



(12) 发明专利申请

(10) 申请公布号 CN 118503697 A

(43) 申请公布日 2024. 08. 16

(21) 申请号 202410570797.2

(51) Int.Cl.

(22) 申请日 2020.11.02

G06F 18/214 (2023.01)

(30) 优先权数据

G06N 3/098 (2023.01)

16/671,001 2019.10.31 US

G06N 3/0464 (2023.01)

G06N 3/084 (2023.01)

(62) 分案原申请数据

202011205589.0 2020.11.02

(71) 申请人 辉达公司

地址 美国加利福尼亚州

(72) 发明人 F·米莱塔里 M·鲍斯特

N·里克 李文琦 徐大光 A·冯

欧榕 程岩

(74) 专利代理机构 北京市磐华律师事务所

11336

专利代理人 高伟

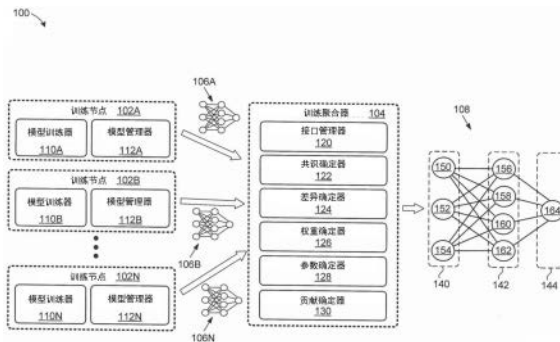
权利要求书1页 说明书94页 附图50页

(54) 发明名称

基于比较模型参数的准确度训练机器学习模型的处理器和系统

(57) 摘要

公开了基于比较模型参数的准确度训练机器学习模型的处理器和系统。协同训练一个或多个机器学习模型的装置、系统和技术。参数审查器可以配置为比较机器学习模型参数信息集,以便生成一个或多个机器学习模型,例如神经网络。



1. 一种处理器, 包括:

一个或多个电路, 用于从服务器接收神经网络权重信息, 其中所述神经网络权重信息至少部分地基于将所述处理器对一个或多个图像内的一个或多个对象的识别准确度与至少第二处理器对所述一个或多个对象的识别准确度进行比较。

2. 根据权利要求1所述的处理器, 其中所述一个或多个电路还至少部分地基于所述神经网络权重信息来更新神经网络的一个或多个参数。

3. 根据权利要求1所述的处理器, 其中所述神经网络权重信息还至少部分地基于所述至少第二处理器对所述一个或多个对象的识别准确度的评分信息, 所述评分信息基于所述比较而生成。

4. 根据权利要求1所述的处理器, 其中从所述服务器接收的所述神经网络权重信息还至少部分地基于所述处理器向所述服务器提供的第二神经网络权重信息。

5. 根据权利要求1所述的处理器, 其中所述处理器属于第一边缘设备, 所述第二处理器属于第二边缘设备。

6. 根据权利要求1所述的处理器, 其中所述至少第二处理器对所述一个或多个图像内的所述一个或多个对象的识别准确度属于至少部分地由所述至少第二处理器训练的神经网络。

7. 一种处理器, 包括:

一个或多个电路, 用于至少部分地基于将第一处理器对一个或多个对象的识别准确度与第二处理器对一个或多个对象的识别准确度进行比较, 来帮助训练神经网络以识别一个或多个图像内的一个或多个对象。

8. 根据权利要求7所述的处理器, 其中所述一个或多个电路还至少部分地基于, 基于所述比较针对所述第一处理器对所述一个或多个对象的识别准确度或所述第二处理器对所述一个或多个对象的识别准确度中的一个或多个生成评分信息, 来帮助训练所述神经网络以识别所述一个或多个图像内的所述一个或多个对象。

9. 一种服务器, 包括:

一个或多个处理器, 用于至少部分地基于第一边缘设备对一个或多个图像内的一个或多个对象的识别准确度与第二边缘设备对所述一个或多个对象的识别准确度的比较来生成一个或多个神经网络。

10. 根据权利要求9所述的服务器, 其中所述一个或多个处理器还将所述一个或多个神经网络的神经网络权重信息发送到所述第一边缘设备或所述第二边缘设备中的一个或多个, 所述神经网络权重信息基于所述比较而更新。

基于比较模型参数的准确度训练机器学习模型的处理器和系统

[0001] 本申请是2020年11月02日提出的第202011205589.0号中国专利申请的分案申请。

背景技术

[0002] 在联合学习中,可以汇总由不同参与实体或节点训练的机器学习模型的参数,以共同地训练机器学习模型。数据预处理中的错误,漏洞,错误的超参数选择,故意的对抗行为或与节点关联的其他特征可能会对共同训练的模型的质量产生负面影响。为了解决这个问题,服务器并不是简单地对来自与相同参数相对应的节点的值取平均,而是可以使用范数裁剪平均进行聚合,以将参数值限制在阈值上。在另一种方法中,每个节点使用随机梯度下降在本地计算更新,然后服务器将这些更新组合在一起。当节点表现出可能对联合机器学习模型的训练产生不利影响的特定类型的特征时,这些方法可能是合适的。

附图说明

- [0003] 图1A示出了至少一个实施例中的协作训练环境;
- [0004] 图1B示出了至少一个实施例中的模型图;
- [0005] 图2A示出了至少一个实施例中的推理和/或训练逻辑;
- [0006] 图2B示出了至少一个实施例中的推理和/或训练逻辑;
- [0007] 图3示出了至少一个实施例中的机器学习模型的训练和部署;
- [0008] 图4示出了至少一个实施例中的用于协作训练的流程图;
- [0009] 图5示出了至少一个实施例中的用于协作训练的网络环境;
- [0010] 图6示出了至少一个实施例中的流程图;
- [0011] 图7示出了至少一个实施例中的流程图;
- [0012] 图8示出了至少一个实施例中的流程图;
- [0013] 图9示出了至少一个实施例中的流程图;
- [0014] 图10示出了至少一个实施例中的流程图;
- [0015] 图11A示出了至少一个实施例中的示例参数审查器;
- [0016] 图11B示出了至少一个实施例中的示例审查聚合器;
- [0017] 图12示出了至少一个实施例中的流程图;
- [0018] 图13示出了至少一个实施例中的流程图;
- [0019] 图14示出了至少一个实施例中的示例数据中心系统;
- [0020] 图15A示出了至少一个实施例中的自主车辆的示例;
- [0021] 图15B示出了至少一个实施例中的图15A的自主车辆的相机位置和视野的示例;
- [0022] 图15C是示出在至少一个实施例中图15A的自主车辆的示例系统架构的框图;
- [0023] 图15D是示出在至少一个实施例中在基于云的服务器与图15A的自主车辆之间进行通信的系统的框图;
- [0024] 图16是示出至少一个实施例中的计算机系统的框图;

- [0025] 图17是示出至少一个实施例中的计算机系统的框图；
- [0026] 图18示出了至少一个实施例中的计算机系统；
- [0027] 图19示出了根据至少一个实施例的计算机系统；
- [0028] 图20A示出了至少一个实施例中的计算机系统；
- [0029] 图20B示出了至少一个实施例中的计算机系统；
- [0030] 图20C示出了至少一个实施例中的计算机系统；
- [0031] 图20D示出了至少一个实施例中的计算机系统；
- [0032] 图20E和图20F示出了至少一个实施例中的共享编程模型；
- [0033] 图21示出了至少一个实施例中的示例性集成电路和相关的图形处理器；
- [0034] 图22A-22B示出了至少一个实施例中的示例性集成电路和相关的图形处理器；
- [0035] 图23A-23B示出了至少一个实施例中的另外的示例性图形处理器逻辑；
- [0036] 图24示出了至少一个实施例中的计算机系统；
- [0037] 图25A示出了至少一个实施例中的并行处理器；
- [0038] 图25B示出了至少一个实施例中的分区单元；
- [0039] 图25C示出了至少一个实施例中的处理集群；
- [0040] 图25D示出了至少一个实施例中的图形多处理器；
- [0041] 图26示出了至少一个实施例中的多图形处理单元 (GPU) 系统；
- [0042] 图27示出了至少一个实施例中的图形处理器；
- [0043] 图28是示出至少一个实施例中的用于处理器的处理器微架构的框图；
- [0044] 图29示出了至少一个实施例中的深度学习应用处理器；
- [0045] 图30是示出至少一个实施例中的示例神经形态处理器的框图；
- [0046] 图31和图32示出了根据一个或多个实施例的图形处理器的至少部分；
- [0047] 图33是在至少一个实施例中的图形处理器核心的至少部分的框图；
- [0048] 图34A-34B示出了在至少一个实施例中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；
- [0049] 图35示出了至少一个实施例中的并行处理单元 (“PPU”)；
- [0050] 图36示出了至少一个实施例中的通用处理集群 (“GPC”)；
- [0051] 图37示出了至少一个实施例中的并行处理单元 (“PPU”) 的存储器分区单元；和
- [0052] 图38示出了至少一个实施例中的流式多处理器。

具体实施方式

[0053] 图1A示出了至少一个实施例中的协作训练环境100。在至少一个实施例中，协作训练环境100可以用于执行推理和/或训练操作。

[0054] 在至少一个实施例中，协作训练环境100可以包括但不限于训练节点102A和102B至102N(也称为“训练节点102”)和训练聚合器104。在至少一个实施例中，训练节点102A和102B至102N和训练聚合器104可以包括硬件、软件或其某种组合。尽管示出了训练节点102A和102B至102N，但是在至少一个实施例中，协作训练环境100可以包括两个或多个训练节点。

[0055] 在至少一个实施例中，训练节点102A包括模型训练器110A和模型管理器112A，训

训练节点102B包括模型训练器110B和模型训练器112B,并且训练节点102N包括模型训练器110N和模型训练器112N。在协作训练环境100包括一个或更多个附加训练节点的情况下,每个训练节点可以包括模型管理器和/或模型训练器,该模型管理器和/或模型训练器可以类似于或不同于任何训练节点102A和102B至102N的模型管理器和/或模型训练器。

[0056] 在至少一个实施例中,训练聚合器104包括接口管理器120、共识确定器122、差异(discrepancy)确定器124、权重确定器126、参数确定器128和贡献确定器130中的一个或更多个。

[0057] 在至少一个实施例中,训练节点102和训练聚合器104可以用于协作地训练机器学习模型(MLM) 108。特别地,一个或更多个MLM 108可以包括一个或更多个经学习的参数(例如权重和/或偏差(bias))。可以通过利用至少部分地由训练节点102的模型训练器110A、模型训练器110B和模型训练器110N(也称为“模型训练器110”)执行的训练来学习机器学习模型108和/或其部分中的一个或更多个的一个或更多个参数的值。

[0058] 在至少一个实施例中,训练节点102A可以使用模型训练器110A和模型管理器112A来训练一个或更多个机器学习模型106A,训练节点102B可以使用模型训练器110B和模型管理器112B来训练一个或更多个机器学习模型106B,训练节点102N可以使用模型训练器110N和模型管理器112N来训练一个或更多个机器学习模型106N。通过训练一个或更多个机器学习模型106A、106B和106C(也称为机器学习模型106),训练节点102可以学习机器学习模型106的一个或更多个参数的值,这些值又可以由训练聚合器104使用以确定一个或更多个机器学习模型108的一个或更多个相应参数的一个或更多个值。为此,训练聚合器104可以使用接口管理器120,共识确定器122,差异确定器124,权重确定器126,参数确定器128和贡献确定器130中的一个或更多个。

[0059] 在至少一个实施例中,一个或更多个机器学习模型108包括使用线性回归,逻辑回归,决策树,支持向量机(SVM),朴素贝叶斯,k近邻(Knn),K均值聚类,随机森林,降维算法,梯度提升算法,神经网络(例如,自动编码器,卷积,递归,感知器,长期/短期记忆(LSTM),Hopfield,Boltzmann,深度信念,反卷积,生成式对抗,液态状态机等)的一个或更多个机器学习模型和/或其他类型的机器学习模型。

[0060] 在图1A中,示出了包括神经网络的一个或更多个机器学习模型108和机器学习模型106的示例,其可以包括一个或更多个层140,一个或更多个层142和一个或更多个层144。在至少一个实施例中,层140包括一个或更多个输入层,层142包括一个或更多个隐藏层,层144包括一个或更多个输出层。一个或更多个层可以包括卷积层,整流线性单元(ReLU)层,池化层,全连接层,归一化层,SoftMax层和/或其他层类型。

[0061] 机器学习模型108和106的层中的每一个层可以包括一个或更多个节点或神经元,其示例包括层140的节点150、152和154,层142的节点156、158、160和162,和层144的节点164。在至少一个实施例中,一个或更多个机器学习模型包括深度神经网络,例如深度卷积神经网络。

[0062] 在至少一个实施例中,协作训练环境100可以被实现为联合推理和/或学习平台。在至少一个实施例中,一个或更多个机器学习模型108可以包括使用训练节点102和机器学习模型106共同训练的联合机器学习模型。在至少一个实施例中,一个或更多个机器学习模型108可以包括中央模型,并且每个机器学习模型106可以包括用于训练一个或更多个机器

学习模型108的中心模型的本地模型。

[0063] 在至少一个实施例中,训练聚合器104的接口管理器120可以从训练节点102接收信息和/或向训练节点102提供信息,诸如以便于一个或更多个机器学习模型108的协作训练。如图1A所示,这样的信息的示例包括机器学习模型106的参数的值(例如,所有参数或参数的子集的值)。共识确定器122可以分析关于来自训练节点102的一个或更多个参数的数据(例如,由接口管理器120接收),以确定训练节点102之间的共识(例如,用于一个或更多个参数的值的一个或更多个共识值)。差异确定器124可以分析与来自训练节点102A,102B和/或102N的一个或更多个参数有关的数据,以确定与训练节点102之间的(例如,由共识确定器122确定的)共识的相应差异量(例如,对于每个训练节点102分别地)。权重确定器126可以至少部分地基于与训练节点102之间的共识的相应差异量(例如,由差异确定器124确定)来确定来自训练节点102A,102B和/或102N的一个或更多个参数的一个或更多个值的一个或更多个权重。参数确定器128可通过使用由权重确定器126确定的一个或更多个相应权重合计(aggregate)来自一个或更多个训练节点102的一个或更多个相应参数的一个或更多个值,来确定一个或更多个机器学习模型108的一个或更多个参数的一个或更多个值。接口管理器120可以将与一个或更多个机器学习模型108的(例如,由参数确定器128确定的)参数的合计值相对应的数据提供给训练节点102。

[0064] 模型管理器112A,模型管理器112B和模型管理器112N(也称为“模型管理器112”)可以使用与参数的合计值相对应的数据来更新一个或更多个机器学习模型106中的相应参数的值。在至少一个实施例中,这包括用一个或更多个机器学习模型108代替机器学习模型106。

[0065] 在至少一个实施例中,除了或代替对一个或更多个机器学习模型108的参数的合计值中的值进行加权,贡献确定器130还可以用于基于权重确定器126确定的权重确定各个训练节点102对训练一个或更多个机器学习模型108的贡献量。在至少一个实施例中,训练聚合器104可以使用贡献量来奖励用于协作训练的相应训练节点102,禁止相应训练节点102进行协作训练(例如,从协作训练的一个或更多个未来和/或后续迭代中),确定(例如,使用权重确定器126)协作训练一个或更多个机器学习模型108的一个或更多个后续迭代的权重,和/或执行其他动作。

[0066] 在至少一个实施例中,一个或更多个机器学习模型108通过多个迭代进行协作训练。在至少一个实施例中,每个迭代可以包括每个训练节点102使用相应的模型训练器110分别训练一个或更多个相应的机器学习模型106以确定机器学习模型106的参数的各个值集。在至少一个实施例中,在一个或更多个迭代开始时,每个机器学习模型106可以包括或者是相同的机器学习模型,具有相同的参数值集,其可以使用模型训练器110进行个体训练。在至少一个实施例中,每个迭代可包括由每个模型训练器110执行的在预定时间段内发生的训练,例如训练时期(epoch),其可以是不同模型训练器110所使用的相同或不同的训练时期。在至少一个实施例中,每个迭代可以包括在预定时间段之后,将机器学习模型106和/或其部分由相应的模型管理器112提供给训练聚合器104的接口管理器120。

[0067] 在至少一个实施例中,一个或更多个迭代可包括训练聚合器104聚合一个或更多个接收的机器学习模型106和/或其部分,并使用参数确定器128确定一个或更多个机器学习模型108的相应参数。在至少一个实施例中,一个或更多个迭代可以包括训练聚合器104

的接口管理器120,然后将一个或多个机器学习模型108和/或其部分分配给一个或多个训练节点102。在至少一个实施例中,一个或多个迭代可以包括一个或多个模型管理器112使用接收的一个或多个机器学习模型108和/或其部分作为对相应机器学习模型106和/或其部分的替代(replacement)。在至少一个实施例中,一个或多个迭代可以包括一个或多个模型训练器110进一步使用对相应的机器学习模型106和/或其部分的替代来训练一个或多个机器学习模型106。

[0068] 在至少一个实施例中,模型训练器110使用图2A和/或图2B的推理和/或训练逻辑215来执行推理和/或训练。例如,每个模型训练器110可以使用推理和/或训练逻辑215的各自实现。在至少一个实施例中,在一个或多个训练节点102之间,推理和/或训练逻辑215至少部分地不同。在至少一个实施例中,在一个或多个训练节点102之间,推理和/或训练逻辑215至少部分地相同。在至少一个实施例中,关于推理和/或训练逻辑215引用的机器学习模型对应于各自的机器学习模型106。

[0069] 在至少一个实施例中,使用图3的训练框架304实现每个模型训练器110,对于每个模型训练器110,其可以是相同的训练框架304或不同的训练框架304。在至少一个实施例中,训练框架304至少部分地在一个或多个训练聚合器104和/或训练节点102上实现。在至少一个实施例中,在训练一个或多个机器学习模型108的一个或多个迭代中,每个机器学习模型106对应于各自的未经训练的机器学习模型306。

[0070] 在至少一个实施例中,训练框架304可以对应于联合学习框架。在至少一个实施例中,每个训练节点102的机器学习模型306由各自的训练框架304部分地训练,然后使用训练聚合器104与机器学习模型306的其他部分进行聚合,以产生一个或多个机器学习模型108。可以通过模型管理器112将机器学习模型306替换为一个或多个机器学习模型108的当前状态。该过程可以重复,直到一个或多个机器学习模型108和/或机器学习模型106对应于图3的经训练的机器学习模型308。

[0071] 在至少一个实施例中,一个或多个模型训练器110使用相同的训练数据集302来训练各自的未经训练的机器学习模型306。在至少一个实施例中,一个或多个模型训练器110使用不同的训练数据集302来训练各自的未经训练的机器学习模型306。在至少一个实施例中,至少一个训练数据集302包括在托管相应训练节点102的设施处现场生成的数据。在至少一个实施例中,训练数据集302或数据代表或训练数据集302或其一部分未到达训练聚合器104和/或托管训练聚合器104的一个或多个服务器(例如,图5的一个或多个聚合服务器504)。

[0072] 在至少一个实施例中,每个训练数据集302包括由一个或多个成像设备(例如,MRI,CT扫描,X射线,超声等),一个或多个测序设备,一个或多个放射学设备,一个或多个基因组学设备和/或其他设备类型(例如,具有相应训练节点102的现场设备)生成的(例如,医学和/或成像)数据。在至少一个实施例中,每个训练数据集302可以包括放射线照相,磁共振成像(MRI),核医学,超声,超声检查,弹性成像,光声成像,层析成像,超声心动图,功能近红外光谱和磁粉成像数据,或者这些类型的任何数据的组合。

[0073] 推理和训练逻辑

[0074] 图2A示出了至少一个实施例中的推理和/或训练逻辑215。下面结合图2A和/或图2B提供关于推理和/或训练逻辑215的细节。

[0075] 在至少一个实施例中,训练逻辑215可以包括或耦合到代码和/或数据存储201以存储图形代码或其他软件,以控制时序和/或顺序,在其中加载权重和/或其他参数信息以配置包括整数和/或浮点单元(统称为算术逻辑单元(ALU))的逻辑。在至少一个实施例中,代码(诸如图形代码)基于神经网络和/或与代码对应的其他机器学习模型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201存储在使用一个或更多个实施例的方面的训练和/或推理期间的输入/输出数据和/或权重参数的正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一层的权重参数和/或输入/输出数据。在至少一个实施例中,代码和/或数据存储201的任何部分都可以与其他片上或片外数据存储一起被包括,包括处理器的L1、L2或L3高速缓存或系统存储器。

[0076] 在至少一个实施例中,代码和/或数据存储201的任何部分可以在一个或更多个处理器或其他硬件逻辑器件或电路的内部或外部。在至少一个实施例中,代码和/或代码和/或数据存储201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)或其他存储器。在至少一个实施例中,对代码和/或编码和/或数据存储201是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对片外的可用存储空间,正在执行的训练和/或推理函数的延迟要求,在神经网络和/或其他机器学习模型的推理和/或训练中使用的数据的批大小或这些因素的某种组合。

[0077] 在至少一个实施例中,推理和/或训练逻辑215可以包括但不限于代码和/或数据存储205,用于存储与神经网络的神经元或层相对应的向后和/或输出权重和/或输入/输出数据,在一个或更多个实施例的方面中,该神经网络被训练为和/或用于进行推理。在至少一个实施例中,代码和/或数据存储205存储在使用一个或更多个实施例的各方面的训练和/或推理期间的输入/输出数据和/或权重参数的反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一层的权重参数和/或输入/输出数据。在至少一个实施例中,训练逻辑215可以包括或耦合到代码和/或数据存储205,用于存储图形代码或其他软件以控制时序和/或顺序,在其中加载权重和/或其他参数信息以配置包括整数和/或浮点单元(统称为算术逻辑单元(ALU))的逻辑。在至少一个实施例中,代码(诸如图形代码)基于神经网络和/或该代码对应的其他机器学习模型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205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205的任何部分可以在一个或更多个处理器或其他硬件逻辑器件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205可以是高速缓存存储器、DRAM、SRAM、非易失性存储器(例如闪存)或其他存储器。在至少一个实施例中,代码和/或数据存储205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,可以取决于片上对片外的可用存储,正在执行的训练和/或推理函数的延迟要求,在神经网络和/或其他机器学习模型的推理和/或训练中使用的数据的批大小或这些因素的某种组合。

[0078] 在至少一个实施例中,代码和/或数据存储201以及代码和/或数据存储205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201以及代码和/或数据存储205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201以及代码和/或

数据存储205可以是部分相同的存储结构和部分分离的存储结构。在至少一个实施例中,代码和/或数据存储201和代码和/或数据存储205的任何部分可以与其他片上或片外数据存储单元包括在一起,包括处理器的L1、L2或L3高速缓存或系统存储器。

[0079] 在至少一个实施例中,推理和/或训练逻辑215可以包括但不限于一个或多个算术逻辑单元(“ALU”)210,包括整数和/或浮点单元,以至少部分地基于训练和/或推理代码(例如,图形代码)或由其指示来执行逻辑和/或数学运算,其结果可能会产生(例如,来自神经网络内部的层或神经元的输出值)存储在激活存储220中的激活,其是存储在代码和/或数据存储201和/或代码和/或数据存储205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 210执行的线性代数和/或基于矩阵的数学生成在激活存储220中存储的激活,其中存储在代码和/或数据存储205中和/或代码和/或将数据存储201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205或代码和/或数据存储201其他片上或片外存储器中。

[0080] 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包括一个或多个ALU 210,而在另一实施例中,一个或多个ALU 210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 210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,代码和/或数据存储201,代码和/或数据存储205以及激活存储220可以在同一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220的任何部分可以与其他片上或片外数据存储单元包括在一起,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。

[0081] 在至少一个实施例中,激活存储220可以是高速缓存存储器、DRAM、SRAM、非易失性存储器(例如,闪存)或其他存储器。在至少一个实施例中,激活存储220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,可以取决于片上对片外可用的存储,进行训练和/或推理函数的延迟要求,在神经网络和/或其他机器学习模型的推理和/或训练中使用的数据的批大小或这些因素的某种组合,选择激活存储220是处理器的内部还是外部,例如,或者包含DRAM、SRAM、闪存或其他存储器类型。在至少一个实施例中,图2A中所示的推理和/或训练逻辑215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2A中所示的推理和/或训练逻辑215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。

[0082] 图2B示出了至少一个实施例中的推理和/或训练逻辑215。在至少一个实施例中,推理和/或训练逻辑215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2B中所示的推理和/或训练逻辑215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2B中所示的推理和/或训练逻辑215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215包括但不限于代码和/或数据存储201以及代码和/或数据存储205,其可以用于存储代码(例如,图形代码)、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2B中所示的至少一个实施例中,代码和/或数据存储201以及代码和/或数据存储205中的每一个都分别与专用的计算资源(例如计算硬件202和计算硬件206)相关联。在至少一个实施例中,计算硬件202和计算硬件206中的每一个包括一个或更多个ALU,这些ALU仅分别对存储在代码和/或数据存储201和代码和/或数据存储205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220中。

[0083] 在至少一个实施例中,代码和/或数据存储201和205以及相应的计算硬件202和206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201和计算硬件202的一个“存储/计算对201/202”得到的激活提供作为代码和/或数据存储205和计算硬件206的下一个“存储/计算对205/206”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201/202和205/206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215中可以包括在存储计算对201/202和205/206之后或与之并行的附加存储/计算对(未示出)。

[0084] 机器学习模型的训练和部署

[0085] 图3示出了在至少一个实施例中的机器学习模型(例如,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302来训练未经训练的机器学习模型306。在至少一个实施例中,训练框架304是PyTorch框架,而在其他实施例中,训练框架304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304训练未经训练的机器学习模型306,并使它能够使用本文所述的处理资源来训练,以生成经训练的机器学习模型308。在至少一个实施例中,权重可以被随机地选择或通过使用深度信念网络预先训练来选择。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。

[0086] 在至少一个实施例中,使用有监督学习来训练未经训练的机器学习模型306,其中训练数据集302包括与输入的期望输出配对的输入,或者其中训练数据集302包括具有已知输出的输入,并对机器学习模型306的输出进行手动排列等级。在至少一个实施例中,以有监督的方式训练未经训练的机器学习模型306,以处理来自训练数据集302的输入,并将结果输出与一组期望或需要的输出进行比较。在至少一个实施例中,然后通过未经训练的机器学习模型306将误差传播回去。在至少一个实施例中,训练框架304调整控制未经训练的机器学习模型306的权重。在至少一个实施例中,训练框架304包括用于监视未经训练的机器学习模型306朝向适合基于已知输入数据(例如新数据312)生成正确答案(例如结果314)的模型(例如经训练的机器学习模型308)收敛的程度的工具。在至少一个实施例中,训练框架304重复地训练未经训练的机器学习模型306,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)来细化未经训练的机器学习模型306的输出。在至少一个实施例中,训练框架304训练未经训练的机器学习模型306,直到未经训练的机器学习模型306达到期望的准确度(accuracy)为止。在至少一个实施例中,然后可以部署经训练的机器学习模型308以实现任何数量的机器学习操作。

[0087] 在至少一个实施例中,使用无监督的学习来训练未经训练的机器学习模型306,其中未经训练的机器学习模型306尝试使用未经标记的数据训练自己。在至少一个实施例中,无监督的学习训练数据集302将包括输入数据而没有任何相关联的输出数据或“地面实况(ground truth)”数据。在至少一个实施例中,未经训练的机器学习模型306可以学习训练数据集302内的分组,并且可以确定各个输入如何与训练数据集302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(self-organizing map),其是经训练的机器学习模型308的一种类型,能够执行对减少新数据312的维度有用的操作。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,其允许识别新数据集312中偏离新数据集312的正常模式的数据点。

[0088] 在至少一个实施例中,可以使用半监督学习,该技术是其中训练数据集302包括标记数据和未标记数据的混合的技术。在至少一个实施例中,训练框架304可以用于执行递增(incremental)学习,诸如通过转移学习技术。在至少一个实施例中,递增学习使得经训练的机器学习模型308能够适应新数据312而不会忘记在初始训练期间灌输的知识。

[0089] 在至少一个实施例中,一个或更多个协作训练机器学习模型108的一个或更多个迭代包括共识确定器122,该共识确定器122分析与来自训练节点102的机器学习模型106的一个或更多个参数相对应的数据(例如,由接口管理器120接收),以确定训练节点102中的一个或更多个之间的共识。在至少一个实施例中,与机器学习模型106的一个或更多个参数相对应的数据表示机器学习模型106的一个或更多个参数的值,一个或更多个机器学习模型108的一个或更多个参数的更新,和/或训练节点102对一个或更多个机器学习模型108的一个或更多个参数的贡献。

[0090] 例如,共识确定器122可将共识计算为表示关于来自机器学习模型106的一个或更多个参数的一个或更多个值的共识的一个或更多个共识值。在至少一个实施例中,共识确定器122使用对异常值具有鲁棒性的算法确定对一个或更多个参数的值的共识,其可以包括使用鲁棒统计,例如一个或更多个参数的值的中位数。

[0091] 在至少一个实施例中,共识值可以对应于多个参数的值的共识。在这样的示例中,计算多于一个参数的共识可以包括共识确定器122合计或组合每个训练节点102的参数的值、更新和/或贡献。为了合计或组合这些值,共识确定器122可以对来自训练节点102的值执行基于矢量的分析。作为示例,合计或组合训练节点102的值可以包括执行参数值之间的欧几里得距离和/或余弦距离。结果,共识确定器122可以为每个训练节点102计算标量值,并且使用每个标量值来计算共识值。

[0092] 在至少一个实施例中,共识值可以对应于由共识确定器122计算的并且表示关于机器学习模型106的至少一个参数的值的共识的统计值。在至少一个实施例中,统计值可以由共识确定器122根据一个或更多个机器学习模型106的一个或更多个参数的值(例如,从训练节点102合计的标量值)来计算。例如,统计值可以对应于来自机器学习模型106的一个或更多个参数的值的中值。统计值的其他示例包括一个或更多个参数的值的集群的平均

值、舍入模式或质心(例如,通过使用一类聚类算法确定集群)。

[0093] 在至少一个实施例中,共识确定器122确定机器学习模型106和/或一个或多个机器学习模型108的所有参数之间的共识。例如,共识确定器122可以来自每个训练节点的相应参数的合计值计算共识值(例如,标量值),然后从其计算共识值。

[0094] 在至少一个实施例中,共识确定器122确定机器学习模型106和/或一个或多个机器学习模型108的参数的子集的共识。例如,共识确定器122可以从来自每个训练节点的相应参数集的合计值计算共识值(例如,标量值),然后从其计算共识值。

[0095] 参数的子集可以对应于机器学习模型106和/或一个或多个机器学习模型108的一个或更多个部分。在至少一个实施例中,机器学习模型106和一个或多个机器学习模型108包括神经网络和/或对应于一个或多个神经网络部分的参数的子集,例如一个或多个节点,层,层类型,节点类型,其组合和/或神经网络的其他划分。作为示例,参数的子集可以对应于节点150、152、154、156、158、160、162和/或164中的一个或多个的参数。在参数的子集对应于层140的情况下,参数的子集可以包括节点150、152和154的参数。在至少一个实施例中,共识确定器122确定关于机器学习模型106的参数的逐层共识。逐层共识可以对应于机器学习模型106的一个或多个层。

[0096] 在至少一个实施例中,共识确定器122确定针对机器学习模型106和/或一个或多个机器学习模型108的多个不同参数集的共识(每个参数集可以对应于一个或多个参数)。例如,共识确定器122可以计算对应于层140的共识值和对应于层142的另一共识值。在至少一个实施例中,共识确定器122分别为每个参数集计算共识值(例如,标量值)。

[0097] 在至少一个实施例中,共识确定器122基于一个或多个因素将机器学习模型106划分为参数集。因素的示例可以包括机器学习模型106的一个或多个输出类别,其中机器学习模型106包括分类器。例如,共识确定器122可基于被激活为产生与一个或多个特定输出类别相对应的一个或多个输出将机器学习模型106的部分(例如,神经元或节点)分组为参数集。在输出类别包括动物类型的情况下,可以将输出狗时激活的部分分组为一个参数集,而将输出猫时激活的部分分组为另一个参数集。附加地或替代地,在至少一个实施例中,参数集可表示机器学习模型106的参数的不同语义分组。在至少一个实施例中,可使用无监督分组或模型部分聚类(例如,基于神经元激活)将机器学习模型106的部分分组为集合。

[0098] 在至少一个实施例中,差异确定器124可以分析与来自训练节点102A,102B和/或102N的一个或多个参数有关的数据,以用训练节点102(例如,由共识确定器122确定)之间的共识确定相应的差异量(例如,针对每个训练节点102)。例如,对于每个训练节点102,差异确定器124可以计算差异值,其表示来自训练节点102的一个或多个参数的更新、贡献和/或值与来自训练节点102的相应共识(例如,由共识确定器122计算)之间的差异量。

[0099] 在至少一个实施例中,共识确定器122使用等式(1)计算共识 $\tilde{d}$

$$[0100] \quad \tilde{d} = \text{Median}(\|p_i - \tilde{p}\|_2) \quad (1)$$

[0101] 为N个训练节点102计算出的跨度(across) $i = p_1, \dots, p_N$, 其中 p_i 表示训练节点i的参数值的矢量(例如,机器学习模型106的所有参数或其子集), $\tilde{p}$ 表示来自训练节点N的合计或组合参数值,例如 p_i 的中值参数值。可以使用等式(1),其中针对训练节点102为参数集

计算共识。

[0102] 在至少一个实施例中,差异确定器124至少部分地基于来自训练节点102的至少一个参数的至少一个值与相应的共识值(例如,统计值)之间的距离来计算差异量。例如,代表差异量的差异值可以被计算为来自训练节点102的至少一个参数相对应的至少一个值与相对应的共识值之间的距离。

[0103] 在至少一个实施例中,至少一个参数的贡献、更新和/或值与共识之间的差异量可以至少部分地基于训练节点102之间贡献、更新和/或值中的离散量。在至少一个实施例中,差异确定器124使用来自训练节点102的更新、值和/或参数之间的差异的规模的估计器(例如,统计估计器)来计算离散量。在一个实施例中,估计器计算训练节点102之间的贡献、更新和/或值中的统计离散度的估计。在至少一个实施例中,差异确定器124使用鲁棒估计器来计算训练节点102之间贡献、更新和/或值中的离散量。

[0104] 在至少一个实施例中,差异确定器124计算离散量作为来自训练节点的相应值(例如,来自合计或组合的参数值,其中多个参数被考虑)的训练节点102之间的贡献、更新和/或值中的统计离散度的度量。由差异确定器124计算的统计离散度的度量的示例包括中值绝对偏差(MAD),标准偏差,四分位数范围,平均绝对差,双权重中方差,平均绝对偏差,距离标准偏差或其组合。

[0105] 在至少一个实施例中,差异确定器124使用等式(2)计算统计离散度 $\tilde{d}^{\text{MAD}}$ 的度量:

$$[0106] \quad \tilde{d}^{\text{MAD}} = \text{Mean}(\|p_i - \tilde{p}\|_2 - \tilde{d}) \quad (2)$$

[0107] 为N个训练节点102计算出的跨度 $i = p_1, \dots, p_N$ 。可以使用等式(1),其中针对关于训练节点102的参数集计算统计离散度。在至少一个实施例中,等式(1)可以与等式(2)一起使用。

[0108] 在至少一个实施例中,差异确定器124使用等式(3)计算差异量 r_i :

$$[0109] \quad r_i = \frac{p_i - \tilde{d}}{a * \tilde{d}^{\text{MAD}}} \quad (3)$$

[0110] 为N个训练节点102计算出的跨度 $i = p_1, \dots, p_N$,其中a可用于使 $\tilde{d}^{\text{MAD}}$ (或可用于其位置的不同估计量)成为渐近正确估计量,用于参数值差异的标准偏差的估计。

[0111] 在至少一个实施例中,权重确定器126至少部分地基于训练节点102之间的共识上的差异量来计算来自训练节点102的值、更新和/或贡献的权重。在至少一个实施例中,来自训练节点102的一个或更多个值、更新和/或贡献的权重使用可以是训练节点102之间的共识上的差异量的函数(例如,成比例)。

[0112] 在至少一个实施例中,具有较大差异量(例如,距共识值的距离较大)的训练节点102比具有较小差异量的训练节点102更有可能是离群值(outlier)。在至少一个实施例中,离群值可以指示数据预处理中的错误,漏洞,错误的超参数选择,故意的对抗行为或与训练节点102关联的其他特征,这可能会对一个或更多个机器学习模型108的协作训练产生负面影响。

[0113] 在至少一个实施例中,权重确定器126为每个训练节点102计算至少一个权重值。在至少一个实施例中,权重确定器126使用模型(诸如函数和/或一个或更多个机器学习模型)将与来自训练节点102的一个或更多个参数相对应的值映射到权重值。在至少一个实施

例中,与来自训练节点102的一个或多个参数相对应的一个或多个值表示训练节点102(例如,由差异确定器124计算)与训练节点102的共识(例如,由共识确定器122计算)之间的差异量。

[0114] 在至少一个实施例中,使用模型(例如,函数)将与来自训练节点102的一个或多个参数相对应的一个或多个值映射到权重值。例如,表示训练节点102和训练节点102的共识之间的差异量的一个或多个值可以用作输出权重值的函数的一个或多个变量。权重确定器126可以使用相同的函数来将每个训练节点102的值映射到相应的权重节点。

[0115] 在至少一个实施例中,权重确定器126使用差异确定器124将与来自训练节点102的一个或多个参数相对应的值拟合(例如缩放)到用于计算权重值的模型(例如函数)。在至少一个实施例中,针对差异确定器124描述的估计器,诸如鲁棒估计器(例如 $\tilde{d}^{MAD}$)被用于将值拟合到用于计算权重值的模型(例如,函数)。例如,等式(3)可以计算 r_i ,使得每个训练节点102的 r_i 拟合到用于计算权重值的模型(例如,函数)。在至少一个实施例中,与来自训练节点102的一个或多个参数相对应的值可以附加地或替代地拟合到使用随机样本共识(RANSAC)来计算权重值的模型(例如,函数)。

[0116] 在至少一个实施例中,随着训练节点102和训练节点102的共识之间的差异量增加,权重确定器126用于计算权重值的模型可以减小到饱和点。图1B示出了至少一个实施例中的模型170的图。权重确定器126可以使用模型170来为来自训练节点102的参数值、贡献和/或更新计算权重值。在图1B中,轴172可以表示与共识的差异量(例如,由差异确定器124计算的),并且轴174可以表示训练节点102的权重值。随着训练节点102和训练节点102的共识之间的差异量增加,模型170可以从最大值176减小(例如,线性或二次)到饱和点178。

[0117] 在至少一个实施例中,模型170对应于双权重函数,绝对误差函数,平方误差函数或Winsorizing函数。在至少一个实施例中,模型170对应于Huber损失函数或Tukey双权重函数。在至少一个实施例中,模型170对应于等式(4),其是Tukey双权重函数的示例:

$$[0118] \quad \hat{w}(r_i) = \begin{cases} \frac{c^2}{b} \left[1 - \left(\frac{r_i}{c} \right)^2 \right]^3, & -c > r_i < c \\ 0, & \text{否则} \end{cases} \quad (4)$$

[0119] 可以为N个训练节点102计算权重值的跨度 $i = p_1, \dots, p_N$,并且b控制最大值176,c控制饱和点178。

[0120] 在至少一个实施例中,权重确定器126可以针对机器学习模型106和/或一个或多个机器学习模型108的一个或多个参数(例如,参数集或所有参数)来确定并归一化训练节点102的权重值。在至少一个实施例中,将权重值归一化可以产生总和为一的归一化权重值。参数确定器128和/或贡献确定器130可以针对训练节点102使用归一化权重值来操作。

[0121] 在至少一个实施例中,权重确定器126可以针对机器学习模型106的任何数量的参数集来确定训练节点102的权重值集(例如,包括每个训练节点102的权重值)。可以为每个参数集计算不同的权重值集。在至少一个实施例中,可以分别使用等式(1)-(4)来计算参数集的权重值集,其中 p_i 表示参数集。

[0122] 在至少一个实施例中,参数确定器128可以通过使用由权重确定器126确定的一个

或更多个相应权重合计来自一个或更多个训练节点102的一个或更多个对应参数的一个或更多个值,来确定一个或更多个机器学习模型108的一个或更多个参数的一个或更多个值。在至少一个实施例中,参数确定器128使用加权平均值来合计参数值,其中,来自特定训练节点102的一个或更多个参数的一个或更多个值用于一个或更多个机器学习模型108的一个或更多个相应参数所计算的权重值加权。例如,在为参数集计算权重值集的情况下,当为一个或更多个机器学习模型108的相应参数合计值时(例如,整个层可以共享应用于参数值的相同权重值),可以用相应的权重值对来自训练节点102的每个参数的每个值进行加权。

[0123] 在至少一个实施例中,参数确定器128可通过合计来自一个或更多个训练节点102的一个或更多个相应参数的一个或更多个值,通过排除来自一个或更多个训练节点102的一个或更多个相应参数的一个或更多个值(除了使用或代替使用一个或更多个值的权重值),来确定一个或更多个机器学习模型108的一个或更多个参数的一个或更多个值。例如,参数确定器128可以至少部分地基于与共识的相应差异量(例如,由差异确定器124确定的),排除来自一个或更多个训练节点102的一个或更多个对应参数的一个或更多个值。作为示例,可以排除与大于阈值的差异量相对应的值。在至少一个实施例中,权重确定器126可以不包括在训练聚合器104中,并且权重值可以不用于合计来自训练节点102的参数值。

[0124] 在至少一个实施例中,参数确定器128用确定的参数值来更新一个或更多个机器学习模型108。在至少一个实施例中,用相应的合计参数值更新一个或更多个机器学习模型108的每个参数。在至少一个实施例中,当机器学习模型108的一个或更多个部分的一个或更多个参数被更新时,一个或更多个机器学习模型108的一个或更多个其他部分的一个或更多个参数可以不被更新。

[0125] 在至少一个实施例中,训练聚合器104可以将由参数确定器128更新的一个或更多个机器学习模型108的一个或更多个部分提供给训练节点102。在至少一个实施例中,模型管理器112可以用一个或更多个机器学习模型108更新相应的机器学习模型106(例如,通过将一个或更多个机器学习模型106的参数值替换为一个或更多个机器学习模型108的参数值)。在至少一个实施例中,随后的协作训练迭代可以单独地训练更新的机器学习模型106,然后将经训练的更新的机器学习模型106提供给训练聚合器104以用于随后的合计到一个或更多个机器学习模型108中。

[0126] 在至少一个实施例中,贡献确定器130基于由权重确定器126确定的权重(例如,预先归一化的或归一化的)来确定训练节点102对训练一个或更多个机器学习模型108的贡献量。在至少一个实施例中,贡献确定器130可基于权重确定器126确定的权重来确定各个训练节点102对训练一个或更多个机器学习模型108的贡献量或贡献分数。

[0127] 在至少一个实施例中,贡献确定器130可以计算贡献分数以至少部分地偏向没有为太接近训练节点102的共识的一个或更多个参数贡献数据的训练节点102。在至少一个实施例中,贡献确定器130可以计算贡献分数,以至少部分地偏向没有为与训练节点102的共识相差太远的一个或更多个参数贡献数据的训练节点102。

[0128] 在至少一个实施例中,贡献确定器130可以计算训练节点102的贡献分数,以使得其从权重值集的最大权重值增加(例如,线性或二次地)到一个或更多个最大点(例如,本地的或全局的),作为训练节点102与训练节点102的共识之间的共识量。在至少一个实施例中,贡献分数的一个或更多个最大值点或一个或更多个截止点可以对应于与来自训练节点

102的参数值的模型的最大值的一个或多个标准偏差或百分位数。例如,在图1B中,最大或截止点180可以基于距最大值174的距离,例如一个标准偏差。在至少一个实施例中,贡献确定器130可以计算训练节点102的贡献分数,使得其从一个或多个最大值点(例如,到一个或多个饱和点)减小(例如,线性或二次地)为训练节点102和训练节点102的共识之间的共识量。

[0129] 在至少一个实施例中,贡献确定器130可以随着时间更新训练节点102的贡献分数。例如,贡献确定器130可以在多个迭代上计算和跟踪训练节点102的贡献分数,例如使用平均值,运行平均值或其他运行贡献分数度量。在至少一个实施例中,贡献确定器130可以事后确定一个或多个贡献分数,例如通过在附加训练后将针对一个或多个机器学习模型108的一种状态的贡献与一个或多个机器学习模型108的一种或更多种未来状态进行比较。在至少一个实施例中,这允许贡献确定器130从当前训练状态中识别可能不容易识别的对训练有意义的贡献。

[0130] 在至少一个实施例中,训练聚合器104可以使用贡献量或贡献分数来奖励相应的训练节点102以进行协作训练。在至少一个实施例中,训练聚合器104可以使用贡献量或贡献分数来禁止相应的训练节点102进行协作训练(例如,从一个或多个未来和/或随后的协作训练迭代)。禁止可以关于一个或多个机器学习模型108和/或机器学习模型106或一个或多个机器学习模型108和/或机器学习模型106整体的一个或多个特定部分。例如,可以为一个或多个机器学习模型108和/或机器学习模型106的参数集计算训练节点的贡献分数,并将其用于实现相应的有限禁止。在至少一个实施例中,例如可以基于一个或多个相应的贡献分数,对于一个或多个机器学习模型108和/或机器学习模型106的一个或多个其他部分可以同时不实施禁止。

[0131] 在至少一个实施例中,禁止导致来自机器学习模型106的训练节点102的值未用于确定训练节点102之间的共识,未用于确定训练节点102与训练节点102的共识之间的差异量,未用于确定训练节点102的权重值,和/或未用于基于与训练节点102相关联的一个或多个贡献分数为一个或多个机器学习模型108合计训练节点102的参数。

[0132] 在至少一个实施例中,禁止可以是永久的或暂时的。在至少一个实施例中,针对训练节点102的禁止是禁止接收更新的一个或多个机器学习模型108和/或其部分。在至少一个实施例中,禁止可以基于训练聚合器104确定贡献分数未能超过阈值。

[0133] 在至少一个实施例中,训练聚合器104可以使用贡献量或贡献分数来确定或计算一个或多个协作训练机器学习模型1018的一个或多个后续迭代的权重(例如,使用权重确定器126),和/或执行其他操作。例如,对于具有较高贡献分数的训练节点102,可以增加权重值,和/或对于具有较低贡献分数的训练节点102,可以减小权重值。可以基于关于一个或多个机器学习模型108和/或机器学习模型106或一个或多个机器学习模型108和/或机器学习模型106整体的一个或多个特定部分的贡献分数来调整权重值。

[0134] 图4示出了至少一个实施例中的用于协作训练的流程图400。在至少一个实施例中,训练节点102可以根据流程图400来协作地训练一个或多个机器学习模型108。例如,流程图400可以用于使用训练节点102实现一个或多个机器学习模型108的参数值的联合学习。

[0135] 训练节点102中的每个可以通过一个或多个通信来传输数据,该数据表示对参

与协作训练一个或更多个机器学习模型108的迭代的请求。在至少一个实施例中,训练节点102A向训练聚合器104提供请求402A,训练节点102B向训练聚合器104提供请求402B,训练节点102N向训练聚合器104提供请求402N。

[0136] 在至少一个实施例中,在框404,接口管理器120分析来自训练节点102的请求402A和402B至402N(也称为“请求402”)。对请求402进行分析可以包括接受一个或更多个请求402和/或拒绝一个或更多个请求402。在至少一个实施例中,接受和/或拒绝一个或更多个请求402可至少部分基于来自协作训练的一个或更多个先前迭代(例如,训练节点102是否被禁止等)的由贡献确定器130确定的贡献分数和/或由权重确定器126确定的权重值。

[0137] 在至少一个实施例中,分析请求402包括认证训练节点102中的一个或更多个(例如,使用请求402中的信息,诸如表示密码和/或密钥的数据)和/或验证对一个或更多个训练节点102的认证。未认证的训练节点102可以拒绝参与协作训练的迭代。

[0138] 在至少一个实施例中,训练聚合器104基于框404处的分析请求402,通过一个或更多个通信来响应一个或更多个训练节点102中的请求402。在至少一个实施例中,请求可以包括表示一个或更多个机器学习模型108的一个或更多个部分的数据。响应406A和406N(也称为“响应406”)是这种响应的示例。每个响应406可以包括一个或更多个机器学习模型108的副本和/或足以使相应训练节点102产生一个或更多个机器学习模型108的当前状态的副本的数据(例如,以使机器学习模型106跨参与的训练节点102同步)。在流程图400中,在框404处,可以基于分析请求402B来拒绝来自训练节点102B的请求402B。

[0139] 在至少一个实施例中,当训练节点102A接收到响应406A时,模型管理器112A可以更新一个或更多个机器学习模型106A以对应于一个或更多个机器学习模型108,然后在框408A,使用模型训练器110A来训练机器学习模型106A(例如,在一个时期内)。在至少一个实施例中,当训练节点102N接收到响应406N时,模型管理器112N可以更新一个或更多个机器学习模型106N以对应于一个或更多个机器学习模型108,然后在框408N,使用模型训练器110N来训练机器学习模型106N(例如,在一个时期内)。

[0140] 在至少一个实施例中,在框408A处训练了一段时间之后,训练节点102A可以通过表示请求410A的一个或更多个通信来传输数据,以将来自一个或更多个机器学习模型106A的参数值聚集到一个或更多个机器学习模型108。同样在框408N训练一段时间后,训练节点102N可以通过表示请求410N的一个或更多个通信来传输数据,以将来自一个或更多个机器学习模型106N的参数值合计到一个或更多个机器学习模型108。

[0141] 在至少一个实施例中,训练聚合器104的接口管理器120可以从可能参与一个或更多个机器学习模型108的协作训练的其他训练节点接收请求410A、410N和/或类似请求。在框412,共识确定器122,差异确定器124,权重确定器126,参数确定器128和/或贡献确定器130可以使用来自请求410A、410N和/或类似请求的信息来更新一个或更多个机器学习模型108和/或计算一个或更多个训练节点102的贡献分数。在至少一个实施例中,然后可以针对一个或更多个协作训练机器学习模型108的任意数量的后续迭代来重复流程图400。

[0142] 图5示出了在至少一个实施例中的用于协作训练的网络环境500。在至少一个实施例中,网络环境500可以包括一个或更多个客户端设备502A和502B至502N(也称为“客户端设备502”)和一个或更多个聚合服务器504。在至少一个实施例中,每个客户端设备502A可以包括一个或更多个训练节点102。例如,一个或更多个客户端设备502A包括训练节点

102A,一个或多个客户端设备502B包括训练节点102B,一个或多个客户端设备502N包括训练节点102N。

[0143] 在至少一个实施例中,一个或多个聚合服务器504包括训练聚合器104的一个或多个部分。在至少一个实施例中,训练聚合器104的一个或多个部分至少部分地跨一个或多个客户端上设备502和/或一个或多个其他客户端设备(其可以包括或不包括训练节点102)分配。在至少一个实施例中,网络环境500包括对等网络环境。在至少一个实施例中,一个或多个聚合服务器504不包括在网络环境500中。在至少一个实施例中,其包括客户端-服务器网络环境。在至少一个实施例中,客户端设备502每个与一个或多个聚合服务器504的客户端或用户相对应。

[0144] 在至少一个实施例中,一个或多个聚合服务器504可以不在网络环境500中和/或用于协作训练的一个或多个方面。例如,网络环境500可以包括由客户端设备502和训练节点102使用的对等网络,用于协作学习(例如,以实现联合学习)。在至少一个实施例中,训练聚合器104的功能可以分布在一个或多个客户端设备502上和/或分配给一个或多个客户端设备502。在至少一个实施例中,客户端设备502可以单独实现训练聚合器104的所有功能。在至少一个实施例中,客户端设备502可以单独地或部分地实现训练聚合器104的某些功能。

[0145] 在至少一个实施例中,图5中所示的每个组件都可以经由网络506彼此通信,网络506可以包括一个或多个局域网(LAN)和/或广域网(WAN)。

[0146] 在至少一个实施例中,一个或多个客户端设备502对应于边缘设备。在至少一个实施例中,每个边缘设备可以是至少部分地由一个或多个聚合服务器504托管的联合或协作学习平台。

[0147] 在至少一个实施例中,客户端设备502和/或边缘设备可以体现为个人计算机(PC),膝上型计算机,移动设备,智能电话,平板计算机,智能手表,可穿戴式计算机,个人数字助理(PDA),MP3播放器,全球定位系统(GPS)或设备,医疗设备,视频播放器,手持通信设备,虚拟现实耳机,游戏设备或系统,娱乐系统,车辆计算机系统,嵌入式系统控制器,遥控器,应用程序,消费电子设备,服务器,数据中心或工作站。在至少一个实施例中,一个或多个客户端设备502和/或边缘设备可以位于彼此不同的医院、医学实验室和/或医学研究设施处。

[0148] 图6示出了至少一个实施例中的方法600的流程图。在框B602,方法600包括确定来自训练节点的贡献之间的共识。在至少一个实施例中,共识确定器122可以确定来自训练节点102的贡献之间的共识。

[0149] 在框B604处,方法600包括基于贡献和贡献的共识之间的差异量来确定至少一个权重。在至少一个实施例中,权重确定器126可以基于来自训练节点102A的贡献和来自训练节点102的贡献的共识之间差异量来确定至少一个权重。

[0150] 在框B606,方法600包括使用至少一个权重值对贡献的合计中的贡献进行加权。在至少一个实施例中,参数确定器128可以使用至少一个权重值在来自训练节点102A的贡献的合计中对来自训练节点102A的贡献进行加权。如图6所示,可以针对协作训练的任何数量的迭代来重复方法600。

[0151] 图7示出了至少一个实施例中的方法700的流程图。在框B702,方法700包括至少部

分地基于来自客户端的更新与客户端之间关于更新的共识之间的差异量来计算权重值。在至少一个实施例中,权重确定器126可以至少部分地基于来自与客户端设备502A相对应的客户端的对一个或多个机器学习模型108的一个或多个参数的更新与对应于客户端设备502的客户端之间关于一个或多个参数的更新的共识之间的差异量来计算权重值。

[0152] 在框B702处,方法700包括用权重值加权来自一个或多个机器学习模型中的客户端的更新。在至少一个实施例中,训练聚合器104可以使用参数确定器128用权重值对来自与一个或多个机器学习模型108中的客户端设备502A相对应的客户端的更新进行加权。如图7所示,方法700可以针对协作训练的任何数量的迭代而重复。

[0153] 图8示出了至少一个实施例中的方法800的流程图。在框B802,方法800包括将对一个或多个机器学习模型的一个或多个参数的更新中的一更新转换为函数的值。在至少一个实施例中,权重确定器126将来自训练节点102A的一个或多个参数的一个或多个值转换为至少部分地基于对来自训练节点102的更新的值的统计分散性以及更新与更新的共识之间的距离的度量将一个或多个机器学习模型108的一个或多个参数的更新转换为函数的值。

[0154] 在框B802处,方法800包括用相应的权重值对一个或多个机器学习模型中的更新加权。在至少一个实施例中,训练聚合器104可以使用参数确定器128用归一化的权重值对来自一个或多个机器学习模型108中的训练节点102A的更新进行加权。如图8所示,方法800可以针对协作训练的任何数量的迭代进行重复。

[0155] 图9示出了至少一个实施例中的方法900的流程图。在框B902,方法900包括使模型适合来自训练节点的至少一个参数。在至少一个实施例中,训练聚合器104可以使用共识确定器122和/或差异确定器124来使模型170适合于来自一个或多个机器学习模型108的训练节点102的至少一个参数的值。

[0156] 在框B904处,方法900包括将来自训练节点的至少一个值映射到模型的值。在至少一个实施例中,权重确定器126可以将来自训练节点102A的至少一个参数的至少一个值映射到模型170的值(例如,使用来自训练节点102的共识的来自训练节点102A的至少一个值的差异量)。

[0157] 在框B906处,方法900包括用模型值对来自训练节点的值进行加权。在至少一个实施例中,训练聚合器104可以使用参数确定器128以用一个或多个机器学习模型108中的模型170的值(例如,归一化的)加权至少一个参数的至少一个值。如图9所示,方法900可以针对协作训练的任何数量的迭代而重复。

[0158] 图10示出了至少一个实施例中的方法1000的流程图。在框B1002,方法1000包括基于贡献和贡献的共识之间的差异量来确定贡献分数。在至少一个实施例中,贡献确定器130可基于来自训练节点102A对一个或多个协作训练机器学习模型108的贡献来确定训练节点102A和/或与客户端设备502A相对应的客户端的贡献分数。

[0159] 在框B1004处,方法1000包括基于贡献分数执行一个或多个动作。例如,训练聚合器104可以使用贡献分数来奖励训练节点102A进行协作训练,禁止训练节点102A进行协作训练(例如,来自一个或多个未来和/或后续的协作训练迭代),确定权重(例如,使用权重确定器126)进行协作训练一个或多个机器学习模型108的一个或多个后续迭代,和/或执行其他操作。

[0160] 参数审查

[0161] 图11A示出了至少一个实施例中的示例参数审查器(reviewer)1102。在至少一个实施例中,参数审查器1102包括接口管理器1104,参数评估器1106和参数比较器1108。图11B示出了至少一个实施例中的示例审查聚合器1110。在至少一个实施例中,审查聚合器1110包括接口管理器1112,审查分析器1114和参数确定器1116。

[0162] 在至少一个实施例中,可以在网络环境500中实现一个或更多个参数审查器1102和一个或更多个审查聚合器1110以进行协作训练。在至少一个实施例中,一个或更多个客户端设备502包括参数审查器1102,并且一个或更多个聚合服务器504包括审查聚合器1110。在至少一个实施例中,一个或更多个聚合服务器504可以不包括审查聚合器1110,例如网络环境500包括对等网络环境。在至少一个实施例中,例如在对等网络环境中,一个或更多个客户端设备502可以包括审查聚合器1110和/或实现审查聚合器1110的功能的一个或更多个部分。

[0163] 一个或更多个参数审查器1102和/或一个或更多个审查聚合器1110可以用于促进协作学习,例如联合学习。在至少一个实施例中,一个或更多个参数审查器1102和/或一个或更多个审查聚合器1110可以用于促进转移学习,例如在图1A的训练节点102之间(例如,机器学习模型106和/或108的参数)。在至少一个实施例中,一个或更多个参数审查器1102和/或一个或更多个审查聚合器1110可用于促进联合转移学习,多任务联合学习,水平联合学习和/或垂直联合学习,例如在图1的训练节点102之间(例如,机器学习模型106和/或108的参数)(例如,机器学习模型106和/或108的参数)。

[0164] 在至少一个实施例中,参数审查器1102的接口管理器1104可以从其他组件接收信息和/或向其他组件提供信息,例如,其他参数审查器1102的其他接口管理器1104和/或一个或更多个审查聚合器1110的接口管理器1112。在至少一个实施例中,这种信息的示例包括图1A的一个或更多个机器学习模型106中的一个或更多个的神经网络权重信息和/或MLM参数信息和/或图1A的一个或更多个机器学习模型108。在至少一个实施例中,接口管理器1104接收MLM参数信息1120A,1120B和/或1120N,其每一个可以对应于一个或更多个机器学习模型106和/或一个或更多个机器学习模型的一个或更多个的神经网络权重信息。在至少一个实施例中,参数评估器1106可以评估MLM参数信息1120A,1120B和/或1120N的一个或更多个部分(例如,分别地)以生成评估结果信息。同样在至少一个实施例中,参数比较器1108可以将来自参数评估器1106评估MLM参数信息1120A,1120B和/或1120N的一个或更多个部分的评估结果信息彼此进行比较。在至少一个实施例中,参数比较器1108至少部分地基于这样的比较来为MLM参数信息1120A,1120B和/或1120N(或其一部分)中的一个或更多个计算得分信息。

[0165] 在至少一个实施例中,审查聚合器1110的接口管理器1112可以从其他组件接收信息和/或向其他组件提供信息,诸如一个或更多个参数审查器1102的一个或更多个接口管理器1104。在至少一个实施例中,这样的信息的示例包括图1A的一个或更多个机器学习模型106和/或图1A的一个或更多个机器学习模型108中的一个或更多个的神经网络权重信息和/或MLM参数信息。在至少一个实施例中,这种信息的示例包括MLM审查信息,诸如来自一个或更多个参数比较器1108的一个或更多个计分信息集。例如,MLM审查信息1118A可以来自一个参数审查器1102,MLM审查信息1118B可以来自另一个参数审查器1102,而MLM审查信

息1118N可以来自另一个参数审查器1102。

[0166] 在至少一个实施例中,审查分析器1114可以分析MLM审查信息,其可以包括分析一组或更多组MLM参数信息1120的得分信息(例如,来自MLM审查信息1118)。在至少一个实施例中,审查分析器1114可以另外地或替代地生成针对神经网络信息的一个或更多个部分的评分信息。在这样的示例中,审查分析器1114可以包括参数比较器1108,并且来自参数审查器1102的MLM审查信息可以包括来自一个或更多个参数评估器1106的评估结果信息(另外一个或更多个参数审查器1102可以不包括参数比较器1108)。

[0167] 在至少一个实施例中,审查聚合器1110的参数确定器1116可以至少部分地基于来自审查分析器1114的分析结果信息来确定一个或更多个MLM的MLM参数信息(例如,神经网络信息),例如一个或更多个机器学习模型108。在至少一个实施例中,确定MLM参数信息的参数确定器1116可以包括基于相应的MLM审查信息更新和/或替换一个或更多个MLM模型(例如,一个或更多个机器学习模型108)的MLM参数信息。在至少一个实施例中,确定MLM参数信息的参数确定器1116可以包括基于相应的MLM审查信息从更新和/或替换一个或更多个MLM模型的MLM参数信息中排除MLM参数信息1120的一个或更多个部分,诸如一个或更多个机器学习模型108。

[0168] 在至少一个实施例中,审查聚合器1110的接口管理器1112接收MLM参数信息1120A至1120N,每个MLM参数信息可以对应于相应的一个或更多个MLM模型。例如,每个MLM参数信息1120可以对应于由相应的训练节点(例如,一个时期)训练了一段时间的相应的MLM模型,例如图1A的训练节点102。如本文所述。作为示例,MLM参数信息1120A可以对应于MLM 106A,MLM参数信息1120B可以对应于MLM 106B,并且MLM参数信息1120N可以对应于MLM 106N。在至少一个实施例中,可以从客户端设备502提供由审查聚合器1110针对特定MLM 106接收的MLM参数信息1120,该客户端设备502包括至少部分地训练了这种MLM 106的训练节点102。例如,每个训练节点102的模型管理器112可以提供相应的MLM参数信息1120。

[0169] 在至少一个实施例中,参数确定器1116可以负责合计MLM参数信息1120A,MLM参数信息1120B和/或MLM参数信息1120N,以便更新MLM 106和/或108中的一个或更多个(例如,用于协作训练)。例如,参数确定器1116可以用于基于MLM参数信息1120来更新MLM108。在至少一个实施例中,参数确定器1116可以对应于图1A的参数确定器128。在至少一个实施例中,参数确定器1116可以使用共识确定器122,差异确定器124,权重确定器126和/或贡献确定器130,如本文所述。在至少一个实施例中,参数确定器1116可以不使用共识确定器122,差异确定器124,权重确定器126和/或贡献确定器130,如本文所述。

[0170] 在至少一个实施例中,为了聚合MLM参数信息1120A,MLM参数信息1120B和/或MLM参数信息1120N,审查聚合器1110可以使用接口管理器1112将MLM参数信息1120A,MLM参数信息1120B,MLM参数信息1120N中的一个或更多个和/或其一部分提供给一个或更多个参数审查器1102进行审查。例如,参数审查器1102被示为接收MLM参数信息1120A至1120N,可以从审查聚合器1110接收其中的任何部分用于审查和/或可以由审查聚合器1110分配给参数审查器1102以进行审查。审查聚合器1110可以类似地将MLM参数信息1120提供给其他参数审查器1102以进行审查。

[0171] 在至少一个实施例中,一个或更多个参数审查器1102可以在与训练节点102相同

的客户端和/或客户端设备502上或与之相关联。在这样的示例中,审查聚合器1110可以不提供这样的具有MLM参数信息1120的对应部分的客户端和/或客户端设备502。例如,每个客户端和/或客户端设备502可以被分配以审查(review)至少一个不同的客户端和/或客户端设备502的MLM参数信息1120。在至少一个实施例中,一个或多个参数查看器1102可以不包括训练节点102和/或可能未提供MLM参数信息1120的一部分给审查聚合器1110以进行审查。

[0172] 在至少一个实施例中,每个参数审查器1102的接口管理器1104接收用于匿名审查的MLM参数信息1120,而无需指示与MLM参数信息1120的对应部分相关联的客户端和/或客户端设备。在至少一个实施例中,除了由相关的训练节点102训练的那些部分(例如,在相同的客户端设备502上和/或在相同的客户端上)之外,每个参数审查器1102审查由训练节点102提供的MLM参数信息1120的每个部分。在至少一个实施例中,审查聚合器1110的接口管理器1112不将MLM参数信息1120的一个或多个部分发送给特定参数审查器1102以进行审查。例如,在审查聚合器1110在图5的一个或多个聚合服务器504上的情况下,或者不包括一个或多个聚合服务器504,相应的数据可以在客户端设备502之间和/或以另一方式对等交换。在至少一个实施例中,MLM参数信息1120的部分分配到参数审查器1102以审查可以是隐式的,诸如由客户端设备502之间的交换算法所定义的(例如,全部交换,环交换等)。可以进行考虑以确保要审查的MLM参数信息1120的对应部分的来源的匿名性。

[0173] 在至少一个实施例中,为了审查MLM参数信息1120,参数审查器1102的参数评估器1106被配置为评估MLM参数信息1120的一个或多个部分以从中产生评估结果信息。评估MLM参数信息1120可以包括参数评估器1106计算应用于一个或多个MLM的MLM参数信息1120的一个或多个对应部分的准确度的一个或多个度量。例如,为了评估MLM参数信息1120A,参数评估器1106可以测试和测量包括由MLM参数信息1120A定义的参数的一个或多个MLM的准确度。这可以包括参数评估器1106对包括由MLM参数信息1120A定义的参数的一个或多个MLM(例如,MLM 106)执行验证,以计算一个或多个输出准确度度量。例如,在一个或多个MLM要识别一个或多个图像内的一个或多个对象的情况下,参数评估器1106可以根据一个或多个图像内的一个或多个对象的识别准确度来评估准确度。与识别准确度有关的度量可以基于所识别对象的数量和/或百分比,正确和/或不正确识别的对象(例如,误报或错误识别),正确和/或不正确分类的对象等。尽管描述了对对象识别作为示例,在至少一个实施例中,被评估的MLM可以不识别一个或多个图像中的一个或多个对象和/或可以执行一个或多个附加任务。

[0174] 在至少一个实施例中,参数审查器1102的参数评估器1106生成评估结果信息,该评估结果信息表示针对不同组的MLM参数信息1120的识别准确度,针对不同组的MLM参数信息1120的识别准确度量度的值,和/或用于不同组的MLM参数信息1120的识别准确度和/或识别准确度量度值的因素。在MLM审查信息1118包括评估结果信息的情况下,此信息的任何组合都可以包括在其中(例如,供审查分析器1114使用)。在至少一个实施例中,参数审查器1102的参数评估器1106可以针对被评估的每组MLM参数信息118计算和/或确定输出准确度度量值。在至少一个实施例中,参数评估器1106可以使用相同的验证数据集(例如,相同的输入图像和地面实况数据集)评估每个MLM和/或相应的MLM参数信息1120。

[0175] 在至少一个实施例中,参数比较器1108比较评估结果信息(例如,来自参数评估器

1106)以生成可以包括在MLM审查信息1118中的比较信息。比较信息的示例包括关于MLM参数信息1120的集合和/或其特定部分的评分信息和/或排名信息。例如,参数比较器1108可以将MLM参数信息1120从最精确到最不精确进行排名,和/或基于评估结果信息(例如,识别准确度度量值)来计算评分信息。在至少一个实施例中,比较信息和/或得分信息可以相对于参考MLM和/或MLM参数信息1120的集。作为示例,在参数审查器1102与特定训练节点102相关联的情况下,参考MLM和/或MLM参数信息1120的集合可对应于至少部分地由该训练节点102训练的MLM。

[0176] 在至少一个实施例中,在MLM参数信息1120A来自图1A的MLM 106A(其由训练节点102A训练)的情况下,模型管理器112A可以向参数审查器1102提供MLM参数信息1120A(例如,在同一客户端设备502内),以用作基准,参考或标准,其他MLM参数信息1120针对该基准,参考或标准由参数比较器1108进行比较。例如,排名或得分信息可以相对于MLM参数信息1120A。在至少一个实施例中,可以基于相应的评估结果信息与MLM参数信息1120A(参考MLM参数信息)的相似程度来对MLM参数信息1120进行排名或打分,其中相似程度较高的结果排名高于相似程度较低的结果。在至少一个实施例中,除了或代替排名或得分信息,比较信息可以对应于关于参数确定器1116在合计用于MLM的参数时是否应使用MLM参数信息1120的特定集合或部分的投票或意见。例如,参数比较器1108可以投票选为包括,其中相应的得分信息、排名信息和/或一个或多个输出准确度度量值在MLM参数信息1120A(参考MLM参数信息)的一个或多个阈值内。

[0177] 在至少一个实施例中,每个参数审查器1102可以独立地确定一部分MLM审查信息1118,并将相应的MLM审查信息1118提供给审查聚合器1110由参数确定器1116进行合计。在至少一个实施例中,采用单个审查聚合器1110。在至少一个实施例中,可以采用多个审查聚合器1110。例如,不同的审查聚合器1110可以合计不同的MLM审查信息集1118,然后可以将其应用于更新一个或多个MLM。在至少一个实施例中,来自不同审查聚合器1110的MLM审查信息1118的合计集可以被合并并且由参数确定器1116使用以确定用于MLM(例如,MLM 108)的参数。

[0178] 在至少一个实施例中,审查分析器1114可以分析MLM审查信息,并且可以由参数确定器1116使用以确定MLM参数信息。在至少一个实施例中,审查分析器1114可以分析由来自一个或多个参数审查器1102的一个或多个参数比较器1108生成的比较信息。在MLM审查信息包括由一个或多个参数评价器1106生成的至少一个评价结果的示例中,审查分析器1114可以包括参数比较器1108以产生和分析至少一些比较信息。

[0179] 在至少一个实施例中,审查分析器1114可以基于来自多个参数审查器1102的MLM审查信息1118,确定MLM参数信息1120的一个或多个部分,以从MLM参数信息1120的合计中排除。例如,在比较信息包括用于包括或排除的投票的情况下,如本文所述,审查分析器1114可以基于这样的投票(例如,票数)确定排除或包括MLM参数信息1120的一部分(例如,来自训练节点和/或其特定子集的所有参数)。在至少一个实施例中,可以至少部分地基于多数票用于包括(或超过一些其他阈值)来包括MLM参数信息1120的一部分,和/或至少部分地基于多数票用于排除(或超出某些其他阈值)来排除MLM参数信息1120的一部分。在至少一个实施例中,审查分析器1114附加地或替代地分析来自不同参数审查器1102的MLM参数信息1120的一部分的得分信息,以确定是否包括或排除这种MLM参数信息1120。在至少一个

实施例中,可以至少部分地基于确定相应得分信息(例如,平均值)的合计超过阈值来包括MLM参数信息1120,和/或至少部分地基于确定相应得分信息(例如,平均值)的合计未能超过阈值来排除MLM参数信息1120。

[0180] 在至少一个实施例中,审查分析器1114可以基于来自多个参数审查器1102的MLM审查信息1118,为MLM参数信息1120的合计确定MLM参数信息1120的一个或更多个权重。例如,审查分析器1114可以使用权重确定器126至少部分地基于相应的比较信息来确定MLM参数信息1120的一个或更多个部分的权重。在至少一个实施例中,MLM参数信息1120的一个或更多个部分可以具有与相应的合计得分信息成比例的权重值。例如,来自训练节点102的MLM参数信息1120的一部分可以具有权重值,该权重值对应于其相对于来自其他训练节点102的MLM参数信息1120的相应部分的分数。在至少一个实施例中,MLM参数信息1120的一个或更多个部分具有与相应的合计投票信息成比例的权重值。例如,来自训练节点102的MLM参数信息1120的一部分可以具有与其相对于来自其他训练节点102的MLM参数信息1120的对应部分的投票数和/或投票总数相对应的权重值。在至少一个实施例中,MLM参数信息1120的一个或更多个部分可以具有权重值,该权重值至少部分地基于相应的训练节点102(和/或客户端或客户端设备502)的贡献信息,如本文所述。

[0181] 如本文所述,在至少一个实施例中,审查聚合器1110的参数确定器1116可以至少部分地基于来自审查分析器1114的分析结果信息确定一个或更多个MLM(例如一个或更多个机器学习模型108)的MLM参数信息(例如,神经网络信息)。在至少一个实施例中,参数确定器1116基于相应的MLM审查信息确定MLM参数信息可以包括更新和/或替换一个或更多个MLM模型(例如,一个或更多个机器学习模型108)的MLM参数信息。在至少一个实施例中,确定MLM参数信息的参数确定器1116可以包括基于相应的MLM评论信息(例如,基于审查分析器1114做出的确定)从更新和/或替换一个或更多个MLM模型(诸如一个或更多个机器学习模型108)的MLM参数信息中排除MLM参数信息1120的一个或更多个部分。

[0182] 在至少一个实施例中,一个或更多个参数审查器1102和一个或更多个审查聚合器1110可以根据图4的流程图400的描述来操作。在至少一个实施例中,一个或更多个参数审查器1102和一个或更多个聚合器1110在多次迭代上协作地训练一个或更多个MLM,以进行联合转移学习,多任务联合学习,水平联合学习和/或垂直联合学习,或另一个协作训练技术。在至少一个实施例中,一个或多个训练节点102可以包括用于更新本地MLM(诸如MLM 106)的参数检查器1102和/或检查聚合器1110。在至少一个实施例中,本地MLM 106的一个或多个部分可以在训练节点102和/或客户端设备502之间(例如,与MLM 108)周期性地同步106(例如,在参数合计之后)。在至少一个实施例中,一个或多个MLM 106或其部分可以不在训练节点102和/或客户端设备502之间同步(例如,用于转移学习或多任务学习)。在至少一个实施例中,一个或更多个MLM 106集或其部分可以被周期性地同步(例如,在参数合计之后),而一个或更多个其他MLM 106集可以不被同步(例如,其中训练节点102被由任务分组)。

[0183] 在至少一个实施例中,不同的审查聚合器1110可以对来自不同组的训练节点102和/或客户端设备502的不同组的MLM评论信息1118进行操作以合计MLM参数信息。在至少一个实施例中,一个或更多个参数审查器1102可以进一步查看这样的MLM参数信息合计集和/或这样合计的MLM参数信息可以由参数确定器1116和/或128进一步合计。

[0184] 模型分支

[0185] 在至少一个实施例中,审查聚合器1110和/或训练聚合器104可以分支MLM 108和/或MLM106。MLM的每个分支可以对应于一个或更多个训练节点102,客户端设备502和/或客户端的不同集。在至少一个实施例中,当更新MLM的分支时,参数确定器128和/或1116可以合计来自对应集的成员的MLM参数信息。例如,可以利用来自训练节点102A和102B的MLM参数信息来更新MLM的一个分支,而利用来自训练节点102N的MLM参数信息来更新MLM的另一分支。在至少一个实施例中,训练节点102,客户端设备502和/或客户端可能不知道存在MLM的特定分支或任何分支和/或已经向其分配了MLM的分支的身份。

[0186] 在至少一个实施例中,审查聚合器1110和/或训练聚合器104可以至少部分地基于成员和共识之间的对应差异量来形成分支和/或将一个或更多个成员分配给MLM的分支。在至少一个实施例中,如本文所述,使用共识确定器122和/或差异确定器124来导出这种信息。在至少一个实施例中,审查聚合器1110和/或训练聚合器104可以至少部分地基于成员对训练的相应贡献来形成分支和/或将一个或更多个成员分配给MLM的分支。在至少一个实施例中,如本文所述,使用贡献确定器130来导出这种信息。在至少一个实施例中,审查聚合器1110和/或训练聚合器104可以至少部分地基于确定来自一个或更多个训练节点102的MLM参数信息(例如,分层参数的)之间的统计不兼容性来形成分支和/或将一个或更多个成员分配给MLM的分支。在一个实施例中,训练节点102可以被分组为统计上兼容的集合,其可以对应于MLM(例如,MLM 108)的分支。在至少一个实施例中,分配有公共分支的成员可以与该分支周期性地同步(例如,根据图4的流程图400)。

[0187] 在至少一个实施例中,审查聚合器1110和/或训练聚合器104可以至少部分地基于成员和共识之间的对应差异量来合并分支和/或将一个或更多个成员分配给MLM的不同分支。在至少一个实施例中,如本文所述,使用共识确定器122和/或差异确定器124来导出这种信息。在至少一个实施例中,审查聚合器1110和/或训练聚合器104可以至少部分地基于成员对训练的相应贡献来合并分支和/或将一个或更多个成员分配给MLM的不同分支。在至少一个实施例中,如本文所述,使用贡献确定器130来导出这种信息。在至少一个实施例中,审查聚合器1110和/或训练聚合器104可以至少部分地基于确定来自一个或更多个训练节点102的MLM参数信息(例如,分层参数的)之间的统计不兼容性来合并分支和/或将一个或更多个成员分配给MLM的不同分支。在至少一个实施例中,训练节点102可以被周期性地分组为统计上兼容的集合,并且MLM(例如,MLM 108)的分支可以被合并和/或分配到分支可以相应地更新。

[0188] 图12示出了至少一个实施例中的方法1200的流程图。在框B1202,方法1200包括比较神经网络权重信息的不同集合的准确度。在至少一个实施例中,参数审查器1102的参数比较器1108或审查聚合器1110可以比较与不同训练节点102,客户端设备502,边缘设备和/或客户关联的MLM参数信息1120的一个或更多个部分的准确度。

[0189] 在框B1204,方法1200包括提供关于神经网络权重信息的不同集合的比较信息。例如,客户端设备502和/或边缘设备生成比较信息,可以将比较信息发送到另一客户端设备502和/或边缘设备和/或一个或更多个聚合服务器504。其中一个或更多个聚合服务器504生成比较信息,比较信息可以提供给审查分析器1114进行分析,可以在一个或更多个聚合服务器504中进行分析。

[0190] 在框B1206处,方法1200包括从服务器接收神经网络权重信息。例如,一个或多个聚合服务器504可以将由参数确定器1116确定的神经网络权重信息提供给一个或多个训练节点102,客户端设备502和/或边缘设备(例如,以使MLM 106或其一个或多个部分同步)。

[0191] 图13示出了至少一个实施例中的方法1300的流程图。在框B1302处,方法1300包括接收MLM审查信息的集合。例如,一个或多个聚合服务器504可以接收MLM审查信息1118(例如,使用审查聚合器1110的接口管理器1112)。

[0192] 在框B1304处,方法1300包括至少部分地基于MLM审查信息来生成一个或多个神经网络。例如,一个或多个聚合服务器504可以使用MLM审查信息1118来生成一个或多个MLM 108(例如,使用审查分析器1114和参数确定器1116)。

[0193] 数据中心

[0194] 图14示出了示例数据中心1400,其中可以使用至少一个实施例。在至少一个实施例中,数据中心1400包括数据中心基础设施层1410、框架层1420、软件层1430和应用层1440。

[0195] 在至少一个实施例中,如图14所示,数据中心基础设施层1410可以包括资源协调器1412、分组的计算资源1414和节点计算资源(“节点C.R.s”)1416(1)-1416(N),其中“N”代表任何完整的正整数。在至少一个实施例中,节点C.R.1416(1)-1416(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),内存设备(例如动态只读存储器),存储器设备(例如固态硬盘或磁盘驱动器),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.1416(1)-1416(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。

[0196] 在至少一个实施例中,分组的计算资源1414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1414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、内存或存储器资源。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。

[0197] 在至少一个实施例中,资源协调器1412可以配置或以其他方式控制一个或多个节点C.R.1416(1)-1416(N)和/或分组的计算资源1414。在至少一个实施例中,资源协调器1412可以包括用于数据中心1400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。

[0198] 在至少一个实施例中,如图14所示,框架层1420包括作业调度器1432、配置管理器1434、资源管理器1436和分布式文件系统1438。在至少一个实施例中,框架层1420可以包括支持软件层1430的软件1432和/或应用程序层1440的一个或多个应用程序1442的框架。在至少一个实施例中,软件1432或应用程序1442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1420可以是但不限于一种免费和开源软件网络应用

框架,例如可以利用分布式文件系统1438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1432可以包括Spark驱动器,以促进对数据中心1400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1434可以能够配置不同的层,例如软件层1430和包括Spark和用于支持大规模数据处理的分布式文件系统1438的框架层1420。在至少一个实施例中,资源管理器1436能够管理映射到或分配用于支持分布式文件系统1438和作业调度器1432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1410上的分组的计算资源1414。在至少一个实施例中,资源管理器1436可以与资源协调器1412协调以管理这些映射的或分配的计算资源。

[0199] 在至少一个实施例中,包括在软件层1430中的软件1432可以包括由节点C.R.1416(1)-1416(N)的至少一部分,分组计算资源1414和/或框架层1420的分布式文件系统1438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。

[0200] 在至少一个实施例中,应用层1440中包括的应用程序1442可以包括由节点C.R.1416(1)-1416(N)的至少一部分、分组的计算资源1414和/或框架层1420的分布式文件系统1438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。

[0201] 在至少一个实施例中,配置管理器1434、资源管理器1436和资源协调器1412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1400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。

[0202] 在至少一个实施例中,数据中心1400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1400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1400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。

[0203] 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。

[0204] 推理和/或训练逻辑215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2A和/或图2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中,推理和/或训练逻辑215可以在图14的系统中使用,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。

[0205] 在至少一个实施例中,一个或更多个客户端设备502,一个或更多个聚合服务器504,训练节点102,训练聚合器104,一个或更多个参数审查器1102和/或一个或更多个检查聚合器1110和/或本文描述的一种或更多种方法可以在一个或更多个数据中心1100上实现。

[0206] 自主车辆

[0207] 图15A示出了根据至少一个实施例的自主车辆1500的示例。在至少一个实施例中,自主车辆1500(在本文中可替代地称为“车辆1500”)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1500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1500可以是飞机、机器人车辆或其他类型的车辆。

[0208] 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1500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1500可能能够进行条件自动化(等级3)、高度自动化(等级4)和/或全自动(等级5)。

[0209] 在至少一个实施例中,车辆1500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1500可以包括但不限于推进系统1550,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1550可以连接至车辆1500的传动系,其可以包括但不限于变速器,以使得能够对车辆1500进行推进。在至少一个实施例中,可以响应于从油门/加速器1552接收信号以控制推进系统1550。

[0210] 在至少一个实施例中,当推进系统1550正在运行时(例如,当车辆行驶时),转向系统1554(其可以包括但不限于方向盘)用于使车辆1500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1554可以从转向致动器1556接收信号。在至少一个实施例中,方向盘不用于全自动化(5级)功能。在至少一个实施例中,制动传感器系统1546可以用于响应于从制动致动器1548和/或制动传感器接收到的信号来操作车辆制动器。

[0211] 在至少一个实施例中,控制器1536可以包括但不限于一个或更多个片上系统(“SoC”) (图15A中未示出) 和/或图形处理单元(“GPU”) 向车辆1500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1536可以发送信号以通过制动致动器1548操作车辆制动,通过转向致动器1556操作转向系统1554,和/或通过节气门/加速器1552操作推进系统1550。控制器1536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1500。在至少一个实施例中,控制器1536可以包括用于自动驾驶功能的第一控制器1536,用于功能安全功能的第二控制器1536,用于人工智能功能(例如计算机视觉)的第三控制器1536,用于信息娱乐功能的第四控制器

1536,用于紧急情况下的冗余的第五控制器1536和/或其他控制器。在至少一个实施例中,单个控制器1536可以处理上述功能中的两个或更多个,两个或更多控制器1536可以处理单个功能和/或其任何组合。

[0212] 在至少一个实施例中,控制器1536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1500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全球导航卫星系统(“GNSS”)传感器1558(例如,全球定位系统传感器)、RADAR传感器1560、超声波传感器1562、LIDAR传感器1564、惯性测量单元(IMU)传感器1566(例如,加速度计、陀螺仪、磁罗盘、磁力计等)、麦克风1596、立体声相机1568、广角相机1570(例如鱼眼相机)、红外相机1572、环绕相机1574(例如,360度相机)、远程相机(图15A中未示出)、中程相机(图15A中未示出)、速度传感器1544(例如,用于测量车辆1500的速度)、振动传感器1542、转向传感器1540、制动传感器(例如,作为制动传感器系统1546的一部分)和/或其他传感器类型接收。

[0213] 在至少一个实施例中,一个或更多个控制器1536可以从车辆1500的仪表板1532接收输入(例如,由输入数据表示)并通过人机接口(“HMI”)显示器1534、声音信号器、扬声器和/或车辆1500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15A中未显示)、位置数据(例如,车辆1500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控制器1536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1534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)。

[0214] 在至少一个实施例中,车辆1500进一步包括网络接口1524,其可以使用无线天线1526和/或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1524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,无线天线1526还可以使用局域网(例如Bluetooth、Bluetooth Low Energy (LE)、Z-Wave、ZigBee等)和/或低功耗广域网(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。

[0215] 推理和/或训练逻辑215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2A和/或图2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中,推理和/或训练逻辑215可以在图15A的系统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。

[0216] 在至少一个实施例中,一个或更多个客户端设备502,一个或更多个聚合服务器504,训练节点102,训练聚合器104,一个或更多个参数审查器1102和/或一个或更多个检查聚合器1110和/或本文描述的一种或更多种方法可以至少部分地在一个或更多个自主车辆1500上实现。

[0217] 图15B示出了根据至少一个实施例的图15A的自主车辆1500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1500上的不同位置。

[0218] 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1500的组件和/或系统一起使用的数字相机。一种或更多种相机可以以汽车安全完整性等级(“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,例如每秒60帧(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的透明(“RCCC”)滤色器阵列、红色透明的透明蓝色(“RCCB”)滤色器阵列、红色蓝色绿色透明

[0219] (“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。

[0220] 在至少一个实施例中,一个或更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,一个或更多个相机(例如,所有相机)可以同时记录并提供图像数据(例如,视频)。

[0221] 在至少一个实施例中,可以将一个或更多个相机安装在安装组件中,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汽车内的反光(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。

[0222] 在至少一个实施例中,相机可以被集成到后视镜中。对于侧视相机,在至少一个实施例中,相机也可以集成在舱室的每个角落的四个支柱内。

[0223] 在至少一个实施例中,具有包括车辆1500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,以及在一个或更多个控制器1536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)。

[0224] 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1570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15B中仅示出了一个广角相机1570,但是,在其他实施例中,车辆1500上可以有任何数量(包括零)的广角相机1570。在至少一个实施例中,任何数量的远程相机1598(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1598也可以用于对象检测和分类以及基本对象跟踪。

[0225] 在至少一个实施例中,任何数量的立体声相机1568也可以包括在前向配置中。在

至少一个实施例中,一个或更多个立体声相机1568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1500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1568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镜头(左右分别一个)和一个图像处理芯片,其可以测量从车辆1500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1568。

[0226] 在至少一个实施例中,具有包括车辆1500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1574(例如,如图15B所示的四个环绕相机1574)可以定位在车辆1500上。在至少一个实施例中,环绕相机1574可以包括但不限于,广角相机1570、鱼目镜头、360度相机和/或类似物的任意数量和组合。例如,在至少一个实施例中,四个鱼目镜头相机可以位于车辆1500的前、后和侧面。在至少一个实施例中,车辆1500可以使用三个环绕相机1574(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。

[0227] 在至少一个实施例中,具有包括车辆1500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前向相机的相机(例如,远程相机1598和/或中程相机1576、立体相机1568、红外相机1572等),如本文所述。

[0228] 推理和/或训练逻辑215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。下面结合图2A和/或图2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中,推理和/或训练逻辑215可以用于图15B的系统中,以至少部分地基于使用本文所述的神经网络训练操作、神经网络功能和/或架构、或神经网络用例计算出的权重参数来推理或预测操作。

[0229] 在至少一个实施例中,由一个或更多个客户端设备502,一个或更多个聚合服务器504,训练节点102,训练聚合器104,一个或更多个参数审查器1102和/或一个或更多个检查聚合器1110使用的训练数据集302对应于由图15B的一个或更多个传感器生成的传感器数据。

[0230] 图15C示出了根据至少一个实施例的图15A的自主车辆1500的示例系统架构的框图。在至少一个实施例中,图15C中的车辆1500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1502连接。在至少一个实施例中,总线1502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总线可以是车辆1500内部的网络,用于帮助控制车辆1500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1502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标识符(例如,CAN ID)。在至少一个实施例中,可以读取总线1502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1502可以是符合ASIL B的CAN

总线。

[0231] 在至少一个实施例中,除了CAN之外或来自CAN,可使用FlexRay和/或以太网(Ethernet)。在至少一个实施例中,可以有任意数量的总线1502,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1502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1502可以用于碰撞避免功能,并且第二总线1502可以用于制动控制。在至少一个实施例中,每个总线1502可以与车辆1500的任何组件通信,并且两个或更多个总线1502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”)1504中的每一个,控制器1536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1500的传感器的输入),并且可以连接到公共总线,例如CAN总线。

[0232] 在至少一个实施例中,车辆1500可以包括一个或更多个控制器1536,诸如本文关于图15A所描述的那些。控制器1536可以用于多种功能。在至少一个实施例中,控制器1536可以耦合到车辆1500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1500、车辆1500的人工智能、车辆1500的信息娱乐等。

[0233] 在至少一个实施例中,车辆1500可以包括任何数量的SoC 1504。SoC1504中的每一个可以包括但不限于中央处理单元(“CPU”)1506、图形处理单元(“GPU”)1508、处理器1510、高速缓存1512、加速器1514、数据存储1516和/或其他未显示的组件和特征。在至少一个实施例中,SoC1504可以用于在各种平台和系统中控制车辆1500。例如,在至少一个实施例中,SoC 1504可以与高清晰度(“HD”)地图1522在系统(例如,车辆1500的系统)中组合,该高清晰度地图1522可以经由网络接口1524从一个或更多个服务器(图15C中未示出)获得地图刷新和/或更新。

[0234] 在至少一个实施例中,CPU 1506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,CPU 1506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,CPU 1506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,CPU 1506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,CPU 1506(例如,CCPLEX)可以配置成支持同时的集群操作,使得CPU 1506的集群的任何组合在任何给定的时间都可以是活跃的。

[0235] 在至少一个实施例中,一个或更多个CPU 1506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;和/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,CPU 1506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。

[0236] 在至少一个实施例中,GPU 1508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, GPU 1508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, GPU 1508, 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, GPU 1508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), 以及两个或更多个流式微处理器可以共享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, GPU 1508可以包括至少八个流式微处理器。在至少一个实施例中, GPU 1508可以使用计算应用程序编程接口(API)。在至少一个实施例中, GPU 1508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)。

[0237] 在至少一个实施例中, 一个或更多个GPU 1508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GPU1508。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 TENSOR CORE、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。

[0238] 在至少一个实施例中, 一个或更多个GPU 1508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”)。

[0239] 在至少一个实施例中, GPU 1508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GPU 1508直接访问CPU 1506页表。在至少一个实施例中, 当GPU 1508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CPU1506。作为响应, 在至少一个实施例中, CPU 1506可以在其页表中查找地址的虚拟-物理的映射并将转换传送回GPU 1508。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CPU 1506和GPU 1508存储器, 从而简化了GPU 1508的编程以及将应用程序移植到GPU 1508。

[0240] 在至少一个实施例中, GPU 1508可以包括任意数量的访问计数器, 其可以跟踪GPU 1508对其他处理器的存储器的访问频率。在至少一个实施例中, 访问计数器可以帮助确保将存储器页面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。

[0241] 在至少一个实施例中, 一个或更多个SoC 1504可以包括任何数量的高速缓存1512, 包括本文所述的那些。例如, 在至少一个实施例中, 高速缓存1512可以包括可用于CPU 1506和GPU 1508(例如, 连接两个CPU 1506和GPU 1508)的三级(“L3”)高速缓存。在至少一个实施例中, 高速缓存1512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一致性协议(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。

[0242] 在至少一个实施例中,一个或更多个SoC 1504可以包括一个或更多个加速器1514(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,SoC 1504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GPU 1508并且卸载GPU 1508的一些任务(例如,释放GPU 1508的更多周期以执行其他任务)。在至少一个实施例中,加速器1514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。

[0243] 在至少一个实施例中,加速器1514(例如,硬件加速集群)可以包括深度学习加速器(“DLA”)。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DLA。在至少一个实施例中,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,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,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1596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。

[0244] 在至少一个实施例中,DLA可以执行GPU 1508的任何功能,并且通过使用推理加速器,例如,设计者可以将DLA或GPU 1508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DLA上,并将其他功能留给GPU 1508和/或其他加速器1514。

[0245] 在至少一个实施例中,加速器1514(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,PVA可以设计和配置为加速用于高级驾驶员辅助系统(“ADAS”)1538、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。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,每个PVA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。

[0246] 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,可以使用一个或更多个集成电路设备、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。

[0247] 在至少一个实施例中,DMA可以使PVA的组件能够独立于CPU 1506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更多个寻址的维度,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。

[0248] 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。

[0249] 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同一图像上同时执行不同的计算机视觉算法,或者甚至在顺序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。

[0250] 在至少一个实施例中,加速器1514(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”),用于为加速器1514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,其包括例如但不限于八个现场可配置的存储器块,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)。

[0251] 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。

[0252] 在至少一个实施例中,一个或更多个SoC 1504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。

[0253] 在至少一个实施例中,加速器1514(例如,硬件加速器集群)具有用于自动驾驶的广泛用途。在至少一个实施例中,PVA可以是可编程视觉加速器,其可以用于ADAS和自动驾驶汽车中的关键处理阶段。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1500)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。

[0254] 例如,在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。

[0255] 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。

[0256] 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,LIDAR传感器1564或RADAR传感器1560)等获得的对象的车辆1500方向、距离、3D位置估计相关的IMU传感器1566的输出。

[0257] 在至少一个实施例中,一个或更多个SoC 1504可以包括数据存储装置1516(例如,存储器)。在至少一个实施例中,数据存储装置1516可以是SoC 1504的片上存储器,其可以存储要在GPU 1508和/或DLA上执行的神经网络。在至少一个实施例中,数据存储装置1516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,数据存储装置1512可以包括L2或L3高速缓存。

[0258] 在至少一个实施例中,一个或更多个SoC 1504可以包括任何数量的处理器1510(例如,嵌入式处理器)。处理器1510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SoC 1504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,SoC 1504热和温度传感器管理和/或SoC 1504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SoC 1504可以使用环形振荡器来检测CPU 1506,GPU 1508和/或加速器1514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SoC 1504置于较低功耗状态和/或将车辆1500置于司机的安全停车图案(例如,使车辆1500安全停车)。

[0259] 在至少一个实施例中,一个或更多个处理器1510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。

[0260] 在至少一个实施例中,处理器1510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。

[0261] 在至少一个实施例中,处理器1510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,处理器1510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,处理器1510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。

[0262] 在至少一个实施例中,处理器1510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广角相机1570、环绕相机1574和/或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 1504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。

[0263] 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。

[0264] 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界面合成,并且不需要GPU 1508来连续渲染新表面。在至少一个实施例中,当对GPU 1508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GPU 1508以改善性能和响应性。

[0265] 在至少一个实施例中,一个或更多个SoC 1504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 1504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。

[0266] 在至少一个实施例中,一个或更多个SoC 1504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。SoC 1504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的)相机、传感器(例如,LIDAR传感器1564,RADAR传感器1560等,其可以通过以太网连接)的数据,来自总线1502的数据(例如,车辆1500的速度、方向盘位置等),来自GNSS传感器1558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 1504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CPU 1506摆脱常规数据管理任务。

[0267] 在至少一个实施例中,SoC 1504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,SoC1504可以比其他系统更快、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,加速器1514当与CPU 1506、GPU 1508以及数据存储装置1516结合时,可以提供用于3-5级自主车辆的快速、有效的平台。

[0268] 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编程语言)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应用和实际3-5级自主车辆中。

[0269] 本文所述的实施例允许同时和/或顺序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, GPU 1520)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。

[0270] 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)”连通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(flashings lights indicate icy conditions)”,其通知车辆的路径规划软件(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GPU 1508上。

[0271] 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1500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,SoC 1504提供防止盗窃和/或劫车的保障。

[0272] 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1596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,SoC 1504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GNSS传感器1558所标识。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,就可以在超声波传感器1562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。

[0273] 在至少一个实施例中,车辆1500可以包括CPU 1518(例如,离散CPU或dCPU),其可以经由高速互连(例如PCIe)耦合到SoC 1504。在至少一个实施例中,CPU 1518可以包括X86处理器,例如CPU 1518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SoC 1504之间潜在的仲裁不一致的结果,和/或监控控制器1536的状态和健康和/或片上信息系统(“信息SoC”)1530。

[0274] 在至少一个实施例中,车辆1500可以包括GPU 1520(例如,离散GPU或dGPU),其可以经由高速互连(例如NVIDIA的NVLINK)耦合到SoC 1504。在至少一个实施例中,GPU 1520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1500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。

[0275] 在至少一个实施例中,车辆1500可以进一步包括网络接口1524,其可以包括但不限于无线天线1526(例如,用于不同通信协议的一个或多个无线天线1526,诸如蜂窝天线、蓝牙天线等)。在至少一个实施例中,网络接口1524可以用于使通过互联网与云(例如,服务器和/或其他网络设备)、其他车辆和/或计算设备(例如乘客的客户端设备)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150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1500提供关于车辆

1500附近的车辆的信息(例如,车辆1500前面、侧面和/或后面的车辆)。在至少一个实施例中,前述功能可以是车辆1500的协作自适应巡航控制功能的一部分。

[0276] 在至少一个实施例中,网络接口1524可以包括SoC,其提供调制和解调功能并使控制器1536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1524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,可以通过各种过程 and/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。

[0277] 在至少一个实施例中,车辆1500可以进一步包括数据存储器1528,其可以包括但不限于片外(例如,SoC 1504)存储器。在至少一个实施例中,数据存储器1528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。

[0278] 在至少一个实施例中,车辆1500可以进一步包括GNSS传感器1558(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1558,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。

[0279] 在至少一个实施例中,车辆1500可以进一步包括一个或更多个RADAR传感器1560。RADAR传感器1560可以由车辆1500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功能安全等级可以是ASIL B。RADAR传感器1560可以使用CAN和/或总线1502(例如,以传输由RADAR传感器1560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1560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1560是脉冲多普勒RADAR传感器。

[0280] 在至少一个实施例中,RADAR传感器1560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,RADAR传感器1560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1538用于紧急制动辅助和向前碰撞警告。包括在远程RADAR系统中的传感器1560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1500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1500的车道的车辆。

[0281] 在至少一个实施例中,作为示例,中程RADAR系统可包括高达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1560。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部和附近的盲点。在至少一个实施例中,短程RADAR系统可以在ADAS系统

1538中用于盲点检测和/或车道改变辅助。

[0282] 在至少一个实施例中,车辆1500可以进一步包括一个或更多个超声传感器1562。可以定位在车辆1500的前、后和/或侧面的超声传感器1562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1562,并且可以将不同的超声传感器1562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1562可以在ASIL B的功能安全等级下操作。

[0283] 在至少一个实施例中,车辆1500可以包括一个或更多个LIDAR传感器1564。LIDAR传感器1564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,LIDAR传感器1564可以是功能安全等级ASIL B。在至少一个实施例中,车辆1500可以包括可以使用以太网(网的多个(例如,两个、四个、六个等)LIDAR传感器1564(例如,将数据提供给千兆以太网交换机)。

[0284] 在至少一个实施例中,LIDAR传感器1564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LIDAR传感器1564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1564。在这样的实施例中,可以将LIDAR传感器1564实现为可以嵌入到车辆1500的前、后、侧面和/或拐角中的小型设备。在至少一个实施例中,LIDAR传感器1564,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LIDAR传感器1564配置为用于45度至135度之间的水平视野。

[0285] 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1500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1500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1500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3D测距点云和共同登记的强度数据形式的反射激光。

[0286] 在至少一个实施例中,车辆还可包括IMU传感器1566。在至少一个实施例中,IMU传感器1566可位于车辆1500的后轴中心,在至少一个实施例中。在至少一个实施例中,IMU传感器1566可以包括,例如但不限于,加速度计、磁力计、陀螺仪、磁罗经和/或其他传感器类型。在至少一个实施例中,例如在六轴应用中,IMU传感器1566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,例如在九轴应用中,IMU传感器1566可以包括但不限于加速度计、陀螺仪和磁力计。

[0287] 在至少一个实施例中,IMU传感器1566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,IMU传感器1566可使车辆1500估算航向而无需来自磁传感器通过直接观察和关联从GPS到IMU传感器1566的速度变化来实现的输入。在至少一个实施例中,IMU传感器1566和GNSS传感器1558可以组合

在单个集成单元中。

[0288] 在至少一个实施例中,车辆1500可以包括放置在车辆1500内和/或周围的麦克风1596。在至少一个实施例中,此外,麦克风1596可以用于紧急车辆检测和识别。

[0289] 在至少一个实施例中,车辆1500可以进一步包括任何数量的相机类型,包括立体相机1568、广角相机1570、红外相机1572、环绕相机1574、远程相机1598、中程相机1576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1500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1500。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1500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1500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15A和图15B更详细地描述了每个相机。

[0290] 在至少一个实施例中,车辆1500可以进一步包括振动传感器1542。振动传感器1542可以测量车辆1500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1542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)。

[0291] 在至少一个实施例中,车辆1500可以包括ADAS系统1538。在一些示例中,ADAS系统1538可以包括但不限于SoC。在至少一个实施例中,ADAS系统1538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。

[0292] 在至少一个实施例中,ACC系统可以使用RADAR传感器1560、LIDAR传感器1564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1500的车辆的距离,并自动调节车辆1500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1500改变车道。在至少一个实施例中,横向ACC与其他ADAS应用有关,例如LC和CW。

[0293] 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1524和/或无线天线1526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1500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1500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。

[0294] 在至少一个实施例中,FCW系统被设计成警告驾驶员危险,以便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RADAR传感器1560,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。

[0295] 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的前向相机和/或RADAR传感器1560。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。

[0296] 在至少一个实施例中,当车辆1500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,其电耦合至诸如显示器、扬声器和/或振动组件。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1500开始离开车道,则LKA系统提供转向输入或制动以校正车辆1500。

[0297] 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的朝后侧的相机和/或RADAR传感器1560,其电耦合到驾驶员反馈,例如显示器、扬声器和/或振动组件。

[0298] 在至少一个实施例中,当在车辆1500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更多个面向后方的RADAR传感器1560,其耦合到专用处理器、DSP、FPGA和/或ASIC,其电耦合到驾驶员反馈,诸如显示器、扬声器和/或振动组件。

[0299] 在至少一个实施例中,其他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,因为其他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行动。在至少一个实施例中,在结果冲突的情况下,车辆1500本身决定是否听从主计算机或副计算机(例如,第一控制器1536或第二控制器1536)的结果。例如,在至少一个实施例中,ADAS系统1538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1538的输出提供给监控MCU。在至少一个实施例中,如果来自自主计算机和辅助计算机的输出冲突,则监督MCU决定如何协调冲突以确保安全操作。

[0300] 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。

[0301] 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SoC 1504的组件。

[0302] 在至少一个实施例中,ADAS系统1538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。

[0303] 在至少一个实施例中,可以将ADAS系统1538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1538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。

[0304] 在至少一个实施例中,车辆1500可以进一步包括信息娱乐SoC 1530(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1530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 1530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1500。例如,信息娱乐SoC 1530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1534、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 1530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1538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。

[0305] 在至少一个实施例中,信息娱乐SoC 1530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 1530可以通过总线1502(例如,CAN总线、以太网等)与车辆1500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 1530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1536(例如,车辆1500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 1530可以使车辆1500进入司机到安全停止模式,如本文所述。

[0306] 在至少一个实施例中,车辆1500可以进一步包括仪表板1532(例如,数字仪表板、电子仪表板、数字仪表操纵板等)。仪表板1532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1532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、安全带警告灯、驻车制动警告灯、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 1530和仪表板1532之间显示和/或共享。在至少一个实施例中,仪表板1532可以被包括作为信息娱乐SoC1530的一部分,反之亦然。

[0307] 推理和/或训练逻辑215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2A和/或2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中,推理和/或训练逻辑215可以在系统图15C中使用,以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。

[0308] 在至少一个实施例中,可以至少部分地在图15C的SoC(B)和/或SoC(A)中的一个或更多个上实现本文描述的客户端设备502,一个或更多个聚合服务器504,训练节点102,训练聚合器104,一个或更多个参数审查器1102和/或一个或更多个审查聚合器1110和/或方法中的一个或更多个。

[0309] 图15D是根据至少一个实施例的在基于云的服务器与图15A的自主车辆1500之间进行通信的系统1576的图。在至少一个实施例中,系统1576可以包括但不限于服务器1578、网络1590以及任何数量和类型的车辆,包括车辆1500。在至少一个实施例中,服务器1578可以包括但不限于,多个GPU 1584(A)-1584(H)(在本文中统称为GPU 1584)、PCIe交换机1582(A)-1582(D)(在本文中统称为PCIe交换机1582),和/或CPU 1580(A)-1580(B)(在本文中统称为CPU 1580)、GPU 1584、CPU 1580和PCIe开关1582可以与高速连接线互连,例如但不限于,由NVIDIA开发的NVLink接口1588和/或PCIe连接1586。在至少一个实施例中,GPU 1584通过NVLink和/或NVSwitchSoC连接,GPU 1584和PCIe开关1582通过PCIe互连连接。在至少一个实施例中,尽管示出了八个GPU 1584、两个CPU 1580和四个PCIe开关1582,但这并不旨在进行限制。在至少一个实施例中,服务器1578中的每一个可以包括但不限于任意数量的GPU 1584、CPU 1580和/或PCIe开关1582的任何组合。例如,在至少一个实施例中,服务器1578可各自包括八个、十六个、三十二个和/或更多个GPU 1584。

[0310] 在至少一个实施例中,服务器1578可以通过网络1590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,服务器1578可以通过网络1590并且向车辆传输神经网络1592、更新的神经网络1592和/或地图信息1594,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1594的更新可以包括但不限于对HD地图1522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1592、更新的神经网络1592和/或地图信息1594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,使用服务器1578和/或其他服务器)。

[0311] 在至少一个实施例中,服务器1578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,通过网络1590传输到车辆,和/或机器学习模型可以被服务器1578使用以远程监控车辆。

[0312] 在至少一个实施例中,服务器1578可以从车辆接收数据并且将数据应用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,服务器1578可以包括由GPU 1584供电的深度学习超级计算机和/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,服务器1578可以包括使用CPU供电的数据中心的深度学习基础设施。

[0313] 在至少一个实施例中,服务器1578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1500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1500接收周期性更新,例如车辆1500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1500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1500中的AI正在发生故障,则服务器1578可以将信号发送到车辆1500,以指示车辆1500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。

[0314] 在至少一个实施例中,服务器1578可以包括GPU 1584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215用于执行一个或多个实施例。下面结合图2A和/或2B提供关于硬件结构215的细节。

[0315] 计算机系统

[0316] 图16是示出在至少一个实施例中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1600,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1600可以包括但不限于组件,例如处理器1602,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1600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1600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。

[0317] 实施例可以用在其他设备中,例如手持设备和嵌入式应用。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更多个指令的任何其他系统。

[0318] 在至少一个实施例中,计算机系统1600可包括但不限于处理器1602,该处理器1602可包括但不限于一个或更多个执行单元1608,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16A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16A可以是多处理器系统。在至少一个实施例中,处理器1602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1602可以耦合到处理器总线1610,该处理器总线1610可以在处理器1602与计算机系统1600中的其他组件之间传输数据信号。

[0319] 在至少一个实施例中,处理器1602可以包括但不限于1级(“L1”)内部高速缓存存储器(“cache”)1604。在至少一个实施例中,处理器1602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1602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1606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。

[0320] 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1608,其也位于处理器1602中。处理器1602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1608可以包括用于处理封装指令集1609的逻辑。在至少一个实施例中,通过将封装指令集1609包括在通用处理器1602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1602中的封装数据来执行许多多媒体应用程序使用的操作。在一个或更多个实施例中,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更多个操作。

[0321] 在至少一个实施例中,执行单元1608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1600可以包括但不限于存储器1620。在至少一个实施例中,存储器1620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设备。在至少一个实施例中,存储器1620可以存储由处理器1602可以执行的由数据信号表示的指令1619和/或数据1621。

[0322] 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1610和存储器1620。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”)1616,

并且处理器1602可以经由处理器总线1610与MCH 1616通信。在至少一个实施例中,MCH 1616可以提供到存储器1620的高带宽存储器路径1618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 1616可以在处理器1602、存储器1620和计算机系统1600中的其他组件之间启动数据信号,并且在处理器总线1610、存储器1620和系统I/O 1622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 1616可以通过高带宽存储器路径1618耦合到存储器1620,并且图形/视频卡1612可以通过加速图形端口 (Accelerated Graphics Port) (“AGP”) 互连1614耦合到MCH 1616。

[0323] 在至少一个实施例中,计算机系统1600可以使用系统I/O 1622作为专有集线器接口总线来将MCH 1616耦合到I/O控制器集线器 (“ICH”) 1630。在至少一个实施例中,ICH 1630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1620、芯片组和处理器1602的高速I/O总线。示例可以包括但不限于音频控制器1629、固件集线器 (“Flash BIOS”) 1628、无线收发器1626、数据存储装置1624、包含用户输入的传统I/O控制器1623和键盘接口、串行扩展端口1627 (例如通用串行总线 (USB)) 和网络控制器1634。数据存储装置1624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。

[0324] 在至少一个实施例中,图16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16可以示出示例性片上系统 (“SoC”)。在至少一个实施例中,图中示出的设备可以与专有互连、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,计算机系统1600的一个或更多个组件使用计算快速链路 (CXL) 互连来互连。

[0325] 推理和/或训练逻辑215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。下面结合图2A和/或2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中,推理和/或训练逻辑215可以在系统图16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。

[0326] 在至少一个实施例中,可以至少部分地在图16的系统1600上实现本文描述的客户端设备502,一个或更多个聚合服务器504,训练节点102,训练聚合器104,一个或更多个参数审查器1102和/或一个或更多个审查聚合器1110和/或方法中的一个或更多个。

[0327] 图17是示出根据至少一个实施例的用于利用处理器1710的电子设备1700的框图。在至少一个实施例中,电子设备1700可以是,例如但不限于,笔记本电脑、塔式服务器、机架服务器、刀片服务器、膝上型计算机、台式机、平板电脑、移动设备、电话、嵌入式计算机或任何其他合适的电子设备。

[0328] 在至少一个实施例中,系统1700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1710。在至少一个实施例中,处理器1710使用总线或接口耦合,诸如I²C总线、系统管理总线 (“SMBus”)、低引脚数 (LPC) 总线、串行外围接口 (“SPI”)、高清音频 (“HDA”) 总线、串行高级技术附件 (“SATA”) 总线、通用串行总线 (“USB”) (1、2、3版) 或通用异步接收器/发送器 (“UART”) 总线。在至少一个实施例中,图17示出了系统,该系统包括互连的硬件设备或“芯片”,而在其他实施例中,图17可以示出示例性片上系统 (“SoC”)。在至少一个实施例中,图17中所示的设备可以与专有互连线、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,图17的一个或更多个组件使用计算快速链路(CXL)互连线来互连。

[0329] 在至少一个实施例中,图17可以包括显示器1724、触摸屏1725、触摸板1730、近场通信单元(“NFC”)1745、传感器集线器1740、热传感器1746、快速芯片组(“EC”)1735、可信平台模块(“TPM”)1738、BIOS/固件/闪存(“BIOS,FW Flash”)1722、DSP 1760、驱动器“SSD或HDD”)1720(例如固态硬盘(“SSD”)或硬盘驱动器(“HDD”))、无线局域网单元(“WLAN”)1750、蓝牙单元1752、无线广域网单元(“WWAN”)1756、全球定位系统(GPS)1755、相机(“USB 3.0相机”)1754(例如USB 3.0相机)和/或实现在例如LPDDR3标准低功耗双倍数据速率(“LPDDR”)存储单元(“LPDDR3”)1715。这些组件可以各自以任何合适的方式实现。

[0330] 在至少一个实施例中,其他组件可以通过以上讨论的组件通信地耦合到处理器1710。在至少一个实施例中,加速度计1741、环境光传感器(“ALS”)1742、罗盘1743和陀螺仪1744可以通信地耦合到传感器集线器1740。在至少一个实施例中,热传感器1739、风扇1737、键盘1746和触摸板1730可以通信地耦合到EC1735。在至少一个实施例中,扬声器1763、耳机1764和麦克风(“mic”)1765可以通信地耦合到音频单元(“音频编解码器和D类放大器”)1764,其又可以通信地耦合到DSP1760。在至少一个实施例中,音频单元1764可以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”)1757可以通信地耦合到WWAN单元1756。在至少一个实施例中,组件(诸如WLAN单元1750和蓝牙单元1752以及WWAN单元1756)可以被实现为下一代形式因素(NGFF)。

[0331] 推理和/或训练逻辑215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2A和/或2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中,推理和/或训练逻辑215可以在系统图17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。

[0332] 在至少一个实施例中,可以至少部分地在图17的一个或更多个电子设备1700上实现本文描述的客户端设备502,一个或更多个聚合服务器504,训练节点102,训练聚合器104,一个或更多个参数审查器1102和/或一个或更多个审查聚合器1110和/或方法中的一个或更多个。

[0333] 图18示出了根据至少一个实施例的计算机系统1800。在至少一个实施例中,计算机系统1800配置为实现贯穿本公开描述的各种过程和方法。

[0334] 在至少一个实施例中,计算机系统1800包括但不限于至少一个中央处理单元(“CPU”)1802,该中央处理单元(“CPU”)1802连接到使用任何合适协议实现的通信总线1810,诸如PCI(“外围设备互连”)、外围组件互连Express(“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1800包括但不限于主存储器1804和控制逻辑(例如,实现为硬件、软件或其组合),并且数据可以采取随机存取存储器(“RAM”)的形式存储在主存储器1804中。在至少一个实施例中,网络接口子系统(“网络接口”)1822提供到其他计算设备和网络的接口,用于从计算机系统1800接收数据并将数据传输到其他系统。

[0335] 在至少一个实施例中,计算机系统1800在至少一个实施例中包括但不限于输入设备1808、并行处理系统1812和显示设备1806,它们可以使用常规的阴极射线管(“CRT”)、液

晶显示器 (“LCD”)、发光二极管 (“LED”)、等离子显示器或其他合适的显示技术实现。在至少一个实施例中,从输入设备1808(诸如键盘、鼠标、触摸板、麦克风等)接收用户输入。

[0336] 推理和/或训练逻辑215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2A和/或2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中,推理和/或训练逻辑215可以在系统图18中使用,以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进行推理或预测操作。

[0337] 在至少一个实施例中,可以至少部分地在图18的一个或更多个计算机系统1800上实现本文描述的客户端设备502,一个或更多个聚合服务器504,训练节点102,训练聚合器104,一个或更多个参数审查器1102和/或一个或更多个审查聚合器1110和/或方法中的一个或更多个。

[0338] 图19示出了根据至少一个实施例的计算机系统1900。在至少一个实施例中,计算机系统1900包括但不限于计算机1910和USB盘1920。在至少一个实施例中,计算机1910可以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1910包括但不限于服务器、云实例、膝上型计算机和台式计算机。

[0339] 在至少一个实施例中,USB盘1920包括但不限于处理单元1930、USB接口1940和USB接口逻辑1950。在至少一个实施例中,处理单元1930可以是任何指令执行系统、装置或能够执行指令的设备。在至少一个实施例中,处理单元1930可以包括但不限于任何数量和类型的处理核心(未示出)。在至少一个实施例中,处理核心1930包括专用集成电路 (“ASIC”),该专用集成电路被优化为执行与机器学习相关联的任何数量和类型的操作。例如,在至少一个实施例中,处理核心1930是张量处理单元 (“TPC”),其被优化以执行机器学习推理操作。在至少一个实施例中,处理核心1930是视觉处理单元 (“VPU”),其被优化以执行机器视觉和机器学习推理操作。

[0340] 在至少一个实施例中,USB接口1940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1940是用于数据和电源的USB 3.0Type-C插座。在至少一个实施例中,USB接口1940是USB 3.0Type-A连接器。在至少一个实施例中,USB接口逻辑1950可以包括使处理单元1930能够经由USB连接器1940与设备(例如计算机1910)相连接的任何数量和类型的逻辑。

[0341] 推理和/或训练逻辑215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2A和/或2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中,推理和/或训练逻辑215可以在系统图19中使用,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。

[0342] 在至少一个实施例中,可以至少部分地在图19的一个或更多个计算机系统1900上实现本文描述的客户端设备502,一个或更多个聚合服务器504,训练节点102,训练聚合器104,一个或更多个参数审查器1102和/或一个或更多个审查聚合器1110和/或方法中的一个或更多个。

[0343] 图20A示出了示例性架构,其中多个GPU 2010-2013通过高速链路2040-2043(例如,总线、点对点互连等)通信地耦合到多个多核心处理器2005-2006。在一个实施例中,高

速链路2040-2043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PCIe4.0或5.0和NVLink 2.0。

[0344] 另外,在一个实施例中,GPU 2010-2013中的两个或更多个通过高速链路2029-2030互连,可以使用与用于高速链路2040-2043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2005-2006可以通过高速链路2028连接,该高速链路可以是以20GB/s、

[0345] 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,图20A中所示的各种系统组件之间的所有通信可以使用相同的协议/链路(例如,通过共同的互连结构)来完成。

[0346] 在一个实施例中,每个多核心处理器2005-2006分别经由存储器互连2026-2027通信地耦合至处理器存储器2001-2002,并且每个GPU

[0347] 2010-2013分别通过GPU存储器互连2050-2053通信地耦合至GPU存储器2020-2023。存储器互连2026-2027和2050-2053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2001-2002和GPU存储器2020-2023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3DXPoint或Nano-Ram。在一个实施例中,处理器存储器2001-2002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)层次结构)。

[0348] 如下所述,尽管各种处理器2005-2006和GPU 2010-2013可以分别物理耦合到特定的存储器2001-2002、2420-2423,可以实现统一的存储器架构,其中相同的虚拟系统地址空间(也称为“有效地址”空间)分布在各种物理存储器之间。例如,处理器存储器2001-2002每个可以包括64GB的系统存储器地址空间,而GPU存储器2020-2023每个可以包括32GB的系统存储器地址空间(在此示例中总共为256GB可寻址存储器)。

[0349] 图20B示出了根据一个示例性实施例的用于多核心处理器2007和图形加速模块2046之间的互连的附加细节。图形加速模块2046可以包括集成在通过高速链路2040耦合到处理器2007的线卡上的一个或更多个GPU芯片。可替代地,图形加速模块2046可以与处理器2007集成在同一封装或芯片上。

[0350] 在至少一个实施例中,所示的处理器2007包括多个核心2060A-2060D,每个核心具有转换后备缓冲区2061A-2061D和一个或更多个高速缓存2062A-2062D。在至少一个实施例中,核心2060A-2060D可以包括用于执行未示出的指令和处理数据的各种其他组件。高速缓存2062A-2062D可以包括1级(L1)和2级(L2)高速缓存。此外,一个或更多个共享高速缓存2056可以被包括在高速缓存2062A-2062D中并且由一组核心2060A-2060D共享。例如,处理器2007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。在这一实施例中,两个相邻核心共享一个或更多个L2和L3高速缓存。处理器2007和图形加速模块2046与系统存储器2014连接,其可以包括图20A的处理器存储器2001-2002。

[0351] 经由一致性总线2064上的核心间通信,为存储在各种高速缓存2062A-2062D、2056和系统存储器2014中的数据和指令保持一致性。例如,每个高速缓存可具有与其关联的高速缓存一致性逻辑/电路以响应于所检测到的对特定高速缓存线的读取或写入通过一致性

总线2064进行通信。在一种实现中,在一致性总线2064上实现了高速缓存监听协议以监听高速缓存访问。

[0352] 在一个实施例中,代理电路2025将图形加速模块2046通信地耦合到一致性总线2064,从而允许图形加速模块2046作为核心2060A-2060D的对等方参与高速缓存一致性协议。特别地,接口2035通过高速链路2040(例如,PCIe总线、NVLink等)提供到代理电路2025的连接,并且接口2037将图形加速模块2046连接到链路2040。

[0353] 在一种实现中,加速器集成电路2036代表图形加速模块2046的多个图形处理引擎2031、2032、N提供高速缓存管理、存储器访问、环境管理和中断管理服务。图形处理引擎2031、2032、N可以各自包括单独的图形处理单元(GPU)。可选地,图形处理引擎2031、2032、N可包括GPU内的不同类型的图形处理引擎,例如图形执行单元、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2046可以是具有多个图形处理引擎2031-2032、N或图形处理引擎2031-2032的GPU,N可以是集成在通用封装、线卡或芯片上的各个GPU。

[0354] 在一个实施例中,加速器集成电路2036包括存储器管理单元(MMU)2039,用于执行各种存储器管理功能,例如虚拟到物理的存储器转换(也称为有效到实际的存储器转换)和用于访问系统存储器2014的存储器访问协议。MMU2039还可以包括转换后备缓冲区(TLB)(未示出),用于将虚拟/有效地址缓存为物理/实际地址转换。在一种实现中,高速缓存2038存储命令和数据,以供图形处理引擎2031-2032、N有效访问。在一个实施例中,存储在高速缓存2038和图形存储器2033-2034、M中的数据与核心高速缓存2062A-2062D、2056和系统存储器2014保持一致。如上所述,这可以经由代理电路2025代表高速缓存2038和存储器2033-2034、M来完成(例如,将与处理器缓存2062A-2062D、2056上的高速缓存行的修改/访问有关的更新发送到高速缓存2038,并从高速缓存2038接收更新)。

[0355] 一组寄存器2045存储由图形处理引擎2031-2032、N执行的线程的环境数据,并且环境管理电路2048管理线程环境。例如,环境管理电路2048可以执行保存和还原操作以在环境切换期间保存和还原各种线程的环境(例如,其中保存第一线程并且存储第二线程,使得第二线程可以由图形处理引擎执行)。例如,在环境切换器上,环境管理电路2048可以将当前寄存器值存储到存储器中的指定区域(例如,由环境指针标识)。然后,当返回环境时,它可以恢复寄存器值。在一个实施例中,中断管理电路2047接收并处理从系统设备接收的中断。

[0356] 在一个实施例中,MMU 2039将来自图形处理引擎2031的虚拟/有效地址转换为系统存储器2014中的实际/物理地址。加速器集成电路2036的一个实施例支持多个(例如4、8、16)图形加速器模块2046和/或其他加速器设备。图形加速器模块2046可以专用于在处理器2007上执行的单个应用,或者可以在多个应用之间共享。在一个实施例中,提出了一种虚拟化的图形执行环境,其中图形处理引擎2031-2032、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,可以基于与虚拟机和/或应用程序相关联的处理要求和优先级,将资源细分为“片”,将其分配给不同的虚拟机和/或应用程序。

[0357] 在至少一个实施例中,加速器集成电路2036充当用于图形加速模块2046的系统的桥梁,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2036可以为主机处理器提供虚拟化设施,以管理图形处理引擎2031-2032、N的虚拟化,中断和存储器管理。

[0358] 因为图形处理引擎2031-2032、N的硬件资源被显式映射到主机处理器2007看到的实际地址空间,所以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2036的一个功能是物理分隔图形处理引擎2031-2032、N,使得它们在系统中表现为独立的单元。

[0359] 在至少一个实施例中,一个或多个图形存储器2033-2034、M分别耦合到图形处理引擎2031-2032、N中的每一个。图形存储器2033-2034、M存储由图形处理引擎2031-2032、N中的每一个处理的指令和数据。图形存储器2033-2034、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。

[0360] 在一个实施例中,为了减少链路2040上的数据流量,使用偏置技术以确存储存在图形存储器2033-2034、M中的数据将是图形处理引擎2031-2032、N最频繁使用的数据,并且最好不被核心2060A-2060D使用(至少不经常使用)。类似地,偏置机制尝试将核心(最好不是图形处理引擎2031-2032、N)所需的数据保留在核心和系统存储器2014的高速缓存2062A-2062D、2056中。

[0361] 图20C示出了另一示例性实施例,其中加速器集成电路2036被集成在处理器2007内。在至少该实施例中,图形处理引擎2031-2032、N经由接口2037和接口2035通过高速链路2040直接通信到加速器集成电路2036(其中,再次可以使用任何形式的总线或接口协议)。加速器集成电路2036可以执行与关于图20B描述的相同的操作,但是鉴于其紧密靠近一致性总线2064和高速缓存2062A-2062D、2056,可能具有较高的吞吐量。至少一个实施例支持不同的编程模型,包括专用过程编程模型(无图形加速模块虚拟化)和共享编程模型(具有虚拟化),其可以包括由加速器集成电路2036控制的编程模型和由图形加速模块2046控制的编程模型。

[0362] 在至少一个实施例中,图形处理引擎2031-2032、N专用于单个操作系统下的单个应用程序或过程。在至少一个实施例中,单个应用程序可以将其他应用请求集中到图形处理引擎2031-2032、N,从而在VM/分区内提供虚拟化。

[0363] 在至少一个实施例中,图形处理引擎2031-2032、N可以由多个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2031-2032、N以允许被每个操作系统访问。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2031-2032、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2031-2032、N以提供对每个进程或应用程序的访问。

[0364] 在至少一个实施例中,图形加速模块2046或单独的图形处理引擎2031-2032、N使用过程句柄来选择进程元素。在至少一个实施例中,处理元件被存储在系统存储器2014中,并且可使用本文所述的有效地址到实地址转换技术来寻址。在至少一个实施例中,过程句柄可以是在向图形处理引擎2031-2032、N注册其环境时向主机过程提供的特定于实现的值(即,调用系统软件以将进程元素添加到进程元素链表)。在至少一个实施例中,过程句柄的低16位可以是进程元素链表中进程元素的偏移量。

[0365] 图20D示出了示例性加速器集成切片2090。如本文所使用的,“片”包括加速器集成电路2036的处理资源的指定部分。系统存储器2014内的应用有效地址空间2082存储进程元素2083。在一个实施例中,响应于来自处理器2007上执行的应用程序2080的GPU调用2481而

存储进程元素2083。进程元素2083包含对应应用程序2080的处理状态。包含在处理元素2083中的工作描述符(WD) 2084可以是应用程序请求的单个作业或可能包含指向作业队列的指针。在至少一个实施例中,WD 2084是指向应用程序地址空间2082中的作业请求队列的指针。

[0366] 图形加速模块2046和/或各个图形处理引擎2031-2032、N可以由系统中的全部或部分进程共享。在至少一个实施例中,可以包括用于建立处理状态并将WD 2084发送到图形加速模块2046以在虚拟化环境中开始作业的基础设施。

[0367] 在至少一个实施例中,专用进程编程模型是针对实现的。在该模型中,单个进程拥有图形加速模块2046或单个图形处理引擎2031。由于图形加速模块2046由单个进程拥有,因此管理程序为拥有的分区初始化加速器集成电路2036,并且当分配图形加速模块2046时操作系统对加速器集成电路2036进行初始化以用于拥有的分区。

[0368] 在操作中,加速器集成切片2090中的WD获取单元2091获取下一个WD 2084,其中包括要由图形加速模块2046的一个或更多个图形处理引擎完成的工作的指示。来自WD 2084的数据可以存储在寄存器2045被MMU 2039、中断管理电路2047和/或环境管理电路2048使用,如图所示。例如,MMU 2039的一个实施例包括用于访问OS虚拟地址空间2085内的段/页表2086的段/页面漫游电路。中断管理电路2047可以处理从图形加速模块2046接收到的中断事件2092。当执行图形操作时,由图形处理引擎2031-2032、N产生的有效地址2093由MMU 2039转换为实际地址。

[0369] 在一个实施例中,为每个图形处理引擎2031-2032、N和/或图形加速模块2046复制相同的寄存器组2045,并且可以由系统管理程序或操作系统来初始化。这些复制的寄存器中的每一个都可以包含在加速器集成切片2090中。表1中显示了可由管理程序初始化的示例性寄存器。

[0370] 表1-管理程序初始化的寄存器

[0371]

1	片控制寄存器
2	实址 (RA) 计划的处理区域指针
3	授权掩码覆盖寄存器
4	中断向量表输入偏移
5	中断向量表入口限制
6	状态寄存器
7	逻辑分区ID
8	实址 (RA) 管理程序加速器利用率记录指针
9	存储描述寄存器

[0372] 表2中示出了可以由操作系统初始化的示例性寄存器。

[0373] 表2-操作系统初始化寄存器

[0374]

1	进程和线程识别
2	有效地址 (EA) 环境保存/还原指针
3	虚拟地址 (VA) 加速器利用率记录指针
4	虚拟地址 (VA) 存储分段表指针
5	权威面具

6	工作描述符
---	-------

[0375] 在一个实施例中,每个WD 2084特定于特定的图形加速模块2046和/或图形处理引擎2031-2032、N。它包含图形处理引擎2031-2032、N进行工作或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,其中应用程序建立了要完成的工作的命令队列。

[0376] 图20E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2098,其中存储了进程元素列表2099。可以通过管理程序2096来访问管理程序实际地址空间2098,其中管理程序2096对图形加速模块引擎进行虚拟化以用于操作系统2095。

[0377] 在至少一个实施例中,共享的编程模型允许系统中全部或部分分区的全部或部分进程使用图形加速模块2046。存在两个共享图形加速模块2046的编程模型由多个进程和分区共享:时间片共享和图形方向共享。

[0378] 在此模型中,系统管理程序2096拥有图形加速模块2046,并且其功能可用于所有操作系统2095。为使图形加速模块2046支持系统管理程序2096进行虚拟化,图形加速模块2046可遵循以下规定: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2046必须提供环境保存和还原机制。2)应用程序的作业请求由图形加速模块2046保证以在指定的时间内完成,包括任何翻译错误,或者图形加速模块2046提供了抢占作业处理的能力。3)在有定向共享编程模型中操作时,必须确保图形加速模块2046在进程之间的公平性。

[0379] 在至少一个实施例中,需要应用程序2080进行具有图形加速模块2046类型、工作描述符(WD)、权限掩码寄存器(AMR)值和环境保存/恢复区域指针(CSRP)的操作系统2095系统调用。在至少一个实施例中,图形加速模块2046类型描述了用于系统调用的目标加速功能。在至少一个实施例中,图形加速模块2046的类型可以是针对系统的值。在至少一个实施例中,WD是专门为图形加速模块2046格式化的,并且可以采用图形加速模块2046命令、指向用户定义结构的有效地址指针、指向命令队列的有效地址指针或任何其他数据结构的形式来描述要由图形加速模块2046完成的工作。在一个实施例中,AMR值是要用于当前过程的AMR状态。在至少一个实施例中,传递给操作系统的值类似于AMR的应用程序设置。如果加速器集成电路2036和图形加速模块2046实现不支持用户权限掩码覆盖寄存器(UAMOR),则操作系统可以在管理程序调用中传递AMR之前将当前UAMOR值应用于AMR值。管理程序2096可以在将AMR放入进程元素2083之前,可选地应用当前的授权掩码覆盖寄存器(AMOR)值。在至少一个实施例中,CSRP是寄存器2045之一,其中包含应用程序有效地址空间2082中某个区域的有效地址,用于图形加速模块2046保存和恢复环境状态。如果不需要在作业之间保存状态或当作业被抢占时,则此指针是可选的。在至少一个实施例中,环境保存/恢复区域可以是固定的系统存储器。

[0380] 在接收到系统调用之后,操作系统2095可以验证应用程序2080已经被注册并且被授予使用图形加速模块2046的权限。然后,操作系统2095使用表3中所示的信息来调用系统管理程序2096。

[0381] 表3-OS到管理程序(Hypervisor)的调用参数

[0382]	1	工作描述符 (WD)
	2	授权屏蔽寄存器 (AMR) 值 (可能被屏蔽)
	3	有效地址 (EA) 环境保存/还原区域指针 (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	存储分段表指针 (SSTP) 的虚拟地址
[0383]	7	逻辑中断服务号码 (LISN)

[0384] 在接收到系统管理程序调用后,系统管理程序2096验证操作系统2095已注册并被授予使用图形加速模块2046的权限。系统管理程序2096然后将进程元素2083放入相应的图形加速模块2046类型的进程元素链表中。进程元素可能包含表4中所示的信息。

[0385] 表4-进程元素信息

[0386]	1	工作描述符 (WD)
	2	授权屏蔽寄存器 (AMR) 值 (可能被屏蔽)。
	3	有效地址 (EA) 环境保存/还原区域指针 (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	存储器分段表指针 (SSTP) 的虚拟地址
	7	逻辑中断服务号码 (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	实地址 (RA) 管理程序加速器利用率记录指针
	12	存储器描述符寄存器 (SDR)

[0387] 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2090寄存器2045。

[0388] 如图20F所示,在至少一个实施例中,使用统一存储器,该统一存储器可通过用于访问物理处理器存储器2001-2002和GPU存储器2020-2023的公共虚拟存储器地址空间来寻址。在该实现中,在GPU 2010-2013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2001-2002,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2001,第二部分被分配给第二处理器存储器2002,第三部分被分配给GPU存储器2020,等等。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2001-2002和GPU存储器2020-2023的每一个中,从而允许任何处理器或GPU访问具有映射到该存储器的虚拟地址的任何

物理存储器。

[0389] 在一个实施例中,一个或多个MMU 2039A-2039E内的偏置/一致性管理电路2094A-2094E确保一个或多个主机处理器(例如2005)和GPU 2010-2013的高速缓存之间的高速缓存一致性,并实现指示物理存储器的偏置技术,其中应存储某些类型的数据。尽管在图20F中示出了偏置/一致性管理电路2094A-2094E的多个实例,可以在一个或多个主机处理器2005的MMU内和/或在加速器集成电路2036内实现偏置/一致性电路。

[0390] 一个实施例允许附有GPU的存储器2020-2023被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,但是没有遭受与完整的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,将附接的GPU的存储器2020-2023作为系统存储器来访问而没有繁重的高速缓存一致性开销的能力为GPU卸载提供了有益的操作环境。这种安排允许主机处理器2005软件设置操作数并访问计算结果,而不会产生传统I/ODMA数据副本的开销。此类传统副本涉及驱动程序调用、中断和存储器映射的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附接的存储器2020-2023的能力对于卸载计算的执行时间是至关重要的。例如,在具有大量流写入存储器流量的情况下,缓存一致性开销会大大降低GPU 2010-2013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、结果访问的效率以及GPU计算的效率可以在确定GPU卸载的效率中起作用。

[0391] 在至少一个实施例中,GPU偏置和主处理器偏置的选择由偏置跟踪器数据结构驱动。可以使用例如偏置表,该偏置表可以是页面粒度结构(即,以存储器页面的粒度来控制),每个GPU连接的存储器页面包括1或2位。在至少一个实施例中,可以在一个或多个GPU附接的存储器2020-2023的被盗存储器范围中实现偏置表,在GPU 2010-2013中具有或不具有偏置缓存(例如,以高速缓存频繁/最近使用的偏置表条目)。可替代地,可以在GPU内维护整个偏置表。

[0392] 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附接的存储器2020-2023的每次访问相关联的偏置表条目,从而导致以下操作。首先,来自GPU 2010-2013的本地请求在GPU偏置中找到其页面被直接转发到相应的GPU存储器2020-2023。来自GPU的在主机偏置中找到其页面的本地请求被转发到处理器2005(例如,通过如上所述的高速链路)。在一个实施例中,来自处理器2005的在主机处理器偏置中找到所请求页面的请求完成了与普通存储器读取类似的请求。备选地,可以将针对GPU偏置的页面的请求转发给GPU 2010-2013。在至少一个实施例中,如果GPU当前不使用页面,则GPU可以将页面转换为主机处理器偏置。在至少一个实施例中,页面的偏置状态可以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,可以纯粹基于硬件的机制来改变。

[0393] 一种用于改变偏置状态的机制采用API调用(例如OpenCL),该调用反过来又调用GPU的设备驱动程序,该设备驱动程序反过来又向GPU发送消息(或排队命令描述符),将其引导到更改偏置状态,并在某些转换中在主机中执行缓存刷新操作。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2005偏置到GPU偏置的转变,但是不用于相反的转变。

[0394] 在一个实施例中,通过暂时渲染由主机处理器2005无法缓存的GPU偏置的页面,来保持缓存一致性。为了访问这些页面,处理器2005可以请求从GPU 2010进行访问,其可以立即授予或不授予访问权限。因此,为了减少处理器2005与GPU 2010之间的通信,有利的是确

保GPU偏置的页面是GPU所需要的页面,而不是主机处理器2005所需要的页面,反之亦然。

[0395] 硬件结构215用于执行一个或更多个实施例。下面结合图2A和/或2B提供关于硬件结构215的细节。

[0396] 图21示出了根据本文所述的各种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。

[0397] 图21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2100上的示例性系统的框图。在至少一个实施例中,集成电路2100包括一个或更多个应用处理器2105(例如,CPU)、至少一个图形处理器2110,并且可以另外包括图像处理器2115和/或视频处理器2120,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2100包括外围或总线逻辑,其包括USB控制器2125、UART控制器2130、SPI/SDIO控制器2135和I.sup.2S/I.sup.2C控制器2140。在至少一个实施例中,集成电路2100可以包括显示设备2145耦合到高清多媒体接口(HDMI)控制器2150和移动工业处理器接口(MIPI)显示接口2155中的一个或更多个。在至少一个实施例中,存储器可以由闪存子系统2160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2165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2170。

[0398] 推理和/或训练逻辑215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2A和/或2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中,推理和/或训练逻辑215可以在集成电路2100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。

[0399] 在至少一个实施例中,可以至少部分地在图21的一个或更多个集成电路2100上实现本文描述的客户端设备502,一个或更多个聚合服务器504,训练节点102,训练聚合器104,一个或更多个参数审查器1102和/或一个或更多个审查聚合器1110和/或方法中的一个或更多个。

[0400] 图22A-22B示出了根据本文所述的各种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。

[0401] 图22A-22B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22A示出了根据至少一个实施例的芯片集成电路上系统的示例性图形处理器2210,其可以使用一个或更多个IP核心来制造。图22B示出了根据至少一个实施例的芯片集成电路上系统的另外示例性图形处理器2240,其可以使用一个或更多个IP核心来制造。在至少一个实施例中,图22A的图形处理器2210是低功耗图形处理器核心。在至少一个实施例中,图22B的图形处理器2240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2210、2240可以是图21的图形处理器2110的变体。

[0402] 在至少一个实施例中,图形处理器2210包括顶点处理器2605和一个或更多个片段处理器2215A-2215N(例如2215A、2215B、2215C、2215D至2215N-1和2215N)。在至少一个实施例中,图形处理器2210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

2205被优化以执行针对顶点着色器程序的操作,而一个或更多个片段处理器2215A-2215N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2205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,片段处理器2215A-2215N使用由顶点处理器2205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,片段处理器2215A-2215N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 3DAPI中所提供的像素着色器程序类似的操作。

[0403] 在至少一个实施例中,图形处理器2210附加地包括一个或更多个存储器管理单元(MMU) 2220A-2220B、高速缓存2225A-2225B和电路互连2230A-2230B。在至少一个实施例中,一个或更多个MMU 2220A-2220B提供用于图形处理器2210的虚拟到物理地址的映射,包括用于顶点处理器2205和/或片段处理器2215A-2215N,其可以引用存储在存储器中的顶点或图像/纹理数据,除了存储在一个或更多个高速缓存2225A-2225B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

[0404] 2220A-2220B可以与系统内的其他MMU同步,包括与图21的一个或更多个应用处理器2105、图像处理器2115和/或视频处理器2120相关联的一个或更多个MMU,使得每个处理器2105-2120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2230A-2230B使图形处理器2210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。

[0405] 在至少一个实施例中,图形处理器2240包括图22A的图形处理器2210的一个或更多个MMU 2220A-2220B、高速缓存2225A-2225B和电路互连2230A-2230B。在至少一个实施例中,图形处理器2240包括一个或更多个着色器核心2255A-2255N(例如,2255A、2255B、2255C、2255D、2255E、2255F、直到2255N-1和2255N),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2240包括核心间任务管理器2245,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2255A-2255N和分块单元2258,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。

[0406] 推理和/或训练逻辑215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2A和/或2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中,推理和/或训练逻辑215可以在集成电路22A和/或22B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。

[0407] 在至少一个实施例中,可以至少部分地在图22A的图形处理器2210和/或图22B的图形处理器2240的一个或更多个上实现本文描述的客户端设备502,一个或更多个聚合服务器504,训练节点102,训练聚合器104,一个或更多个参数审查器1102和/或一个或更多个审查聚合器1110和/或方法中的一个或更多个。

[0408] 图23A-23B示出了根据本文描述的实施例的附加的示例性图形处理器逻辑。在至少一个实施例中,图23A示出了可以包括在图21的图形处理器2110内的图形核心2300,在至

少一个实施例中,可以是图22B中统一的着色器核心2255A-2255N。图23B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2330。

[0409] 在至少一个实施例中,图形核心2300包括共享指令高速缓存2302、纹理单元2318和高速缓存/共享存储器2320,它们是图形核心2300内的执行资源所共有的。在至少一个实施例中,图形核心2300可以包括多个切片2301A-2301N或每个核心的分区,图形处理器可以包括图形核心2300的多个实例。切片2301A-2301N可以包括支持逻辑,该支持逻辑包括本地指令高速缓存2304A-2304N、线程调度器2306A-2306N、线程分派器2308A-2308N和一组寄存器2310A-2310N。在至少一个实施例中,切片2301A-2301N可以包括一组附加功能单元 (AFU 2312A-2312N)、浮点单元 (FPU 2314A-2314N)、整数算术逻辑单元 (ALU 2316-2316N)、地址计算单元 (ACU 2313A-2313N)、双精度浮点单元 (DPFPU 2315A-2315N) 和矩阵处理单元 (MPU 2317A-2317N)。

[0410] 在至少一个实施例中,FPU 2314A-2314N可以执行单精度 (32位) 和半精度 (16位) 浮点运算,而DPFPU 2315A-2315N可以执行双精度 (64位) 浮点运算点操作。在至少一个实施例中,ALU 2316A-2316N可以以8位、16位和32位精度执行可变精度整数运算,并且可以被配置用于混合精度运算。在至少一个实施例中,MPU 2317A-2317N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 2317A-2317N可以执行各种矩阵操作以加速机器学习应用框架,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 2312A-2312N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,Sine、Cosine等)。

[0411] 推理和/或训练逻辑215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2A和/或2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中,推理和/或训练逻辑215可以在图形核心2300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。

[0412] 在至少一个实施例中,可以至少部分地在图23A的一个或更多个图形处理器2300上实现本文描述的客户端设备502,一个或更多个聚合服务器504,训练节点102,训练聚合器104,一个或更多个参数审查器1102和/或一个或更多个审查聚合器1110和/或方法中的一个或更多个。

[0413] 图23B示出了在至少一个实施例中的通用处理单元 (GPGPU) 2330,其可以被配置为使得高度并行的计算操作能够由图形处理单元阵列来执行。在至少一个实施例中,GPGPU 2330可以直接链路到GPGPU 2330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 2330包括主机接口2332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2332是PCI Express接口。在至少一个实施例中,主机 interface 2332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 2330从主机处理器接收命令,并使用全局调度器2334将与那些命令相关联的执行线程分派给一组计算集群

[0414] 2336A-2336H。在至少一个实施例中,计算集群2336A-2336H共享高速缓存存储器2338。在至少一个实施例中,高速缓存存储器2338可以用作计算集群2336A-2336H内的高速缓存存储器的高级高速缓存。

[0415] 在至少一个实施例中, GPGPU 2330包括经由一组存储器控制器2342A-2342B与计算集群2336A-2336H耦合的存储器2344A-2344B。在至少一个实施例中, 存储器2344A-2344B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。

[0416] 在至少一个实施例中, 计算集群2336A-2336H各自包括一组图形核心, 诸如图23A的图形核心2300, 其可以包括多种类型的整数和浮点逻辑单元, 其可以以包括用于在适用于机器学习计算的精度范围内执行计算操作。例如, 在至少一个实施例中, 每个计算集群2336A-2336H中的浮点单元的至少一个子集可以配置为执行16位或32位浮点运算, 而不同的浮点单元的子集可以配置为执行64位浮点运算。

[0417] 在至少一个实施例中, GPGPU 2330的多个实例可以被配置为操作为计算集群。在至少一个实施例中, 计算集群2336A-2336H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 2330的多个实例通过主机接口2332进行通信。在至少一个实施例中, GPGPU 2330包括I/O集线器2339, 其将GPGPU 2330与GPU链路2340耦合, 使得能够直接连接至GPGPU2330的其他的实例。在至少一个实施例中, GPU链路2340耦合到专用GPU到GPU桥接器, 其使得能够在GPGPU 2330的多个实例之间进行通信和同步。在至少一个实施例中, GPU链路2340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, GPGPU2330的多个实例位于单独的数据处理系统中, 并经由可经由主机接口2332访问的网络设备进行通信。在至少一个实施例中, GPU链路2340可被配置为能够连接到主机处理器, 附加或替代主机接口2332。

[0418] 在至少一个实施例中, GPGPU 2330可以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 2330。在其中使用GPGPU 2330进行推理的至少一个实施例中, 相对于当使用GPGPU训练神经网络时, GPGPU可以包括更少的计算集群2336A-2336H。在至少一个实施例中, 与存储器2344A-2344B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 2330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。

[0419] 推理和/或训练逻辑215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。下面结合图2A和/或2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中, 推理和/或训练逻辑215可以在GPGPU 2330中用于至少部分地基于本文描述的使用神经网络训练操作、神经网络功能和/或架构或神经网络用例计算的权重参数来推理或预测操作。

[0420] 在至少一个实施例中, 可以至少部分地在图23B的一个或多个GPGPU 2330上实现本文描述的客户端设备502, 一个或多个聚合服务器504, 训练节点102, 训练聚合器104, 一个或多个参数审查器1102和/或一个或多个审查聚合器1110和/或方法中的一个或多个。

[0421] 图24是示出根据至少一个实施例的计算系统2400的框图。在至少一个实施例中, 计算系统2400包括具有一个或多个处理器2402的处理子系统2401和经由可以包括存储器集线器2405的互连路径通信的系统存储器2404。在至少一个实施例中, 存储器集线器

2405可以是芯片组组件内的单独组件,也可以集成在一个或多个处理器2402内。在至少一个实施例中,存储器集线器2405通过通信链路2406与I/O子系统2411耦合。在一实施例中,I/O子系统2411包括I/O集线器2407,其可以使计算系统2400能够接收来自一个或多个输入设备2408的输入。在至少一个实施例中,I/O集线器2407可以使能显示控制器,其包括在一个或多个处理器2402中,用于向一个或多个显示设备2410A提供输出。在至少一个实施例中,与I/O集线器2407耦合的一个或多个显示设备2410A可以包括本地、内部或嵌入式显示设备。

[0422] 在至少一个实施例中,处理子系统2401包括经由总线或其他通信链路2413耦合到存储器集线器2405的一个或多个并行处理器2412。在至少一个实施例中,通信链路2413可以是许多基于标准的通信链路技术或协议中的一种,例如但不限于PCI Express,或者可以是针对供应商的通信接口或通信结构。在至少一个实施例中,一个或多个并行处理器2412形成计算集中的并行或向量处理系统,该系统可以包括大量的处理核心和/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,一个或多个并行处理器2412形成可以将像素输出到经由I/O集线器2407耦合的一个或多个显示设备2410A之一的图形处理子系统。在至少一个实施例中,一个或多个并行处理器2412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2410B。

[0423] 在至少一个实施例中,系统存储单元2414可以连接到I/O集线器2407,以提供用于计算系统2400的存储机制。在至少一个实施例中,I/O开关2416可以用于提供接口机制,以实现I/O集线器2407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2418和/或无线网络适配器2419,以及可以通过一个或多个附加设备2420添加的各种其他设备。在至少一个实施例中,网络适配器2418可以是以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2419可以包括Wi-Fi、蓝牙、近场通信(NFC)的一个或多个或其他包括一个或多个无线电的网络设备。

[0424] 在至少一个实施例中,计算系统2400可以包括未明确示出的其他组件,包括USB或其他端口连接、光存储驱动器、视频捕获设备等,也可以连接到I/O集线器2407。在至少一个实施例中,对图24中的各个组件进行互连的通信路径可以使用任何合适的协议来实现,诸如基于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(例如,NV-链路高速互连或互连协议)。

[0425] 在至少一个实施例中,一个或多个并行处理器2412包括针对图形和视频处理而优化的电路(包括例如视频输出电路),并构成图形处理单元(GPU)。在至少一个实施例中,一个或多个并行处理器2412包括针对通用处理而优化的电路。在至少一个实施例中,计算系统2400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2412、存储器集线器2405、处理器2402和I/O集线器2407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,计算系统2400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,计算系统2400的组件的至少一部分可以被集成到多芯片模块(MCM)中,该多芯片模块可以与其他多芯片模块互连到模块化计算系统中。

[0426] 推理和/或训练逻辑215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。下面结合图2A和/或2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例

中,推理和/或训练逻辑215可以在系统图2400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出的权重参数来推理或预测操作。

[0427] 在至少一个实施例中,可以至少部分地在图24的一个或多个计算系统2400上实现本文描述的客户端设备502,一个或多个聚合服务器504,训练节点102,训练聚合器104,一个或多个参数审查器1102和/或一个或多个审查聚合器1110和/或方法中的一个或多个。

[0428] 处理器

[0429] 图25A示出了根据至少一个实施例的并行处理器2500。在至少一个实施例中,并行处理器2500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2500是根据示例性实施例的图24所示的一个或多个并行处理器2412的变体。

[0430] 在至少一个实施例中,并行处理器2500包括并行处理单元2502。在至少一个实施例中,并行处理单元2502包括I/O单元2504,其使得能够与其他设备进行通信,包括并行处理单元2502的其他实例。在至少一个实施例中,I/O单元2504可以直接连接到其他设备。在至少一个实施例中,I/O单元2504通过使用集线器或交换机接口(例如,存储器集线器2405)与其他设备连接。在至少一个实施例中,存储器集线器2405与I/O单元2504之间的连接形成通信链路2413。在至少一个实施例中,I/O单元2504与主机接口2506和存储器交叉开关2516连接,其中主机接口2506接收用于执行处理操作的命令,而存储器交叉开关2516接收用于执行存储器操作的命令。

[0431] 在至少一个实施例中,当主机接口2506经由I/O单元2504接收命令缓冲区时,主机接口2506可以引导工作操作以执行那些命令到前端2508。在至少一个实施例中,前端2508与调度器2510耦合,调度器2510配置成将命令或其他工作项分配给处理集群阵列2512的处理集群阵列2512。在至少一个实施例中,调度器2510确保在将任务分配给处理集群阵列2512之前,处理集群阵列2512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2510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2510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2512上执行的线程的快速抢占和环境切换。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2512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2510的微控制器内的调度器2510逻辑在处理阵列2512上自动分配。

[0432] 在至少一个实施例中,处理集群阵列2512可以包括多达“N”个处理集群(例如,集群2514A、集群2514B到集群2514N)。在至少一个实施例中,处理集群阵列2512的每个集群2514A-2514N可以执行大量并发线程。在至少一个实施例中,调度器2510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2512的集群2514A-2514N,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2510动态地处理,或者可以在配置为由处理集群阵列2512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2512的不同的集群2514A-2514N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。

[0433] 在至少一个实施例中,处理集群阵列2512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2512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2512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。

[0434] 在至少一个实施例中,处理集群阵列2512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2512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2512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2502可以经由I/O单元2504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2522),然后将其写回到系统存储器。

[0435] 在至少一个实施例中,当并行处理单元2502用于执行图形处理时,调度器2510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2512的多个集群2514A-2514N。在至少一个实施例中,处理集群阵列2512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2514A-2514N中的一个或多个产生的中间数据存储到缓冲区中,以允许在集群2514A-2514N之间传输中间数据以进行进一步处理。

[0436] 在至少一个实施例中,处理集群阵列2512可以经由调度器2510接收要执行的处理任务,该调度器2510从前端2508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2510可以配置成获取与任务相对应的索引,或者可以从前端2508接收索引。在至少一个实施例中,前端2508可以配置成确保在启动由传入命令缓冲区(例如,批处理缓冲区、推入缓冲区等)指定的工作负载之前,处理集群阵列2512配置成有效状态。

[0437] 在至少一个实施例中,并行处理单元2502的一个或多个实例中的每一个可以与并行处理器存储器2522耦合。在至少一个实施例中,可以经由存储器交叉开关2516访问并行处理器存储器2522,所述存储器交叉开关2516可以接收来自处理集群阵列2512以及I/O单元2504的存储器请求。在至少一个实施例中,存储器交叉开关2516可以经由存储器接口2518访问并行处理器存储器2522。在至少一个实施例中,存储器接口2518可以包括多个分区单元(例如,分区单元2520A、分区单元2520B、直到分区单元2520N),其可各自耦合至并行处理器存储器2522的一部分(例如,存储单元)。在至少一个实施例中,多个分区单元2520A-2520N为配置为等于存储单元的数量,使得第一分区单元2520A具有相应的第一存储单元2524A,第二分区单元2520B具有相应的存储单元2520B,第N分区单元2520N具有相应的第N存储单元2524N。在至少一个实施例中,分区单元2520A-2520N的数量可以不等于存储设备的数量。

[0438] 在至少一个实施例中,存储器单元2524A-2524N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2524A-2524N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2524A-2524N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2520A-2520N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2522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2522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。

[0439] 在至少一个实施例中,处理集群阵列2512的集群2514A-2514N中的任何一个都可以处理将被写入并行处理器存储器2522内的任何存储器单元2524A-2524N中的数据。在至少一个实施例中,存储器交叉开关2516可以配置为将每个集群2514A-2514N的输出传输到任何分区单元2520A-2520N或另一个集群2514A-2514N,集群2514A-2514N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2514A-2514N可以通过存储器交叉开关2516与存储器接口2518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2516具有到存储器接口2518的连接以与I/O单元2504通信,以及到并行处理器存储器2522的本地实例的连接,从而使不同处理集群2514A-2514N内的处理单元与系统存储器或不是并行处理单元2502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2516可以使用虚拟通道来分离集群2514A-2514N和分区单元2520A-2520N之间的业务流。

[0440] 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2502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2502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2502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2502或并行处理器2500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。

[0441] 图25B是根据至少一个实施例的分区单元2520的框图。在至少一个实施例中,分区单元2520是图25A的分区单元2520A-2520N之一的实例。在至少一个实施例中,分区单元2520包括L2高速缓存2521、帧缓冲区接口2525和光栅操作单元 (“ROP”) 2526。L2高速缓存2521是读/写高速缓存,其配置成执行从存储器交叉开关2516和ROP 2526接收的加载和存储操作。在至少一个实施例中,L2高速缓存2521将读取未命中和紧急回写请求输出到帧缓冲区接口2525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2525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2525与并行处理器存储器中的存储器单元 (诸如图25的存储器单元2524A-2524N (例如,在并行处理器存储器2522内)) 之一相互作用。

[0442] 在至少一个实施例中,ROP 2526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 2526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 2526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。在至少一个实施例中,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 2526执行的压缩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。

[0443] 在至少一个实施例中,ROP 2526包括在每个处理集群内(例如,图25的集群2514A-2514N),而不是在分区单元2520内。在至少一个实施例中,通过存储器交叉开关2516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24的一个或多个显示设备2410之一)显示,由处理器2402路由以供进一步处理,或者由图25A的并行处理器2500内的处理实体之一路由以供进一步处理。

[0444] 图25C是根据至少一个实施例的并行处理单元内的处理集群2514的框图。在至少一个实施例中,处理集群是图25的处理集群2514A-2514N之一的实例。在至少一个实施例中,一个或多个处理集群2514中的一个或多个可以配置成并行执行许多线程,其中术语“线程”可以至少一个实施例中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。

[0445] 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2532来控制处理集群2514的操作。在至少一个实施例中,管线管理器2532从图25的调度器2510接收指令,通过图形多处理器2534和/或纹理单元2536管理这些指令的执行。在至少一个实施例中,图形多处理器2534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2514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2514内可以包括图形多处理器2534的一个或多个实例。在至少一个实施例中,图形多处理器2534可以处理数据,并且数据交叉开关2540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2532可以通过指定要针对数据交叉开关2540分配的处理数据的目的地来促进处理数据的分配。

[0446] 在至少一个实施例中,处理集群2514内的每个图形多处理器2534可以包括相同的一组功能执行逻辑(例如,算术逻辑单元、负载存储单元等)。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。

[0447] 在至少一个实施例中,传送到处理集群2514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2534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2534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2534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2534内的处理引擎更多的线程时,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2534上同时执行多个线程组。

[0448] 在至少一个实施例中,图形多处理器2534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2534可以放弃内部高速缓存并使用处理集群2514内的高速缓存存储器(例如,L1高速缓存2548)。在至少一个实施例中,每个图形多处理器2534还可以访问分区单元(例如,图25的分区单元2520A-2520N)内的L2高速缓存,这些分区单元在所有处理集群2514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2534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多个。在至少一个实施例中,并行处理单元2502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2514包括图形多处理器2534的多个实例,它们可以共享可以存储在L1高速缓存2548中的公共指令和数据。

[0449] 在至少一个实施例中,每个处理集群2514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”)2545。在至少一个实施例中,MMU 2545的一个或多个实例可以驻留在图25的存储器接口2518内。在至少一个实施例中,MMU 2545包括一组页表条目(PTE),其用于将虚拟地址映射到图块(详细讨论图块)的物理地址以及在至少一个实施例中映射到高速缓存存储器行。在至少一个实施例中,MMU 2545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2534或L1高速缓存或处理集群2514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。

[0450] 在至少一个实施例中,可以配置处理集群2514,使得每个图形多处理器2534耦合到纹理单元2536,以执行纹理映射操作,例如,确定纹理样本位置、读取纹理数据以及过滤纹理数据。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2534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2534将处理后的任务输出到数据交叉开关2540,以将处理后的任务提供给另一处理集群2514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2516的系统存储器中。在至少一个实施例中,preROP2542(光栅前操作单元)配置成从图形多处理器2534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25A的分区单元2520A-2520N)一起定位。在至少一个实施例中,PreROP2542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。

[0451] 推理和/或训练逻辑215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。下面结合图2A和/或2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中,推理和/或训练逻辑215可以在图形处理集群2514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、神经网络功能和/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。

[0452] 在至少一个实施例中,可以至少部分地在一个或多个并行处理器2500上实现本

文描述的客户端设备502,一个或更多个聚合服务器504,训练节点102,训练聚合器104,一个或更多个参数审查器1102和/或一个或更多个审查聚合器1110和/或方法中的一个或更多个。

[0453] 图25D示出了根据至少一个实施例的图形多处理器2534。在至少一个实施例中,图形多处理器2534与处理集群2514的管线管理器2532耦合。在至少一个实施例中,图形多处理器2534具有执行管线,该执行管线包括但不限于指令高速缓存2552、指令单元2554、地址映射单元2556、寄存器文件2558、一个或更多个通用图形处理单元(GPGPU)核心2562和一个或更多个加载/存储单元2566。GPGPU核心2562和加载/存储单元2566与高速缓存存储器2572和共享存储器2570通过存储器和高速缓存互连2568耦合。

[0454] 在至少一个实施例中,指令高速缓存2552从管线管理器2532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2552中并将其分派以供指令单元2554执行。在一个实施例中,指令单元2554可以分派指令作为线程组(例如,线程束),将每个线程组分配给GPGPU核心2562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2556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2566访问的不同的存储器地址。

[0455] 在至少一个实施例中,寄存器文件2558为图形多处理器2534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2558为连接到图形多处理器2534的功能单元(例如,GPGPU核心2562、加载/存储单元2566)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2558,使得为每个功能单元分配寄存器文件2558的专用部分。在至少一个实施例中,寄存器文件2558在图形多处理器2534正在执行的不同线程束之间划分。

[0456] 在至少一个实施例中,GPGPU核心2562可以各自包括用于执行图形多处理器2534的指令的浮点单元(FPU)和/或整数算术逻辑单元(ALU)。GPGPU核心2562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2562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2534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。

[0457] 在至少一个实施例中,GPGPU核心2562包括能够对多组数据执行单个指令的SIMD逻辑。在至少一个实施例中,GPGPU核心2562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。

[0458] 在至少一个实施例中,存储器和高速缓存互连2568是将图形多处理器2534的每个功能单元连接到寄存器文件2558和共享存储器2570的互连网络。在至少一个实施例中,存

储器和高速缓存互连2568是交叉开关互连,其允许加载/存储单元2566在共享存储器2570和寄存器文件2558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2558可以以与GPGPU核心2562相同的频率操作,从而在GPGPU核心2562和寄存器文件2558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2570可以用于启用在图形多处理器2534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2572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2536之间通信的纹理数据。在至少一个实施例中,共享存储器2570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2572中的自动高速缓存的数据之外,在GPGPU核心2562上执行的线程还可以以编程方式将数据存储在共享存储器中。

[0459] 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。

[0460] 推理和/或训练逻辑215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2A和/或图2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中,推理和/或训练逻辑215可以在图形多处理器2534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、神经网络功能和/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,。

[0461] 在至少一个实施例中,可以至少部分地在一个或更多个图形多处理器2534上实现本文描述的客户端设备502,一个或更多个聚合服务器504,训练节点102,训练聚合器104,一个或更多个参数审查器1102和/或一个或更多个审查聚合器1110和/或方法中的一个或更多个。

[0462] 图26示出了根据至少一个实施例的多GPU计算系统2600。在至少一个实施例中,多GPU计算系统2600可包括经由主机接口开关2604耦合到多个通用图形处理单元(GPGPU) 2606A-D的处理器2602。在至少一个实施例中,主机接口开关2604是PCI高速开关设备,其将处理器2602耦合到PCI高速总线,处理器2602可以通过PCI高速总线与GPGPU

[0463] 2606A-D通信。GPGPU 2606A-D可以经由一组高速点对点GPU到GPU链路2616来互连。在至少一个实施例中,GPU到GPU链路2616经由专用GPU链路连接到GPGPU 2606A-D中的每一个。在至少一个实施例中,P2P GPU链路2616使得能够在每个GPGPU 2606A-D之间直接进行通信,而无需通过处理器2602所连接的主机接口总线2604进行通信。在至少一个实施例中,在GPU到GPU业务流指向P2P GPU链路2616的情况下,主机接口总线2604保持可用于系统存储器访问或者例如经由一个或更多个网络设备与多GPU计算系统2600的其他实例通信。虽然在至少一个实施例中,GPGPU 2606A-D经由主机接口开关2604连接到处理器2602,但是在至少一个实施例中,处理器2602包括对P2P GPU链路2616的直接支持,并且可以直接连接到GPGPU 2606A-D。

[0464] 推理和/或训练逻辑215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。下面结合图2A和/或图2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中,推理和/或训练逻辑215可以在多GPU计算系统2600中用于至少部分地基于使用本文所述的神经网络训练操作,神经网络功能和/或或架构或神经网络用例计算出的权重参数来进行推理或预测操作。

[0465] 在至少一个实施例中,可以至少部分地在一个或更多个GPU计算系统2600上实现本文描述的客户端设备502,一个或更多个聚合服务器504,训练节点102,训练聚合器104,一个或更多个参数审查器1102和/或一个或更多个审查聚合器1110和/或方法中的一个或更多个。

[0466] 图27是根据至少一个实施例的图形处理器2700的框图。在至少一个实施例中,图形处理器2700包括环形互连2702、管线前端2704、媒体引擎2737和图形核心2780A-2780N。在至少一个实施例中,环形互连2702将图形处理器2700耦合到其他处理单元,包括其他图形处理器或一个或更多个通用处理器核心。在至少一个实施例中,图形处理器2700是集成在多核心处理系统内的许多处理器之一。

[0467] 在至少一个实施例中,图形处理器2700经由环形互连2702接收多批命令。在至少一个实施例中,输入命令由管线前端2704中的命令流服务器(command streamer) 2703解释。在至少一个实施例中,图形处理器2700包括可缩放执行逻辑,以经由图形核心2780A-2780N执行3D几何处理和媒体处理。在至少一个实施例中,对于3D几何处理命令,命令流服务器2703将命令提供给几何管线2736。在至少一个实施例中,对于至少一些媒体处理命令,命令流服务器2703将命令提供给视频前端2734,其与媒体引擎2737耦合。在至少一个实施例中,媒体引擎2737包括用于视频和图像后处理的视频质量引擎(VQE) 2730,以及用于提供硬件加速媒体数据编码和解码的多格式编码/解码(MFX) 2733引擎。在至少一个实施例中,几何管线2736和媒体引擎2737各自生成用于由至少一个图形核心2780A提供的线程执行资源的执行线程。

[0468] 在至少一个实施例中,图形处理器2700包括以模块化核心2780A-2780N(有时称为核心切片)为特征的可缩放线程执行资源,每个模块核心具有多个子核心2750A-2750N、2760A-2760N(有时称为核心子切片)。在至少一个实施例中,图形处理器2700可以具有任意数量的图形核心2780A至2780N。在至少一个实施例中,图形处理器2700包括具有至少第一子核心2750A和第二子核心2760A的图形核心2780A。在至少一个实施例中,图形处理器2700是具有单个子核心(例如2750A)的低功率处理器。在至少一个实施例中,图形处理器2700包括多个图形核心2780A-2780N,每个图形核心包括一组第一子核心2750A-2750N和一组第二子核心2760A-2760N。在至少一个实施例中,第一子核心2750A-2750N中的每个子核心至少包括第一组执行单元2752A-2752N和媒体/纹理采样器2754A-2754N。在至少一个实施例中,第二子核心2760A-2760N中的每个子核心至少包括第二组执行单元2762A-2762N和采样器2764A-2764N。在至少一个实施例中,每个子核心2750A-2750N、2760A-2760N共享一组共享资源3170A-3170N。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。

[0469] 推理和/或训练逻辑215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2A和/或图2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例

中,推理和/或训练逻辑215可以在图形处理器2700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、神经网络功能和/或架构或神经网络用例计算的权重参数来推理或预测操作。

[0470] 在至少一个实施例中,可以至少部分地在一个或多个图形处理器2700上实现本文描述的客户端设备502,一个或多个聚合服务器504,训练节点102,训练聚合器104,一个或多个参数审查器1102和/或一个或多个审查聚合器1110和/或方法中的一个或多个。

[0471] 图28是示出根据至少一个实施例的用于处理器3200的微架构的框图,该处理器2800可以包括执行指令的逻辑电路。在至少一个实施例中,处理器2800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2810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MMX™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2810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。

[0472] 在至少一个实施例中,处理器2800包括有序前端(“前端”)3201,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2801可以包括几个单元。在至少一个实施例中,指令预取器2826从存储器中获取指令并将指令提供给指令解码器2828,指令解码器2828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2828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”)的一个或多个操作。在至少一个实施例中,指令解码器2828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2830可以将解码的微指令组装成微指令队列2834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2830遇到复杂指令时,微码ROM 2832提供完成操作所需的微指令。

[0473] 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2828可以访问微码ROM 2832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2828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 2832中。在至少一个实施例中,追踪高速缓存器2830参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 2832读取微码序列以完成一个或多个指令。在至少一个实施例中,在微码ROM 2832完成对指令的微操作排序之后,机器的前端2801可以恢复从追踪高速缓存2830获取微操作。

[0474] 在至少一个实施例中,乱序执行引擎(“乱序引擎”)2803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。在至少一个实施例中,乱序执行引擎2803包

包括但不限于分配器/寄存器重命名器2840、存储器微指令队列2842、整数/浮点微指令队列2844、存储器调度器2846、快速调度器2802、慢速/通用浮点调度器(“慢速/通用FP调度器”)2804和简单浮点调度器(“简单FP调度器”)2806。在至少一个实施例中,快速调度器2802、慢速/通用浮点调度器2804和简单浮点调度器2806也统称为“微指令调度器2802、2804、2806”。在至少一个实施例中,分配器/寄存器重命名器2840分配每个微指令按顺序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2840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2840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2842用于存储器操作和整数/浮点微指令队列2844用于非存储器操作,在存储器调度器2846和微指令调度器2802、2804、2806的前面。在至少一个实施例中,微指令调度器2802、2804、2806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2802可以在主时钟周期的每个一半上调度,而慢速/通用浮点调度器2804和简单浮点调度器2806可以在每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2802、2804、2806对调度端口进行仲裁,以调度用于执行的微指令。

[0475] 在至少一个实施例中,执行块b11包括但不限于整数寄存器文件/支路网络2808、浮点寄存器文件/支路网络(“FP寄存器文件/支路网络”)2810、地址生成单元(“AGU”)2812和2814、快速算术逻辑单元(“快速ALU”)2816和2818、慢速算术逻辑单元(“慢速ALU”)2820、浮点ALU(“FP”)2822和浮点移动单元(“FP移动”)2824。在至少一个实施例中,整数寄存器文件/支路网络2808和浮点寄存器文件/旁路网络2810在本文中也称为“寄存器文件2808、2810”。在至少一个实施例中,AGU 2812和2814、快速ALU 2816和2818、慢速ALU 2820、浮点ALU 2822和浮点移动单元2824在本文中也称为“执行单元2812、2814、2816、2818、2820、2822和2824”。在至少一个实施例中,执行框b11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)。

[0476] 在至少一个实施例中,寄存器文件2808、2810可以布置在微指令调度器2802、2804、2806与执行单元2812、2814、2816、2818、2820、2822和2824之间。在至少一个实施例中,整数寄存器文件/支路网络2808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2810执行浮点操作。在至少一个实施例中,寄存器文件2808、2810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2808、2810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2808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2810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。

[0477] 在至少一个实施例中,执行单元2812、2814、2816、2818、2820、2822、2824可以执行指令。在至少一个实施例中,寄存器文件2808、2810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2800可以包括但不限于任何数量的执行单元2812、2814、2816、2818、2820、2822、2824及其组合。在至少一个实施例中,浮点ALU 2822和浮点移动单元2824,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 2822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 2816、2818。在至少一个实施例中,快速ALU 2816、2818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 2820,因为慢速ALU 2820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AGUS 2812、2814执行。在至少一个实施例中,快速ALU 2816、快速ALU 2818和慢速ALU 2820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 2816、快速ALU 2818和慢速ALU 2820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 2822和浮点移动单元2824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 2822和浮点移动单元2824可以结合SIMD和多媒体指令对128位宽封装数据操作数进行操作。

[0478] 在至少一个实施例中,微指令调度器2802、2804、2806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2800中推测性地调度和执行微指令,处理器2800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。

[0479] 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。

[0480] 推理和/或训练逻辑215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2A和/或图2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中,可以将推理和/或训练逻辑215的部分或全部并入EXE块2811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EXE块2811中示出的一个或更多个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EXE块2811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。

[0481] 在至少一个实施例中,可以至少部分地在一个或更多个处理器2800上实现本文描述的客户端设备502,一个或更多个聚合服务器504,训练节点102,训练聚合器104,一个或更多个参数审查器1102和/或一个或更多个审查聚合器1110和/或方法中的一个或更多个。

[0482] 图29示出了根据至少一个实施例的深度学习应用处理器2900。在至少一个实施例

中,深度学习应用处理器2900使用指令,如果由深度学习应用处理器2900执行,则指令使深度学习应用处理器2900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,深度学习应用处理器2900是专用集成电路(ASIC)。在至少一个实施例中,应用处理器2900执行矩阵乘法运算或者“硬连线”到硬件中,作为执行一个或更多个指令或两者的结果。在至少一个实施例中,深度学习应用处理器2900包括但不限于处理集群2910(1)-2910(12)、芯片间链路(“ICL”)2920(1)-2920(12)、芯片间控制器(“ICC”)2930(1)-2930(2)、第二代高带宽内存(“HBM2”)2940(1)-2940(4)、存储器控制器(“Mem Ctrlrs”)2942(1)-2942(4)、高带宽存储器物理层(“HBM PHY”)2944(1)-2944(4)、管理控制器中央处理单元(“管理控制器CPU”)2950、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”)2960、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”)2970、以及十六通道外围组件互连快速端口(“PCI Express x 16”)2980。

[0483] 在至少一个实施例中,处理集群2910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2910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,深度学习应用处理器2900可以包括任何数量和类型的处理集群2900。在至少一个实施例中,芯片间链路2920是双向的。在至少一个实施例中,芯片间链路2920和芯片间控制器2930使多个深度学习应用处理器2900能够交换信息,包括从执行一个或更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,深度学习应用处理器2900可以包括任意数量(包括零)和类型的ICL 2920和ICC 2930。

[0484] 在至少一个实施例中,HBM2 2940提供总共32GB的存储器。HBM2 2940(i)与存储器控制器2942(i)和HBM PHY 2944(i)都相关联。在至少一个实施例中,任何数量的HBM2 2940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2942和HBM PHY 2944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 2960、PCIe控制器和DMA 2970和/或PCIe 2980,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。

[0485] 推理和/或训练逻辑215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2A和/或2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中,深度学习应用处理器2900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处理器2900的信息。在至少一个实施例中,深度学习应用处理器2900用于基于已经由另一处理器或系统或由深度学习应用处理器2900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2900可以用于执行本文所述的一个或更多个神经网络用例。

[0486] 在至少一个实施例中,可以至少部分地在一个或更多个处理器2900上实现本文描述的客户设备502,一个或更多个聚合服务器504,训练节点102,训练聚合器104,一个或更多个参数审查器1102和/或一个或更多个审查聚合器1110和/或方法中的一个或更多个。

[0487] 图30是根据至少一个实施例的神经形态处理器3000的框图。在至少一个实施例中,神经形态处理器3000可以从神经形态处理器3000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3000内的一个或更多个神经元3002。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3002及其组件。在至少一个实施例中,神经形态处理器3000可以包括但不限于成千上万个神经元3002的实例,但是可以使用任何合适数量的神经元3002。在至少一个实施例中,神经元3002的每个实例可以包括神经元输入3004和神经元输出3006。在至少一个实施例中,神经元3002可以生成可以传输到神经元3002的其他实例的输入的输出。在至少一个实施例中,神经元输入3004和神经元输出3006可以经由突触3008互连。

[0488] 在至少一个实施例中,神经元3002和突触3008可以互连,使得神经形态处理器3000操作以处理或分析由神经形态处理器3000接收的信息。在至少一个实施例中,当通过神经元输入3004接收到的输入超过阈值时,神经元3002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3002可以对在神经元输入3004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3002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3002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3004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3004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3002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3002可以包括但不限于当将传递函数应用于神经元输入3004的结果超过阈值时在神经元输出3006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3002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3002可以在合适的时间段(或修复期)之后恢复正常操作。

[0489] 在至少一个实施例中,神经元3002可以通过突触3008互连。在至少一个实施例中,突触3008可以操作以将从第一神经元3002的输出的信号传输到第二神经元3002的输入。在至少一个实施例中,神经元3002可以在一个以上的突触3008实例上传输信息。在至少一个实施例中,神经元输出3006的一个或更多个实例可以通过突触3008的实例连接到同一神经元3002中神经元输入3004的实例。在至少一个实施例中,相对于突触3008的那个实例,神经元3002的实例产生要在突触3008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3008的实例,神经元3002的实例接收通过突触3008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3008的各种实例,因为神经元3002的实例可以接收来自一个或更多个突触3008实例的输入,并且还可以通过一个或更多个突触3008实例传输输出,因此神经元3002的单个实例可以既是“突触前神经元”又是“突触后神经元”。

[0490] 在至少一个实施例中,神经元3002可以被组织成一层或更多层。神经元3002的每个实例可以具有一个神经元输出3006,该神经元输出3006可以通过一个或更多个突触3008扇出到一个或更多个神经元输入3004。在至少一个实施例中,第一层3010中的神经元3002的神经元输出3006可以连接到第二层3012中的神经元3002的神经元输入3004。在至少一个实施例中,层3010可以被称为“前馈层”。在至少一个实施例中,在第一层3010的实例中神经元3002的每个实例可以扇出到第二层3012中的神经元3002的每个实例。在至少一个实施例

中,第一层3010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3012的每个实例中的神经元3002的每个实例扇出到少于在第三层3014中的神经元3002的所有实例。在至少一个实施例中,第二层3012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3012中的神经元3002可以扇出到多个其他层中的神经元3002,包括(相同)第二层3012中的神经元3002。在至少一个实施例中,第二层3012可以被称为“循环层”。在至少一个实施例中,神经形态处理器3000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。

[0491] 在至少一个实施例中,神经形态处理器3000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3008连接到神经元3002。在至少一个实施例中,神经形态处理器3000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3002。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3008连接到神经元3002。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。

[0492] 在至少一个实施例中,可以至少部分地在一个或多个神经形态处理器3000上实现本文描述的客户端设备502,一个或多个聚合服务器504,训练节点102,训练聚合器104,一个或多个参数审查器1102和/或一个或多个审查聚合器1110和/或方法中的一个或多个。

[0493] 图31是根据至少一个实施例的处理系统的框图。在至少一个实施例中,系统3100包括一个或多个处理器3102和一个或多个图形处理器3108,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3102或处理器核心3107的服务器系统。在至少一个实施例中,系统3100是结合在片上系统(SoC)集成电路内的处理平台,以用于移动、手持或嵌入式设备。

[0494] 在至少一个实施例中,系统3100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3100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3100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3100是电视或机顶盒设备,其具有一个或多个处理器3102以及由一个或多个图形处理器3108生成的图形接口。

[0495] 在至少一个实施例中,一个或多个处理器3102每个包括一个或多个处理器核心3107,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3107中的每一个被配置为处理特定指令集3109。在至少一个实施例中,指令集3109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,多个处理器核心3107可以各自处理不同的指令集3109,该指令集3109可以包括有助于仿真其他指令集的指令。在至少一个实施例中,处理器核心3107还可以包括其他处理设备,例如数字信号处理器(DSP)。

[0496] 在至少一个实施例中,处理器3102包括高速缓存存储器3104。在至少一个实施例中,处理器3102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3102的各个组件之间共享。在至少一个实施例中,处理器

3102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),其可以使用已知的高速缓存一致性技术在处理器核心3107之间共享该逻辑。在至少一个实施例中,处理器3102中另外包括寄存器文件3106,处理器3102可以包括用于存储不同类型的的不同类型的寄存器的不同数据类型的寄存器(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3106可以包括通用寄存器或其他寄存器。

[0497] 在至少一个实施例中,一个或多个处理器3102与一个或多个接口总线3110耦合,以在处理器3102与系统3100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3110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3110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3102包括集成存储器控制器3116和平台控制器集线器3130。在至少一个实施例中,存储器控制器3116促进存储设备与系统3100的其他组件之间的通信,而平台控制器集线器(PCH)3130通过本地I/O总线提供到I/O设备的连接。

[0498] 在至少一个实施例中,存储设备3120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过程存储器。在至少一个实施例中,存储设备3120可以用作系统3100的系统存储器,以存储数据3122和指令3121,以在一个或多个处理器3102执行应用或过程时使用。在至少一个实施例中,存储器控制器3116还与可选的外部图形处理器3112耦合,其可以与处理器3102中的一个或多个图形处理器3108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3111可以连接至处理器3102。在至少一个实施例中,显示设备3111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便携式计算机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。在至少一个实施例中,显示设备3111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。

[0499] 在至少一个实施例中,平台控制器集线器3130使外围设备能够通过高速I/O总线连接到存储设备3120和处理器3102。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3146、网络控制器3134、固件接口3128、无线收发器3126、触摸传感器3125、数据存储设备3124(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3124可以经由存储器接口(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、PCI Express)。在至少一个实施例中,触摸传感器3125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3126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3128使能与系统固件的通信,并且可以是例如统一的可扩展固件接口(UEFI)。在至少一个实施例中,网络控制器3134可以启用到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3110耦合。在至少一个实施例中,音频控制器3146是多通道高清晰度音频控制器。在至少一个实施例中,系统3100包括可选的遗留I/O控制器3140,用于将遗留(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3130还可以连接到一个或多个通用串行总线(USB)控制器3142,该控制器连接输入设备,诸如键盘和鼠标3143组合、相机3144或其他USB输入设备。

[0500] 在至少一个实施例中,存储器控制器3116和平台控制器集线器3130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3112。在至少一个实施例中,平台控制器集线器3130和/或存储控制器3116可以在一个或更多个处理器3102的外部。例如,在至少一个实施例中,系统3100可以包括外部存储控制器3116和平台控制器集线器3130,其可以配置成在与处理器3102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。

[0501] 推理和/或训练逻辑215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2A和/或2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中,推理和/或训练逻辑215的部分或全部可以被合并到图形处理器3100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种或更多种体现在3D管线3112中的ALU。此外,在至少一个实施例中,可以使用除图2A或图2B所示的逻辑以外的逻辑来完成本文所述的推理和/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或/或寄存器(示出或未示出)中,其配置图形处理器3100的ALU以执行一种或更本文描述的多种机器学习算法、神经网络架构、用例或训练技术。

[0502] 在至少一个实施例中,可以至少部分地在一个或更多个图形处理器3100上实现本文描述的客户端设备502,一个或更多个聚合服务器504,训练节点102,训练聚合器104,一个或更多个参数审查器1102和/或一个或更多个审查聚合器1110和/或方法中的一个或更多个。

[0503] 图32是根据至少一个实施例的具有一个或更多个处理器核心3202A-3202N、集成存储器控制器3214和集成图形处理器3208的处理器3200的框图。在至少一个实施例中,处理器3200可以包括直至并包括由虚线框表示的附加核心3202N的附加核心。在至少一个实施例中,每个处理器核心3202A-3202N包括一个或更多个内部高速缓存单元3204A-3204N。在至少一个实施例中,每个处理器核心还可以访问一个或更多个共享高速缓存的单元3206。

[0504] 在至少一个实施例中,内部高速缓存单元3204A-3204N和共享高速缓存单元3206表示处理器3200内的高速缓存存储器层次结构。在至少一个实施例中,高速缓存存储器单元3204A-3204N可以包括每个处理器核心内的至少一级指令和数据以及共享中级缓存中的一级或更多级缓存,例如2级(L2)、3级(L3)、4级(L4)或其他级别的缓存,其中在外部存储器之前将最高级别的缓存归类为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3206和3204A-3204N之间的一致性。

[0505] 在至少一个实施例中,处理器3200还可包括一组一个或更多个总线控制器单元3216和系统代理核心3210。在至少一个实施例中,一个或更多个总线控制器单元3216管理一组外围总线,例如一个或更多个PCI或PCI Express总线。在至少一个实施例中,系统代理核心3210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3210包括一个或更多个集成存储器控制器3214,以管理对各种外部存储器设备(未示出)的访问。

[0506] 在至少一个实施例中,一个或更多个处理器核心3202A-3202N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3210包括用于在多线程处理期间协调和操作核心3202A-3202N的组件。在至少一个实施例中,系统代理核心3210可以另外包括电源控制单元(PCU),该电源控制单元包括逻辑和组件以调节处理器核心3202A-3202N和图形处理器3208的一个或更多个电源状态。

[0507] 在至少一个实施例中,处理器3200另外包括图形处理器3208以执行图形处理操作。在至少一个实施例中,图形处理器3208与共享高速缓存单元3206和包括一个或更多个集成存储器控制器3214的系统代理核心3210耦合。在至少一个实施例中,系统代理核心3210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3211。在至少一个实施例中,显示器控制器3211也可以是经由至少一个互连与图形处理器3208耦合的独立模块,或者可以集成在图形处理器3208内。

[0508] 在至少一个实施例中,基于环的互连单元3212用于耦合处理器3200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3208经由I/O链路3213与环形互连3212耦合。

[0509] 在至少一个实施例中,I/O链路3213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3218(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3202A-3202N和图形处理器3208中的每一个使用嵌入式存储器模块3218作为共享的最后一级高速缓存。

[0510] 在至少一个实施例中,处理器核心3202A-3202N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3202A-3202N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3202A-3202N执行公共指令集,而一个或更多个其他处理器核心3202A-36-02N执行公共指令集或不同指令集的子集。在至少一个实施例中,就微架构而言,处理器核心3202A-3202N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3200可以实现在一个或更多个芯片上或被实现为SoC集成电路。

[0511] 推理和/或训练逻辑215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2A和/或图2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中,可以将推理和/或训练逻辑215的部分或全部合并到图形处理器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图32中的3D管线3112、图形核心3215A,共享功能逻辑3216,图形核心3215B,共享功能逻辑3220或其他逻辑。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2A或2B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3210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。

[0512] 在至少一个实施例中,可以至少部分地在一个或更多个图形处理器3210上实现本文描述的客户端设备502,一个或更多个聚合服务器504,训练节点102,训练聚合器104,一个或更多个参数审查器1102和/或一个或更多个审查聚合器1110和/或方法中的一个或更多个。

[0513] 图33是根据本文所述的至少一个实施例的图形处理器核心3300的硬件逻辑的框图。在至少一个实施例中,图形处理器核心3300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3300(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3300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3300可以包括与多个子核心3301A-3301F耦合的固定功能块

3330,也称为子切片,其包括通用和固定功能逻辑的模块块。

[0514] 在至少一个实施例中,固定功能块3330包括几何/固定功能管线3336,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/固定功能管线3336可以由图形处理器3300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3336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。

[0515] 在固定的至少一个实施例中,功能块3330还包括图形SoC接口3337、图形微控制器3338和媒体管线3339。在固定的至少一个实施例中,图形SoC接口3337提供了图形核心3300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3338是可编程子处理器,其可配置为管理图形处理器3300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3339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3339经由对子核心3301-3301F内的计算或采样逻辑的请求来实现媒体操作。

[0516] 在至少一个实施例中,SoC接口3337使图形核心3300能够与通用应用处理器核心(例如,CPU)和/或SoC内的其他组件通信,包括存储器层次结构元素,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3337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3300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3337还可以实现用于图形核心3300的电源管理控制,并且启用图形核心3300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3337使得能够从命令流服务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3339,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3336、几何形状和固定功能管线3314)。

[0517] 在至少一个实施例中,图形微控制器3338可以配置为对图形核心3300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3338可以在子核心3301A-3301F中的执行单元(EU)阵列3302A-3302F、3304A-3304F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3300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服务器、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3338还可以促进图形核心3300的低功率或空闲状态,从而为图形核心3300提供在图形核心3300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。

[0518] 在至少一个实施例中,图形核心3300可以具有比所示的子核心3301A-3301F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3300还可以包括共享功能逻辑3310、共享和/或高速缓存存储器3312、几何/固定功能管线3314以及附加的固定功能逻辑3316以加速各种图形和计算处理操作。在至少一个实施例中,共享功能逻

辑3310可以包括可由图形核心3300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/或缓存存储器3312可以是图形核心3300内的N个子核心3301A-3301F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3314来代替固定功能块3330内的几何/固定功能管线3336,并且可以包括相同或相似的逻辑单元。

[0519] 在至少一个实施例中,图形核心3300包括附加的固定功能逻辑3316,其可以包括供图形核心3300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3316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3316、3336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3316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3316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。

[0520] 在至少一个实施例中,附加的固定功能逻辑3316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。

[0521] 在至少一个实施例中,在每个图形子核心3301A-3301F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3301A-3301F包括多个EU阵列3302A-3302F、3304A-3304F,线程分派和线程间通信(TD/IC)逻辑3303A-3303F,3D(例如,纹理)采样器3305A-3305F,媒体采样器3306A-3306F,着色器处理器3307A-3307F和共享本地存储器(SLM)3308A-3308F。EU阵列3302A-3302F、3304A-3304F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3303A-3303F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3305A-3305F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3306A-3306F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3301A-3301F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3301A-3301F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3308A-3308F,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。

[0522] 推理和/或训练逻辑215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2A和/或图2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中,推理和/或训练逻辑215的部分或全部可以被合并到图形处理器3310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3310、图形微控制器3338、几何和固定功能管线3314和3336或图32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2A或图2B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,所述寄存器配置图形处理器3300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。

[0523] 在至少一个实施例中,可以至少部分地在一个或更多个图形处理器3300上实现本文描述的客户端设备502,一个或更多个聚合服务器504,训练节点102,训练聚合器104,一个或更多个参数审查器1102和/或一个或更多个审查聚合器1110和/或方法中的一个或更多个。

[0524] 图34A-34B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3400。图34A示出了至少一个实施例,其中使用了线程执行逻辑3400。图34B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。

[0525] 如图34A中所示,在至少一个实施例中,线程执行逻辑3400包括着色器处理器3402、线程分派器3404、指令高速缓存3406、包括多个执行单元3408A-3408N的可缩放执行单元阵列、采样器3410、数据高速缓存3412和数据端口3414。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3408A、3408B、3408C、3408D、从3408N-1到3408N中的任何一个)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3400包括通过指令高速缓存3406、数据端口3414、采样器3410和执行单元3408A-3408N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3408A)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3408A-3408N的阵列可缩放以包括任意数量的单独执行单元。

[0526] 在至少一个实施例中,执行单元3408A-3408N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3402可以处理各种着色器程序并经由线程分派器3404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3404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3408A-3408N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3404还可以处理来自执行着色器程序的运行时线程产生请求。

[0527] 在至少一个实施例中,执行单元3408A-3408N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翻译即可执行。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3408A-3408N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3408A-3408N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。

[0528] 在至少一个实施例中, 执行单元3408A-3408N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3408A-3408N支持整数和浮点数据类型。

[0529] 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。

[0530] 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对于融合EU的线程控制逻辑 (3407A-3407N) 的融合执行单元3409A-3409N。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组中的每个EU可以配置为执行单独的SIMD硬件线程。融合的EU组中的EU的数量可以根据各种实施例而变化。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3409A-3409N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3409A包括第一EU 3408A、第二EU 3408B以及第一EU 3408A和第二EU 3408B共有的线程控制逻辑3407A。在至少一个实施例中, 线程控制逻辑3407A控制在融合图形执行单元3409A上执行的线程, 从而允许融合执行单元3409A-3409N内的每个EU使用公共指令指针寄存器来执行。

[0531] 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3406) 被包括在线程执行逻辑3400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3412) 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3410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3410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。

[0532] 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3400。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3402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3402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3402经由线程分派器3404将线程分派到执行单元(例如3408A)。在至少一个实施例中,着色器处理器3402使用采样器3410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,或者丢弃一个或多个像素以进行进一步处理。

[0533] 在至少一个实施例中,数据端口3414提供了一种用于线程执行逻辑3400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3414包括或耦合到一个或多个高速缓存存储器(例如,数据高速缓存3412)以高速缓存数据以便经由数据端口进行存储器访问。

[0534] 如图34B所示,在至少一个实施例中,图形执行单元3408可以包括指令获取单元3437、通用寄存器文件阵列(GRF)3424、架构寄存器文件阵列(ARF)3426、线程仲裁器3422、发送单元3430、分支单元3432、一组SIMD浮点单元(FPU)3434,以及在至少一个实施例中,一组专用整数SIMD ALU3435。在至少一个实施例中,GRF 3424和ARF 3426包括一组与可以在图形执行单元3408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 3426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 3424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 3426中的线程专用寄存器中。

[0535] 在至少一个实施例中,图形执行单元3408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。

[0536] 在至少一个实施例中,图形执行单元3408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3408的线程仲裁器3422可以将指令分派到发送单元3430、分支单元3442或SIMD FPU 3434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 3424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 3424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 3424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。

[0537] 在至少一个实施例中,经由消息传递发送单元3430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3432促进SIMD发散和最终收敛。

[0538] 在至少一个实施例中,图形执行单元3408包括一个或更多个SIMD浮点单元(FPU) 3434,以执行浮点操作。在至少一个实施例中,FPU 3434还支持整数计算。在至少一个实施例中,FPU 3434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,FPU中的至少一个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 3435,并且可以被专门优化以执行与机器学习计算相关的操作。

[0539] 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3408的多个实例的阵列。在至少一个实施例中,执行单元3408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3408上执行的每个线程在不同的通道上执行。

[0540] 推理和/或训练逻辑215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2A和/或图2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中,推理和/或训练逻辑215的部分或全部可以被结合到执行逻辑3400中。此外,在至少一个实施例中,可以使用除了图2A或图2B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3400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。

[0541] 在至少一个实施例中,可以至少部分地使用一个或更多个执行逻辑3400实现本文描述的客户端设备502,一个或更多个聚合服务器504,训练节点102,训练聚合器104,一个或更多个参数审查器1102和/或一个或更多个审查聚合器1110和/或方法中的一个或更多个。

[0542] 图35示出了根据至少一个实施例的并行处理单元(“PPU”) 3500。在至少一个实施例中,PPU 3500配置有机器可读代码,该机器可读代码如果由PPU 3500执行,则使得PPU 3500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 3500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 3500执行的一组指令的实例。在至少一个实施例中,PPU 3500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道,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 3500用于执行计算,诸如线性代数运算和机器学习运算。图35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容设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。

[0543] 在至少一个实施例中,一个或更多个PPU 3500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 3500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。

[0544] 在至少一个实施例中,PPU 3500包括但不限于输入/输出(“I/O”)单元3506、前端单元3510、调度器单元3512、工作分配单元3514、集线器3516、交叉开关(“Xbar”)3520、一个或更多个通用处理集群(“GPC”)3518和一个或更多个分区单元(“存储器分区单元”)3522。在至少一个实施例中,PPU 3500通过一个或更多个高速GPU互连(“GPU互连”)3508连接到主机处理器或其他PPU 3500。在至少一个实施例中,PPU 3500通过互连3502连接到主机处理器或其他外围设备。在一实施例中,PPU 3500连接到包括一个或更多个存储器设备(“存储器”)3504的本地存储器。在至少一个实施例中,存储器设备3504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。

[0545] 在至少一个实施例中,高速GPU互连3508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 3500(“CPU”),支持PPU 3500和CPU之间的缓存一致性以及CPU主控。在至少一个实施例中,高速GPU互连3508通过集线器3516将数据和/或命令传输到PPU 3500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35中可能未明确示出的其他组件。

[0546] 在至少一个实施例中,I/O单元3506配置为通过系统总线3502从主机处理器(图35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3506直接通过系统总线3502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3506可以经由系统总线3502与一个或更多个其他处理器(例如一个或更多个PPU 3500)通信。在至少一个实施例中,I/O单元3506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3506实现用于与外部设备通信的接口。

[0547] 在至少一个实施例中,I/O单元3506对经由系统总线3502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 3500执行各种操作的命令。在至少一个实施例中,I/O单元3506如命令所指定的那样将解码的命令发送到PPU 3500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3510和/或被发送到集线器3516或PPU 3500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35中未明确示出)。在至少一个实施例中,I/O单元3506配置为在PPU 3500的各种逻辑单元之间路由通信。

[0548] 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 3500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 3500两者访问(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3506通过系统总线3502传输的存储器请求连接到系统总线3502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 3500,使得前端单元3510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 3500的各个单元。

[0549] 在至少一个实施例中,前端单元3510耦合到调度器单元3512,该调度器单元3512配置各种GPC 3518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3512配置为跟踪与调度器单元3512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 3518,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3512管理在一个或更多个GPC 3518上执行的多个任务。

[0550] 在至少一个实施例中,调度器单元3512耦合到工作分配单元3514,该工作分配单元3514配置为分派任务以在GPC 3518上执行。在至少一个实施例中,工作分配单元3514跟踪从调度器单元3512接收到的多个调度任务并且工作分配单元3514管理每个GPC 3518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 3518处理的任务;活跃任务池可包括用于由GPC 3518主动处理的任务的多个时隙(例如4个时隙),以使随着GPC 3518中的一个完成任务的执行,该任务将从GPC 3518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之一,并安排其在GPC 3518上执行。在至少一个实施例中,如果活跃任务在GPC 3518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 3518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 3518上执行。

[0551] 在至少一个实施例中,工作分配单元3514经由XBar3520与一个或更多个GPC 3518通信。在至少一个实施例中,XBar3520是互连网络,其将PPU 3500的许多单元耦合到PPU 3500的其他单元,并且可以配置为将工作分配单元3514耦合到特定的GPC3518。在至少一个实施例中,一个或更多个PPU 3500的其他单元也可以通过集线器3516连接到XBar3520。

[0552] 在至少一个实施例中,任务由调度器单元3512管理,并由工作分配单元3514分配给GPC 3518之一。GPC3518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 3518中的其他任务消耗,通过XBar3520路由到不同的GPC 3518或存储在存储器3504中。在至少一个实施例中,结果可以通过分区单元3522写到存储器3504中,其实现了用于向存储器3504写入数据或从存储器3504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3508传输到另一PPU 3504或CPU。在至少一个实施例中,PPU 3500包括但不限于U个分区单元3522,其等于耦合到PPU 3500的分离且不同的存储设备3504的数量。在至少一个实施例中,下面结合图37更详细地描述分区单元3522。

[0553] 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 3500上执行。在一个实施例中,多个计算应用由PPU 3500同时执行,并且PPU 3500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 3500执行,并且驱动器核心将任务输出至由PPU 3500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。在至少一个实施例中,结合图37根据至少一个实施例更详细地描述了线程和协作线程。

[0554] 推理和/或训练逻辑215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。下面结合图2A和/或2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中,深度学习应用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 3500的信息。在至少一个实施例中,深度学习应用处理器3500用于基于已由另一处理器或系统或PPU 3500经训练的经训练的机器学习模型(例如,神经网络)推理或预测信息。在至少一个实施例中,PPU 3500可用于执行本文所述的一个或多个神经网络用例。

[0555] 在至少一个实施例中,可以至少部分地使用一个或多个PPU 3500实现本文描述的客户端设备502,一个或多个聚合服务器504,训练节点102,训练聚合器104,一个或多个参数审查器1102和/或一个或多个审查聚合器1110和/或方法中的一个或多个。

[0556] 图36示出了根据至少一个实施例的通用处理集群(“GPC”)3600。在至少一个实施例中,GPC 3600是图35的GPC 3518。在至少一个实施例中,每个GPC 3600包括但不限于用于处理任务的多个硬件单元,并且每个GPC 3600包括但不限于管线管理器3602、预光栅操作单元(“PROP”)3604、光栅引擎3608、工作分配交叉开关(“WDX”)3616、存储器管理单元(“MMU”)3618、一个或多个数据处理集群(“DPC”)3606,以及部件的任何合适组合。

[0557] 在至少一个实施例中,GPC 3600的操作由管线管理器3602控制。在至少一个实施例中,管线管理器3602管理一个或多个DPC 3606的配置,以处理分配给GPC 3600的任务。在至少一个实施例中,管线管理器3602配置一个或多个DPC 3606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 3606配置为在可编程流式多处理器(“SM”)3614上执行顶点着色器程序。在至少一个实施例中,管线管理器3602配置为将从工作分配单元接收的数据包路由到GPC 3600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 3604和/或光栅引擎3608中的固定功能硬件单元,而可以将其他数据包路由到DPC 3606以由原始引擎3612进行处理。在至少一个实施例中,管线管理器3602配置DPC 3606中的至少一个以实现神经网络模型和/或计算管线。

[0558] 在至少一个实施例中,PROP单元3604配置为在至少一个实施例中将由光栅引擎3608和DPC 3606生成的数据路由到分区单元3522中的光栅操作(“ROP”)单元,上面结合图35更详细地描述。在至少一个实施例中,PROP单元3604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3608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3608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3608的输出包括将由任何适当的实体(例如,由在DPC 3606内实现的片段着色器)处理的片段。

[0559] 在至少一个实施例中,包括在GPC 3600中的每个DPC 3606包括但不限于M管道控制器(“MPC”)3610;图元引擎3612;一个或多个SM 3614;及其任何合适的组合。在至少一

个实施例中, MPC 3610控制DPC 3606的操作, 将从管线管理器3602接收的分组路由到DPC 3606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3612, 图元引擎3612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 3614。

[0560] 在至少一个实施例中, SM 3614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 3614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 3614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的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。下面更详细地描述SM 3614的至少一个实施例。

[0561] 在至少一个实施例中, MMU 3618在GPC 3600和存储器分区单元 (例如, 图35的分区单元3522) 之间提供接口, 并且MMU 3618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 3618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。

[0562] 推理和/或训练逻辑215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。下面结合图2A和/或图2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中, 深度学习应用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 3600的信息。在至少一个实施例中, GPC 3600用于基于已由另一处理器或系统或GPC 3600经训练的机器学习模型 (例如, 神经网络) 推理或预测信息。在至少一个实施例中, GPC 3600可用于执行本文所述的一个或多个神经网络用例。

[0563] 在至少一个实施例中, 可以至少部分地使用一个或多个GPC 3600上实现本文所述的客户端设备502, 一个或多个聚合服务器504, 训练节点102, 训练聚合器104, 一个或多个参数审查器1102和/或一个或多个审查聚合器1110和/或方法中的一个或多个。

[0564] 图37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3700。在至少一个实施例中, 存储器分区单元3700包括但不限于光栅操作 (“ROP”) 单元3702; 二级 (“L2”) 高速缓存3704; 存储器接口3706; 及其任何合适的组合。存储器接口3706耦合到存储器。存储器接口3706可以实现32、64、128、1024位数据总线, 或者类似的用于高速数据传输。在至少一个实施例中, PPU包括U个存储器接口3706, 每对分区单元3700一个存储器接口3706, 其中每对分区单元3700连接到相应的存储器设备。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同步动态随机地址存储器 (“GDDR5 SDRAM”)。

[0565] 在至少一个实施例中, 存储器接口3706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存储器堆栈与PPU位于同一物理封装上,与其他GDDR5 SDRAM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(“SECCDED”)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提供更高的可靠性。

[0566] 在至少一个实施例中,PPU实现了多级存储器层次结构。在至少一个实施例中,存储器分区单元3700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3508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。

[0567] 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3700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面,并且复制过程是透明的。

[0568] 根据至少一个实施例,来自图35的存储器3504或其他系统存储器的数据由存储器分区单元3700获取,并将其存储在L2高速缓存3704中,L2高速缓存3704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3700包括但不限于与相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 3614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 3614的私有存储器,并且从L2高速缓存3704中获取数据并将其存储在每个L1高速缓存中,用于在SM 3614的功能单元中进行处理。在至少一个实施例中,L2高速缓存3704耦合到存储器接口3706和XBar3520。

[0569] 在至少一个实施例中,ROP单元3702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3702结合光栅引擎3608实施深度测试,从光栅引擎3608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,如果片段通过了针对样本位置的深度测试,则ROP单元3702更新深度缓冲区,并将深度测试的结果发送给光栅引擎3608。将意识到,分区单元3700的数量可以不同于光栅单元400的数量,因此,可以在至少一个实施例中将每个ROP单元3702耦合到每个GPC。在至少一个实施例中,ROP单元3702追踪从不同GPC接收到的分组,并确定通过XBar3520将ROP单元3702产生的结果路由到哪个。

[0570] 图38示出了根据至少一个实施例的流式多处理器(“SM”)3800。在至少一个实施例中,SM 3800是图36的SM。在至少一个实施例中,SM 3800包括但不限于指令高速缓存3802;一个或更多个调度器单元3804;寄存器文件3808;一个或更多个处理核心(“核心”)3810;一个或更多个特殊功能单元(“SFU”)3812;一个或更多个加载/存储单元(“LSU”)3814;互连网

络3816;共享存储器/一级(“L1”)高速缓存3818;及其任何合适的组合。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 3800之一。在至少一个实施例中,调度器单元3804从工作分配单元接收任务并管理分配给SM 3800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3804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3804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3810、SFU 3812和LSU 3814)。

[0571] 在至少一个实施例中,合作组可以指用于组织通信线程组的编程模型,其允许开发人员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,其他编程模型的应用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。

[0572] 在至少一个实施例中,调度单元3806配置为将指令发送到功能单元中的一个或更多个,并且调度器单元3804包括但不限于两个调度单元3806,该两个调度单元3806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3804包括单个调度单元3806或附加调度单元3806。

[0573] 在至少一个实施例中,每个SM 3800在至少一个实施例中包括但不限于寄存器文件3808,该寄存器文件3808为SM 3800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3808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3808的专用部分。在至少一个实施例中,寄存器文件3808在由SM 3800执行的不同线程束之间划分,并且寄存器文件3808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 3800包括但不限于多个L个处理核心3810。在至少一个实施例中,SM 3800包括但不限于大量(例如128个或更多)不同的处理核心3810。在至少一个实施例中,每个处理核心3810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 754-1608标准。在至少一个实施例中,处理核心3810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。

[0574] 在至少一个实施例中,张量核心配置为执行矩阵运算。在至少一个实施例中,一个

或更多个张量核心包括在处理核心3810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。

[0575] 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,张量核心对具有32位浮点累积的16位浮点输入数据进行操作。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的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。

[0576] 在至少一个实施例中,每个SM 3800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 3812。在至少一个实施例中,SFU 3812包括但不限于配置为遍历分层树数据结构的树遍历单元。在至少一个实施例中,SFU 3812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 3800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3818中。在至少一个实施例中,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。在至少一个实施例中,每个SM 3800包括但不限于两个纹理单元。

[0577] 在至少一个实施例中,每个SM 3800包括但不限于实现共享存储器/L1高速缓存3818与寄存器文件3808之间的加载和存储操作的N个LSU 3814。在至少一个实施例中,每个SM 3800包括但不限于互连网络3816,互连网络3816将每个功能单元连接到寄存器文件3808,并且LSU 3814连接到寄存器文件3808和共享存储器/L1高速缓存3818。在至少一个实施例中,互连网络3816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3808中的任何寄存器,并且将LSU 3814连接到寄存器文件3808和共享存储器/L1高速缓存3818中的存储器位置。

[0578] 在至少一个实施例中,共享存储器/L1高速缓存3818是片上存储器的阵列,其在至少一个实施例中允许SM 3800与图元引擎之间以及SM 3800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3818包括但不限于128KB的存储容量,并且位于从SM 3800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3818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3818、L2高速缓存和存储器中的一个或更多个是后备存储器。

[0579] 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,则纹理和加载/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3818内的集成使共享存储器/L1高速缓存3818能够用作用于流传输数据的高吞吐量管道,

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 3800执行程序并执行计算,使用共享存储器/L1高速缓存3818在线程之间进行通信,以及使用LSU 3814通过共享存储器/L1高速缓存3818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 3800向调度器单元3804写入可以用来在DPC上启动新工作的命令。

[0580] 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。

[0581] 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。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,PPU可以是包括在主板的芯片组中的集成图形处理单元(“iGPU”)。

[0582] 推理和/或训练逻辑215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。下面结合图2A和/或2B提供关于推理和/或训练逻辑215的细节。在至少一个实施例中,深度学习应用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 3800的信息。在至少一个实施例中,SM 3800用于基于已由另一处理器或系统或由SM 3800经训练的机器学习模型(例如,神经网络)推理或预测信息。在至少一个实施例中,SM 3800可用于执行一个或更多个本文所述的神经网络用例。

[0583] 在至少一个实施例中,可以至少部分地使用一个或更多个SMU3800上实现本文描述的客户端设备502,一个或更多个聚合服务器504,训练节点102,训练聚合器104,一个或更多个参数审查器1102和/或一个或更多个审查聚合器1110和/或方法中的一个或更多个。

[0584] 在至少一个实施例中,单个半导体平台可以指唯一的单一的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,并且相对于利用中央处理单元(“CPU”)和总线实施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。

[0585] 在至少一个实施例中,将以机器可读的可执行代码或计算机控制逻辑算法的形式计算机程序存储在主存储器1804和/或辅助存储器中。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1800能够执行各种功能。在至少一个实施例中,存储器1804、存储器和/或任何其他存储器是计算机可读介质的可能示例。在至少一个实施例中,辅助存储器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,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,各种先前附图的架构和/或功能是

在CPU 1802;并行处理系统1812;能够具有至少两个CPU 1802的能力的至少一部分的集成电路;并行处理系统1812;芯片组(例如,一组设计成工作并作为执行相关功能的单元出售的集成电路等);以及集成电路的任何适当组合的环境中实现的。

[0586] 在至少一个实施例中,各种先前附图的架构和/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1800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。

[0587] 在至少一个实施例中,并行处理系统1812包括但不限于多个并行处理单元(“PPU”)1814和相关联的存储器1816。在至少一个实施例中,PPU 1814通过互连1818和开关1820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1812在可并行化的PPU 1814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 1814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 1814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 1814的操作,其中块中的所有线程(例如,跨多个PPU 1814执行)在进行之前到达某个代码执行点。

[0588] 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,但是其某些示出的实施例在附图中示出并且已经在上面进行了详细描述。但是,应当理解,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公开内容的精神和范围内的所有修改、替代构造和等同物。

[0589] 除非另有说明,除非另有说明或显然与环境矛盾,否则在描述所公开的实施例的环境中(特别是在所附权利要求的环境中)对术语“一”,“一个”和“该”以及类似指代的使用应解释为涵盖单数和复数,并且不作为术语的定义。术语“包含”,“具有”,“包括”和“内含”应解释为开放式术语(意思是“包括但不限于”)。术语“连接”在未经修改时指的是物理连接,应理解为部分或全部包含在,连接到或连接在一起的部分或全部,即使有任何介入。除非在此另外指出,否则本文中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速记方法,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。除非环境另外指出或矛盾,否则术语“组”(例如“一组项目”)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,除非环境另外指出或矛盾,否则相应集合的术语“子集”不一定表示相应集合的适当子集,而是子集和相应集合可以相等。

[0590] 除非以其他方式明确指出或与环境明显矛盾,否则诸如“A、B和C中的至少一个”或“A、B和C的至少一个”形式的词组等联合语言在环境中理解为通常用来表示项目,术语等可以是A或B或C,也可以是A和B和C集合的任何非空子集。例如,在具有三个成员,连接短语“A、B和C中的至少一个”和“A、B和C的至少一个”是指以下任意集合: {A}, {B}, {C}, {A, B}, {A, C}, {B, C}, {A, B, C}。因此,这种联合语言通常不意图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,除非另有说明或与环境矛盾,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)。复数是至少两个项目,但是当明确地或

通过环境指示时可以是多个。此外,除非另有说明或从环境中清楚得知,否则短语“基于”是指“至少部分基于”而不是“仅基于”。

[0591] 除非本文另外指出或与环境明显矛盾,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)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)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,代码例如以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,其不包括暂时性信号(例如,传播的瞬态电或电磁传输),但包括暂时性信号的收发器中的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(或其他存储器以存储可执行指令)的一组一个或更多个非暂时性计算机可读存储介质上,当由计算机系统的一个或更多个处理器执行(即,由于被执行)而导致的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以及缺少所有代码的多个非暂时性计算机可读存储介质的一个或更多个单个非暂时性存储介质,而多个非暂时性计算机可读存储介质共同存储所有代码。在至少一个实施例中,执行可执行指令,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。

[0592] 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描述的过程的操作的一个或更多个服务,并且这样的计算机系统配置有能够实现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,是一种分布式计算机系统,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。

[0593] 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,除非另有声明。说明书中的任何语言都不应被解释为表示对于实施公开必不可少的任何未要求保护的要素。

[0594] 本文引用的所有参考文献,包括出版物、专利申请和专利,均以引用的方式并入本文,如同每个参考文献被单独且具体地指出以引用的方式并入本文一样。

[0595] 在描述和权利要求中,可以使用术语“耦合”和“连接”及其派生词。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”可用于指示两个或更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更多个元素彼此不直接接触,但仍彼此协作或交互。

[0596] 除非另外特别说明,否则可以理解,在整个说明书中,诸如“处理”、“计算”、“运算”、“确定”等,是指计算机或计算系统的动作和/或过程。或类似的电子计算设备,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。

[0597] 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设备的一部分。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”过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。而且,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。因为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,术语“系统”和“方法”在本文中可互换使用。

[0598] 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。可以以多种方式来完成获得、获取、接收过程或输入模拟和数字数据,例如通过接收作为函数调用或对应用程序接口的调用的参数的数据。在一些实施方式中,获得、获取、接收或输入模拟或数字数据的过程可以通过经由串行或并行接口传输数据来完成。在另一实施方式中,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、传输、发送或呈现模拟或数字数据。在各种示例中,提供、输出、传输、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。

[0599] 尽管上面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,并且意图在本公开的范围内。此外,尽管出于讨论目的在上面定义了具体的责任分配,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。

[0600] 此外,尽管已经用针对结构特征和/或方法动作的语言描述了主题,但是应该理解,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。

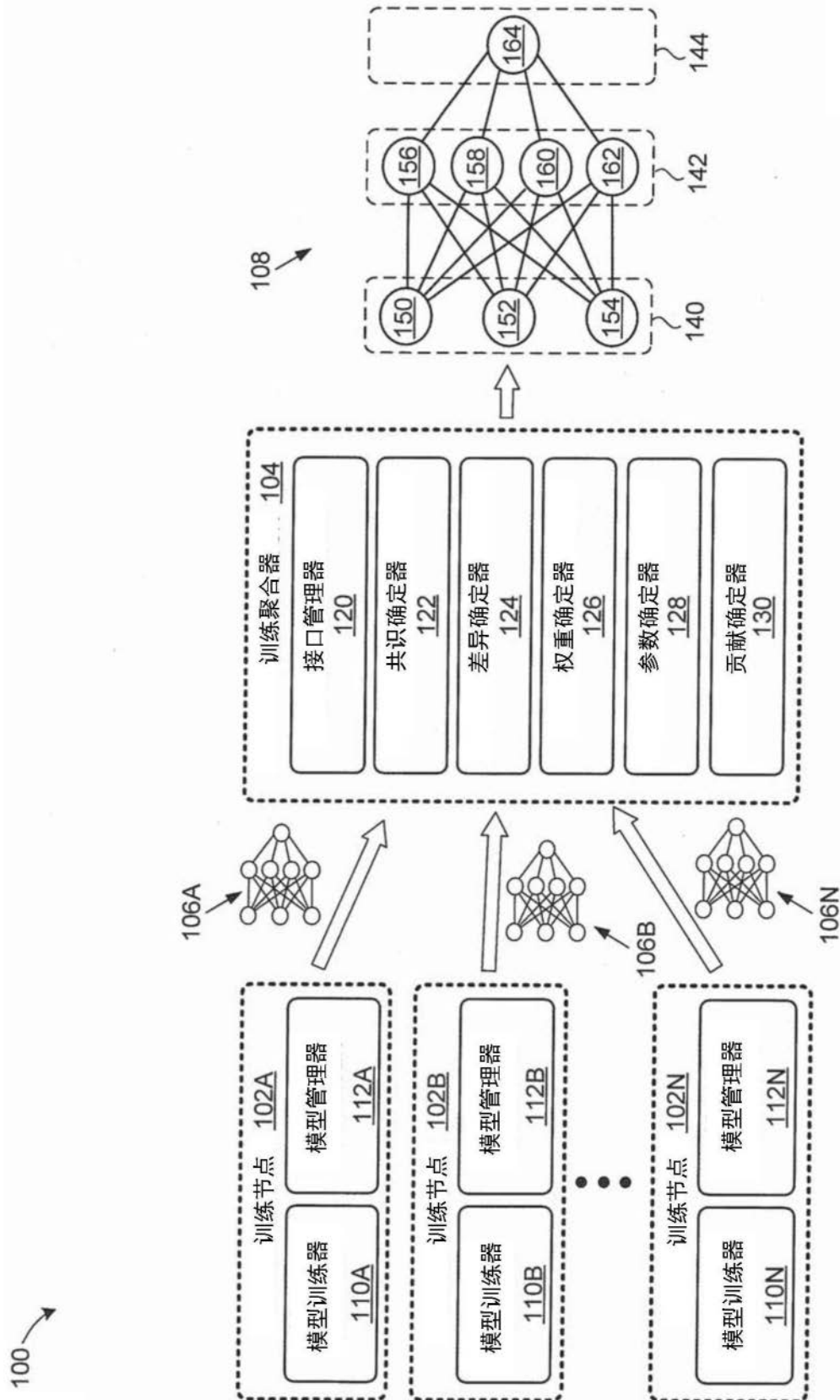


图1A

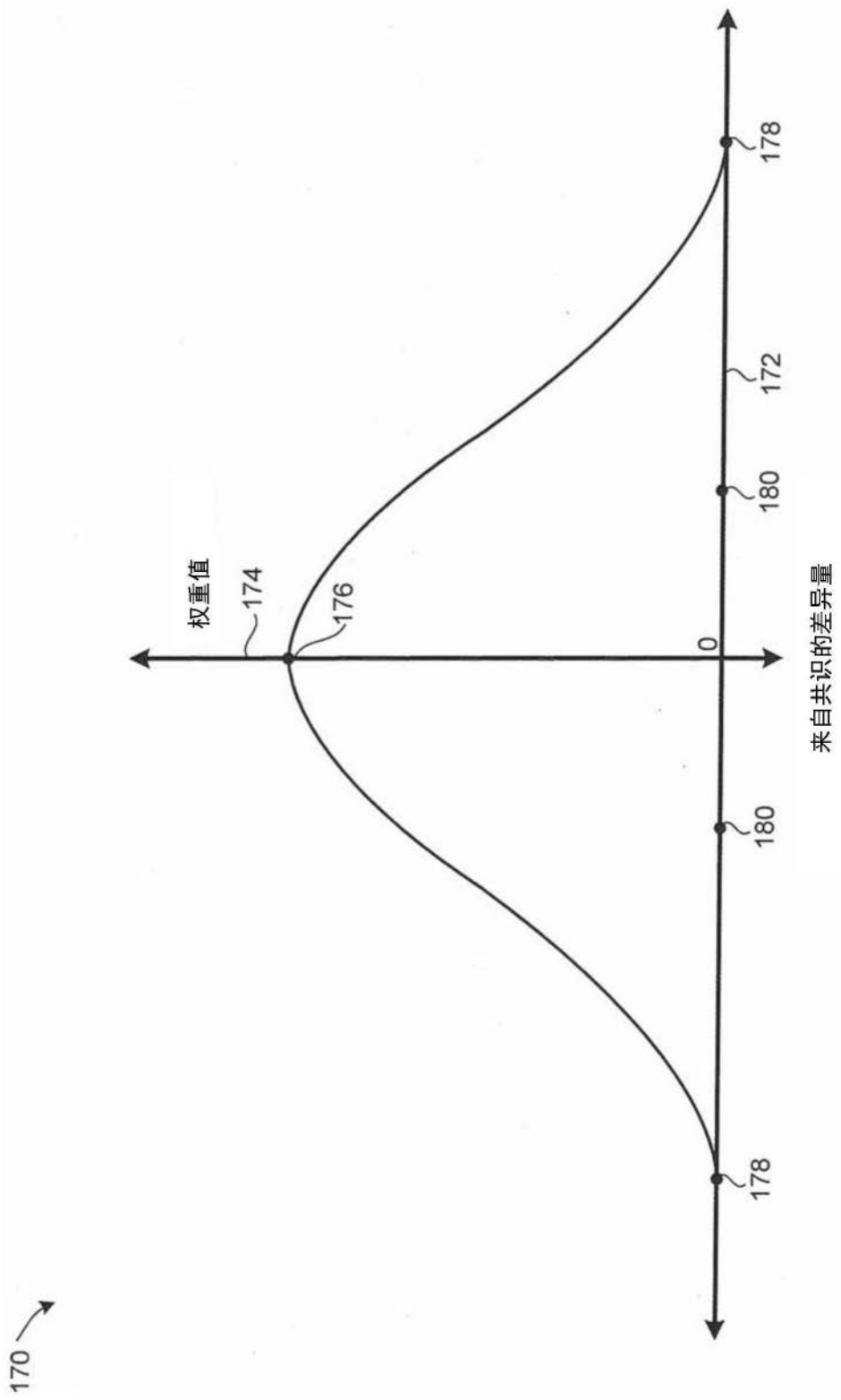


图1B

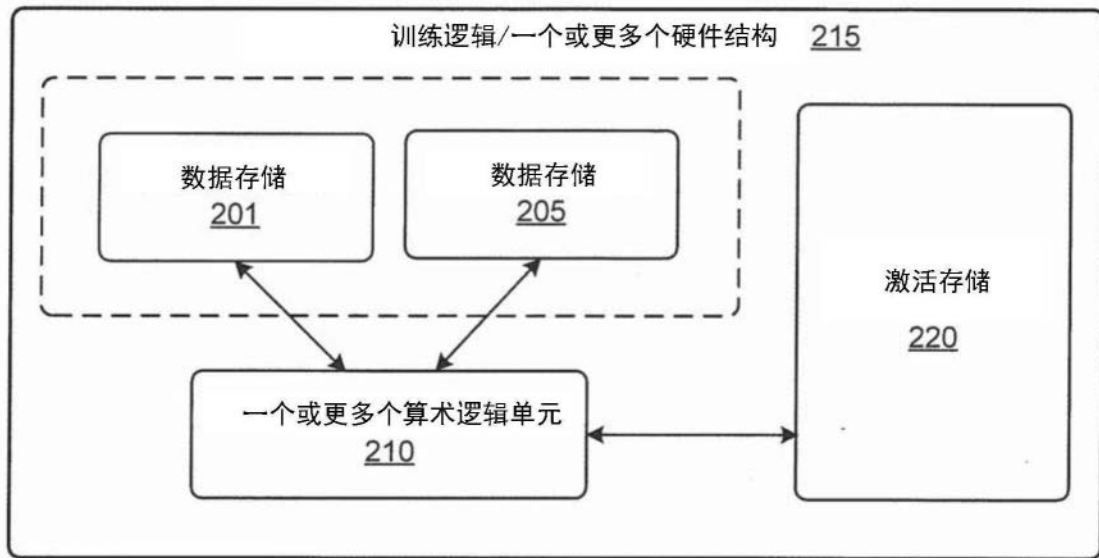


图2A

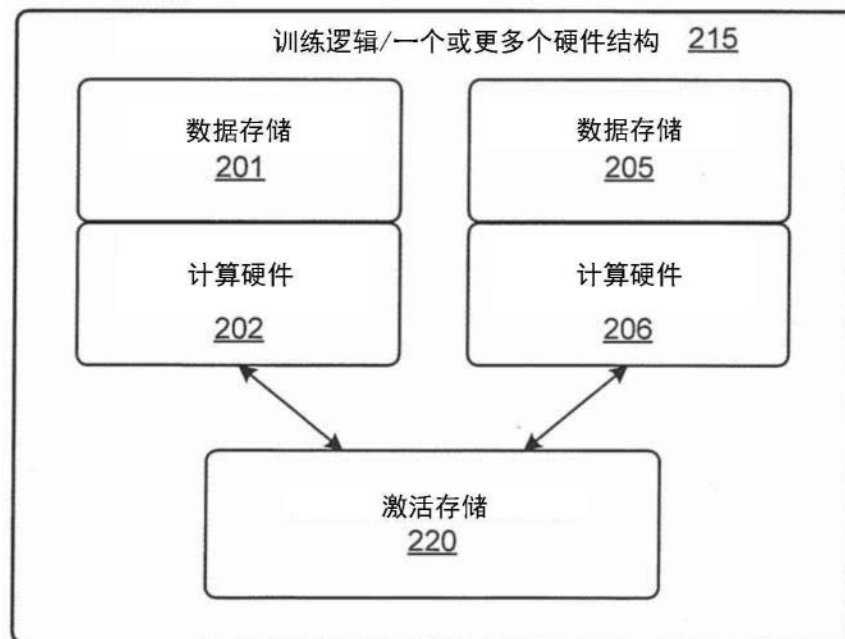


图2B

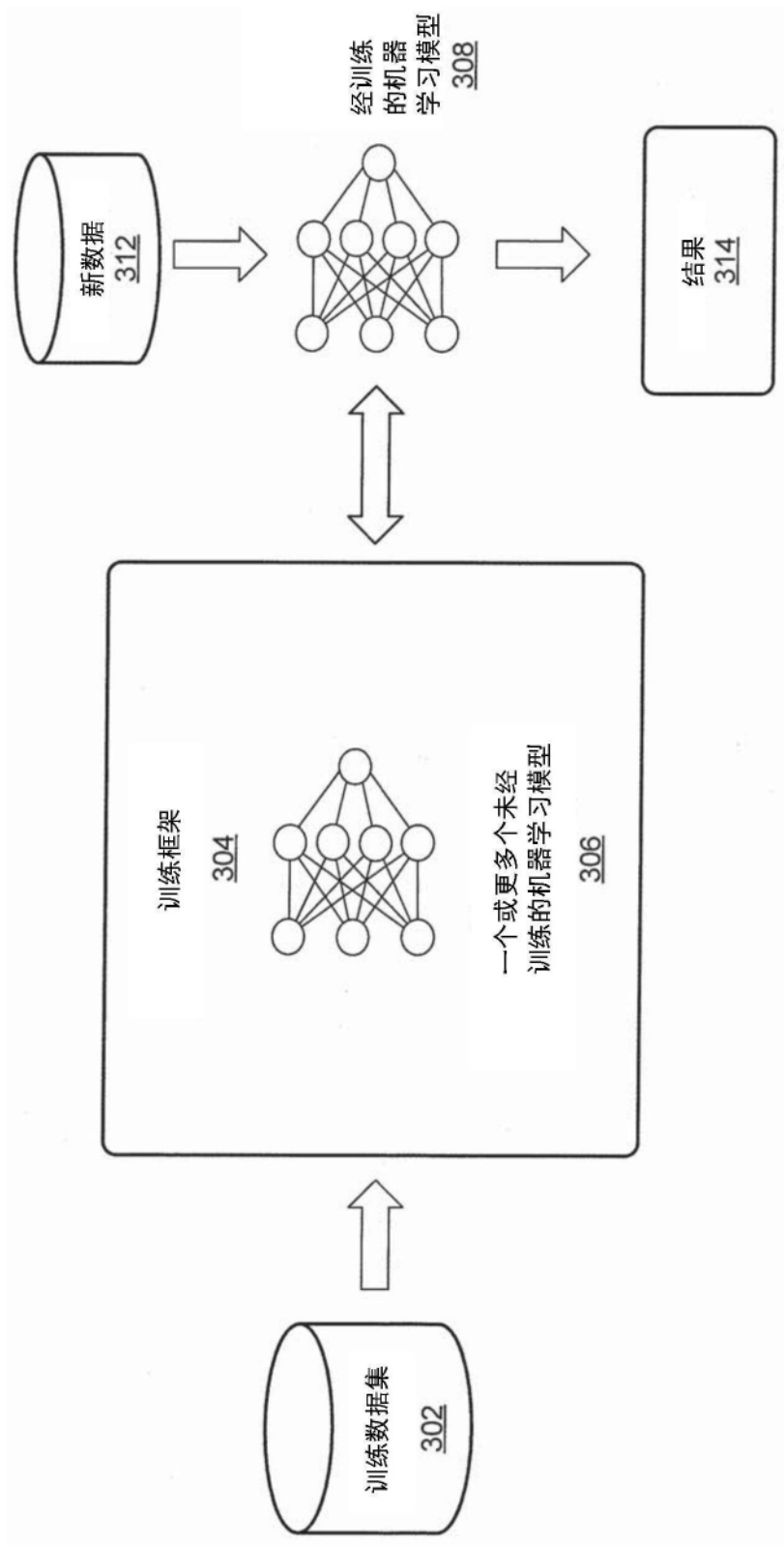


图3

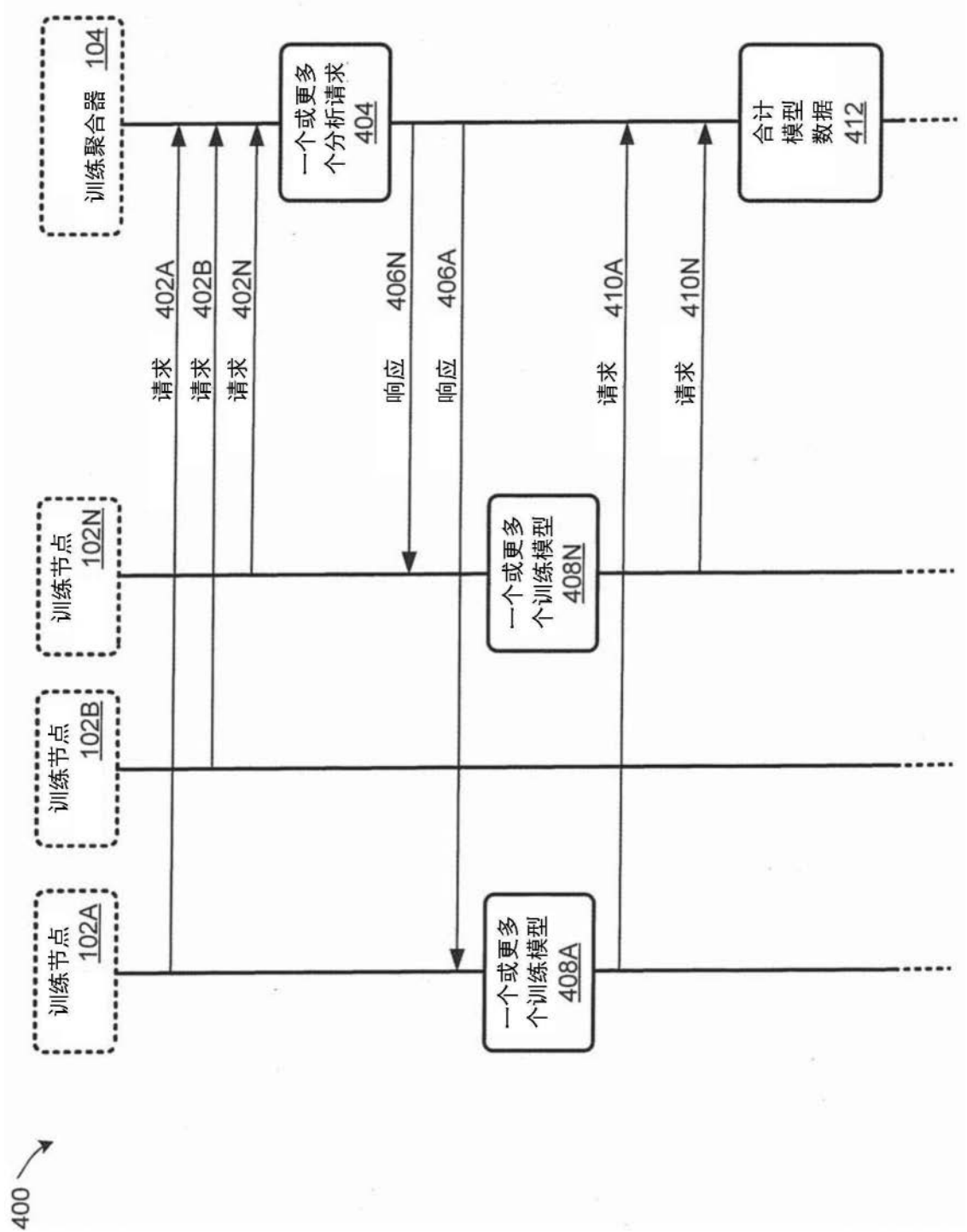


图4

500

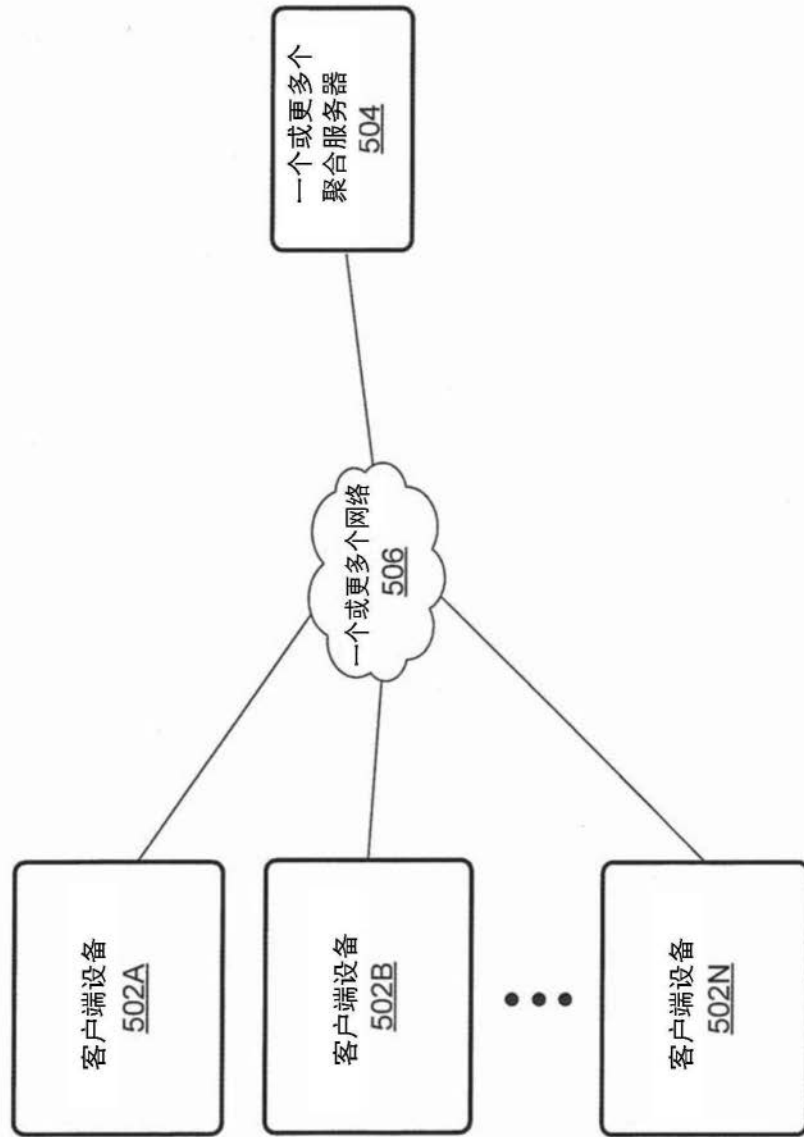


图5

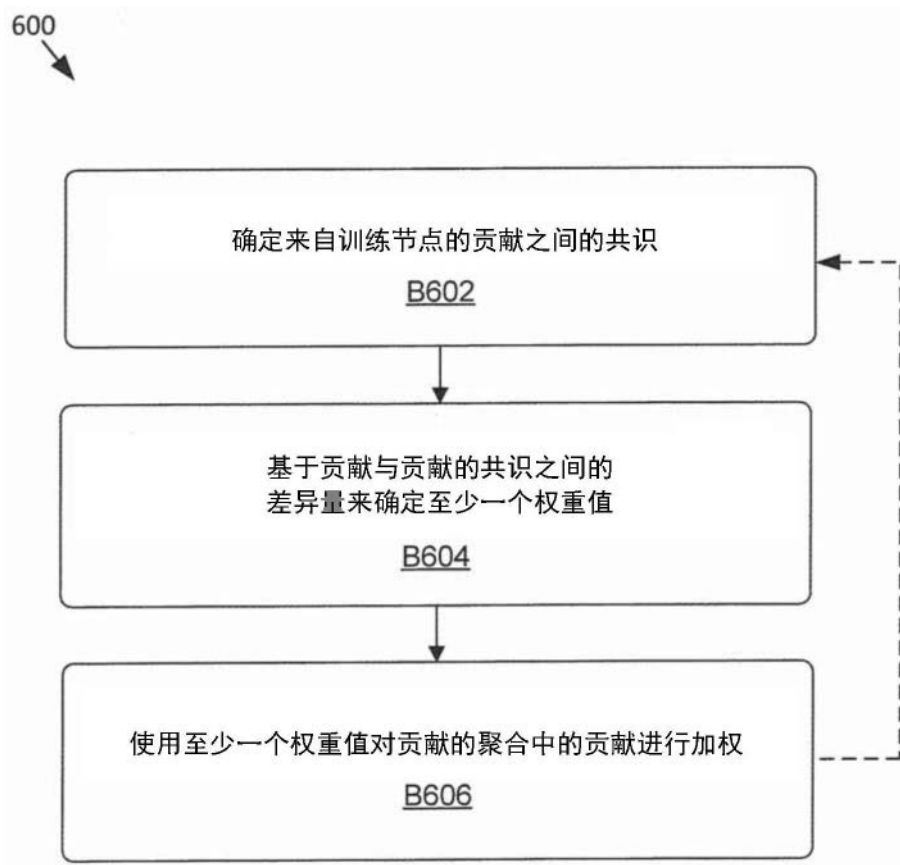


图6

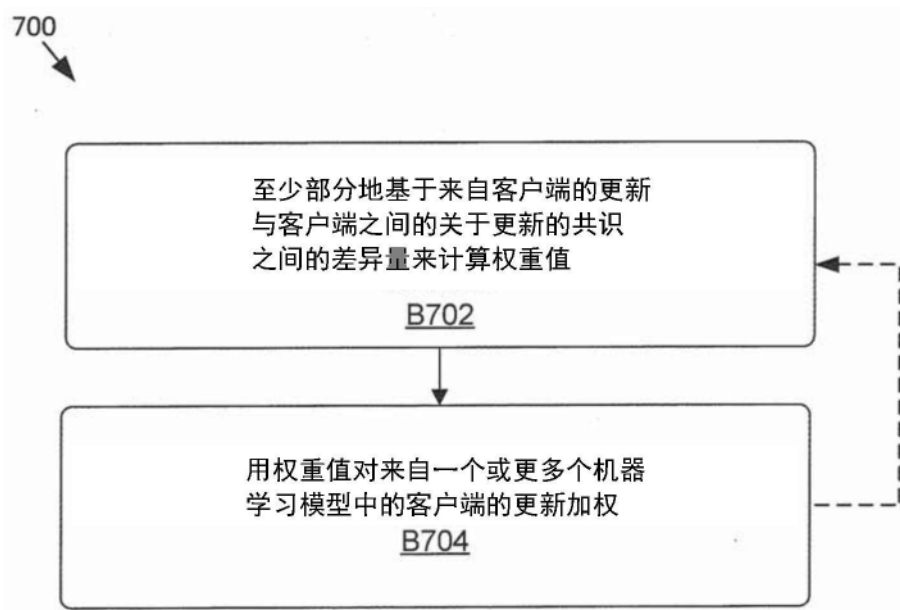


图7

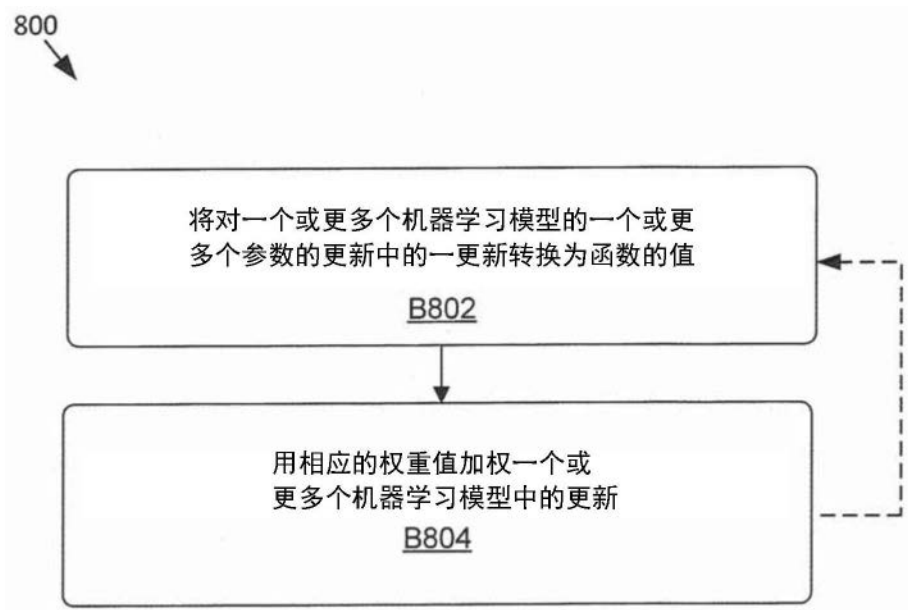


图8

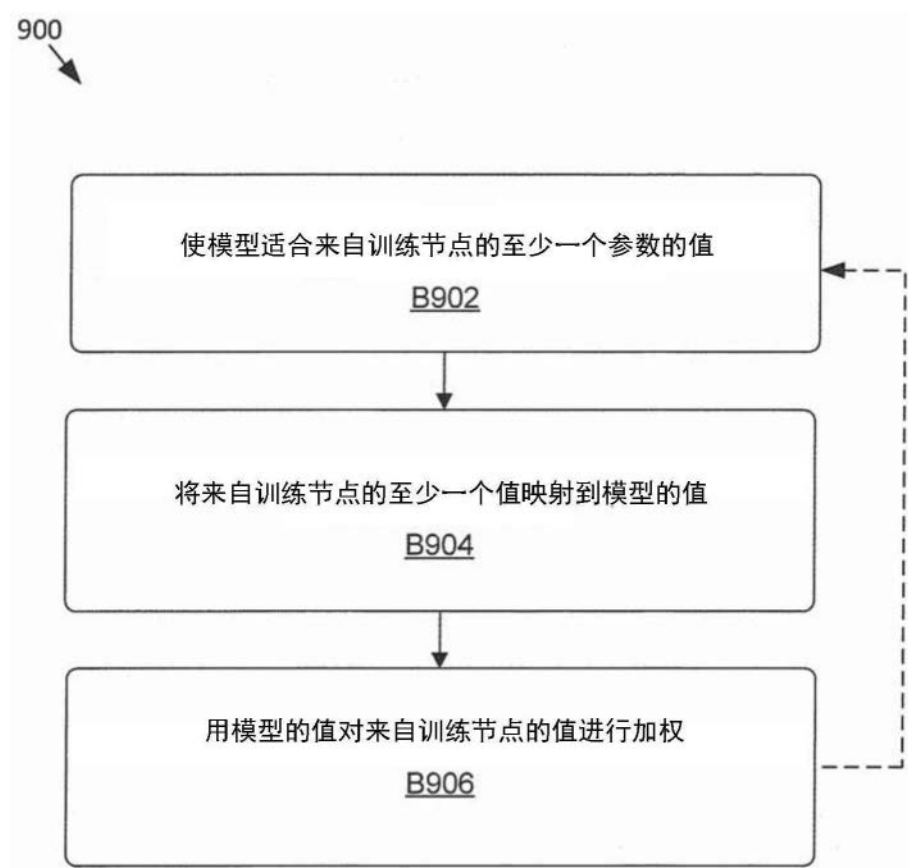


图9

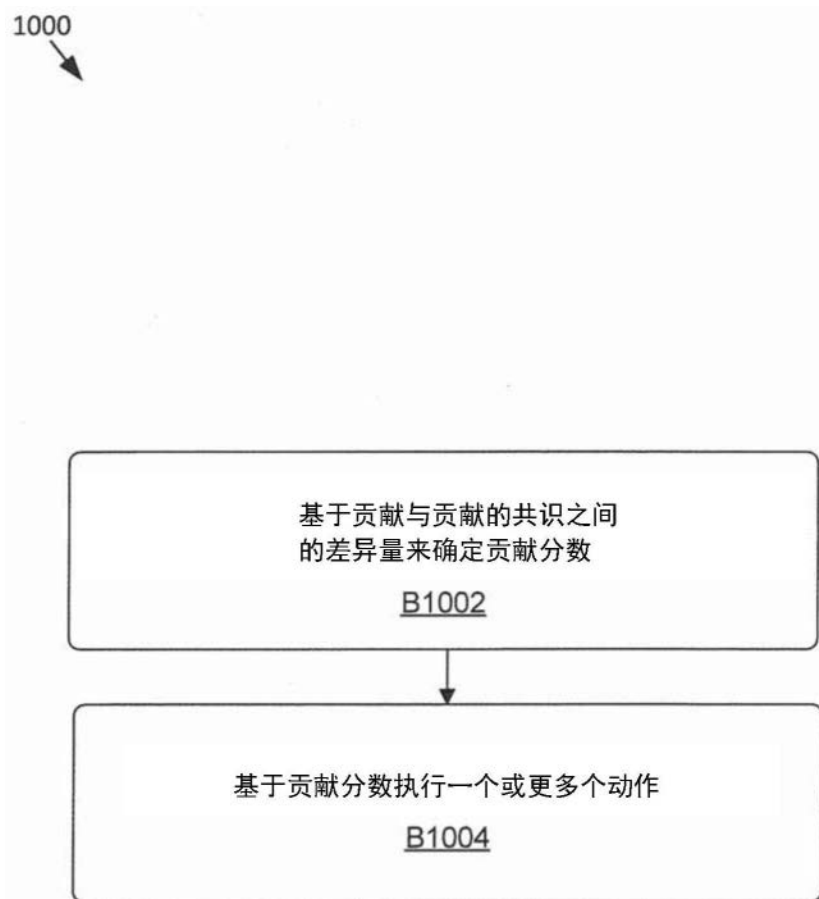


图10

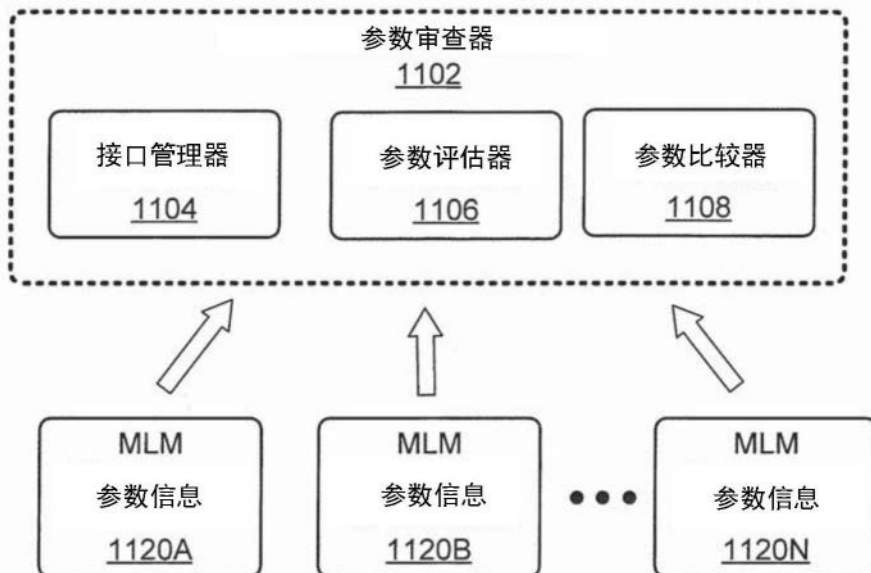


图11A

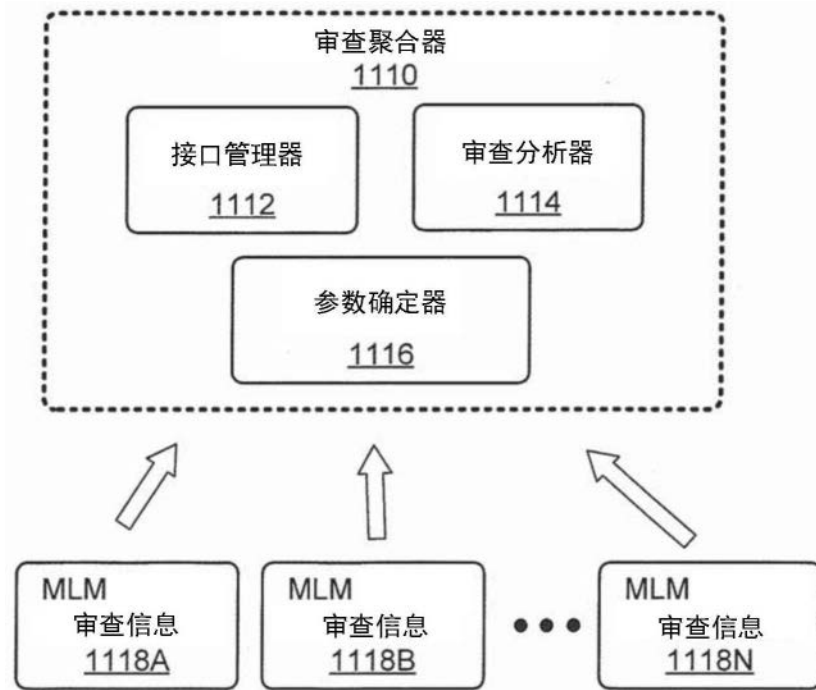


图11B

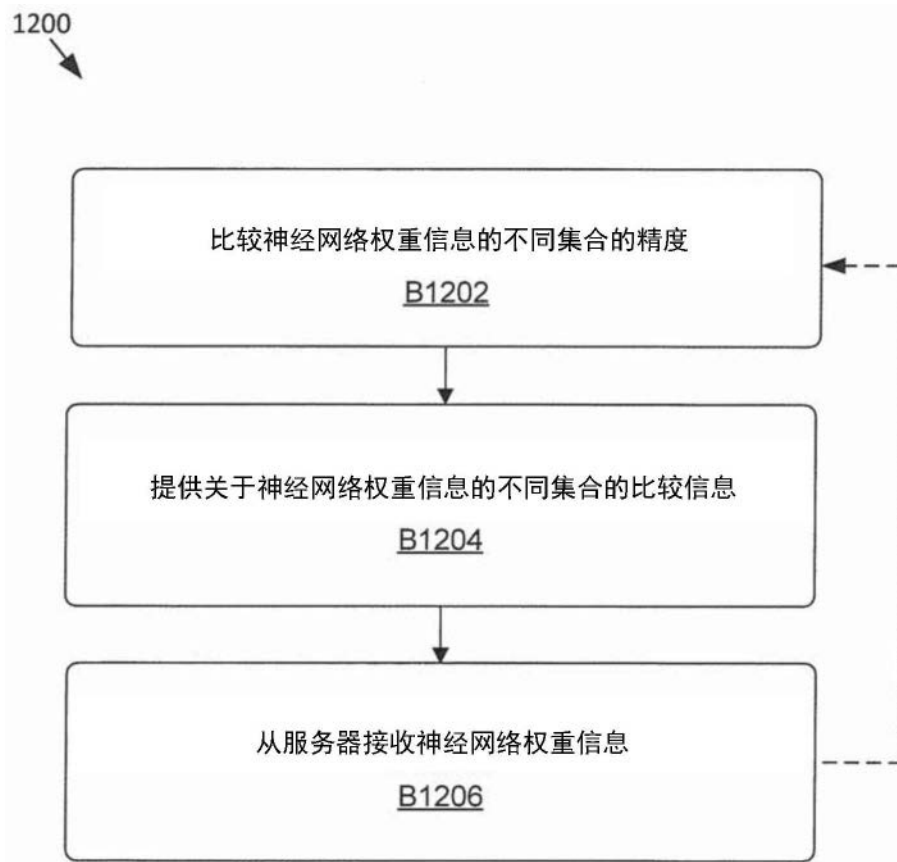


图12

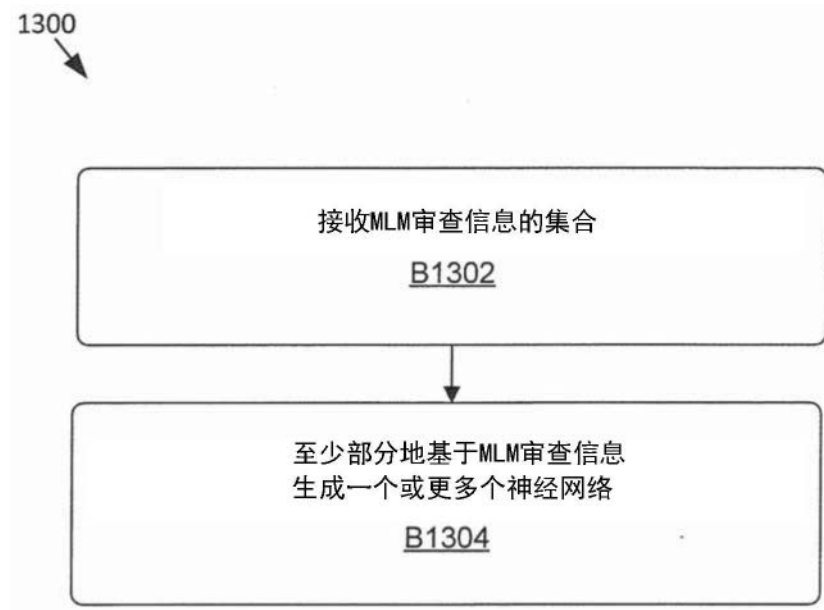


图13



图14

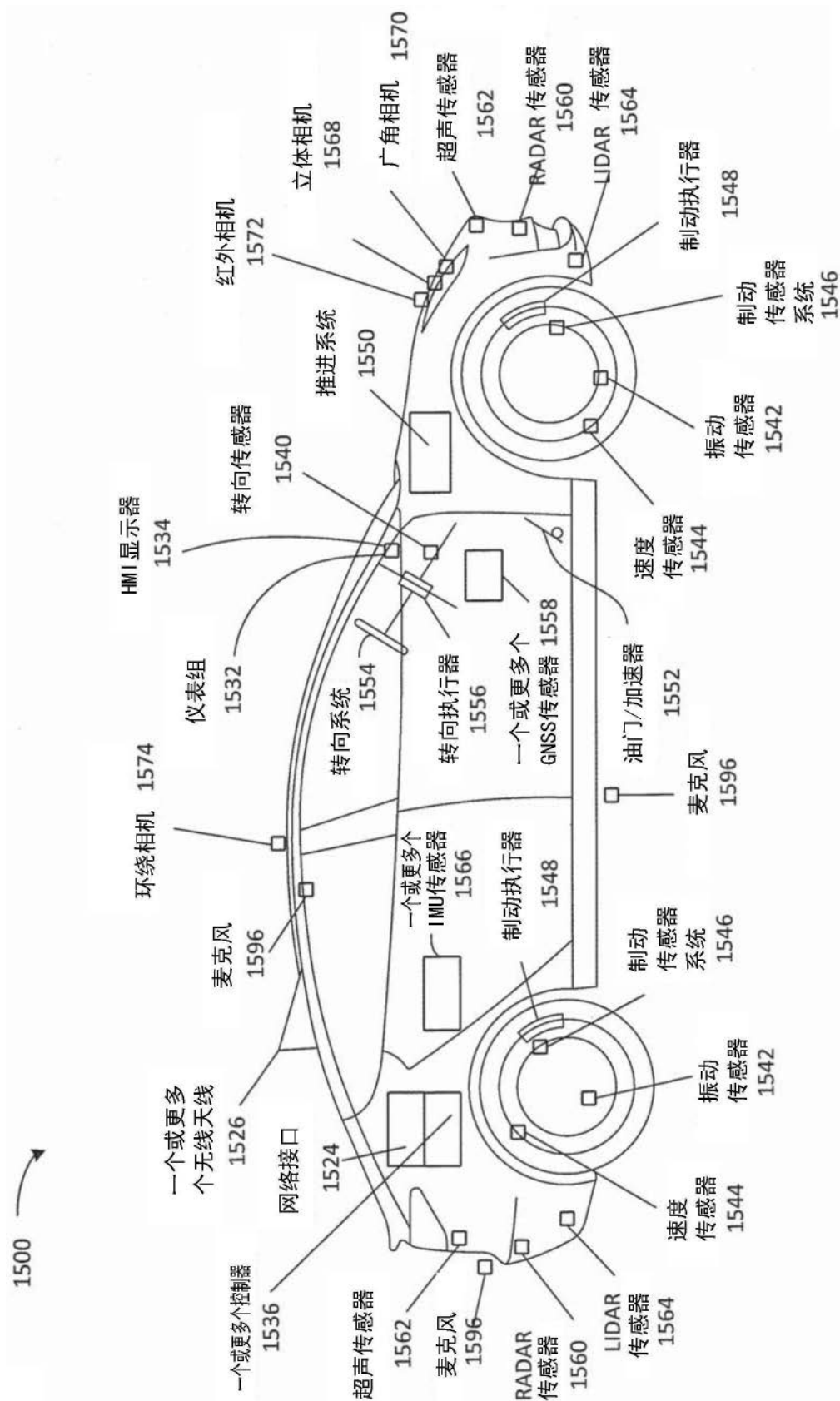


图15A

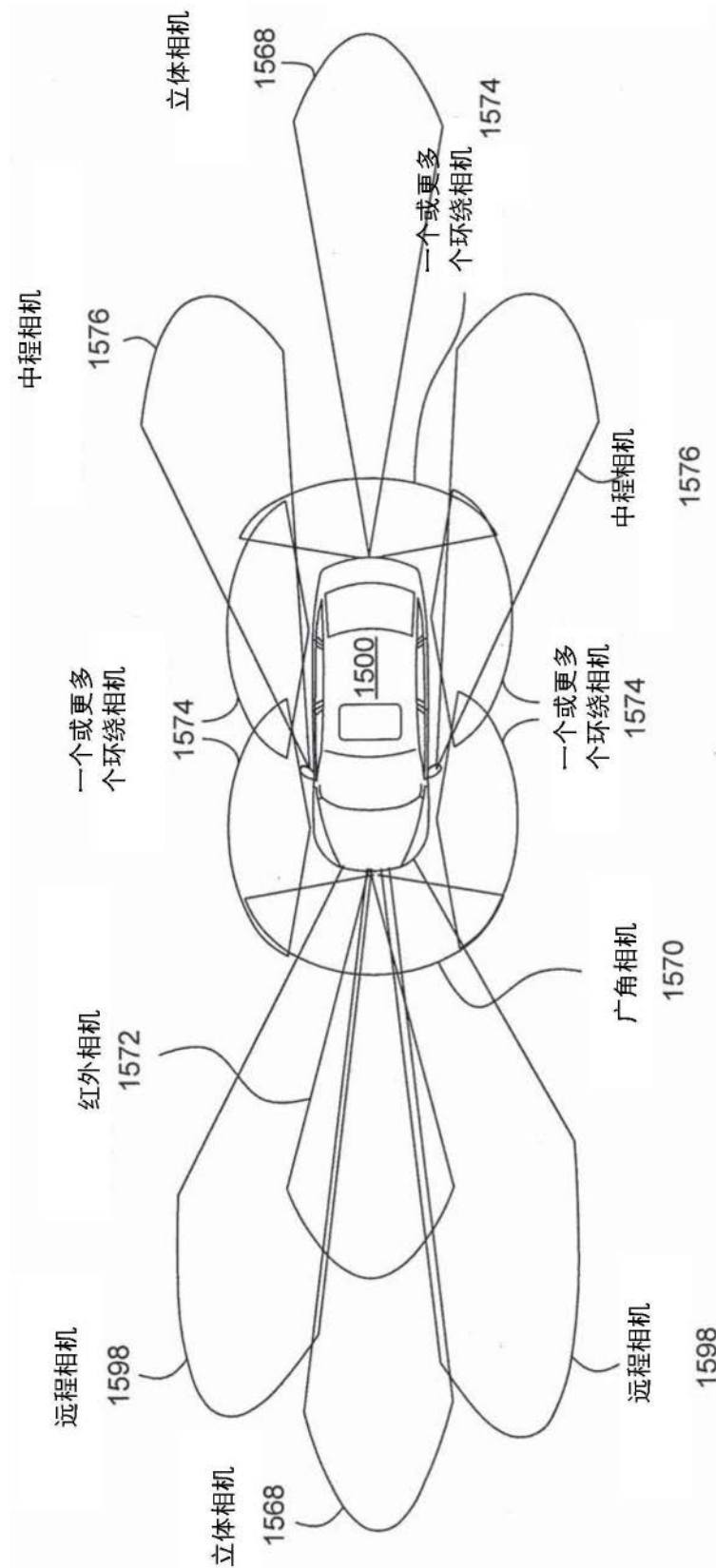


图15B

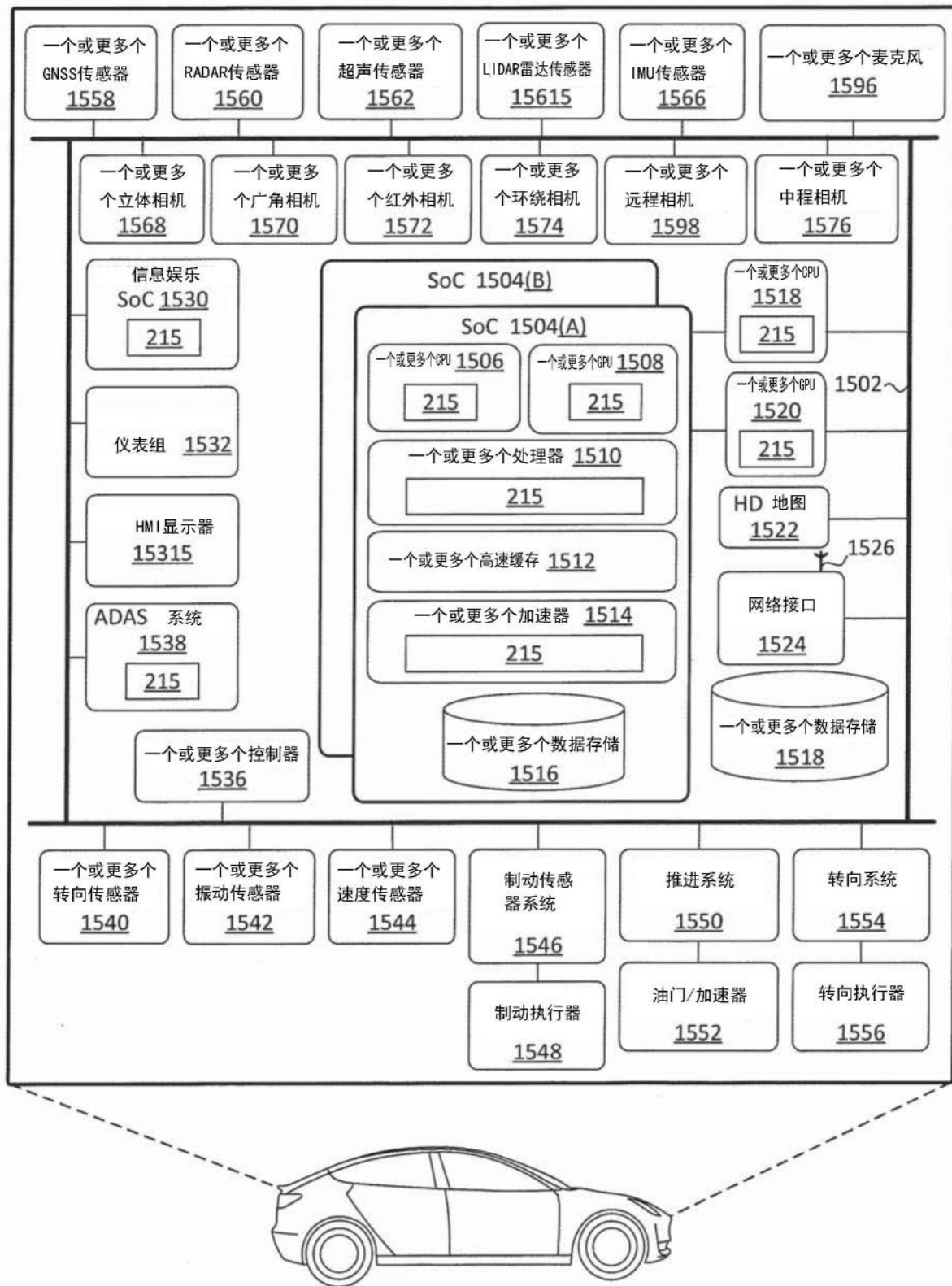


图15C

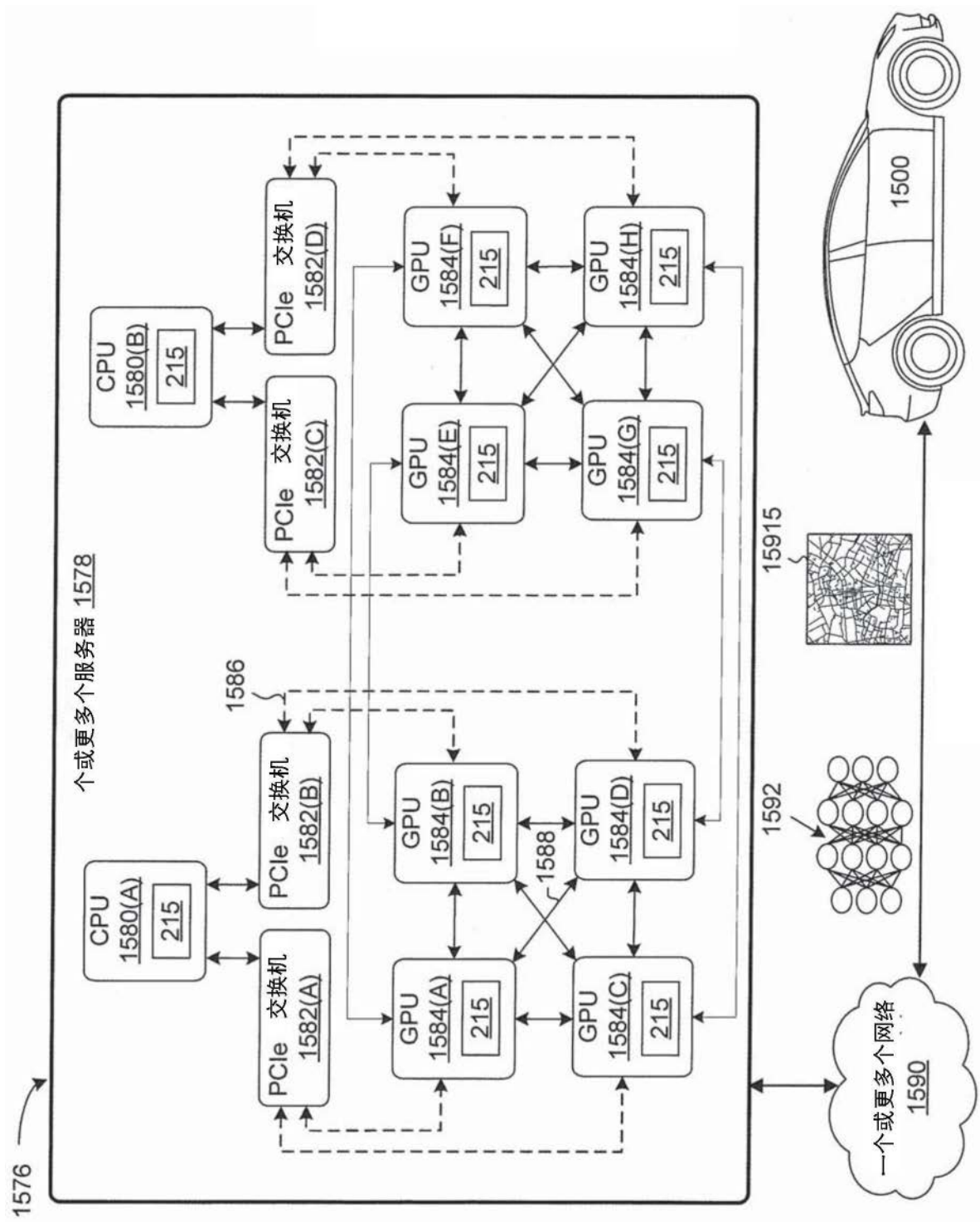


图15D

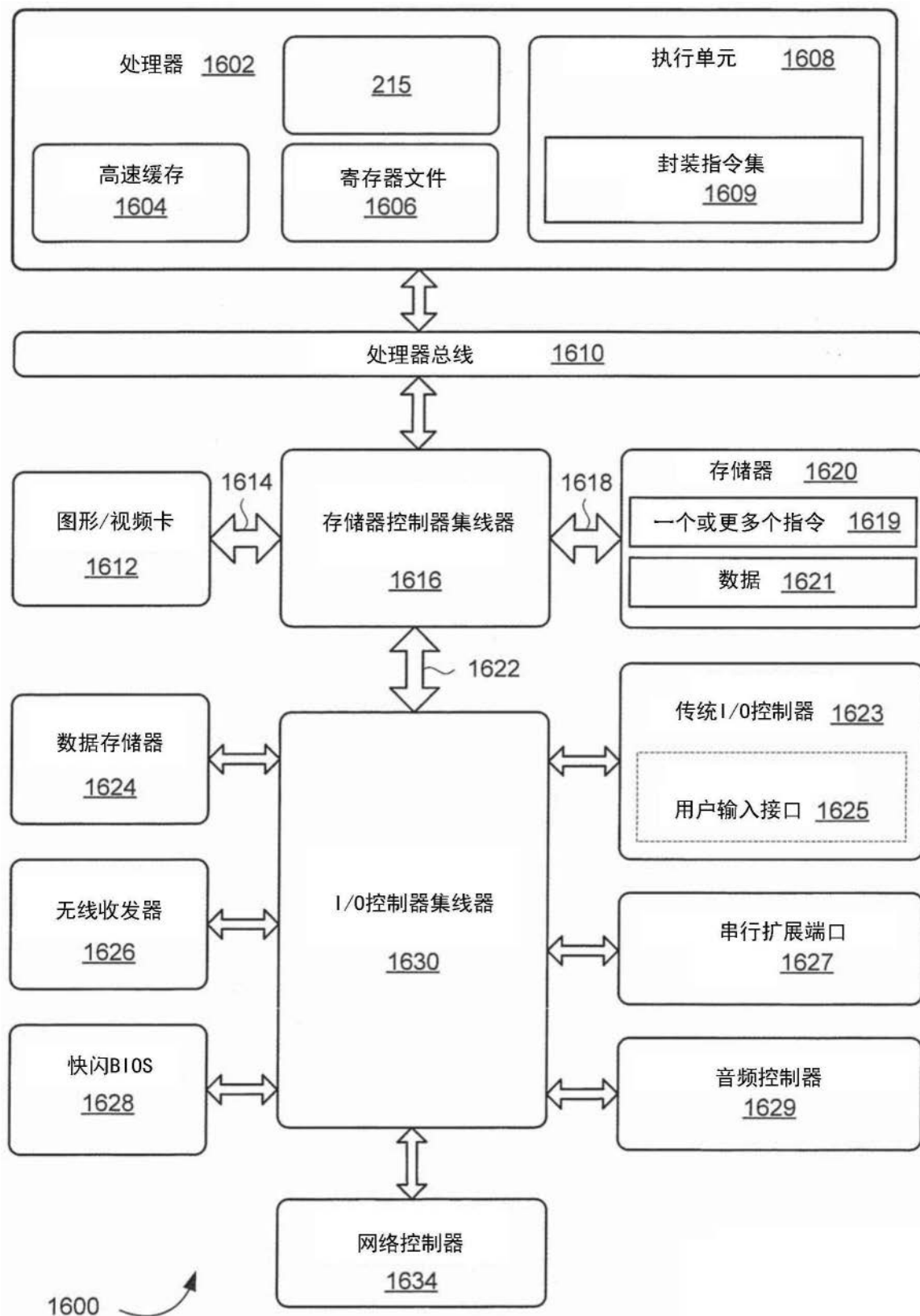


图16

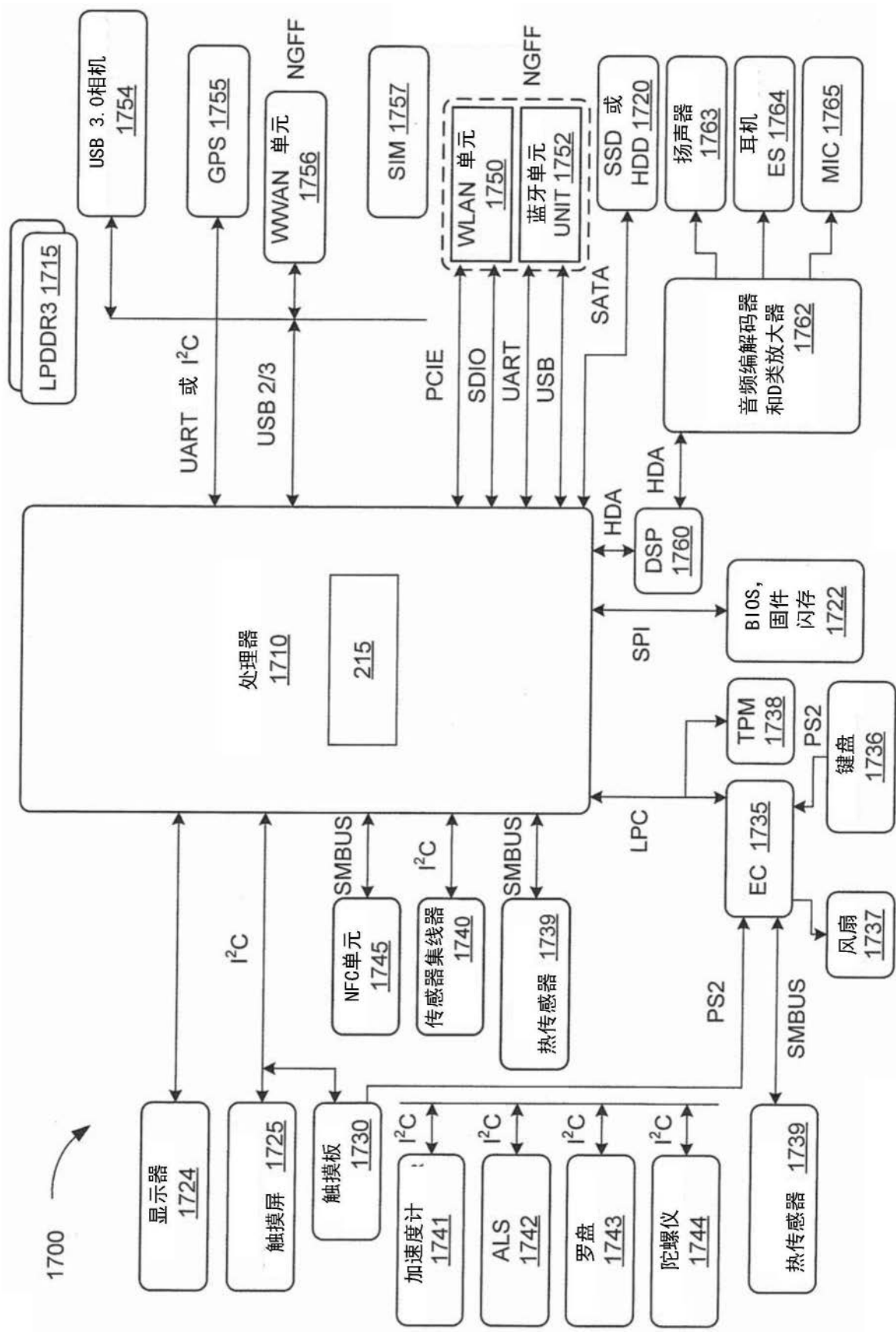


图17

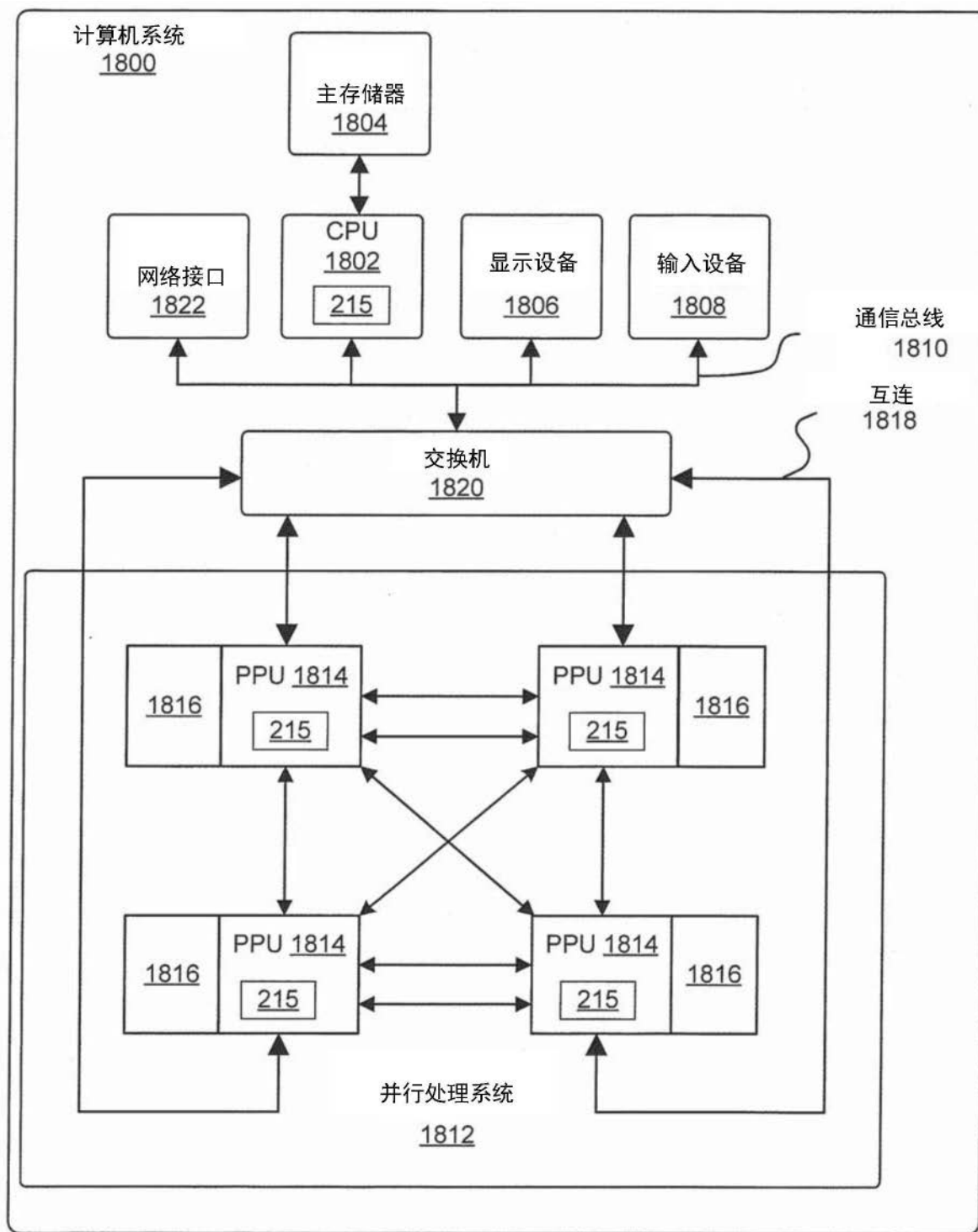


图18

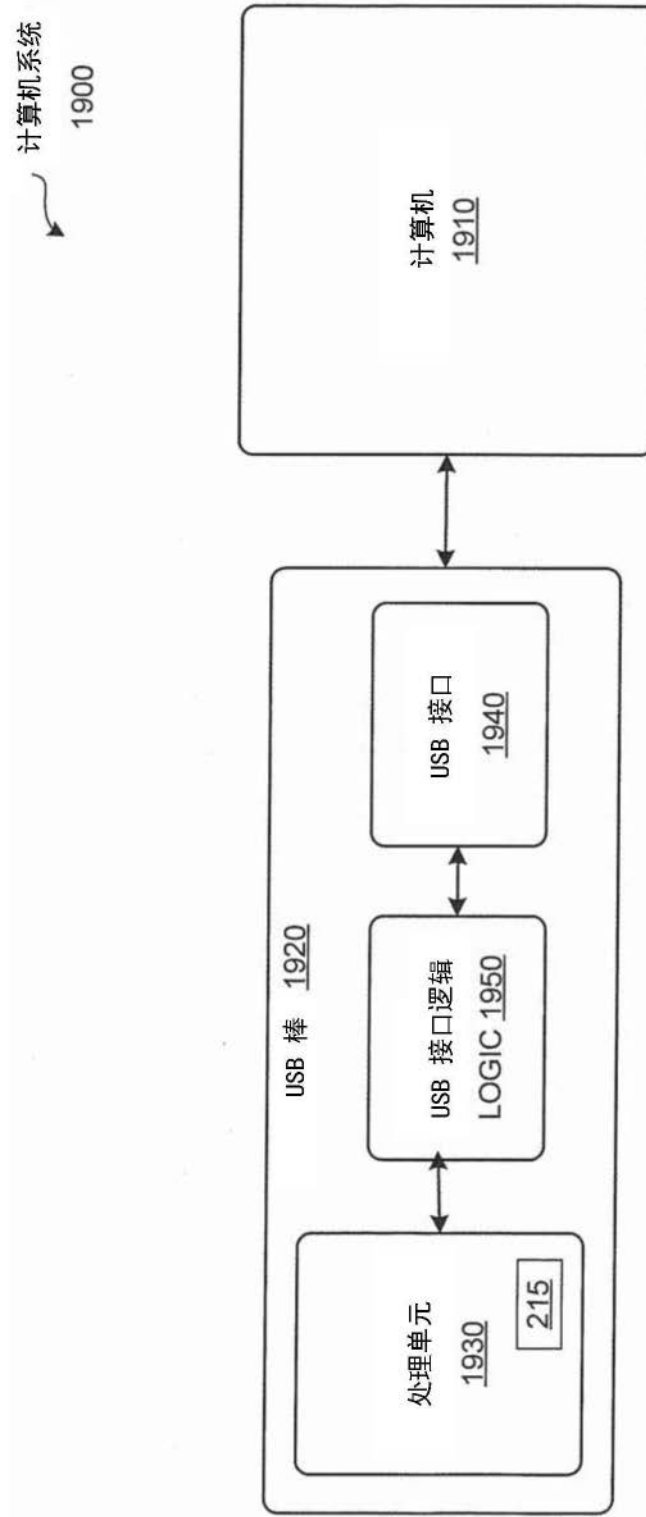


图19

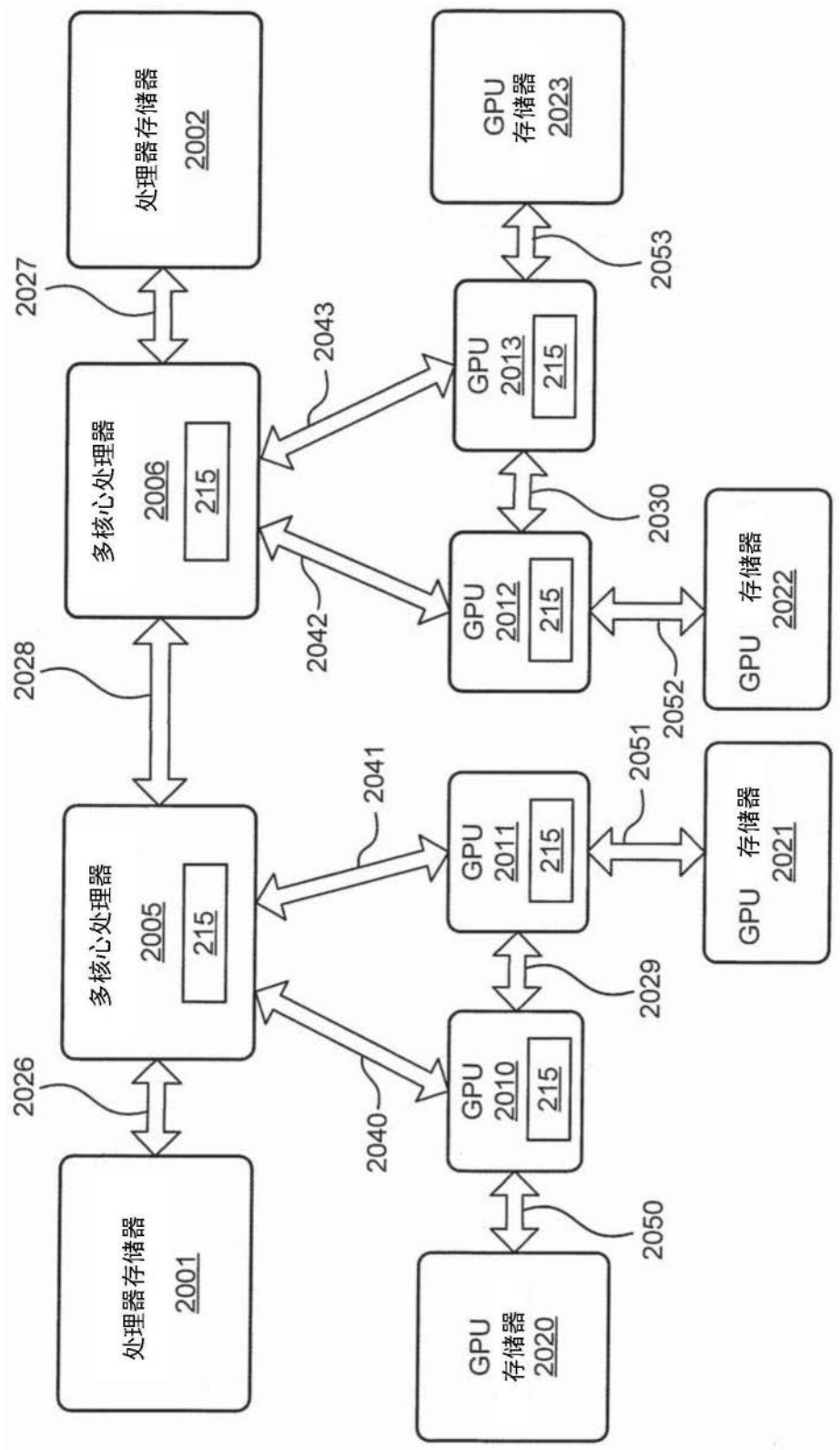


图20A

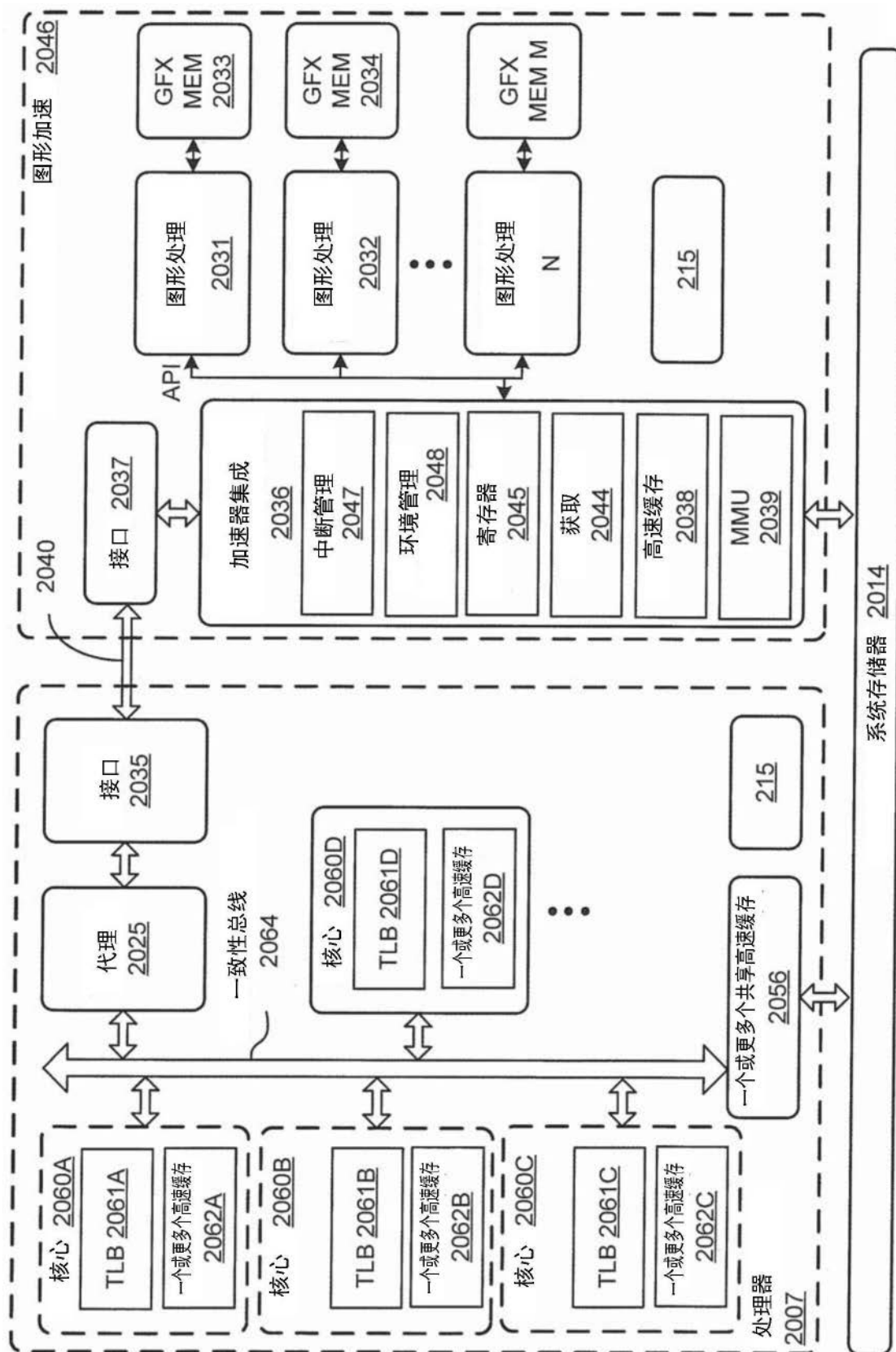


图20B

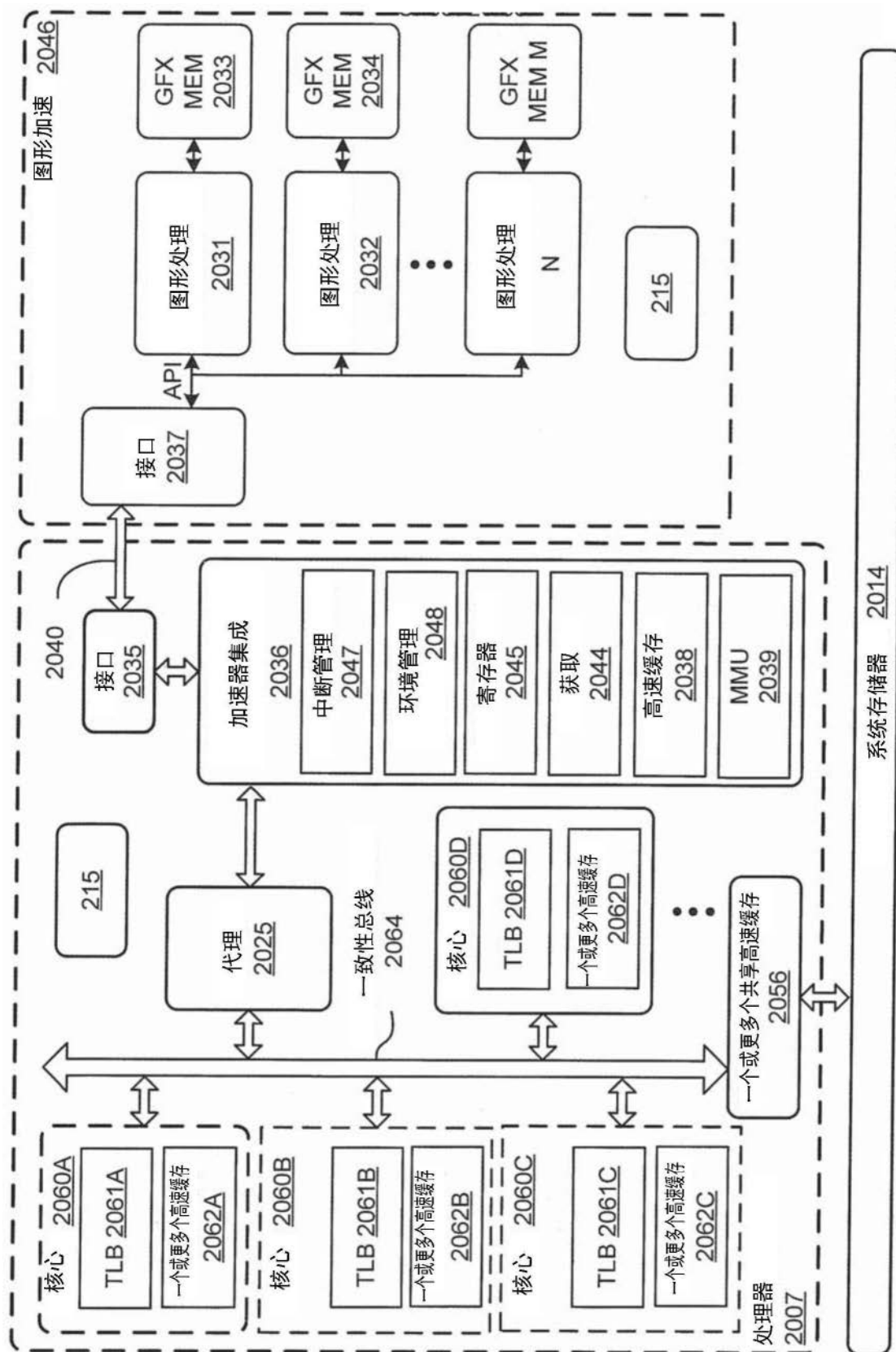


图20C

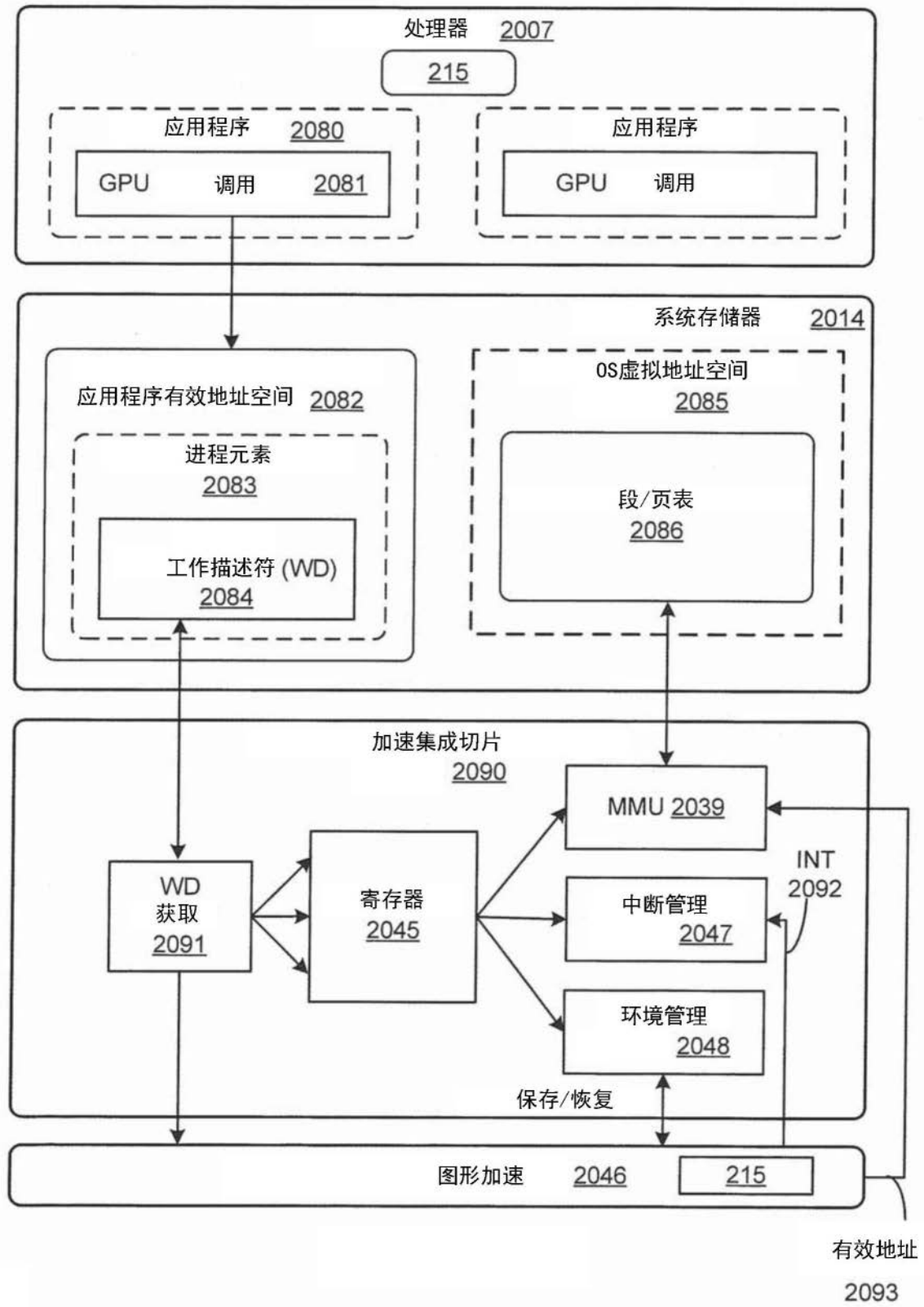


图20D

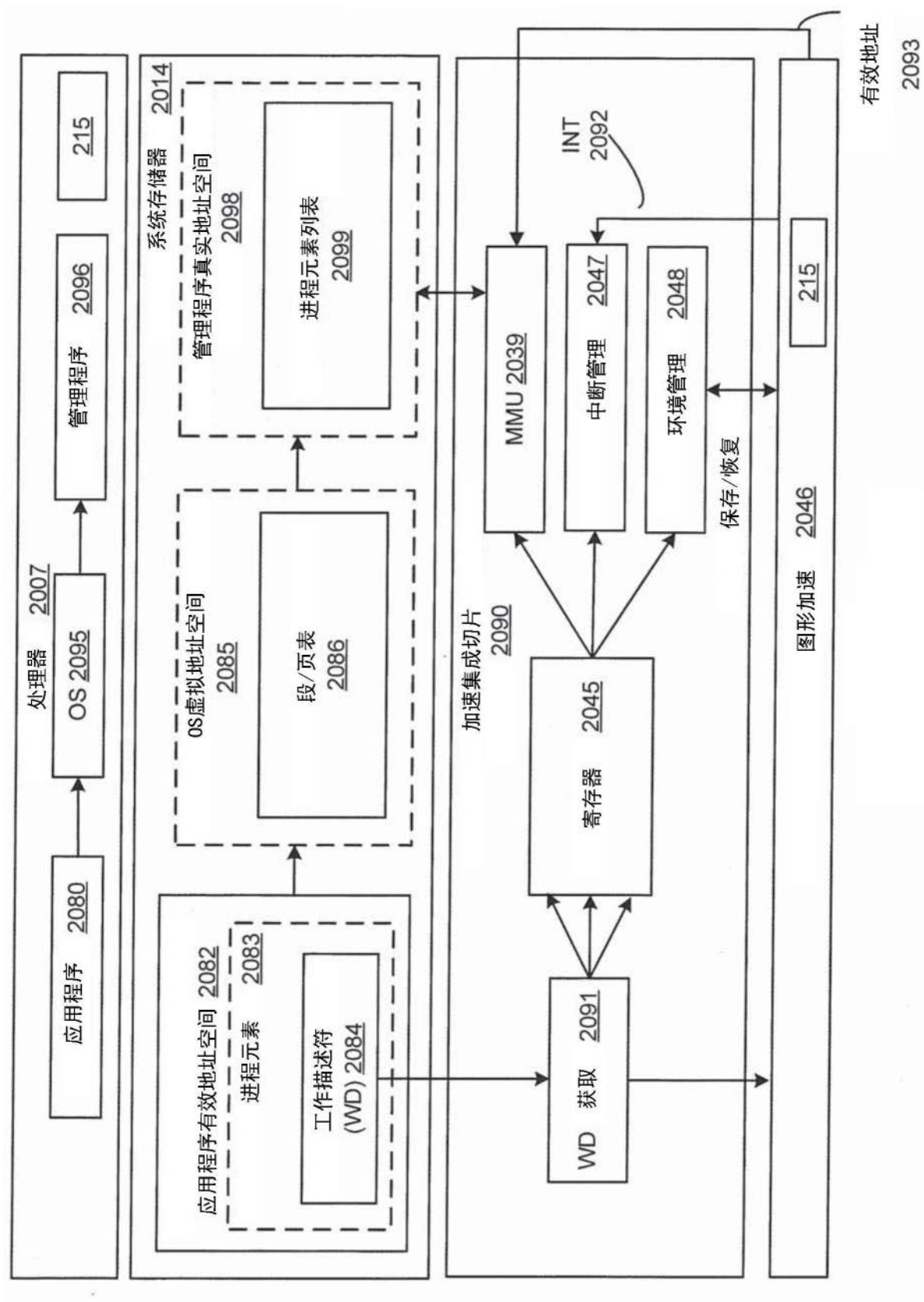


图20E

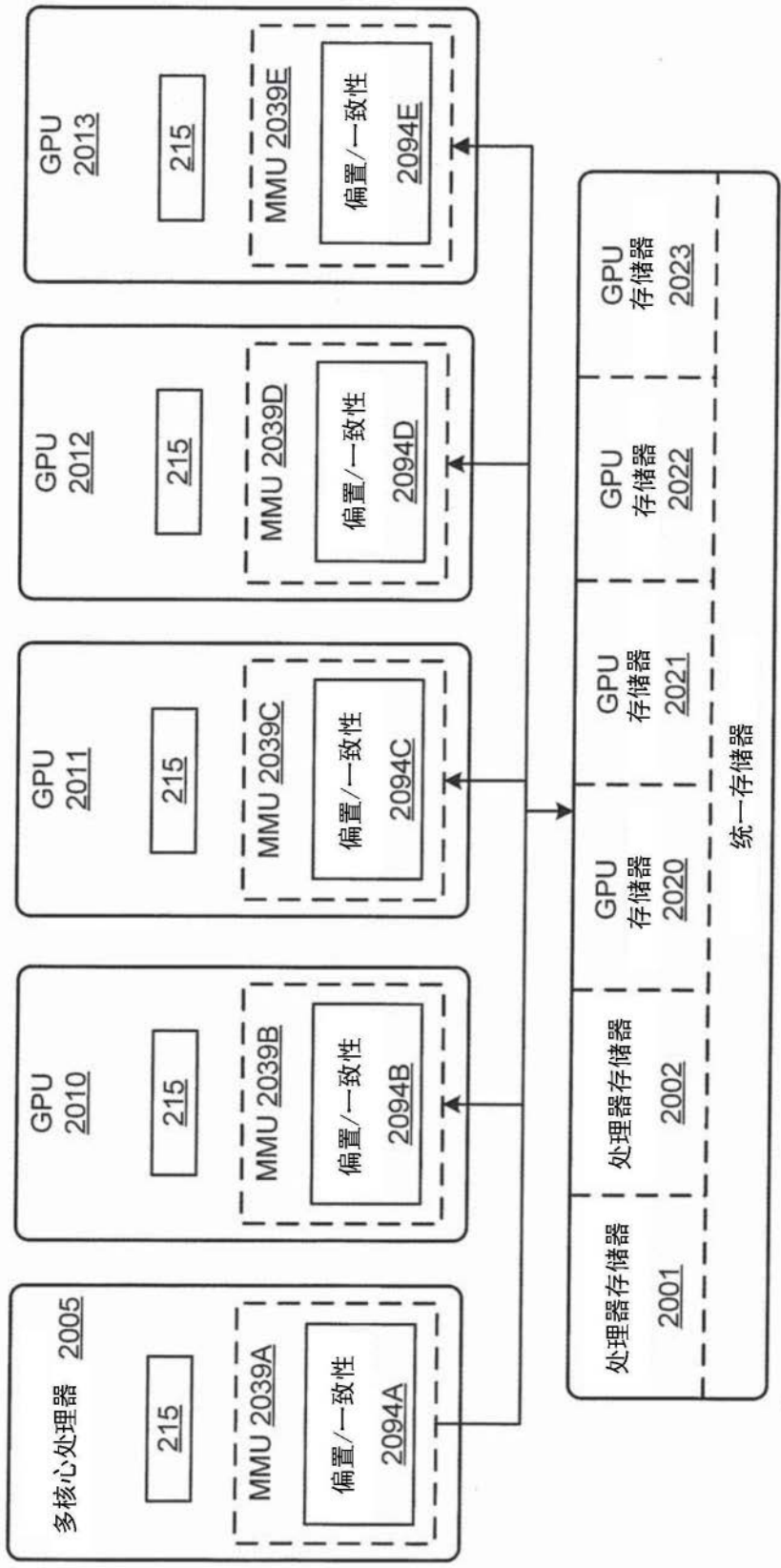


图20F

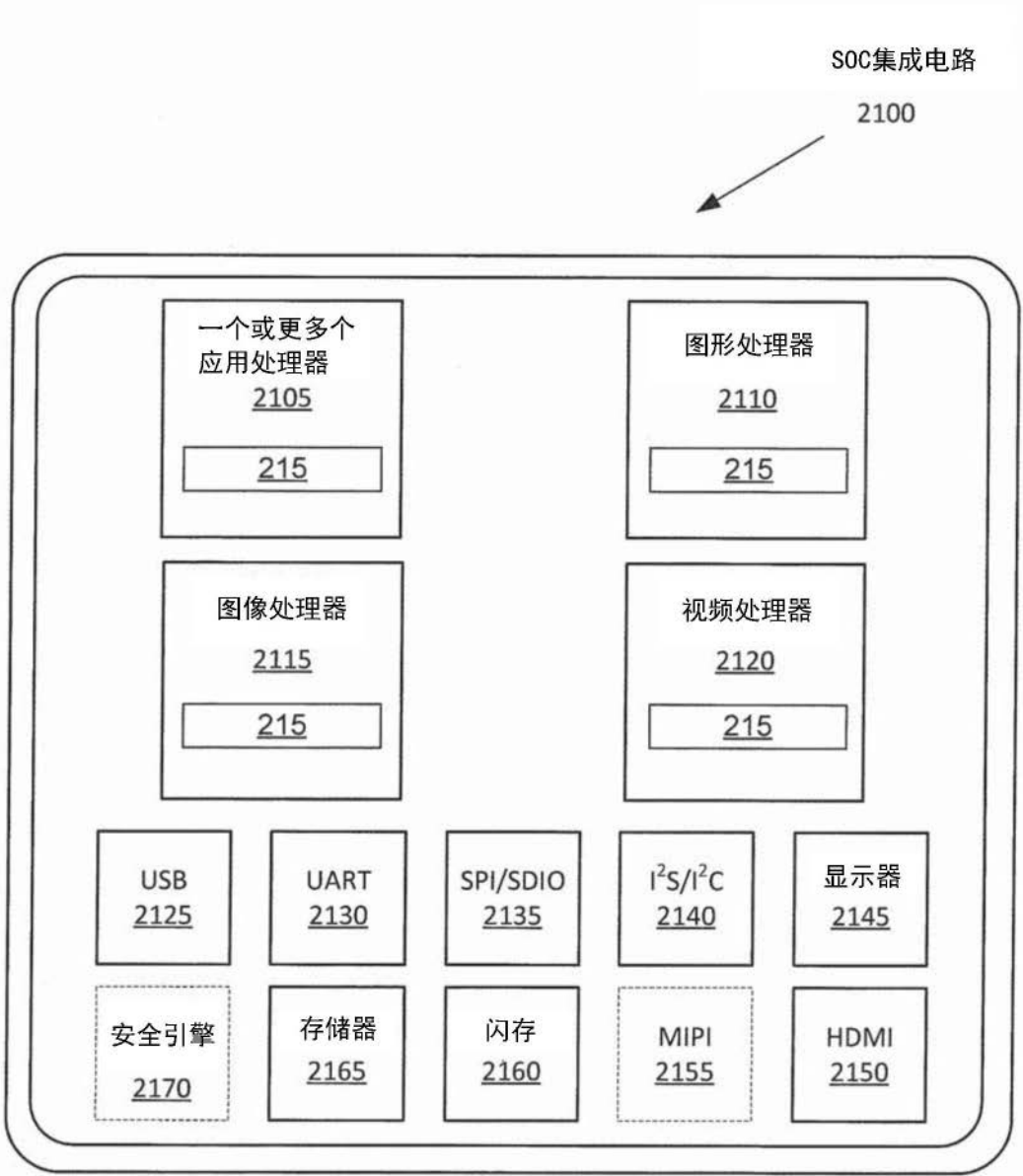


图21



图22A



图22B

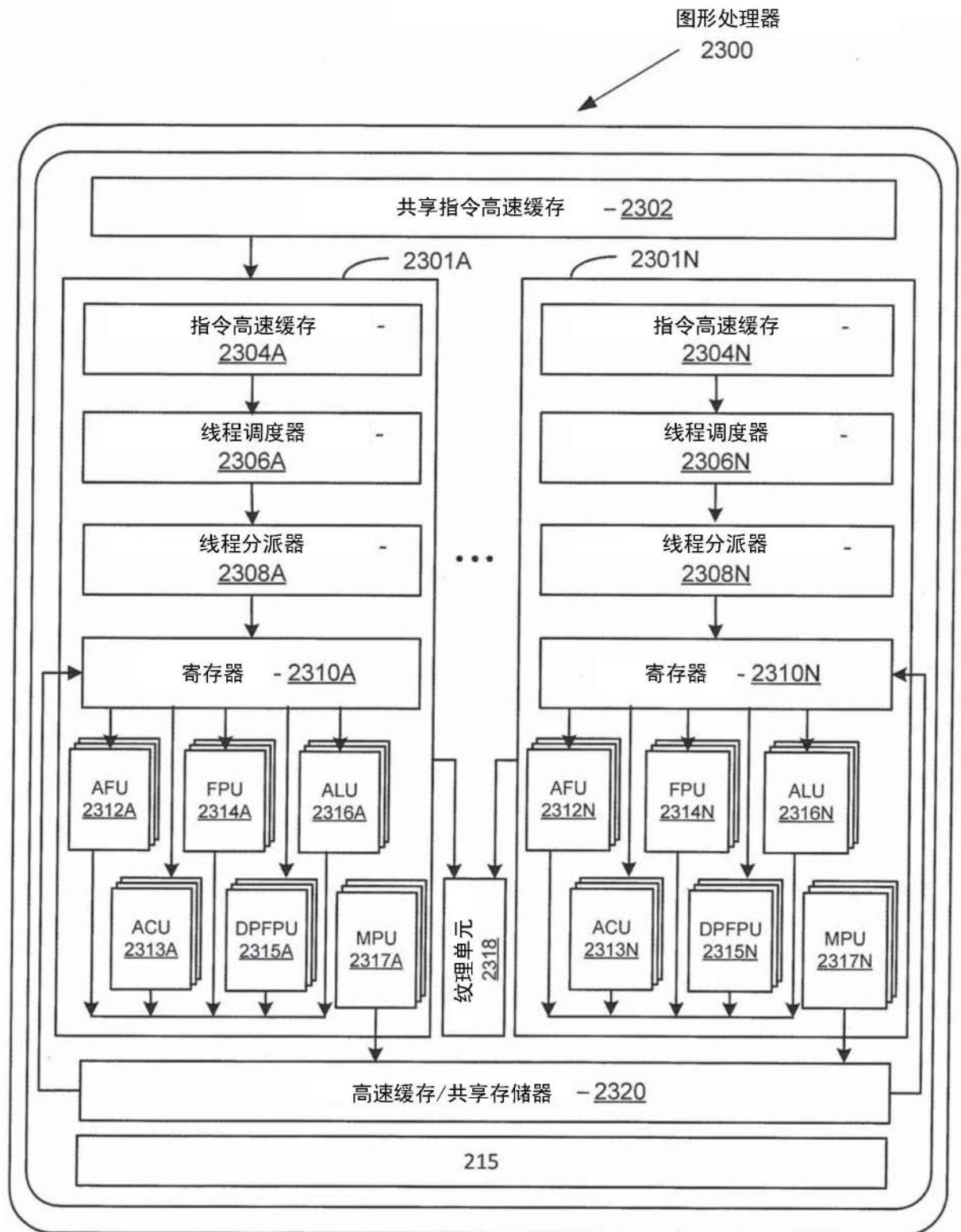


图23A

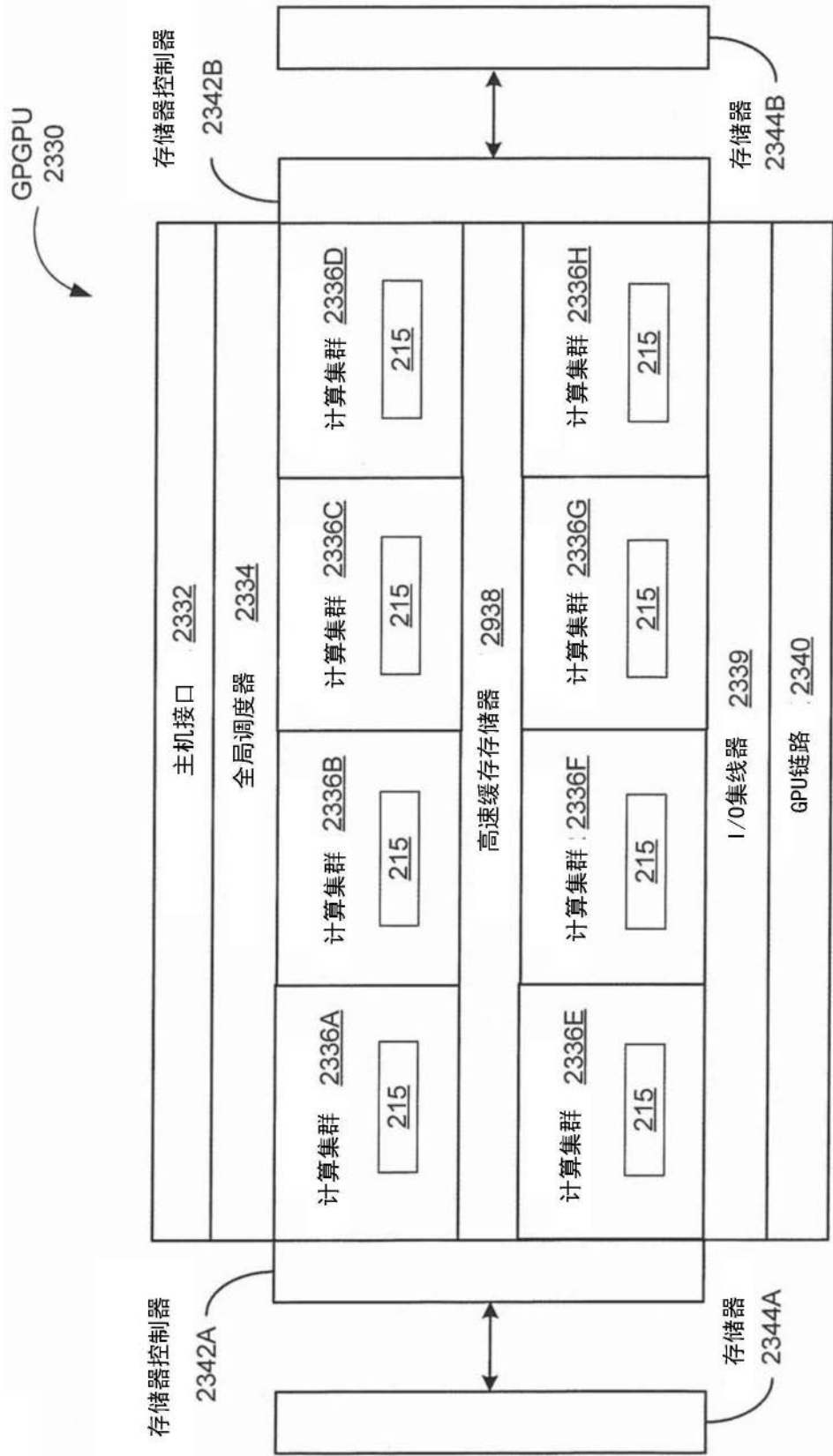


图23B

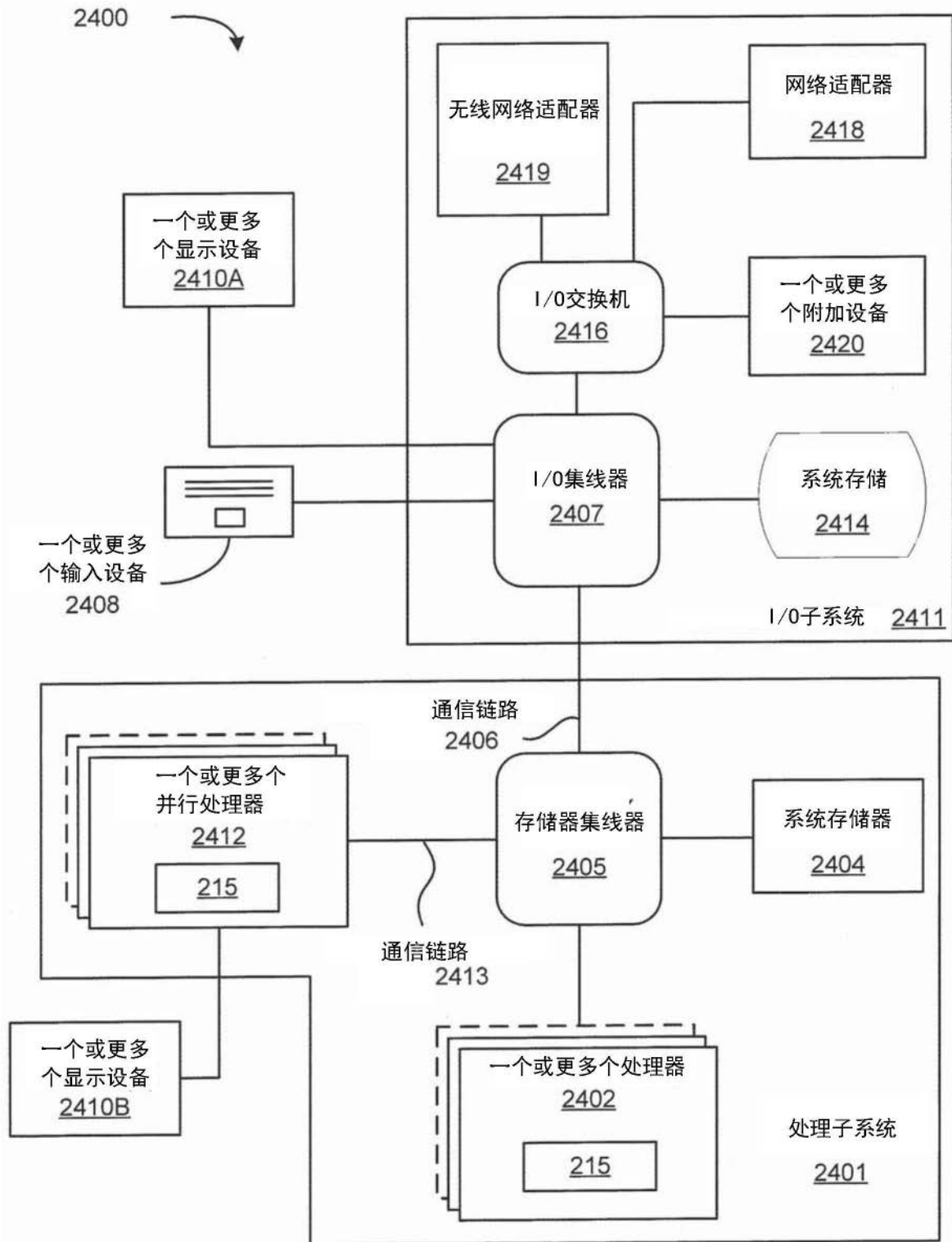


图24

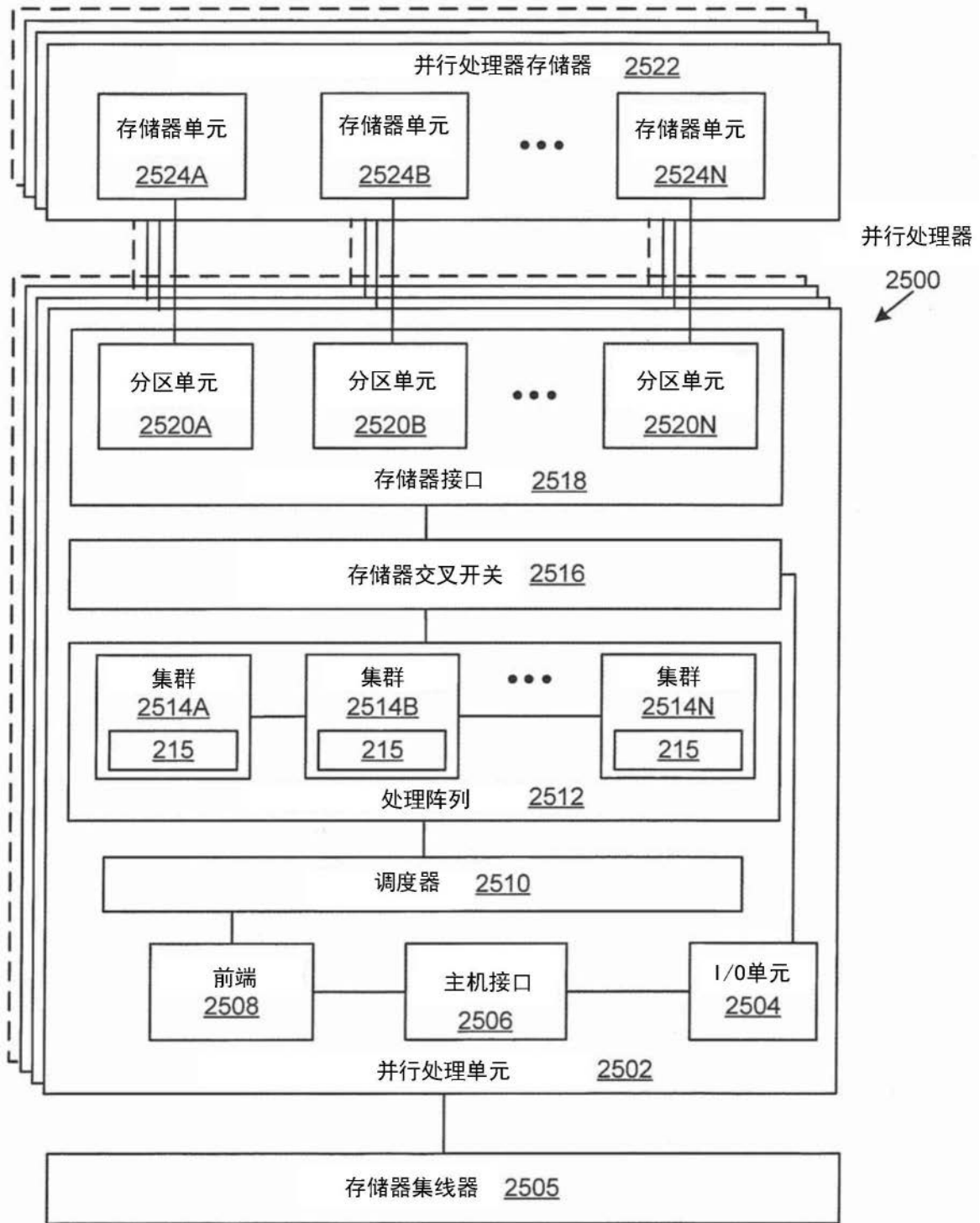


图25A

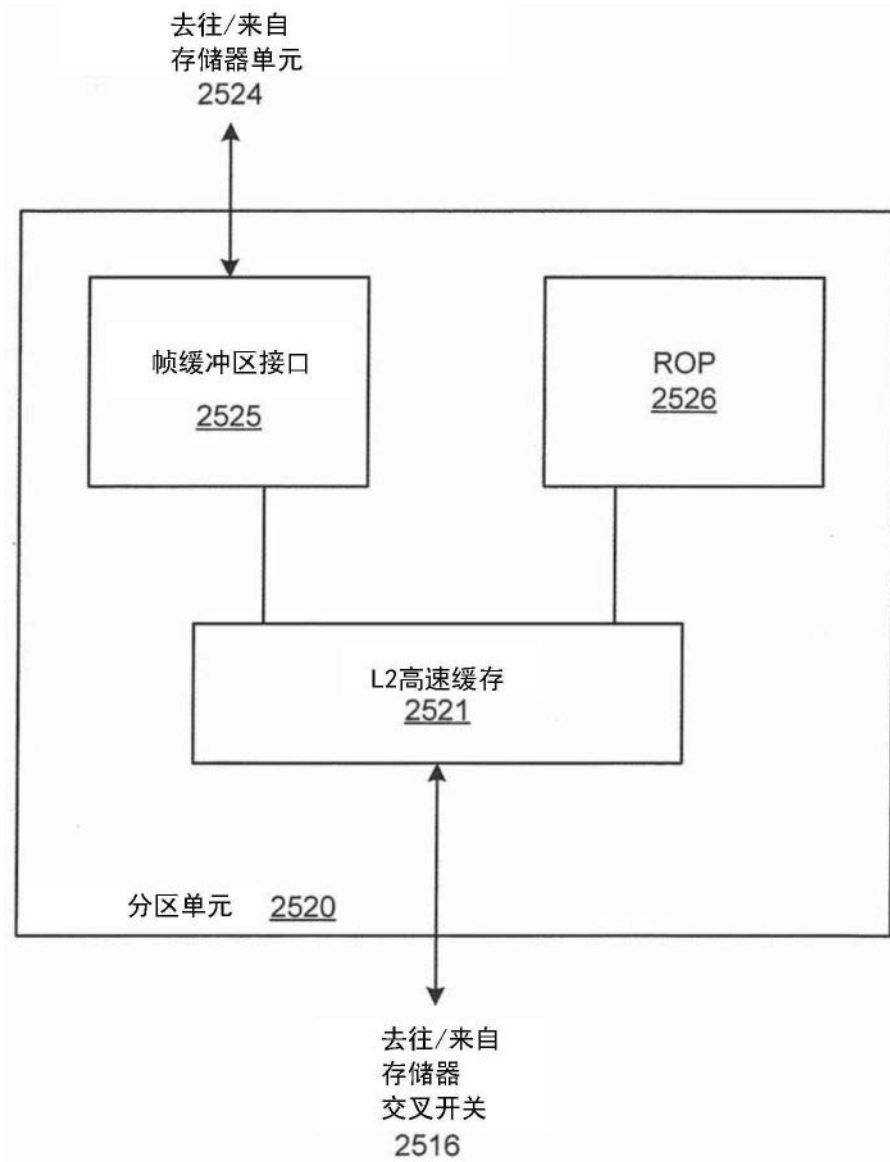


图25B

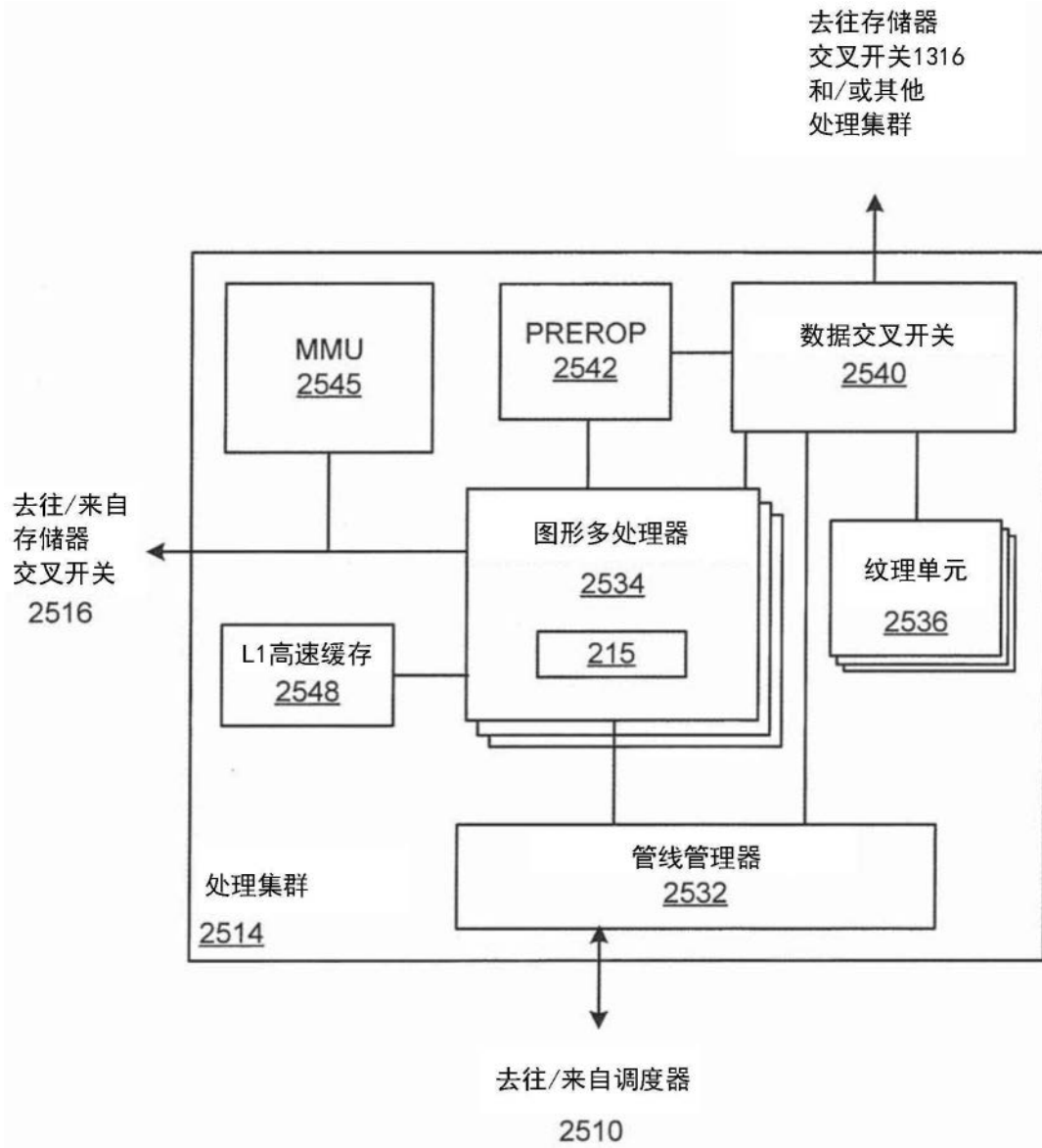


图25C

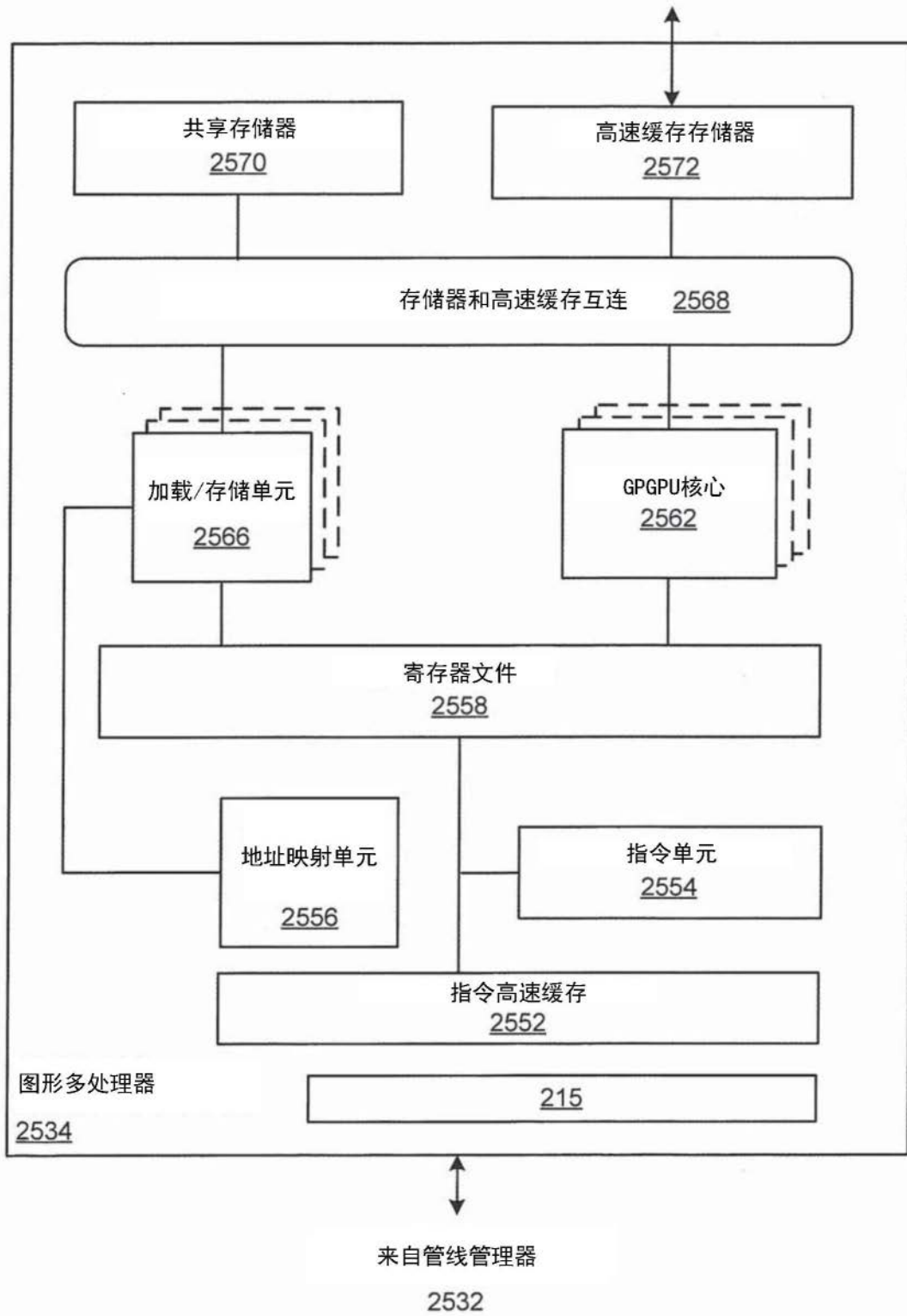


图25D

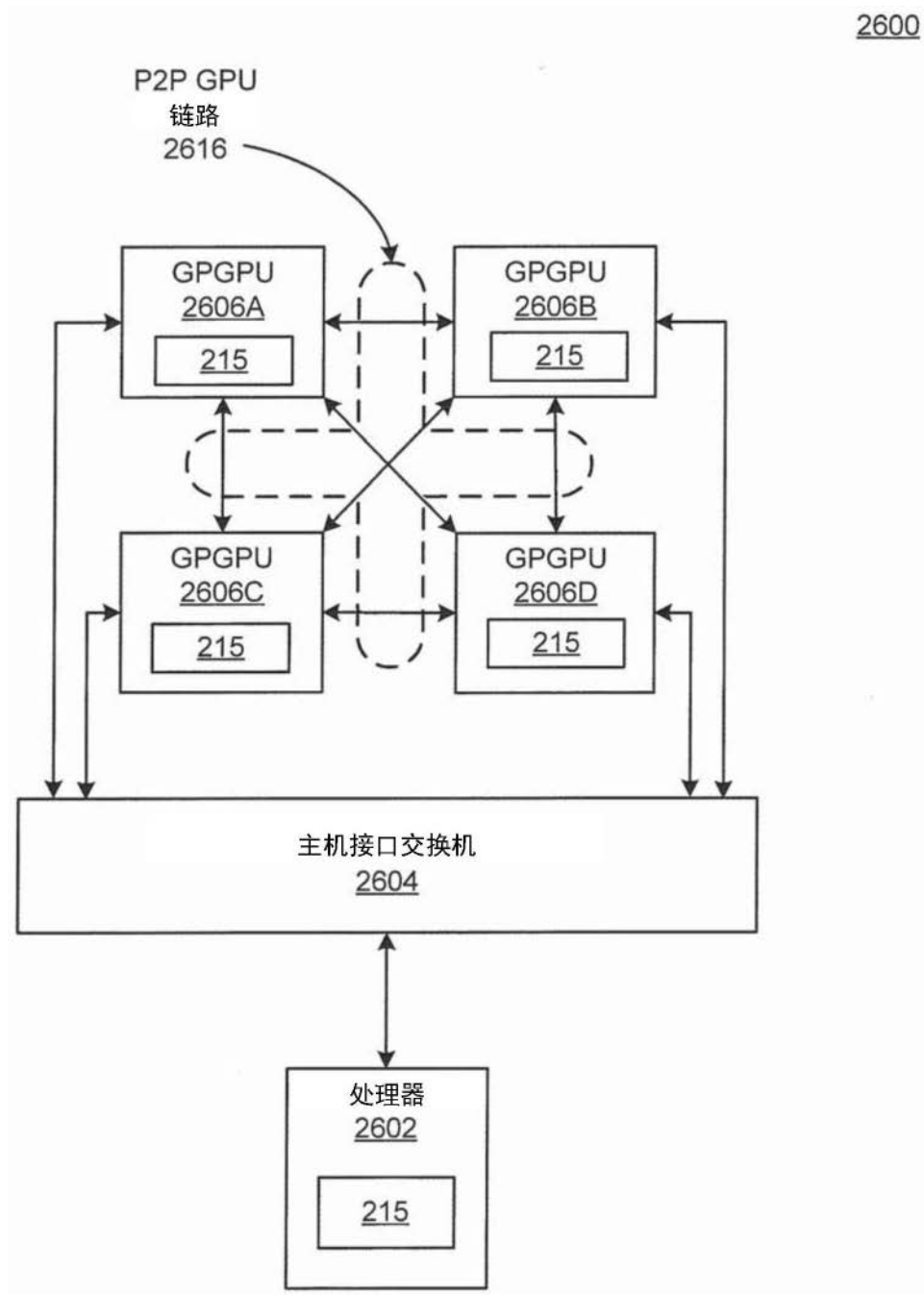


图26

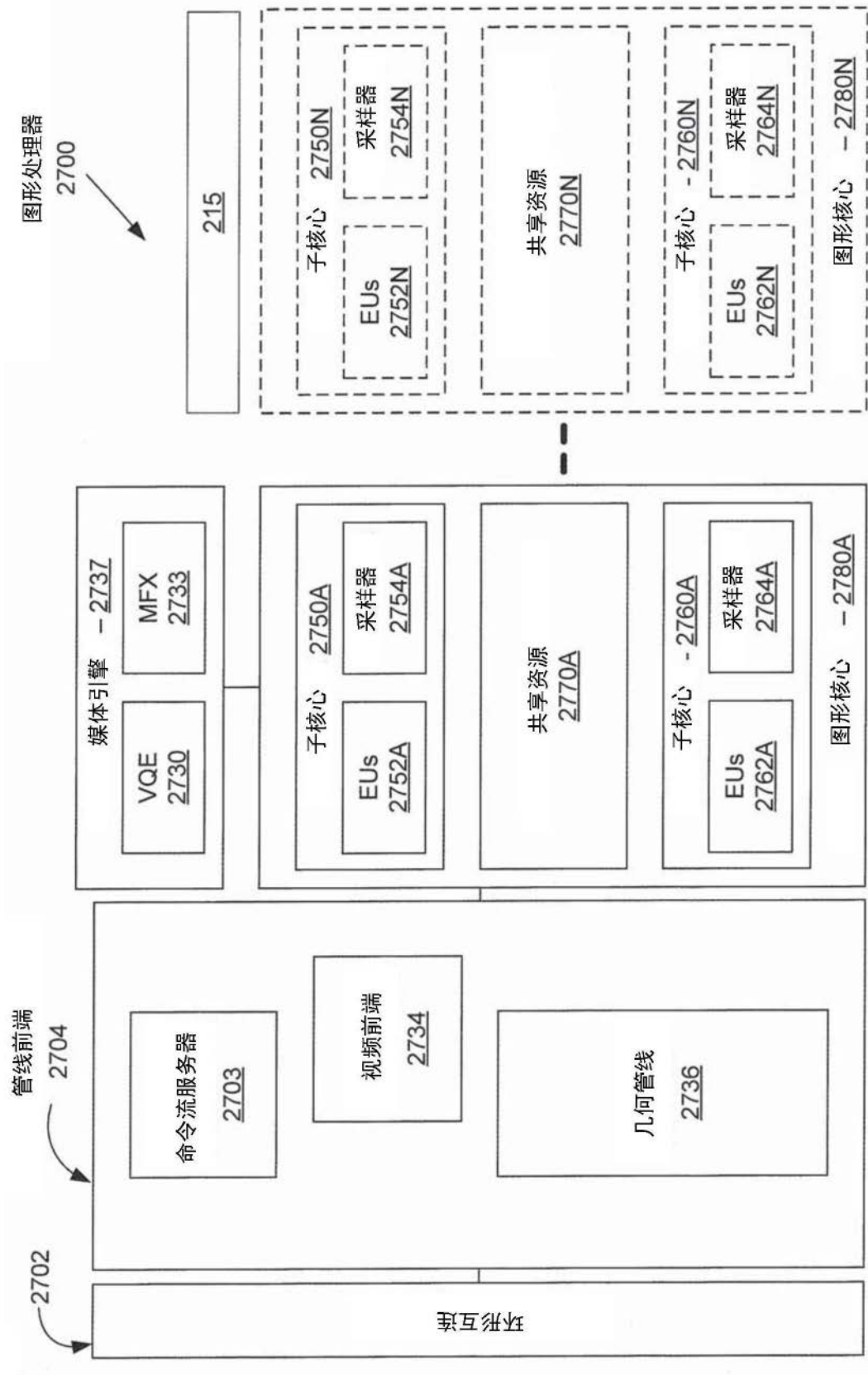


图27

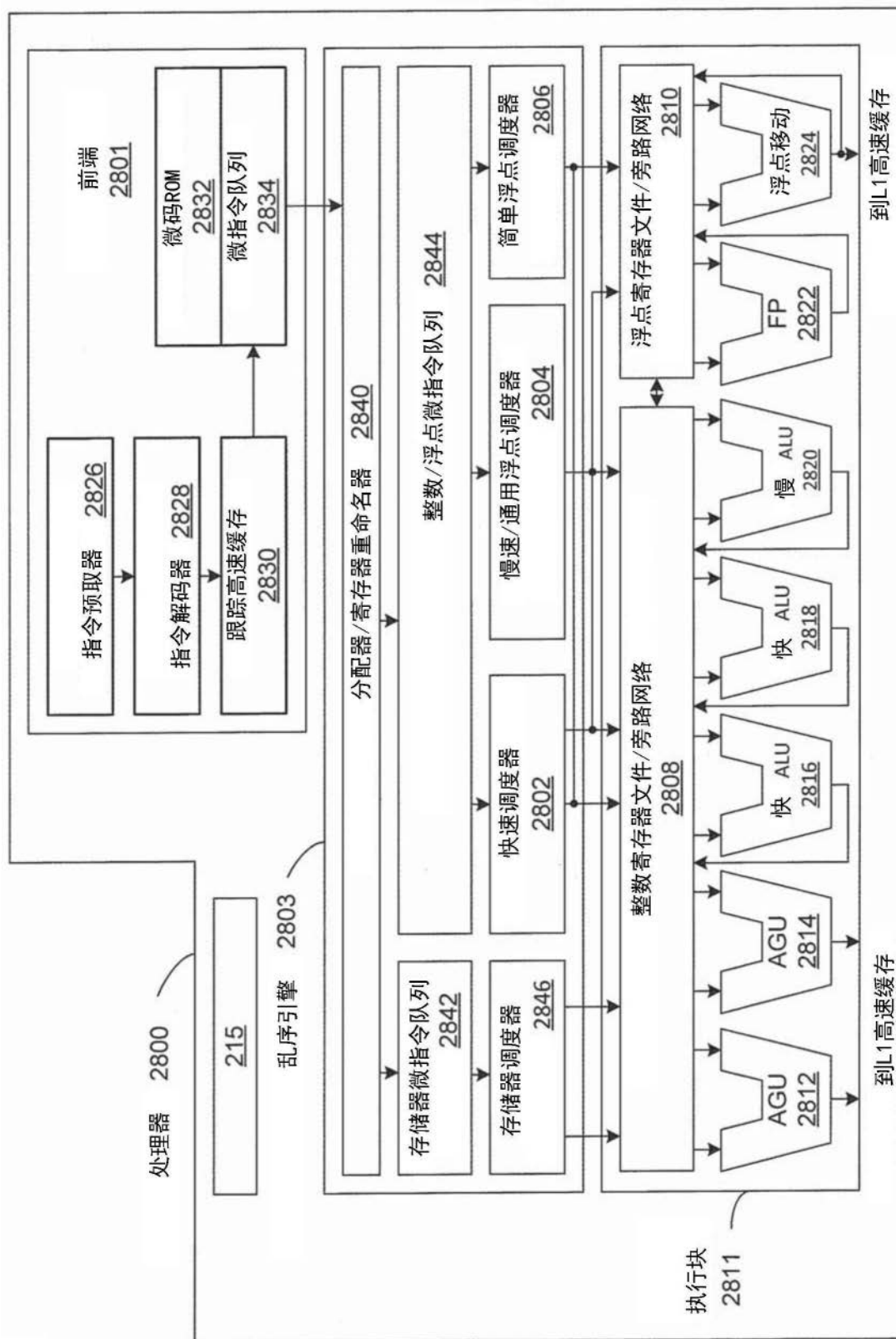


图28

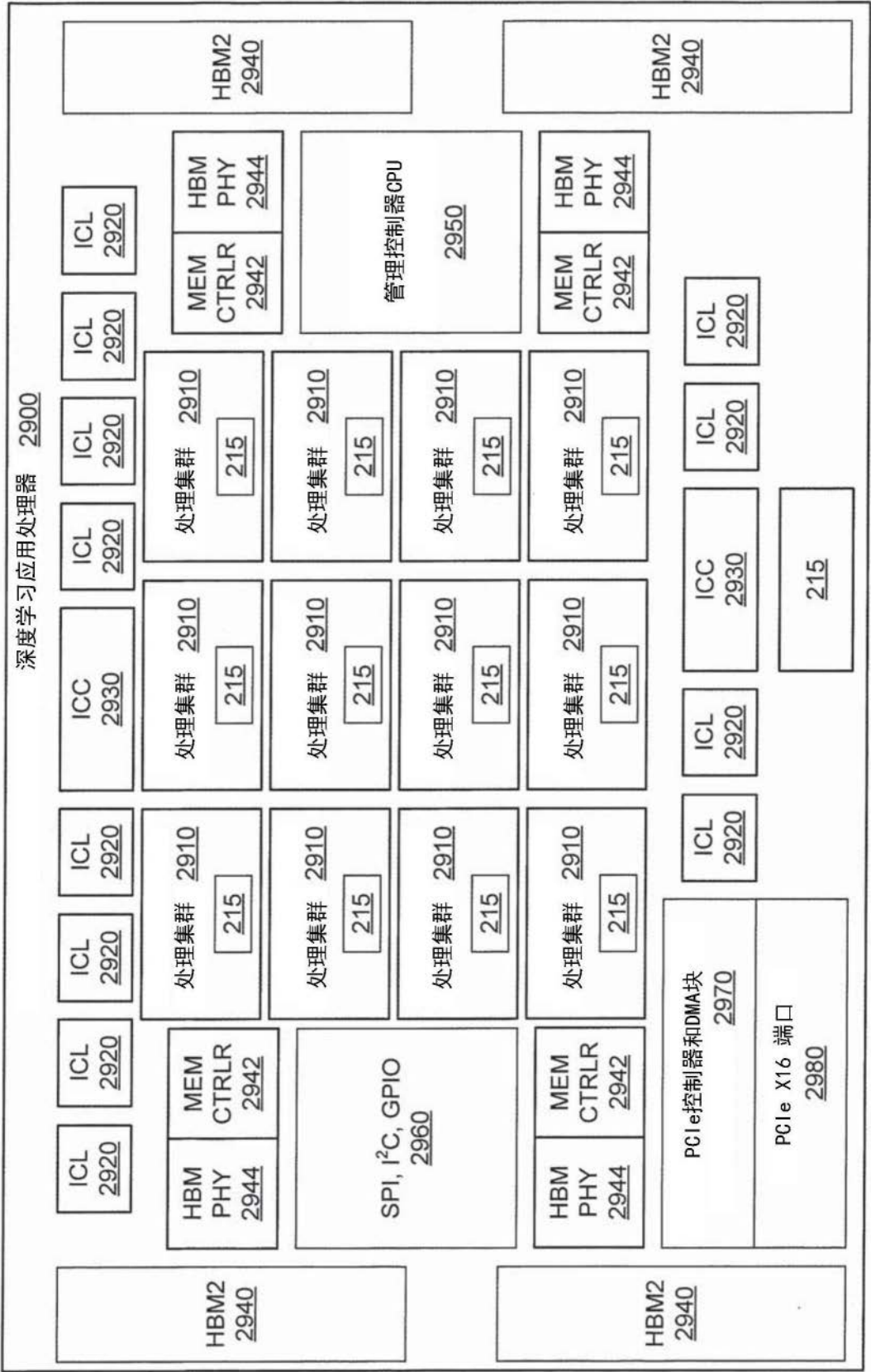


图29

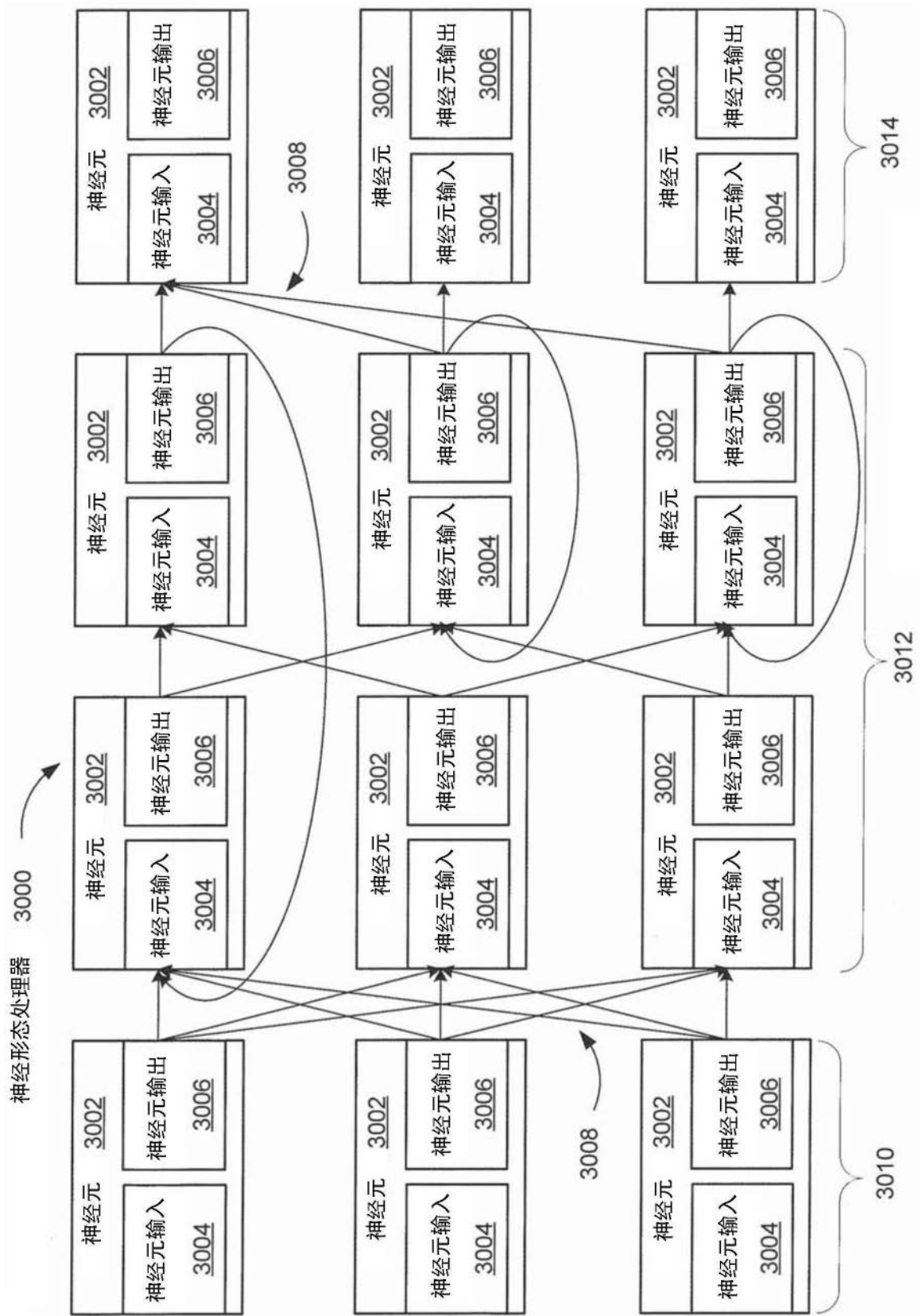


图30

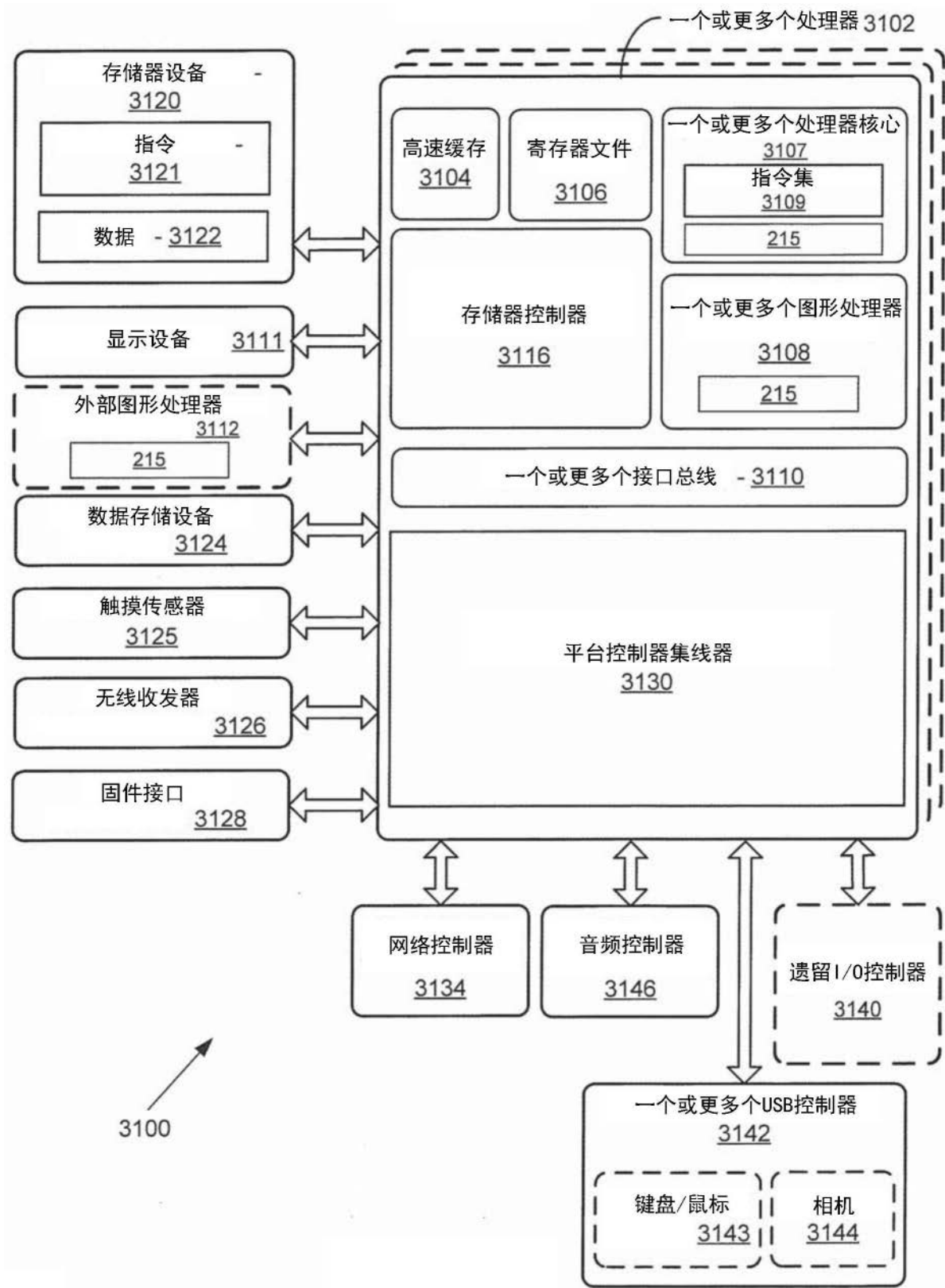


图31

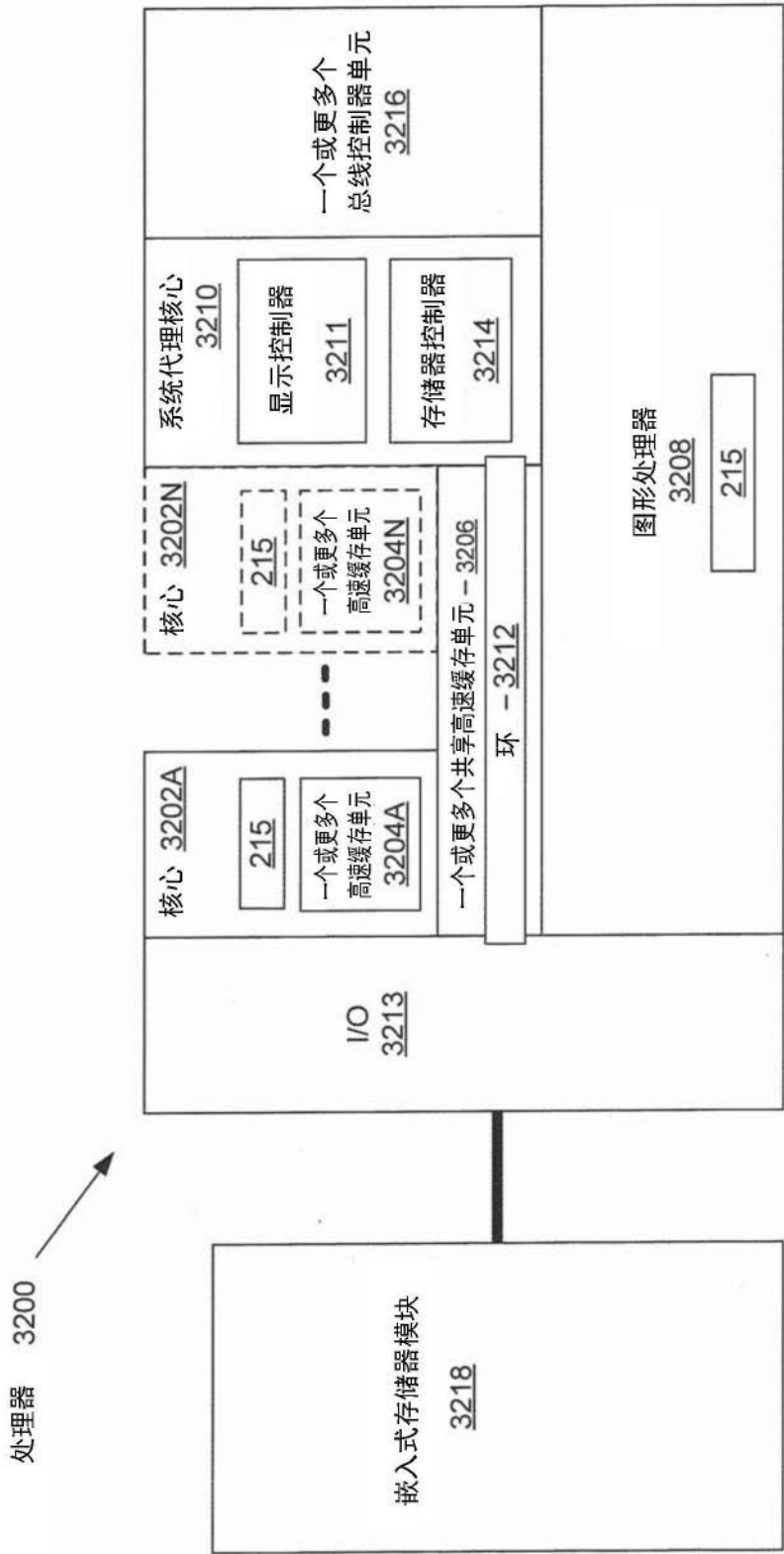


图32

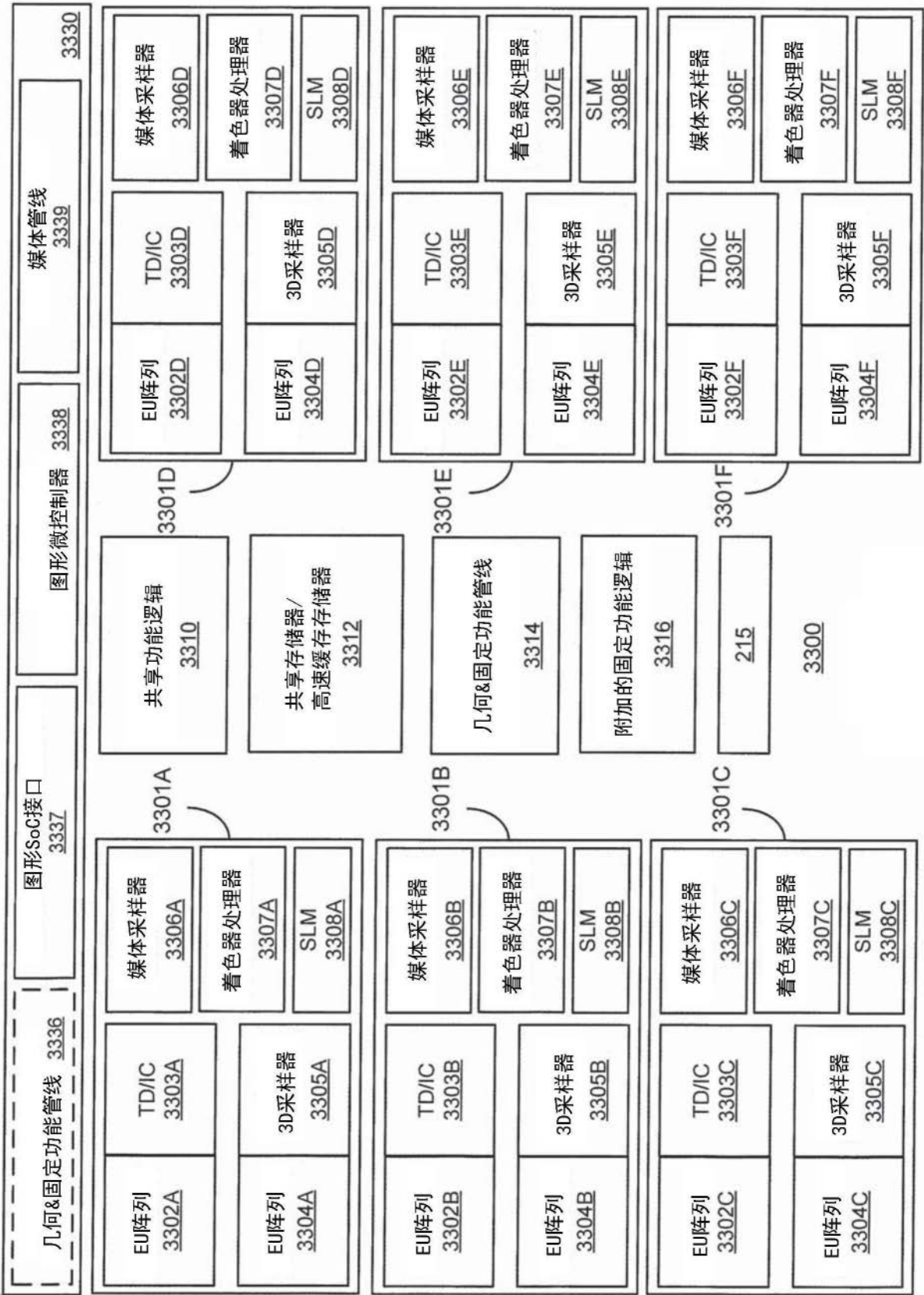


图33

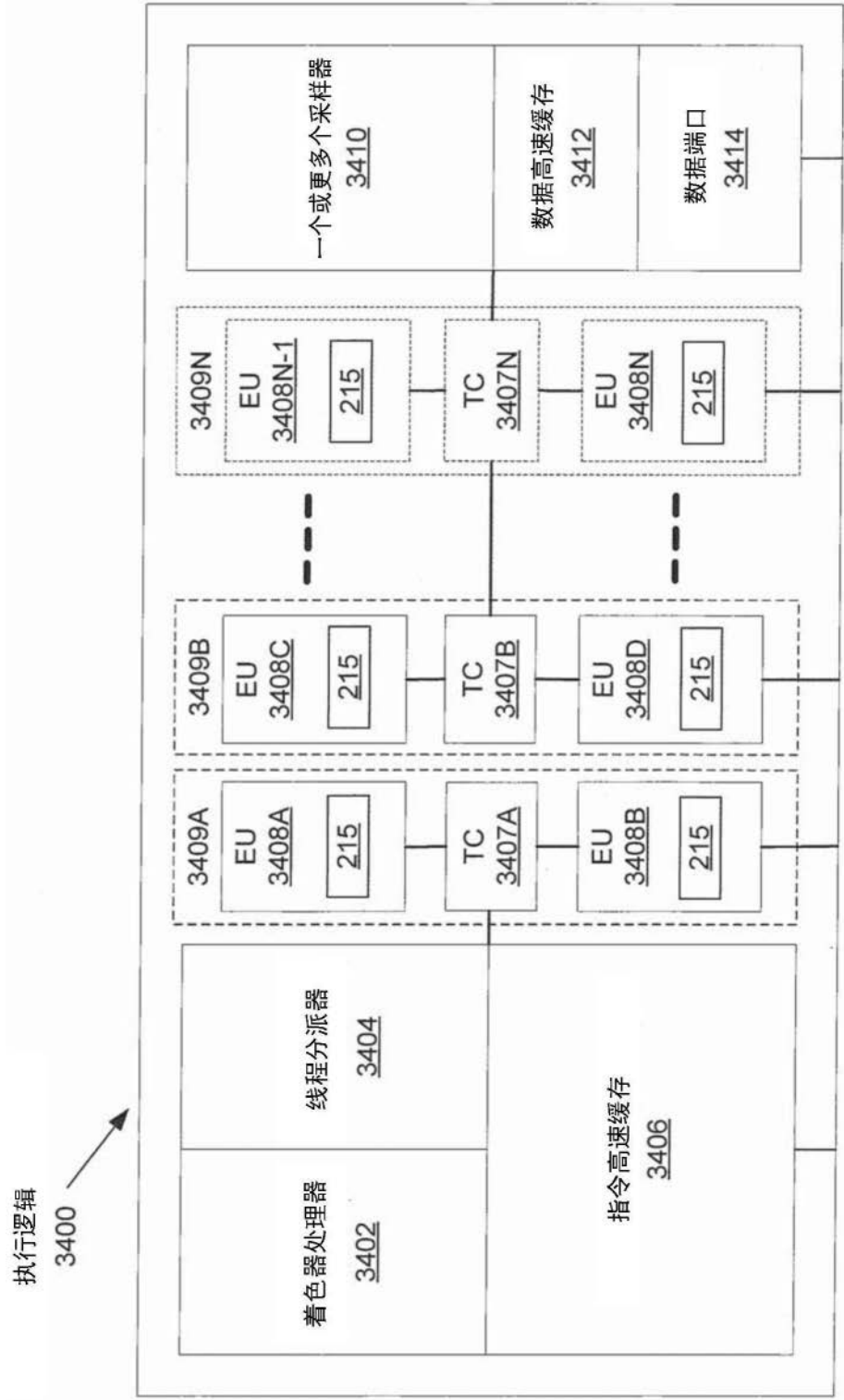


图34A

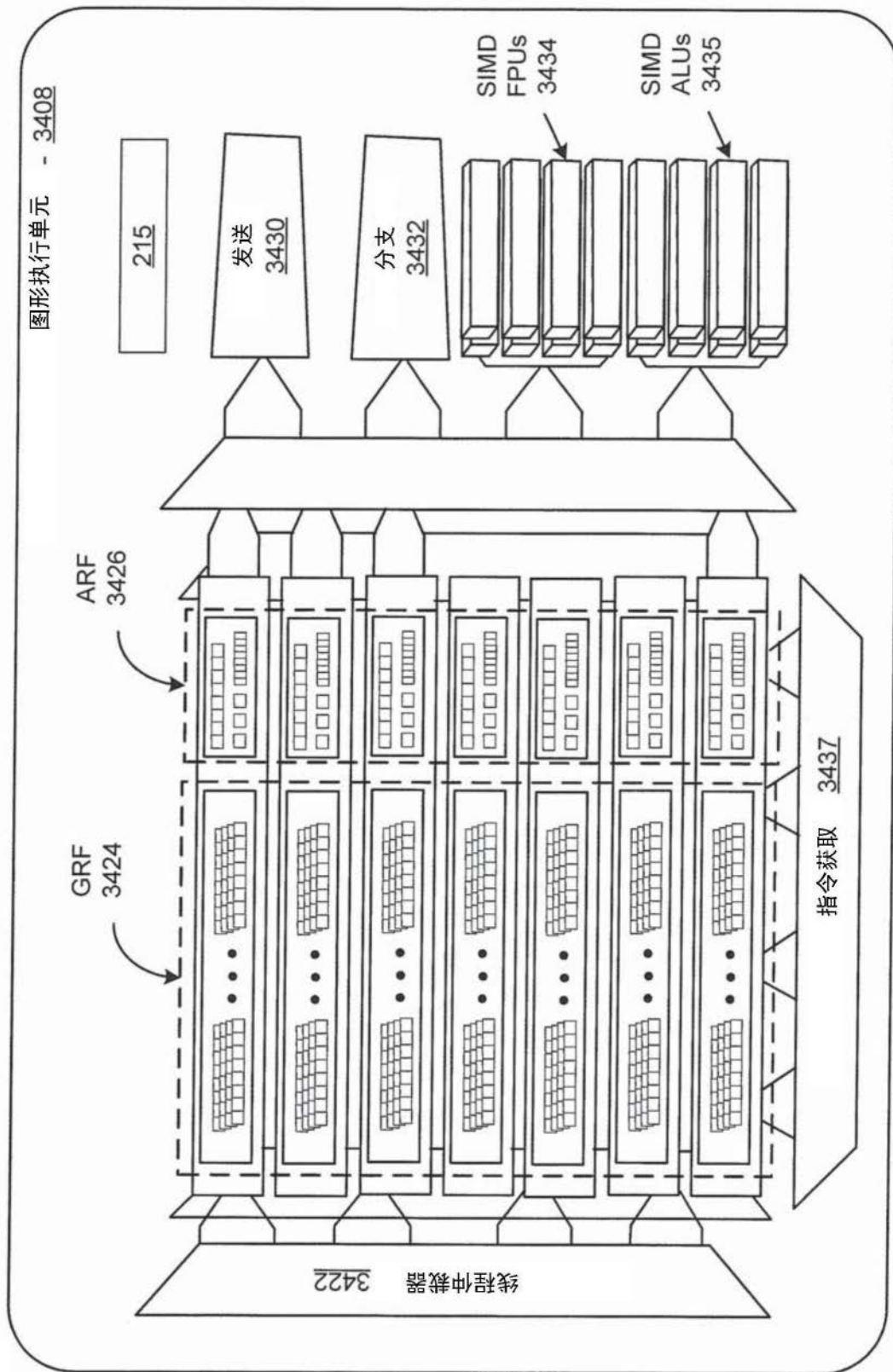


图34B

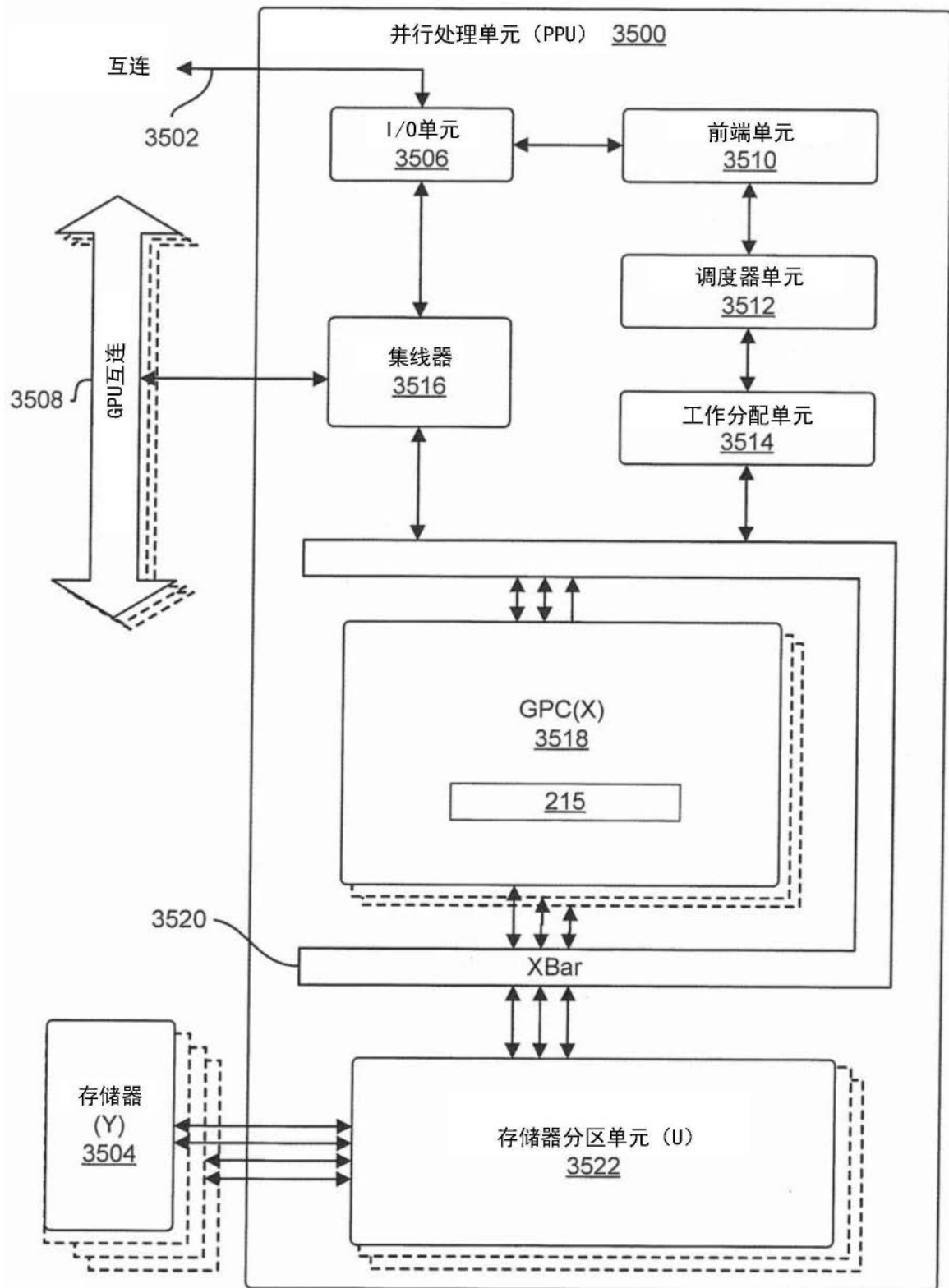


图35

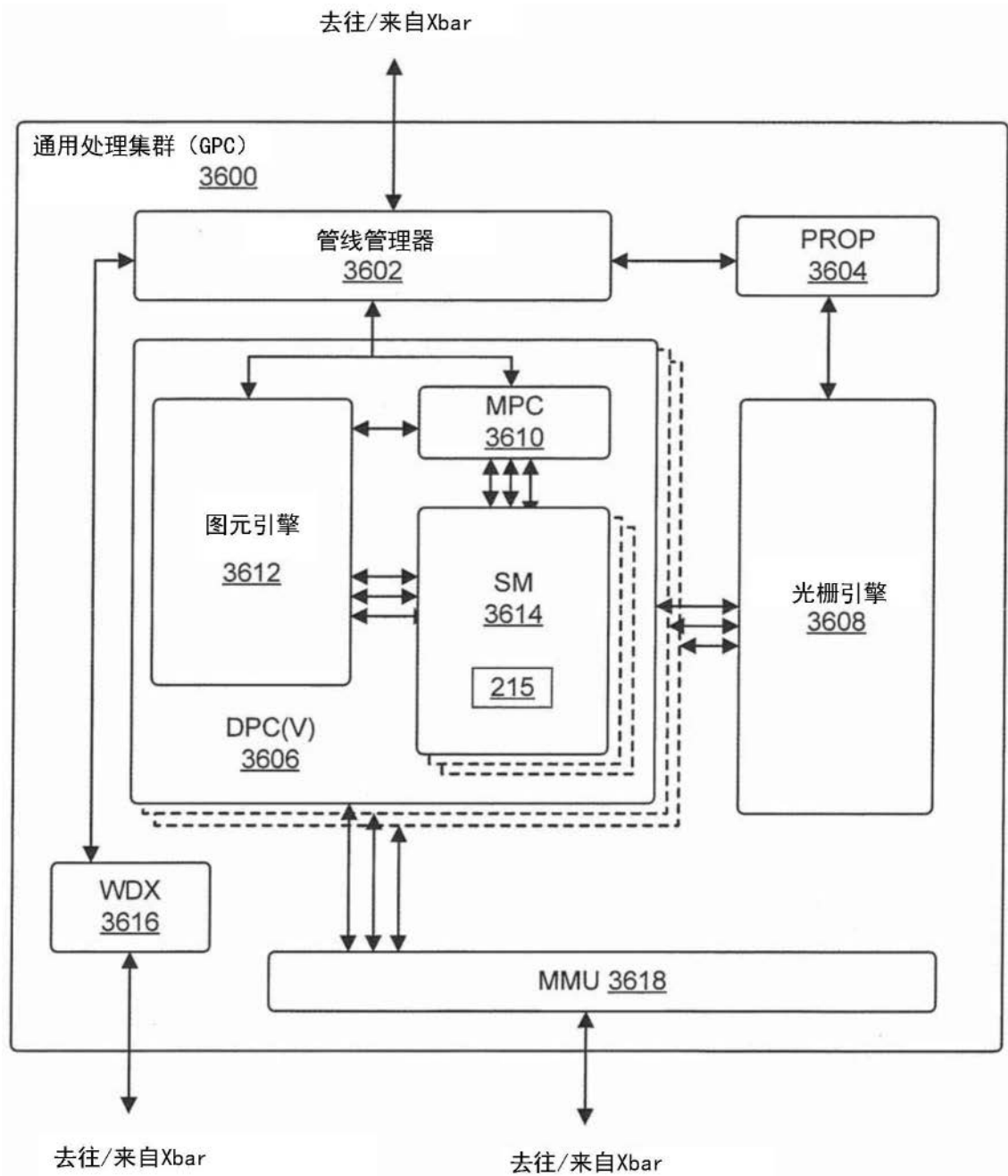


图36

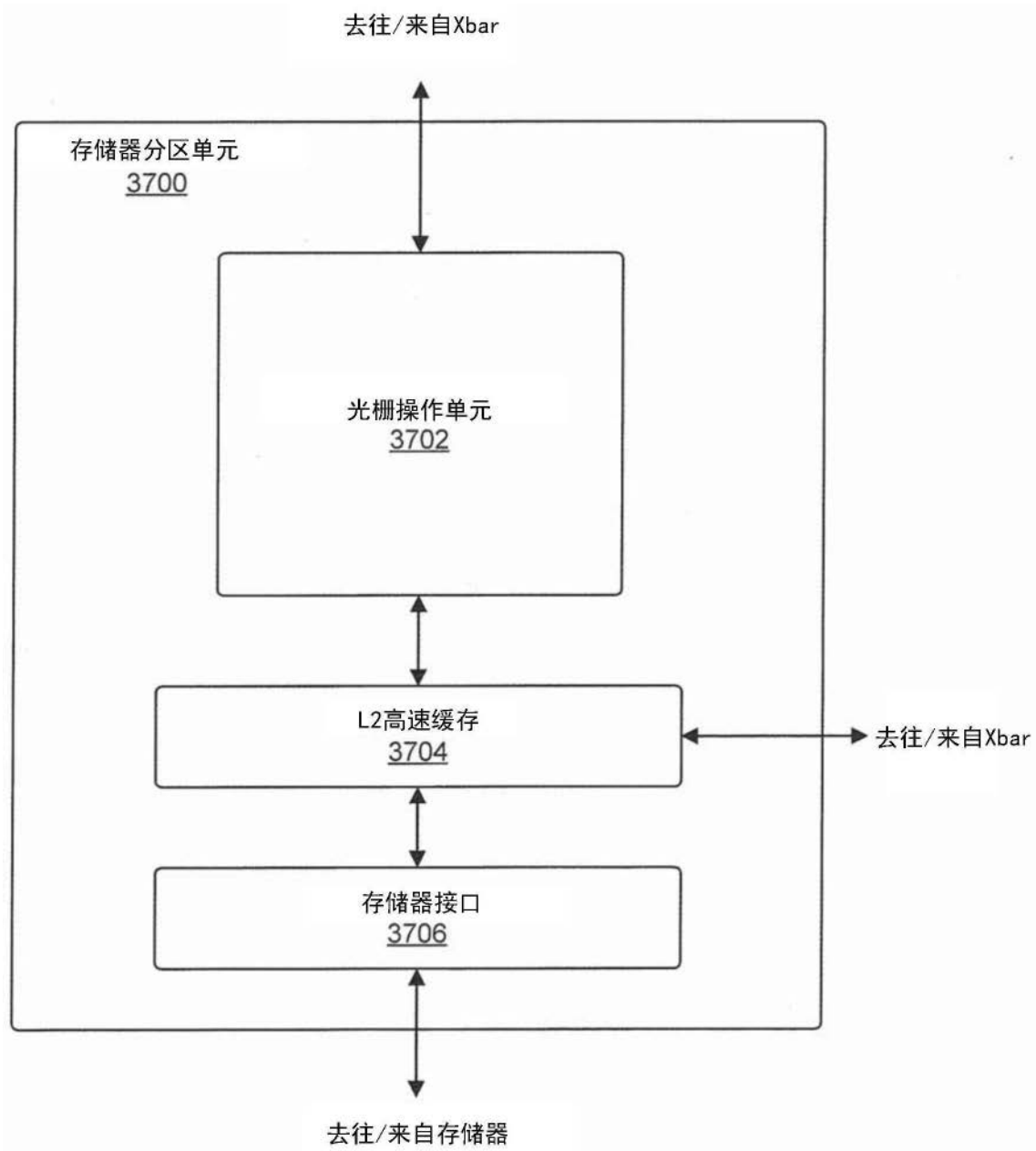


图37

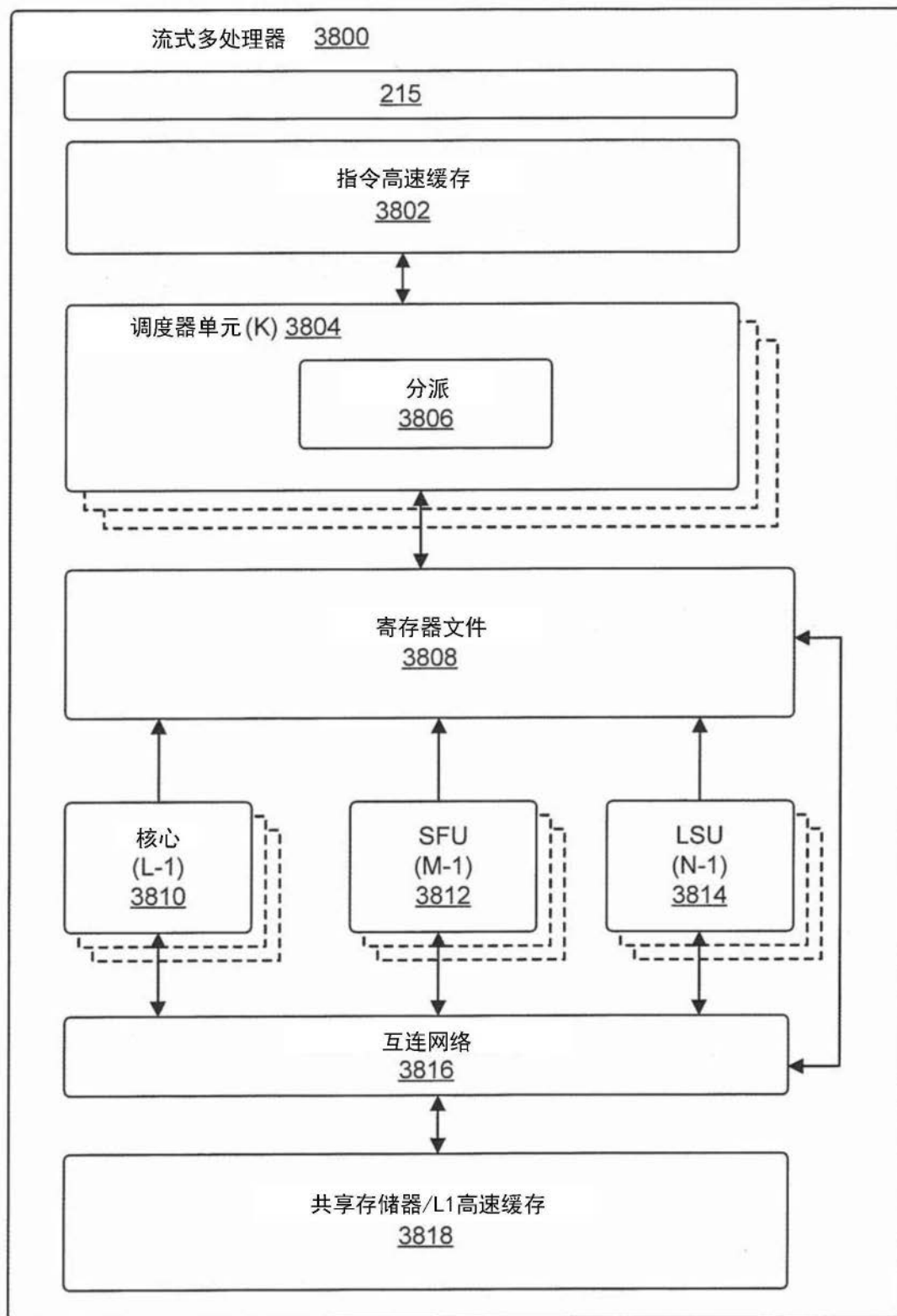


图38