doi:10.3772/j.issn.1002-0470.2021.05.003

基于 NVDLA 与 FPGA 结合的神经网络加速器平台设计①

管兆康②* 张志伟③**

(*华中科技大学人工智能与自动化学院 武汉 430074)

(**中国科学院自动化研究所 北京 100190)

摘 要 随着深度神经网络对算力的需求不断增加,传统通用处理器在完成推理运算过程中出现了性能低、功耗高的缺点,因此通过专用硬件对深度神经网络进行加速逐步成为了深度神经网络的重要发展趋势。现场可编程门阵列(FPGA)具有重构性强、开发周期短以及性能优越等优点,适合用作深度神经网络的硬件加速平台。英伟达深度学习加速器(NVDLA)是英伟达开源的神经网络硬件加速器,其凭借自身出色的性能被学术界和工业界高度认可。本文主要研究 NVDLA 在 FPGA 平台上的优化映射问题,通过多种优化方案高效利用 FPGA 内部的硬件资源,同时提高其运行性能。基于搭建的 NVDLA 加速器平台,本文实现了对 RESNET-50 神经网络的硬件加速,完成了在 ImageNet 数据集上的图像分类任务。研究结果表明,优化后的 NVDLA 能显著提高硬件资源使用效率,处理性能最高可达 30.8 fps,实现了较边缘中央处理器(CPU)加速器平台 28 倍的性能提升。

关键词 英伟达深度学习加速器(NVDLA);现场可编程门阵列(FPGA);硬件加速;模块优化

0 引 言

随着人工智能的飞速发展,神经网络作为一种重要的深度学习框架愈发引起人们的重视。相比于传统算法,神经网络凭借其在图像识别等领域的优越表现,已经在无人驾驶[1]、人脸识别[2]以及目标跟踪[3]等众多领域得到广泛应用。然而,神经网络在实际的应用过程中依然存在计算量大、存储复杂等问题,因此为神经网络的应用选择合适的计算平台对于推动神经网络的发展至关重要。神经网络推理的大部分计算工作都基于数学运算,其主要包含卷积运算、激活函数运算、池化运算和规范化运算 4 部分。以上操作的内存访问模式是有规律、可预测且易于并行进行的,适合通过特殊用途的硬件来实现。近年来,各种神经网络硬件加速器平台应运而生。在 ASIC 设计方面,谷歌提出了张量处理器(tensor processing unit, TPU)[4],其通过脉动矩阵的方式实现神经网络加速,主要针对 TensorFlow 框架进行设计。寒武纪推出了 DianNao 系列芯片[5],其中包括基于单核的 DianNao[6] 和基于多核架构的 DaDianNao[7],以及针对多种机器学习算法进行优化的 PuDianNao[8] 和针对深度卷积神经网络数据访存问题进行优化的 ShiDianNao[9] 等。Farabet 等人[10]提出了一种可扩展的数据流硬件体系结构神经网络硬件加速器,专门针对通用视觉算法进行了硬件加速。此外 Chen 等人[11]也针对卷积神经网络硬件加速器的功耗问题进行优化,提出了一种低功耗的加速器 Eyeriss。在现场可编程门阵列(field-programmable gate array, FPGA)设计方面,Zhang 等人[12]采用 roofline model 对神经网络推理的计算吞

① 中国科学院战略性先导科技专项(XDB32000000)资助项目。

② 男,1995 年生,硕士生;研究方向:基于 FPGA 的神经网络硬件加速;E-mail: 981625562@qq.com

③ 通信作者,E-mail: zhiwei.zhang@ia.ac.cn

(收稿日期:2020-04-01)

吐量和所需的内存带宽进行定量分析，选择最佳设计方案，在 FPGA 上实现了 61.62 GFLOPS 的处理性能。余子健等人[13]基于粗粒并行层对卷积神经网络进行并行化加速，实现了对 MNIST 手写数字字符的识别。陈鹏等人[14]提出了一种基于 SDSoc 的 FPGA 设计方案，该设计方案采用流水线的层间复用方法，实现了在 FPGA 上更低功耗的深度神经网络硬件加速。Ding 等人[15]同时从深度神经网络算法和硬件设计两方面进行考虑，提出一种异构权重量化方法，即交替方向乘数法（alternating direction method of multipliers，ADMM），并使用块循环矩阵方法对卷积模块进行设计，提升了 FPGA 上加速器的处理性能。

神经网络硬件加速器平台中，基于 FPGA 的加速器平台由于高能效、高并行性、灵活性以及安全性等优势被广泛应用于深度学习加速器。英伟达深度学习加速器（NVIDIA deep learning accelerator，NVDLA）是英伟达开源的神经网络硬件加速器项目。该项目以 IP 核的形式提供 RTL 级的综合模型和仿真模型，并鼓励开发者们在此基础上进行进一步地开发和设计。但是 NVDLA 本身是从 ASIC 的角度出发进行设计的，并没有针对 FPGA 平台进行性能、设计尺寸和功耗等方面的优化。

本文首先针对 NVDLA 在 FPGA 上综合实现过程中存在的问题进行分析，提出相应的优化方案，使其能充分利用 FPGA 内部的各种硬件资源，实现较高性能的神经网络推理。其次，基于优化的 NVDLA 模块搭建 FPGA 的神经网络硬件加速器平台并设计相应的神经网络编译器和运行时程序。最后，利用设计的加速器平台实现 RESTNET-50 深度卷积神经网络的硬件加速并通过实验验证加速器的性能。

1　NVDLA 的介绍

NVDLA 提供了一种简单、灵活、强大的推理加速解决方案，其采用模块化结构，具有良好的可扩展性、高度可配置性以及可移植性，能够根据具体的应用场景进行设计和集成，支持多种规模的物联网设备。

1.1　硬件结构

NVDLA 的内部硬件架构如图 1 所示。其主要通过 4 个接口与外部进行数据交互，4 个接口分别为控制信号接口、中断接口以及两个数据传输接口。控制信号接口是一个同步的、低带宽的、低功耗的 32 位控制总线接口，用于中央处理器（central processing unit，CPU）访问 NVDLA 配置寄存器，控制 NVDLA 实现特定的功能。中断接口是一个 1 bit 的输出接口，当 NVDLA 完成任务或者出现错误时，通过此接口向 CPU 发送中断信息。数据传输接口是一种同步、高速、高度可配置的数据总线接口。其中，主数据传输接口与外部存储器相连。二级数据传输接口是可选的，与高数据吞吐量、低数据访问延迟的边上存储器相连，以实现更快的数据交互。

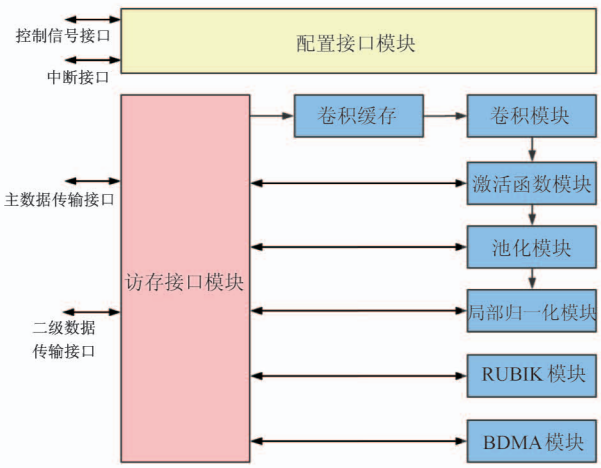

图 1　NVDLA 的内部硬件架构图

NVDLA 内部采用模块化的设计方式，并且每个模块之间是相互独立的。例如，不需要池化功能，亦可直接将池化模块裁剪。或者，想要优化卷积模块的性能，只需对卷积模块进行调整，无需修改加速器的其他模块。这样的设计方式在保持设计的灵活性的同时简化了集成，但需要专用微处理器或者 CPU 来完成对每个模块的调度操作。

1.2　编译器

编译器是负责将通用深度学习框架描述的网络模型，直接映射至专有硬件的软件。目前 NVDLA 的编译器支持 caffe 网络框架，通过定点化来减少模型参数，并提高计算效率。编译的流程包括解析网

络结构、经典计算图构建、NVDLA 计算图构建、图优化和操作输出。解析网络结构从模型定义读入网络并配置属性,同时检查网络的结构合法性、配置文件的匹配性。经典计算图构建直接从模型定义构建计算图,以节点的形式描述每个层的操作。NVDLA 计算图的构建根据 NVDLA 硬件结构的特性,进一步调整经典计算图的结构。图优化是进一步优化计算图的结构,对计算图内的节点进行拆分合并,并为每一个节点分配内存。操作输出负责将优化后的节点操作映射至具体的硬件寄存器配置。编译完成后,软件生成自定义格式的可执行文件供运行时程序使用,文件主要组成包括权重和特征分配的内存物理地址、各层的配置参数以及网络的输入输出格式。

2 NVDLA 的优化

本节对 NVDLA 在 FPGA 平台综合实现过程中出现的问题进行分析,同时提出解决相应问题的优化策略和方案。

2.1 门控时钟的优化

在 ASIC 设计中通常会采用门控时钟的策略来降低功耗。当系统中的某个模块处于不工作状态时,系统通过关闭时钟的方式使其处于非激活的状态,从而达到降低功耗的目的,NVDLA 也引入了门控时钟来降低功耗。在 FPGA 中,常规的时钟信号从管脚进入芯片后,首先会被送入全局时钟缓冲器(global clock buffer, BUFG),再经专用的全局时钟网络扇到各级寄存器中,故采用全局时钟网络能有效地减少寄存器之间的时钟偏斜。但由于 FPGA 无法主动对门控时钟插入 BUFG,导致门控时钟模块的输出时钟只能占用传统布线资源,无法使用全局时钟网络进行布线。时钟信号采用传统布线资源一方面会抢占其他关键路径的布线资源,另一方面会导致到各寄存器之间的时延差距增大,进而导致严重的时序违例。表 1 列出了门控时钟优化前后时序分析工具对模块保持时间的分析结果。由表 1 可知,采取门控时钟策略的电路中 37.5% 的时钟路径都无法实现时序收敛,此类布线生成的电路将无法正常工作。而不采取门控时钟策略的电路则没有出现保持时间的违例。

表 1 门控时钟优化前后时序分析工具对模块保持时间的分析结果

	门控时钟优化前/ns	门控时钟优化后/ns
最差保持时间松弛	-5.393	0.010
总共违例保持时间松弛	-361 338.374	0.000
时序违例的时钟路径个数	371 338	0
总共的时钟路径个数	990 093	990 093

根据保持时间松弛的计算公式对优化前保持时间违例最严重的时钟路径进行分析。

$$CPS = SCD + CPR - DCD \tag{5}$$

$$HS = CPS + DPD \tag{6}$$

其中 *CPS* 为时钟路径偏斜,*SCD* 为源时钟延时,*CPR* 为时钟悲观度余量,*DCD* 为目的时钟延时,*HS* 为保持时间松弛,*DPD* 为数据路径延时。表 2 列出了该时钟路径在门控时钟优化前后,时序分析工具给出的各项参数。由表 2 可得,优化前后 *DPD* 和 *CPR* 只有 2 ps 以内的差距,故 *DCD* 过大是保持时间违例的主要原因。因此剔除 NVDLA 的门控时钟模块,使时钟信号进入全局时钟网络,方能减小各寄存器之间的时钟偏斜,确保 NVDLA 在 FPGA 上的正常工作。

表 2 门控时钟优化前后关键时钟路径的各项时序参数

	门控时钟优化前/ns	门控时钟优化后/ns
SCD	8.533	6.841
CPR	-0.145	-0.143
DCD	13.946	6.848
DPD	0.165	0.166

2.2 模块的裁剪

模块的裁剪能够减少 NVDLA 对 FPGA 硬件资源的占用,有利于其在 FPGA 上的实现。

2.2.1 片上缓存的裁剪

NVDLA 本身是基于 ASIC 进行设计,其工作频率能够达到 1 GHz 以上,端口拥有很高的数据传输带宽。而外部存储器受限于功耗和性能,数据传输的带宽通常无法满足 NVDLA 的数据传输带宽。故 NVDLA 引入了高带宽的片上存储器以缓解数据传输带宽不足的问题。有别于 ASIC 芯片,FPGA 被设计成了岛状逻辑块矩阵电路以满足可重构的特性,每个逻辑块内部有实现任意电路的硬件资源。模块化的电路结构导致 FPGA 在实现同样的电路功能时,需要占用更多的逻辑资源、采用更复杂的布线结构。以上两点导致 FPGA 内部的网络延时较大,亦限制了 FPGA 的工作频率。有相关研究表明,在 FPGA 上实现一个只有组合和时序逻辑的电路,平均需要相当于 ASIC 电路 40 倍的面积和 3 ~ 4 倍的关键路径延时[16]。在 FPGA 中,NVDLA 由于受限于工作频率,其端口的数据传输带宽一般情况下小于外部存储器的带宽。以本设计为例,NVDLA 的数据位宽为 256 bit,最高工作频率为 188 MHz,可得 NVDLA 数据端口的最大理论带宽为 256 × 188/8 = 6.016 GB/s。其外部存储器为双倍速率同步动态随机存储器(double data rate synchronous dynamic random access memory, DDR SDRAM),数据位宽为 64 bit,工作频率为 1200 MHz,理论带宽为 2 × 1200 × 64/8 = 19.2 GB/s,故外部存储器已经满足 NVDLA 的带宽需求,无需添加额外片上缓存。此外,NVDLA 需要引入 32 MB 以上的片上存储器才能保证其正常工作。然而 FPGA 内部的片上存储资源有限,且均布局在 FPGA 内部的固定位置,若使用过多的片上存储资源则会增大时序收敛的难度。因此在 FPGA 设计中可以裁剪用于实现片上存储器和片外存储器之间数据传输的 BDMA 模块和片上存储器模块。

2.2.2 CDP 模块的裁剪

CDP 模块主要实现了局部响应标准化层(local response normalization,LRN)的功能,LRN 层多出现于早期的神经网络结构中(例如 ALEXNET[17] 和 GOOGLENET[18]),其主要是实现数据局部归一化的功能,但日后的研究证明 LRN 层对神经网络的训练结果的提升并不明显[19],故目前普遍应用的神经网络结构均不采用这一层(例如 RESNET[20] 和 VGG[19])。因此,在 FPGA 设计中该模块亦可进行裁剪。

2.3 乘加器阵列的优化

乘加器阵列模块是 NVDLA 的核心,主要实现了卷积过程中部分和计算的功能。NVDLA 在 FPGA 上主要利用查找表(look up table,LUT)资源实现乘加器阵列模块的功能。一方面,该方法占用了 FPGA 内部大量的查找表资源,加大了布局的难度,甚至会导致布局无法完成;另一方面,利用查找表实现的乘法器和加法器在布线过程中存在网络延时(net delay)较长的问题,增加了时序收敛的难度,限制了 NVDLA 工作频率,影响了整体运行性能。

DSP48 模块是 XILINX 公司 FPGA 内部的数字信号处理器(digital signal processor,DSP)资源,由加法器、乘法器和算数逻辑单元 3 部分组成,可以实现乘法、乘法累加、幅度比较器等复杂的算术操作。相比用 FPGA 内部的 LUT 资源实现同样的功能,DSP48 模块占用的面积更小,工作频率更快,性能更高。但 NVDLA 要调用 FPGA 内部的 DSP 资源实现乘加器阵列模块的功能,因此需要对整个模块进行优化。

2.3.1 乘加器阵列电路调整

优化前的 NVDLA 内部乘加器模块的电路结构如图2所示,其中4对输入特征与权重首先分别通

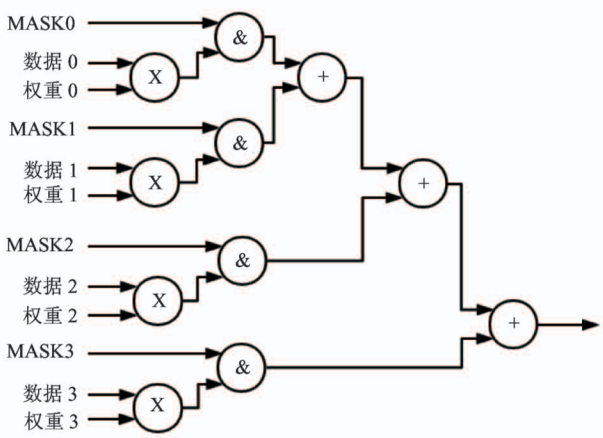

图 2 优化前 NVDLA 内部乘加器模块的电路结构

过乘法器获得运算结果,再和掩码(MASK)信号进行与运算,最后电路将与运算结果求和输出。此电路结构无法利用 DSP 高效实现,其原因在于单个 DSP 模块内部只有一个算数逻辑单元,无法同时完成对乘法结果的按位与和累加操作,故需要更多逻辑资源对 4 个与运算结果进行求和。为解决上述问题,本设计对乘加器阵列的电路进行调整,将权重和 MASK 信号进行与运算后作为输入送入 DSP 模块和数据进行乘法运算,并利用 DSP 模块的乘法累加运算,逐级累加获得最终输出结果,优化后电路如图 3 所示。中间插入多级流水的目的是减少多个 DSP 模块串联带来的逻辑延时(logic delay),有利于时序的收敛。通过这种方式,只需 4 个 DSP 模块就可实现乘累加的功能。

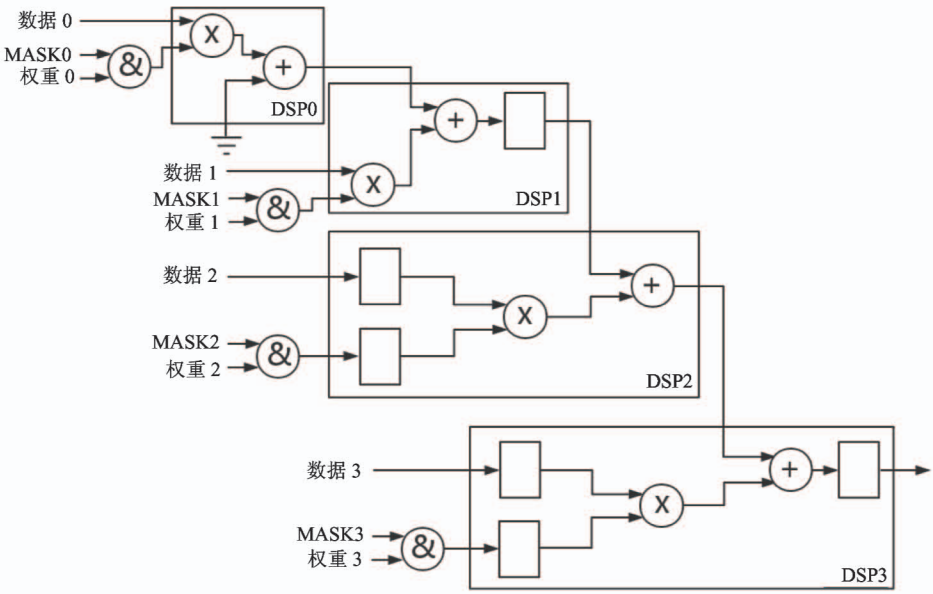

图 3　优化后 NVDLA 内部乘加器模块的电路结构

2.3.2　DSP 模块的复用

NVDLA 支持 8 位定点数的乘法运算,远少于 DSP 模块支持的乘法运算位数,为此可以考虑一个 DSP 模块并行执行两个 8 位乘法运算,从而提高复用,减少 FPGA 资源使用。为了实现在一个 DSP 模块并行完成 1 组输入特征与 2 组权重的乘法操作,针对 DSP 的设计需要满足以下两点要求。

(1) 高位的乘积结果不受低位乘积结果的影响。

(2) 低位乘累加结果对高位乘累加结果的影响是可恢复的。

NVDLA 内部支持的数据类型为 8 位定点数,1 对输入特征和权重的乘积结果的位宽为 8 + 8 = 16 位。为满足第 1 点要求,高位乘积结果的最低有效位不得进入 16 位,即高位的输入权重至少从 17 位开始。故 DSP 模块的被乘数位宽至少为 16 + 8 = 24 位,输出数据位宽至少为 16 + 16 = 32 位。本设计中 DSP 模块内部的乘法器被乘数位宽最大为 27 位,乘数位宽最大为 18 位,输出位宽最大为 48 位,具体实现 DSP 复用的乘累加功能的方式如图 4 所示。

外部电路将权重 1 左移 18 位送入输入端口 A,权重 2 和特征以二进制补码形式分别送入输入端口 B 和 D。以防低位累加结果溢出,进而导致低位累加结果对高位累加结果产生不可恢复的影响,故此处权重 1 被左移 18 位。权重 1 和权重 2 在预加法器求和后与特征一起送入乘法器完成运算,获得乘积结果。最后乘积结果和送入 C 端口的累加输入进行累加,获得输出,送往下一级 DSP 模块。

进一步分析低位乘累加结果对高位结果产生影响的原因。首先根据图 4 可以推出 DSP 模块的输出为

$$O = (W1 \ll 18 + W2) \times F + (I1 \ll 18 + I2) \quad (1)$$

$$= (W1 \times F + I1) \ll 18 + (W2 \times F + I2) \quad (2)$$

其中 O 为 DSP 的输出,$W1$ 和 $W2$ 为权重 1 和权重

2，F 为特征，$I1$ 和 $I2$ 为高位和低位的累加输入。可得，DSP 的输出为二进制补码格式的两个 27 + 18 = 45 位项之和（左移 18 位的高位累加输出和未进行移位的低位累加输出）。进一步可得到高位和低位乘累加的输出结果和 DSP 输出 O 的关系。

$$O[35:18] = (W1 \times F + I1) + S \tag{3}$$

$$O[17:0] = (W2 \times F + I2) \tag{4}$$

其中，S 为低位符号位对高位结果的影响，当低位结果为负值时 S 为 18'h3FFFF，低位结果为正值时 S 为 18'h0。因此在经历多级 DSP 乘累加之后，只需根据最后一级 DSP 的低位乘累加结果的符号对输出结果 $O[35:18]$ 进行校正即可得到正确的高位累加结果，即当低位乘累加结果为正值时，高位结果为 $O[35:18]$；当低位乘累加结果为负值时，高位结果为（$O[35:18]+1$）。

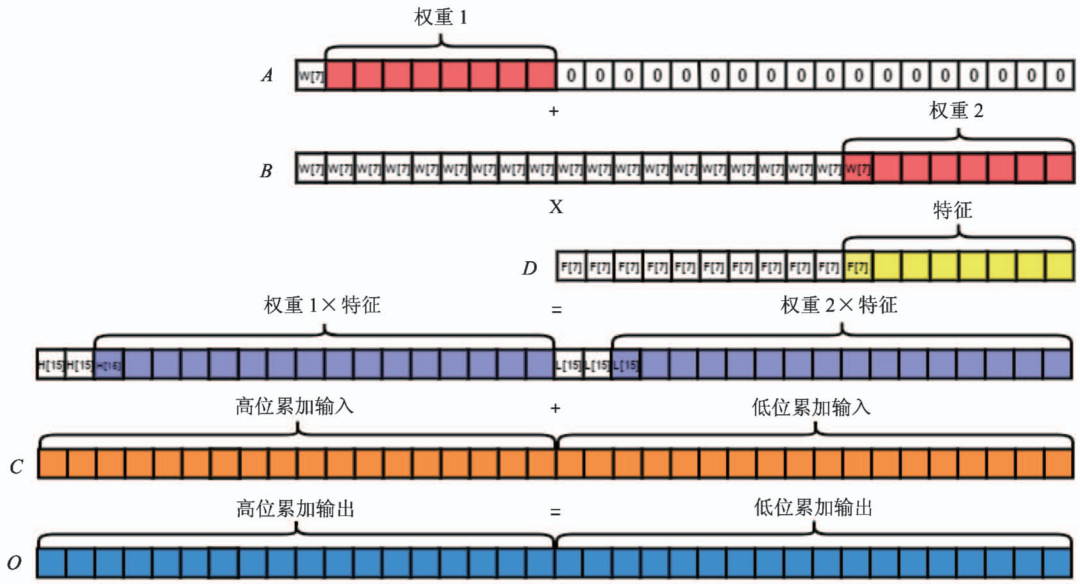

图 4　DSP 模块的复用原理图

3　神经网络加速器平台的设计

本节主要对硬件平台设计的具体细节以及软件平台中运行时程序的实现方式进行介绍。

3.1　硬件平台的架构

硬件平台架构框图如图 5 所示。系统主要由两条先进的可扩展接口（advanced extensible interface，AXI）总线、MicroBlaze 处理器、外部存储器、网口模块、NVDLA 模块以及其他外设模块组成。MicroBlaze 是 XILINX 公司自主研发的软核处理器，采用 RISC 架构和哈佛结构的 32 位指令并通过 AXI 总线与外部进行数据交互。本设计中，MicroBlaze 通过指令缓存（AXI _ IC）、数据缓存（AXI _ DC）接口经由 AXI _ HP 总线从外部存储器中读取运行的程序，并访问其中的数据。同时，MicroBlaze 将其产生的指令以配置寄存器的方式通过数据外设（AXI _ DP）接口经由 AXI _ LITE 总线发送到各个模块。网口模块主要起到与 PC 端进行数据交互的作用。当加速器平台工作时，网口模块将 PC 端发送来的权重数据和输入图像数据通过 SG _ DMA 模块传输到外部存储器中供 NVDLA 模块使用。SG _ DMA 模块是一种特殊功能的直接存储器访问模块（direct memory access，DMA），它能将数据连续地传输到非连续物理地址的存储空间中，这一特性适合 NVDLA 权重的存储。NVDLA 模块通过 AXI _ LITE 总线接收 MicroBlaze 配置加速器的参数和启动指令，并通过 AXI _ HP 总线从外部存储器读取权重和特征数据。运算完成后，NVDLA 将运算结果写入外部存储器中，同时通过中断控制器向 MicroBlaze 发送处理完成的中断信号。

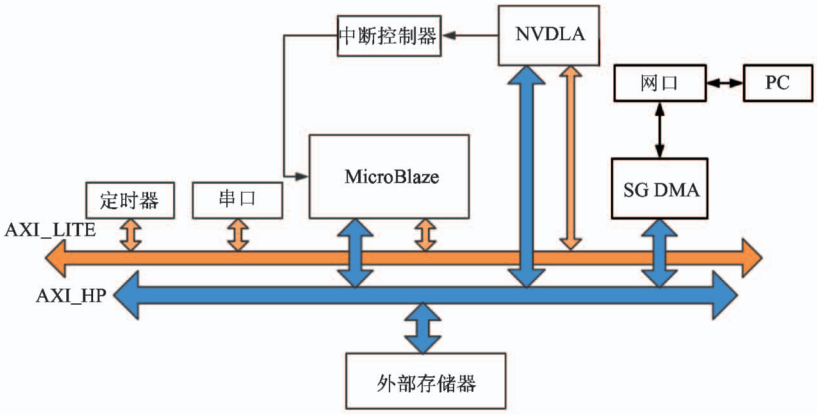

图 5　硬件平台架构框图

3.2　软件平台的实现

运行时(runtime)程序的主要功能为解析编译器生成的可执行文件,并调度 NVDLA 完成整个神经网络的推理工作。本设计的运行时程序运行在非操作系统下,主要流程如图 6 所示。一个网络到来后,MicroBlaze 首先要进行内存初始化的配置并将相应参数按照 NVDLA 的使用需要放置在内存中。完成配置后将输入图像送入网络中,通过计算图发射执行整个网络推理。计算图发射包括每个节点的配置,为隐藏节点配置对计算性能的影响,硬件上每个模块均采用 ping-pong 寄存器结构。采用 ping-pong 寄存器结构的硬件模块内部含有两组配置参数的寄存器,当硬件模块根据第1组寄存器的参数进行运算时,MicroBlaze 对第 2 组寄存器进行配置。在完成当前层运算之后硬件模块发送中断信号给 MicroBlaze,MicroBlaze 进行响应,清除中断,并使硬件模块根据第 2 组寄存器的配置进行下一层运算,同时对第 1 组寄存器进行配置。以此反复循环,硬件模块即可在运算过程中进行节点参数配置,隐藏节点配置的时间,从而提高计算性能。

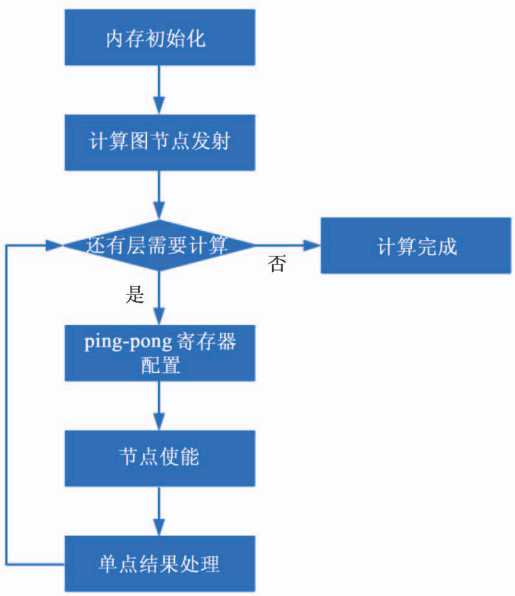

图 6　运行时程序流程

4　实验验证

4.1　实验设置

本设计选择在 XILINX 公司的 VCU118 FPGA 评估套件上进行开发,VCU118 内部搭载了 Virtex UltraScale + XCVU9P 的 FPGA 芯片、2 片 DDR4 存储器以及支持千兆网的 SGMII 以太网接口,满足硬件平台的设计需求。加速器的综合和实现均于 Vivado 2019.1 上完成,运行时程序的开发则于 Vivado SDK 2019.1 上进行。

4.2　实验结果与分析

本设计对 RESNET-50 的深度卷积神经网络进行硬件加速,完成了在 ImageNet 数据集上的图像分类任务。

表 3 列出了裁剪前后 FPGA 资源的使用情况。由表 3 可得,裁剪后双级随机存储器(bipolar random access memory,BRAM)资源的占用比裁剪前减少了 94.3%,其余资源的占用也有不同程度减少,因此模块裁剪更有利于 NVDLA 在 FPGA 上的实现。表 4 列出了 DSP 优化前后的 NVDLA 完成 RESNET-50

深度神经网络推理的性能和资源使用情况。在性能方面,优化前 NVDLA 由于受到自身内部乘加器阵列模块时序的限制,最高工作频率仅为 125 MHz,处理图像的性能为 25.3 fps。优化后的 NVDLA 利用 DSP 资源实现乘加器模块的功能,缓解了布局布线过程中的拥塞问题,使 NVDLA 的最高工作频率可达 188 MHz,处理图像的性能也提升到 30.8 fps。在资源的使用方面,优化后的 NVDLA 的查找表资源使用量明显少于优化前。采用 DSP 复用策略进行优化后,DSP 的使用量相较于只进行电路调整的使用量减少了 1024 个,实现了对 FPGA 内部硬件资源的合理利用。表 5 列出了工作频率为 125 MHz 时优化前后功耗的对比结果。由表 5 可知,由于 DSP 使用量较大,故只进行电路调整的 NVDLA 功耗较大。对乘加器模块的 DSP 进行复用后,DSP 的使用量减少了 50%,功耗从 3.558 W 降低到 2.719 W,性能功耗比也从 7.11 fps/W 提升到9.30 fps/W。综上,通过优化,处理性能从 25.3 fps 提升至 30.8 fps,性能功耗比从 9.22 fps/W 提升至 9.30 fps/W。

表 3 NVDLA 模块裁剪前后资源使用对比

	LUT	FF	DSP	BRAM
裁剪前	495 269	362 171	214	970
片上缓存	475 007	328 213	214	59.5
片上缓存 + CDP	413 619	284 088	214	55.5

表 4 NVDLA DSP 优化前后的性能和资源使用对比

	工作频率/MHz	处理性能/fps	LUT	FF	DSP
优化前	125	25.3	413 619	284 088	214
电路调整	188	30.8	262 866	275 432	2262
电路调整 + DSP 复用	188	30.8	263 560	276 225	1238

表 5 DSP 优化前后功耗对比

	总功耗/W	每瓦性能/fps/W
优化前	2.744	9.22
电路调整	3.558	7.11
电路调整 + DSP 复用	2.719	9.30

4.3 对比与比较

表 6 列出了本设计的加速器平台与边缘 CPU 加速器平台的性能比较。本设计选用单核 ARM CortexA73 处理器和四核 ARM CortexA73 处理器的边缘 CPU 加速器平台进行研究对比。由表 6 可得,与以上两类加速器平台相比,本设计的硬件加速器平台分别实现了 28 倍和 16.2 倍的加速。

表 6 与 CPU 的性能比较

	处理时间/ms	处理性能/fps
单核 ARM CortexA73	935	1.1
四核 ARM CortexA73	522	1.9
本设计硬件加速器	32	30.8

5 结 论

本文主要完成了 NVDLA 的 FPGA 平台优化映射工作,首先通过采用调整乘加器电路结构、裁剪冗余模块以及替换门控时钟等方法对 NVDLA 进行优化,之后利用优化后的 NVDLA 搭建硬件加速器平台并完成了相应软件平台的开发。最后,基于加速器平台对 RESNET-50 神经网络进行硬件加速,完成了在 ImageNet 数据集上的图像分类任务。研究结果表明,优化后的 NVDLA 显著提高了 FPGA 内部硬件资源的使用效率,大幅度提升了网络处理性能,与此同时有效降低了功耗。接下来的工作将从软件和硬件两个方面展开,继续丰富硬件加速器的功能。进一步对目标检测、语义分割等不同领域的深度神经网络模型进行硬件加速,从而解决更多工程中遇到的实际问题。

参考文献

[1] Bresson G, Alsayed Z, Li Y, et al. Simultaneous localization and mapping: a survey of current trends in autono-

mous driving[J]. *IEEE Transactions on Intelligent Vehicles*, 2017, 2(7): 194-220

[2] 景晨凯, 宋涛, 庄雷,等. 基于深度卷积神经网络的人脸识别技术综述[J]. 计算机应用与软件,2018,35(1):223-231

[3] 胡硕, 赵银妹, 孙翔. 基于卷积神经网络的目标跟踪算法综述[J]. 高技术通讯,2018,28(3):207-213

[4] Jouppi N P, Young C, Patil N, et al. In-datacenter performance analysis of a tensor processing unit[C]//Proceedings of the 2017 International Symposium on Computer Architecture, Toronto, Canada, 2017: 1-12

[5] Chen Y J, Chen T S, Xu Z W, et al. DianNao family: energy-efficient hardware accelerators for machine learning[J]. *Communications of the ACM*, 2016, 59(11): 105-112

[6] Chen T S, Du Z D, Sun N H, et al. DianNao: a small-footprint high-throughput accelerator for ubiquitous machine-learning[C]//Proceedings of International Conference on Architectural Support for Programming Languages and Operating Systems, Salt Lake City, USA, 2014: 269-284

[7] Chen Y J, Luo T, Liu S L, et al. DaDianNao: a machine-learning supercomputer[C]//Proceedings of the 47th Annual IEEE/ACM International Symposium on Microarchitecture, Cambridge, UK, 2014: 609-622

[8] Liu D F, Chen T S, Liu S L, et al. PuDianNao: a polyvalent machine learning accelerator[C]//Proceedings of International Conference on Architectural Support for Programming Languages and Operating Systems, Istanbul, Turkey, 2015: 369-381

[9] Du Z D, Fasthuber R, Chen T S, et al. ShiDianNao: shifting vision processing closer to the sensor[C]//Proceedings of the 2015 International Symposium on Computer Architecture, Portland, USA, 2015: 92-104

[10] Farabet C, Martini B, Corda B, et al. Neuflow: a runtime reconfigurable dataflow processor for vision[C]//Proceedings of Computer Vision and Pattern Recognition Workshops, Colorado Springs, USA, 2011:109-116

[11] Chen Y H, Emer J, Sze V. Eyeriss: a spatial architecture for energy-efficient dataflow for convolutional neural networks[C]//Proceedings of the 2016 International Symposium on Computer Architecture, Seoul, Korea, 2016:367-379

[12] Zhang C, Li P, Sun G Y, Guan Y J, et al. Optimizing FPGA-based accelerator design for deep convolutional neural networks[C]//Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, Monterey, USA, 2015: 161-170

[13] 余子健, 马德, 严晓浪, 等. 基于 FPGA 的卷积神经网络加速器[J]. 计算机工程,2017,43(1): 109-114,119

[14] 陈鹏, 陈庆清, 王海霞, 等. 基于改进动态配置的 FPGA 卷积神经网络加速器的优化方法[J]. 高技术通讯, 2020,30(3):240-247

[15] Ding C W, Wang S, Liu N, et al. REQ-YOLO: a resource-aware, efficient quantization fameworkfor object detection on FPGAs[J]. *arXiv*: 1909.13396, 2018

[16] Kuon I, Rose J. Measuring the gap between FPGAs and ASICs[C]//Proceedings of the Internation Symposium on Field Programmable Gate Arrays, Monterey, USA,2006: 21-30

[17] Krizhevsky A, Sutskever I, Hinton G. ImageNet classification with deep convolutional neural networks[C]//Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, USA, 2012: 1097-1105

[18] Szegedy C, Liu W, Jia Y Q, et al. Going deeper with convolutions[C]//Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, USA, 2015: 1-9

[19] Simonyan K, Zisserman A. A very deep convolutional networks for large-scale image recognition[J]. *arXiv*: 1409.1556, 2014

[20] He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition[C]//Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, 2016: 770-778

Design of neural network accelerator platform based on NVDLA and FPGA

Guan Zhaokang[*], Zhang Zhiwei[**]

([*] School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074)

([**] Institute of Automation, Chinese Academy of Sciences, Beijing 100190)

Abstract

With the increasing demand for computing power of deep neural networks, traditional general-purpose processors have the disadvantages of low performance and high power consumption in the process of completing inference operations. Therefore, it has become an important development trend to accelerate deep neural networks through dedicated hardware. Field programmable gate array (FPGA) has the advantages of strong reconfigurability, short development cycle, and superior performance. It is very suitable as a hardware acceleration platform for deep neural networks. NVIDIA deep learning accelerator (NVDLA) is NVIDIA's open source neural network hardware accelerator. With its excellent performance, it is highly recognized by the academic and industrial circles. This article mainly studies the optimization mapping problem of NVDLA on the FPGA platform. Through various optimization schemes, the hardware resources inside the FPGA are efficiently utilized and the running performance is improved. Based on the built NVDLA accelerator platform, hardware acceleration of the RESNET-50 neural network is achieved, and the image classification task on the ImageNet dataset is completed. The test results show that the optimized NVDLA can significantly improve the utilization efficiency of hardware resources, and the processing performance can reach up to 30.8 fps, which is a 28 times performance improvement over the edge central processing unit (CPU) accelerator platform.

Key words: NVIDIA deep learning accelerator (NVDLA), field programmable gate array (FPGA), hardware acceleration, module optimization