



(10) 授权公告号 CN 107003988 B

(45) 授权公告日 2021.04.02

(21) 申请号 201580062941.0
(22) 申请日 2015.11.19
(65) 同一申请的已公布的文献号
申请公布号 CN 107003988 A

(43) 申请公布日 2017.08.01

(30) 优先权数据
14382551.1 2014.12.19 EP

(85) PCT国际申请进入国家阶段日
2017.05.19

(86) PCT国际申请的申请数据
PCT/US2015/061455 2015.11.19

(87) PCT国际申请的公布数据
W02016/099780 EN 2016.06.23

(73) 专利权人 英特尔公司
地址 美国加利福尼亚州

(72) 发明人 E.埃雷罗阿韦利亚纳斯
G.图尔纳维蒂斯 F.C.普拉塔斯
M.卢庞 F.拉托雷 P.洛佩斯
A.J.福尔肯

(74) 专利代理机构 中国专利代理(香港)有限公司
72001

代理人 张凌苗 张涛

(51) Int.Cl.
G06F 17/15 (2006.01)
G06T 1/60 (2006.01)

(56) 对比文件
US 2009/0055565 A1,2009.02.26
CN 104205825 A,2014.12.10
CN 101436121 A,2009.05.20
US 2009/0055565 A1,2009.02.26

审查员 李易玮

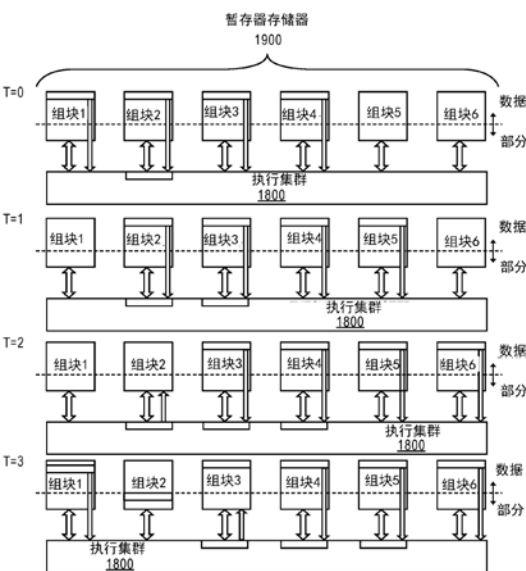
权利要求书4页 说明书23页 附图47页

(54) 发明名称

用于执行卷积运算的存储设备和方法

(57) 摘要

描述了用于执行卷积运算的存储设备和方法。例如,执行卷积运算的装置的一个实施例包括对输入数据和部分结果执行卷积运算的多个处理单元;统一暂存器存储器,其包括通过多个读/写端口通信地耦合到所述多个处理单元的多个存储器组块,将所述多个存储器组块中的每个进行划分以存储输入数据和部分结果;控制单元,用于将输入数据和部分结果分配给存储器组块,以根据指定数量的读/写端口和要执行的指定卷积运算来确保最低服务质量。



1. 一种执行卷积运算的装置,包括:

多个处理单元,所述多个处理单元用于对输入数据和部分结果执行卷积运算;

统一暂存器存储器,所述统一暂存器存储器包括通过多个读/写端口通信地耦合到所述多个处理单元的多个存储器组块,将所述多个存储器组块中的每个进行划分以存储输入数据和部分结果二者;

控制单元,所述控制单元用于将所述输入数据和部分结果分配给所述存储器组块,以根据指定数量的读/写端口和要执行的指定卷积运算来确保最低服务质量;

其中,将所述存储器组块中的每个在逻辑上划分成用于输入数据的第一区域和用于部分结果的第二区域,其中基于要执行的指定卷积运算来动态地调整分区的大小。

2. 根据权利要求1所述的装置,其中,所述控制单元将所述输入数据和部分结果分配给所述存储器组块,以确保在每个处理周期中访问存储器组块的不同集合。

3. 根据权利要求1所述的装置,其中,所述输入数据包括图像或矩阵,并且其中所述控制单元将映射所述图像/矩阵的每一行,因此针对所述处理单元在所述输入数据和部分结果上的每个相邻操作,每一行在不同的存储器组块中开始。

4. 根据权利要求3所述的装置,其中,对于针对在给定周期中进行处理所需的图像或矩阵的N个元素的列,所述控制单元将分配所述输入数据和部分结果以确保所有N个元素在所述周期中都可以从不同存储器组块读取。

5. 根据权利要求1所述的装置,其中,所述控制单元将所述输入数据和部分结果分配给所述存储器组块,以确保在指定卷积窗口内的所有输入数据和部分结果在每个处理周期上都对处理单元可用。

6. 根据权利要求5所述的装置,其中,所述卷积窗口基于指定的滤波器大小和同时要处理的滤波器的实例的数量。

7. 根据权利要求1所述的装置,其中,所述存储器组块中的每个包括单个读/写端口。

8. 根据权利要求1所述的装置,还包括:

一个或多个部分写回缓冲器,用于在存储在所述多个存储器组块之一中之前临时缓冲由处理单元中的一个或多个生成的新的部分结果。

9. 根据权利要求1所述的装置,还包括:

输入多路复用器,其通信地耦合以从输入/输出接口接收新的输入数据,所述输入多路复用器由所述控制单元控制以使所述输入数据对准并存储在指定的存储器组块中。

10. 根据权利要求1所述的装置,还包括:

输出多路复用器,其通过处理单元接口通信地耦合到所述处理单元和所述存储器组块的读/写端口,所述输出多路复用器由控制单元控制以对准从所述存储器组块接收的输入数据和部分结果以及将所述输入数据和部分结果提供给一个或多个指定的处理单元。

11. 一种方法,包括:

使用输入数据和部分结果在多个处理单元上执行卷积运算;

对统一暂存器存储器的多个存储器组块中的每个进行划分以存储所述输入数据和部分结果两者,所述多个存储器组块通过多个读/写端口通信地耦合到所述多个处理单元;

将所述输入数据和部分结果分配给所述存储器组块,以根据指定数量的读/写端口和要执行的指定卷积运算来确保最低服务质量;

其中,将所述存储器组块中的每个在逻辑上划分成用于输入数据的第一区域和用于部分结果的第二区域,其中基于要执行的指定卷积运算来动态地调整分区的大小。

12.根据权利要求11所述的方法,其中,将所述输入数据和部分结果分配给所述存储器组块,以确保在每个处理周期中访问存储器组块的不同集合。

13.根据权利要求11所述的方法,其中,所述输入数据包括图像或矩阵,并且其中将所述图像/矩阵的每一行映射到所述存储器组块,因此针对所述处理单元在所述输入数据和部分结果上的每个相邻的操作,每一行在不同的存储器组块中开始。

14.根据权利要求13所述的方法,其中,对于针对在给定周期中进行处理所需的图像或矩阵的N个元素的列,分配所述输入数据和部分结果以确保所有N个元素在所述周期中都可以从不同的存储器组块可用。

15.根据权利要求11所述的方法,其中,将所述输入数据和部分结果分配给所述存储器组块,以确保在指定卷积窗口内的所有输入数据和部分结果在每个处理周期上都对所述处理单元可用。

16.根据权利要求15所述的方法,其中,所述卷积窗口基于指定的滤波器大小和同时要处理的滤波器的实例的数量。

17.根据权利要求11所述的方法,其中,存储器组块中的每个包括单个读/写端口。

18.根据权利要求11所述的方法,还包括:

在存储在所述多个存储器组块之一中之前临时缓冲由处理单元中的一个或多个生成的新的部分结果。

19.根据权利要求11所述的方法,还包括:

在输入多路复用器处接收新的输入数据,所述输入多路复用器被控制以使所述输入数据对准并存储在指定的存储器组块中。

20.根据权利要求11所述的方法,还包括:

对准从所述存储器组块接收的输入数据和部分结果;以及

将对准的输入数据和部分结果提供给一个或多个指定的处理单元。

21.一种系统,包括:

存储器,其用于存储指令和数据;

多个核,其用于执行所述指令并处理所述数据;

图形处理器,其用于响应于某些指令来执行图形操作;

网络接口,其用于通过网络接收和传输数据;

接口,其用于接收来自鼠标或光标控制设备的用户输入;和

装置,用于执行卷积运算,其包括:

多个处理单元,所述多个处理单元用于对输入数据和部分结果执行卷积运算;

统一暂存器存储器,所述统一暂存器存储器包括通过多个读/写端口通信地耦合到所述多个处理单元的多个存储器组块,将所述多个存储器组块中的每个进行划分以存储输入数据和部分结果二者;

控制单元,所述控制单元用于将所述输入数据和部分结果分配给所述存储器组块,以根据指定数量的读/写端口和要执行的指定卷积运算来确保最低服务质量;

其中,将所述存储器组块中的每个在逻辑上划分成用于输入数据的第一区域和用于部

分结果的第二区域,其中基于要执行的指定卷积运算来动态地调整分区的大小。

22.根据权利要求21所述的系统,其中,所述控制单元将所述输入数据和部分结果分配给所述存储器组块,以确保在每个处理周期中访问存储器组块的不同集合,因此不存在与忙组块的冲突。

23.根据权利要求21所述的系统,其中,所述输入数据包括图像或矩阵,并且其中所述控制单元将映射所述图像/矩阵的每一行,因此针对所述处理单元在所述输入数据和部分结果上的每个相邻操作,每一行在不同的存储器组块中开始。

24.一种装置,包括:

用于使用输入数据和部分结果在多个处理单元上执行卷积运算的装置;

用于对统一暂寄存器存储器的多个存储器组块中的每个进行划分以存储所述输入数据和部分结果两者的装置,所述多个存储器组块通过多个读/写端口通信地耦合到所述多个处理单元;

用于将所述输入数据和部分结果分配给所述存储器组块以根据指定数量的读/写端口和要执行的指定卷积运算来确保最低服务质量的装置;

其中,将所述存储器组块中的每个在逻辑上划分成用于输入数据的第一区域和用于部分结果的第二区域,其中基于要执行的指定卷积运算来动态地调整分区的大小。

25.根据权利要求24所述的装置,其中,将所述输入数据和部分结果分配给所述存储器组块,以确保在每个处理周期中访问存储器组块的不同集合。

26.根据权利要求24所述的装置,其中,所述输入数据包括图像或矩阵,并且其中将所述图像/矩阵的每一行映射到所述存储器组块,因此针对所述处理单元在所述输入数据和部分结果上的每个相邻的操作,每一行在不同的存储器组块中开始。

27.根据权利要求26所述的装置,其中,对于针对在给定周期中进行处理所需的图像或矩阵的N个元素的列,分配所述输入数据和部分结果以确保所有N个元素在所述周期中都可以从不同的存储器组块可用。

28.根据权利要求24所述的装置,其中,将所述输入数据和部分结果分配给所述存储器组块,以确保在指定卷积窗口内的所有输入数据和部分结果在每个处理周期上都对所述处理单元可用。

29.根据权利要求28所述的装置,其中,所述卷积窗口基于指定的滤波器大小和同时要处理的滤波器的实例的数量。

30.根据权利要求24所述的装置,其中,存储器组块中的每个包括单个读/写端口。

31.根据权利要求24所述的装置,还包括:

用于在存储在所述多个存储器组块之一中之前临时缓冲由处理单元中的一个或多个生成的新的部分结果的装置。

32.根据权利要求24所述的装置,还包括:

用于在输入多路复用器处接收新的输入数据的装置,所述输入多路复用器被控制以使所述输入数据对准并存储在指定的存储器组块中。

33.根据权利要求24所述的装置,还包括:

用于对准从所述存储器组块接收的输入数据和部分结果的装置;以及

用于将对准的输入数据和部分结果提供给一个或多个指定的处理单元的装置。

34.一种具有存储于其上的指令的计算机可读介质,当所述指令被执行时使得计算设备执行根据权利要求11-20中任一项所述的方法。

用于执行卷积运算的存储设备和方法

[0001] 相关申请的交叉引用

[0002] 本申请根据35 U.S.C. 第119条要求于2014年12月19日提交的题为“Storage Device And Method For Performing Convolution Operations”的欧洲专利申请号EP14382551.1的优先权,其全部内容通过引用并入本文。

技术领域

[0003] 本发明一般涉及计算机处理器的领域。更具体地,本发明涉及一种用于执行卷积运算的存储设备和方法。

背景技术

[0004] ANN通常被呈现为可以计算来自输入的值的互连“神经元”的系统。ANN表示用于学习和识别模式的最相关和广泛的技术之一。因此,ANN已经形成一种用于改善用户体验的直观的人类/设备交互的有效解决方案,这是一种称为“认知计算”的新计算范例(paradigm)。除了其他使用之外,ANN可以用于成像处理、语音和对象识别或自然语言处理。卷积神经网络(CNN)或深度信念网络(DBN)只是采用ANN算法的计算范例的几个示例。

附图说明

[0005] 可以从结合以下附图的以下详细描述获得对本发明的更好理解,在附图中:

[0006] 图1A是图示了根据本发明的实施例的示例性按顺序取得(fetch)、解码、退役(retire)流水线和示例性寄存器重命名、乱序发布/执行流水线两者的框图;

[0007] 图1B是图示了根据本发明的实施例的在处理器中包括的按顺序取得、解码、退役核的示例性实施例和示例性寄存器重命名、乱序发布/执行架构核两者的框图;

[0008] 图2是根据本发明的实施例的具有集成的存储器控制器和图形的多核处理器和单核处理器的框图;

[0009] 图3图示了根据本发明的一个实施例的系统的框图;

[0010] 图4图示了根据本发明的实施例的第二系统的框图;

[0011] 图5图示了根据本发明的实施例的第三系统的框图;

[0012] 图6图示了根据本发明的实施例的片上系统(SoC)的框图;

[0013] 图7图示了根据本发明的实施例的对比使用软件指令转换器将源指令集中的二进制指令转换为目标指令集中的二进制指令的框图;

[0014] 图8图示了具有多个层的完全连接的人工神经网络(ANN)的方案;

[0015] 图9图示了示例性神经形态(neuromorphic)加速器架构;

[0016] 图10图示了针对单片配置和基于每层神经元的数量来利用分布式配置的本发明实施例二者的带宽需求;

[0017] 图11图示了单片架构与基于每层神经元的数量的分布式神经形态架构之间的比较;

- [0018] 图12A图示了其中神经形态加速器通信地耦合到处理器的一个实施例；
- [0019] 图12B图示了其中神经形态加速器集成在处理器内的一个实施例；
- [0020] 图13图示了包括处理单元(PU)的集合的神经形态加速器的一个实施例；
- [0021] 图14A-图14B图示了根据本发明的一个实施例的完全连接的1对1操作可以如何映射；
- [0022] 图15A-图15B图示了根据本发明的另一实施例的完全连接的1对N操作可以如何映射；
- [0023] 图16图示了执行变化的滤波器大小和多个存储器大小的卷积(convolution)所需的能量；
- [0024] 图17A-图17B图示了用于执行集群(cluster)的不同存储布置；
- [0025] 图18A图示了存储器组织的一个实施例,其中在不同类型的数据之间共享存储器组块(memory bank)和互连；
- [0026] 图18B图示了根据一个实施例的可以对其应用滤波器的示例性图像；
- [0027] 图19图示了根据本发明的一个实施例的包括多个组块的暂存器(scratchpad)存储器；
- [0028] 图20A-图20B图示了针对暂存器存储器的不同示例性利用；
- [0029] 图21图示了作为多个滤波器的函数的外部存储器访问中的百分比减少；
- [0030] 图22图示了根据本发明的一个实施例的卷积神经网络加速器；
- [0031] 图23示出了在卷积器中使用的输入数据的示例集合；
- [0032] 图24示出了来自图23的元素可以如何在物理存储器组块中映射；
- [0033] 图25图示了用于图像处理的通用加速器的高级框图；
- [0034] 图26示出了其中以1像素的步幅从左上到右下角应用 4×4 像素的内核的示例性图像；
- [0035] 图27图示了作为卷积配置的函数的多个读取端口；
- [0036] 图28图示了具有4个集群的加速器的一个实施例；
- [0037] 图29图示了根据本发明的一个实施例的数据容器；
- [0038] 图30图示了示出在本发明的一个实施例中实现的对准模式的表格；
- [0039] 图31图示了示出在本发明的一个实施例中采用的逻辑的表格；
- [0040] 图32图示了根据本发明的一个实施例的如何将容器、对准逻辑和选择逻辑互连以产生输出分组；
- [0041] 图33图示了基于神经元的加权输入来计算神经元的输出的示例性激活函数(Activation Function)(AF)；
- [0042] 图34A-图34B图示了使用查找表(LUT)的通过固定值近似的函数中的每个点的计算；
- [0043] 图35A-图35B图示了每个点的计算可以使用LUT由具有给定斜率的线段来近似；
- [0044] 图36图示了包括针对AF的特性进行优化的分段(piecewise)插值近似单元的本发明的一个实施例；
- [0045] 图37A-图37B图示了数据集合,其示出了多态(polymorphic)解码器的实施例如何将每个输入映射到横坐标空间中的范围；

- [0046] 图38图示了具有最大误差 2^{-5} 和5个线段的sigmoid函数的示例；
- [0047] 图39图示了包括LUT和多态解码器的架构的一个实施例；和
- [0048] 图40图示了多态解码器的一个实施例。

具体实施方式

[0049] 在随后的描述中,为了解释的目的,阐述了许多具体细节,以便提供对下面描述的本发明的实施例的透彻理解。然而,对于本领域技术人员清楚的是:可以在没有这些具体细节中的一些的情况下实践本发明的实施例。在其他情况下,以框图形式示出公知的结构和设备以避免模糊本发明的实施例的基本原理。

[0050] 示例性处理器架构和数据类型

[0051] 图1A是图示了根据本发明的实施例的示例性按顺序取得、解码、退役流水线和示例性寄存器重命名、乱序发布/执行流水线两者的框图。图1B是图示了根据本发明的实施例的在处理器中包括的按顺序取得、解码、退役核和示例性寄存器重命名、乱序发布/执行架构核的框图。图1A-图1B中的实线框图示了流水线和核的按顺序部分,而虚线框的可选添加则图示了寄存器重命名、乱序发布/执行流水线和核。

[0052] 在图1A中,处理器流水线100包括取得级102、长度解码级104、解码级106、分配级108、重命名级110、调度(也称为分派(dispatch)或发布)级112、寄存器读取/存储器读取级114、执行级116、写回/存储器写入级118、异常处理级122和提交(commit)级124。

[0053] 图1B示出了处理器核190,其包括耦合到执行引擎单元150的前端单元130,并且这两者都耦合到存储器单元170。核190可以是精简指令集计算(RISC)核、复杂指令集计算(CISC)核、超长指令字(VLIW)核或混合或替代的核类型。作为又一选项,核190可以是专用核,诸如例如网络或通信核、压缩引擎、协处理器核、通用计算图形处理单元(GPGPU)核、图形核等。

[0054] 前端单元130包括耦合到指令高速缓存单元134的分支预测单元132,该指令高速缓存单元134耦合到指令转译后备缓冲器(TLB)136,该指令转译后备缓冲器(TLB)136耦合到指令取得单元138,该指令取得单元138耦合到解码单元140。解码单元140(或解码器)可以对指令进行解码,并且生成作为输出的一个或多个微操作、微代码入口点、微指令、其他指令或其他控制信号,其解码自、或否则反映、或导出自原始指令。解码单元140可以使用各种不同的机制来实现。合适机制的示例包括但不限于查找表、硬件实现、可编程逻辑阵列(PLA)、微代码只读存储器(ROM)等。在一个实施例中,核190包括微代码ROM或存储用于某些宏指令的微代码(例如,在解码单元140中或以否则在前端单元130内)的其他介质。解码单元140耦合到执行引擎单元150中的重命名/分配器单元152。

[0055] 执行引擎单元150包括重命名/分配器单元152,该重命名/分配器单元152耦合到退役单元154和一个或多个调度器单元156的集合。(一个或多个)调度器单元156表示任意数量的不同调度器,包括保留站、中心指令窗口等。(一个或多个)调度器单元156耦合到(一个或多个)物理寄存器文件单元158。(一个或多个)物理寄存器文件单元158中的每个表示一个或多个物理寄存器文件,其中不同的物理寄存器文件存储一个或多个不同的数据类型,诸如标量整数、标量浮点、打包(packed)整数、打包浮点、向量整数、向量浮点、状态(例如,作为要执行的下一个指令的地址的指令指针)等。在一个实施例中,(一个或多个)物理

寄存器文件单元158包括向量寄存器单元、写掩码(mask)寄存器单元和标量寄存器单元。这些寄存器单元可以提供架构向量寄存器、向量掩码寄存器和通用寄存器。(一个或多个)物理寄存器文件单元158被退役单元154重叠以图示可以在其中(例如,使用(一个或多个)重新排序缓冲器和(一个或多个)退役寄存文件;使用(一个或多个)未来的文件、(一个或多个)历史缓冲器和(一个或多个)退役寄存器文件;使用寄存器映射和寄存器池等)实现寄存器重命名和乱序执行的各种方式。退役单元154和(一个或多个)物理寄存器文件单元158耦合到(一个或多个)执行集群160。(一个或多个)执行集群160包括一个或多个执行单元162的集合和一个或多个存储器访问单元164的集合。执行单元162可以并且在各种类型的数据(例如,标量浮点、打包整数、打包浮点、向量整数、向量浮点)上执行各种操作(例如,移位(shift)、加法、减法、乘法)。虽然一些实施例可以包括专用于特定功能或功能集合的数个执行单元,但是其他实施例可以仅包括一个执行单元或者全部执行全部功能的多个执行单元。(一个或多个)调度器单元156、(一个或多个)物理寄存器文件单元158和(一个或多个)执行集群160被示出为可能是复数的,因为某些实施例针对某些类型的数据/操作创建单独的流水线(例如,标量整数流水线、标量浮点/打包整数/打包浮点/向量整数/向量浮点流水线,和/或存储器访问流水线,其中的每个具有它们各自的调度器单元、(一个或多个)物理寄存器文件单元和/或执行集群--并且在单独的存储器访问流水线的情况下,某些实施例被实现为其中只有该流水线的执行集群具有(一个或多个)存储器访问单元164)。还应该理解,在使用单独的流水线的情况下,这些流水线中的一个或多个可以是乱序发布/执行,并且其余的可以是按顺序的。

[0056] 存储器访问单元164的集合耦合到存储器单元170,该存储器单元170包括数据TLB单元172,该数据TLB单元172耦合到数据高速缓存单元174,该数据高速缓存单元174耦合到第2级(L2)高速缓存单元176。在一个示例性实施例中,存储器访问单元164可以包括加载单元、存储地址单元和存储数据单元,其每个单元均耦合到存储器单元170中的数据TLB单元172。指令高速缓存单元134进一步耦合到存储器单元170中的第2级(L2)高速缓存单元176。L2高速缓存单元176耦合到一个或多个其他级别的高速缓存并且最终耦合到主存储器。

[0057] 通过示例的方式,示例性寄存器重命名、乱序发布/执行核架构可以实现流水线100如下:1)指令取得138执行取得和长度解码级102和104;2)解码单元140执行解码级106;3)重命名/分配器单元152执行分配级108和重命名级110;4)(一个或多个)调度器单元156执行调度级112;5)(一个或多个)物理寄存器文件单元158和存储单元170执行寄存器读取/存储读取级114;执行集群160执行执行级116;6)存储器单元170和(一个或多个)物理寄存器文件单元158执行写回/存储器写入级118;7)在异常处理级122中可能涉及各种单元;以及8)退役单元154和(一个或多个)物理寄存器文件单元158执行提交级124。

[0058] 核190可以支持一个或多个指令集(例如,x86指令集(具有已经向较新版本添加的一些扩展);加利福尼亚州桑尼维尔的MIPS Technologies的MIPS指令集;加利福尼亚州桑尼维尔的ARM Holdings的ARM指令集(具有诸如NEON的可选附加扩展)),包括本文所述的(一个或多个)指令。在一个实施例中,核190包括支持打包数据指令集扩展(例如,AVX1、AVX2和/或一些形式的通用向量友好指令格式(U=0和/或U=1)),如下所述)的逻辑,从而允许使用打包数据来执行许多多媒体应用所使用的操作。

[0059] 应当理解,核可以支持多线程(执行操作或线程的两个或更多并行集合),并且可

以以各种方式这样做,各种方式包括时间分片多线程、同时多线程(其中单个物理核针对物理核是同时多线程的线程中的每个提供逻辑核)或其组合(例如,诸如在Intel®超线程技术中的此后的时间分片取得和解码以及同时多线程)。

[0060] 虽然在乱序执行的上下文中描述了寄存器重命名,但是应当理解,可以在按顺序的架构中使用寄存器重命名。虽然处理器的图示实施例还包括单独的指令和数据高速缓存单元134/174和共享L2高速缓存单元176,但是替代实施例可以具有用于指令和数据二者的诸如例如第1级(L1)内部高速缓存的单个内部高速缓存,或多级内部高速缓存。在一些实施例中,系统可以包括内部高速缓存和在核和/或处理器外部的的外部高速缓存的组合。可替代地,所有高速缓存可以在核和/或处理器外部。

[0061] 图2是根据本发明的实施例的处理器200的框图,处理器200可以具有多于一个核、可以具有集成存储器控制器并且可以具有集成图形。图2中的实线框示出了具有单个核202A、系统代理210、一个或多个总线控制器单元216的集合的处理器200,而虚线框的可选添加图示了具有多个核202A-N、在系统代理单元210中的一个或多个集成存储器控制器单元214的集合和专用逻辑208的替代处理器200。

[0062] 因此,处理器200的不同实现可以包括:1)CPU,其具有是集成图形和/或科学(吞吐量)逻辑的专用逻辑208(其可以包括一个或多个核),以及是一个或多个通用核的核202A-N(例如,通用按顺序核、通用乱序核、两者的组合);2)协处理器,其具有是主要用于图形和/或科学(吞吐量)的大量专用核的核202A-N;和3)协处理器,其具有是大量通用按顺序核的核202A-N。因此,处理器200可以是通用处理器、协处理器或专用处理器,诸如例如网络或通信处理器、压缩引擎、图形处理器、GPGPU(通用图形处理单元)、高吞吐量多集成核(MIC)协处理器(包括30个或更多个核)、嵌入式处理器等。处理器可以在一个或多个芯片上实现。处理器200可以是一个或多个衬底的一部分和/或使用诸如例如BiCMOS、CMOS或NMOS之类的多种工艺技术中的任何技术在一个或多个衬底上实现。

[0063] 存储器层级包括核内的一个或多个级别的高速缓存、集合或一个或多个共享高速缓存单元206和耦合到集成存储器控制器单元214的集合的外部存储器(未示出)。共享高速缓存单元206的集合可以包括一个或多个中级高速缓存,诸如第2级(L2)、第3级(L3)、第4级(L4)或其他级别的高速缓存、最后级别高速缓存(LLC)和/或其组合。虽然在一个实施例中,基于环的互连单元212互连集成图形逻辑208、共享高速缓存单元206的集合和系统代理单元210/(一个或多个)集成存储器控制器单元214,但是替代实施例可以使用任何数量的用于对此类单元进行互连的公知技术。在一个实施例中,在一个或多个高速缓存单元206和核202-A-N之间保持一致性(coherency)。

[0064] 在一些实施例中,一个或多个核202A-N能够进行多线程。系统代理210包括协调和操作核202A-N的那些部件。系统代理单元210可以包括例如功率控制单元(PCU)和显示单元。PCU可以是或包括调整核202A-N和集成图形逻辑208的功率状态所需的逻辑和部件。显示单元用于驱动一个或多个外部连接的显示器。

[0065] 在架构指令集方面,核202A-N可以是同构或异构的;也就是说,核202A-N中的两个或更多核可能能够执行相同的指令集,而其他核可能能够只执行该指令集的子集或者不同指令集。在一个实施例中,核202A-N是异构的并且包括下面描述的“小”核和“大”核两者。

[0066] 图3-图6是示例性计算机架构的框图。本领域已知的用于膝上型计算机、台式计算

机、手持PC、个人数字助理、工程工作站、服务器、网络设备、网络集线器、交换机、嵌入式处理器、数字信号处理器(DSP)、图形设备、视频游戏设备、机顶盒、微控制器、蜂窝电话、便携式媒体播放器、手持设备和各种其他电子设备的其他系统设计和配置也是合适的。通常,能够合并如本文所公开的处理器和/或其他执行逻辑的各种各样的系统或电子设备通常是合适的。

[0067] 现在参考图3,示出的是根据本发明的一个实施例的系统300的框图。系统300可以包括耦合到控制器集线器320的一个或多个处理器310、315。在一个实施例中,控制器集线器320包括图形存储器控制器集线器(GMCH)390和输入/输出集线器(IOH)350(其可以在单独的芯片上);GMCH 390包括存储器和图形控制器,存储器340和协处理器345耦合到GMCH 390上;IOH 350将输入/输出(I/O)设备360耦合到GMCH 390。可替代地,存储器和图形控制器之一或二者集成在处理器内(如本文所述),存储器340和协处理器345直接耦合到处理器310,并且利用IOH 350与在单个芯片中的控制器集线器320耦合。

[0068] 图3中用虚线标示附加处理器315的可选性质。每个处理器310、315可以包括本文描述的一个或多个处理核并且可以是某版本的处理器200。

[0069] 存储器340可以是例如动态随机存取存储器(DRAM)、相变存储器(PCM)或两者的组合。对于至少一个实施例,控制器集线器320经由诸如前端总线(FSB)的多点总线(multi-drop bus)、诸如快速通道互连(QPI)之类的点对点接口或类似连接395来与(一个或多个)处理器310、315通信。

[0070] 在一个实施例中,协处理器345是专用处理器,诸如例如高吞吐量MIC处理器、网络或通信处理器、压缩引擎、图形处理器、GPGPU、嵌入式处理器等。在一个实施例中,控制器集线器320可以包括集成图形加速器。

[0071] 在包括架构、微架构、热、功耗特性等在内的品质度量的范围方面在物理资源310、315之间可以存在各种差异。

[0072] 在一个实施例中,处理器310执行控制一般类型的数据处理操作的指令。嵌入在指令内的可以是协处理器指令。处理器310将这些协处理器指令识别为应该由附加的协处理器345执行的类型。因此,处理器310将协处理器总线或其他互连上的这些协处理器指令(或表示协处理器指令的控制信号)发布到协处理器345。(一个或多个)协处理器345接受并执行接收到的协处理器指令。

[0073] 现在参考图4,示出的是根据本发明实施例的第一个更具体的示例性系统400的框图。如图4中所示,多处理器系统400是点对点互连系统,并且包括经由点对点互连450耦合的第一处理器470和第二处理器480。处理器470和480中的每个可以是某版本的处理器200。在本发明的一个实施例中,处理器470和480分别是处理器310和315,而协处理器438是协处理器345。在另一个实施例中,处理器470和480分别是处理器310协处理器345。

[0074] 处理器470和480被示出分别包括集成存储器控制器(IMC)单元472和482。处理器470还包括作为其总线控制器单元的一部分的点对点(P-P)接口476和478;类似地,第二处理器480包括P-P接口486和488。处理器470、480可以使用P-P接口电路478、488经由点对点(P-P)接口450来交换信息。如图4中所示,IMC 472和482将处理器耦合到相应的存储器,即存储器432和存储器434,它们可以是本地附接到相应的处理器的主存储器的部分。

[0075] 处理器470、480中的每个可以使用点对点接口电路476、494、486、498经由单独P-P

接口452、454来与芯片组490交换信息。芯片组490可以可选地经由高性能接口439来与协处理器438交换信息。在一个实施例中,协处理器438是专用处理器,诸如例如高吞吐量MIC处理器、网络或通信处理器、压缩引擎、图形处理器、GPGPU、嵌入式处理器等。

[0076] 可以将共享高速缓存(未示出)包括在任一处理器中或在两个处理器外部、但是仍经由P-P互连与处理器连接,使得如果处理器被置于低功耗模式中则可以将一个或两个处理器的本地高速缓存信息存储在共享高速缓存中。

[0077] 芯片组490可以经由接口496耦合到第一总线416。在一个实施例中,第一总线416可以是外围部件互连(PCI)总线或诸如PCI Express总线或另一第三代I/O互连总线之类的总线,但是本发明的范围不限于此。

[0078] 如图4中所示,各种I/O设备414可以耦合到第一总线416连同总线桥418,总线桥418将第一总线416耦合到第二总线420。在一个实施例中,一个或多个附加处理器415,诸如协处理器、高吞吐量MIC处理器、GPGPU、加速器(诸如例如图形加速器或数字信号处理(DSP)单元)、现场可编程门阵列或任何其他处理器,耦合到第一总线416。在一个实施例中,第二总线420可以是低引脚数(LPC)总线。在一个实施例中,各种设备可以耦合到第二总线420,各种设备包括例如键盘和/或鼠标422、通信设备427和诸如盘驱动器或其他可以包括指令/代码和数据430的大容量存储设备的存储单元428。此外,音频I/O 424可以耦合到第二总线420。注意,其他架构是可能的。例如,代替图4的点对点架构,系统可以实现多点总线或其他此类架构。

[0079] 现在参考图5,示出的是根据本发明的实施例的第二个更具体的示例性系统500的框图。图4和图5中的类似元素具有类似的参考标号,并且已经从图5省略了图4的某些方面,以避免模糊图5的其他方面。

[0080] 图5图示了处理器470、480可以分别包括集成存储器和I/O控制逻辑(“CL”)472和482。因此,CL 472、482包括集成存储器控制器单元并且包括I/O控制逻辑。图5不仅图示了耦合到CL 472、482的存储器432、434,而且图示了I/O设备514也耦合到控制逻辑472、482。旧有I/O设备515耦合到芯片组490。

[0081] 现在参考图6,示出的是根据本发明的实施例的SoC 600的框图。图2中的类似元素具有类似的参考标号。此外,虚线框是在更先进的SoC上的可选功能。在图6中,(一个或多个)互连单元602耦合到:应用处理器610,其包括一个或多个核202A-N的集合和(一个或多个)共享高速缓存单元206;系统代理单元210;(一个或多个)总线控制器单元216;(一个或多个)集成存储器控制器单元214;可以包括集成图形逻辑、图像处理、音频处理器和视频处理器的集合或一个或多个协处理器620;静态随机存取存储器(SRAM)单元630;直接存储器存取(DMA)单元632;以及用于耦合到一个或多个外部显示器的显示单元640。在一个实施例中,(一个或多个)协处理器620包括专用处理器,诸如例如网络或通信处理器、压缩引擎、GPGPU、高吞吐量MIC处理器、嵌入式处理器等。

[0082] 本文公开的机制的实施例可以以硬件、软件、固件或此类实现方法的组合来实现。本发明的实施例可以被实现为在包括至少一个处理器、存储系统(包括易失性和非易失性存储器和/或存储元件)、至少一个输入设备和至少一个输出设备的可编程系统上执行的计算机程序或程序代码。

[0083] 可以将诸如图4中所图示的代码430之类的程序代码应用于输入指令以执行本文

所述的功能并生成输出信息。输出信息可以以已知的方式应用于一个或多个输出设备。为了本申请的目的,处理系统包括具有诸如例如数字信号处理器(DSP)、微控制器、专用集成电路(ASIC)或微处理器之类的处理器的任何系统。

[0084] 程序代码可以以高级程序或面向对象编程语言来实现,以与处理系统通信。如果需要,程序代码也可以以汇编或机器语言来实现。事实上,本文所述的机制在范围上不限于任何特定的编程语言。在任何情况下,语言可能是经编译或解释的语言。

[0085] 至少一个实施例的一个或多个方面可以通过存储在表示处理器内的各种逻辑的机器可读介质上的代表性指令来实现,当指令被机器读取时使得机器制造用于执行本文所描述的技术的逻辑。称为“IP核”的这种表示可以存储在有形的机器可读介质上,并提供给各种客户或制造工厂以加载到实际制造逻辑或处理器的制造机器中。

[0086] 这样的机器可读存储介质可以包括但不限于由机器或设备制造或形成的制品的非临时的、有形的布置,包括诸如硬盘、包括软盘、光盘、压缩盘只读存储器(CD-ROM)、压缩盘可重写(CD-RW)和磁光盘的任何其他类型的盘、半导体器件,诸如只读存储器(ROM)、诸如动态随机存取存储器(DRAM)、静态随机存取存储器(SRAM)之类的随机存取存储器(RAM)、可擦除可编程只读存储器(EPROM)、闪存、电可擦除可编程只读存储器(EEPROM)、相变存储器(PCM),磁卡或光卡,或适于存储电子指令的任何其他类型的介质在内的存储介质。

[0087] 因此,本发明的实施例还包括非临时的、有形的机器可读介质,其包含指令或包含设计数据,诸如定义本文所描述的结构、电路、装置、处理器和/或系统特征的硬件描述语言(HDL)。这样的实施例也可以称为程序产品。

[0088] 在一些情况下,可以使用指令转换器将来自源指令集的指令转换为目标指令集。例如,指令转换器可以将指令转译(例如,使用静态二进制转译,包括动态编译的动态二进制转译)转变(morph)、仿真或以否则转换为要由核处理的一个或多个其他指令。指令转换器可以以软件、硬件、固件或其组合来实现。指令转换器可以在处理器上、在处理器外、或部分在处理器上并且部分在处理器外。

[0089] 图7是根据本发明的实施例的对比使用软件指令转换器将源指令集中的二进制指令转换为目标指令集中的二进制指令的框图。在图示的实施例中,指令转换器是软件指令转换器,但是可替代地,指令转换器可以以软件、固件、硬件或其各种组合来实现。图7示出了可以使用x86编译器704编译高级语言702的程序,以生成可由具有至少一个x86指令集核的处理器716本地执行的x86二进制代码706。具有至少一个x86指令集核的处理器716表示任何如下处理器:所述任何处理器可以通过兼容地执行或以否则处理(1) Intel x86指令集核的指令集的大部分或者(2)目标是在具有至少一个x86指令集内核的Intel处理器上运行的应用或其他软件的目标代码版本以便实现与具有至少一个x86指令集核的Intel处理器基本相同的结果,来执行与具有至少一个x86指令集核的Intel处理器基本上相同的功能。x86编译器704表示可操作以生成可以在具有或没有附加联动(linkage)处理的情况下在具有至少一个x86指令集核的处理器716上执行的x86二进制代码706(例如,目标代码)的编译器。类似地,图7示出了可以使用替代的指令集编译器708来编译高级语言702的程序,以生成可由没有至少一个x86指令集核的处理器714(例如,具有执行加利福尼亚州桑尼维尔的MIPS Technologies的MIPS指令集和/或执行加利福尼亚州桑尼维尔的ARM Holdings的ARM指令集的核的处理器)在本地执行的替代指令集二进制代码710。指令转换器712用于

将x86二进制代码706转换为可由没有x86指令集核的处理器714在本地执行的代码。该转换后的代码不太可能与替代指令集二进制代码710相同,因为能够实现此的指令转换器很难制造;然而,转换后的代码将完成一般操作,并由来自替代指令集的指令组成。因此,指令转换器712表示通过仿真、模拟或任何其他过程来允许不具有x86指令集处理器或核的处理器或其他电子设备执行x86二进制代码706的软件、固件、硬件或其组合。

[0090] 用于人工神经网络中的分布式与协作计算的方法与装置

[0091] 人工神经网络(ANN)可以被设计为包含大量“神经元”的完全连接的层的集合。每个神经元通过“突触”与属于相邻层的所有神经元连接。跨越突触的两个神经元之间的连接的强度或幅度被称为“突触权重”(或者仅称为“权重”),其可以被表示为数值。因此,神经元的输出可以通过由它们的突触权重衡量(ponder)的来自前一层的所有输入神经元的相加,一种被称为点积的运算,来计算。图8图示了具有多层(层1、层i-1、层1、层N)的完全连接的ANN的方案,并示出了可以如何使用点积运算来组合多个权重 w_{0j} - $w_{(m-1)j}$ 。

[0092] 大脑启发的(brain-inspired)算法这样的ANN可能需要非常高的计算需求,这在传统低功率设备中由于它们有限的功率预算和处理能力而可能是禁止的。为了克服这个问题,工业和学术界两者最近都对引入专门的神经形态架构表现出很大的兴趣,所述专门的神经形态架构提供了好于常规方法数量级的能量效率。

[0093] 不幸的是,所提出的神经形态架构需要巨大的存储器带宽,这在低功率片上系统(SoC)架构中是不可用的。例如,已经表明,由简单CNN系统中的完全连接层所需要的带宽为120 GB/s,而SoC中的可用带宽要小得多。即使在使用平铺(tiling)技术和内部存储器的情况下,带宽需求仍保持是价格昂贵的。

[0094] 图9示出了神经形态加速器架构900,其中单个处理单元(“PU”)901掌管针对每个逻辑神经元计算点积运算,并累积部分结果直到已经遍历所有输入神经元并且结果是最终的。使用点对点总线将输入和权重从输入/输出(I/O)接口902带到PU 901,所述点对点总线将来自该单元的每个元件与内部存储器或外部世界连接。

[0095] 诸如图9中所示的当前设计的主要问题是输入和权重在PU 901内部不重用。由于每个逻辑神经元都有它自己的权重,所以必须每个周期取得新的数据,从而在PU 901和I/O接口902之间添加显著的流量。此外,由于在硬件中仅映射单个逻辑神经元,所以针对包含在给定的完全连接层中的每个神经元必须再次访问输入,从而失去重用输入的机会。此外,由于必须在每次计算时递送输入和权重二者,所以从I/O接口902到PU的总线太宽,使得设计低效。更好地重用输入数据并减少针对每次计算而传输的数据量将大大改进架构的能量效率。

[0096] 下面描述的本发明的实施例包括非常有效地计算完全连接的神经网络的优化架构。该架构的一个实施例由分布式处理单元(PU)的集合组成,所述分布式处理单元(PU)协作工作以最小化带宽需求并减少外部读取和写入的数量。实现具体机制的集合以(i)在PU内部重用尽可能多的数据、(ii)在PU之间共享数据以避免广播数据以及(iii)提前请求数据以避免空闲的PU周期。标量和模块化方法允许系统在从高性能设备变化到低功率设备的许多市场划分中极有效率。

[0097] ANN可以在常规架构(例如高端处理器)或专门的神经形态加速器中执行。已经表明,与前者相比,后者提供了数量级的改进的能量效率,并且因此更适合于功率受限的那些

划分,如SoC或可穿戴设备。然而,目前的神经形态架构呈现出具有巨大带宽需求的单片结构,这在上述领域中是不可用的。该障碍阻碍了那些加速器在ANN的认知能力将是特别有用的(例如,移动设备)市场划分中的实现。

[0098] 下面描述的本发明的实施例使得能够将神经形态加速器集成在低功率设备中,通过在完全连接的操作上最大化数据重用减少数据流量,并且减少连接PU与I/O接口的总线的宽度。此设计允许系统节省面积(因为需要较少的连接)和功率(因为访问外部存储器次数较少)。此外,外部带宽显著降低。这些性质允许本发明的实施例满足当前SoC技术中的物理约束。

[0099] 图10示出了针对单片配置和利用分布式配置的本发明的实施例二者的带宽需求。Y轴示出了带宽(GB/s),而X轴示出了给定ANN层中的输出神经元的数量(输入神经元数量是固定数量)。单片配置引入了可以在1 GHz(全速模式)或0.25 GHz(带宽节省模式)工作并包含64个8位乘积累加(multiply-accumulate) (MAC)单元的单个PU。分布式配置的一个实施例以0.25 GHz操作、支持两种配置(1对1和1对N)并包含十六个每个具有4个8位乘法器(总共为64个8位MAC单元,与单片配置相同的资源)的PU。

[0100] 图10示出了单片配置在大型ANN中需要高达128 GB/s(1 GHz),并且以它们频率的四分之一操作时需要32 GB/s。无论ANN层的输出神经元的数量如何,这都是固定数量。如果在不同部件之间共享芯片连接,则低功率划分不能支持这种带宽。通过支持1对1和1对N两种配置,分布式架构需要少于14 GB/s(与输出神经元数量无关的最大带宽需求),相比于在以相同频率操作的单片架构中这减少至1/2.3并且相比于全速操作的单片架构所需的带宽这减少至约1/9.8。在一个实施例中,这是因为将计算分布在不同PU之间更好地允许了数据重用和对外部存储器的更少的访问而实现了这一点。注意,1对1配置更好地适于具有几个输出神经元的ANN,而1对N配置在计算具有大量输出神经元的大型ANN时工作更好。本文中呈现的分布式架构支持两种执行模式。

[0101] 而且,单片架构比分布式架构需要更多的与外部元件(例如内部或外部存储器)的连接。遵循图10中所示的配置示例,单片配置需要128字节宽的总线,以用于将每个周期64B数据和64B权重递送到单个PU,而本文中所呈现的分布式解决方案仅需要在PU之间共享的20字节宽的总线来递送4B数据和16B权重。这导致电路中的75%的面积节省和在互连功耗中的类似减少。

[0102] 图11图示了单片和分布式神经形态架构之间的比较,其中Y轴示出了能量消耗(以mJ为单位)并且X轴示出了输出神经元的数量。查看附图,可以看出在实现分布式架构时获得的关于能量消耗的益处,这被转化为多于2倍的能耗改进。这是因为访问主存储器是功耗巨大(power-hungry)的操作,其可以在多个小型PU协作工作时得到缓解。

[0103] 图12A-图12B图示了可以在其上实现本发明的实施例的不同系统架构。两个示例性架构都包括处理器1200,处理器1200包括多个核1201-1203,N和包括用于互连核1201-1203,N的通信总线结构或构造的“非核(uncore)”部件1210。非核部件1210还包括存储器控制器1211,其将核1201-1203,N中的每个通信地耦合到系统存储器1220(例如,动态随机存取存储器(DRAM)或其他类型的存储设备)。

[0104] 在图12A中所示的实施例中,用于实现本文所述的本发明的实施例的神经形态加速器1230被配置为外部加速器,其经由总线或通信构造而耦合到处理器。在图12B中所示的

替代实施例中,神经形态加速器1231集成在处理器1200内。在任一情况下,经由集成的存储器控制器1211向神经形态加速器1230提供对系统存储器1220的访问。虽然为了说明起见在图12A-图12B中示出了这两个特定架构,但是应当注意,本发明的基本原理不限于任何特定的处理器架构。

[0105] 如图13中所示,神经形态加速器1230的一个实施例包括通过共享数据和部分输入总线(在图中标记为1)和共享结果输出总线(标记为2)而连接到IO接口1310的PU 1300-1303的集合。第一总线(1)用于将数据、内核和权重带到PU 1300-1303,而第二(2)用于递送部分或最终结果。使用单向点对点(P2P)链路(标记为3)互连PU 1300-1303,单向点对点(P2P)链路(标记为3)用于将部分计算从一个PU发送到另一个。全局控制单元1305掌管将数据从IO接口1310驱动到PU 1300-1303,以及配置每个PU以在任何时间点执行适当的完全连接配置。

[0106] 在一个实施例中,为了优化功率/性能,PU 1300-1303可以以比IO接口1310低的频率操作。本发明的基本原理不限于共享总线的任何数量的PU 1300-1303和/或每个PU中包含的任何特定资源。在一个实施例中,每个PU 1300-1303包含十六个8位乘法器,并且该架构包括协作工作的四个分布式PU 1300-1303。可以采用各种总线宽度,但是在特定实施例中,共享数据输入总线(在图中标记为1)的宽度为20B。

[0107] 图13中所图示的协作方法具有超过图9中所示的单片神经形态架构的许多益处。首先,单片架构在单个PU中包含所有处理元件(MAC单元和特定电路),而本发明的实施例将计算分布在多个PU 1300-1303之间,在PU之间需要内部锁存器和特定通信信道以交换部分结果。第二,单片架构需要PU和IO接口之间的更宽的总线来每个周期发送/接收尽可能多的数据,而本发明的实施例在PU之间共享总线。第三,本发明的实施例引入了一些内部锁存器,其用于重用数据输入并存储未来计算所需的其他数据。最后,在一个实施例中,当执行完全连接的层以满足带宽约束时PU 1300-1303以较低的频率操作(尽管其他层可以全速操作)。假设从相同总线挂起四个PU 1300-1303,则每个PU可以以总线频率的四分之一(1 PU 周期 = 4个总线周期)操作。注意,在一个实施例中,PU 1300-1303始终保持运行,尽管每秒发送较少的数据。

[0108] 如所提及的那样,本发明的一个实施例支持两种执行模式:(1)完全连接的1对1和(2)完全连接的1对N操作。如上所解释的那样,取决于目标ANN配置,以一种模式或以另一模式来执行可能是更适合的。执行模式之间的主要差异在于处理输入和权重的方式。虽然第一模式需要输入和权重同时到达给定的PU 1300-1303,但是第二模式首先加载神经元的输入并且然后通过仅在每个周期取得权重来遍历输出神经元。图14A-图14B图示了根据本发明的一个实施例的完全连接的1对1操作可以如何映射并且图15A-图15B图示了根据本发明的另一实施例的完全连接的1对N操作可以如何映射。

[0109] 首先转到图14A-图14B,完全连接的1对1操作将一个逻辑神经元映射到在单个PU 1300-1303中计算的一个物理神经元。为了执行点积运算,发送输入神经元和突触(权重)1400-1403二者。由于每个神经元需要它自己的权重,所以必须针对每个操作取得输入神经元和权重二者。从总线挂起的每个实体在不同的时间接收打包在一起的它们的权重(在一个实施例中使用16B来进行编码)。然而,输入神经元在PU 1300-1303之间共享,并且因此可以广播它们以避免从IO接口1310多次读取,从而每PU周期仅需要4B。因此,用于图示实施例

的总带宽需求一次是20B,如图14B中所示。因此,在图14B中,在四个PU周期(1-4)之后,所有PU 1300-1303都具有计算所需的输入和权重两者(在图示实施例中总共为128B),因此它们可以执行点积运算,并且如果神经元具有许多输入,则将结果与以前的计算进行累积。在一个实施例中,只有当逻辑神经元的最后输入被处理时,PU 1300-1303才将产出结果发回到IO接口1310。

[0110] 现在转到图15A-图15B中所图示的完全连接的1对N操作,当以许多输入神经元操作时,PU 1300-1303可以被配置为映射多个物理神经元中的一个逻辑神经元并且因此更有效率。在这种方法中,每个PU 1300-1303仅计算逻辑神经元的子集,并将部分结果传递到邻近(contiguous)的PU,其累积它自己的计算与先前的结果。因此,神经元的最终产出仅在最后的PU中可用。

[0111] 注意,PU 1300-1303可以每个周期计算不同的逻辑神经元。也就是说,它们需要新的权重,但是它们不从输入神经元再次取得值,这些值保留在本地锁存器中以增加数据重用。图15B示出了使用1对N配置进行操作时如何分布输入和权重。像在1对1情况中那样,PU 1300-1303也可以以其最大频率的四分之一来操作。尽管如此,每个PU周期都不广播输入;它们仅在改变输入神经元时被发送(例如当需要计算新的层时)。为了避免空闲周期,IO接口1310可以预先发送针对接下来要计算的逻辑神经元的输入,并且这些值被内部锁存在PU中,直到它们被实际使用。如图15B中所示,在单独的PU周期中发送那些输入,并且与权重一起,仅由单个PU消耗。因此,在图示实施例中,20B宽的总线足以维持整个计算。

[0112] 尽管为了解释的目的而在上面提供了若干具体架构示例,但是本发明的基本原理可以在包括移动设备(例如,智能电话)、配备有照相机或麦克风的平板计算机或可穿戴设备的各种不同架构中实现。事实上,这些实施例可以在任何形式的设备上实现,以减少机器学习算法的带宽需求并且改进像人工神经网络(例如卷积神经网络或深度信念神经网络)的新型计算机范例上的能量效率。

[0113] 用于执行卷积运算的存储设备和方法

[0114] 图像处理在片上系统(SoC)架构的设计中越来越重要。移动设备中的图像信号处理器(ISP)负责处理越来越大的图像并且被期望保持或减少其功率预算。卷积神经网络(CNN),由于其在新型人机接口中的应用而获得兴趣的生物学启发算法,也广泛利用2D卷积。目前,最好的对象识别算法利用卷积神经网络并且已经实现了显著优于以前顶级(top-performing)算法的识别率。

[0115] 因为卷积运算依赖于数据的高度重用,所以它们大大受益于具有专用的记忆装置。图16示出了执行变化的滤波器大小和多个存储器大小的卷积所需的能量。可以看出,所需的能量可以呈数量级地减少,特别是对于所需数据在处理单元中不适合的大型滤波器大小。因此,如果存在足够的面积预算,则希望在加速器中使用专门的记忆装置以便增加总体能量效率。

[0116] 用于这些种类的应用的若干加速器在工业和学术界中已经提出并且大多数依赖于用于如图17A中所图示的不同类型的数据和处理单元的专用存储区域1701-1706。然而,如下所讨论的,对于固定量的存储,最佳存储器映射取决于问题类型和大小而改变。因此,具有静态存储器分区导致次优存储器使用。

[0117] 对于这个问题的直接解决方案是具有有专用于不同类型数据的多个读/写端口的

统一存储区域1710,如图17B中所示。然而,由存储器组块所消耗的面积和能量随着端口数量而显著增加。仅通过添加一个额外的端口,阵列面积可以变为双倍并且动态和泄漏能量两者可以增加约35%。此外,如果针对两种类型的数据需要最低服务质量,则应将专用互连或虚拟通道分配给每种类型的数据。

[0118] 因此,现有加速器的存储器受到两个冲突目标的限制。专用存储器1701-1706是简单且能量高效的,但没有提供存储的高效的、可适配的使用。此外,共享存储器1710可能适于问题需求但需要更复杂和较低效的存储器阵列和互连。

[0119] 下面描述的本发明的实施例包括用于硬件卷积器和神经网络加速器的暂存器存储器设计。这些实施例能够使用简单的一个端口存储器阵列和共享互连,并且还能够基于问题/应用来适配存储器分配。

[0120] 图18A图示了存储器组织的一个实施例,其中在执行集群1800(例如,包括多个处理单元(PU))内执行的不同类型的数据(例如,输入图像和部分结果)之间共享存储器组块1801-1806和互连1811-1816。由于访问模式对于特定问题是公知的,所以在存储器阵列中不需要标签,从而节省了大量的面积和能量。可以取决于问题需求改变专用于每种类型数据的存储百分比,因此优化其使用。

[0121] 利用卷积运算的规则访问模式,可以实现在两种类型的数据之间的独立性。图18B示出了将在其上应用滤波器的图像1820。在卷积的给定周期中,逻辑单元必须具有卷积窗口1821的数据,其基于要同时处理的滤波器的实例的数量和滤波器大小。这个卷积窗口1821基于步幅1823并且在卷积行1822内在每个卷积运算上向右移动,直到其到达图像1820的边缘。当到达图像的边缘时,卷积窗口1821向下移动并处理下一个卷积行。重复此过程,直到达到图像的末尾。在卷积器的稳态操作上,重用卷积窗口1821的大部分并且只有新的部分需要从存储器加载。每个周期需要加载的数据取决于问题并由卷积窗口1821高度以及由步幅1823确定。

[0122] 为了实现两种类型的数据的期望的服务质量,本发明的一个实施例使用确保在两种数据类型(通常为输入数据和部分结果)之间的完全独立性的映射。图19示出了根据本发明的一个实施例的包括多个组块1-6的暂存器存储器1900的操作原理。这个映射的一个思想是以如下这样的方式分配数据,即,在每个周期中访问不同的组块1-6。示出了从T=1到T=4的四个周期。在图示的示例中,在T=0访问组块2;在T=1访问组块2和3;在T=2访问组块2-4;并且在T=3访问组块3-5。一种技术映射图像/矩阵的每行,使得它针对每个相邻操作在不同的组块中开始。这样,如果需要在稳态中读取N个元素的列,则可以在一个周期内读取全部,因为它们在不同的组块中。该映射也确保了每个周期我们将访问N个组块并且这些组块将针对每个访问而改变。

[0123] 由于以规则和均匀的方式访问组块,所以需要存储的任何部分结果将在固定和已知的最大时间之前进行。因此,在一个实施例中,将部分结果缓冲在小的输入缓冲器中,直到它们可以访问它们的目的地地组块。部分结果可能停顿(stall)一些周期的事实不是问题,因为它们不在关键路径中。

[0124] 因此,本发明的一个实施例包括用于卷积加速器中的两种类型的数据,输入数据和部分结果,的统一暂存器存储器1900。在这个暂存器存储器中,所有组块都被分区成两个区域(输入数据和部分结果),并且可以取决于问题/应用而改变专用于每个数据类型的量。

共享可用存储容量允许针对所有问题大小的最优使用,这导致每操作的较低带宽需求和较低能量。

[0125] 本发明的实施例还包括一种映射技术,即使在使用仅具有一个读/写端口和共享互连的存储器组块时,该映射技术也确保了两种类型的数据的最低服务质量。允许仅具有一个端口的存储器组块的使用减少了暂存器存储器1900所需的面积和能量。

[0126] 该统一设计的一个优点是它实现了暂存器存储器1900的可用容量的最佳利用,并且最重要的是,不需要通常需要更多面积并消耗更多功率的多端口存储器组块或附加阵列总线。此外,更好的暂存器存储器利用导致显著的外部存储器带宽降低,以及因此更低的功率和能量消耗。

[0127] 在图20A中,上部阴影区域示出了与卷积滤波器的数量相关的留着未被使用的典型暂存器存储器的部分。更清楚的是,具有相等大小(即50:50)的用于输入数据和部分结果的专用暂存器存储器的现有设计不能有效地利用总可用存储器区域。事实上,如图20A中所示,对于在面部检测CNN的卷积层1和2中使用的配置参数,静态分区仅分别利用了可用暂存器容量的65%和58%。平均而言,50:50的分割暂存器配置使30%的暂存器存储器未被使用。

[0128] 相比之下,可以以最佳方式对图20B中示出了其结果的统一暂存器1900动态地进行划分,因此针对任何给定卷积问题配置而实际上利用暂存器存储器的整个容量。注意,静态暂存器组织未充分利用可用容量,因为输入数据和部分结果之间的比取决于滤波器的数量和大小以及输入的数量和维度而显著变化。通过比较针对不同数量的滤波器(x轴)的底部区(输入数据)和中间区(部分),可以容易地在图20A中观察到这一点。

[0129] 有效利用是重要的,根据经验法则(a rule of thumb),越高的暂存器利用导致对外部存储器的越少访问。对于此的原因是:通常,可用内部存储器不足以将完全的问题存储在本地存储器内(例如,全HD图像的1个通道为~2MB)。因此,必须将输入打破成适当大小的分区,以便将输入数据和部分结果二者适配在暂存器存储器中。然而,分区导致从外部存储器取得输入数据的一部分不止一次。因此,分区越多,外部存储器带宽开销越大。图21图示了使用提出的统一暂存器设计实现的对外部存储器的访问的减少。对于给定的总暂存器存储器大小的益处可能取决于通道和滤波器的数量而变化。例如,对于图20A-图20B和图21中使用的面部检测CNN的示例,对于卷积层1和2的配置,外部存储器访问的减少分别从3.1%变动至11.3%。重要的是要注意,随着我们移向更高清晰度的标准,分区的影响变得甚至更高。例如,对于4K UHD输入图像,外部存储器访问的减少针对层1和2而言分别增加至的5.6%和17.7%。

[0130] 更清楚的是,统一设计的灵活性是减少外部存储器访问的能量的关键。事实上,外部存储器访问支配着CNN计算的整个能量成本,其虑及针对全套(a wide set of)CNN配置的总能量的95%以上。这些发现进一步强调了减少冗余的外部存储器访问的技术的重要性。

[0131] 最重要的是,本发明的实施例在不需要通常需要更多面积并消耗更多功率的多端口存储器组块或附加暂存器阵列总线的情况下提供了外部存储器带宽的该减少。代之以,下面详述的机制使能使用只具有一个读/写端口的存储器组块,类似于用于通用处理器核中的中间级别高速缓存(MLC)的高密度存储器块。此外,可以使用相同的共享阵列总线来服

务针对多个存储器块的读取和写入请求两者,这避免专用总线的区域的过度增加。然而,使用更简单的存储器阵列需要处理共享总线中的读/写冲突和专门的数据映射,以保证针对输入数据和部分结果二者所需的服务质量。然而,如下所示,可以以直接的方式解决两个问题而且没有显著的功率和面积成本。

[0132] 本发明的基本原理可以以多种方式实现,但是一个特定实施例中在图22中示出,其已经针对NeuraTM加速器,一种用于可穿戴和移动设备的超低功率卷积神经网络加速器进行了开发。在这种情况下,使用通过32个总线SAB0-SAB31连接的具有1个读/写端口的 4×32 个存储器组块2201。每个总线有4个组块(A0-D0,A1-D1等)允许写和读以最大加速器频率的 $1/4$ 操作。图22中所示的四个附加结构是输入多路复用器(MUXI) 2202、输出多路复用器(MUXO) 2203、部分写回缓冲器(PWBB) 2204和控制单元2205。

[0133] 在一个实施例中,MUXI 2202是多路复用器的集合,其对准出自加速器外部的数据并且用于在第一次访问数据时填充暂存器存储器组块2201。MUXO 2203是多路复用器的另一集合,其对准出自内部总线并经由PU接口2206被发送到执行集群1800的处理单元(PU)的数据。

[0134] 在一个实施例中,PWBB 2204是负责在目的地组块忙时保持由执行集群1800的PU所提供的部分结果的缓冲器集合。由于最坏情况模式是已知的,所以可以对这些缓冲器的设置维度(dimension)以存储在最坏情况的场景下可以生成的部分的最大数量。PWBB 2204还处理部分写入,并且当不同的总线未被用于读取数据时请求控制单元2205向它们写入。

[0135] 最后,控制单元2205负责控制存储器2201。特别地,控制单元2205的一个实施例向多路复用器2202-2203指示在每个周期哪些总线SAB0-SAB31活跃并且向存储器组块2201指示何时开始读或写操作以及在哪些线(line)上开始。控制单元2205处理的主要操作是读数据、写数据、读部分和旁路数据(例如,当数据来自加速器外部时使用)。它还准许对PWBB 2204在空闲总线和组块中写入部分的许可。

[0136] 在一个实施例中,针对每个组块2201所执行的数据映射将前N行用于输入数据(例如,输入图像),并且其余的用于部分。部分结果可以以任何同质(homogeneous)的方式,通常按到达顺序,进行存储。存储输入数据,使得在每个周期中改变被访问的组块。因为在这个实施例中每个总线有4个组块(Ax、Bx、Cx和Dx),所以可以对图像进行划分,使得Ax组块存储偶数行和列元素,Bx组块存储奇数行和偶数列元素,Cx组块存储偶数行和奇数列元素并且Dx组块存储奇数行和列元素。

[0137] 图23示出了在卷积器中使用的输入数据(图像)的示例,以及图24示出了这些元素可以如何在物理存储器组块中映射。在图中可以看到行2的映射(即(2,0)、(2,2)、(2,4))。由于它是偶数行,所以将它存储在组块A和组块B中,如所图示的那样。可以看出,该行在第二A组块中开始(即,A1中示出(2,0)),并且随后的行在右侧的不同组块上开始。

[0138] 被访问的行的数量根据待解决的问题(例如,基于滤波器大小、同时计算的滤波器的数量等)而改变。因此,取决于问题,需要同时访问不同数量的组块。例如,如果卷积行由6个图像行组成,则系统将访问(A0、A1、A2-B0、B1、B2-C0、C1、C2-D0、D1、D2-A1、A2、A3……)。本文描述的存储器组织支持不同的卷积行大小,其仅需要来自每个组块的一个元素。因此,该特定实施例支持在两个存储器周期中访问卷积所需的所有新数据。可以取决于问题需求和功率/面积限制,通过添加更多专用总线来选择所需的周期数量。

[0139] 上述实施例的一个有益方面是数据不需要识别单独的标签。由于访问模式是高度规则的,所以控制单元2205能够跟踪不同组块的索引而不需要标签阵列。不使用标签在面积、功率和延迟方面具有显著优势,并且提供了比传统存储结构显著改进的能量效率。

[0140] 用于低功率加速器的细粒度存储接口和方法

[0141] 图像处理算法由于其在新颖人机接口中的多种应用而获得兴趣,这使得更好的用户体验成为可能。图像处理中最重要的运算之一是卷积。除其他应用之外,卷积广泛用于将滤波器应用于在图像信号处理器 (ISP) 中的图像以及用于卷积神经网络 (CNN) 中的图像识别。

[0142] 卷积运算将数字的两个阵列相乘在一起,数字的两个阵列通常具有不同的大小但具有相同的维度,以产生第三输出阵列。在图像处理中,输入阵列之一是输入图像。第二阵列被称为内核,通常比输入图像小得多。通过在图像上滑动内核,通常从左上角开始,来执行卷积运算。每个内核应用生成通过将内核的值与基础子图像值相乘所计算出的输出像素,并将所有结果相加在一起。在数学上,卷积可以被描述为:

$$[0143] \quad O(x,y) = \sum_{k=1}^m \sum_{l=1}^n I(x+k-1,y+l-1)K(k,l)$$

[0144] 其中 I 是输入图像、 K 是内核并且 $O(x,y)$ 表示输出图像的坐标 x 、 y 中的像素。常数 m 和 n 分别是内核宽度和高度。

[0145] 图像处理算法具有非常高的计算需求,这对于传统的低功率划分而言由于其有限的功率预算和处理能力而可能是禁止的。为了克服这个问题,许多团体最近都在致力于开发被称为“加速器”的专门架构,它提供了好于常规方法数量级的能量效率。这些架构通常具有用于执行大量并行的乘法和加法的若干处理单元 (PU)。图25示出了用于图像处理的通用加速器的高级框图。PU 0-15形成执行集群2500。以同步的方式收集和处理来自PU的输出,以便产生预期的结果。经由I/O接口2503将数据从存储器2501传送到执行集群2500或者经由I/O接口2503 (例如使用点对点总线) 将数据从外部接口2502传送到外部世界。PU通常分组成为通过互连网络与存储器2501和外部接口2502通信的集群。

[0146] 然而,这些架构需要巨大的存储器带宽以在执行卷积时想所有处理单元进行馈送,这减少了系统的整体能量效率并且需要复杂的存储器互连的实现——对于低功率片上系统 (SOC) 设计而言是禁止的需求。

[0147] 目前的设计提出了传统的数据高速缓存组织来减少带宽需求。在这些设计中,将高速缓存结构放置在I/O接口2503和执行集群2500之间。然而,这些解决方案并没有完全利用卷积运算的特性,这导致非最佳的结果。在这些方法中,每个处理单元或处理单元的子集单独地请求数据,这需要高数量的高速缓存读取端口。事实上,在类似于图25中所呈现的加速器的加速器中执行16x16步幅1卷积时,为了提供足够的数据需要多达18个读取端口。鉴于存储器组块消耗的的面积和能量随着端口数量的增加而显著增加 (仅仅添加一个额外的端口,则阵列面积可以加倍,并且动态和泄漏能量两者都可以增加约36%),端口数量对于实现加速器而言是价格昂贵的。此外,当增加处理单元的数量以更快地执行卷积时,这个问题恶化。幸运的是,卷积的访问模式是非常可预测的,并且它们展现出高的空间和时间数据局部性 (locality),这可以被利用来减少存储器访问的数量和所需的读取端口数量。

[0148] 本发明的实施例包括细粒度存储器接口,其允许在具有约束带宽、面积和功率需求的神经网络加速器和图像处理器中非常有效地执行卷积。所呈现的实施例利用新颖的存储组织和数据混洗器(shuffler)机制,它们协作工作以向执行集群提供图像数据,从而将对较高的高速缓存级的读取数量最小化。该方案利用卷积运算的规则访问模式,并且使能将图像作为一系列像素列、像素行或两者的组合进行处理,这对于向处理单元有效地提供数据是非常方便的。此外,本发明的实施例是模块化和可扩展的,涵盖了从小滤波器(2x2像素)变动到大滤波器(64x64像素)的应用的多种多样的卷积问题。

[0149] 所提出的实施例使能将以卷积运算为目标的神经网络加速器和图像处理集成在低功率和带宽受限设备中。这些实施例的目的是通过提供足够的图像数据来对所有处理单元进行馈送、同时将存储器流量保持在合理限制内来非常有效地处理卷积运算。这在一个实施例中通过以最方便的方式处理图像来执行卷积运算而实现。

[0150] 卷积具有非常高的空间和时间数据局部性。内核的连续应用共享大量的输入图像数据。图26示出了图像2600,其中4x4像素2601的内核以1像素的步幅从左上应用到右下角。当卷积运算开始时,将内核应用在4x4像素2601的正方形图像区上。一旦计算出第一输出像素,则内核向右移位,如图像中的虚线区2602所指示的。由于已经将其他像素提供给先前的内核操作,所以只需要请求表示四行的列的仅仅四个额外像素就可执行该新的操作。此外,当内核如图像中的交替虚线区2603所指示的向下移位时,由于先前已经提供了其他像素,所以仅需要四个像素的单行。

[0151] 传统的方法将图像数据存储为连续的像素行或列,当将数据存储为行时,需要多个存储器访问以生成列,或者当将数据存储为列时,需要多个存储器访问以生成行。这些额外的访问显著增加了存储器读取端口的数量,并且因此增加了加速器和带宽的总功耗。当在加速器中同时执行相同内核的多个实例时,这恶化了。本发明的实施例内部地组织图像以利用数据局部性并提供细粒度访问以便为每个内核应用精确地提供所需的像素,从而显著减少所需端口的数量。

[0152] 图27图示了在与图25中所呈现的加速器类似的加速器中执行不同类型的卷积所需的读取端口的数量。卷积配置(内核大小和步幅)示出在x轴上,而端口的数量示出在y轴上。在使用本发明的实施例和不使用本发明的实施例的情况下(高速缓存+数据混洗器对比传统高速缓存)对实现传统高速缓存结构的加速器进行比较。如可以观察到的,传统设计中所需的读取端口的数量从在步幅2的4x4像素的卷积的情况下的4变动到在步幅1的16x16像素的卷积的情况下的18。对比而言,使用本发明的实施例,读取端口的数量总是等于1,而与卷积配置无关。通过利用卷积中观察到的规则模式并且高度利用在下面详细描述的数据局部性来实现高速缓存读取端口的数量的这种急剧减少。

[0153] 如所提及的那样,本发明的实施例使能通过显著减少存储器读取端口的数量来非常有效地在具有约束带宽、面积和功率需求的图像处理器和神经网络加速器中执行卷积。这些性质通过以下来实现:

[0154] 1. 基于用于卷积的模式的数据组织和编索引。以利用空间和时间局部性的方式组织数据。可以遍历像素行、像素列或两者的组合来处理图像。对外部存储器和/或较高的高速缓存级的访问次数低于传统方法中的次数,这显著减少了加速器的功耗和带宽。

[0155] 2. 无标签(tag-less)的存储组织,这将加速器面积保持在合理的限制内。

[0156] 3. 分布式数据组织, 这简化了较高的高速缓存级和处理单元之间的连接。

[0157] 上面提及的性质允许本文所述的本发明的实施例满足当前SoC技术的物理约束。

[0158] 具有4个集群的加速器的一个实施例的高级框图在图28中示出。本发明的这个实施例包括四个部件: (1) 输入和分布逻辑2850、(2) 容器2800-2803、(3) 对准逻辑2810-2813和(4) 选择逻辑2820-2823。容器、对准逻辑和选择逻辑部件可以划分成组, 每个连接到处理单元的不同的集群2830。对照而言, 在一个实施例中, 输入和分布逻辑2850由所有集群共享。如所图示的, 四个部件顺序连接。输入和分布逻辑2850直接连接到容器2800-2803, 容器与对准逻辑2810-2813连接, 并且对准逻辑与选择逻辑2820-2823连接。在一个实施例中, 所有部件以同步方式工作, 并且它们进行协作以同时为所有处理单元集群2830提供图像数据。

[0159] 输入和分布逻辑

[0160] 在一个实施例中, 通过在整个原始图像上应用内核来执行卷积。处理单元2830进行协作以遍历图像, 将卷积运算应用于像素(乘法和加法), 并对结果进行分组以生成最终的输出图像。在该过程中, 处理单元的每个集群2830与来自输入图像的像素子集一起工作。输入逻辑2850从较高的高速缓存级得到图像数据, 并且根据处理单元的每个集群的计算需求来选择性地将其广播到容器2800-2803。事实上, 将像素内部存储在每个集群的容器中, 以用于它们的以后使用。事实上, 该模块化和可扩展的方法允许设计覆盖从小内核(2x2像素)变动到非常大的内核(64x64像素)的多个卷积配置。

[0161] 容器、对准逻辑和选择逻辑

[0162] 在一个实施例中, 当执行卷积时, 加速器处理两种类型的信息: 输入图像数据(像素)和内核权重。将两种类型的信息都存储在容器2800-2803中, 可以将容器2800-2803实现为允许并行读取和写入多个连续像素的无标签存储单元。在图29中图示了容器的一个实施例, 其包括固定宽度的缓冲器2901的集合和用于处理元素的写入和读取的简单逻辑。特别地, 对缓冲器2901的输入由应用到7b位掩码移位器2902的写使能信号(wr_en)来控制。在一个实施例中, 由读使能信号(rd_en)和写使能信号(wr_en)控制的计数器2904的输出与如所图示的写使能信号一起进行“或”(OR)运算以控制移位器2902。在一个实施例中, 移位器2902响应于移位信号(Init/Shift)而移位通过地址以选择每个缓冲器条目(entry)2901。位掩码移位器2902的输出选择写入输入数据的缓冲器条目。尽管总是按顺序写入像素(像在FIFO缓冲器中那样), 但可以随机读取它们。特别地, 在一个实施例中, 多路复用器2903响应于读取地址信号(rd_idx)选择缓冲器条目之一来输出(D_out)。

[0163] 在本发明的一个实施例中采用两种不同类型的容器: 用于图像数据的行容器以及用于内核权重的权重容器。在一个特定实施例中, 存在48个行容器和4个权重容器, 表示总共2048个字节。

[0164] 在将图像像素发送到执行集群之前将图像像素存储在行容器中。实际上, 每个单独的行容器存储单个图像行的多个像素。多个行容器的联合覆盖原始输入图像的一部分。容器使得同时获取对来自不同行的多个像素的访问而不需要对较高的高速缓存级的多次访问成为可能。在一个实施例中, 行容器的全集合能够从N个不同行提供多达4个连续的像素, 其中N是行容器的数量。为了执行卷积以自然的方式存储数据, 并且可以通过标识将应用操作的行来访问像素。因此, 一个实施例的容器不需要标签。

[0165] 在一个实施例中,来自行容器的输出连接到对准逻辑2810-2813,其给出了对行条目的单独像素的访问。对准逻辑2810-2813使得应用诸如生成图像列所需的数据转置(transpose)之类的简单的数据变换成为可能。图30中的表格示出了在本发明的一个实施例中实现的特定对准模式。第一列标识对准的类型,而其他列表示给定来自通用行容器(O_x)的内容的情况下的对准输出($A0_x$)。可以容易地扩展这个逻辑以支持本发明的不同实施例中的其他对准。

[0166] 在一个实施例中,权重以与处理单元所预期的相同格式存储在存储器中。因此,权重容器输出不需要对准,并且直接连接到选择逻辑2820-2823的输出。

[0167] 在一个实施例中,以处理单元对权重期望的相同格式在存储器中存储权重。因此,权重容器输出不需要对准并且直接连接到选择逻辑2820-2823的输出。

[0168] 在一个实施例中,选择逻辑2820-2823掌管以准备由处理单元消耗的格式来创建输出数据。通常,选择逻辑2820-2823知道需要从哪些容器取什么数据,以及需要将其放置在哪里以生成将对处理单元进行馈送的输出分组。为此,选择逻辑2820-2823的一个实施例从一个或多个容器2800-2803读取对准的输出,并将该信息放置到用于执行集群2830的输出分组中。在本发明的一个实施例中采用的逻辑在图31中的表格中示出。第一列是指正在考虑的图像访问模式:行、列或正方形。四个接下来的列是指输出分组。注意, C_x 是指容器编号 x ,并且 $A0_x$ 是指位置 x 中的容器对准输出。如其所示,通过读取4B行模式中的单个容器的四个对准输出来生成输出。在4B列模式下,通过选择四个不同容器的第一对准位置来生成输出。最后,在4B正方形模式中,通过选择两个不同容器的前两个对准位置(例如, 2×2 像素正方形区)来生成输出。这些模式允许以像素行、像素列或两者的组合遍历图像数据。这可以非常方便地向执行集群2830提供卷积模式,以高度利用在卷积中观察到的数据局部性。

[0169] 图32图示了容器2800-2803、对准逻辑2810-2813和选择逻辑2820-2823的一个实施例如何互连以产生用于执行集群2830的输出分组3210。多路复用器3201-3204的每个集合来自对准逻辑2810-2813的输出之一选择输出分组3210的一部分。注意,跨不同集群的部件以协作的方式工作,并且同时产生输出分组3210。在从选择逻辑2820-2823接收到数据之后,处理单元以锁步(lockstep)执行。

[0170] 容器2800-2803、对准逻辑2810和选择逻辑2820-2823一起使得以用于执行卷积的最方便的方式向所有处理单元提供数据成为可能。虽然处理单元的每个集群2830可能具有不同的数据需求,但是这个逻辑提供数据而不对较高的高速缓存级进行多次访问。将数据临时存储在容器2800-2803中,并且对准逻辑2810-2813和选择逻辑2820-2823以灵活的方式提供它以满足每个集群2830的需求。例如,图示实施例可以同时提供来自非连续行的多个图像列,总共仅需要一次存储器访问,而在传统的高速缓存方案中,这个操作需要针对每个访问行的一个单个访问。

[0171] 用于人工神经网络的可重配置功能单元和方法

[0172] 如上所提及,ANN被配置为具有通过“突触”连接的神经元的互连“神经元”层的集合。突触权重(或仅权重)是指两个神经元之间的连接的强度或幅度,其可以表示为数值。因此,神经元的输出可以通过由它们的突触权重衡量的来自前一层的所有输入神经元的相加来计算。该计算的输出然后通过激活函数(AF),该激活函数(AF)是一种数学函数,它基于神经元的加权输入来计算神经元的输出,如图33中所示。

[0173] 为了使神经网络近似或预测非线性行为,必须使用非线性激活函数。虽然sigmoid、有界可调(rectified)线性和双曲正切是最常见的函数,但是可以实现若干非线性函数。卷积神经网络(CNN)中的一个问题是选择什么AF。研究表明,不同的AF可提供不同的准确度,并且也可能需要较大或较短的训练时间。理想地,灵活的硬件设计应能够实现各种AF,每个AF都针对具体问题。

[0174] 在通用CPU上提供的当前解决方案提供了灵活性,但是大脑启发的算法可能具有非常高的计算需求,这在传统低功率划分中由于有限的功率预算和处理能力而可能是禁止的。为了克服这个问题,工业和学术界两者都对专门的神经形态架构表现出很大的兴趣,所述专门的神经形态架构提供了好于常规方法数量级的能量效率。

[0175] 然而,当前的神经形态架构提供非最优的AF实现。例如,许多现有的实现仅支持一种类型的AF,通常是在基准的巨大的集合上提供最佳结果的AF。例如,所选择的AF对于面部检测问题可能是优秀的,但对于语音识别而言是次最佳的。具有多个AF实现或者可以被重新配置以提供多个AF实现的架构可以解决这个问题。

[0176] 另一方面,虽然有试图用可重新配置的AF来实现神经形态核的一些提议,但是它们使用的方法并不有效。首先,那些现有的实现使用均匀分割,其不是最佳的,因为段的大小受最坏情况(例如最小的大小)的限制。此外,函数中每个点的计算由固定值来近似,这如图34A中所示可以用简单查找表(LUT)3400来实现。图34B示出了具有点的示例性sigmoid近似。可替代地,如图35A中所示,每个点的计算可以使用LUT 3500由具有给定斜率的线段来近似。图35B示出了具有线性函数的示例性sigmoid近似。

[0177] 本发明的一个实施例包括用于计算神经网络中的AF的优化和灵活的微架构。如图36中所示的,架构的一个实施例基于针对AF的特性而优化的分段(piecewise)近似单元3603。它还包括支持非均匀函数分割的多态解码器3601以及具有不同线段的参数的LUT 3602。如下所讨论的,它还可以包括执行计算的附加逻辑。考虑到CNN中使用的AF的特性,所提出的解决方案将最小准确度需求的近似段数量最小化,同时还减少了能量。该设计可以根据准确度需求而容易地缩放,并允许系统在从高性能划分变动到低功率划分的许多市场划分中是极高效的。

[0178] ANN可以在常规架构(例如高端处理器)或专门的神经形态加速器中执行。若干研究已经表明,后者提供了好于前者数量级的能量效率。因此,它们更适合于功率受限的划分,如SoC或可穿戴设备。然而,目前的神经形态加速器中的AF实现不是最佳的,在灵活性方面存在显著的约束(例如,它们仅支持一个AF),或者当可重新配置时,它们对于可接受的准确度级别不是高效的,因为它们使用需要大LUT块的具有均匀量化的简单点近似(参见图34A);或者需要较小LUT块但需要较大算术单元(乘法器和加法器)的具有均匀量化的分段近似(参见图35A)。

[0179] 本文描述的本发明的实施例克服了这些限制,并且能够利用小LUT和合理大小的算术单元来提供高准确度和效率两者。为了图示这些问题,表格1示出了对于固定点[s2.7]表示和固定最大误差 2^{-7} 在当前解决方案(A和B)中和在本发明的实施例中所需的LUT条目的数量。这被确定为面部检测CNN实现中的最小可接受误差。表格中的前三行示出了支持三个类型的AF中的每个所需的LUT条目的数量。最后一行示出了针对支持所有三个AF的可重新配置的解决方案所需的条目的数量。对于 2^{-7} 的固定误差,C只需要相比于B的50%的条目以

及相比于A的3.1%的条目。

[0180]	AF	A-均匀(点)	B-均匀(分段)	C-本发明的实施例
	双曲正切	512	32	16
	Sigmoid	256	16	16
	有界ReLU	512	4	4
	全部	512	32	16

[0181] 表格1。

[0182] 表格2示出了当针对所有实现类型考虑相同数量的LUT条目(16)(即,对于面积和功耗施加限制的情况)时,误差如何变化。

[0183]	AF	A-均匀(点)	B-均匀(分段)	C-提出的技术方案
	双曲正切	2^{-3}	2^{-6}	2^{-7}
	Sigmoid	2^{-4}	2^{-7}	2^{-7}
	有界ReLU	2^{-2}	2^{-7}	2^{-7}

[0184] 表格2。

[0185] 此外,对于算术计算,本发明的实施例仅需要移位器而不是在分段近似解决方案中通常使用的乘法器,因此也减少了功率和面积需求。

[0186] 本发明的实施例包括微架构,所述微架构被优化以用于有效地近似ANN中通常使用的激活函数,即双曲正切、sigmoid和有界ReLU。然而,微架构足够灵活以支持其他函数,只要它们共享在ANN中使用的AF中通常发现的共同特性,其他函数即限于原点附近的输入改变以及对原点附近的输入改变更敏感的函数。所提出的设计是基于使用线性函数的分段近似的原理。此外,在一个实施例中,每个线段是等式(1)中所示的类型:

$$y(x) = \alpha + \beta x \quad (1)$$

[0187]

$$\text{其中 } \beta = \{-2^t, 0, 2^t\}; t \in N$$

[0188] 激活函数单元概述

[0189] 激活函数(AF)单元的一个实施例包括图36中所示的三个主要部件:(a)多态解码器3601、(b)LUT 3602和(c)分段插值近似单元3603。

[0190] 在一个实施例中,多态解码器3601将每个输入X映射到横坐标空间中的范围,并且利用所考虑的激活函数在原点附近需要较多精确度(即较小的量化间隔),并且对于较大的|X|需要较少精确度的事实(见图37A)。该特性对于用于对存在于神经元的响应中的非线性建模的激活函数是常见的。因此,如果我们将输入量化间隔的固定初始集合考虑为图37A中所示的那些,则多态解码器3601可以被配置为选择间隔的较小的非均匀子集,诸如图37B中所示的那些(如粗线所指示的)。该方法支持根据近似激活函数的特性的输入X与存储在LUT 2602中的分段线段的索引之间的有效且灵活的变换。因此,针对给定函数所选择的间隔与其特性相关。

[0191] 在一个实施例中,LUT单元3602是存储线性插值段的参数的部件。如所图示的,它使用多态解码器3601的结果作为输入。对于给定的线段,存储在LUT 3602中的参数由分段插值近似单元3603用于根据上述等式(1)来计算最终结果。

[0192] 方法概述

[0193] 根据本发明的一个实施例实现以下方法。首先,使用任何分段近似算法将所考虑的激活函数分为线段,所述任何分段近似算法保证:(1)最大误差 ϵ 和(2)所有斜率都遵守规则 $\beta = \{-2^t, 0, 2^t\}$

[0194] 为了图示的目的,图38中示出了具有最大误差为 2^{-5} 和5个线段的sigmoid函数的示例。将对于不同线段而离线获得的参数存储到LUT单元3602中,并且根据相应的X范围配置多态解码器3601。由于可以控制最大误差 ϵ ,并且由于所需的线段的数量取决于误差本身(参见表格1和2),因此可能使用针对近似计算解决方案的这个方法,其中,根据目标误差,重新调整LUT 3602中的条目的数量以节省额外的功率(例如,使用时钟门控)。

[0195] 另外,可以将AF单元并入神经形成加速器1230的处理单元中,诸如图12A-图12B中所图示的加速器。此外,为了利用单元的可配置能力,以下机制可能暴露给软件:

[0196] (1)根据目标函数将近似参数加载到LUT 3602中的机制。可以通过诸如“Load M (Rx)→AF”的指令来暴露该机制。

[0197] (2)如果支持多个AF(例如通过使用影子寄存器或支持复制寄存器状态的其他类型的实施例),则实施附加指令以在不同AF之间切换,例如“SwitchAF AFID#”。

[0198] 一个示例性实施例

[0199] 图39中图示了一个示例性实施例,其还包括多态解码器3902和LUT 3903。模X块(|X|)3901利用了双曲和sigmoid函数具有一些对称性以进一步将LUT条目数减半(halve)的事实。如上面所提及的,LUT 3902维持AF线性分割所需的不同参数。在这种情况下,有6个参数 $(t, \alpha', \alpha'', z', z'', e)$ 。

[0200] 分段近似算术单元的一个实施例包括执行指定功能的以下部件:

[0201] (i) Sgn(x)块3910--X的符号用于选择偏移参数(α),当 $\beta=0$ (z)时为零参数,(参见上面的等式(1)),以及调整Y的最终符号。数学上我们有:

$$[0202] \quad y(x) = \text{sgn}(x) \times \begin{cases} \alpha' + z'\beta x, & \text{if } x \geq 0 \\ \alpha'' + z''\beta x, & \text{if } x < 0 \end{cases} \wedge \beta = 2^t \wedge z', z'' \in \{0,1\}$$

[0203] (ii) 移位器块3911--移位器块用于计算 $\beta=2^t$ 。当移位时,可能发生一些精度损失。本发明的一个实施例以两种方式减少了该损失:(1)当右移位时,将LSB作为保护位存储并用作加法中的进位(carry-in);和(2)取决于“ α ”和“ t ”,编程器可以配置算术单元以两种方式计算等式(1):

$$[0204] \quad y(x) = \text{sgn}(x) \times (\alpha + \beta x) \text{ 或 } y(x) = \text{sgn}(x) \times \left(\beta \left(\frac{\alpha}{\beta} + x \right) \right)$$

[0205] 第一种情况,其中第二种情况可以用来避免在加法发生之前右移位。

[0206] (iii) 加法器块3912--最后,加法器块用于执行加法。

[0207] 图40中示出了多态解码器3902的可能实施例。它由小LUT 4001、4002、4003的组组成,如图37A中所示,所述组对函数X范围的不同段提供可变精度间隔。然后将LUT条目配置为将X个间隔映射到某个分段段。

[0208] 具有固定精度[s2.7]的输入X和[s0.7]的输出Y以及对于具有最大误差为 2^{-7} 的sigmoid、有界可调线性和双曲正切函数的支持的AF单元的实施例将需要具有16个条目的LUT和具有2个LUT的多态解码器,一个具有 25×4 位并且第二个具有 24×4 位。

[0209] 在前面的说明书中,已经参照本发明的特定示例性实施例描述了本发明。然而,清楚的是,在不脱离如所附权利要求中阐述的本发明的更广泛的精神和范围的情况下,可以对其进行各种修改和改变。因此,说明书和附图被认为是说明性的而不是限制性的。

[0210] 本发明的实施例可以包括上面已经描述的各种步骤。这些步骤可以体现在机器可执行指令中,所述机器可执行指令可以用于使通用或专用处理器执行步骤。可替代地,这些步骤可以由包含用于执行步骤的硬连线逻辑的特定硬件部件或由编程的计算机部件和定制硬件部件的任何组合来执行。

[0211] 如本文所述,指令可以指代被配置为执行某些操作或具有存储在体现在非临时计算机可读介质中的存储器中的预定功能或软件指令的硬件的特定配置,诸如专用集成电路(ASIC)。因此,图中所示的技术可以使用在一个或多个电子设备(例如,终端站、网络元件等)上存储和执行的代码和数据来实现。这样的电子设备使用诸如非临时计算机机器可读存储介质(例如,磁盘;光盘;随机存取存储器;只读存储器;闪速存储设备;相变存储器)和临时计算机机器可读通信介质(例如,电、光、声或其他形式的传播信号-诸如载波、红外信号、数字信号等)之类的计算机机器可读介质来存储和(内部地和/或通过网络与其他电子设备)传送代码和数据。

[0212] 此外,这样的电子设备通常包括耦合到一个或多个其他部件的一个或多个处理器的集合,所述一个或多个其他部件诸如是一个或多个存储设备(非临时性机器可读存储介质)、用户输入/输出设备(例如键盘、触摸屏和/或显示器)和网络连接。处理器集合和其他部件的耦合通常通过一个或多个总线和桥(也称为总线控制器)。携带网络流量的存储设备和信号分别表示一个或多个机器可读存储介质和机器可读通信介质。因此,给定电子设备的存储设备通常存储用于在该电子设备的一个或多个处理器的集合上执行的代码和/或数据。当然,可以使用软件、固件和/或硬件的不同组合来实现本发明的实施例的一个或多个部分。贯穿该具体实施方式,出于解释的目的,阐述了许多具体细节以便提供对本发明的透彻理解。然而,对于本领域技术人员清楚的是,可以在没有这些具体细节中的一些的情况下实践本发明。在某些情况下,为了避免模糊本发明的主题,未以详细的细节描述公知的结构和功能。因此,本发明的范围和精神应根据以下权利要求来判断。

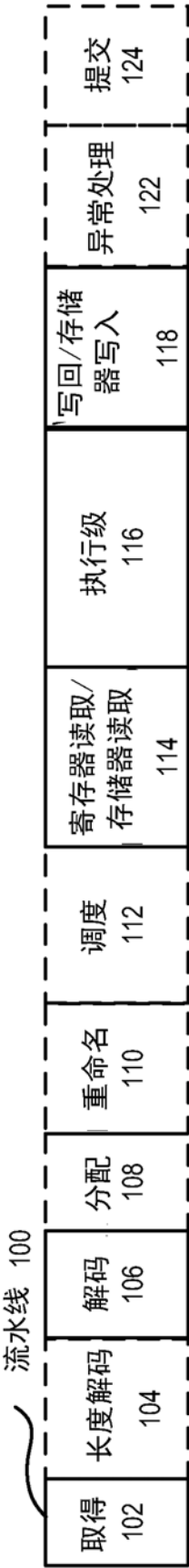


图 1A

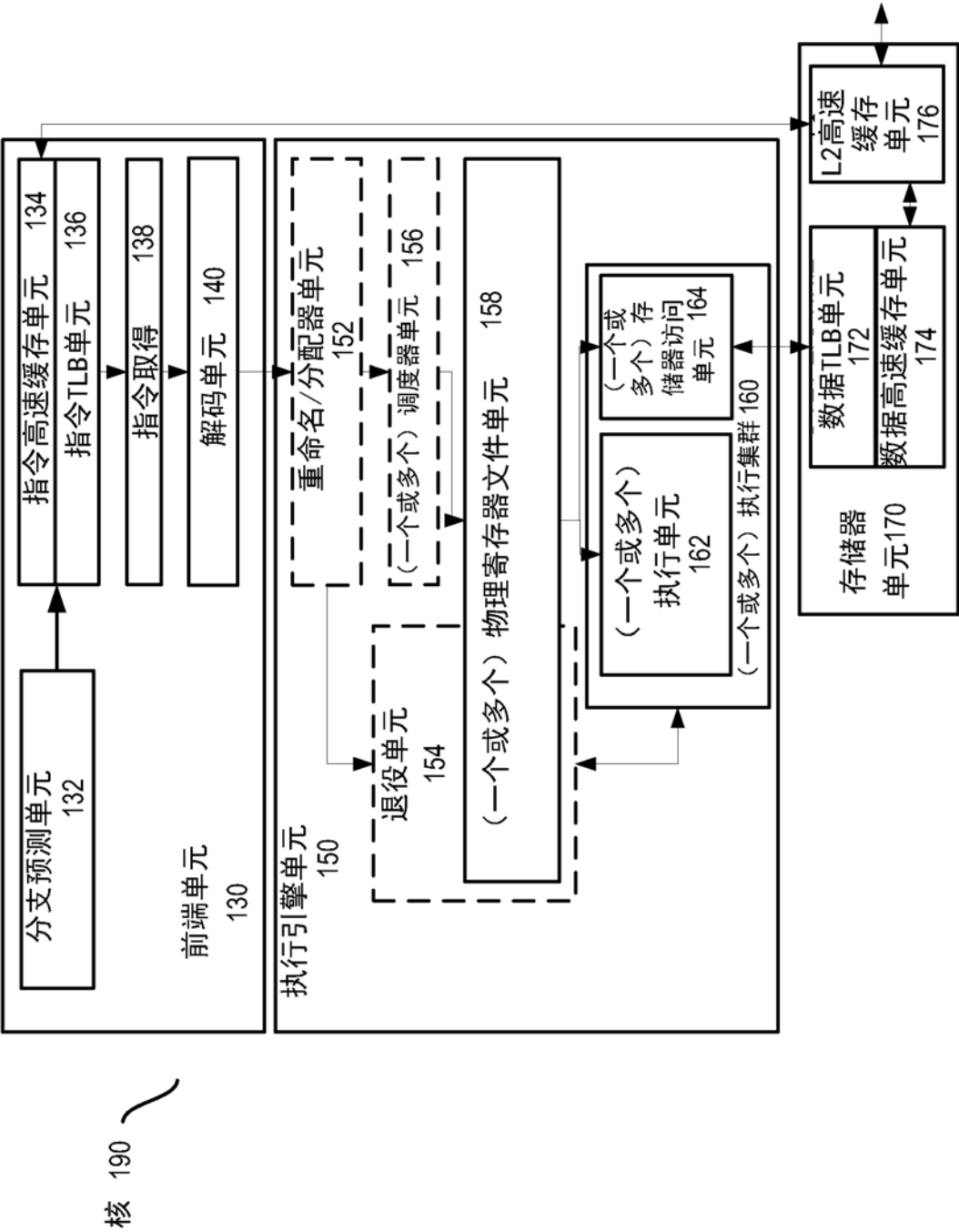


图 1B

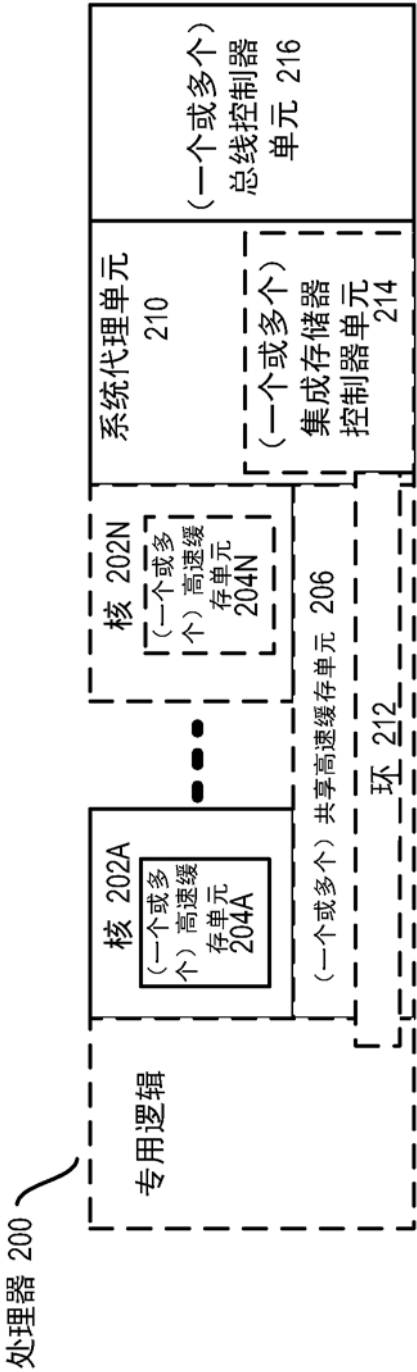


图 2

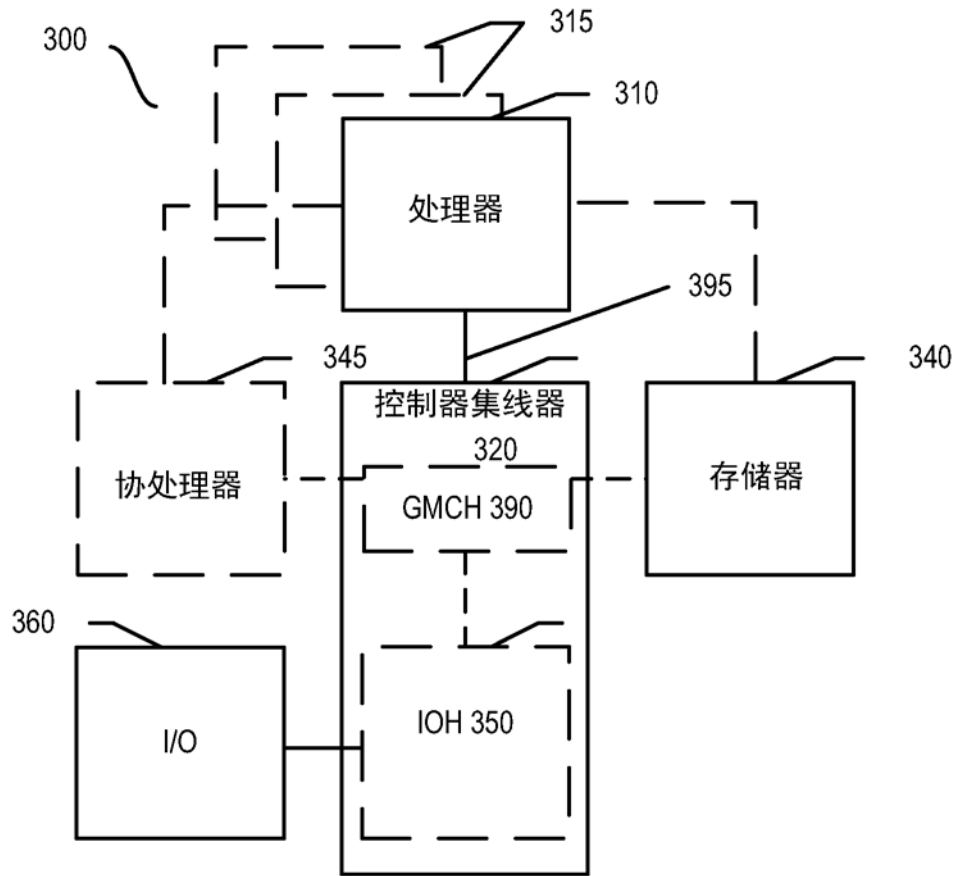


图 3

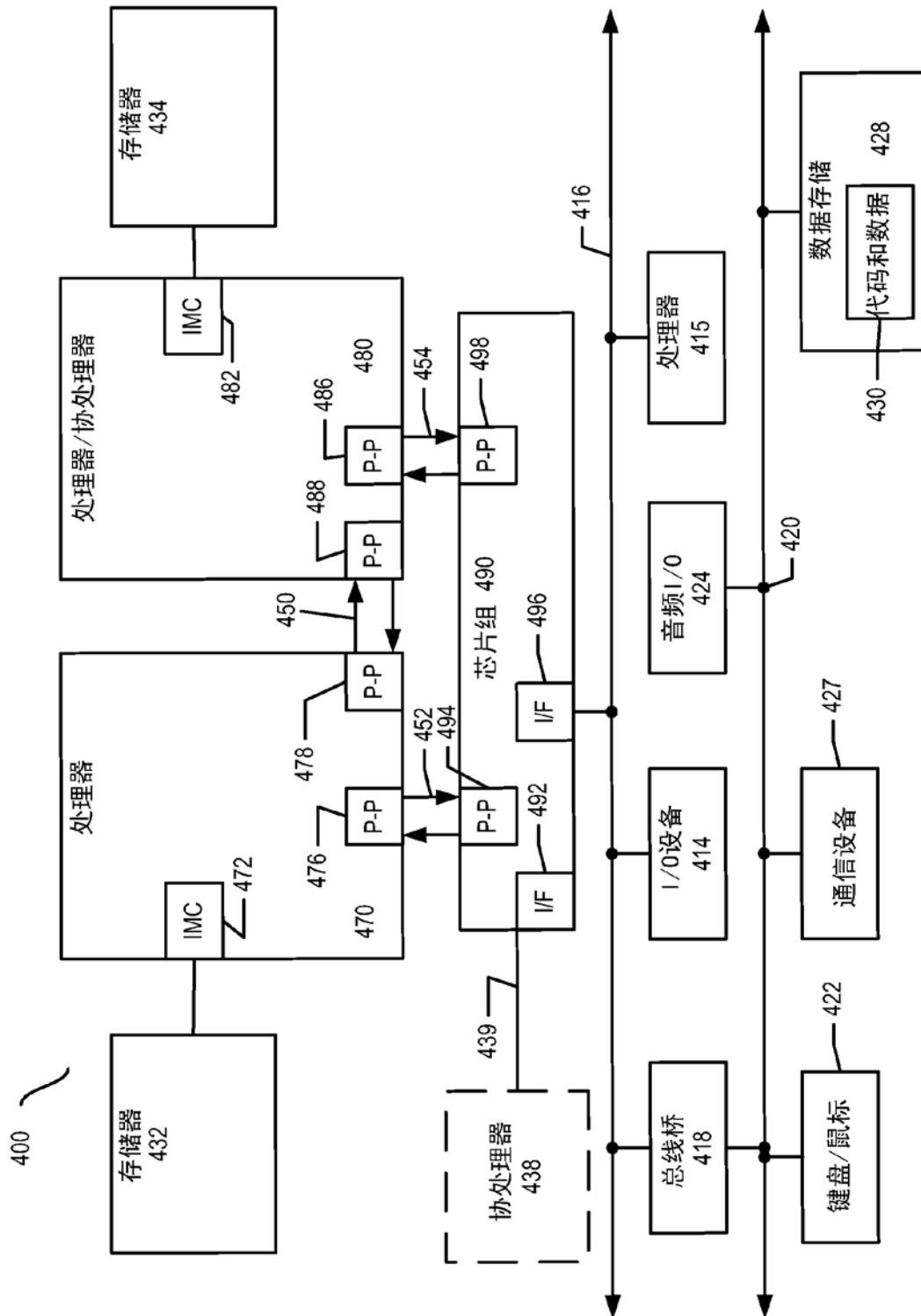


图 4

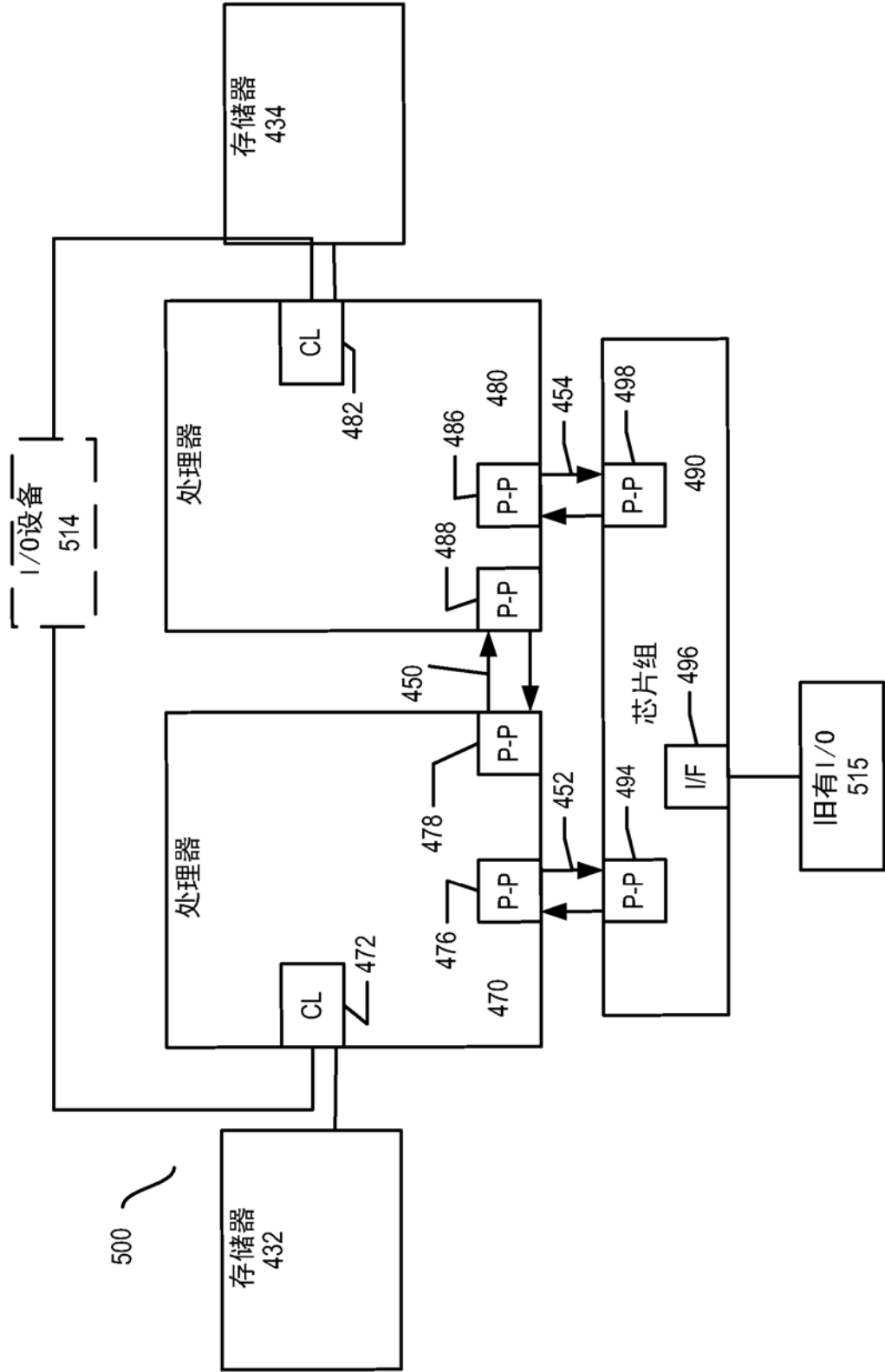


图 5

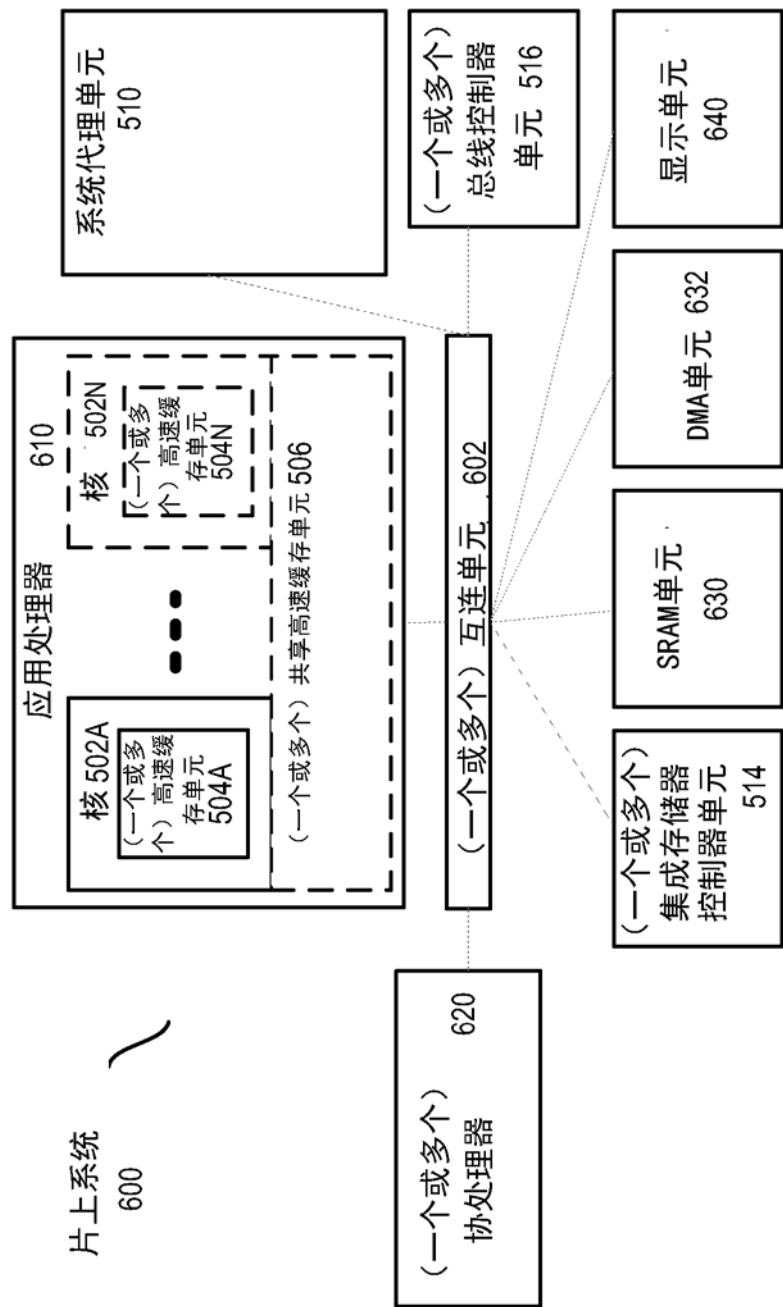


图 6

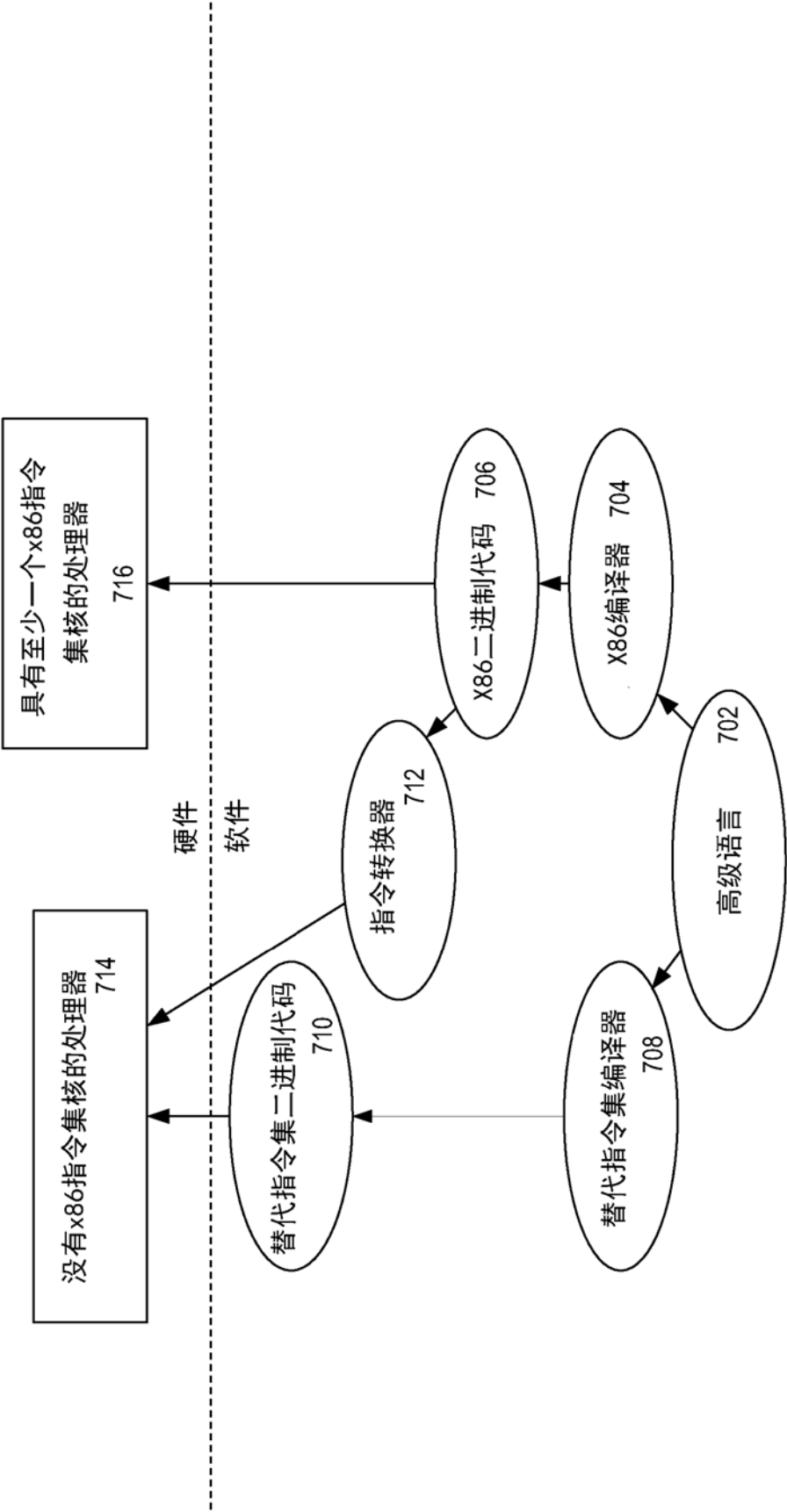


图 7

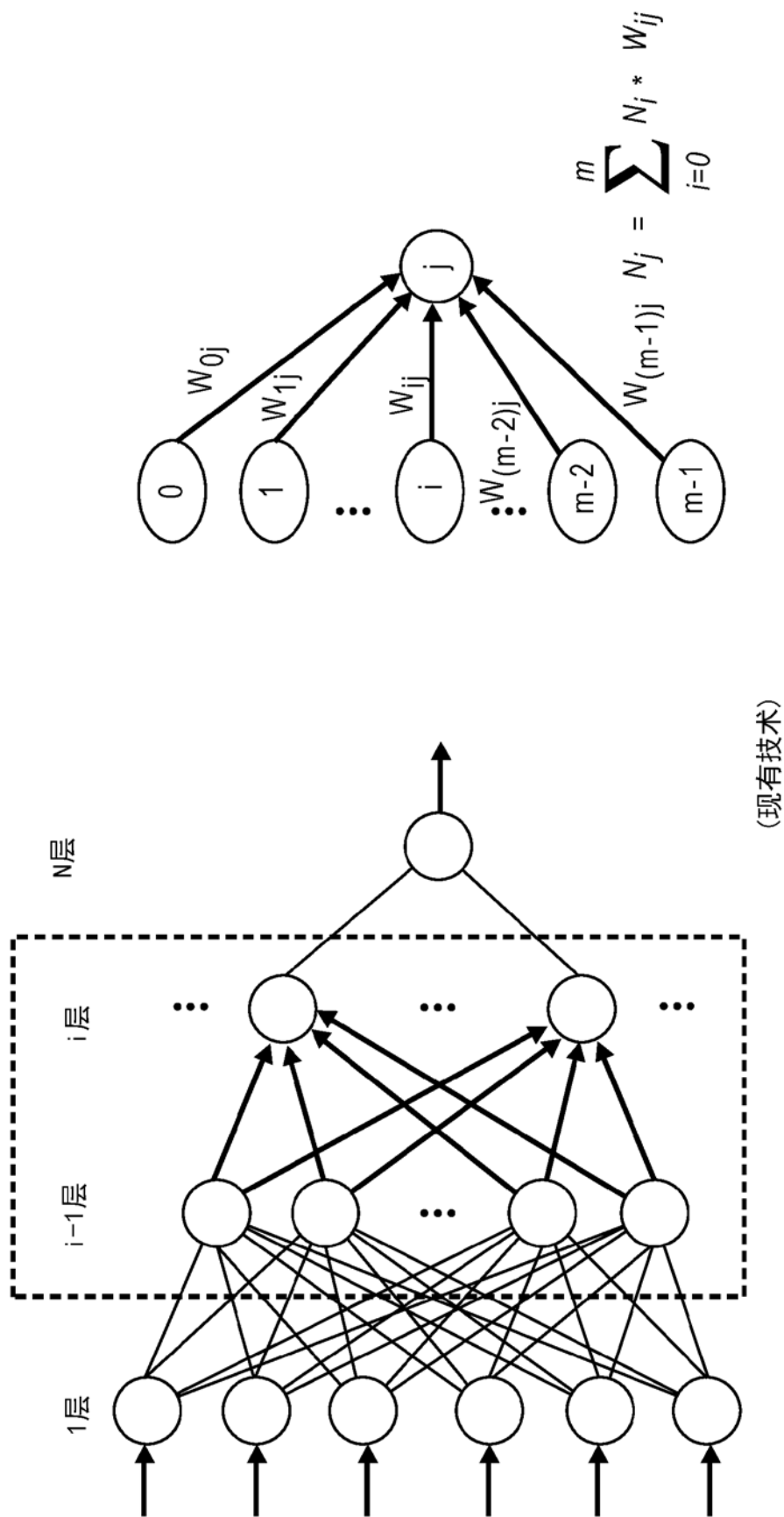


图 8

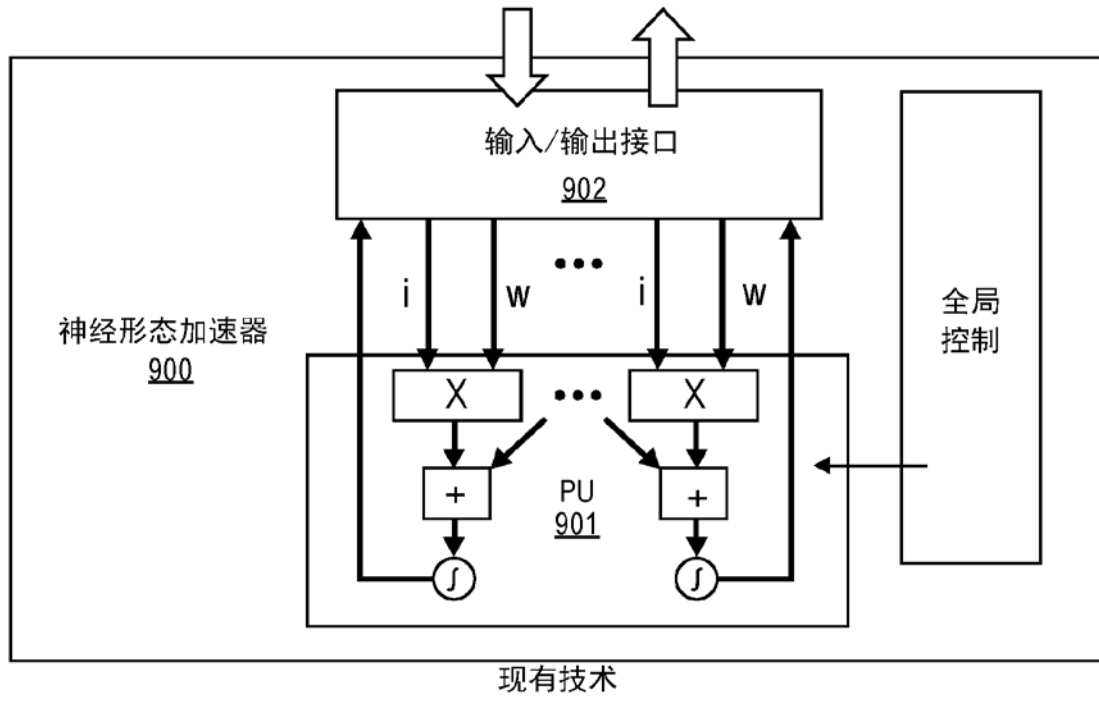
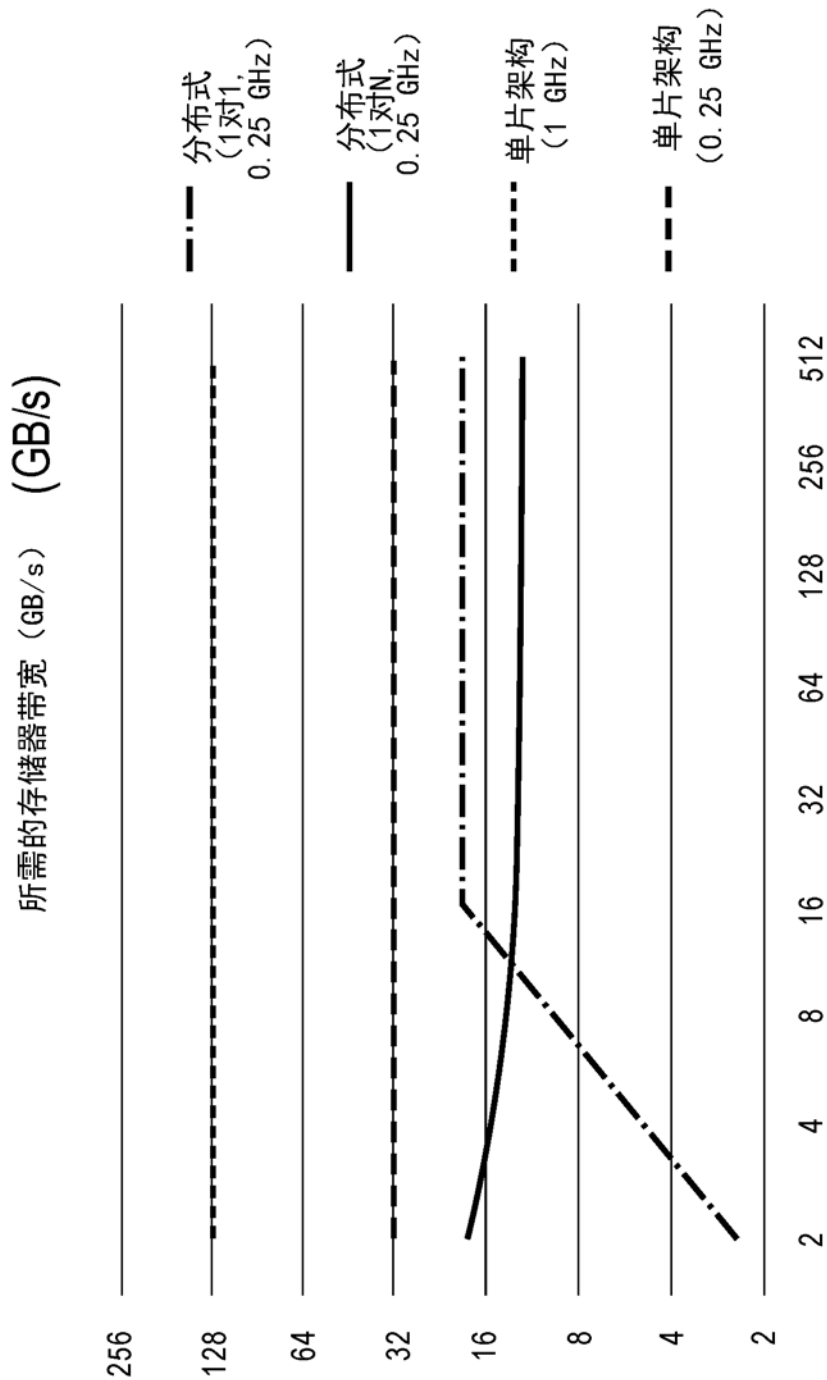


图 9



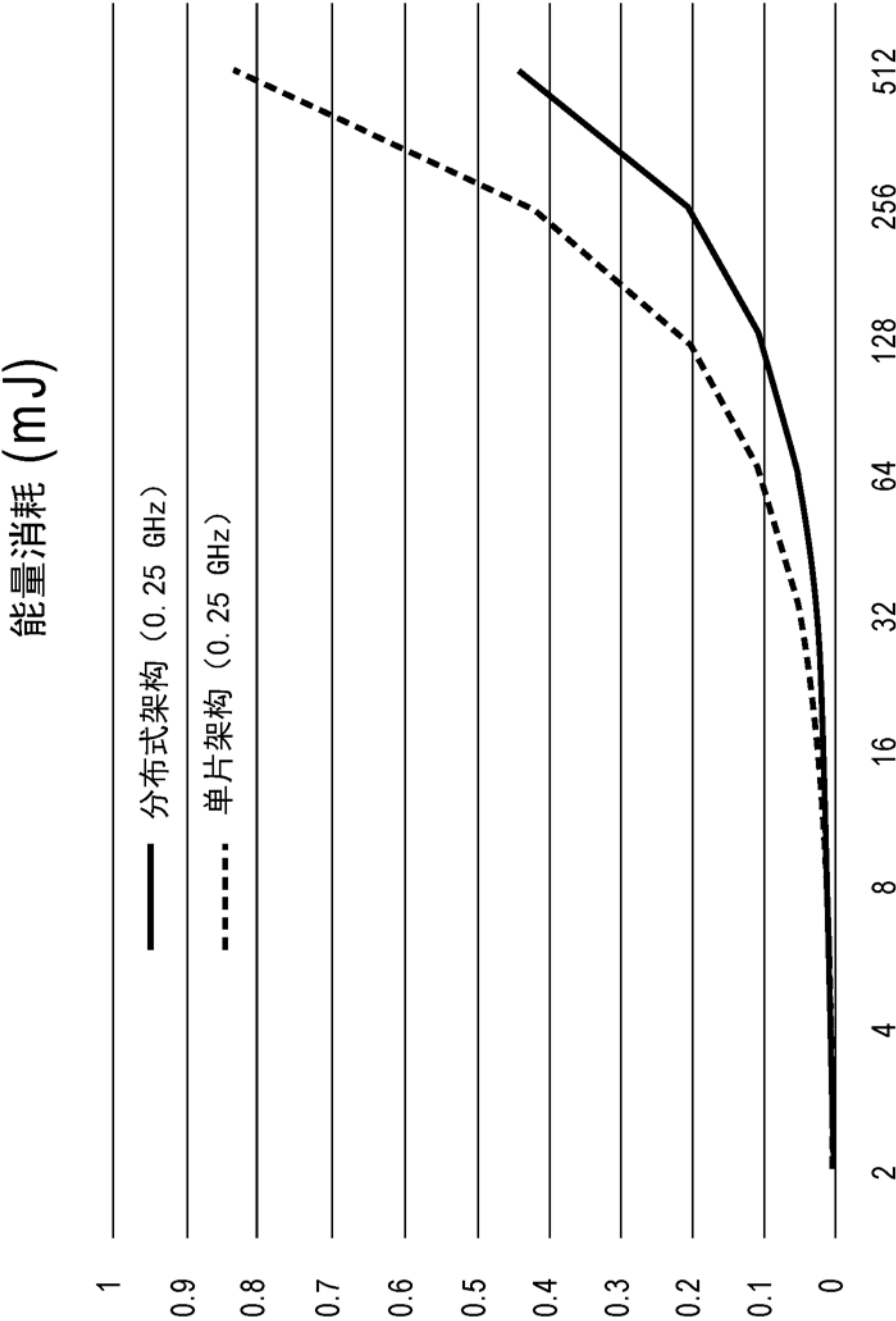


图 11

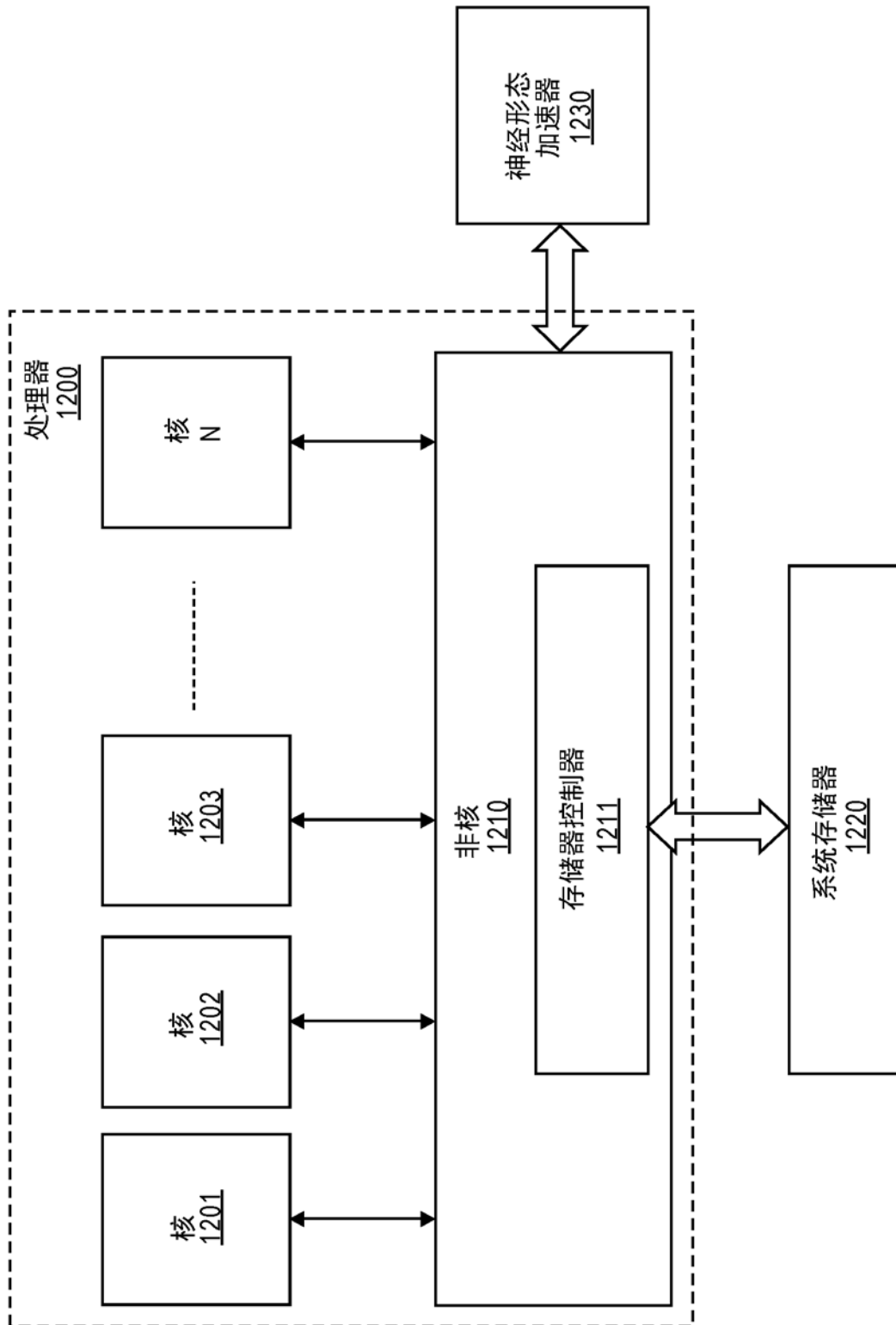


图 12A

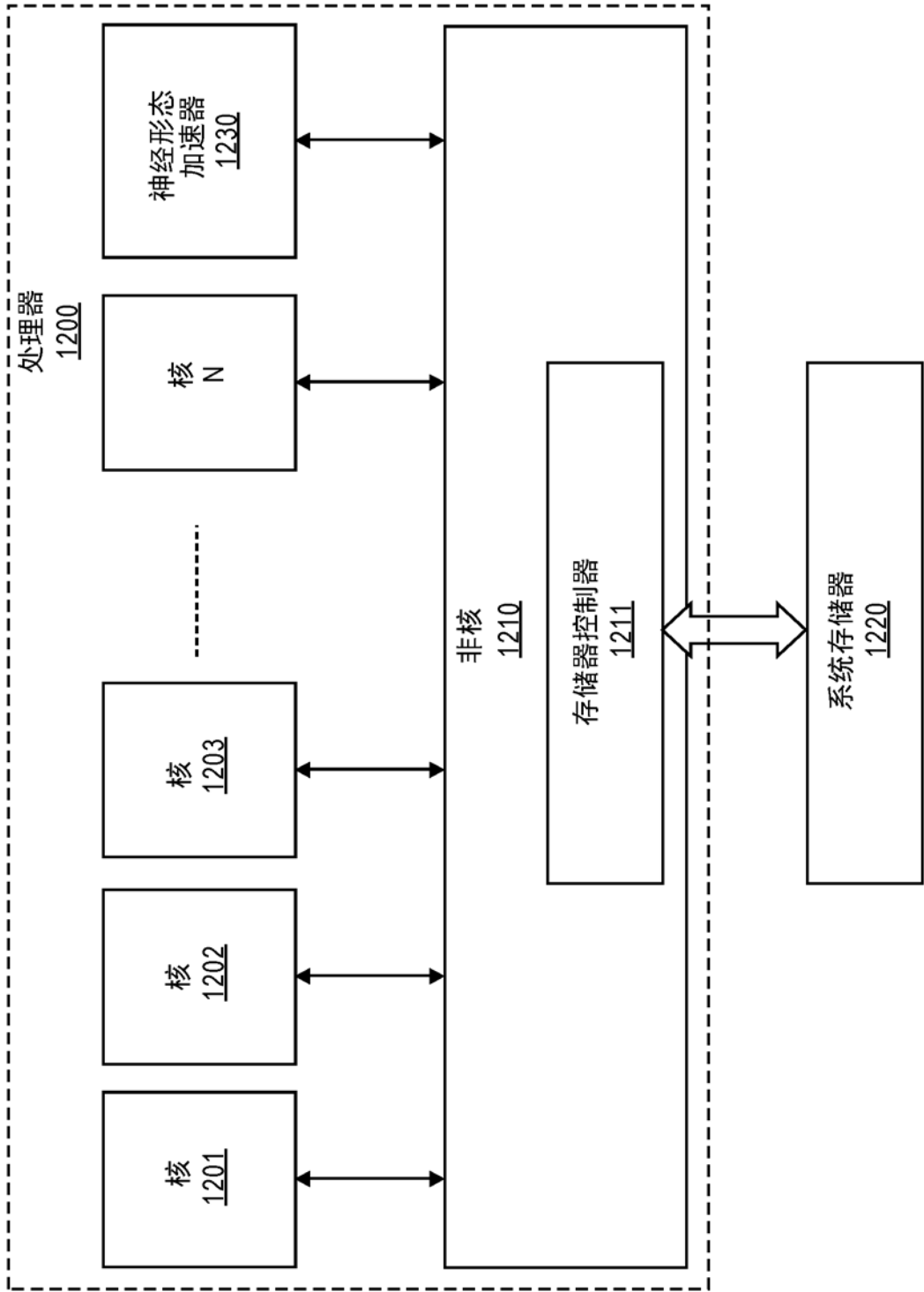


图 12B

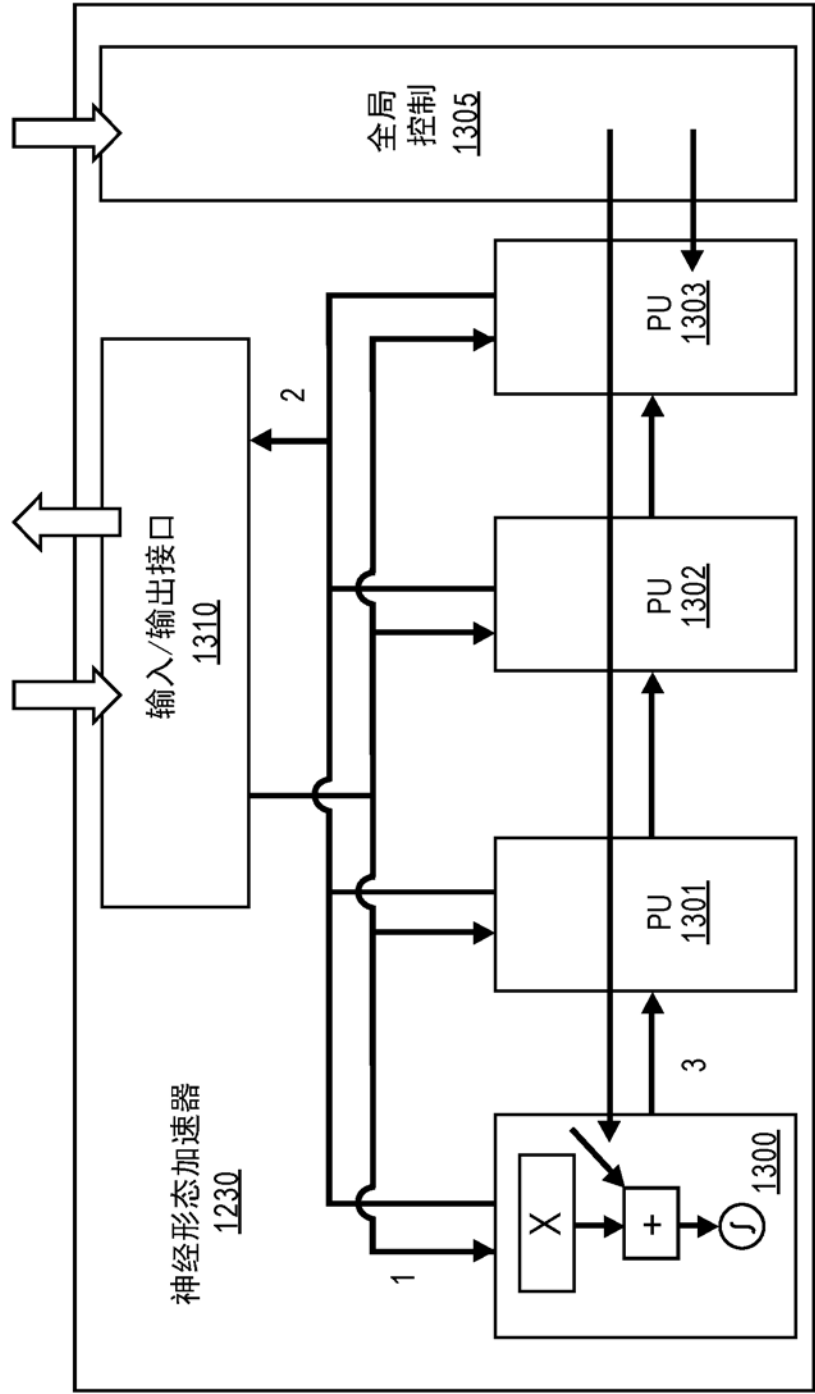


图 13

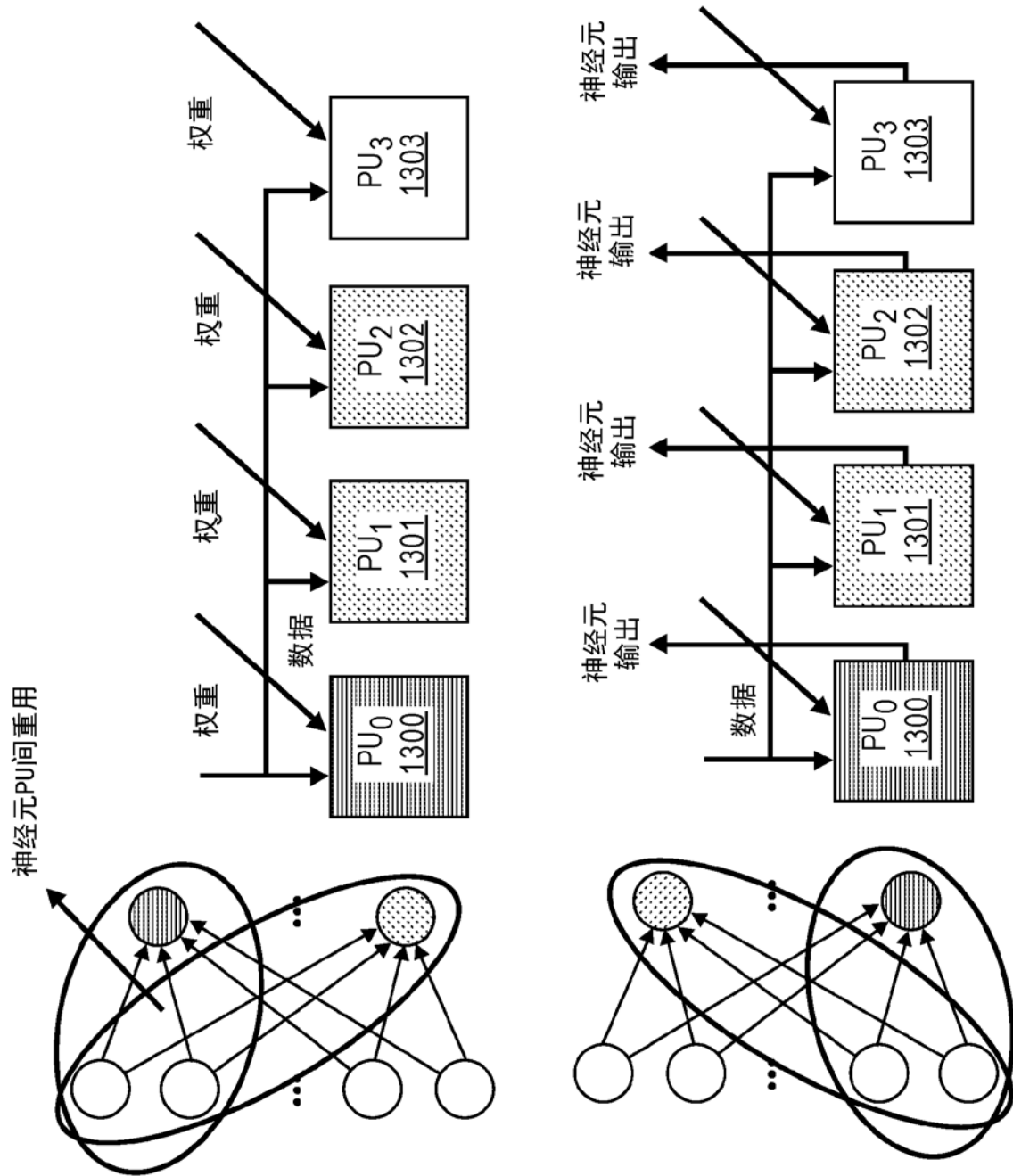


图 14A

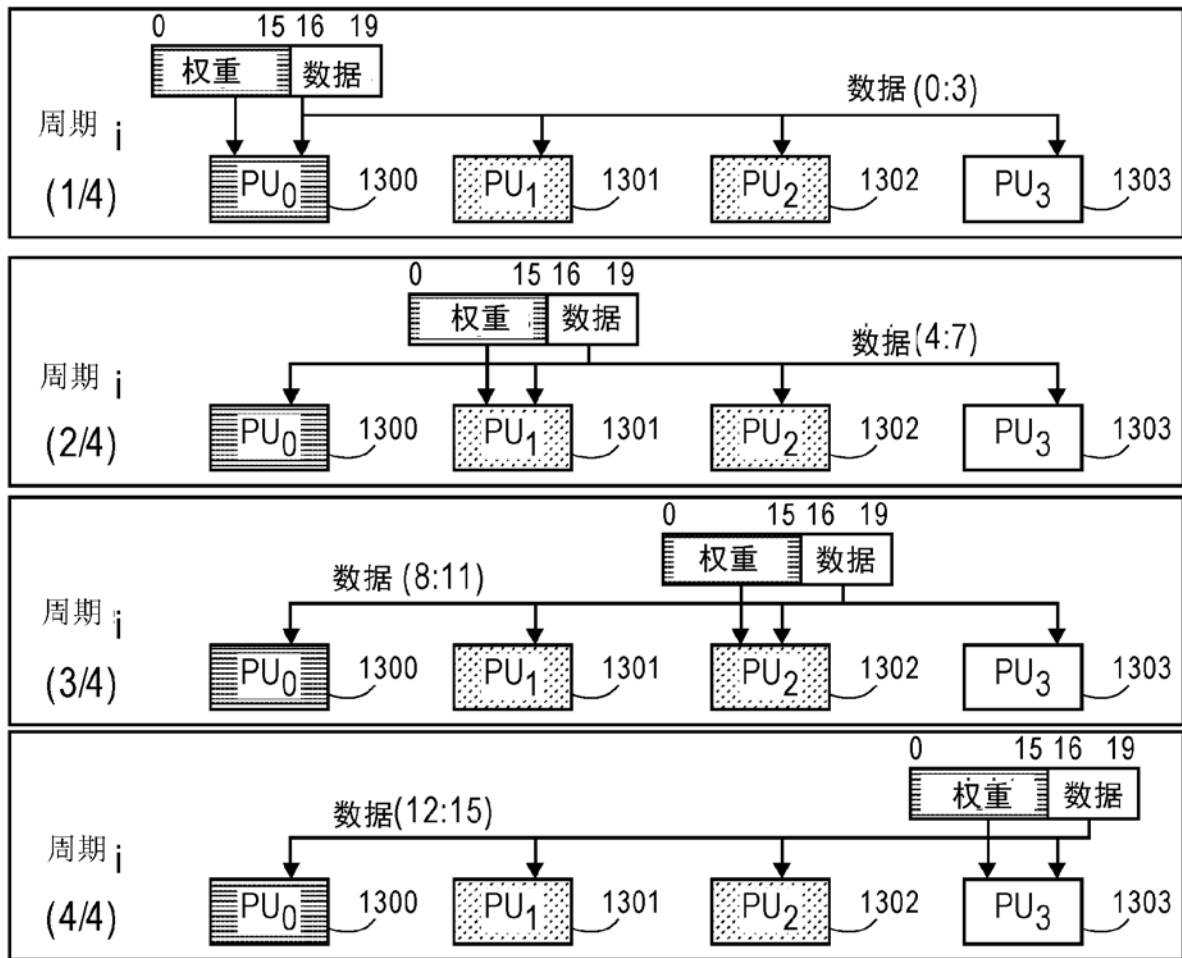


图 14B

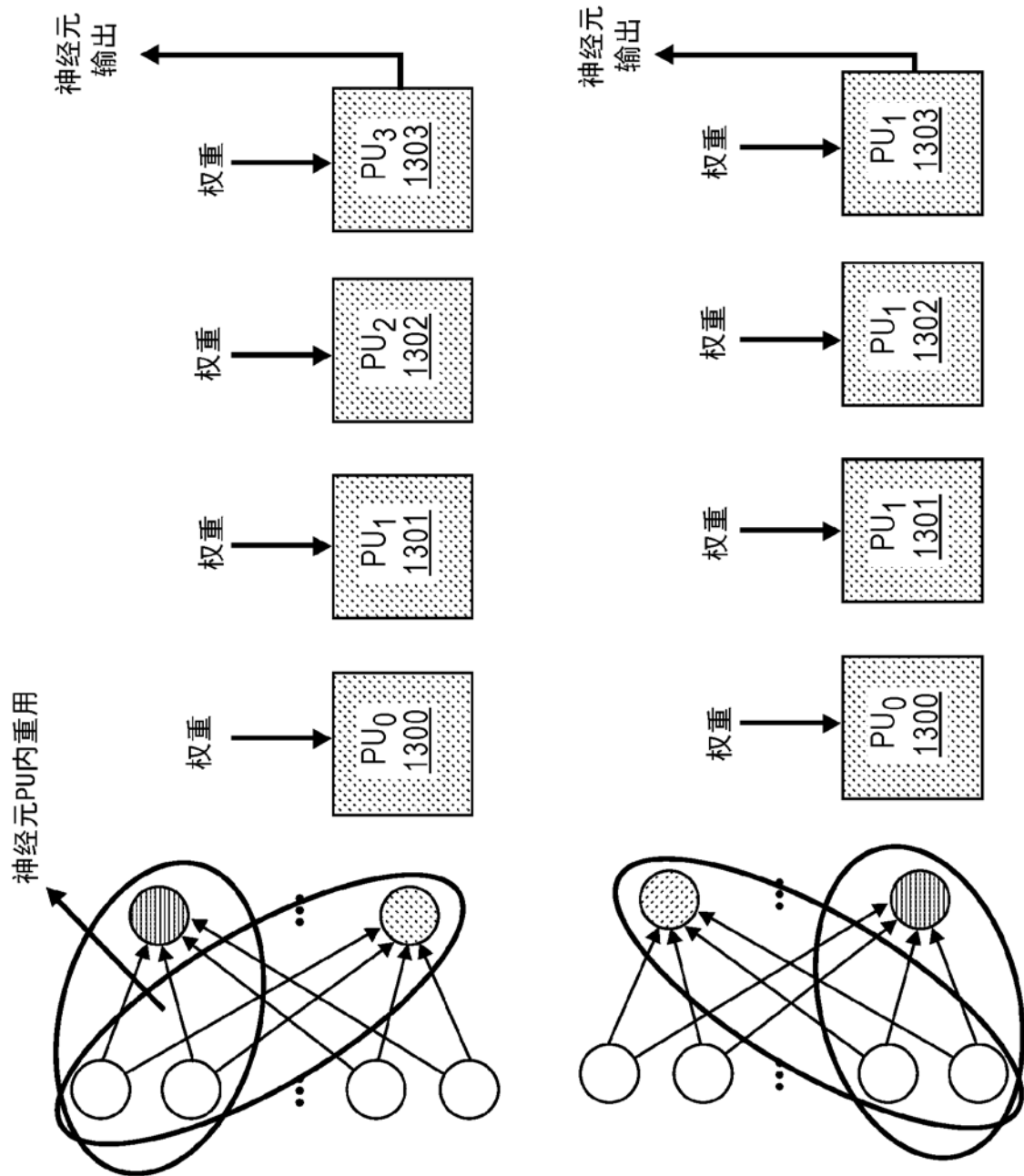


图 15A

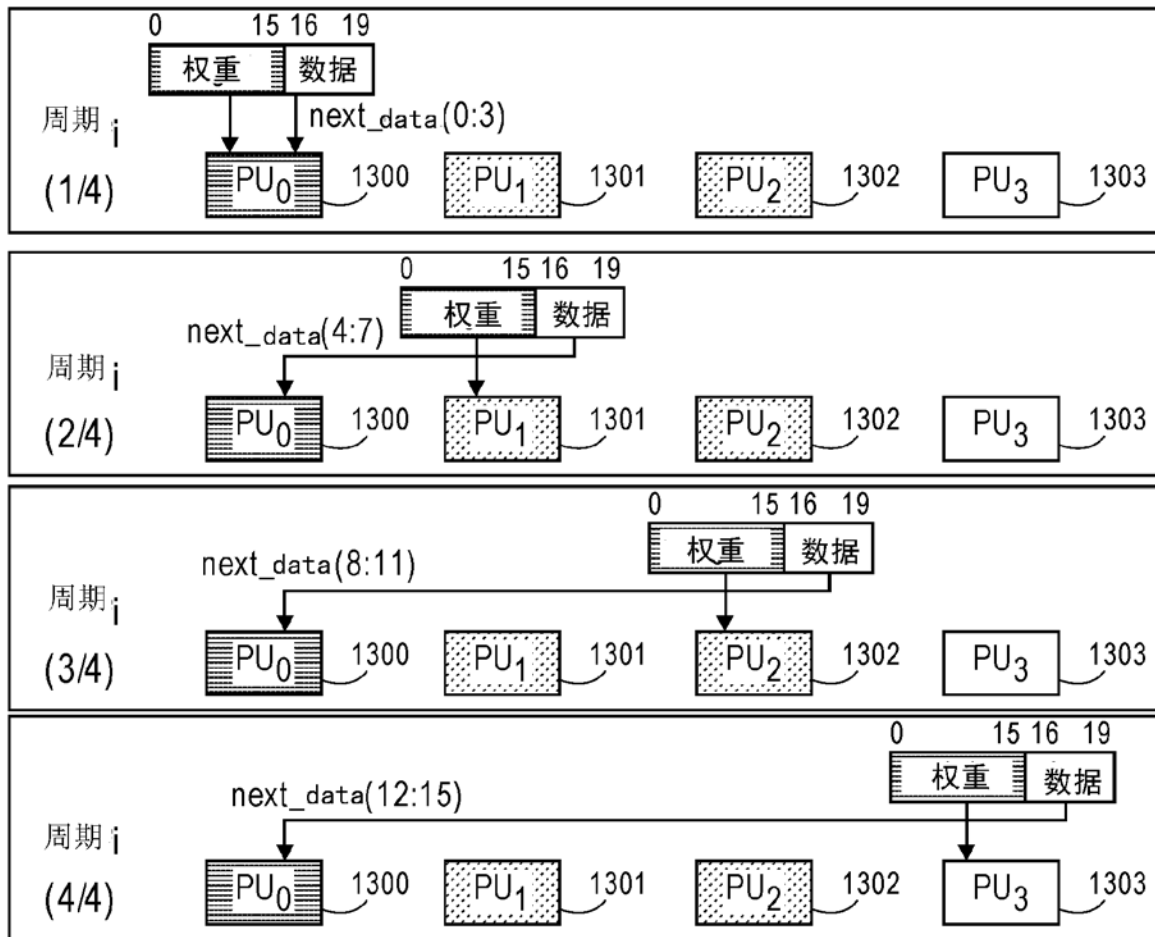


图 15B

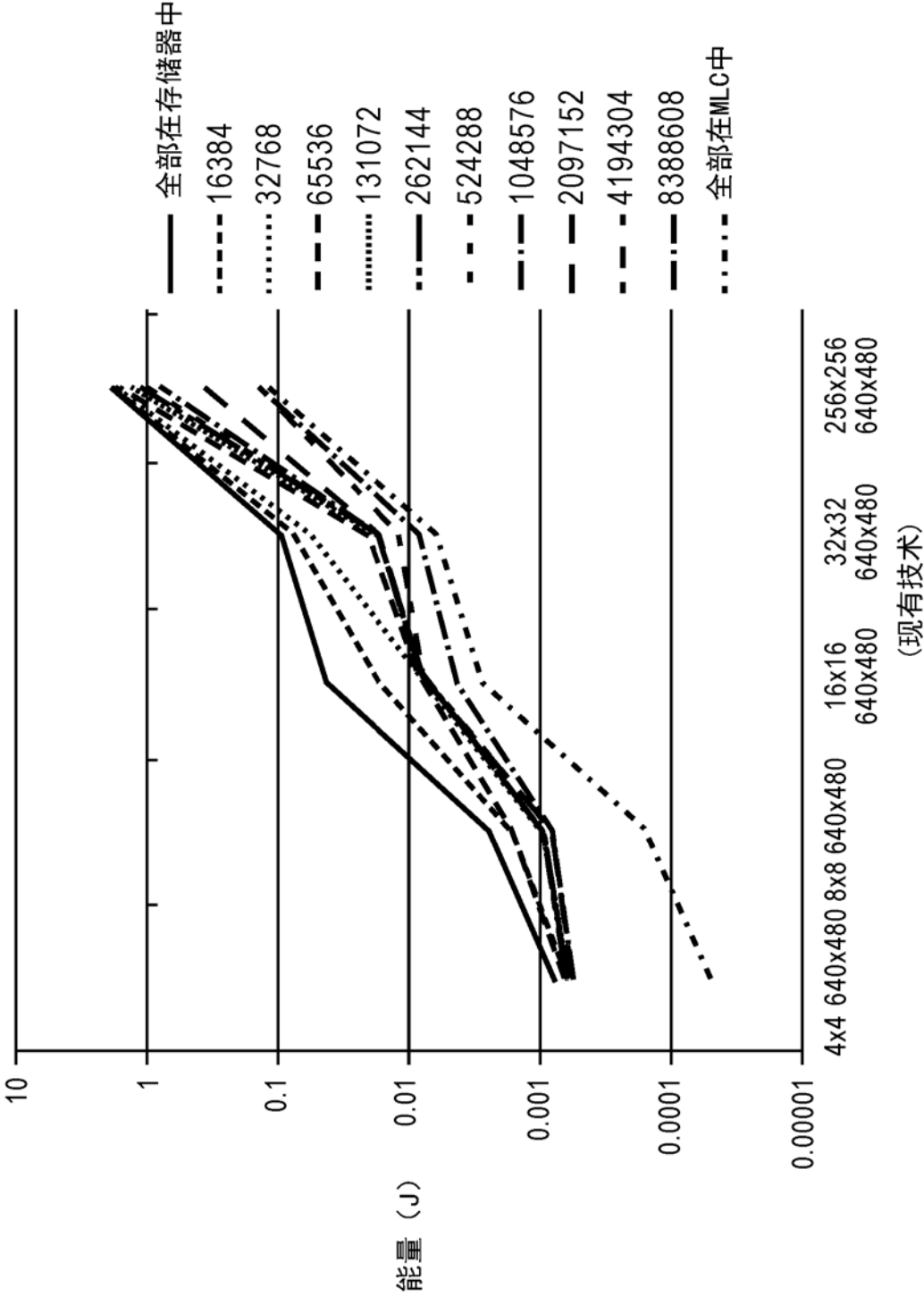


图 16

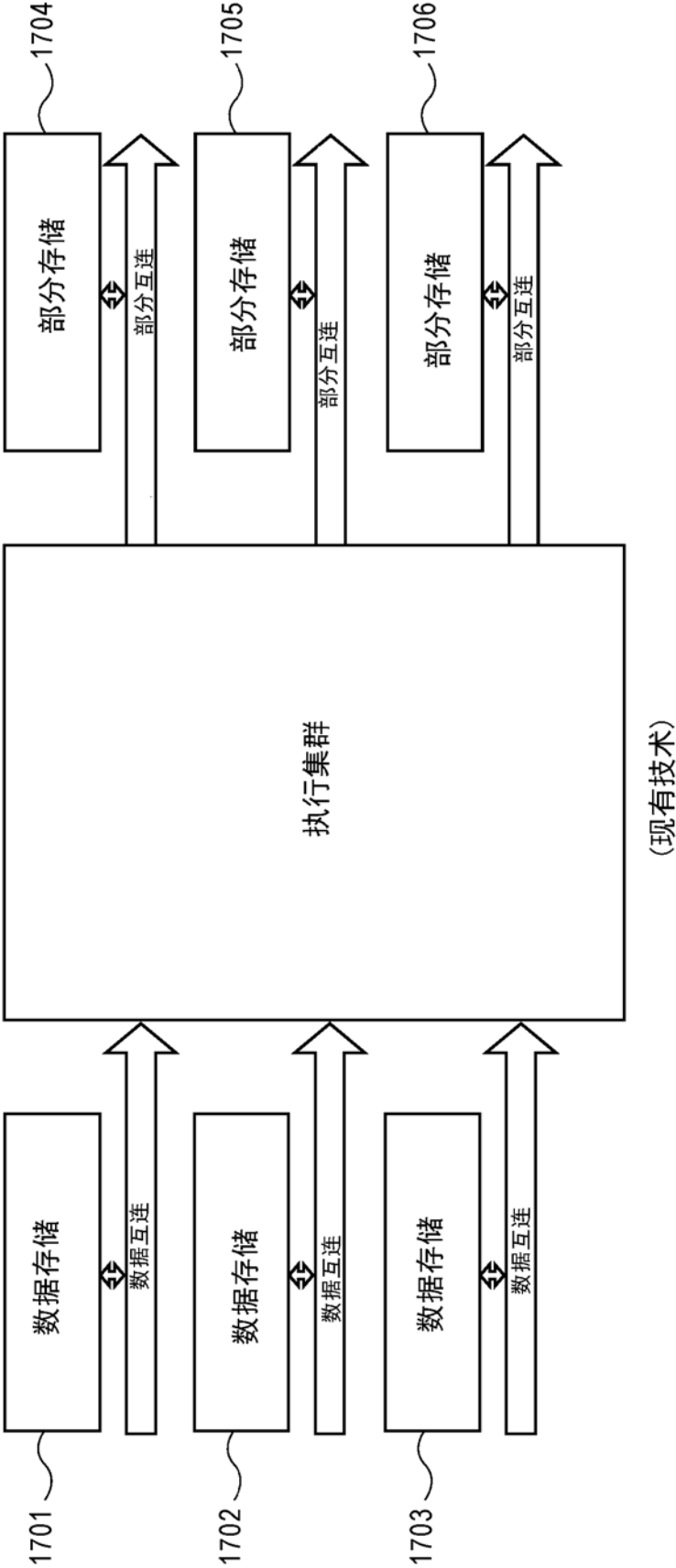


图 17A

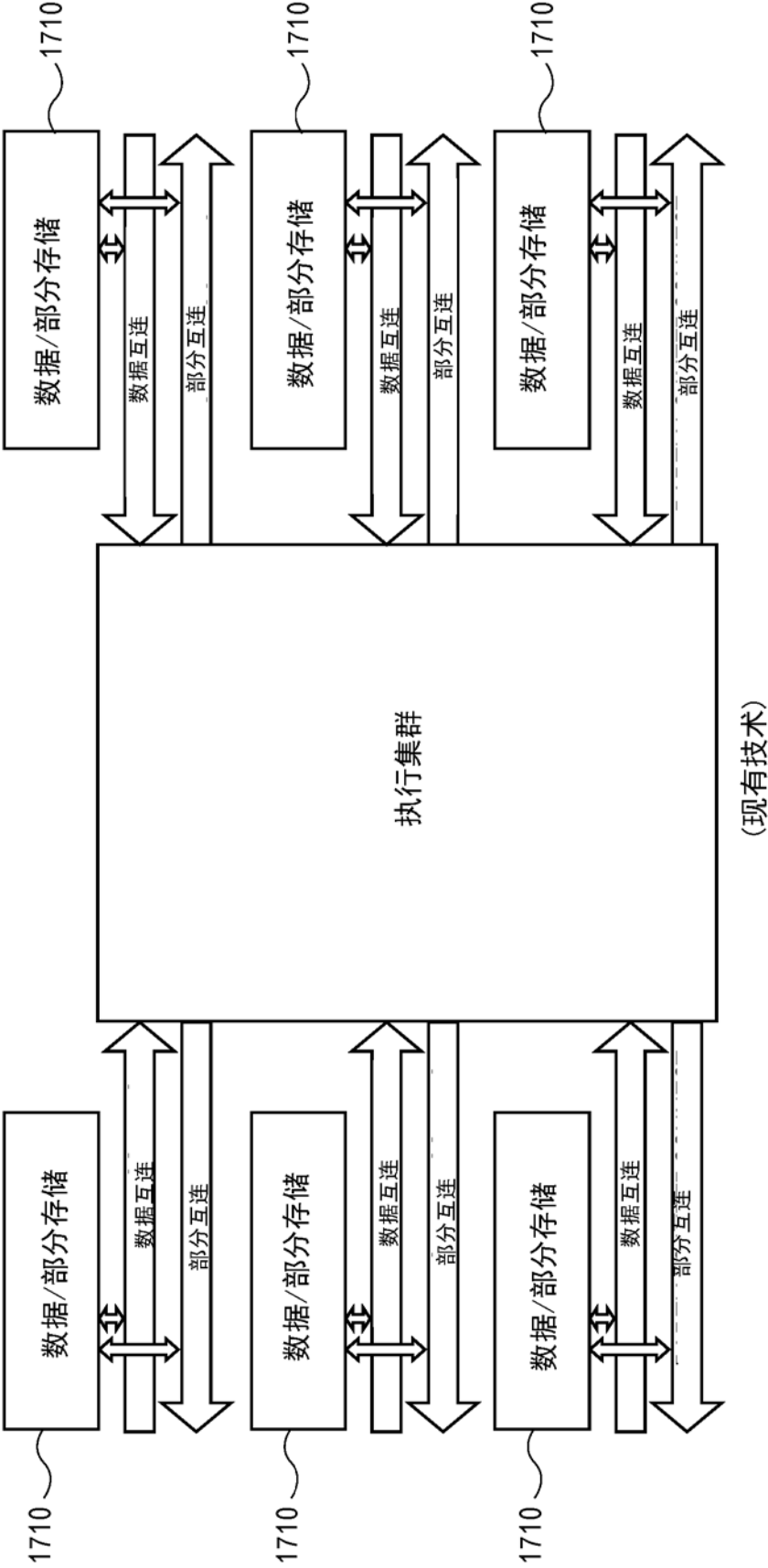


图 17B

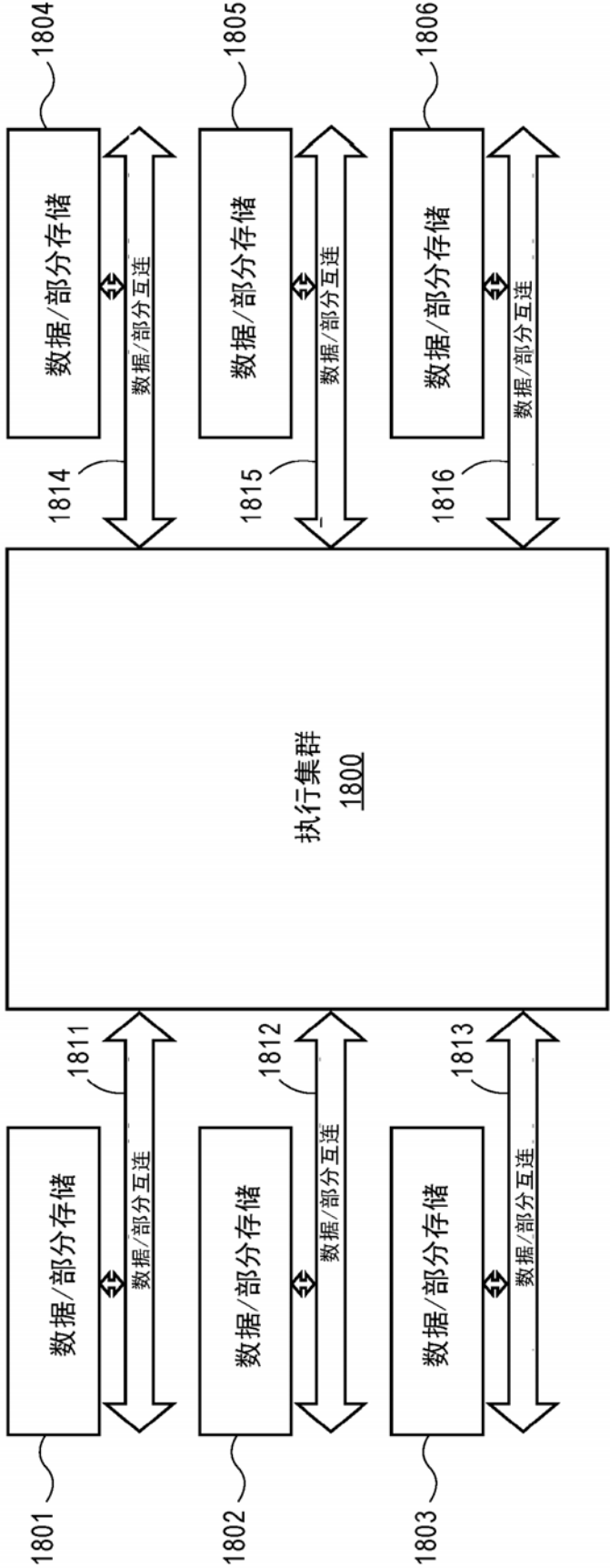


图 18A

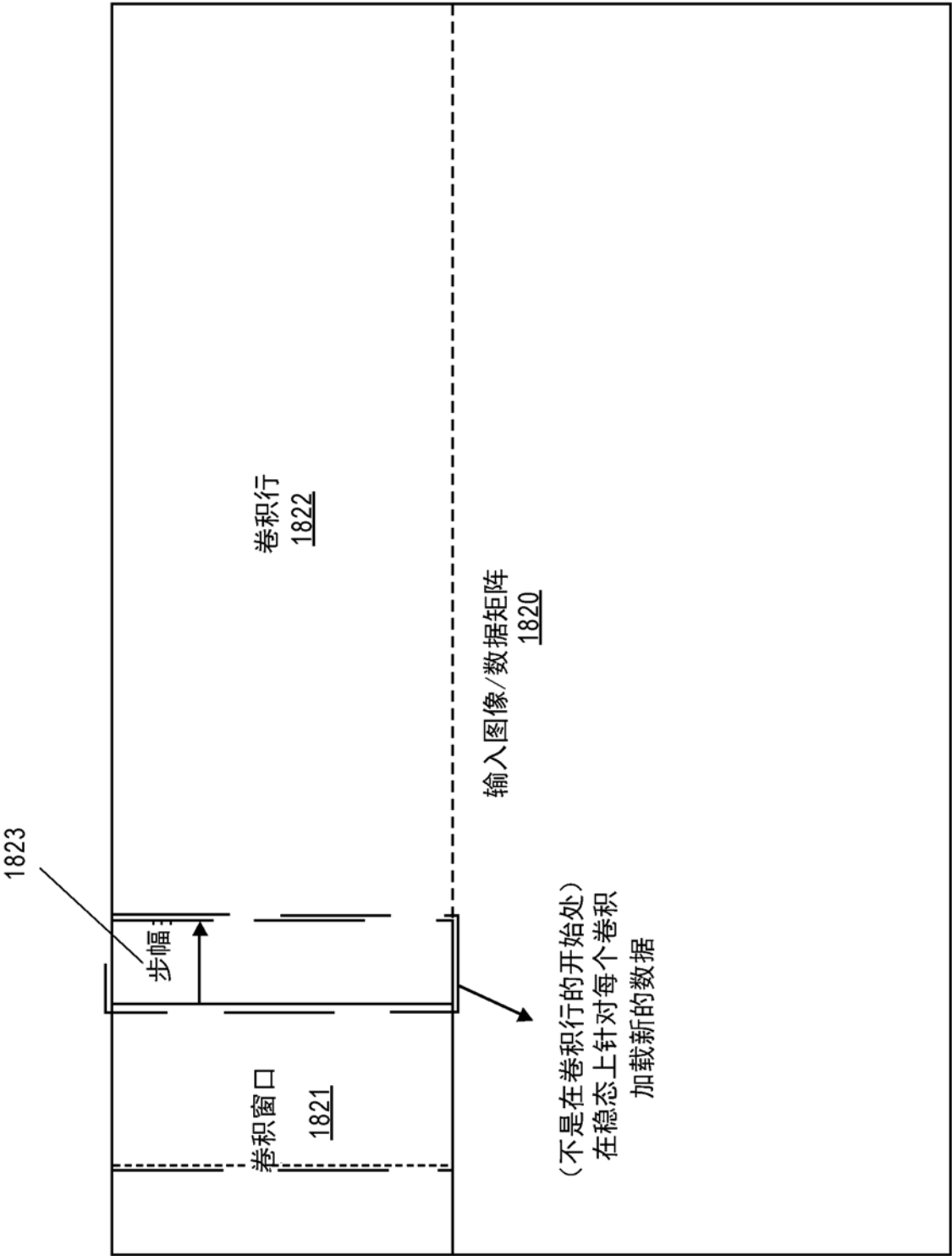


图 18B

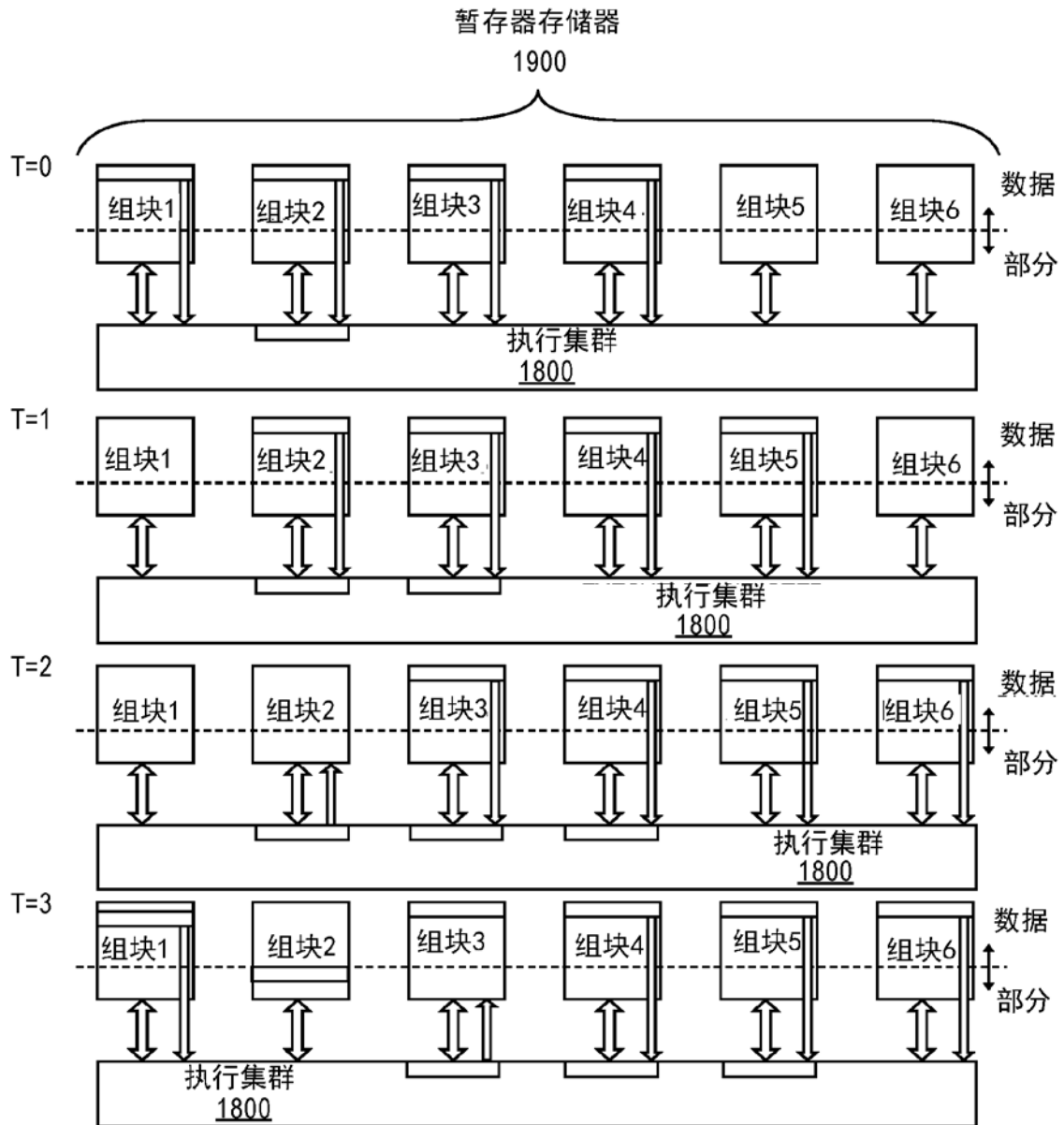


图 19

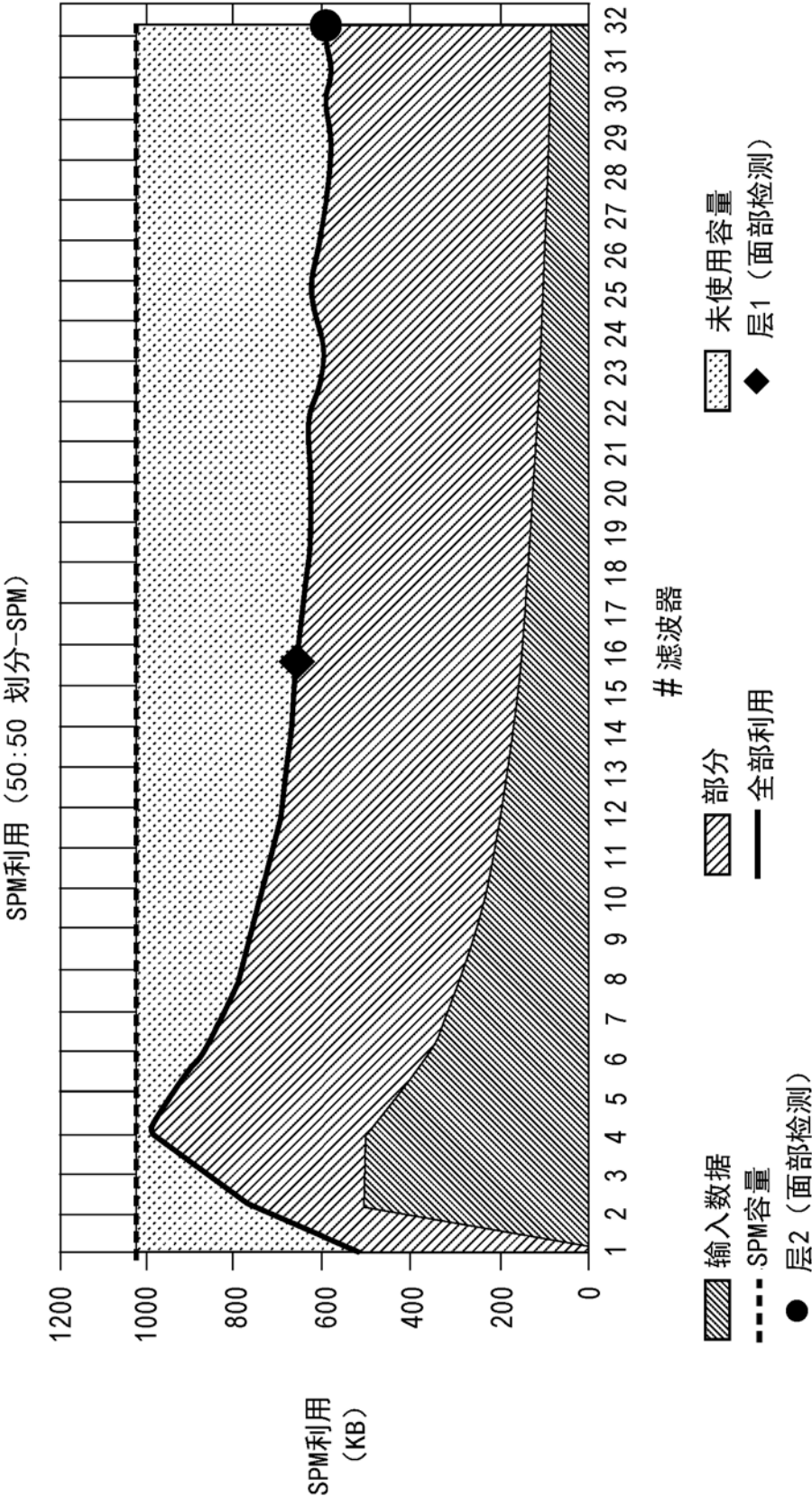


图 20A

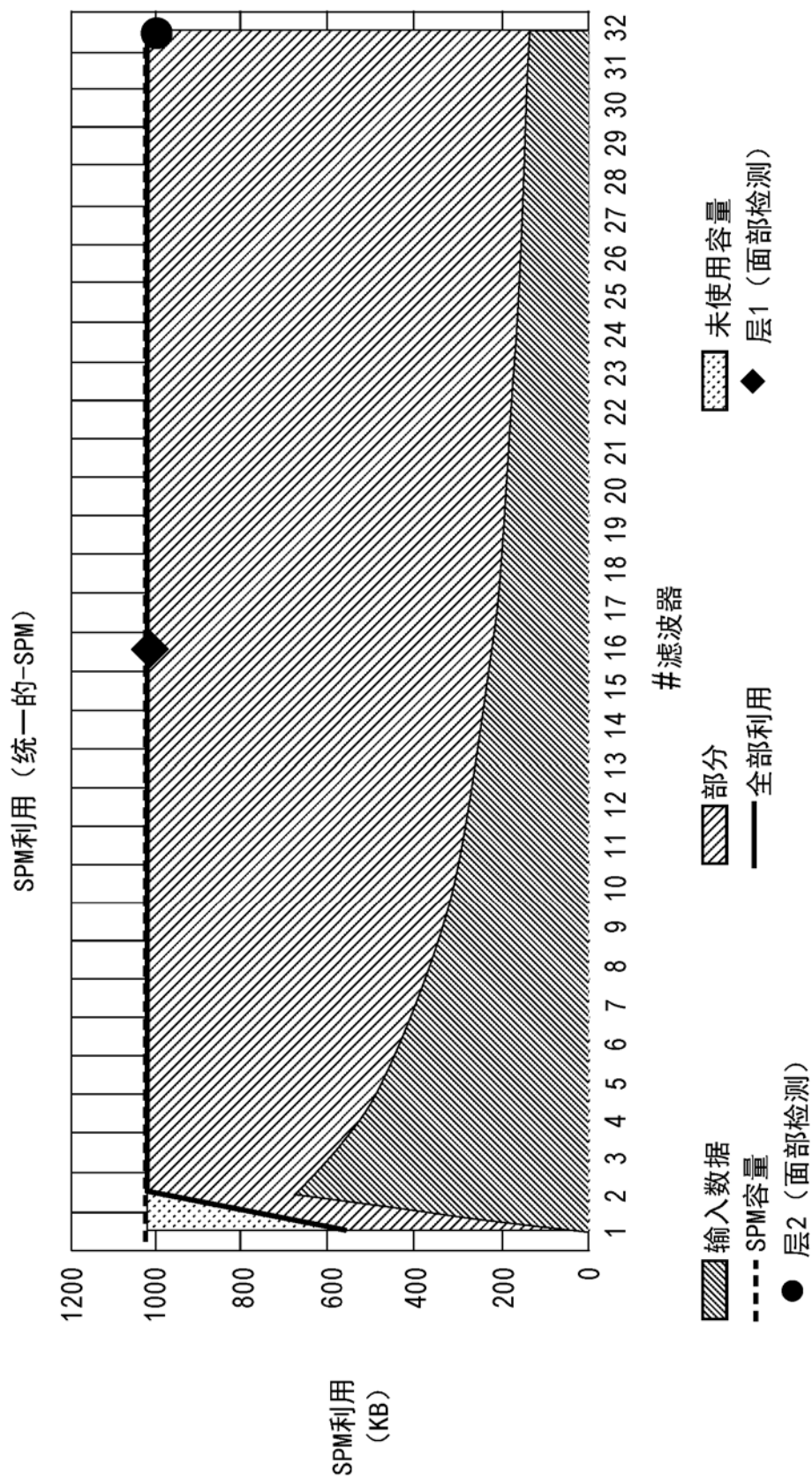


图 20B

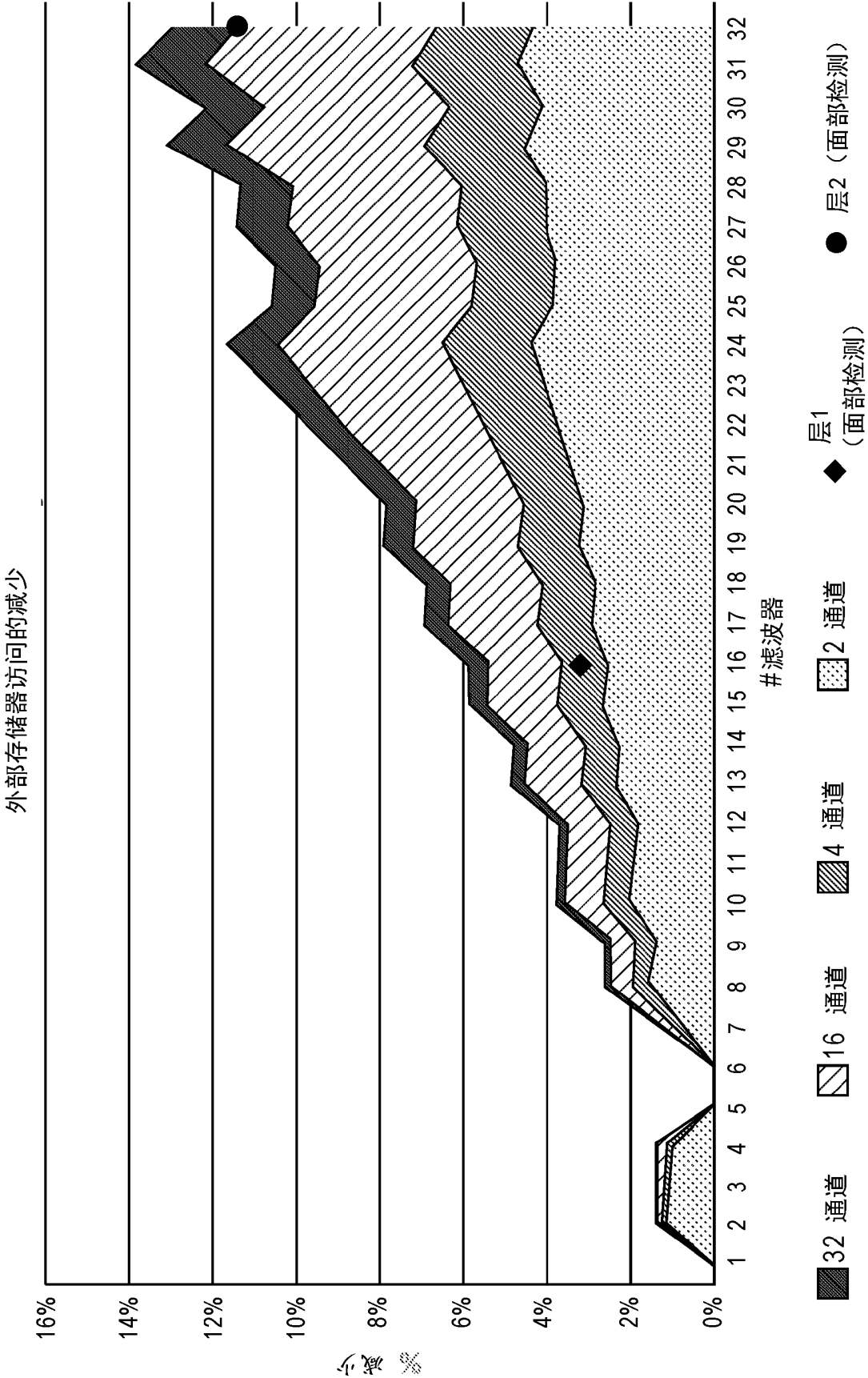


图 21

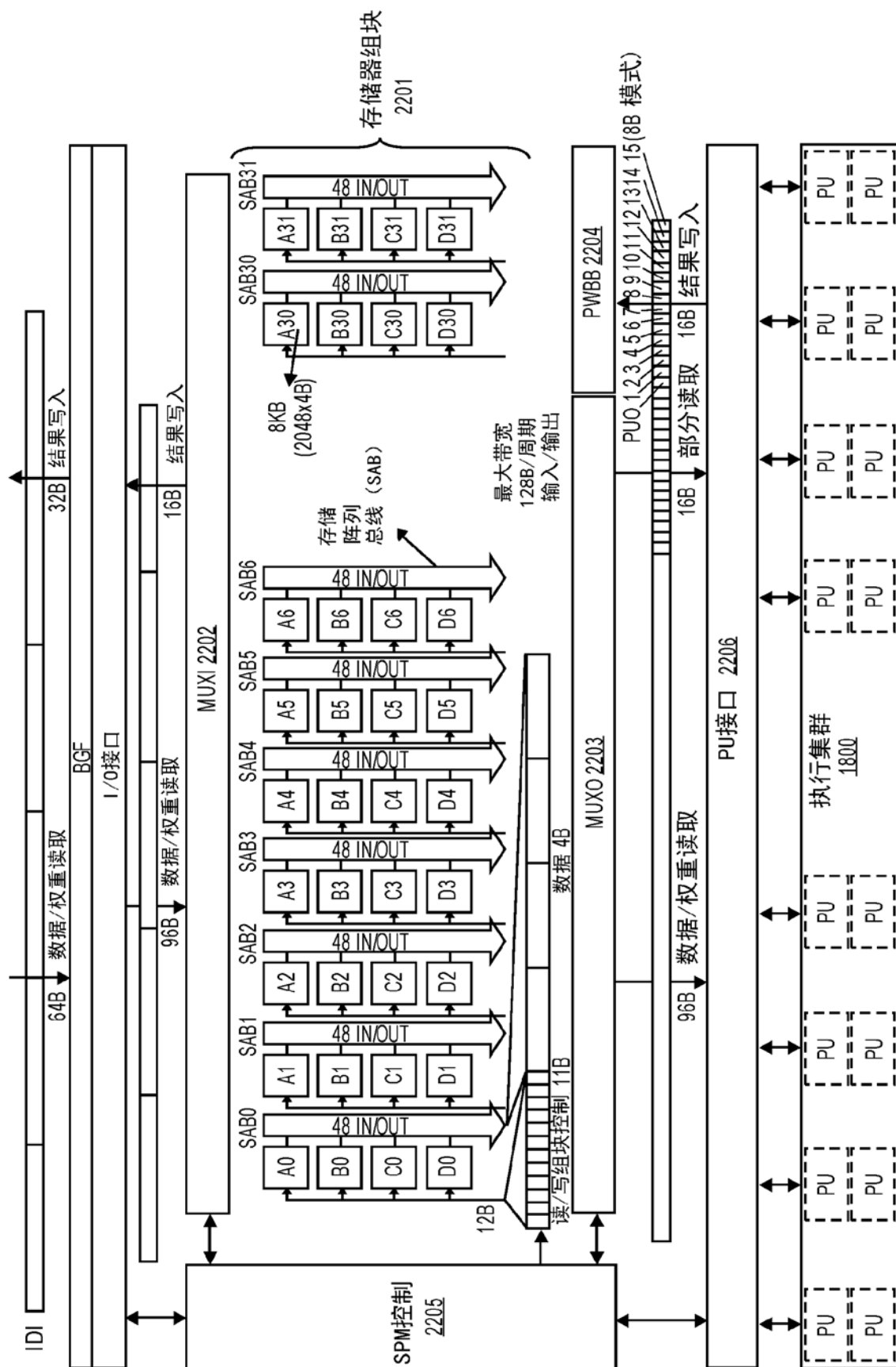


图 22

[illegible]

图 23

	0	1	2	3	4	5	6	7	16	17	31
A	0.0	2.0	4.0	6.0	8.0	10.0	12.0	14.0	32.0	34.0	62.0
	64.0	66.0	68.0	70.0	72.0	74.0	76.0	78.0	96.0	98.0	126.0
	128.0	130.0	132.0	134.0	136.0	138.0	140.0	142.0	160.0	162.0	190.0
	192.0	0.2	2.2	4.2	6.2	8.2	10.2	12.2	30.2	32.2	60.2
	62.2	64.2	66.2	68.2	70.2	72.2	74.2	76.2	94.2	96.2	124.2
	126.2	128.2	130.2	132.2	134.2	136.2	138.2	140.2	158.2	160.2	188.2
	190.2	192.2	0.4	2.4	4.4	6.4					
			64.4								
B	0.1	2.1	4.1	6.1	8.1	10.1	12.1	14.1	32.1	34.1	64.1
	64.1										
	192.1	0.3	2.3	4.3	6.3						
		64.3									
	190.3	192.3	0.5	2.5	4.5	6.5				160.3	188.3
			64.5								
C	1.0	3.0	5.0	7.0	9.0	11.0	13.0	15.0	33.0	35.0	63.0
	65.0										
	193.0	1.2	3.2	5.2	7.2						
		65.2									
		193.2	1.4	3.4							
			65.4								
D	1.1	3.1	5.1	7.1	9.1	11.1	13.1	15.1	33.1	35.1	63.1
	65.1	67.1	69.1	71.1	73.1	75.1	77.1	79.1			
	193.0	1.3	3.3	5.3	7.3						
		65.3									
		193.3	1.5	3.5	5.5	7.5					
			65.5								

图 24

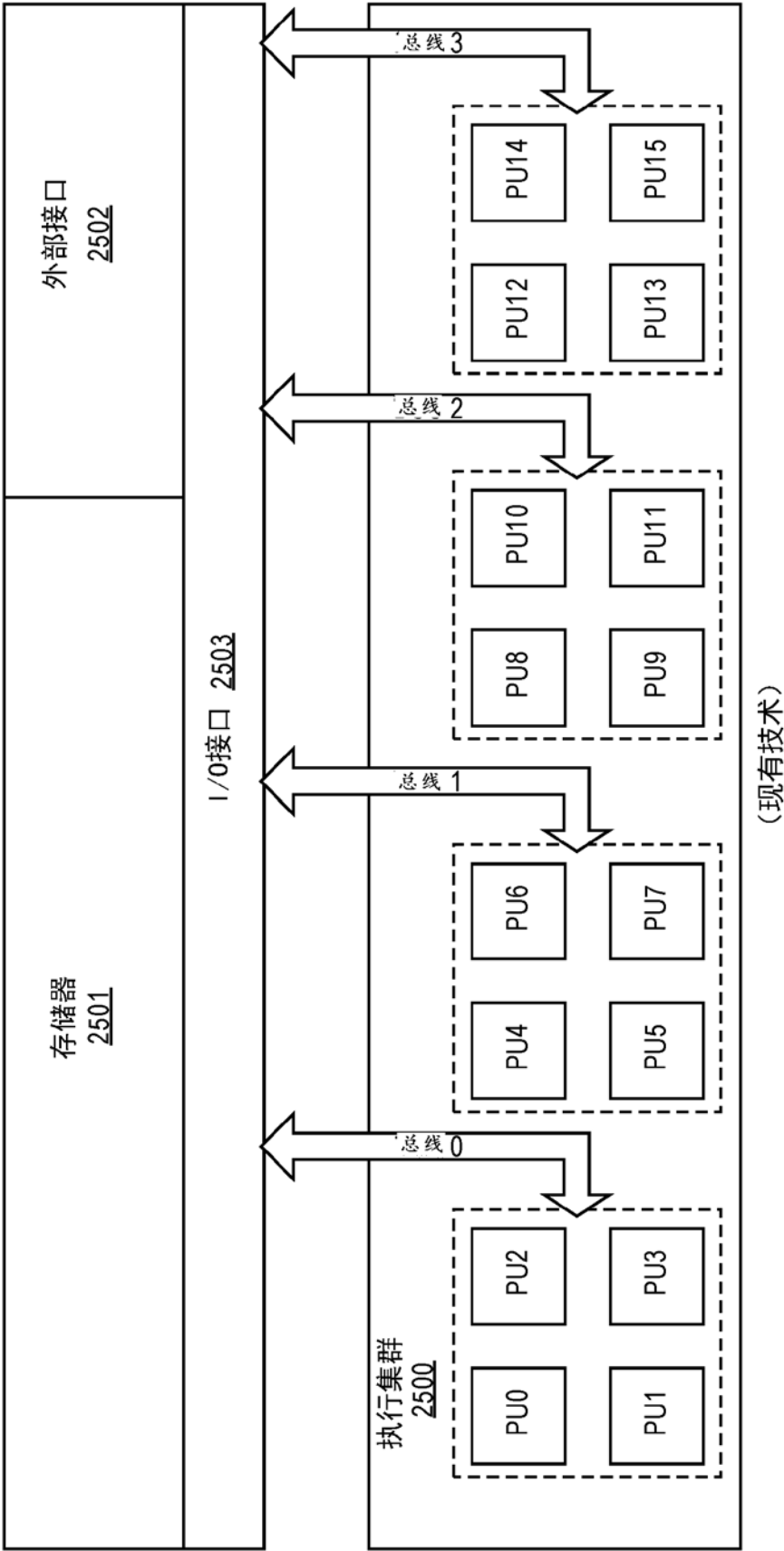


图 25

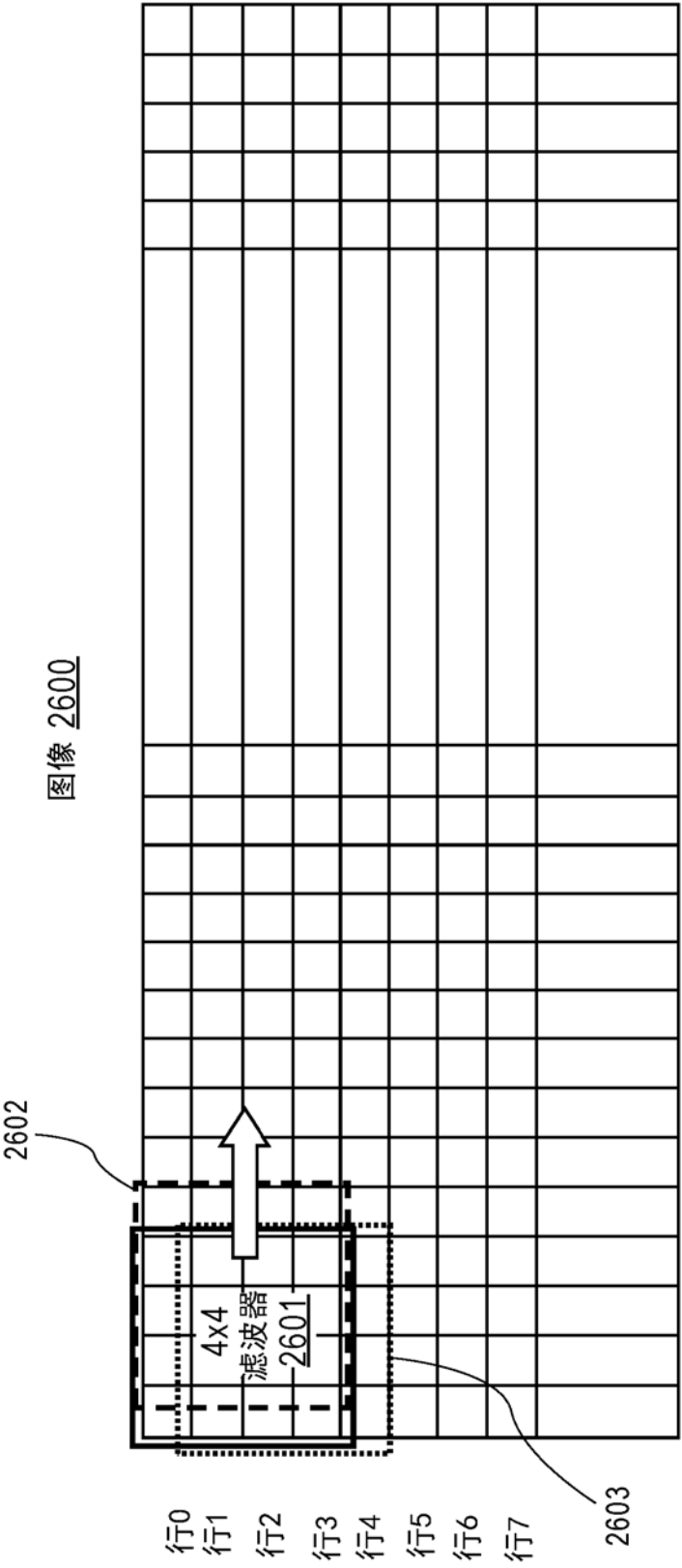


图 26

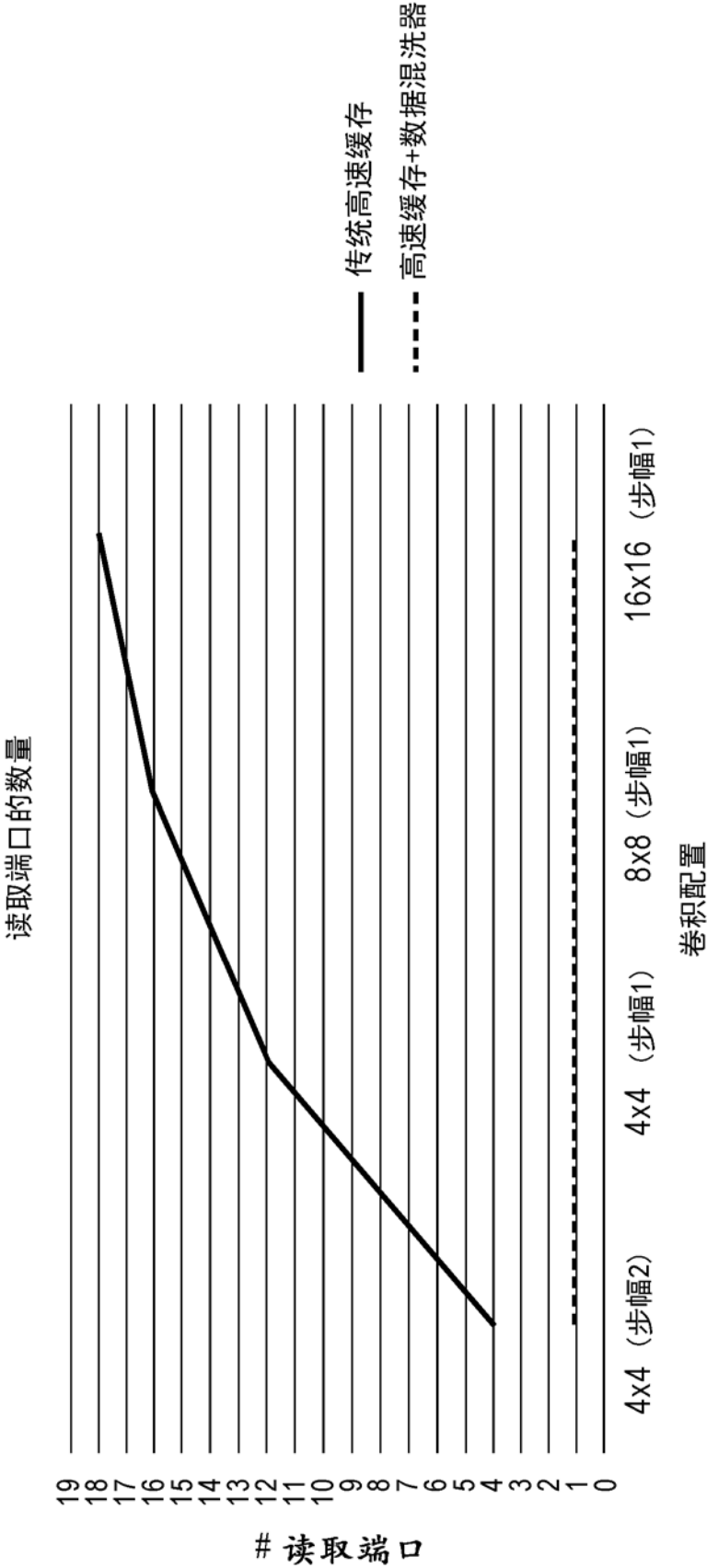


图 27

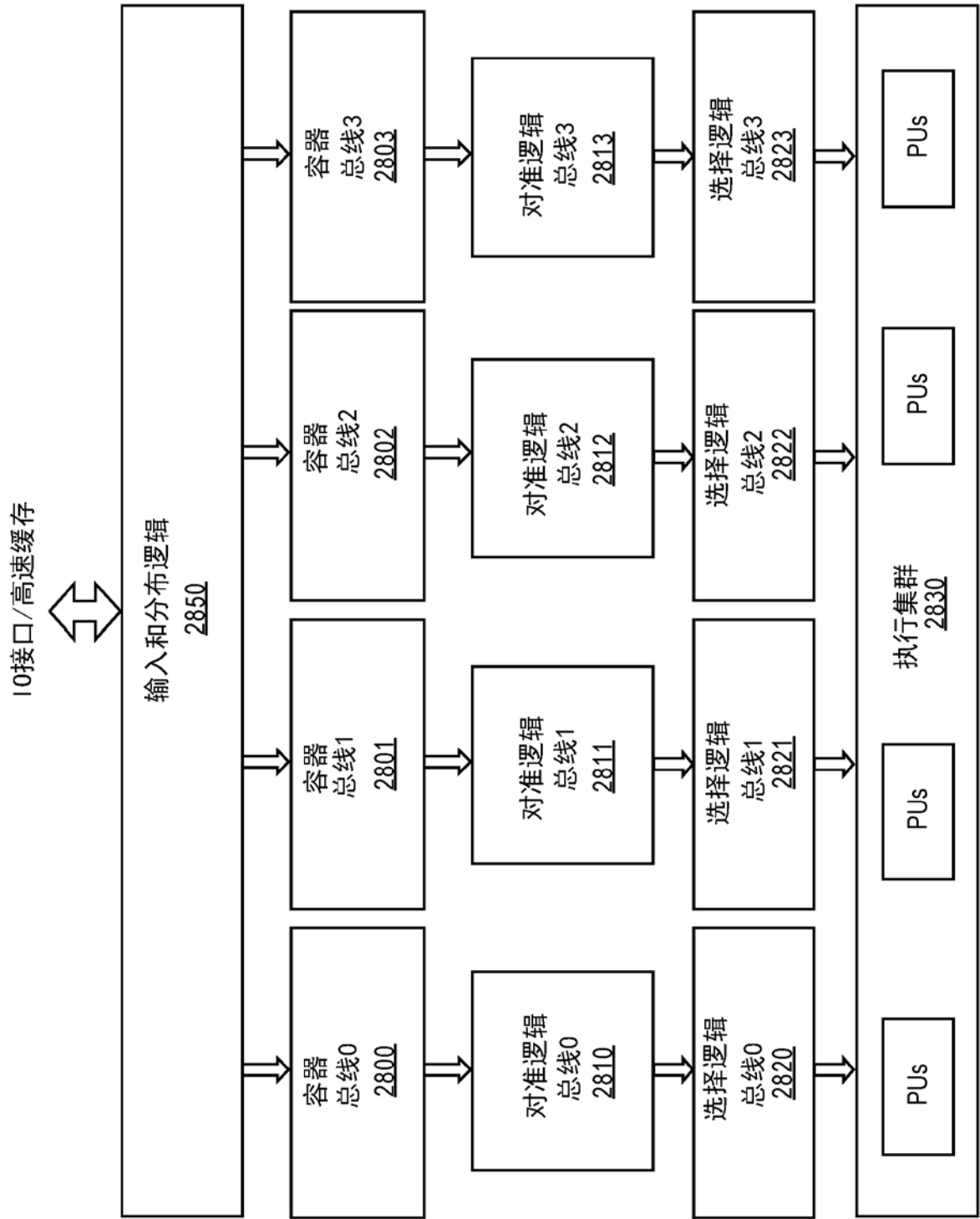


图 28

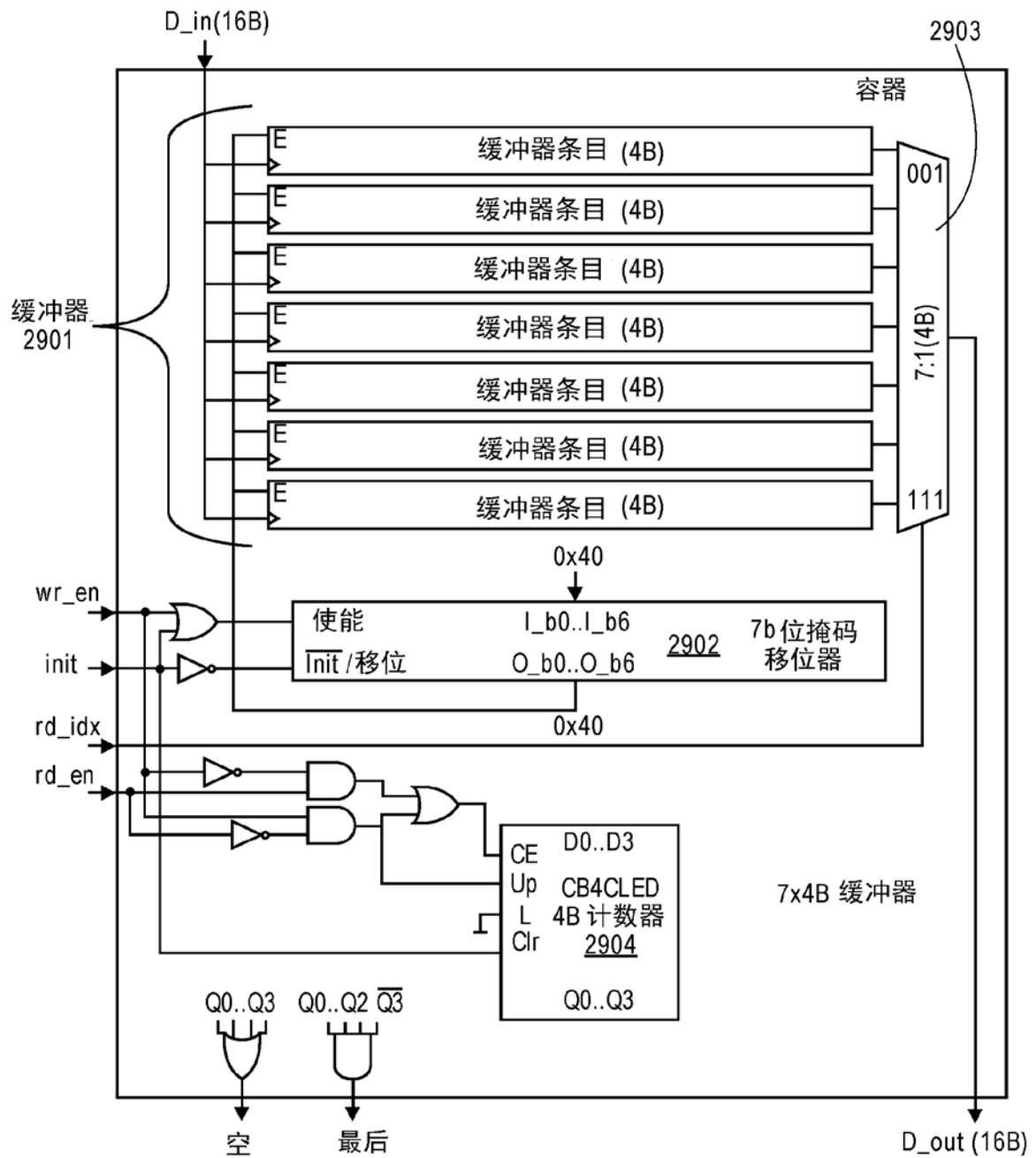


图 29

对准	AO ₀	AO ₁	AO ₂	AO ₃
4元素1B/1元素4B	O ₀	O ₁	O ₂	O ₃
1元素2B (位置0)	O ₀	O ₁	-	-
1元素2B (位置1)	O ₂	O ₃	-	-
1元素1B (位置0)	O ₀	-	-	-
1元素1B (位置1)	O ₁	-	-	-
1元素1B (位置2)	O ₂	-	-	-
1元素1B (位置3)	O ₃	-	-	-

图 30

输出模式	Output_Pos0	Output_Pos1	Output_Pos2	Output_Pos3
4B行模式	C _X _AO0	C _X _AO1	C _X _AO2	C _X _AO3
4B列模式	C _W _AO0	C _X _AO0	C _Y _AO0	C _Z _AO0
4B正方形模式 (2x2)	C _X _AO0	C _X _AO1	C _Y _AO0	C _Y _AO1

图 31

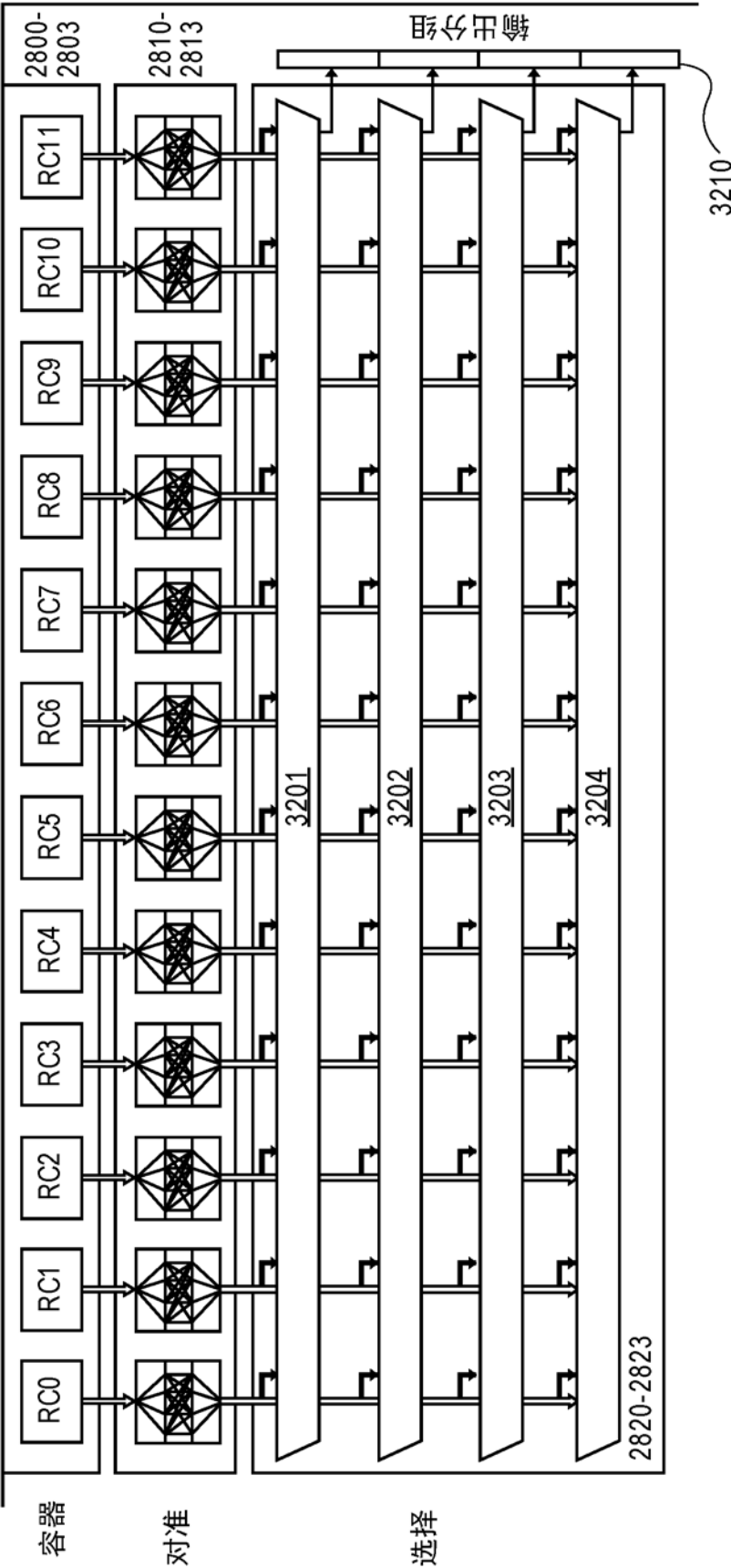
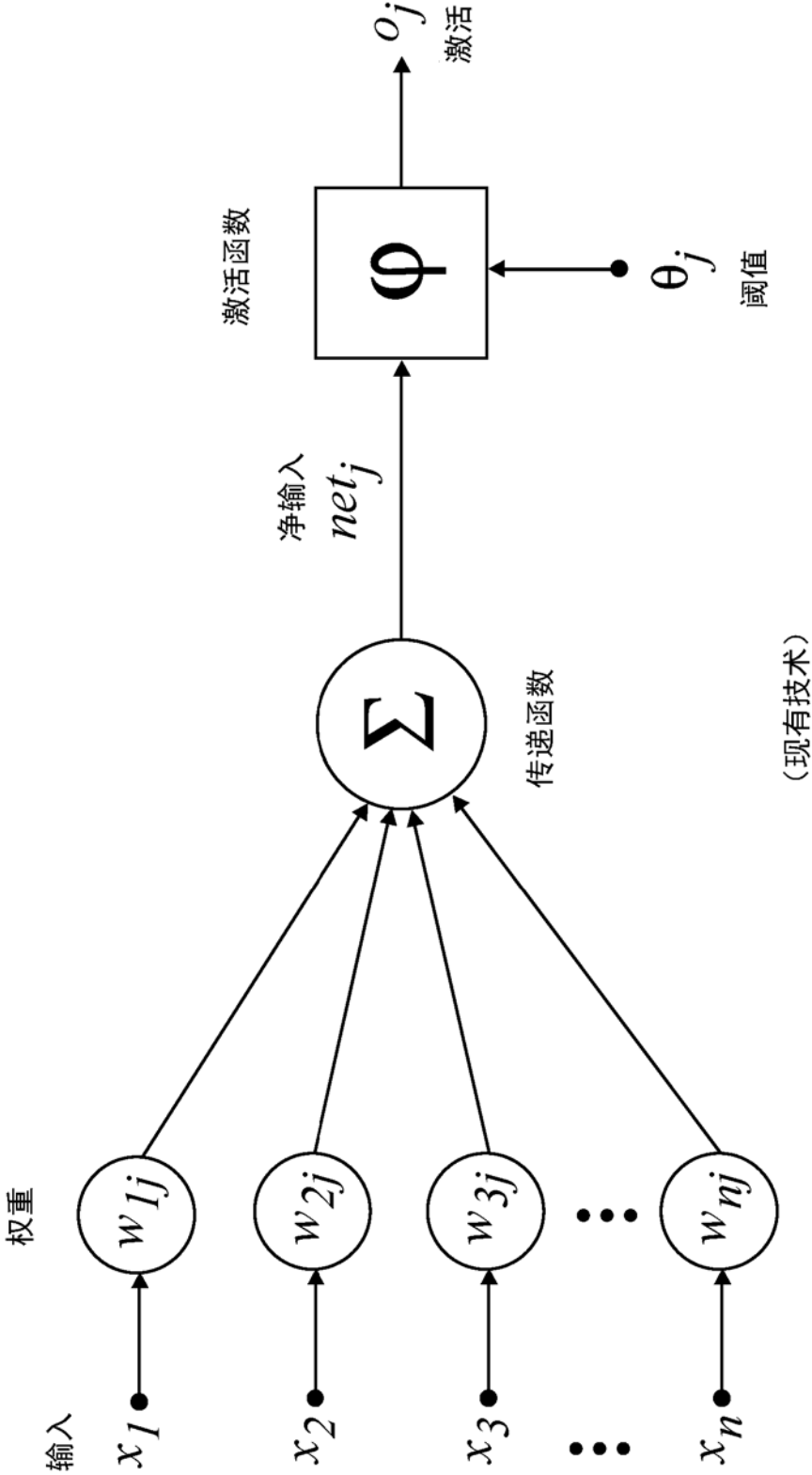


图 32



(现有技术)

图 33

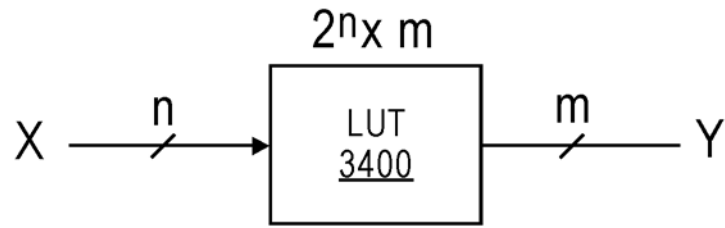


图 34A

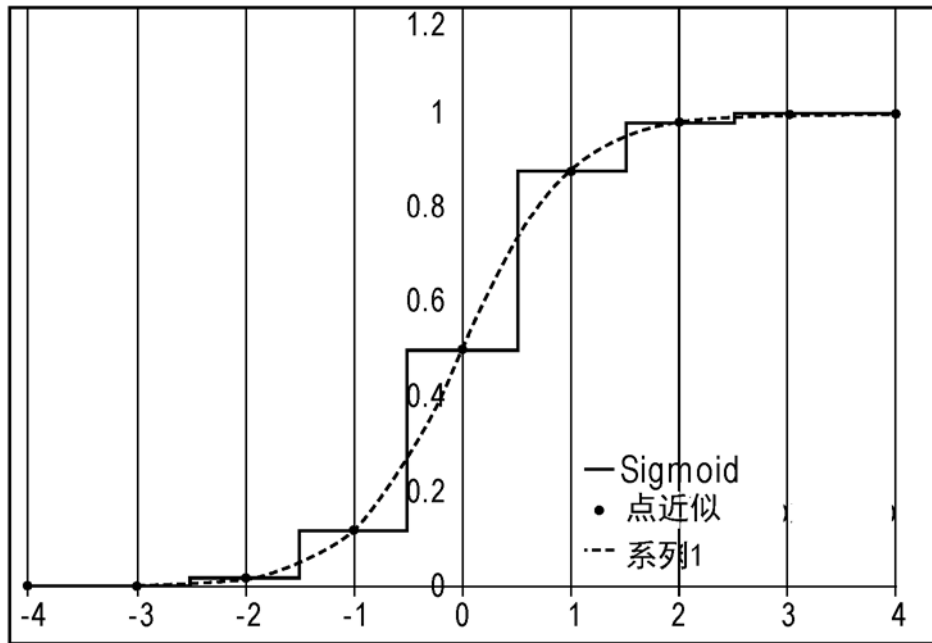


图 34B

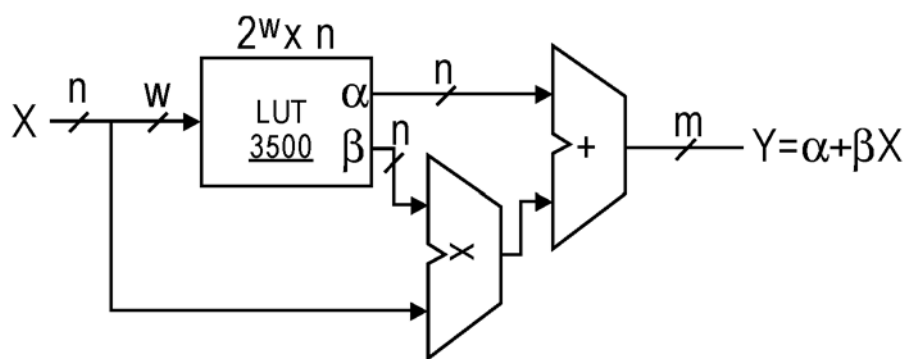


图 35A

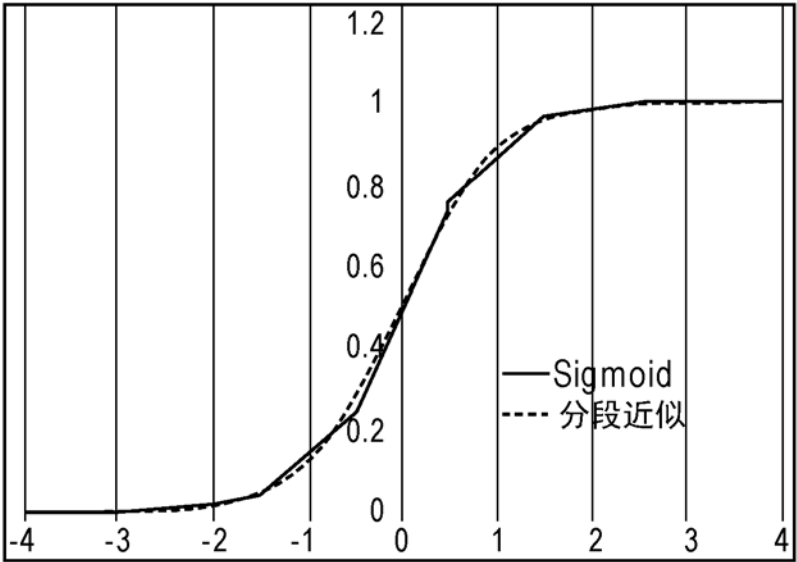


图 35B

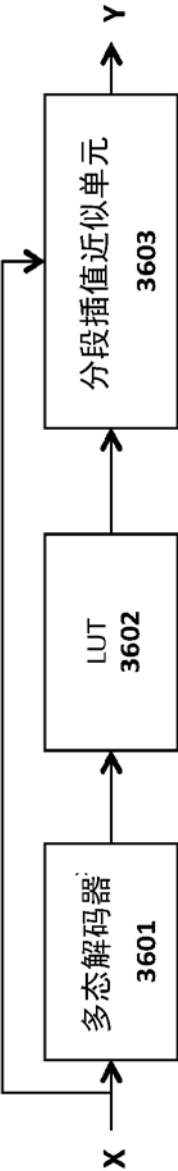


图 36

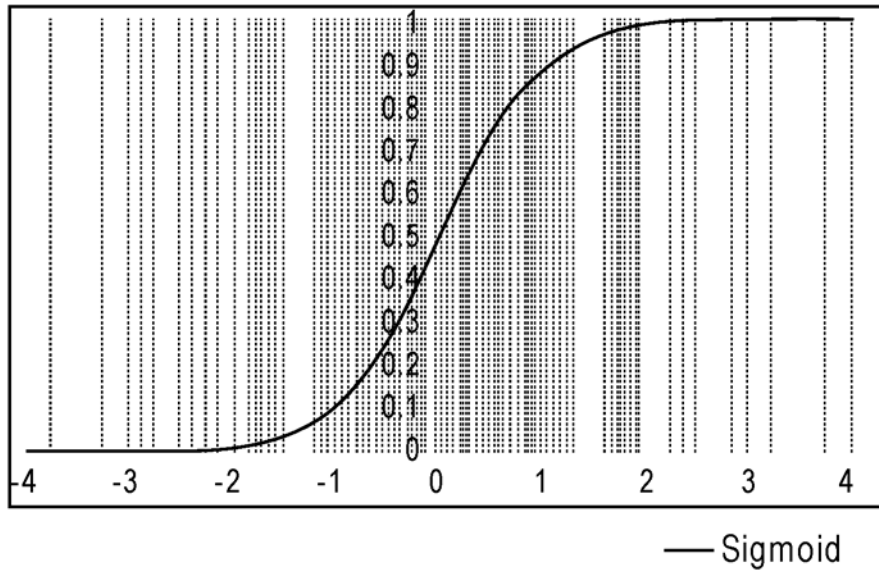


图 37A

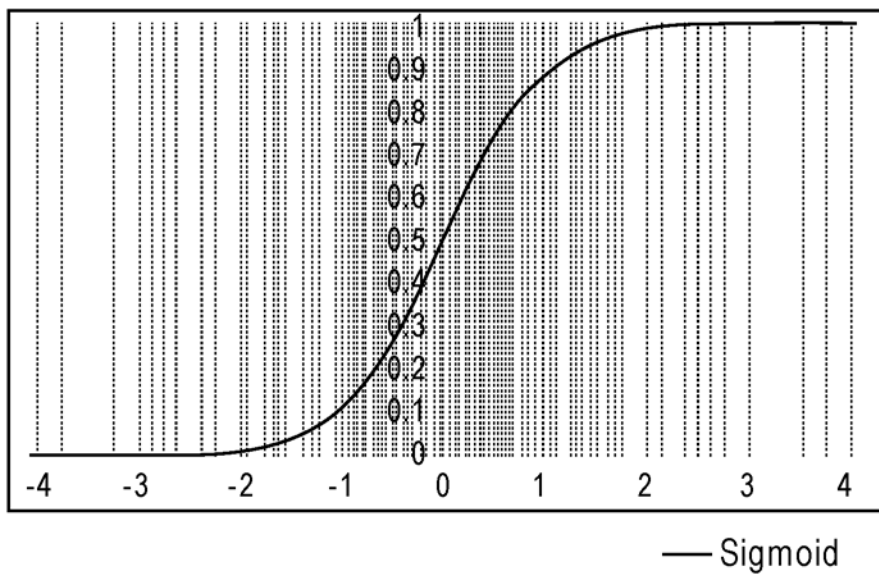


图 37B

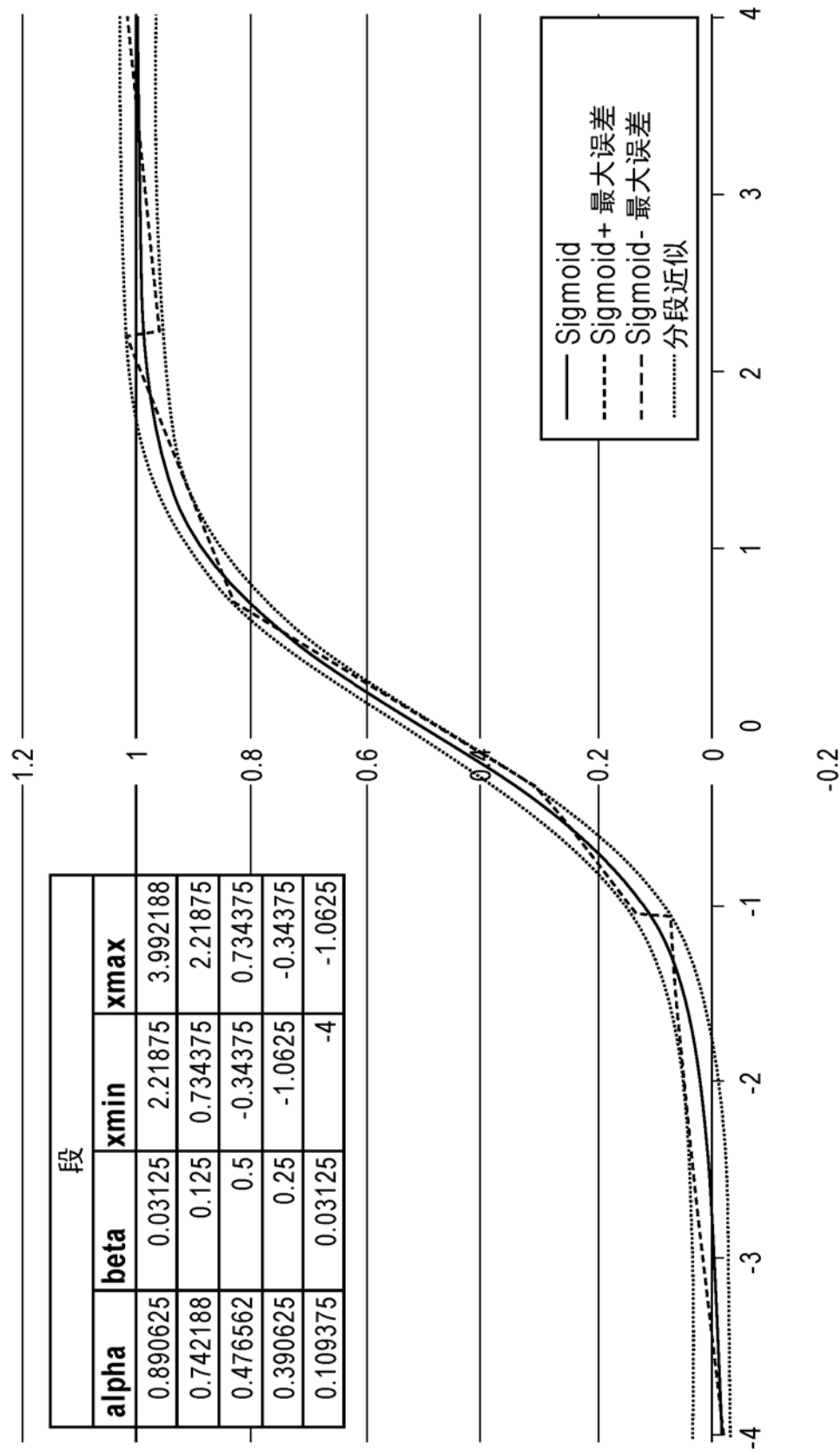


图 38

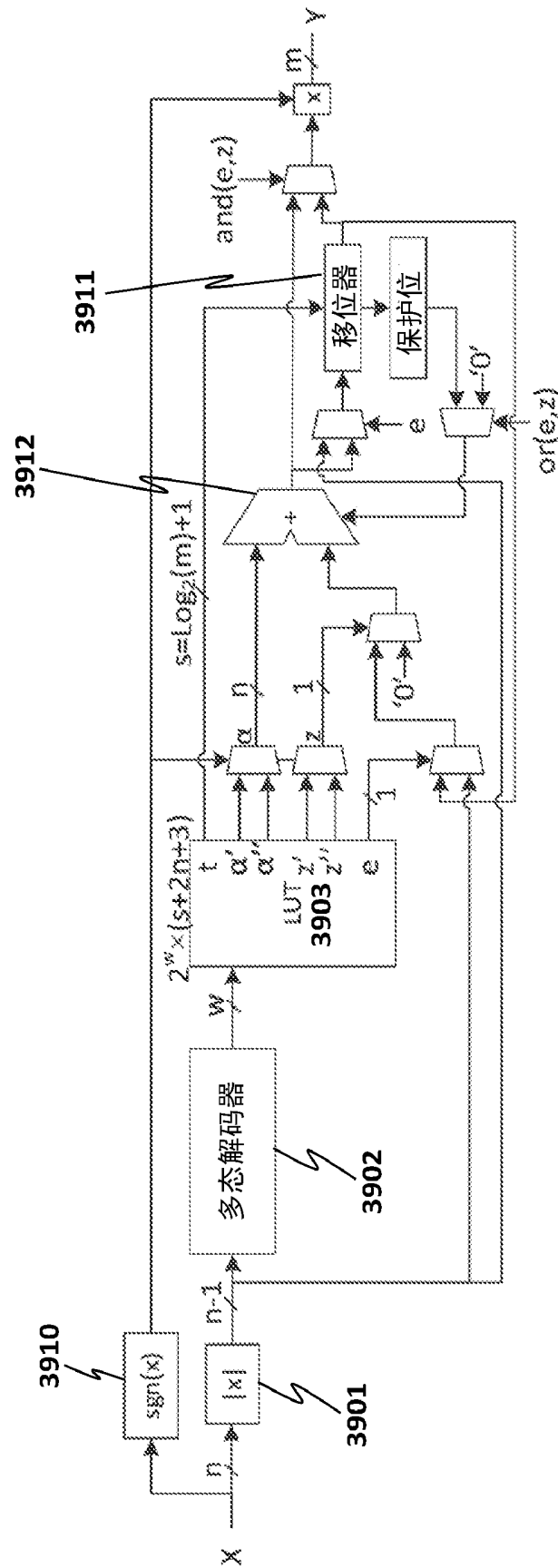


图 39

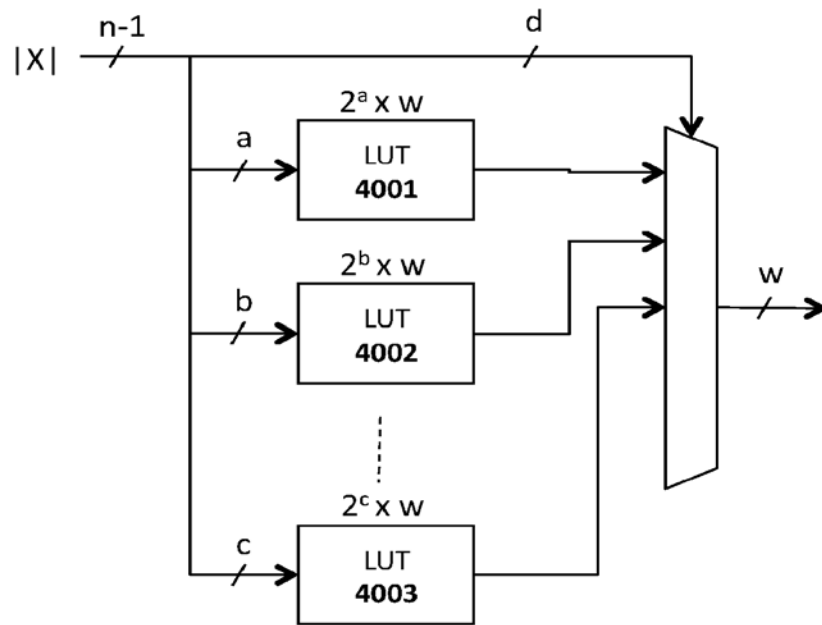


图 40