基于ImageMake硬算子的研究实现

摘要 随着 CNN 在端侧推理场景中的普及，面向 FPGA SoC 的前处理硬件加速器（Pre-Processing Accelerator, PPA）成为降低系统延迟与功耗的关键部件。本文以上海复旦微电子集团股份有限公司发布的 ImageMake（v1.0 & v1.1）硬算子为研究对象，设计并实现了一套可扩展的 SystemVerilog Testbench。该 Testbench 具备 AXI-Lite 寄存器配置、跨时钟域启动同步、DDR 写事务监视、在线自检与自动比对等功能，可在 100 MHz GP 时钟与 200 MHz DDR 时钟域下完成 64 × 64 × 4 通道 16-bit 图像数据的完整验证。实验结果表明，仿真结果基本达到预期目标，为后续的进一步学习提供了很好的基础支撑。

关键字 像素预处理、padding、内存打包、AXI 写事务、嵌入式视觉

1介绍（Introduction）

在深度学习推理系统中，图像前处理（例如减均值、按通道缩放、padding、位宽转换与数据重排）是保证后续网络得到正确输入并以高效形式利用内存带宽的关键环节。传统上这些操作多由 CPU 或软件框架完成，但当系统需要更高吞吐或更低延迟时，单靠 CPU 的做法会成为瓶颈，因而出现了大量将前处理或部分前处理下放到专用硬件或加速器的研究（见对硬件加速器与系统性优化的综述和案例研究）。例如，有综述系统性讨论了在各种硬件平台上高效执行 DNN 的方法，而另一些工作则聚焦于在 FPGA 上对卷积类网络进行结构与数据流的优化以提升性能和资源利用率 [1][2]。

现代 FPGA 提供了丰富的片上资源（BRAM、分布式 RAM、DSP 单元等），这些资源非常适合实现并行、流水线式的数据预处理模块。许多研究和实现展示了如何利用这些硬件资源把常见算子（如乘加、量化与数据重排）实现为流水线或硬算子，从而减轻主处理器负担并提升整体能效和吞吐。关于 OpenCL/FPGA 实现和面向 FPGA 的设计优化，有多项工作给出了实践经验和工程化策略 [2,3,4]。

另一个重要的问题是内存与带宽的有效利用：在面向大规模图像或视频的 DNN 推理中，如何组织数据布局、分块传输与突发写入，直接影响 DDR 带宽利用率与能效。研究人员对在数据流设计中减少冗余传输、对齐内存访问以及如何与后端加速核（如矩阵乘/卷积阵列）衔接提出了多种优化方法，这些研究帮助指导了在硬件中实现高效写流与打包策略的设计 [5,6]。

在实际应用层面，经典的图像处理与神经网络输入预处理（例如像素减均值、尺度变换和位宽规范化）是很多视觉任务的常见步骤，这些任务在不同的网络与应用中被反复证明重要（例如大规模视觉分类与实时检测任务对输入格式与吞吐的要求不同）。因此，除了芯片级和架构级的优化，很多工作还比较了 FPGA 与 GPU 等平台在推理场景中的性能与能效以确定在边缘或数据中心部署时的折中策略 [7,8,9][10]。

在这样的背景下，ImageMake 硬算子被提出用于把图像预处理流水线硬件化：其将按通道的减均值与 scale、可配置 padding、位宽转换（8/16 bit）以及将多通道像素以 512‑bit 打包写回 PL DDR 的能力整合在一个模块中。通过寄存器接口（AXI‑Lite）提供灵活配置，ImageMake 可以适配不同分辨率、通道数与数据类型的输入，从而为后续 ICORE/PL 推理单元提供高带宽且对齐良好的数据流。

2 ImageMake 硬算子介绍

ImageMake 硬算子是上海复旦微电子集团股份有限公司专为其AI加速平台设计的一款高性能图像预处理硬核IP（Hard Operator）。该算子主要用于将来自PS DDR或摄像头输入源的图像数据搬运至PL DDR，并在数据传输过程中完成一系列关键的预处理操作，旨在提升神经网络前处理效率，减轻CPU负担，实现端到端的硬件加速。

ImageMake 硬算子将来自上游的多通道像素流（每通道 8/16 bit，可达 1~4 通道）做减均值与 scale 处理（Y = (X - mean) \* scale），做必要的 padding，然后以 ICORE 要求的 512‑bit 打包格式写入 PL DDR。

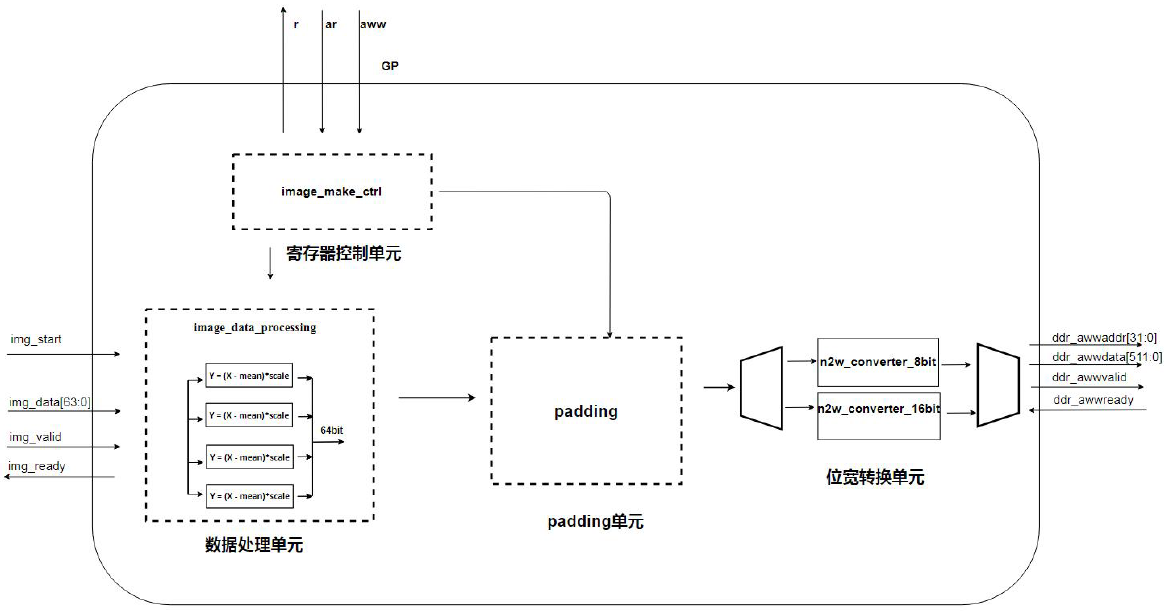


图1. ImageMake 模块结构框图

如图1所示，ImageMake采用模块化设计，主要由四个子模块构成：寄存器控制单元（Image\_make\_ctrl）：负责解析配置寄存器，设置操作参数。数据处理单元（Image\_data\_processing）：核心计算单元，对每个像素通道执行 Y = (X - mean) \* scale 的线性变换，其中mean（均值）和scale（缩放因子）可针对各通道独立配置。填充单元（Img\_padding）：支持对输入特征图进行灵活的边缘填充（Padding），四个方向（上、下、左、右）的填充宽度可独立设置（最大255像素），且各通道的填充值可不同。位宽转换与地址生成模块（n2w\_bwc\_addr）：负责将处理后的数据拼接并转换为后续计算单元（如ICORE）所需的硬件格式。它支持两种输出模式：8bit模式（将数据补至4通道后，16个32bit数据拼接为512bit）和16bit模式（8个64bit数据拼接为512bit），最终以高带宽（512bit @ 100MHz）写入PL DDR。

ImageMake具备高度的可配置性和灵活性，其功能通过一组丰富的存储器映射寄存器进行控制，包括图像尺寸、填充参数、各通道的mean/scale值、输出模式选择等。该算子在诸如YOLOv3、YOLOv5s、YOLOv7等常见神经网络模型上，对于416x416至1280x1280的不同输入尺寸，处理耗时仅在毫秒级别，显著提升了整体推理流程的效率。

综上所述，ImageMake硬算子通过硬件化常用的图像预处理流程，有效解决了软件预处理带来的性能瓶颈，是复旦微电子AI芯片平台中实现高效数据吞吐和低延迟推理的重要组成部分。

3. testbench 代码功能分析与结果

（增加过度语言，引出下面对各个TB代码的写的理由原因，即目的是要得到什么）

**3.1各功能模块**

（讲下TB代码中主要模块的作用）

**3.2仿真结果**

（讲下TB代码中主要的结果，根据picture文件夹中的图片来）

4结束语

（讲下通过该硬算子的TB实现，丰富了我的知识，基本达到了课程的最低要求，但是由于是初次接触，研究不够深入，因此对部分功能实现上没有完全实现。下一步进一步学习）

参考文献

[1] Sze, Vivienne,Chen,et al. Efficient Processing of Deep Neural Networks: A Tutorial and Survey[J]. Proceedings of the IEEE,2017,Vol.105(12): 2295-2329.

[2] Mittal, Sparsh (AUTHOR)( sparsh@iith.ac.in). A survey of FPGA-based accelerators for convolutional neural networks.[J]. Neural Computing & Applications,2020,Vol.32(4): 1109-1139.

[3] Naveen Suda,Vikas Chandra,Ganesh Dasika,et al. Throughput-Optimized OpenCL-based FPGA Accelerator for Large-Scale Convolutional Neural Networks[C]//FPGA '16: Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. 2016.

[4] Chen, TS (Chen, Tianshi),et al. DianNao: A Small-Footprint High-Throughput Accelerator for Ubiquitous Machine-Learning[J]. ACM SIGPLAN Notices: A Monthly Publication of the Special Interest Group on Programming Languages,2014,Vol.49(4): 269-284.

[5] Jouppi, N.P.,Young,et al. In-Datacenter Performance Analysis of a Tensor Processing Unit[J]. ACM SIGARCH Computer Architecture News,2017,Vol.45(2): 1-12.

[6] Masayuki Shimoda,Youki Sada,Hiroki Nakahara. FPGA-Based Inter-layer Pipelined Accelerators for Filter-Wise Weight-Balanced Sparse Fully Convolutional Networks with Overlapped Tiling[J]. Journal of Signal Processing Systems,2021,Vol.93(5): 499-512.

[7] Alex Krizhevsky,Ilya Sutskever,Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks[C]//NIPS'12: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. 2012.

[8] Joseph Redmon,Santosh Divvala,Ross Girshick,et al. You Only Look Once: Unified, Real-Time Object Detection[J]. 2016,.

[9] Eriko Nurvitadhi,Ganesh Venkatesh,Jaewoong Sim,et al. Can FPGAs Beat GPUs in Accelerating Next-Generation Deep Neural Networks?[C]//FPGA '17: Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. 2017.

[10] Andrew G. Howard,Menglong Zhu,Bo Chen,et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications[J]. 2017,.