应用于极致边缘计算场景的卷积神经网络加速器架构设计

吴瑞东[①] 刘 冰[*①] 付 平[①] 纪兴龙[②] 鲁文帅[②]

[①](哈尔滨工业大学电子与信息工程学院 哈尔滨 150000)

[②](启元实验室 北京 100089)

摘 要：针对卷积神经网络在极致边缘计算(UEC)场景应用中的性能和功耗需求，该文针对场景中16 Bit量化位宽的网络模型提出一种不依赖外部存储的卷积神经网络(CNN)加速器架构，该架构基本结构设计为基于现场可编程逻辑门阵列(FPGA)的多核CNN全流水加速器。在此基础上，实现了该加速器的层内映射与层间融合优化。然后，通过构建资源评估模型在理论上完成架构中的计算资源与存储资源评估，并在该理论模型指导下，通过设计空间探索来最大化资源使用率与计算效率，进而充分挖掘加速器在计算资源约束条件下的峰值算力。最后，以纳型无人机(UAV)自主快速人体检测UEC场景为例，通过实验完成了加速器架构性能验证与分析。结果表明，在实现基于Single Shot multibox Detector (SSD)的人体检测神经网络推理中，加速器在100 MHz和25 MHz主频下分别实现了137 FPS和34 FPS的推理速度，对应功耗分别为0.514 W和0.263 W，满足纳型无人机自主计算这种典型UEC场景对图像实时处理的性能与功耗需求。

关键词：极致边缘计算；卷积神经网络；现场可编程逻辑门阵列；加速器架构

中图分类号：TP331 **文献标识码**：A **文章编号**：1009-5896(2022)00-0001-11

DOI: 10.11999/JEIT220130

Convolutional Neural Network Accelerator Architecture Design for Ultimate Edge Computing Scenario

WU Ruidong[①] LIU Bing[①] FU Ping[①] JI Xinglong[②] LU Wenshuai[②]

[①](*School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin* 150000, *China*)

[②](*Qiyuan Laboratory, Beijing* 100089, *China*)

Abstract: In order to meet the requirements of performance and power in Ultimate Edge Computing (UEC) scenario, a Convolutional Neural Network (CNN) accelerator architecture is proposed with 16 Bit quantization model that does not rely on external memory. The basic structure of proposed architecture is Field Programmable Gate Array (FPGA) with multi-core CNN full pipeline accelerator. On this basis, the optimization of intra-layer mapping and inter-layer fusion of accelerator is realized. Then, the evaluation of computing resource and memory resource is theoretically completed by building the corresponding model. Under the guidance of this model, the resource utilization and computing efficiency are maximized through design space exploration, and the peak computing power of accelerator is fully exploited with limited resource constraint. Finally, taking fast human detection of nano Unmanned Aerial Vehicle (UAV) as an example, the verification and analysis of architecture are completed through experiments. Experimental results show that in the inference of human body detection neural network based-on Single Shot multibox Detector (SSD), the performance is achieved with the speed of 137 FPS and 34 FPS at 100 MHz and 25 MHz, and the corresponding power is 0.514 W and 0.263 W, respectively, which meets the performance and power requirements of real-time image processing in typical UEC scenarios such as autonomous computing of nano-UAV.

Key words: Ultimate Edge Computing(UEC); Convolutional Neural Network(CNN); Field Programmable Gate Array (FPGA); Accelerator architecture

收稿日期：2022-02-15；改回日期：2022-07-10；网络出版：2022-07-15

*通信作者： 刘冰 liubing66@hit.edu.cn

基金项目：国家自然科学基金(62171156)

Foundation Item: The National Natural Science Foundation of China (62171156)

1 引言

近年来，物联网技术的快速发展使得边缘计算被广泛应用于各行各业，边缘计算是指在网络边缘执行计算的一种新型计算模型[1]。物联网场景下的边缘计算将计算任务在接近数据源的计算资源上运行，而非以云计算模型为核心的集中式大数据处理方式，不仅降低数据传输带宽与运行功耗，同时较好的保护隐私数据[2]。极致边缘计算(Ultimate Edge Computing, UEC)是传统边缘计算场景的进一步延伸，其特点是面临极小空间、极致性能和极致功耗等设计挑战与需求。自从2012年AlexNet网络的提出[3]，卷积神经网络(Convolutional Neural Network, CNN)在物体分类[4]、目标识别[5]、自然语言处理[6]等领域取得优越的处理性能，并被广泛应用于工业机器人、可穿戴设备、可听戴设备、纳型无人机等UEC场景[7–10]。其中，Crazyflie纳型无人机是一种轻量型飞行器，重量为27 g，机翼间长度为92 mm，飞行最大承重为15 g[11]。基于CrazyFlie无人机的SSD (Single Shot multibox Detector)自主快速人体检测是一种典型UEC应用场景，在该场景中实现快速人体目标检测与跟踪，且计算系统通常面临尺寸(<3 cm×3 cm)、重量(<15 g)、功耗(<0.5 W)、性能(>30 FPS)等因素限制。

UEC场景中承载CNN计算的常用硬件平台包括：嵌入式微控制器(Micro Controller Unit, MCU)、嵌入式图形处理器(Graphics Processing Unit, GPU)、现场可编程逻辑门阵列(Field Programmable Gate Array, FPGA)、基于并行超低功耗(Parallel Ultra-Low-Power, PULP)架构的专用集成电路(Application Specific Integrated Circuit, ASIC)芯片等[12–14]。其中，FPGA具备接口丰富、在线可重构、并行处理能力强、系统功耗低、芯片尺寸小等特性[15–17]，使得FPGA成为适合UEC场景应用的热门平台。

近年来，基于FPGA的CNN加速方法已经得到学术界与工业界的广泛研究[18–24]。在学术界领域，Gong等人[18]提出将CNN所有网络层映射至FPGA，以层间流水的方式完成计算加速；Wang等人[19]基于FPGA提出深度学习加速器单元(Deep Learning Accelerator Unit, DLAU)，与Intel Core2相比实现了36.1倍速度提升；Ding等人[20]面向深度可分离卷积提出层间双缓冲策略，与CPU相比实现了17.6倍性能提升； Blott等人[21]面向低位宽网络提出FPGA加速架构，实现设计空间探索与高度定制推理引擎的自动化设计； Bai等人[22]在FPGA开展面向MobileNetV2加速的可扩展深度分离卷积优化。现有研究分别从计算资源、存储空间、交互带宽等方面进行优化，但是在实现中需要在片外扩充高速存储器，同时多个计算单元需要额外的控制器来完成数据同步与任务调度。在工业界领域，深度学习处理器单元(Deep learning Processing Unit, DPU)作为工业界成熟的FPGA加速器知识产权(Intellectual Property, IP)核，能够直接应用于异构处理器与云端服务器[24]。然而DPU的运行需要额外协处理器，并且无法在部分硬件资源紧张的FPGA上部署。综上所述，虽然已有研究完成了对CNN加速的探索，但是在UEC场景中更加侧重于网络计算效率与性能功耗比，使得现有研究并不能直接应用于资源极端受限的UEC场景，对于如何在面临空间、性能与功耗的极致需求下完成CNN加速器架构设计，是一项具有挑战性的任务。

在应用于资源极端受限的UEC场景下的CNN加速器设计中，本文以降低性能约束(>30 FPS)条件下的系统功耗需求为首要设计目标，分别从加速架构、部署策略和优化方案3个方面进行研究。其中，加速架构适配应用场景需求与任务，通过构建多核CNN加速器匹配不同网络层对加速计算并行度的需求；部署策略在最大化资源利用率的前提下保障对网络层加速的负载均衡，进而提升计算效率；优化方案依托于卷积计算特性降低存储空间需求，实现加速器在多个网络层间的复用。

为了解决上述问题，本文以纳型无人机自主快速人体检测的UEC场景为例，基于FPGA完成16 Bit量化位宽的Body Detection网络部署，对加速器架构进行性能验证与分析。主要完成如下研究工作：(1)针对UEC场景中16 Bit量化位宽的轻量化网络加速，提出一种不依赖外部存储的卷积神经网络加速器架构，该架构基本结构设计为基于FPGA的多核CNN全流水加速器；(2)实现全流水加速器的层内映射与层间融合优化，降低加速器对存储空间与计算资源的消耗；(3)构建加速器资源评估模型，完成对架构中计算资源与存储资源评估；(4)在资源评估模型指导下，通过设计空间探索来最大化资源使用率与计算效率，进而充分挖掘加速器在计算资源受限条件下的峰值算力，提升性能功耗比。

2 纳型无人机平台

纳型无人机平台作为UEC场景中的典型应用平台，其计算系统设计具备代表性特征，本节将从纳型无人机结构特点出发，引入基于FPGA的计算系统。

2.1 纳型无人机结构

如图1所示为纳型无人机平台实物图，该平台

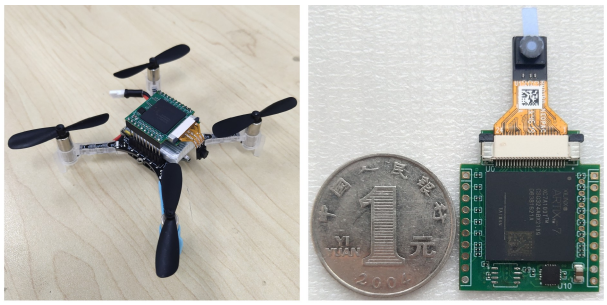
主要由飞控系统、供电系统和计算系统组成。其中，飞控系统由单片机控制器完成飞行姿态控制与调整工作；供电系统由锂电池及其充放电电路组成；计算系统由自主设计的FPGA电路与外接低功耗摄像头组成，完成图像采集和CNN加速计算。

计算系统的设计尺寸为24 mm×27 mm，重量为2.432 g，满足纳型无人机结构尺寸与飞行最大承重需求。与基于PULP架构的GAP8计算系统5 g的重量相比，本文的计算系统减少了纳型无人机在飞行过程中由承重所带来的能量损耗，提升供电系统的能量转换效率，具有延长飞行时间的特点[12]。

2.2 基于FPGA的计算系统

基于FPGA的计算系统包括系统硬件设计与FPGA功能模块设计。

如图2所示为计算系统硬件框图，主要包括连接器、电源、时钟、配置存储器、FPGA和摄像头。其中，FPGA通过4线串行外设接口(Quad Serial Peripheral Interface, QSPI)读取串行配置存储器，在有源晶振驱动下提供主时钟(Main CLocK, MCLK)至摄像头。FPGA与摄像头通过集成电路总线(Inter-Integrated Circuit, IIC)完成内部寄存器配置，并经由数字视频端口(Digital Video Port, DVP)读取原始图像数据，经过计算系统加速后的处理结果通过通用异步收发传输器(Universal Asynchronous Receiver/Transmitter, UART)发送至飞控系统，完成后续的目标跟踪功能。



(a) 纳型无人机　　(b) FPGA计算系统

图 1 纳型无人机结构图

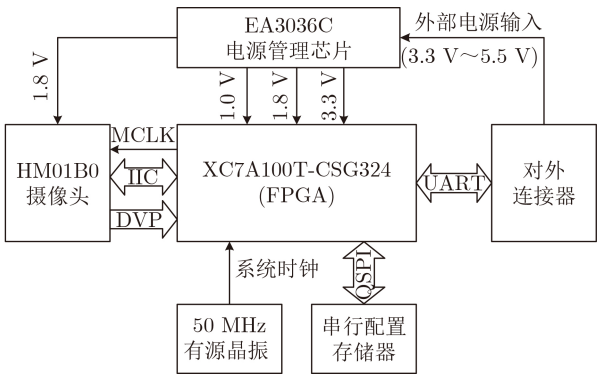


图 2 计算系统硬件框图

如图3所示为FPGA功能模块框图，该模块包括HM01B0初始化、DVP接口图像采集、CNN多核加速与SSD后处理、UART接口输出等。依据时钟域可划分为IIC时钟域、像素时钟(Pixel CLocK, PCLK)域、UART时钟域和多核加速器时钟域。其中IIC时钟域完成摄像头上电初始化配置，PCLK时钟域将采集结果跨时钟传输至CNN多核加速器，多核加速器时钟域完成图像至目标检测的端到端网络推理加速，最后由UART时钟域完成目标检测结果的输出。

3 多核加速器架构

多核加速器架构由多个加速核组成，在运行过程中不依赖外部存储器，并根据网络的层级特征适配算子单元的并行度，优化计算资源与存储资源消耗。合理的架构设计不仅能有效地提升加速性能，也能够降低资源需求与输出延迟，满足UEC应用场景中对性能与功耗的要求。

3.1 加速器设计

UEC场景中CNN加速器的设计不仅需要考虑计算灵活性，即能够适配不同卷积类型、卷积核大小、多维度并行加速等需求，还需要考虑加速器的计算高效性。如图4所示为通用加速器示意图，该加速器包含输入存储、算子单元、池化单元和输出存储。其中，存储器设计采用紧密耦合内存(Tightly Coupled Memory, TCM)层次结构，并依据数据来源划分为特征图紧密耦合内存(Feature-map Tightly Coupled Memory, FTCM)和权重紧密耦合内存(Weight Tightly Coupled Memory, WTCM)。算子单元在空间中层叠，实现FTCM和WTCM在多

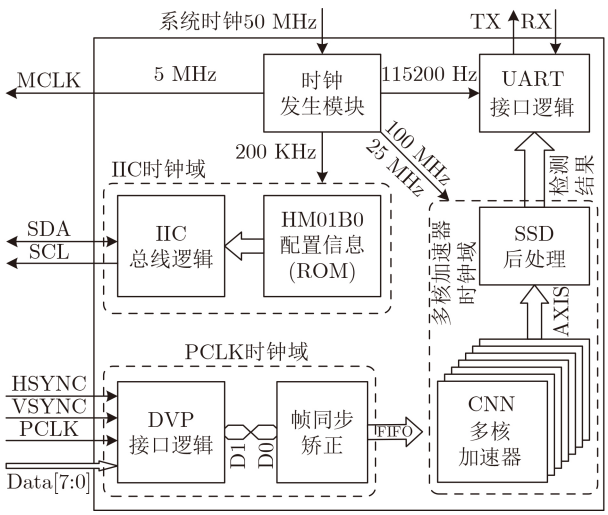


图 3 FPGA功能模块框图

个维度的数据复用，对应加速过程中特征图与多个卷积核并行计算。同理，算子单元输入寄存器维度对应乘加树的输入宽度与计算深度，对应卷积计算中输入通道并行。考虑到卷积神经网络中的池化层操作，本文在加速器设计中融合算子单元与池化单元，简化数据通路设计且减少逻辑占用与系统功耗。

如图5所示为卷积流水计算示意图，依据加速器功能模块分为读取FTCM、读取WTCM、算子单元处理和计算结果输出。在初始阶段t_0等待输入同步信号，t_1开启卷积处理，t_2等待输出同步信号。其模块间多个信号同步采用深度为2的先进先出(First Input First Output, FIFO)存储器，使得在当前操作结束后能够开启下一次计算，在连续多周期流水计算中，将保持如t_2时刻的3级流水线状态，即理论上达到算子单元的100%计算效率。

在完成通用加速器设计后，考虑到UEC场景下所面临的计算资源与存储资源有限的问题，将结合网络层级特征实现加速器资源占用的优化设计。

3.2 层内映射优化

层内映射优化用于解决UEC场景中FTCM存储需求较大的问题，适用于单网络层的卷积计算加速，其优点在于能够根据单网络层特征适配算子单元的数量与输入维度，并结合流水计算特性调整TCM的数据位宽与深度，实现最小存储资源消耗。

如图6所示为针对FTCM的层内映射优化示意图，其中FTCM,WTCM和算子单元构成一个完整的处理单元(Processing Element, PE)。由于多个PE以串行方式连接，结合卷积计算的局部特性，理论上存在与卷积核大小相同的特征图区域即可开启当前计算。与图4中特征图存储方式不同，优化后的存储采用行缓冲策略，此时FTCM存储深度与原始深度相比，输出行数减少为卷积核行数加1，并且在计算效率方面，多个PE以独立串行方式连接，保证优化后的FTCM不影响原有计算效率。

根据层内映射优化策略可知，层内映射优化在输入特征图尺寸较大的网络层具有更好的优化效果，通常对应浅层网络加速，在后续章节中将使用PE表示层内映射优化的加速器。

3.3 层间融合优化

层间融合优化用于解决UEC场景中WTCM存储效率和算子单元计算效率过低的问题，适用于面向连续多网络层(融合层数≥2)的卷积计算加速，其中算子单元采取分时复用的方法依次完成卷积加速，减少计算资源的消耗，同时在分时复用策略下WTCM将支持连续多层权重混合存储在同一内存区域中，进而提高存储空间的有效利用率。

如图7所示为层间融合优化示意图，由算子单元、WTCM、算子FTCM、输入和输出FTCM所组成，其中输入FTCM与上一层加速器共用，输出FTCM与下一层加速器共用。层间融合根据融合层数将权重数据($W0$,$W1$和$W2$)采取混合存储的方式放置在WTCM中，由统一寻址简化存储器在时间上的读写逻辑设计，尤其是在单层权重数据不足以填充单块存储器时，混合存储方式以减少位宽和提升深度的方式提高单个存储空间的有效利用率，降低WTCM存储资源的开销。同时算子单元采用分

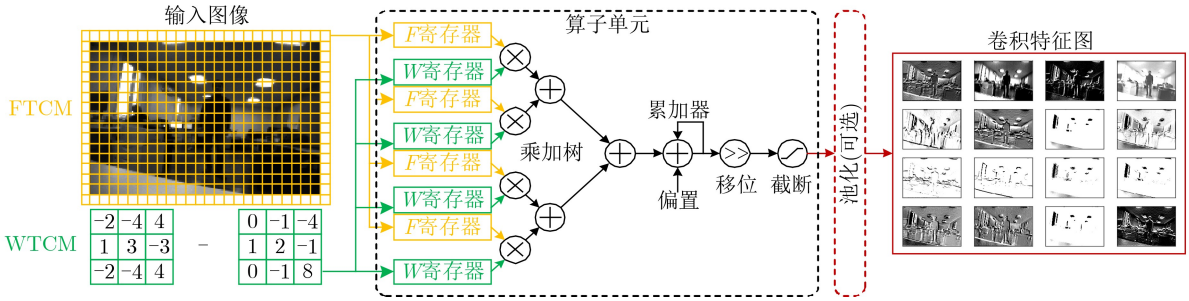


图 4 通用加速器示意图

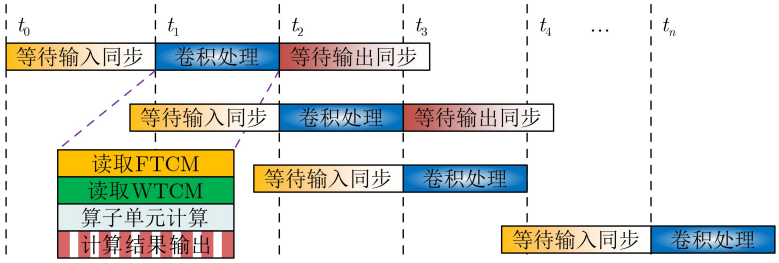


图 5 卷积流水计算示意图

时复用的运行机制，与原有的单层加速器相比降低了对计算资源的需求，减少了同步计算所带来的延迟时间。

根据上述计算和存储优化策略可知，层间融合优化适用于层内并行度低且权重数据分散的网络层，通常对应深层网络加速，在后续章节中使用X表示经过层间融合优化的加速器。

3.4 多核CNN加速

多核CNN加速是对上述所提出加速优化的集成，以最小化UEC场景中存储资源与计算资源消耗为目的，实现网络从输入图像到输出检测结果的端到端多核协同加速。

如图8所示为连续输入的多核CNN加速示意图，其中输入数据为实时采集的原始灰度图像，输出为目标检测的类别与位置信息。多个PE在初始阶段依次启动，待输入稳定后在时间上全并行流水计算，且每个PE的最大运行时间小于图像输入间隔$T0$。同理对于层间融合加速器，在时间上逐个启动，支持多个层间融合加速器并行执行，且融合后的加速器能够减少单次推理的输出延迟。依据在网络中的位置，在多核CNN加速中的对应位置插入SSD加速器完成后处理计算。当连续多帧连续图像输入至多核CNN加速器架构时，每个加速器将以周期并行计算方式完成网络的高效推理。

根据以上分析可知，多核CNN加速器架构将多个优化后的加速器进行有序连接，实现对多帧连

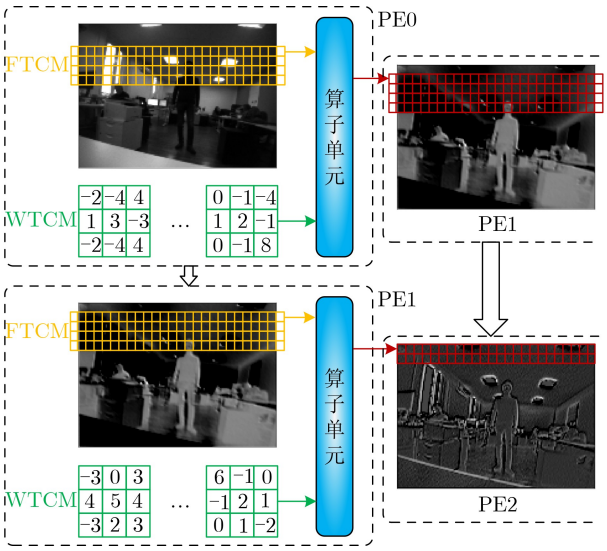


图 6 层内映射示意图

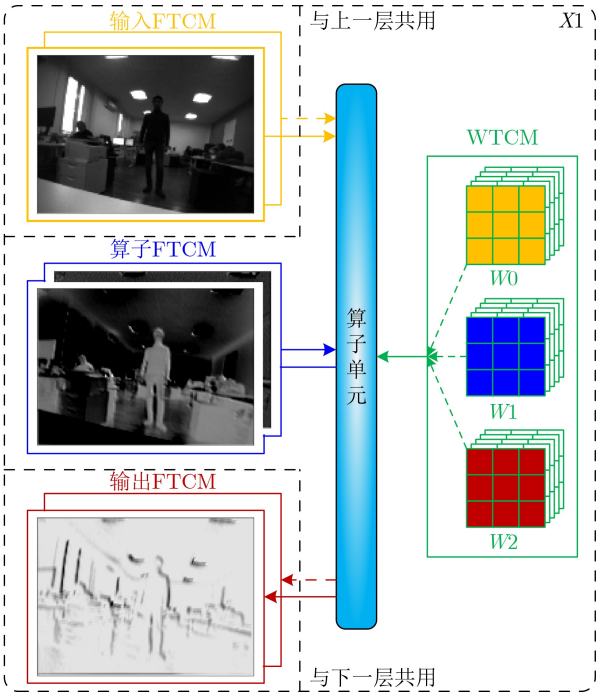


图 7 层间融合优化示意图

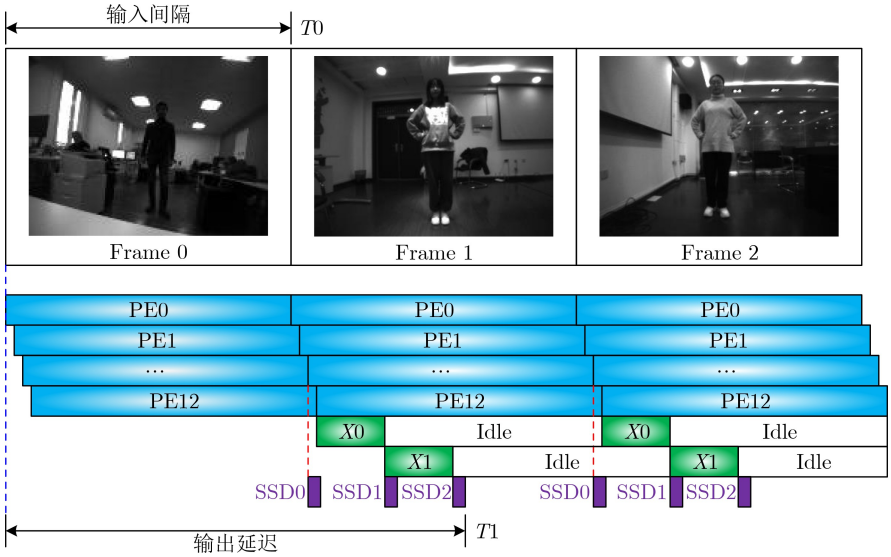


图 8 多核 CNN 加速示意图

续图像的实时处理，保证每个加速器运行在理论最高计算效率状态下，增强推理性能并降低输出延迟。

4 资源评估模型

资源评估模型是对CNN加速器架构中每个加速器资源占用与运行周期的理论评估，以片内可用资源作为约束条件，探索最大化资源利用率与计算效率的设计空间，获取加速器的并行度参数，进而实现最佳推理性能和降低系统功耗的目的。

4.1 计算资源模型

计算资源模型是用于评估加速器所用数字信号处理(Digital Signal Processing, DSP)IP核的数量，直接影响FPGA能达到的峰值算力，间接影响加速后网络完成单次推理所需要的时钟周期数。

典型的CNN计算所需要的操作数OP如式(1)所示，其中R_{O}为输出特征图的行，C_{O}为输出特征的列，M为输入特征图的通道，N为输出特征图的通道，K为卷积核大小

$$\mathrm{OP}=R_{\mathrm{O}}\times C_{\mathrm{O}}\times M\times N\times K\times K\times 2 \tag{1}$$

依据图4中所示算子单元中乘加树结构设计，并行加速包括特征图行并行T_{R}和列并行T_{C}、输入通道并行T_M、输出通道并行T_N和卷积核并行T_K。因此经过并行加速后所需要的时钟周期C为

$$C=\left\lceil\frac{R_{\mathrm{O}}}{T_{\mathrm{R}}}\right\rceil\times\left\lceil\frac{C_{\mathrm{O}}}{T_{\mathrm{C}}}\right\rceil\times\left\lceil\frac{M}{T_M}\right\rceil\times\left\lceil\frac{N}{T_N}\right\rceil\times\left\lceil\frac{K\times K}{T_K}\right\rceil \tag{2}$$

在上述并行条件下，完成并行加速所消耗的DSP数量V为

$$V=T_{\mathrm{R}}\times T_{\mathrm{C}}\times T_M\times T_N\times T_K\times\alpha \tag{3}$$

其中，α为不同数据类型在算子单元中完成单次乘加所消耗的DSP数量，例如16 Bit定点数α为1，8 Bit定点数α为0.5。因此，完成L层CNN加速所消耗总DSP数量V_{D}为

$$V_{\mathrm{D}}=\alpha\sum_{i=1}^{L}T_{\mathrm{R}_i}\times T_{\mathrm{C}_i}\times T_{M_i}\times T_{N_i}\times T_{K_i} \tag{4}$$

考虑到实际部署中存在层间融合优化来降低DSP使用量，则多核CNN加速器架构的总DSP数量V_{D}进一步更新为

$$\left.\begin{aligned}&V_{\mathrm{D}}=\alpha\left(V_{\mathrm{PE}}+V_X\right)\\&V_{\mathrm{PE}}=\sum_{i=1}^{\beta}T_{\mathrm{R}_i}\times T_{\mathrm{C}_i}\times T_i\times T_{N_i}\times T_{K_i}\\&V_X=\sum_{i=1}^{\gamma}T_{\mathrm{R}_i}\times T_{\mathrm{C}_i}\times T_{M_i}\times T_{N_i}\times T_{K_i}\\&L=\beta+\sum_{j=1}^{\gamma}\gamma_j\end{aligned}\right\} \tag{5}$$

其中，β为层内映射优化加速器PE的数量，γ为层间融合优化加速器X的数量，γ_j为第j个加速器融合的层数。

则基于上述加速策略，多核CNN加速器架构中每个加速器对应的时钟周期为

$$C_{\mathrm{PE}_i}=\left\lceil\frac{R_{\mathrm{O}_i}}{T_{\mathrm{R}_i}}\right\rceil\times\left\lceil\frac{C_{\mathrm{O}_i}}{T_{\mathrm{C}_i}}\right\rceil\times\left\lceil\frac{M_i}{T_{M_i}}\right\rceil\times\left\lceil\frac{N_i}{T_{N_i}}\right\rceil\times\left\lceil\frac{K_i\times K_i}{T_{K_i}}\right\rceil \tag{6}$$

$$C_{X_j}=\sum_{n=1}^{\gamma_j}\left\lceil\frac{R_{\mathrm{O}_n}}{T_{\mathrm{R}_j}}\right\rceil\times\left\lceil\frac{C_{\mathrm{O}_n}}{T_{\mathrm{C}_i}}\right\rceil\times\left\lceil\frac{M_n}{T_{M_j}}\right\rceil\times\left\lceil\frac{N_n}{T_{N_j}}\right\rceil\times\left\lceil\frac{K_n\times K_n}{T_{K_j}}\right\rceil \tag{7}$$

其中，C_{PE}为加速器PE的时钟周期，C_X为加速器X的时钟周期。因此若要保证对输入图像的实时处理，则理论上加速器的最大时钟周期应小于等于图像的输入间隔周期，如式(8)所示

$$\max\left(C_{\mathrm{PE}_i},C_{X_j}\right)\le T_0, i\in[1,\beta], j\in[1,\gamma] \tag{8}$$

综上所述可知，计算资源模型能够预估出所消耗的总DSP数量和最大时钟周期数，便于多核CNN加速器的负载均衡，为后续FPGA加速性能的评估以及适配输入图像帧率做参考。

4.2 存储资源模型

存储资源模型用于评估片内存储器 (Block Random Access Memory, BRAM)的资源消耗，包括FTCM和WTCM存储模型。

由图4中通用加速器的FTCM和WTCM存储策略可知，FTCM由并行度和输入特征图大小所决定，若网络采用的量化数据位宽为ε，则FTCM对应的位宽F_{W}和深度F_{D}如式(9)和式(10)所示

$$F_{\mathrm{W}}=T_{\mathrm{R}}\times T_{\mathrm{C}}\times T_M\times\varepsilon \tag{9}$$

$$F_{\mathrm{D}}=\left\lceil\frac{R_{\mathrm{I}}\times C_{\mathrm{I}}\times M}{T_{\mathrm{R}}\times T_{\mathrm{C}}\times T_M}\right\rceil \tag{10}$$

其中，R_{I}和C_{I}为输入的行与列，同理WTCM的位宽W_{W}和深度W_{D}为

$$W_{\mathrm{W}}=T_M\times T_N\times\varepsilon \tag{11}$$

$$W_{\mathrm{D}}=\left\lceil\frac{M\times N\times K\times K}{T_M\times T_N\times T_K}\right\rceil \tag{12}$$

FPGA片内BRAM基本单元为BRAM18K，在不同数据位宽ε下对应的深度$D_{(\varepsilon)}$为

$$D_{(\varepsilon)}=\begin{cases}2048,\varepsilon\in[5,9]\\1024,\varepsilon\in[10,18]\\512,\ \ \varepsilon\in[19,36]\end{cases} \tag{13}$$

因此，FTCM和WTCM消耗的BRAM18K数量F_{B}和W_{B}为

$$F_{\mathrm{B}}=\left\lceil\frac{F_{\mathrm{D}}\times F_{\mathrm{W}}}{D_{(\varepsilon)}}\right\rceil \tag{14}$$

$$W_{\mathrm{B}}=\left\lceil\frac{W_{\mathrm{D}}\times W_{\mathrm{W}}}{D_{(\varepsilon)}}\right\rceil \tag{15}$$

根据图6中所示行缓冲策略，优化后的FTCM深度F_{DK}为如式(16)所示，通过与式(10)对比，分子项由原来的R_{I}降低为K+1，实现了优化存储深度需求与BRAM资源消耗的目的。

$$F_{\mathrm{DK}}=\left\lceil\frac{(K+1)\times C_{\mathrm{I}}\times M}{T_{\mathrm{R}}\times T_{\mathrm{C}}\times T_{M}}\right\rceil \tag{16}$$

同理根据图7中所示的混合存储策略，优化后第j个层间融合加速器中WTCM深度为

$$W_{\mathrm{D}_j}=\left\lceil\frac{\sum_{n=1}^{\gamma_j}M_n\times N_n\times K_n\times K_n}{T_{M_j}\times T_{N_j}\times T_{K_j}}\right\rceil \tag{17}$$

基于上述优化后的存储策略，由于多核CNN加速器架构对BRAM18K的消耗主要由FTCM存储V_{F}和WTCM存储V_{W}组成，因此优化后的BRAM总消耗V_{B}为

$$\left.\begin{aligned}V_{\mathrm{B}}&=V_{\mathrm{F}}+V_{\mathrm{W}}\\V_{\mathrm{F}}&=\sum_{i=1}^{\beta}\left\lceil\frac{F_{\mathrm{DK}_i}\times F_{\mathrm{W}_i}}{D_{(\varepsilon)}}\right\rceil+5\times\sum_{j=1}^{\gamma}F_{\mathrm{B}_j}\\V_{\mathrm{W}}&=\sum_{i=1}^{\beta}W_{\mathrm{B}_i}+\sum_{j=1}^{\gamma}\left\lceil\frac{W_{\mathrm{D}_j}\times W_{\mathrm{W}_j}}{D_{(\varepsilon)}}\right\rceil\end{aligned}\right\} \tag{18}$$

综上所述，存储资源模型从多核CNN加速器架构出发，优化后的特征图与权重存储深度，为片上存储资源极其受限的UEC场景中加速器部署做准备。

5　实验验证与对比

实验部分由资源评估模型提供理论指导，在设计空间内探索FPGA加速器的并行度参数，验证资源消耗、加速性能和运行功耗，并与ARM和GAP8平台对比网络推理速度与性能功耗比。

5.1　设计空间探索

如表1所示为基于SSD的Body Detection网络结构，其中量化方式采用16 Bit动态定点量化。考虑到FPGA片内资源的限制，本文采用缩小设计空间探索区域的方法并将搜索空间限制在T_M与T_N内，探索结果如表1所示。

根据表1中所提供的并行度进行理论分析，得到表2中计算资源DSP与存储资源BRAM36K的理论值。其中浅层网络的计算量较大，需要大量DSP资源来提升并行度；随着网络深度的增加，网络计算量与权重参数开始减少，使得中间网络层的DSP与BRAM消耗保持在较低水平；最后是深层网络的层间融合加速，存储部分需要额外的FTCM用于融合后特征图，计算部分通过复用算子单元的方式提升了深层网络的计算效率。

通过上述设计空间探索得到DSP与BRAM的理论资源消耗，以及负载均衡后运行周期的参考值，为后续板级部署的资源与性能评估做参考。

5.2　资源占用与功耗

依据上述并行度探索结果，完成多核CNN加速器架构与功能模块部署，部署后片内资源消耗与占比如表3所示。

结合表3内容可知，BRAM36K与DSP的消耗占比远大于其他资源，理论上DSP消耗占比越大表明峰值算力越高，对应加速效果越好，即并行度探索结果最优。BRAM36K用于构建FTCM与WTCM，合理的BRAM资源占比简化了硬件设计，在无需外扩存储器的前提下保证加速性能。

为了进一步验证资源评估模型的有效性，SSD

表 1　网络结构参数与并行度探索结果

网络层	输入大小	卷积核大小	T_M	T_N	周期
0	160×120×1	3×3×1×32	1	8	691200
1	160×120×32	1×1×32×32	32	1	614400
2	160×120×32	3×3×32	1	8	691200
3	160×120×32	1×1×32×32	32	1	614400
4	80×60×32	3×3×32×16	32	1	691200
5	80×60×16	1×1×16×16	2	1	614400
6	80×60×16	3×3×16	1	1	691200
7	80×60×16	1×1×16×16	2	1	614400
8	40×30×16	3×3×16×16	4	1	691200
9	40×30×16	1×1×16×16	1	1	307200
10	40×30×16	3×3×16	1	1	172800
11	40×30×16	1×1×16×16	1	1	307200
SSD0	40×30×16	–	–	–	–
12	20×15×16	3×3×16×16	1	1	691200
13	20×15×16	1×1×16×16			
14	20×15×16	3×3×16	1	1	196800
15	20×15×16	1×1×16×16			
SSD1	20×15×16	–	–	–	–
16	10×7×16	3×3×16×16			
17	10×7×16	1×1×16×16	1	1	207200
18	10×7×16	3×3×16			
19	10×7×16	1×1×16×16			
SSD2	10×7×16	–	–	–	–

表 2 加速器资源消耗理论值

网络层	加速器	DSP	BRAM36K
0	PE0	8	0.5
1	PE1	32	8
2	PE2	8	12
3	PE3	32	8
4	PE4	32	8
5	PE5	2	1.5
6	PE6	1	3
7	PE7	2	1.5
8	PE8	4	2
9	PE9	1	1
10	PE10	1	1.5
11	PE11	1	1
12	PE12	1	2.5
13～15	X0	1	10.5
16～19	X1	1	6
总计	15	127	67

表 3 部署资源消耗

资源类别	消耗	片内总计	占比(%)
LUT	24814	63400	39.14
LUTRAM	1236	19000	6.51
FF	17516	126800	13.81
BRAM36K	106	135	78.52
DSP	156	240	65.00
IO	17	210	8.10

后处理部分资源消耗如表4所示，结合表3内容可知，DSP资源实际消耗与表2所提供的消耗理论值相对应，BRAM资源实际消耗符合理论消耗的预估结果，表明本文所提出的资源评估模型能够完成对DSP计算资源与BRAM存储的理论预估。

同时，板级部署在不同频率下的功耗结果如表5所示。对比功耗组成，在降低频率过程中静态功耗保持不变，动态功耗降低。在静态功耗方面，由于计算系统存在额外元器件的能量消耗，实际测量值与评估值相比上升至0.184 W；在动态功耗方面，频率提升所消耗的动态功耗变化量为0.251 W，符合理论评估值变化量0.244 W的预期结果。在25 MHz频率下功耗实际测量值为0.263 W，符合UEC场景中最大功耗0.5 W的应用需求。

5.3 加速性能对比

本文采用Cortex-A53高性能ARM处理器和基于PULP架构的GAP8处理器作为对比平台，且上述平台在尺寸、重量等方面符合CrazyFlie无人机对计算系统的要求。

表 4 SSD资源消耗

	DSP	BRAM36K
SSD0	18	9
SSD1	6	6
SSD2	4	6
总计	28	21

表 5 不同频率下功耗结果

类别	100 MHz 功耗(W)	25 MHz功耗(W)
Clock	0.062	0.016
Logic	0.073	0.019
Signal	0.098	0.022
BRAM	0.040	0.010
DSP	0.050	0.012
IO	<0.001	<0.001
Static	0.109	0.109
评估值	0.432	0.188
测量值	0.514	0.263

与ARM处理器的软硬件性能对比采用如下测试方法：(1)输入单帧图像，记录从图像开始传输到结果输出的处理时间；(2)输入多帧图像，记录完成100次推理结果输出的总时间，即对应峰值计算性能。测试结果如表6所示，FPGA加速器架构的频率低于ARM平台，但由于加速器架构采用层内并行、层间流水的多核工作方式，获得优于ARM平台的峰值计算性能。从输入间隔结果可知，ARM平台输入间隔等于所有网络层计算时间146.277 ms的总和，对应帧率为7FPS，而FPGA平台具备多核流水计算能力，图像输入间隔为7.287 ms，对应帧率为137 FPS，与ARM平台相比获得20倍的性能提升。

同理，本文在GAP8平台对比完成Body Detection检测网络的推理性能与功耗测试，实验测试条件如下：使能GAP8片内8核并行处理单元，并设置并行单元的时钟频率为200 MHz，由Autotiler工具完成网络分块切片与逐层调度工作，测量网络单次推理时间与运行功耗。测试结果分别从算力、网络推理时间、运行功耗和性能功耗比4个完成性能对比，结果如表7所示。在算力与推理时间方面，本文的多核CNN加速器架构充分利用片内计算资源，获得优于ARM与GAP8的结果；在功耗方面，基于PULP架构的GAP8平台拥有最低的运行功耗；在性能功耗比方面，由于GAP8平台具备更低的功耗，因而其性能功耗比优于ARM平台，更进一步，

由于FPGA平台具备更高的算力，其性能功耗比优于其他平台，在100 MHz下达到43.018 GOPS/W的计算性能。

基于上述实验结果，考虑到UEC场景中纳型无人机自主快速人体检测对性能的要求，本文所设计的加速器架构25 MHz频率下实现34 FPS网络推理性能，且运行功耗为0.263 W，对应性能功耗比为21.019 GOPS/W，满足UEC场景下对性能与功耗的应用需求。

5.4 相关工作比较

与已有基于FPGA的CNN加速器对比结果如表8所示，本文分别从网络类型、量化位宽、算力、计算效率、性能功耗比等方面展开。受限于应用场景对芯片选型的限制，虽然可用计算资源DSP数量远小于对比文献，但是在计算效率方面，本文所设计的加速器架构在单位周期内每个DSP贡献的平均计算量为1.728，优于文献[20]和文献[22]所提方案。在与同类型架构的比较中，文献[23]中加速器运行中不需要与外界存储器交互，因而其性能功耗比优于文献[20]，表明减少带宽交互能够降低系统功耗。本文所提架构在25 MHz下达到21.019 GOPS/W，高于文献[20,23]中实验结果，同时在100 MHz下其动态功耗占比优于25 MHz，使得性能功耗比能够进一步提升至43.108 GOPS/W。

表 6　软硬件处理性能对比结果

计算区间	ARM计算时间(ms) 1.2 GHz	FPGA加速时间(ms) 100 MHz
PE0～PE12	140.424	9.291
PE0～X1	141.179	13.280
SSD0	2.782	1.117
SSD1	1.952	0.939
SSD2	0.364	0.233
输入间隔	146.277	7.287
帧率	7 FPS	137 FPS

表 7　不同平台性能对比结果

类别	ARM	GAP8	FPGA	
	1.2 GHz	200 MHz	25 MHz	100 MHz
算力(GOPS)	1.101	0.482	5.528	22.111
推理时间(ms)	146.277	334.625	29.148	7.287
功耗(W)	2.617	0.196	0.263	0.514
性能功耗比(GOPS/W)	0.421	2.459	21.019	43.018

6　结论

本文提出一种不依赖外部存储的卷积神经网络加速器架构，该架构基本结构设计为基于FPGA的多核CNN全流水加速器。同时本文实现加速器的层内映射与层间融合优化，降低加速器对存储空间与计算资源的需求。通过构建资源评估模型在理论上完成计算资源与存储资源评估，并在该理论指导下通过设计空间探索最大化资源使用率与计算效率，充分挖掘加速器在资源约束条件下的峰值算力，提升性能功耗比。实验结果表明，本文所提出的加速器架构在实现基于SSD的人体检测神经网络推理中，ARM平台软件处理性能高为7 FPS，对应功耗为2.617 W。FPGA平台在100MHz频率下实现137 FPS的推理速度，在25 MHz频率下获得34 FPS推理速度与0.263 W功耗，并达到了21.019 GOPS/W的性能功耗比，与ARM平台相比获得了5倍的性能提升，功耗降低90%，性能功耗比提升50倍。综上所述，本文所提出的加速器架构设计能够满足纳型无人机自主计算这种典型UEC场景对图像实时处理的性能与功耗需求。

参 考 文 献

[1] BIANCHI V, BASSOLI M, LOMBARDO G, *et al*. IoT wearable sensor and deep learning: An integrated approach

表 8　与相关工作比较结果

类别	文献[20]	文献[20]	文献[22]	文献[23]	本文	
平台	GX1150	GX1150	10AS066N	XC7Z045	XC7A100T	
网络类型	卷积	深度分离卷积	MobileNetV2	Face Detector	Body Detection	
量化位宽(Bit)	16	16	16	16	16	
DSP	760	712	1278	–	128	
频率(MHz)	150	180	133	150	25	100
算力(GOPS)	87.500	98.910	170.600	137	5.528	22.111
功耗(W)	8.69	8.52	–	9.63	0.263	0.514
计算效率(OPS/DSP/f)	0.768	0.772	1.004	–	1.728	1.727
性能功耗比(GOPS/W)	10.069	11.609	–	14.226	21.019	43.018

for personalized human activity recognition in a smart home environment[J]. *IEEE Internet of Things Journal*, 2019, 6(5): 8553–8562. doi: 10.1109/JIOT.2019.2920283.

[2] 施巍松, 张星洲, 王一帆, 等. 边缘计算: 现状与展望[J]. 计算机研究与发展, 2019, 56(1): 69–89. doi: 10.7544/issn1000-1239.2019.20180760.
SHI Weisong, ZHANG Xingzhou, WANG Yifan, *et al*. Edge computing: State-of-the-art and future directions[J]. *Journal of Computer Research and Development*, 2019, 56(1): 69–89. doi: 10.7544/issn1000-1239.2019.20180760.

[3] KRIZHEVSKY A, SUTSKEVER I, and HINTON G E. ImageNet classification with deep convolutional neural networks[C]. The 25th International Conference on Neural Information Processing Systems, Lake Tahoe, USA, 2012: 1097–1105.

[4] ROY S K, KRISHNA G, DUBEY S R, *et al*. HybridSN: Exploring 3-D–2-D CNN feature hierarchy for hyperspectral image classification[J]. *IEEE Geoscience and Remote Sensing Letters*, 2020, 17(2): 277–281. doi: 10.1109/LGRS.2019.2918719.

[5] LIN T Y, GOYAL P, GIRSHICK R, *et al*. Focal loss for dense object detection[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020, 42(2): 318–327. doi: 10.1109/TPAMI.2018.2858826.

[6] USAMA M, AHMAD B, SONG Enmin, *et al*. Attention-based sentiment analysis using convolutional and recurrent neural network[J]. *Future Generation Computer Systems*, 2020, 113: 571–578. doi: 10.1016/j.future.2020.07.022.

[7] WAN Shaohua and GOUDOS S. Faster R-CNN for multi-class fruit detection using a robotic vision system[J]. *Computer Networks*, 2020, 168: 107036. doi: 10.1016/j.comnet.2019.107036.

[8] ACHARYA J and BASU A. Deep neural network for respiratory sound classification in wearable devices enabled by patient specific model tuning[J]. *IEEE Transactions on Biomedical Circuits and Systems*, 2020, 14(3): 535–544. doi: 10.1109/TBCAS.2020.2981172.

[9] WANG Yu, YANG Jie, LIU Miao, *et al*. LightAMC: Lightweight automatic modulation classification via deep learning and compressive sensing[J]. *IEEE Transactions on Vehicular Technology*, 2020, 69(3): 3491–3495. doi: 10.1109/TVT.2020.2971001.

[10] WU Huaqiang, LYU Feng, ZHOU Conghao, *et al*. Optimal UAV caching and trajectory in aerial-assisted vehicular networks: A learning-based approach[J]. *IEEE Journal on Selected Areas in Communications*, 2020, 38(12): 2783–2797. doi: 10.1109/JSAC.2020.3005469.

[11] Bitcraze. Crazyflie 2.1[EB/OL]. https://www.bitcraze.io/products/crazyflie-2-1/, 2022.

[12] PALOSSI D, LOQUERCIO A, CONTI F, *et al*. A 64-mW DNN-based visual navigation engine for autonomous nano-drones[J]. *IEEE Internet of Things Journal*, 2019, 6(5): 8357–8371. doi: 10.1109/JIOT.2019.2917066.

[13] NICULESCU V, LAMBERTI L, CONTI F, *et al*. Improving autonomous nano-drones performance via automated end-to-end optimization and deployment of DNNs[J]. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 2021, 11(4): 548–562. doi: 10.1109/JETCAS.2021.3126259.

[14] PALOSSI D, ZIMMERMAN N, BURRELLO A, *et al*. Fully onboard AI-powered human-drone pose estimation on ultralow-power autonomous flying nano-UAVs[J]. *IEEE Internet of Things Journal*, 2022, 9(3): 1913–1929. doi: 10.1109/JIOT.2021.3091643.

[15] 刘勤让, 刘崇阳. 利用参数稀疏性的卷积神经网络计算优化及其FPGA加速器设计[J]. 电子与信息学报, 2018, 40(6): 1368–1374. doi: 10.11999/JEIT170819.
LIU Qinrang and LIU Chongyang. Calculation optimization for convolutional neural networks and FPGA-based accelerator design using the parameters sparsity[J]. *Journal of Electronics & Information Technology*, 2018, 40(6): 1368–1374. doi: 10.11999/JEIT170819.

[16] 秦华标, 曹钦平. 基于FPGA的卷积神经网络硬件加速器设计[J]. 电子与信息学报, 2019, 41(11): 2599–2605. doi: 10.11999/JEIT190058.
QIN Huabiao and CAO Qinping. Design of convolutional neural networks hardware acceleration based on FPGA[J]. *Journal of Electronics & Information Technology*, 2019, 41(11): 2599–2605. doi: 10.11999/JEIT190058.

[17] YUAN Tian, LIU Weiqiang, HAN Jie, *et al*. High performance CNN accelerators based on hardware and algorithm co-optimization[J]. *IEEE Transactions on Circuits and Systems I:Regular Papers*, 2021, 68(1): 250–263. doi: 10.1109/TCSI.2020.3030663.

[18] GONG Lei, WANG Chao, LI Xi, *et al*. MALOC: A fully pipelined FPGA accelerator for convolutional neural networks with all layers mapped on chip[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018, 37(11): 2601–2612. doi: 10.1109/TCAD.2018.2857078.

[19] WANG Chao, GONG Lei, YU Qi, *et al*. DLAU: A scalable deep learning accelerator unit on FPGA[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2017, 36(3): 513–517. doi: 10.1109/TCAD.2016.2587683.

[20] DING Wei, HUANG Zeyu, HUANG Zunkai, *et al*. Designing efficient accelerator of depthwise separable convolutional neural network on FPGA[J]. *Journal of*

Systems Architecture, 2019, 97: 278–286. doi: 10.1016/j.sysarc.2018.12.008.

[21] BLOTT M, PREUßER T B, FRASER N J, *et al*. FINN-*R*: An end-to-end deep-learning framework for fast exploration of quantized neural networks[J]. *ACM Transactions on Reconfigurable Technology and Systems*, 2018, 11(3): 16. doi: 10.1145/3242897.

[22] BAI Lin, ZHAO Yiming, and HUANG Xinming. A CNN accelerator on FPGA using depthwise separable convolution[J]. *IEEE Transactions on Circuits and Systems II:Express Briefs*, 2018, 65(10): 1415–1419. doi: 10.1109/TCSII.2018.2865896.

[23] GUO Kaiyuan, SUI Lingzhi, QIU Jiantao, *et al*. Angel-eye: A complete design flow for mapping CNN onto embedded FPGA[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018, 37(1): 35–47. doi: 10.1109/TCAD.2017.2705069.

[24] ZHU Jiang, WANG Lizan, LIU Haolin, *et al*. An efficient task assignment framework to accelerate DPU-based convolutional neural network inference on FPGAs[J]. *IEEE Access*, 2020, 8: 83224–83237. doi: 10.1109/ACCESS.2020.2988311.

吴瑞东：男，1992年生，博士生，研究方向为高性能异构计算.

刘　冰：男，1982年生，副教授，研究方向为高性能计算、计算视觉.

付　平：男，1965年生，教授，研究方向为自动测试.

纪兴龙：男，1988年生，研究员，研究方向为嵌入式智能计算.

鲁文帅：男，1987年生，研究员，研究方向为智能微系统.

责任编辑：余　蓉