MRNDA:一种基于资源受限片上网络的深度神经网络加速器组播机制研究

欧阳一鸣[1],王 奇[2*],汤飞扬[1],周 武[1],李建华[1]
(1. 合肥工业大学计算机与信息学院,安徽合肥 230009;2. 合肥工业大学微电子学院,安徽合肥 230009)

摘 要: 片上网络(Network-on-Chip,NoC)在多处理器系统中得到了广泛的应用. 近年来,有研究提出了基于NoC的深度神经网络(Deep Neural Network,DNN)加速器. 基于NoC的DNN加速器设计利用NoC连接神经元计算设备,能够极大地减少加速器对片外存储的访问从而减少加速器的分类延迟和功耗. 但是,若采用传统的单播NoC,大量的一对多数据包会极大的提高加速器的通信延迟. 并且,目前的深度神经网络规模往往非常庞大,而NoC的核心数量是有限的. 因此,文中提出了一种针对资源受限的NoC的组播方案. 该方案利用有限数量的处理单元(Processor Element,PE)来计算大型的DNN,并且利用特殊的树形组播加速网络来减少加速器的通信延迟. 仿真结果表明,和基准情况相比,本文提出的组播机制使加速器的分类延迟最高降低了86.7%,通信延迟最高降低了88.8%,而它的路由器面积和功耗仅占基准路由器的9.5%和10.3%.

关键词: 片上网络;深度神经网络加速器;组播;路由器架构;多物理网络
基金项目: 国家自然科学基金(No.61874157,No.71971151)
中图分类号: TP302 **文献标识码:** A **文章编号:** 0372-2112(2024)03-0872-13
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20220106

MRNDA: A Multicast Mechanism for Resource-Constrained Noc-Based Deep Neural Network Accelerators

OUYANG Yi-ming[1], WANG Qi[2,*], TANG Fei-yang[1], ZHOU Wu[1], LI Jian-hua[1]
(1. *School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, Anhui* 230009, *China*;
2. *School of Microelectronics, Hefei University of Technology, Hefei, Anhui* 230009, *China*)

Abstract: Network-on-Chip (NoC) devices have been widely used in multiprocessor systems. In recent years, NoC-based deep neural network (DNN) accelerators have been proposed to connect neural computing devices using NoCs. Such designs dramatically reduce off-chip memory accesses of these platforms thus reduce the accelerators' classification latency and power consumption. However, the large number of one-to-many packet transfers significantly increase the communication latency with traditional unicast channels. We proposed a multicast mechanism for resource-constrained noc-based deep neural network accelerators (MRNDA) to compute large DNN models by using limited number of processor elements (PEs). This paper proposes a tree-based multicast acceleration network to decrease the communication latency of DNN accelerators. Simulation results show that, compared with the baseline method, the multicast mechanism proposed in this paper reduces the classification latency of the accelerator by up to 86.7% and the communication latency by up to 88.8%, while its router's area and power only account for 9.5% and 10.3% of the baseline routers.

Key words: network-on-chip; deep neural network accelerator; multicast; router architecture; multiple network
Foundation Item(s): National Natural Science Foundation of China (No.61874157, No.71971151)

1 引言

片上网络(Network-on-Chip,NoC)为多核架构提供了高扩展性、高吞吐量和低延迟的互联网络[1]. 传统的总线结构只能连接有限数量的核心,并不符合当前多核心互联的需求[2]. 因此,传统的总线结构已经成为多处理器系统性能的瓶颈.

收稿日期:2022-01-21;修回日期:2022-07-15;责任编辑:王天慧
*通讯作者:王奇

NoC是一种包交换网络,是由路由器和链路组成的.目前有很多关于有线[3~5]和无线[6~9]NoC的研究.在NoC中,每个路由器会连接一个或者一组PE,路由器之间通过链路相互连接[10].不同的PE核通过NoC相互通信.在NoC中,拓扑结构决定了路由器和链路在芯片上的位置和连接的方式.2D Mesh是目前最常见的拓扑结构之一,因为它能够很好地适应芯片的几何形状,同时提供优秀的可扩展性和更高的吞吐量[2].路由算法为数据包在特定的拓扑结构中选择从源路由器到目的路由器的路径.路由器是NoC中最重要的部分,它能够接收数据包并将其存入输入缓冲器(buffer)中,根据特定的路由算法为数据包选择合适的输出端口,并将数据包转发到下一个路由器中[11].

近年来,研究者提出了新型的基于NoC的DNN设计范式[10].目前的研究提出了尖峰神经网络加速器(如SpiNNaker[12, 13])、DNN加速器(如Eyriss-v2[14], Neu-NoC[15]),以及DNN加速器模拟器(如NN-Noxim[16], CNN-Noxim[17], DNNoC-Sim[18]).这些加速器都是基于NoC架构的.NoC能够有效减少这些加速器对片外存储器的访问,并增加设计的灵活性[10].NoC高吞吐量和低延迟的特性能够有效减少这些基于多核架构加速器的通信延迟.

基于NoC的DNN加速器中的流量模式和传统研究中NoC的流量模式有很大的不同.在这种加速器中,存在大量的一对多(one-to-many)流量[19].如果采用传统单播方式来实现多播,会向网络中注入大量相同的冗余数据包,这会显著增加数据包的排队延迟,降低网络传输数据包的能力.与数据流架构(Dataflow architecture)[20]相似,基于NoC的DNN加速器也面临着数据包注入率极高的问题,网络需要为PE提供较低的传输延迟才能有效提高加速器的分类速度.

因此解决这些问题的最好方法是设计一种有效的组播机制来减少大量冗余数据包对网络带宽的影响.因此,本文提出了一种基于NoC的DNN加速器组播机制.该机制能够有效减少冗余数据包对网络带宽的冲击,从而减少加速器的分类延迟.

NoC上的PE数量往往是有限的,而目前的DNN模型却变的越来越复杂.本文主要将基于NoC的DNN加速器分为两种:基于资源不受限NoC的DNN加速器和基于资源受限NoC的DNN加速器.基于资源不受限NoC的DNN加速器指NoC的节点数量不受限制,DNN模型能够完全映射到NoC上.而基于资源受限NoC的DNN加速器指NoC的节点数量固定,DNN模型不能够完全映射到NoC上.本文主要针对资源受限的NoC平台的DNN加速器设计了一种组播机制MRNDA(Multicast mechanism for Resource-constrained Noc-based Deep neural network Accelerators).MRNDA能够使利用小型的NoC平台也能够计算大型的DNN模型,并且能够有效减少NoC中的冗余数据包数量,从而减少加速器的分类延迟.

在传统的NoC网络中一般使用多微片(multi-flit)机制传输数据包,这虽然能够减少NoC路由器的面积,但是在基于NoC的DNN加速器中会极大地增加数据包的排队延迟.因此,本文提出的方案使用单微片(single-flit)数据包传输网络中的数据.

此外,目前有很多基于多网络的NoC研究[21~23].这类方案利用多个物理网络协作以减少网络延迟或降低网络功耗.本文也采用了多网络的设计,在传统的NoC网络上额外增加一层树形组播网络.树形组播网络的结构简单,面积和功耗开销小,并且能以较低的延迟传播组播数据包,减少加速器分类延迟.

本文的贡献如下:

(1)设计了一种Tree网络和Mesh网络混合的组播方案MRNDA;

(2)为MRNDA组播方案设计了一种轻量化的路由器架构,数据包只需要更少的跳数就能到达目的节点集,并且路由器开销很小;

(3)衡量了不同组播机制对DNN加速器的影响,实验结果表明,MRNDA在分类延迟、通信延迟和面积,功耗方面都比传统方案具有一定的优势.

2　相关工作

2.1　基于片上网络的深度神经网络加速器

目前最常见的DNN加速器平台是基于CPU,GPU,ASIC和FPGA的.多核CPU(如48-core Qualcomm Centriq 2400[24], 72-core Intel Xeon Phi[25])拥有很强的计算能力.GPU在DNN计算中十分流行,因为GPU具有内在的并行计算的能力.但是CPU和GPU是为通用计算而设计的,因此在计算DNN时会消耗更多的能量[10].FPGA由于其可重编程能力,具有比ASIC更高的灵活性[26].在ASIC中,片上计算单元针对特定的应用程序[27~29]进行了优化,因此它具有比CPU和GPU更好的性能和更低的功耗[10].但是基于ASIC的DNN加速器是针对特定的DNN模型所设计的,因此与CPU和GPU相比,ASIC和FPGA在运行时的可重构能力较低[10].

目前流行的DNN模型,如AlexNet[30]和VGG-16[31],由大量的神经元和参数组成,计算复杂度很高.为了加快DNN的计算速度,加速器设计人员需要发掘其中的计算并行性.由于DNN的多层以及每层具有的并行神经元结构,加速器可以采用多核架构设计.因此,有研究提出了基于NoC的DNN加速器架构,因为NoC具有为多核系统提供高带宽和低延迟通信的能力.

文献[10]提出了一种基于NoC的DNN加速器设计. 作者分析了传统神经网络加速器的优缺点,提出了一种基于NoC的DNN加速器设计方案. 该设计解耦了DNN操作:在PE中进行计算,在NoC中进行数据传输. 该基于NoC的DNN加速器设计支持不同的DNN数据流,增加了设计灵活性和运行时可重配置性. 实验结果表明,该方法大大减少了片外存储器访问,降低了功耗,提高了分类速度.

文献[15]提出了DNN加速器称为Neu-NoC. Neu-NoC是一种ring-mesh混合网络. 在Neu-NoC中,同一层的神经元通过一个环形网络连接,同一个环共享相同的数据以提升数据包传输效率. 作者还分析了加速器中数据流量的特点,提出了一种复杂的神经网络感知映射算法和组播传输方案. 该方案均衡了网络流量,减少了冗余流量. 但是,这种组播方案需要在数据包中添加大量的地址位,这会增加硬件开销. 同时,环形网络会减少网络整体的吞吐量并增加传输延迟.

文献[16~18]提出了一系列的基于Noxim[32]开发的基于NoC的DNN加速器. NN-Noxim[16]是一个周期精确的NoC模拟器,可以支持人工神经网络(Artificial Neural Network, ANN)模型. 该模拟器将神经元簇映射到一个NoC上,并执行计算任务. CNN-Noxim[17]是专为卷积神经网络(Convolutional Neural Networks, CNN)模型设计的. 该模拟器首先将CNN模型平坦化并转化为ANN模型,然后将神经元簇映射到NoC上. DNNoC-Sim[18]将神经元动态映射到NoC上,适用于资源受限的NoC平台上的大规模DNN模型. 本文参考了文献[16~18]的设计,假设DNN模型的训练是在服务器上进行的,部署到边缘设备的DNN模型只需要完成前向传播的计算即可. 实验模拟器的搭建也与文献[16~18]的模拟器结构基本相同.

2.2 组播机制和多网络

广播(broadcast)和组播(multicast)是将同一消息从源节点发送到多个目标节点的流量模式. 在ANN或DNN模型中,每一层的所有神经元都必须将自己的计算结果发送到下一层的所有神经元中,然后才能进行计算. 因此,必须为基于NoC的DNN加速器设计合适的组播机制.

最常用的组播路由算法通常是基于树的[33, 34]和基于路径的[35, 36]. 在基于树的算法中,多播包首先尽可能沿公共路径传输,然后根据目的节点的位置和组播路由算法将该数据包复制为多个副本,路由器再将这些副本传输到不同的目的节点. 在基于路径的组播路由算法中,路由算法会将组播数据包的所有目的节点串连到一条公共路径上,然后再将数据包通过这条公共路径传输到每个目的节点. 在基于路径的方法中,数据包只有在到达目的地节点时才被复制. 在基于树的组播路由算法中,阻塞一个分支会影响另一个分支的传输. 因此基于树的方法容易发生拥塞,但平均数据包跳数比基于路径的方法短. 基于路径的方法不容易阻塞,但平均数据包跳数更长.

研究人员针对NoC提出了许多组播机制. 一种名为VCTM(Virtual Circuit Tree Multicast)[37]的组播机制在发送组播包之前会先发送一个建立组播树的建立包. 组播报文通过建立的组播树进行传输. VCTM需要额外的存储空间来维护多播树,这会增加芯片面积和功耗. VCTM还会使用额外的周期发送组播树的设置包,这会增加网络的传输延迟. VCTM还需要构建大量的多播树来服务于不同的节点集,这限制了VCTM的可伸缩性. 另一种名为RPM(Recursive Partitioning Multicast)的组播机制[2]根据数据包的位置将整个网络分成8个部分. 数据包的输出端口由数据包的目的节点在这8个部分中的分布决定. 与VCTM相比,RPM不需要在路由器中设置额外的VCT和CAM表,这能够减少路由器的面积和功耗. RPM数据包头flit包含与网络节点数相同长度的比特位. 这些比特位记录了数据包的目的节点的位置. 在网络规模较大的情况下,RPM的硬件开销会增大. RPM使用两个虚拟子网来避免死锁,这可能会造成流量不平衡并降低网络带宽.

MMNNN(Multicast Mechanism for NoC-based deep Neural Network)[38]是一种专用于基于NoC的DNN加速器的组播机制. MMNNN组播机制主要包括一个基于树的组播路由算法和能够支持该组播路由算法的组播路由器. MMNNN中的组播路由算法根据神经元簇在片上网络中的空间局部性来控制数据包的分裂并保证传输路径的最小化和数据包数量的最小化. MMNNN中的路由器架构是无头阻塞的,并且能够减少复制数据包带来的额外功耗开销. 但是MMNNN只能够支持能够将神经网络模型完全映射到NoC平台的情况,不适用于NoC资源受限的DNN加速器.

SmartFork[39]是一种专门用于组播路由算法的路由器微体系结构. SmartFork是针对传统Mesh网络设计的,它对输出端口进行分组,组内端口串行服务,而组间端口并行服务. 该路由器在保证无死锁的同时,增加了交换级的限制,使路由器在没有引入高硬件成本的情况下增加了输入缓冲区的利用和分配效率.

多网络(Multiple networks)常用于扩展NoC的带宽并避免具有不同消息类的协议级死锁. 目前也有研究发现,多网络可以很好地用于功率门控以及低延迟NoC的设计.

文献[21]提出了一种Tree-Mesh混合的NoC设计. 该设计将网络直径低但吞吐量低的树形NoC和网络直径高但吞吐量高的Mesh NoC结合,在网络负载较低的情况下使用树形NoC减少零负载延迟,在网络负载较

高的情况下使用Mesh NoC保证高吞吐量.

Catnap[22]将传统的单NoC(Single NoC)划分成了具有4个不同优先级子网的多NoC(Multiple NoC). Catnap会保持0号子网一直开启,以此来降低低流量情况下数据包的传输延迟. 在0子网拥塞程度很低的情况下,其他子网保持关闭状态以节省功耗. 在0子网拥塞程度较高的情况下,会打开1子网的路由器传输数据包,增大NoC的带宽. 以此类推,在底层子网判定为拥塞时会打开上层子网的路由器,从而最终打开所有子网. 该方案不仅减少了低流量负载情况下网络的数据包延迟,而且也节省了相当可观的静态功耗.

本文参考了传统基于NoC的DNN加速器平台设计方案,设计了一种基于资源受限NoC的DNN加速器平台. 该平台使用了传统单播Mesh-NoC和组播Tree-NoC相结合的方案,减少了网络中冗余数据包的数量,减少了数据包的平均传输延迟,减少了加速器的分类延迟.

3　MRNDA设计

在本节中,我们将先介绍基准的基于资源受限NoC的DNN加速器的基本架构,然后再介绍我们的组播机制设计方案.

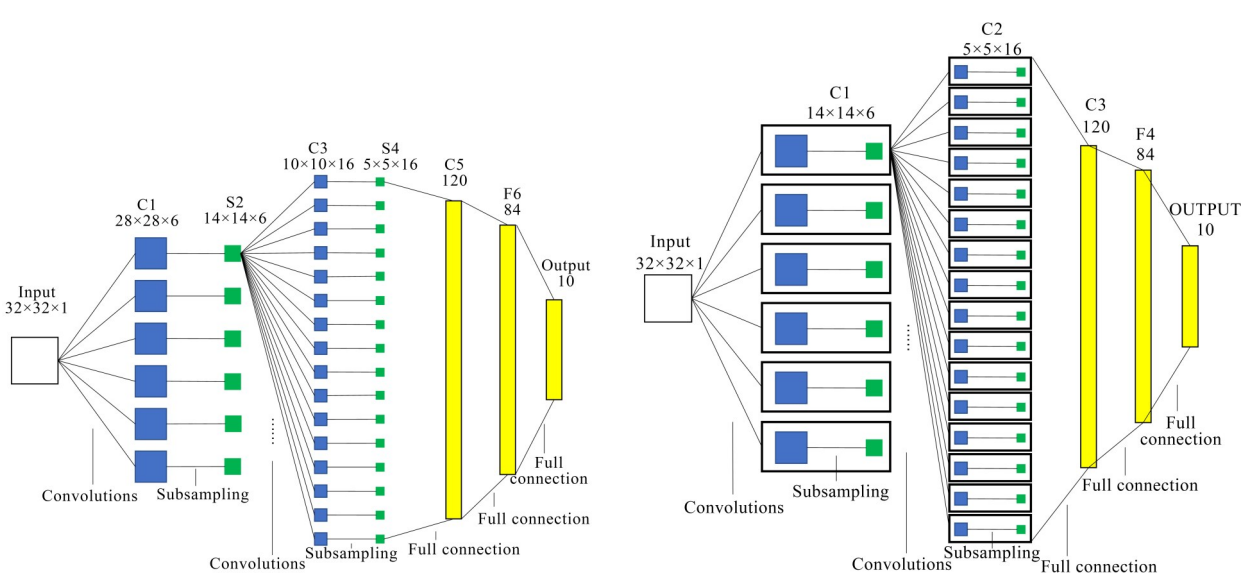
3.1　基于资源受限NoC的DNN加速器基准架构

3.1.1　神经网络的变形(Reshape)、聚簇(Clustering)和映射(Mapping)

本节将主要介绍神经网络的变形、聚簇和映射过程的具体细节. 在基于NoC的DNN加速器中,若每个PE只计算一个神经元,那么将需要在网络中添加大量的PE,并且NoC会传输过多数量的数据包,这会严重影响加速器的网络传输速率,也不适用于资源受限的NoC. 因此,我们需要设计合适的变形、聚簇和映射策略使一个PE能够计算多个神经元,并保证网络负载处于一个合理的范围之内.

本文主要研究DNN中的卷积神经网络(Convolutional Neural Network, CNN)的前向传播过程. 我们将以一个经典的CNN模型——Lenet-5[40]来演示我们的变形、聚簇和映射策略.

图1(*a*)为Lenet-5的基本结构,它主要由输入层、卷积层、池化层、全连接层和输出层组成. 从图1中可以看出,在Lenet-5中,卷积层和池化层是一一对应的,也就是说在卷积层的神经元之后往往会有一个池化层的神经元与之相连. 池化层的主要作用是减少卷积层输出的特征图(feature map)的大小,同时保留特征图的主要特征. 如果我们将所有卷积层和池化层的神经元都映射到网络中,网络将传输大量的特征图数据包,从而降低网络的传输速率. 因此,将池化层神经元及其卷积层对应的神经元映射到相同的PE上可以减少传输数据包的数量. 池化层神经元可以在PE中直接利用卷积层的输出特征图进行计算,PE输出的计算结果是已经降维后的特征图,这能够进一步降低网络中传输数据包的数量. 并且加速器不需要再为池化层单独映射一次,从而能够降低访存次数.



(*a*) Lenet-5基本结构　　(*b*) Lenet-5变形后结构

图1　Lenet-5网络模型架构转换

因此,如图1(b)所示,我们将卷积层和池化层的对应神经元合并一个神经元,这样就可以将网络变形为一个类似于全连接的ANN网络. 由于每个PE会计算多个神经元的计算任务,因此我们需要确定每个PE计

算神经元的数目. 在此,我们定义一个MPC值,MPC值代表每一层的神经元可以占用多少个PE进行计算,那么每个PE承担计算的神经元的数目为⌊NM/MPC⌋(NM表示每层有多少个神经元,且⌊NM/MPC⌋最小值为1),网络中最后一个PE承担神经元的数量为NM−⌊NM/MPC⌋×(MPC−1),这也意味着我们将同一层⌊NM/MPC⌋个神经元聚为一簇. 图2为MPC为2时Lenet-5的聚簇情况.

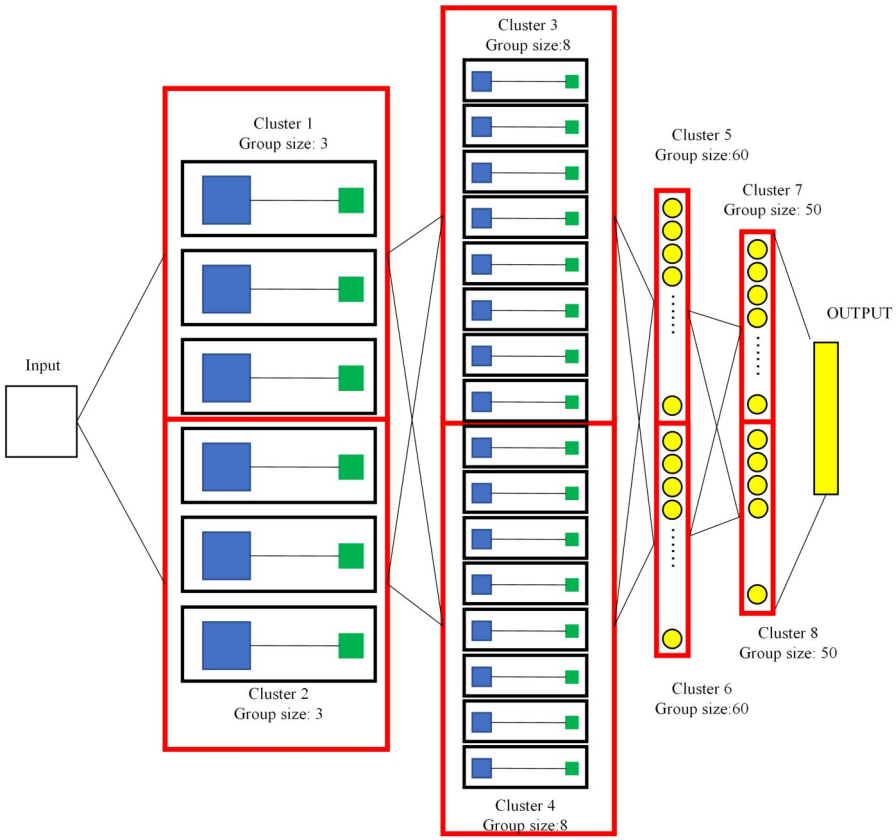


图2 聚簇策略

当神经网络聚簇完毕后,我们将每一层的簇按照Mesh网络的X轴正方向依次映射到NoC上. 具体的映射策略和NoC平台的数据流将在下一小节具体介绍.

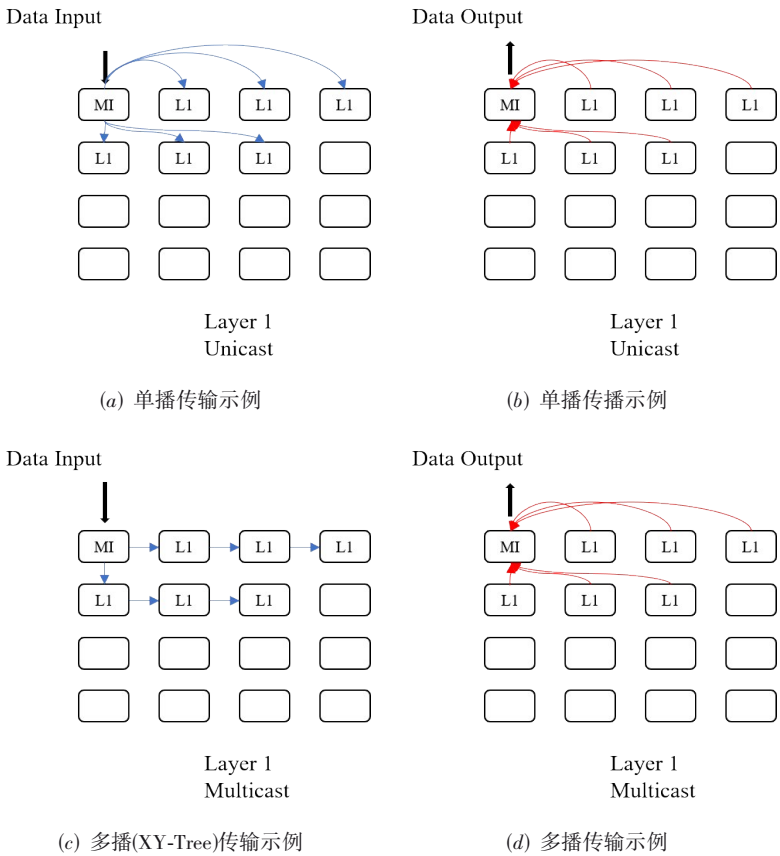
3.1.2 基于资源受限NoC的DNN加速器工作流程

本文参考了文献[18]和文献[41]的基于NoC的DNN加速器的设计,提出了如下文所示的基于资源受限NoC的DNN加速器基准设计. 如图3所示,本文使用4×4的Mesh NoC作为基准平台. 在图3中,MI(Memory Interface)为内存接口,作用是从内存中取出上一层神经元的计算结果并发送给现在网络中正在计算神经元的PE,并将网络中PE计算完成的结果输出至内存. MI在NoC上固定占用0号节点(左上角节点). 由于MI固定占用一个节点,因此在本文所示的平台中,MPC(Maximum number of PEs that can be occupied for each Convolutional layer)值固定为15.

由于PE的数量是有限的,而深度神经网络往往规模都比较庞大,因此为了利用有限数量的PE,本文将每一层的神经元簇按照时间次序映射到NoC上. 以Lenet-5为例,如图3(a)所示,MI首先会取出输入层的输入图片发送至映射到NoC上的第一层的神经元簇(L1). 当第一层的神经元簇的计算任务完成,每个PE会将计算结果发送至MI节点,将计算结果存入内存中. 然后,再将第二层的神经元簇映射到NoC上,MI发送第一层的计算结果至承担计算任务的PE中,每个PE计算完成后再将计算结果发送到MI,以此类推,直至完成所有层的计算任务.

图3(a)和图3(b)是以单播方式来传输数据的. 图3(a)以单播方式将上一层的计算结果传送到需要的PE中,从中我们可以发现,在这个过程中所传输的数据包内容都是相同的,因此这些数据包会降低NoC的传输速率. 由于每个神经元的计算结果可能都是不同的,因此要以单播方式将计算结果发送到MI中,图3(b)和图3(d)均是如此. 而图3(c)则使用XY-Tree组播路由算法来传输组播数据包. XY-Tree是最经典的路由算法之一. 在XY-Tree组播路由算法中,数据包从源节点到所有目的节点的路径都遵循XY路由算法. 使用XY-Tree组播路

由算法能够减少冗余数据包的传输数量,因此XY-Tree的传输速率会比单播方式快很多. 本文参考了RPM[2]的路由器架构设计,修改了传统五端口Mesh路由器,使其能够支持组播路由算法.



(*a*) 单播传输示例　(*b*) 单播传播示例

(*c*) 多播(XY-Tree)传输示例　(*d*) 多播传输示例

图3　基于资源受限NoC的DNN加速器数据传输过程示例

3.2　MRNDA

我们在研究过程中发现,考虑到现有的DNN模型、映射策略和NoC的大小,MI节点往往需要传输数据包到所有PE当中,在图3(*c*)的组播方法中,MI传输数据包到所有PE节点时会开启所有的路由器,这会降低NoC的传输速率并消耗大量的功耗. 因此,我们需要提出一种延迟和功耗更低的组播机制.

本文提出了一种针对基于片上资源受限NoC的DNN加速器组播机制,具体工作流程如图4所示.

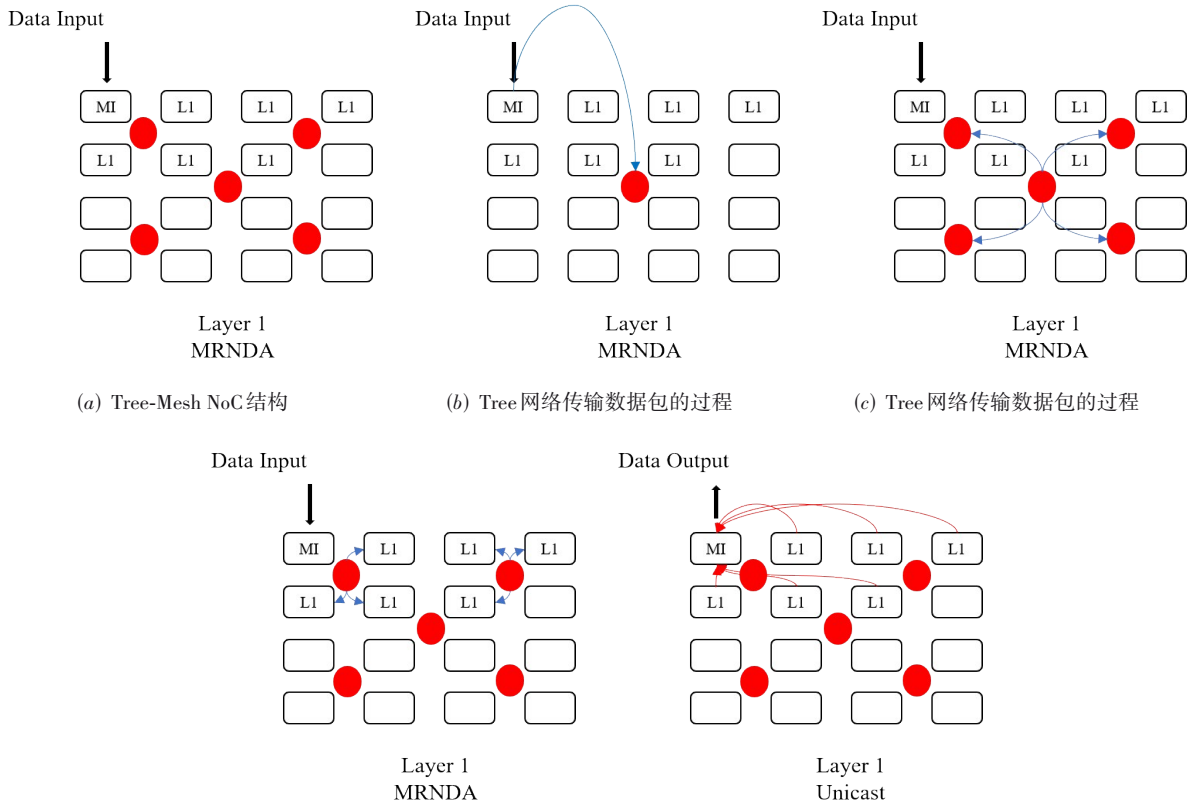
如图4(*a*)所示,在MRNDA中,NoC除了由传统Mesh网络的基准路由器组成外,还由5个Tree路由器构成,其中1个为根路由器,其余4个为叶子路由器. 根路由器连接MI节点和所有叶子路由器. 叶子路由器连接相邻4个PE.

如图4(*b*)所示,当MI需要传输一个组播数据包至网络中的PE时,MI会将数据包传送到根路由器中. 当叶子路由器连接的PE需要该数据包时,PE会向叶子路由器发送一个举手信号,而若叶子路由器需要数据包的话(收到了PE的举手信号)则会向根路由器发送请求数据包的举手信号. 如图4(*c*)所示,根路由器根据叶子路由器的请求会将数据包发送到有请求的叶子路由器中. 如图4(*d*)所示,叶子路由器收到数据包时,会根据PE的请求将数据包发送到对应PE中. 如图4(*e*)所示,当所有承担计算任务的PE计算完成之后,会利用NoC中Mesh网络的单播功能将计算结果发送到MI中.

由于本文采用了多网络机制,文中的Tree网络和Mesh网络的运行是互不干扰的,因此它们是可以同时运行的. 所以,图4(*b*)~(*e*)所示的过程是可以同时进行的. 在PE将自己的计算任务完成之后,它们就可以向MI节点发送计算结果. MI节点收到计算结果后,就可以将结果作为下一层的输入通过Tree网络发送到需要的PE中. 这样就可以并行的利用Tree网络和Mesh网络,减少了整体的分类延迟.

对于本文的路由决策机制而言,最重要的就是PE能够根据自身的需要向叶子路由器举手,叶子路由器收到了举手信号之后才能向上级路由器举手. 算法1为PE如何根据需要发送举手信号的伪代码.

算法1的输入主要有4个. neural_network_size是PE中存储的所要运行的神经网络模型每一层的神经元数量;current_layer则是PE当前需要收到神经网络哪一层的数据;MPC的定义和上文相同,指的是每一层的神



(*a*) Tree-Mesh NoC结构 (*b*) Tree网络传输数据包的过程 (*c*) Tree网络传输数据包的过程

(*d*) Tree网络传输数据包的过程 (*e*) PE计算完成后发送计算结果至MI

图4 MRNDA工作过程

算法1 PE举手过程伪代码

输入: array<int> neural_network_size, int current_layer, int MPC, int local_id

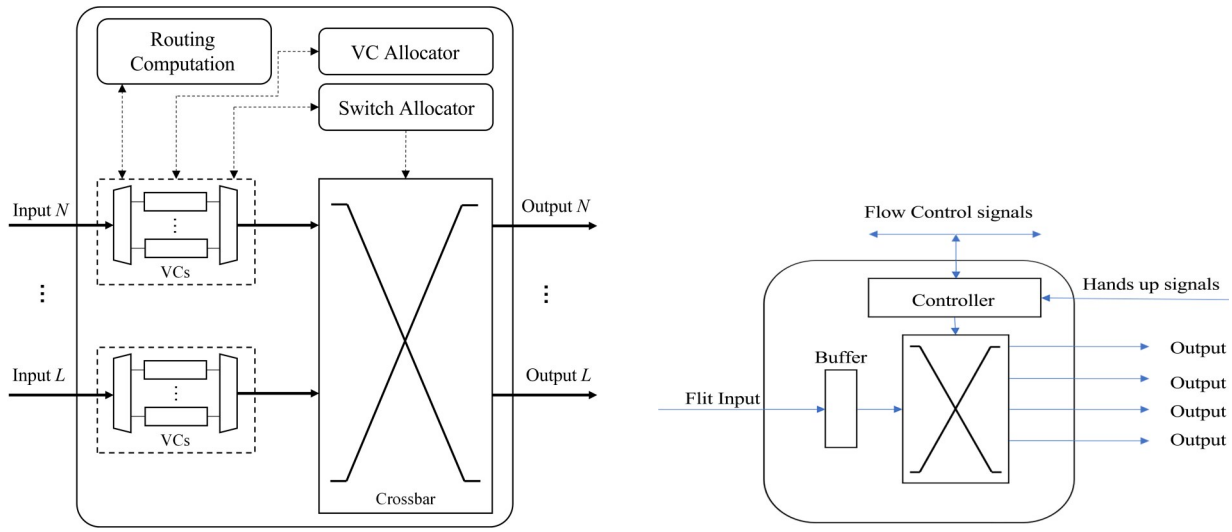
输出: bool hands_up_signal

```
IF (neural_network_size[current_layer] < MPC) THEN
  IF (local_id > neural_network_size[current_layer]) THEN
    hands_up_signal = false
  ELSE
    hands_up_signal = true
  ENDIF
ELSE
  hands_up_signal = true
ENDIF
```

经元可以占用多少个PE进行计算;local_id则是代表当前PE在Mesh网络中的地址. 算法1的输出是PE是否向叶子路由器发送举手信号hands_up_signal. 由于本文的映射方案会尽可能地利用PE来进行计算,因此只有当神经网络某一层的神经元数量小于MPC时,才可能会出现某些PE不会发送举手信号至路由器的情况. 只有neural_network_size[current_layer]小于MPC,且local_id的值大于neural_network_size[current_layer],才说明该PE不会承担计算任务,不需要发送举手信号.

图5为基准路由器和MRNDA路由器架构示意图. MRNDA路由器只需要一个输入端口即可. 当flit从上游传输过来时,若buffer为空,则会被存放至buffer中. Controller负责接收下游路由器或PE的举手信号,检查输入的数据包需要输出到哪些输出端口,并且和下游路由器或PE进行通信,确定是否可以将flit传输到目的地中. 当下游路由器或PE可以接收flit时,Controller会打开交叉开关,使flit分裂到对应的输出端口中. 当所有输出端口都传输完毕,Controller会释放flit占用的buffer,使新的flit进入路由器中.

本文参考了Noxim[32]的路由器的握手方式,在MRNDA路由器采用ABP协议(Alternating Bit Protocol)进行握手,该协议主要用于实现点对点的异步通信. 图6展示了MRNDA路由器发送端口和接收端口之间的信号线. 发送端口会首先通过buffer_info端口读取下游接收端口buffer的状态信息,若接收端口buffer非空,且有flit需要发送时,则会通过置反tx_req信号向接收端口发送传输flit的请求. 在发送端口中,tx_req主要用于发送传输flit的请求信号,接收端口通过rx_req收到相应请求后会开启buffer从flit_channel_tx中读取flit,



(a) 基准路由器架构　　(b) MRNDA路由器架构

图5　路由器架构

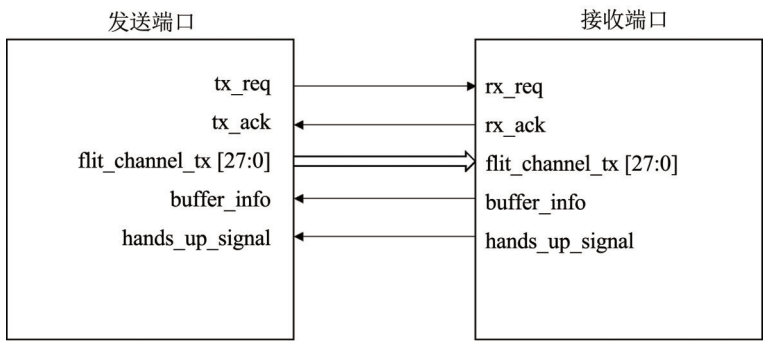


图6　发送端口和接收端口握信号

读取完成后将rx_ack信号置反,表示flit的传输结束.发送端口会通过hands_up_signal读取下游路由器的举手信号并传输给路由计算单元当中.

3.3　数据包格式

目前,NoC一般采用多flit(multi-flit)机制来减少缓冲区和数据总线的尺寸.多flit机制将一个较大的数据包分解成若干个flit,使用flit作为网络中最小的传输单元.然而,在基于NoC的深度神经网络加速器中,数据包的注入率非常高,并且加速器的分类速度对通信延迟很敏感.采用多flit机制会提高网络中数据包的排队延迟,从而增加数据包的传输时间.此外,若使用多flit机制,同一数据包的不同flit可能会分布在多个路由器上,这会进一步增加网络中的拥塞.

在基于NoC的DNN加速器中,每个数据包的有效载荷并不是很大.因此,将一个数据包划分成多个flit并不是一种合适的方法.所以在我们的方案中采用单flit机制来提高数据包的传输速度.

图7展示了3种不同组播方案的数据包大小.

当采取传统路由器的单播方案(Unicast)时,数据包的长度为32 bits.其中,路由信息(Routing Info)占4 bits,控制信息(Control Info)占12 bits,数据(Data)占16 bits.路由信息主要是数据包目的节点的地址,控制

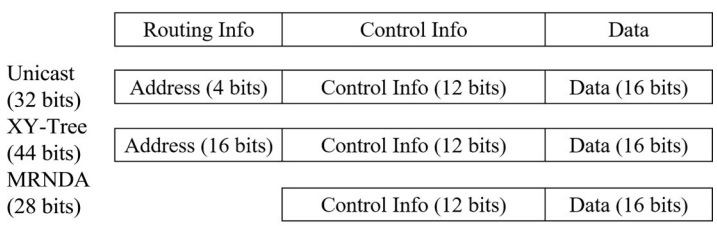


图7　不同方案数据包长度

信息可以包含一些数据的校验信息、数据包的排序信息以及一些用户自定义的字段,而数据部分则是代表向每个节点输入的数据. 在本文的Unicast和XY-Tree方案中,控制信息字段主要用于携带数据包的排序信息,因为不同数据包从源节点到目的节点的顺序可能会发生变化,因此目的节点需要通过数据包中的排序信息对数据进行整序. 设计者可以根据运行的神经网络模型以及实际的需要自由增加或减少控制信息的字段大小. 在本文中,若运行的神经网络模型为Lenet-5,那么控制信息字段为12 bits就已经足够,若运行的是VGG-16,控制信息字段则至少需要22 bits. 与Unicast方案以及XY-Tree方案不同的是,MRNDA的树形网络的控制字段不需要携带数据包的排序信息,因为树形网络的数据包是按顺序到达各个PE中的. 在XY-Tree方案中,地址部分占16 bits,这是因为在XY-Tree方案中,数据包需要携带和网络节点相同数量的比特位来完成组播信息. 而在MRNDA中,则不需要路由信息,这是因为MRNDA的路由决策是依赖下级路由器的举手信息来完成的.

4 实验评估

4.1 仿真环境配置

本文在Noxim[32]模拟器上进行扩展所评估的MRNDA方案的各项性能. 本文在4×4的Mesh网络中,将MRNDA和传统单播方案以及XY-Tree组播方案进行对比. 具体的仿真环境配置如表1所示.

表1 实验基本参数设置表

参数	设置
网络拓扑	4×4 Mesh
缓冲区大小/端口	4 flits
虚通道数量/端口	4
flit大小	32 bits ~ 42 bits (Unicast) 44 bits ~54 bits (XY-Tree) 28 bits (MRNDA)
路由算法	XY (Unicast)
PE性能	86.4 GOPS
路由器频率	1 GHz
内存带宽	2 GB/s
DNN 模型	Lenet-5, AlexNet, VGG-16

在本文的实验方案中,传统单播方案和XY-Tree组播方案的路由器每个端口均有4个虚通道,每个虚通道为4 flit buffer. MRNDA buffer大小也为4 flits. 本文中Unicast和XY-Tree方案中flit的大小根据所运行的神经网络模型的不同会有一定的变化. 考虑到现有的DNN加速器的PE设计,在本文中,每个PE的性能为86.4 GOPS,与UNPU(Unified Neural Processing Unit)[42]的设计相同. 所有方案下的路由器均为单周期路由器,频率为1 GHz. MI节点从内存中存取数据的带宽为2 GB/s. 在本文的实验方案中,我们选取了3种不同规格的DNN模型用于实验中,分别是Lenet-5,AlexNet和VGG-16.

4.2 性能分析

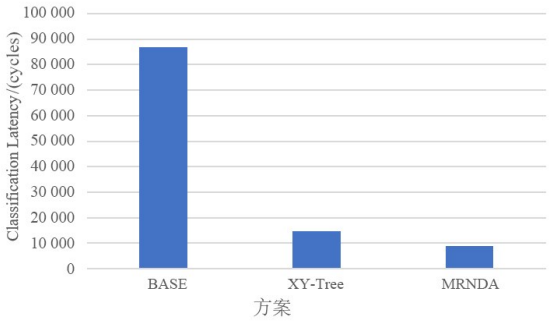
为了衡量不同的组播机制对DNN加速器性能的影响,本文主要使用两个指标来衡量不同组播机制的性能:分类延迟和通信延迟. 分类延迟指的是一张图片通过NoC输入到加速器中到分类结果通过MI结构输出到内存中所用的周期数. 而通信延迟指的是在加速器工作过程中用于数据传输的周期数.

图8为3种不同规模的卷积神经网络Lenet-5, AlexNet, VGG-16运行时所使用的分类延迟. 从图7中我们可以看出,使用组播方案(XY-Tree和MRNDA)比使用单播方案(Basement Schem,BASE)分类延迟有显著的降低. 在图8中XY-Tree方案比BASE方案在3种不同神经网络中分类延迟降低了83.1%, 82.1%和75.6%,MRNDA方案比BASE方案的分类延迟降低了86.7%,85.4%和81.6%. 而MRNDA相比XY-Tree在3种不同神经网络中的分类延迟降低了21.3%, 18.7%, 24.5%. 这说明在不同规格的神经网络下,MRNDA方案相比XY-Tree方案都有较为不错的提升.

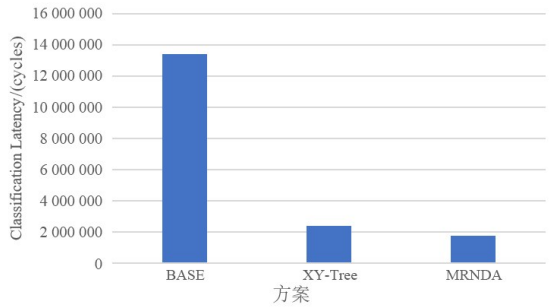
图9为归一化后3种不同规模的卷积神经网络Lenet-5, AlexNet, VGG-16运行时所使用的通信延迟. 从图9中我们可以看出,使用组播方案(XY-Tree和MRNDA)比使用单播方案(BASE)能够有效降低NoC网络中数据传输所用的时间. 在图9中XY-Tree方案比BASE方案在3种不同神经网络中通信延迟降低了83.9%,85.0%和82.3%,MRNDA方案比BASE方案的通信延迟降低了87.6%, 88.4%和88.8%. 而MRNDA比XY-Tree在3种不同神经网络中的通信延迟降低了22.7%,23.1%,36.7%. XY-Tree方案显著减少了网络中冗余数据包的传输,因此XY-Tree方案的通信延迟远低于BASE方案. MRNDA则是使用了额外一层的树形网络传输数据包,树形网络的规模更小,平均跳距比Mesh网络更小,并且可以和Mesh网络同时工作. 因此相比XY-Tree方案,MRNDA有效地减少了NoC的通信延迟.

4.3 准确性分析

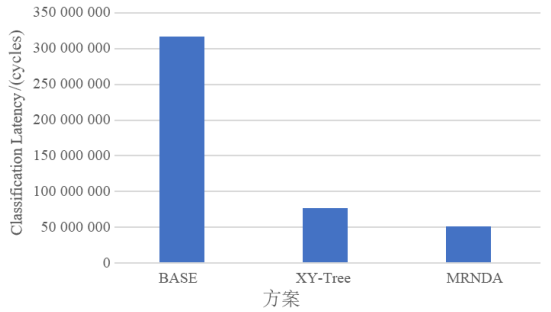
图10展示了数据包污染率增加对3种网络模型Lenet-5, AlexNet和vgg-16的平均Top-1识别准确度的影响. 图中横坐标是数据包污染率的比例,纵坐标代表识别结果为正确类别任务占总任务数的比例. 我们对3种模型进行了全网络错误数据包随机注入,模拟传输过程中瞬时性故障的数据包污染. 相对于Lenet-5模



(*a*) Lenet-5平均分类延迟

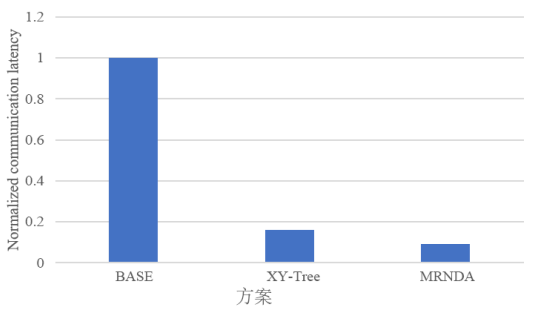


(*b*) Alexnet平均分类延迟

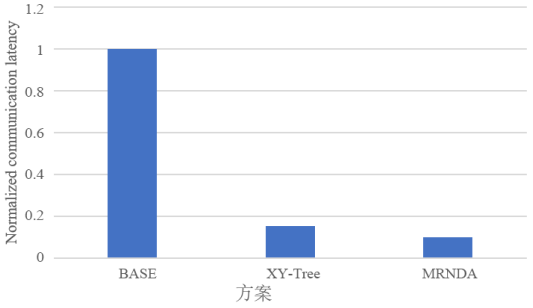


(*c*) VGG-16平均分类延迟

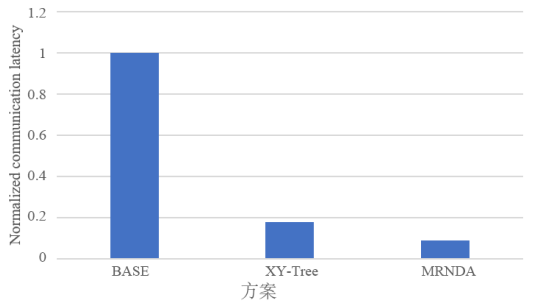
图8　各网络模型平均分类延迟



(*a*) Lenet-5归一化通信延迟



(*b*) Alexnet归一化通信延迟



(*c*) VGG-16归一化通信延迟

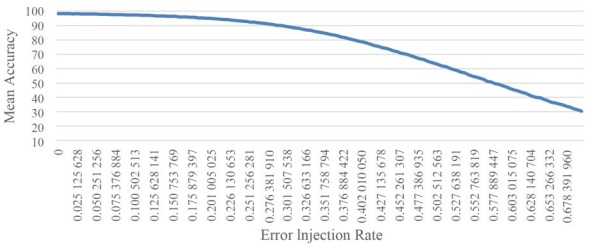
图9　各网络模型归一化通信延迟

型,AlexNet和vgg-16的准确度随着污染数据包注入率的提升,下降的趋势更为明显.因为这2个网络深度较大,参数量也较大,相对于Lenent-5进行等比例数据包污染时,污染数据包的绝对数量更大,更容易在关键数据位产生差错,也更有可能直接在推断结果传输中产生差错,造成总体的识别准确度下降.神经网络中数据的稀疏性和大量冗余神经元为其提供了天然的容错能力.从图中可以看出,3%以内的数据包污染对网络最终识别精度的影响极小.同时考虑到在纯有线的2-D片上网络中,数据传输的比特错误率通常在10^{-14}量级[43],除非大量数据包污染发生在结果输出传输过程中,其影响可以忽略不计.

4.4　功耗和面积分析

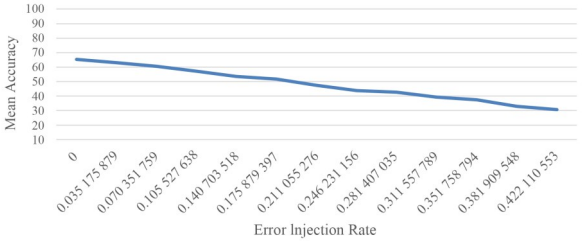
本文使用Verilog HDL描述文中3种方案路由器的硬件逻辑结构,使用ModelSim-SE 2019.2进行功能和结构验证,使用Synopsis Design Compiler对硬件结构进行综合.我们使用台积电(TSMC)的65 nm CMOS标准单元库模拟了路由器的面积和功耗,路由器工作频率为1 GHz.仿真结果如表2所示.

表2中的BASE方案和XY-Tree方案的面积和功耗是在flit大小32 bits和44 bits情况下取得的.从表2中可以看出,MRNDA方案是一个很轻量化的路由器,面积和功耗开销均很小.和XY-Tree方案相比,MRNDA方案采用树形网络组播数据包,在传输同量组播数据



(*a*) Lenet-5分类准确性

(*b*) Alexnet分类准确性



(*c*) VGG-16分类准确性

图10 网络模型分类准确性

表2 面积功耗表

方案	面积/μm^2	总功耗/mW
BASE	50 659.56	40.41
XY-Tree	67 043.88	52.28
MRNDA	4 812.65	4.16

包的情况下只需要启动5个树形路由器,而XY-Tree则需要启动所有Mesh路由器,并且我们设计的路由器面积和功耗更小,因此我们的方案比XY-Tree方案能够节约更多的功耗.

表3为MRNDA路由器单个端口握手逻辑部分和PE举手单元的面积和功耗开销.

从表3中可以看出,MRNDA路由器中发送端口和接收端口握手逻辑的功耗和面积只占路由器总功耗和面积开销的很小一部分,因此可以认为MRNDA的缓冲区和交叉开关的面积和功耗才是路由器总功耗的主要来源. 而PE举手单元的面积和功耗也非常小,因此可以认为MRNDA的路由决策机制在PE中的面积和功耗开销是几乎可以忽略不计的.

表3 单个端口握手逻辑和PE举手单元面积功耗表

方案	面积/μm^2	总功耗/mW
发送端口	40.68	0.036
接收端口	58.32	0.042
PE举手单元	19.44	0.010

5 总结

近年来,有研究提出了基于NoC的DNN加速器. 然而,当中存在的一对多流量意味着为NoC设计合适的组播机制能够有效地提升加速器的分类速度. 并且,目前的深度神经网络规模非常庞大,但是NoC的核心数量往往是有限的. 因此,在本文中,我们提出了一种针对资源受限的NoC的组播方案MRNDA. MRNDA使用在传统Mesh网络上附加一层树形组播网络的方式进行组播,能够极大减少传输数据包的数量和分类延迟. MRNDA的路由器结构也相对更为简单,对芯片的面积和功耗影响很小. 因此MRNDA能够使基于资源受限NoC的DNN加速器突破通讯瓶颈,提升加速器分类速度,并且硬件开销在可接受的范围内.

参考文献

[1] GOOSSENS K, DIELISSEN J, RADULESCU A. AEthereal network on chip: Concepts, architectures, and implementations[J]. IEEE Design & Test of Computers, 2005, 22(5): 414-421.

[2] WANG L, JIN Y, KIM H, et al. Recursive partitioning multicast: A bandwidth-efficient routing for networks-on-chip[C]//2009 3rd ACM/IEEE International Symposium on Networks-on-Chip. New York: ACM, 2009: 64-73.

[3] PEH L S, DALLY W J. A delay model and speculative architecture for pipelined routers[C]//Proceedings HPCA Seventh International Symposium on High-Performance Computer Architecture. New York: ACM, 2001: 255-266.

[4] KUMAR A, PEH L S, KUNDU P, et al. Express virtual channels: Towards the ideal interconnection fabric[J]. ACM Sigarch Computer Architecture News, 2007, 35(2): 150-161.

[5] MATSUTANI H, KOIBUCHI M, AMANO H, et al. Prediction router: Yet another low latency on-chip router architecture[C]//2009 IEEE 15th International Symposium on High Performance Computer Architecture. Piscataway: IEEE, 2009: 367-378.

[6] DEB S, GANGULY A, PANDE P P, et al. Wireless NoC

as interconnection backbone for multicore chips: Promises and challenges[J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2012, 2(2): 228-239.

[7] OUYANG Y, YANG J, XING K, et al. An improved communication scheme for non-HOL-blocking wireless NoC [J]. Integration, 2018, 60: 240-247.

[8] OUYANG Y, LI Z, LI J, et al. CPCA: An efficient wireless routing algorithm in WiNoC for cross path congestion awareness[J]. Integration, 2019, 69: 75-84.

[9] OUYANG Y, WANG Q, HU L, et al. DVFS based error avoidance strategy in wireless network-on-chip[J]. Journal of Electronic Testing, 2019, 35(6): 767-777.

[10] CHEN K C, EBRAHIMI M, WANG T Y, et al. NoC-based DNN accelerator: A future design paradigm[C]// Proceedings of the 13th IEEE/ACM International Symposium on Networks-on-Chip. New York: ACM, 2019: 1-8.

[11] DALLY W J, TOWLES B P. Principles and Practices of Interconnection Networks[M]. Amsterdam: Morgan Kaufmann Publishers, 2004.

[12] PAINKRAS E, PLANA L A, GARSIDE J, et al. SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation[J]. IEEE Journal of Solid-State Circuits, 2013, 48(8): 1943-1953.

[13] CARRILLO S, HARKIN J, MCDAID L J, et al. Scalable hierarchical network-on-chip architecture for spiking neural network hardware implementations[J]. IEEE Transactions on Parallel and Distributed Systems, 2012, 24(12): 2451-2461.

[14] CHEN Y H, YANG T J, EMER J, et al. Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices[J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2019, 9(2): 292-308.

[15] LIU X, WEN W, QIAN X, et al. Neu-NoC: A high-efficient interconnection network for accelerated neuromorphic systems[C]//2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC). Piscataway: IEEE, 2018: 141-146.

[16] CHEN K C, WANG T Y. NN-noxim: High-level cycle-accurate NoC-based neural networks simulator[C]//2018 11th International Workshop on Network on Chip Architectures (NoCArc). Piscataway:IEEE, 2018: 1-5.

[17] CHEN K C J, WANG T Y G, YANG Y C A. Cycle-accurate noc-based convolutional neural network simulator [C]//Proceedings of the International Conference on Omni-Layer Intelligent Systems. New York: ACM, 2019: 199-204.

[18] CHEN K C J, EBRAHIMI M, WANG T Y, et al. A NoC-based simulator for design and evaluation of deep neural networks[J]. Microprocessors and Microsystems, 2020, 77: 103145.

[19] XIAO S, GUO Y, LIAO W, et al. Neuronlink: An efficient chip-to-chip interconnect for large-scale neural network accelerators[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2020, 28(9): 1966-1978.

[20] SHEN X W, YE X C, TAN X, et al. An efficient network-on-chip router for dataflow architecture[J]. Journal of Computer Science and Technology, 2017, 32(1): 11-25.

[21] HAN S, LEE J, CHOI K. Tree-mesh heterogeneous topology for low-latency noc[C]//Proceedings of the 2014 International Workshop on Network on Chip Architectures. New York: ACM, 2014: 19-24.

[22] DAS R, NARAYANASAMY S, SATPATHY S K, et al. Catnap: Energy proportional multiple network-on-chip[J]. ACM SIGARCH Computer Architecture News, 2013, 41 (3): 320-331.

[23] LIU S, CHEN T, LING L, et al. IMR: High-performance low-cost multi-ring NoCs[J]. IEEE Transactions on Parallel & Distributed Systems, 2016, 27(6): 1700-1712.

[24] SPEIER T, WOLFORD B, DILEEP B. Qualcomm centriq 2400 processor[C]//Hot Chips: A Symposium on High Performance Chips. New York: ACM, 2017:1-17.

[25] JEFFERS J, REINDERS J, SODANI A. Intel Xeon Phi Processor High Performance Programming: Knights Landing Edition[M]. Amsterdam: Morgan Kaufmann Publishers Inc, 2016.

[26] LIAN X, LIU Z, SONG Z, et al. High-performance FPGA-based CNN accelerator with block-floating-point arithmetic[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2019, 27(8): 1874-1885.

[27] FARABET C, POULET C, LECUN Y. An fpga-based stream processor for embedded real-time vision with convolutional networks[C]//2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops. Piscataway: IEEE, 2009: 878-885.

[28] GUPTA S, AGRAWAL A, GOPALAKRISHNAN K, et al. Deep learning with limited numerical precision[C]//International Conference on Machine Learning. New York: ACM, 2015: 1737-1746.

[29] MOONS B, VERHELST M. A 0.3-2.6 TOPS/W precision-scalable processor for real-time large-scale ConvNets[C]//2016 IEEE Symposium on VLSI Circuits (VL-

SI-Circuits). Piscataway:IEEE, 2016: 1-2.
[30] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[J]. Advances in Neural Information Processing Systems, 2012, 25: 1097-1105.
[31] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[EB/OL]. (2014)[2022]. https:arxiv.org/abs/1709.06158v1.
[32] CATANIA V, MINEO A, MONTELEONE S, et al. Cycle-accurate network on chip simulation with noxim[J]. ACM Transactions on Modeling and Computer Simulation (TOMACS), 2016, 27(1): 1-25.
[33] KUMAR D R, NAJJAR W A, SRIMANI P K. A new adaptive hardware tree-based multicast routing in k-ary n-cubes[J]. IEEE Transactions on Computers, 2001, 50(7): 647-659.
[34] HU W, LU Z, JANTSCH A, et al. Power-efficient tree-based multicast support for networks-on-chip[C]//16th Asia and South Pacific Design Automation Conference (ASP-DAC 2011). New York: ACM, 2011: 363-368.
[35] LIN X, MCKINLEY P K, NI L M. Deadlock-free multicast wormhole routing in 2-D mesh multicomputers[J]. IEEE Transactions on Parallel and Distributed Systems, 1994, 5(8): 793-804.
[36] EBRAHIMI M, DANESHTALAB M, LILJEBERG P, et al. HAMUM-A novel routing protocol for unicast and multicast traffic in MPSoCs[C]//2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing. Piscataway: IEEE, 2010: 525-532.
[37] JERGER N E, L-S PEH, LIPASTI M. Virtual circuit tree multicasting: A case for on-chip hardware multicast support [C]//2008 International Symposium on Computer Architecture. Piscataway: IEEE, 2008: 229-240.
[38] OUYANG Y, TANG F, HU C, et al. MMNNN: A tree-based multicast mechanism for NoC-based deep neural network accelerators[J]. Microprocessors and Microsystems, 2021, 85(5): 104242.
[39] KONSTANTINOU D, NICOPOULOS C, LEE J, et al. SmartFork: Partitioned multicast allocation and switching in network-on-chip routers[C]//2020 IEEE International Symposium on Circuits and Systems (ISCAS). Piscataway: IEEE, 2020.
[40] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.
[41] ASCIA G, CATANIA V, MONTELEONE S, et al. Analyzing networks-on-chip based deep neural networks[C]// Proceedings of the 13th IEEE/ACM International Symposium on Networks-on-Chip. New York: ACM,2019: 1-2.
[42] LEE J, KIM C, KANG S, et al. UNPU: An energy-efficient deep neural network accelerator with fully variable weight bit precision[J]. IEEE Journal of Solid-State Circuits, 2018, 54(1): 173-185.
[43] AGYEMAN M O, VIEN Q T, AHMADINIA A, et al. A resilient 2-d waveguide communication fabric for hybrid wired-wireless noc design[J]. IEEE Transactions on Parallel and Distributed Systems, 28(2): 359-373.

作者简介

欧阳一鸣 男,1963年出生,安徽合肥人. 合肥工业大学计算机与信息学院教授、博导,中国计算机学会高级会员. 主要研究方向为片上网络(NoC)与片上系统(SoC)、嵌入式系统的综合与测试、数字系统设计自动化.
E-mail: oyym@hfut.edu.cn

王 奇 男,1994年出生,安徽滁州人. 2016年于吉林大学获学士学位,2018年于美国斯蒂文斯理工学院获硕士学院. 合肥工业大学微电子学院博士生. 主要研究方向为片上网络容错设计、智能片上网络、神经网络加速器设计.
E-mail: keywenchester@outlook.com