(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 112579043 A
(43)申请公布日 2021.03.30

(21)申请号 202010578598.8

(22)申请日 2020.06.23

(30)优先权数据
16/586648 2019.09.27 US

(71)申请人 英特尔公司
地址 美国加利福尼亚州

(72)发明人 H.E.苏姆布尔 陈耕和 P.克纳格 R.库马尔 R.克里什纳墨菲

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

(51)Int.Cl.
G06F 7/544(2006.01)
G06F 17/16(2006.01)

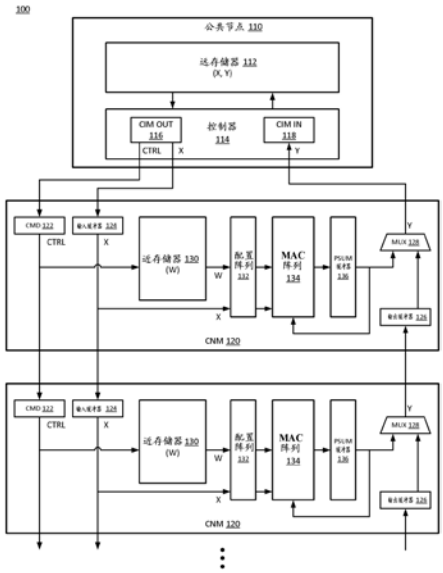
权利要求书2页 说明书19页 附图10页

(54)发明名称

存储器中计算/近存储器计算(CIM)电路架构

(57)摘要

一种存储器电路包括动态可配置的多个(X个)乘法累加(MAC)电路。MAC电路可基于输入向量的X个元素与权重向量的计算来计算输出,或者基于输入向量的单个元素与权重向量的计算来计算输出,其中每个元素具有一位或多位长度。第一存储器可保存具有宽度为X个元素的输入向量,而第二存储器可存储权重向量。MAC电路包括具有第一存储器的芯片上的MAC阵列。



CN 112579043 A

1.一种用于乘法累加(MAC)运算的设备,包括:

第一存储器,所述第一存储器用于提供具有宽度为X个元素的输入向量;以及

具有所述第一存储器的芯片上的乘法累加(MAC)阵列,所述MAC阵列包括:

第二存储器,所述第二存储器用于存储权重向量;以及

X个MAC电路,所述X个MAC电路动态地可配置成:基于所述输入向量的所述X个元素与所述权重向量的计算来计算输出,或者基于所述输入向量的单个元素与所述权重向量的计算来计算所述输出。

2.如权利要求1所述的设备,其中所述MAC阵列在具有所述第一存储器的公共存储器管芯上,其中所述第一存储器是处理器的高速缓存存储器。

3.如权利要求1所述的设备,其中所述MAC阵列在具有所述第一存储器的公共存储器管芯上,其中所述第一存储器是处理器的高速暂存存储器。

4.如权利要求1所述的设备,其中所述MAC阵列在具有所述第一存储器的片上系统内,其中所述第一存储器是处理器的高速缓存存储器。

5.如权利要求1所述的设备,其中所述MAC阵列在具有所述第一存储器的片上系统内,其中所述第一存储器是处理器的高速暂存存储器。

6.如权利要求1到5中任一项所述的设备,所述MAC阵列进一步包括:

复用器(mux),所述mux用于在所述第一存储器和所述MAC阵列之间提供替代路径;以及

mux控制器,所述mux控制器用于控制所述mux在所述替代路径之间进行选择。

7.如权利要求6所述的设备,其中所述mux控制器针对一维(1D)MxV计算要控制所述mux以用于一个输入向量元素到所有X个MAC电路。

8.如权利要求6所述的设备,其中所述mux控制器针对二维(2D)MxM计算要控制所述mux以用于X个不同的输入向量元素分别到所述X个MAC电路。

9.如权利要求1到8中任一项所述的设备,其中所述第一存储器包括静态随机存取存储器(SRAM)。

10.如权利要求1到8中任一项所述的设备,其中所述第二存储器包括寄存器堆。

11.如权利要求1到10中任一项所述的设备,其中所述第二存储器包括静态随机存取存储器(SRAM)。

12.一种用于乘法累加(MAC)运算的系统,包括:

处理单元的高速暂存存储器,所述处理单元的高速暂存存储器用于提供具有宽度为X个元素的输入向量;以及

硬件加速器,所述硬件加速器耦合到所述处理单元的所述高速暂存存储器,所述硬件加速器包括具有乘法累加(MAC)阵列的存储器中计算/近存储器计算(CIM)电路系统,所述MAC阵列包括:

本地存储器,所述本地存储器用于存储权重向量;以及

X个MAC电路,所述X个MAC电路动态可配置成:基于所述输入向量的所述X个元素与所述权重向量的计算来计算输出,或者基于所述输入向量的单个元素与所述权重向量的计算来计算所述输出。

13.如权利要求12所述的系统,所述MAC阵列进一步包括:

复用器(mux),所述mux用于在所述高速暂存存储器和所述MAC阵列之间提供替代路径;

以及

mux控制器,所述mux控制器用于控制所述mux在所述替代路径之间进行选择。

14.如权利要求13所述的系统,其中所述mux控制器针对一维(1D)MxV计算要控制所述mux以用于一个输入向量元素到所述MAC阵列的所有X个MAC电路。

15.如权利要求13所述的系统,其中所述mux控制器针对二维(2D)MxM计算要控制所述mux以用于X个不同的输入向量元素分别到所述MAC阵列的所述X个MAC电路。

16.如权利要求12所述的系统,其中:

所述处理单元包括多核主机处理器装置;

所述系统进一步包括通信地耦合到主机处理器的显示器;

所述系统进一步包括通信地耦合到主机处理器的网络接口;或者

所述系统进一步包括为所述系统供电的电池。

存储器中计算/近存储器计算(CIM)电路架构

技术领域

[0001] 描述一般涉及存储器中计算/近存储器计算(compute-in/near-memory)(CIM),并且更特别的描述涉及用公共CIM硬件在矩阵-矩阵计算和矩阵-向量计算之间进行动态切换。

背景技术

[0002] 已经在机器学习上特别是使用深度学习技术构建了计算机人工智能(AI)。通过深度学习,被组织为神经网络的计算系统计算输入数据与现有计算的数据匹配的统计可能性。神经网络涉及多个互连的处理节点,这些节点使数据分析能够将输入与“受训练的”数据进行比较。受训练的数据涉及对已知数据的性质进行计算分析,以开发用于比较输入数据的模型。

[0003] 某些深度学习应用,诸如应用单批量或小批量MLP(多层感知器)、RNN(递归神经网络)和LSTM(长短期存储器——具有反馈的递归神经网络)的那些应用,通常执行矩阵-向量(MxV)乘法,其中神经元激活的向量(或窄矩阵)与神经网络权重矩阵相乘。

[0004] 其它深度学习应用,诸如应用CNN(卷积神经网络,诸如用于图像处理、视频处理或二维(2D)图像的计算或分类)或大批量MLP的那些应用,一般被映射到矩阵-矩阵(MxM)乘法,其中矩阵与神经网络权重矩阵相乘。

[0005] 通过使用乘法累加(multiply-accumulate)(MAC)算术单元的阵列来执行对于MxV和MxM两者的点积运算。MAC单元的二维(2D)脉动阵列(systolic array)能为MxM运算提供具有良好性能和数据重复使用的硬件解决方案。一维(1D)计算阵列方法比2D脉动阵列更适合最大化向量元素的数据重复使用。从而,1D计算阵列为MxV运算提供了良好的性能。因此,MxV乘法通常很好地映射到1D计算阵列,而MxM乘法通常很好地映射到2D计算阵列。

[0006] 然而,在2D脉动阵列上执行MxV使计算与存储器带宽(BW)的比率急剧下降。类似地,在1D阵列上执行MxM会串行化(serialize)点积的高度可并行化的任务,引起性能和能量问题。结果,传统的硬件方法必须适合要执行的主要运算,这意味着针对MxM运算或者针对MxV运算进行优化。一些系统实现执行两种类型的操作,导致性能损失,因为硬件解决方案不能为非优化的工作负荷(workload)提供最大性能。

附图说明

[0007] 以下描述包括对图的讨论,这些图具有通过实现示例的方式给出的图示。应该通过示例而非限制的方式来理解附图。如本文中所使用的,对一个或多个示例的引用要被理解为描述包括在本发明的至少一个实现中的特定特征、结构或特性。本文中出现的诸如“在一个示例中”或“在备选示例中”的短语提供了本发明的实现的示例,并且不一定全都指的是相同实现。然而,它们也不一定相互排斥。

[0008] 图1是用动态可配置近存储器计算(CNM)电路执行计算的存储器中计算/近存储器计算系统的示例的框图。

[0009] 图2是动态可配置CNM核的示例的框图。

[0010] 图3A是配置用于矩阵-矩阵计算的动态可配置CNM核的计算操作的示例的框图。

[0011] 图3B是配置用于矩阵-向量计算的动态可配置CNM核的计算操作的示例的框图。

[0012] 图4A是用配置用于矩阵-矩阵计算的动态可配置CNM核的计算操作的周期(cycle)的图解示例。

[0013] 图4B是用配置用于矩阵-向量计算的动态可配置CNM核的计算操作的周期的图解示例。

[0014] 图5是具有处理器和动态可配置的存储器中计算/近存储器计算的电路系统的系统的示例的框图。

[0015] 图6是用能动态计算矩阵-矩阵或矩阵-向量计算的架构的存储器中计算/近存储器计算的过程的示例的流程图。

[0016] 图7是计算系统的示例的框图,其中能实现用动态可配置MAC阵列的存储器中计算/近存储器计算(CIM)操作。

[0017] 图8是多节点网络的示例的框图,其中能实现用动态可配置MAC阵列的存储器中计算/近存储器计算(CIM)操作。

[0018] 下面是对某些细节和实现的描述,所述描述包括对图的非限制性描述,它们可描绘一些或所有示例,以及其它潜在的实现。

具体实施方式

[0019] 如本文中所描述的,一种存储器电路包括动态可配置的多个乘法累加(MAC)电路。MAC电路或MAC算术单元可被配置成基于X个位的输入向量与权重向量的计算来计算输出,或者基于单个位的输入向量与权重向量的计算来计算输出。第一存储器提供输入向量,并且第二存储器提供权重向量。输入向量和权重向量两者可具有相同数量的位(例如,X个位),并且取决于配置,这些位以不同的组合相对彼此递送和计算。MAC电路按阵列被配置,并且该配置能确定阵列的操作。在一个示例中,阵列被配置成计算MxV(矩阵-向量)计算。在一个示例中,相同阵列被配置成计算MxM(矩阵-矩阵)计算。

[0020] MAC阵列的动态可配置性使相同的电路系统能够针对MxM运算或者MxV运算进行优化。将理解,与优化或进行优化相关的表达指的是相对于电路系统的其它配置的性能改进。此类表达不一定指的是绝对最优,也不暗示不能进行附加改进。例如,对MxM运算的优化涉及与改进MxM运算的性能的配置有关的改进阵列性能的MAC阵列配置。备选地,可说特定的配置集中于一种类型的操作或另一种类型的操作,或者配置针对特定类型的操作。用此类可配置性,相同的电路系统能为任一类型的操作提供高性能。将存在用于该配置的成本,与为特定操作类型单独优化的电路系统相比,该配置可能会降低性能,但是相对于为一种类型的操作配置的电路系统执行另一种类型的操作而言,该配置将提供改进的性能。在一个示例中,可重配置MAC阵列硬件在专用硬件(诸如,专用处理器或加速器)上实现。

[0021] 从而,相同的电路系统能为2D运算提供2D阵列配置,以计算用于推理或训练任务的大量神经网络(NN)参数的深度学习矩阵运算工作负荷。类似地,相同的电路系统能提供有利于1D计算操作的1D阵列配置。计算能包括用于单批量或小批量MLP、RNN和LSTM的1D计算,其中神经元激活的向量(或窄矩阵)与NN权重相乘。该计算可包括用于CNN或大批量MLP

的2D计算。在一个示例中，用于2D运算或1D运算的阵列配置能以脉动阵列的形式。

[0022] 脉动阵列在本领域中被理解为相似元素的模块化阵列，其具有在返回到主处理单元或存储器之前通过阵列元素传播数据和结果的架构，并且将不进行详细描述。许多描述提及脉动阵列，这将被理解为示例。在某些实现中，脉动阵列架构能提供优于非脉动阵列架构的改进性能，并且从而将用于描述下面的示例。可使用除了脉动阵列之外的架构。此类架构能使用不同类型的元素或元素的不同排列。

[0023] 神经网络计算的传统方法包括通用多核硬件或专用加速器。通用多核方法通常将SIMD(单指令多数据)类型向量化、分布式的解决方案用于诸如在GPU(图形处理单元)硬件或协处理器硬件上处置矩阵运算。通用方法为矩阵运算提供了足够的计算/带宽效率；然而，与专用处理器相比，它们提供较低性能。

[0024] 取决于所选定的任务和工作负荷，专用加速器(诸如，独立芯片、加速器、FPGA(现场可编程门阵列)平台或其它专用硬件)通常使用阵列式乘法器，诸如2D脉动阵列或1D计算阵列。传统的加速器解决方案为所选定的任务实现了非常高的能效；然而，当切换到另一个任务，诸如从MxM运算切换到MxV运算时，它们受到计算效率或能效限制以及带宽限制。有限的效率意味着，在许多实现中，将使用两种类型的专用硬件，这增加了成本并且降低了总体效率，因为某个硬件在某些操作中不可避免地保持不使用。

[0025] 用可配置MAC阵列，系统能将模块化近存储器计算(CNM)电路块实现为提供高能效和良好的计算/带宽比的用于MxM运算和MxV运算两者的统一解决方案。CNM电路块是存储器中计算/近存储器计算(CIM)电路的示例。该电路可以是包括执行计算的存储器的存储器中计算电路。近存储器计算电路更具体地指的是CIM电路，其中存储器是多级存储器架构的近存储器，其包括具有比远存储器级的易失性或非易失性存储器更快的访问的至少一级易失性存储器。CIM电路是能够实现在存储器电路中进行计算操作而不必将数据发送给处理器的电路系统。CIM加速器直接在存储器阵列内执行基本操作，诸如向量的点积和绝对差，这减少了将数据转移到计算引擎或处理器以用于计算的需要。存储器和处理单元之间数据移动的减少可用于加速存储器带宽受限的算法。数据移动的减少也能减少数据转移能量开销。基于模拟操作的CIM加速器根据每列访问多位数据读出允许(allow for)较低计算成本和较高有效存储器带宽。

[0026] 将轻量级再配置性电路集成到具有小的近存储器和MAC单元阵列的CNM模块中，能够实现两种不同的矩阵访问模式。结果，CNM核阵列可被动态配置成作为MxM运算的2D阵列执行，而作为MxV运算的1D计算阵列执行。将计算定位在存储器分区附近最大化了可用的本地存储器带宽并且最小化了用于计算的位传输的能量成本。模块化CNM MAC单元的再配置性允许基于选择的硬件配置或DNN(深度神经网络)特性或两者来平衡输入、权重和输出重复使用，这能够实现各种用例的高能效增益。

[0027] 所描述的CNM阵列的模块化本性能够实现用于不同应用的可配置操作以及可配置大小。可配置阵列可应用在从服务器到边缘装置的任何数量的计算装置中。该阵列对不同的应用是可缩放的，并且对于不同的系统使用是灵活的。

[0028] 图1是用动态可配置的近存储器计算(CNM)电路执行计算的存储器中计算系统的示例的框图。系统100提供了能在可配置硬件上实现MAC运算的系统组件的示例。系统100包括公共节点110和由图示的两个CNM单元表示的CNM阵列。

[0029] 在一个示例中，公共节点110包括远存储器112，远存储器是与近存储器相比具有更长访问时间的存储器资源。近存储器与将执行计算的硬件位于相同位置，而远存储器与近存储器相比具有对计算硬件的更长访问时间。通常，远存储器112比近存储器更大。

[0030] 在一个示例中，远存储器112是SRAM(同步随机存取存储器)装置。在一个示例中，每个近存储器130是SRAM装置。在一个示例中，近存储器130是存储器装置的寄存器堆。

[0031] 在一个示例中，公共节点110包括控制电路和各种混杂的算术逻辑、或者矩阵运算逻辑、或者算术和矩阵逻辑的组合。该逻辑可被实现为处理器或可编程逻辑阵列的部分或独立硬件。控制器114表示逻辑和控制电路。控制器114可包括到CNM阵列的存储器中计算/近存储器计算(CIM)硬件的接口。CIM OUT 116表示提供来自远存储器112的控制(CTRL)和数据(X)的接口。来自远存储器112的数据X表示要由CNM阵列执行的矩阵计算的操作数之一。CIM IN 118表示接收来自CNM阵列的数据(Y)的接口。数据Y表示计算结果。数据可被存储在远存储器112中。

[0032] 公共节点110包括作为用于存储X和Y的远存储器112的相对较大的存储器装置。近存储器130小于远存储器112。控制器114可以是或者可包括用于数据通信和对存储器进行访问的一个或多个控制块，以及实现深度神经网络(DNN)机器学习所需的各种算术运算。取决于目标工作负荷，公共节点算术运算可包括但不限于逐元素算术运算、归约(reduction)、池化(pooling)、非线性运算、矩阵转置、数据重塑、压缩或这些的组合或者其它运算。只要下游和上游控制是用CNM核实现的，可配置CNM阵列的使用就与公共节点110的任何设计选择和在公共节点110中实现的任何算术运算兼容。

[0033] CNM阵列包括标识为CNM 120的多个CNM单元或CNM核。系统100为CNM块提供顶层布局(top-level layout)。对于任何问题大小，矩阵和向量可根据CNM片大小(tile-size)划分成片，以实现具有最高可能利用率的MxV/MxM。CNM 120的片大小指的是被包括的MAC单元的数量。从而，图2的示例具有4个MAC单元，给出4的片大小和M=4的值。V小于M，诸如V=1。

[0034] CNM指的是近存储器计算资源，其中计算单元由存储器直接集成，或非常靠近存储器。位于存储器附近的计算单元提供细粒度的分布式计算和存储器阵列。虽然对存储器的典型访问将发送命令并在执行所有计算的处理核处等待数据结果，但是某些计算可直接在具有CNM架构的存储器处进行，从而减少了需要从存储器转移到处理器的数据量。相反，存储器本身具有计算资源，这些计算资源能返回计算的结果而不是要在计算中使用的数据。

[0035] 在系统100的配置中，系统向在CNM 120的CNM核处的近存储器130分配权重。在一个示例中，公共节点110的远存储器112存储输入矩阵(X)。类似地，输出Y矩阵被发送回到公共节点110以被处理或存储在远存储器112中。在一个示例中，系统100内的所有存储器访问被分片，其中将相同的地址用于每个CNM 120中的所有近存储器130。在一个示例中，公共节点110生成到近存储器130的地址或地址信息。

[0036] 在一个示例中，系统100将少量地址和读/写控制位与(一个或多个)输入X向量一起发送到下游数据转移(downstream data transfer)。在一个示例中，CNM 120以1D脉动阵列的方式向下游传递相同的数据。作为向下游传递相同数据的结果，CNM核将仅需要轻量级控制电路来实现控制操作。

[0037] CNM 120包括用于下游通信的输入缓冲器(In Buff)124和用于上游通信的输出缓冲器(Out Buff)126。下游通信指的是从公共节点110到CNM核的通信，或者从更靠近公共节

点110的CNM核到更远离公共节点110的CNM核的通信。类似地，上游通信指的是从CNM核到公共节点110的通信，或者从更远离公共节点110的CNM核到更靠近公共节点110的CNM核的通信。在一个示例中，在CNM 120之间向下游和向上游的1D脉动阵列通信以触发器到触发器(flop-to-flop)的方式实现。因此，输入缓冲器124可用于双重目的：存储(一个或多个)X输入向量和将(一个或多个)X输入向量向下游传送到下一个CNM核。在一个示例中，CNM 120在输入缓冲器124旁边的CMD(命令)122中包括额外的触发器，以保持并向下游传送地址和控制位。

[0038] 来自CIM OUT 116的CTRL表示来自公共节点110的控制位或命令位。CNM 120包括CMD或命令122，其表示用于控制信号的缓冲器。命令的缓冲和转发可类似于上面关于输入缓冲器124和输入X所描述的。在一个示例中，(一个或多个)输入X向量可从输入缓冲器124提供到指示为配置阵列132的控制硬件。在一个示例中，该命令使近存储器130向配置阵列132生成权重矩阵数据W。

[0039] 配置阵列132为由公共节点110所请求的所选运算配置MAC阵列134。配置阵列132可配置MAC阵列134的累加运算。累加运算使得MAC阵列在发送最终累加部分积以写入到PSUM缓冲器(部分和缓冲器)136之前，对一定数量的运算执行累加。PSUM缓冲器136能够实现向上游发送部分和。

[0040] 向上游发送部分和将根据操作类型(矩阵-向量或矩阵-矩阵)的访问模式。CNM 120经由输出Y端口向上游传送部分和。在一个示例中，输出Y端口由mux(复用器)128驱动，mux 128选择来自PSUM缓冲器136的本地部分和或者在OUT Buff 126接收的下游CNM的部分和。在一个示例中，CNM 120响应于累加信号(控制信号的部分——在系统100中未具体示出，但在系统200中示出)在1个时钟周期之后立即发送部分和。在一个示例中，CNM 120在时钟周期的数量等于指示CNM单元在阵列中位置的值之后发送部分和。指示特定CNM单元位置的值可由这样的数字来指示，该数字还指示在每个CNM内侧的MAC阵列中MAC单元的数量(在一个实现中，其可被称为“片大小”)。在阵列大小或片大小数量的时钟周期之后，CNM单元可输出其部分和。

[0041] 在一个示例中，在完成最终累加之后，CNM 120将发送出其PSUM缓冲器136的内容，并将PSUM缓冲器重置为0。发送出内容之后的重置使CNM 120能够在下一个时钟周期开始时立即准备好新的一组累加运算。这样，只要存在来自输入流水线的输入，就能在每个时钟周期利用MAC阵列134。在IN Buff 124中缓冲输入信号，并且在OUT Buff 126中缓冲输出信号，这使得CNM 120的阵列能够以流水线方式工作，在1D脉动阵列中的相邻CNM核之间具有一个时钟周期的操作延迟。在一个示例中，如果在每个CNM 120中执行足够量的就地(in-place)累加，则输出数据传输是无冲突的，并且不需要额外的通信控制。

[0042] 近存储器130包括存储器元件(cell)或位元件(bitcell)的阵列。在一个示例中，该阵列基于动态随机存取存储器(DRAM)技术。在一个示例中，该阵列基于一基于电阻式随机存取存储器(RAM)元件(诸如，电阻式RAM(ReRAM)、磁阻式RAM(MRAM)、铁电RAM(FeRAM)、电介质RAM或其它存储器)、基于三维交叉点(3DXP)相变的RAM或其它字节可寻址随机存取存储器技术。在一个示例中，远存储器112基于SRAM。在一个示例中，远存储器112基于上面标识的其它技术之一。

[0043] 在一个示例中，公共节点110是包括处理器附近的电路系统和存储器的硬件平台。

这样，在系统100中被标记为远存储器112的可以是处理器的高速缓存或高速暂存(scratch-pad)存储器。相对于由本地存储器或位于MAC阵列134附近的近存储器130执行的MAC计算，远存储器112在系统100中是“远”的。作为高速缓存或高速暂存存储器，远存储器112相对于计算装置的主机处理器或主机处理单元实际上可能是“近存储器”，系统100被合并到所述计算装置中。

[0044] 高速缓存存储器可指的是存储器层级中的一级存储器。对高速缓存存储器的访问可能受协议约束，该协议定义如何能访问存储器以及访问何时及何地能发生。高速暂存存储器可指的是类似于高速缓存的存储器，但是可由处理器在任何时间随机访问，而没有可应用于高速缓存的特定访问约束(例如，指令集架构(ISA)规则)。如所描述的CNM的应用可应用于高速缓存或高速暂存存储器。将理解，CNM对高速缓存的应用可包括某些高速缓存访问规则的应用。

[0045] 在一个示例中，系统100被集成在具有主机处理器的公共管芯上，其中CNM 120及其关联的存储器和MAC单元是硬件加速电路的部分。CNM 120的阵列可集成在硬件加速器管芯上，而不是集成在主机处理器管芯上，该硬件加速器管芯与主机处理器一起合并在公共衬底或主板上，或者被包括在具有主机处理器的片上系统中。从而，可将MAC阵列合并到具有主机处理器的管芯上，或者可作为硬件加速器在远离处理器的片外实现。

[0046] 图2是动态可配置CNM核的示例的框图。系统200提供根据图1的系统100的系统的一个示例。CNM核210表示根据图1的CNM 120的CNM单元或CNM核。在一个示例中，CNM 210能与较大的存储器(诸如，SRAM(静态随机存取存储器)、DRAM(动态随机存取存储器)或其它类型的存储器)集成。

[0047] CNM核210表示模块化CNM核电路的示例的细节。CNM核210包括用于接收和分配输入向量的输入缓冲器212和输入mux或输入选择器214。CNM核210包括用于根据接收到的控制信号来配置CNM核的操作的控件(control)(CTRL)216。CNM核210包括用于本地地存储权重数据(W)的近存储器(近mem)220，以及用于向MAC阵列提供所选位的mux 222。CNM核210包括MAC阵列230和部分和缓冲器(PSUM)232。当完成计算时，部分和能被反馈到MAC阵列230中或提供给输出。CNM核210包括用于接收和缓冲来自下游的输出数据的输出缓冲器240，以及用于转发本地计算或转发从下游接收到的数据的选择器或输出mux 242。控件(CTRL)244表示配置输出mux 242的操作的控制功能性。

[0048] 在一个示例中，MAC阵列230包括多个MAC单元。在一个示例中，当系统200被合并在以推理工作负荷为目标的深度学习加速器中时，用整数算术来实现MAC单元和累加。在一个示例中，当系统200被合并在以训练工作负荷为目标的深度学习加速器中时，CNM计算单元将用浮点算术构建。取决于目标深度学习工作负荷，可在MAC阵列230中实现基于位精度(例如，INT32、全精度浮点或其它固定位精度)、可重配置位精度(例如，INT8/4/2/1或其它可重配置精度)或混合精度(例如，bfloat16/INT8或其它)算术的MAC单元的单个选择。

[0049] 如图所示，输入缓冲器212、近存储器220、MAC阵列230、累加(“部分和”或“PSUM”)缓冲器232以及上游输出缓冲器240与M=4的设计选择相匹配，如4个元素的向量所示。这4个元素可被称为片大小，其是能与输入宽度和输出阵列大小一致的值。虽然该示例具备4个MAC元素或M=4，但是将理解，该系统是可缩放的，并且可用更多或更少的元素构成。阵列中二进制数的使用可提供改进的效率(例如，具有M=2、4、8、16、32...)。二进制数的使用不是

必须的，并且所描述的应该适用于大于1的任何整数值。在一个示例中，将存在匹配每个CNM核210内MAC阵列中的MAC元素的数量的多个CNM核。从而，系统200图示了输入X[3:0]、存储的数据W[3:0]以及在部分和Psum[3:0]上累加的MAC阵列的4个元素向量的示例，其中输出Y[3:0]，其中M=4并且其它元素通常是[(M-1):0]的形式。当使用不同数量的MAC元素时，M的值将相应地不同。在一个示例中，一元素是一个位。在一个示例中，一元素是多位元素。作为一般示例，对于X个MAC，将有一对X个元素，其中每个元素可以是单个位或多个位。

[0050] 输入缓冲器212接收并存储输入向量“X”，如输入X[3:0]所图示的那样。数据被称为下游数据，因为它从命令源和输入向量源向下游流动。在一个示例中，1D阵列中的第一CNM核接收来自远存储器的输入X，并将数据传递到下一个核，下一个核又以1D脉动方式将该数据向下游传递到后续核。输入向量X可被存储在输入缓冲器212中，在一个示例中，该输入缓冲器可以是双稳态触发器(flip flop)(FF)的阵列。当输入缓冲器被实现为双稳态触发器的阵列时，CNM核210能以触发器到触发器方式将输入X向下游传递到下一个CNM核。在一个示例中，输入缓冲器212被实现为锁存器阵列或小尺寸存储器(诸如，寄存器堆或SRAM)。输入缓冲器212的实现将取决于硬件要求、矩阵大小和可用的硬件预算。在一个示例中，输入缓冲器212可被实现为并行输入缓冲器。在备选示例中，取决于缓冲器电路选择、带宽和可用硬件资源，输入缓冲器212可被实现为串行输入FIFO(先进先出)缓冲器。在图中，实线图示了传递到输入缓冲器212和mux/选择器214的输入。虚线图示了信号还被传递到输入直通(input passthrough)，以向下游发送。

[0051] 近存储器220存储要在存储器的读取端口中缓冲的本地权重“W”的向量。在一个示例中，近存储器220被实现为具有高本地带宽的小尺寸存储器。具有高本地带宽的存储器示例可包括寄存器堆(RF)或小型SRAM。取决于硬件和工作负荷规格，近存储器220可具有单访问能力或多端口访问能力。从而，近存储器220可以是各种类型的RF中的任何RF，诸如1RW、1R1W或其它多端口存储器。备选地，近存储器220可以是各种小型SRAM变体中的任何小型SRAM变体，诸如1RW、1R1W或其它多端口存储器。

[0052] 在一个示例中，近存储器220用作与输入缓冲器212的触发器同步的时序电路，以用流水线输送(pipeline)MAC运算。在一个示例中，mux阵列214连接到输入缓冲器212，并且mux阵列222连接到近存储器220。在一个示例中，mux阵列的输入是向量，其中元素的数量等于片大小。mux阵列的输出可以是相同长度向量。

[0053] CNM核210接收控制(CTRL)位，诸如有效信号、累加(ACC)信号、地址(ADR)信号以及一个或多个其它配置(CONFIG)位。有效信号可被视为硬件执行操作的“进行”信号。ACC信号可被视为“完成”信号，这意味着操作被完成，并且累加的结果可被发送。ADR信号指示在计算操作中使用的存储器位置的地址。CONFIG位可指的是用于配置该配置阵列的任何其它信号。配置阵列指的是控制到MAC阵列230的位流的组件。

[0054] 对于CNM 210，系统本身不需要“配置”电路系统。在一个示例中，通过设置了一个或多个配置或控制信号，从输入和近存储器到MAC阵列的位流将改变，这改变了由MAC阵列执行的操作。作为输入要根据其进行计算的进入MAC阵列230中的位流将改变由MAC阵列执行的逻辑运算，即使MAC阵列操作在每种情况下都是相同的。用作输入的位和用于计算的周期数量可确定MxV和MxM运算的不同。在一个示例中，MxM运算能以多种不同的方式执行。例如，CNM核210的配置可被设置成在每个操作周期内向MAC阵列提供不同的位。作为备选方

案,MxV运算的运算可通过多次重复MxV运算模式来扩展,以产生MxM输出。该方法的灵活性允许根据问题大小和硬件预算定制阵列的大小和操作,以将相同的硬件用于不同的操作。

[0055] 在一个示例中,配置信号(一个或多个配置位)控制mux阵列的访问模式。CNM核210在控件(CTRL)216接收控制位以分配控制信号。用于控件216的逻辑可以是简单的路由电路系统或小型控制器或具有对控制信号进行定时控制的其它电路系统。在一个示例中,控件216将使能信号(EN)传递到输入阵列212和近存储器220。使能信号能够实现近存储器和输入缓冲器的操作,并且可响应于从上游断言的有效信号而生成。使能信号还可能够实现部分和缓冲器232的操作。

[0056] 控件216还向近存储器220提供地址信息(ADR),以标识要从存储器读取的向量或位的地址。在一个示例中,控件216向控件244提供累加信号(ACC),以控制mux阵列242的操作。在一个示例中,控件244可以是控件216的部分。在一个示例中,控件216根据要执行的操作向mux阵列214和222提供配置,即,要将mux阵列配置成基于MxM运算或MxV运算来组合输出向量。

[0057] 在一个示例中,mux阵列214和222响应于(一个或多个)配置信号实现两种特定的访问模式:“所有到所有(all-to-all)”和“一到所有(one-to-all)”。这两种特定的访问模式被用在MxM运算和MxV运算两者中。在一个示例中,“所有到所有”访问配置使mux阵列将向量原样传递到其输出。向量原样的传递向不同的MAC单元传送不同的权重。例如,在片大小被选定为4的情况下,对于[X0, X1, X2, X3]的输入向量,mux阵列的输出向量等于[X0, X1, X2, X3]。在一个示例中,“一到所有”访问配置使mux阵列将单个选择的元素传递到所有输出元素。传递单个选择的元素可被称为广播,其中该元素由位宽log2(tile_size)的选择位“S”选定,并被传送到所有计算单元。例如,在片大小被选定为4的情况下,对于[X0, X1, X2, X3]的输入向量和选择在位置<1>的元素[X1]的“S=01”,mux阵列的输出向量是[X1, X1, X1, X1]。

[0058] 在一个示例中,控件216包括计数器218,以从0计数到“tile_size-1”。在一个示例中,CNM核210将计数器218用作控制mux阵列的选择位“S”的元素指针。在一个示例中,对于MxV访问模式,计数器218实现了用于连接到输入缓冲器212的mux阵列214的指针。在一个示例中,对于MxV访问模式,CNM核每个周期访问近存储器220。在一个示例中,对于MxM访问模式,计数器218实现了用于在连接到近存储器220的mux阵列222的读出端口处的W向量的元素指针。在一个示例中,对于MxM访问模式,CNM核在操作的所有周期内只访问近存储器220一次,并且随着新的输入X向量在每个周期继续到达,继续以连续的方式访问近存储器220,直到累加触发CNM核210向上游传递结果为止。

[0059] 在一个示例中,系统200用流水线输送CNM 210的输出,以能够实现特定CNM核的输出与来自下游CNM核的结果的在公共节点处的累加。从而,系统200能利用多个CNM核210或核阵列来提供操作结果。公共节点没有在系统200中具体示出,但是耦合到CNM核阵列。用每个CNM核中的适当硬件,在每个CNM核中都可能发生来自下游的累加。将理解,此类实现将要求每个核中的附加电路系统来执行累加。当公共节点接收来自每个CNM核的输出并执行所有结果的累加时,在CNM核中不需要此类硬件。根据所描述的,每个CNM核能被配置用于1D或2D运算。通常,CNM核210的阵列将全部被配置用于相同类型的操作。用CNM核210,单个硬件核能用于MxV问题或MxM问题。在这两种配置中,CNM核210能充分利用MAC阵列230。在每个计

算周期内利用每个CNM核210中的所有MAC单元，最大化了系统200的操作的OPs/W(每秒每瓦的操作)。

[0060] CNM核210能够实现在公共硬件加速器上2D和1D矩阵运算之间的重配置。MAC阵列230是具有近存储器220计算的可重配置MAC阵列。为了能效，近存储器220与MAC电路靠近(在相同位置)。在一个示例中，MAC阵列230包括标准的元件数字MAC单元(cell digital MAC unit)。在一个示例中，MAC阵列230包括模拟乘法器电路。

[0061] 在一个示例中，系统200可将CNM核210重配置为用于MxV运算的1D脉动阵列。阵列不一定是脉动的。在一个示例中，系统200可将CNM核210重配置为用于MxM运算的2D脉动阵列。阵列不一定是脉动的。如上所述，可通过配置连接到输入缓冲器212的mux阵列214和连接到近存储器220的读出器的mux阵列222来实现重配置。用于“一到所有”和“所有到所有”的配置能以互斥的方式实现。控件216能向mux阵列或其它选择器硬件提供配置。

[0062] 如上所述，MAC阵列230的MAC单元能在MxM配置和MxV配置两者中得到充分利用，并且累加缓冲器或PSUM缓冲器232累加点积。在一个示例中，PSUM缓冲器232用双稳态触发器实现。在一个示例中，PSUM缓冲器232用锁存器实现。累加位宽或PSUM缓冲器232的宽度根据目标问题大小来选定。虽然在系统200中图示了4个元素，但是将理解，PSUM缓冲器232可比输入的位的数量更宽。从而，图示的4个元素可能各保持多位。作为特定示例，对于系统200中图示的4位输入，PSUM缓冲器232的4个元素可能保持32位。

[0063] 在一个示例中，除了标准累加电路之外，当输出位宽小于输入元素时，CNM核210包括额外的量化电路。在此类情况下，一旦完成累加，部分和就能直接传送到上游CNM模块或公共节点。为了确保在没有冲突的情况下数据转移发生，在一个示例中，在输出端口240接收的输出Y向量被缓冲。输出端口240能作为下游单元的上游输入缓冲器进行操作，并且上游输出是用mux阵列242将当前核的部分和以及输入上游累加向量复用的。

[0064] 在一个示例中，可通过减少上游数据以每周期仅发送核当中一个元素来最小化布线和数据传送能量。此类配置降低了系统200设计的布线复杂性。在此类实现中，CNM核210将需要在将输出数据写回到存储器之前执行更多的就地累加，以实现对MAC阵列230的充分利用。

[0065] 在一个示例中，单个位有效信号能控制用于近存储器220的读使能位和用于输入缓冲器212的写使能。在其中CNM 210包括用于地址位的附加触发器或缓冲器的示例中，有效信号还能控制用于地址位的写使能。在一个示例中，当有效位为高时，有效位指示传入数据转移(输入X和地址位)。从CNM核210的控制位输入到CNM核210的输出控制位的虚线指示：包括有效位在内的控制位可类似地被缓冲到输入缓冲器，所述输入缓冲器能够实现系统200中的位的核到核传输。

[0066] 在一个示例中，累加信号也被接收为类似于有效位的单个位。在一个示例中，累加信号指示要执行点积的最终累加。一旦接收到ACC信号，CNM核210就根据配置的访问模式(MxV、MxM)来执行其最终累加，并将部分和传送到上游输出Y端口。对于MxM，在一个示例中，当接收到ACC时，在1个时钟周期之后立即发送出部分和。对于MxV，在一个示例中，在发生tile_size个数量的时钟周期(例如，由计数器218所计数)之后，发送出部分和。

[0067] 在一个示例中，ACC信号充当mux阵列242的上游mux的mux控制，以控制上游通信的冲突。在一个示例中，ACC信号生成有效信号以用于以类似于输入有效信号控制下游通信的

传输的方式输出Y数据转移。在一个示例中，上游有效信号控制用于输出缓冲器的写使能。由于输入和输出信号被缓冲，诸如CNM核210的核能以流水线输送的方式工作。用流水线输送的操作，系统200可在系统200的1D核阵列中的相邻CNM核之间具有一个时钟周期操作延迟。如果在每个CNM核中执行足够数量的就地累加，则输出数据转移是无冲突的，并且将不需要额外的通信控制。

[0068] 图3A是配置用于矩阵-矩阵计算的动态可配置CNM核的计算操作的示例的框图。系统302图示了用于根据图1的系统100的系统的MxM运算的二维(2D)脉动阵列。系统302的2D脉动配置是根据图2的系统200的模块化、可重配置硬件的配置。

[0069] 系统302包括大的远存储器310以存储矩阵操作数(X)和矩阵结果(Y)。控件330将CNM 320配置用于系统302中的MxM运算。MxM config表示一个或多个控制信号，它们控制CNM 320的硬件以响应于输入来控制MAC单元的操作。控件330表示向CNM 320提供配置的硬件。将理解，控件330能至少部分地在CNM 320内部实现。

[0070] 远存储器310以逐行方式和逐列方式向CNM 320的2D阵列广播元素X。X的广播由到每个MAC单元324的单独实线图示。此外，W从近存储器322传递到所有MAC单元324。基于控件330的配置使CNM以MxM方式接收输入并对其进行操作。X的元素可被称为输入激活数据。行/列元素以相同的逐行/逐列方向从一个CNM 320传递到下一个，以最大化数据重复使用并摊销(amortize)存储器访问的高能量成本。

[0071] 每个计算单元CNM 320用MAC单元324执行MAC运算，同时MAC累加部分和。MAC单元324表示并行MAC算术单元的阵列。CNM 320对从远存储器310接收的X和从近存储器322访问的W执行MAC运算。W表示本地存储的权重数据。在系统302中，CNM 320作为2D阵列来执行MAC运算，其中单个W元素在CNM核内被广播到所有MAC单元324，并与多个X元素相乘。在一个示例中，对于接收到的片大小数量的X向量，仅访问近存储器322一次，并且以相同的方式重复，直到执行最终累加为止，并且Y被发送到远存储器310。

[0072] 将理解，MAC单元324包括在系统302中未具体示出的累加电路。在多次累加之后，MAC单元324将最终点积结果发送回到远存储器310以被存储。无论是由系统设计者选择还是为了满足应用要求，系统302与各种不同的算术设计选择中的任何算术设计选择兼容。例如，可用整数算术或浮点算术来实现MAC单元324，并且MAC单元324可基于固定位精度、可重配置位精度或混合精度。MAC单元324被图示为没有明确位精度的MAC。类似地，系统302与不同的向量大小兼容，并且假定X向量和W向量的位宽与作为计算单元的设计选择而规定并且未被具体指示的算术标准匹配。

[0073] 图3B是配置用于矩阵-向量计算的动态可配置CNM核的计算操作的示例的框图。系统304图示了用于根据图1的系统100的系统的MxV运算的一维(1D)脉动阵列。系统304的1D脉动配置是根据图2的系统200的可重配置硬件的配置。系统304是系统302的硬件的替代配置。从而，远存储器310、CNM 320、近存储器322、MAC 324和控件330的硬件描述在此将不重复，而是被理解成与系统302的相同。

[0074] 控件330将CNM 320配置用于系统304中的MxV运算。MxV config表示一个或多个控制信号，它们控制CNM 320的硬件以响应于输入来控制MAC单元的操作。虽然系统302被配置为2D阵列，但是系统304被配置为用于MxV运算的1D阵列。1D阵列方法能最大化向量元素的数据重复使用。远存储器310向CNM 320的1D阵列广播元素X。MAC单元324基于来自远存储器

310的X和从近存储器322的本地存储体(local memory bank)访问的矩阵操作数来执行计算。MAC单元324是执行MAC运算的1D计算阵列,其中部分和被累加,并且最终点积被发送回到远存储器310以被存储。

[0075] 在一个示例中,对于系统304的MxV配置,控件330将CNM 320配置成作为1D计算阵列执行,其中从近存储器322读取的多个W元素被分配给并行MAC单元324,并与从远存储器310接收的单个X元素相乘。从远存储器310到CNM 320的单个X线表示单个X线的使用,与如系统302中的不同X线相反。系统304能将不同的W元素分配给MAC单元以用于操作,如由平行的W线所示的那样。如同系统302的2D配置一样,系统304的1D配置能充分利用MAC单元。

[0076] 每个计算单元CNM 320用MAC单元324执行MAC运算,同时MAC累加部分和。MAC单元324表示并行MAC算术单元的阵列。CNM 320对从远存储器310接收的X和从近存储器322访问的W执行MAC运算。W表示本地存储的权重数据。

[0077] MAC单元324表示执行计算和累加的MAC单元和累加电路。在多次累加之后,MAC单元324将最终点积结果发送回到远存储器310以被存储。无论是由系统设计者选择还是为了满足应用要求,系统304与各种不同的算术设计选择中的任何算术设计选择兼容。例如,可用整数算术或浮点算术来实现MAC单元324,并且MAC单元324可基于固定位精度、可重配置位精度或混合精度。MAC单元324被图示为没有明确位精度的MAC。类似地,系统304与不同的向量大小兼容,并且假定X向量和W向量的位宽与作为计算单元的设计选择而规定并且未被具体指示的算术标准匹配。

[0078] 系统302的配置或系统304的配置的应用将取决于要计算的特定工作负荷。将理解,取决于操作,不同的配置将具有不同的功率结果。不同的配置在不同的环境下将提供更好的性能。

[0079] 所执行的操作将每输入产生一列输出(对于MxM运算),或者每输入产生一行输出(对于MxV运算)。尽管输出不同,但将理解,硬件没有定向(orientation)。相反,硬件只是响应于输入而生成输出,并且输出可基于要执行的操作的类型来解释。

[0080] 将理解,与仅执行MxM运算或MxV运算的传统电路相比,系统302和304的硬件将遭受损失。例如,仅执行MxM运算的电路将比用于MxM运算的系统302更高效,但是如果将需要执行MxV运算,则它将不太高效,而系统302可能被重配置成用于MxV运算的系统304。备选地,仅执行MxV运算的电路将比用于MxV运算的系统304更高效,但是如果将需要执行MxM运算,则它将不太高效,而系统304可能被重配置成用于MxM运算的系统302。

[0081] 图4A是用配置用于矩阵-矩阵计算的动态可配置CNM核的计算操作的周期的图解示例。表400图示了根据系统100和系统200的可重配置CNM核的MxM访问模式。

[0082] 表400示出了示例片大小为4的MxM运算的访问模式配置。输入X的元素被存储在输入缓冲器,而从近存储器访问W。在PSUM缓冲器累加部分和。图示的示例执行形式为[4x4] * [4x4] = [4x4]的MxM运算,其中逐列接收到输入缓冲器的输入X是大小为[4x4]的矩阵,从近存储器访问的权重矩阵的大小为[4x4],并且逐列累加在PSUM缓冲器上的输出Y是大小为[4x4]的矩阵。

[0083] MxM运算被配置成MxV(如图4B的表410中所示)的相对访问模式。在一个示例中,连接到输入缓冲器的mux阵列被配置成所有到所有访问模式。当部署CNM核时,访问模式将CNM阵列配置为2D阵列。对于4的示例片大小,表400示出了在4个时钟周期中MxM运算期间输出Y

矩阵的单列的点积。在周期0中，输入X的列0乘以权重向量W的位0。在周期1中，输入X的列1乘以权重向量W的位1。在周期2中，输入X的列2乘以权重向量W的位2。在周期3中，输入X的列3乘以权重向量W的位3。对于此示例，存储在近存储器中的W矩阵的列0在初始时钟被访问一次，并重复使用了4个周期。结果，X矩阵的一列在每个周期作为向量发送到MAC阵列。

[0084] 这样，每个时钟周期实现单个W元素与X矩阵的一列的乘法和累加。在每个周期结束时，在PSUM缓冲器累加部分和。结果，乘积[4x4] * [4x1] = [4x1]在CNM核中在4个时钟周期实现，从而在每个周期充分利用了MAC阵列。对于这个示例，通过对连续的W列重复图4A中所示的4个时钟周期操作来计算输出矩阵Y的连续列，在16个周期中计算输出矩阵Y，因为随后的时钟周期将对输入和近存储器的不同元素进行计算，直到整个矩阵被计算为止。例如，周期4-7将用W的下一个元素来计算输出Y的第二列，继续直到完成计算为止。

[0085] 图4B是用配置用于矩阵-向量计算的动态可配置CNM核的计算操作的周期的图解示例。表410图示了根据系统100和系统200的可重配置CNM核的MxV访问模式。

[0086] 表410示出了示例片大小为4的MxV运算的访问模式配置。输入X的元素被存储在输入缓冲器，而从近存储器访问W。在PSUM缓冲器累加部分和。图示的示例执行形式为[1x4] * [4x4] = [1x4]的MxV运算，其中输入X是大小为[1x4]的向量，从近存储器访问的权重矩阵的大小为[4x4]，并且在PSUM缓冲器上累加的输出Y的大小为[1x4]。

[0087] MxV运算被配置成MxM(如图4A的表400中所示)的相对访问模式。在一个示例中，连接到输入缓冲器的mux阵列被配置成一到所有访问模式。当部署CNM核时，这个访问模式将CNM阵列配置为1D阵列。对于4的示例片大小，表410示出了在CNM核中在4个时钟周期实现的MxV运算。在一个示例中，输入X向量元素在4个连续周期中被访问，并被广播到MAC阵列。在一个示例中，W矩阵以行优先(row-major)格式被存储，并且来自近存储器的单次读出提供了W矩阵的一行，其是大小为[1x tile_size = 4]的W行向量。因此，在每个连续周期，W矩阵的一行被读出并发送到MAC阵列。

[0088] 在周期0中，输入X的一位乘以权重向量W的行0。在周期1中，输入X的一位乘以权重向量W的行1。在周期2中，输入X的一位乘以权重向量W的行2。在周期3中，输入X的一位乘以权重向量W的行3。结果，乘积[1x4] * [4x4] = [1x4]在CNM核中在4个时钟周期实现，从而在每个周期充分利用了MAC阵列。对于这个示例，输入X在每个时钟周期对一个元素进行访问，同时重复使用在初始时钟周期接收的相同输入X向量。

[0089] 将观察到，也可重复表410中所示的MxV运算以实现MxM运算。与根据表400的MxM运算相对，执行重复的MxV运算的能量益处和成本取决于特定的硬件选择，诸如CNM近存储器130和公共节点远存储器112大小、可用BW、1D阵列中CNM核的数量、深度神经网络参数和其它考虑因素。

[0090] 图5是具有处理器和动态可配置的存储器中计算/近存储器计算的电路系统的系统的示例的框图。系统500可包括具有根据上面描述的任何示例的可重配置CNM电路系统的CIM电路。CIM电路可以是用于机器学习或其它计算密集型运算的CIM加速器。加速器可被称为协处理器。

[0091] 系统500包括处理器510，该处理器可以是或包括通用CPU(中央处理单元)、GPU(图形处理单元)、DSP(数字信号处理器)阵列或其它处理器。处理器510表示数字处理器，并且可被称为主机处理器。在一个示例中，处理器510包括多个核512。处理器510执行访问数据

以用于计算密集型运算的应用或软件程序。

[0092] 在一个示例中，处理器510包括第一级高速缓存514，其表示为热数据提供物理上靠近的存储资源的处理器管芯或处理器SOC(片上系统)内的高速缓存资源。在一个示例中，处理器510包括未明确示出的多级片上/本地高速缓存。将理解，高速缓存资源可以是特定核512的本地资源，由多个核512共享，或者是共享高速缓存资源和本地化高速缓存资源的组合。

[0093] 系统500包括最后一级高速缓存520。在一个示例中，最后一级高速缓存520也在处理器510上。在一个示例中，最后一级高速缓存520在远离处理器510的片外。最后一级高速缓存520比第一级高速缓存514大，但是花更长的时间来访问。在第一级高速缓存514和最后一级高速缓存520之间可有任何数量的其它高速缓存层。

[0094] 主存储器550表示系统主存储器。主存储器550通常比高速缓存资源大几个数量级，但是相对于高速缓存资源，要花更长的时间来访问。主存储器550可包括当电力中断时具有不确定状态的易失性存储器资源。主存储器550可包括非易失性存储器资源，这些资源即使在到存储器的电力被中断时也维持状态。高速缓存资源通常是易失性的，但是可包括非易失性存储器资源。

[0095] 系统500包括核512的指令执行流水线的描绘。在一个示例中，每个核512可包括多个执行流水线(例如，多线程)。指令流水线要被理解为一般性解释，并且特定细节未被提供。在一个示例中，指令流水线包括指令提取562，其中核提取要执行的指令。指令解码564表示由核在准备处理器电路系统以用于执行时对指令的解码。在一个示例中，指令解码564将该指令标识为触发CIM电路以用于操作的使用的命令的部分。

[0096] 操作数提取566表示为指令提取或获得要在其上执行的操作数。在一个示例中，操作数是神经网络的权重向量，或者其它数学函数操作数。在一个示例中，操作数在或被放置在与执行流水线关联的寄存器空间中。执行568表示在(一个或多个)操作数上或用(一个或多个)操作数来执行指令。在一个示例中，执行可包括将操作发送到CIM电路系统以用于执行。在一个示例中，指令被发送到处理器ALU(算术逻辑单元)，其可触发CIM加速器的执行。写回570指的是将执行结果写入结果寄存器中以用于返回到存储器，或者传递到另一个寄存器以用于另一次执行。在使用CIM电路系统的情况下，执行568和写回570可包括将操作发送到CIM电路系统，CIM电路系统将执行该操作并返回功能输出，然后该功能输出可被写回。该执行可包括根据本文中描述的任何示例的CNM电路系统的配置。从而，传统上会要求将操作数加载到处理器510内的ALU或其它计算引擎中的指令可被发送到CIM电路系统，而不必从存储器读取，并且然后从CIM电路系统而不是从本地计算引擎接收计算结果。

[0097] CIM 530表示在最后一级高速缓存520中实现的CIM电路系统。CIM 530可以是最后一级高速缓存520的全部或部分。在一个示例中，最后一级高速缓存520包括被配置为CIM电路系统的存储器阵列，以及不包括CIM电路系统的存储器阵列。系统能选择性地将数据存储在CIM使能的存储器阵列中以用于CIM加速。

[0098] CIM 540表示与系统500的高速缓存架构分开的CIM电路系统。从而，CIM 540表示专用于CIM加速操作的执行的存储器资源。此类CIM存储器仍然能包括传统的访问硬件，以能够实现在不执行计算的情况下从存储器中读取。CIM 530和CIM 540包括CIM硬件，以能够实现根据本文中描述的任何示例的功能读取。

[0099] CIM 530和CIM 540包括来自处理器510的访问路径。当高速缓存架构的部分作为CIM 530时，互连硬件可与用于高速缓存装置的互连硬件相同。当高速缓存架构的部分不作为CIM 540时，CIM互连可与高速缓存装置或主存储器装置或耦合到处理器510的其它存储器类似或相同。CIM 530和CIM 540耦合到主存储器550(未具体示出)，以加载用于CIM加速操作的操作数。

[0100] 在一个示例中，计算密集型函数的调用触发将命令指向CIM加速器。例如，处理器510可包括其指令集架构(ISA)中对触发CIM加速器的操作的某些命令的支持。在一个示例中，CIM加速器的调用可在将配置信息加载到加速器中之前或与之相关联。此类配置信息可例如定义节间连接的权重、定义要执行的数学函数或其它配置。系统500可从系统主存储器或从非易失性大容量存储装置或其组合中加载配置信息。

[0101] 在一个示例中，CIM 530或CIM 540或两者的CIM电路系统可包括一级或多级神经网络。在一个示例中，在配置CIM电路系统之后，指令流水线将输入值应用于所配置的神经网络以用于处理。CIM电路系统生成写回到寄存器空间或系统存储器的结果，并且CIM电路系统可向调用CIM电路系统的执行线程指示处理的完成。在一个示例中，如果在CIM电路系统中物理实现的神经网络级或每级神经元的数量少于要处理的神经网络的级/神经元的数量，则可通过重复加载用于处理的CIM电路系统来迭代通过神经网络的处理，直到已经处理了神经网络的所有级为止。

[0102] 图6是用能动态计算矩阵-矩阵或矩阵-向量计算的架构的存储器中计算的过程的示例的流程图。过程600表示用根据本文中任何描述的可配置MAC阵列进行MAC计算的过程。在一个示例中，包括可配置MAC阵列的系统确定是执行MxM计算还是MxV计算(框602)。该确定可根据用于要执行的神经网络操作的矩阵工作负荷。在一个示例中，系统生成配置位(框604)，诸如其中主控制器发送CNM核上的控制逻辑将用于配置操作的控制位。

[0103] 在一个示例中，CNM单元响应于配置位来配置MAC阵列(框606)。CNM单元访问一个或多个远存储器位输入X(框608)。CNM单元还访问一个或多个近存储器位输入W(框610)。CNM单元用配置的MAC阵列来计算X*W的乘积(框612)。

[0104] 在一个示例中，MAC阵列继续执行乘法和累加运算，直到接收到完成信号为止。如果CNM单元要继续累加部分和信息(框614“是”分支)，则MAC单元执行累加(框616)，并返回到访问一个或多个远存储器位(框608)。将理解，在累加之后的重复中，访问远存储器或近存储器可能指的是从缓冲器访问数据，而不是再次访问特定的存储器装置。

[0105] 一旦完成所有累加运算(框614“否”分支)，CNM单元就可向上游发送结果数据位向量Y(框618)。系统可将结果位应用于最终操作，以生成期望的输出(框620)。

[0106] 图7是计算系统的示例的框图，其中可实现用动态可配置MAC阵列的存储器中计算(CIM)操作。系统700表示根据本文中任何示例的计算装置，并且可以是膝上型计算机、桌上型计算机、平板计算机、服务器、游戏或娱乐控制系统、扫描仪、复印机、打印机、路由或切换装置、嵌入式计算装置、智能电话、可穿戴装置、物联网装置或其它电子装置。

[0107] 在一个示例中，系统700包括CIM电路系统。CIM电路系统包括根据本文中描述的任何示例的可重配置的CNM电路系统。CIM电路系统的一个示例可以是CIM 792，其被表示为包括在处理器SOC 710中的CIM电路系统。CIM 792可以是或包括处理器SOC衬底或管芯上的存储器装置中的CIM电路系统。例如，CIM 792可以是处理器SOC 710上的高速缓存的部分。CIM

电路系统的另一个示例可以是CIM 794，其被表示为包括在存储器子系统720中的CIM电路系统。更具体地，CIM 794被图示为存储器730的部分，指的是系统700的存储器资源。CIM 794表示可作为存储器资源被包括的CIM电路系统，处理器SOC 710可将计算密集型运算卸载到所述存储器资源。计算密集型运算可包括MxM运算和MxV运算。可配置CNM电路系统使CIM 792或CIM 794能够针对两种类型的工作负荷是可配置的。根据要执行的操作来动态地配置电路系统。

[0108] 系统700包括处理器SOC 710(后续称为“处理器710”)，其为系统700提供处理、操作管理和指令执行。处理器710可包括任何类型的微处理器、中央处理单元(CPU)、图形处理单元(GPU)、处理核或者为系统700提供处理的其它处理硬件，或者处理器的组合。处理器710可包括单核或多核。处理器710控制系统700的总体操作，并且可以是或者包括一个或多个可编程通用或专用微处理器、数字信号处理器(DSP)、可编程控制器、专用集成电路(ASIC)、可编程逻辑装置(PLD)等，或者此类装置的组合。

[0109] 在一个示例中，系统700包括耦合到处理器710的接口712，该接口可表示用于需要较高带宽连接的系统组件(诸如，存储器子系统720或图形接口组件740)的较高速度接口或高吞吐量接口。接口712表示接口电路，其可以是独立组件或被集成到处理器管芯上。在存在的情况下，图形接口740通过接口连接到图形组件以用于向系统700的用户提供视觉显示。在一个示例中，图形接口740可驱动向用户提供输出的高清晰度(HD)显示器。高清晰度可涉及具有近似100 PPI(每英寸像素)或更大的像素密度的显示器，并且可包括诸如全HD(例如，1080p)、视网膜显示(retina display)、4K(超高清晰度或UHD)或其它的格式。在一个示例中，显示器可包括触摸屏显示器。在一个示例中，图形接口740基于存储在存储器730中的数据或者基于由处理器710执行的操作或者两者来生成显示。

[0110] 存储器子系统720表示系统700的主存储器，并且为要由处理器710执行的代码或要在执行例程中使用的数据值提供存储。存储器子系统720可包括一个或多个存储器装置730，诸如只读存储器(ROM)、闪速存储器、一种或多种随机存取存储器(RAM)(诸如，DRAM)或其它存储器装置或此类装置的组合。存储器730除了其它事项外还存储和托管(host)操作系统(OS)732，以为系统700中指令的执行提供软件平台。此外，应用734可在来自存储器730的OS 732的软件平台上执行。应用734表示具有它们自己的操作逻辑以执行一个或多个功能的执行的程序。过程736表示向OS 732或一个或多个应用734或其组合提供辅助功能的代理或例程。OS 732、应用734和过程736提供软件逻辑来为系统700提供功能。在一个示例中，存储器子系统720包括存储器控制器722，该存储器控制器是生成命令并向存储器730发布命令的存储器控制器。将理解，存储器控制器722可能是处理器710的物理部分或者接口712的物理部分。例如，存储器控制器722可以是集成的存储器控制器，集成到具有处理器710的电路上。

[0111] 虽然没有具体图示，但是将理解，系统700可在装置之间包括一个或多个总线或总线系统，诸如存储器总线、图形总线、接口总线或其它总线。总线或其它信号线可将组件通信地或电气地耦合在一起，或者既通信地又电气地耦合组件。总线可包括物理通信线、点对点连接、桥接器、适配器、控制器或其它电路系统或其组合。总线可包括例如以下项中的一个或多个：系统总线、外围组件互连(PCI)总线、超传输或工业标准架构(ISA)总线、小计算机系统接口(SCSI)总线、通用串行总线(USB)或电气与电子工程师协会(IEEE)标准1394总

线。

[0112] 在一个示例中，系统700包括接口714，该接口可耦合到接口712。与接口712相比，接口714可以是较低速度接口。在一个示例中，接口714表示接口电路，其可包括独立组件和集成的电路系统。在一个示例中，多个用户接口组件或外围组件或两者耦合到接口714。网络接口750给系统700提供了通过一个或多个网络与远程装置(例如，服务器或其它计算装置)通信的能力。网络接口750可包括以太网适配器、无线互连组件、蜂窝网络互连组件、USB(通用串行总线)或其它基于有线或无线标准的接口或专有接口。网络接口750可与远程装置交换数据，这可包括发送存储在存储器中的数据，或者接收要存储在存储器中的数据。

[0113] 在一个示例中，系统700包括一个或多个输入/输出(I/O)接口760。I/O接口760可包括一个或多个接口组件，用户通过所述接口组件与系统700进行交互(例如，音频、字母数字、触觉/触摸或其它接口连接(interfacing))。外围接口770可包括上面没有具体提到的任何硬件接口。外围一般指的是从属地连接到系统700的装置。从属连接是其中系统700提供软件平台或硬件平台或两者的连接，在软件平台或硬件平台或两者上执行操作，并且用户与其交互。

[0114] 在一个示例中，系统700包括以非易失性方式存储数据的存储子系统780。在一个示例中，在某些系统实现中，存储子系统780的至少某些组件可与存储器子系统720的组件重叠。存储子系统780包括(一个或多个)存储装置784，其可以是或包括用于以非易失性方式存储大量数据的任何传统介质，诸如一个或多个基于磁的、固态的或光的盘或其组合。存储装置784将代码或指令和数据786保持在持久状态(即，尽管到系统700的电力中断，该值仍被保留)。尽管存储器730通常是向处理器710提供指令的执行或操作存储器，但是存储装置784一般可被认为是“存储器”。尽管存储装置784是非易失性的，但是存储器730可包括易失性存储器(即，如果到系统700的电力被中断，则数据的状态或值是不确定的)。在一个示例中，存储子系统780包括与存储装置784通过接口连接的控制器782。在一个示例中，控制器782是接口714或处理器710的物理部分，或者可包括处理器710和接口714两者中的电路或逻辑。

[0115] 电源702向系统700的组件提供电力。更具体地，电源702通常通过接口连接到系统700中的一个或多个电力供应704，以向系统700的组件提供电力。在一个示例中，电力供应704包括AC到DC(交流到直流)适配器，以插入到壁装电源插座中。此类AC电力可以是可再生能(例如，太阳能)电源702。在一个示例中，电源702包括DC电源，诸如外部AC到DC转换器。在一个示例中，电源702或电力供应704包括经由接近充电场进行充电的无线充电硬件。在一个示例中，电源702可包括内部电池或燃料电池源。

[0116] 图8是多节点网络的示例的框图，其中可实现用动态可配置MAC阵列的存储器中计算(CIM)操作。系统800表示根据本文中描述的任何示例的节点网络。在一个示例中，系统800表示数据中心。在一个示例中，系统800表示服务器场(server farm)。在一个示例中，系统800表示数据云或处理云。

[0117] 节点830包括存储器840，该存储器在一个示例中包括CIM 842。CIM电路系统包括根据本文中描述的任何示例的可重配置的CNM电路系统。用CIM 842，存储器840变成节点830的计算资源。更具体地，CIM 842使节点830能够提供对存储器/数据密集型以及计算密集型的工作负荷的改进处理。CIM 842使存储器能够执行计算并将功能结果返回给调用处

理器832，而不是从存储器到处理器来回移动数据。CIM 842可被实现为存储器840内的独立资源或者实现为加速器。在一个示例中，CIM 842被实现为处理器832上的高速缓存。在一个示例中，CIM 842是作为对于处理器资源的刀片(blade)的加速器的独立刀片。取决于要执行的工作负荷类型，可配置CNM电路系统使CIM 842能够被配置用于MxM运算或者MxV运算。根据要执行的操作来动态配置电路系统。

[0118] 一个或多个客户端802通过网络804向系统800提出请求。网络804表示一个或多个局域网、或广域网、或组合。客户端802可以是人类或机器客户端，其生成由系统800执行操作的请求。系统800执行由客户端802请求的应用或数据计算任务。

[0119] 在一个示例中，系统800包括一个或多个机架(rack)，机架表示容纳和互连多个计算节点的结构资源和互连资源。在一个示例中，机架810包括多个节点830。在一个示例中，机架810托管多个刀片组件820。托管指的是提供电力、结构或机械支持以及互连。刀片820可指印刷电路板(PCB)上的计算资源，其中PCB容纳一个或多个节点830的硬件组件。在一个示例中，刀片820不包括除了由机架810提供的机箱(chassis)或外壳或其它“盒子”之外的机箱或外壳或其它“盒子”。在一个示例中，刀片820包括具有连接到机架810中的暴露的连接器的外壳。在一个示例中，系统800不包括机架810，并且每个刀片820包括机箱或外壳，该机箱或外壳能堆叠或以其它方式驻留在其它刀片附近，并且允许节点830互连。

[0120] 系统800包括组构(fabric)870，该组构表示节点830的一个或多个互连器。在一个示例中，组构870包括多个交换机872或路由器或其它硬件，以在节点830之间路由信号。此外，组构870能将系统800耦合到网络804，以供客户端802访问。除了路由设备之外，组构870可被认为包括将节点830耦合在一起的电缆或端口或其它硬件设备。在一个示例中，组构870具有一个或多个关联的协议来管理通过系统800的信号的路由。在一个示例中，一个或多个协议至少部分地取决于系统800中使用的硬件设备。

[0121] 如图所示，机架810包括N个刀片820。在一个示例中，除了机架810之外，系统800还包括机架850。如图所示，机架850包括M个刀片860。M不一定与N相同；从而，将理解，各种不同的硬件设备组件可被使用，并通过组构870一起耦合到系统800中。刀片860可与刀片820相同或类似。节点830可以是任何类型的节点，并且不一定全都是相同类型的节点。系统800不限于同质(homogenous)，也不限于不同质。

[0122] 为简单起见，仅详细图示了刀片820[0]中的节点。然而，系统800中的其它节点可相同或类似。至少一些节点830是计算节点，具有处理器(proc)832和存储器840。计算节点指的是具有执行操作系统并能接收和处理一个或多个任务的处理资源(例如，一个或多个处理器)的节点。在一个示例中，至少一些节点830是服务器节点，其中服务器作为由处理器832和存储器840表示的处理资源。存储服务器指的是比计算节点具有更多的存储资源的节点，并且存储服务器包括管理对存储服务器内的存储节点的访问的处理资源，而不具有用于执行任务的处理器。

[0123] 在一个示例中，节点830包括接口控制器834，该接口控制器表示控制由节点830对组构870的访问的逻辑。该逻辑可包括与物理互连硬件互连的硬件资源。该逻辑可包括管理互连的软件或固件逻辑。在一个示例中，接口控制器834是或包括主机组构接口，该主机组构接口可以是根据本文中描述的任何示例的组构接口。

[0124] 处理器832可包括一个或多个单独处理器。每个单独处理器可包括单个处理单元、

多核处理单元或组合。处理单元可以是诸如CPU(中央处理单元)的主处理器、诸如GPU(图形处理单元)的外围处理器或组合。存储器840可以是或包括存储器装置和存储器控制器。

[0125] 对存储器装置的引用可适用于不同的存储器类型。存储器装置一般涉及易失性存储器技术。易失性存储器是如果到装置的电力被中断则其状态(以及因此其上存储的数据)不确定的存储器。非易失性存储器指的是即使到装置的电力被中断其状态也确定的存储器。动态易失性存储器要求刷新存储在装置中的数据以维持状态。动态易失性存储器的一个示例包括DRAM(动态随机存取存储器),或者某种变体,诸如同步DRAM(SDRAM)。如本文中所描述的存储器子系统可与多个存储器技术兼容,诸如DDR3(最初由JEDEC(联合电子装置工程委员会)于2007年6月发布的双倍数据速率版本3 JESD79-3)、DDR4(DDR版本4 JESD79-4,由JEDEC于2012年9月公布的初始规范)、LPDDR3(JEDEC的2013年8月的低功率DDR版本3 JESD209-3)、LPDDR4(最初由JEDEC于2014年8月公布的LPDDR版本4 JESD209-4)、WIO2(最初由JEDEC于2014年8月公布的宽I/O 2(WideIO2) JESD229-2)、HBM(最初由JEDEC于2013年10月公布的高带宽存储器 JESD235)、DDR5(目前由JEDEC讨论中的DDR版本5)、LPDDR5(最初由JEDEC于2019年2月公布的LPDDR版本5 JESD209-5)、HBM2(目前由JEDEC讨论中的HBM版本2),或者其它存储器技术或其组合,以及基于此类规范的衍生或扩展的技术。

[0126] 一般而言,关于本文中的描述,在一个示例中,一种设备包括:提供具有宽度为X个元素的输入向量的第一存储器;以及具有所述第一存储器的芯片上的乘法累加(MAC)阵列,所述MAC阵列包括:存储权重向量的第二存储器;以及X个MAC电路,其动态地可配置成:基于输入向量的X个元素与权重向量的计算来计算输出,或者基于输入向量的单个元素与权重向量的计算来计算输出。

[0127] 在一个示例中,第一存储器包括静态随机存取存储器(SRAM)。在一个示例中,第二存储器包括寄存器堆。在一个示例中,第二存储器包括静态随机存取存储器(SRAM)。在一个示例中,MAC阵列在具有第一存储器的公共存储器管芯上,其中所述第一存储器是处理器的高速缓存存储器。在一个示例中,MAC阵列在具有第一存储器的公共存储器管芯上,其中第一存储器是处理器的高速暂存存储器。在一个示例中,MAC阵列在具有第一存储器的片上系统内,其中所述第一存储器是处理器的高速缓存存储器。在一个示例中,MAC阵列在具有第一存储器的片上系统内,其中第一存储器是处理器的高速暂存存储器。在一个示例中,MAC阵列进一步包括:在第一存储器和MAC阵列之间提供替代路径的复用器(mux);以及控制mux在替代路径之间进行选择的mux控制器。在一个示例中,mux控制器针对一维(1D)MxV计算要控制mux以用于一个输入向量元素到所有X个MAC电路。在一个示例中,mux控制器针对二维(2D)MxM计算要控制mux以用于X个不同的输入向量元素分别到所述X个MAC电路。

[0128] 一般而言,关于本文中的描述,在一个示例中,一种系统包括:提供具有宽度为X个元素的输入向量的处理单元的高速暂存存储器;以及耦合到处理单元的高速暂存存储器的硬件加速器,所述硬件加速器包括具有乘法累加(MAC)阵列的存储器中计算/近存储器计算(CIM)电路系统,所述MAC阵列包括:存储权重向量的本地存储器;以及X个MAC电路,其动态地可配置成:基于输入向量的X个元素与权重向量的计算来计算输出,或者基于输入向量的单个元素与权重向量的计算来计算输出。

[0129] 在一个示例中,高速缓存或高速暂存存储器包括静态随机存取存储器(SRAM)。在一个示例中,本地存储器包括寄存器堆。在一个示例中,本地存储器包括静态随机存取存储

器(SRAM)。在一个示例中,硬件加速器被集成在具有高速缓存或高速暂存存储器的公共存储器管芯上。在一个示例中,硬件加速器被集成在具有所述高速缓存或高速暂存存储器的片上系统上。在一个示例中,MAC阵列进一步包括:在高速缓存或高速暂存存储器和MAC阵列之间提供替代路径的复用器(mux);以及控制mux在替代路径之间进行选择的mux控制器。在一个示例中,mux控制器针对一维(1D)MxV计算要控制mux以用于一个输入向量元素到MAC阵列的所有X个MAC电路。在一个示例中,mux控制器针对二维(2D)MxM计算要控制mux以用于X个不同的输入向量元素分别到MAC阵列的X个MAC电路。在一个示例中,其中:处理单元包括多核主机处理器装置;该系统进一步包括通信地耦合到主机处理器的显示器;该系统进一步包括通信地耦合到主机处理器的网络接口;或者该系统进一步包括为该系统供电的电池。

[0130] 如本文中所图示的流程图提供了各种过程动作的序列的示例。流程图可指示要由软件或固件例程执行的操作,以及物理操作。流程图可图示有限状态机(FSM)的状态的实现的示例,有限状态机可用硬件和/或软件实现。尽管以特定顺序或次序示出,但除非另有规定,否则可修改动作的次序。从而,所图示的图应该仅被理解为示例,并且该过程能以不同的次序执行,并且一些动作可并行执行。此外,一个或多个动作可被省略;从而,并不是所有的实现都会执行所有的动作。

[0131] 就本文中描述的各种操作或功能而言,它们可被描述或定义为软件代码、指令、配置和/或数据。内容可以是直接可执行的(“对象”或“可执行的”形式)、源代码或差分代码(“Δ”或“补丁(patch)”代码)。本文中所描述的软件内容能经由其上存储有内容的制品(article of manufacture)提供,或经由操作通信接口以经由通信接口发送数据的方法提供。机器可读存储介质能使机器执行所描述的功能或操作,并且包括以由机器(例如,计算装置、电子系统等)可访问形式存储信息的任何机构(mechanism),诸如可记录/不可记录介质(例如,只读存储器(ROM)、随机存取存储器(RAM)、磁盘存储介质、光存储介质、闪速存储器装置等)。通信接口包括与硬连线、无线、光等介质中的任何介质通过接口连接以与另一装置通信的任何机构(诸如,存储器总线接口、处理器总线接口、因特网连接、盘控制器等)。可通过提供配置参数和/或发送信号来准备通信接口以提供描述软件内容的数据信号而配置通信接口。可经由发送到通信接口的一个或多个命令或信号访问该通信接口。

[0132] 本文中描述的各种组件可以是用于执行所描述的操作或功能的部件。本文中描述的每个组件包括软件、硬件或这些的组合。组件能被实现为软件模块、硬件模块、专用硬件(例如,专用硬件、专用集成电路(ASIC)、数字信号处理器(DSP)等)、嵌入式控制器、硬连线电路系统等。

[0133] 除了本文中所描述的之外,还能对本发明的实现和所公开的内容进行各种修改而不脱离它们的范围。因此,本文中的图示和示例应该以说明性的而非约束性的意义来解释。本发明的范围应该仅参考接下来的权利要求书来衡量。

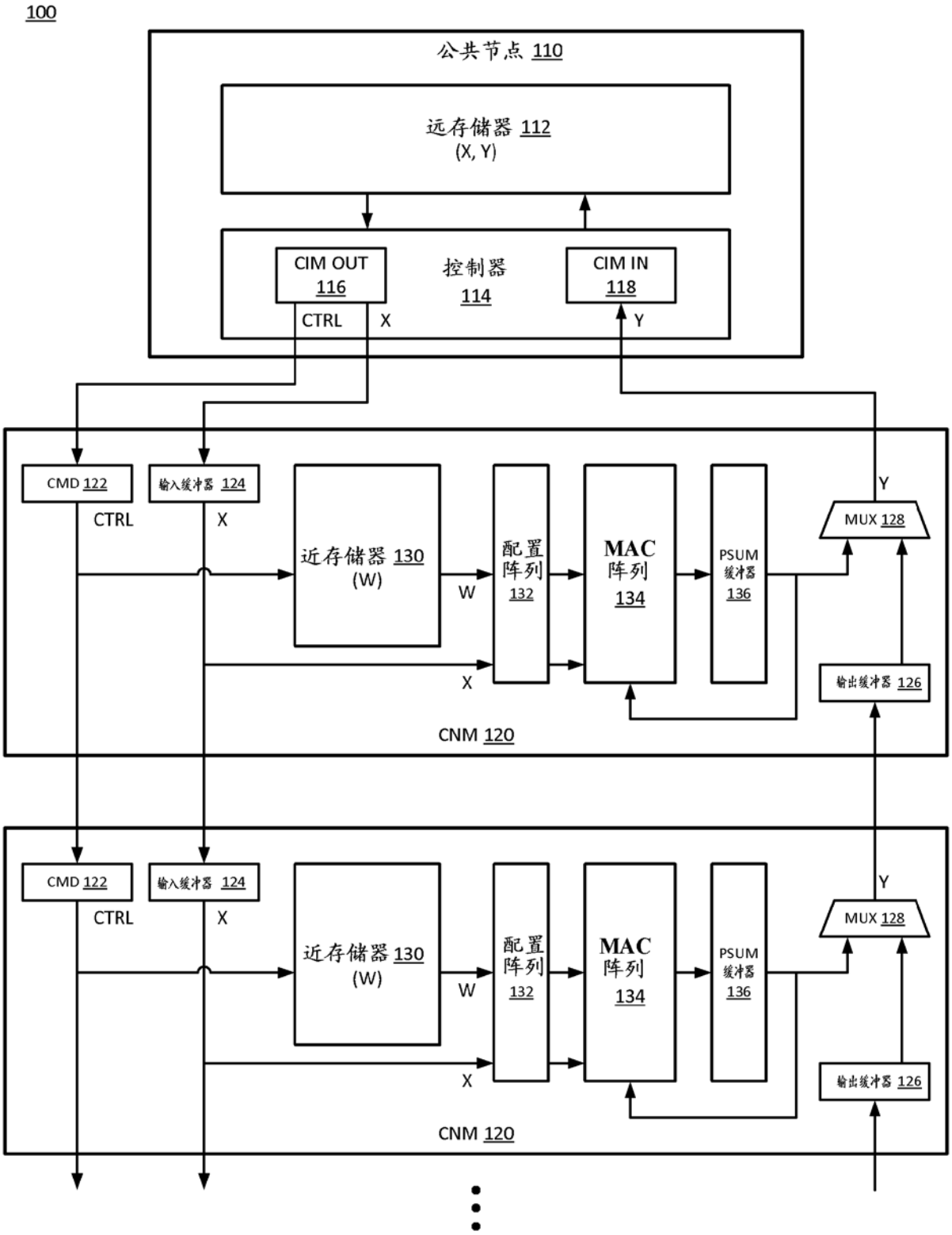

100
公共节点 110
远存储器 112
(X, Y)
CIM OUT
116
控制器
114
CIM IN
118
CTRL
X
Y
CMD 122
输入缓冲器 124
CTRL
X
近存储器 130
(W)
W
配置
阵列
132
MAC
阵列
134
PSUM
缓冲器
136
Y
MUX 128
输出缓冲器 126
X
CNM 120
CMD 122
输入缓冲器 124
CTRL
X
近存储器 130
(W)
W
配置
阵列
132
MAC
阵列
134
PSUM
缓冲器
136
Y
MUX 128
输出缓冲器 126
X
CNM 120

图 1

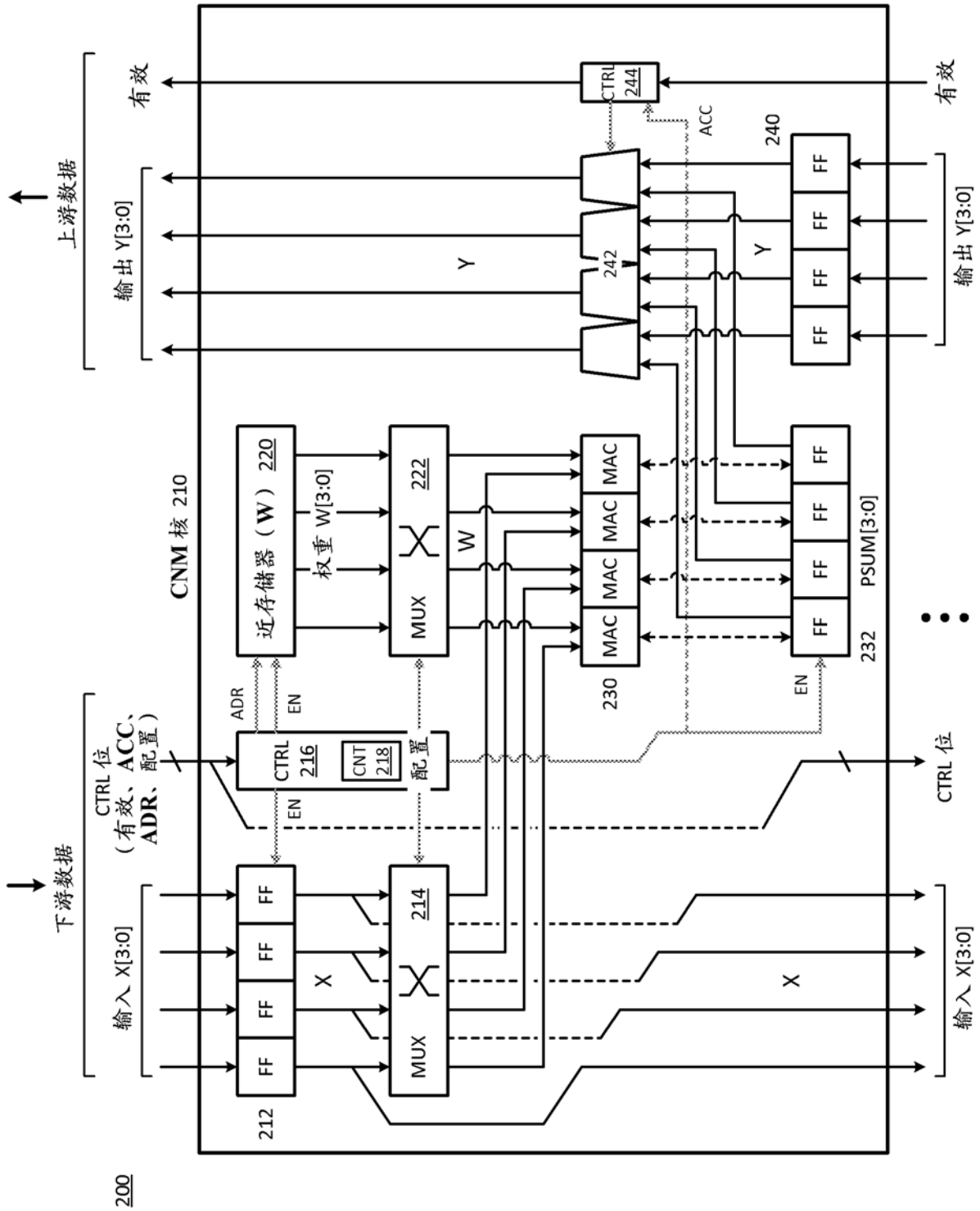

上游数据
输出 Y[3:0]
有效
CTRL 244
ACC
240
242
Y
FF
输出 Y[3:0]
有效
CNM 核 210
近存储器（W） 220
权重 W[3:0]
222
MUX
W
MAC
230
232
PSUM[3:0]
ADR
EN
CTRL 位
（有效、ACC、ADR、配置）
CTRL 216
CNT 218
配置
CTRL 位
下游数据
输入 X[3:0]
212
214
X
输入 X[3:0]
200

图 2

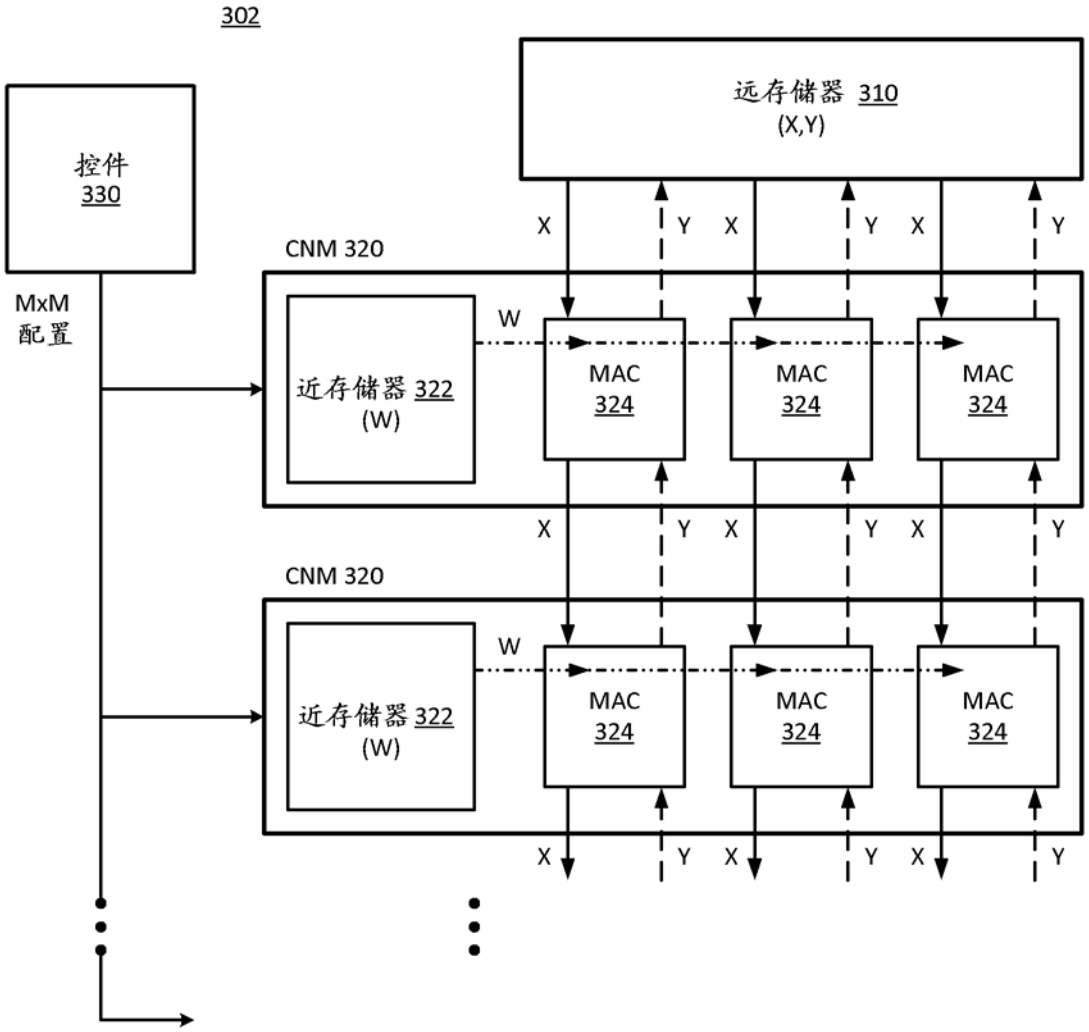

302
远存储器 310
(X,Y)
控件
330
MxM
配置
CNM 320
近存储器 322
(W)
W
MAC
324
X
Y
CNM 320
近存储器 322
(W)
W
MAC
324

图 3A

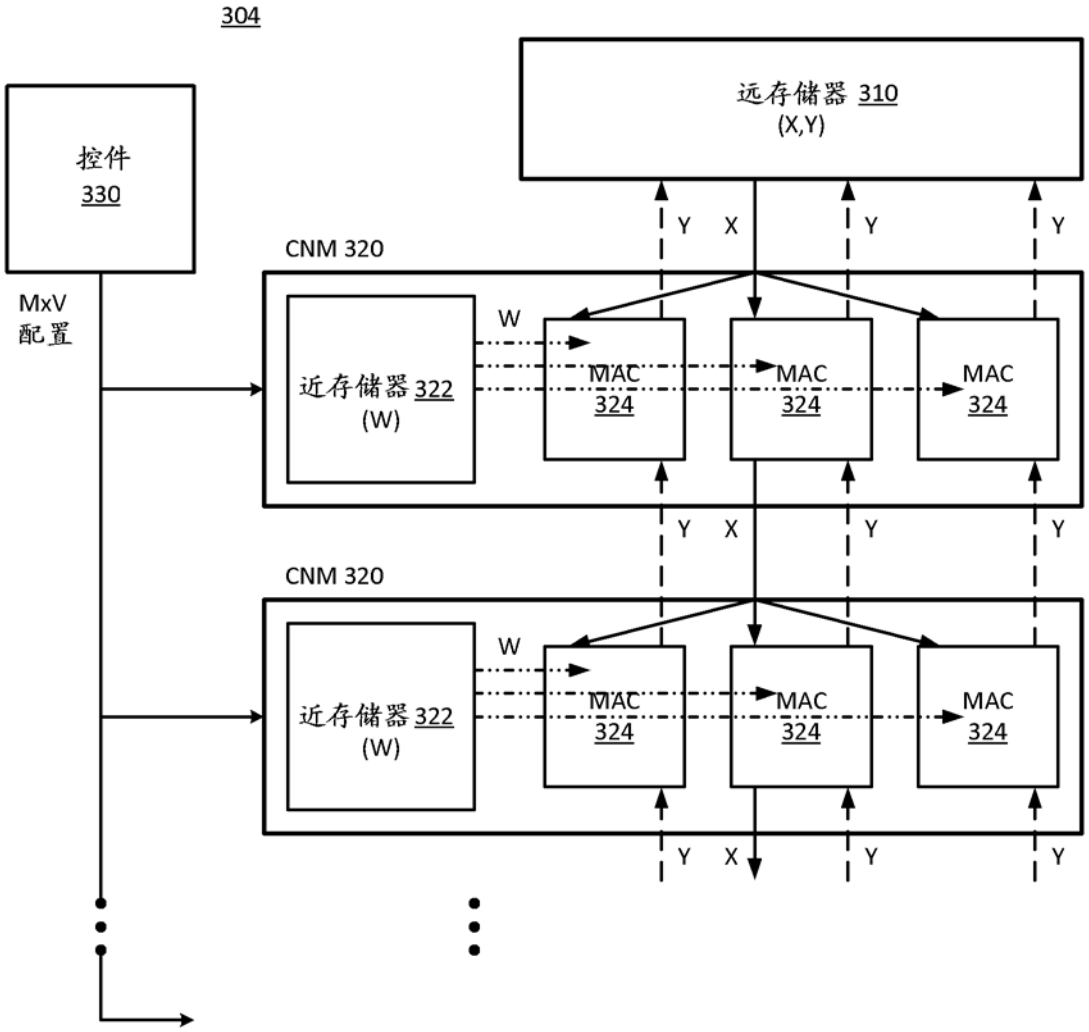

304
远存储器 310
(X,Y)
控件
330
MxV
配置
CNM 320
近存储器 322
(W)
W
MAC
324
Y
X
CNM 320
近存储器 322
(W)
W
MAC
324
Y
X

图 3B

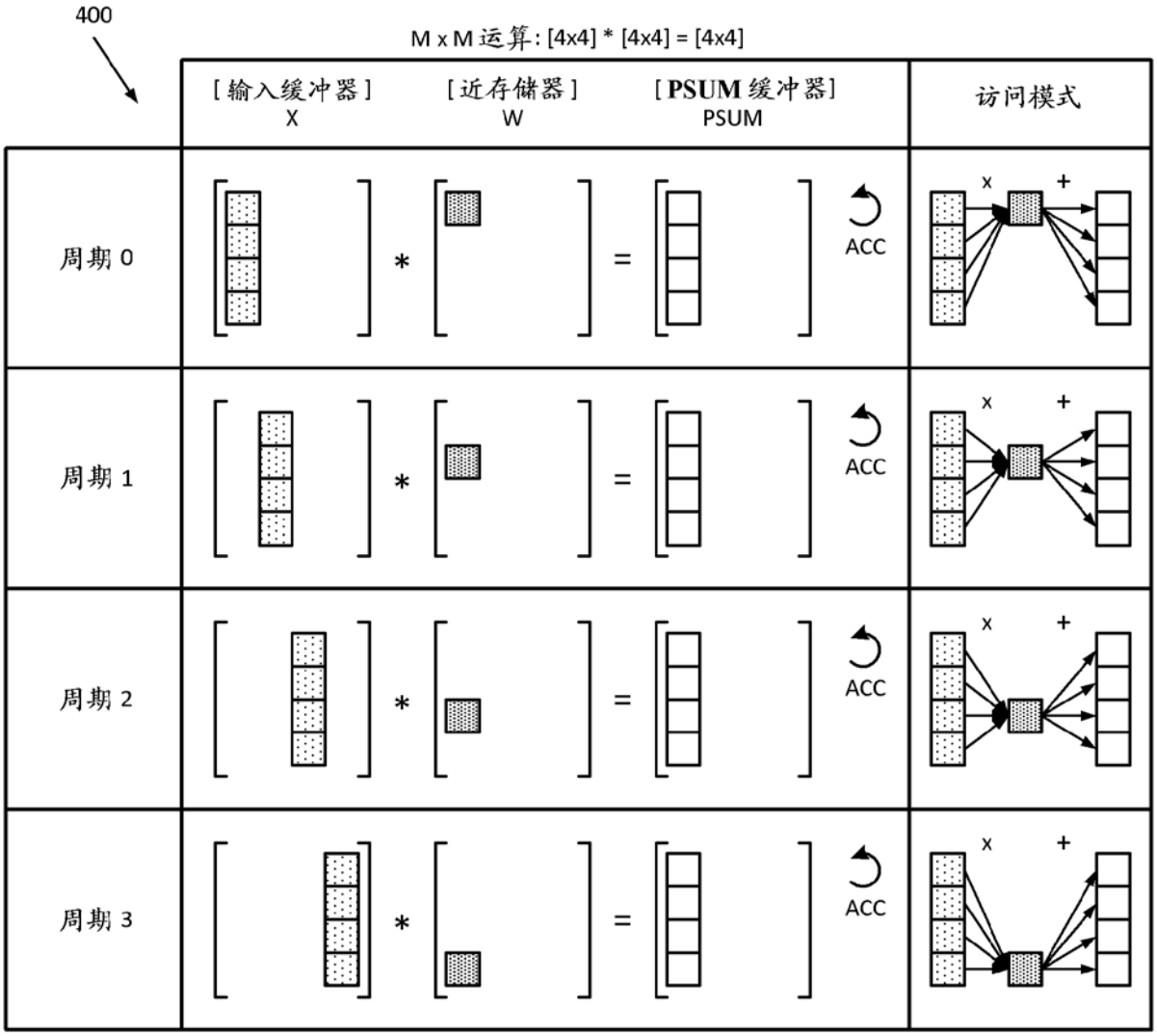

400
M x M 运算: [4x4] * [4x4] = [4x4]
[输入缓冲器]
X
[近存储器]
W
[PSUM 缓冲器]
PSUM
访问模式
周期 0
周期 1
周期 2
周期 3
ACC
x
+

图 4A

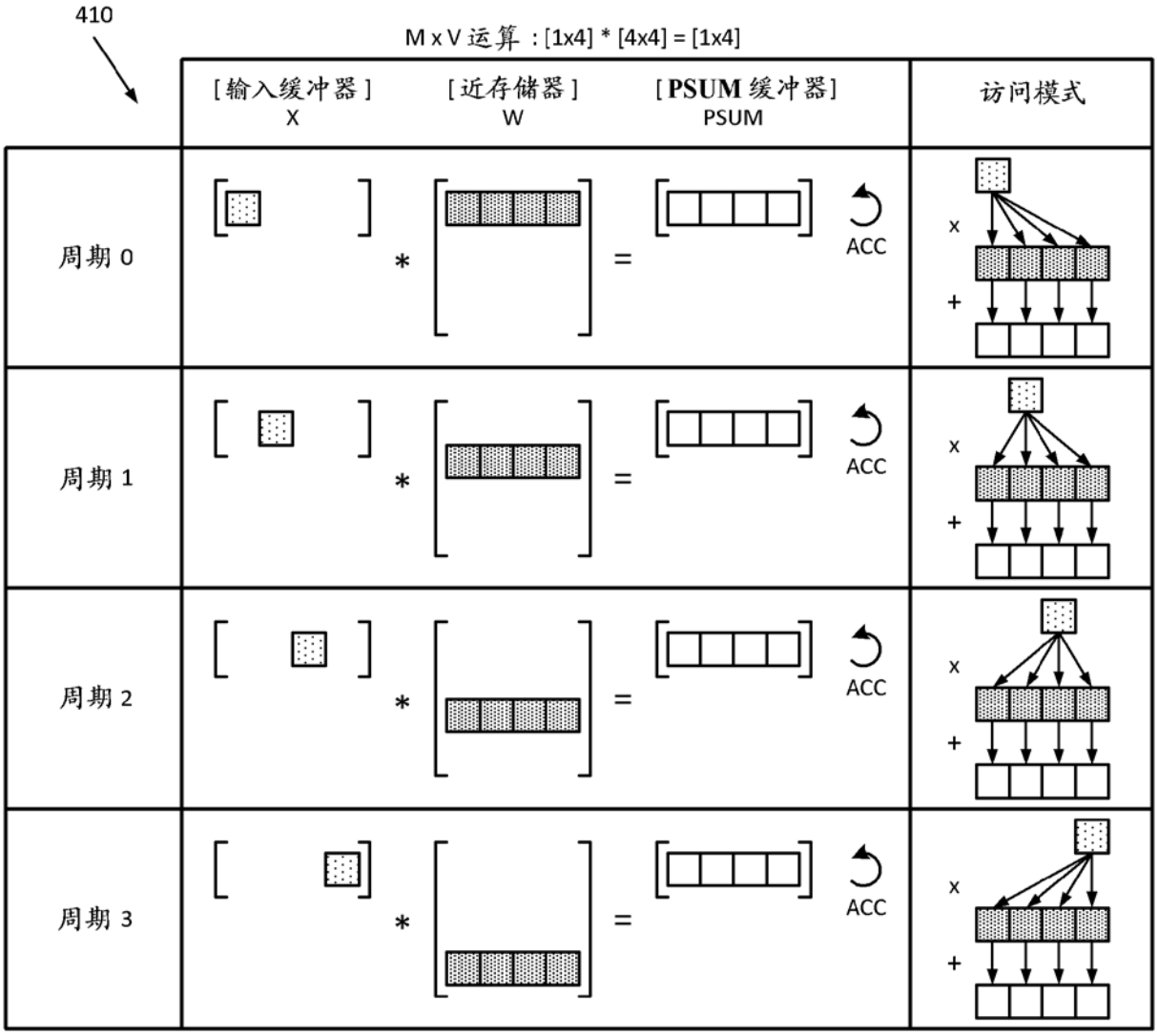

410
M x V 运算 : [1x4] * [4x4] = [1x4]
[输入缓冲器]
X
[近存储器]
W
[PSUM 缓冲器]
PSUM
访问模式
周期 0
周期 1
周期 2
周期 3
ACC
x
+

图 4B

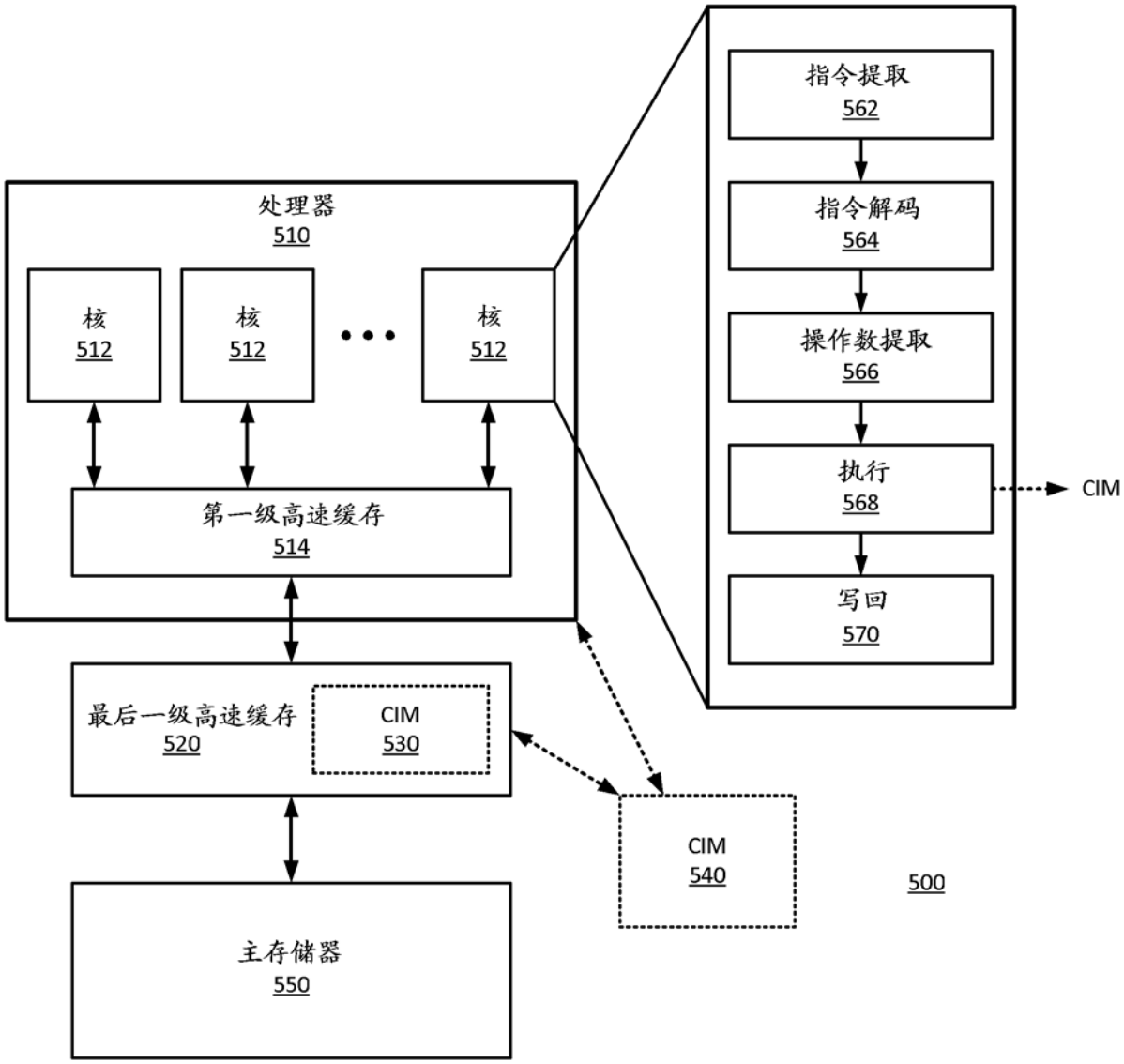

处理器
510
核
512
核
512
核
512
第一级高速缓存
514
最后一级高速缓存
520
CIM
530
CIM
540
主存储器
550
500
指令提取
562
指令解码
564
操作数提取
566
执行
568
写回
570
CIM

图 5

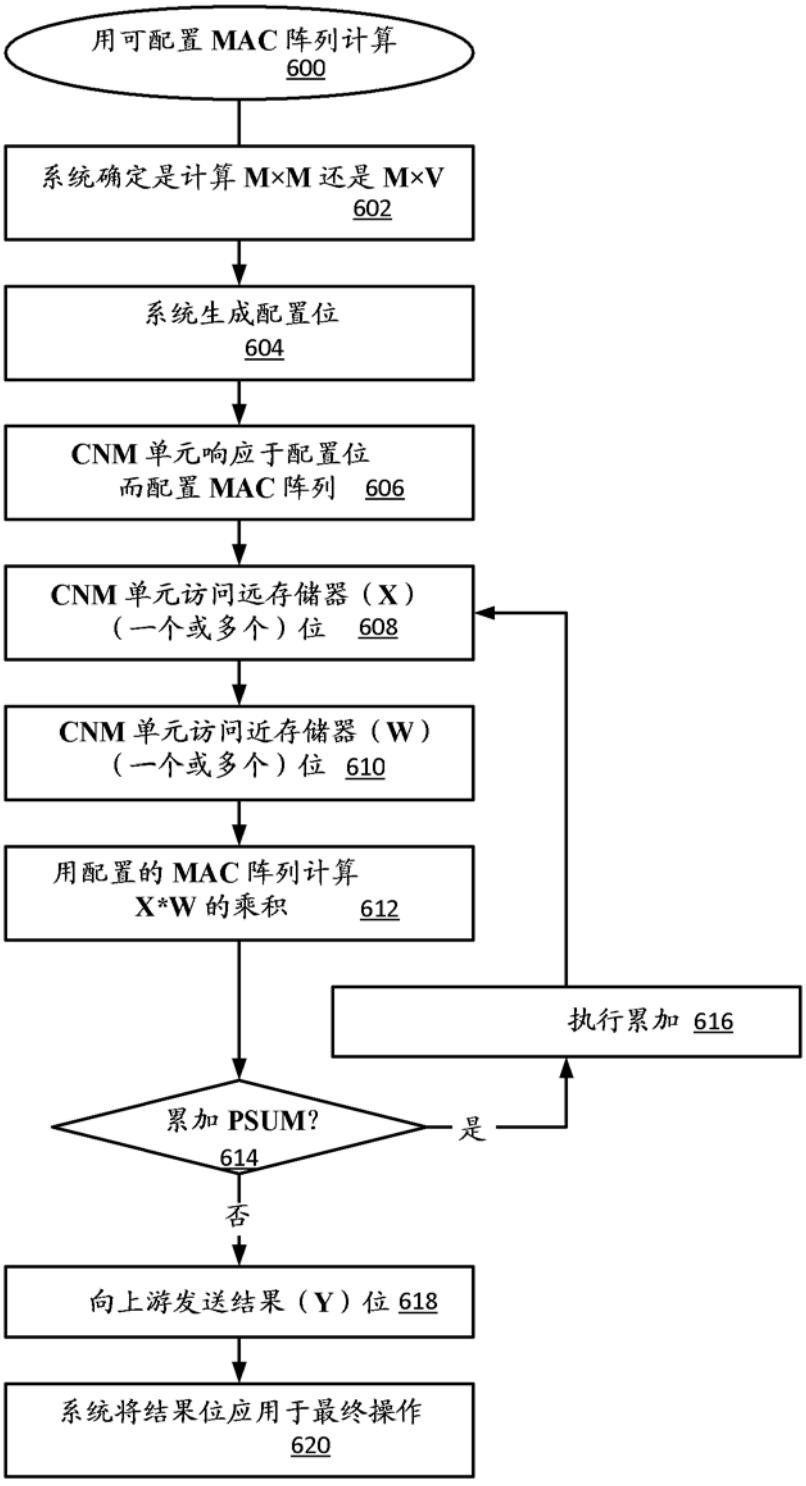

用可配置 MAC 阵列计算
600
系统确定是计算 M×M 还是 M×V
602
系统生成配置位
604
CNM 单元响应于配置位
而配置 MAC 阵列 606
CNM 单元访问远存储器（X）
（一个或多个）位 608
CNM 单元访问近存储器（W）
（一个或多个）位 610
用配置的 MAC 阵列计算
X*W 的乘积 612
执行累加 616
累加 PSUM?
614
是
否
向上游发送结果（Y）位 618
系统将结果位应用于最终操作
620

图 6

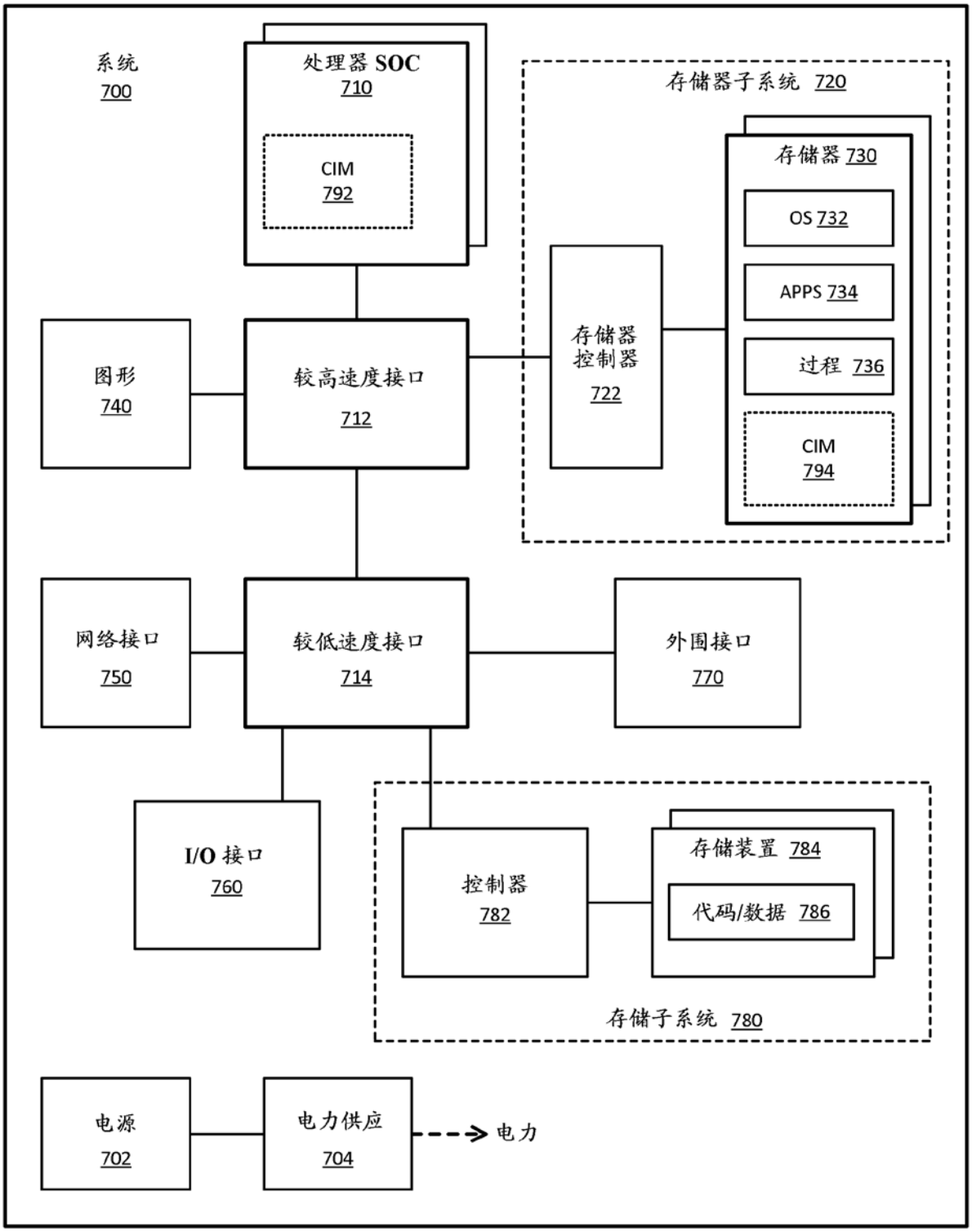

系统
700
处理器 SOC
710
CIM
792
存储器子系统 720
存储器 730
OS 732
APPS 734
过程 736
CIM
794
存储器
控制器
722
图形
740
较高速度接口
712
网络接口
750
较低速度接口
714
外围接口
770
I/O 接口
760
控制器
782
存储装置 784
代码/数据 786
存储子系统 780
电源
702
电力供应
704
电力

图 7

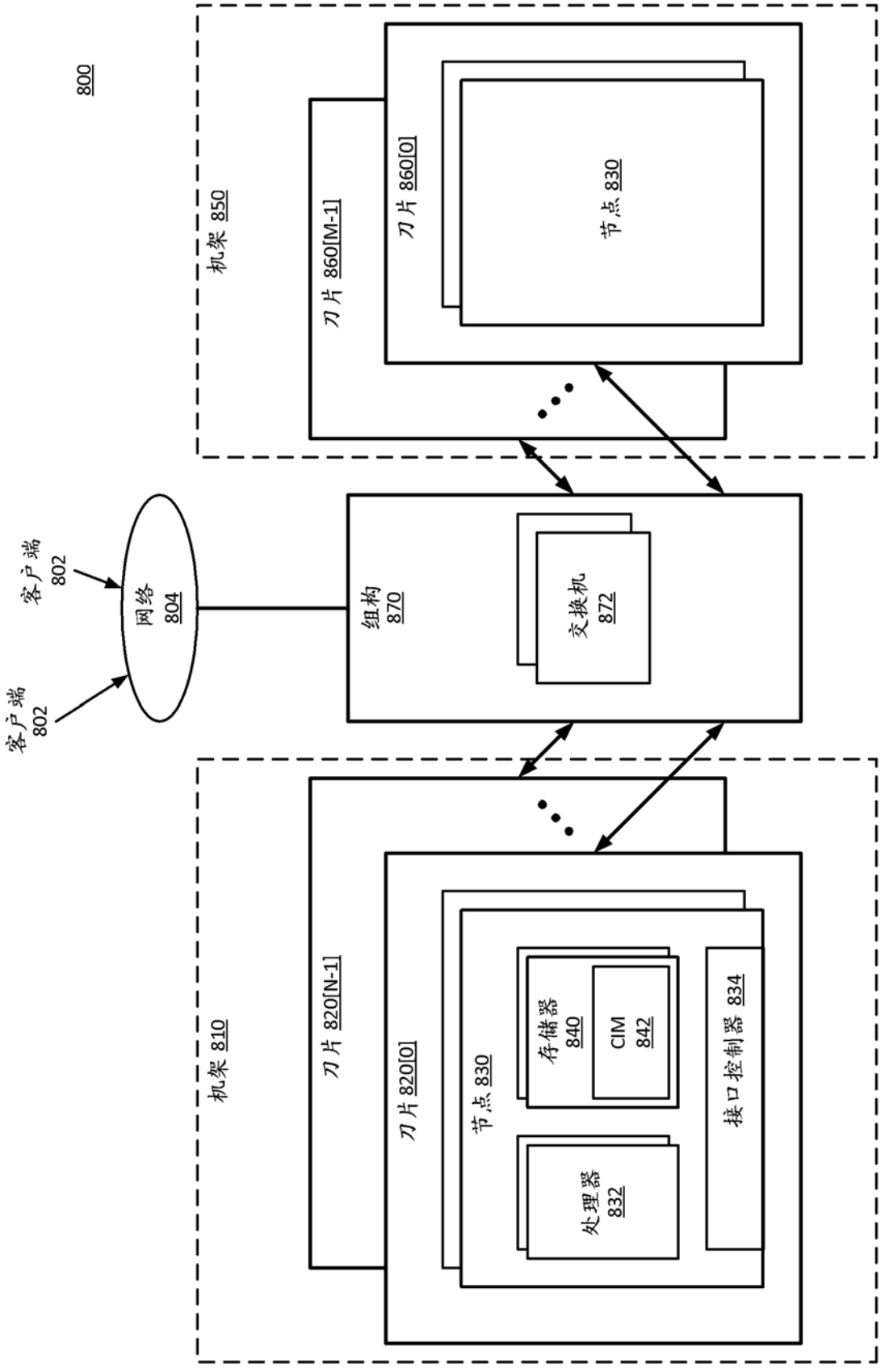

800
机架 850
刀片 860[M-1]
刀片 860[0]
节点 830
客户端 802
网络 804
客户端 802
组构 870
交换机 872
机架 810
刀片 820[N-1]
刀片 820[0]
节点 830
存储器 840
CIM 842
处理器 832
接口控制器 834

图 8