，该波束图设计为以较高的速度记录车辆1200的周围环境，而相邻车道的交通干扰最小。在至少一个实施例中，其他两个天线可以扩大视野，从而可以快速检测进入或离开车辆1200的车道。

[0201] 在至少一个实施例中，作为示例，中程RADAR系统可包括例如高达160m（前）或80m（后）的范围，以及高达42度（前）或150度（后）的视野。在至少一个实施例中，短程RADAR系统可以包括但不限于设计成安装在后保险杠的两端的任意数量的RADAR传感器1260。当安装在后保险杠的两端时，在至少一个实施例中，RADAR传感器系统可以产生两个光束，该两个光束不断地监测车辆后面和附近的盲点。在至少一个实施例中，短程RADAR系统可以在ADAS系统1238中用于盲点检测和/或车道改变辅助。

[0202] 在至少一个实施例中，车辆1200可以进一步包括一个或更多个超声传感器1262。可以定位在车辆1200的前、后和/或两侧的一个或更多个超声传感器1262可以用于停车辅助和/或创建和更新占用光栅。在至少一个实施例中，可以使用各种各样的超声传感器1262，并且可以将不同的超声传感器1262用于不同的检测范围（例如2.5m、4m）。在至少一个实施例中，超声传感器1262可以在ASIL B的功能安全级别下操作。

[0203] 在至少一个实施例中，车辆1200可以包括一个或更多个LIDAR传感器1264。一个或更多个LIDAR传感器1264可以用于对象和行人检测、紧急制动、避免碰撞和/或其他功能。在至少一个实施例中，一个或更多个LIDAR传感器1264可以是功能安全级别ASIL B。在至少一个实施例中，车辆1200可以包括可以使用以太网的多个（例如，两个、四个、六个等）LIDAR传感器1264（例如，将数据提供给千兆以太网交换机）。

[0204] 在至少一个实施例中，一个或更多个LIDAR传感器1264可能能够提供针对360度视野的对象及其距离的列表。在至少一个实施例中，市售的一个或更多个LIDAR传感器1264例如可以具有大约100m的广告范围，具有2cm-3cm的精度，并且支持100Mbps的以太网连接。在至少一个实施例中，可以使用一个或更多个非突出的LIDAR传感器1264。在这样的实施例中，一个或更多个LIDAR传感器1264可以被实现为可以嵌入到车辆1200的前、后、侧面和/或角落位置中的小型设备。在至少一个实施例中，一个或更多个LIDAR传感器1264，在这样的实施例中，即使对于低反射率的对象，也可以提供高达120度的水平视野和35度的垂直视野，并且具有200m的范围。在至少一个实施例中，可将前向一个或更多个LIDAR传感器1264配置为用于45度至135度之间的水平视野。

[0205] 在至少一个实施例中，也可以使用LIDAR技术（诸如3D闪光LIDAR）。3D闪光LIDAR使用激光闪光作为传输源，以照亮车辆1200周围大约200m。在至少一个实施例中，闪光LIDAR单元包括但不限于接收器，该接收器记录激光脉冲传播时间和每个像素上的反射光，该像素又对应于从车辆1200到对象的范围。在至少一个实施例中，闪光LIDAR可以允许利用每个

激光闪光来生成周围环境的高度准确且无失真的图像。在至少一个实施例中,可以部署四个闪光LIDAR传感器,在车辆1200的每一侧部署一个传感器。在至少一个实施例中,3D闪光LIDAR系统包括但不限于除了风扇(例如非扫描LIDAR设备)以外没有移动部件的固态3D视线阵列LIDAR相机。在至少一个实施例中,闪光LIDAR设备可以每帧使用5纳秒的I类(人眼安全)激光脉冲,并且可以捕获反射激光,以3D测距点云和共同登记的强度数据的形式。

[0206] 在至少一个实施例中,车辆1200还可包括一个或更多个IMU传感器1266。在至少一个实施例中,一个或更多个IMU传感器1266可位于车辆1200的后轴中心,在至少一个实施例中。在至少一个实施例中,一个或更多个IMU传感器1266可以包括,例如但不限于,一个或更多个加速度计、一个或更多个磁力计、一个或更多个陀螺仪、一个磁罗盘、多个磁罗盘和/或其他传感器类型。在至少一个实施例中,例如在六轴应用程序中,一个或更多个IMU传感器1266可以包括但不限于加速度计和陀螺仪。在至少一个实施例中,例如在九轴应用程序中,一个或更多个IMU传感器1266可以包括但不限于加速度计、陀螺仪和磁力计。

[0207] 在至少一个实施例中,一个或更多个IMU传感器1266可以实现为结合了微机电系统(“MEMS”)惯性传感器,高灵敏度GPS接收器和先进的卡尔曼滤波算法的微型高性能GPS辅助惯性导航系统(“GPS/INS”),以提供位置、速度和姿态的估算;在至少一个实施例中,一个或更多个IMU传感器1266可使车辆1200估算航向而无需来自磁传感器通过直接观测和关联从GPS到一个或更多个IMU传感器1266的速度变化来实现的输入。在至少一个实施例中,一个或更多个IMU传感器1266和一个或更多个GNSS传感器1258可以组合在单个集成单元中。

[0208] 在至少一个实施例中,车辆1200可以包括放置在车辆1200内和/或周围的一个或更多个麦克风1296。在至少一个实施例中,此外,一个或更多个麦克风1296可以用于紧急车辆检测和识别。

[0209] 在至少一个实施例中,车辆1200可以进一步包括任何数量的相机类型,包括一个或更多个立体相机1268、一个或更多个广角相机1270、一个或更多个红外相机1272、一个或更多个环绕相机1274、一个或更多个远程相机1298、一个或更多个中程相机1276和/或其他相机类型。在至少一个实施例中,相机可用于捕获车辆1200的整个外围周围的图像数据。在至少一个实施例中,所使用的相机的类型取决于车辆1200。在至少一个实施例中,相机类型的任何组合可以是用于在车辆1200周围提供必要覆盖范围。在至少一个实施例中,相机的数量可以根据实施例而不同。例如,在至少一个实施例中,车辆1200可以包括六个相机、七个相机、十个相机、十二个相机或其他数量的相机。相机可以作为示例但不限于支持千兆位多媒体串行链路(“GMSL”)和/或千兆位以太网。在至少一个实施例中,本文先前参照图12A和图12B可以更详细地描述了每个相机。

[0210] 在至少一个实施例中,车辆1200可以进一步包括一个或更多个振动传感器1242。一个或更多个振动传感器1242可以测量车辆1200的部件(例如,轴)的振动。例如,在至少一个实施例中,振动的变化可以指示路面的变化。在至少一个实施例中,当使用两个或更多个振动传感器1242时,振动之间的差异可以用于确定路面的摩擦或打滑(例如,当在动力驱动轴和自由旋转轴之间存在振动差异时)。

[0211] 在至少一个实施例中,车辆1200可以包括ADAS系统1238。ADAS系统1238可以包括但不限于SoC。在至少一个实施例中,ADAS系统1238可以包括但不限于任何数量的自主/自适应/自动巡航控制(“ACC”)系统、协作自适应巡航控制(“CACC”)系统、前撞警告(“FCW”)系

统、自动紧急制动 (“AEB”) 系统、车道偏离警告 (“LDW”) 系统、车道保持辅助 (“LKA”) 系统、盲区警告 (“BSW”) 系统、后方交叉交通警告 (“RCTW”) 系统、碰撞警告 (“CW”) 系统、车道对中 (“LC”) 系统和其他系统、特征和/或功能及其组合。

[0212] 在至少一个实施例中, ACC系统可以使用一个或更多个RADAR传感器1260、一个或更多个LIDAR传感器1264和/或任何数量的相机。在至少一个实施例中, ACC系统可以包括纵向ACC系统和/或横向ACC系统。在至少一个实施例中, 纵向ACC系统监控并控制到紧邻车辆1200的车辆的距离, 并自动调节车辆1200的速度以保持与前方车辆的安全距离。在至少一个实施例中, 横向ACC系统执行距离保持, 并在需要时建议车辆1200改变车道。在至少一个实施例中, 横向ACC与其他ADAS应用程序有关, 例如LC和CW。

[0213] 在至少一个实施例中, CACC系统使用来自其他车辆的信息, 该信息可以经由网络接口1224和/或一个或更多个无线天线1226从其他车辆接收经由无线链路或者间接经由网络连接 (例如, 经由互联网) 接收。在至少一个实施例中, 直接链路可以由车辆到车辆 (“V2V”) 的通信链路提供, 而间接链路可以由基础设施到车辆 (“I2V”) 的通信链路提供。通常, V2V通信概念提供关于紧接在前的车辆 (例如, 紧接在车辆1200之前并与之在同一车道上的车辆) 的信息, 而I2V通信概念提供关于更前方交通的信息。在至少一个实施例中, CACC系统可以包括I2V和V2V信息源之一或两者。在至少一个实施例中, 在给定车辆1200之前的车辆的信息的情况下, CACC系统可以更可靠, 并且具有改善交通流的平滑度并减少道路拥堵的潜力。

[0214] 在至少一个实施例中, FCW系统被设计成警告驾驶员危险, 如此驾驶员可以采取纠正措施。在至少一个实施例中, FCW系统使用前向相机和/或一个或更多个RADAR传感器1260, 其耦合至专用处理器、DSP、FPGA和/或ASIC, 其电耦合至驾驶员反馈, 例如显示器、扬声器和/或振动组件。在至少一个实施例中, FCW系统可以提供警告, 例如以声音、视觉警告, 振动和/或快速制动脉冲的形式。

[0215] 在至少一个实施例中, AEB系统检测到与另一车辆或其他对象的即将发生的向前碰撞, 并且如果驾驶员在指定的时间或距离参数内未采取纠正措施, 则可以自动施加制动。在至少一个实施例中, AEB系统可以使用耦合到专用处理器、DSP、FPGA和/或ASIC的一个或更多个前向相机和/或一个或更多个RADAR传感器1260。在至少一个实施例中, 当AEB系统检测到危险时, AEB系统通常首先警告驾驶员采取纠正措施以避免碰撞, 并且, 如果驾驶员没有采取纠正措施, 则AEB系统可以自动施加制动器以试图防止或至少减轻预测碰撞的影响。在至少一个实施例中, AEB系统可以包括诸如动态制动器支持和/或即将发生碰撞的制动的技术。

[0216] 在至少一个实施例中, 当车辆1200越过车道标记时, LDW系统提供视觉、听觉和/或触觉警告, 例如方向盘或座椅振动, 以警告驾驶员。在至少一个实施例中, 当驾驶员通过激活转向信号灯指示有意的车道偏离时, LDW系统不活跃。在至少一个实施例中, LDW系统可以使用耦合到专用处理器、DSP、FPGA和/或ASIC的面向正面的相机, 其被电耦合到诸如显示器、扬声器和/或振动组件之类的驾驶员反馈。在至少一个实施例中, LKA系统是LDW系统的一种变型。如果车辆1200开始离开车道, LKA系统提供转向输入或制动以校正车辆1200。

[0217] 在至少一个实施例中, BSW系统检测并警告汽车盲区中的车辆驾驶员。在至少一个实施例中, BSW系统可以提供视觉、听觉和/或触觉警报, 以指示合并或改变车道是不安全

的。在至少一个实施例中，当驾驶员使用转向灯时，BSW系统可以提供附加警告。在至少一个实施例中，BSW系统可以使用耦合到专用处理器、DSP、FPGA和/或ASIC的一个或多个朝后侧的相机和/或一个或多个RADAR传感器1260，其电耦合到驾驶员反馈，例如显示器、扬声器和/或振动组件。

[0218] 在至少一个实施例中，当在车辆1200倒车时在后相机范围之外检测到对象时，RCTW系统可以提供视觉、听觉和/或触觉通知。在至少一个实施例中，RCTW系统包括AEB系统，以确保应用程序车辆制动器以避免碰撞。在至少一个实施例中，RCTW系统可以使用一个或多个面向后方的RADAR传感器1260，其耦合到专用处理器、DSP、FPGA和/或ASIC，其被电耦合到诸如显示器、扬声器和/或振动组件之类的驾驶员反馈。

[0219] 在至少一个实施例中，常规的ADAS系统可能易于产生误报结果，这可能使驾驶员烦恼和分散注意力，但通常不是灾难性的，因为常规的ADAS系统会警告驾驶员并允许驾驶员决定安全状况是否真正存在并采取相应动作。在至少一个实施例中，在结果冲突的情况下，车辆1200本身决定是否听从主计算机或副计算机（例如，控制器1236的第一控制器1236或第二控制器1236）的结果。例如，在至少一个实施例中，ADAS系统1238可以是用于将感知信息提供给备份计算机合理性模块的备用和/或辅助计算机。在至少一个实施例中，备用计算机合理性监控器可以在硬件组件上运行冗余的各种软件，以检测感知和动态驾驶任务中的故障。在至少一个实施例中，可以将来自ADAS系统1238的输出提供给监控MCU。在至少一个实施例中，如果主计算机和辅助计算机的输出冲突，监督MCU决定如何协调冲突以确保安全操作。

[0220] 在至少一个实施例中，主计算机可以配置为向监督MCU提供置信度分数，以指示主计算机对所选结果的置信度。在至少一个实施例中，如果置信度得分超过阈值，则监督MCU可以遵循主计算机的指示，而不管辅助计算机是否提供冲突或不一致的结果。在至少一个实施例中，在置信度得分不满足阈值的情况下，并且在主计算机和辅助计算机指示不同的结果（例如，冲突）的情况下，监督MCU可以在计算机之间仲裁以确定适当的结果。

[0221] 在至少一个实施例中，监督MCU可以配置为运行神经网络，该神经网络被训练和配置为至少部分地基于来自主计算机和辅助计算机的输出来确定该辅助计算机提供错误警报的条件。在至少一个实施例中，监督MCU中的神经网络可以学习何时可以信任辅助计算机的输出，以及何时不能信任。例如，在至少一个实施例中，当辅助计算机是基于RADAR的FCW系统时，监督MCU中的神经网络可以学习FCW系统何时识别实际上不是危险的金属对象，例如会触发警报的排水格栅或井盖。在至少一个实施例中，当辅助计算机是基于相机的LDW系统时，当存在骑自行车的人或行人并且实际上车道偏离是最安全的操作时，监督MCU中的神经网络可以学会覆盖LDW。在至少一个实施例中，监督MCU可以包括适合于运行具有相关联的存储器的神经网络的DLA或GPU中的至少一个。在至少一个实施例中，监督MCU可以包括和/或被包括为一个或多个SoC 1204的组件。

[0222] 在至少一个实施例中，ADAS系统1238可以包括使用传统的计算机视觉规则执行ADAS功能的辅助计算机。在至少一个实施例中，辅助计算机可以使用经典计算机视觉规则（如果-则），并且监督MCU中的神经网络的存在可以提高可靠性、安全性和性能。例如，在至少一个实施例中，多样化的实现方式和有意的非同一性使得整个系统更加容错，尤其是对于由软件（或软件-硬件接口）功能引起的故障。例如，在至少一个实施例中，如果在主计算

机上运行的软件中存在软件漏洞或错误,并且在辅助计算机上运行的不相同的软件代码提供了相同的总体结果,则监督MCU可以更有把握地认为总体结果是正确,并且主计算机上的软件或硬件中的漏洞不会导致重大错误。

[0223] 在至少一个实施例中,可以将ADAS系统1238的输出输入到主计算机的感知模块和/或主计算机的动态驾驶任务模块中。例如,在至少一个实施例中,如果ADAS系统1238由于正前方的对象而指示向前碰撞警告,则感知块可以在识别对象时使用该信息。在至少一个实施例中,如本文所述,辅助计算机可以具有其自己的神经网络,该神经网络经过训练从而降低了误报的风险。

[0224] 在至少一个实施例中,车辆1200可以进一步包括信息娱乐SoC 1230(例如,车载信息娱乐系统(IVI))。尽管被示出和描述为SoC,但是在至少一个实施例中,信息娱乐系统1230可以不是SoC,并且可以包括但不限于两个或更多个分立组件。在至少一个实施例中,信息娱乐SoC 1230可以包括但不限于硬件和软件的组合,其可以用于提供音频(例如,音乐、个人数字助理、导航指令、新闻、广播等)、视频(例如,电视、电影、流媒体等)、电话(例如,免提通话)、网络连接(例如,LTE、WiFi等)和/或信息服务(例如,导航系统、后停车辅助、无线电数据系统、与车辆相关的信息,例如燃油水平、总覆盖距离、制动燃油水平、油位、车门打开/关闭、空气滤清器信息等)到车辆。例如,信息娱乐SoC 1230可以包括收音机、磁盘播放器、导航系统、视频播放器、USB和蓝牙连接、汽车、车载娱乐系统、WiFi、方向盘音频控制、免提语音控制、抬头显示器(“HUD”)、HMI显示器1234、远程信息处理设备、控制面板(例如,用于控制各种组件、特征和/或系统和/或与之交互)和/或其他组件。在至少一个实施例中,信息娱乐SoC 1230可以进一步用于向车辆的用户提供信息(例如,视觉和/或听觉的),诸如来自ADAS系统1238的信息、自动驾驶信息(诸如计划的车辆操纵)、轨迹、周围环境信息(例如,交叉路口信息、车辆信息、道路信息等)和/或其他信息。

[0225] 在至少一个实施例中,信息娱乐SoC 1230可以包括任何数量和类型的GPU功能。在至少一个实施例中,信息娱乐SoC 1230可以通过总线1202(例如CAN总线、以太网等)与车辆1200的其他设备、系统和/或组件通信。在至少一个实施例中,信息娱乐SoC 1230可以是耦合到监控MCU,使得信息娱乐系统的GPU可以在主控制器1236(例如,车辆1200的主计算机和/或备用计算机)发生故障的情况下执行一些自动驾驶功能。在至少一个实施例中,信息娱乐SoC 1230可以使车辆1200进入司机到安全停止模式,如本文所述。

[0226] 在至少一个实施例中,车辆1200可以进一步包括仪表板1232(例如,数字仪表板、电子仪表板、数字仪表操纵板等)。仪表板1232可以包括但不限于控制器和/或超级计算机(例如,离散控制器或超级计算机)。在至少一个实施例中,仪表板1232可以包括但不限于一组仪表的任何数量和组合,例如车速表、燃料水平、油压、转速表、里程表、转弯指示器、换挡位置指示器、一个或更多个安全带警告灯、一个或更多个驻车制动警告灯、一个或更多个发动机故障灯、辅助约束系统(例如安全气囊)信息、照明控件、安全系统控件、导航信息等。在某些示例中,信息可能是在信息娱乐SoC 1230和仪表板1232之间显示和/或共享。在至少一个实施例中,仪表板1232可以被包括作为信息娱乐SoC 1230的一部分,反之亦然。

[0227] 推理和/或训练逻辑915用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图9A和/或图9B提供关于推理和/或训练逻辑915的细节。在至少一个实施例中,推理和/或训练逻辑915可以在系统图12C中用于至少部分地基于使用神经网络训练操

作\神经网络功能和/或架构或本文所述的神经网络用例计算出的权重参数来推理或预测操作。

[0228] 在至少一个实施例中,空间自适应可分离卷积层7可以和系统图12C一起用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构、或本文所述的神经网络用例计算的权重参数来推理和预测操作。

[0229] 图12D是根据至少一个实施例的在基于云的服务器与图12A的自动驾驶车辆1200之间进行通信的系统1276的图。在至少一个实施例中,系统1276可以包括但不限于一个或更多个服务器1278、一个或更多个网络1290以及任何数量和类型的车辆,包括车辆1200。一个或更多个服务器1278可以包括但不限于,多个GPU 1284 (A) -1284 (H) (在本文中统称为GPU 1284)、PCIe交换机1282 (A) -1282 (H) (在本文中统称为PCIe交换机1282),和/或CPU 1280 (A) -1280 (B) (在本文中统称为CPU 1280)。GPU 1284、CPU 1280和PCIe交换机1282可以与高速连接线互连,例如但不限于,由NVIDIA开发的NVLink接口1288和/或PCIe连接1286。在至少一个实施例中,GPU 1284通过NVLink和/或NVSwitchSoC连接,GPU 1284和PCIe交换机1282通过PCIe互连连接。在至少一个实施例中,尽管示出了八个GPU 1284、两个CPU 1280和四个PCIe交换机1282,但这并不旨在进行限制。在至少一个实施例中,一个或更多个服务器1278中的每一个可以包括但不限于任意数量的GPU 1284、CPU 1280和/或PCIe交换机1282的任何组合。例如,在至少一个实施例中,一个或更多个服务器1278可各自包括八个、十六个、三十二个和/或更多个GPU 1284。

[0230] 在至少一个实施例中,一个或更多个服务器1278可以通过一个或更多个网络1290并从车辆接收表示图像的图像数据,该图像示出了意外的或改变的道路状况,例如最近开始的道路工程。在至少一个实施例中,一个或更多个服务器1278可以通过一个或更多个网络1290并且向车辆传输经更新的神经网络1292,和/或地图信息1294,包括但不限于关于交通和道路状况的信息。在至少一个实施例中,对地图信息1294的更新可以包括但不限于对HD地图1222的更新,例如关于建筑工地、坑洼、便道、洪水和/或其他障碍物的信息。在至少一个实施例中,神经网络1292、经更新的神经网络1292和/或地图信息1294可能是由从环境中的任何数量的车辆接收的数据中表示的新训练和/或经验产生的,和/或至少基于在数据中心执行的训练(例如,使用一个或更多个服务器1278和/或其他服务器)。

[0231] 在至少一个实施例中,一个或更多个服务器1278可以用于至少部分地基于训练数据来训练机器学习模型(例如,神经网络)。训练数据可以由车辆产生,和/或可以在模拟中产生(例如,使用游戏引擎)。在至少一个实施例中,标记任何数量的训练数据(例如,在相关的神经网络受益于监督学习的情况下)和/或经历其他预处理。在至少一个实施例中,没有对任何数量的训练数据进行标记和/或预处理(例如,在相关联的神经网络不需要监督学习的情况下)。在至少一个实施例中,一旦机器学习模型被训练,机器学习模型就可以被车辆使用(例如,通过一个或更多个网络1290传输到车辆,和/或机器学习模型可以被一个或更多个服务器1278使用以远程监控车辆)。

[0232] 在至少一个实施例中,一个或更多个服务器1278可以从车辆接收数据并且将数据应用程序于最新的实时神经网络以用于实时智能推理。在至少一个实施例中,一个或更多个服务器1278可以包括由一个或更多个GPU 1284供电的深度超级计算机和/或专用AI计算机,例如由NVIDIA开发的DGX和DGX Station机器。然而,在至少一个实施例中,一个或

更多服务器1278可以包括使用CPU供电的数据中心的深度学习基础设施。

[0233] 在至少一个实施例中,一个或多个服务器1278的深度学习基础结构可能能够进行快速、实时的推理,并且可以使用该能力来评估和验证车辆1200中处理器、软件和/或相关硬件的健康。例如,在至少一个实施例中,深度学习基础设施可以从车辆1200接收周期性更新,例如车辆1200在该图像序列中所定位的图像序列和/或对象(例如,通过计算机视觉和/或其他机器学习对象分类技术)。在至少一个实施例中,深度学习基础设施可以运行其自己的神经网络以识别对象并将它们与车辆1200所识别的对象进行比较,并且,如果结果不匹配和深度学习基础设施断定车辆1200中的AI正在发生故障,则一个或多个服务器1278可以将信号发送到车辆1200,以指示车辆1200的故障安全计算机采取控制、通知乘客并完成安全停车操作。

[0234] 在至少一个实施例中,一个或多个服务器1278可以包括一个或多个GPU 1284和一个或多个可编程推理加速器(例如NVIDIA的TensorRT 3)。在至少一个实施例中,GPU驱动的服务器和推理加速的组合可以使实时响应成为可能。在至少一个实施例中,例如在性能不太关键的情况下,可以将由CPU、FPGA和其他处理器驱动的服务器用于推理。在至少一个实施例中,硬件结构915用于执行一个或多个实施例。本文结合图9A和/或图9B提供关于硬件结构915的细节。

[0235] 计算机系统

[0236] 图13是示出根据至少一个实施例示例性计算机系统的框图,该示例性计算机系统可以是具有互连的设备和组件的系统,片上系统(SOC)或它们的某种形成有处理器的组合1300,该处理器可以包括执行单元以执行指令。在至少一个实施例中,根据本公开,例如本文所述的实施例,计算机系统1300可以包括但不限于组件,例如处理器1302,其执行单元包括逻辑以执行用于过程数据的算法。在至少一个实施例中,计算机系统1300可以包括处理器,例如可从加利福尼亚圣塔克拉拉的英特尔公司(Intel Corporation of Santa Clara, California)获得的PENTIUM®处理器家族、Xeon™、Itanium®、XScale™和/或StrongARM™,Intel®Core™或Intel®Nervana™微处理器,尽管也可以使用其他系统(包括具有其他微处理器的PC、工程工作站、机顶盒等)。在至少一个实施例中,计算机系统1300可以执行可从华盛顿州雷蒙德市的微软公司(Microsoft Corporation of Redmond, Wash.)获得的WINDOWS操作系统版本,尽管其他操作系统(例如UNIX和Linux)、嵌入式软件和/或图形用户界面也可以使用。

[0237] 实施例可以用在其他设备中,例如手持设备和嵌入式应用程序。手持设备的一些示例包括蜂窝电话、互联网协议(Internet Protocol)设备、数码相机、个人数字助理(“PDA”)和手持PC。在至少一个实施例中,嵌入式应用程序可以包括微控制器、数字信号处理器(“DSP”)、片上系统、网络计算机(“NetPC”)、机顶盒、网络集线器、广域网(“WAN”)交换机,或根据至少一个实施例可以执行一个或多个指令的任何其他系统。

[0238] 在至少一个实施例中,计算机系统1300可包括但不限于处理器1302,该处理器1302可包括但不限于一个或多个执行单元1308,以根据本文描述的技术执行机器学习模型训练和/或推理。在至少一个实施例中,系统13是单处理器台式机或服务器系统,但是在另一实施例中,系统13可以是多处理器系统。在至少一个实施例中,处理器1302可以包括但不限于复杂指令集计算机(“CISC”)微处理器、精简指令集计算(“RISC”)微处理器、超长指

令字 (“VLIW”) 微处理器、实现指令集组合的处理器, 或任何其他处理器设备, 例如数字信号处理器。在至少一个实施例中, 处理器1302可以耦合到处理器总线1310, 该处理器总线1310可以在处理器1302与计算机系统1300中的其他组件之间传输数据信号。

[0239] 在至少一个实施例中, 处理器1302可以包括但不限于1级 (“L1”) 内部高速缓存存储器 (“cache”) 1304。在至少一个实施例中, 处理器1302可以具有单个内部高速缓存或多级内部缓存。在至少一个实施例中, 高速缓存存储器可以驻留在处理器1302的外部。根据特定的实现和需求, 其他实施例也可以包括内部和外部高速缓存的组合。在至少一个实施例中, 寄存器文件1306可以在各种寄存器中存储不同类型的数据, 包括但不限于整数寄存器、浮点寄存器、状态寄存器和指令指针寄存器。

[0240] 在至少一个实施例中, 包括但不限于执行整数和浮点运算的逻辑的执行单元1308, 其也位于处理器1302中。处理器1302还可以包括微码 (“ucode”) 只读存储器 (“ROM”), 用于存储某些宏指令的微代码。在至少一个实施例中, 执行单元1308可以包括用于处理封装指令集1309的逻辑。在至少一个实施例中, 通过将封装指令集1309包括在通用处理器1302的指令集中, 以及要执行指令的相关电路, 可以使用通用处理器1302中的封装数据来执行许多多媒体应用程序使用的操作。在一个或多个实施例中, 可以通过使用处理器的数据总线的全宽度来在封装的数据上执行操作来加速和更有效地执行许多多媒体应用程序, 这可能不需要在处理器的数据总线上传输较小的数据单元来一次执行一个数据元素的一个或多个操作。

[0241] 在至少一个实施例中, 执行单元1308也可以用在微控制器、嵌入式处理器、图形设备、DSP和其他类型的逻辑电路中。在至少一个实施例中, 计算机系统1300可以包括但不限于存储器1320。在至少一个实施例中, 存储器1320可以被实现为动态随机存取存储器 (“DRAM”) 设备、静态随机存取存储器 (“SRAM”) 设备、闪存设备或其他存储器设备。存储器1320可以存储由处理器1302可以执行的由数据信号表示的指令1319和/或数据1321。

[0242] 在至少一个实施例中, 系统逻辑芯片可以耦合到处理器总线1310和存储器1320。在至少一个实施例中, 系统逻辑芯片可以包括但不限于存储器控制器集线器 (“MCH”) 1316, 并且处理器1302可以经由处理器总线1310与MCH 1316通信。在至少一个实施例中, MCH 1316可以提供到存储器1320的高带宽存储器路径1318以用于指令和数据存储以及用于图形命令、数据和纹理的存储。在至少一个实施例中, MCH 1316可以在处理器1302、存储器1320和计算机系统1300中的其他组件之间启动数据信号, 并且在处理器总线1310、存储器1320和系统I/O 1322之间桥接数据信号。在至少一个实施例中, 系统逻辑芯片可以提供用于耦合到图形控制器的图形端口。在至少一个实施例中, MCH 1316可以通过高带宽存储器路径1318耦合到存储器1320, 并且图形/视频卡1312可以通过加速图形端口 (Accelerated Graphics Port) (“AGP”) 互连1314耦合到MCH 1316。

[0243] 在至少一个实施例中, 计算机系统1300可以使用系统I/O 1322, 所述系统I/O 1322是专有集线器接口总线来将MCH 1316耦合到I/O控制器集线器 (“ICH”) 1330。在至少一个实施例中, ICH 1330可以通过本地I/O总线提供与某些I/O设备的直接连接。在至少一个实施例中, 本地I/O总线可以包括但不限于用于将外围设备连接到存储器1320、芯片组和处理器1302的高速I/O总线。示例可以包括但不限于音频控制器1329、固件集线器 (“Flash BIOS”) 1328、无线收发器1326、数据存储1324、包含用户输入和键盘接口的传统I/O控制器

1323、串行扩展端口1327(例如通用串行总线(USB))和网络控制器1334。数据存储1324可以包括硬盘驱动器、软盘驱动器、CD-ROM设备、闪存设备或其他大容量存储设备。

[0244] 在至少一个实施例中,图13示出了一种系统,该系统包括互连的硬件设备或“芯片”,而在其他实施例中,图13可以示出示例性系统芯片(SoC)。在至少一个实施例中,图cc中所示出的设备可以与专用互连、标准化互连(例如,PCIe)或其某种组合互连。在至少一个实施例中,系统1300的一个或更多个组件使用快速计算链路(CXL)互连来互连。

[0245] 推理和/或训练逻辑915用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图9A和/或9B提供了关于推理和/或训练逻辑915的细节。在至少一个实施例中,推理和/或训练逻辑915可以用于系统图13中,用于至少部分地基于使用在此描述的神经网络训练操作、神经网络功能和/或架构或神经网络使用情况计算的权重参数来推理或预测操作。

[0246] 在至少一个实施例中,空间自适应可分离卷积层7可以和系统图13一起用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构、或本文所述的神经网络用例计算的权重参数来推理和预测操作。

[0247] 图14是示出根据至少一个实施例的用于利用处理器1410的电子设备1400的框图。在至少一个实施例中,电子设备1400可以是例如但不限于笔记本、塔式服务器、机架式服务器、刀片式服务器、膝上型计算机、台式计算机、平板计算机、移动设备、电话、嵌入式计算机或任何其他合适的电子设备。

[0248] 在至少一个实施例中,系统1400可以包括但不限于通信地耦合到任何合适数量或种类的组件、外围设备、模块或设备的处理器1410。在至少一个实施例中,处理器1410使用总线或接口耦合,诸如I²C总线、系统管理总线(“SMBus”)、低引脚计数(LPC)总线、串行外围接口(“SPI”),高清晰度音频(“HDA”)总线、串行高级技术附件(“SATA”)总线、通用串行总线(“USB”) (版本1、2、3)或通用异步接收器/发送器(“UART”)总线。在至少一个实施例中,图14示出了包括互连的硬件设备或“芯片”的系统,而在其他实施例中,图14可以示出示例性系统芯片(SoC)。在至少一个实施例中,图14中所示出的设备可以与专用互连、标准化互连(例如,PCIe)或其某种组合互连。在至少一个实施例中,使用计算快速链路(CXL)互连来互连图14的一个或更多个组件。

[0249] 在至少一个实施例中,图14可包括显示器1424、触摸屏1425、触摸板1430、近场通信单元(“NFC”)1445、传感器集线器1440、热传感器1446、快速芯片组(“EC”)1435、受信平台模块(“TPM”)1438、BIOS/固件/闪存(“BIOS,FW闪存”)1422、DSP 1460、驱动器1420“SSD或HDD”(如固态硬盘(“SSD”)或硬盘驱动器(“HDD”))、无线局域网单元(“WLAN”)1450、蓝牙单元1452、无线广域网单元(“WWAN”)1456、全球定位系统(GPS)1455、相机(“USB 3.0相机”)1454(如USB 3.0相机)或以例如LPDDR3标准实现的低功率双倍数据速率(“LPDDR”)存储器单元(“LPDDR3”)1415。这些组件可以各自以任何合适的方式实现。

[0250] 在至少一个实施例中,其他组件可通过以上讨论的组件通信地耦合到处理器1410。在至少一个实施例中,加速度计1441、环境光传感器(“ALS”)1442、罗盘1443和陀螺仪1444可以通信地耦合至传感器集线器1440。在至少一个实施例中,热传感器1439、风扇1437、键盘1446和触摸板1430可以通信地耦合到EC 1435。在至少一个实施例中,扬声器1463、耳机1464和麦克风(“mic”)1465可以通信地耦合至音频单元(“音频编解码器和d类放

大器”) 1464, 该音频单元又可以通信地耦合至DSP 1460。在至少一个实施例中, 音频单元1464可包括, 例如但不限于, 音频编码器/解码器(“编解码器”)和D类放大器。在至少一个实施例中, SIM卡(“SIM”) 1457可以通信地耦合到WWAN单元1456。在至少一个实施例中, 可以在下一代形状因子(“NGFF”)中实现诸如WLAN单元1450和蓝牙单元1452以及WWAN单元1456之类的组件。

[0251] 推理和/或训练逻辑915用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图9A和/或9B提供了关于推理和/或训练逻辑915的细节。在至少一个实施例中, 推理和/或训练逻辑915可以在图14的系统中用于至少部分地基于使用在此描述的神经网络训练操作、神经网络功能和/或架构或神经网络使用情况计算的权重参数来推理或预测操作。

[0252] 在至少一个实施例中, 空间自适应可分离卷积层7可以和系统图14一起用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构、或本文所述的神经网络用例计算的权重参数来推理和预测操作。

[0253] 图15示出了根据至少一个实施例的计算机系统1500。在至少一个实施例中, 计算机系统1500被配置为实现贯穿本公开所描述的不同过程和方法。

[0254] 在至少一个实施例中, 计算机系统1500包括但不限于至少一个中央处理单元(“CPU”) 1502, 该中央处理单元连接至使用任何合适的协议实现的通信总线1510, 诸如PCI(“外围组件互连”)、快速外围组件互连(“PCI-Express”)、AGP(“加速图形端口”)、超传输或任何其他总线或点对点通信协议。在至少一个实施例中, 计算机系统1500包括但不限于主存储器1504和控制逻辑(例如, 被实现为硬件、软件或其组合), 并且数据被存储在主存储器1504中, 该主存储器可以采取随机存取存储器(“RAM”)的形式。在至少一个实施例中, 网络接口子系统(“网络接口”) 1522提供到其他计算设备和网络的接口, 该接口用于来自系统1500的其他系统接收数据和向具有计算机系统1500的其他系统传输数据。

[0255] 在至少一个实施例中, 在至少一个实施例中, 计算机系统1500包括但不限于输入设备1508、并行处理系统1512、以及可以使用常规阴极射线管(“CRT”)、液晶显示器(“LCD”)、发光二极管(“LED”)、等离子体显示器或其他适合的显示技术实现的显示设备1506。在至少一个实施例中, 从诸如键盘、鼠标、触摸板、麦克风或更多的输入设备1508中接收用户输入。在至少一个实施例中, 前述每个模块可位于单个半导体平台上以形成处理系统。

[0256] 推理和/或训练逻辑915用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图9A和/或9B提供了关于推理和/或训练逻辑915的细节。在至少一个实施例中, 推理和/或训练逻辑915可以用于系统图15中, 用于至少部分地基于使用在此描述的神经网络训练操作、神经网络功能和/或架构或神经网络使用情况计算的权重参数来推理或预测操作。

[0257] 在至少一个实施例中, 空间自适应可分离卷积层7可以和系统图15一起用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构、或本文所述的神经网络用例计算的权重参数来推理和预测操作。

[0258] 图16示出了根据至少一个实施例的计算机系统1600。在至少一个实施例中, 计算机系统1600包括但不限于计算机1610和USB棒1620。在至少一个实施例中, 计算机1610可包

包括但不限于任何数量和类型的处理器(未示出)和存储器(未示出)。在至少一个实施例中,计算机1610包括但不限于服务器、云实例、膝上型计算机和台式计算机。

[0259] 在至少一个实施例中,USB棒1620包括但不限于处理单元1630、USB接口1640和USB接口逻辑1650。在至少一个实施例中,处理单元1630可以是能够执行指令的任何指令执行系统、装置或设备。在至少一个实施例中,处理核心1630可以包括但不限于任何数量和类型的处理核(未示出)。在至少一个实施例中,处理核心1630包括被优化成执行与机器学习相关联的任何量和类型的操作的专用集成电路(“ASIC”)。例如,在至少一个实施例中,处理核心1630是被优化为执行机器学习推理操作的张量处理单元(“TPC”)。在至少一个实施例中,处理核心1630是被优化为执行机器视觉和机器学习推理操作的视觉处理单元(“VPU”)。

[0260] 在至少一个实施例中,USB接口1640可以是任何类型的USB连接器或USB插座。例如,在至少一个实施例中,USB接口1640是用于数据和电力的USB 3.0类型C插座。在至少一个实施例中,USB接口1640是USB 3.0类型A连接器。在至少一个实施例中,USB接口逻辑1650可以包括使得处理单元1630能够经由USB连接器1640与设备(例如,计算机1610)接合的任何量和类型的逻辑。

[0261] 推理和/或训练逻辑915用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图9A和/或图9B提供了关于推理和/或训练逻辑915的细节。在至少一个实施例中,推理和/或训练逻辑915可以用于至少部分地基于使用在此描述的神经网络训练操作、神经网络功能和/或架构、或神经网络使用情况计算的权重参数来推理或预测操作。

[0262] 在至少一个实施例中,空间自适应可分离卷积层7可以和系统图16一起用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构、或本文所述的神经网络用例计算的权重参数来推理和预测操作。

[0263] 图17A示出了示例性架构,其中多个GPU 1710-1713通过高速链路1740-1743(例如,总线/点对点互连等)通信地耦合到多个多核心处理器1705-1706。在一个实施例中,高速链路1740-1743支持4GB/s、30GB/s、80GB/s或更高的通信吞吐量。可以使用各种互连协议,包括但不限于PCIe 4.0或5.0以及NVLink 2.0。

[0264] 此外,在一个实施例中,两个或更多个GPU 1710-1713通过高速链路1729-1730互连,该高速链路可以使用与用于高速链路1740-1743的协议/链路相同或不同的协议/链路来实现。类似地,两个或更多个多核心处理器1705-1706可以通过高速链路1728连接,该高速链路可以是以20GB/s、30GB/s、120GB/s或更高的速度运行的对称多处理器(SMP)总线。可替代地,可以使用相同的协议/链路(例如,通过公共互连结构)来完成图17A中所示的各种系统组件之间的所有通信。

[0265] 在至少一个实施例中,每个多核心处理器1705-1706分别经由存储器互连1726-1727通信地耦合到处理器存储器1701-1702,并且每个GPU 1710-1713分别通过GPU存储器互连1750-1753通信地耦合到GPU存储器1720-1723。存储器互连1726-1727和1750-1753可以利用相同或不同的存储器访问技术。作为示例而非限制,处理器存储器1701-1702和GPU存储器1720-1723可以是易失性存储器,诸如动态随机存取存储器(DRAM)(包括堆叠的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,例如3D XPoint或Nano-Ram。在一个实施例中,处理器存储器1701-1702的某些部分可以是易失性存储器,而另一部分可以是非易失性存储器(例如,使用两级存储器

(2LM) 层次)。

[0266] 如本文所述, 尽管各种心处理器1705-1706和GPU 1710-1713可以分别物理地耦合到特定存储器1701-1702、1720-1723, 可以实现统一存储器架构, 其中相同的虚拟系统地址空间 (也称为“有效地址”空间) 分布在各个物理存储器之间。例如, 处理器存储器1701-1702可以各自包含64GB的系统存储器地址空间, 并且GPU存储器1720-1723可以各自包含32GB的系统存储器地址空间 (在此示例中, 导致总计256GB的可寻址存储器大小)。

[0267] 图17B示出了根据一个示例性实施例的用于多核心处理器1707和图形加速模块1746之间互连的附加细节。图形加速模块1746可以包括集成在线路卡上的一个或更多个GPU芯片, 该线路卡经由高速链路1740耦合到处理器1707。可选择地, 图形加速模块1746可以集成在和处理器1707相同的封装或芯片上。

[0268] 在至少一个实施例中, 图示的处理器1707包括多个核心1760A-1760D, 每个核心都具有转换后备缓冲区1761A-1761D和一个或更多个高速缓存1762A-1762D。在至少一个实施例中, 核心1760A-1760D可以包括未示出的各种其他组件, 用于执行指令和处理数据。高速缓存1762A-1762D可以包括级别1 (L1) 和级别2 (L2) 高速缓存。此外, 一个或更多个共享高速缓存1756可以被包括在高速缓存1762A-1762D中, 并且由各组核心1760A-1760D共享。例如, 处理器1707的一个实施例包括24个核心, 每个核心具有其自己的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。在该实施例中, 两个相邻核心共享一个或更多个L2和L3高速缓存。处理器1707和图形加速模块1746与系统存储器1714连接, 该系统存储器1714可以包括图17A中的处理器存储器1701-1702。

[0269] 通过一致性总线1764经由核心间通信为存储在各个高速缓存1762A-1762D、1756和系统存储器1714中的数据和指令维护一致性。例如, 每个高速缓存可以具有与其相关联的高速缓存一致性逻辑/电路, 以响应于检测到对特定高速缓存行的读取或写入通过一致性总线1764进行通信。在一个实现中, 通过一致性总线1764执行高速缓存监听协议, 以监听(snoop) 高速缓存访问。

[0270] 在一个实施例中, 代理电路1725将图形加速模块1746通信地耦合到一致性总线1764, 从而允许图形加速模块1746作为核心1760A-1760D的对等方参与高速缓存一致性协议。特别地, 接口1735通过高速链路1740 (例如, PCIe总线、NVLink等) 提供到代理电路1725的连接, 并且接口1737将图形加速模块1746连接到链路1740。

[0271] 在一个实现中, 加速器集成电路1736代表图形加速模块的多个图形处理引擎1731、1732, N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1731、1732, N, 可各自包括单独的图形处理单元 (GPU)。可选择地, 图形处理引擎1731、1732, N可以包括GPU内的不同类型的图形处理引擎, 诸如图形执行单元、媒体处理引擎 (例如, 视频编码器/解码器)、采样器和blit引擎。在至少一个实施例中, 图形加速模块1746可以是具有多个图形处理引擎1731-1732, N的GPU, 或者图形处理引擎1731-1732, N可以是集成在通用封装、线路卡或芯片上的各个GPU。

[0272] 在一个实施例中, 加速器集成电路1736包括存储器管理单元 (MMU) 1739, 用于执行各种存储器管理功能, 例如虚拟到物理存储器转换 (也称为有效到真实存储器转换), 还包括用于访问系统存储器1714的存储器访问协议。MMU 1739还可包括转换后备缓冲区 (TLB) (未示出), 用于高速缓存虚拟/有效到物理/真实地址转换。在一个实现中, 高速缓存1738存

储命令和数据,用于图形处理引擎1731-1732,N有效地访问。在至少一个实施例中,将存储在高速缓存1738和图形存储器1733-1734,M中的数据与核心高速缓存1762A-1762D、1756和系统存储器1714保持一致。如前所述,可以经由代表高速缓存1738和图形存储器1731-1732,M的代理电路1725来完成该任务(例如,将与处理器高速缓存1762A-1762D、1756上的高速缓存行的修改/访问有关的更新发送到高速缓存1738,并从高速缓存1738接收更新)。

[0273] 一组寄存器1745存储由图形处理引擎1731-1732,N执行的线程的上下文数据,并且上下文管理电路1748管理线程上下文。例如,上下文管理电路1748可以执行保存和恢复操作,以在上下文切换期间保存和恢复各个线程的上下文(例如,其中保存第一线程并且存储第二线程,以便可以由图形处理引擎执行第二线程)。例如,上下文管理电路1748在上下文切换时,可以将当前寄存器值存储到存储器中的(例如,由上下文指针标识的)指定区域。然后,当返回上下文时可以恢复寄存器值。在一个实施例中,中断管理电路1747接收并处理从系统设备接收的中断。

[0274] 在一个实现中,MMU 1739将来自图形处理引擎1731的虚拟/有效地址转换为系统存储器1714中的真实/物理地址。加速器集成电路1736的一个实施例支持多个(例如,4、8、16)图形加速器模块1746和/或其他加速器设备。图形加速器模块1746可以专用于在处理器1707上执行的单个应用程序,或者可以在多个应用程序之间共享。在一个实施例中,呈现了虚拟化的图形执行环境,其中图形处理引擎1731-1732,N的资源与多个应用程序或虚拟机(VM)共享。在至少一个实施例中,可以基于处理要求和与VM和/或应用程序相关联的优先级,将资源细分为“切片”,其被分配给不同的VM和/或应用程序。

[0275] 在至少一个实施例中,加速器集成电路1736作为图形加速模块1746的系统的桥来执行,并提供地址转换和系统存储器高速缓存服务。另外,加速器集成电路1736可以为主机处理器提供虚拟化设施,以管理图形处理引擎1731-1732的虚拟化、中断和存储器管理。

[0276] 由于图形处理引擎1731-1732,N的硬件资源被明确地映射到主机处理器1707看到的真实地址空间,因此任何主机处理器都可以使用有效地址值直接寻址这些资源。在一个实施例中,加速器集成电路1736的一个功能是物理分离图形处理引擎1731-1732,N,使得它们在系统看来为独立的单元。

[0277] 在至少一个实施例中,一个或更多个图形存储器1733-1734,M分别耦合到每个图形处理引擎1731-1732,N。图形存储器1733-1734,M存储指令和数据,所述指令和数据由每个图形处理引擎1731-1732,N处理。图形存储器1733-1734,M可以是易失性存储器,例如DRAM(包括堆叠的DRAM)、GDDR存储器(例如,GDDR5,GDDR6)或HBM,和/或可以是非易失性存储器,例如3D XPoint或Nano-Ram。

[0278] 在一个实施例中,为了减少链路1740上的数据流量,使用偏置技术以确存储存在图形存储器1733-1734,M中的数据是图形处理引擎1731-1732,N最常使用的,并且最好核心1760A-1760D不使用(至少不经常使用)的数据。类似地,偏置机制试图将核心(并且优选地不是图形处理引擎1731-1732,N)需要的数据保持在核心的高速缓存1762A-1762D、1756核和系统存储器1714中。

[0279] 图17C示出了另一个示例性实施例,其中加速器集成电路1736被集成在处理器1707内。在该实施例中,图形处理引擎1731-1732,N经由接口1737和接口1735(同样可以利用任何形式的总线或接口协议)通过高速链路1740直接与加速器集成电路1736通信。加速

器集成电路1736可以执行与关于图17B描述的操作相同的操作。但是由于它紧密靠近一致性总线1764和高速缓存1762A-1762D、1756,可能具有更高的吞吐量。一个实施例支持不同的编程模型,包括专用进程编程模型(无图形加速模块虚拟化)和共享编程模型(具有虚拟化),所述编程模型可以包括由加速器集成电路1736控制的编程模型和由图形加速模块1746控制的编程模型。

[0280] 在至少一个实施例中,图形处理引擎1731-1732,N专用于单个操作系统下的单个应用程序或进程。在至少一个实施例中,单个应用程序可以将其他应用程序请求汇聚(funnel)到图形处理引擎1731-1732,N,从而在VM/分区内提供虚拟化。

[0281] 在至少一个实施例中,图形处理引擎1731-1732,N可以被多个VM/应用程序分区共享。在至少一个实施例中,共享模型可以使用系统管理程序来虚拟化图形处理引擎1731-1732,N,以允许每个操作系统进行访问。对于没有管理程序的单分区系统,操作系统拥有图形处理引擎1731-1732,N。在至少一个实施例中,操作系统可以虚拟化图形处理引擎1731-1732,N,以提供对每个进程或应用程序的访问。

[0282] 在至少一个实施例中,图形加速模块1746或个体图形处理引擎1731-1732,N使用进程句柄来选择进程元素。在一个实施例中,进程元素被存储在系统存储器1714中,并且可使用本文所述的有效地址到真实地址转换技术来寻址。在至少一个实施例中,进程句柄可以是特定于实现方式的值,其在向图形处理引擎1731-1732,N注册其上下文时提供给主机进程(即,调用系统软件以将进程元素添加到进程元素链接列表)。在至少一个实施例中,进程句柄的较低16位可以是所述进程元素在进程元素链接列表中的偏移量。

[0283] 图17D示出了示例性加速器集成切片1790。如本文所使用的,“切片”包括加速器集成电路1736的处理资源的指定部分。应用程序是系统存储器1714中的有效地址空间1782,其存储进程元素1783。在一个实施例中,响应于来自在处理器1707上执行的应用程序1780的GPU调用1781,存储进程元素1783。进程元素1783包含相应的应用程序1780的进程状态。包含在进程元素1783中的工作描述符(WD) 1784可以是由应用程序请求的单个作业,或者可以包含指向作业队列的指针。在至少一个实施例中,WD 1784是指向应用程序的地址空间1782中的作业请求队列的指针。

[0284] 图形加速模块1746和/或各个图形处理引擎1731-1732,N可以由系统中所有进程或进程子集共享。在至少一个实施例中,可以包括用于设置进程状态并将WD 1784发送到图形加速模块1746以在虚拟化环境中开始作业的基础设施。

[0285] 在至少一个实施例中,专用进程编程模型是特定于实现方式的。在该模型中,单个进程拥有图形加速模块1746或个体图形处理引擎1731。由于图形加速模块1746由单个进程拥有,管理程序初始化用于所拥有的分区的加速器集成电路,当指派了图形加速模块1746时,操作系统初始化用于所拥有的进程的加速器集成电路1736。

[0286] 在操作中,加速器集成切片1790中的WD获取单元1791获取下一个WD 1784,其包括要由图形加速模块1746的一个或更多个图形处理引擎完成的工作的指示。来自WD 1784的数据可以存储在寄存器1745中,并由MMU 1739、中断管理电路1747和/或上下文管理电路1748使用,如图所示。例如,MMU 1739的一个实施例包括用于访问OS虚拟地址空间1785内的段/页表1786的段/页漫游电路。中断管理电路1747可以处理从图形加速模块1746接收的中断事件1792。当执行图形操作时,由图形处理引擎1731-1732,N生成的有效地址1793被MMU

1739转换为实地址。

[0287] 在一个实施例中,为每个图形处理引擎1731-1732,N和/或图形加速模块1746复制相同的一组寄存器1745,并且所述寄存器1745可以由管理程序或操作系统初始化。这些复制的寄存器中的每一个可以被包括在加速器集成切片1790中。可以由管理程序初始化的示例性寄存器在表1中示出。

[0288] 表1-管理程序初始化的寄存器

[0289]

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

[0290] 表2中示出了可由操作系统初始化的示例性寄存器。

[0291] 表2-操作系统初始化的寄存器

[0292]

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) 加速器利用率记录指针
4	虚拟地址 (VA) 存储段表指针
5	权限屏蔽
6	工作描述符

[0293]

[0294] 在一个实施例中,每个WD 1784特定于特定的图形加速模块1746和/或图形处理引擎1731-1732,N。它包含图形处理引擎1731-1732,N完成工作所需的所有信息,或者它可以是指向存储器位置的指针,在该存储器位置应用程序已经设置了要完成的工作的命令队列。

[0295] 图17E示出了共享模型的一个示例性实施例的附加细节。该实施例包括管理程序真实地址空间1798,其中存储了进程元素列表1799。可经由管理程序1796来访问管理程序实地址空间1798,所述管理程序1796虚拟化用于操作系统1795的图形加速模块引擎。

[0296] 在至少一个实施例中,共享编程模型允许来自系统中全部分区或分区子集的全部

进程或进程子集使用图形加速模块1746。存在两种编程模型，其中图形加速模块1746由多个进程和分区共享，时间切片共享和图形定向共享。

[0297] 在该模型中，系统管理程序1796拥有图形加速模块1746，并使其功能可用于所有操作系统1795。对于图形加速模块1746通过系统管理程序1796支持虚拟化，图形加速模块1746可以遵守下列条件：1) 应用程序的作业请求必须是自主的 (即，不需要在作业之间保持状态)，或者图形加速模块1746必须提供上下文保存和恢复机制。2) 图形加速模块1746保证应用程序的作业请求在指定的时间量内完成，包括任何转换错误，或者图形加速模块1746提供了抢占作业处理的能力。3) 在有向共享编程模型中进行操作时，必须确保图形加速模块1746进程之间的公平性。

[0298] 在至少一个实施例中，需要应用程序1780使用图形加速模块1746类型、工作描述符 (WD)、权限屏蔽寄存器 (AMR) 值和上下文保存/恢复区域指针 (CSRP) 进行操作系统1795系统调用。在至少一个实施例中，图形加速模块1746类型描述了用于系统调用的目标加速函数。在至少一个实施例中，图形加速模块1746类型可以是系统特定的值。在至少一个实施例中，WD是专门为图形加速模块1746格式化的，并且可以采用图形加速模块1746命令、指向用户定义的结构的有效地址指针、指向命令队列的有效地址指针的形式，或描述要由图形加速模块1746完成的工作的任何其他数据结构。在一个实施例中，AMR值是用于当前进程的AMR状态。在至少一个实施例中，传递给操作系统的值与设置AMR的应用程序类似。如果加速器集成电路1736和图形加速模块1746的实现不支持用户权限屏蔽覆写寄存器 (UAMOR)，则在管理程序调用中传递AMR之前，操作系统可以将当前UAMOR值应用于AMR值。管理程序1796可以在将AMR放入进程元素1783中之前选择性地应用当前权限屏蔽覆写寄存器 (AMOR) 值。在至少一个实施例中，CSRP是寄存器1745中的一个，所述寄存器包含应用程序的地址空间1782中的区域的有效地址，供图形加速模块1746保存和恢复上下文状态。如果不需要在作业之间保存状态或者当作业被抢占时，则该指针是可选的。在至少一个实施例中，上下文保存/恢复区域可以是固定的系统存储器。

[0299] 在接收到系统调用时，操作系统1795可以验证应用程序1780已经注册并且被授予使用图形加速模块1746的权限。然后，操作系统1795使用表3中所示的信息来调用管理程序1796。

[0300] 表3-操作系统到管理程序的调用参数

[0301]	1	工作描述符 (WD)
	2	权限屏蔽寄存器 (AMR) 值 (可能被屏蔽)
	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) 加速器利用率记录指针 (AURP)
	6	存储段表指针的虚拟地址 (SSTP)
	7	逻辑中断服务号 (LISN)

[0302] 在接收到管理程序调用时，管理程序1796验证操作系统1795已注册并被授予使用图形加速模块1746的权限。然后，管理程序1796将进程元素1783放入相应的图形加速模块1746类型的进程元素链接列表中。进程元素可以包括表4中所示的信息。

[0303] 表4-进程元素信息

[0304]	1	工作描述符 (WD)
	2	权限屏蔽寄存器 (AMR) 值 (可能被屏蔽)
[0305]	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) 加速器利用率记录指针 (AURP)
	6	存储段表指针的虚拟地址 (SSTP)
	7	逻辑中断服务号 (LISN)
	8	从管理程序调用参数派生的中断向量表
	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率记录指针
	12	存储描述符寄存器 (SDR)

[0306] 在至少一个实施例中,管理程序初始化多个加速器集成切片1790寄存器1745。

[0307] 如图17F所示,在至少一个实施例中,使用统一存储器,所述统一存储器可经由用于访问物理处理器存储器1701-1702和GPU存储器1720-1723的公共虚拟存储器地址空间来寻址。在该实现方式中,在GPU 1710-1713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1701-1702,反之亦然,从而简化了可编程性。在一个实施例中,虚拟/有效地址空间的第一部分被分配给处理器存储器1701,第二部分被分配给第二处理器存储器1702,第三部分被分配给GPU存储器1720,以此类推。在至少一个实施例中,整个虚拟/有效存储器空间(有时称为有效地址空间)由此分布在处理器存储器1701-1702和GPU存储器1720-1723的每一个中,从而允许任何处理器或GPU采用映射到任何物理存储器的虚拟地址访问该存储器。

[0308] 在一个实施例中,一个或更多个MMU 1739A-1739E内的偏置/一致性管理电路1794A-1794E确保一个或更多个主机处理器(例如,1705)与GPU 1710-1713的高速缓存之间的高速缓存一致性,并实现指示应在其中存储某些类型的数据的物理存储器的偏置技术。虽然在图17F中示出了偏置/一致性管理电路1794A-1794E的多个实例,但可以在一个或更多个主机处理器1705的MMU内和/或在加速器集成电路1736内实现偏置/一致性电路。

[0309] 一个实施例允许将GPU附属存储器1720-1723映射为系统存储器的一部分,并使用共享虚拟存储器(SVM)技术进行访问,但不会遭受与完整系统高速缓存一致性相关的性能缺陷。在至少一个实施例中,将GPU附属存储器1720-1723作为系统存储器来访问而无需繁重的高速缓存一致性开销的能力为GPU卸载提供了有利的操作环境。该布置允许主机处理器1705软件设置操作数并访问计算结果,而没有传统的I/O DMA数据拷贝的开销。这样的传统拷贝包括驱动程序调用、中断和存储器映射I/O(MMIO)访问,相对于简单的存储器访问而言,这些访问效率均较低。在至少一个实施例中,在没有高速缓存一致性开销的情况下访问

GPU附属存储器1720-1723的能力对于卸载的计算的执行时间可能是关键的。例如,在具有大量流式写入存储器流量的情况下,高速缓存一致性开销可以显著降低GPU 1710-1713所看到的有效写入带宽。在至少一个实施例中,操作数设置的效率、结果访问的效率和GPU计算的效率可能会在确定GPU卸载的有效性方面发挥作用。

[0310] 在至少一个实施例中,GPU偏置和主机处理器偏置的选择由偏置跟踪器数据结构驱动。例如,可以使用偏置表,所述偏置表可以是页面粒度结构(例如,以存储器页面的粒度来控制),该页面粒度结构包括每个GPU附加的存储器页面1或2位。在至少一个实施例中,在GPU 1710-1713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)的情况下,可以在一个或更多个GPU附属存储器1720-1723的被盗存储器范围中实现偏置表。替代地,可以在GPU内维护整个偏置表。

[0311] 在至少一个实施例中,在实际访问GPU存储器之前,访问与对GPU附加存储器1720-1723的每次访问相关联的偏置表条目,从而引起以下操作。首先,来自GPU 1710-1713的在GPU偏置中找到其页面的本地请求被直接转发到对应的GPU存储器1720-1723。来自GPU的在主机偏置中找到其页面的本地请求被转发至处理器1705(例如,通过以上讨论的高速链路)。在一个实施例中,来自处理器1705的在主机处理器偏置中找到所请求页面的请求完成了与正常存储器读取类似的请求。替代地,可以将指向GPU偏置页面的请求转发到GPU 1710-1713。在至少一个实施例中,如果GPU当前不使用页面,则GPU可随后将页面迁移到主机处理器偏置。在至少一个实施例中,页面的偏置状态可通过基于软件的机制、基于硬件辅助的软件的机制、或者在有限的情况下通过纯粹基于硬件的机制来改变。

[0312] 一种用于改变偏置状态的机制采用API调用(例如OpenCL),所述API调用随后调用GPU的设备驱动程序,所述设备驱动程序随后发送消息(或使命令描述符入队)到GPU,引导GPU改变偏置状态,并在某些迁移中在主机中执行高速缓存刷新操作。在至少一个实施例中,高速缓存刷新操作用于从主机处理器1705偏置到GPU偏置的迁移,但是不用于相反的迁移。

[0313] 在一个实施例中,高速缓存一致性是通过暂时渲染主机处理器1705无法高速缓存的GPU偏置页面来维护的。为了访问这些页面,处理器1705可以请求来自GPU 1710的访问,GPU 1710可以或可以不立即授予访问权限。因此,为了减少处理器1705和GPU 1710之间的通信,确保GPU偏置页面是GPU所需的页面而不是主机处理器1705所需的页面是有益的,反之亦然。

[0314] 一个或更多个硬件结构915用于执行一个或更多个实施例。在本文中结合图9A和/或图9B提供关于一个或更多个所述硬件结构915的细节。

[0315] 图18示出了根据本文所述的各个实施例的示例性集成电路和相关联的图形处理器,其可以使用一个或更多个IP核心来制造。除了图示之外,在至少一个实施例中可以包括其他逻辑和电路,包括附加的图形处理器/核心、外围接口控制器或通用处理器核心。

[0316] 图18是示出根据至少一个实施例的可使用一个或更多个IP核心制造的芯片集成电路1800上的示例性系统的框图。在至少一个实施例中,集成电路1800包括一个或更多个应用程序处理器1805(例如,CPU)、至少一个图形处理器1810,并且可以另外包括图像处理器1815和/或视频处理器1820,其中任意一个可能是模块化IP核心。在至少一个实施例中,集成电路1800包括外围或总线逻辑,其包括USB控制器1825、UART控制器1830、SPI/SDIO控

制器1835和sup.2S/I.sup.2C控制器1840。在至少一个实施例中,集成电路1800可以包括显示设备1845耦合到高清多媒体接口(HDMI)控制器1850和移动工业处理器接口(MIPI)显示接口1855中的一个或多个。在至少一个实施例中,存储可以由闪存子系统1860提供,包括闪存和闪存控制器。在至少一个实施例中,可以经由存储器控制器1865提供存储器接口以用于访问SDRAM或SRAM存储器设备。在至少一个实施例中,一些集成电路还包括嵌入式安全引擎1870。

[0317] 推理和/或训练逻辑915用于执行与一个或多个实施例相关联的推理和/或训练操作。本文结合图9A和/或图9B提供关于推理和/或训练逻辑915的细节。在至少一个实施例中,推理和/或训练逻辑915可以在集成电路1800中用于至少部分地基于使用神经网络训练操作、神经网络功能和/或架构或本文描述的神经网络用例计算的权重参数来推理或预测操作。

[0318] 在至少一个实施例中,空间自适应可分离卷积层7可用于集成电路1800中,用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构、或本文所述的神经网络用例计算的权重参数来推理或预测操作。

[0319] 图19A-19B示出了根据本文所述的各个实施例的示例性集成电路和相关联的图形处理器,其可以使用一个或多个IP核心来制造。除了图示之外,在至少一个实施例中可以包括其他逻辑和电路,包括附加的图形处理器/核心、外围接口控制器或通用处理器核心。

[0320] 图19A-19B是示出根据本文描述的实施例的在SoC内使用的示例性图形处理器的框图。图19A示出了根据至少一个实施例的片上系统集成电路的示例性图形处理器1910,其可以使用一个或多个IP核心来制造。图19B示出了根据至少一个实施例的片上系统集成电路的附加示例性图形处理器1940,其可以使用一个或多个IP核心来制造。在至少一个实施例中,图19A的图形处理器1910是低功耗图形处理器核心。在至少一个实施例中,图19B的图形处理器1940是更高性能的图形处理器核心。在至少一个实施例中,每个图形处理器1910、1940可以是图18的图形处理器1810的变体。

[0321] 在至少一个实施例中,图形处理器1910包括顶点处理器1905和一个或多个片段处理器1915A-1915N(例如1915A、1915B、1915C、1915D至1915N-1和1915N)。在至少一个实施例中,图形处理器1910可以经由单独的逻辑来执行不同的着色器程序,使得顶点处理器1905被优化以执行针对顶点着色器程序的操作,而一个或多个片段处理器1915A-1915N执行片段(例如,像素)着色操作用于片段或像素或着色器程序。在至少一个实施例中,顶点处理器1905执行3D图形管线的顶点处理阶段并生成图元和顶点数据。在至少一个实施例中,一个或多个片段处理器1915A-1915N使用由顶点处理器1905生成的图元和顶点数据来生成在显示设备上显示的帧缓冲区。在至少一个实施例中,一个或多个片段处理器1915A-1915N被优化以执行如在OpenGL API中所提供的片段着色器程序,其可以用于执行与在Direct 3D API中所提供的像素着色器程序类似的操作。

[0322] 在至少一个实施例中,图形处理器1910附加地包括一个或多个存储器管理单元(MMU) 1920A-1920B、一个或多个高速缓存1925A-1925B和一个或多个电路互连1930A-1930B。在至少一个实施例中,一个或多个MMU 1920A-1920B提供用于图形处理器1910的虚拟到物理地址的映射,包括用于顶点处理器1905和/或片段处理器1915A-1915N,其可以引用存储在存储器中的顶点或图像/纹理数据,除了存储在一个或多个高速缓存1925A-

1925B中的顶点或图像/纹理数据之外。在至少一个实施例中,一个或更多个MMU 1920A-1920B可以与系统内的其他MMU同步,包括与图18的一个或更多个应用程序处理器1805、图像处理器1815和/或视频处理器1820相关联的一个或更多个MMU,使得每个处理器1805-1820可以参与共享或统一的虚拟存储器系统。在至少一个实施例中,一个或更多个电路互连1930A-1930B使图形处理器1910能够经由SoC的内部总线或经由直接连接与SoC内的其他IP核心相连接。

[0323] 在至少一个实施例中,图形处理器1940包括一个或更多个MMU1920A-1920B,高速缓存1925A-1925B和图19A的图形处理器1910的电路互连1930A-1930B。在至少一个实施例中,图形处理器1940包括一个或更多个着色器核心1955A-1955N(例如,1955A、1955B、1955C、1955D、1955E、1955F到1955N-1和1955N),其提供了统一的着色器核心架构,其中单个核心或类型或核心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。在至少一个实施例中,多个着色器核心可以变化。在至少一个实施例中,图形处理器1940包括核心间任务管理器1945,其充当线程分派器以将执行线程分派给一个或更多个着色器核心1955A-1955N和分块单元1958,以加速基于图块渲染的分块操作,其中在图像空间中细分了场景的渲染操作,例如,以利用场景内的局部空间一致性或优化内部缓存的使用。

[0324] 推理和/或训练逻辑915用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图9A和/或图9B提供关于推理和/或训练逻辑915的细节。在至少一个实施例中,推理和/或训练逻辑915可以在集成电路图19A和/或图19B中用于至少部分地基于使用神经网络训练操作、神经网络函数或架构,或本文所述的神经网络用例计算的权重参数来进行推理或预测操作。

[0325] 在至少一个实施例中,空间自适应可分离卷积层7可用于集成电路19A和/或19B中,以用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构、或本文所述的神经网络用例计算的权重参数来推理或预测操作。

[0326] 图20A-20B示出了根据本文描述的实施例的附加示例性图形处理器逻辑。在至少一个实施例中,图20A示出了可以包括在图18的图形处理器1810内的图形核心2000,并且在至少一个实施例中,其可以是如图19B所示的统一着色器核心1955A-1955N。图20B示出了在至少一个实施例中的适用于在多芯片模块上部署的高度并行的通用图形处理单元2030。

[0327] 在至少一个实施例中,图形核心2000包括共享指令高速缓存2002、纹理单元2018和高速缓存/共享存储器2020,它们对于图形核心2000内的执行资源是通用的。在至少一个实施例中,图形核心2000可包括多个切片2001A-2001N或每个核心的分区,并且图形处理器可包括图形核心2000的多个实例。切片2001A-2001N可包括支持逻辑,所述逻辑包括本地指令高速缓存2004A-2004N、线程调度器2006A-2006N、线程分派器2008A-2008N和一组寄存器2010A-2010N。在至少一个实施例中,切片2001A-2001N可以包括一组附加功能单元(AFU 2012A-2012N)、浮点单元(FPU 2014A-2014N)、整数算术逻辑单元(ALU 2016A-2016N)、地址计算单元(ACU 2013A-2013N)、双精度浮点单元(DPFPU 2015A-2015N)和矩阵处理单元(MPU 2017A-2017N)。

[0328] 在至少一个实施例中,FPU 2014A-2014N可以执行单精度(32位)和半精度(16位)浮点运算,而DPFPU 2015A-2015N则执行双精度(64位)浮点运算点操作。在至少一个实施例

中,ALU 2016A-2016N可以以8位、16位和32位精度执行可变精度整数运算,并且可以配置为混合精度运算。在至少一个实施例中,MPU 2017A-2017N还可被配置用于混合精度矩阵运算,包括半精度浮点运算和8位整数运算。在至少一个实施例中,MPU 2017A-2017N可以执行各种矩阵运算以加速机器学习应用程序框架,包括使得能够支持加速的通用矩阵到矩阵乘法(GEMM)。在至少一个实施例中,AFU 2012A-2012N可以执行浮点数或整数单元不支持的附加逻辑运算,包括三角运算(例如,正弦,余弦等)。

[0329] 推理和/或训练逻辑915用于执行与一个或更多个实施例相关联的推理和/或训练操作。这里结合图9A和/或图9B提供关于推理和/或训练逻辑915的细节。在至少一个实施例中,推理和/或训练逻辑915可以在图形核心2000中使用,用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构或本文所述的神经网络用例计算的权重参数来推理或预测操作。

[0330] 在至少一个实施例中,空间自适应可分离卷积层7可用于图形核心2000中,以用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构、或本文所述的神经网络用例计算的权重参数来推理或预测操作。

[0331] 图20B示出了在至少一个实施例中的通用处理单元(GPGPU) 2030,其可以被配置为使得高度并行的计算操作能够由一组图形处理单元来执行。在至少一个实施例中,GPGPU 2030可以直接链接到GPGPU 2030的其他实例,以创建多GPU集群以提高用于深度神经网络的训练速度。在至少一个实施例中,GPGPU 2030包括主机接口2032,以实现与主机处理器的连接。在至少一个实施例中,主机接口2032是PCI Express接口。在至少一个实施例中,主机接口2032可以是厂商专用的通信接口或通信结构。在至少一个实施例中,GPGPU 2030接收主机处理器的命令,并使用全局调度器2034,以将与那些命令相关联的执行线程分配给一组计算集群2036A-2036H。在至少一个实施例中,计算群集2036A-2036H共享高速缓存存储器2038。在至少一个实施例中,高速缓存存储器2038可以用作计算群集2036A-2036H内的高速缓存存储器的更高级别的高速缓存。

[0332] 在至少一个实施例中,GPGPU 2030包括存储器2044A-2044B,所述存储器2044A-2044B经由一组存储器控制器2042A-2042B与计算集群2036A-2036H耦合。在至少一个实施例中,存储器2044A-2044B可以包括各种类型的存储器设备,包括动态随机存取存储器(DRAM)或图形随机存取存储器,例如同步图形随机存取存储器(SGRAM),其包括图形双倍数据速率(GDDR)存储器。

[0333] 在至少一个实施例中,计算集群2036A-2036H每个都包括一组图形核心,例如如图20A的图形核心2000,所述图形核心可以包括多种类型的整数和浮点逻辑单元,所述逻辑单元可以在计算机各种精度范围上执行计算操作,包括适用于机器学习计算的精度。例如,在至少一个实施例中,每个计算集群2036A-2036H中的浮点单元的至少一个子集可以被配置为执行16位或32位浮点运算,而浮点单元的不同子集可以配置为执行64位浮点运算。

[0334] 在至少一个实施例中,GPGPU 2030的多个实例可以被配置为用作计算集群。在至少一个实施例中,计算集群2036A-2036H用于同步和数据交换的通信在实施例之间变化。在至少一个实施例中,GPGPU 2030的多个实例通过主机接口2032进行通信。在至少一个实施例中,GPGPU 2030包括I/O集线器2039,所述集线器将GPGPU 2030与GPU链路2040耦合,使得能够直接连接到GPGPU 2030的其他实例。在至少一个实施例中,GPU链路2040耦合到专用

GPU到GPU桥,所述桥使得GPGP 2030的多个实例之间能够通信和同步。在至少一个实施例中,GPU链路2040与高速互连耦合,以向其他GPGPU或并行处理器发送和接收数据。在至少一个实施例中,GPGPU 2030的多个实例位于单独的数据处理系统中,并通过可通过主机接口2032访问的网络设备进行通信。在至少一个实施例中,GPU链路2040可被配置为使得能够连接到主机除主机接口2032之外或作为其替代的处理器。

[0335] 在至少一个实施例中,GPGPU 2030可以被配置为训练神经网络。在至少一个实施例中,可以在推理平台内使用GPGPU 2030。在至少一个实施例中,在其中使用GPGPU 2030进行推理的情况下,相对于使用GPGPU训练神经网络时,GPGPU可以包括更少的计算集群2036A-2036H。在至少一个实施例中,与存储器2044A-2044B相关联的存储器技术可以在推理和训练配置之间有所不同,其中更高带宽的存储器技术专用于训练配置。在至少一个实施例中,GPGPU 2030的推理配置可以支持推理特定指令。例如,在至少一个实施例中,推理配置可以提供对一个或更多个8位整数点积指令的支持,该指令可以在部署的神经网络的推理操作期间使用。

[0336] 推理和/或训练逻辑915用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图9A和/或图9B提供关于推理和/或训练逻辑915的细节。在至少一个实施例中,推理和/或训练逻辑915可以在GPGPU 2030中使用,用于至少部分地基于使用神经网络训练操作、神经网络功能和/或架构或本文所述的神经网络用例计算的权重参数来推理或预测操作。

[0337] 在至少一个实施例中,空间自适应可分离卷积层7可用于GPGPU 2030中,以用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构、或本文所述的神经网络用例计算的权重参数来推理或预测操作。

[0338] 图21示出了根据至少一个实施例的计算机系统2100的框图。在至少一个实施例中,计算机系统2100包括具有经由可包括存储器集线器2105的互连路径通信的一个或更多个处理器2102和系统存储器2104的处理子系统2101。在至少一个实施例中,存储器集线器2105可以是芯片组部件内的单独部件,或者可以集成在一个或更多个处理器2102内。在至少一个实施例中,存储器集线器2105通过通信链路2106与I/O子系统2111耦合。在一个实施例中,I/O子系统2111包括I/O集线器2107,所述I/O集线器可以使计算机系统2100能够接收来自一个或更多个输入设备2108的输入。在至少一个实施例中,I/O集线器2107可以使显示控制器向一个或更多个显示设备2110A提供输出,所述显示控制器可以包括在一个或更多个处理器2102中。在至少一个实施例中,与I/O集线器2107耦合的一个或更多个显示设备2110A可以包括本地,内部或嵌入式显示设备。

[0339] 在至少一个实施例中,处理子系统2101包括经由总线或其他通信链路2113耦合到存储器集线器2105的一个或更多个并行处理器2112中。在至少一个实施例中,通信链路2113可以是任何一种许多基于标准的通信链路技术或协议,例如但不限于PCI Express,或者可以是特定于供应商的通信接口或通信结构。在至少一个实施例中,一个或更多个并行处理器2112形成计算集中的并行或矢量处理系统,所述系统可以包括大量处理核心和/或处理集群,例如多集成核心(MIC)处理器。在至少一个实施例中,一个或更多个并行处理器2112形成图形处理子系统,所述图形处理子系统可以将像素输出到经由I/O集线器2107耦合的一个或更多个显示设备2110A之一。在至少一个实施例中,一个或更多个并行处理器

2112还可以包括显示控制器和显示接口(未示出),以使得能够直接连接到一个或多个显示设备2110B。

[0340] 在至少一个实施例中,系统存储单元2114可以连接到I/O集线器2107,以提供用于计算机系统2100的存储机制。在至少一个实施例中,I/O交换机2116可以用于提供一个接口机制,以实现I/O集线器2107与其他组件之间的连接,例如可以集成到平台中的网络适配器2118和/或无线网络适配器2119,以及可以通过一个或多个附加设备2120添加的各种其他设备。在至少一个实施例中,网络适配器2118可以是有以太网适配器或另一有线网络适配器。在至少一个实施例中,无线网络适配器2119可以包括Wi-Fi、蓝牙、近场通信(NFC)中的一个或多个,或包括一个或多个无线电设备的其他网络设备。

[0341] 在至少一个实施例中,计算机系统2100可以包括未明确示出的其他组件,所述其他组件包括USB或其他端口连接、光学存储驱动器、视频捕获设备等,所述其他组件也可以连接到I/O集线器2107。在至少一个实施例中,可以使用任何合适的协议(例如基于PCI(外围组件互连)的协议(例如PCI-Express)或其他总线或点对点通信接口和/或协议)来实现互连图21中各个组件的通信路径,例如NV-Link高速互连或互连协议。

[0342] 在至少一个实施例中,一个或多个并行处理器2112包括为图形和视频处理而优化的电路,所述电路包括例如视频输出电路,并构成图形处理单元(GPU)。在至少一个实施例中,一个或多个并行处理器2112包括为通用处理而优化的电路。在至少一个实施例中,计算机系统2100的组件可以与单个集成电路上的一个或多个其他系统元件集成。例如,在至少一个实施例中,一个或多个并行处理器2112、存储器集线器2105、处理器2102和I/O集线器2107,可以被集成到片上系统(SoC)集成电路中。在至少一个实施例中,计算机系统2100的组件可以被集成到单个封装中,以形成系统级封装(SIP)配置。在至少一个实施例中,计算机系统2100的组件的至少一部分可以被集成到多芯片模块(MCM)中,所述多芯片模块可以与其他多芯片模块互连到模块化计算机系统中。

[0343] 推理和/或训练逻辑915用于执行与一个或多个实施例相关联的推理和/或训练操作。本文结合图9A和/或图9B提供关于推理和/或训练逻辑915的细节。在至少一个实施例中,推理和/或训练逻辑915可以在图21的系统2100中使用,用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构或本文所述的神经网络用例计算的权重参数来推理或预测操作。

[0344] 在至少一个实施例中,空间自适应可分离卷积层7可用于系统图2100中,以用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构、或本文所述的神经网络用例计算的权重参数来推理或预测操作。

[0345] 处理器

[0346] 图22A示出了根据至少一个实施例的并行处理器2200。在至少一个实施例中,并行处理器2200的各种组件可以使用一个或多个集成电路设备来实现,例如可编程处理器、专用集成电路(ASIC)或现场可编程门阵列(FPGA)。在至少一个实施例中,所示的并行处理器2200是根据示例性实施例的图21所示的一个或多个并行处理器2112的变体。

[0347] 在至少一个实施例中,并行处理器2200包括并行处理单元2202。在至少一个实施例中,并行处理单元2202包括I/O单元2204,其使得能够与其他设备进行通信,包括并行处理单元2202的其他实例。在至少一个实施例中,I/O单元2204可以直接连接到其他设备。在

至少一个实施例中，I/O单元2204通过使用集线器或交换机接口（例如，存储器集线器2105）与其他设备连接。在至少一个实施例中，存储器集线器2105与I/O单元2204之间的连接形成通信链路2113。在至少一个实施例中，I/O单元2204与主机接口2206和存储器交叉开关2216连接，其中主机接口2206接收用于执行处理操作的命令，而存储器交叉开关2216接收用于执行存储器操作的命令。

[0348] 在至少一个实施例中，当主机接口2206经由I/O单元2204接收命令缓冲区时，主机接口2206可以引导工作操作以执行那些命令到前端2208。在至少一个实施例中，前端2208与调度器2210耦合，调度器2210配置成将命令或其他工作项分配给处理集群阵列2212。在至少一个实施例中，调度器2210确保在将任务分配给处理集群阵列2212中的处理集群阵列2212之前，处理集群阵列2212被正确地配置并且处于有效状态。在至少一个实施例中，调度器2210通过在微控制器上执行的固件逻辑来实现。在至少一个实施例中，微控制器实现的调度器2210可配置成以粗粒度和细粒度执行复杂的调度和工作分配操作，从而实现对在处理阵列2212上执行的线程的快速抢占和上下文切换。在至少一个实施例中，主机软件可以证明用于通过多个图形处理门铃之一在处理阵列2212上进行调度的工作负载。在至少一个实施例中，工作负载然后可以由包括调度器2210的微控制器内的调度器2210逻辑在处理阵列2212上自动分配。

[0349] 在至少一个实施例中，处理集群阵列2212可以包括多达“N”个处理集群（例如，集群2214A、集群2214B到集群2214N）。在至少一个实施例中，处理集群阵列2212的每个集群2214A-2214N可以执行大量并发线程。在至少一个实施例中，调度器2210可以使用各种调度和/或工作分配算法将工作分配给处理集群阵列2212的集群2214A-2214N，其可以根据每种程序或计算类型产生的工作负载而变化。在至少一个实施例中，调度可以由调度器2210动态地处理，或者可以在配置为由处理集群阵列2212执行的程序逻辑的编译期间部分地由编译器逻辑来辅助。在至少一个实施例中，可将处理集群阵列2212的不同的集群2214A-2214N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。

[0350] 在至少一个实施例中，处理集群阵列2212可以配置成执行各种类型的并行处理操作。在至少一个实施例中，处理集群阵列2212配置成执行通用并行计算操作。例如，在至少一个实施例中，处理集群阵列2212可以包括执行处理任务的逻辑，该处理任务包括对视频和/或音频数据的过滤，执行建模操作，包括物理操作以及执行数据转换。

[0351] 在至少一个实施例中，处理集群阵列2212配置成执行并行图形处理操作。在至少一个实施例中，处理集群阵列2212可以包括附加逻辑以支持这种图形处理操作的执行，包括但不限于执行纹理操作的纹理采样逻辑，以及镶嵌逻辑和其他顶点处理逻辑。在至少一个实施例中，处理集群阵列2212可以配置成执行与图形处理有关的着色器程序，例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。在至少一个实施例中，并行处理单元2202可以经由I/O单元2204从系统存储器传送数据以进行处理。在至少一个实施例中，在处理期间，可以在处理期间将传送的数据存储到片上存储器（例如，并行处理器存储器2222），然后将其写回到系统存储器。

[0352] 在至少一个实施例中，当并行处理单元2202用于执行图形处理时，调度器2210可以配置成将处理工作负载划分为近似相等大小的任务，以更好地将图形处理操作分配给处理集群阵列2212的多个集群2214A-2214N。在至少一个实施例中，处理集群阵列2212的部分

可以配置成执行不同类型的处理。例如,在至少一个实施例中,第一部分可以配置成执行顶点着色和拓扑生成,第二部分可以配置成执行镶嵌和几何着色,并且第三部分可以配置成执行像素着色或其他屏幕空间操作,以生成用于显示的渲染图像。在至少一个实施例中,可以将由集群2214A-2214N中的一个或多个产生的中间数据存储在缓冲区中,以允许在集群2214A-2214N之间传输中间数据以进行进一步处理。

[0353] 在至少一个实施例中,处理集群阵列2212可以经由调度器2210接收要执行的处理任务,该调度器2210从前端2208接收定义处理任务的命令。在至少一个实施例中,处理任务可以包括要被处理的数据的索引,例如,表面(补丁)数据、原始数据、顶点数据和/或像素数据,以及状态参数和定义如何处理数据的命令(例如,要执行什么程序)。在至少一个实施例中,调度器2210可以配置成获取与任务相对应的索引,或者可以从前端2208接收索引。在至少一个实施例中,前端2208可以配置成确保在启动由传入命令缓冲区(例如,批缓冲区(batch-buffer)、推送缓冲区等)指定的工作负载之前,处理集群阵列2212配置成有效状态。

[0354] 在至少一个实施例中,并行处理单元2202的一个或多个实例中的每一个可以与并行处理器存储器2222耦合。在至少一个实施例中,可以经由存储器交叉开关2216访问并行处理器存储器2222,所述存储器交叉开关2216可以接收来自处理集群阵列2212以及I/O单元2204的存储器请求。在至少一个实施例中,存储器交叉开关2216可以经由存储器接口2218访问并行处理器存储器2222。在至少一个实施例中,存储器接口2218可以包括多个分区单元(例如,分区单元2220A、分区单元2220B到分区单元2220N),其可各自耦合至并行处理器存储器2222的一部分(例如,存储器单元)。在至少一个实施例中,多个分区单元2220A-2220N为配置为等于存储器单元的数量,使得第一分区单元2220A具有对应的第一存储器单元2224A,第二分区单元2220B具有对应的存储器单元2224B,第N分区单元2220N具有对应的第N存储器单元2224N。在至少一个实施例中,分区单元2220A-2220N的数量可以不等于存储器设备的数量。

[0355] 在至少一个实施例中,存储器单元2224A-2224N可以包括各种类型的存储器设备,包括动态随机存取存储器(DRAM)或图形随机存取存储器,例如同步图形随机存取存储器(SGRAM),包括图形双倍数据速率(GDDR)存储器。在至少一个实施例中,存储器单元2224A-2224N还可包括3D堆叠存储器,包括但不限于高带宽存储器(HBM)。在至少一个实施例中,可以跨存储器单元2224A-2224N来存储诸如帧缓冲区或纹理映射的渲染目标,从而允许分区单元2220A-2220N并行地写入每个渲染目标的部分,以有效地使用并行处理器存储器2222的可用带宽。在至少一个实施例中,可以排除并行处理器存储器2222的本地实例,以有利于利用系统存储器与本地高速缓存存储器结合的统一存储器设计。

[0356] 在至少一个实施例中,处理集群阵列2212的集群2214A-2214N中的任何一个都可以处理将被写入并行处理器存储器2222内的任何存储器单元2224A-2224N中的数据。在至少一个实施例中,存储器交叉开关2216可以配置为将每个集群2214A-2214N的输出传输到任何分区单元2220A-2220N或另一个集群2214A-2214N,集群2214A-2214N可以对输出执行其他处理操作。在至少一个实施例中,每个集群2214A-2214N可以通过存储器交叉开关2216与存储器接口2218通信,以从各种外部存储设备读取或写入各种外部存储设备。在至少一个实施例中,存储器交叉开关2216具有到存储器接口2218的连接以与I/O单元2204通信,以

及到并行处理器存储器2222的本地实例的连接,从而使不同处理集群2214A-2214N内的处理单元与系统存储器或不是并行处理单元2202本地的其他存储器进行通信。在至少一个实施例中,存储器交叉开关2216可以使用虚拟通道来分离集群2214A-2214N和分区单元2220A-2220N之间的业务流。

[0357] 在至少一个实施例中,可以在单个插入卡上提供并行处理单元2202的多个实例,或者可以将多个插入卡互连。在至少一个实施例中,并行处理单元2202的不同实例可以配置成相互操作,即使不同实例具有不同数量的处理核心,不同数量的本地并行处理器存储器和/或其他配置差异。例如,在至少一个实施例中,并行处理单元2202的一些实例可以包括相对于其他实例而言更高精度的浮点单元。在至少一个实施例中,结合并行处理单元2202或并行处理器2200的一个或更多个实例的系统可以以各种配置和形式因素来实现,包括但不限于台式机、膝上型计算机或手持式个人计算机、服务器、工作站、游戏机和/或嵌入式系统。

[0358] 图22B是根据至少一个实施例的分区单元2220的框图。在至少一个实施例中,分区单元2220是图22A的分区单元2220A-2220N之一的实例。在至少一个实施例中,分区单元2220包括L2高速缓存2221、帧缓冲区接口2225和ROP 2226(光栅操作单元)。L2高速缓存2221是读/写高速缓存,其配置成执行从存储器交叉开关2216和ROP 2226接收的加载和存储操作。在至少一个实施例中,L2高速缓存2221将读取未命中和紧急回写请求输出到帧缓冲区接口2225以进行处理。在至少一个实施例中,还可以经由帧缓冲区接口2225将更新发送到帧缓冲区以进行处理。在至少一个实施例中,帧缓冲区接口2225与并行处理器存储器中的存储器单元(诸如图22A的存储器单元2224A-2224N(例如,在并行处理器存储器2222内))之一相互作用。

[0359] 在至少一个实施例中,ROP 2226是一种处理单元,其执行光栅操作,诸如模版、z测试、混合等。在至少一个实施例中,ROP 2226然后输出存储在图形存储器中的处理后的图形数据。在至少一个实施例中,ROP 2226包括压缩逻辑以压缩被写入存储器的深度或颜色数据并解压缩从存储器读取的深度或颜色数据。压缩逻辑可以是利用多种压缩算法中的一种或更多种的无损压缩逻辑。ROP 2226执行的压缩的类型可以基于要压缩的数据的统计特性而变化。例如,在至少一个实施例中,基于每图块基础上的深度和颜色数据执行增量颜色压缩。

[0360] 在至少一个实施例中,ROP 2226包括在每个处理集群内(例如,图22的集群2214A-2214N),而不是在分区单元2220内。在至少一个实施例中,通过存储器交叉开关2216而不是像素片段数据传输对像素数据的读取和写入请求。在至少一个实施例中,经处理的图形数据可以在显示设备上(诸如图21的一个或更多个显示设备2110之一)显示,由处理器2102路由以供进一步处理,或者由图22A的并行处理器2200内的处理实体之一路由以供进一步处理。

[0361] 图22C是根据至少一个实施例的并行处理单元内的处理集群2214的框图。在至少一个实施例中,处理集群是图22的处理集群2214A-2214N之一的实例。在至少一个实施例中,处理集群2214可以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。在至少一个实施例中,单指令多数据(SIMD)指令发布技术用于支持大量线程的并行执行而无需提供多个独立的指令单元。在至少一个实施例中,使

用单指令多线程 (SIMT) 技术来支持并行执行大量一般同步的线程,这使用了公共指令单元,该公共指令单元配置成向每个处理集群内的一组处理引擎发出指令。

[0362] 在至少一个实施例中,可以通过将处理任务分配给SIMT并行处理器的管线管理器2232来控制处理集群2214的操作。在至少一个实施例中,管线管理器2232从图22的调度器2210接收指令,通过图形多处理器2234和/或纹理单元2236管理这些指令的执行。在至少一个实施例中,图形多处理器2234是SIMT并行处理器的示例性实例。然而,在至少一个实施例中,处理集群2214内可以包括不同架构的各种类型的SIMT并行处理器。在至少一个实施例中,在处理集群2214内可以包括图形多处理器2234的一个或更多个实例。在至少一个实施例中,图形多处理器2234可以处理数据,并且数据交叉开关2240可以用于将处理后的数据分发到多个可能的目的(包括其他着色器单元)地之一。在至少一个实施例中,管线管理器2232可以通过指定要经由数据交叉开关2240分配的处理后的数据的目的地来促进处理后的数据的分配。

[0363] 在至少一个实施例中,处理集群2214内的每个图形多处理器2234可以包括相同的一组功能执行逻辑(例如,算术逻辑单元、加载存储单元等)。在至少一个实施例中,可以以管线方式配置功能执行逻辑,其中可以在先前的指令完成之前发出新的指令。在至少一个实施例中,功能执行逻辑支持多种操作,包括整数和浮点算术、比较操作、布尔运算、移位和各种代数函数的计算。在至少一个实施例中,可以利用相同的功能单元硬件来执行不同的操作,并且可以存在功能单元的任何组合。

[0364] 在至少一个实施例中,传送到处理集群2214的指令构成线程。在至少一个实施例中,跨一组并行处理引擎执行的一组线程是线程组。在至少一个实施例中,线程组在不同的输入数据上执行程序。在至少一个实施例中,线程组内的每个线程可被分配给图形多处理器2234内的不同处理引擎。在至少一个实施例中,线程组可包括比图形多处理器2234内的多个处理引擎更少的线程。在至少一个实施例中,当线程组包括的线程数少于处理引擎的数量时,一个或更多个处理引擎在正在处理该线程组的循环期间可能是空闲的。在至少一个实施例中,线程组还可以包括比图形多处理器2234内的多个处理引擎更多的线程。在至少一个实施例中,当线程组包括比图形多处理器2234内的处理引擎的数量更多的线程时,可以在连续的时钟周期内执行处理。在至少一个实施例中,可以在图形多处理器2234上同时执行多个线程组。

[0365] 在至少一个实施例中,图形多处理器2234包括内部高速缓存存储器,以执行加载和存储操作。在至少一个实施例中,图形多处理器2234可以放弃内部高速缓存并使用处理集群2214内的高速缓存存储器(例如,L1高速缓存2248)。在至少一个实施例中,每个图形多处理器2234还可以访问分区单元(例如,图22A的分区单元2220A-2220N)内的L2高速缓存,这些分区单元在所有处理集群2214之间共享并且可以用于在线程之间传输数据。在至少一个实施例中,图形多处理器2234还可以访问片外全局存储器,其可以包括本地并行处理器存储器和/或系统存储器中的一个或更多个。在至少一个实施例中,并行处理单元2202外部的任何存储器都可以用作全局存储器。在至少一个实施例中,处理集群2214包括图形多处理器2234的多个实例,它们可以共享可以存储在L1高速缓存2248中的公共指令和数据。

[0366] 在至少一个实施例中,每个处理集群2214可以包括配置成将虚拟地址映射为物理地址的存储器管理单元(“MMU”)2245。在至少一个实施例中,MMU 2245的一个或更多个实例

可以驻留在图22的存储器接口2218内。在至少一个实施例中,MMU 2245包括一组页表条目(PTE),其用于将虚拟地址映射到图块(更多的讨论分块)的物理地址以及可选地映射到高速缓存行索引。在至少一个实施例中,MMU 2245可以包括地址转换后备缓冲区(TLB)或可以驻留在图形多处理器2234或L1高速缓存或处理集群2214内的高速缓存。在至少一个实施例中,处理物理地址以分配表面数据访问局部性,以便在分区单元之间进行有效的请求交织。在至少一个实施例中,高速缓存行索引可以用于确定对高速缓存行的请求是命中还是未命中。

[0367] 在至少一个实施例中,可以配置处理集群2214,使得每个图形多处理器2234耦合到纹理单元2236,以执行纹理映射操作,所述操作确定纹理样本位置、读取纹理数据以及过滤纹理数据。在至少一个实施例中,根据需从内部纹理L1高速缓存(未示出)或从图形多处理器2234内的L1高速缓存中读取纹理数据,并从L2高速缓存、本地并行处理器存储器或系统存储器中获取纹理数据。在至少一个实施例中,每个图形多处理器2234将处理后的任务输出到数据交叉开关2240,以将处理后的任务提供给另一处理集群2214以进行进一步处理或将处理后的任务存储在L2高速缓存、本地并行处理器存储器、或经由存储器交叉开关2216的系统存储器中。在至少一个实施例中,preROP 2242(光栅前操作单元)配置成从图形多处理器2234接收数据,将数据引导至ROP单元,该ROP单元可以与本文所述的分区单元(例如,图22的分区单元2220A-2220N)一起定位。在至少一个实施例中,PreROP 2242单元可以执行用于颜色混合的优化、组织像素颜色数据以及执行地址转换。

[0368] 推理和/或训练逻辑915用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图9A和/或图9B提供关于推理和/或训练逻辑915的细节。在至少一个实施例中,推理和/或训练逻辑915可以在图形处理集群2214中用于至少部分地基于使用本文描述的神经网络训练操作、神经网络函数和/或架构或神经网络用例计算的权重参数来进行推理或预测操作。

[0369] 在至少一个实施例中,空间自适应可分离卷积层7可用于图形处理集群2214中,以用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构、或本文所述的神经网络用例计算的权重参数来推理或预测操作。

[0370] 图22D示出了根据至少一个实施例的图形多处理器2234。在至少一个实施例中,图形多处理器2234与处理集群2214的管线管理器2232耦合。在至少一个实施例中,图形多处理器2234具有执行管线,该执行管线包括但不限于指令高速缓存2252、指令单元2254、地址映射单元2256、寄存器文件2258、一个或更多个通用图形处理单元(GPGPU)核心2262和一个或更多个加载/存储单元2266。GPGPU核心2262和加载/存储单元2266与高速缓存存储器2272和共享存储器2270通过存储器和高速缓存互连2268耦合。

[0371] 在至少一个实施例中,指令高速缓存2252从管线管理器2232接收要执行的指令流。在至少一个实施例中,将指令高速缓存在指令高速缓存2252中并将其分派以供指令单元2254执行。在一个实施例中,指令单元2254可以分派指令作为线程组(例如,线程束),将线程组的每个线程分配给GPGPU核心2262内的不同执行单元。在至少一个实施例中,指令可以通过在统一地址空间内指定地址来访问任何本地、共享或全局地址空间。在至少一个实施例中,地址映射单元2256可以用于将统一地址空间中的地址转换成可以由加载/存储单元2266访问的不同的存储器地址。

[0372] 在至少一个实施例中,寄存器文件2258为图形多处理器2234的功能单元提供了一组寄存器。在至少一个实施例中,寄存器文件2258为连接到图形多处理器2234的功能单元(例如,GPGPU核心2262、加载/存储单元2266)的数据路径的操作数提供了临时存储。在至少一个实施例中,在每个功能单元之间划分寄存器文件2258,使得为每个功能单元分配寄存器文件2258的专用部分。在至少一个实施例中,寄存器文件2258在图形多处理器2234正在执行的不同线程束之间划分。

[0373] 在至少一个实施例中,GPGPU核心2262可以各自包括用于执行图形多处理器2234的指令的浮点单元(FPU)和/或整数算术逻辑单元(ALU)。GPGPU核心2262在架构上可以相似或架构可能有所不同。在至少一个实施例中,GPGPU核心2262的第一部分包括单精度FPU和整数ALU,而GPGPU核心的第二部分包括双精度FPU。在至少一个实施例中,FPU可以实现用于浮点算法的IEEE 754-2008标准或启用可变精度浮点算法。在至少一个实施例中,图形多处理器2234可以另外包括一个或更多个固定功能或特殊功能单元,以执行特定功能,诸如复制矩形或像素混合操作。在至少一个实施例中,GPGPU核心中的一个或更多个也可以包括固定或特殊功能逻辑。

[0374] 在至少一个实施例中,GPGPU核心2262包括能够对多组数据执行单个指令的SIMD逻辑。在一个实施例中,GPGPU核心2262可以物理地执行SIMD4、SIMD8和SIMD16指令,并且在逻辑上执行SIMD1、SIMD2和SIMD32指令。在至少一个实施例中,用于GPGPU核心的SIMD指令可以在编译时由着色器编译器生成,或者在执行针对单程序多数据(SPMD)或SIMT架构编写和编译的程序时自动生成。在至少一个实施例中,可以通过单个SIMD指令来执行为SIMT执行模型配置的程序的多个线程。例如,在至少一个实施例中,可以通过单个SIMD8逻辑单元并行执行执行相同或相似操作的八个SIMT线程。

[0375] 在至少一个实施例中,存储器和高速缓存互连2268是将图形多处理器2234的每个功能单元连接到寄存器文件2258和共享存储器2270的互连网络。在至少一个实施例中,存储器和高速缓存互连2268是交叉开关互连,其允许加载/存储单元2266在共享存储器2270和寄存器文件2258之间实现加载和存储操作。在至少一个实施例中,寄存器文件2258可以以与GPGPU核心2262相同的频率操作,从而在GPGPU核心2262和寄存器文件2258之间进行数据传输的延迟非常低。在至少一个实施例中,共享存储器2270可以用于启用在图形多处理器2234内的功能单元上执行的线程之间的通信。在至少一个实施例中,高速缓存存储器2272可以用作例如数据高速缓存,以高速缓存在功能单元和纹理单元2236之间通信的纹理数据。在至少一个实施例中,共享存储器2270也可以用作程序管理的高速缓存。在至少一个实施例中,除了存储在高速缓存存储器2272中的自动高速缓存的数据之外,在GPGPU核心2262上执行的线程还可以以编程方式将数据存储到共享存储器中。

[0376] 在至少一个实施例中,如本文所述的并行处理器或GPGPU通信地耦合到主机/处理器核心,以加速图形操作、机器学习操作、图案分析操作以及各种通用GPU(GPGPU)功能。在至少一个实施例中,GPU可以通过总线或其他互连(例如,诸如PCIe或NVLink之类的高速互连)通信地耦合到主机处理器/核心。在至少一个实施例中,GPU可以与核心集成在相同的封装或芯片上,并通过内部处理器总线/互连(即,封装或芯片的内部)通信地耦合到核心。在至少一个实施例中,不管GPU连接的方式如何,处理器核心可以以工作描述符中包含的命令/指令序列的形式向GPU分配工作。在至少一个实施例中,GPU然后使用专用电路/逻辑来

有效地处理这些命令/指令。

[0377] 推理和/或训练逻辑915用于执行与一个或多个实施例相关联的推理和/或训练操作。本文结合图9A和/或9B提供了关于推理和/或训练逻辑915的细节。在至少一个实施例中,推理和/或训练逻辑915可以在图形多处理器2234中被用于至少部分地基于使用在此描述的神经网络训练操作、神经网络功能和/或架构或神经网络使用情况计算的权重参数来推理或预测操作。

[0378] 在至少一个实施例中,空间自适应可分离卷积层7可用于图形多处理器2234中,以用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构、或本文所述的神经网络用例计算的权重参数来推理或预测操作。

[0379] 图23示出了根据至少一个实施例的多GPU计算系统2300。在至少一个实施例中,多GPU计算系统2300可以包括经由主机接口交换机2304耦合到多个通用图形处理单元(GPGPU) 2306A-D的处理器2302。在至少一个实施例中,主机接口交换机2304是快速PCI交换机设备,其将处理器2302耦合到快速PCI总线,处理器2302可以通过所述快速PCI总线与GPGPU 2306A-D进行通信。GPGPU 2306A-D可以经由一组高速点对点GPU到GPU链路2316互连。在至少一个实施例中,GPU到GPU链路2316经由专用GPU链路连接到GPGPU 2306A-D中的每一个。在至少一个实施例中,P2P GPU链路2316实现GPGPU 2306A-D中的每一个之间的直接通信,而不需要处理器2302连接到的主机接口总线2304上的通信。在至少一个实施例中,通过指向P2P GPU链路2316的GPU到GPU流量,主机接口总线2304保持可用于系统存储器访问或例如经由一个或多个网络设备与多GPU计算系统2300的其他实例通信。虽然在至少一个实施例中,GPGPU 2306A-D经由主机接口交换机2304连接到处理器2302,但是在至少一个实施例中,处理器2302包括对P2P GPU链路2316的直接支持,并且可以直接连接到GPGPU 2306A-D。

[0380] 推理和/或训练逻辑915用于执行与一个或多个实施例相关联的推理和/或训练操作。本文结合图9A和/或9B提供了关于推理和/或训练逻辑915的细节。在至少一个实施例中,推理和/或训练逻辑915可以用于多GPU计算系统2300中,用于至少部分地基于使用在此描述的神经网络训练操作、神经网络功能和/或架构或神经网络使用情况计算的权重参数来推理或预测操作。

[0381] 在至少一个实施例中,空间自适应可分离卷积层7可用多GPU计算系统2300中,以用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构、或本文所述的神经网络用例计算的权重参数来推理或预测操作。

[0382] 图24是根据至少一个实施例的图形处理器2400的框图。在至少一个实施例中,图形处理器2400包括环形互连2402、管线前端2404、媒体引擎2437和图形核心2480A-2480N。在至少一个实施例中,环形互连2402将图形处理器2400耦合到其他处理单元,包括其他图形处理器或一个或多个通用处理器核心。在至少一个实施例中,图形处理器2400是集成在多核心处理系统内的许多处理器之一。

[0383] 在至少一个实施例中,图形处理器2400经由环形互连2402接收多批命令。在至少一个实施例中,传入命令由管线前端2404中的命令流转化器2403来解释。在至少一个实施例中,图形处理器2400包括用于经由一个或多个图形核心2480A-2480N执行3D几何处理和媒体处理的可扩展执行逻辑。在至少一个实施例中,对于3D几何处理命令,命令流转化器

2403向几何管线2436提供命令。在至少一个实施例中,对于至少一些媒体处理命令,命令流转化器2403向与媒体引擎2437耦合的视频前端2434提供命令。在至少一个实施例中,媒体引擎2437包括用于视频和图像后处理的视频质量引擎(VQE) 2430和用于提供硬件加速的媒体数据编码和解码的多格式编码/解码(MFX) 2433引擎。在至少一个实施例中,几何管线2436和媒体引擎2437各自为由至少一个图形核心2480A提供的线程执行资源生成执行线程。

[0384] 在至少一个实施例中,图形处理器2400包括以模块化核心2480A-2480N(有时被称为核心切片)为特征的可扩展线程执行资源,每个图形核心具有多个子核心2450A-2450N、2460A-2460N(有时被称为核心子切片)。在至少一个实施例中,图形处理器2400可以具有任何数量的图形核心2480A至2480N。在至少一个实施例中,图形处理器2400包括至少具有第一子核心2450A和第二子核心2460A的图形核心2480A。在至少一个实施例中,图形处理器2400是具有单个子核心(例如,2450A)的低功率处理器。在至少一个实施例中,图形处理器2400包括多个图形核心2480A-2480N,每个图形核心包括一组第一子核心2450A-2450N和一组第二子核心2460A-2460N。在至少一个实施例中,第一子核心2450A-2450N中的每个子核心至少包括第一组执行单元2452A-2452N和媒体/纹理采样器2454A-2454N。在至少一个实施例中,第二子核心2460A-2460N中的每个子核心至少包括第二组执行单元2462A-2462N和采样器2464A-2464N。在至少一个实施例中,每个子核心2450A-2450N、2460A-2460N共享一组共享资源2470A-2470N。在至少一个实施例中,共享资源包括共享高速缓存存储器和像素操作逻辑。

[0385] 推理和/或训练逻辑915用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图9A和/或图9B提供了关于推理和/或训练逻辑915的细节。在至少一个实施例中,推理和/或训练逻辑915可以在图形处理器2400中用于至少部分地基于使用在此描述的神经网络训练操作、神经网络功能和/或架构或神经网络使用情况计算的权重参数来推理或预测操作。

[0386] 在至少一个实施例中,空间自适应可分离卷积层7可用于图形处理器2400中,以用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构、或本文所述的神经网络用例计算的权重参数来推理或预测操作。

[0387] 图25是示出根据至少一个实施例的可以包括用于执行指令的逻辑电路的处理器2500的微架构的框图。在至少一个实施例中,处理器2500可以执行指令,包括x86指令、ARM指令、用于专用集成电路(ASIC)的专用指令等。在至少一个实施例中,处理器2510可以包括用于存储封装数据的寄存器,例如作为加利福尼亚州圣克拉拉市英特尔公司采用MMX技术启用的微处理器中的64位宽MMXTM寄存器。在至少一个实施例中,整数和浮点数形式可用的MMX寄存器可以与封装的数据元素一起运行,所述封装的数据元素伴随单指令多数据(“SIMD”)和流式SIMD扩展(“SSE”)指令。在至少一个实施例中,与SSE2、SSE3、SSE4、AVX或更高版本(一般称为“SSEx”)技术有关的128位宽XMM寄存器可以保存此类封装数据操作数。在至少一个实施例中,处理器2510可以执行指令以加速机器学习或深度学习算法、训练或推理。

[0388] 在至少一个实施例中,处理器2500包括有序前端(“前端”)2501,以提取要执行的指令并准备稍后在处理器管线中使用的指令。在至少一个实施例中,前端2501可以包括几

个单元。在至少一个实施例中,指令预取器2526从存储器中获取指令并将指令提供给指令解码器2528,指令解码器2528又对指令进行解码或解释。例如,在至少一个实施例中,指令解码器2528将接收到的指令解码为机器可执行的所谓的“微指令”或“微操作”(也称为“微操作”或“微指令”)的一个或更多个操作。在至少一个实施例中,指令解码器2528将指令解析为操作码以及相应的数据和控制字段,其可以由微架构用来使用以根据至少一个实施例来执行操作。在至少一个实施例中,跟踪高速缓存2530可以将解码的微指令组装成微指令队列2534中的程序排序的序列或追踪以供执行。在至少一个实施例中,当追踪高速缓存2530遇到复杂指令时,微码ROM 2532提供完成操作所需的微指令。

[0389] 在至少一个实施例中,可以将一些指令转换成单个微操作,而另一些指令则需要几个微操作来完成全部操作。在至少一个实施例中,如果需要多于四个的微指令来完成一条指令,则指令解码器2528可以访问微码ROM 2532以执行指令。在至少一个实施例中,可以将指令解码为少量的微指令以在指令解码器2528处进行处理。在至少一个实施例中,如果需要多个微指令完成操作,则可以将指令存储在微码ROM 2532中。在至少一个实施例中,追踪高速缓存器2530参考入口点可编程逻辑阵列(“PLA”)以确定正确的微指令指针,用于根据至少一个实施例从微码ROM 2532读取微码序列以完成一个或更多个指令。在至少一个实施例中,在微码ROM 2532完成对指令的微操作排序之后,机器的前端2501可以恢复从追踪高速缓存2530获取微操作。

[0390] 在至少一个实施例中,乱序执行引擎(“乱序引擎”)2503可以准备用于执行的指令。在至少一个实施例中,乱序执行逻辑具有多个缓冲区,以使指令流平滑并重新排序,以在指令沿管线下落并被调度执行时优化性能。乱序执行引擎2503包括但不限于分配器/寄存器重命名器2540、存储器微指令队列2542、整数/浮点微指令队列2544、存储器调度器2546、快速调度器2502、慢速/通用浮点调度器(“慢速/通用FP调度器”)2504和简单浮点调度器(“简单FP调度器”)2506。在至少一个实施例中,快速调度器2502、慢速/通用浮点调度器2504和简单浮点调度器2506也统称为“微指令调度器2502、2504、2506”。分配器/寄存器重命名器2540分配每个微指令按序列执行所需要的机器缓冲区和资源。在至少一个实施例中,分配器/寄存器重命名器2540将逻辑寄存器重命名为寄存器文件中的条目。在至少一个实施例中,分配器/寄存器重命名器2540还为两个微指令队列之一中的每个微指令分配条目,存储器微指令队列2542用于存储器操作和整数/浮点微指令队列2544用于非存储器操作,在存储器调度器2546和微指令调度器2502、2504、2506的前面。在至少一个实施例中,微指令调度器2502、2504、2506基于它们的从属输入寄存器操作数源的就绪性和需要完成的执行资源微指令的可用性来确定何时准备好执行微指令。在至少一个实施例中,至少一个实施例的快速调度器2502可以在每半个主时钟周期上调度,而慢速/通用浮点调度器2504和简单浮点调度器2506可以在每一个主处理器时钟周期调度一次。在至少一个实施例中,微指令调度器2502、2504、2506对调度端口进行仲裁,以调度用于执行的微指令。

[0391] 在至少一个实施例中,执行块2511包括但不限于整数寄存器文件/支路网络2508、浮点寄存器文件/支路网络(“FP寄存器文件/支路网络”)2510、地址生成单元(“AGU”)2512和2514、快速算术逻辑单元(“快速ALU”)2516和2518、慢速算术逻辑单元(“慢速ALU”)2520、浮点ALU(“FP”)2522和浮点移动单元(“FP移动”)2524。在至少一个实施例中,整数寄存器文件/支路网络2508和浮点寄存器文件/旁路网络2510在本文中也称为“寄存器文件2508、

2510”。在至少一个实施例中，AGU 2512和2514、快速ALU 2516和2518、慢速ALU 2520、浮点ALU 2522和浮点移动单元2524在本文中也称为“执行单元2512、2514、2516、2518、2520、2522和2524”。在至少一个实施例中，执行块2511可以包括但不限于任意数量(包括零)和类型的寄存器文件、支路网络、地址生成单元和执行单元(以任何组合)。

[0392] 在至少一个实施例中，寄存器文件2508、2510可以布置在微指令调度器2502、2504、2506与执行单元2512、2514、2516、2518、2520、2522和2524之间。在至少一个实施例中，整数寄存器文件/支路网络2508执行整数运算。在至少一个实施例中，浮点寄存器文件/支路网络2510执行浮点操作。在至少一个实施例中，寄存器文件2508、2510中的每一个可以包括但不限于支路网络，该支路网络可以绕过或转发尚未写入寄存器文件中的刚刚完成的结果到新的从属对象。在至少一个实施例中，寄存器文件2508、2510可以彼此通信数据。在至少一个实施例中，整数寄存器文件/支路网络2508可以包括但不限于两个单独的寄存器文件、一个寄存器文件用于低阶32位数据，第二寄存器文件用于高阶32位数据。在至少一个实施例中，浮点寄存器文件/支路网络2510可以包括但不限于128位宽的条目，因为浮点指令通常具有宽度为64至128位的操作数。

[0393] 在至少一个实施例中，执行单元2512、2514、2516、2518、2520、2522、2524可以执行指令。在至少一个实施例中，寄存器文件2508、2510存储微指令需要执行的整数和浮点数据操作数值。在至少一个实施例中，处理器2500可以包括但不限于任何数量的执行单元2512、2514、2516、2518、2520、2522、2524及其组合。在至少一个实施例中，浮点ALU 2522和浮点移动单元2524，可以执行浮点、MMX、SIMD、AVX和SSE或其他操作，包括专门的机器学习指令。在至少一个实施例中，浮点ALU 2522可以包括但不限于64位乘64位浮点除法器，以执行除法、平方根和余数微操作。在至少一个实施例中，可以用浮点硬件来处理涉及浮点值的指令。在至少一个实施例中，可以将ALU操作传递给快速ALU 2516、2518。在至少一个实施例中，快速ALU 2516、2518可以以半个时钟周期的有效延迟执行快速操作。在至少一个实施例中，大多数复杂的整数运算进入慢速ALU 2520，因为慢速ALU 2520可以包括但不限于用于长延迟类型操作的整数执行硬件，例如乘法器、移位、标志逻辑和分支处理。在至少一个实施例中，存储器加载/存储操作可以由AGU 2512、2514执行。在至少一个实施例中，快速ALU 2516、快速ALU 2518和慢速ALU 2520可以对64位数据操作数执行整数运算。在至少一个实施例中，可以实现快速ALU 2516、快速ALU 2518和慢速ALU 2520以支持包括十六、三十二、128、256等的各种数据位大小。在至少一个实施例中，浮点ALU 2522和浮点移动单元2524可以实现为支持具有各种宽度的位的一定范围的操作数。在至少一个实施例中，浮点ALU 2522和浮点移动单元2524可以在可以结合SIMD和多媒体指令对128位宽封装数据操作数上进行操作。

[0394] 在至少一个实施例中，微指令调度器2502、2504、2506在父加载完成执行之前调度从属操作。在至少一个实施例中，由于可以在处理器2500中推测性地调度和执行微指令，处理器2500还可以包括用于处理存储器未命中的逻辑。在至少一个实施例中，如果数据高速缓存中的数据加载未命中，则可能存在在管线中正在运行的从属操作，其使调度器暂时没有正确的数据。在至少一个实施例中，一种重放机制追踪并重新执行使用不正确数据的指令。在至少一个实施例中，可能需要重放从属操作并且可以允许完成独立操作。在至少一个实施例中，处理器的至少一个实施例的调度器和重放机制也可以设计为捕获用于文本串比较操作的指令序列。

[0395] 在至少一个实施例中,术语“寄存器”可以指代可以用作识别操作数的指令的一部分的机载处理器存储位置。在至少一个实施例中,寄存器可以是那些可以从处理器外部使用的寄存器(从程序员的角度来看)。在至少一个实施例中,寄存器可能不限于特定类型的电路。相反,在至少一个实施例中,寄存器可以存储数据、提供数据并执行本文描述的功能。在至少一个实施例中,本文描述的寄存器可以通过处理器内的电路使用多种不同技术来实现,例如专用物理寄存器、使用寄存器重命名动态分配的物理寄存器、专用和动态分配的物理寄存器的组合等。在至少一个实施例中,整数寄存器存储32位整数数据。至少一个实施例的寄存器文件还包含八个用于封装数据的多媒体SIMD寄存器。

[0396] 推理和/或训练逻辑915用于执行与一个或多个实施例相关联的推理和/或训练操作。本文结合图9A和/或图9B提供关于推理和/或训练逻辑915的细节。在至少一个实施例中,可以将推理和/或训练逻辑915的部分或全部并入执行块2511以及示出或未示出的其他存储器或寄存器。例如,在至少一个实施例中,本文描述的训练和/或推理技术可以使用执行块2511中示出的一个或多个ALU。此外,权重参数可以存储在片上或片外存储器和/或寄存器(示出或未示出)中,该寄存器和/或寄存器配置执行块2511的ALU以执行一种或更多种本文所述的机器学习算法、神经网络架构、用例或训练技术。

[0397] 图26示出了根据至少一个实施例的深度学习应用程序处理器2600。在至少一个实施例中,深度学习应用程序处理器2600使用指令,如果由深度学习应用程序处理器2600执行,则指令使深度学习应用程序处理器2600执行贯穿本公开描述的一些或全部过程和技术。在至少一个实施例中,深度学习应用程序处理器2600是专用集成电路(ASIC)。在至少一个实施例中,应用程序处理器2600执行矩阵乘法运算或者“硬连线”到硬件中,作为执行一个或多个指令或两者的结果。在至少一个实施例中,深度学习应用程序处理器2600包括但不限于处理集群2610(1)-2610(12)、芯片间链路(“ICL”)2620(1)-2620(12)、芯片间控制器(“ICC”)2630(1)-2630(2)、第二代高带宽存储器(“HBM2”)2640(1)-2640(4)、存储器控制器(“Mem Ctrlr”)2642(1)-2642(4)、高带宽存储器物理层(“HBM PHY”)2644(1)-2644(4)、管理控制器中央处理单元(“管理控制器CPU”)2650、串行外围设备接口、内部集成电路和通用输入/输出块(“SPI、I2C、GPIO”)2660,外围组件互连快速控制器和直接存储器访问块(“PCIe控制器和DMA”)2670、以及十六通道外围组件互连快速端口(“PCI Express x 16”)2680。

[0398] 在至少一个实施例中,处理集群2610可以执行深度学习操作,包括基于一种或更多种训练技术计算的权重参数的推理或预测操作,包括本文所述的那些技术。在至少一个实施例中,每个处理集群2610可以包括但不限于任何数量和类型的处理器。在至少一个实施例中,深度学习应用程序处理器2600可以包括任何数量和类型的处理集群2600。在至少一个实施例中,芯片间链路2620是双向的。在至少一个实施例中,芯片间链路2620和芯片间控制器2630使多个深度学习应用程序处理器2600能够交换信息,包括从执行一个或多个神经网络中体现的一种或更多种机器学习算法而产生的激活信息。在至少一个实施例中,深度学习应用程序处理器2600可以包括任意数量(包括零)和类型的ICL 2620和ICC 2630。

[0399] 在至少一个实施例中,HBM2 2640提供总共32GB的存储器。HBM2 2640(i)与存储器控制器2642(i)和HBM PHY 2644(i)都相关联。在至少一个实施例中,任何数量的HBM2 2640可以提供任何类型和总量的高带宽存储器,并且可以与任何数量(包括零)和类型的存储器

控制器2642和HBM PHY 2644相关联。在至少一个实施例中,可以用任何数量和类型的块替换SPI、I2C、GPIO 3360、PCIe控制器2660和DMA 2670和/或PCIe2680,以任何技术上可行的方式实现任何数量和类型的通信标准。

[0400] 推理和/或训练逻辑915用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图9A和/或图9B提供关于推理和/或训练逻辑915的细节。在至少一个实施例中,深度学习应用程序处理器用于训练机器学习模型(例如神经网络),以预测或推理提供给深度学习应用程序处理器2600的信息。在至少一个实施例中,深度学习应用程序处理器2600用于基于已经由另一处理器或系统或由深度学习应用程序处理器2600训练的经训练的机器学习模型(例如,神经网络)来推理或预测信息。在至少一个实施例中,处理器2600可以用于执行本文所述的一个或更多个神经网络用例。

[0401] 在至少一个实施例中,空间自适应可分离卷积层7可用于一个深度学习应用程序处理器2600中,以用于至少部分地基于使用神经网络训练操作、神经网络函数和/或架构、或本文所述的神经网络用例计算的权重参数来推理或预测操作。

[0402] 图27是根据至少一个实施例的神经形态处理器2700的框图。在至少一个实施例中,神经形态处理器2700可以从神经形态处理器2700外部的源接收一个或更多个输入。在至少一个实施例中,这些输入可以被传输到神经形态处理器2700内的一个或更多个神经元2702。在至少一个实施例中,可以使用包括一个或更多个算术逻辑单元(ALU)的电路或逻辑来实现神经元2702及其组件。在至少一个实施例中,神经形态处理器2700可以包括但不限于成千上万个神经元2702的实例,但是可以使用任何合适数量的神经元2702。在至少一个实施例中,神经元2702的每个实例可以包括神经元输入2704和神经元输出2706。在至少一个实施例中,神经元2702可以生成可以传输到神经元2702的其他实例的输入的输出生成。在至少一个实施例中,神经元输入2704和神经元输出2706可以经由突触2708互连。

[0403] 在至少一个实施例中,神经元2702和突触2708可以互连,使得神经形态处理器2700操作以处理或分析由神经形态处理器2700接收的信息。在至少一个实施例中,当通过神经元输入2704接收到的输入超过阈值时,神经元2702可以发送输出脉冲(或“触发”或“峰值”)。在至少一个实施例中,神经元2702可以对在神经元输入2704处接收到的信号进行求和或积分。例如,在至少一个实施例中,神经元2702可以实现为有泄漏的积分-触发神经元,其中如果求和(称为“膜电位”)超过阈值,则神经元2702可以使用诸如sigmoid或阈值函数的传递函数来产生输出(或“触发”)。在至少一个实施例中,泄漏的积分-触发神经元可以将神经输入2704处接收到的信号求和成膜电位,并且可以应用衰减因子(或泄漏)以减小膜电位。在至少一个实施例中,如果在神经元输入2704处接收到足够快以超过阈值的多个输入信号(即,在膜电势衰减得太低而不能触发之前),则泄漏的积分-触发神经元可能会触发。在至少一个实施例中,神经元2702可以使用接收输入、将输入积分到膜电位、并衰减膜电位的电路或逻辑来实现。在至少一个实施例中,可以对输入求平均,或者可以使用任何其他合适的传递函数。此外,在至少一个实施例中,神经元2702可以包括但不限于当将传递函数应用于神经输入2704的结果超过阈值时在神经元输出2706处产生输出尖峰的比较器电路或逻辑。在至少一个实施例中,一旦神经元2702触发,它可以通过例如将膜电位复位为0或另一合适的默认值来忽略先前接收的输入信息。在至少一个实施例中,一旦膜电位被重置为0,则神经元2702可以在合适的时间段(或修复期)之后恢复正常操作。

[0404] 在至少一个实施例中,神经元2702可以通过突触2708互连。在至少一个实施例中,突触2708可以操作以将从第一神经元2702的输出的信号传输到第二神经元2702的输入。在至少一个实施例中,神经元2702可以在一个以上的突触2708实例上传输信息。在至少一个实施例中,神经元输出2706的一个或更多个实例可以通过突触2708的实例连接到同一神经元2702中神经元输入2704的实例。在至少一个实施例中,相对于突触2708的那个实例,神经元2702的实例产生要在突触2708的实例上传输的输出可以被称为“突触前神经元”。在至少一个实施例中,相对于突触2708的实例,神经元2702的实例接收通过突触2708的实例传输的输入可以被称为“突触后神经元”。在至少一个实施例中,关于突触2708的各种实例,因为神经元2702的实例可以接收来自一个或更多个突触2708实例的输入,并且还可以通过一个或更多个突触2708实例传输输出,因此神经元2702的单个实例可以既是“突触前神经元”又是“突触后神经元”。

[0405] 在至少一个实施例中,神经元2702可以被组织成一层或更多层。神经元2702的每个实例可以具有一个神经元输出2706,该神经元输出2706可以通过一个或更多个突触2708扇出到一个或更多个神经元输入2704。在至少一个实施例中,第一层2710中的神经元2702的神经元输出2706可以连接到第二层2712中的神经元2702的神经元输入2704。在至少一个实施例中,层2710可以被称为“前馈层”。在至少一个实施例中,在第一层2710的实例中神经元2702的每个实例可以扇出到第二层2712中的神经元2702的每个实例。在至少一个实施例中,第一层2710可以被称为“完全连接的前馈层”。在至少一个实施例中,在第二层2712的每个实例中的神经元2702的每个实例扇出到少于在第三层2714中的神经元2702的所有实例。在至少一个实施例中,第二层2712可以被称为“稀疏连接的前馈层”。在至少一个实施例中,第二层2712中的神经元2702可以扇出到多个其他层中的神经元2702,包括扇出到(相同的)第二层2712中的神经元2702。在至少一个实施例中,第二层2712可以被称为“循环层”。神经形态处理器2700可以包括但不限于循环层和前馈层的任何合适的组合,包括但不限于稀疏连接的前馈层和完全连接的前馈层。

[0406] 在至少一个实施例中,神经形态处理器2700可以包括但不限于可重新配置的互连架构或专用硬连线互连,以将突触2708连接到神经元2702。在至少一个实施例中,神经形态处理器2700可以包括但不限于电路或逻辑,其根据神经网络拓扑结构和神经元扇入/扇出,允许根据需要将突触分配给不同神经元2702。例如,在至少一个实施例中,可以使用互连结构(诸如片上网络)或通过专用连接将突触2708连接到神经元2702。在至少一个实施例中,可以使用电路或逻辑来实现突触互连及其组件。

[0407] 图28示出了根据至少一个实施例的处理系统。在至少一个实施例中,系统2800包括一个或更多个处理器2802和一个或更多个图形处理器2808,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2802或处理器核心2807的服务器系统。在至少一个实施例中,系统2800是结合在片上系统(SoC)集成电路内的处理平台,以在移动、手持或嵌入式设备使用。

[0408] 在至少一个实施例中,系统2800可以包括或结合在基于服务器的游戏平台中,包括游戏和媒体控制台的游戏控制台、移动游戏控制台、手持游戏控制台或在线游戏控制台。在至少一个实施例中,系统2800是移动电话、智能电话、平板计算设备或移动互联网设备。在至少一个实施例中,处理系统2800还可包括与可穿戴设备耦合或集成在可穿戴设备中,

例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。在至少一个实施例中,处理系统2800是电视或机顶盒设备,其具有一个或多个处理器2802以及由一个或多个图形处理器2808生成的图形界面。

[0409] 在至少一个实施例中,一个或多个处理器2802每个包括一个或多个处理器核心2807,以处理指令,该指令在被执行时执行针对系统和用户软件的操作。在至少一个实施例中,一个或多个处理器核心2807中的每一个被配置为处理特定指令集2809。在至少一个实施例中,指令集2809可以促进复杂指令集计算(CISC)、精简指令集计算(RISC),或通过超长指令字(VLIW)进行计算。在至少一个实施例中,处理器核心2807可以各自处理不同的指令集2809,该指令集可以包括有助于模拟其他指令集的指令。在至少一个实施例中,处理器核心2807还可以包括其他处理设备,例如数字信号处理器(DSP)。

[0410] 在至少一个实施例中,处理器2802包括高速缓存存储器2804。在至少一个实施例中,处理器2802可以具有单个内部高速缓存或多个级别的内部高速缓存。在至少一个实施例中,高速缓存存储器在处理器2802的各个组件之间共享。在至少一个实施例中,处理器2802还使用外部高速缓存(例如,三级(L3)高速缓存或最后一级高速缓存(LLC))(未示出),可以使用已知的高速缓存一致性技术在处理器核心2807之间共享该外部高速缓存。在至少一个实施例中,处理器2802中另外包括寄存器文件2806,处理器可以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)。在至少一个实施例中,寄存器文件2806可以包括通用寄存器或其他寄存器。

[0411] 在至少一个实施例中,一个或多个处理器2802与一个或多个接口总线2810耦合,以在处理器2802与系统2800中的其他组件之间传输通信信号,例如地址、数据或控制信号。在至少一个实施例中,接口总线2810在一个实施例中可以是处理器总线,例如直接媒体接口(DMI)总线的版本。在至少一个实施例中,接口2810不限于DMI总线,并且可以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。在至少一个实施例中,处理器2802包括集成存储器控制器2816和平台控制器集线器2830。在至少一个实施例中,存储器控制器2816促进存储器设备与处理系统2800的其他组件之间的通信,而平台控制器集线器(PCH)2830通过本地I/O总线提供到输入/输出(I/O)设备的连接。

[0412] 在至少一个实施例中,存储器设备2820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、相变存储设备或具有适当的性能以用作处理器存储器。在至少一个实施例中,存储设备2820可以用作处理系统2800的系统存储器,以存储数据2822和指令2821,以在一个或多个处理器2802执行应用程序或过程时使用。在至少一个实施例中,存储器控制器2816还与可选的外部图形处理器2812耦合,其可以与处理器2802中的一个或多个图形处理器2808通信以执行图形和媒体操作。在至少一个实施例中,显示设备2811可以连接至处理器2802。在至少一个实施例中,显示设备2811可以包括内部显示设备中的一个或多个,例如在移动电子设备或膝上型设备或通过显示器接口(例如显示端口(DisplayPort)等)连接的外部显示设备中。在至少一个实施例中,显示设备2811可以包括头戴式显示器(HMD),诸如用于虚拟现实(VR)应用或增强现实(AR)应用中的立体显示设备。

[0413] 在至少一个实施例中,平台控制器集线器2830使外围设备能够通过高速I/O总线

连接到存储设备2820和处理器2802。在至少一个实施例中，I/O外围设备包括但不限于音频控制器2846、网络控制器2834、固件接口2828、无线收发器2826、触摸传感器2825、数据存储设备2824（例如，硬盘驱动器、闪存等）。在至少一个实施例中，数据存储设备2824可以经由存储接口（例如，SATA）或经由外围总线来连接，诸如外围组件互连总线（例如，PCI、PCIe）。在至少一个实施例中，触摸传感器2825可以包括触摸屏传感器、压力传感器或指纹传感器。在至少一个实施例中，无线收发器2826可以是Wi-Fi收发器、蓝牙收发器或移动网络收发器，诸如3G、4G或长期演进（LTE）收发器。在至少一个实施例中，固件接口2828使能与系统固件的通信，并且可以是例如统一可扩展固件接口（UEFI）。在至少一个实施例中，网络控制器2834可以启用有线网络的网络连接。在至少一个实施例中，高性能网络控制器（未示出）与接口总线2810耦合。在至少一个实施例中，音频控制器2846是多通道高清晰度音频控制器。在至少一个实施例中，处理系统2800包括可选的传统（legacy）I/O控制器2840，用于将传统（例如，个人系统2（PS/2））设备耦合到系统。在至少一个实施例中，平台控制器集线器2830还可以连接到一个或多个通用串行总线（USB）控制器2842，该控制器连接输入设备，诸如键盘和鼠标2843组合、相机2844或其他USB输入设备。

[0414] 在至少一个实施例中，存储器控制器2816和平台控制器集线器2830的实例可以集成到离散的外部图形处理器中，例如外部图形处理器2812。在至少一个实施例中，平台控制器集线器2830和/或存储器控制器2816可以在一个或多个处理器2802的外部。例如，在至少一个实施例中，系统2800可以包括外部存储器控制器2816和平台控制器集线器2830，其可以配置成在与处理器2802通信的系统芯片组中的存储器控制器集线器和外围控制器集线器。

[0415] 推理和/或训练逻辑915用于执行与一个或多个实施例相关联的推理和/或训练操作。本文结合图9A和/或图9B提供关于推理和/或训练逻辑915的细节。在至少一个实施例中，部分或全部推理和/或训练逻辑915可以结合到图形处理器2800中。例如，在至少一个实施例中，本文描述的训练和/或推理技术可以使用一个或多个ALU，所述ALU体现在3D管线2812中。此外，在至少一个实施例中，本文描述的推理和/或训练操作可以使用除图9A或图9B所示的逻辑之外的逻辑来完成。在至少一个实施例中，权重参数可以存储在片上或片外存储器和/或寄存器（示出或未示出）中，其配置图形处理器2800的ALU，以执行一种或更多种本文所述的机器学习算法、神经网络架构、用例或训练技术。

[0416] 图29是根据至少一个实施例的具有一个或多个处理器核心2902A-2902N、集成存储器控制器2914和集成图形处理器2908的处理器2900的框图。在至少一个实施例中，处理器2900可以包含附加核心，多达并包括以虚线框表示的附加核心2902N。在至少一个实施例中，每个处理器核心2902A-2902N包括一个或多个内部高速缓存单元2904A-2904N。在至少一个实施例中，每个处理器核心还可以访问一个或多个共享高速缓存单元2906。

[0417] 在至少一个实施例中，内部高速缓存单元2904A-2904N和共享高速缓存单元2906表示处理器2900内的高速缓存存储器层次。在至少一个实施例中，高速缓存存储器单元2904A-2904N可以包括每个处理器核心内的至少一级指令和数据高速缓存以及共享中级高速缓存中的一级或更多级缓存，例如2级（L2）、3级（L3）、4级（L4）或其他级别的高速缓存，其中将外部存储器之前的最高级别的高速缓存归类为LLC。在至少一个实施例中，高速缓存一致性逻辑维持各种高速缓存单元2906和2904A-2904N之间的一致性。

[0418] 在至少一个实施例中,处理器2900还可包括一组一个或更多个总线控制器单元2916和系统代理核心2910。在至少一个实施例中,一个或更多个总线控制器单元2916管理一组外围总线,例如一个或更多个PCI或PCIe总线。在至少一个实施例中,系统代理核心2910为各种处理器组件提供管理功能。在至少一个实施例中,系统代理核心2910包括一个或更多个集成存储器控制器2914,以管理对各种外部存储器设备(未示出)的访问。

[0419] 在至少一个实施例中,一个或更多个处理器核心2902A-2902N包括对多线程同时进行的支持。在至少一个实施例中,系统代理核心2910包括用于在多线程处理期间协调和操作核心2902A-2902N的组件。在至少一个实施例中,系统代理核心2910可以另外包括电源控制单元(PCU),该电源控制单元包括用于调节处理器核心2902A-2902N和图形处理器2908的一个或更多个电源状态的逻辑和组件。

[0420] 在至少一个实施例中,处理器2900还包括用于执行图处理操作的图形处理器2908。在至少一个实施例中,图形处理器2908与共享高速缓存单元2906和包括一个或更多个集成存储器控制器2914的系统代理核心2910耦合。在至少一个实施例中,系统代理核心2910还包括用于驱动图形处理器输出到一个或更多个耦合的显示器的显示器控制器2911。在至少一个实施例中,显示器控制器2911也可以是经由至少一个互连与图形处理器2908耦合的独立模块,或者可以集成在图形处理器2908内。

[0421] 在至少一个实施例中,基于环的互连单元2912用于耦合处理器2900的内部组件。在至少一个实施例中,可以使用替代性互连单元,例如点对点互连、交换互连或其他技术。在至少一个实施例中,图形处理器2908经由I/O链路2913与环形互连2912耦合。

[0422] 在至少一个实施例中,I/O链路2913代表多种I/O互连中的至少一种,包括促进各种处理器组件与高性能嵌入式存储器模块2918(例如eDRAM模块)之间的通信的封装I/O互连。在至少一个实施例中,处理器核心2902A-2902N和图形处理器2908中的每一个使用嵌入式存储器模块2918作为共享的最后一级高速缓存。

[0423] 在至少一个实施例中,处理器核心2902A-2902N是执行公共指令集架构的同质核心。在至少一个实施例中,处理器核心2902A-2902N在指令集架构(ISA)方面是异构的,其中一个或更多个处理器核心2902A-2902N执行公共指令集,而一个或更多个其他处理器核心2902A-2902N执行公共指令集的子集或不同指令集。在至少一个实施例中,就微架构而言,处理器核心2902A-2902N是异构的,其中具有相对较高功耗的一个或更多个核心与具有较低功耗的一个或更多个功率核心耦合。在至少一个实施例中,处理器2900可以在一个或更多个芯片上实现或被实现为SoC集成电路。

[0424] 推理和/或训练逻辑915用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图9A和/或图9B提供关于推理和/或训练逻辑915的细节。在至少一个实施例中,部分或全部推理和/或训练逻辑915可以结合到图形处理器2908中。例如,在至少一个实施例中,本文描述的训练和/或推理技术可以使用一个或更多个ALU,所述ALU体现在图29中的3D管线2812、图形核心2915A、共享功能逻辑2916,图形核心2915B,共享功能逻辑2920,或其他逻辑中。此外,在至少一个实施例中,本文描述的推理和/或训练操作可以使用除图9A或图9B所示的逻辑之外的逻辑来完成。在至少一个实施例中,权重参数可以存储在片上或片外存储器和/或寄存器(示出或未示出)中,其配置图形处理器2910的ALU以执行一种或更多种本文所述的机器学习算法、神经网络架构、用例或训练技术。

[0425] 图30是图形处理器3000的框图,该图形处理器可以是分立的图形处理单元,或者可以是与多个处理核心集成的图形处理器。在至少一个实施例中,图形处理器3000经由存储器映射的I/O接口与图形处理器3000上的寄存器以及放置在存储器中的命令进行通信。在至少一个实施例中,图形处理器3000包括用于访问存储器的存储器接口3014。在至少一个实施例中,存储器接口3014是到本地存储器、一个或更多个内部高速缓存、一个或更多个共享的外部高速缓存和/或到系统存储器的接口。

[0426] 在至少一个实施例中,图形处理器3000还包括用于将显示输出数据驱动到显示设备3020的显示控制器3002。在至少一个实施例中,显示控制器3002包括用于显示设备3020的一个或更多个覆盖平面的硬件以及多层视频或用户接口元素的组合。在至少一个实施例中,显示设备3020可以是内部或外部显示设备。在至少一个实施例中,显示设备3020是头戴式显示设备,例如虚拟现实(VR)显示设备或增强现实(AR)显示设备。在至少一个实施例中,图形处理器3000包括视频编解码器引擎3006,以将媒体编码、解码或转码为一种或更多种媒体编码格式,从一种或更多种媒体编码格式编码、解码或转码,或在一种或更多种媒体编码格式之间进行编码、解码或转码,所述媒体编码格式包括但不限于运动图像专家组(MPEG)格式(例如MPEG-2),高级视频编码(AVC)格式(例如H.264/MPEG-4AVC,以及美国电影电视工程师协会(SMPTE)421M/VC-1)和联合图像专家组(JPEG)格式(例如JPEG)和Motion JPEG(MJPEG)格式。

[0427] 在至少一个实施例中,图形处理器3000包括块图像传送(BLIT)引擎3004,以执行二维(2D)光栅化器操作,包括例如位边界块传送。但是,在至少一个实施例中,使用图形处理引擎(GPE)3010的一个或更多个组件来执行2D图形操作。在至少一个实施例中,GPE 3010是用于执行图形操作(包括三维(3D)图形操作和媒体操作)的计算引擎。

[0428] 在至少一个实施例中,GPE 3010包括用于执行3D操作的3D管线3012,例如使用对3D图元形状(例如,矩形、三角形等)进行操作的处理功能来渲染三维图像和场景。3D管线3012包括执行各种任务和/或产生到3D/媒体子系统3015的执行线程的可编程和固定功能元素。虽然3D管线3012可用于执行媒体操作,但是在至少一个实施例中,GPE 3010还包括媒体管线3016,其用于执行媒体操作,诸如视频后处理和图像增强。

[0429] 在至少一个实施例中,媒体管线3016包括固定功能或可编程逻辑单元,用于执行一种或更多种专门的媒体操作,例如视频解码加速,视频去隔行和视频编码加速,代替或代表视频编解码器引擎3006。在至少一个实施例中,媒体管线3016还包括线程产生单元,用于产生线程以在3D/媒体子系统3015上执行。在至少一个实施例中,产生的线程在3D/媒体子系统3015中包含的一个或更多个图形执行单元上执行媒体操作的计算。

[0430] 在至少一个实施例中,3D/媒体子系统3015包括用于执行3D管线3012和媒体管线3016产生的线程的逻辑。在至少一个实施例中,3D管线3012和媒体管线3016将线程执行请求发送到3D/媒体子系统3015,其包括用于仲裁各种请求并将其分派给可用线程执行资源的线程分派逻辑。在至少一个实施例中,执行资源包括用于处理3D和媒体线程的图形执行单元的阵列。在至少一个实施例中,3D/媒体子系统3015包括用于线程指令和数据的一个或更多个内部高速缓存。在至少一个实施例中,子系统3015还包括共享存储器,其包括寄存器和可寻址存储器,以在线程之间共享数据并存储输出数据。

[0431] 推理和/或训练逻辑915用于执行与一个或更多个实施例相关联的推理和/或训练

操作。本文结合图9A和/或图9B提供关于推理和/或训练逻辑915的细节。在至少一个实施例中,可以将推理和/或训练逻辑915的部分或全部合并到处理器3000中。例如,在至少一个实施例中,本文描述的训练和/或推理技术可以使用3D管线3012中包含的一个或更多个ALU。此外,在至少一个实施例中,本文描述的推理和/或训练操作可以使用除图9A或图9B所示的逻辑以外的逻辑来完成。在至少一个实施例中,权重参数可以存储在片上或片外存储器和/或寄存器(示出或未示出)中,其配置图形处理器3000的ALU以执行一种或更多种机器学习算法、神经网络架构、用例或本文介绍的训练技术。

[0432] 图31是根据至少一个实施例的图形处理器的图形处理引擎3110的框图。在至少一个实施例中,图形处理引擎(GPE) 3110是图30中所示的GPE 3010的版本。在至少一个实施例中,媒体管线3106是可选的,并且可以不显式地包括在GPE 3110中。在至少一个实施例中,单独的媒体和/或图像处理耦合到GPE 3110。

[0433] 在至少一个实施例中,GPE 3110耦合到或包括命令流转化器3103,其向3D管线3012和/或媒体管线3016提供命令流。在至少一个实施例中,命令流转化器3103耦合到存储器,所述存储器可以是系统存储器,也可以是内部高速缓存存储器和共享高速缓存存储器中的一个或更多个。在至少一个实施例中,命令流转化器3103从存储器接收命令,并且将命令发送到3D管线3012和/或媒体管线3016。在至少一个实施例中,命令是从环形缓冲区中获取的指令、基元或微操作,该环形缓冲区存储用于3D管线3012和媒体管线3016的命令。在至少一个实施例中,环形缓冲区还可以包括存储各批多个命令的批命令缓冲区。在至少一个实施例中,用于3D管线3012的命令还可以包括对存储在存储器中的数据引用,例如但不限于用于3D管线3012的顶点和几何数据和/或用于媒体管线3016的图像数据和存储器对象。在至少一个实施例中,3D管线3012和媒体管线3016通过执行操作或通过将一个或更多个执行线程分派到图形核心阵列3114,来处理命令和数据。在至少一个实施例中,图形核心阵列3114包括一个或更多个图形核心块(例如,一个或更多个图形核心3115A、一个或更多个图形核心3115B),每个块包括一个或更多个图形核心。在至少一个实施例中,每个图形核心包括一组图形执行资源,所述图形执行资源包括通用和图形特定的执行逻辑,用于执行图形和计算操作,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,包括图9A和图9B中的推理和/或训练逻辑915。

[0434] 在至少一个实施例中,3D管线3012包括固定功能和可编程逻辑,用于通过处理指令并将执行线程分派到图形核心阵列3114,来处理一个或更多个着色器程序,例如顶点着色器、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。在至少一个实施例中,图形核心阵列3114提供统一的执行资源块,所述执行资源块用于处理着色器程序。在至少一个实施例中,在图形核心阵列3114的图形核心3115A-3115B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,并且可以执行与多个着色器关联的多个同时执行线程。

[0435] 在至少一个实施例中,图形核心阵列3114还包括执行逻辑,用于执行媒体功能,诸如视频和/或图像处理。在至少一个实施例中,除了图形处理操作之外,执行单元还包括可编程以执行并行通用计算操作的通用逻辑。

[0436] 在至少一个实施例中,输出数据可以将数据输出到统一返回缓冲区(URB) 3118中的存储器,所述输出数据由在图形核心阵列3114上执行的线程生成。URB 3118可以存储多

个线程的数据。在至少一个实施例中,URB 3118可以用于在图形核心阵列3114上执行的不同线程之间发送数据。在至少一个实施例中,URB 3118还可用于图形核心阵列3114上的线程与共享功能逻辑3120内的固定功能逻辑之间的同步。

[0437] 在至少一个实施例中,图形核心阵列3114是可缩放的,使得图形核心阵列3114包括可变数量的图形核心,每个图形核心具有基于GPE 3110的目标功率和性能水平的可变数量的执行单元。在至少一个实施例中,执行资源是动态可伸缩的,使得执行资源可以根据需要被启用或禁用。

[0438] 在至少一个实施例中,图形核心阵列3114耦合到共享功能逻辑3120,该共享功能逻辑包括在图形核心阵列3114中的图形核心之间共享的多个资源。在至少一个实施例中,由共享功能逻辑3120执行的共享功能体现在向图形核心阵列3114提供专门的补充功能的硬件逻辑单元中。在至少一个实施例中,共享功能逻辑3120包括但不限于采样器3121、数学3122和线程间通信(ITC) 3123逻辑。在至少一个实施例中,一个或更多个高速缓存3125被包含在或耦合到共享功能逻辑3120中。

[0439] 在至少一个实施例中,如果对专用功能的需求不足以包含在图形核心阵列3114中,则使用共享功能。在至少一个实施例中,专用功能的单个实例在共享功能逻辑3120中使用,并且在图形核心阵列3114内的其他执行资源之间共享。在至少一个实施例中,可以将图形核心阵列3114内的共享功能逻辑3116内的特定共享功能包含在图形核心阵列3114广泛使用的共享功能逻辑3116内。在至少一个实施例中,图形核心阵列3114内的共享功能逻辑3116可包括共享功能逻辑3120内的一些或全部逻辑。在至少一个实施例中,共享功能逻辑3120内的所有逻辑元件可在图形核心阵列3114的共享功能逻辑3116内复制。在至少一个实施例中,排除共享功能逻辑3120,以支持图形核心阵列3114内的共享功能逻辑3116。

[0440] 推理和/或训练逻辑915用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图9A和/或9B提供关于推理和/或训练逻辑915的细节。在至少一个实施例中,部分或全部推理和/或训练逻辑915可以结合到图形处理器3110中。例如,在至少一个实施例中,本文描述的推理和/或推理技术可以使用一个或更多个ALU,所述ALU体现在3D管线3012、图形核心3115A、共享功能逻辑3116、图形核心3115B、共享功能逻辑3120或图31中的其他逻辑中。此外,在至少一个实施例中,本文描述的推理和/或训练操作可以使用除图9A或图9B所示的逻辑之外的逻辑来完成。在至少一个实施例中,权重参数可以存储在片上或片外存储器和/或寄存器(示出或未示出)中,其配置图形处理器3110的ALU,以执行一种或更多种本文所述的机器学习算法、神经网络架构、用例或训练技术。

[0441] 图32是根据本文所述的至少一个实施例的图形处理器核心3200的硬件逻辑的框图。在至少一个实施例中,图形处理器核心3200被包括在图形核心阵列内。在至少一个实施例中,图形处理器核心3200(有时称为核心切片)可以是模块化图形处理器内的一个或更多个图形核心。在至少一个实施例中,图形处理器核心3200是一个图形核心切片的示例,并且本文所述的图形处理器可以基于目标功率和性能包络线包括多个图形核心切片。在至少一个实施例中,每个图形核心3200可以包括与多个子核心3201A-3201F耦合的固定功能块3230,也称为子切片,其包括通用和固定功能逻辑的模块块。

[0442] 在至少一个实施例中,固定功能块3230包括几何或固定功能管线3236,例如,在较低性能和/或较低功率的图形处理器实施方式中,该几何和固定功能管线3236可以由图形

处理器3200中的所有子核心共享。在至少一个实施例中,几何/固定功能管线3236包括3D固定功能管线、视频前端单元,线程产生器和线程分派器以及管理统一返回缓冲区的统一返回缓冲区管理器。

[0443] 在固定的至少一个实施例中,固定功能块3230还包括图形SoC接口3237、图形微控制器3238和媒体管线3239。图形SoC接口3237提供了图形核心3200以及片上集成电路系统中的其他处理器核心之间的接口。在至少一个实施例中,图形微控制器3238是可编程子处理器,其可配置为管理图形处理器3200的各种功能,包括线程分派、调度和抢占。在至少一个实施例中,媒体管线3239包括有助于对包括图像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。在至少一个实施例中,媒体管线3239经由对子核心3201-3201F内的计算或采样逻辑的请求来实现媒体操作。

[0444] 在至少一个实施例中,SoC接口3237使图形核心3200能够与通用应用程序处理器核心(例如,CPU)和/或SoC内的其他组件通信,包括存储器层次元素,诸如共享的最后一级高速缓存、系统RAM和/或嵌入式片上或封装DRAM。在至少一个实施例中,SoC接口3237还可以使得能够与SoC内的固定功能设备(例如,相机成像管线)进行通信,并且使得能够使用和/或实现可以在图形核心3200和SoC内部的CPU之间共享的全局存储器原子。在至少一个实施例中,SoC接口3237还可以实现用于图形核心3200的电源管理控制,并且启用图形核心3200的时钟域与SoC内的其他时钟域之间的接口。在至少一个实施例中,SoC接口3237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区,其配置为向图形处理器内的一个或更多个图形核心中的每一个提供命令和指令。在至少一个实施例中,当要执行媒体操作时,可以将命令和指令分派给媒体管线3239,或者当要执行图形处理操作时,可以将其分配给几何形状和固定功能管线(例如,几何形状和固定功能管线3236,几何形状和固定功能管线3214)。

[0445] 在至少一个实施例中,图形微控制器3238可以配置为对图形核心3200执行各种调度和管理任务。在至少一个实施例中,图形微控制器3238可以在子核心3201A-3201F中的执行单元(EU)阵列3202A-3202F、3204A-3204F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。在至少一个实施例中,在包括图形核心3200的SoC的CPU核心上执行的主机软件可以提交多个图形处理器门铃之一的工作负载,其调用适当的图形引擎上的调度操作。在至少一个实施例中,调度操作包括确定接下来要运行哪个工作负载、将工作负载提交给命令流转化器、抢先在引擎上运行的现有工作负载、监控工作负载的进度以及在工作负载完成时通知主机软件。在至少一个实施例中,图形微控制器3238还可以促进图形核心3200的低功率或空闲状态,从而为图形核心3200提供在图形核心3200内独立于操作系统和/或系统上的图形驱动程序软件的跨低功率状态转换的保存和恢复寄存器的能力。

[0446] 在至少一个实施例中,图形核心3200可以具有比所示的子核心3201A-3201F多或少达N个模块化子核心。对于每组N个子核心,在至少一个实施例中,图形核心3200还可以包括共享功能逻辑3210、共享和/或高速缓存存储器3212、几何/固定功能管线3214以及附加的固定功能逻辑3216以加速各种图形和计算处理操作。共享功能逻辑3210可以包括可由图形核心3200内的每个N个子核心共享的逻辑单元(例如,采样器、数学和/或线程间通信逻辑)。共享和/或高速缓存存储器3212可以是图形核心3200内的N个子核心3201A-3201F的最后一级高速缓存,并且还可以用作可由多个子核心访问的共享存储器。在至少一个实施例

中,可以包括几何/固定功能管线3214来代替固定功能块3230内的几何/固定功能管线3236,并且可以包括相同或相似的逻辑单元。

[0447] 在至少一个实施例中,图形核心3200包括附加的固定功能逻辑3216,其可以包括供图形核心3200使用的各种固定功能加速逻辑。在至少一个实施例中,附加的固定功能逻辑3216包括用于仅位置着色中使用的附加的几何管线。在仅位置着色中,存在至少两个几何管线,而在几何/固定功能管线3216、3236内的完整几何管线和剔除管线中,其是可以包括在附加的固定功能逻辑3216中的附加几何管线。在至少一个实施例中,剔除管线是完整几何管线的修整版。在至少一个实施例中,完整管线和剔除管线可以执行应用程序的不同实例,每个实例具有单独的环境。在至少一个实施例中,仅位置着色可以隐藏被丢弃的三角形的长剔除运行,从而在某些情况下可以更早地完成着色。例如,在至少一个实施例中,附加固定功能逻辑3216中的剔除管线逻辑可以与主应用程序并行执行位置着色器,并且通常比完整管线更快地生成关键结果,因为剔除管线获取并遮蔽顶点的位置属性,无需执行光栅化和将像素渲染到帧缓冲区。在至少一个实施例中,剔除管线可以使用生成的临界结果来计算所有三角形的可见性信息,而与这些三角形是否被剔除无关。在至少一个实施例中,完整管线(在这种情况下可以称为重播管线)可以消耗可见性信息来跳过剔除的三角形以仅遮盖最终传递到光栅化阶段的可见三角形。

[0448] 在至少一个实施例中,附加的固定功能逻辑3216还可包括机器学习加速逻辑,例如固定功能矩阵乘法逻辑,用于实现包括用于机器学习训练或推理的优化。

[0449] 在至少一个实施例中,在每个图形子核心3201A-3201F内包括一组执行资源,其可用于响应于图形管线、媒体管线或着色器程序的请求来执行图形、媒体和计算操作。在至少一个实施例中,图形子核心3201A-3201F包括多个EU阵列3202A-3202F、3204A-3204F,线程分派和线程间通信(TD/IC)逻辑3203A-3203F,3D(例如,纹理)采样器3205A-3205F,媒体采样器3206A-3206F,着色器处理器3207A-3207F和共享本地存储器(SLM)3208A-3208F。EU阵列3202A-3202F、3204A-3204F每个都包含多个执行单元,这些执行单元是通用图形处理单元,能够为图形、媒体或计算操作提供服务,执行浮点和整数/定点逻辑运算,包括图形、媒体或计算着色器程序。在至少一个实施例中,TD/IC逻辑3203A-3203F为子核心内的执行单元执行本地线程分派和线程控制操作,并促进在子核心的执行单元上执行的线程之间的通信。在至少一个实施例中,3D采样器3205A-3205F可以将与纹理或其他3D图形相关的数据读取到存储器中。在至少一个实施例中,3D采样器可以基于与给定纹理相关联的配置的采样状态和纹理格式来不同地读取纹理数据。在至少一个实施例中,媒体采样器3206A-3206F可以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。在至少一个实施例中,每个图形子核心3201A-3201F可以可替代地包括统一的3D和媒体采样器。在至少一个实施例中,在每个子核心3201A-3201F内的执行单元上执行的线程可以利用每个子核心内的共享本地存储器3208A-3208F,以使在线程组内执行的线程能够使用片上存储器的公共池来执行。

[0450] 推理和/或训练逻辑915用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图9A和/或图9B提供关于推理和/或训练逻辑915的细节。在至少一个实施例中,推理和/或训练逻辑915的部分或全部可以被合并到图形处理器3210中。例如,在至少一个实施例中,本文描述的训练和/或推理技术可以使用在3D管线3210、图形微控制器3238、

几何和固定功能管线3214和3236或图29中的其他逻辑中体现的一个或更多个ALU。此外,在至少一个实施例中,本文描述的推理和/或训练操作可以使用除图9A或图9B所示的逻辑以外的逻辑来完成。在至少一个实施例中,权重参数可以存储在片上或片外存储器和/或寄存器(示出或未示出)中,其配置图形处理器3200的ALU以执行一种或更多种本文介绍的机器学习算法、神经网络架构、用例或训练技术。

[0451] 图33A-33B示出了根据至少一个实施例的包括图形处理器核心的处理元件的阵列的线程执行逻辑3300。图33A示出了至少一个实施例,其中使用了线程执行逻辑3300。图33B示出了根据至少一个实施例的执行单元的示例性内部细节。

[0452] 如图33A中所示,在至少一个实施例中,线程执行逻辑3300包括着色器处理器3302、线程分派器3304、指令高速缓存3306、包括多个执行单元3308A-3308N的可缩放执行单元阵列、采样器3310、数据高速缓存3312和数据端口3314。在至少一个实施例中,可缩放执行单元阵列可以例如基于工作负载的计算要求,通过启用或禁用一个或更多个执行单元(例如,执行单元3308A,3308B,3308C,3308D到3308N-1和3308N中的任意一个)来动态缩放。在至少一个实施例中,可缩放执行单元通过链路到每个执行单元的互连结构互连。在至少一个实施例中,线程执行逻辑3300包括通过指令高速缓存3306、数据端口3314、采样器3310和执行单元3308A-3308N中的一个或更多个到存储器(诸如系统存储器或高速缓存存储器)的一个或更多个连接。在至少一个实施例中,每个执行单元(例如3308A)是独立的可编程通用计算单元,其能够执行多个同时的硬件线程,同时针对每个线程并行处理多个数据元素。在至少一个实施例中,执行单元3308A-3308N的阵列可缩放以包括任意数量的单独执行单元。

[0453] 在至少一个实施例中,执行单元3308A-3308N主要用于执行着色器程序。在至少一个实施例中,着色器处理器3302可以处理各种着色器程序并经由线程分派器3304来分派与着色器程序相关联的执行线程。在至少一个实施例中,线程分派器3304包括用于仲裁来自图形和媒体管线的线程初始化庆祝以及在执行单元3308A-3308N中的一个或更多个执行单元上实例化请求的线程的逻辑。例如,在至少一个实施例中,几何管线可以将顶点、镶嵌或几何着色器分派到线程执行逻辑以进行处理。在至少一个实施例中,线程分派器3304还可以处理来自执行着色器程序的运行时线程产生请求。

[0454] 在至少一个实施例中,执行单元3308A-3308N支持一种指令集,该指令集包括对许多标准3D图形着色器指令的本机支持,从而使图形库(例如Direct 3D和OpenGL)中的着色器程序只需最少的转换即可执行。在至少一个实施例中,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)。在至少一个实施例中,每个执行单元3308A-3308N包括一个或更多个算术逻辑单元(ALU),能够执行多发出单指令多数据(SIMD),并且多线程操作实现了高效的执行环境尽管有更高的延迟存储器访问。在至少一个实施例中,每个执行单元内的每个硬件线程具有专用的高带宽寄存器文件和相关的独立线程状态。在至少一个实施例中,执行是每个时钟到管线的多次发出,管线能够进行整数、单精度和双精度浮点运算、SIMD分支功能、逻辑运算、先验运算和其他运算。在至少一个实施例中,在等待来自存储器或共享功能之一的数据时,执行单元3308A-3308N内的依赖性逻辑使等待线程休眠直到返回了所请求的数据。在至少一个实施例中,当等待线程正在休眠时,硬件资源可以专用于处

理其他线程。例如,在至少一个实施例中,在与顶点着色器操作相关联的延迟期间,执行单元可以对像素着色器、片段着色器或另一类型的着色器程序(包括不同的顶点着色器)执行操作。

[0455] 在至少一个实施例中,执行单元3308A-3308N中的每一个执行单元在数据元素的阵列上进行操作。在至少一个实施例中,多个数据元素是“执行大小”或指令的通道数。在至少一个实施例中,执行通道是用于指令内的数据元素访问、屏蔽和流控制的执行的逻辑单元。在至少一个实施例中,多个通道可以独立于用于特定图形处理器的多个物理算术逻辑单元(ALU)或浮点单元(FPU)。在至少一个实施例中,执行单元3308A-3308N支持整数和浮点数据类型。

[0456] 在至少一个实施例中,执行单元指令集包括SIMD指令。在至少一个实施例中,各种数据元素可以作为封装数据类型存储在寄存器中,并且执行单元将基于那些元素的数据大小来处理各种元素。例如,在至少一个实施例中,当对256位宽的向量进行操作时,将向量的256位存储在寄存器中,并且执行单元对向量进行操作,作为四个单独的64位封装数据元素(四字(QW)大小数据元素)、八个单独的32位封装数据元素(双字(DW)大小数据元素)、十六个单独的16位封装数据元素(单词(W)大小数据元素)或三十二个单独的8位数据元素(字节(B)大小的数据元素)。然而,在至少一个实施例中,不同的向量宽度和寄存器大小是可能的。

[0457] 在至少一个实施例中,一个或更多个执行单元可以被组合成具有执行融合EU共用的线程控制逻辑(3307A-3307N)的融合执行单元3309A-3309N。在至少一个实施例中,可以将多个EU合并成一个EU组。在至少一个实施例中,融合EU组中的EU的数量可以配置为执行单独的SIMD硬件线程,融合的EU组中的EU的数量可根据各个实施例而变化。在至少一个实施例中,每个EU可以执行各种SIMD宽度,包括但不限于SIMD8、SIMD16和SIMD32。在至少一个实施例中,每个融合图形执行单元3309A-3309N包括至少两个执行单元。例如,在至少一个实施例中,融合执行单元3309A包括第一EU 3308A、第二EU 3308A以及第一EU 3308A和第二EU 3308B共有的线程控制逻辑3307A。在至少一个实施例中,线程控制逻辑3307A控制在融合图形执行单元3309A上执行的线程,从而允许融合执行单元3309A-3309N内的每个EU使用公共指令指针寄存器来执行。

[0458] 在至少一个实施例中,一个或更多个内部指令高速缓存(例如3306)被包括在线程执行逻辑3300中以高速缓存用于执行单元的线程指令。在至少一个实施例中,包括一个或更多个数据高速缓存(例如3312)以在线程执行期间高速缓存线程数据。在至少一个实施例中,包括采样器3310以提供用于3D操作的纹理采样和用于媒体操作的媒体采样。在至少一个实施例中,采样器3310包括专门的纹理或媒体采样功能,以在将采样数据提供给执行单元之前在采样过程中处理纹理或媒体数据。

[0459] 在执行期间,在至少一个实施例中,图形和媒体管线通过线程产生和分派逻辑将线程发起请求发送到线程执行逻辑3300。在至少一个实施例中,一旦一组几何对象已经被处理并光栅化成像素数据,则在着色器处理器3302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)被调用以进一步计算输出信息并且导致将结果写入输出表面(例如,颜色缓冲区、深度缓冲区、模板缓冲区等)。在至少一个实施例中,像素着色器或片段着色器计算要在光栅化对象上插值的各种顶点属性的值。在至少一个实施例中,着色器处理

器3302内的像素处理器逻辑然后执行应用程序接口 (API) 提供的像素或片段着色器程序。在至少一个实施例中,为了执行着色器程序,着色器处理器3302经由线程分派器3304将线程分派到执行单元(例如3308A)。在至少一个实施例中,着色器处理器3302使用采样器3310中的纹理采样逻辑来访问存储在存储器中的纹理贴图纹理数据。在至少一个实施例中,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,或者丢弃一个或更多个像素以进行进一步处理。

[0460] 在至少一个实施例中,数据端口3314提供了一种用于线程执行逻辑3300的存储器访问机制,以将处理后的数据输出到存储器以在图形处理器输出管线上进行进一步处理。在至少一个实施例中,数据端口3314包括或耦合到一个或更多个高速缓存存储器(例如,数据高速缓存3312)以高速缓存数据以便经由数据端口进行存储器访问。

[0461] 如图33B所示,在至少一个实施例中,图形执行单元3308可以包括指令获取单元3337、通用寄存器文件阵列 (GRF) 3324、架构寄存器文件阵列 (ARF) 3326、线程仲裁器3322、发送单元3330、分支单元3332、一组SIMD浮点单元 (FPU) 3334,以及在至少一个实施例中,一组专用整数SIMD ALU 3335。GRF 3324和ARF 3326包括一组与可以在图形执行单元3308中活跃的每个同时硬件线程相关联的通用寄存器文件和架构寄存器文件。在至少一个实施例中,在ARF 3326中维护每个线程架构状态,而在线程执行期间使用的数据存储GRF 3324中。在至少一个实施例中,每个线程的执行状态,包括每个线程的指令指针,可以被保存在ARF 3326中的线程专用寄存器中。

[0462] 在至少一个实施例中,图形执行单元3308具有一种架构,该架构是同时多线程 (SMT) 和细粒度交错多线程 (IMT) 的组合。在至少一个实施例中,架构具有模块化配置,该模块化配置可以在设计时基于同时线程的目标数量和每个执行单元的寄存器数量来进行微调,其中执行单元资源在用于执行多个同时线程的逻辑上分配。

[0463] 在至少一个实施例中,图形执行单元3308可以共同发布多个指令,每个指令可以是不同的指令。在至少一个实施例中,图形执行单元线程3308的线程仲裁器3322可以将指令分派到发送单元3330、分支单元3342或SIMD FPU 3334之一以供执行。在至少一个实施例中,每个执行线程可以访问GRF 3324中的128个通用寄存器,其中每个寄存器可以存储32个字节,可以作为32位数据元素的SIMD 8元素向量进行访问。在至少一个实施例中,每个执行单元线程可以访问GRF 3324中的4KB,尽管实施例不限于此,并且在其他实施例中可以提供更多或更少的寄存器资源。在至少一个实施例中,尽管每个执行单元的线程数量也可以根据实施例而变化,但是最多可以同时执行七个线程。在其中七个线程可以访问4KB的至少一个实施例中,GRF 3324可以存储总共28KB。在至少一个实施例中,灵活的寻址模式可以允许将寄存器一起寻址以有效地建立更宽的寄存器或表示跨步的矩形块数据结构。

[0464] 在至少一个实施例中,经由消息传递发送单元3330执行的“发送”指令来调度存储器操作、采样器操作和其他更长延迟的系统通信。在至少一个实施例中,将分支指令分派到专用分支单元3332促进SIMD发散和最终收敛。

[0465] 在至少一个实施例中,图形执行单元3308包括一个或更多个SIMD浮点单元 (FPU) 3334,以执行浮点操作。在至少一个实施例中,一个或更多个FPU 3334还支持整数计算。在至少一个实施例中,一个或更多个FPU 3334可以SIMD执行多达M个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。在至少一个实施例中,至少一个FPU提供

扩展的数学能力以支持高吞吐量的先验数学函数和双精度64位浮点。在至少一个实施例中,还存在一组8位整数SIMD ALU 3335,并且可以被专门优化以执行与机器学习计算相关的操作。

[0466] 在至少一个实施例中,可以在图形子核心分组(例如,子切片)中实例化图形执行单元3308的多个实例的阵列。在至少一个实施例中,执行单元3308可以跨多个执行通道执行指令。在至少一个实施例中,在图形执行单元3308上执行的每个线程在不同的通道上执行。

[0467] 推理和/或训练逻辑915用于执行与一个或更多个实施例相关联的推理和/或训练操作。下面结合图9A和/或图9B提供关于推理和/或训练逻辑915的细节。在至少一个实施例中,推理和/或训练逻辑915的部分或全部可以被结合到执行逻辑3300中。此外,在至少一个实施例中,可以使用除了图9A或图9B中所示的逻辑之外的逻辑来完成在此描述的推理和/或训练操作。在至少一个实施例中,权重参数可以存储在片上或片外存储器和/或寄存器(示出或未示出)中,其配置执行逻辑3300的ALU以执行一种或更多种机器学习算法、神经网络架构、用例或本文介绍的训练技术。

[0468] 图34示出了根据至少一个实施例的并行处理单元(“PPU”)3400。在至少一个实施例中,PPU 3400配置有机器可读代码,该机器可读代码如果由PPU 3400执行,则使得PPU 3400执行贯穿本公开描述的一些或全部过程和技术。在至少一个实施例中,PPU 3400是在一个或更多个集成电路设备上实现的多线程处理器,并且利用多线程作为被设计为处理在多个线程上并行执行的计算机可读指令(也称为机器可读指令或简单的指令)的延迟隐藏技术。在至少一个实施例中,线程是指执行线程,并且是被配置为由PPU 3400执行的一组指令的实例。在至少一个实施例中,PPU 3400是图形处理单元(“GPU”),图形处理单元配置为实现用于处理三维(“3D”)图形数据的图形渲染管线,以便生成用于在显示设备(诸如液晶显示器(“LCD”)设备)上显示的二维(“2D”)图像数据。在至少一个实施例中,PPU 3400用于执行计算,诸如线性代数运算和机器学习运算。图34仅出于说明性目的示出了示例并行处理器,并且应被解释为在本公开的范围内设想的处理器架构的非限制性示例,并且可以采用任何适当的处理器来对其进行补充和/或替代。

[0469] 在至少一个实施例中,一个或更多个PPU 3400配置成加速高性能计算(“HPC”)、数据中心和机器学习应用程序。在至少一个实施例中,PPU 3400配置成加速深度学习系统和应用程序,包括以下非限制性示例:自动驾驶汽车平台、深度学习、高精度语音、图像、文本识别系统、智能视频分析、分子模拟、药物发现、疾病诊断、天气预报、大数据分析、天文学、分子动力学模拟、财务建模、机器人技术、工厂自动化、实时语言翻译、在线搜索优化以及个性化用户推荐等。

[0470] 在至少一个实施例中,PPU 3400包括但不限于输入/输出(“I/O”)单元3406、前端单元3410、调度器单元3412、工作分配单元3414、集线器3416、交叉开关(“Xbar”)3420、一个或更多个通用处理集群(“GPC”)3418和一个或更多个分区单元(“存储器分区单元”)3422。在至少一个实施例中,PPU 3400通过一个或更多个高速GPU互连(“GPU互连”)3408连接到主机处理器或其他PPU 3400。在至少一个实施例中,PPU 3400通过互连3402连接到主机处理器或其他外围设备。在至少一个实施例中,PPU 3400连接到包括一个或更多个存储器设备(“存储器”)3404的本地存储器。在至少一个实施例中,存储器设备3404包括但不限于一个

或更多个动态随机存取存储器 (“DRAM”) 设备。在至少一个实施例中, 一个或更多个DRAM设备配置和/或可配置为高带宽存储器 (“HBM”) 子系统, 并且在每个设备内堆叠有多个DRAM管芯。

[0471] 在至少一个实施例中, 高速GPU互连3408可以指代系统使用其来进行缩放的基于线的多通道通信链路, 并包括与一个或更多个中央处理单元结合的一个或更多个PPU 3400 (“CPU”), 支持PPU 3400和CPU之间的缓存相干以及CPU主控。在至少一个实施例中, 高速GPU互连3408通过集线器3416将数据和/或命令传输到PPU 3400的其他单元, 例如一个或更多个复制引擎、视频编码器、视频解码器、电源管理单元和/或在图34中可能未明确示出的其他组件。

[0472] 在至少一个实施例中, I/O单元3406配置为通过系统总线3402从主机处理器 (图34中未示出) 发送和接收通信 (例如, 命令、数据)。在至少一个实施例中, I/O单元3406直接通过系统总线3402或通过一个或更多个中间设备 (例如存储器桥) 与主机处理器通信。在至少一个实施例中, I/O单元3406可以经由系统总线3402与一个或更多个其他处理器 (例如一个或更多个PPU 3400) 通信。在至少一个实施例中, I/O单元3406实现外围组件互连Express (“PCIe”) 接口, 用于通过PCIe总线进行通信。在至少一个实施例中, I/O单元3406实现用于与外部设备通信的接口。

[0473] 在至少一个实施例中, I/O单元3406对经由系统总线3402接收的分组进行解码。在至少一个实施例中, 至少一些分组表示被配置为使PPU 3400执行各种操作的命令。在至少一个实施例中, I/O单元3406如命令所指定的那样将解码的命令发送到PPU 3400的各种其他单元。在至少一个实施例中, 命令被发送到前端单元3410和/或被发送到集线器3416或PPU 3400的其他单元, 例如一个或更多个复制引擎、视频编码器、视频解码器、电源管理单元等 (图34中未明确示出)。在至少一个实施例中, I/O单元3406配置为在PPU 3400的各种逻辑单元之间路由通信。

[0474] 在至少一个实施例中, 由主机处理器执行的程序在缓冲区中对命令流进行编码, 该缓冲区将工作负载提供给PPU 3400以进行处理。在至少一个实施例中, 工作负载包括指令和要由那些指令处理的数据。在至少一个实施例中, 缓冲区是可由主机处理器和PPU 3400两者访问 (例如, 读/写) 的存储器中的区域—主机接口单元可以配置为访问经由I/O单元3406通过系统总线3402传输的存储器请求连接到系统总线3402的系统存储器中的缓冲区。在至少一个实施例中, 主机处理器将命令流写入缓冲区, 然后将指示命令流开始的指针发送给PPU 3400, 使得前端单元3410接收指向一个或更多个命令流指针并管理一个或更多个命令流, 从命令流中读取命令并将命令转发到PPU 3400的各个单元。

[0475] 在至少一个实施例中, 前端单元3410耦合到调度器单元3412, 该调度器单元3412配置各种GPC 3418以处理由一个或更多个命令流定义的任务。在至少一个实施例中, 调度器单元3412配置为跟踪与调度器单元3412管理的各种任务有关的状态信息, 其中状态信息可以指示任务被分配给哪个GPC 3418, 任务是活跃的还是非活跃的, 与任务相关联的优先级等等。在至少一个实施例中, 调度器单元3412管理在一个或更多个GPC 3418上执行的多个任务。

[0476] 在至少一个实施例中, 调度器单元3412耦合到工作分配单元3414, 该工作分配单元3414配置为分派任务以在GPC 3418上执行。在至少一个实施例中, 工作分配单元3414跟

踪从调度器单元3412接收到的多个调度任务并且工作分配单元3414管理每个GPC 3418的待处理任务池和活跃任务池。在至少一个实施例中,待处理任务池包括多个时隙(例如32个时隙),这些时隙包含分配给要由特定的GPC 3418处理的任务;活跃任务池可包括用于由GPC 3418主动处理的任务的多个时隙(例如4个时隙),以使随着GPC 3418中的一个完成任务的执行,该任务将从GPC 3418的活动任务池中逐出,并且从待处理任务池中选择其他任务中的一个,并安排其在GPC 3418上执行。在至少一个实施例中,如果活跃任务在GPC 3418上处于空闲状态,例如在等待数据依赖性解决时,则活跃任务从GPC 3418中驱逐并返回到待处理任务池,同时选择了待处理任务池中的另一个任务并调度在GPC 3418上执行。

[0477] 在至少一个实施例中,工作分配单元3414经由XBar 3420与一个或更多个GPC 3418通信。在至少一个实施例中,XBar 3420是互连网络,其将PPU 3400的许多单元耦合到PPU 3400的其他单元,并且可以配置为将工作分配单元3414耦合到特定的GPC 3418。在至少一个实施例中,一个或更多个PPU 3400的其他单元也可以通过集线器3416连接到XBar 3420。

[0478] 在至少一个实施例中,任务由调度器单元3412管理,并由工作分配单元3414分配给GPC 3418之一。GPC 3418被配置为处理任务并产生结果。在至少一个实施例中,结果可以由GPC 3418中的其他任务消耗,通过XBar 3420路由到不同的GPC 3418或存储在存储器3404中。在至少一个实施例中,结果可以通过分区单元3422写到存储器3404中,其实现了用于向存储器3404写入数据或从存储器3404读取数据的存储器接口。在至少一个实施例中,结果可以经由高速GPU互连3408传输到另一PPU 3404或CPU。PPU 3400包括但不限于U个分区单元3422,其等于耦合到PPU 3400的分离且不同的存储器设备3404的数量。在至少一个实施例中,分区单元3422将在本文中结合图36更详细地描述。

[0479] 在至少一个实施例中,主机处理器执行驱动器核心,该驱动程序核心实现应用程序编程接口(API),该应用程序编程接口使在主机处理器上执行的一个或更多个应用程序能够调度操作以在PPU 3400上执行。在一个实施例中,多个计算应用程序由PPU 3400同时执行,并且PPU 3400为多个计算应用程序提供隔离、服务质量("QoS")和独立的地址空间。在至少一个实施例中,应用程序生成指令(例如,以API调用的形式),该指令使驱动器核心生成一个或更多个任务以供PPU 3400执行,并且驱动器核心将任务输出至由PPU 3400处理的一个或更多个流。在至少一个实施例中,每个任务包括一个或更多个相关线程组,其可以被称为线程束(warp)。在至少一个实施例中,线程束包括可以并行执行的多个相关线程(例如32个线程)。在至少一个实施例中,协作线程可以指代多个线程,包括用于执行任务并且通过共享存储器交换数据的指令,结合图36根据至少一个实施例更详细地描述了线程和协作线程。

[0480] 推理和/或训练逻辑915用于执行与一个或更多个实施例相关联的推理和/或训练操作。本文结合图9A和/或图9B提供关于推理和/或训练逻辑915的细节。在至少一个实施例中,深度学习应用程序处理器用于训练机器学习模型(诸如神经网络),以预测或推理提供给PPU 3400的信息。在至少一个实施例中,深度学习应用程序处理器3400用于基于已由另一处理器或系统或PPU 3400训练过的训练过的机器学习模型(例如,神经网络)推理或预测信息。在至少一个实施例中,PPU 3400可用于执行本文所述的一个或更多个神经网络用例。

[0481] 图35示出了根据至少一个实施例的通用处理集群("GPC") 3500。在至少一个实施

例中, GPC 3500是图34的GPC 3418。在至少一个实施例中, 每个GPC 3500包括但不限于用于处理任务的多个硬件单元, 并且每个GPC 3500包括但不限于管线管理器3502、预光栅操作单元(“PROP”) 3504、光栅引擎3508、工作分配交叉开关(“WDX”) 3516、存储器管理单元(“MMU”) 3518、一个或更多个数据处理集群(“DPC”) 3506, 以及部件的任何合适组合。

[0482] 在至少一个实施例中, GPC 3500的操作由管线管理器3502控制。在至少一个实施例中, 管线管理器3502管理一个或更多个DPC 3506的配置, 以处理分配给GPC 3500的任务。在至少一个实施例中, 管线管理器3502配置一个或更多个DPC 3506中的至少一个以实现图形渲染管线的至少一部分。在至少一个实施例中, DPC 3506配置为在可编程流式多处理器(“SM”) 3514上执行顶点着色器程序。在至少一个实施例中, 管线管理器3502配置为将从工作分配单元接收的数据包路由到GPC 3500内的适当逻辑单元, 以及在至少一个实施例中, 可以将一些数据包路由到PROP 3504和/或光栅引擎3508中的固定功能硬件单元, 而可以将其其他数据包路由到DPC 3506以由原始引擎3512或SM 3514进行处理。在至少一个实施例中, 管线管理器3502配置DPC 3506中的至少一个以实现神经网络模型和/或计算管线。

[0483] 在至少一个实施例中, PROP单元3504配置为在至少一个实施例中将由光栅引擎3508和DPC 3506生成的数据路由到分区单元3422中的光栅操作(“ROP”) 单元, 上面结合图34更详细地描述。在至少一个实施例中, PROP单元3504配置为执行用于颜色混合的优化、组织像素数据、执行地址转换等等。在至少一个实施例中, 光栅引擎3508包括但不限于配置为执行各种光栅操作的多个固定功能硬件单元, 并且在至少一个实施例中, 光栅引擎3508包括但不限于设置引擎、粗光栅引擎、剔除引擎、裁剪引擎、精细光栅引擎、图块聚合引擎及其任意合适的组合。在至少一个实施例中, 设置引擎接收变换后的顶点并生成与由顶点定义的几何图元相关联的平面方程; 平面方程式被传送到粗光栅引擎以生成基本图元的覆盖信息(例如, 图块的x、y覆盖范围掩码); 粗光栅引擎的输出将传输到剔除引擎, 在剔除引擎中与z测试失败的图元相关联的片段将被剔除, 并传输到剪切引擎, 在剪切引擎中剪切位于视锥范围之外的片段。在至少一个实施例中, 将经过裁剪和剔除的片段传递给精细光栅引擎, 以基于设置引擎生成的平面方程式生成像素片段的属性。在至少一个实施例中, 光栅引擎3508的输出包括将由任何适当的实体(例如, 由在DPC 3506内实现的片段着色器)处理的片段。

[0484] 在至少一个实施例中, 包括在GPC 3500中的每个DPC 3506包括但不限于M管线控制器(“MPC”) 3510; 图元引擎3512; 一个或更多个SM 3514; 及其任何合适的组合。在至少一个实施例中, MPC 3510控制DPC 3506的操作, 将从管线管理器3502接收的分组路由到DPC 3506中的适当单元。在至少一个实施例中, 将与顶点相关联的分组路由到图元引擎3512, 图元引擎3512配置为从存储器中获取与顶点关联的顶点属性; 相反, 可以将与着色器程序相关联的数据包发送到SM 3514。

[0485] 在至少一个实施例中, SM 3514包括但不限于可编程流式处理器, 其配置为处理由多个线程表示的任务。在至少一个实施例中, SM 3514是多线程的并且配置为同时执行来自特定线程组的多个线程(例如32个线程), 并且实现单指令、多数据(“SIMD”) 架构, 其中将一组线程(例如, 线程束)中的每个线程配置为基于相同的指令集来处理不同的数据集。在至少一个实施例中, 线程组中的所有线程执行相同的指令。在至少一个实施例中, SM 3514实施单指令、多线程(“SIMT”) 架构, 其中一组线程中的每个线程配置为基于相同指令集来处

理不同的数据集,但是其中线程组中的各个线程允许在执行期间发散。在至少一个实施例中,为每个线程束维护程序计数器、调用栈和执行状态,从而当线程束中的线程发散时,实现线程束和线程束内的串行执行之间的并发性。在另一个实施例中,为每个单独的线程维护程序计数器、调用栈和执行状态,从而使得在线程束内和线程束之间的所有线程之间具有相等的并发性。在至少一个实施例中,为每个单独的线程维持执行状态,并且可以收敛并并行地执行执行相同指令的线程以提高效率。本文更详细地描述SM 3514的至少一个实施例。

[0486] 在至少一个实施例中,MMU 3518在GPC 3500和存储器分区单元(例如,图34的分区单元3422)之间提供接口,并且MMU 3518提供虚拟地址到物理地址的转换、存储器保护以及存储器请求的仲裁。在至少一个实施例中,MMU 3518提供一个或多个转换后备缓冲区(“TLB”),用于执行虚拟地址到存储器中的物理地址的转换。

[0487] 推理和/或训练逻辑915用于执行与一个或多个实施例相关联的推理和/或训练操作。本文结合图9A和/或图9B提供关于推理和/或训练逻辑915的细节。在至少一个实施例中,深度学习应用程序处理器用于训练机器学习模型(诸如神经网络),以预测或推理提供给GPC 3500的信息。在至少一个实施例中,GPC 3500用于基于已由另一处理器或系统或GPC 3500训练过的机器学习模型(例如,神经网络)推理或预测信息。在至少一个实施例中,GPC 3500可用于执行本文所述的一个或多个神经网络用例。

[0488] 图36示出了根据至少一个实施例的并行处理单元(“PPU”)的存储器分区单元3600。在至少一个实施例中,存储器分区单元3600包括但不限于光栅操作(“ROP”)单元3602;二级(“L2”)高速缓存3604;存储器接口3606;及其任何合适的组合。存储器接口3606耦合到存储器。存储器接口3606可以实现32、64、128、1024位数据总线,或者类似的实现方式用于高速数据传输。在至少一个实施例中,PPU包括U个存储器接口3606,每对分区单元3600一个存储器接口3606,其中每对分区单元3600连接到对应的存储器设备。例如,在至少一个实施例中,PPU可以连接至多达Y个存储器设备,例如高带宽存储器堆栈或图形双数据速率版本5同步动态随机存取存储器(“GDDR5 SDRAM”)。

[0489] 在至少一个实施例中,存储器接口3606实现高带宽存储器第二代(“HBM2”)存储器接口,并且Y等于U的一半。在至少一个实施例中,HBM2存储器堆栈位于与PPU相同的物理封装上,与传统的GDDR5SDRAM系统相比,可提供大量功率并节省面积。在至少一个实施例中,每个HBM2堆栈包括但不限于四个存储器管芯,且Y等于4,每个HBM2堆栈包括每个管芯两个128位通道,用于总共8个通道和1024位的数据总线宽度。在至少一个实施例中,存储器支持单错误校正双错误检测(“SECCED”)错误校正码(“ECC”)以保护数据。ECC为对数据损坏敏感的计算应用程序提供更高的可靠性。

[0490] 在至少一个实施例中,PPU实现了多级存储器层次。在至少一个实施例中,存储器分区单元3600支持统一存储器以为中央处理单元(“CPU”)和PPU存储器提供单个统一虚拟地址空间,从而实现虚拟存储器系统之间的数据共享。在至少一个实施例中,追踪PPU对位于其他处理器上的存储器的访问频率,以确保将存储器页面移动到更频繁地访问页面的PPU的物理存储器。在至少一个实施例中,高速GPU互连3408支持地址转换服务,其允许PPU直接访问CPU的页表,并通过PPU提供对CPU存储器的完全访问。

[0491] 在至少一个实施例中,复制引擎在多个PPU之间或PPU与CPU之间传输数据。在至少

一个实施例中,复制引擎可以为未被映射到页表中的地址生成页面错误,并且存储器分区单元3600然后为页面错误提供服务,将地址映射到页表中,之后复制引擎执行传输。在至少一个实施例中,为多个处理器之间的多个复制引擎操作固定(即不可分页)存储器,从而实质上减少了可用存储器。在至少一个实施例中,在硬件页面故障的情况下,可以将地址传递给复制引擎,而无需考虑是否驻留存储器页,并且复制过程是透明的。

[0492] 根据至少一个实施例,来自图34的存储器3404或其他系统存储器的数据由存储器分区单元3600获取,并将其存储在L2高速缓存3604中,L2高速缓存3604位于芯片上并且在各种GPC之间共享。在至少一个实施例中,每个存储器分区单元3600包括但不限于与对应的存储器设备相关联的L2高速缓存的至少一部分。在至少一个实施例中,在GPC内的各个单元中实现较低级别的高速缓存。在至少一个实施例中,每个SM 3514可以实现一级(“L1”)高速缓存,其中L1高速缓存是专用于特定SM 3514的私有存储器,并且从L2高速缓存3604中获取数据并将其存储在每个L1高速缓存中,用于在SM 3514的功能单元中进行处理。在至少一个实施例中,L2高速缓存3604耦合到存储器接口3606和XBar 3420。

[0493] 在至少一个实施例中,ROP单元3602执行与像素颜色有关的图形光栅操作,诸如颜色压缩、像素混合等。在至少一个实施例中,ROP单元3602结合光栅引擎3508实施深度测试,从光栅引擎3508的剔除引擎接收与像素片段相关联的样本位置的深度。在至少一个实施例中,针对在与片段关联的样本位置的深度缓冲区中的相应深度测试深度。在至少一个实施例中,如果该片段通过了针对该样本位置的该深度测试,则ROP单元3602更新深度缓冲区,并将该深度测试的结果发送给光栅引擎3508。将意识到,分区单元3600的数量可以不同于GPC的数量,因此,可以在至少一个实施例中将每个ROP单元3602耦合到每个GPC。在至少一个实施例中,ROP单元3602追踪从不同GPC接收到的分组,并且确定ROP单元3602生成的结果要通过XBar 3420路由到哪个分组。

[0494] 图37示出了根据至少一个实施例的流式多处理器(“SM”)3700。在至少一个实施例中,SM 3700是图35的SM。在至少一个实施例中,SM 3700包括但不限于指令高速缓存3702;一个或更多个调度器单元3704;寄存器文件3708;一个或更多个处理核心(“核心”)3710;一个或更多个特殊功能单元(“SFU”)3712;一个或更多个加载/存储单元(“LSU”)3714;互连网络3716;共享存储器/一级(“L1”)高速缓存3718;和/或其任何合适的组合。在至少一个实施例中,工作分配单元调度任务以在并行处理单元(“PPU”)的通用处理集群(“GPC”)上执行,并且每个任务被分配给GPC内部的特定数据处理集群(“DPC”),并且如果任务与着色器程序相关联,则将任务分配给SM 3700之一。在至少一个实施例中,调度器单元3704从工作分配单元接收任务并管理分配给SM 3700的一个或更多个线程块的指令调度。在至少一个实施例中,调度器单元3704调度线程块以作为并行线程的线程束来执行,其中每个线程块被分配至少一个线程束。在至少一个实施例中,每个线程束执行线程。在至少一个实施例中,调度器单元3704管理多个不同的线程块,将线程束分配给不同的线程块,然后在每个时钟周期内将来自多个不同的协作组的指令分派给各种功能单元(例如,处理核心3710、SFU 3712和LSU 3714)。

[0495] 在至少一个实施例中,协作组可以指用于组织通信线程组的编程模型,其允许开发者表达线程正在通信的粒度,从而能够表达更丰富、更有效的并行分解。在至少一个实施例中,协作启动API支持线程块之间的同步以执行并行算法。在至少一个实施例中,常规编

程模型的应用程序提供了用于同步协作线程的单一、简单的构造:跨线程块的所有线程的屏障(例如, `syncthreads()` 函数)。但是,在至少一个实施例中,程序员可以在小于线程块粒度的情形下来定义线程组,并在所定义的组内进行同步,以实现更高的性能、设计灵活性以及以集合组范围功能接口的形式实现软件重用。在至少一个实施例中,协作组使程序员能够以子块(即,小到单个线程)和多块粒度明确定义线程组,并执行集合操作,例如对协作组中的线程进行同步。编程模型支持跨软件边界的干净组合,从而库和实用程序功能可以在其本地环境中安全地同步,而不必进行关于收敛的假设。在至少一个实施例中,协作组图元使协作并行的新图案成为可能,包括但不限于生产者-消费者并行,机会主义并行以及整个线程块网格上的全局同步。

[0496] 在至少一个实施例中,分派单元3706配置为将指令发送到功能单元中的一个或多个,并且调度器单元3704并包括但不限于两个分派单元3706,该两个分派单元3706使得来自相同线程束的两个不同指令能够在每个时钟周期被调度。在至少一个实施例中,每个调度器单元3704包括单个分派单元3706或附加分派单元3706。

[0497] 在至少一个实施例中,每个SM 3700在至少一个实施例中包括但不限于寄存器文件3708,该寄存器文件3708为SM 3700的功能单元提供了一组寄存器。在至少一个实施例中,寄存器文件3708在每个功能单元之间划分,从而为每个功能单元分配寄存器文件3708的专用部分。在至少一个实施例中,寄存器文件3708在由SM 3700执行的不同线程束之间划分,并且寄存器文件3708为连接到功能单元的数据路径的操作数提供临时存储。在至少一个实施例中,每个SM 3700包括但不限于多个L个处理核心3710。在至少一个实施例中,SM 3700包括但不限于大量(例如128个或更多)不同的处理核心3710。在至少一个实施例中,每个处理核心3710在至少一个实施例中,包括但不限于全管线、单精度、双精度和/或混合精度处理单元,其包括但不限于浮点算术逻辑单元和整数算术逻辑单元。在至少一个实施例中,浮点算术逻辑单元实现用于浮点算术的IEEE 754-2008标准。在至少一个实施例中,处理核心3710包括但不限于64个单精度(32位)浮点核心、64个整数核心、32个双精度(64位)浮点核心和8个张量核心。

[0498] 根据至少一个实施例,张量核心配置为执行矩阵运算。在至少一个实施例中,一个或多个张量核心包括在处理核心3710中。在至少一个实施例中,张量核心配置为执行深度学习矩阵算术,例如用于神经网络训练和推理的卷积运算。在至少一个实施例中,每个张量核心在 4×4 矩阵上操作并且执行矩阵乘法和累加运算 $D = A \times B + C$,其中A、B、C和D是 4×4 矩阵。

[0499] 在至少一个实施例中,矩阵乘法输入A和B是16位浮点矩阵,并且累加矩阵C和D是16位浮点或32位浮点矩阵。在至少一个实施例中,张量核心对16位浮点输入数据进行32位浮点累加运算。在至少一个实施例中,16位浮点乘法使用64个运算,并得到全精度乘积,然后使用32位浮点加法与其他中间乘积累加起来,以进行 $4 \times 4 \times 4$ 矩阵乘法。在至少一个实施例中,张量核心用于执行由这些较小元件构成的更大的二维或更高维度的矩阵运算。在至少一个实施例中,API(诸如CUDA 9C++API)公开专门的矩阵加载、矩阵乘法和累加以及矩阵存储操作,以有效地使用来自CUDA-C++程序的张量核心。在至少一个实施例中,在CUDA级别,线程束级别接口假定跨越所有32个线程束线程的 16×16 大小的矩阵。

[0500] 在至少一个实施例中,每个SM 3700包括但不限于执行特殊功能(例如,属性评估、

倒数平方根等)的M个SFU 3712。在至少一个实施例中,SFU 3712包括但不限于配置为遍历层次树数据结构的树遍历单元。在至少一个实施例中,SFU 3712包括但不限于配置为执行纹理映射过滤操作的纹理单元。在至少一个实施例中,纹理单元配置为从存储器中加载纹理映射(例如,纹理像素的2D阵列)和采样纹理映射,以产生采样的纹理值以供由SM 3700执行的着色器程序使用。在至少一个实施例中,将纹理映射存储在共享存储器/L1高速缓存3718中。在至少一个实施例中,根据至少一个实施例,纹理单元使用mip映射(mip-maps)(例如,细节级别不同的纹理映射)来实现纹理操作(诸如过滤操作)。在至少一个实施例中,每个SM 3700包括但不限于两个纹理单元。

[0501] 在至少一个实施例中,每个SM 3700包括但不限于实现共享存储器/L1高速缓存3718与寄存器文件3708之间的加载和存储操作的N个LSU 3714。在至少一个实施例中,每个SM 3700包括但不限于互连网络3716,它将每个功能单元连接到寄存器文件3708,并且LSU 3714连接到寄存器文件3708和共享存储器/L1高速缓存3718。在至少一个实施例中,互连网络3716是交叉开关,其可以配置为将任何功能单元连接到寄存器文件3708中的任何寄存器,并且将LSU 3714连接到寄存器文件3708和共享存储器/L1高速缓存3718中的存储器位置。

[0502] 在至少一个实施例中,共享存储器/L1高速缓存3718是片上存储器的阵列,其在至少一个实施例中允许SM 3700与图元引擎之间以及SM 3700中的线程之间的数据存储和通信。在至少一个实施例中,共享存储器/L1高速缓存3718包括但不限于128KB的存储容量,并且位于从SM 3700到分区单元的路径中。在至少一个实施例中,共享存储器/L1高速缓存3718在至少一个实施例中用于高速缓存读取和写入。在至少一个实施例中,共享存储器/L1高速缓存3718、L2高速缓存和存储器中的一个或更多个是后备存储。

[0503] 在至少一个实施例中,将数据高速缓存和共享存储器功能组合到单个存储器块中,为两种类型的存储器访问提供了改进的性能。在至少一个实施例中,容量由不使用共享存储器的程序使用或将其用作高速缓存,例如如果共享存储器配置为使用一半容量,并且纹理和加载/存储操作可以使用剩余容量。根据至少一个实施例,在共享存储器/L1高速缓存3718内的集成使共享存储器/L1高速缓存3718能够用作用于流传输数据的高吞吐量管线,同时提供对频繁重用的数据的高带宽和低延迟访问。在至少一个实施例中,当配置用于通用并行计算时,与图形处理相比,可以使用更简单的配置。在至少一个实施例中,绕过固定功能图形处理单元,从而创建了更加简单的编程模型。在至少一个实施例中,在通用并行计算配置中,工作分配单元直接将线程的块分配和分布给DPC。在至少一个实施例中,块中的线程执行相同的程序,在计算中使用唯一的线程ID以确保每个线程生成唯一的结果,使用SM 3700执行程序并执行计算,使用共享存储器/L1高速缓存3718在线程之间进行通信,以及使用LSU 3714通过共享存储器/L1高速缓存3718和存储器分区单元来读写全局存储器。在至少一个实施例中,当被配置用于通用并行计算时,SM 3700向调度器单元3704写入可以用来在DPC上启动新工作的命令。

[0504] 在至少一个实施例中,PPU被包括在台式计算机、膝上型计算机、平板电脑、服务器、超级计算机、智能电话(例如,无线、手持设备)、个人数字助理(“PDA”)、数码相机、车辆、头戴式显示器、手持式电子设备等中或与之耦合。在至少一个实施例中,PPU被实现在单个半导体衬底上。在至少一个实施例中,PPU与一个或更多个其他设备(例如附加的PPU、存储

器、精简指令集计算机(“RISC”)CPU,一个或更多个存储器管理单元(“MMU”)、数模转换器(“DAC”)等)一起被包括在片上系统(“SoC”)中。

[0505] 在至少一个实施例中,PPU可以被包括在包括一个或更多个存储设备的图形卡上。该图形卡可以配置为与台式计算机主板上的PCIe插槽相连接。在至少一个实施例中,该PPU可以是包括在主板的芯片组中的集成图形处理单元(“iGPU”)。

[0506] 推理和/或训练逻辑915用于执行与一个或更多个实施例相关的推理和/或训练操作。本文结合图9A和/或图9B提供关于推理和/或训练逻辑915的细节。在至少一个实施例中,深度学习应用程序处理器用于训练机器学习模型(诸如神经网络),以预测或推理提供给SM 3700的信息。在至少一个实施例中,SM 3700用于基于已由另一处理器或系统或由SM 3700训练过的机器学习模型(例如,神经网络)推理或预测信息。在至少一个实施例中,SM 3700可用于执行一个或更多个本文所述的神经网络用例。

[0507] 在至少一个实施例中,单个半导体平台可以指唯一的单一基于半导体的集成电路或芯片。在至少一个实施例中,可以使用具有增加的连接性的多芯片模块,其模拟芯片上的操作,并且相对于利用传统的中央处理单元(“CPU”)和总线实现方式进行了实质性的改进。在至少一个实施例中,根据用户的需求,各种模块也可以分开放置或以半导体平台的各种组合放置。

[0508] 在至少一个实施例中,机器可读的可执行代码或计算机控制逻辑算法形式的计算机程序被存储在主存储器1504和/或辅助存储中。根据至少一个实施例,如果由一个或更多个处理器执行,则计算机程序使系统1500能够执行各种功能。存储器1504、存储和/或任何其他存储是计算机可读介质的可能示例。在至少一个实施例中,辅助存储可以指代任何合适的存储设备或系统,例如硬盘驱动器和/或可移除存储驱动器,其代表软盘驱动器、磁带驱动器、光盘驱动器、数字多功能盘(“DVD”)驱动器、记录设备、通用串行总线(“USB”)闪存等。在至少一个实施例中,各个先前附图的架构和/或功能是在CPU 1502;并行处理系统1512;能够具有两个CPU 1502的至少部分能力的集成电路;并行处理系统1512;芯片组(例如,设计成作为执行相关功能的单元工作并出售的一组集成电路等);和集成电路的任何适当组合的环境中实现的。

[0509] 在至少一个实施例中,各个先前附图的架构和/或功能在通用计算机系统、电路板系统、专用于娱乐目的的游戏控制台系统、专用系统等的环境中实现。在至少一个实施例中,计算机系统1500可以采取台式计算机、膝上型计算机、平板电脑、服务器、超级计算机、智能电话(例如,无线、手持设备)、个人数字助理(“PDA”)、数码相机、车辆、头戴式显示器、手持式电子设备、移动电话设备、电视、工作站、游戏机、嵌入式系统和/或任何其他类型的逻辑的形式。

[0510] 在至少一个实施例中,并行处理系统1512包括但不限于多个并行处理单元(“PPU”)1514和相关联的存储器1516。在至少一个实施例中,PPU 1514经由互连1518和交换机1520或多路复用器连接到主机处理器或其他外围设备。在至少一个实施例中,并行处理系统1512在可并行化的PPU 1514上分配计算任务,例如,作为跨多个图形处理单元(“GPU”)线程块的计算任务分布的一部分。在至少一个实施例中,在PPU 1514中的一些或全部之间共享和访问存储器(例如,用于读取和/或写入访问),尽管这种共享存储器可能引发相对于使用本地存储器和驻留在PPU 1514上的寄存器的性能损失。在至少一个实施例中,通过使

用命令(诸如__syncthreads())来同步PPU 1514的操作,其中块中的所有线程(例如,跨多个PPU 1514执行)在进行之前到达某个代码执行点。

[0511] 其他变型在本公开的精神内。因此,尽管公开的技术易于进行各种修改和替代构造,但是某些示出的其实例在附图中示出并且已经在上面进行了详细描述。然而,应理解,无意将公开内容限制为所公开的一种或更多种特定形式,而是相反,其意图是涵盖落入如所附权利要求书所定义的本公开内容的精神和范围内的所有修改、替代构造和等同物。

[0512] 除非另有说明或显然与上下文矛盾,否则在描述所公开的实施例的上下文中(特别是在所附权利要求的上下文中),术语“一”和“一个”和“该”以及类似指代的使用应被解释为涵盖单数和复数,而不是作为术语的定义。除非另有说明,否则术语“包括”、“具有”、“包含”和“含有”应被解释为开放式术语(意味着“包括但不限于”)。术语“连接”(在未经修改时指的是物理连接)应解释为部分或全部包含在内、附接到或连接在一起,即使有某些介入。除非本文另外指出,否则本文中对数值范围的引用仅旨在用作分别指代落入该范围内的每个单独值的简写方法,并且每个单独值都被并入说明书中,就如同其在本文中被单独叙述一样。除非另外指出或与上下文矛盾,否则术语“集”(例如“项目集”)或“子集”的使用应解释为包括一个或更多个成员的非空集合。此外,除非另外指出或与上下文矛盾,否则术语相应集的“子集”不一定表示对应集的适当子集,而是子集和对应集可以相等。

[0513] 除非以其他方式明确指出或与上下文明显矛盾,否则诸如“A,B和C中的至少一个”或“A,B与C中的至少一个”形式的短语之类的连接语在上下文中理解为通常用来表示项目、条款等,其可以是A或B或C,也可以是A和B和C集的任何非空子集。例如,在具有三个成员的集的说明性示例中,连接短语“A,B和C中的至少一个”和“A,B与C中的至少一个”是指以下任意集:{A},{B},{C},{A,B},{A,C},{B,C},{A,B,C}。因此,这种连接语言通常不旨在暗示某些实施例要求存在A中的至少一个,B中的至少一个和C中的至少一个。另外,除非另有说明或与上下文矛盾,否则术语“多个”表示复数的状态(例如,“多个项目”表示多个项目)。多个项目中项目的数量至少为两个,但如果明确指示或通过上下文指示,则可以更多。此外,除非另有说明或从上下文中可以清楚得知,否则短语“基于”是指“至少部分地基于”而不是“仅基于”。

[0514] 除非本文另外指出或与上下文明显矛盾,否则本文描述的过程的操作可以任何合适的顺序执行。在至少一个实施例中,诸如本文所述的那些过程(或其变形和/或其组合)之类的过程在配置有可执行指令的一个或更多个计算机系统的控制下执行,并且被实现为代码(例如,可执行指令,一个或更多个计算机程序或一个或更多个应用程序),该代码通过硬件或其组合在一个或更多个处理器上共同执行。在至少一个实施例中,代码以例如计算机程序的形式存储在计算机可读存储介质上,该计算机程序包括可由一个或更多个处理器执行的多个指令。在至少一个实施例中,计算机可读存储介质是非暂时性计算机可读存储介质,其排除了暂时性信号(例如,传播的瞬态电或电磁传输),但包括非暂时性数据存储电路(例如,缓冲区、高速缓存和队列)。在至少一个实施例中,代码(例如,可执行代码或源代码)被存储在其上存储有可执行指令的一组一个或更多个非暂时性计算机可读存储介质(或用于存储可执行指令的其他存储器)上,该可执行指令在由计算机系统的一个或更多个处理器执行时(即,作为被执行的结果),使得计算机系统执行本文所述的操作。在至少一个实施例中,一组非暂时性计算机可读存储介质包括多个非暂时性计算机可读存储介质,并且多

个非暂时性计算机可读存储介质中的个体非暂时性存储介质中的一个或多个缺少全部代码,而是多个非暂时性计算机可读存储介质共同存储全部代码。在至少一个实施例中,可执行指令被执行,以使得不同的指令由不同的处理器执行,例如,非暂时性计算机可读存储介质存储指令,并且主中央处理单元(“CPU”)执行一些指令,而图形处理单元(“GPU”)执行其他指令。在至少一个实施例中,计算机系统的不同组件具有单独的处理器,并且不同的处理器执行指令的不同子集。

[0515] 因此,在至少一个实施例中,计算机系统被配置为实现单独地或共同地执行本文所述的过程的操作的一个或多个服务,并且这样的计算机系统被配置有使能实施操作的适用的硬件和/或软件。此外,实现本公开的至少一个实施例的计算机系统是单个设备,并且在另一实施例中是分布式计算机系统,其包括以不同方式操作的多个设备,使得分布式计算机系统执行本文所述的操作,并且使得单个设备不执行所有操作。

[0516] 本文提供的任何和所有示例或示例性语言(例如,“诸如”)的使用仅旨在更好地阐明本公开的实施例,并且不对公开的范围构成限制,除非另有要求。说明书中的任何语言都不应被解释为表示任何未要求保护的要素对于实践公开内容是必不可少的。

[0517] 本文引用的所有参考文献,包括出版物、专利申请和专利,均通过引用并入本文,其程度就如同每个参考文献被单独且具体地指示为以引用的方式并入本文并且其全部内容在本文中阐述一样。

[0518] 在说明书和权利要求中,可以使用术语“耦合”和“连接”以及它们的派生词。应当理解,这些术语可能不旨在作为彼此的同义词。相反,在特定示例中,“连接”或“耦合”可用于指示两个或多个元件彼此直接或间接物理或电接触。“耦合”也可能意味着两个或多个元素彼此不直接接触,但仍彼此协作或交互。

[0519] 除非另有明确说明,否则可以理解,在整个说明书中,诸如“处理”、“计算”、“计算”、“确定”等之类的术语,是指计算机或计算系统或类似的电子计算设备的动作和/或过程,其将计算系统的寄存器和/或存储器中表示为物理量(例如电子)的数据处理和/或转换为类似表示为计算系统的存储器、寄存器或其他此类信息存储、传输或显示设备中的物理量的其他数据。

[0520] 以类似的方式,术语“处理器”可以指处理来自寄存器和/或存储器的电子数据并将该电子数据转换成可以存储在寄存器和/或存储器中的其他电子数据的任何设备或存储器的一部分。作为非限制性示例,“处理器”可以是CPU或GPU。“计算平台”可以包括一个或多个处理器。如本文所使用的,“软件”进程可以包括例如随时间执行工作的软件和/或硬件实体,诸如任务、线程和智能代理。同样,每个过程可以指代多个过程,以连续地或间歇地顺序地或并行地执行指令。术语“系统”和“方法”在本文中可以互换使用,只要系统可以体现一种或更多种方法,并且方法可以被认为是系统。

[0521] 在本文件中,可以参考获得、获取、接收或将模拟或数字数据输入子系统、计算机系统或计算机实现的机器中。可以通过多种方式来完成获得、获取、接收或输入模拟和数字数据的过程,例如通过接收作为函数调用或对应用程序编程接口的调用的参数的数据。在一些实现中,可以通过经由串行或并行接口传输数据来完成获得、获取、接收或输入模拟或数字数据的过程。在另一个实现中,可以通过经由计算机网络将数据从提供实体传输到获取实体来完成获得、获取、接收或输入模拟或数字数据的过程。也可以参考提供、输出、传

送、发送或呈现模拟或数字数据。在各种示例中,提供、输出、传送、发送或呈现模拟或数字数据的过程可以通过将数据作为函数调用的输入或输出参数、应用程序编程接口或进程间通信机制的参数进行传输来实现。

[0522] 尽管以上的讨论阐述了所描述的技术的示例实现,但是其他架构可以用于实现所描述的功能,并且旨在落入本公开的范围。此外,尽管出于讨论的目的在上面定义了具体的职责分配,但是根据情况,可以以不同的方式分配和划分各种功能和职责。

[0523] 此外,尽管已经用特定于结构特征和/或方法动作的语言描述了主题,但是应当理解,所附权利要求书所要求保护的主体不必限于所描述的特定特征或动作。而是,公开了特定的特征和动作作为实现权利要求的示例性形式。

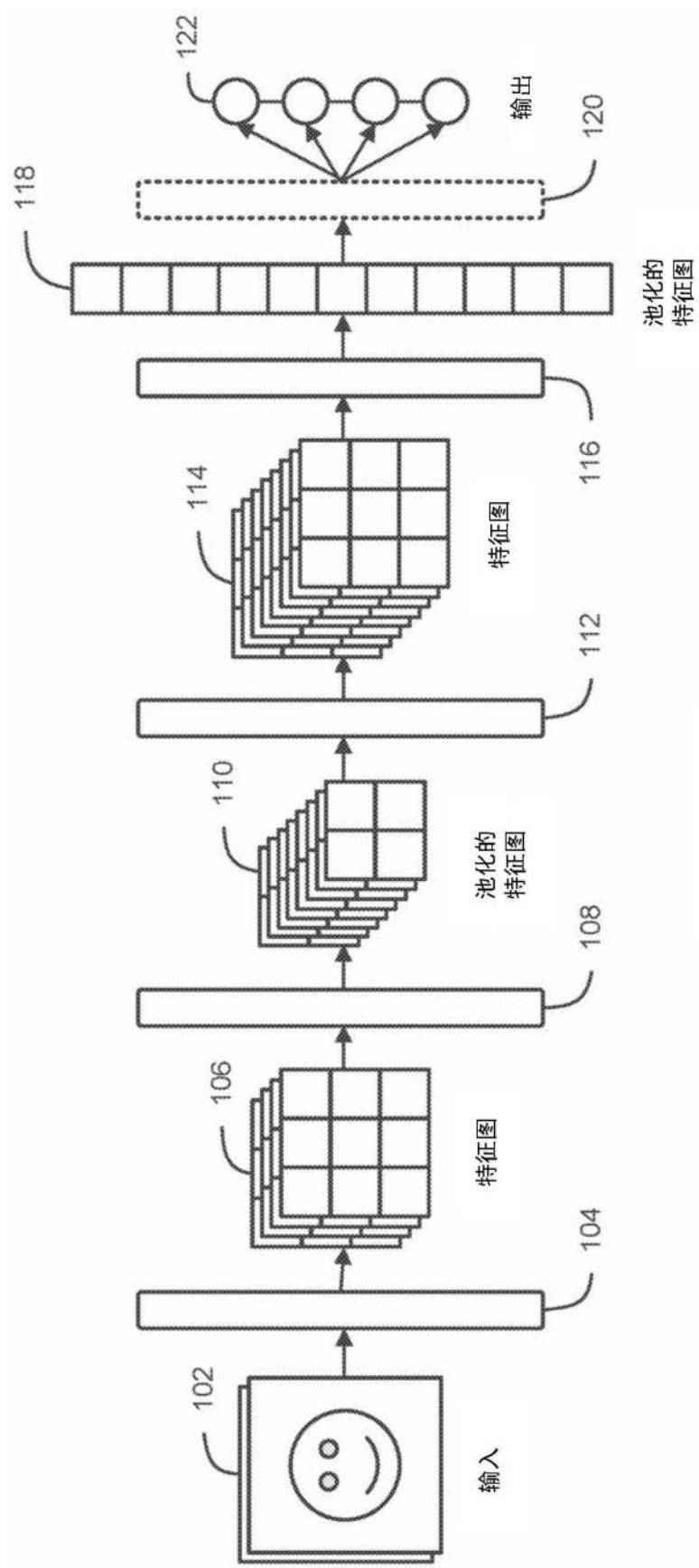


图1

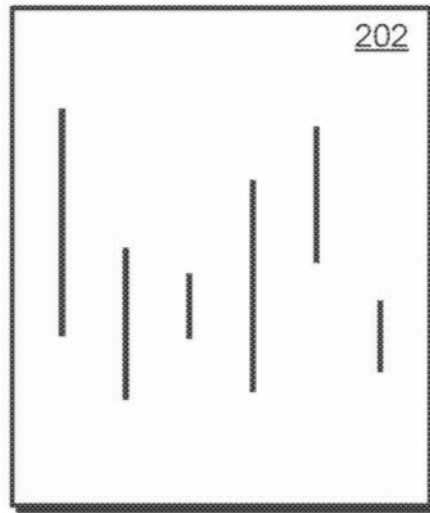


图2A



图2B

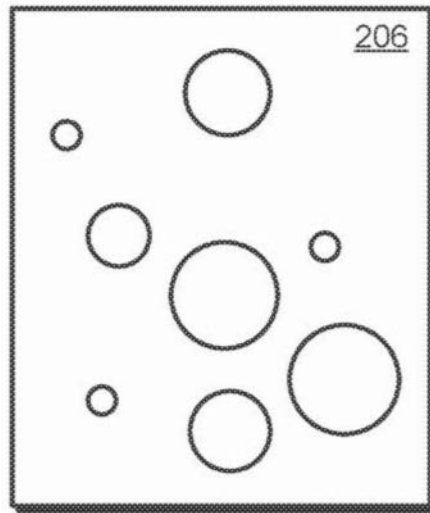


图2C

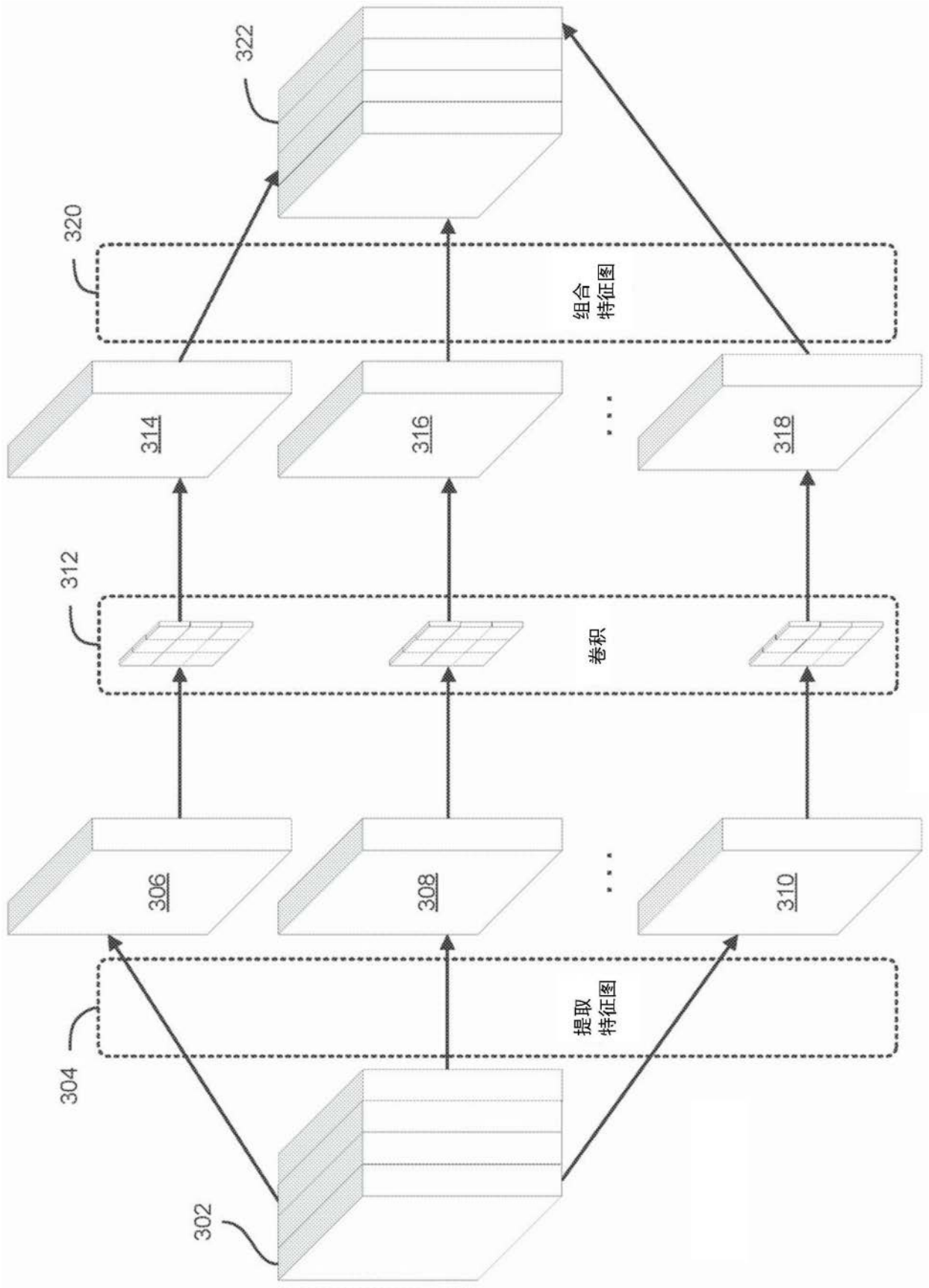


图3

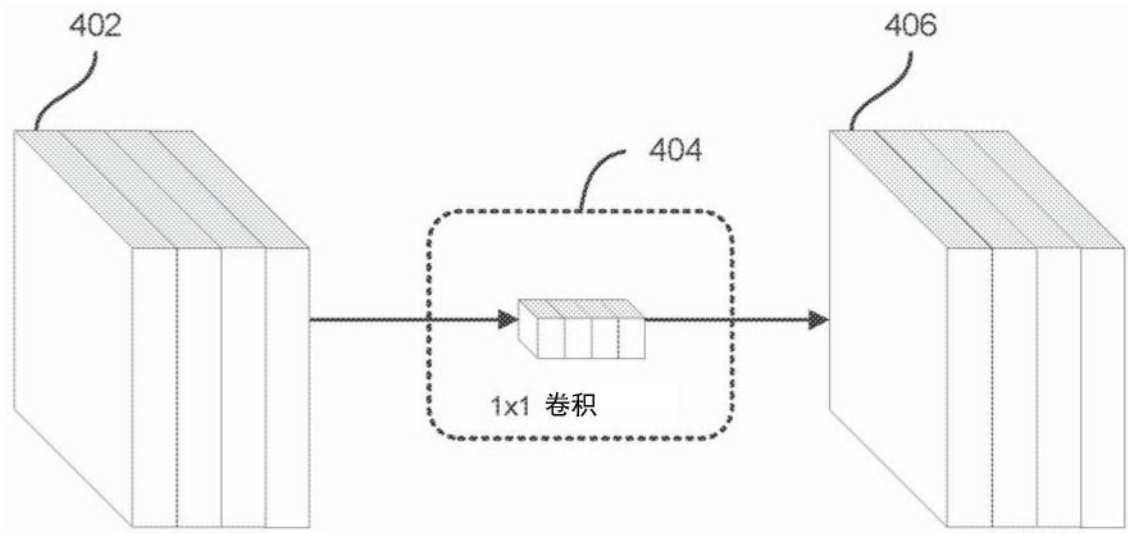


图4

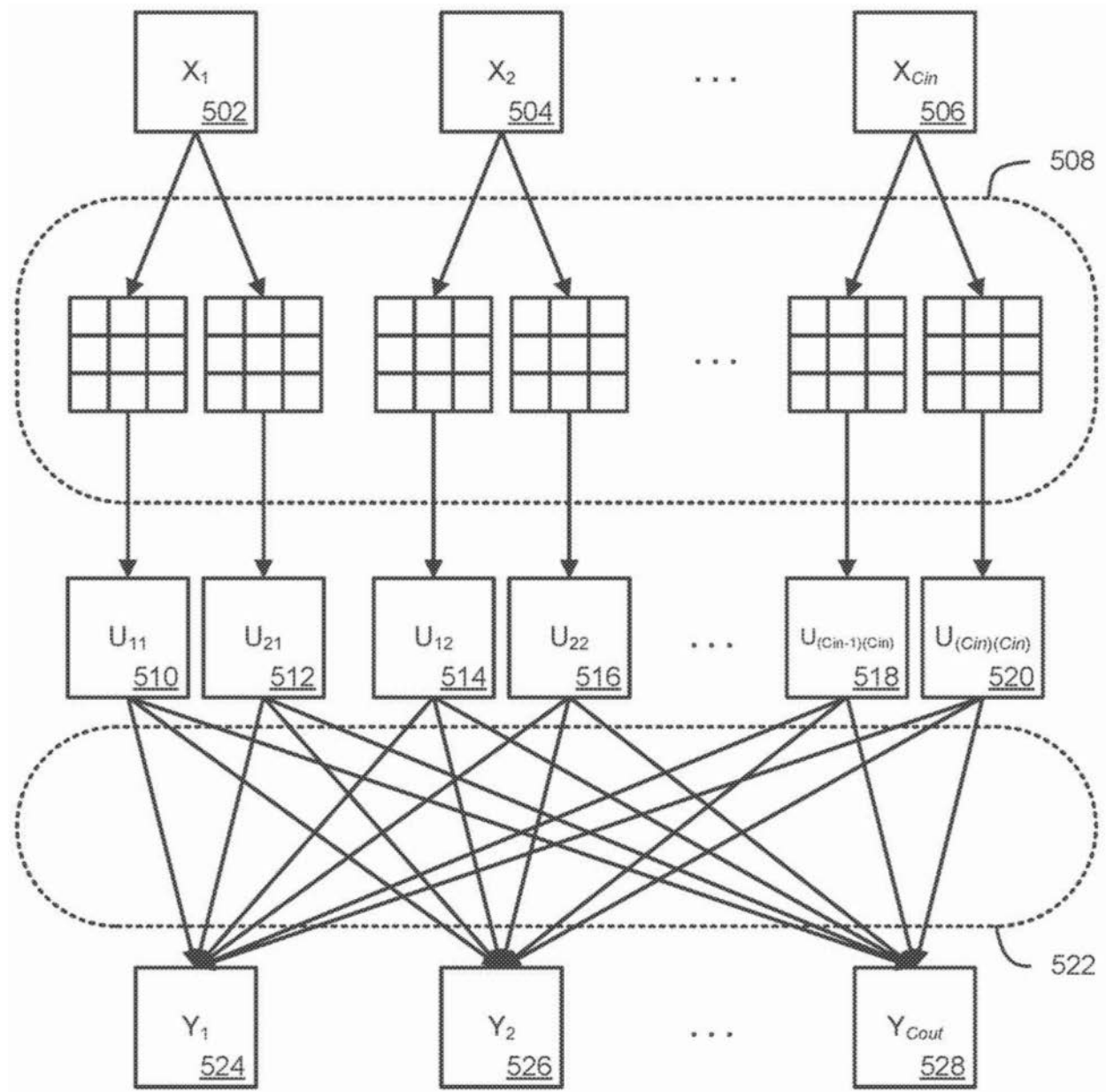


图5

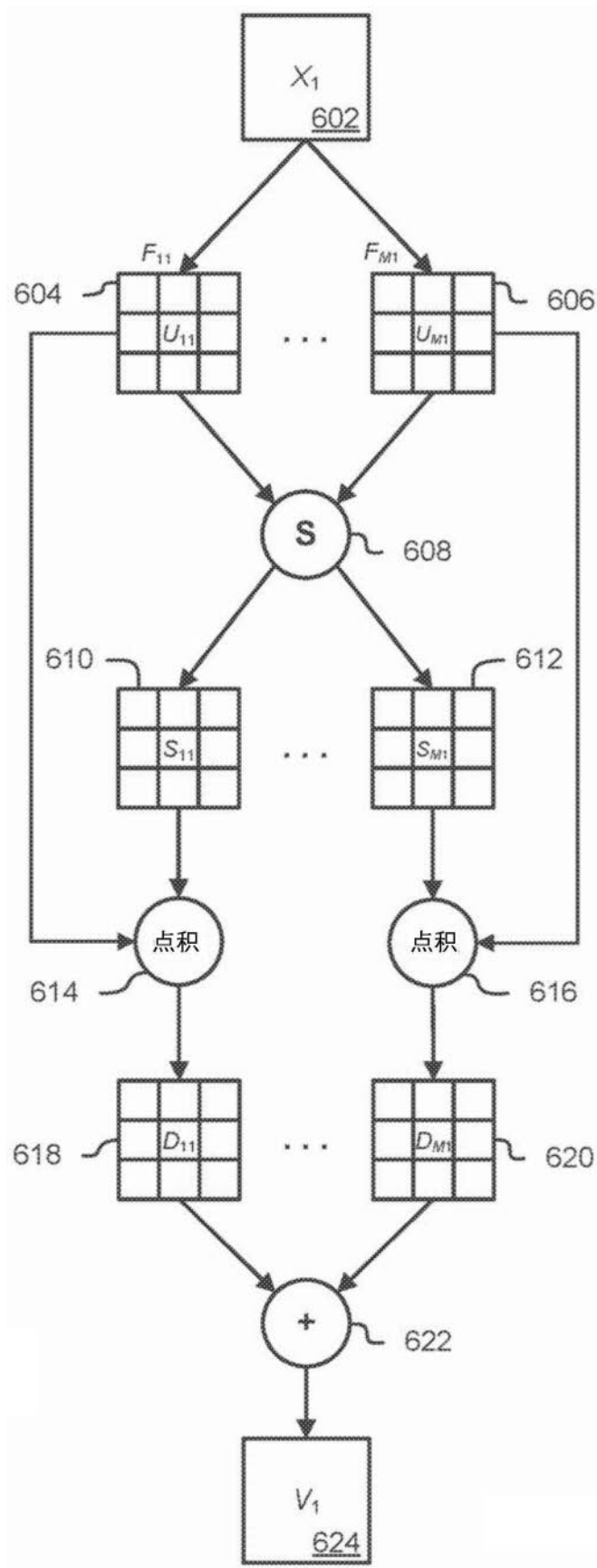


图6

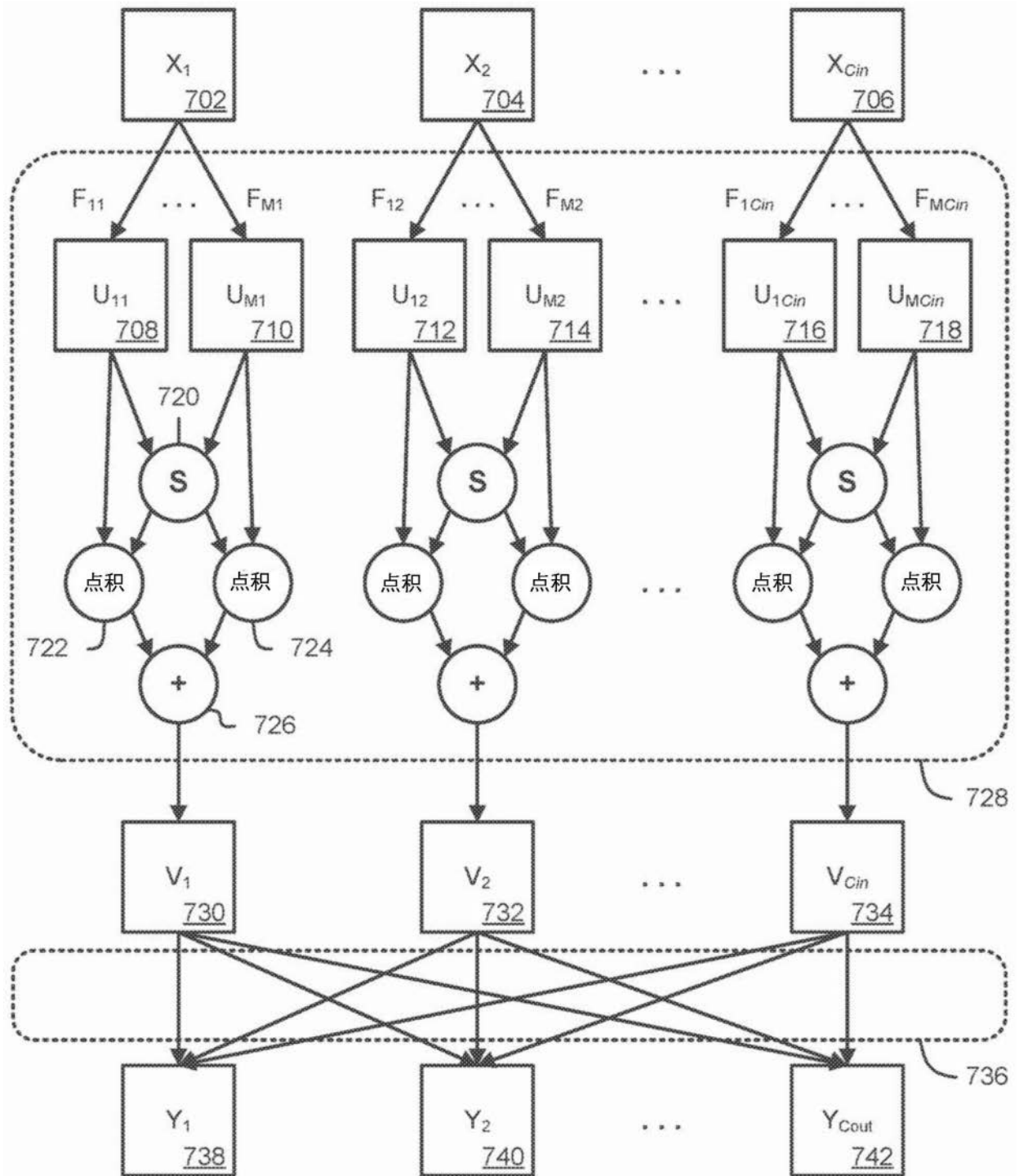


图7

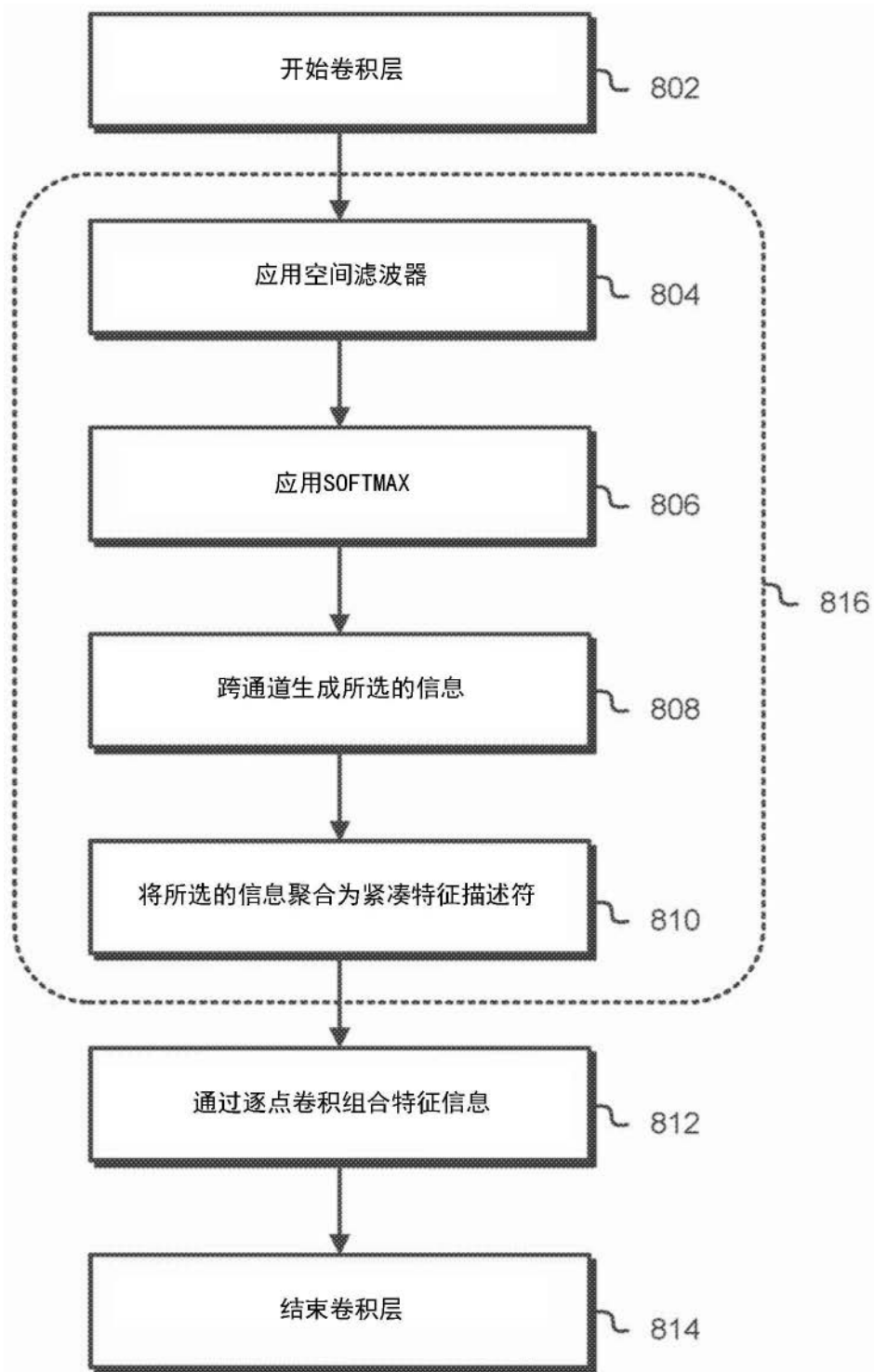


图8

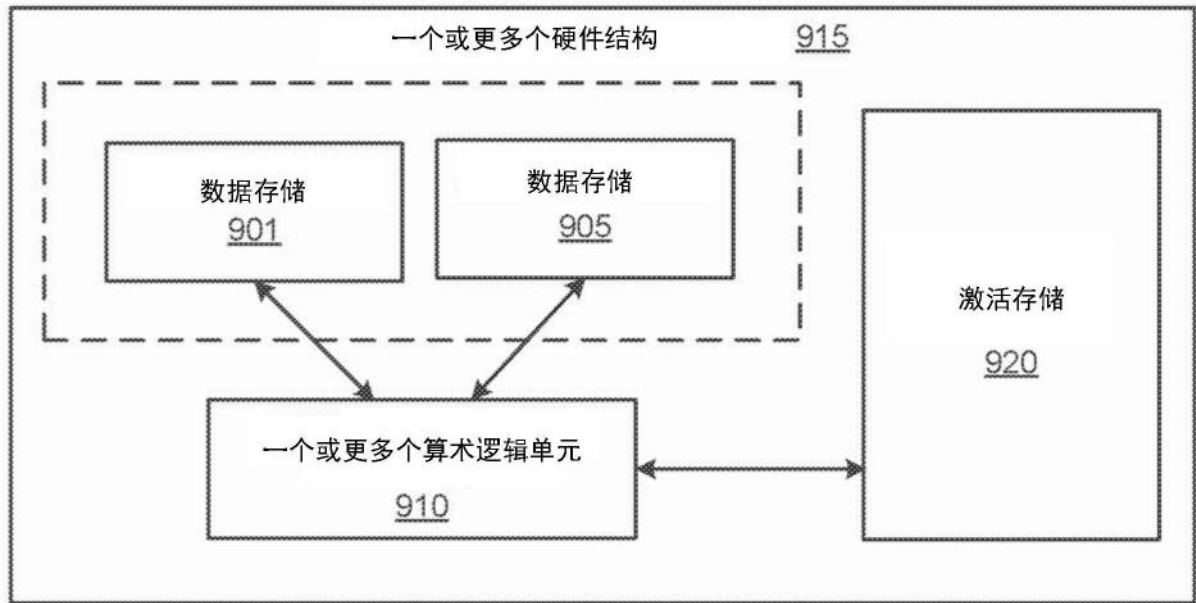


图9A

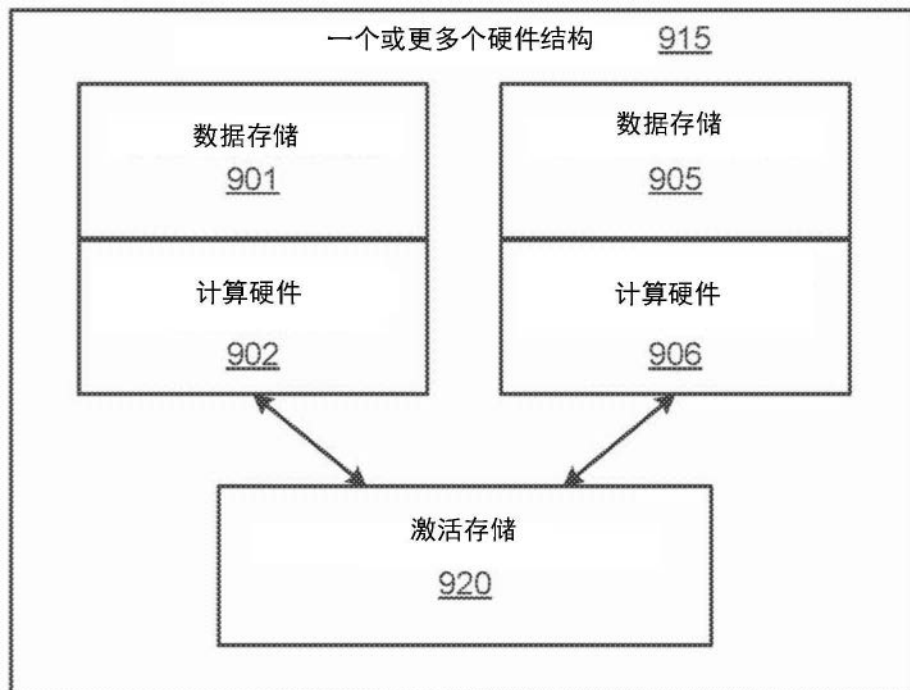


图9B

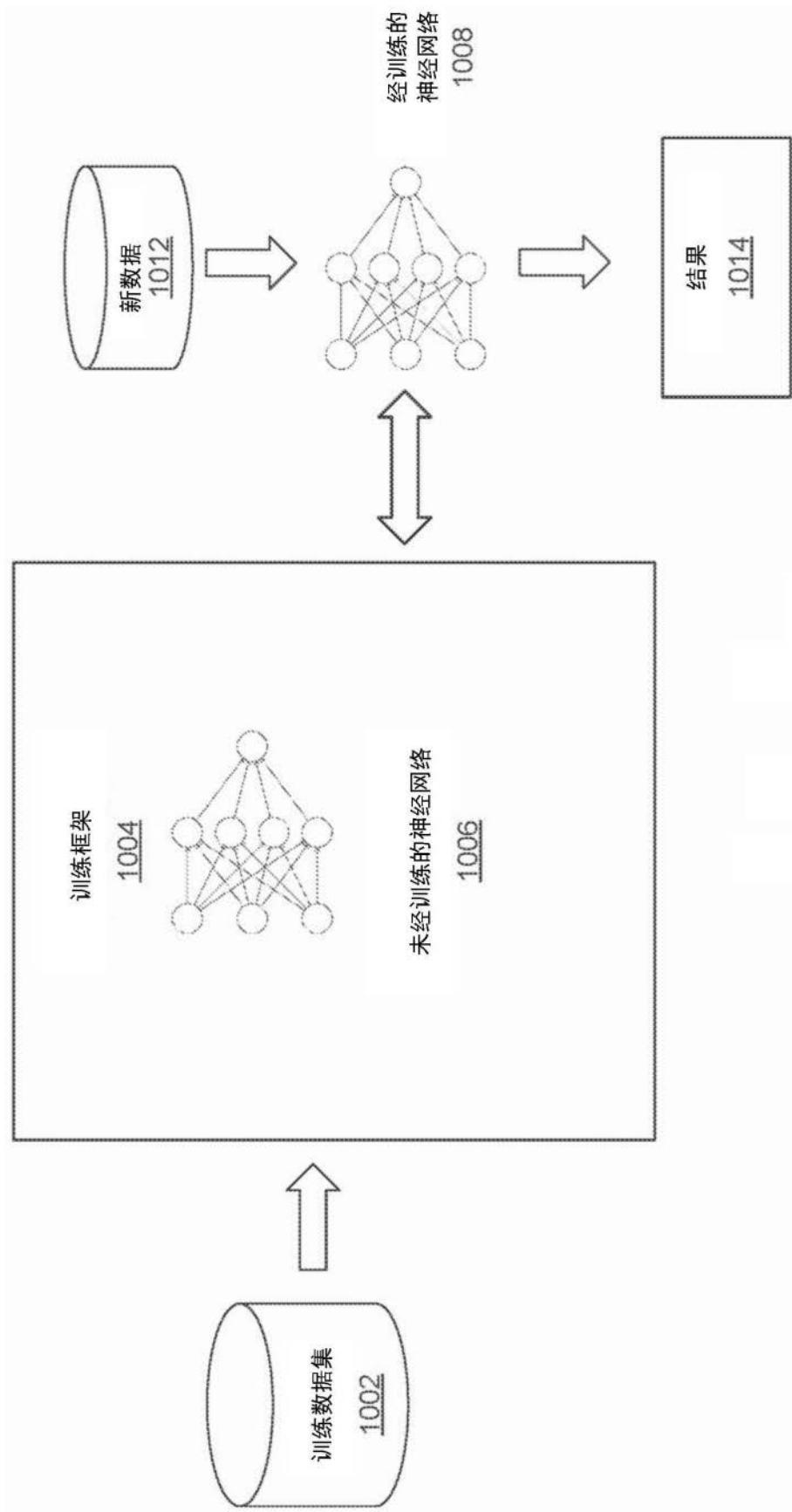


图10

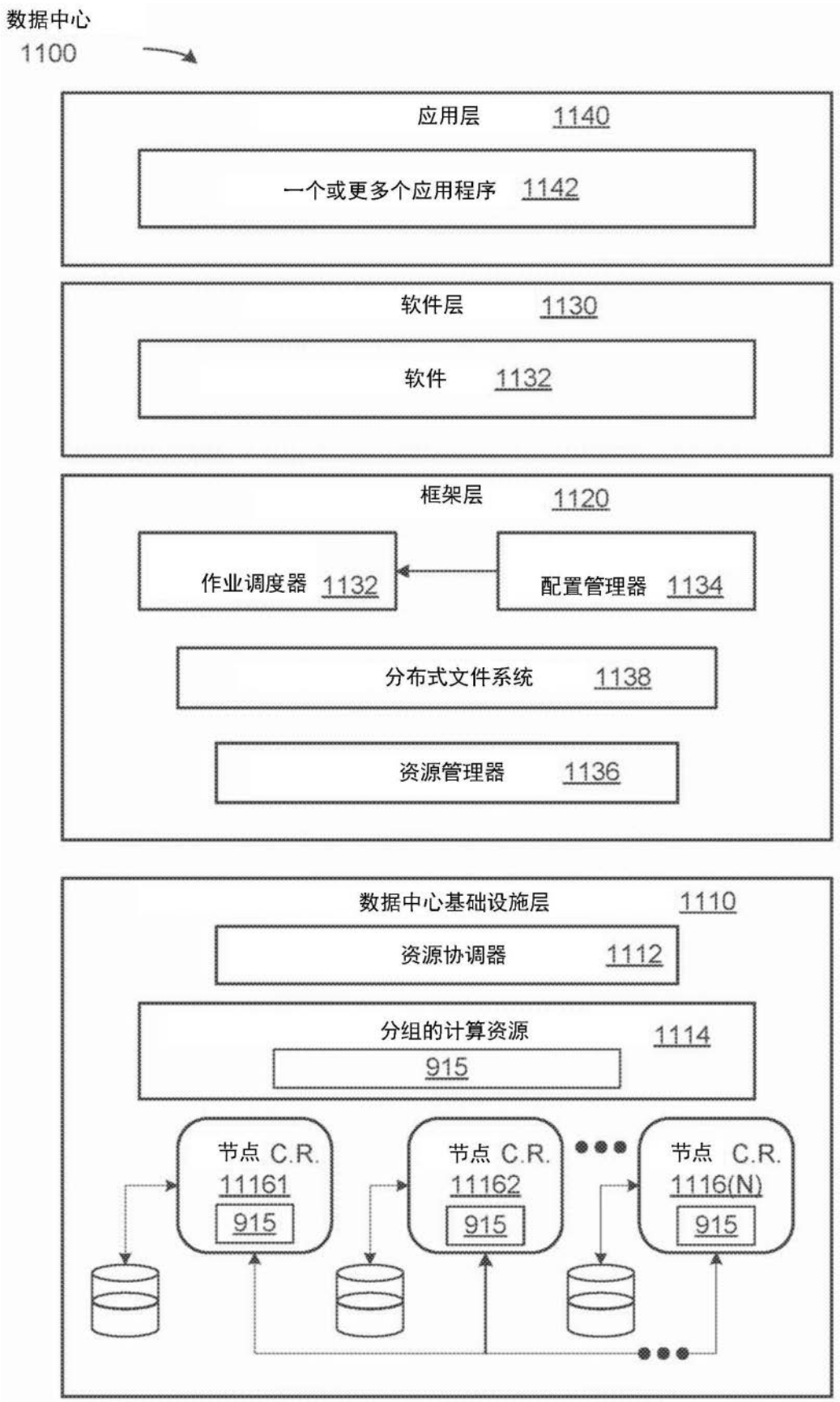


图11

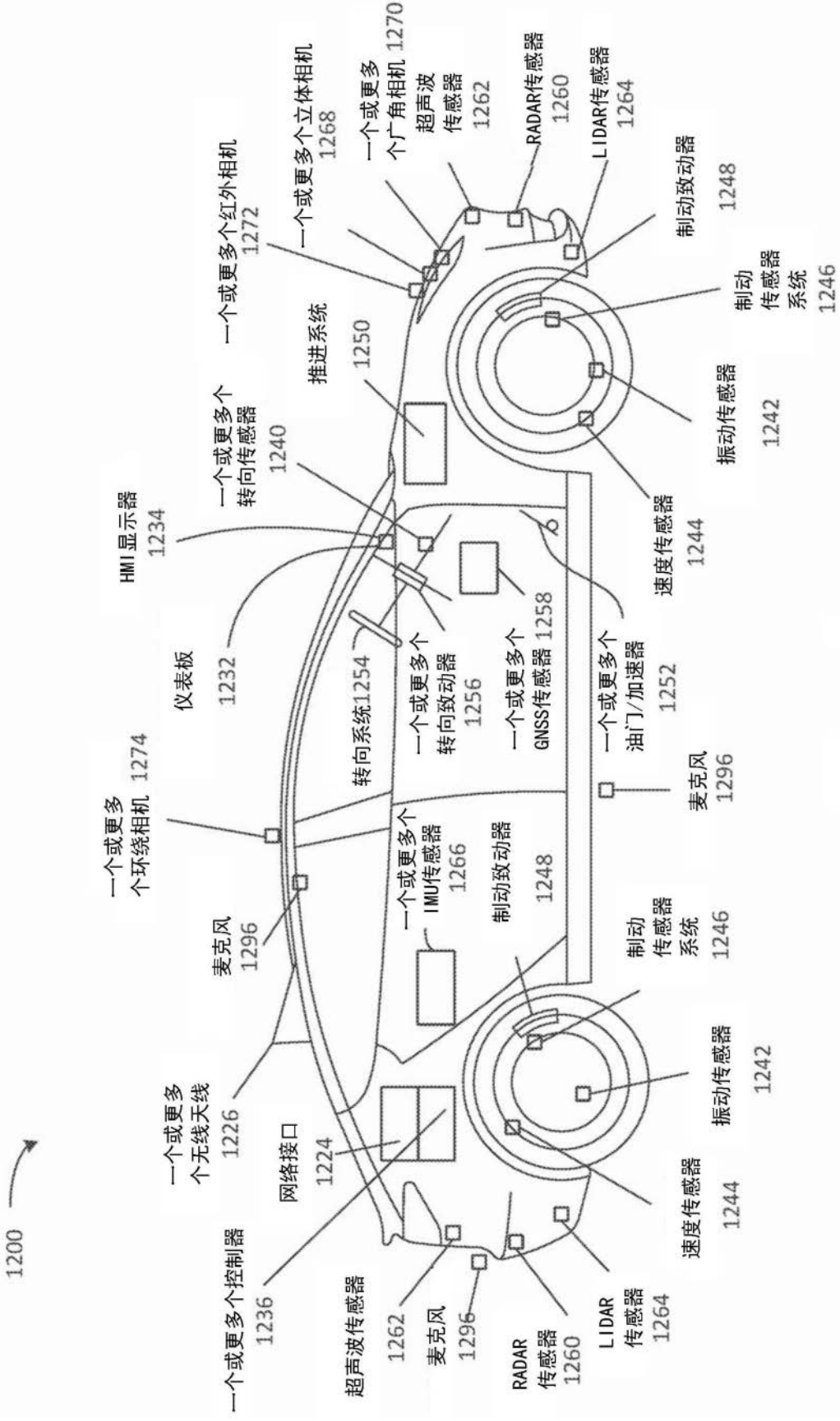


图12A

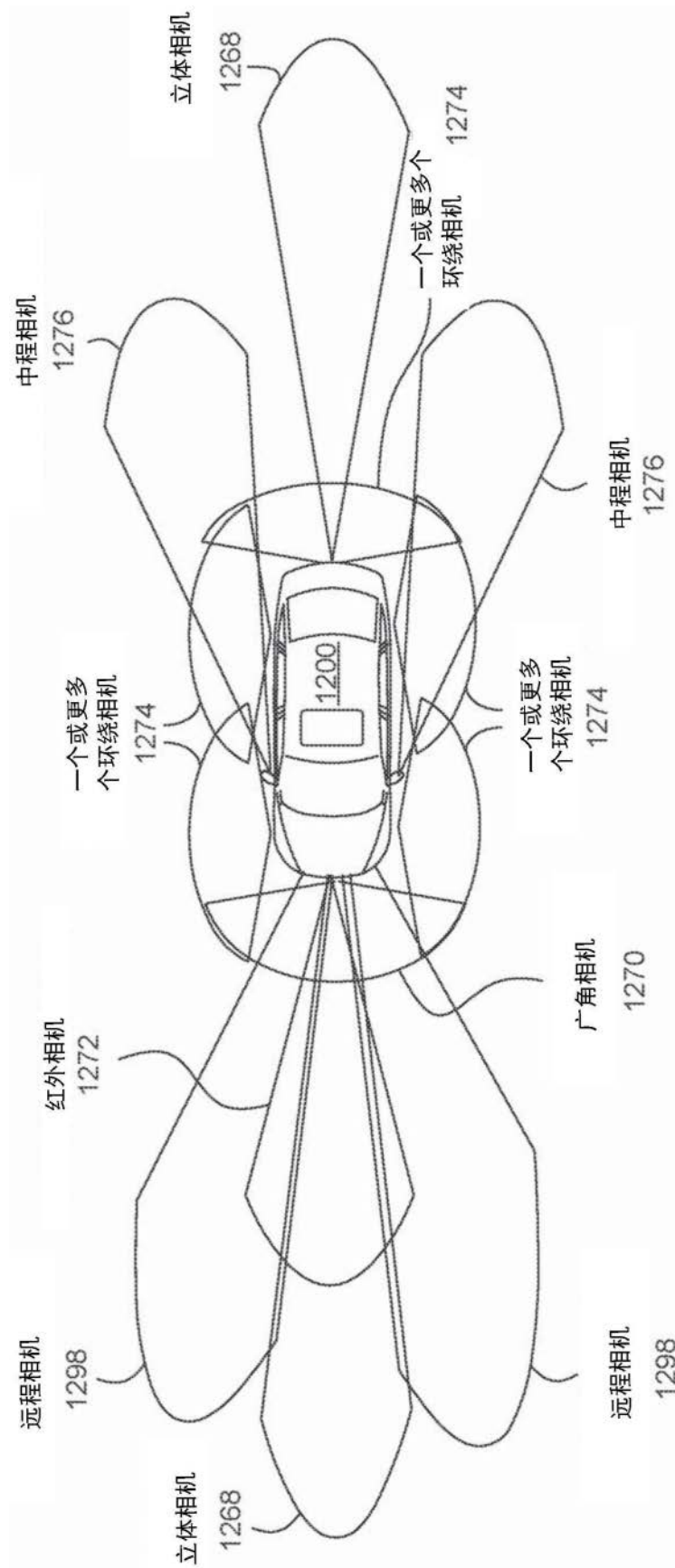


图12B

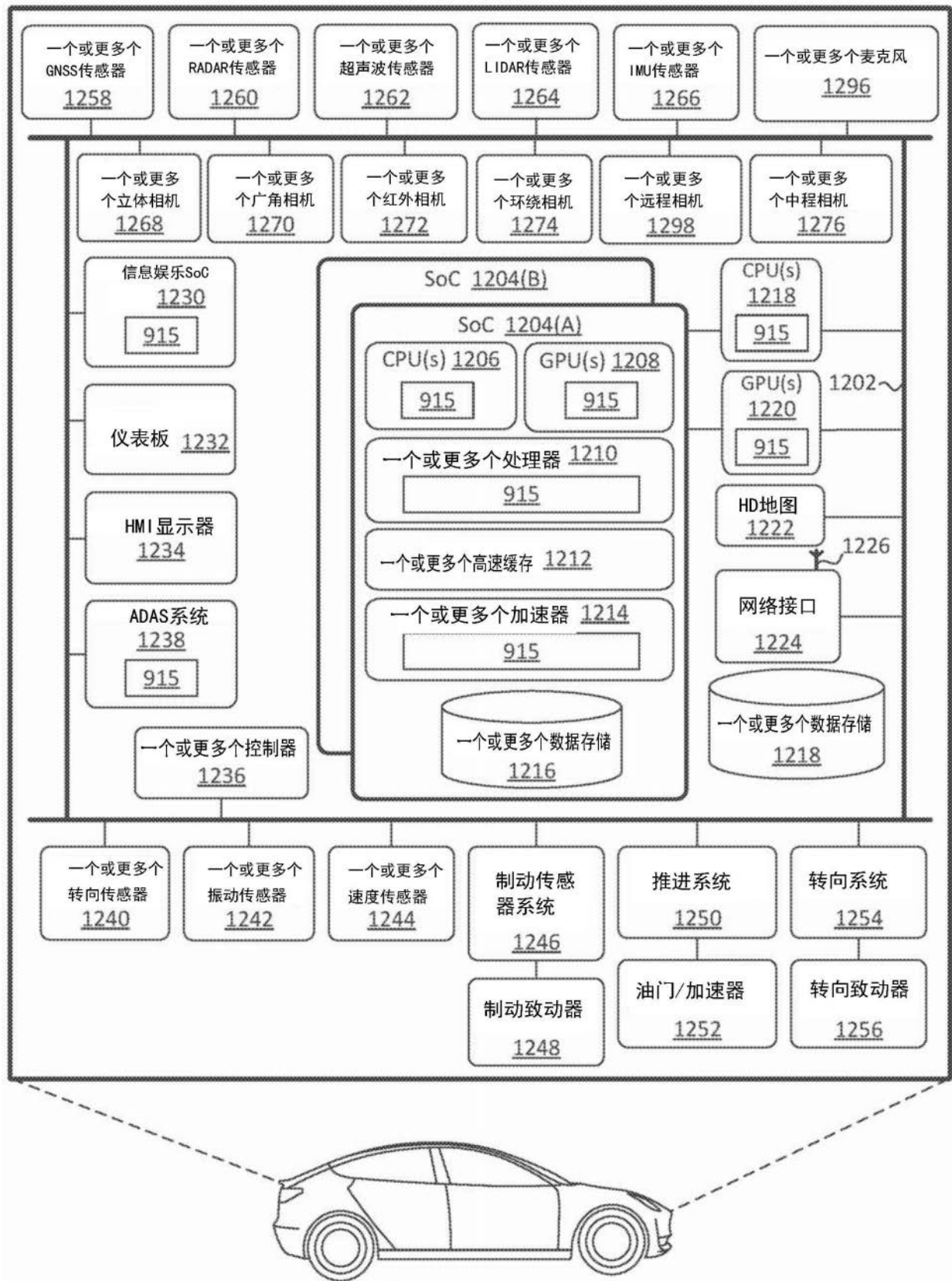


图12C

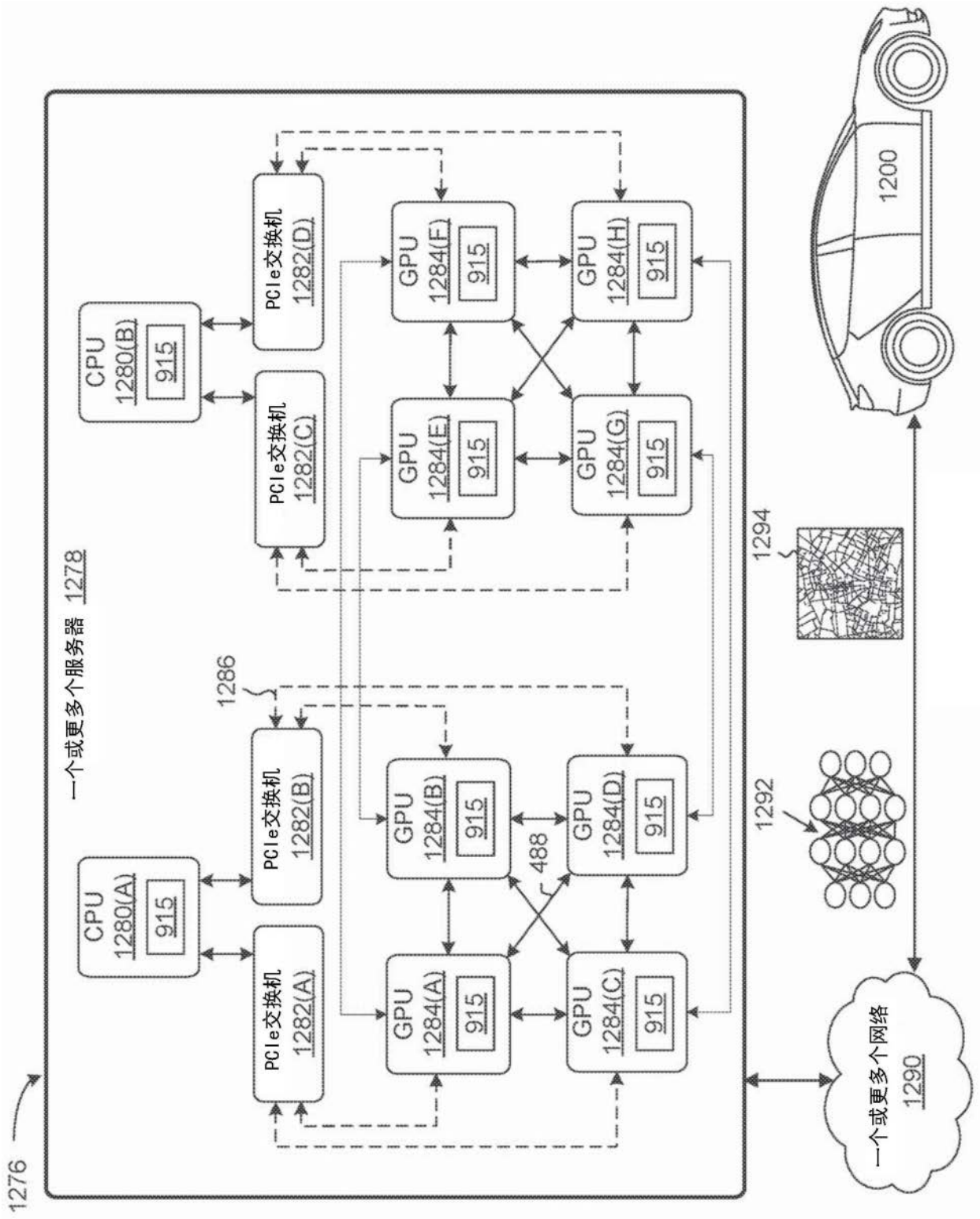


图12D

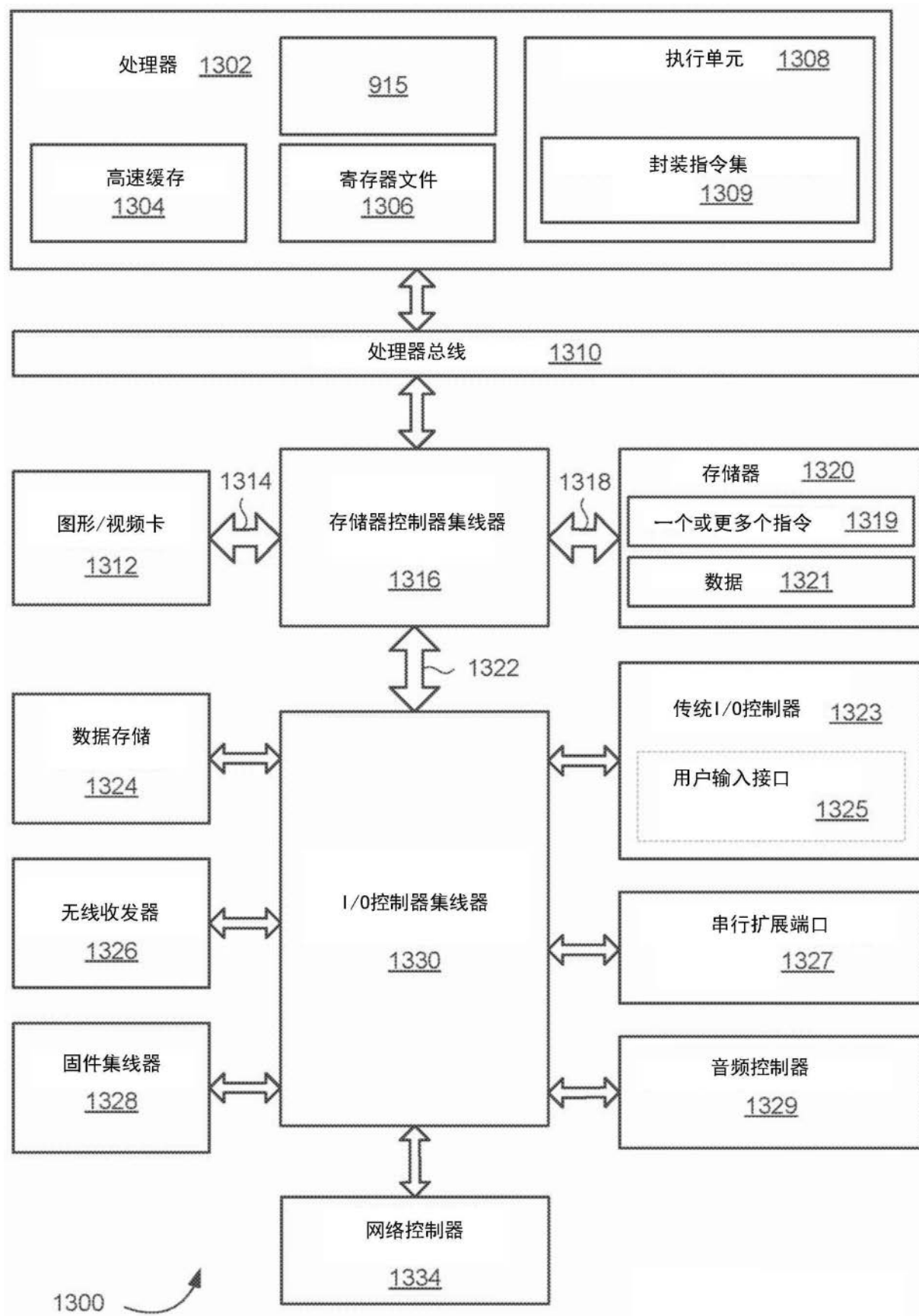


图13

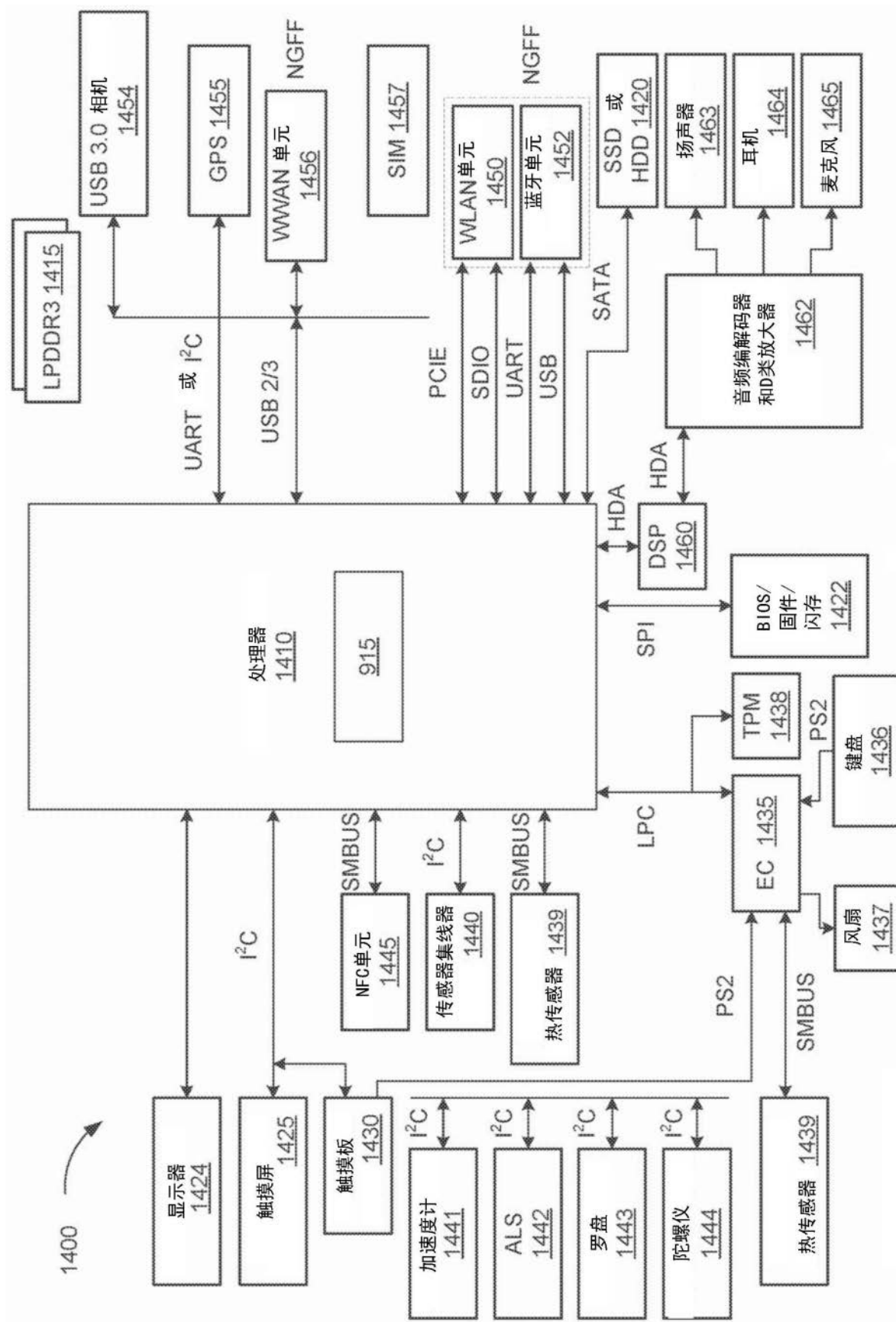


图14

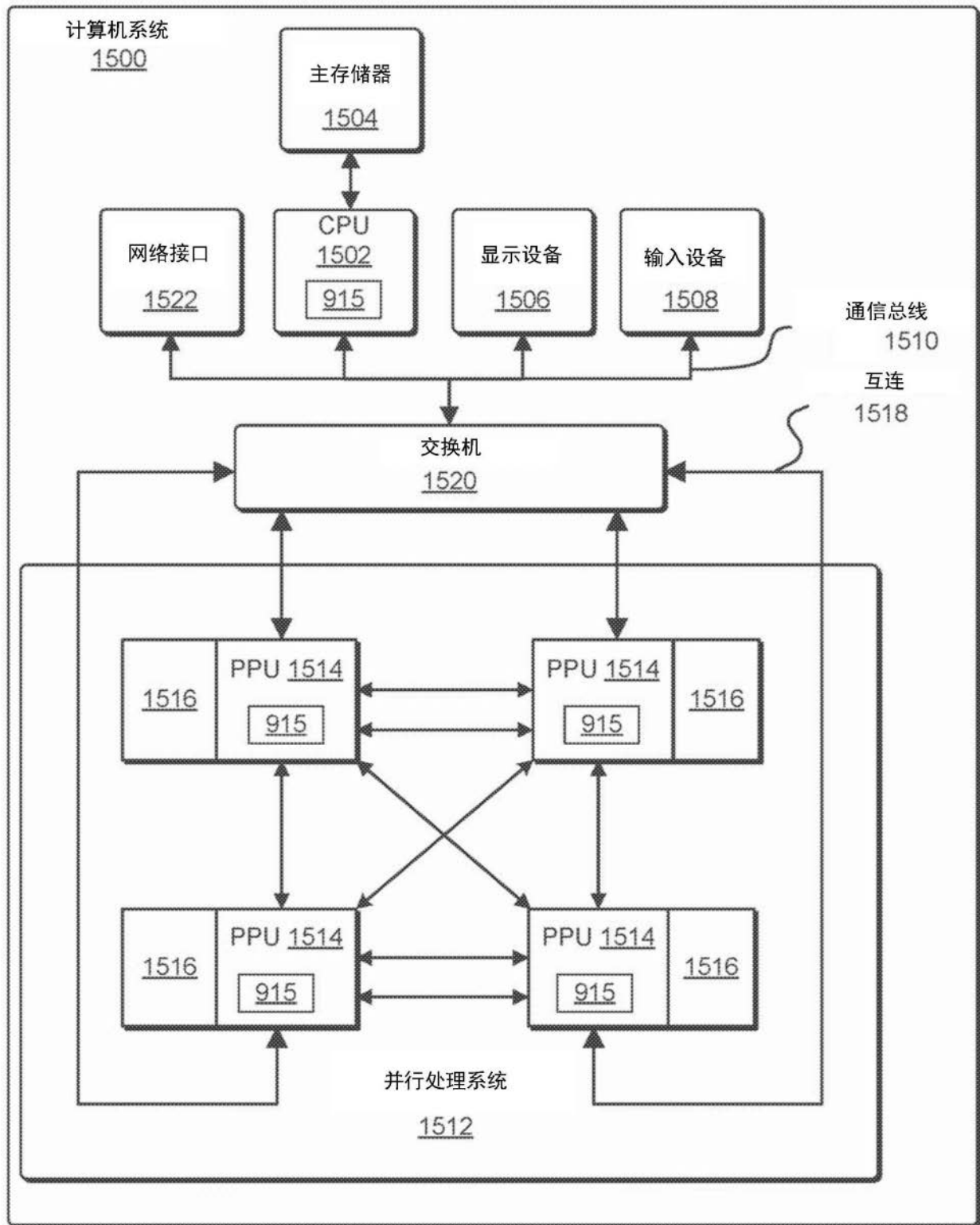


图15

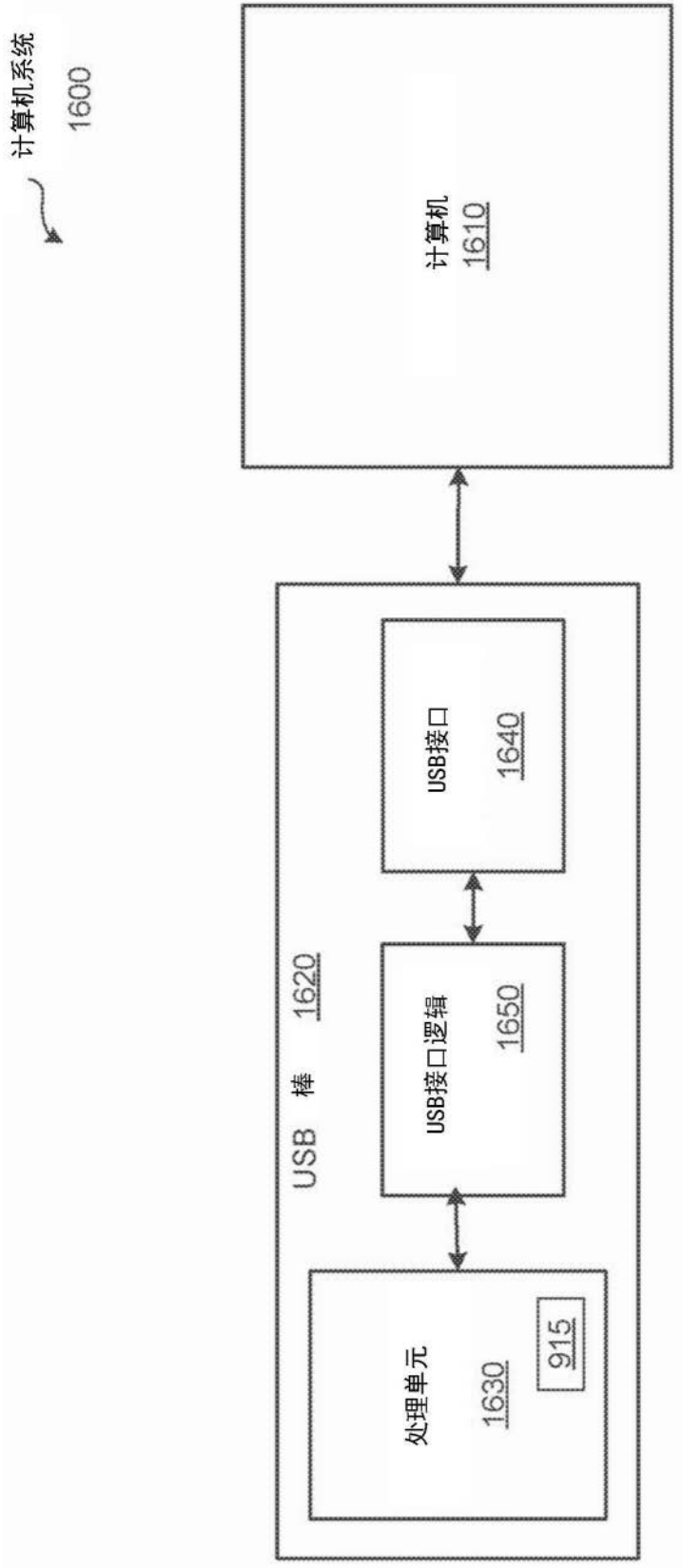


图16

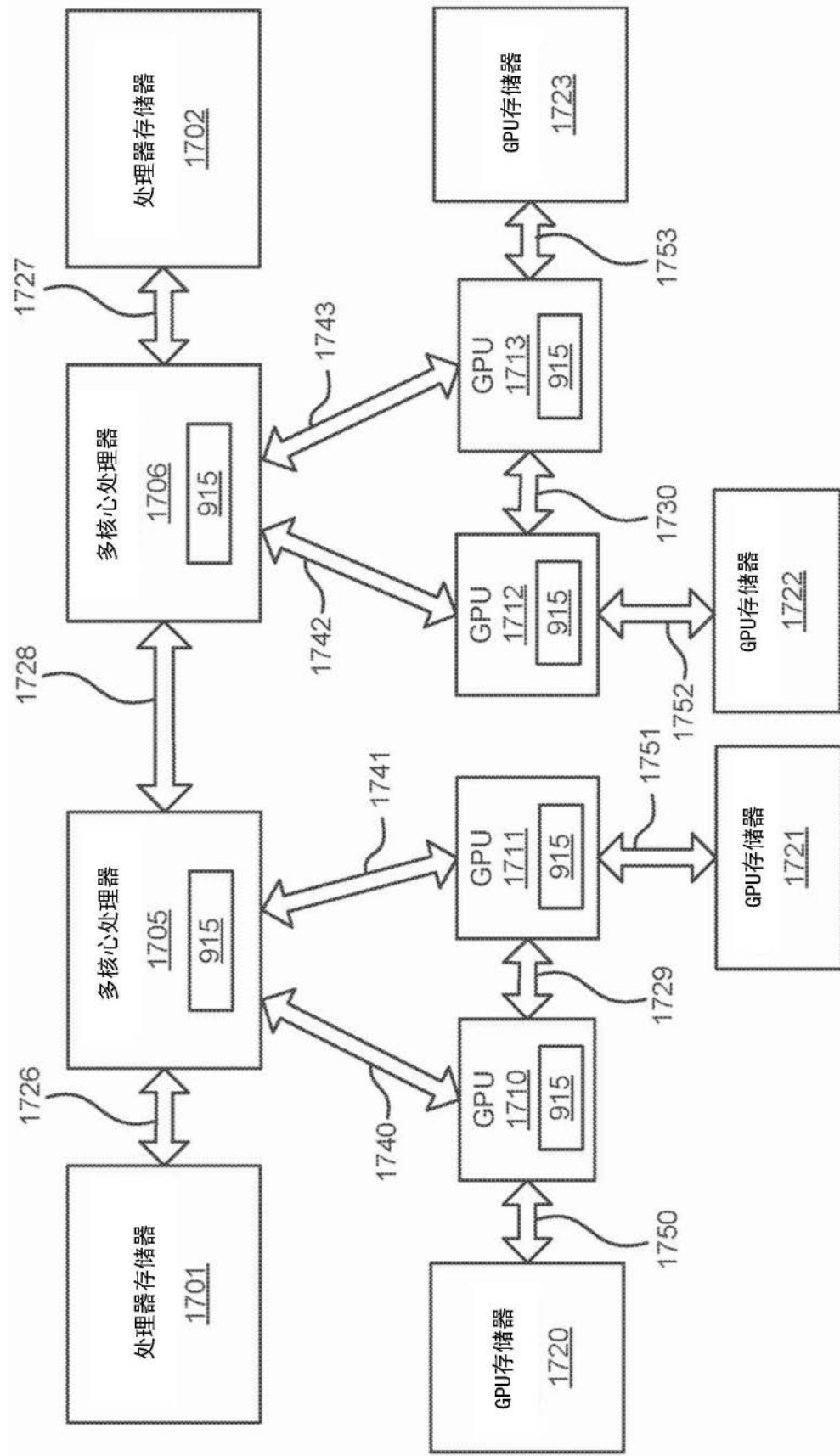


图17A

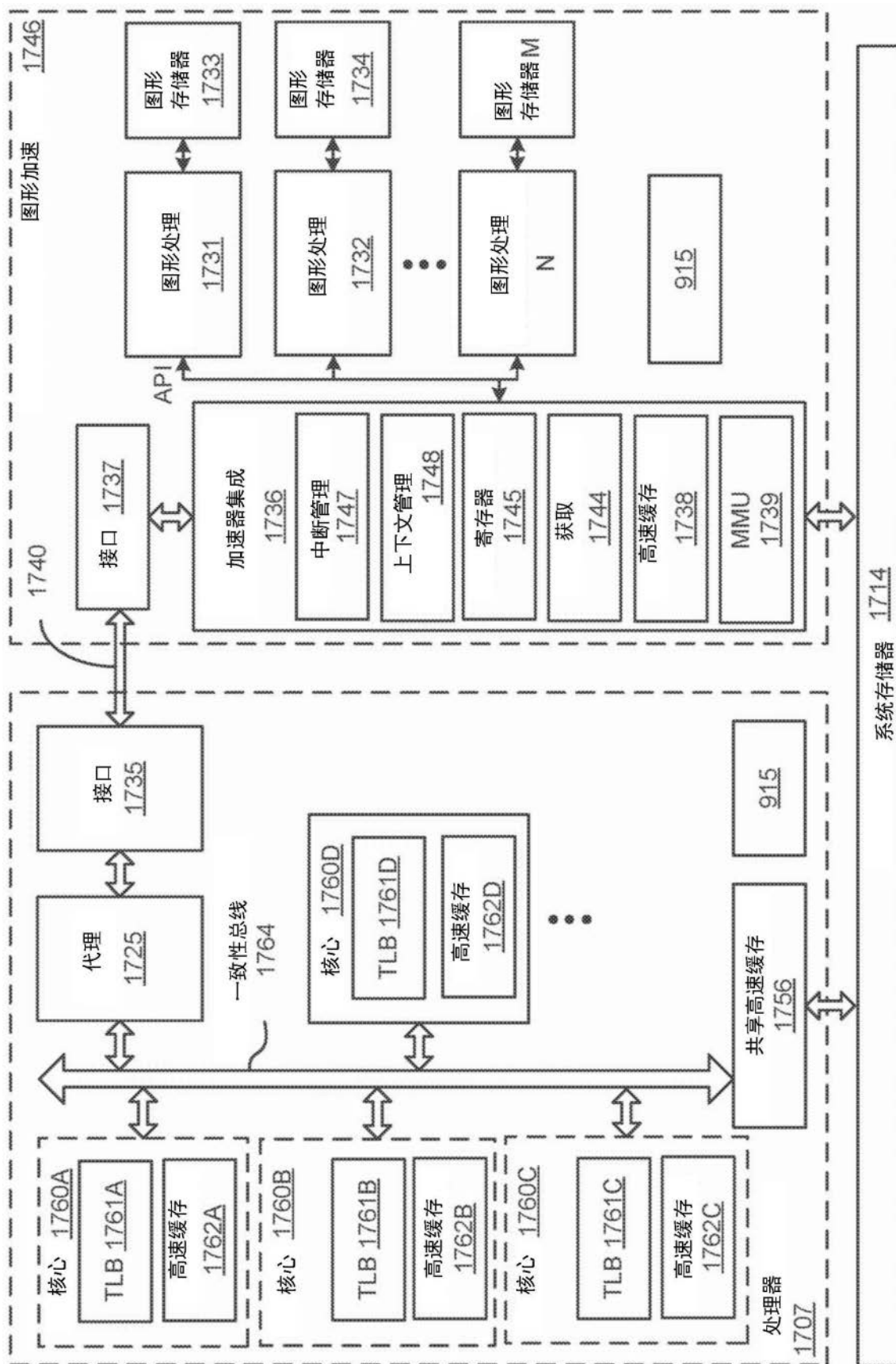


图17B

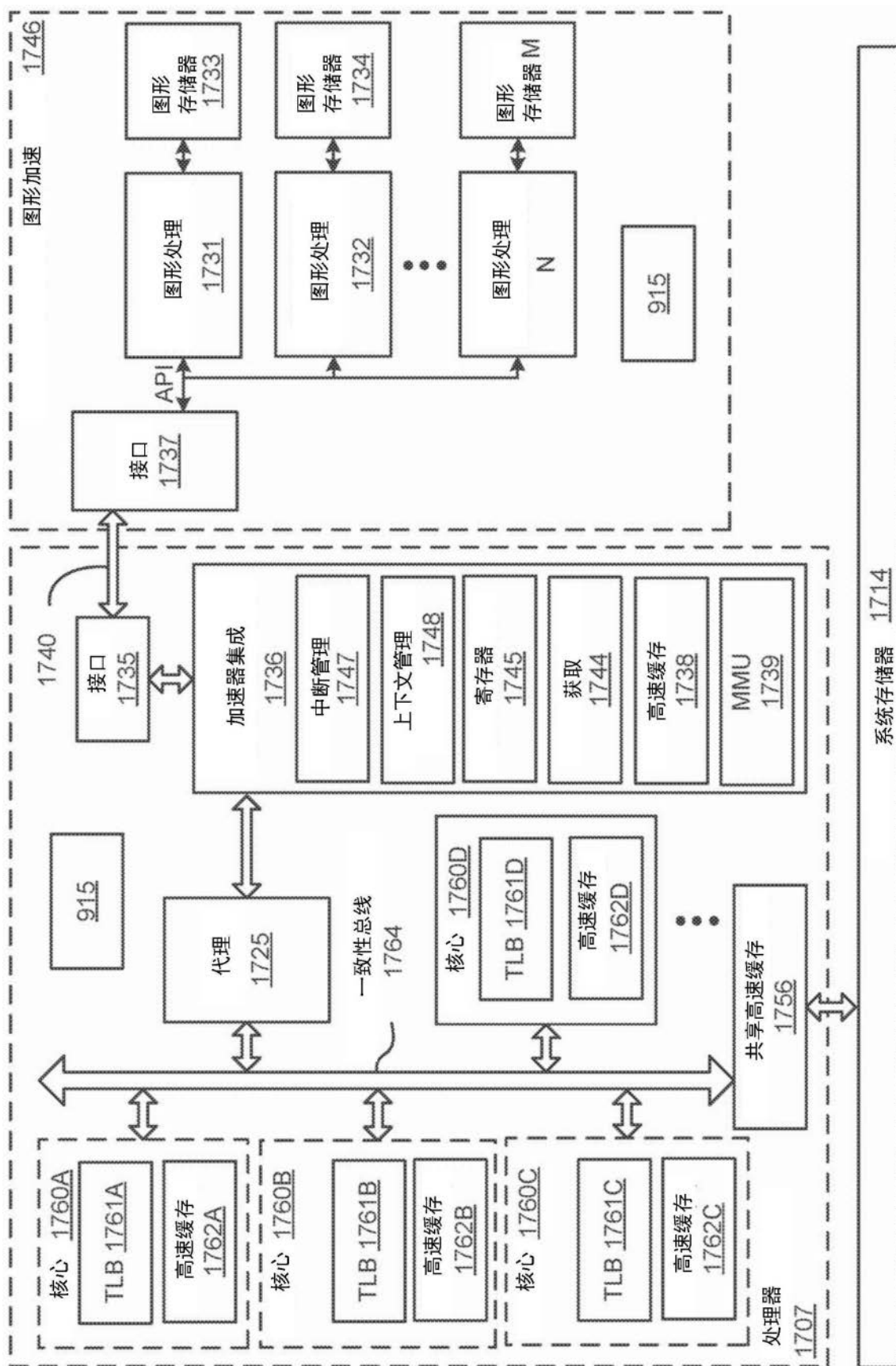


图17C

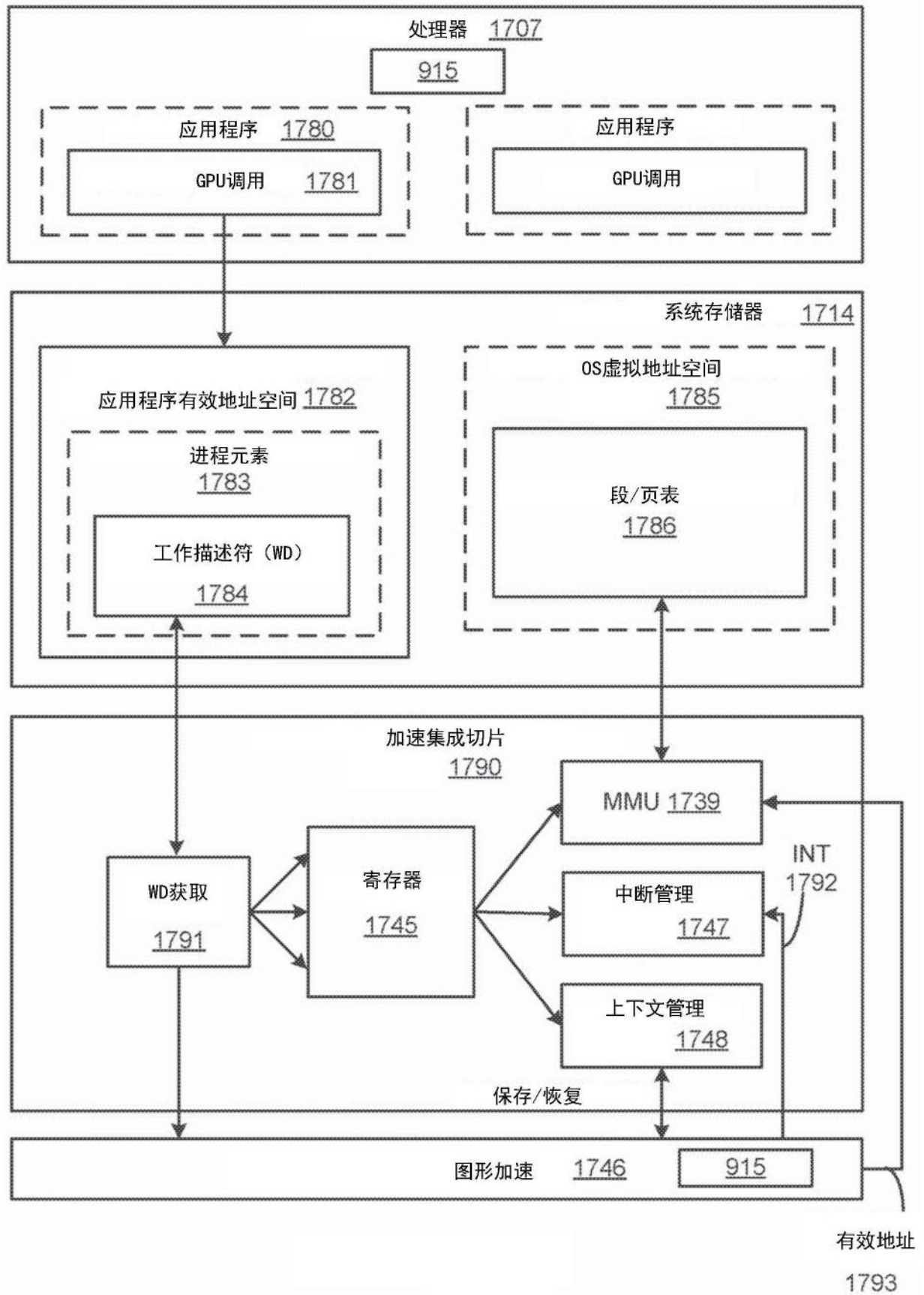


图17D

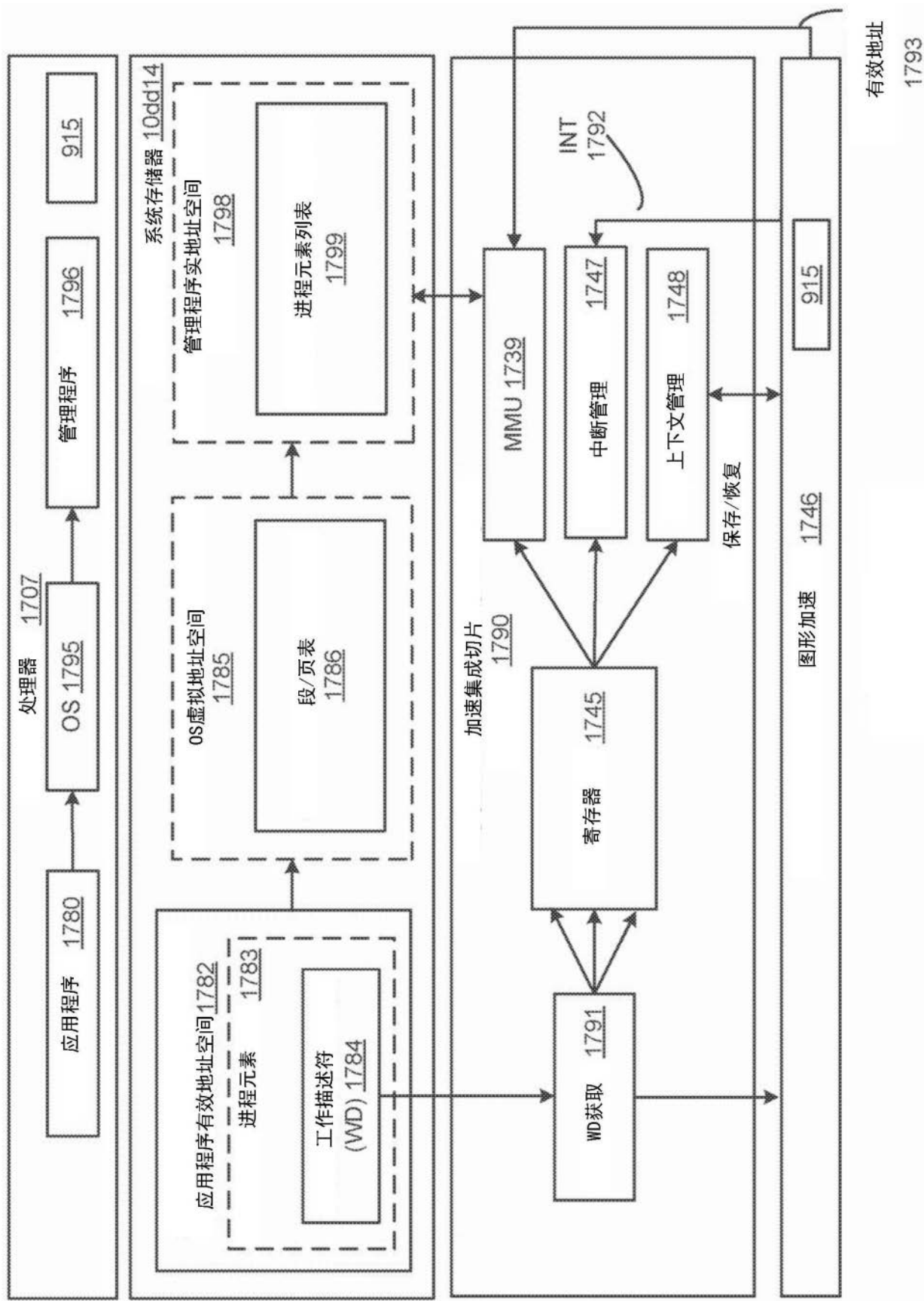


图17E

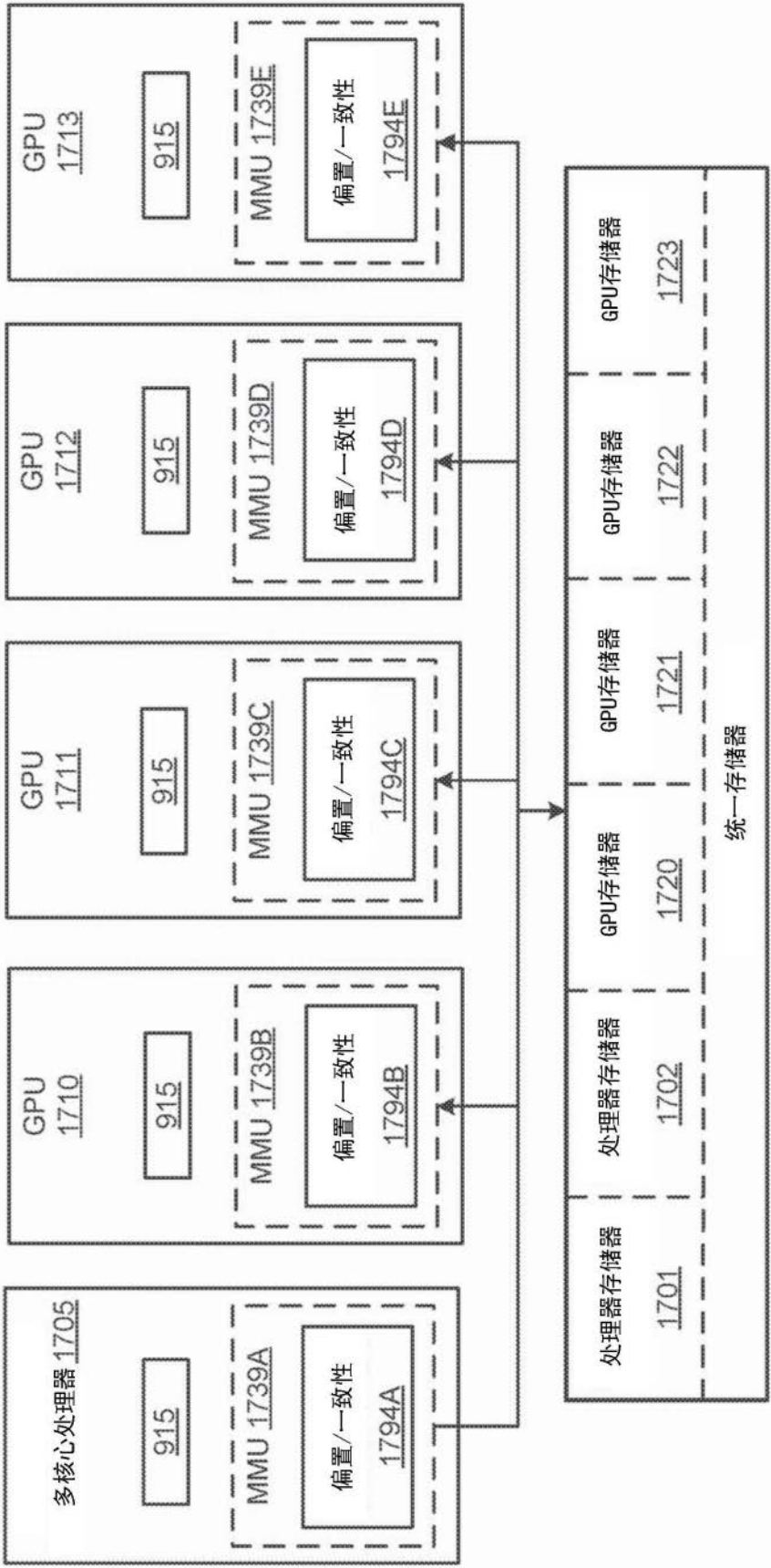


图17F

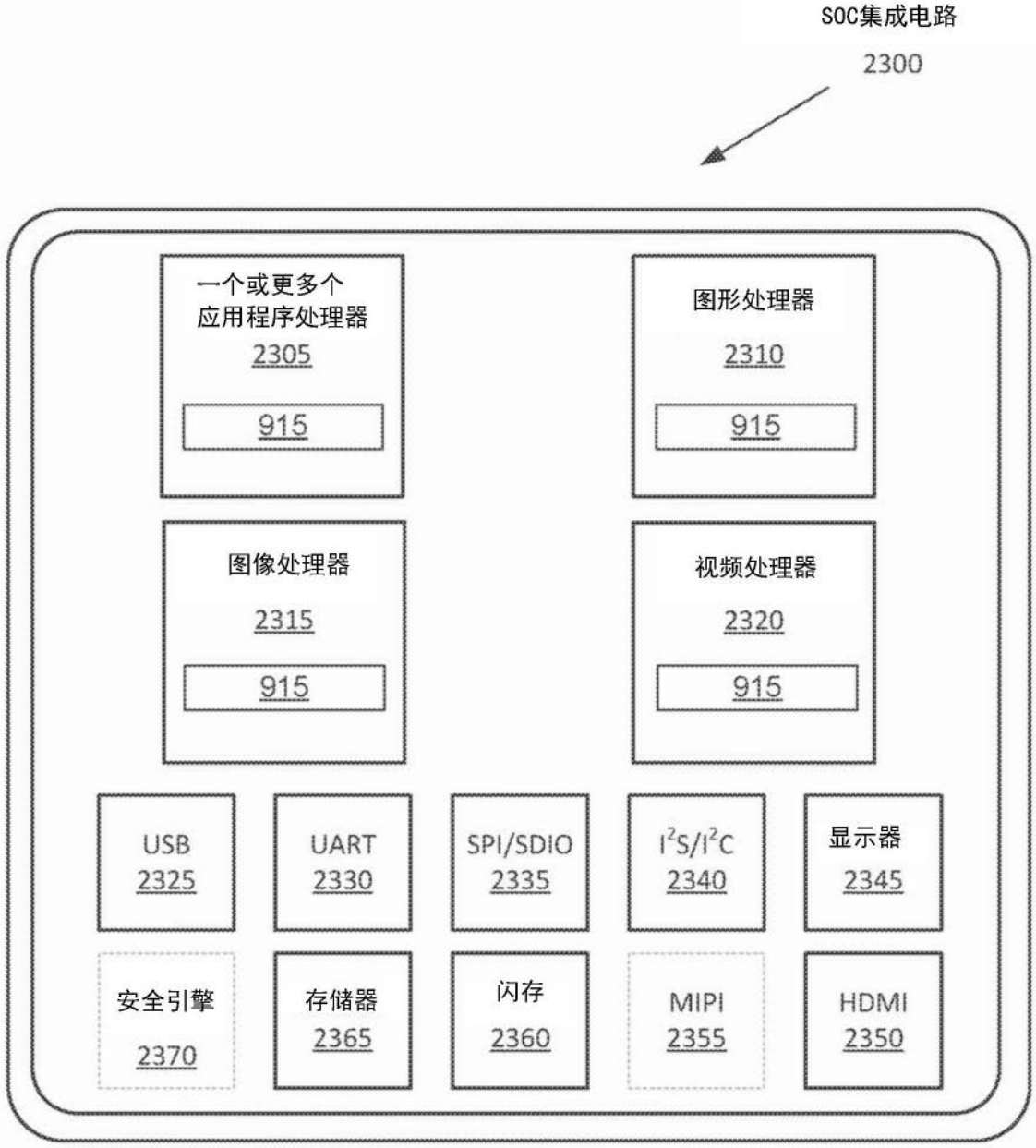


图18



图19A



图19B

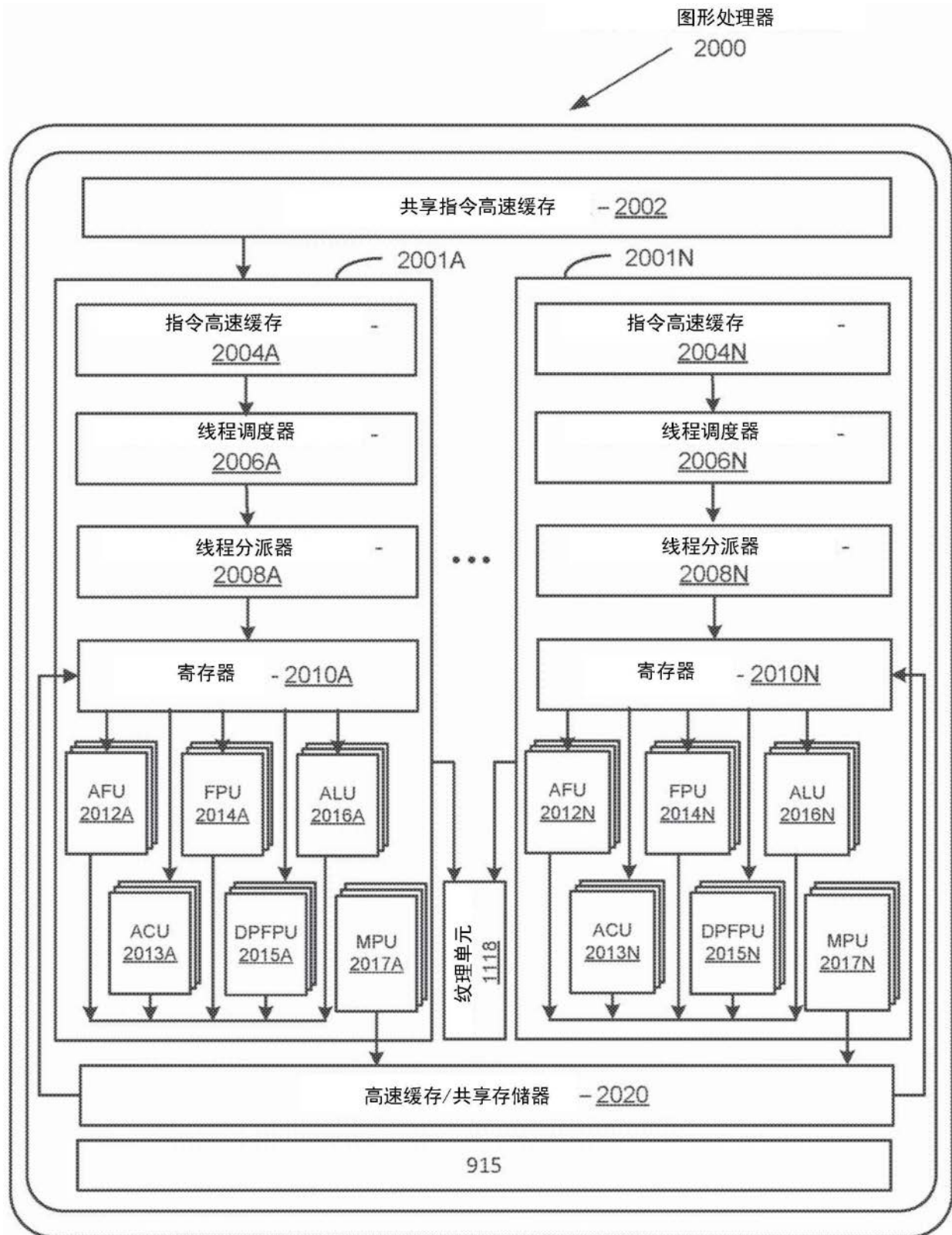


图20A

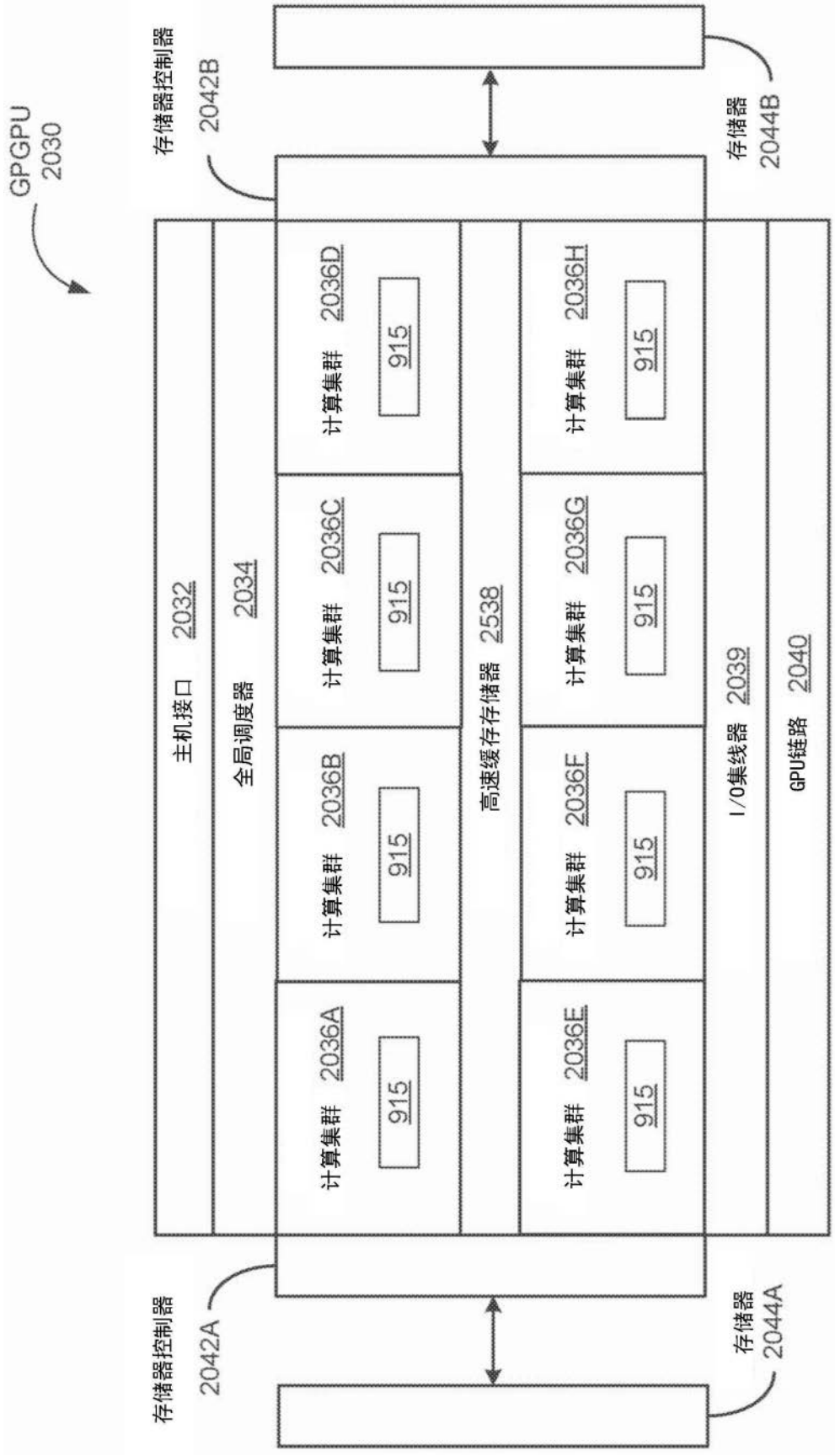


图20B

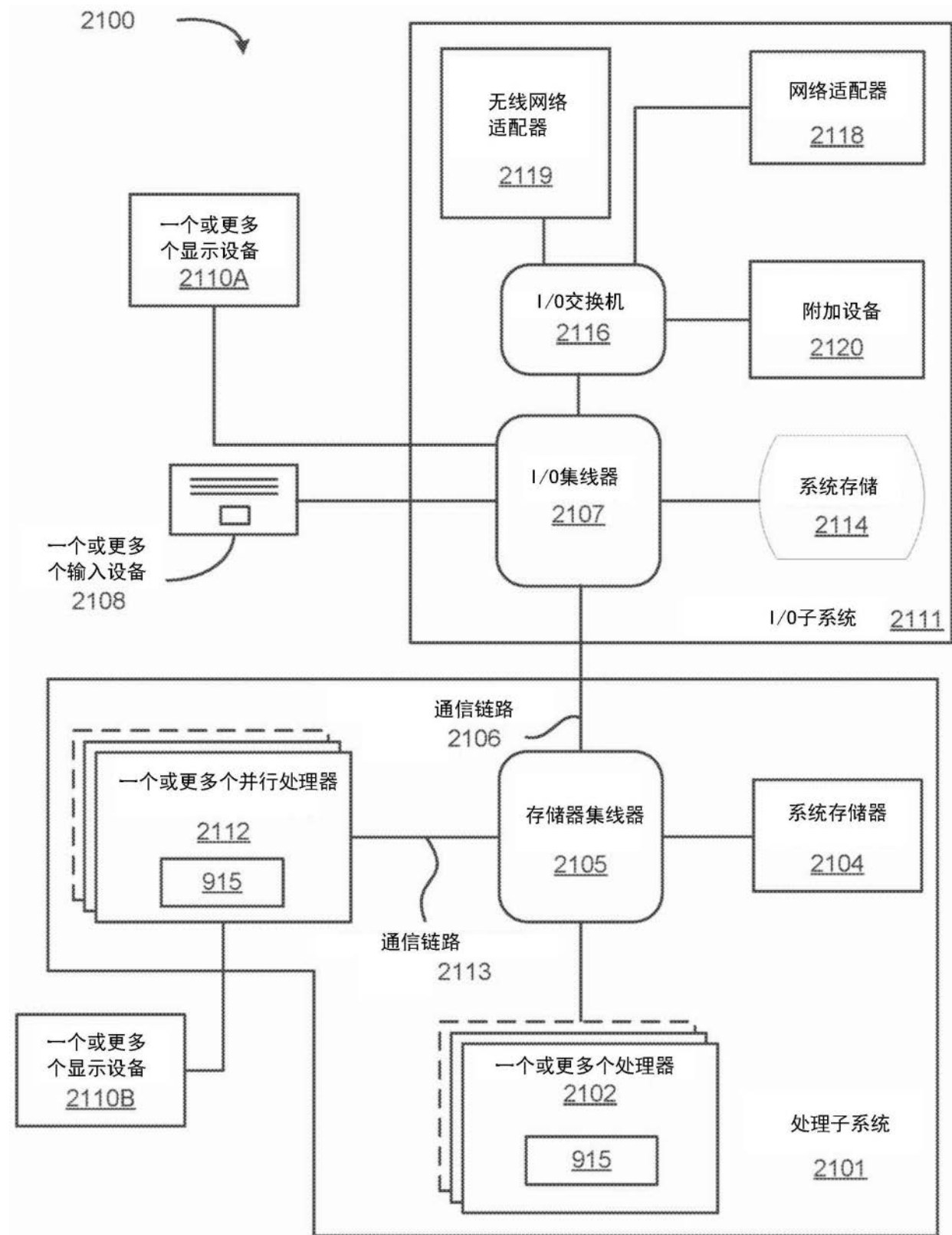


图21

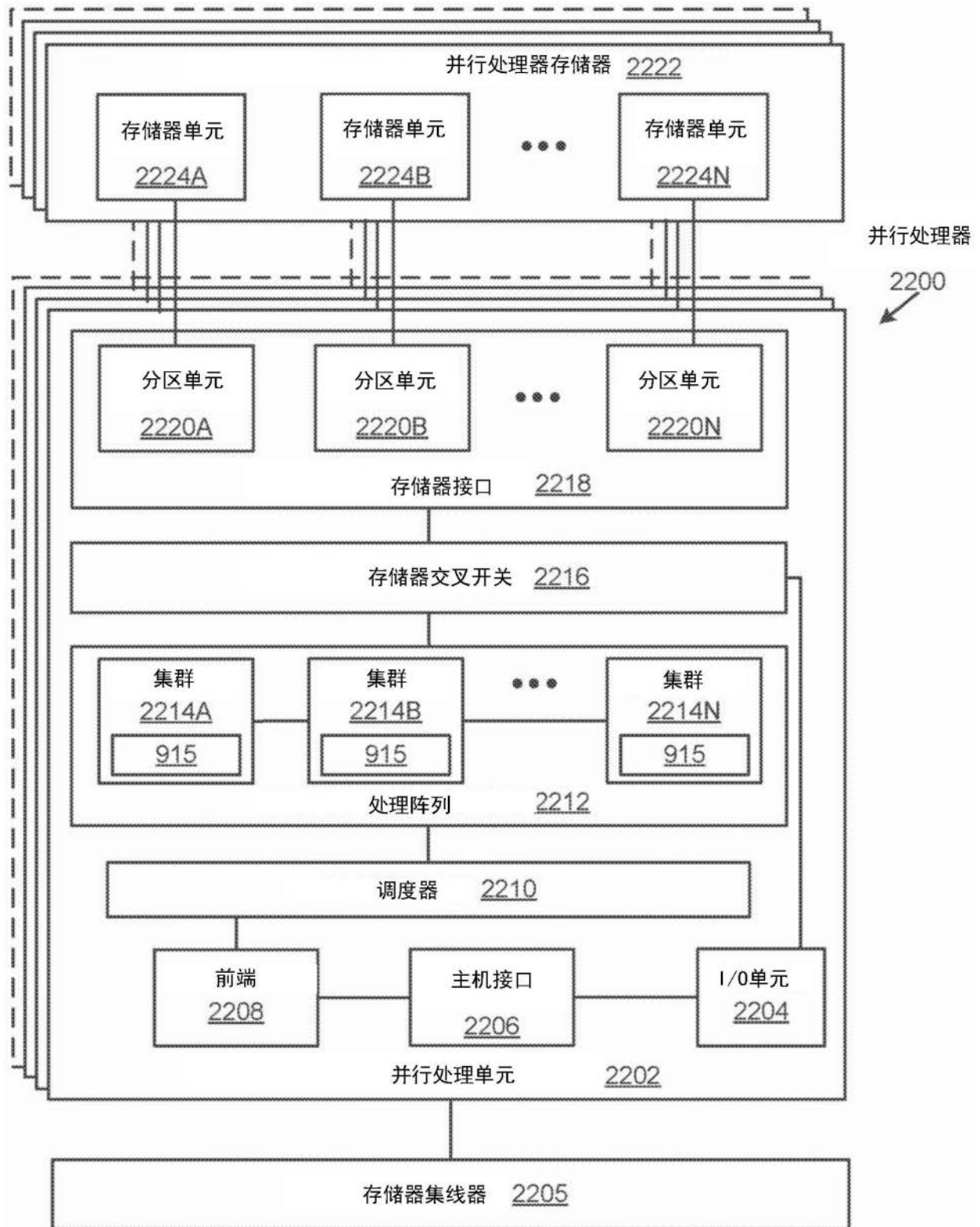


图22A

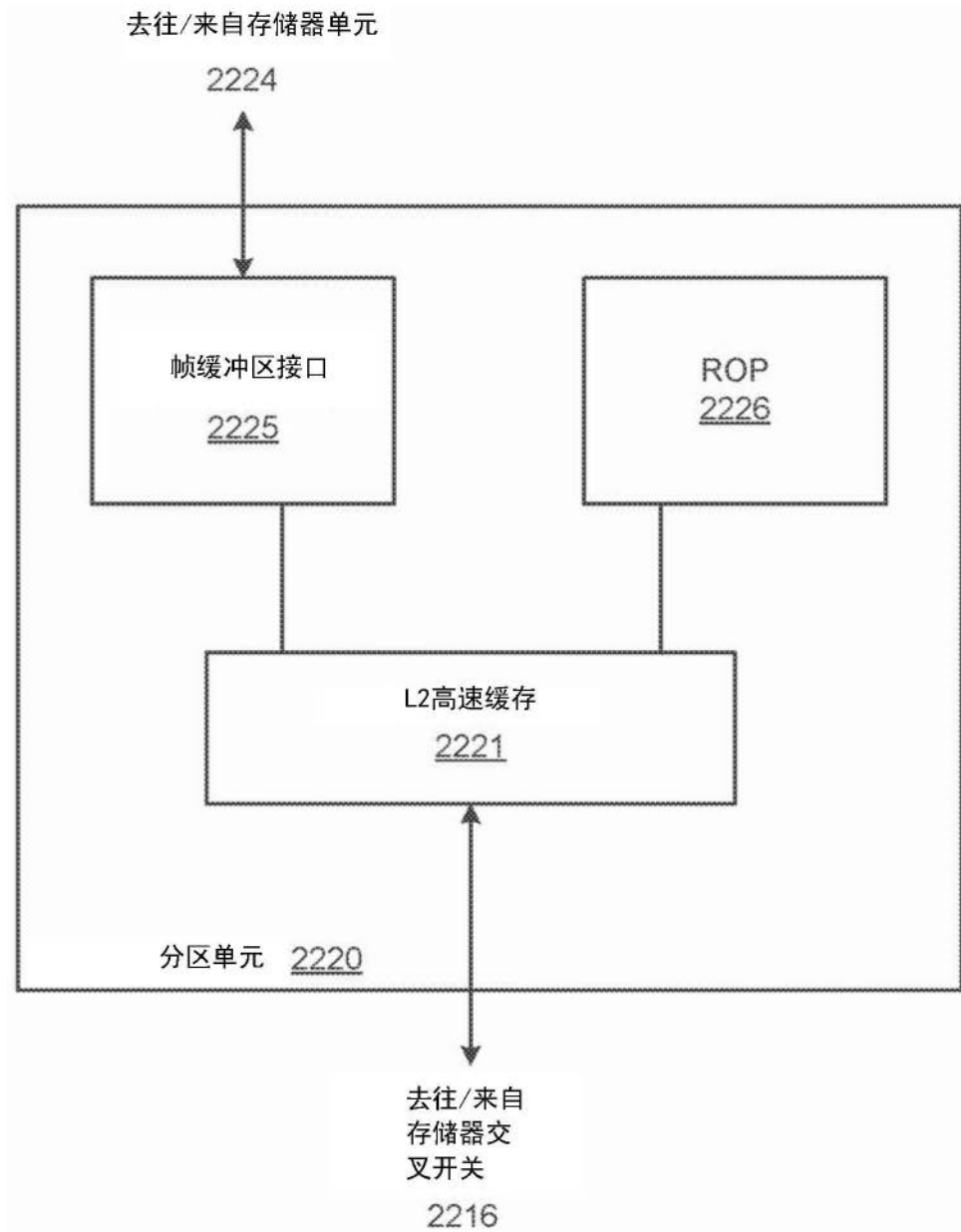


图22B

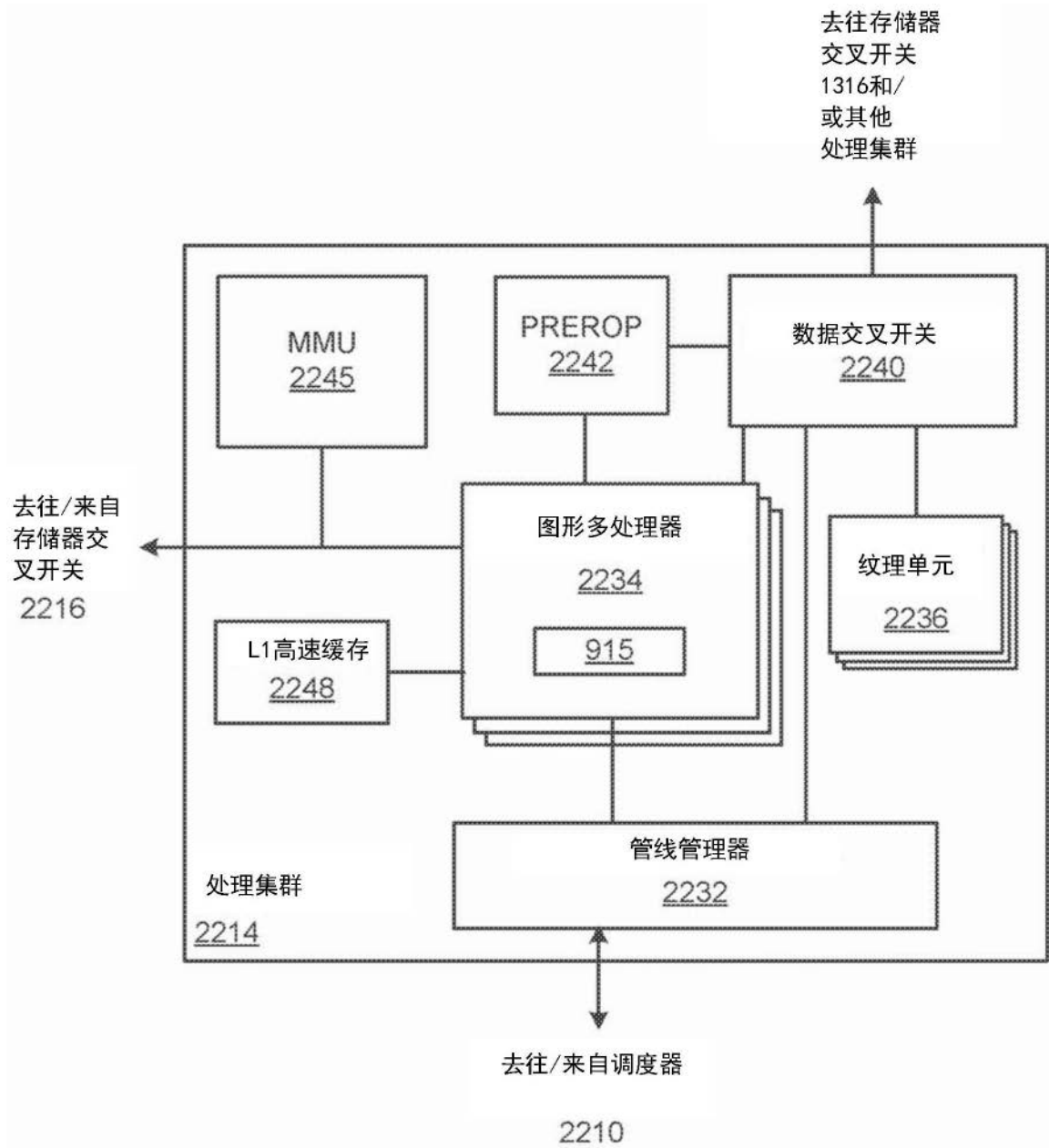


图22C

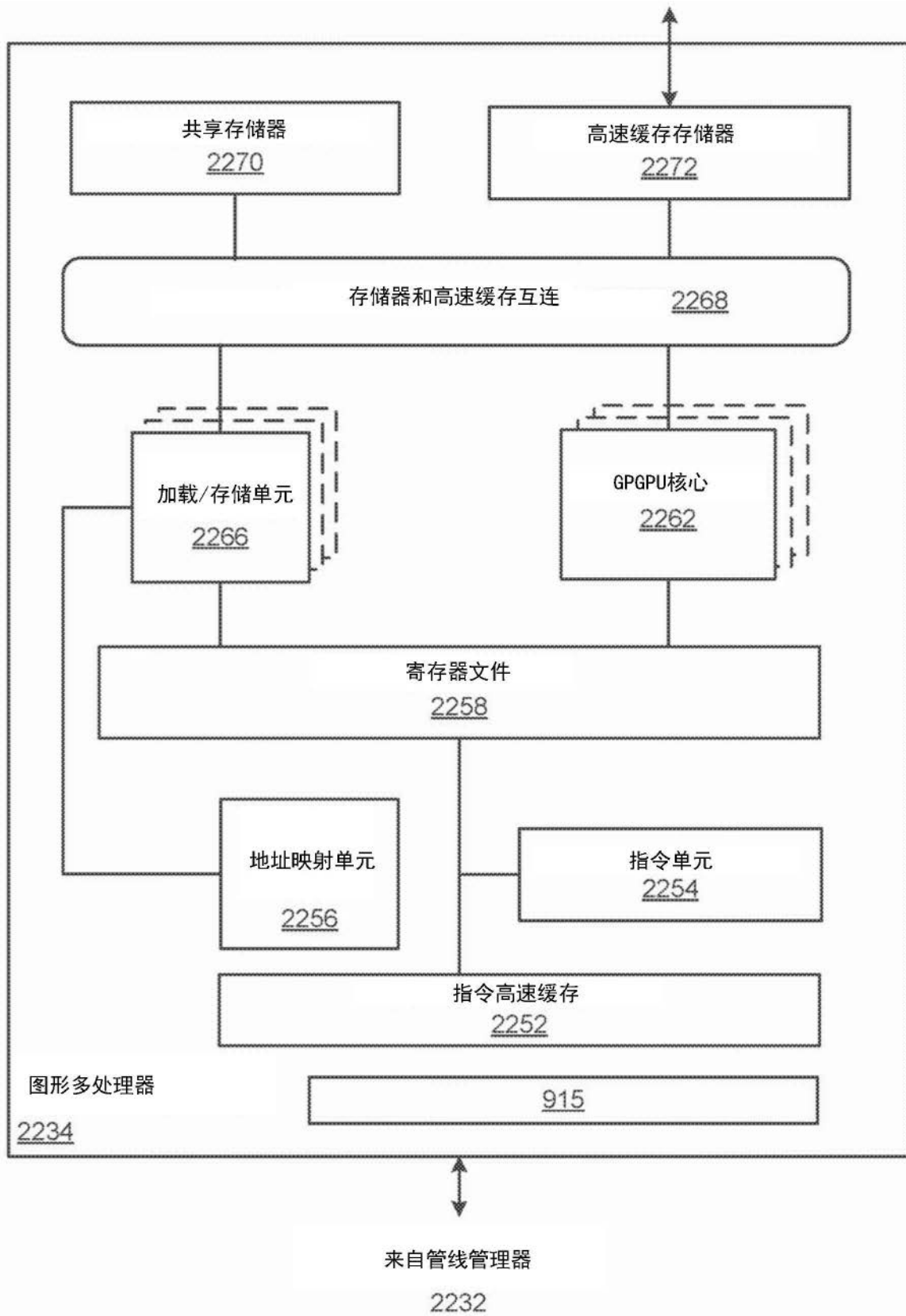


图22D

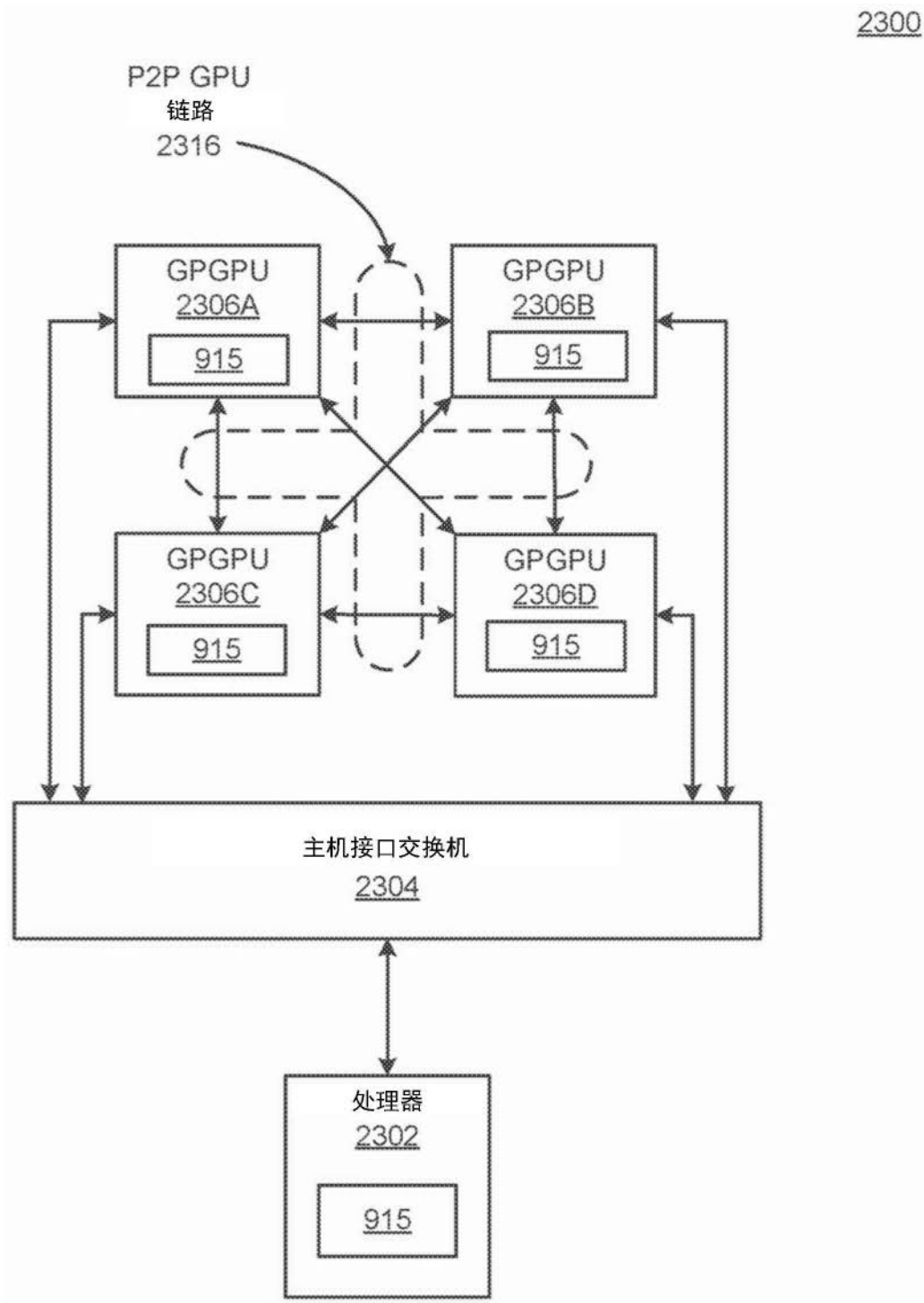


图23

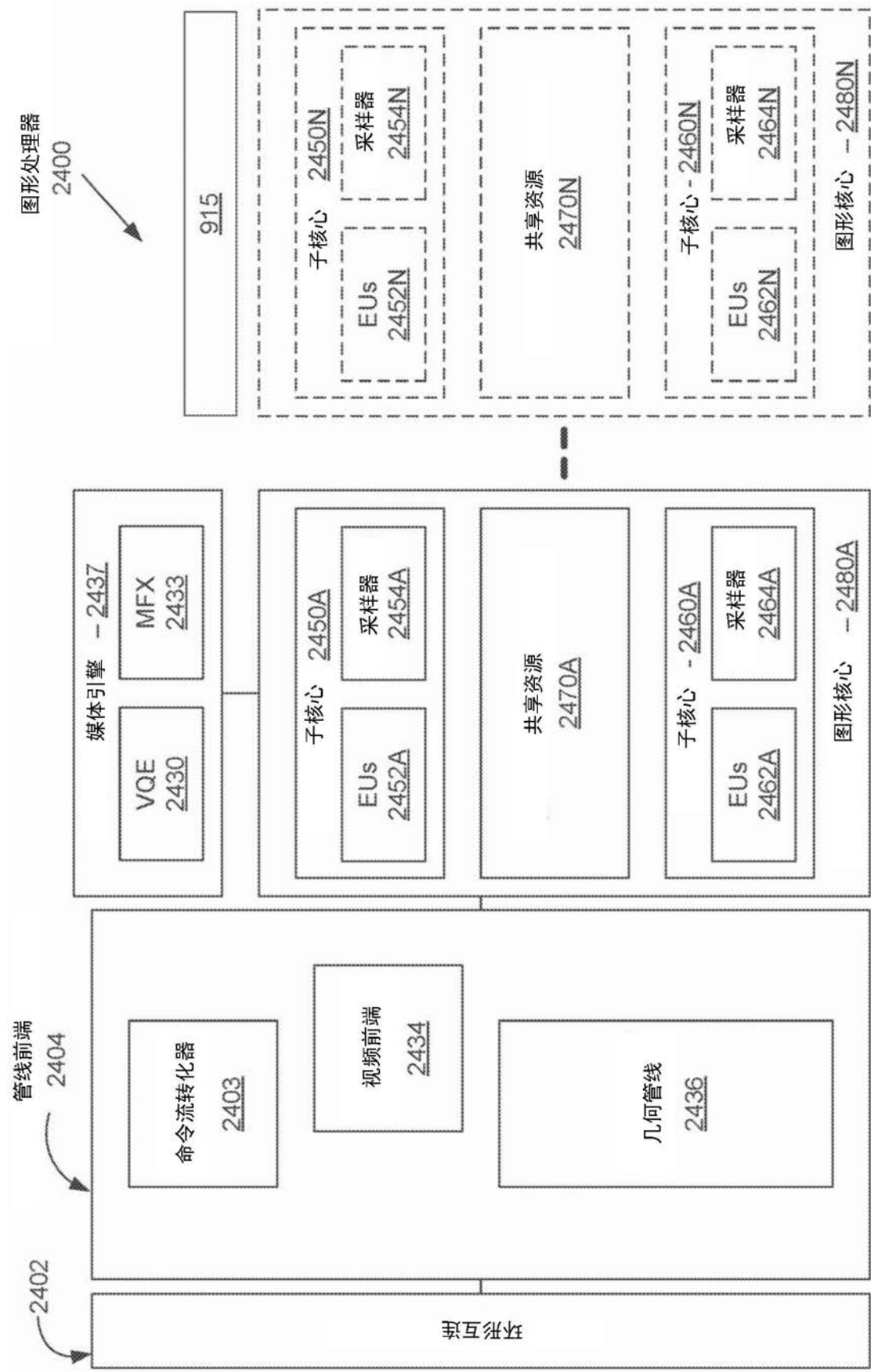


图24

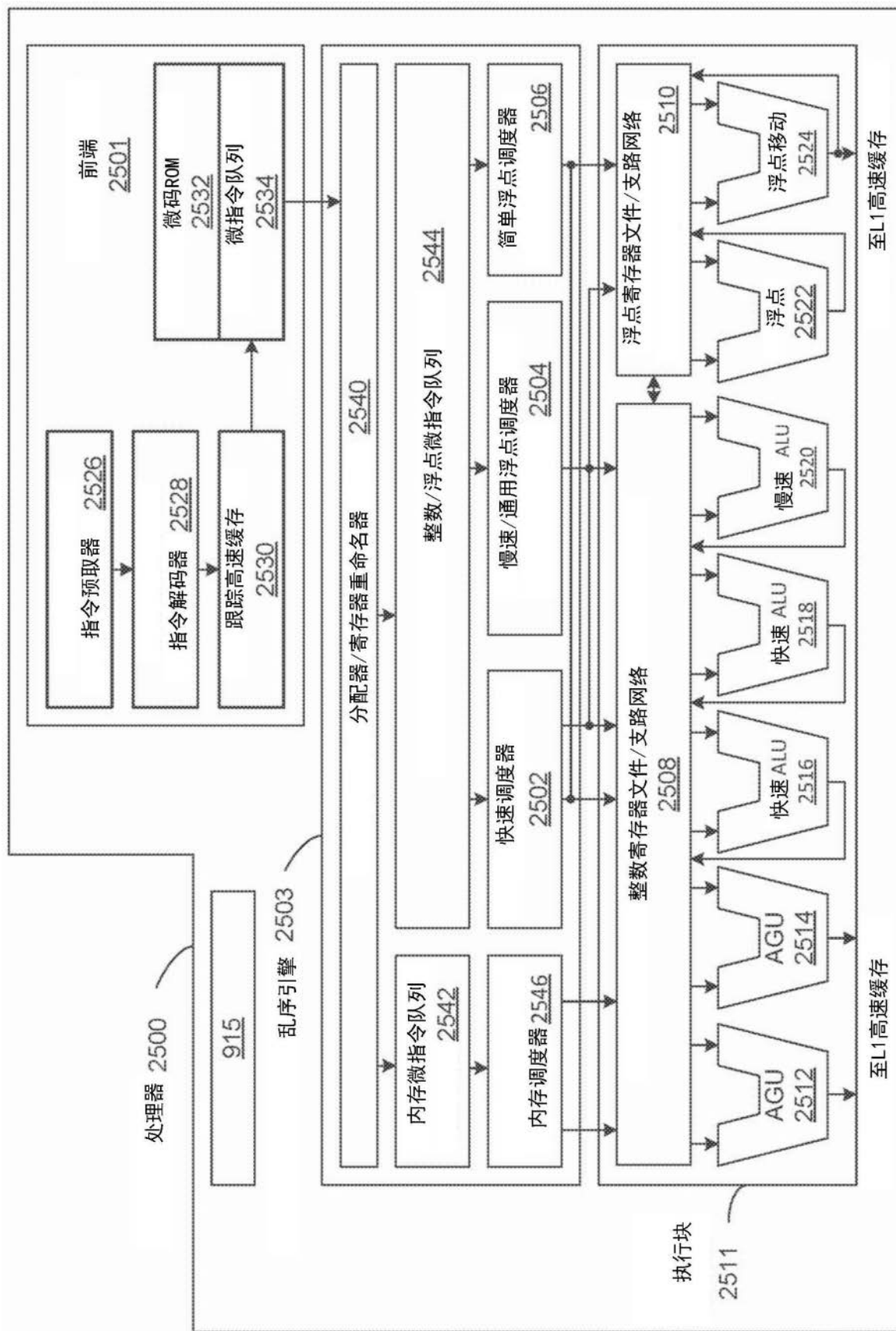


图25

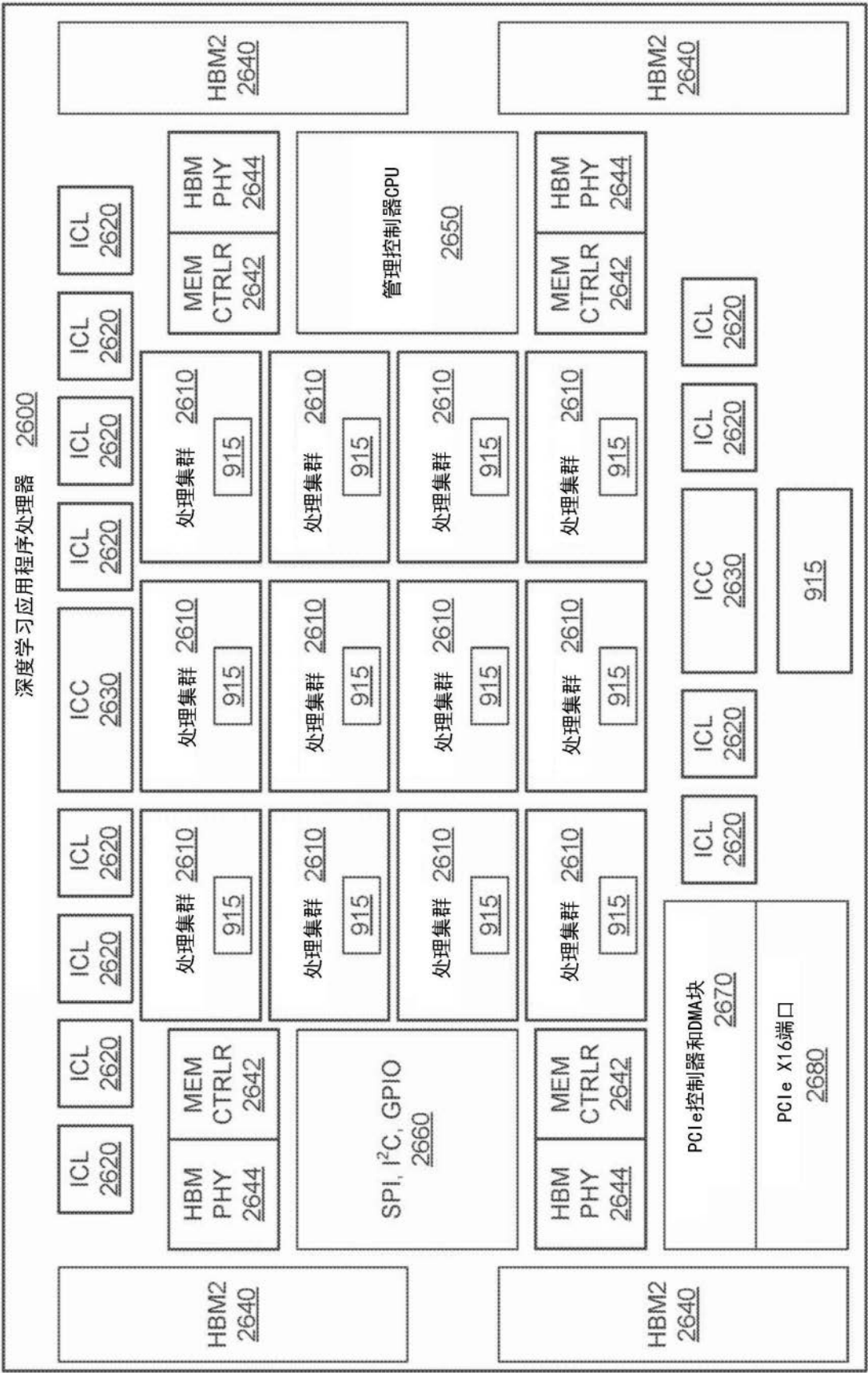


图26

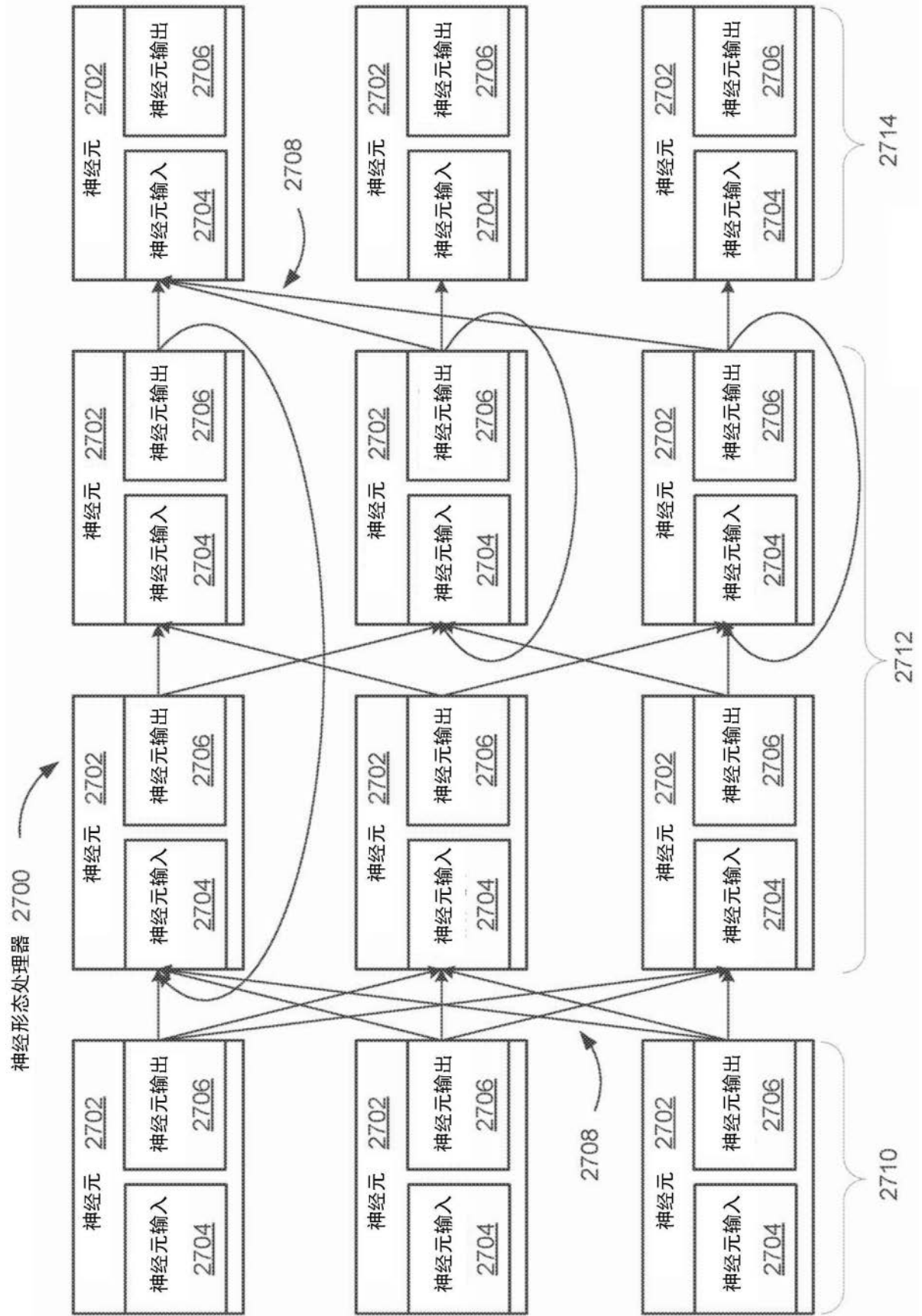


图27

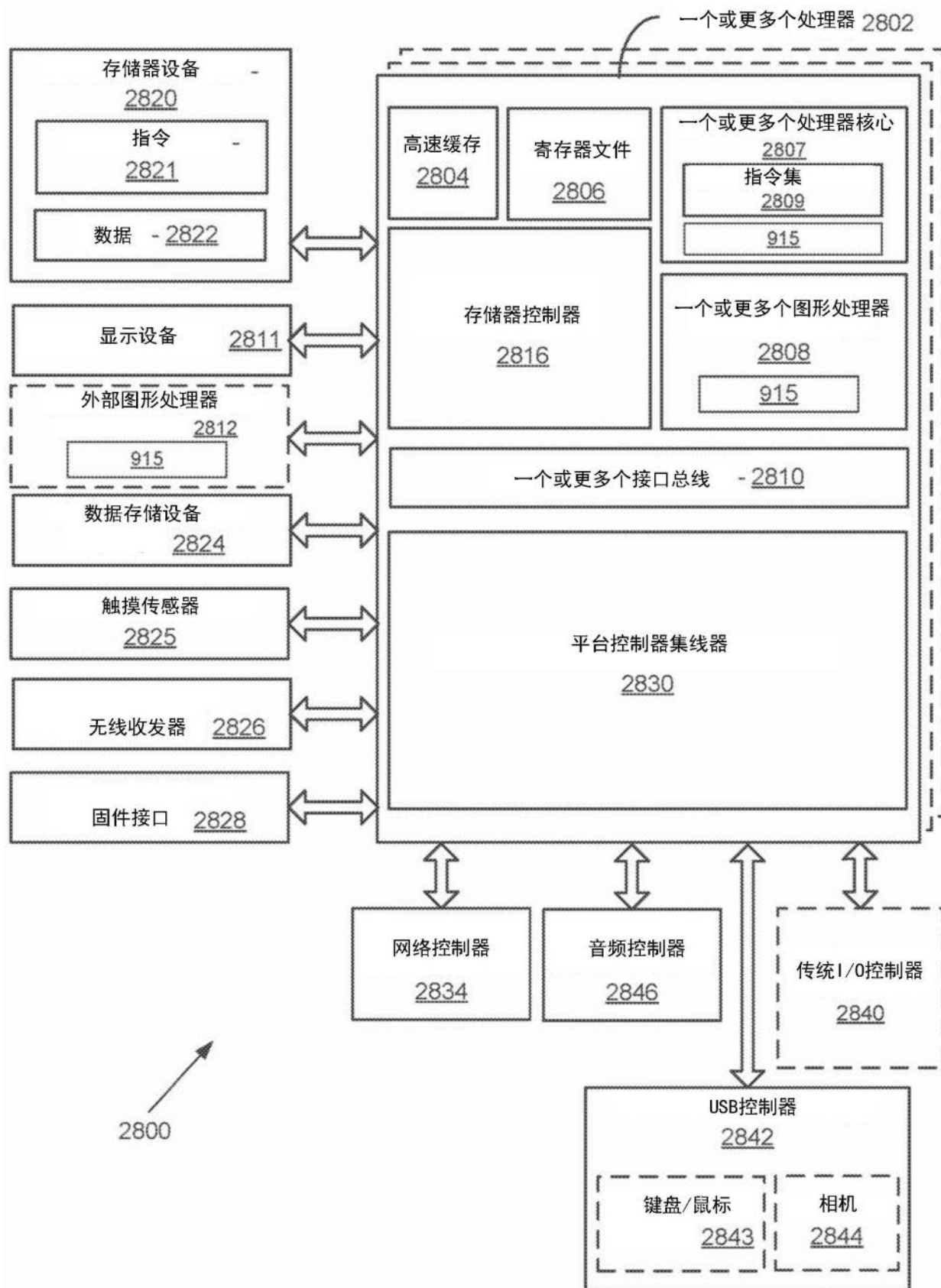


图28

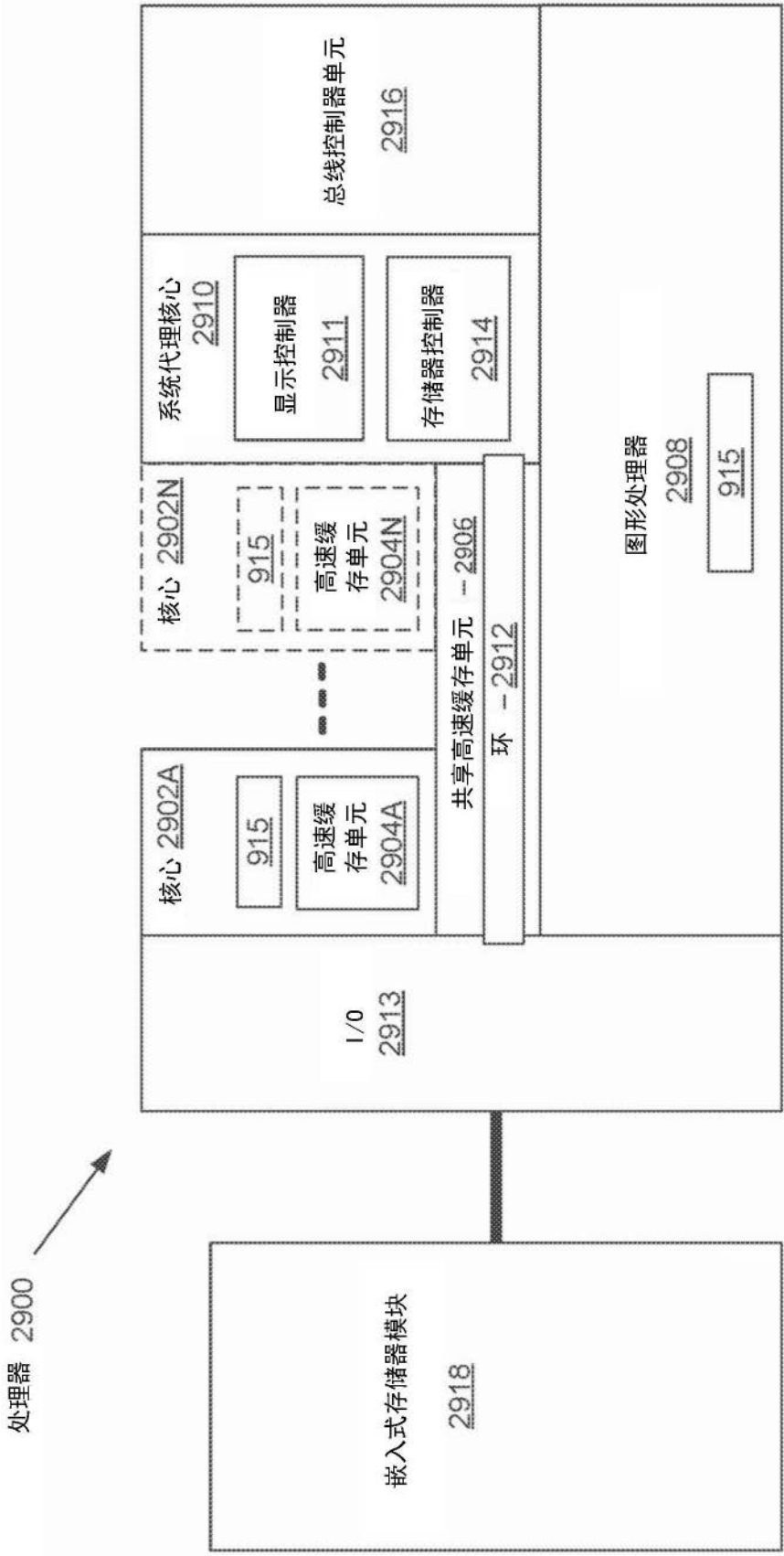


图29

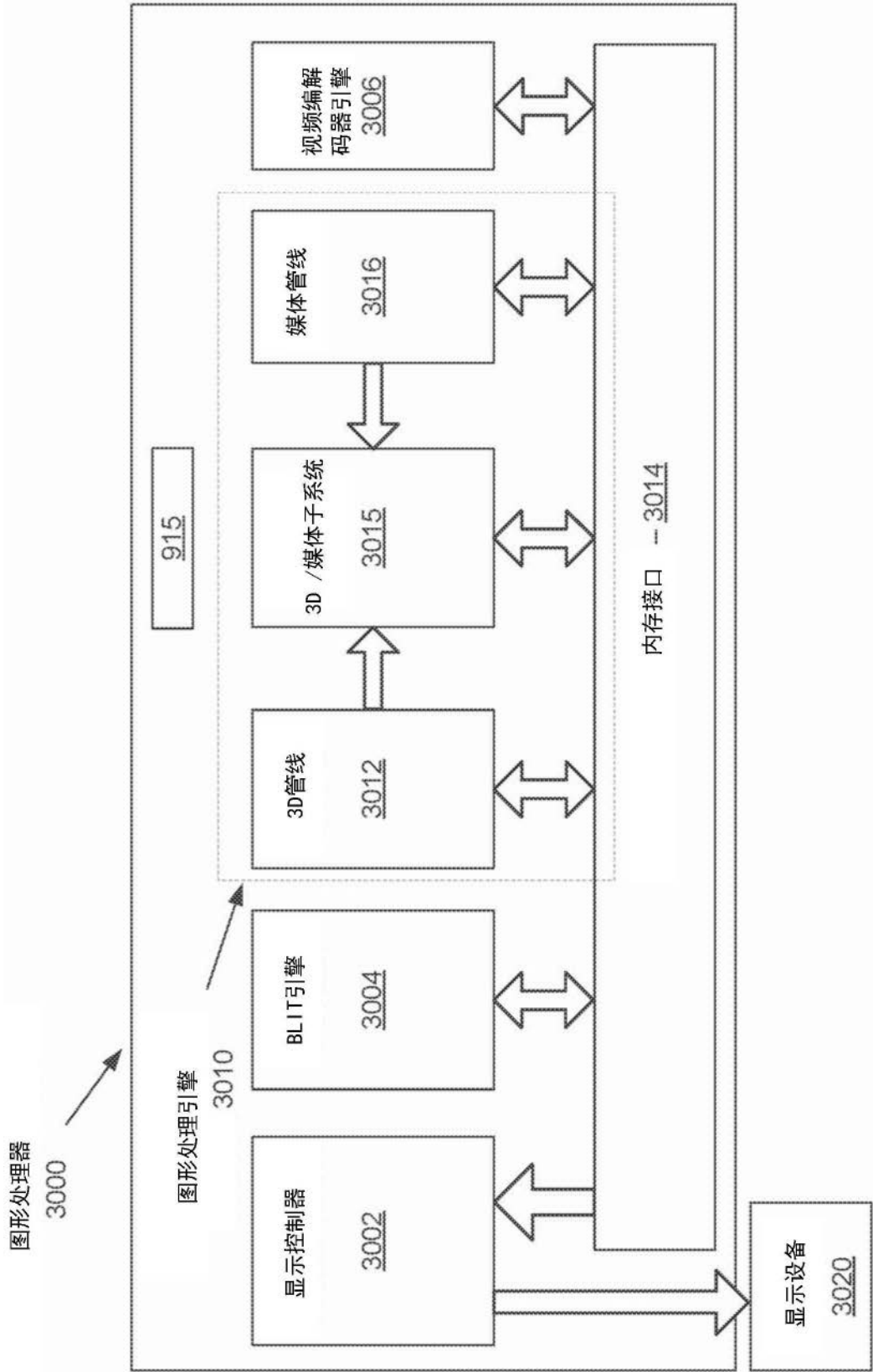


图30

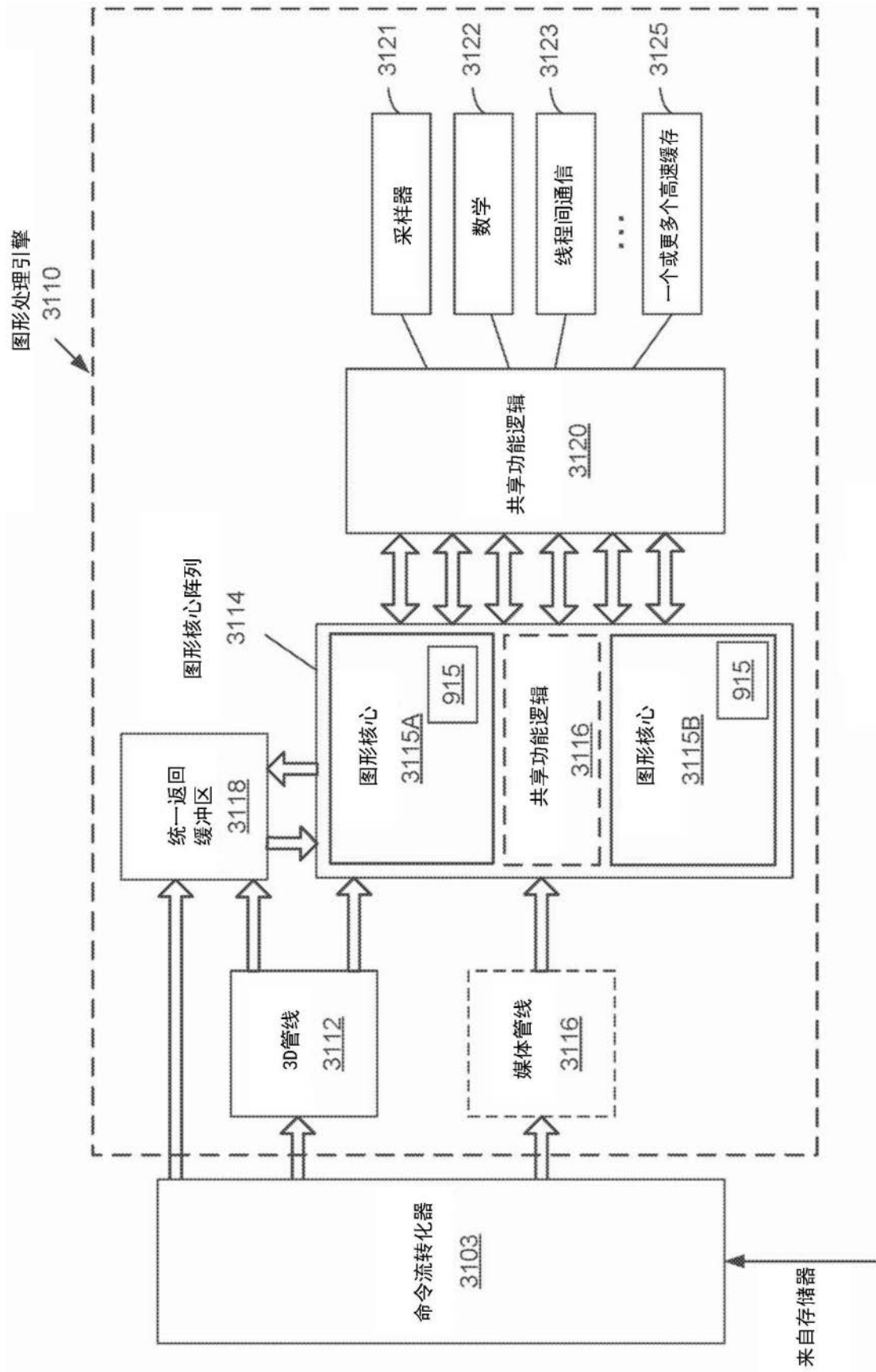


图31

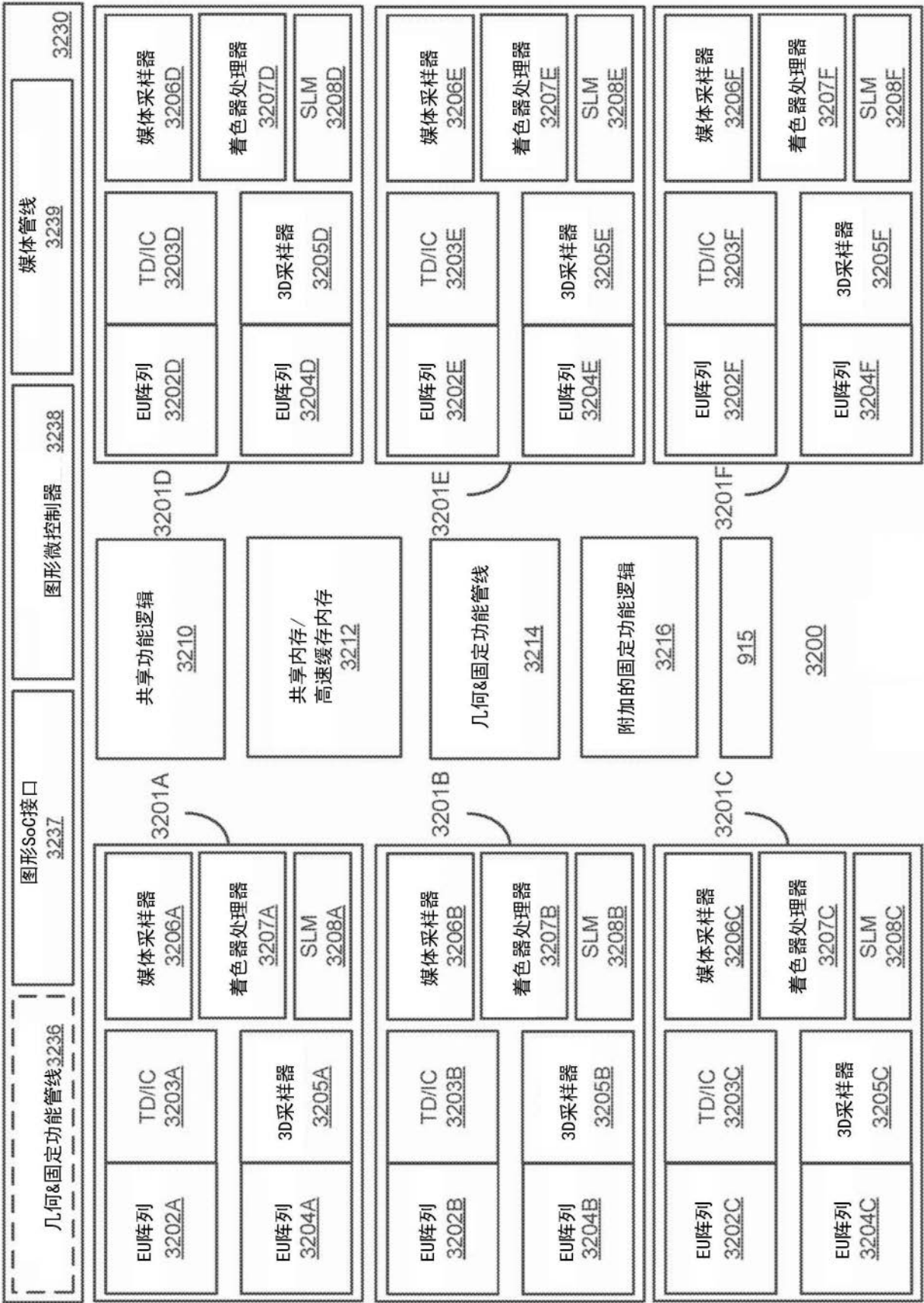


图32

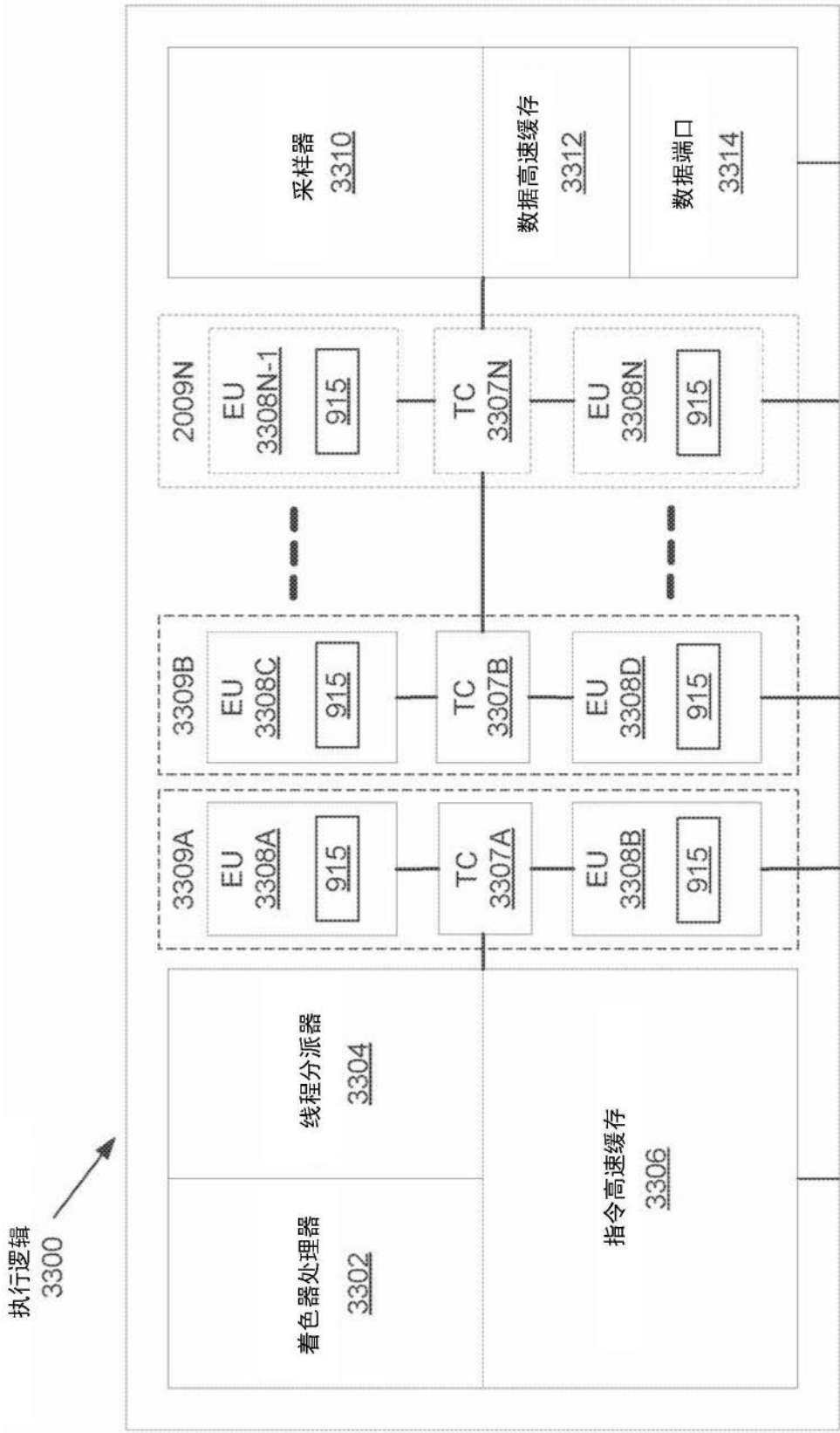


图33A

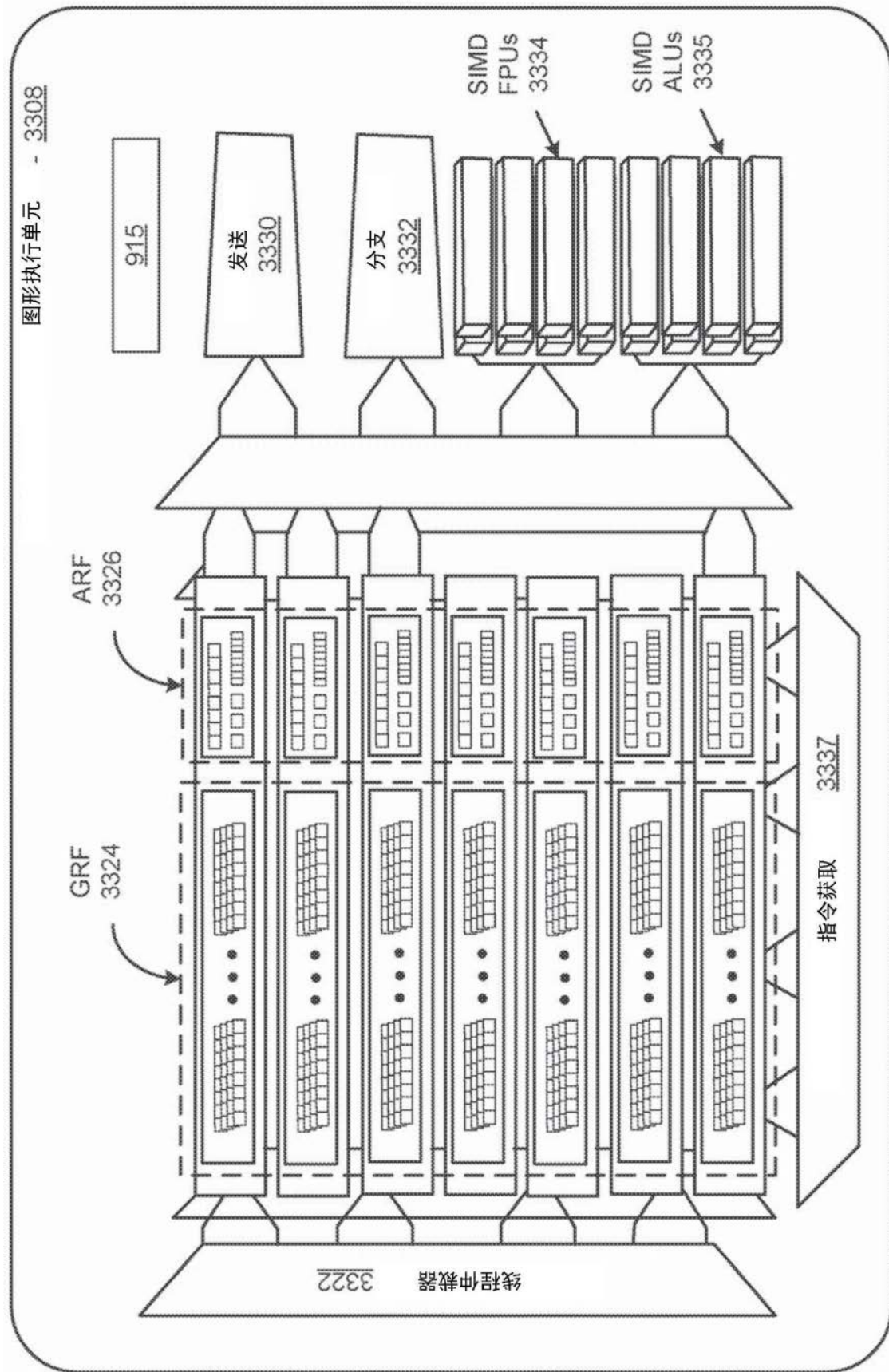


图33B

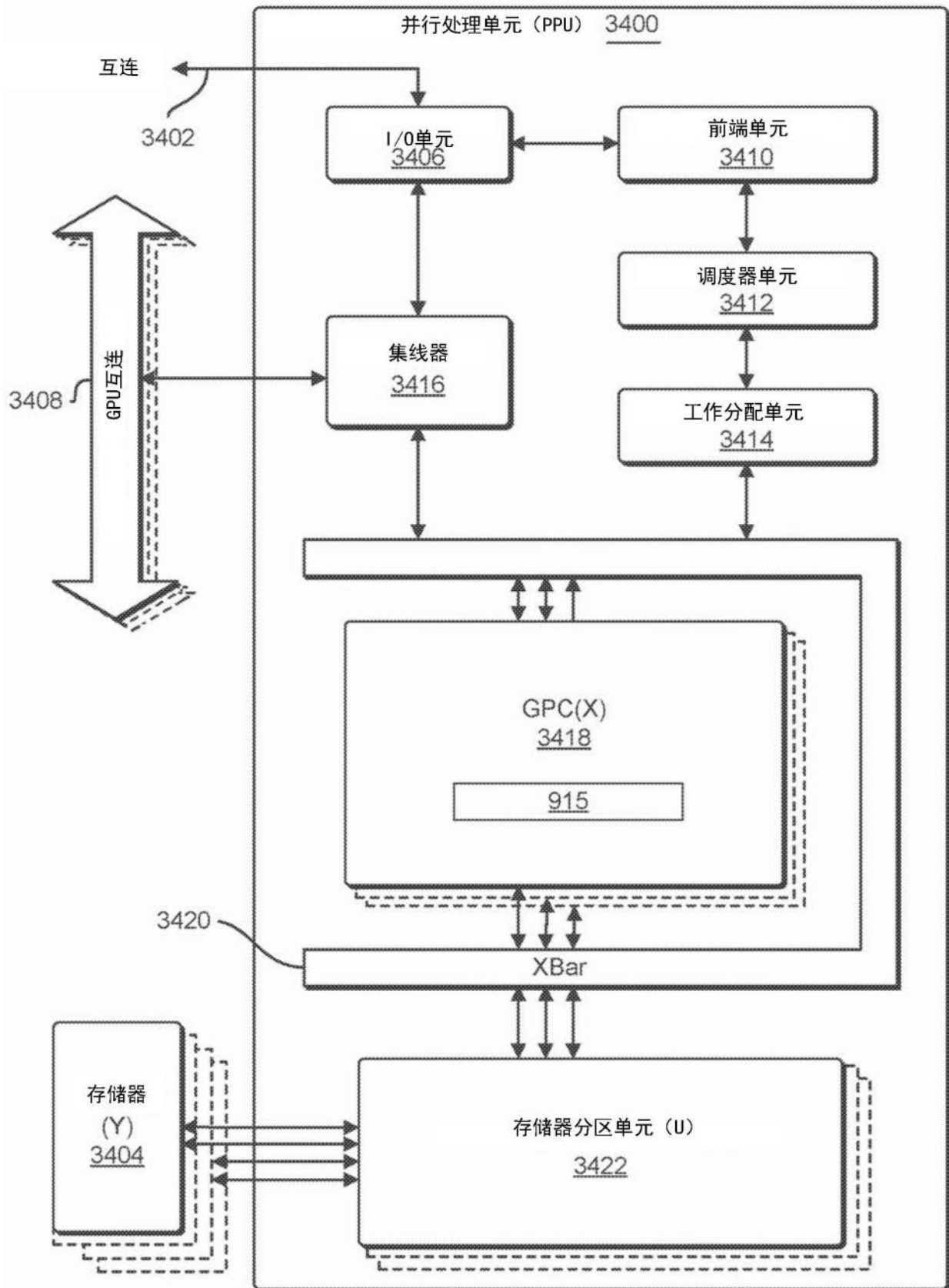


图34

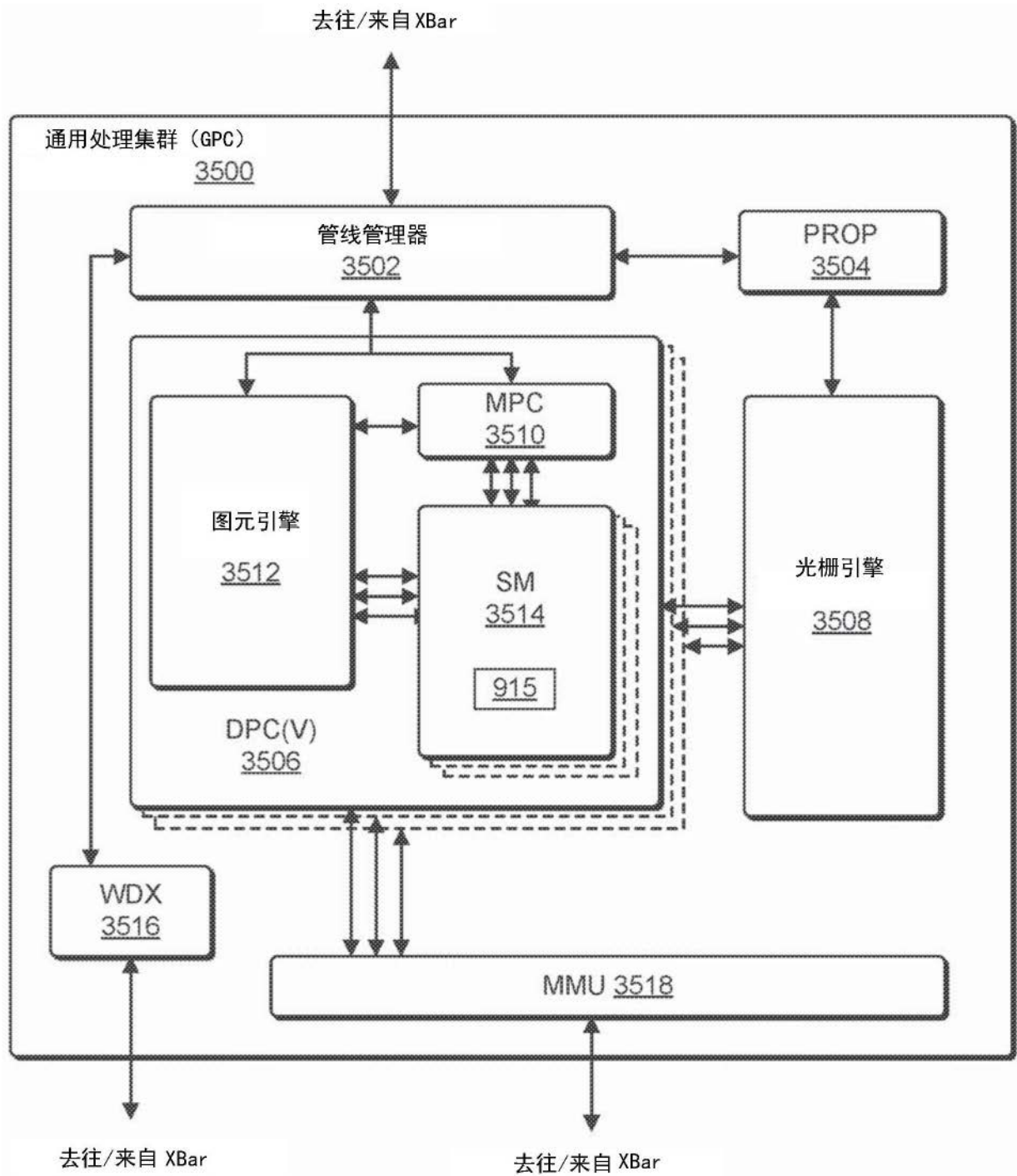


图35

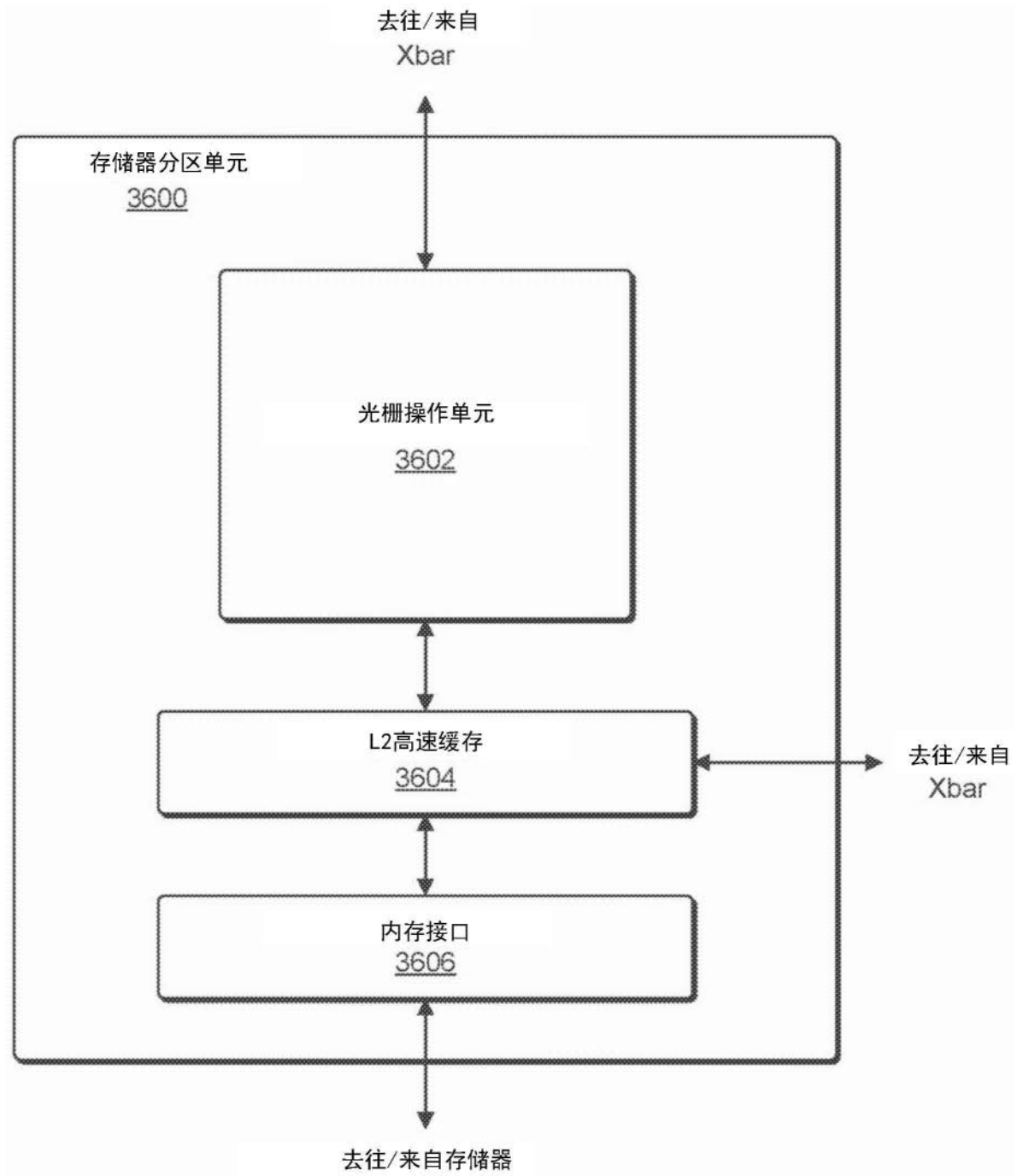


图36

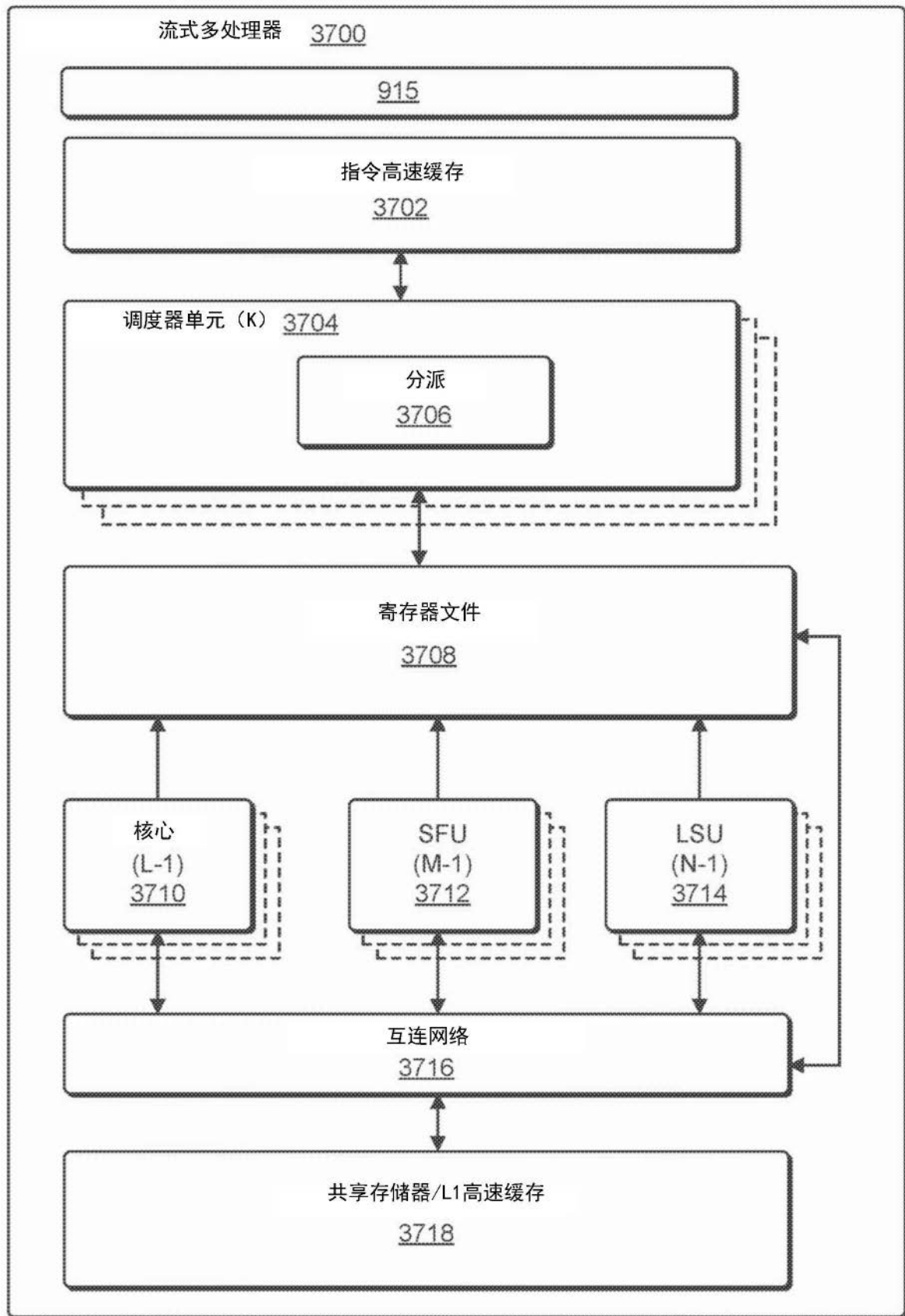


图37