(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 110537194 A
(43)申请公布日 2019.12.03

(21)申请号 201880025488.X

(22)申请日 2018.04.13

(30)优先权数据
62/486,432 2017.04.17 US
15/950,550 2018.04.11 US

(85)PCT国际申请进入国家阶段日
2019.10.16

(86)PCT国际申请的申请数据
PCT/US2018/027674 2018.04.13

(87)PCT国际申请的公布数据
WO2018/194939 EN 2018.10.25

(71)申请人 微软技术许可有限责任公司
地址 美国华盛顿州

(72)发明人 C·B·麦克布赖德
A·A·安巴德卡 K·D·塞多拉
G·彼得 L·M·瓦尔
B·博布罗夫

(74)专利代理机构 北京市金杜律师事务所 11256
代理人 赵林琳 郭星

(51)Int.Cl.
G06N 3/063(2006.01)
G06F 9/48(2006.01)

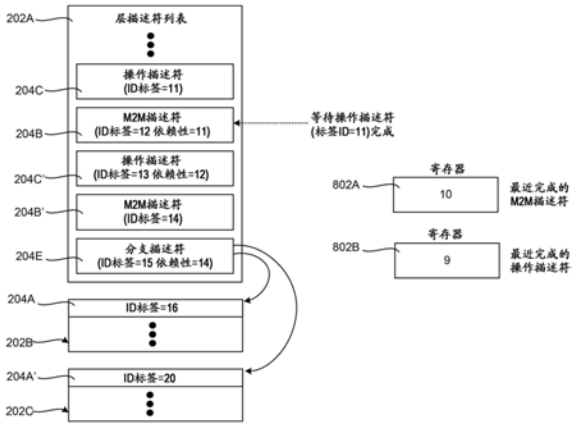
权利要求书2页 说明书18页 附图15页

(54)发明名称

被配置用于层和操作防护和依赖性管理的功率高效的深度神经网络模块

(57)摘要

一种深度神经网络(DNN)处理器被配置为执行层描述符列表中的层描述符。描述符定义用于由DNN处理器执行DNN的前向传递的指令。层描述符也可以用于管理通过DNN模块的描述符流。例如,层描述符可以定义对其他描述符的依赖性。定义依赖性的描述符将不执行,直到它们所依赖的描述符已经完成。层描述符也可以定义“防护”或“屏障”功能,该功能可以用于阻止对上游层描述符的处理,直到所有下游层描述符的处理完成。防护位保证了在处理具有要声明的防护的层描述符之前,DNN处理流水线中没有其他层描述符。



CN 110537194 A

1.一种神经网络处理器,包括:

存储器设备,存储包括针对神经网络的第一层描述符的层描述符列表,所述第一层描述符指定针对所述第一层描述符的执行所依赖的第二层描述符的标识符(ID);

硬件寄存器,存储最近完成的层描述符的ID;以及

控制器,被配置为:

确定被存储在所述硬件寄存器中的所述最近完成的层描述符的所述ID是否小于所述第二层描述符的ID,

响应于确定所述最近完成的层描述符的所述ID不小于所述第二层描述符的所述ID,使得所述神经网络处理器执行所述第一层描述符,以及

响应于确定所述最近完成的层描述符的所述ID小于所述第二层描述符的所述ID,使得所述神经网络处理器暂缓所述第一层描述符的执行。

2.根据权利要求1所述的神经网络处理器,其中所述第一层描述符的执行被暂缓,直到被存储在所述硬件寄存器中的所述最近完成的层描述符的所述ID等于所述第二层描述符的所述ID。

3.根据权利要求1所述的神经网络处理器,其中所述神经网络处理器还被配置为执行所述层描述符列表中的层描述符,并且将所述最近完成的层描述符的所述ID存储在所述硬件寄存器中。

4.根据权利要求1所述的神经网络处理器,其中标识符以单调递增顺序被分配给所述层描述符列表中的所述层描述符。

5.根据权利要求1所述的神经网络处理器,其中所述第一层描述符或所述第二层描述符包括以下各项中的一项:存储器到存储器移动(M2M)描述符、操作描述符、主机通信描述符、配置描述符、分支描述符、或者同步描述符。

6.根据权利要求1所述的神经网络处理器,其中所述层描述符列表包括指定防护操作的第三层描述符,并且其中所述控制器还被配置为:

至少部分地基于所述最近完成的层描述符的所述ID,确定所述层描述符列表中具有ID小于所述第三层描述符的ID的所有层描述符是否已经完成执行,

响应于确定所述层描述符列表中ID小于所述第三层描述符的所述ID的所有层描述符已经完成执行,使得所述神经网络处理器执行所述第三层描述符,以及

响应于确定所述层描述符列表中具有ID小于所述第三层描述符的所述ID的所有层描述符尚未完成执行,使得所述神经网络处理器暂缓所述第三层描述符的执行。

7.一种神经网络处理器,包括:

存储器设备,存储包括针对神经网络的第一层描述符的层描述符列表;

硬件寄存器,存储最近完成的层描述符的标识符(ID);以及

控制器,被配置为:

至少部分地基于所述最近完成的层描述符的所述ID,确定所述层描述符列表中具有ID小于所述第一层描述符的所述ID的所有层描述符是否已经完成执行,

响应于确定所述层描述符列表中具有ID小于所述第一层描述符的所述ID的所有层描述符已经完成执行,使得所述神经网络处理器执行所述第一层描述符,以及

响应于确定所述层描述符列表中具有ID小于所述第一层描述符的所述ID的所有层描

述符尚未完成执行，使得所述神经网络处理器暂缓所述第一层描述符的执行。

8.根据权利要求7所述的神经网络处理器，其中所述第一层描述符的执行被暂缓，直到所述描述符列表中具有ID小于所述第一层描述符的所述ID的所有层描述符已经完成执行。

9.根据权利要求7所述的神经网络处理器，其中所述神经网络处理器还被配置为执行所述层描述符列表中的层描述符，并且将所述最近完成的层描述符的所述ID存储在所述硬件寄存器中。

10.根据权利要求7所述的神经网络处理器，其中标识符以单调递增顺序被分配给所述层描述符列表中的所述层描述符。

11.根据权利要求7所述的神经网络处理器，其中所述第一层描述符包括以下各项中的一项：存储器到存储器移动(M2M)描述符、操作描述符、主机通信描述符、配置描述符、分支描述符、或者同步描述符。

12.一种计算机实现的方法，包括：

通过神经网络模块访问包括针对神经网络的第一层描述符的层描述符列表，所述第一层描述符指定针对所述第一层描述符的执行所依赖的第二层描述符的标识符(ID)；

确定最近完成的层描述符的标识符(ID)是否小于所述第二层描述符的ID；

响应于确定所述最近完成的层描述符的所述ID不小于所述第二层描述符的所述ID，通过所述神经网络模块执行所述第一层描述符；以及

响应于确定所述最近完成的层描述符的所述ID小于所述第二层描述符的所述ID，暂缓通过所述神经网络模块对所述第一层描述符的执行。

13.根据权利要求12所述的计算机实现的方法，其中所述第一层描述符的执行被暂缓，直到所述最近完成的层描述符的所述ID等于所述第二层描述符的所述ID。

14.根据权利要求12所述的计算机实现的方法，还包括：

执行所述层描述符列表中的层描述符；以及

将所述最近完成的层描述符的所述ID存储在所述神经网络模块的硬件寄存器中。

15.根据权利要求12所述的计算机实现的方法，其中所述第一层描述符和所述第二层描述符包括以下各项中的一项：存储器到存储器移动(M2M)描述符、操作描述符、主机通信描述符、配置描述符、分支描述符、或者同步描述符。

被配置用于层和操作防护和依赖性管理的功率高效的深度神经网络模块

背景技术

[0001] 在诸如人脑的生物神经系统中，在信息处理和通信模式之后，对深度神经网络(“DNN”)进行松散建模。DNN可以用来解决复杂的分类问题，诸如但不限于对象检测、语义标记和特征提取。结果，DNN形成了很多人工智能(“AI”)应用的基础，诸如计算机视觉、语音识别和机器翻译。在很多领域，DNN都可以达到或甚至超过人类的准确性。

[0002] DNN的高级性能源于它们在对大数据集使用统计学习以获取输入空间的有效表示之后，从输入数据中提取高级特征的能力。但是，DNN的优越性能是以高计算复杂度为代价的。诸如图形处理单元(“GPU”)的高性能通用处理器通常用于提供很多DNN应用所需要的高水平计算性能。

[0003] 尽管诸如GPU的通用处理器可以为实现DNN提供高水平的计算性能，但是这些类型的处理器通常不适合在低功耗至关重要的计算设备中长时间执行DNN操作。例如，诸如GPU的通用处理器可能不适合在电池供电的便携式设备(诸如智能手机或替代/虚拟现实(AR/VR)设备)中执行长时间运行的DNN任务，其中需要降低功耗以延长电池寿命。

[0004] 在执行诸如人体移动的检测的连续DNN任务时，降低功耗在诸如以太网供电(“POE”)安全相机的非电池供电的设备中也很重要。在该特定示例中，POE交换机只能提供有限的电量，并且减少诸如安全相机的POE设备的功耗允许使用提供更少电量的POE交换机。

[0005] 与通用处理器相比，已经开发出可以在降低功耗的同时提供高性能DNN处理的专用集成电路(“ASIC”)。尽管该领域取得了进步，但仍然需要提高执行DNN处理的ASIC的性能，并且降低执行DNN处理的ASIC的功耗，尤其是在低功耗至关重要的计算设备中。

[0006] 关于这些和其他技术挑战，提出了本文中进行的公开内容。

发明内容

[0007] 公开了一种神经网络模块或处理器，其可以以减少DNN计算执行期间的等待时间的方式，来执行层描述符列表中的层描述符(本文中可以称为“描述符”)。所公开的神经网络模块还可以实现用于管理通过DNN模块的描述符流的功能。通过所公开的技术的实现，可以优化通过DNN模块的描述符流，从而使得DNN模块能够更快地完成其处理。更快地完成处理可以使得DNN模块能够更早地关闭，从而节省了功率。还可以通过所公开的主题的实现来实现本文中未具体提及的其他技术益处。

[0008] 为了实现上面简要提及的技术益处以及潜在的其他益处，公开了一种DNN模块，该DNN模块能够取回和执行层描述符列表中包含的描述符以实现DNN。层描述符列表(本文中可以称为“描述符列表”)由诸如编译器的软件预编译，并且包括用于由DNN模块执行神经网络的前向传递的指令。

[0009] 描述符列表中的描述符也可以用来配置DNN模块的操作的各方面，包括用于实现神经网络的DNN模块中的神经元的配置。在一个实施例中，描述符列表存储在用作用于DNN

模块的主机的计算设备的主存储器中，并且由DNN模块加载以进行即时执行。

[0010] 描述符列表可以包括几种类型的DNN层描述符(本文中可以称为“描述符”)：存储器到存储器移动(“M2M”)描述符；操作描述符；主机通信描述符；配置描述符；分支描述符；以及同步描述符。这些描述符类型中的每个的配置和操作将在下面详细描述。

[0011] 如上所述，所公开的DNN模块还可以被配置用于层和操作防护和依赖性管理。特别地，上述层描述符可以用于管理通过DNN模块的描述符流。例如，层描述符可以定义对其他描述符的依赖性。每个描述符可以包括在整个处理流水线中随描述符一起携带的唯一标识(“ID”)标签。当描述符完成处理之后，指示针对描述符类型的最近完成的描述符的描述符ID放置在寄存器中。每种描述符类型可以具有用于跟踪与完成的层描述符相关联的ID的单独的寄存器。

[0012] 如果层描述符指定对另一层描述符的依赖性，则DNN模块将最后完成的描述符的层ID与当前描述符所依赖的描述符的ID进行比较。如果最后完成的描述符ID高于当前描述符所依赖的描述符的ID，则当前描述符可以开始处理，因为依赖性已经满足。但是，如果最后完成的描述符的ID低于当前描述符所依赖的描述符的ID，则当前描述符将暂缓，直到其依赖性满足。

[0013] 层描述符还可以定义“防护”或“屏障”功能，该功能用于防止对上游层描述符的处理，直到所有下游层描述符的处理完成。例如，在一个实施例中，M2M描述符可以通过防护位来指定不对其进行处理，直到它之前的所有描述符完成并且接收到它们的完成存储的写入响应。防护位保证了在处理具有要声明的防护的层描述符之前，DNN处理流水线中没有其他层描述符。

[0014] 如将在下面更详细描述的，M2M描述符与操作描述符分开地被缓存，并且可以在明确地设置依赖性的情况下尽快被执行。结果，可以减少通常在获取数据以供DNN模块中的神经元进行处理时所使得的等待时间，因此，神经元可以比其他方式更快地完成其处理。然后，DNN模块可以比其他方式更早地断电，从而节省了功率。

[0015] 如下面也将更详细描述的，所公开的技术可以使得能够优化通过DNN模块的描述符流，从而使得DNN模块能够更快地完成其处理，从而节省功率。还可以通过所公开的技术的实现来实现本文中未具体确定的其他技术益处。

[0016] 应当理解，上述主题可以被实现为计算机控制的装置、计算机实现的方法、计算设备或诸如计算机可读介质的制品。通过阅读以下“具体实施方式”并且查看相关附图，这些和各种其他特征将变得很清楚。

[0017] 提供本“发明内容”以便以简化的形式介绍下面在“具体实施方式”中进一步描述的所公开的技术的一些方面。本“发明内容”既不旨在标识所要求保护的主题的关键特征或必要特征，也不旨在用于限制所要求保护的主题的范围。此外，所要求保护的主题不限于能够解决在本公开的任何部分中指出的任何或所有缺点的实现。

附图说明

[0018] 图1是示出根据一个实施例的可以实现本文中公开的技术的DNN模块的配置和操作的各方面的计算架构图；

[0019] 图2是示出根据一个实施例的用于创建和执行层描述符列表的说明性环境的各方

面的计算系统图；

[0020] 图3是示出根据一个实施例的图2所示的层描述符列表的附加方面的数据结构图；

[0021] 图4A是示出根据一个实施例的操作描述符的配置的各方面的数据结构图；

[0022] 图4B是示出根据一个实施例的M2M描述符的配置的各方面的数据结构图；

[0023] 图5是示出根据一个实施例的关于能够执行层描述符列表中的描述符的DNN模块的配置的细节的计算架构图；

[0024] 图6是示出根据本文中公开的一个实施例的例程的流程图，该例程示出了用于执行层描述符列表中的描述符的DNN模块的操作的各方面；

[0025] 图7是示出根据一个实施例的DNN模块的配置和操作的附加细节的计算架构图，DNN模块被配置用于执行层描述符列表中的描述符，并且提供层和操作防护和依赖性管理；

[0026] 图8是示出根据一个实施例的用于执行防护操作的DNN模块的操作的各方面的数据结构图；

[0027] 图9A至图9C是示出根据一个实施例的用于执行操作依赖性的DNN模块的操作的各方面的数据结构图；

[0028] 图10是示出根据本文中公开的一个实施例的例程的流程图，该例程示出了用于提供层和操作防护和依赖性管理的DNN模块的操作的各方面；

[0029] 图11是示出根据一个实施例的可以用作本文中提出的用于DNN模块的应用主机的计算设备的说明性计算机硬件和软件架构的计算机架构图；以及

[0030] 图12是示出根据本文中呈现的各种实施例的可以在其中实现所公开的技术的各方面的分布式计算环境的网络图。

具体实施方式

[0031] 以下详细描述涉及一种神经网络模块，其可以执行层描述符列表中的描述符以有效地执行DNN计算。如上所述，DNN模块还可以实现用于管理通过DNN模块的描述符流的功能。通过所公开的技术的实现，可以优化通过DNN模块的描述符流，从而使得DNN模块能够更快地完成其处理。更快地完成处理可以使得DNN模块能够更早地关闭，从而节省了功率。还可以通过所公开的主题的实现来实现本文中未具体提及的其他技术益处。

[0032] 虽然本文中描述的主题是在硬件DNN模块的一般上下文中介绍的，但本领域技术人员将认识到，可以结合其他类型的计算系统和模块来执行其他实现。本领域技术人员还将认识到，本文中描述的主题可以与其他计算机系统配置一起实践，包括手持式设备、多处理器系统、基于微处理器的或可编程的消费电子产品、嵌入在设备(诸如可穿戴计算设备、汽车、家庭自动化等)中的计算或处理系统、小型计算机、大型计算机等。

[0033] 在下面的详细描述中，参考构成其一部分的附图，并且通过说明的方式示出了具体的配置或示例。现在参考附图，其中贯穿几个附图，相同的数字表示相同的元素，将描述可以执行层描述符列表中的描述符以有效执行DNN计算的神经网络模块的各个方面。

[0034] 图1是示出根据一个实施例的实现本文中公开的技术的DNN模块105的配置和操作的各方面的计算架构图。在一些实施例中，本文中公开的DNN模块105被配置为解决分类问题(和相关问题)，诸如但不限于对象检测、语义标记和特征提取。

[0035] 为了提供该功能，DNN模块105可以实现仅召回神经网络，并且以编程方式支持多

种网络结构。由DNN模块105实现的网络的训练可以在服务器场、数据中心或其他合适的计算环境中脱机执行。训练DNN的结果是被称为“权重”或“内核”的一组参数。这些参数表示可以应用于输入的转换函数，其结果是分类或语义标记输出。

[0036] 本文中公开的DNN模块105可以被认为是超标量处理器。DNN模块105可以将一个或多个指令分派给多个执行单元(称为神经元105F)。执行单元可以是“同时分派同时完成”，其中每个执行单元与每个其他执行单元同步。DNN模块105可以被分类为单指令流多数据流(“SIMD”)架构。

[0037] DNN模块105包括多个神经元105F(例如，二的幂)。神经元105F是人工神经网络中用于对大脑中的生物神经元进行建模的基本单元。神经元105F的模型可以包括输入矢量的内积，其中权重矢量被添加到偏置，并且应用了非线性。由本文中描述的DNN模块105中的神经元105F执行的处理被紧密地映射到人造神经元。

[0038] DNN模块105中的每个神经元105F能够执行加权总和、最大合并、旁路和潜在的其他类型的操作。神经元105F在每个时钟周期处理输入和权重数据。就内核内的进度而言，每个神经元105F与所有其他神经元105F同步，以最小化DNN模块105内的内核数据流。

[0039] 每个神经元105F可以包含乘法器、加法器、比较器和多个累加器(图1中未示出)。通过具有多个累加器，神经元105F可以一次为多个不同的活动内核保持上下文。每个累加器能够从BaSRAM 150(如下所述)的读取来加载。累加器可以将其自身与来自其他神经元105F的其他累加器的内容求和。

[0040] DNN模块105接受平面数据作为输入，诸如图像数据。然而，DNN模块105的输入不限于图像数据。而是，DNN模块105可以对以统一平面格式呈现给DNN模块105的任何输入数据进行操作。在一个特定实施例中，DNN模块105可以接受多平面一字节或两字节数据帧作为输入。

[0041] 每个输入帧可以与一组NxKxHxW个内核进行卷积，其中N是内核数，K是每个内核的通道数，H是高度，W是宽度。在跨输入数据的重叠间隔上执行卷积，其中间隔由X和Y方向上的跨度定义。这些函数由神经元105F执行，并且由DNN模块105和软件可见控制寄存器进行管理。

[0042] DNN模块105支持三种主要的数据类型：权重；输入数据/特征图；以及激活数据。在大多数情况下，输入数据/特征图和激活数据是针对同一数据的两个名称，区别在于，在涉及层的输出时，使用术语激活数据。当涉及层的输入时，使用术语输入数据/特征图。

[0043] DNN模块105中的神经元105F计算其输入的加权总和，并且使加权总和通过“激活函数”或“传递函数”。传递函数通常具有S形形状，但也可以采用以下形式：分段线性函数、阶跃函数或另一种函数。激活函数允许神经元105F在分类边界为非线性的情况下训练到更大的一组输入和期望输出。

[0044] DNN模块105对与神经网络的层相对应的层描述符列表进行操作。DNN模块105可以将层描述符列表视为指令。这些描述符可以从存储器中预先提取到DNN模块105中并且按顺序执行。描述符列表用作DNN模块105的一组指令。可以在DNN模块105外部的设备上执行软件工具和/或编译器，以创建在DNN模块105上执行的描述符列表。

[0045] 通常，可以有两个主要类别的描述符：M2M描述符；以及操作描述符。M2M描述符可以用于将数据往返于主存储器与本地缓冲器(即，下述行缓冲器125)之间来回移动以供操

作描述符消费。M2M描述符遵循与操作描述符不同的执行流水线。用于M2M描述符的目标流水线可以是内部DMA引擎105B或配置寄存器105G，而用于操作描述符的目标流水线可以是神经元105F。

[0046] 操作描述符包括定义神经元105F应当对位于本地静态随机存取存储器(“SRAM”)存储器中的数据结构执行的特定操作的数据。操作描述符按顺序处理，并且能够执行很多不同的层操作，本文中描述其中的至少一些操作。关于M2M描述符、操作描述符和几种其他类型的描述符以及用于执行描述符的机制的附加细节将在下面关于图2-6提供。

[0047] 如图1所示，DNN模块105具有带有唯一的L1和L2缓冲器结构的存储器子系统。图1所示的L1和L2缓冲器是专门为神经网络处理而设计的。作为示例，L2缓冲器150可以通过以选择的频率(例如，每秒十六吉比特(16GBps))操作的高速专用接口来保持选择的存储容量(例如，一兆字节(1MB))。L1缓冲器125可以保持可以在内核数据与激活数据之间分配的选择的存储容量(例如，八千字节(8KB))。L1缓冲器125在本文中可以称为“行缓冲器125”，并且L2缓冲器150在本文中可以称为BaSRAM 150。

[0048] 在一些实施例中，计算数据(即，输入数据、权重和激活数据)存储在BaSRAM 150行主序中。可以将计算数据组织为两个行缓冲器，其中一个行缓冲器包含输入数据(本文中可以称为“输入缓冲器”)，另一行缓冲器(本文中可以称为“权重缓冲器”)包含内核权重。行缓冲器由加载/存储单元105C从BaSRAM 150填充。数据在每个行缓冲器中累积，直到达到预定容量。然后，在一些实施例中，将行缓冲器数据复制到阴影缓冲器，并且呈现给神经元105F。

[0049] DNN模块105还可以包括其他组件，包括但不限于寄存器接口105G、预取单元105A、保存/恢复单元105E、层控制器105D和寄存器接口105G。在一些实施例中，DNN模块105可以包括附加或备选组件。

[0050] 在一些配置中，DNN模块105与其他外部计算组件相结合操作。例如，在一些实施例中，DNN模块105连接到片上主机应用处理器系统(“主机SoC”)130。DNN模块105可以例如通过PCIe接口连接到主机SoC 130。适当的PCIe组件(诸如PCIe端点135)可以用于启用这些连接。

[0051] 在一些实施例中，主机SoC 130用作用于DNN模块105的应用处理器。主操作系统、应用和辅助传感器处理由主机SoC 130执行。主机SoC 130还可以连接到向DNN模块105提供输入数据(诸如图像数据)的输入数据源102(诸如外部相机)。

[0052] DDR DRAM 155还可以连接到主机SoC 130，DDR DRAM 155可以用作主系统存储器。该存储器通过存储器控制器145在高带宽结构120(例如，PCIe总线)上从主机SoC 130可访问。高带宽结构120提供双向直接存储器访问(“DMA”)小型消息传递事务和较大DMA事务。桥接器115和低带宽结构110可以将DNN模块105连接到主机SoC 130以用于子模块配置和其他功能。

[0053] DNN模块105可以包括被配置为将数据移入和移出主存储器155的DMA引擎105B。在一些实施例中，DMA引擎105B具有两个通道。一个通道专用于获取操作描述符，而另一通道专用于M2M操作。DMA描述符可以嵌入在M2M描述符中。在这种情况下，描述符是用于移动存储器的内容的DMA描述符，请勿与本文中描述的操作描述符混淆。

[0054] 为了卸载本地BaSRAM存储器150，并且为了为输入数据和权重数据提供更多空间，激活输出可以可选地直接流传输到DDR存储器155。当将数据流传输到DDR存储器155时，DNN

模块105将累积足以用于高带宽结构120上的突发事务的数据，并且将缓冲足以最小化神经元105F上的背压的事务。下面将提供关于DNN模块105的操作的附加细节。

[0055] 图2是示出根据一个实施例的用于创建和执行层描述符列表202的说明性环境的各方面的计算系统图。如以上简要描述的，DNN模块105可以取回和执行层描述符列表202中包含的层描述符204以实现DNN。

[0056] 层描述符列表202由诸如在开发计算设备208上执行的编译器206等软件预编译，并且对应于神经网络的层。层描述符列表202可以在开发计算设备208上或在另一环境中创建，并且被部署到托管DNN模块105的设备210(在本文中可以称为“主机210”)。DNN模块105将层描述符204视为指令，并且可以执行层描述符204以执行神经网络的前向传递。

[0057] 在一个实施例中，层描述符列表202存储在主机210的主存储器中，并且由DNN模块加载以用于即时执行。描述符204可以按顺序从主机210的存储器预取到DNN模块105中并且执行。

[0058] 层描述符列表204可以包括几种类型的DNN层描述符204：M2M描述符204B；操作描述符204C；主机通信描述符204D；配置描述符204A；分支描述符204E(在图3中示出并且在下面描述)；以及同步描述符204F。这些描述符类型中的每个在下面描述。在其他实施例中，可以使用其他类型的层描述符。

[0059] M2M描述符204B可以用于将数据往返于主机计算设备210的主存储器与本地缓冲器(即，下面描述的行缓冲器125)以供操作描述符204消耗，如下所述。在一个实施例中，DNN模块105中的DMA引擎(图1中未示出)利用M2M描述符204B来执行DMA操作。

[0060] M2M描述符204B包括指定定义去往和来自任何存储器地址的多维跨度DMA操作的参数的字段。例如而非限制，可以执行M2M描述符以将要由DNN模块105中的神经元105F操作的输入数据和权重数据从主机计算设备210的存储器传输到DNN模块105中的存储器，诸如高速缓存存储器。M2M描述符204B包括定义这种存储器传输的参数的数据。关于这些参数中的至少一些的细节将在下面关于图4B提供。

[0061] 操作描述符204C指定DNN模块105中的神经元105F应当对由M2M描述符获取的数据执行的操作。例如，操作描述符204C可以定义要由神经元105F执行的算术运算，诸如但不限于加法组合、标量乘法和加法、卷积、解卷积、最大池化或完全连接层。

[0062] 操作描述符204C还可以指定要由神经元105F使用的激活函数(诸如但不限于ReLU激活函数和基于查找表的激活函数)、以及要由神经元105F在执行这些操作时使用的数学精度。

[0063] 操作描述符204C还可以包括用于配置DNN模块105的硬件的操作的各方面的微代码。操作描述符204C可以包括其他字段，这些字段包含定义用于实现DNN的神经元105F的配置的各方面的数据，其中的一些在下面参考图4A描述。

[0064] 配置描述符204A启用DNN模块105的配置状态的修改。例如，可以执行配置描述符204A，以配置DNN模块105如何执行舍入操作、功率管理或启用和禁用神经元。

[0065] 主机通信描述符204D使得DNN模块105能够中断主机计算设备210，以提供状态消息和/或其他类型的数据。例如，DNN模块105可以执行主机通信描述符204D，以向主机计算设备210提供关于DNN的层的状态或完成的数据。

[0066] 通过指示神经元105F暂停其处理直到神经网络的其他神经元105F完成其处理，可

以利用同步描述符204F同步DNN模块105中多个神经元105F的执行。在其他实施例中，可以定义和执行其他类型的描述符。

[0067] 在一个实施例中，编译器206向每个描述符204分配唯一的标识ID标签212(可以称为“ID 212”)。ID标签212是编译器206基于描述符在层描述符列表202中的位置而分配给每个描述符204的单调递增数字。

[0068] 在图2所示的示例中，例如，配置描述符204A将具有列表202中的描述符204的最低ID标签212A。针对描述符204D的ID标签212G将具有列表202中的描述符204的最高标签ID 212G。

[0069] 如将在下面关于4A和4B详细描述的，在一些实施例中，标签ID 212存储在描述符204内。如还将在下面更详细描述的，标签ID 212可以用于执行依赖性并且执行防护操作。

[0070] 图3是示出根据一个实施例的图2所示的层描述符列表202的其他方面的数据结构图。特别地，图3示出了包括分支描述符204E的层描述符列表202A的各方面。当指定条件满足时，分支描述符204E使得执行能够在描述符204或层描述符列表202之间分支。

[0071] 在图3所示的示例中，例如，已经定义了分支描述符204E，分支描述符204E在被执行时，将基于指定条件的评估来将执行分支到层描述符列表202B的头部或层描述符列表202C的头部。在其他实施例中，可以以其他方式执行对描述符204的执行的分支。

[0072] 图4A是示出根据一个实施例的用于操作描述符204C的配置的各方面的数据结构图。如图4A所示，操作描述符204C可以包括字段402A-402I，字段402A-402I存储定义特定操作的数据，DNN模块105中的神经元105F应当对通过M2M描述符204B的执行而获取的数据执行该特定操作。在一个实施例中，操作描述符204C是对该信息进行编码的128字节宽的数据结构。在其他实施例中，操作描述符204C可以以其他方式实现。

[0073] 在一个实施例中，操作描述符204C包括字段402A，字段402A存储定义要由神经元105F执行的操作类型的数据，诸如但不限于加法组合、标量乘法和加法、卷积、解卷积、最大池化或完全连接层。操作描述符204C还可以包括字段402B，字段402B存储指定要由神经元105F在指定处理操作期间使用的激活函数的数据，诸如但不限于ReLU激活函数和基于查找表的激活函数。操作描述符204C还可以包括字段402G，字段402G指定要由神经元105F在执行操作时利用的数学精度。

[0074] 操作描述符204C还可以包括字段402C，字段402C存储指示神经元105F在其处理完成时暂停的数据。操作描述符204C还可以包括字段402D，字段402D存储数据，该数据将使得DNN模块105阻止对描述符204的进一步处理直到在其之前的所有描述符204(即，具有较低ID标签)已完成其处理并且已从其完成存储接收到写入响应(该过程在本文中称为“防护”)。这可以用来确保在描述符处理流水线中没有剩余其他描述符204。在一个实施例中，用于操作描述符204的ID标签212存储在字段402R中。下面关于图7、8和10提供关于防护操作的执行的附加细节。

[0075] 操作描述符204C还可以包括字段402E，字段402E嵌入用于配置DNN模块105的硬件的操作的各方面的微代码。例如而非限制，字段402E中的微代码可以由DNN模块105提取并且被执行，以配置实现神经元105F的硬件。作为特定示例，微代码可以被执行，以配置如何执行卷积操作、配置迭代器、和/或配置神经元105F的操作的其他方面。

[0076] 操作描述符204C还可以包括字段402F，字段402F存储定义对另一描述符204的完

成执行的依赖性的数据。具有依赖性集合表示描述符将不被执行，直到在字段402F中标识的描述符已经完成其操作。字段402F可以被设置为零以指示描述符204不具有依赖性，并且因此应当对该描述符204禁用依赖性检查。

[0077] 为了启用依赖性检查，DNN模块105可以保持存储最近完成的描述符204的ID标签212的寄存器。可以保持用于标识最近完成的M2M描述符204B和最近完成的操作描述符204C的单独的寄存器。DNN模块105将释放依赖性并且基于这些寄存器中存储的值来执行防护。下面关于图7、9A-9C和10提供关于依赖性管理的附加细节。

[0078] 应当理解，由于M2M描述符204B是按顺序执行的，因此不必设置两个M2M描述符204B之间的依赖性。因此，M2M描述符204B的执行可以取决于操作描述符204C的执行的完成，并且操作描述符204C的执行可以取决于M2M描述符204B的执行的完成。在某些情况下，操作描述符204B的执行也可以取决于另一操作描述符204B的完成。

[0079] M2M描述符204B和操作描述符204C还可以取决于以上标识的其他描述符类型的完成执行。例如，在图2所示的示例中，M2M描述符204B的执行可以取决于配置描述符204A的完成执行。类似地，主机通信描述符204D的执行可以取决于第二操作描述符204C的完成执行。在本文中公开的实施例中，以上标识的不同类型的描述符204可以彼此依赖。

[0080] 操作描述符204C还可以包括字段402H和402I，字段402H和402I存储定义到操作的输入数据和由操作生成的输出数据(本文中可以称为“斑点”)的各方面的数据。这些字段可以包括例如标识以下各项的数据：用于输入和输出数据的存储器地址、输入和输出数据的尺寸、输入和输出数据的精度、特征计数、输入数据的高度和宽度、通道数、输出数据的高度和宽度、填充配置和跨步配置。在其他实施例中，可以在操作描述符204C中指定输入和输出数据的其他方面。

[0081] 应当理解，字段402A-402I仅是说明性的。在其他实施例中，操作描述符204C可以包括附加或备选字段，其存储定义要由神经元105F执行以实现DNN的操作的其他方面的数据。

[0082] 图4B是示出根据一个实施例的M2M描述符204B的配置的各方面的数据结构图。如上所述，M2M描述符204B包括指定定义往返于任何存储器地址的多维跨度DMA操作的参数的字段。在一个实施例中，M2M描述符204B是编码该信息的128字节宽的数据结构。在其他实施例中，M2M描述符204B可以以其他方式实现。

[0083] 在一个实施例中，M2M描述符204B包括指示描述符是M2M描述符204B的字段402A。M2M描述符204B还可以包括存储诸如以上针对M2M描述符204B描述的ID标签212的字段402R。M2M描述符204B还可以包括也以上述方式指定对另一描述符204的完成执行的依赖性的字段402F。类似地，M2M描述符204B可以包含字段402D，字段402D存储数据，该数据将使得DNN模块105阻止对描述符204B的进一步处理、直到在其之前的所有描述符204完成其处理并且从其完成存储接收到写入响应(即，上述防护操作)。下面将关于图7-10提供关于这些防护和依赖性管理操作的附加细节。

[0084] M2M描述符204B还可以包括指定用户定义的传输ID的字段402J。通过在该字段中存储唯一编号并且监测DNN模块105的操作状态寄存器中的相应字段，软件可以标识当前正在执行的传输。

[0085] M2M描述符204B还可以包括指定要传输的数据的各方面的字段402K-402P。例如而

非限制,这些字段可以分别存储标识源跨度(即,DMA传输的连续行的第一字节之间的字节数)、目的地跨度、X和Y维度上的操作大小、源存储器地址和目的地存储器地址。在其他实施例中,M2M描述符204B可以包括存储定义M2M操作的其他方面的数据的附加或备选字段。

[0086] 图5是示出根据一个实施例的关于被配置为执行层描述符列表202中的描述符204的DNN模块105的配置的细节的计算架构图。在该实施例中,DNN模块105包括执行用于取回和路由层描述符列表202中的描述符204的操作的描述符列表控制器("DLC")500。

[0087] 如图5所示,主机CPU 502(即,主机计算设备210中的CPU)向DLC 500提供标识要执行的层描述符列表202的数据。在一个实施例中,该数据存储在描述符队列504中。描述符获取单元506又从描述符队列504中取回数据,并且指示DMA引擎105B取回所标识的层描述符列表202。DMA引擎105B从主机计算机210的DRAM155中取回层描述符列表202,并且将层描述符列表202存储在由DLC 500提供的高速缓存508中。

[0088] 在一些实施例中,依赖性/防护检查单元510从高速缓存508中取回描述符204,并且执行上述防护和依赖性检查。关于这些处理操作的细节将在下面关于图7-10提供。在依赖性/防护检查单元510处理之后,层描述符列表202中的描述符204被提供给路由引擎512。

[0089] 路由引擎512将不同类型的描述符路由到不同的流水线。例如,在图5所示的示例中,路由引擎512已经将M2M描述符204B路由到DMA引擎105B。如上所述,DMA引擎105B可以利用M2M描述符204B的内容,将要由DNN模块105中的神经元105F进行操作的输入数据和权重数据从主机计算设备210的存储器传输到DNN模块105中的存储器,诸如高速缓存。

[0090] 在图5所示的示例中,路由引擎512还已经将操作描述符204C路由到操作控制器514以执行。操作控制器514将利用操作描述符204C来配置神经元105F。一旦被配置,神经元105F可以处理通过M2M描述符204B的执行而取回到的数据。可以针对诸如上面描述的附加描述符204、和附加层描述符列表202重复该过程。以下关于图7更详细地讨论关于由DLC 500实现的处理流水线的附加细节。

[0091] 还如图5所示,路由引擎512可以利用配置描述符204A来设置定义DNN模块105的配置的配置寄存器516的状态。还如图5所示,主机CPU 502还可以独立地访问配置寄存器516,以设置或取回DNN模块105的配置状态。例如,主机CPU 502可以以使得DNN模块105在DNN处理完成之后断电的方式设置配置寄存器516。在其他实施例中,主机CPU 502可以利用配置寄存器516来设置DNN模块105的配置状态的其他方面。寄存器516可以在包含DNN模块105的功率岛外部,以便在DNN模块105断电时允许主机210访问寄存器516。

[0092] 如上所述,并且下面关于图7进一步讨论,DLC 500与操作描述符204C分开地缓冲M2M描述符204B。以这种方式,在明确设置依赖性的情况下,可以尽快执行M2M描述符204B。结果,可以减少通常在获取数据以供DNN模块105中的神经元105F处理时所引起的等待时间,因此,神经元105F可以比其他情况下更快地完成其处理。然后,DNN模块105可以比其他方式更早地断电,从而节省功率。还可以通过所公开的技术的实现来实现本文中未具体确定的其他技术益处。

[0093] 图6是示出根据本文中公开的一个实施例的例程600的流程图,该例程600示出了用于执行层描述符列表202中的描述符204的参考图1-5描述的DNN模块105的操作的各方面。应当理解,本文中关于图6和其他图描述的逻辑操作可以实现为(1)在计算设备上运行的一系列计算机实现的动作或程序模块,和/或(2)诸如DNN模块105的计算设备内的互连的

机器逻辑电路或电路模块。

[0094] 本文中公开的技术的特定实现是取决于计算设备的性能和其他要求的选择问题。因此，本文中描述的逻辑操作被不同地称为状态、操作、结构设备、动作或模块。这些状态、操作、结构设备、动作和模块可以用硬件、软件、固件、专用数字逻辑及其任何组合来实现。应当理解，可以执行比附图中示出和本文中描述的更多或更少的操作。这些操作也可以以与本文中描述的顺序不同的顺序执行。

[0095] 例程600开始于操作602，在操作602，DLC 500从主机CPU 502接收描述在主机DRAM 155中的描述符列表202的位置的数据。该数据放置在描述符队列504中。然后，例程600从操作602前进到操作604，在操作604，描述符获取单元506指示DMA引擎105B从主机DRAM 155中取回层描述符列表202。DMA引擎105B又取回层描述符列表202，并且将列表202存储在高速缓存508中。然后，例程600从操作604前进到操作606。

[0096] 在操作606，从高速缓存508中取回描述符列表202中的第一描述符204。然后，例程600前进到操作608，在操作608，依赖性/防护检查单元510确定描述符204是否指示防护或依赖性。如果是，则例程600从操作608前进到操作610，在操作610，依赖性/防护检查单元510确定是否满足指定的防护或依赖性条件。如果是，则例程600从操作612分支到操作614。如果在操作608，依赖性/防护检查单元510确定描述符204未指示防护或依赖性，则例程600从操作608前进到操作614。下面关于图7-10提供关于用于执行依赖性和防护检查的处理操作的细节。

[0097] 在操作614，路由引擎512确定当前描述符204是否是分支描述符204E。如果是，则例程600从操作614前进到操作616，在操作616，确定是否满足由分支描述符204E指定的条件。如果是，则例程600从操作616前进到操作618，在操作618，描述符204的执行分支到由分支描述符204E标识的描述符204。例程600然后从操作618回到操作608，在操作608，可以处理下一描述符204。

[0098] 如果在操作614，路由引擎512确定当前描述符204不是分支描述符204E，则例程600从操作614前进到操作620。在操作620，路由引擎512确定当前描述符是否204是M2M描述符204B。如果是，则例程600从操作614前进到操作622，在操作622，路由引擎512将当前描述符204B路由到DMA引擎105B，以便执行指定的M2M操作。如果当前描述符204不是M2M描述符204B，则例程600从操作620前进到操作624。

[0099] 在操作624，路由引擎512确定当前描述符204是否是操作描述符204C。如果是，则例程600从操作624前进到操作626，在操作626，路由引擎512将当前描述符204C路由到操作控制器514，以便配置神经元105F并且执行由操作描述符204C指定的处理操作。如果当前描述符204不是操作描述符204C，则例程600从操作624前进到操作628。

[0100] 在操作628，路由引擎512确定当前描述符204是否是主机通信描述符204D。如果是，则例程600从操作628前进到操作630，在操作630，主机210可以被中断以便将数据从DNN模块104传输到主机210。如果当前描述符204不是主机通信描述符204D，则例程600从操作628前进到操作632。

[0101] 在操作632，路由引擎512确定当前描述符204是否是同步描述符。如果是，则例程600从操作632前进到操作634，在操作634，操作控制器514以上述方式同步神经元105F。如果当前描述符204不是同步描述符，则例程600从操作632前进到操作636，在操作636，可以

执行其他描述符类型。然后，例程600从操作636前进到操作638。

[0102] 在操作638，DLC 500确定在描述符列表202中是否存在要执行的附加描述符204。如果否，则例程600从操作638前进到操作642，在操作642，例程600结束。如果还有待处理的附加描述符204，则例程600从操作638前进到操作640，在操作640，取回描述符列表202中的下一描述符204。然后，例程600返回到操作608，在操作608，可以以上述方式处理描述符204。

[0103] 图7是示出根据一个实施例的被配置用于执行层描述符列表202中的描述符204、并且提供层和操作防护和依赖性管理的DNN模块105的配置和操作的附加细节的计算架构图。如图7所示并且在上面简要描述，路由引擎512可以将不同类型的描述符路由到不同的流水线702。

[0104] 在图7所示的示例配置中，DLC 500实现三个流水线：M2M流水线702A；操作流水线702B；和配置流水线702C，每个将在下面进行描述。流水线702独立地操作，因此，例如，如果没有具体定义依赖性，则在流水线702B执行操作描述符204C的同时，流水线702A可以执行下一M2M描述符204B。如下所述，如果在流水线702种的一个流水线的描述符204中标识出未满足的依赖性，则其他流水线702可以继续执行描述符204。

[0105] 路由引擎512将M2M描述符204B路由到M2M流水线702A。如图7所示，M2M流水线702A包括依赖性检查单元704A和先进先出(“FIFO”)存储器706A。依赖性检查单元704A确定M2M描述符204B是否指定依赖性(即，字段402F标识M2M描述符204的执行所依赖的另一描述符204)、或者指示在执行M2M描述符204B之前要执行防护操作(即，字段402D指示将要诸如通过被设置为逻辑“1”来执行防护操作)。

[0106] 如上所述，可以通过确定存储在DNN模块105的硬件寄存器中的最近完成的层描述符204的ID标签212是否小于其所依赖的层描述符204的ID标签212来执行依赖性检查。如果最近完成的层描述符204的ID标签212不小于其所依赖的描述符的ID标签212，则将执行当前描述符204。

[0107] 如果最近完成的层描述符204的ID标签212小于当前描述符所依赖的描述符的ID标签212，则当前描述符204的执行将被暂缓。当前描述符204的执行将被暂缓，直到最近完成的层描述符204的ID标签212等于当前描述符204所依赖的描述符204的ID标签212。

[0108] 应当理解，必须检查依赖性的所有组合。例如，必须确定M2M描述符是否依赖于另一M2M描述符，M2M描述符是否依赖于操作描述符204C，操作描述符204C是否依赖于另一操作描述符204C，以及操作描述符204C是否依赖于M2M描述符204B。另外，还应当理解，某些依赖性情况仍然允许描述符204流过流水线702。例如，如果描述符204由于对另一描述符204的依赖性而被暂缓，并且在暂缓的描述符之后存在没有依赖性的M2M描述符204B，则M2M描述符204B可以在暂缓的描述符204周围移动并且执行。

[0109] 将进一步认识到，描述符204的执行可以取决于读取操作或写入操作。例如，如果流水线702中的当前描述符204将写入由另一描述符204当前正在从其读取的存储器位置，则当前描述符必须等待直到读取操作完成。类似地，如果当前描述符204将从由另一描述符204当前正在写入的位置读取，则当前描述符204必须等待直到写入操作完成。

[0110] 可以通过确定层描述符列表202中ID标签212小于当前层描述符204A的标签ID 212的所有层描述符204是否已经完成执行来执行防护检查操作。这可以通过将当前描述符

204的标签ID 212与最近完成的层描述符204的标签ID 212进行比较来实现。如果层描述符列表202中ID标签212小于第一层描述符204A的ID标签212的所有层描述符204已经完成执行，则将执行当前层描述符204。如果层描述符列表202中ID标签212小于第一层描述符204A的ID标签212的所有层描述符204尚未完成执行，则将不执行当前层描述符204。

[0111] 一旦解决了M2M流水线702A中的M2M描述符204B中的任何防护或依赖性，就可以将当前的M2M描述符204B通过FIFO存储器706A路由到DMA引擎105B，以便以上述方式进行处理。以这种方式，防护操作保证了在处理具有要声明的防护的描述符204之前，流水线702中不存在其他描述符204。

[0112] 路由引擎512还将操作描述符204C路由到操作流水线702B。依赖性检查单元704B然后可以以上述方式评估由当前操作描述符204C指定的任何防护或依赖性。一旦解决了操作流水线702B中的操作描述符204C中的任何防护或依赖性，就可以将当前操作描述符204C通过FIFO存储器706B路由到DMA引擎105B，以便以上述方式进行处理。

[0113] 路由引擎512还将配置描述符204A路由到配置流水线702C。依赖性检查单元704C然后可以以上述方式评估由当前配置描述符204A指定的任何防护或依赖性。一旦解决了由当前配置描述符204A指定的防护或依赖性，就可以如上所述将当前配置描述符204A通过FIFO存储器706C路由到配置寄存器516。

[0114] 图8是示出根据一个实施例的用于执行防护操作的DNN模块105的操作的各方面的数据结构图。在图8所示的示例中，层描述符列表202A包括操作描述符204C、M2M描述符204B、操作描述符204C'、M2M描述符204B'和分支描述符204E。硬件寄存器802A用于存储最近完成的M2M描述符204B的ID标签212，并且硬件寄存器802B用于存储最近完成的操作描述符204C的ID标签212。如上所述，可以保持单独的寄存器802以存储每种描述符类型的最近完成的描述符204的ID标签212。

[0115] 在图8所示的示例中，首先执行操作描述符204C，并且在完成时使得寄存器802B中的值被设置为11(即，操作描述符204C的ID标签212的值)。然后，执行M2M描述符204B，并且在完成时使得寄存器802A中的值被设置为12。操作描述符204C和M2M描述符204B可以并行执行，因为这些描述符尚未设置依赖性。

[0116] 然后，执行操作描述符204C'，并且当完成操作时，寄存器802B的值被设置为13。然后，开始M2M描述符204B'的执行。因为M2M描述符204B'没有指定任何依赖性，所以分支描述符204E的执行可以由流水线204C并行执行。分支描述符204E指示如果指定条件满足，则将进行到描述符列表202B的头部的分支。如果指定条件不满足，则将进行到层描述符列表202C的头部的分支。

[0117] 描述符列表202B中的第一描述符204是配置描述符204A。描述符204A指示要执行防护操作。类似地，描述符列表202C中的第一描述符204是配置描述符204A'。该描述符204A'还指示要执行防护操作。因此，当描述符列表202B和描述符列表202C的执行开始时，将执行防护操作。

[0118] 如上所述，防护操作要求在可以执行当前描述符204之前，描述符列表202中ID标签212低于当前描述符204的所有描述符204都已经完成。在图8所示的示例中，这表示，ID标签212小于16(即，描述符204A的ID标签212)的描述符204必须在描述符204A的执行可以开始之前已经完成。通过检查寄存器802A和802B，DLC 500可以确定最近完成的描述符204是

操作描述符204C'，其ID标签212为13。因此，配置描述符204A(或配置204A')的执行将被暂缓，直到M2M描述符204B'完成并且寄存器802A中的值设置为14。

[0119] 图9A-9C是示出根据一个实施例的用于执行操作依赖性的DNN模块105的操作的各方面的数据结构图。如在图8所示的示例中，图9A-9C所示的说明性层描述符列表202A包括操作描述符204C、M2M描述符204B、操作描述符204C'、M2M描述符204B'和分支描述符204E。硬件寄存器802A用于存储最近完成的M2M描述符204B的ID标签212，并且硬件寄存器802B用于存储最近完成的操作描述符204C的ID标签212。

[0120] 在该示例中，操作描述符204C的执行首先开始。因为操作描述符204C尚未设置任何依赖性，所以可以开始M2M描述符204B的执行。然而，在该示例中，M2M描述符204B指定对操作描述符204C的依赖性。因此，如图9B所示，M2M描述符204B的执行将被暂缓，直到操作描述符204C完成并且寄存器802B的值设置为11。

[0121] 如图9B所示，操作描述符204C'具有对M2M描述符204B的依赖性。因此，操作描述符204C'的执行将被暂缓，直到M2M描述符204B完成并且寄存器802A的值设置为11。一旦发生这种情况，操作描述符204C'将执行，并且在完成时将使得寄存器802B中的值被设置为13。M2M描述符204B'也将开始执行。

[0122] 如图9C所示，分支描述符204E也具有依赖性集合，在这种情况下，是对M2M描述符204B'的依赖性集合。因此，分支描述符204E将被暂缓，直到M2M描述符204B'完成并且寄存器802A的值设置为14。一旦发生，执行可以分支到描述符204A或204A'，这取决于对指定条件的评估结果。应当理解，图8和9A-9C所示的示例仅是说明性的，并且出于讨论目的已经被简化。

[0123] 图10是示出根据本文中公开的一个实施例的例程1000的流程图，该例程1000示出了用于提供层和操作防护和依赖性管理的DNN模块105的操作的各方面。例程1000开始于操作1002，在操作1002，DLC 500确定是否要执行依赖性或防护。如果要执行依赖性，则例程1000从操作1002前进到操作1004。

[0124] 在操作1004，DLC 500确定最近完成的层描述符204的ID标签212是否小于指定依赖性的描述符204的标签ID 212。如果最近完成的层描述符204的ID标签212小于指定依赖性的描述符204的标签ID 212，则表示依赖性尚未解决。因此，例程1000从操作1006前进到操作1008，在操作1008，指定依赖性的描述符204的执行被暂缓。然后，例程1000从操作1008前进到操作1012。在操作1012，处理前进到以上关于图6描述的例程600的操作612。

[0125] 如果最近完成的层描述符204的ID标签212不小于指定依赖性的描述符204的标签ID 212，则表示依赖性已经解决。因此，例程1000从操作1006前进到操作1010，在操作1010，开始指定依赖性的描述符204的执行。然后，例程1000从操作1010前进到操作1012。在操作1012，处理前进到以上关于图6描述的例程600的操作612。

[0126] 在操作1002，如果DLC 500确定要执行防护，则例程1000从操作1002前进到操作1014。在操作1014，DLC 500确定层描述符列表202中ID标签212小于当前层描述符204的ID标签212的所有层描述符204的执行是否已经完成。如果具有较低ID标签212的所有描述符204的执行尚未完成，则例程1000从操作1016前进到操作1020，在操作1020，当前层描述符204的执行被暂缓。例程1000然后从操作1020前进到操作1012。在操作1012，处理前进到以上关于图6描述的例程600的操作612。

[0127] 如果ID标签212小于当前描述符204的所有描述符204的执行已经完成，则例程1000从操作1016前进到操作1018，在操作1018，可以开始当前描述符204的执行。然后，例程1000从操作1018前进到操作1012。在操作1012，处理前进到以上关于图6描述的例程600的操作612。

[0128] 图11是示出可以用作本文中提出的针对DNN模块105的应用主机的计算设备的说明性计算机硬件和软件架构的计算机架构图。特别地，图11所示的架构可以用于实现服务器计算机、移动电话、电子阅读器、智能电话、台式计算机、AR/VR设备、平板计算机、膝上型计算机、或适合于与DNN模块105一起使用的另一种类型的计算设备。

[0129] 图11所示的计算机1100包括中央处理单元1102(“CPU”)、系统存储器1104(包括随机存取存储器1106(“RAM”)和只读存储器(“ROM”)1108)、以及将存储器1104耦合到CPU 1102的系统总线1110。基本输入/输出系统(“BIOS”或“固件”)可以存储在ROM 1108中，基本输入/输出系统包含用于帮助诸如在启动过程中在计算机1100内的各个元件之间传输信息的基本例程。计算机1100还包括用于存储操作系统1122、应用程序和其他类型的程序的大容量存储设备1112。大容量存储设备1112还可以被配置为存储其他类型的程序和数据。

[0130] 大容量存储设备1112通过连接到总线1110的大容量存储控制器(未示出)连接到CPU 1102。大容量存储设备1112及其关联的计算机可读介质为计算机1100提供非易失性存储。尽管本文中包含的计算机可读介质的描述是指大容量存储设备，诸如硬盘、CD-ROM驱动器、DVD-ROM驱动器或USB存储密钥，但是本领域技术人员应当理解，计算机可读介质可以是计算机1100可以访问的任何可用的计算机存储介质或通信介质。

[0131] 通信介质包括诸如载波或其他传输机制等调制数据信号中的计算机可读指令、数据结构、程序模块或其他数据，并且包括任何传递介质。术语“调制数据信号”是指具有以能够将信息编码在信号中的方式来改变或设置其一个或多个特性的信号。作为示例而非限制，通信介质包括诸如有线网络或直接有线连接等有线介质、以及诸如声学、射频、红外和其他无线介质等无线介质。以上任何内容的组合也应当被包括在计算机可读介质的范围内。

[0132] 作为示例而非限制，计算机存储介质可以包括以用于存储诸如计算机可读指令、数据结构、程序模块或其他数据等信息的任何方法或技术实现的易失性和非易失性、可移动和不可移动介质。例如，计算机存储介质包括但不限于RAM、ROM、EPROM、EEPROM、闪存或其他固态存储技术、CD-ROM、数字多功能磁盘(“DVD”)、HD-DVD、BLU-RAY或其他光学存储器、磁带盒、磁带、磁盘存储器或其他磁性存储设备、或者可以用于存储期望信息并且可以由计算机1100访问的任何其他介质。出于权利要求的目的，短语“计算机存储介质”及其变体不包括波或信号本身或通信介质。

[0133] 根据各种配置，计算机1100可以使用通过诸如网络1120等网络到远程计算机的逻辑连接来在联网环境中操作。计算机1100可以通过连接到总线1110的网络接口单元1116连接到网络1120。应当理解，网络接口单元1116也可以用于连接到其他类型的网络和远程计算机系统。计算机1100还可以包括用于接收和处理来自很多其他设备(包括键盘、鼠标、触摸输入、电子笔(图11中未示出)或物理传感器，诸如视频相机)的输入的输入/输出控制器1118。类似地，输入/输出控制器1118可以向显示屏或其他类型的输出设备(在图11中也未示出)提供输出。

[0134] 应当理解，本文中描述的软件组件在被加载到CPU 1102中并且被执行时，可以将CPU 1102和整个计算机1100从通用计算设备转换为被定制为促进本文中介绍的功能的专用计算设备。CPU 1102可以由可以个体或共同地呈现任何数目的状态的任何数目的晶体管或其他分立电路元件构成。更具体地，响应于本文中公开的软件模块中包含的可执行指令，CPU 1102可以作为有限状态机操作。这些计算机可执行指令可以通过指定CPU 1102如何在状态之间转换来对CPU 1102进行转换，从而对构成CPU 1102的晶体管或其他分立硬件元件进行转换。

[0135] 对本文中提出的软件模块进行编码还可以变换本文中提出的计算机可读介质的物理结构。在本说明书的不同实现中，物理结构的特定变换取决于各种因素。这样的因素的示例包括但不限于用于实现计算机可读介质的技术、计算机可读介质的特征是主要存储还是辅助存储等。例如，如果计算机可读介质被实现为基于半导体的存储器，则可以通过变换半导体存储器的物理状态来将本文中公开的软件编码在计算机可读介质上。例如，该软件可以变换构成半导体存储器的晶体管、电容器或其他分立电路元件的状态。该软件还可以转换这些组件的物理状态，以便在其上存储数据。

[0136] 作为另一示例，本文中公开的计算机可读介质可以使用磁性或光学技术来实现。在这样的实现中，当软件被编码在其中时，本文中提出的软件可以变换磁性或光学介质的物理状态。这些变换可以包括改变给定磁性介质内的特定位置的磁性特性。这些变换还可以包括改变给定光学介质内的特定位置的物理特征或特性，以改变这些位置的光学特性。在不背离本说明书的范围和精神的情况下，物理介质的其他变换是可能的，其中提供前述示例仅是为了促进该讨论。

[0137] 鉴于以上所述，应当理解，在计算机1100中发生了很多类型的物理变换以便存储和执行本文中提出的软件组件。还应当理解，图11中针对计算机1100示出的架构或类似架构可以用于实现其他类型的计算设备，包括手持计算机、视频游戏设备、嵌入式计算机系统、移动设备(诸如智能手机、平板电脑和AR/VR设备)、以及本领域技术人员已知的其他类型的计算设备。还可以想到，计算机1100可以并非包括图11所示的所有组件，可以包括图11中未明确示出的其他组件，或者可以使用与图11所示的架构完全不同的架构。

[0138] 图12是示出根据本文中呈现的各种实施例的可以在其中实现所公开的技术的各方面的分布式网络计算环境1200的网络图。如图12所示，一个或多个服务器计算机1200A可以经由通信网络1120(其可以是固定有线或无线LAN、WAN、内联网、外联网、对等网络、虚拟专用网络、因特网、蓝牙通信网络、专有低压通信网络或其他通信网络)与多个客户端计算设备(诸如但不限于平板电脑1200B、游戏机1200C、智能手表1200D、电话1200E(诸如智能电话)、个人计算机1200F和AR/VR设备1200G)互连。

[0139] 例如，在通信网络1120是因特网的网络环境中，服务器计算机1200A可以是专用服务器计算机，该专用服务器计算机可操作以经由多种已知协议中的任何一种来处理与客户端计算设备1200B-1200G的数据以及与客户端计算设备1200B-1200G传送数据，诸如超文本传输协议(“HTTP”)、文件传输协议(“FTP”)或简单对象访问协议(“SOAP”)。另外，网络计算环境1200可以利用各种数据安全协议，诸如安全套接字层(“SSL”)或相当好的隐私(“PGP”)。每个客户端计算设备1200B-1200G可以配备有操作系统，该操作系统可操作以支持一个或多个计算应用或终端会话，诸如网络浏览器(图12中未示出)或其他图形用户界面

(图12中未示出)或移动桌面环境(图12中未显示),以获取对服务器计算机1200A的访问。

[0140] 服务器计算机1200A可以通信地耦合到其他计算环境(图12中未示出),并且接收有关参与用户的交互/资源网络的数据。在说明性操作中,用户(图12中未示出)可以与在客户端计算设备1200B-1200G上运行的计算应用交互以获取期望数据和/或执行其他计算应用。

[0141] 数据和/或计算应用可以存储在一个或多个服务器1200A上,并且通过示例性通信网络1120通过客户端计算设备1200B-1200G传送到合作用户。参与用户(图12中未示出)可以请求访问全部或部分容纳在服务器计算机11800A上的特定数据和应用。这些数据可以在客户端计算设备1200B-1200G与服务器计算机1200A之间传送以进行处理和存储。

[0142] 服务器计算机1200A可以托管用于数据、应用的生成、认证、加密和通信的计算应用、过程和小程序,并且可以与其他服务器计算环境(图12中未示出)、第三方服务供应商(图12中未示出)、网络附加存储("NAS")和存储区域网络("SAN")协作以实现应用/数据交易。

[0143] 应当理解,图11所示的计算架构和图12所示的分布式网络计算环境为了便于讨论而被简化。还应当理解,计算架构和分布式计算网络可以包括和利用本文中未具体描述的更多的计算组件、设备、软件程序、网络设备和其他组件。

[0144] 本文中提出的公开内容还涵盖以下条款中阐述的主题:

[0145] 条款1.一种神经网络处理器,包括:存储器设备,存储包括针对神经网络的第一层描述符的层描述符列表,所述第一层描述符指定针对所述第一层描述符的执行所依赖的第二层描述符的标识符(ID);硬件寄存器,存储最近完成的层描述符的ID;以及控制器,被配置为:确定存储在所述硬件寄存器中的最近完成的层描述符的ID是否小于所述第二层描述符的ID,响应于确定所述最近完成的层描述符的ID不小于所述第二层描述符的ID,使得所述神经网络处理器执行所述第一层描述符,以及响应于确定所述最近完成的层描述符的ID小于所述第二层描述符的ID,使得所述神经网络处理器暂缓所述第一层描述符的执行。

[0146] 条款2.根据条款1所述的神经网络处理器,其中所述第一层描述符的执行被暂缓,直到存储在所述硬件寄存器中的最近完成的层描述符的ID等于所述第二层描述符的ID。

[0147] 条款3.根据条款1-2中任一项所述的神经网络处理器,其中所述神经网络处理器还被配置为执行所述层描述符列表中的层描述符,并且将所述最近完成的层描述符的ID存储在所述硬件寄存器中。

[0148] 条款4.根据条款1-3中任一项所述的神经网络处理器,标识符以单调递增顺序被分配给所述层描述符列表中的层描述符。

[0149] 条款5.根据条款1-4中任一项所述的神经网络处理器,其中所述第一层描述符包括以下各项中的一项:存储器到存储器移动(M2M)描述符、操作描述符、主机通信描述符、配置描述符、分支描述符或同步描述符。

[0150] 条款6.根据条款1-5中任一项所述的神经网络处理器,其中所述第二层描述符包括以下各项中的一项:存储器到存储器移动(M2M)描述符、操作描述符、主机通信描述符、配置描述符、分支描述符或同步描述符。

[0151] 条款7.根据条款1-6中任一项所述的神经网络处理器,其中所述层描述符列表包括指定防护操作的第三层描述符,并且其中所述控制器还被配置为:至少部分地基于所述

最近完成的层描述符的ID来确定所述层描述符列表中具有ID小于所述第三层描述符的ID的所有层描述符是否已经完成执行,响应于确定所述层描述符列表中ID小于所述第三层描述符的ID的所有层描述符已经完成执行,使得所述神经网络处理器执行所述第三层描述符,以及响应于确定所述层描述符列表中具有ID小于所述第三层描述符的ID的所有层描述符尚未完成执行,使得所述神经网络处理器暂缓所述第三层描述符的执行。

[0152] 条款8:一种神经网络处理器,包括:存储器设备,存储包括针对神经网络的第一层描述符的层描述符列表;硬件寄存器,存储最近完成的层描述符的标识符(ID);以及控制器,被配置为:至少部分地基于所述最近完成的层描述符的ID来确定所述层描述符列表中具有ID小于所述第一层描述符的ID的所有层描述符是否已经完成执行,响应于确定所述层描述符列表中具有ID小于所述第一层描述符的ID的所有层描述符已经完成执行,使得所述神经网络处理器执行所述第一层描述符,以及响应于确定所述层描述符列表中具有ID小于所述第一层描述符的ID的所有层描述符尚未完成执行,使得所述神经网络处理器暂缓所述第一层描述符的执行。

[0153] 条款9.根据条款8所述的神经网络处理器,其中所述第一层描述符的执行被暂缓,直到所述描述符列表中具有ID小于所述第一层描述符的ID的所有层描述符已经完成执行。

[0154] 条款10:根据条款8-9中任一项所述的神经网络处理器,其中所述神经网络处理器还被配置为执行所述层描述符列表中的层描述符,并且将所述最近完成的层描述符的ID存储在所述硬件寄存器中。

[0155] 条款11.根据条款8-10中任一项所述的神经网络处理器,其中标识符以单调递增顺序被分配给所述层描述符列表中的层描述符。

[0156] 条款12.根据条款8-11中任一项所述的神经网络处理器,其中所述第一层描述符包括以下各项中的一项:存储器到存储器移动(M2M)描述符、操作描述符、主机通信描述符、配置描述符、分支描述符或同步描述符。

[0157] 条款13.根据条款8-12中任一项所述的神经网络处理器,其中所述层描述符列表中的描述符包括以下各项中的一项或多项:存储器到存储器移动(M2M)描述符、操作描述符、主机通信描述符、配置描述符、分支描述符或同步描述符。

[0158] 条款14.根据条款8-13中任一项所述的神经网络处理器,其中所述层描述符列表还包括第二层描述符,所述第二层描述符指定针对所述第二层描述符的执行所依赖的第三层描述符的标识符ID,并且其中所述控制器还被配置为:确定存储在所述硬件寄存器中的最近完成的层描述符的ID是否小于所述第三层描述符的ID,响应于确定所述最近完成的层描述符的ID不小于所述第三层描述符的ID,使得所述神经网络处理器执行所述第二层描述符,以及响应于确定所述最近完成的层描述符的ID小于所述第三层描述符的ID,使得所述神经网络处理器暂缓所述第二层描述符的执行。

[0159] 条款15:一种计算机实现的方法,包括:通过神经网络模块访问包括针对神经网络的第一层描述符的层描述符列表,所述第一层描述符指定所述第一层描述符的执行所依赖的第二层描述符的标识符(ID);确定最近完成的层描述符的标识符(ID)是否小于所述第二层描述符的ID;响应于确定所述最近完成的层描述符的ID不小于所述第二层描述符的ID,通过所述神经网络模块执行所述第一层描述符;以及响应于确定所述最近完成的层描述符的ID小于所述第二层描述符的ID,通过所述神经网络模块暂缓所述第一层描述符的执行。

[0160] 条款16.根据条款15所述的计算机实现的方法，其中所述第一层描述符的执行被暂缓，直到所述最近完成的层描述符的ID等于所述第二层描述符的ID。

[0161] 条款17.根据条款15-16中任一项所述的计算机实现的方法，还包括：执行所述层描述符列表中的层描述符；以及

[0162] 将所述最近完成的层描述符的ID存储在所述神经网络模块的硬件寄存器中。

[0163] 条款18：根据条款15-17中任一项所述的计算机实现的方法，其中标识符以单调递增顺序被分配给所述层描述符列表中的层描述符。

[0164] 条款19.根据条款15-18中任一项所述的计算机实现的方法，其中所述第一层描述符和所述第二层描述符包括以下各项中的一项：存储器到存储器移动(M2M)描述符、操作描述符、主机通信描述符、配置描述符、分支描述符或同步描述符。

[0165] 条款20.根据条款15-19中任一项所述的计算机实现的方法，其中所述层描述符列表包括指定防护操作的第三层描述符，并且其中计算机实现的方法还包括：至少部分地基于所述最近完成的层描述符的ID，确定所述层描述符列表中具有ID小于所述第三层描述符的ID的所有层描述符是否已经完成执行；响应于确定所述层描述符列表中具有ID小于所述第三层描述符的ID的所有层描述符已经完成执行，通过所述神经网络处理器执行所述第三层描述符；以及响应于确定所述层描述符列表中具有ID小于所述第三层描述符的ID的所有层描述符尚未完成执行，暂缓通过所述神经网络处理器对所述第三层描述符的执行。

[0166] 基于前述内容，应当理解，本文中已经公开了被配置用于层和操作防护和依赖性管理的神经网络模块。尽管已经以计算机结构特征、方法和转换动作、特定的计算机器和计算机可读介质专用的语言描述了本文中介绍的主题，但是应当理解，所附权利要求书中提出的主题不必限于本文中描述的特定功能、动作或介质。相反，特定特征、动作和介质被公开作为实现所要求保护的主题的示例形式。

[0167] 上述主题仅以示例的方式提供，并且不应当被解释为是限制性的。可以在不遵循示出和描述的示例配置和应用的情况下，并且在不脱离在所附权利要求中阐述的本公开的范围的情况下，对本文中描述的主题进行各种修改和改变。

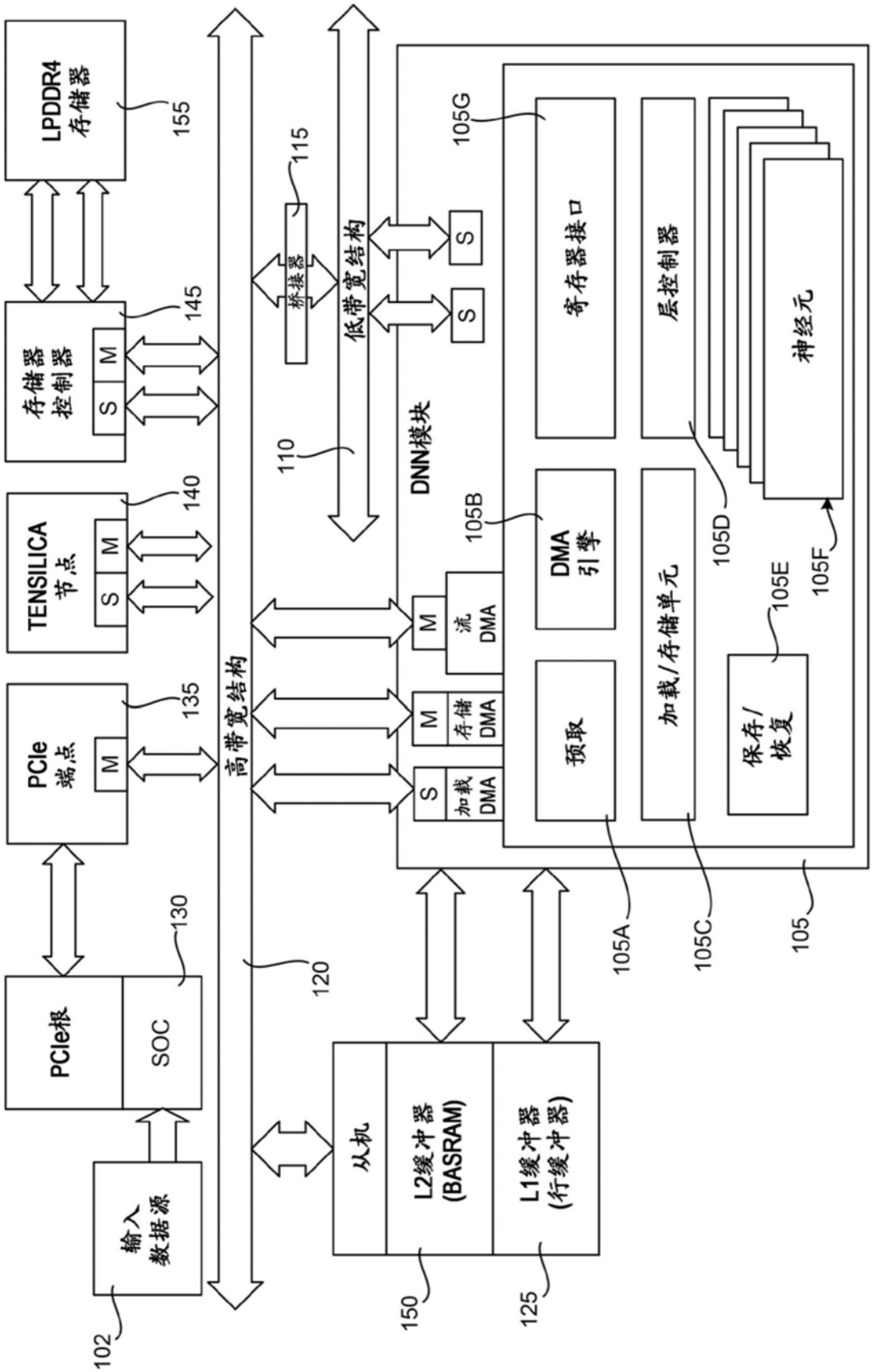

LPDDR4
存储器
155
存储器
控制器
S
M
145
TENSILICA
节点
S
M
140
PCIe
端点
M
135
PCIe根
SOC
130
输入
数据源
102
高带宽结构
120
桥接器
115
低带宽结构
110
DNN模块
105B
S
M
流
DMA
存储
DMA
加载
DMA
预取
DMA
引擎
加载/存储单元
保存/
恢复
105E
105F
105G
寄存器接口
层控制器
105D
神经元
105A
105C
105
从机
L2缓冲器
(BASRAM)
150
L1缓冲器
(行缓冲器)
125

图1

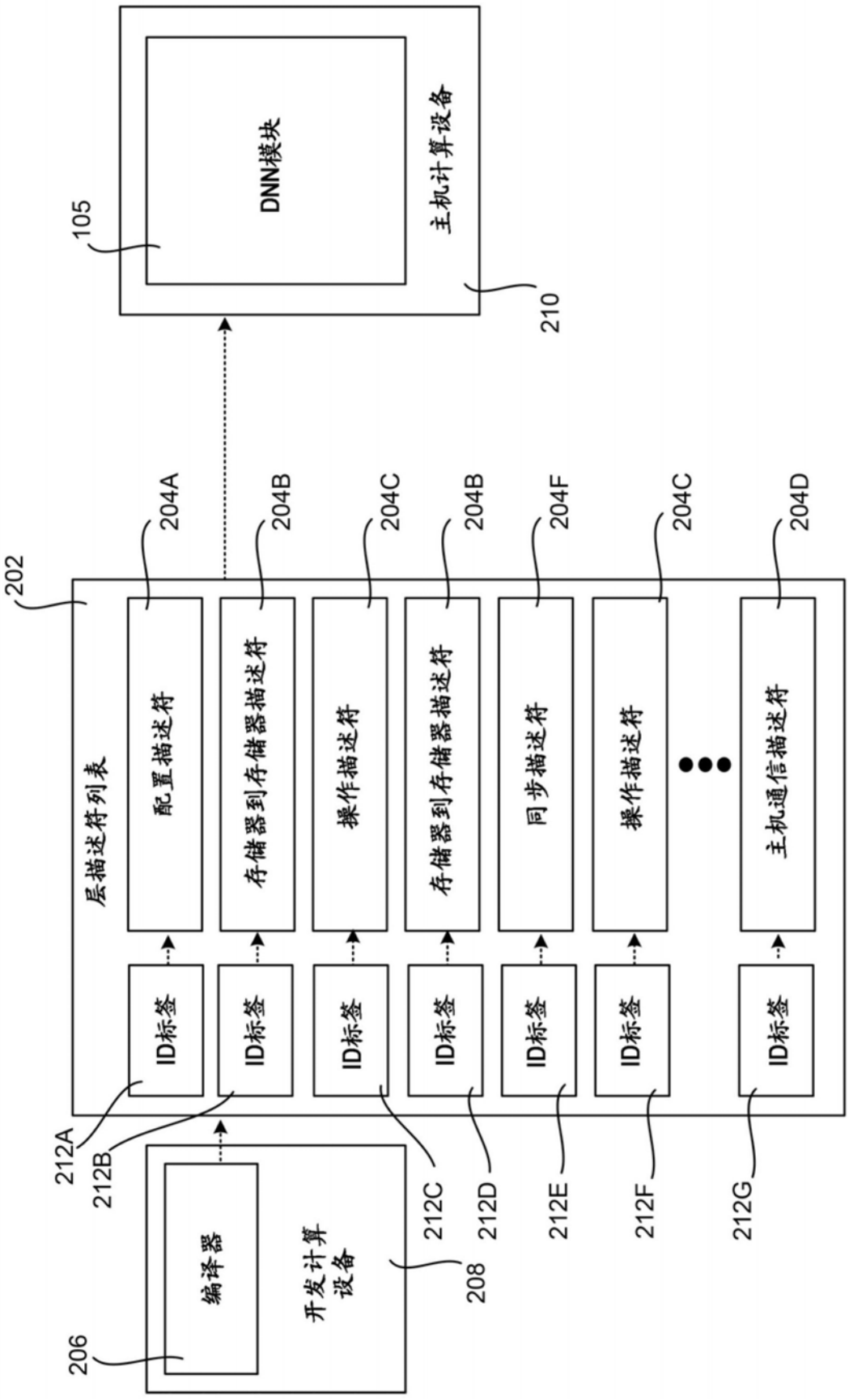

105
DNN模块
主机计算设备
210
202
204A
204B
204C
204B
204F
204C
204D
层描述符列表
配置描述符
存储器到存储器描述符
操作描述符
存储器到存储器描述符
同步描述符
操作描述符
主机通信描述符
ID标签
ID标签
ID标签
ID标签
ID标签
ID标签
ID标签
212A
212B
212C
212D
212E
212F
212G
编译器
开发计算设备
206
208

图2

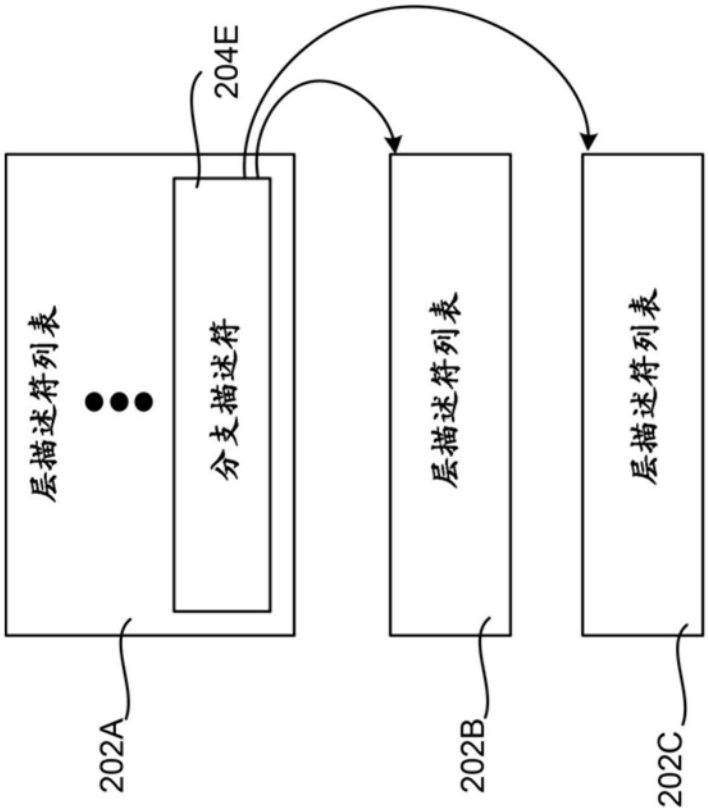


图3

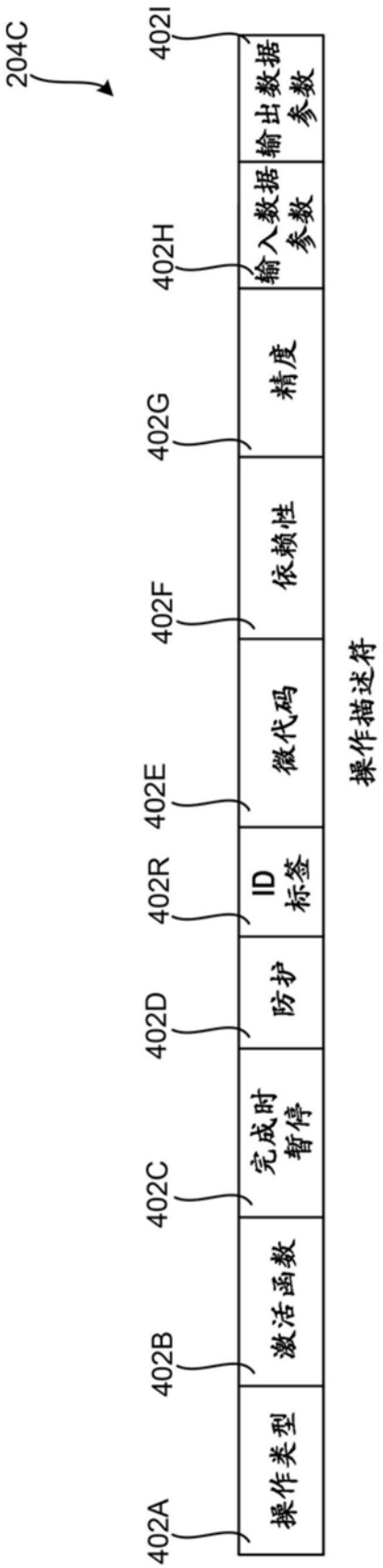

204C
操作描述符
402A
操作类型
402B
激活函数
402C
完成时暂停
402D
防护
402R
ID标签
402E
微代码
402F
依赖性
402G
精度
402H
输入数据参数
402I
输出数据参数

图4A

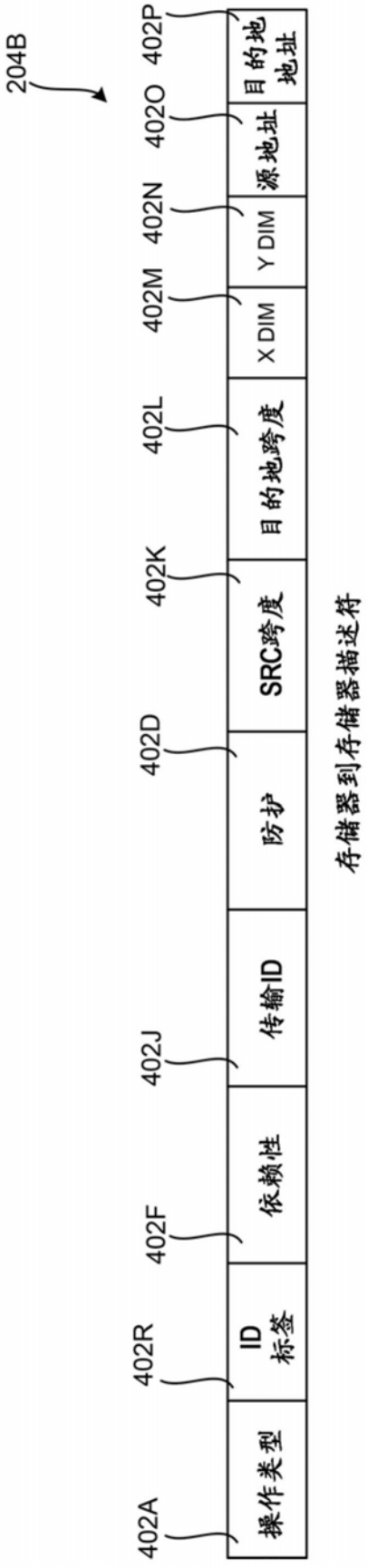

204B
402A
操作类型
402R
ID
标签
402F
依赖性
402J
传输ID
402D
防护
402K
SRC跨度
402L
目的地跨度
402M
X DIM
402N
Y DIM
402O
源地址
402P
目的地址
存储器到存储器描述符

图4B

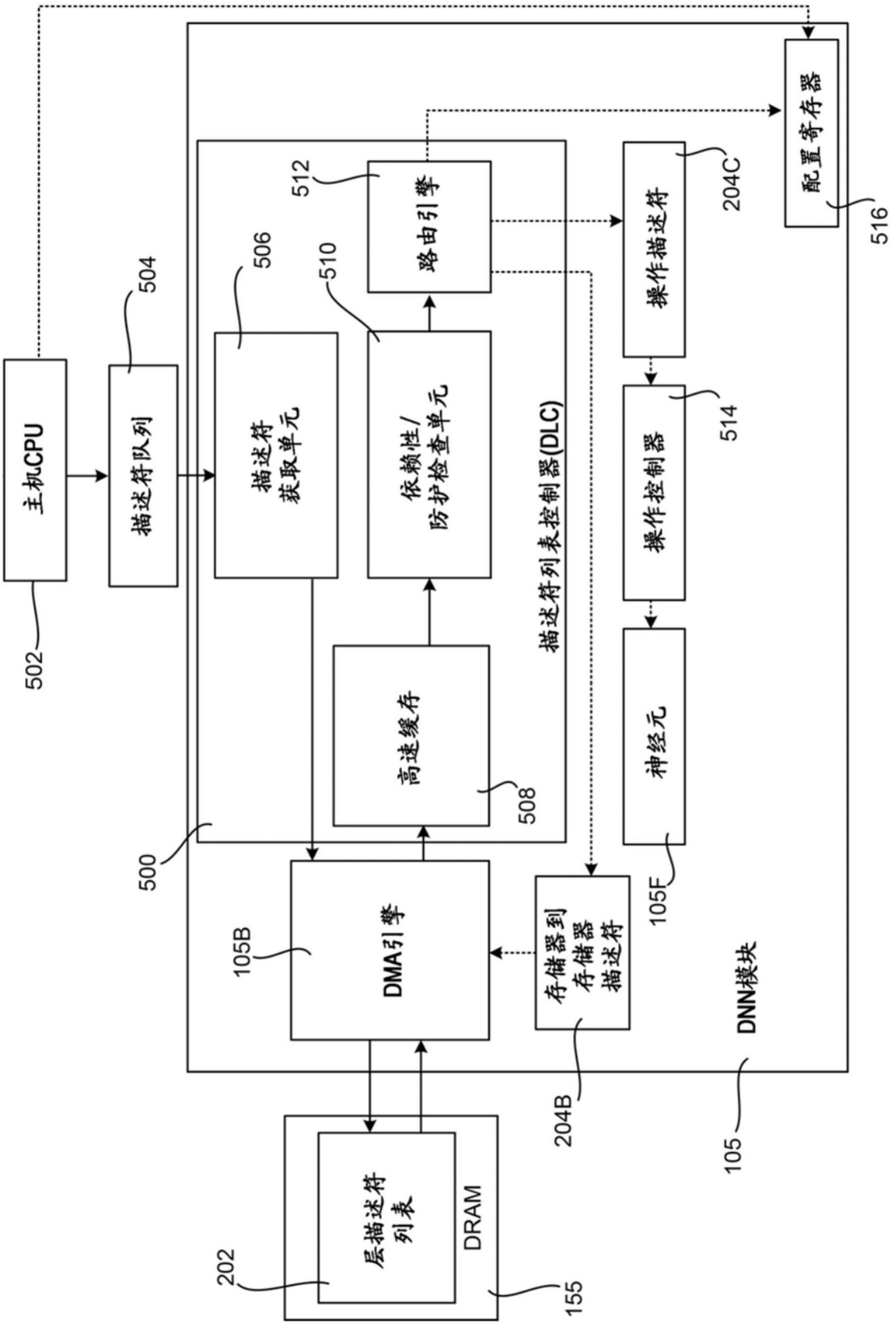

主机CPU
502
描述符队列
504
描述符
获取单元
506
依赖性/
防护检查单元
510
路由引擎
512
操作描述符
204C
配置寄存器
516
描述符列表控制器(DLC)
高速缓存
508
操作控制器
514
神经元
105F
500
DMA引擎
105B
存储器到
存储器
描述符
204B
DNN模块
105
层描述符
列表
DRAM
202
155

图5

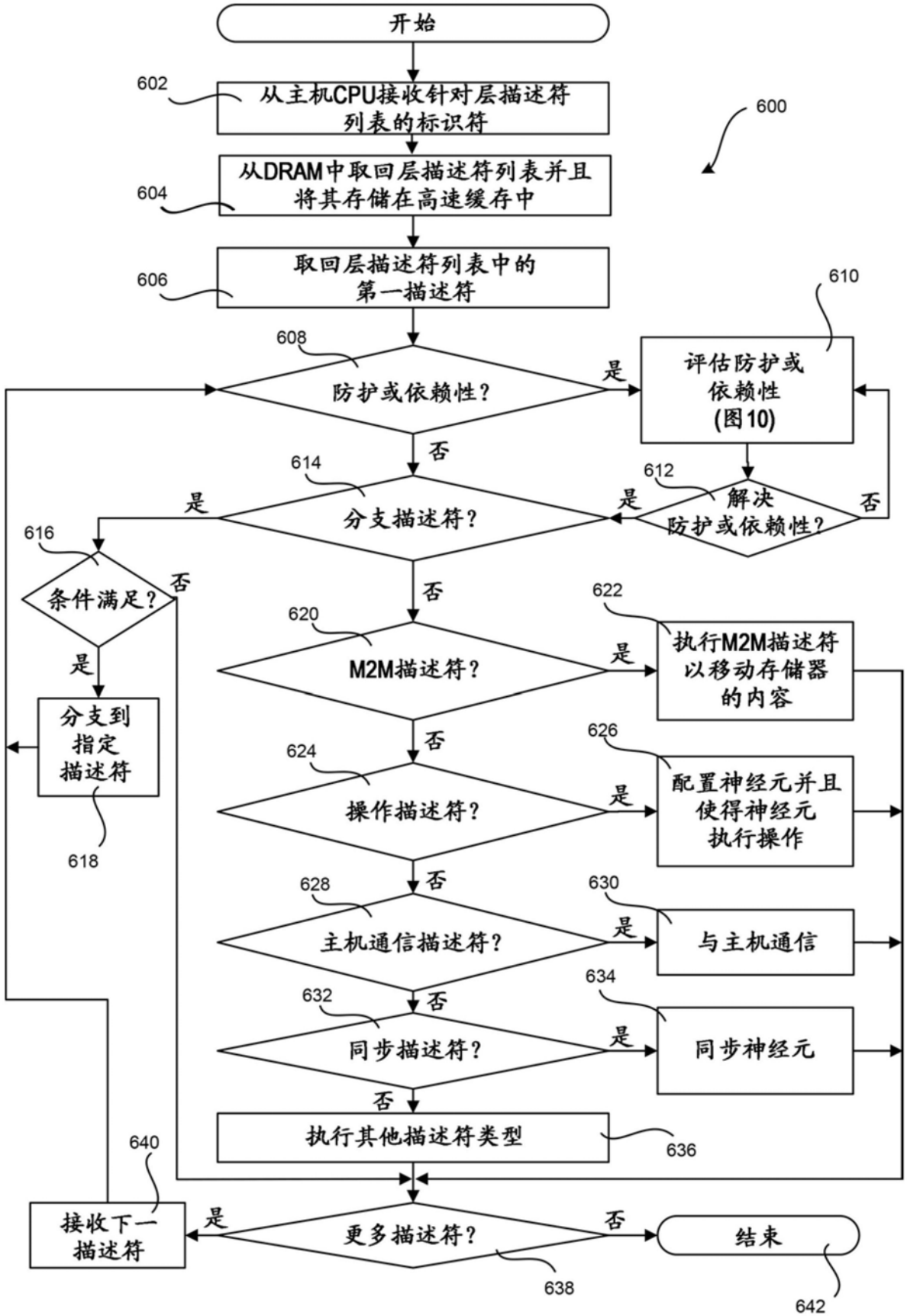

开始
602
从主机CPU接收针对层描述符列表的标识符
600
604
从DRAM中取回层描述符列表并且将其存储在高速缓存中
606
取回层描述符列表中的第一描述符
610
608
防护或依赖性？
是
评估防护或依赖性(图10)
否
614
612
分支描述符？
是
是
解决防护或依赖性？
否
616
条件满足？
否
否
620
622
M2M描述符？
是
执行M2M描述符以移动存储器的内容
是
分支到指定描述符
否
626
624
操作描述符？
是
配置神经元并且使得神经元执行操作
618
否
628
630
主机通信描述符？
是
与主机通信
否
634
632
同步描述符？
是
同步神经元
否
执行其他描述符类型
636
640
接收下一描述符
是
更多描述符？
否
结束
638
642

图6

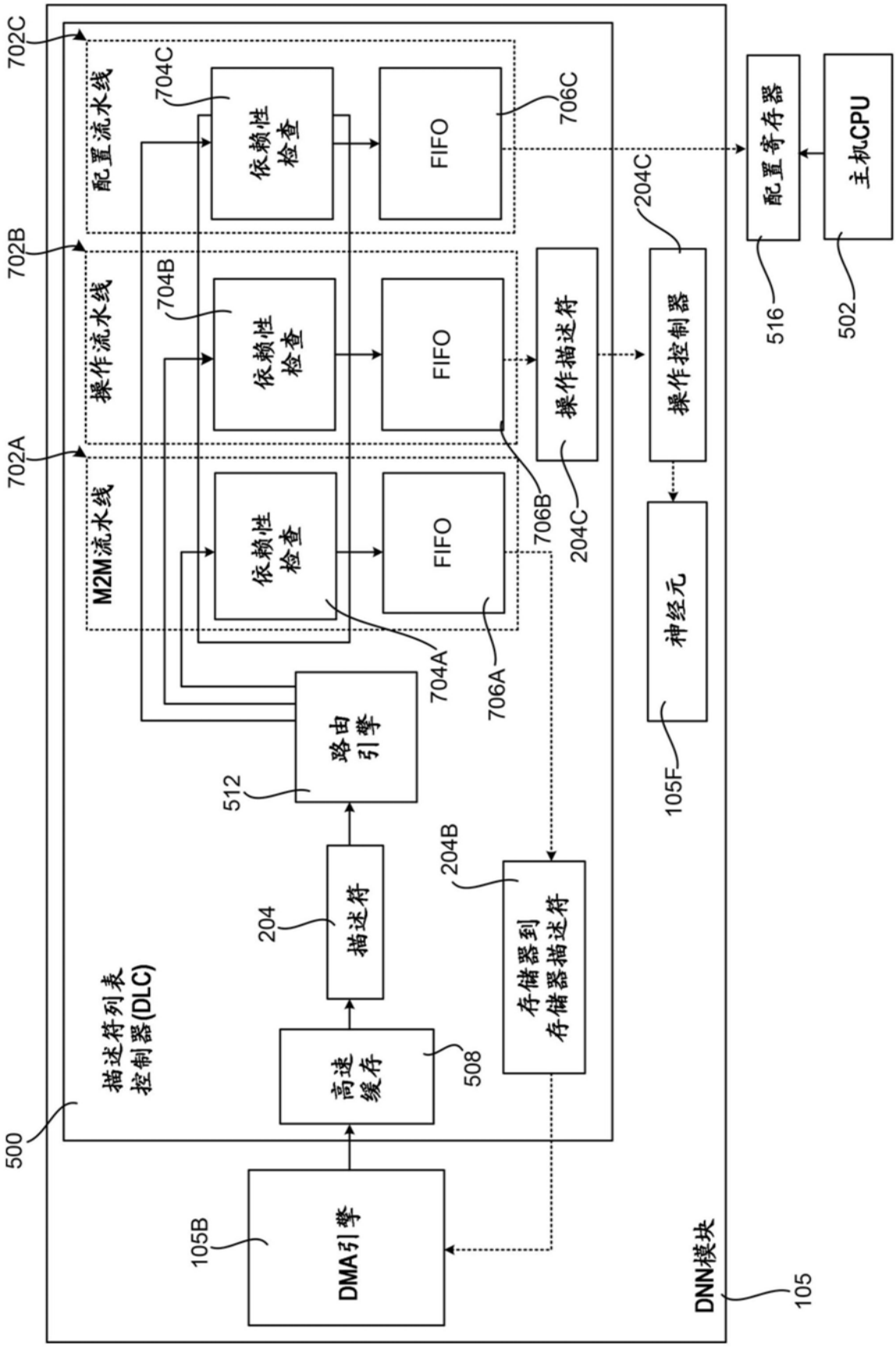

DNN模块
105
DMA引擎
105B
描述符列表
控制器(DLC)
500
高速
缓存
508
描述符
204
存储器到
存储器描述符
204B
路由
引擎
512
M2M流水线
操作流水线
配置流水线
702A
702B
702C
依赖性
检查
704A
704B
704C
FIFO
706A
706B
706C
操作描述符
204C
操作控制器
204C
神经元
105F
配置寄存器
516
主机CPU
502

图7

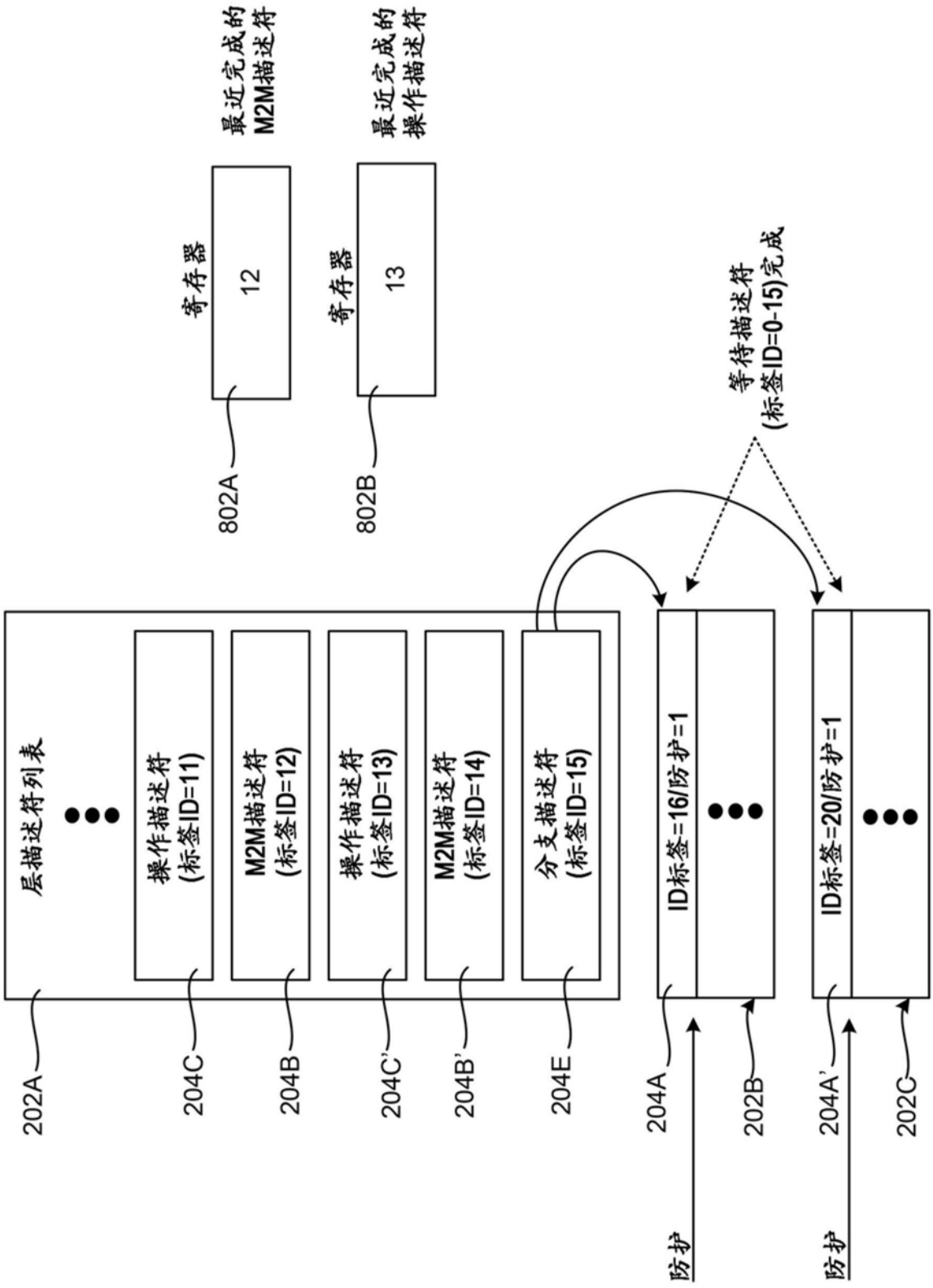

最近完成的M2M描述符
最近完成的操作描述符
寄存器
12
寄存器
13
802A
802B
等待描述符(标签ID=0-15)完成
层描述符列表
操作描述符(标签ID=11)
M2M描述符(标签ID=12)
操作描述符(标签ID=13)
M2M描述符(标签ID=14)
分支描述符(标签ID=15)
ID标签=16/防护=1
ID标签=20/防护=1
202A
204C
204B
204C'
204B'
204E
204A
202B
204A'
202C
防护
防护

图8

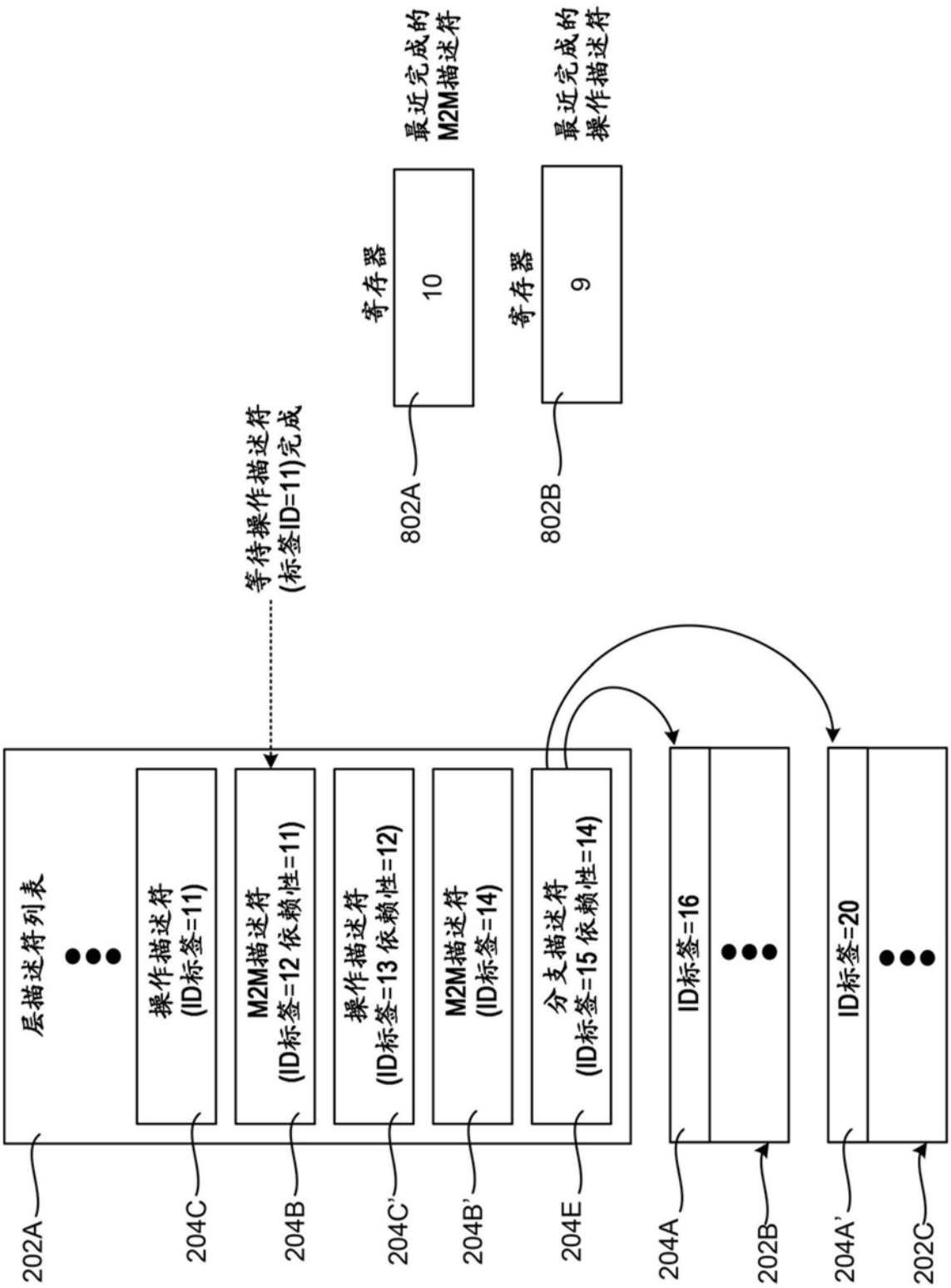

层描述符列表
操作描述符
(ID标签=11)
M2M描述符
(ID标签=12 依赖性=11)
操作描述符
(ID标签=13 依赖性=12)
M2M描述符
(ID标签=14)
分支描述符
(ID标签=15 依赖性=14)
ID标签=16
ID标签=20
202A
204C
204B
204C'
204B'
204E
204A
202B
204A'
202C
等待操作描述符
(标签ID=11)完成
寄存器
10
802A
最近完成的
M2M描述符
寄存器
9
802B
最近完成的
操作描述符

图9A

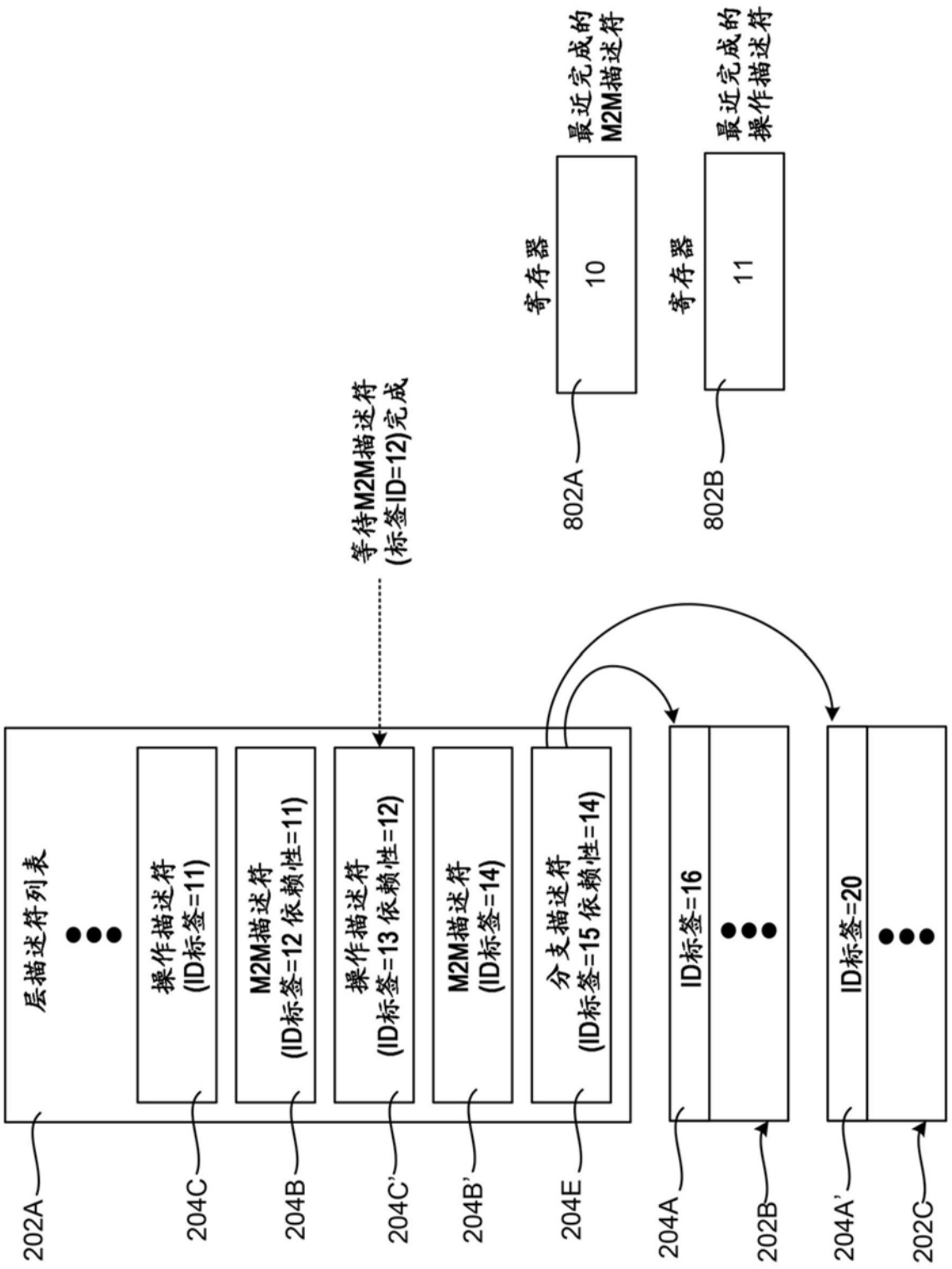

层描述符列表
操作描述符
(ID标签=11)
M2M描述符
(ID标签=12 依赖性=11)
操作描述符
(ID标签=13 依赖性=12)
M2M描述符
(ID标签=14)
分支描述符
(ID标签=15 依赖性=14)
ID标签=16
ID标签=20
等待M2M描述符
(标签ID=12)完成
寄存器
10
寄存器
11
最近完成的
M2M描述符
最近完成的
操作描述符
202A
204C
204B
204C'
204B'
204E
204A
202B
204A'
202C
802A
802B

图9B

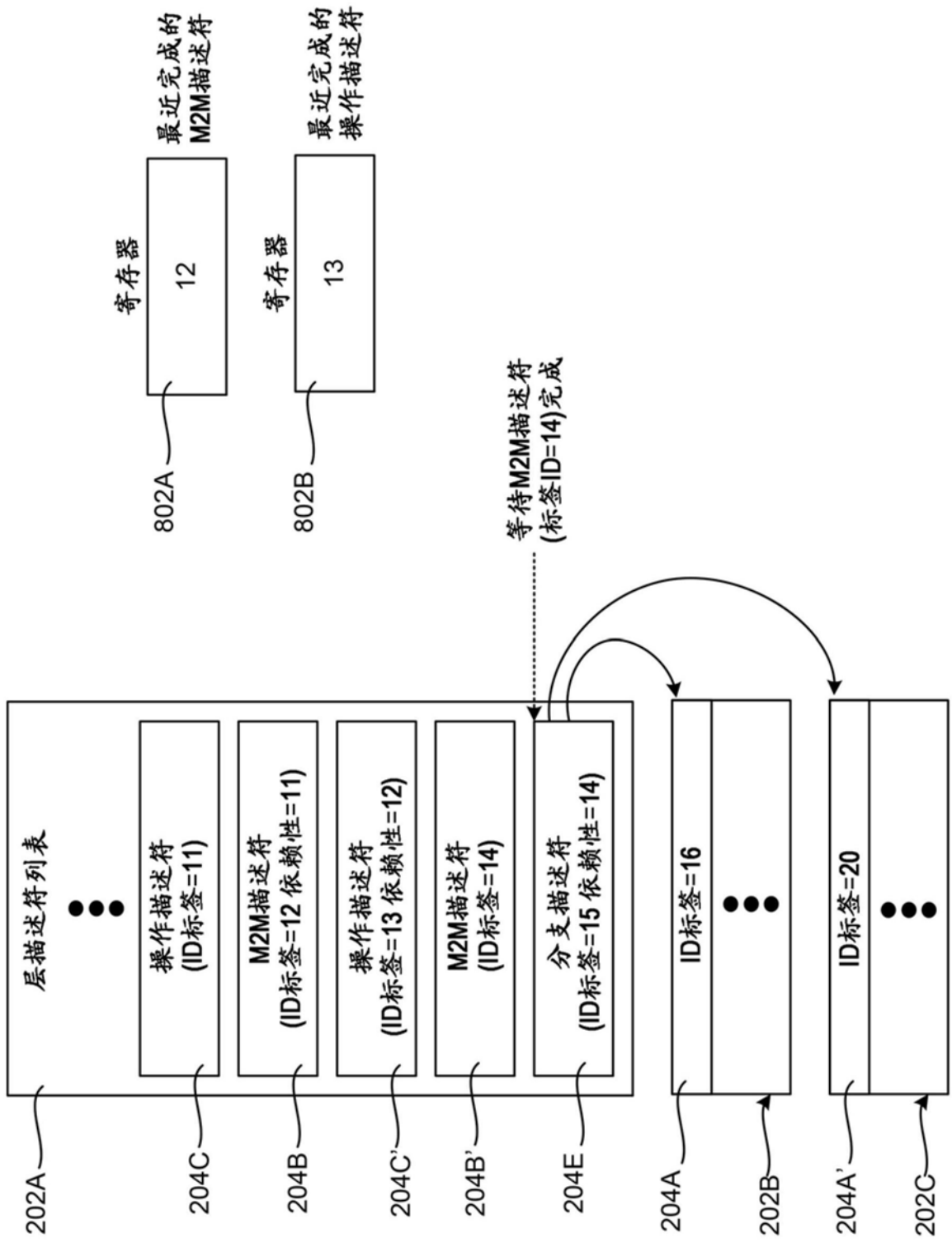

寄存器
12
最近完成的M2M描述符
802A
寄存器
13
最近完成的操作描述符
802B
等待M2M描述符(标签ID=14)完成
层描述符列表
操作描述符(ID标签=11)
M2M描述符(ID标签=12 依赖性=11)
操作描述符(ID标签=13 依赖性=12)
M2M描述符(ID标签=14)
分支描述符(ID标签=15 依赖性=14)
ID标签=16
ID标签=20
202A
204C
204B
204C'
204B'
204E
204A
202B
204A'
202C

图9C

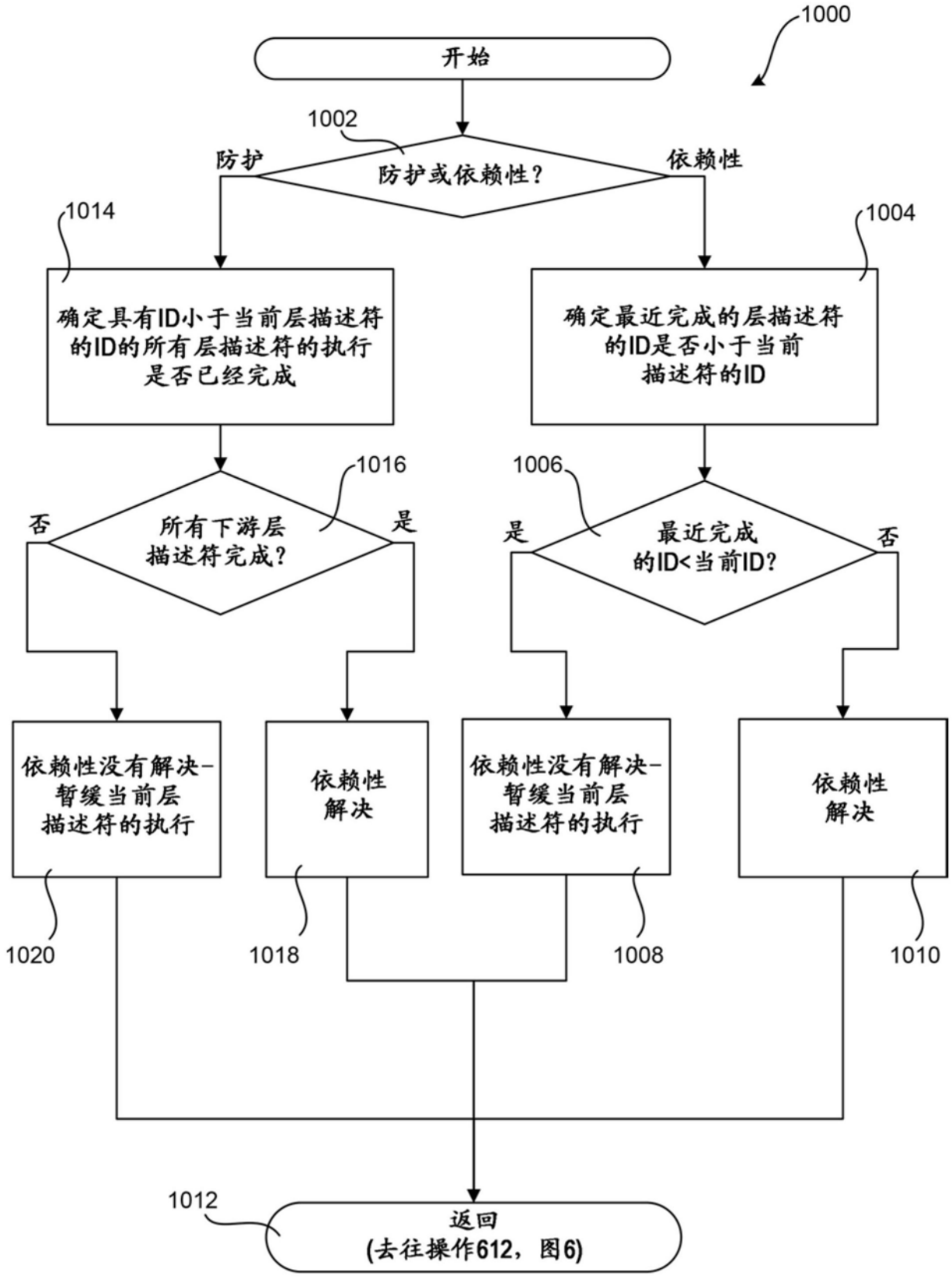

1000
开始
1002
防护
防护或依赖性?
依赖性
1014
1004
确定具有ID小于当前层描述符的ID的所有层描述符的执行是否已经完成
确定最近完成的层描述符的ID是否小于当前描述符的ID
1016
1006
否
所有下游层描述符完成?
是
是
最近完成的ID<当前ID?
否
依赖性没有解决-暂缓当前层描述符的执行
依赖性解决
依赖性没有解决-暂缓当前层描述符的执行
依赖性解决
1020
1018
1008
1010
1012
返回
(去往操作612，图6)

图10

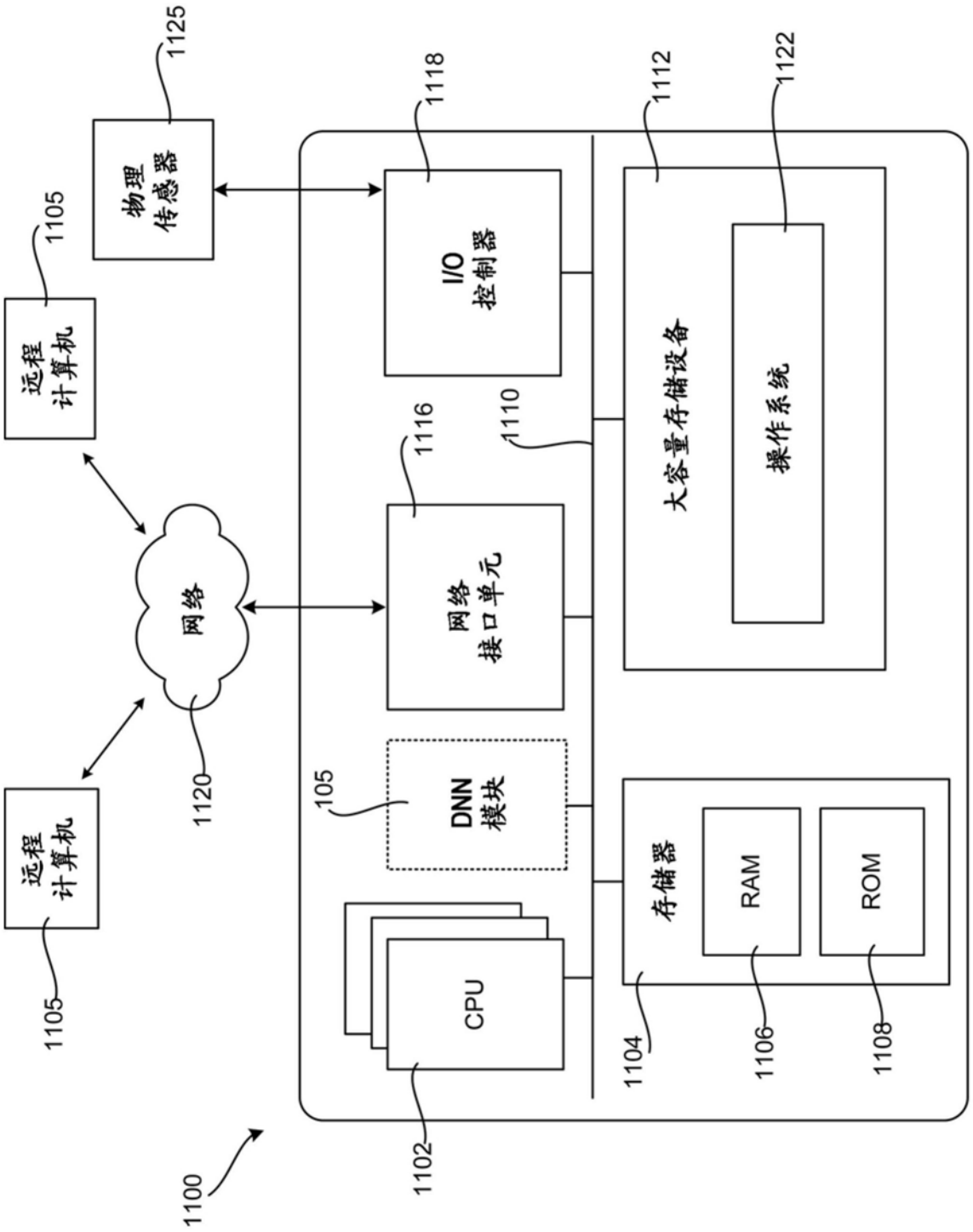

1125
物理传感器
1118
1112
1122
1105
远程计算机
I/O控制器
大容量存储设备
操作系统
1116
1110
网络
网络接口单元
1120
105
DNN模块
1105
远程计算机
存储器
RAM
ROM
CPU
1104
1106
1108
1102
1100

图11

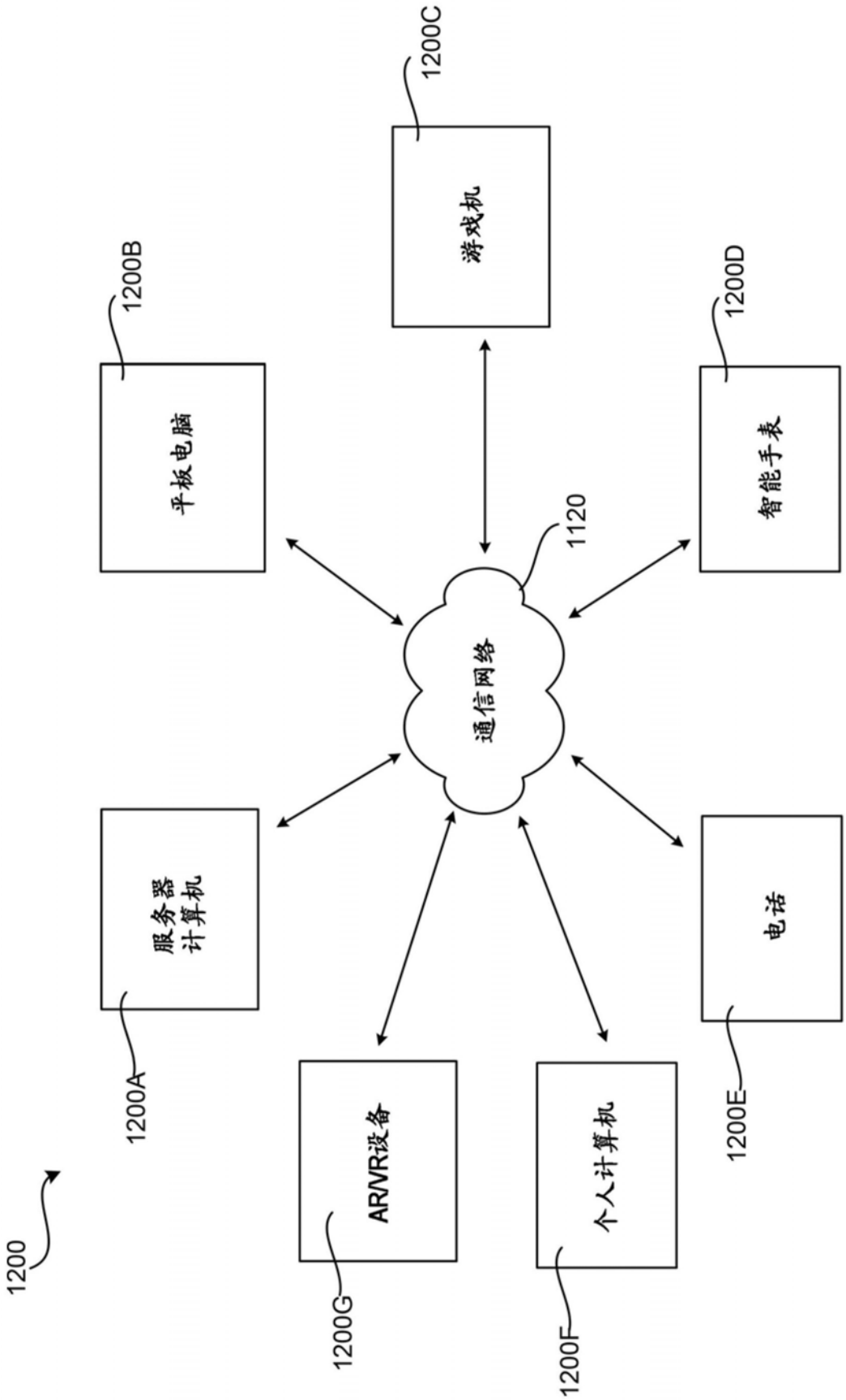

1200
1200A
服务器
计算机
1200B
平板电脑
1200C
游戏机
1200D
智能手表
1200E
电话
1200F
个人计算机
1200G
AR/VR设备
1120
通信网络

图12