



(10) 授权公告号 CN 111630505 B

(45) 授权公告日 2024. 06. 28

(21) 申请号 201980009631.0

(22) 申请日 2019.01.23

(65) 同一申请的已公布的文献号
申请公布号 CN 111630505 A

(43) 申请公布日 2020.09.04

(30) 优先权数据
62/621,368 2018.01.24 US

(85) PCT国际申请进入国家阶段日
2020.07.22

(86) PCT国际申请的申请数据
PCT/US2019/014801 2019.01.23

(87) PCT国际申请的公布数据
W02019/147708 EN 2019.08.01

(73) 专利权人 阿里巴巴集团控股有限公司

地址 英属开曼群岛大开曼资本大厦一座四
层847号邮箱

(72) 发明人 周擎岗 金铃铃

(74) 专利代理机构 北京清源汇知识产权代理事
务所(特殊普通合伙) 11644
专利代理师 冯德魁 张艳梅

(51) Int.Cl.
G06F 15/173 (2006.01)
H04L 45/02 (2022.01)

(56) 对比文件
CN 102063408 A, 2011.05.18
US 2009064140 A1, 2009.03.05
US 2010111088 A1, 2010.05.06

审查员 刘启军

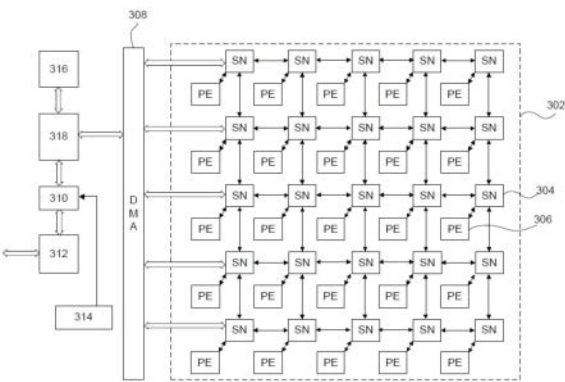
权利要求书2页 说明书9页 附图8页

(54) 发明名称

深度学习加速器系统及其方法

(57) 摘要

本公开涉及一种机器学习加速器系统以及使用该机器学习加速器系统传输数据的方法。机器学习加速器系统可以包括交换网络,该交换网络包括交换节点阵列和处理元件阵列。处理元件阵列的每个处理元件与交换节点阵列的交换节点相连接,并被配置为生成可经由交换节点传输的数据。该方法可以包括:使用交换节点从数据源接收输入数据;基于输入数据使用与所述交换节点相连接的处理元件生成输出数据。该方法可以包括使用交换节点将所生成的输出数据传输到目的地处理元件。



1. 一种机器学习加速器系统,包括:交换网络,

所述交换网络包括:

交换节点阵列;和

处理元件阵列,其中所述处理元件阵列中的每个处理元件与所述交换节点阵列中的一交换节点相连接,所述处理元件阵列中的每个处理元件被配置为生成可通过所述交换节点传输的数据,所生成的数据在一个或多个数据包中传输;所述处理元件阵列提供数据所需的计算资源和数据存储资源;

所述交换节点阵列的交换节点被配置为基于所述交换节点阵列或处理元件阵列中的至少一个的预定义配置,沿所述交换网络中的路径传输数据包,包括:基于所述交换节点的操作状态临时推回所述数据包;

所述机器学习加速器系统包括全局管理器,所述全局管理器包括一个或多个任务管理器,一个或多个所述任务管理器被配置为与一个或多个所述处理元件配合。

2. 根据权利要求1所述的系统,还包括:交换节点阵列的目的地交换节点和与所述目的地交换节点相连接目的地处理元件。

3. 根据权利要求2所述的系统,其中,所述一个或多个数据包包括与所述目的地处理元件的位置、所述目的地处理元件内的存储位置以及所述生成的数据相关的信息。

4. 根据权利要求3所述的系统,其中,与所述目的地处理元件的位置相关的信息包括在所述处理元件阵列内的所述目的地处理元件(x,y)坐标。

5. 根据权利要求3和4中的任一项所述的系统,其中,基于对所述交换网络中的数据流模式的分析,沿一路径传输所述数据包。

6. 根据权利要求5所述的系统,其中,所述路径包括水平路径、垂直路径或二者的组合。

7. 根据权利要求1所述的系统,其中,所述交换节点阵列的交换节点被配置为基于所述交换节点的操作状态来拒绝接收所述数据包。

8. 根据权利要求1所述的系统,其中,所述交换节点阵列中的交换节点被配置为基于所述交换节点的操作状态来修改所述数据包的路径。

9. 根据权利要求1所述的系统,其中,所述处理元件包括:

处理器核心,所述处理器核心被配置为生成所述数据;和

存储器缓冲器,所述存储器缓冲器被配置为存储所生成的数据。

10. 一种在机器学习加速器系统中传输数据的方法,该方法包括:

使用交换网络的交换节点阵列中的交换节点从数据源接收输入数据;

基于所述输入数据,使用连接到所述交换节点并且是处理元件阵列的一部分的处理元件生成输出数据,包括形成一个或多个数据包;所述处理元件阵列提供数据所需的计算资源和数据存储资源;

使用所述交换节点通过所述交换网络将生成的所述输出数据传输到处理元件阵列的目的地处理元件;

基于所述交换节点阵列或所述处理元件阵列的预定配置,沿所述交换网络中的路径传输所述一个或多个数据包,包括:基于所述交换节点的操作状态临时推回所述数据包;

所述机器学习加速器系统包括全局管理器,所述全局管理器包括一个或多个任务管理器,一个或多个所述任务管理器被配置为与一个或多个所述处理元件配合。

11. 根据权利要求10所述的方法, 所述一个或更多个数据包包含与目的地处理元件在所述处理元件阵列内的位置、所述目的地处理元件内的存储位置、以及所生成的输出数据有关的信息。

12. 根据权利要求11所述的方法, 其进一步包括, 将所生成的输出数据存储于所述处理元件阵列内的目的地处理元件的存储器缓冲器中。

13. 根据权利要求11和12中的任一项所述的方法, 其中, 基于对所述交换网络中的数据流模式的分析, 沿所述交换网络中的路径来传输所述数据包。

14. 根据权利要求13所述的方法, 其中, 所述路径包括水平路径、垂直路径或二者的组合。

15. 根据权利要求10所述的方法, 其中, 所述交换节点阵列的交换节点被配置为基于所述交换节点阵列的所述交换节点的操作状态来修改所述一个或更多个数据包的路径。

16. 根据权利要求10所述的方法, 其中, 所述交换节点阵列中的交换节点被配置为基于所述交换节点的操作状态拒绝接收所述数据包。

17. 一种非暂时性计算机可读介质, 其存储有指令集, 该指令集可由机器学习加速器系统的一个或更多个处理器执行, 以使所述机器学习加速器系统执行用于传输数据的方法, 该方法包括:

生成路由指令, 用于将由处理元件阵列的处理元件生成的输出数据传输到处理元件阵列的目的地处理元件, 所述输出数据基于由所述处理元件通过交换网络接收的输入数据而生成, 包括形成一个或更多个数据包, 其中, 所述处理元件阵列的每个处理元件与所述交换网络的交换节点阵列中的一交换节点相连接; 所述处理元件阵列提供数据所需的计算资源和数据存储资源;

基于所述交换节点阵列或所述处理元件阵列的预定配置, 沿所述交换网络中的路径传输所述一个或更多个数据包, 包括: 基于所述交换节点的操作状态临时推回所述数据包;

所述机器学习加速器系统包括全局管理器, 所述全局管理器包括一个或多个任务管理器, 一个或多个所述任务管理器被配置为与一个或多个所述处理元件配合。

18. 根据权利要求17所述的非暂时性计算机可读介质, 其中, 可由所述机器学习加速器系统的一个或更多个处理器执行的指令集使所述机器学习加速器系统进一步执行:

所述一个或更多个数据包包括与目的地处理元件在处理元件阵列内的位置、目的地处理元件内的存储位置以及所生成的输出数据有关的信息。

深度学习加速器系统及其方法

[0001] 相关申请的交叉引用

[0002] 本申请基于2018年1月24日提交的美国临时申请No.62/621368题为“使用具有二维处理单元阵列的轻量级网格网络的深度学习加速器方法”并要求其优先权,上述申请的全部内容通过引用合并于此。

背景技术

[0003] 随着整个业务部门中基于神经网络的深度学习应用的呈指数增长,基于商品中央处理单元(CPU)/图形处理单元(GPU)的平台不再是支持性能、功率和经济可扩展性方面不断增长的计算需求的合适计算基础。开发神经网络处理器以加速基于神经网络的深度学习应用程序已在许多业务领域获得了重大关注,包括成熟的芯片制造商、初创公司以及大型互联网公司。

[0004] 现有的神经处理单元(NPU)或张量处理单元(TPU)具有可编程的确定性执行流水线。该流水线的关键部分包括一个具有256x256个8位乘法器-累加器单元(MAC)的矩阵单元和一24MiB存储器缓冲器。但是,随着半导体技术扩展到7nm节点,晶体管密度有望增加10倍以上,在此配置下,实现有效的数据传输可能需要增加矩阵单元的大小和缓冲器的大小,这可能会带来更多的挑战。

发明内容

[0005] 本公开涉及一种机器学习加速器系统和用于在其中交换数据的方法。所述机器学习加速器系统可以包括交换网络,该交换网络包括交换节点阵列和处理元件阵列。处理元件阵列的每个处理元件可以与交换节点阵列的交换节点相连接,并且被配置为生成可经由所述交换节点传输的数据。可以在一个或多个数据包中传输所生成的数据,该一个或多个数据包包括与目的地处理元件的位置、目的地处理元件内的存储位置以及所生成的数据有关的信息。

[0006] 本公开提供了一种在机器学习加速器系统中传输数据的方法。该方法可以包括使用交换网络的交换节点阵列中的交换节点从数据源接收输入数据。该方法可以包括:使用连接到所述交换节点并且是处理元件阵列的一部分的处理元件,基于所述输入数据来生成输出数据;以及使用所述交换节点经由所述交换网络将所生成的输出数据传输到所述处理元件阵列的目的地处理元件。

[0007] 与一些公开的实施例一致,提供了一种计算机可读存储介质,其包括可由至少一个处理器执行以执行前述方法的一组指令。

[0008] 与其他公开的实施例一致,非暂时性计算机可读存储介质可以存储由至少一个处理设备执行并执行本文所述的前述方法的程序指令。

附图说明

[0009] 在以下详细描述和附图中示出了本公开的实施例和各个方面。图中所示的各种特

征未按比例绘制。

[0010] 图1示出了根据本公开的实施例的示例性深度学习加速器系统。

[0011] 图2示出了根据本公开的实施例的示例性深度学习加速器系统的框图。

[0012] 图3A示出了根据本公开实施例的示例性基于网格的深度学习加速器系统的框图。

[0013] 图3B示出了根据本公开的实施例的示例性深度学习加速器系统的处理元件的框图。

[0014] 图4示出了根据本公开的实施例的示例性数据包的框图。

[0015] 图5示出了根据本公开实施例的用于深度学习加速器系统中的数据传输的示例性路径。

[0016] 图6示出了根据本公开实施例的用于深度学习加速器系统中的数据传输的示例性路径。

[0017] 图7示出了根据本公开实施例的用于深度学习加速器系统中的数据传输的示例性路径。

[0018] 图8示出了根据本公开实施例的用于深度学习加速器系统中的数据传输的示例性方法的处理流程图。

具体实施方式

[0019] 现在将详细参考示例性实施例,其示例在附图中示出。以下描述参考附图,在所述附图中除非另有说明,否则不同附图中的相同数字表示相同或相似的元件。在以下对示例性实施例的描述中阐述的实施方式并不代表与本发明一致的所有实施方式。相反,它们仅仅是与如所附权利要求中所述的与本发明有关的方面一致的装置和方法的示例。

[0020] 如上所述,常规的加速器具有几个缺陷。例如,常规的图形处理单元(GPU)可能具有数千个着色器核心,这些着色器核心具有完整的指令集、动态的工作调度程序以及复杂的存储器层次结构,从而导致大量的功耗和深度学习工作负载的额外工作。

[0021] 常规数据处理单元(DPU)可以具有基于数据流的粗粒度可重构体系结构(CGRA)。此CGRA可以配置为32x32集群的网格,每个集群可以配置为16个数据流处理元件(PE)。PE可以将数据直接传递到其邻居,使得数据可通过该网格传递。这可能需要PE花费几个周期来传递数据,而非专注于计算,从而使数据流效率低下。

[0022] 本发明的实施例克服了常规加速器的这些问题。例如,所述实施例提供了一种轻量级的交换网络,从而允许PE专注于计算。另外,计算和存储资源分布在多个PE中。借助于2D网格连接,可以在PE之间传递数据。软件可以更灵活划分神经网络工作负载和数据为PE阵列,并据此对数据流进行编程。由于类似的原因,在不增加打包更多工作和数据的难度的情况下,很容易添加额外的资源。

[0023] 图1示出了根据本公开的实施例的示例性深度学习加速器架构100。在本公开的上下文中,深度学习加速器系统也可以被称为机器学习加速器。机器学习和深度学习在这里可以互换使用。如图1所示,深度学习加速器系统架构结构100可以包括片上通信系统102、主机存储器104、存储器控制器106、直接存储器访问(DMA)单元108、联合测试动作组(JTAG)/测试访问端(TAP)控制器110、外围接口112、总线114、全局存储器116等。应当理解,片上通信系统102可以基于所传递的数据来执行算法操作。此外,加速器系统架构100可以

包括全局存储器116,该全局存储器具有片上存储块(例如4个8GB第二代高带宽存储器(HBM2)的块)以用作主存储器。

[0024] 片上通信系统102可以包括全局管理器122和多个处理元件124。全局管理器122可以包括一个或更多个任务管理器126,一个或更多个任务管理器126被配置为与一个或更多个处理元件124配合。每个任务管理器126可以与一个为神经网络提供突触/神经元电路的处理元件阵列124相关联。例如,图1的处理元件的顶层可以提供代表神经网络的输入层的电路,而处理元件的第二层可以提供代表神经网络的一个或更多个隐藏层的电路。如图1所示,全局管理器122可以包括两个任务管理器126,其被配置为与处理元件124的两个阵列相配合。在一些实施例中,加速器系统架构100可以被称作神经网络处理单元(NPU)架构100。

[0025] 处理元件124可以包括一个或更多个处理元件,每个处理元件包括单指令多数据(SIMD)体系架构,该SIMD体系架构包括一个或更多个配置为在全局管理器122的控制下对通信的数据执行一个或更多个操作(例如,乘法、加法、乘法累加等)的处理单元。处理元件124可以包括核心和存储器缓冲器。每个处理元件可以包括任何数量的处理单元。在一些实施例中,处理元件124可以被认为是瓦片或类似物。

[0026] 主机存储器104可以是例如主机CPU的存储器的片外存储器。例如,主机存储器104可以是双倍数据速率同步动态随机存取存储器(DDR-SDRAM)存储器或类似物。与集成在一个或更多个处理器中的片上存储器相比,主机存储器104可以配置为以较低的访问速度存储大量数据,以用作更高级别的高速缓存。

[0027] 存储器控制器106可以管理对全局存储器116内的存储器块(例如HBM2)的数据读写,例如,存储器控制器106可以管理来自外部芯片通信系统(例如,来自DMA单元108或与另一个NPU对应的DMA单元)或片上通信系统102(例如,经由全局管理器122的任务管理器126控制的2D网格来自处理元件124中的本地存储器)的数据读写。此外,虽然在图1中示出了一个存储控制器,但是,参照图1,应当理解,在NPU架构100中可以提供一个以上的存储器控制器。例如,在全局存储器116内的每个存储器块(例如,HBM2)可以存在一个存储器控制器。

[0028] 存储器控制器106可以生成存储器地址并且启动存储器读取或写入周期。存储器控制器106可包含可由一个或更多个处理器写入和读取的几个硬件寄存器。这些寄存器可以包括存储器地址寄存器、字节计数寄存器、一个或更多个控制寄存器以及其他类型的寄存器。这些寄存器可以指定源、目的地、传输方向(从输入/输出(I/O)设备读取或写入I/O设备)、传输单元的大小、一个突发中传输的字节数、和/或存储控制器的其他典型功能中的某种组合。

[0029] DMA单元108可以辅助在主机存储器104和全局存储器116之间传送数据。另外,DMA单元108可以辅助在多个加速器之间传送数据。DMA单元108可以允许片外设备访问片内和片外存储器而不会引起CPU中断。因此,DMA单元108还可生成存储器地址并启动存储器读取或写入周期。DMA单元108还可包含可由一个或更多个处理器写入和读取的几个硬件寄存器,包括存储器地址寄存器、字节计数寄存器、一个或更多个控制寄存器、以及其他类型的寄存器。这些寄存器可以指定源、目的地、传输方向(从输入/输出(I/O)设备读取或写入I/O设备)、传输单元的大小、和/或一个突发中传输的字节数中的某种组合。应当理解,加速器架构100可以包括第二DMA单元,其可以用于在其他加速器架构之间传输数据,以允许多个加速器架构直接通信而不涉及主机CPU。

[0030] JTAG/TAP控制器110可以指定专用调试端口,该专用调试端口实现用于对加速器的低开销访问的串行通信接口(例如,JTAG接口),而无需直接外部访问系统地址和数据总线。JTAG/TAP控制器110还可以具有片上测试访问接口(例如,TAP接口),该片上测试访问接口被配置为实现协议以访问呈现各部分的芯片逻辑电平和设备能力的一组测试寄存器。

[0031] 外围接口112(例如PCIe接口),如果存在的话,可以用作(并且通常是)芯片间总线,以提供加速器和其他设备之间的通信。

[0032] 总线114包括芯片内总线和芯片间总线。芯片内总线将所有内部组件相互连接,这是系统结构所要求的。尽管并非所有组件都与每个其他组件都连接,但所有组件确实与它们需要与之通信的其他组件建立了某种连接。芯片间总线将加速器与其他设备相连,例如片外存储器或外围设备。通常,如果存在外围接口112(例如,芯片间总线),则总线114仅与芯片内总线有关,尽管在某些实现中,总线114仍可能与专门的总线间通信有关。

[0033] 尽管图1的加速器架构100通常针对NPU架构(如下文进一步描述),但是应当理解,所公开的实施例可以应用于用于加速诸如深度学习之类的某些应用的任何类型的加速器。此类芯片可以是例如GPU、具有矢量/矩阵处理能力的CPU或用于深度学习的神经网络加速器。SIMD或矢量结构通常用于支持具有数据并行性的计算设备,例如图形处理和深度学习。

[0034] 现在参考图2,其示出了根据本公开的实施例的示例性深度学习加速器系统200的框图。深度学习加速器系统200可以包括神经网络处理单元(NPU)202、NPU存储器204、主机CPU 208、与主机CPU 208相关联的主机存储器210、和磁盘212。

[0035] 如图2所示,NPU 202可以通过外围接口(例如,图1的外围接口112)连接到主机CPU 208。如本文所提到的,神经网络处理单元(例如,NPU 202)可以是用于加速神经网络计算任务的计算设备。在一些实施例中,NPU 202可以被配置为用作主机CPU 208的协处理器。

[0036] 在一些实施例中,NPU 202可以包括编译器(未示出)。编译器可以是将以一种编程语言编写的计算机代码转换为NPU指令以创建可执行程序或计算机软件。在机器应用中,编译器可执行各种操作,例如预处理、词法分析、解析、语义分析、输入程序到中间表示的转换、代码优化、代码生成或其组合。

[0037] 在一些实施例中,编译器可以在主机单元(例如,图2的主机CPU 208或主机存储器210)上,被配置为将一个或多个命令推入NPU202。基于这些命令,任务管理器(例如,图1的任务管理器126)可以将任何数量的任务分配给一个或多个处理元件(例如,图1的处理元件124)。所述命令中的一些可以指示DMA单元(例如,图1的DMA单元108)将指令和数据从主机存储器(例如,图1的主机存储器104)加载到全局存储器中。然后,可以将加载的指令分发给分配有相应任务的每个处理元件124,并且一个或多个处理元件124可以处理这些指令。

[0038] 应当理解,由处理元件接收的前几条指令可以指示处理元件将数据从全局存储器加载/存储到处理元件的一个或多个本地存储器中(例如,处理元件的存储器或每个活动处理元件的本地内存)。然后,每个处理元件可以启动指令流水线,该指令流水线包括从本地存储器中获取指令(例如,经由获取单元)、对指令进行解码(例如,经由指令解码器)以及生成本地存储器地址(例如,与操作数相对应)、读取源数据、执行或加载/存储操作,然后写回结果。

[0039] 主机CPU 208可以与主机存储器210和磁盘212相关联。在一些实施例中,主机存储

器210可以是与主机CPU 208关联的集成存储器或外部存储器。主机存储器210可以是本地或全局存储器。在一些实施例中,磁盘212可以包括配置为向主机CPU 208提供附加存储器的外部存储器。

[0040] 现在参照图3A,其示出了根据本公开实施例的示例性深度学习加速器系统300。深度学习加速器系统300可以包括一交换网络302,该交换网络302包括交换节点304阵列和处理元件306阵列、DMA单元308、由控制单元314控制的主机CPU 310、外围接口312、高带宽存储器316和高带宽存储器接口318。可以理解,深度学习加速器系统300可以包括这里未示出的其他组件。

[0041] 在一些实施例中,交换网络302可以包括交换节点304的阵列。可以以形成交换节点304的二维(2D)阵列的方式来布置交换节点304。在一些实施例中,如图3A所示,交换网络302可以包括一交换网络,该交换网络包含使得交换网络中的每个交换节点304可以与紧邻的交换节点304连接的交换节点的2D网状连接。交换节点304可以被配置为将数据路由至交换网络302或者从交换网络302路由数据,或在交换网络302内路由数据。数据可以从交换网络302的另一个交换节点304内部接收,也可以从DMA单元308外部接收。路由数据可以包括接收数据并将其传输到其他相关组件,例如,深度学习加速器系统300的另一个交换节点304或处理元件306。在一些实施例中,交换节点304可以从DMA 308、处理元件306和交换网络302的一个或更多个相邻交换节点304接收数据。

[0042] 如图3A所示,每个交换节点304可以与对应的处理元件306相关联。处理元件306可以类似于图1的处理元件124。深度学习加速器系统300可以包含处理元件306的2D阵列,每个处理元件与交换网络302的对应交换节点304连接。处理元件306可以配置为生成数据包形式的数据(稍后描述)。在一些实施例中,处理元件306可以被配置为基于计算机可执行程序、软件、固件或预定义的配置来生成数据。处理元件306也可以被配置为将数据发送到交换节点304。

[0043] 在一些实施例中,交换节点304可以被配置为基于交换节点304的操作状态来响应处理元件306。例如,如果交换节点304正忙于路由数据包,则交换节点304可以拒绝数据包或临时推回来自处理元件306的数据包。在一些实施例中,交换节点304可以重新路由数据包,例如,交换节点304可以根据操作状态或整个系统状态,将数据包的流向从水平路径改变为垂直路径,或者从垂直路径改变为水平路径。

[0044] 在一些实施例中,交换网络302可以包括交换节点304的2D阵列,每个交换节点连接到对应的单独的处理元件306。交换节点304可以被配置为将数据从一个位置传输到另一位置,而处理元件306可以被配置为计算输入数据以生成输出数据。计算和传输资源的这种分配可以允许交换网络302是轻量级的和高效的。轻量级2D交换网络可能具有此处讨论的部分或全部优点。

[0045] (i) 基于简单交换的设计-提出的2D交换网络包括简单的交换来控制网络内的数据流。交换节点的使用实现了处理元件的2D阵列之间的点对点通信。

[0046] (ii) 高计算效率-数据流管理(包括在网络的交换节点之间交换和传输数据)由可执行程序(例如软件或固件)执行。该软件允许根据数据流模式、工作负载特征、数据流量等来调度数据流,从而形成高效的深度学习加速器系统。

[0047] (iii) 增强的性能和较低的功耗-提出的轻量级交换网络依靠去中心化的资源分

配,从而提高了整个系统的性能。例如,计算资源和数据存储资源分布在处理元件阵列中,而不是在中央核心或处理元件中枢。简单的基于网格的连接可以实现处理元件之间的通信。

[0048] (iv) 设计的灵活性和可扩展性-软件可以将神经网络的工作负载和数据灵活地划分至处理元件阵列,并据此对数据流进行编程。这样可以添加资源以计算更大数量的数据,同时保持计算效率和整体系统效率。

[0049] (v) 数据路由策略的灵活性-提出的2D交换网络可以不需要用于死锁检测、避免拥塞或数据冲突管理的复杂流控制机制。由于网状网络和连接性,可以采用简单有效的路由策略。

[0050] (vi) 软件兼容性-软件或固件可以在运行前基于对工作负载、数据流模式和数据存储的静态分析,调度处理元件的任务以生成避免拥塞和死锁的数据包。

[0051] 在一些实施例中,DMA单元308可以类似于图1的DMA单元108。DMA单元308可以包括骨干网,并且深度学习加速器系统可以包括两个单独的总线系统(例如,图1的总线114)。一个总线系统可以使得在交换网络的交换节点304之间进行通信,而另一总线系统可以使得DMA单元308和骨干网之间进行通信。DMA单元308可配置为控制和组织流入和流出交换机网络302的数据流。

[0052] 深度学习加速器系统300可以包括主机CPU 310。在一些实施例中,主机CPU 310可以与控制单元314电连接。主机CPU 310也可以连接到外围接口312和高带宽接口318。DMA单元308可以通过高带宽存储器接口318与主机CPU 310或高带宽存储器316通信。在一些实施例中,高带宽存储器316可以类似于图1所示的深度学习加速器系统100的全局存储器116。

[0053] 现在参照图3B,其示出了根据本公开的实施例的示例性处理元件的框图。除了其他组件之外,处理元件306可以包括处理核心320和存储器缓冲器322。处理核心320可以被配置为处理从DMA单元308或从交换网络302的另一个处理元件306接收的输入数据。在一些实施例中,处理核心320可以被配置为处理输入数据、以数据包的形式生成输出数据,以及将所生成的输出数据包传递给相邻处理元件306。视情况而定,存储器缓冲器322可以包括本地存储器、全局共享存储器或其组合。存储器缓冲器322可以被配置为存储输入数据或输出数据。

[0054] 现在参考图4,其示出了根据本公开实施例的示例性数据包。数据包400可以被格式化为包含与目的地位置和数据本身相关联的信息。在一些实施例中,数据包400可以包括与目的地位置有关的信息和要被传送到目的地位置的数据410。与目的地位置有关的信息可以包括交换网络中的目的地处理元件306的(X,Y)坐标以及数据偏移。在一些实施例中, PE_x 可以包括目的地处理元件306的X坐标404, PE_y 可以包括目的地处理元件306的Y坐标406,并且 PE_{OFFSET} 可以包括与处理元件306的存储器缓冲器322内的位置相关联的信息。例如,如果存储器缓冲器322是256位存储器并且存储器中的每一行(line)是32位,则存储器具有8行。在这样的配置中, PE_{OFFSET} 信息可以指示数据410所属的存储器内的目的地行号。数据包400可以由交换网络内的交换节点304基于数据业务、数据传输效率、共享的数据类型等并使用一种或多种路由策略来路由。本文讨论了数据的路由策略的一些示例。应当理解,可以适当地采用其他路由策略。

[0055] 图5示出了根据本公开实施例的用于深度学习加速器系统中的数据传输的示例性

路径500。沿着传输路径500传输数据可以包括水平传输数据包502、504、506和508,如图5所示。可以以与图4所示的数据包400类似的方式格式化数据包502、504、506和508。虽然仅示出了四个数据包,但是深度学习加速器系统可以包括数据计算所需的任何数量的数据包。深度学习加速器系统的计算工作量可以被划分并分配给处理元件306。

[0056] 在一些实施例中,如图5所示,水平流水线数据传输是指将数据或包含数据(例如,图4的数据410)的数据包在交换网络中从具有(X,Y)坐标的交换节点304传输到具有(X+i,Y)坐标的交换节点304,其中“i”是一正整数。在一些实施例中,目的地交换节点304可以具有(X-i,Y)坐标。数据包的移动可能是从左到右或从右到左,具体取决于目的地交换节点。

[0057] 作为示例,图5示出了用于四个数据包(例如,数据包502、504、506和508,每个数据包在图中用不同的线格式标注)的数据传输路径。每个数据包的目标位置是(X+4,Y)。这可以在四个周期中完成,称为周期0、周期1、周期2和周期3。每个周期只能由一个交换节点304移动一数据包。在一些实施例中,移动数据包到目的地交换节点所需的周期数可以等于在特定方向上传输数据包所需的交换节点数。在一些实施例中,沿着X方向的行中或沿着Y方向的列中的交换节点304可以被称为深度学习加速器系统的层。

[0058] 在一些实施例中,与交换节点304相关联的处理元件306可以被配置为接收数据包(例如,图4的数据包400或图5的502)并且将数据存储在处理元件306的存储器缓冲器322中。可以基于接收到的数据包的 PE_{OFFSET} 将数据存储存储在存储器缓冲器322内。

[0059] 现在参考图6,图6示出了根据本公开实施例的用于深度学习加速器系统中的数据传输的示例性路径600。沿着传输路径600传输数据可以包括垂直传输数据包602、604和606,如图6所示。数据包602、604和606可以类似于图4所示的数据包400。

[0060] 在一些实施例中,如图6中所示的垂直流水线数据传输是指将数据或包含数据的数据包(例如,图4的数据410)在交换网络中从具有(X,Y)坐标的交换节点304传输到具有(X,Y+i)坐标的交换节点304,其中“i”是一正整数。在一些实施例中,目的地交换节点304可以具有(X,Y-i)坐标。数据包的移动可以是下至上或从上到下,具体取决于目的地交换节点。

[0061] 现在参考图7,其示出了根据本公开实施例的用于在深度学习加速器系统中进行数据传输的示例性路径700。在一些实施例中,处理元件阵列中的处理元件306可以从DMA单元(例如,图3A的DMA单元308)或其他数据源外部接收数据。基于接收到的数据,处理元件306可以生成包括计算数据和所计算的数据的目的地位位置信息的数据包。图7示出了在水平和垂直两个方向上都传输的数据包702、704、706和708。在这样的配置中,可以采用两步处理。在第一步中,数据包702、704、706和708可以沿着Y坐标在垂直方向上传输,直到到达目的地交换节点304为止。在到达目的地Y坐标后,在第二步骤中,数据包702、704、706和708可以沿着X坐标在水平方向上传输,直到到达目的地交换节点304为止。

[0062] 在一些实施例中,数据流的方向可以由软件在执行之前或运行时间之前确定。例如,软件可以在处理元件306生成包括计算结果的输出数据时,以流水线的方式确定水平数据流,并且在处理元件306与其相邻处理元件共享输入数据时,以流水线的方式确定垂直数据流。

[0063] 现在参考图8,其示出了根据本公开的实施例的在深度学习加速器系统(例如,图1的深度学习加速器系统100)中传输数据的示例性方法的处理流程图800。该方法可以包括:

使用交换节点从内部或外部数据源接收数据;使用处理元件基于接收到的输入数据生成输出数据;以及将所述输出数据传输到目的地处理元件。

[0064] 在步骤810中,可以将交换节点(例如,图3A的交换节点304)配置为从数据源接收数据。数据源可以是内部数据源,例如,交换节点阵列中的另一个交换节点或处理元件(例如,图3A的处理元件306)。在一些实施例中,数据源可以是外部数据源,例如,DMA单元(例如,图3A的DMA单元308)。DMA单元可以被配置为控制主机CPU(例如,图3A的主机CPU 310)和2D交换网络(例如,图3A的交换网络302)之间的数据流。在一些实施例中,DMA单元可以与交换网络的一个或更多个交换节点304通信并交换数据。

[0065] DMA单元可以帮助在主机存储器(例如,主机CPU的本地存储器)和高带宽存储器(例如,图3A的高带宽存储器316)之间传输数据。另外,DMA单元可以被配置为在多个处理单元之间传输数据。在一些实施例中,DMA单元可以允许片外设备访问片内和片外存储器而不会引起CPU中断。因此,DMA单元还可生成存储器地址并启动存储器读或写周期。DMA单元还可以包含可以由一个或更多个处理器读写的几个硬件寄存器,包括存储器地址寄存器、字节计数寄存器、一个或更多个控制寄存器以及其他类型的寄存器。

[0066] 交换节点可以被配置为接收输入数据并将接收到的输入数据或输出数据从处理元件传输到交换网络内的目的地位置。网络交换网络可以实现处理元件的2D阵列之间的点对点数据通信。

[0067] 在步骤820中,处理元件(例如,图3A的处理元件306)可以基于内部或外部接收的输入数据来生成输出数据。网络交换网络可以包括处理元件的2D阵列。网络交换网络的每个处理元件可以与至少一个交换节点相关联。在一些实施例中,基于系统设计和性能要求,多个处理元件可以与一个交换节点相关联。

[0068] 处理元件可以包括处理器核心(例如,图3B的处理器核心320)和存储器(例如,图3B的存储器缓冲器322)。处理器核心可以被配置为计算和生成输出数据,而存储器缓冲器可以被配置为存储所生成的输出数据。在一些实施例中,存储器缓冲器还可存储计算输出数据所需的数据和指令。可以以数据包(例如,图4的数据包400)的形式生成和传输输出数据。所述数据包可以被格式化为包括目的地处理元件的(X,Y)坐标、输出数据以及目的地处理元件的存储器缓冲器内数据需要存储的位置。例如,数据包可以包括 PE_x 、 PE_y 、 PE_{OFFSET} 和数据。在此, PE_x 可以指示目的地处理元件的X坐标, PE_y 可以指示目的地处理元件的Y坐标,并且 PE_{OFFSET} 可以指示存储器缓冲器中的存储空间的位置地址(Bit line address)。

[0069] 处理元件可以包括本地存储器或全局共享存储器。处理元件的本地存储器可以由处理元件的处理器核心320访问,而全局共享存储器可以由网络交换网络中的任何处理元件的任何处理器核心访问。

[0070] 在步骤830中,可以基于存储在处理元件的存储器缓冲器中的目的地信息,将生成的输出数据或数据包传输到目的地处理元件。数据可以通过一条或多条路径被传送到目的地处理元件。数据传输路径可以基于网络交换网络中的交换节点阵列或处理元件阵列处的预定义配置。软件、固件或计算机可执行程序可以在运行之前确定路由。

[0071] 在一些实施例中,数据或数据包可以沿着通过至少对数据流模式、数据流流量或数据量等的静态地分析而确定的路线来传输。软件(例如,诸如主机CPU中的编译器)还可以调度处理元件的任务并对处理元件进行编程,以生成避免拥塞和死锁的数据包。确定的路

径可以是如图5所示的水平路径,或图6所示的垂直路径,或者如图7所示的水平和垂直路径的组合,也可以酌情使用其他路由策略。

[0072] 在方法步骤或过程的一般上下文中描述了本文所述的各种示例实施例,该方法步骤或过程可以在一个方面中由体现在计算机可读介质中的计算机程序产品来实现,该计算机程序产品包括计算机可执行指令,例如程序代码,由网络环境中的计算机执行。计算机可读介质可以包括可移动和不可移动存储设备,包括但不限于只读存储器 (ROM)、随机存取存储器 (RAM)、光盘 (CD)、数字通用光盘 (DVD) 等。程序模块可以包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、组件、数据结构等。计算机可执行指令、关联的数据结构和程序模块代表用于执行本文公开的方法的步骤的程序代码的示例。这样的可执行指令或相关联的数据结构的特定序列表示用于实现在这样的步骤或过程中描述的功能的相应动作的示例。

[0073] 在前述说明书中,已经参照可能随实施方式而变化的许多具体细节描述了实施例。可以对所描述的实施例进行某些修改和修改。考虑到本文公开的发明的说明书和实践,其他实施例对于本领域技术人员而言可能是显而易见的。说明书和实施例仅被认为是示例性的,本发明的真实范围和精神由所附权利要求书指示。附图中所示的步骤顺序也旨在仅用于说明目的,并且不旨在限于任何特定的步骤顺序。这样,本领域技术人员可以理解,在实现相同方法的同时,可以以不同顺序执行这些步骤。

[0074] 在附图和说明书中,已经公开了示例性实施例。但是,可以对这些实施例进行许多变化和修改。因此,尽管采用了特定术语,但是它们仅在一般性和描述性意义上使用,而不是出于限制的目的,实施例的范围由所附权利要求限定。

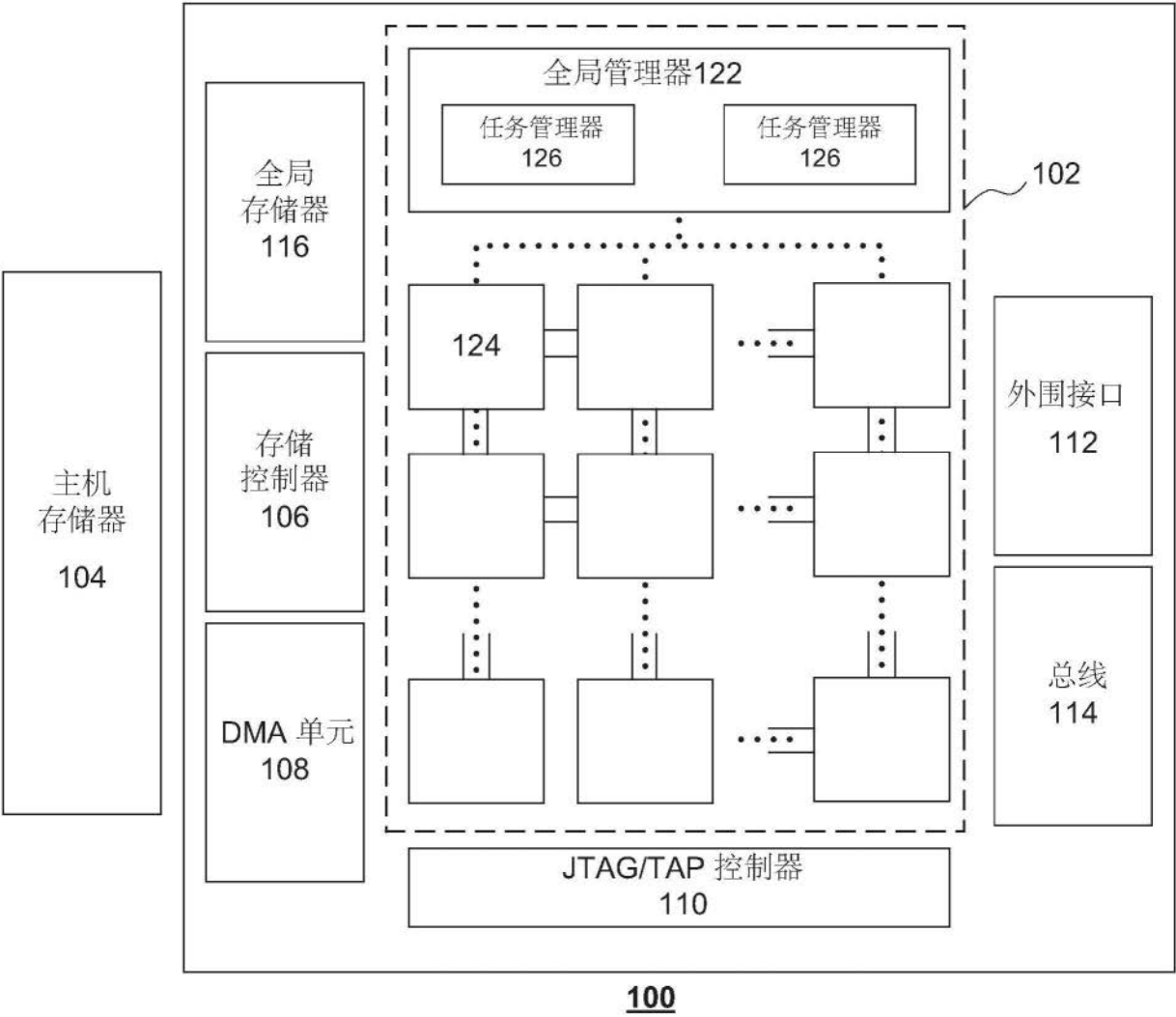


图1

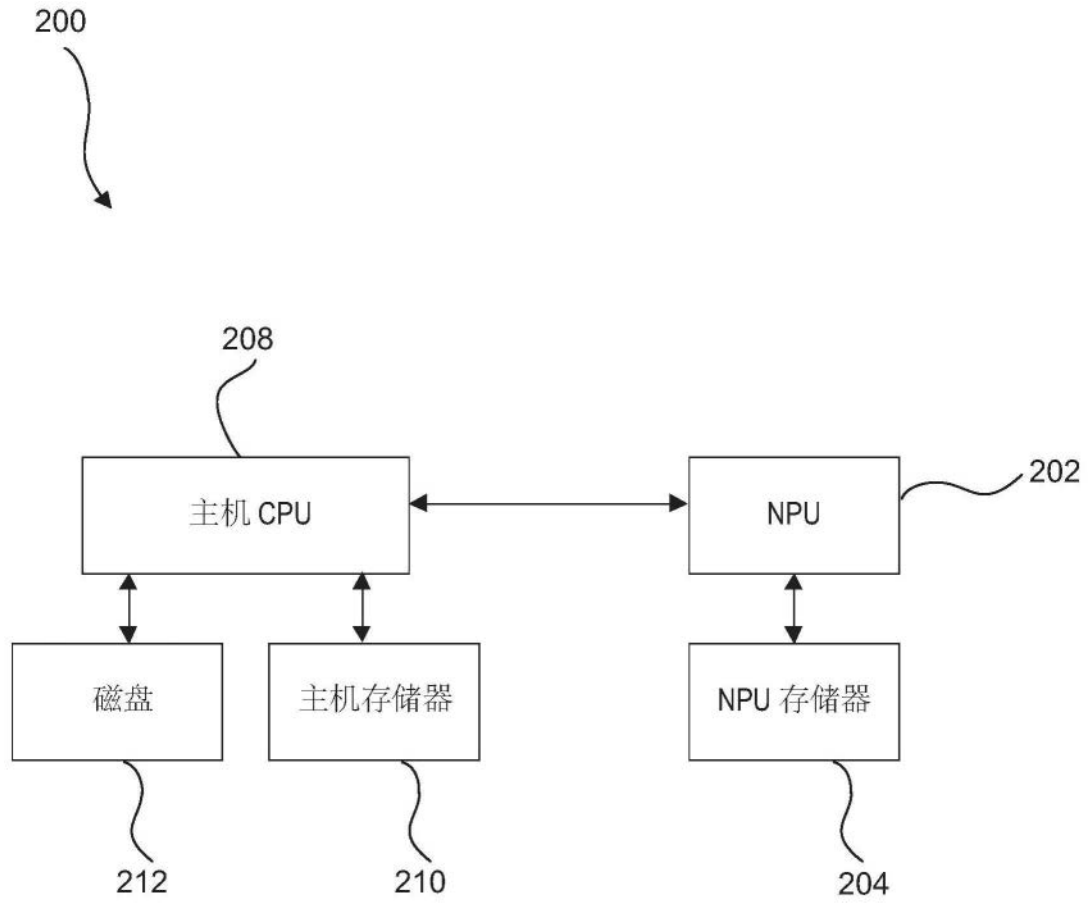


图2

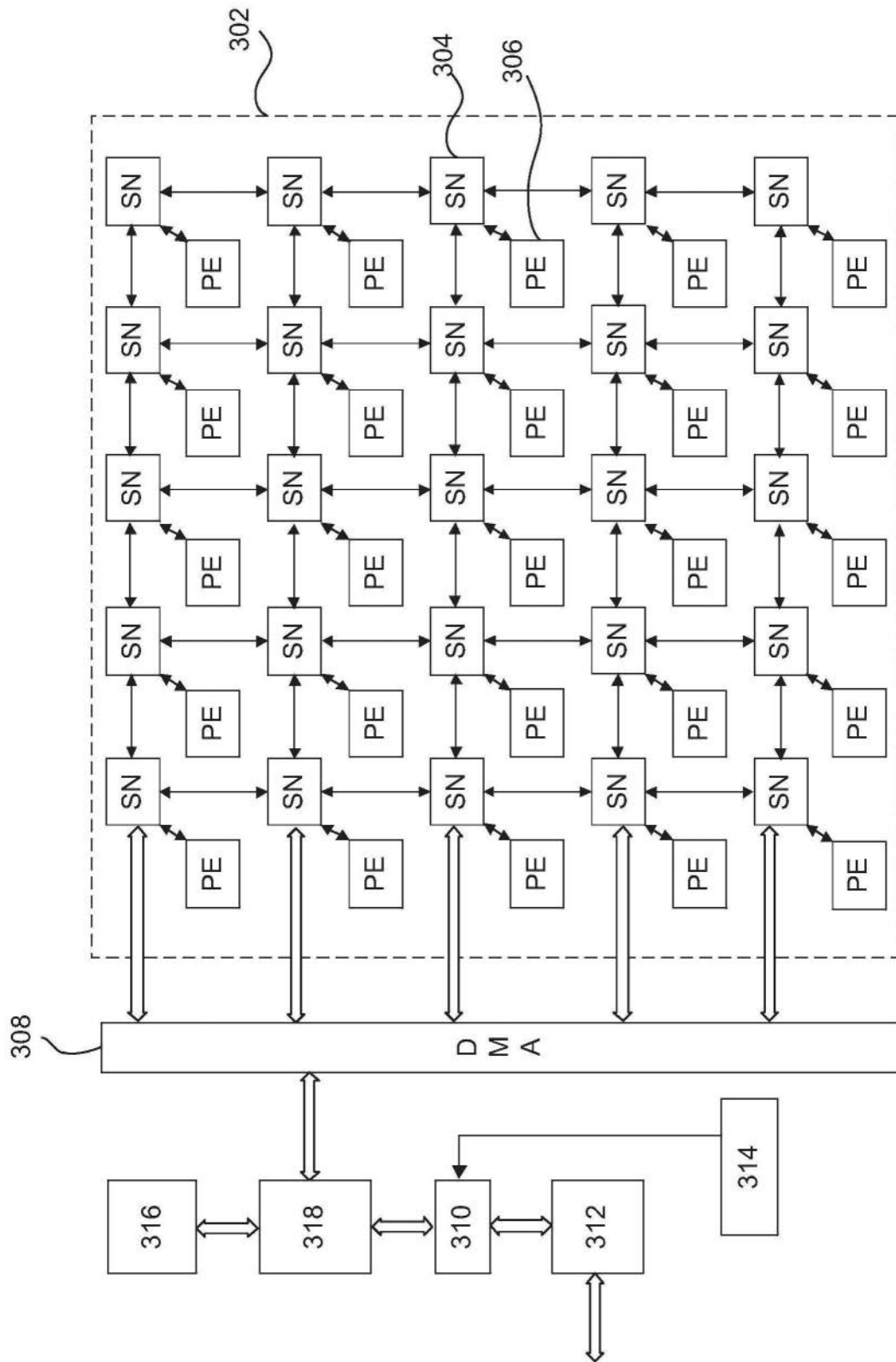


图3A

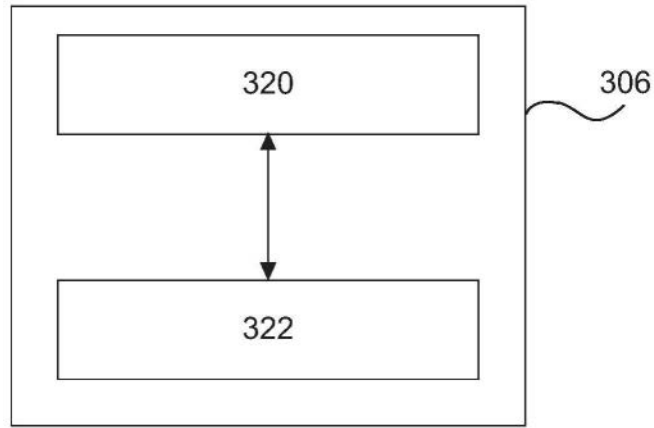


图3B

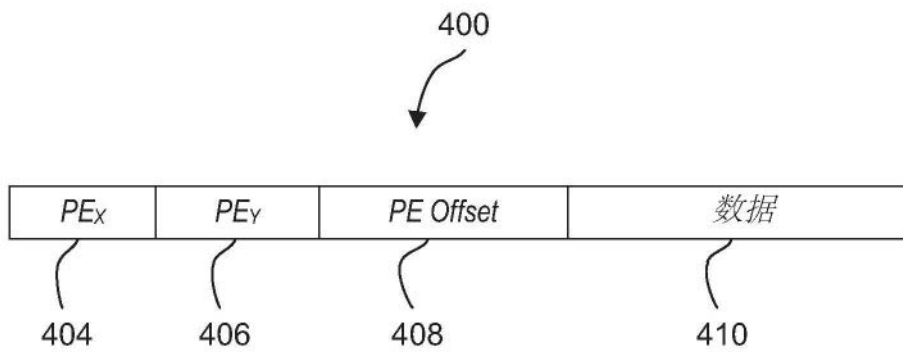


图4

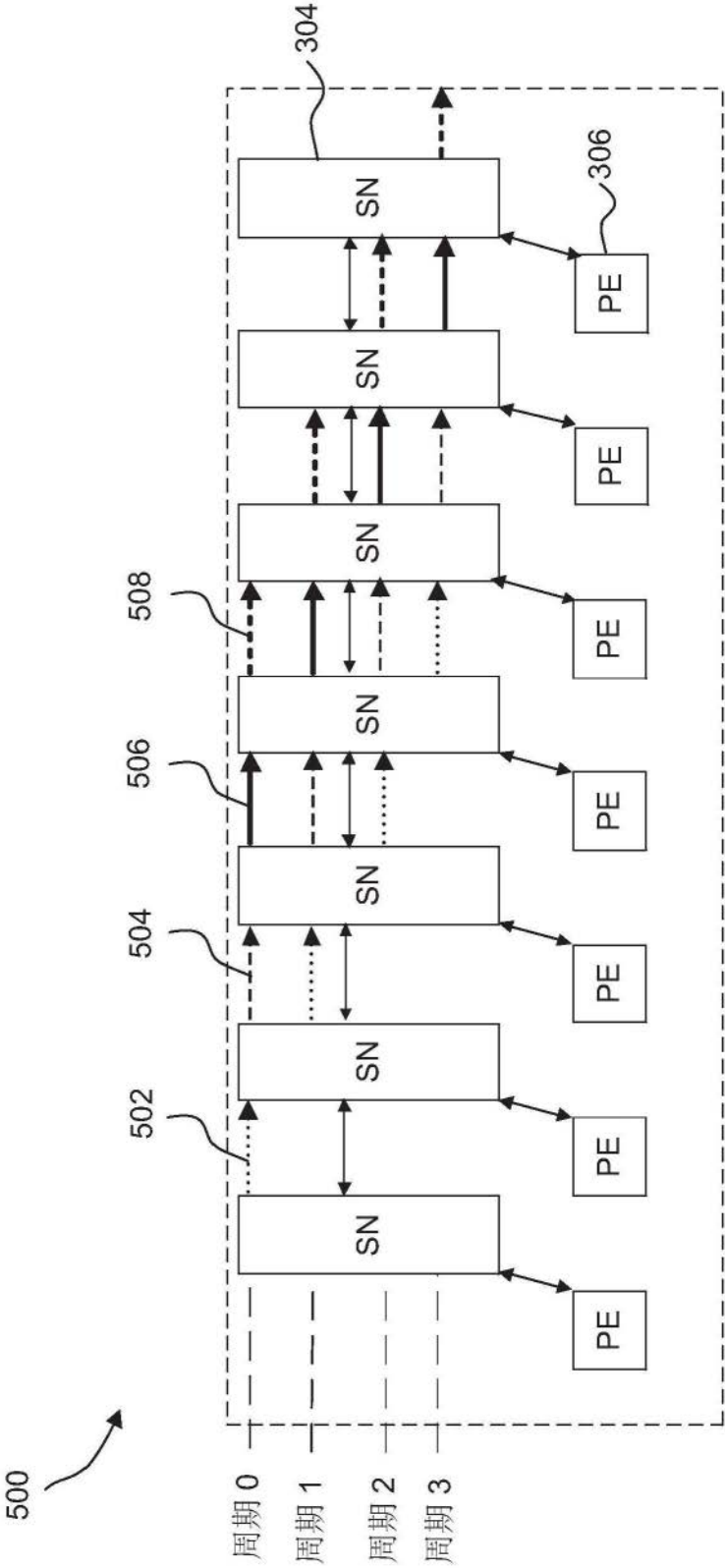


图5

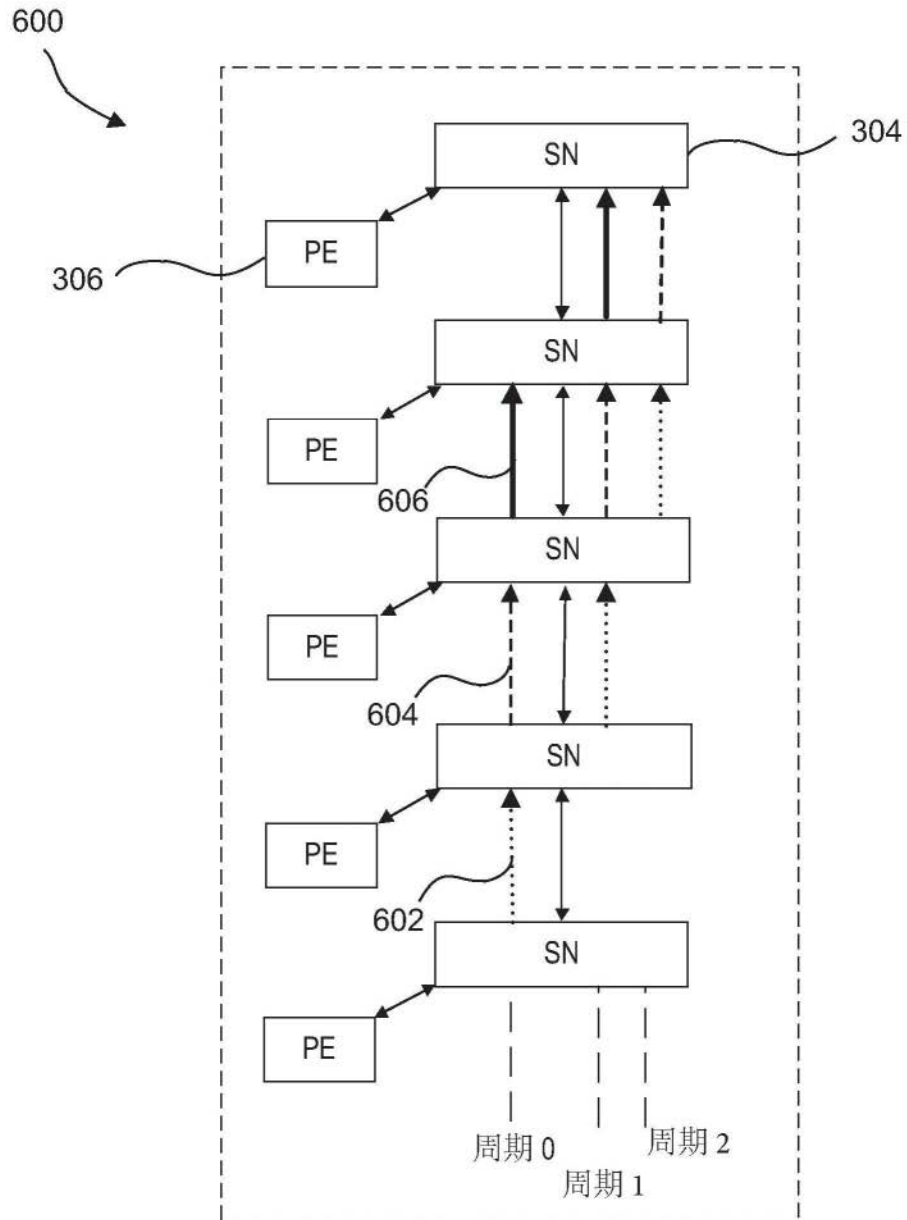


图6

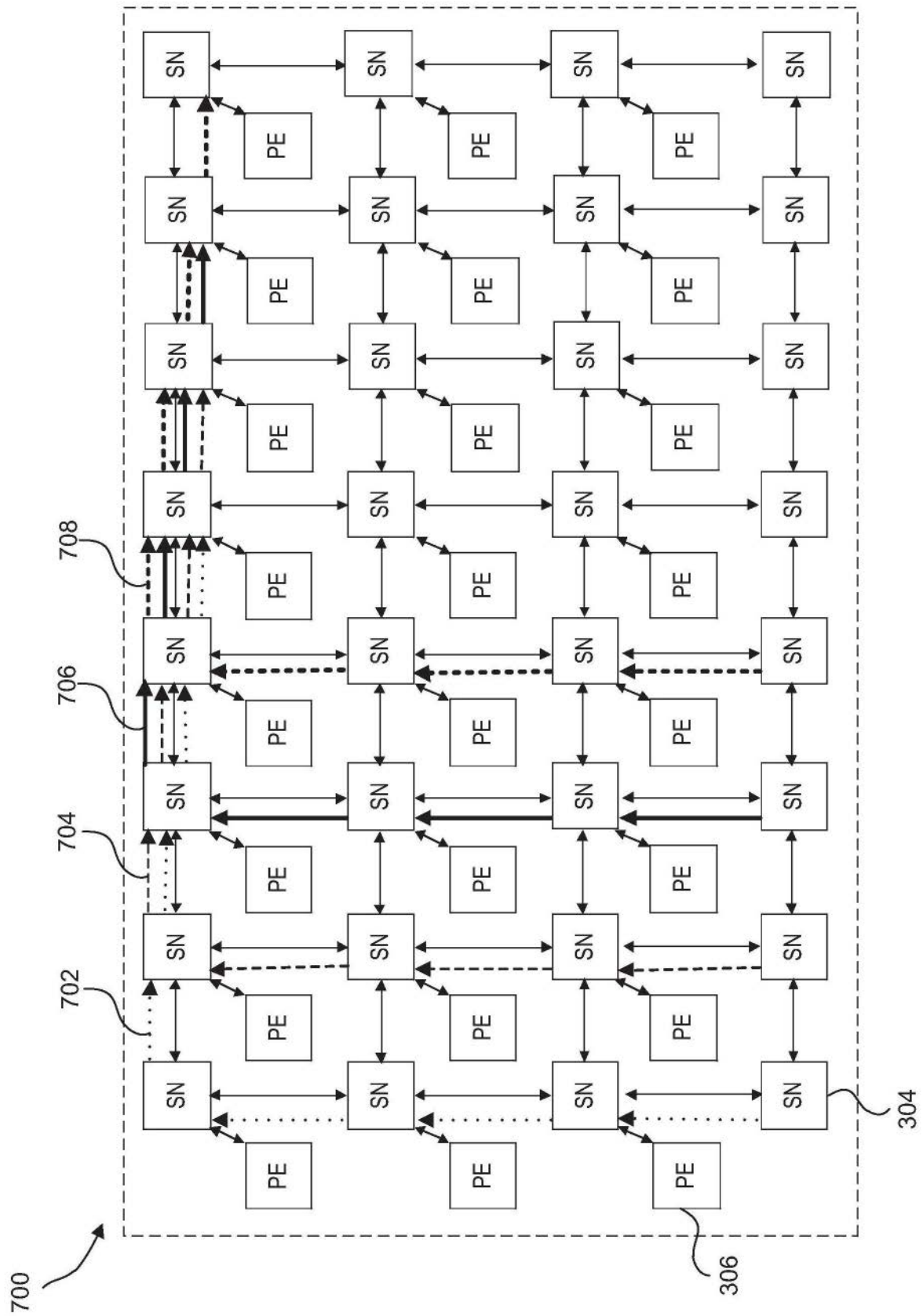


图7

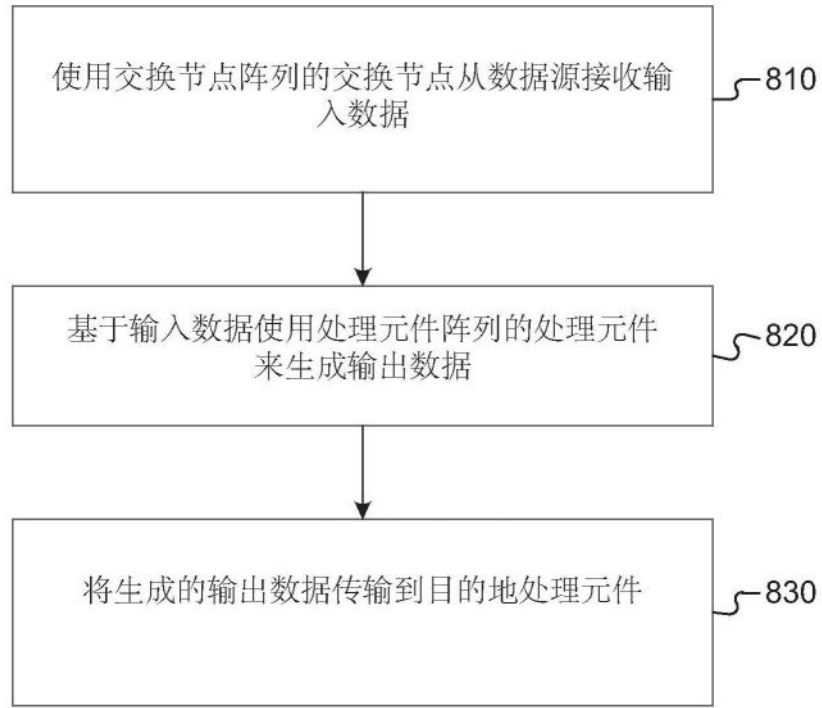
800

图8