DOI:10.19651/j.cnki.emt.2417613

基于 FPGA 的语义信息处理加速器设计*

李俊锋[1] 谭北海[2] 郑宇凡[1] 陈汉杰[1] 余 荣[1]
(1.广东工业大学自动化学院 广州 510006;2.广东工业大学集成电路学院 广州 510006)

摘 要:在语义通信中,图像语义信息处理高度依赖于计算复杂度高的卷积神经网络,尤其在处理高分辨率图像时,对计算性能要求更高,这对语义通信在边缘场景中的应用提出了巨大挑战。为此,本文提出了一种基于 FPGA 的语义信息处理加速器,创新性地将卷积神经网络编码器和 rANS 编码融合在同一硬件加速器中。具体而言,加速器采用融合乘累加器的脉动阵列架构、循环分块策略和双缓存结构,以充分利用 FPGA 的并行计算能力与片上存储资源,提升数据传输效率与计算性能。每个处理单元集成多个乘累加单元,可在每个时钟周期完成两个 INT8 乘法并局部累加。最终,对输出特征采用 rANS 进行 8 路并行编码,进一步压缩特征数据。实验结果表明,在 ZCU104 平台上,本设计在处理 1080P 图像时达到 300.5 GOPS 的吞吐量,能效比为 66.77 GOPS/W,处理速度比 Intel CPU 提升约 6 倍,比 ARM CPU 提升约 58 倍。与其他 FPGA 加速器相比,BRAM 效率分别提升约 730%、40%和 63%,能效比分别提升约 802%、60%和 3%,DSP 效率分别提升约 476%、70%和 133%。所提出的加速器在性能上具有显著优势,可高效处理图像语义信息,具有广泛的实际应用意义。

关键词:卷积神经网络;语义通信;图像压缩;FPGA;硬件加速器
中图分类号:TN46 **文献标识码**:A **国家标准学科分类代码**:510.4030

Design of a semantic information processing accelerator based on FPGA

Li Junfeng[1] Tan Beihai[2] Zheng Yufan[1] Chen Hanjie[1] Yu Rong[1]
(1. School of Automation, Guangdong University of Technology, Guangzhou 510006, China;
2. School of Integrated Circuit, Guangdong University of Technology, Guangzhou 510006, China)

Abstract: In semantic communication, image semantic information processing heavily relies on computationally intensive convolutional neural networks, which require higher computational performance, especially when handling high-resolution images. This presents a significant challenge for the application of semantic communication in edge scenarios. To address this, this paper proposes an FPGA-based semantic information processing accelerator, which innovatively integrates the convolutional neural network encoder and rANS encoding in the same hardware accelerator. Specifically, the accelerator adopts a systolic array architecture combined with multiply-accumulate units, loop tiling strategy, and a dual-buffer structure to fully leverage the parallel computing capabilities and on-chip storage resources of the FPGA, improving data transmission efficiency and computational performance. Each processing unit integrates multiple multiply-accumulate units, capable of performing two INT8 multiplications and local accumulation in each clock cycle. Finally, rANS is used for 8-way parallel encoding of the output features, further compressing the feature data. Experimental results show that, on the ZCU104 platform, the design achieves a throughput of 300.5 GOPS with a power efficiency of 66.77 GOPS/W when processing 1080P images, providing a processing speed approximately 6 times faster than Intel CPUs and 58 times faster than ARM CPUs. Compared with other FPGA accelerators, the BRAM efficiency improves by approximately 730%, 40%, and 63%, the energy efficiency by approximately 802%, 60% and 3%, and the DSP efficiency by approximately 476%, 70% and 133%. The proposed accelerator demonstrates significant performance advantages and can efficiently process image semantic information, offering broad practical application potential.

Keywords: convolutional neural network; semantic communication; image compression; FPGA; hardware accelerator

0 引 言

语义通信[1-3]作为一项新兴技术,旨在提取并传递图像和视频的核心语义信息,以在减少数据传输量的同时,保持信息的完整性和有效性。与传统基于比特层面[4-6]的压缩方法不同,语义通信高度依赖卷积神经网络[7-8]来实现高层

收稿日期:2024-12-12
* 基金项目:国家自然科学基金重点项目(U22A2054)资助

次的特征提取和信息压缩。尽管神经网络在提升压缩效率方面表现优异,但其计算复杂度较高,尤其在处理高分辨率图像时,对计算资源的需求显著增加。这种计算负荷在资源有限的边缘计算环境中更为明显,限制了现有语义通信技术在这些场景中的应用效率。

现场可编程门阵列(field programmable gate array,FPGA)以其独特的并行处理能力和高效的硬件资源利用率,成为实现高性能深度学习模型的理想平台。相较于传统的中央处理器(central processing unit,CPU)和图像处理器(graphics processing unit,GPU),FPGA 在功耗和延迟方面具有显著优势,在实时处理和低功耗应用场景中,得到了广泛关注和应用[9-10]。文献[11]提出了一种可调节并行度的神经网络加速器设计,可根据平台的资源量平衡性能和功耗,但该加速器使用 16 bit 浮点进行计算,导致片上资源消耗较大,并且数字信号处理器(digital signal processor,DSP)的利用率不高。文献[12]使用 Winograd 快速算法[13]减少了卷积计算所需的乘法次数,在降低 DSP 消耗的同时提升了计算性能。但该算法使数据位宽由 8 bit 增加到 10 bit,需要消耗额外的资源才能在一个 DSP 中同时执行两个 INT8 乘法操作。文献[14]提出一种双缓存脉动阵列计算阵列,在处理系统(process system,PS)端进行数据重排,在可编程逻辑(programmable logic,PL)端的脉动阵列中执行通用矩阵乘法操作。该设计提高了数据复率和并行计算效率,但数据重排过程需要额外的存储空间来保存重排后的数据,并增加了数据搬运的带宽开销。此外,这种设计需要 PS 端增加复杂的控制逻辑来协调数据重排与 PL 端矩阵计算过程,进而提升了系统的整体复杂性。

神经网络编码器在对输入图像进行特征压缩后,还需要通过算术编码对这些特征进行进一步的压缩,从而减少数据量。区间非对称数字系统(range asymmetric numeral systems,rANS)[15]熵编码器具有优越的编码性能和压缩率,目前已经广泛应用于图像语义信息压缩领域[16-18]。然而,传统的 rANS 并行实现方式通常会将输入符号序列划分为多个分区,然后为每个分区分配独立的编码器来进行并行编码[19]。这种分区编码方式需要为每个分区提供独立的存储空间来缓存输入和输出数据,且导致数据访问缺乏连续性,进而增加存储和带宽的压力,使其难以在 FPGA 上实现高效部署。

针对上述挑战,本文设计并实现了一种基于 FPGA 的语义信息处理加速器,优化计算与数据传输效率。首先,本文设计了一种脉动阵列与乘累加器相结合的计算架构,支持高并行 INT8 卷积运算,每个计算单元在每个时钟周期可执行两个 INT8 乘法并局部累加,从而提升 DSP 利用率并降低计算资源消耗。其次,为优化数据存储与传输,提出了一种循环分块策略,以减少特征数据的存储需求并降低带宽压力,同时通过双缓存机制,实现计算与数据加载的高效流水并行,进一步提升系统吞吐量。最后,采用交错 rANS 编码方法优化存储与带宽利用,通过紧凑的数据布局减少存储需求,并借助流水处理确保数据访问连续性,降低带宽开销,提高吞吐率,同时简化控制逻辑,降低实现复杂度。

1 相关技术介绍

1.1 图像语义信息压缩

早期神经网络技术在图像与视频数据高效传输与存储上,主要将传统混合编码框架的模块与神经网络结合来优化性能,但这种模块级别的改进提升有限且实现复杂[20]。随着技术发展,语义通信的兴起为图像编码开辟了新视角,更聚焦于深层语义信息,追求更精准高效的信息传递。

近年来,基于深度学习的端到端图像语义信息压缩技术受到了研究人员们的广泛关注[21-22]。这种算法通过构建包含编码器和解码器的完整模型,每个模块都可以通过学习的方式进行优化,以对图像数据进行深度分析和优化压缩。与传统方法相比,端到端模型不仅显著提高了压缩效率,更能保留图像中的关键语义信息,为后续的图像识别和分析提供有力支持。

本文采用基于分解先验(factorized prior)结构的图像语义信息压缩模型,并在 FPGA 上对编码器部分进行硬件设计加速,以满足边缘端对高并行和低功耗处理的需求。如图 1 所示,该模型结构灵活,参数 N 和 M 可表示输入通道数量,可根据实际需求调整。在训练阶段,可通过调整权重参数 λ 来控制压缩过程中图像质量损失和压缩比之间的平衡。这种灵活性和可控性使得该模型能够适用于多种语义通信场景,助力图像数据的实时传输与高效处理。

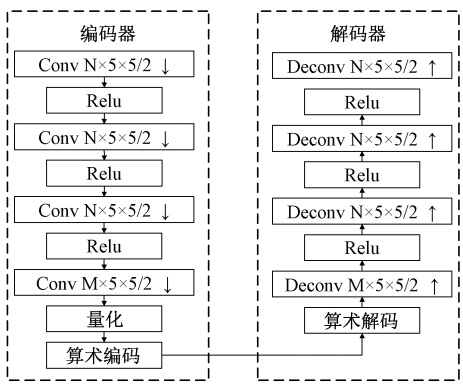

图 1 压缩模型结构图

Fig. 1 Compression model structure diagram

在编码器部分,模型包括四层卷积层,卷积核大小为 5×5,卷积步长为 2。每层卷积层之后(除了最后一层)均接入激活函数以增强模型非线性处理能力。输入图像历经四次卷积下采样 后,通过量化(采用四舍五入策略)和算术编码技术,被高效压缩为紧凑的比特流。

解码器部分则由四层转置卷积层(transposed

convolution)组成,卷积核大小为 5×5,卷积步长为 2。接收到的比特流首先通过算术解码恢复特征数据,随后经过四次转置卷积操作逐步还原图像细节,最终实现高质量的图像重构。

1.2　rANS 编码

rANS 凭借其高效的编码性能和压缩率,已广泛应用于图像语义信息压缩。本文采用 rANS 对卷积神经网络编码器的输出特征进行进一步压缩,以减少冗余特征数据。

给定一个有限符号表$\mathcal{A}$(对于 8 位图像,符号表大小为 256)和一个已知的符号概率分布 $\mathbb{P}(s)$,rANS 通过更新一个自然数(状态) x 来编码输入的符号序列。

如式(1)所示,rANS 通过编码函数将输入符号 s 和当前状态 x 映射到一个新的状态x':

$$\mathrm{Enc}(s,x)=\lfloor x/f_s\rfloor\cdot k+(x\ \mathrm{bmod}f_s)+b_s \tag{1}$$

式中:f_s 是符号 s 的频度。k 为一个大整数,负责对符号概率$\mathbb{P}(s)$ 进行量化,使得每个$\mathbb{P}(s)$ 能够通过 f_s/k 来近似。$b_s=f_0+f_1+\cdots+f_{s-1}$ 表示累加的符号频率偏移值,bmod 表示取模运算。

解码函数是编码函数的逆函数,需要从新状态 x' 检索出当前的符号 s 和先前状态 x,如式(2)所示。

$$\mathrm{Dec}(x')=f_s\cdot\lfloor x'/k\rfloor+(x'\ \mathrm{bmod}k)-b_s \tag{2}$$

在实际应用中,通常将符号的概率分布 $\mathbb{P}(s)$ 量化在 2^n 范围内,即符号概率近似为 $f_s/2^n$,其中 $\sum_{s\in\mathcal{A}}f_s=2^n$。经过这种概率分布的量化,可以简化编码函数和解码函数,即式(1)可以改写为式(3):

$$\mathrm{Enc}(s,x)=\lfloor x/f_s\rfloor<<n+(x\ \mathrm{bmod}f_s)+b'_s \tag{3}$$

式中:b'_s 表示量化后的b_s。式(2)可以改写为式(4):

$$\mathrm{Dec}(x')=f_s\cdot(x'>>n)+(x'\ \mathrm{bmod}2^n)-b'_s \tag{4}$$

随着符号不断被编码,状态 x 的值会逐渐增大,如果待编码的符号序列足够大,状态 x 可能会超过机器所能处理的整数范围,导致溢出错误。因此,当 x 超出给定范围时,需要通过正规化(renormalization)操作将其重新调整到指定范围内。假设给定状态范围是 $[2^L,2^{2L}-1]$,在编码过程中,如果 x 大于给定上界,则写出较低的 L 位,并将状态 x 右移一定的位数。同样的,在解码过程中,如果 x 小于给定下界,则将当前状态 x 左移一定的位数,并读入较低的 L 位。值得注意的是,编码和解码的正规化操作是对称的。

2　FPGA 硬件加速器设计

2.1　加速器整体框架

基于 FPGA 的图像语义信息处理加速器顶层架构如图 2 所示。加速器主要划分为控制器、片上存储模块、计算模块和 rANS 编码模块。控制器主要负责控制各个模块的运行调度以及运行过程中各个直接内存访问(direct memory access,DMA)的读写配置。偏置(bias)数据由于数据量较少,会一次性将所有层的数据预先写入到偏置缓存中。片上存储模块采用“乒-乓”双缓存策略,分为输入缓存、权重缓存和输出缓存,通过交替使用两个缓冲区实现高效的数据存取,一边进行数据处理的同时,另一边进行数据加载,提高数据传输效率,减少等待时间。计算模块包括计算阵列、激活和后处理,计算阵列采用脉动阵列架构,每个处理单元包含多个乘累加单元,执行高效的卷积运算。rANS 编码模块负责对特征数据进行进一步压缩,提高数据压缩效率和性能。

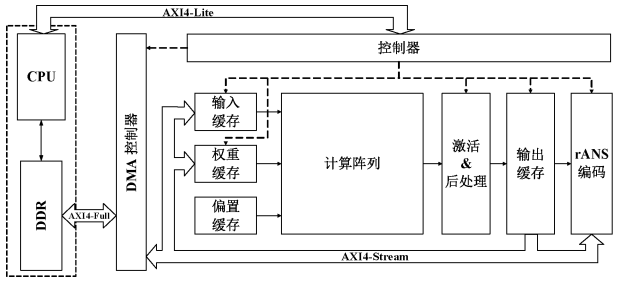

图 2　加速器整体架构

Fig. 2　Overall architecture of the accelerator

系统运行流程如下。CPU 通过 AXI4-Lite 总线向控制器发送启动信号,控制器随即初始化加速器并调度各个模块开始执行卷积操作。该过程涉及启动各个模块、参数配置,控制 DMA 从双倍速率同步动态随机存储器(double data rate synchronous dynamic random access memory,DDR SDRAM)(简称 DDR)读取权重与输入特征,以及将输出特征写回到 DDR。必要时,控制器还需控制部分模块的暂停,以确保数据对齐和避免数据覆盖。计算阵列负责完成卷积操作,并将卷积结果传递到下游进行激活和后处理。本文设计的加速器架构采用 INT8 计算,激活后的数据需进行 INT8 量化。为了 rANS 编码获取当前输入符号序列的概率分布,在最后一层卷积层执行完 INT8 量化后还需要进行 INT8 反量化,然后进行量化操作。最后一层卷积的输出特征在传输给 DMA 的同时,也输入到 rANS 编码模块以同步构建概率分布表。所有卷积层处理完后,控制器启动 rANS 编码模块执行编码操作,编码完成后发送结束信号给 CPU,并重新初始化加速器。

2.2　循环分块策略

FPGA 的片上存储资源有限,常规的通道优先循环分块策略需要存储整个通道的卷积计算中间结果,对于高分辨率特征图,这一要求可能会超出片上存储的容量。本设计提出了一种改进的循环分块数据复用策略,以更有效地利用片上存储资源并提高数据传输效率。

如图 3 所示,对于一个通道为 C、高为 H、宽为 W 的输入特征图,在高度上划分为若干个 Slice,每个 Slice 之间存在高度为卷积步长的重叠区域。在每个 Slice 内部,进一步在通道上均匀划分为若干个通道为 C' 的 Block。这种划分策略,能够显著减少单个 Block 的片上存储需求,而且每个 Block 存储在连续的地址空间内,DMA 能够高效地读取连

续地址的数据,提高数据传输效率和带宽利用率。在每个 Slice 中,通过遍历每个 Block 的计算方式,还能减少部分和的缓存需求。此外,每个 Block 在计算上可以高度复用,与多组卷积核进行卷积操作,提高了输入特征图和权重数据的复用率。对于高分辨率特征图,这种划分策略能够有效地解决存储瓶颈问题,同时确保计算的高效性。

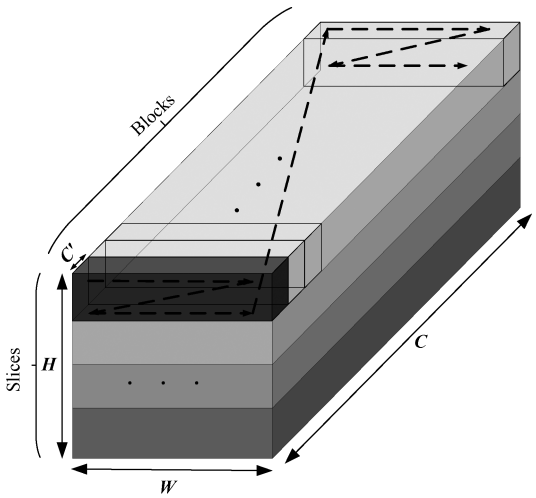

图 3 输入特征图分块

Fig. 3 Partitioning of input feature maps

2.3 计算阵列设计

卷积运算计算量大且数据传输频繁,对硬件资源和带宽提出了较高的要求。本设计的计算阵列采用脉动阵列计算架构。如图 4 所示,计算阵列包含 16 个处理单元(processing element,PE),输入特征从左向右依次传递给各个 PE,权重从上到下依次传递,实现输入特征和权重的复用。

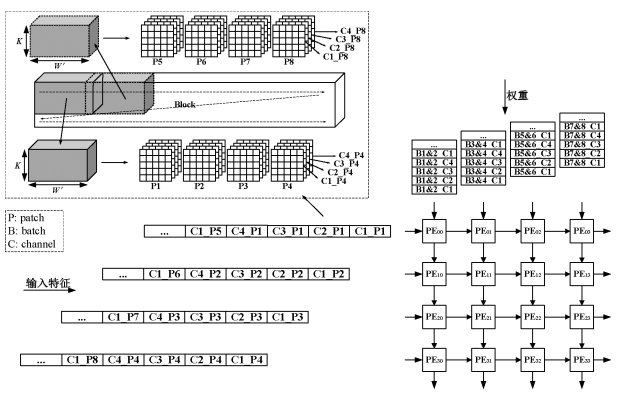

图 4 脉动阵列计算架构

Fig. 4 Systolic array computing architecture

为适配本设计的脉动阵列架构,每个 Block 包含 4 个通道的特征数据。对于每个 Block 的输入特征,首先依次提取维度为 $4\times K\times W'$ 的特征数据块 sub-block,相邻 sub-block 之间存在维度为 $4\times K\times(stride+1)$ 的重叠区域,然后对 sub-block 在 4 个通道上分别分割,构造出连续的 4 个卷积特征图(patch),这些特征图将按顺序传递到 PE 中进行处理。其中 K 为卷积核尺寸,W' 为构造卷积特征图所需的宽度,*stride* 为卷积步长。例如,对于 *stride* 为 2,K 为 5 的 sub-block 维度为 $4\times5\times11$。

计算阵列采用 4×4 脉动阵列计算架构。两组权重数据拼接在一起,从上到下依次传递,在 PE 内完成两组卷积操作。每个 patch 从左向右传递到计算阵列,累计与 8 个 batch 的权重数据完成卷积操作。例如,PE00 会在 4 级流水线中依次接收到输入特征 C1_P1(第 1 个通道第 1 个 patch)、C2_P1、C3_P1 和 C4_P1,权重数据 B1&2_C1(第 1 个 batch 和第 2 个 batch 的第 1 个通道两组数据)、B1&2_C2、B1&2_C3 和 B1&2_C4。该设计不仅能够提高计算单元的利用率,还减少数据传输的延迟,使得计算阵列在实现高效卷积计算的同时,充分利用 FPGA 的并行计算能力和片上存储资源,提升整体系统性能。

2.4 处理单元设计

卷积计算的核心是乘累加运算,其效率直接影响整体计算性能。本设计的 PE 包含 25 个并行的乘累加运算单元(multiply-accumulate unit, MAC),每个 MAC 在每个时钟周期可以完成两个 INT8 乘法运算[23]。MAC 流水计算结构如图 5 所示,核心处理单元为 FPGA 内部的 DSP 单元。每一级流水线中,两个通道相同但 batch 不同的权重 w_i 和单个输入特征 I_j 在 DSP 中完成两个 INT8 乘法操作,DSP 输出将会被分离出两个乘积项,并分别在累加器中与上一级结果完成累加操作。每 4 级流水线表示一个运算周期,每完成一个周期运算,累加器会输出累加结果(Y_1 和 Y_2)并进行清零操作,以避免下一周期执行累加时采用上一周期的累加数据。

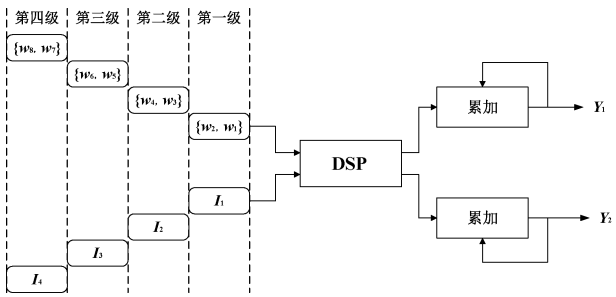

图 5 MAC 流水线计算结构

Fig. 5 MAC pipelined computing structure

PE 单元内部结构如图 6 所示。在 PE 中,每 4 级流水最多可输出 2×25 个中间结果。为了节省片上缓存资源,这两组中间结果会分别传递给加法树,与前一个 Block 的部分和(prev_psum)进行累加操作。如果当前 Block 是当前 Slice 的最后一个 Block,加法树的最终结果(cur_psum)将会直接输出给后续处理单元;否则 cur_psum 将会被存储在局部缓存中,以便后续快速访问。由于加法树计算所需的 prev_psum 和输出的 cur_psum 在索引上相对应,因此局部缓存不需要采用双缓存结构,从而简化设计并节省片

上缓存资源。

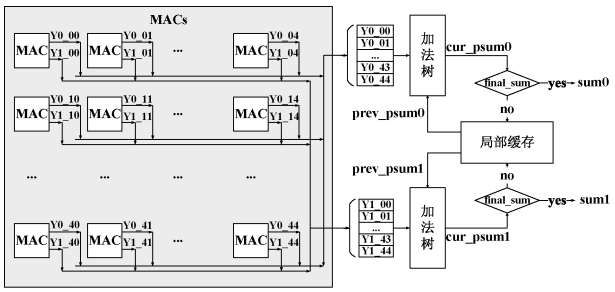

图 6　PE 单元计算结构

Fig. 6　Internal structure of the PE unit

2.5　rANS 编码设计

本文所实现的图像语义信息压缩的编码器，通过神经网络提取图像特征后，会进一步对特征进行 rANS 编码操作，以压缩数据量并减少特征冗余信息。如图 7 所示，rANS 编码主要分为构建概率表、初始化参数和编码 3 个部分。构建概率表需要获取当前图像最后一层卷积的输出特征(待编码符号序列)。为了减少数据在 DDR 之间传输产生的时延，最后一层卷积的输出特征会分为两路完全相同的数据流，一路接入 DMA 写回到 DDR，另一路传输到 rANS 编码模块构建概率表。构建完概率表后会进行初始化参数，包括初始化状态和对编码函数的运算参数进行转换，提高编码效率。

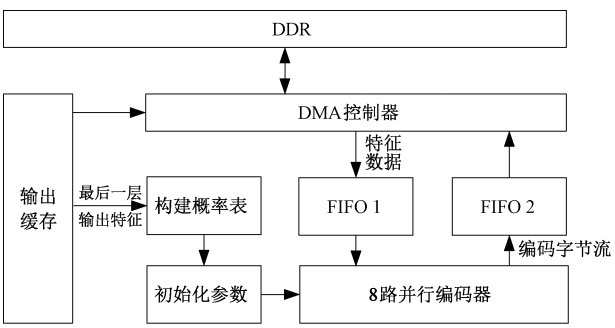

图 7　rANS 编码顶层结构

Fig. 7　rANS encoding top-level structure

编码流程如图 8 所示。连续的输入符号序列(8 bit 特征数据)首先缓存在 FIFO 1 中，以避免数据传输中的瓶颈和拥塞，确保编码数据流的稳定性。接着，从 FIFO 1 读取的 8 个符号并行传输到 8 个独立的编码器进行编码，每个编码器每 8 个时钟周期完成一次编码操作。为解决多路并行编码输出序列的离散性，本设计使用 FIFO 2 对编码输出进行缓存，转换为连续的编码字节，便于后续的数据传输。具体为，编码器输出的编码字节将通过有效信号 val 控制 FIFO 2 的写使能，决定当前字节是否需要写入。这种交错编码方式通过连续的读写操作减少了跨地址读写的频繁发生，从而节省带宽，降低资源消耗并简化设计复杂度。

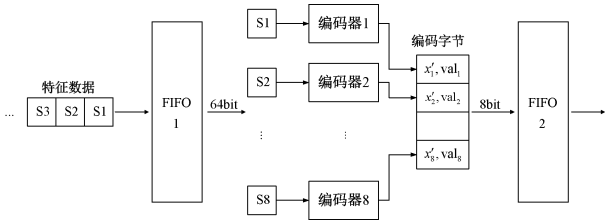

图 8　编码流程

Fig. 8　Encoding process

3　实验与结果分析

3.1　实验设置

本文实现基于分解先验结构的图像语义信息压缩模型的编码器部分，通道参数 N 和 M 均设定为 128。该模型可对 1080P(1 920×1 080×3)分辨率的输入图像进行编码压缩，整体模型的运算量为 149 GOP。为了进一步提升压缩效率，经过神经网络编码器提取特征后，本设计会对特征数据采用 8 路并行 rANS 编码进行进一步压缩。

本实验采用 Xilinx 的 ZYNQ UltraScale＋ MPSoC ZCU104 开发板，该开发板搭载 XCZU7EV-FFVC1156 型号芯片，包括大量的逻辑单元和可编程逻辑块，此外还配备了 2GB 在 PS 端的 DDR4 存储组件，以满足高带宽数据传输需求。加速器使用 Verilog 硬件描述语言进行编写，并通过 Vivado 2 018.3 设计工具进行仿真，以及对整体设计进行综合和布局布线。

3.2　实验结果

本文设计的加速器，其核心电路的工作频率为 200 MHz，在满足时序和通过布局布线的情况下，加速器资源消耗情况如表 1 所示。其中查找表(look-up table，LUT)、查找表型随机访问存储器(look-up table random access memory，LUTRAM)和触发器(flip flop，FF)的资源占用率分别为 37.56%、3.96%和 29.93%，主要用于逻辑控制和数据存储；随机存取内存块(block random access memory，BRAM)使用了 89.58%，主要用于缓存特征图、权重和概率表等数据，以及构建双缓存结构；DSP 共使用了 23.15%，用于组成计算阵列中的乘累加单元。

表 1　加速器资源占用率

Table 1　Accelerator resource usage

FPGA 资源	LUT	LUTRAM	FF	BRAM	DSP
可用	230 400	101 760	460 800	312	1 728
已用	86 544	4 029	137 934	279.5	400
占比/%	37.56	3.96	29.93	89.58	23.15

基于实际硬件测试，表 2 展示了本文加速器与 CPU 的处理时延和功耗对比结果。在 200 MHz 工作频率下，本文加速器的处理速度比 Intel I5-13400 提升约 6 倍，而功耗仅为其 1/14。在与 ARM Cortex-A57 功耗相近的情况下，本

文加速器的处理速度是 4 核 ARM Cortex-A57 的 58 倍,展现出显著的性能和能效优势。

表 2 与 CPU 的时延和功耗对比

Table 2 Comparison with CPU latency and power consumption

设备平台	频率/GHz	处理时延/s	功耗/W
Intel I5-13400	1.6	3.05	65
ARM Cortex-A57	1.43	28.88	5
本文 ZCU104	0.2	0.497	4.5

表 3 展示了本文加速器与其他 FPGA 加速器的对比结果。由于不同 FPGA 评估板在计算和存储资源方面存在差异,且加速器的时钟频率和数据精度也有所不同,因此本文采用结合相应资源的常用性能指标进行评估。其中,BRAM 效率(GOPS/BRAM)表示每个 BRAM 的平均吞吐量,反映存储资源的利用效率;能效比(GOPS/W)表示单位功耗下的平均吞吐量,用于衡量加速器的能耗性能;DSP 效率(GOPS/DSP)表示每个 DSP 单元的平均吞吐量,反映硬件资源的利用率。

表 3 加速器性能对比

Table 3 Accelerator performance comparison

加速器	平台	数据精度/bit	频率/MHz	BRAM	DSP	吞吐量(GOPS)	BRAM 效率(GOPS/BRAM)	能效比(GOPS/W)	DSP 效率(GOPS/DSP)
文献[24]	Ultra96 V2	8	250	248	242	31.5	0.13	7.40	0.13
文献[25]	VC709	8	180	528	920	406.1	0.77	41.61	0.44
文献[26]	VC707	16	100	288	528	189.03	0.66	64.87	0.321
本文	ZCU104	8	200	279.5	400	300.5	1.08	66.77	0.75

文献[24]采用通用矩阵乘法和脉动阵列架构,为了提高灵活性,在设计上牺牲了一定的性能。其 BRAM 效率远低于本文,表明其存储资源利用率较低,可能未针对特定计算模式进行优化,导致较多的冗余存储开销。DSP 效率也显著低于本文,虽然采用 8 bit 低位宽计算,但未能充分发挥 DSP 的并行计算能力,灵活性设计增加了数据复用的复杂度,影响了整体吞吐率。此外,其能效比仅为 7.40 GOPS/W,远低于本文的 66.77 GOPS/W,在单位功耗下,性能表现相对较差。文献[25]通过软硬件协同优化方法,在单个 DSP 上执行两个 INT8 乘法,取得了较高的吞吐率,在 BRAM 效率、能效比和 DSP 效率上均表现出色。然而,与本文相比,其 BRAM 效率仍然偏低,这可能是由于其存储架构未能充分优化数据访问模式,导致存储带宽未被高效利用,影响了数据的存取效率。文献[26]采用乘加树与脉动阵列结合的架构,在能效比方面与本文相近,但 BRAM 效率和 DSP 效率仍低于本文。主要原因在于其采用 16 bit 数据精度计算未能充分利用 DSP,且存储需求增大,未能充分优化存储资源分配。此外,文献[26]可能未充分利用卷积运算的计算特点,导致对片外存储的依赖性较高,增加了存储带宽的开销。

总体而言,与文献[24]、文献[25]和文献[26]中的设计相比,本文 BRAM 效率分别提升约 730%、40%和 63%,能耗比分别提升约 802%、60%和 3%,DSP 效率分别提升约 476%、70%和 133%。这些性能提升得益于加速器在计算架构设计、数据复用和 rANS 编码设计上的优化。通过脉动阵列与乘累加器结合的架构,每个时钟周期执行两个 INT8 乘法并局部累加,提升了 DSP 利用率和计算效率。采用循环分块策略和双缓存机制,减少了数据存储需求,降低了带宽压力,并实现了计算与数据加载的高效并行,进一步提升了系统吞吐能力。同时,结合交错 rANS 编码优化存储与带宽利用,紧凑的数据布局减少了存储需求,确保数据访问的连续性并降低带宽开销。这些优化使得加速器在处理大规模图像数据时展现出显著的优势。

4 结 论

本文设计并实现了一种基于 FPGA 的语义信息处理加速器,旨在满足边缘计算环境中对高分辨率图像进行实时处理的需求。该加速器集成了卷积神经网络编码器和 rANS 编码,采用融合乘累加器的脉动阵列计算架构和循环分块策略,充分利用 FPGA 的并行计算能力和片上存储资源。实验结果表明,本文加速器在处理 1 080 P 图像时达到了 300.5 GOPS 的高吞吐量,BRAM 效率、能效比和 DSP 效率分别达到了 1.08 GOPS/BRAM、66.77 GOPS/W 和 0.75 GOPS/DSP,显著提升了计算性能和能效。该设计验证了基于 FPGA 的语义信息处理加速器在实际应用中的有效性,并为实现高性能、低功耗的边缘计算提供了可行的解决方案。未来的研究将聚焦于进一步优化硬件架构,提升计算性能和资源利用效率,并扩展加速器的应用范围,支持更高分辨率图像处理及其他语义信息处理任务。

参考文献

[1] QIN ZH J, TAO X M, LU J H, et al. Semantic communications: Principles and challenges[J]. ArXiv preprint arXiv:2201.01389, 2021.

[2] HUANG D L, GAO F F, TAO X M, et al. Toward semantic communications: Deep learning-based image

semantic coding[J]. IEEE Journal on Selected Areas in Communications, 2022, 41(1): 55-71.

[3] 张振国,杨倩倩,贺诗波.基于深度学习的图像语义通信系统[J].中兴通讯技术,2023,29(2):54-61.
ZHANG ZH G, YANG Q Q, HE SH B. Deep learning-based image semantic communication system[J]. ZTE Communications, 2023, 29(2): 54-61.

[4] WALLACE G K. The JPEG still picture compression standard[J]. Communications of the ACM, 1991, 34(4): 30-44.

[5] TAUBMAN D S, MARCELLIN M W, RABBANI M. JPEG2000: Image compression fundamentals, standards and practice[J]. Journal of Electronic Imaging, 2002, 11(2): 286-287.

[6] BELLARD F. BPG image format[EB/OL]. 2017-01-30[2024-07-17]. http://bellard.org/bpg/.

[7] 侯保军,田金鹏,杨洁,等.基于多通道采样和注意力重构的图像压缩感知[J].电子测量技术,2022, 45(16): 102-108.
HOU B J, TIAN J P, YANG J, et al. Image compressive sensing based on multi-channel sampling and attention reconstruction[J]. Electronic Measurement Technology, 2022, 45(16): 102-108.

[8] 孙洋舟,严天峰,孙文灏,等.基于 Swin Transformer 的图像语义通信系统[J].电子测量技术,2024, 47(24):85-92.
SUN Y ZH, YAN T F, SUN W H, et al. Image semantic communication system based on swin transformer[J]. Electronic Measurement Technology, 2024, 47(24): 85-92.

[9] 彭宇,姬森展,于希明,等.语义分割网络的 FPGA 加速计算方法综述[J].仪器仪表学报,2021,42(9): 1-12.
PENG Y, JI S ZH, YU X M, et al. A review of FPGA-accelerated computing methods for semantic segmentation network[J]. Chinese Journal of Scientific Instrument, 2021, 42(9): 1-12.

[10] 杨统,肖昊.基于低成本 FPGA 的深度卷积神经网络加速器设计[J].电子测量技术,2024,47(10): 184-190.
YANG T, XIAO H. Design of deep convolutional neural network accelerator based on low-cost FPGA[J]. Electronic Measurement Technology, 2024, 47(10): 184-190.

[11] 张立国,黄文汉,金梅. FPGA 实现卷积神经网络加速器[J]. 高技术通讯,2023,33(10):1060-1067.
ZHANG L G, HUANG W H, JIN M. Implementation of a convolutional neural network on an FPGA[J]. High Technology Letters, 2023, 33(10): 1060-1067.

[12] 王帅帅,陈强,郭剑博,等.基于 Winograd 算法的高效神经网络加速器及 FPGA 实现[J].合肥工业大学学报(自然科学版),2023,46(12):1659-1665.
WANG SH SH, CHEN Q, GUO J B, et al. Design of high-efficiency convolutional neural network accelerator andimplementation of FPGA based on Winograd algorithm[J]. Journal of Hefei University of Technology (Natural Science), 2023, 46(12): 1659-1665.

[13] LAVIN A, GRAY S. Fast algorithms for convolutional neural networks[C]. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016: 4013-4021.

[14] 姜明飞,冯凤阳,冯赟,等.基于 YOLOv4-Tiny 的硬件加速系统的设计与实现[J].电脑知识与技术,2024, 20(10):11-14.
JIANG M F, FENG F Y, FENG Y, et al. Design and implementation of a hardware acceleration system based on YOLOv4-Tiny[J]. Computer Knowledge and Technology, 2024, 20(10): 11-14.

[15] DUDA J. Asymmetric numeral systems: entropy coding combining speed of huffman coding with compression rate of arithmetic coding[J]. ArXiv preprint arXiv:1311.2540, 2013.

[16] RIPPEL O, BOURDEV L. Real-time adaptive image compression[C]. International Conference on Machine Learning, PMLR, 2017: 2922-2930.

[17] HOOGEBOOM E, PETERS J, VAN D B R, et al. Integer discrete flows and lossless compression[C]. Advances in Neural Information Processing Systems, 2019.

[18] KANG N, QIU SH ZH, ZHANG SH F, et al. Pilc: Practical image lossless compression with an end-to-end gpu oriented neural framework[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022: 3739-3748.

[19] LIN F ZH, ARUNRUANGSIRILERT K, SUN H M, et al. Recoil: parallel rANS decoding with decoder-adaptive scalability[C]. 52nd International Conference on Parallel Prlcessing, 2023: 31-40.

[20] 陈积敏,林泽昊.基于端到端学习的图像编码研究及进展[J].激光与光电子学进展,2020,57(22):28-38.
CHEN J M, LIN Z H. End-to-end learning-based image compression: A review[J]. Laser & Optoelectronics Progress, 2020, 57(22): 28-38.

[21] BALLE J, LAPARRA V, SIMONCELLI E P. End-to-end optimized image compression[J]. ArXiv

preprint arXiv:1611.01704, 2016.

[22] BALLE J, MINNEN D, SINGH S, et al. Variational image compression with a scale hyperprior[J]. ArXiv preprint arXiv:1802.01436, 2018.

[23] 苗鑫,周欢欢,陆栋洵. 基于 ZCU102 DSP 的 CNN 卷积运算加速方法[J]. 自动化技术与应用,2022,41(12):64-67.
MIAO X, ZHOU H H, LU D X. Acceleration method for CNN convolution based on ZCU102 DSP [J]. Automation Technology and Application, 2022, 41(12): 64-67.

[24] ADIONO T, PUTRA A, SUTISNA N, et al. Low latency YOLOv3-tiny accelerator for low-cost FPGA using general matrix multiplication principle[J]. IEEE Access, 2021, 9: 141890-141913.

[25] 张雨豪,叶有时,彭宇,等. 一种基于 FPGA 的深度神经网络硬件加速器系统[J]. 空间控制技术与应用,2024,50(2):83-92.
ZHANG Y H, YE Y SH, PENG Y, et al. An FPGA-based hardware accelerator system for deep neural networks [J]. Aerospace Control and Application, 2024, 50(2): 83-92.

[26] 梅志伟,王维东. 基于 FPGA 的卷积神经网络加速模块设计[J]. 南京大学学报(自然科学),2020,56(4):581-590.
MEI ZH W, WANG W D. Design of convolutional neural network acceleration module based on FPGA[J]. Journal of Nanjing University (Natural Sciences), 2020, 56(4): 581-590.

作者简介

李俊锋,硕士研究生,主要研究方向为 AI 算法芯片设计、人工智能等。

E-mail: 445099619@qq.com

谭北海(通信作者),副教授,硕士生导师,主要研究方向为信号与信息处理、人工智能、AI 算法芯片设计等。

E-mail:bhtan@gdut.edu.cn

郑宇凡,硕士研究生,主要研究方向为 AI 算法芯片设计、人工智能等。

陈汉杰,硕士研究生,主要研究方向为 AI 算法芯片设计、人工智能等。

余荣,教授,博士生导师,主要研究方向为分布式系统与边缘计算。