基于 FPGA 的脉冲 Transformer 硬件高能效加速器实现*

邹涛[1)] 项水英[1)2)†] 卢小峰[1)] 黄志权[1)] 侯悦[1)] 郭星星[1)]
张雅慧[1)] 郑凌[3)] 潘伟涛[1)‡] 郝跃[2)]

1) (西安电子科技大学, 空天地一体化综合业务网全国重点实验室, 西安 710071)

2) (西安电子科技大学, 宽禁带半导体国家工程研究中心, 西安 710071)

3) (西安邮电大学通信与信息工程学院, 西安 710121)

(2026 年 1 月 16 日收到; 2026 年 2 月 6 日收到修改稿)

脉冲神经网络 (spiking neural networks, SNNs) 凭借低功耗、事件驱动和稀疏计算等特性, 在动态视觉等任务中展现出显著潜力, 但其算法优势在实际部署中仍受到传统计算架构的制约. 为突破事件驱动计算在能效与延迟上的硬件瓶颈, 本文针对 Spikformer 模型开展算法与硬件协同优化, 提出了一种基于现场可编程门阵列 (field-programmable gate array, FPGA) 的脉冲 Transformer 通用加速器架构. 算法层面, 通过卷积层与批归一化 (batch normalization, BN) 层融合以及量化感知训练, 将 Spikformer-1-384 模型参数规模由 15.92 MB 压缩至原来的 1/4, 并将精度损失控制在 1% 以内. 硬件层面, 基于 Verilog 设计了面向脉冲数据流的可配置加速器, 支持多时间步并行计算以及卷积、全连接、残差与注意力算子的灵活组合, 并提升并行度与存储带宽利用效率. 实验结果表明, 在 Xilinx Zynq UltraScale+MPSoC (xczu7ev-ffvc1156-2-i) 平台上, 该加速器在 CIFAR-10 数据集上时间步长 4 的端到端推理延迟约为 53 ms, 其中卷积特征提取与注意力模块计算时间分别为 48 ms 和 4.634 ms; 端到端系统功耗为 7.181 W, 对应能效达到 2.63 FPS/W, 整体性能与能效均优于 Intel i9 CPU; 对于自注意力机制和前馈神经网络 (multilayer perceptron, MLP) 计算, 较 GPU 和 CPU 分别加速 1.70× 和 5.73×. 本研究开源链接: https://github.com/tooddler/FPGA_SpikingTransformer.

关键词: 脉冲神经网络, 现场可编程门阵列, Transformer, 高能效加速器

DOI: 10.7498/aps.75.20260085 **CSTR:** 32037.14.aps.75.20260085

1 引 言

人工智能 (artificial intelligence, AI) 作为引领新一轮科技革命与产业变革的战略性技术, 近年来在模型结构与应用形态上均取得了突破性进展. Transformer 架构[1] 自提出以来, 已从最初的机器翻译任务迅速扩展至自然语言处理、计算机视觉、多模态理解以及具身智能等多个关键领域. 在自然语言处理方向, 以 BERT[2] 模型以及国产模型 DeepSeek[3]、文心一言[4]、通义千问[5] 为代表的大语言模型, 在文本生成、语义理解和逻辑推理等方面展现出卓越性能; 在计算机视觉领域, ViT (vision transformer)[6] 通过将图像划分为序列化图块并引入自注意力机制, 显著提升了对全局上下文信息的建模能力[7], 而 OpenAI 提出的 Sora 模型[8] 进一

* 国家重点研发计划 (批准号: 2021YFB2801900)、国家自然科学基金 (批准号: 62535015) 和中央高校基本科研业务费 (批准号: QTZX23041) 资助的课题.

† 通信作者. E-mail: syxiang@xidian.edu.cn

‡ 通信作者. E-mail: wtpan@mail.xidian.edu.cn

步展示了 Transformer 在跨模态视频生成任务中的强大潜力. 在自动驾驶与机器人控制场景中, BEVFormer[9] 和 RT-2[10] 等模型利用注意力机制融合多源感知与语义信息, 有效提升了系统在复杂环境下的感知与决策能力.

在视觉模型结构演进方面, 图像分类技术经历了由手工特征到卷积神经网络 (convolutional neural networks, CNN) 主导的范式迁移; 随后受 Transformer 在自然语言处理中成功经验启发, 研究者开始将其引入视觉领域并不断发展, 从部分替换卷积模块逐渐过渡到完全基于自注意力的视觉 Transformer. 除 ViT 外, Swin Transformer[11] 等层级式视觉模型通过滑动窗口注意力机制有效缓解了高分辨率视觉任务中的计算与建模困难 [12], 推动了 Transformer 在视觉领域的广泛落地. 然而, Transformer 架构性能不断提升的同时, 其模型规模和计算复杂度也持续膨胀. 以深度学习为代表的第二代人工神经网络 (artificial neural network, ANN) 高度依赖高精度数值计算和密集矩阵运算, 在训练与推理过程中需要频繁进行片外存储访问, 导致计算与存储之间的数据搬移能耗显著增大, 冯·诺依曼架构下的“存储墙”问题愈发突出. 这一矛盾在边缘端和嵌入式设备中更为尖锐: 有限的片上存储与带宽使得模型推理延迟与能耗难以同时兼顾, 从而制约了高性能视觉模型的低功耗部署与规模化应用.

为突破传统 ANN 在能效方面的限制, 研究者将目光转向更加接近生物神经系统的信息处理模型. 脉冲神经网络 (spiking neural network, SNN) 作为第 3 代神经网络模型 [13], 通过离散脉冲事件在时间维度上传递信息, 具备事件驱动、时序计算和低功耗等显著特性 [14]. 在数字电路实现中, 脉冲信号可利用单个比特进行表示, 其突触计算过程可由简单的逻辑运算或多路复用操作完成, 相较于 FP32, FP16 及 INT8 等传统数据类型, 在理论上具备更优的能效潜力. 随着训练方法和编码机制的不断发展, SNN 在静态与动态视觉任务中的性能差距逐渐缩小, 尤其在动态视觉传感器 (dynamic vision sensor, DVS)[15] 相关数据集上, 其对事件流输入的天然适配性使其在减少冗余计算、提升实时能效方面展现出独特优势.

在此基础上, 研究者进一步将 Transformer 架构与 SNN 相结合, 提出了以 Spikformer[16] 为代表的脉冲 Transformer 模型. 在 CIFAR[17], ImageNet[18] 和 DVS 等数据集上取得了兼顾精度与能效的优异表现. 在基于 ImageNet 数据集的图像分类实验中, 参数量为 16.81M 的 Spikformer-8-384 模型在从头训练条件下取得了 70.24% 的 top-1 准确率, 优于参数量达 60.19M 的 SEW-ResNet-152 模型 [19] (69.26%), 同时其计算量 (6.82 GSOPs) 和理论能耗 (0.525 mJ) 显著更低. 随着模型规模扩大, Spikformer-8-512 以 29.68M 参数量实现了 73.38% 的当前最高准确率, 表明性能随模型深度增加持续提升. 在 CIFAR 系列数据集中, Spikformer-4-384 在 CIFAR-10 上达到 95.19% 的准确率, 且扩展训练周期可进一步提升性能, 尤其在更复杂的 CIFAR-100 数据集上改进更为显著. 在动态视觉数据集上的实验进一步验证了其高效性: 在 DVS128 手势识别任务中, 参数量仅 2.59 M 的模型以 16 时间步长取得 98.2% 的准确率, 优于 SEW-ResNet (97.9%); 在 CIFAR10-DVS 上, 其以 10 步和 16 步二进制脉冲分别实现 78.9% 和 80.9% 的准确率, 较原有 SOTA 方法 DSR[20] (准确率 77.3%) 提升显著, 凸显了 Spikformer 在精度、能效与时序数据处理方面的综合优势.

尽管 Spikformer 在算法层面展现出显著潜力, 但其优势尚未在实际硬件平台上得到充分释放. 从系统实现角度来看, 现有主流 CPU 和 GPU 架构主要围绕密集数值计算进行优化, 难以高效支持 SNN 所固有的事件驱动与稀疏计算特性, 导致其在通用计算平台上的部署效率受限 [21]. 尤其对于多时间步推理过程, 若仍采用逐时间步循环执行, 不仅会引入较高的控制与同步开销, 也难以充分挖掘时间维度上的并行性; 与此同时, 脉冲数据表示与存储系统在访问粒度与数据组织方式上的不兼容, 会造成权重、膜电位等状态量在读写过程中产生额外的访存冗余, 从而降低有效带宽利用率并放大能耗与延迟. 近年来, 面向 Transformer 的 FPGA 加速研究不断涌现, 例如 Lu 等 [22]、Sun 等 [23] 和 Wang 等 [24] 分别从可重构架构、自动化部署框架和模型定制化加速等角度提升了视觉 Transformer 的推理效率. 然而, 这类工作主要面向传统 ANN 模型, 其数据流与计算模式以连续值和同步矩阵运算为主, 对脉冲数据表示、多时间步时序展开以及事件驱动计算机制缺乏针对性支持, 难以直接迁移至脉冲 Transformer 的高效部署场景.

在 SNN 专用硬件方向, Qi 等[25] 和 Li 等[26] 的研究在神经元模型灵活性和能效方面取得了重要进展, 但现有多数 SNN 加速器主要面向卷积或全连接结构优化, 通常采用单时间步循环执行方式, 时间维度并行性不足. 同时, 脉冲数据在表示形式与访存模式上具有不规则特征, 与传统存储系统的数据组织方式存在不匹配问题, 神经元状态与权重访问呈现动态触发特性, 容易导致带宽利用率下降与访存延迟增加.

当注意力机制被引入脉冲神经网络后, 模型中进一步包含 QKV 生成、大规模矩阵乘法及全局相关性建模等算子, 其数据依赖范围更广、访存模式更复杂, 与传统以局部连接和规则数据流为主的 SNN 加速架构存在范式差异. 此外, 部分现有 SNN 硬件设计对算子类型与网络结构支持较为固定, 扩展性有限, 难以灵活适配包含多头注意力与前馈大矩阵计算的脉冲 Transformer 模型.

因此, 现有 SNN 加速器在时间并行性、访存效率与算子通用性方面仍存在结构性约束, 难以同时兼顾事件驱动特性与注意力算子高吞吐计算需求. 基于上述背景, 本文以 Spikformer 模型为研究对象, 围绕 SNN 在硬件部署中面临的并行度不足、访存效率低及模型适配性受限等问题, 开展算法与硬件协同优化研究.

2 网络设计与轻量化处理

Spikformer 作为融合 Transformer 架构与脉冲神经网络的模型, 在结构与计算特性上天然适合 FPGA 等可重构硬件平台的部署. 其脉冲神经元及注意力计算可采用低比特甚至二值数据表示, 相较于传统 Transformer 中高精度乘加运算, 算子复杂度显著降低, 从而使大规模乘法运算可由逻辑运算与加法操作替代, 有效减少数字信号处理元件 (digital signal processing elements, DSP) 资源占用并降低功耗. 此外, Spikformer 在多类视觉任务中较传统卷积神经网络表现出更优的性能, 进一步体现了其作为通用视觉模型的潜力.

在网络结构设计上, Spikformer 采用批归一化 (batch normalization, BN) 而非层归一化 (layer normalization, LN), 该选择充分考虑了脉冲神经网络的事件驱动特性与训练稳定性. 相比 LN, BN 在 SNN 中更有利于稳定脉冲发放分布, 并在部署阶段可通过算子融合进一步简化计算流程[27,28]. 在模型量化方面, 本文采用量化感知训练 (quantization aware training, QAT) 策略. 与训练后量化 (post training quantization, PTQ) 相比, QAT 能在训练过程中显式建模量化误差, 从而在模型参数规模较大的情况下更有效地保持推理精度[29]. 同时, 由于 SNN 的输出天然为二值脉冲序列, 仅需对权重和神经元阈值进行量化, 而无需额外的激活量化步骤[30].

鉴于 Spikformer 模型参数规模处于数十兆量级, 本文在完成 BN 算子与前一层的卷积或全连接层进行融合后采用 QAT 进行量化, 其整体流程如图 1 所示.

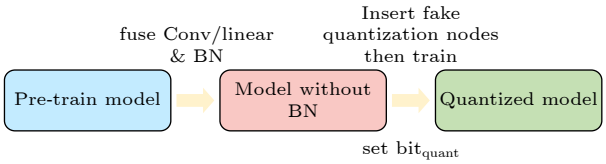


图 1 QAT 量化流程

Fig. 1. QAT quantitative process.

前向传播:

$$w_{\mathrm{o}} = \mathrm{round}\left(\mathrm{Scale}\cdot w_i\right). \tag{1}$$

首先需要找出权重绝对值当中的最大值记为 $w_{\max}$, 用于计算 Scale. 值得注意的是, 在 FPGA 中, 乘法运算比左移运算, 即乘以 2 的整数次幂需要消耗更多的资源和功耗, 因此选择牺牲掉一定精度, 选择将 Scale 设为 2 的整数次幂进行量化:

$$w_{\max} = \max\left(|w|\right), \tag{2}$$

$$S = \mathrm{floor}\left[\log_2\left(\frac{2^{k-1}}{w_{\max}}\right)\right], \tag{3}$$

$$\mathrm{Scale} = 2^S. \tag{4}$$

反向传播. 跳过了不可微的部分, 即 round 函数, 实现一个直通的梯度传递:

$$\frac{\partial c}{\partial r_{\mathrm{i}}} = \frac{\partial c}{\partial r_{\mathrm{o}}}, \tag{5}$$

其中以 Spikformer-1-384 为例, 具体结构如表 1. Patch Enbedding 模块如图 2(a) 所示. 对于 RGB 数据, 首先通过脉冲卷积编码, 得到脉冲数据, 此时通道数从原来的 3 变成了 embed_dim/8, 然后通过脉冲卷积模块, 将通道数逐步变为 embed_dim, 最后在通过一个输入与输出通道相同的卷积模块

表 1 Spikformer-1-384 网络结构

Table 1. Spikformer-1-384 network structure.

模块	层类型	参数配置	维度变换	说明
Patch emdebding	Conv2d+BN+LIF Maxpool	3→48→96→192→384 3×3 conv kernel	$(B, 3, h, w)$→$(B, 3, h/4, w/4)$	特征提取
Transformer block	Attention+MLP	num_heads = 12 hidden_dim = 1536	$(B, 384)$→$(B, 384)$	自注意力机制
Classification layer	Linear	384→10	$(B, 384)$→$(B, 10)$	输出分类结果

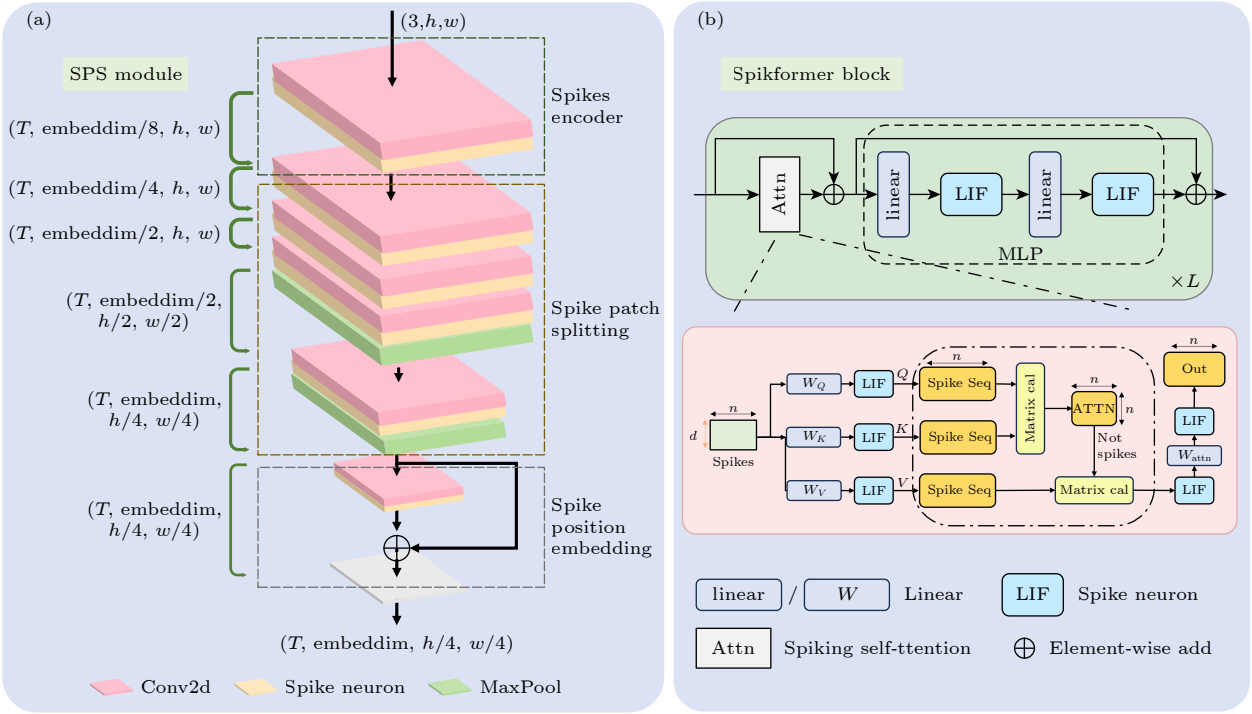


图 2 Spikformer 网络结构 (a) 特征提取卷积计算; (b) 脉冲自注意力机制计算

Fig. 2. Spikformer network structure: (a) Feature extraction convolution computation; (b) spiking self-attention mechanism computation.

作为脉冲位置嵌入. Transformer Block 具体如图 2(b) 所示, 脉冲特征提取 (spike patch splitting, SPS) 模块的输出再经过数据重组 (reshape) 后的数据作为输入. 该模块可以级联, 以提升网络的表达能力. 脉冲自注意力机制将连续值特征映射为时间展开的脉冲序列, 通过线性变换与脉冲神经元非线性激活生成脉冲形式的 Q, K, V 表示, 以脉冲计数或时间维度刻画特征强度. 注意力权重由脉冲 Q 与 K 的相关运算直接得到, 并引入缩放因子抑制数值放大, 再与 V 分支加权融合. 最终结果经归一化与脉冲神经元变换输出, 实现从连续值注意力到脉冲注意力的计算范式转换.

采用 AdamW 优化器进行模型训练 (weight decay = 6×10^{-2}). 初始学习率设为 5×10^{-4}, 并采用 cosine 学习率调度策略逐步衰减至 1×10^{-5}, 前 20 个 epoch 使用 warmup 策略 ($lr_{\text{warmup}}=1\times10^{-5}$), 训练末期设置 10 个 epoch 的 cooldown 以增强收敛稳定性. 模型首先在 CIFAR-10 数据集上进行 300 轮全精度 (FP32) 训练, 以获得收敛稳定的基线模型及最优权重参数. 全精度训练完成后, 对网络结构进行部署友好的重参数化处理, 将 BN 吸收并与相邻卷积或线性层融合, 以消除推理阶段的 BN 计算开销并简化算子链路. 在此基础上采用量化感知训练 (QAT), 通过在训练图中插入伪量化节点 (fake quantization nodes) 模拟定点量化误差, 使模型在训练过程中自适应量化扰动. 量化阶段保持其余训练超参数不变, 将训练轮数设为 100 轮进行微调, 最终将权重参数量化为 INT8 表示, 以降低存储开销并提升硬件推理效率. 网络中的脉冲神经元采用带泄漏整合发放 (leaky integrate-and-fire, LIF) 模型.

图 3 表明随着时间步长 T 的增大, 神经元在时间维度上的积分窗口变长, 能够累积更多来自输入脉冲的时序信息, 从而提升特征表达的充分性并在

一定程度上改善分类精度, 尤其是在脉冲活动较为稀疏或输入信息分布较为复杂的场景中. 然而, 时间步的增加也会带来显著的计算与存储开销[31].

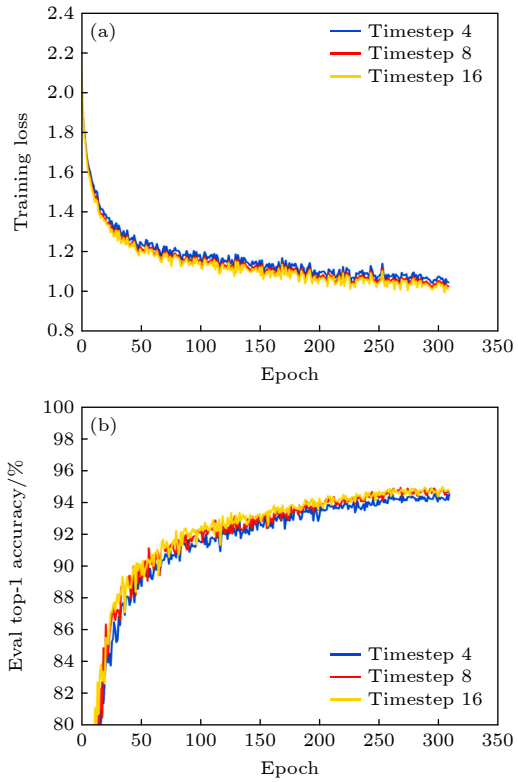


图 3 不同时间步长在 CIFAR10 数据集上表现 (a) 各时间步长训练过程 loss 曲线; (b) 各时间步长 top-1 准确率变化

Fig. 3. Performance on CIFAR10 dataset with different timesteps: (a) Train loss comparison; (b) evaluation top-1 accuracy comparison.

表 2 列出了不同时间步长对模型准确率的影响, 可知权重参数减少约 4 倍的情况下, QAT 后准确率下降值保证在 1% 以内.

表 2 Spikformer 量化前后在 CIFAR10 上的 top-1 准确率表现

Table 2. The top-1 accuracy performance of Spikformer before and after quantization on the CIFAR10 dataset.

Model	Param/MB	$T=4$ Acc	$T=8$ Acc	$T=16$ Acc
Spikformer-1-384-FP32	15.92	94.57%	94.93%	94.97%
Spikformer-1-384-INT8	3.98	94.35%	94.81%	94.75%

3 处理器硬件系统的设计与实现

3.1 加速器整体框架设计

基于 FPGA 硬件加速器设计的总体架构图如图 4 所示. 主机端通过以太网与 Zynq MPSoC 平台建立通信, 利用处理系统 (Processing System, PS) 端轻量级 TCP/IP 协议栈 (light weight IP, lwIP) 完成权重、偏置以及待推理图像数据的传输, 并通过 (direct memory access, DMA) 将其写入 (double data rate SDRAM, DDR) 的预分配地址空间. 随后, PS 端通过 AXI-Lite 对各专用加速器进行寄存器级配置 (包括基地址、数据尺寸、算子/层参数及控制状态等). 当主机下发触发信号后, 系统进入推理阶段: 各加速器按照预定的数据依赖关系与调度策略执行计算, 推理完成后由 PS 汇总中间/最终结果, 并通过串口将推理输出回传至主机端进行显示与记录.

在可编程逻辑 (programmable logic, PL) 端的计算模块划分中, 各硬件加速单元与 Spikformer 网络中的主要算法算子一一对应. 脉冲编码器 (spikes encoder accelerator) 面向帧式图像输入, 实现前端卷积编码与脉冲化表示生成, 对应算法中的输入编码与脉冲转换阶段, 为后续事件驱动计算提供统一脉冲数据接口; 脉冲卷积加速器 (spiking conv accelerator) 负责脉冲特征的卷积与局部特征提取, 对应网络中的卷积块与残差结构, 其算子序列由 Code Scheduler 模块按层级顺序调度执行, 以支持卷积、池化及残差加和等算子的可配置组合; 自注意力加速器 (self-attention accelerator) 对应 Transformer Block 中的脉冲自注意力与前馈网络 (MLP) 模块, 其中 QKV 矩阵乘法由专门的流水线脉冲矩阵乘法计算器计算, 其他都复用设置的脉动阵列进行分块并行计算.

在硬件实现中, 网络权重统一存储于 DDR 地址空间, 由 Weight_FIFO 按阈值触发突发读请求预取至片上 BRAM, 实现访存与计算流水重叠. 为降低地址生成与控制复杂度, 权重在软件端通过脚本进行离线重排, 使 DDR 访问模式尽量满足连续自增或局部循环的线性地址形式. 卷积层权重按原始 Conv2d 通道布局顺序存储并顺序突发加载, 单通道卷积完成后即可释放缓存. 针对 QKV 生成与 MLP 等大矩阵计算, 采用 3 组并行脉动阵列结构, 将权重按阵列划分映射至独立 DDR 地址区间, 并通过 reshape 与分块重排支持并行计算与顺序拼接, 权重存储方式如图 5 所示. 该权重组织策略使 Weight_FIFO 仅需在局部地址空间循环突发读取即可稳定供数, 在 FIFO 深度与突发长度协同设计下可有效减少访存反压与计算气泡.

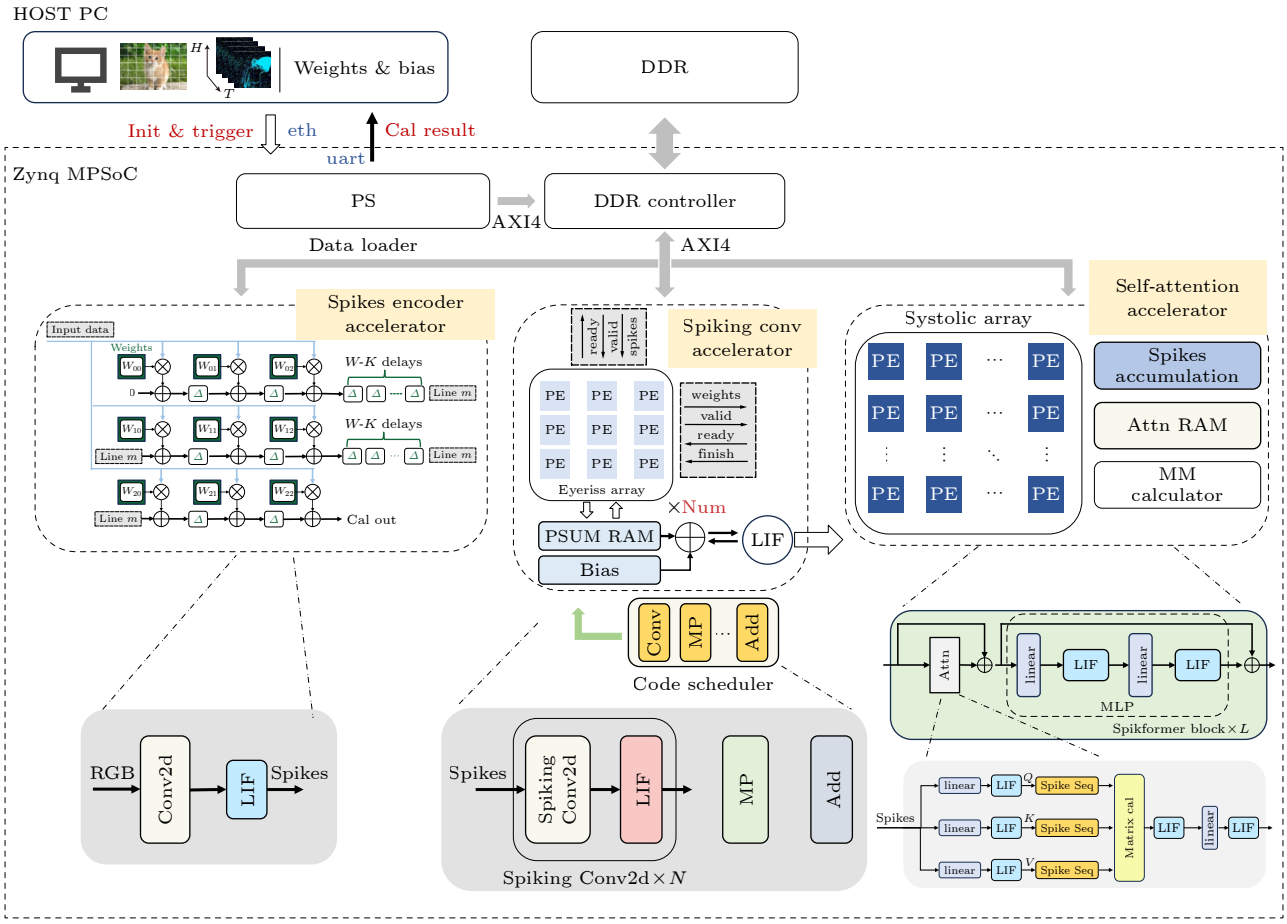


图 4 脉冲 Transformer 处理器硬件总体架构图

Fig. 4. Overall hardware architecture of the spiking transformer processor.

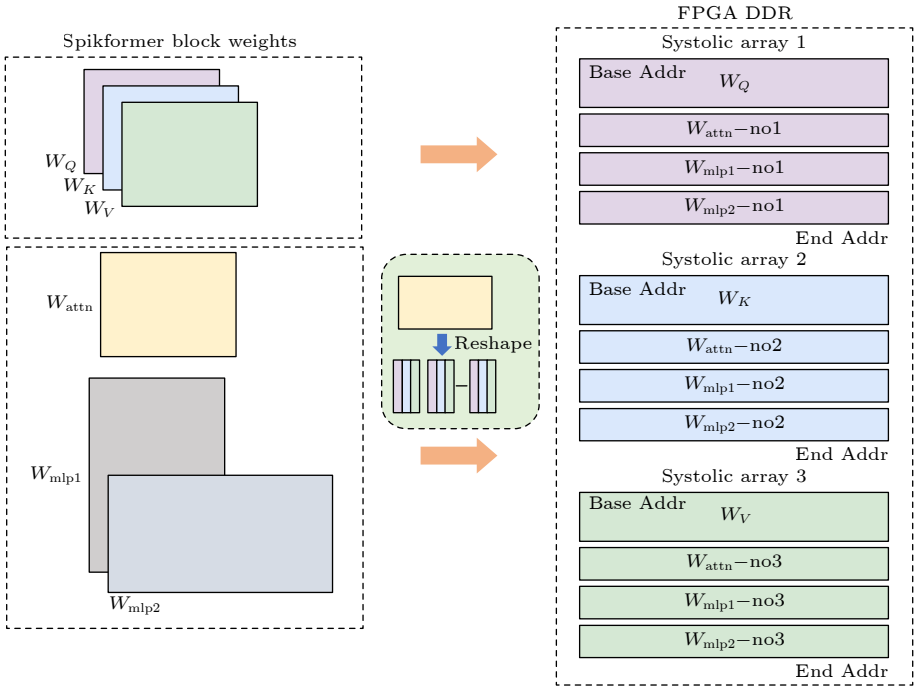


图 5 Spikformer block 权重在 DDR 中的映射关系

Fig. 5. Mapping relationship of spikformer block weights in DDR.

3.2 关键计算内核设计

3.2.1 脉冲神经元并行计算设计

考虑到脉冲数据量较少, 计算单元面积较小, DDR 的带宽利用不高, 且片上的各类资源利用不合理, 这里最大支持 SNN 中 4 个时间步长的并行计算. 神经元的兼容设计如图 6 所示. 4 个时间步长的数据同拍进入模块, 0 时刻的数据与前面存储的膜电位值进行非线性计算后的膜电位给到 1 时刻 (delay 1 clk), 同理类推. 在第 3 时刻输出的膜电位需要送回存储, 供下一次使用 (场景为 $T > 4$ 的脉冲数据推理).

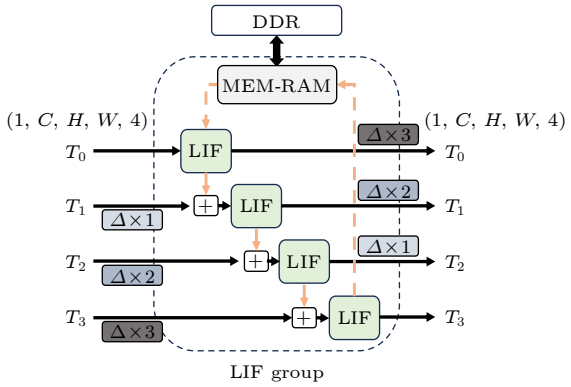


图 6 LIF 神经元结构图

Fig. 6. LIF neuron structure.

3.2.2 脉冲编码器

作为深度神经网络中的核心操作, 卷积在带来优异性能的同时, 也引入了沉重的计算与存储负担[32]. 首先, 卷积层通常包含数百万至数十亿个参数, 其存储与传输需要占用极高的内存带宽. 其次, 卷积运算需要在高维张量间进行滑窗或矩阵乘法, 这导致了极其庞大的操作数, 消耗了绝大部分的计算资源[33]. 因此, 卷积操作的效率已成为制约处理器性能的关键因素, 针对其进行优化是提升整体计算效率最直接的途径.

本节主流传感器获取的图像数据的存放方式都是 (H, W, 3), 以 RGB888 为例, DDR 的数据位宽为 64 bit, 只需要进行位宽转换即可让 3 个通道并行计算, 缩短数据重排的时间. 利用图 7 所示结构, 构造 3 个这样的计算核心, 能够以流水线计算得到卷积输出的某一层结果并写回 DDR. 图 7 将 3×3 卷积的权重系数直接映射到一个 3×3 的处理单元 (process element, PE) 阵列中, 使每个 PE 固定存储一个权重. 工作时, 输入数据被广播至所有 PE 进行并行乘加运算, 产生的部分和则在相邻 PE 间按预定时序传递并累加, 最终输出结果.

3.2.3 脉冲卷积加速器

本设计采用了在处理稀疏数据上表现更高效的类 Eyeriss 架构[34]. 通过行复用 (row-stationary, RS) 数据流最大化输入特征图、权重和部分和 (partial sums, Psum) 的多级复用, 从而大幅减少数据在片外与片内层级之间的传输量.

本设计的整体硬件框架在图 4 中体现, 其核心计算单元结构如图 8(a) 所示. 该计算核包含一个 3×3 的 PE 阵列, 专门用于计算权重向量 (w_0, w_1, w_2)

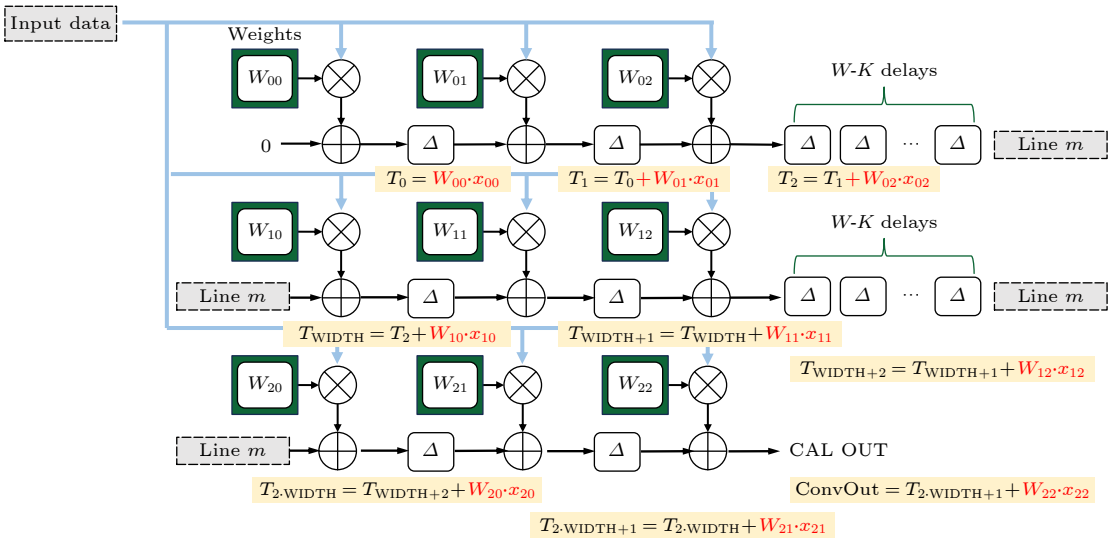


图 7 脉冲编码器加速模块

Fig. 7. Spikes encoder accelerator module.

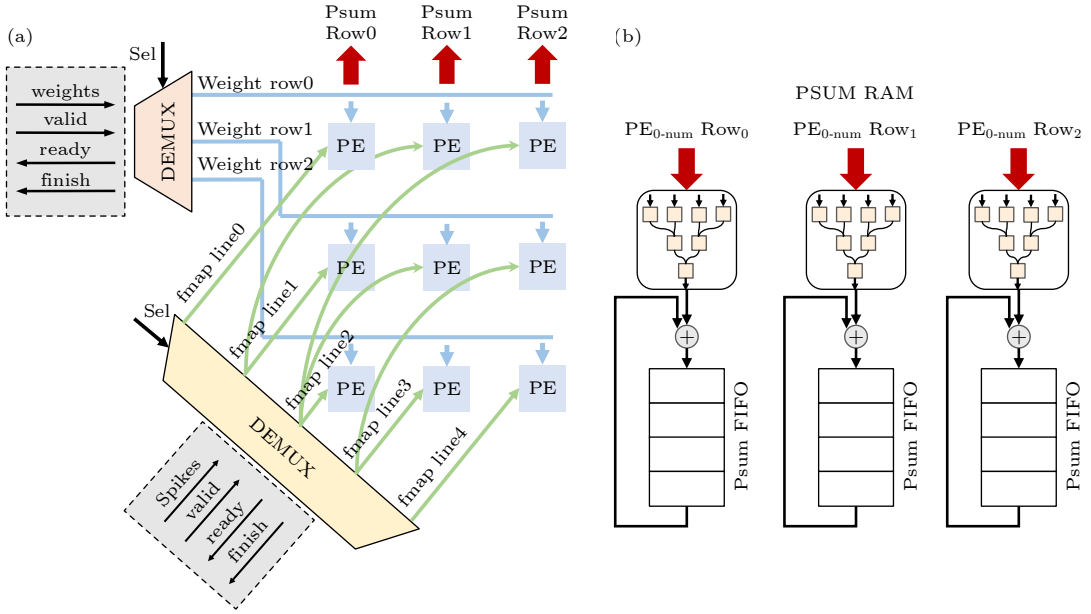


图 8 脉冲卷积加速器细节示意 (a) 脉冲卷积计算核心单元结构; (b) 部分和计算流水线架构

Fig. 8. Detailed schematic diagram of the spiking convolution accelerator: (a) Core unit structure for spiking convolution computation; (b) partial computational pipeline architecture.

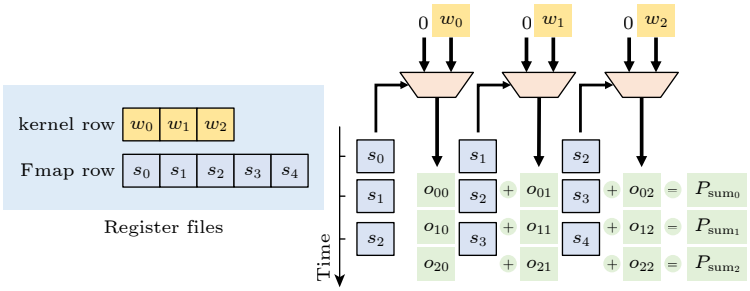


图 9 PE 中一维卷积原语的处理序列

Fig. 9. Processing sequence of one-dimensional convolution primitives in PE.

与脉冲输入向量 $(\text{spike}_0, \text{spike}_1, \cdots, \text{spike}_n)$ 的卷积, 如图 9 所示. 工作时, 权重通过 sel 信号选择广播至指定行; 脉冲数据则通过对应的 sel 信号以斜向 (对角线) 方式广播, 共广播 5 行. 如图 8(b) 阵列中, 每一列 3 个 PE 的计算结果先进行竖向累加, 其输出再与其他 PE 的结果经由加法树进一步汇总, 生成当前的部分和, 并暂存于 PsumRAM 中. 待该卷积核的所有计算完成后, 最终结果从外部读出, 即得到一个通道的输出.

3.2.4 自注意力机制加速器

自注意力机制的计算核心主要由大规模的矩阵乘法构成, 例如 Q, K, V 的生成, MLP 的计算等. 相比之下, 卷积神经网络中的权重在整个卷积层内是共享的, 并通过滑动窗口的方式在输入特征图上进行局部特征提取. 考虑到计算效率与硬件并行性, 矩阵运算通常采用脉动阵列 (systolic array) 这一专用硬件架构来实现. 在 Spikformer 中, 由于残差连接的原因, 导致了原本的单比特特征图数据发生变化, 需要 2 bit 数据进行表示, 计算核心如图 10 所示. 但 Q, K, V 之间的计算依然保留了脉冲的格式, 此处选择将 attention 的算子做融合,

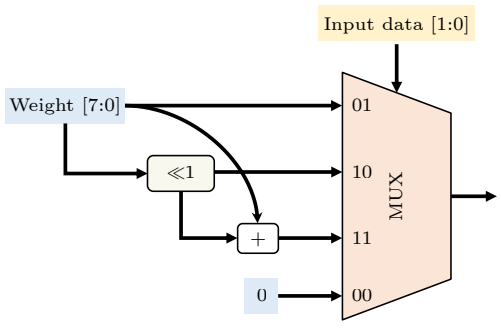


图 10 2 bit 计算核心示意图

Fig. 10. Schematic diagram of a 2 bit computing unit.

以片上 Block RAM 为缓冲, 乘法由逻辑与实现, 实现流水线计算.

图 11 展示了采用行固定 RS 数据流的脉动阵列结构. 为提升效率, 每个 PE 内设置了两份寄存器文件用于权重存储, 以此实现乒乓操作, 从而有效消除数据等待时间, 实现连续计算. 如图 12(a) 所示, 后续的 QKV 计算由于采用纯脉冲数据, 具备更为高效的处理方式. 在 QK 矩阵乘法的计算过程中, 通过按位逻辑运算与多周期分步累加相结合的策略, 高效生成注意力得分. 输入数据首先经过按位与操作, 其结果送入脉冲计数模块进行累加得到注意力权重值 (记为 Attn). 脉冲计数单元中, 借助全加器组合统计每组 8 bit 数据中“1”的个数, 如图 12(b) 所示.

对于 Attn 进行中间缓存, 借助中间两块 BRAM 存储 Attn 值进行乒乓, 当其中 Attn RAM 存储满一部分后, 会通知下游继续计算, 也会通过 ready 信号反压上级 QK 计算, 说明当前是否有空闲 RAM 存储. 如图 12(c) 所示, 下游计算中, V 矩阵为脉冲格式作为选路器的 select 信号, 输入从 Attn RAMs 中按顺序输出的 Attn 值, 输出整体矩阵计算结果.

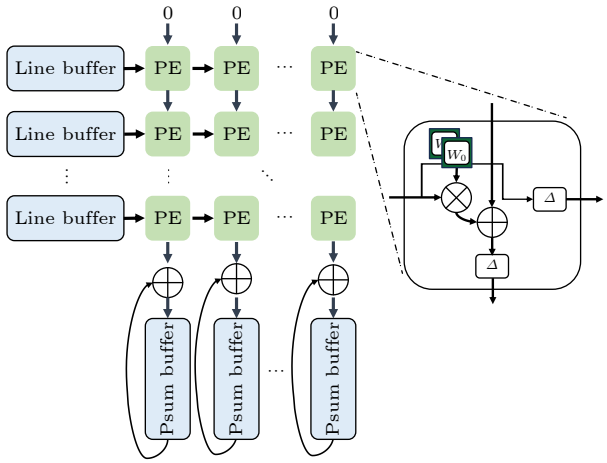


图 11 脉动阵列计算示意图

Fig. 11. Schematic diagram of pulsating array calculation.

4 数据分析与讨论

硬件加速器的卷积核总共开启 8 组计算核心, 完成 Spikformer-1-384 模型端到端的推理, 与其他 SNN 加速器的资源使用情况在表 3 列出. 加速器运行时钟为 200 MHz, 时间步长为 4 的数据集 CIFAR-10 在硬件上准确率保持在 94.35%, 脉冲卷积模块的数据加载带宽为 1.6 GB/s, 数据回写带宽为 2.0 GB/s. 对于模型 Spikformer-1-384, 卷积部分所需的运算操作数为 1.47×10^9 次脉冲的乘加计算, 推理时间用时 48 ms, 硬件利用率为 48.11%, 一个注意力机制块的计算时间为 4.634 ms, 在 CIFAR-10 数据集的图像推理时间仅为 53 ms. 本研究加速器的峰值吞吐率为 336.0 GSOP/s, 能效为 46.8 GSOP/W.

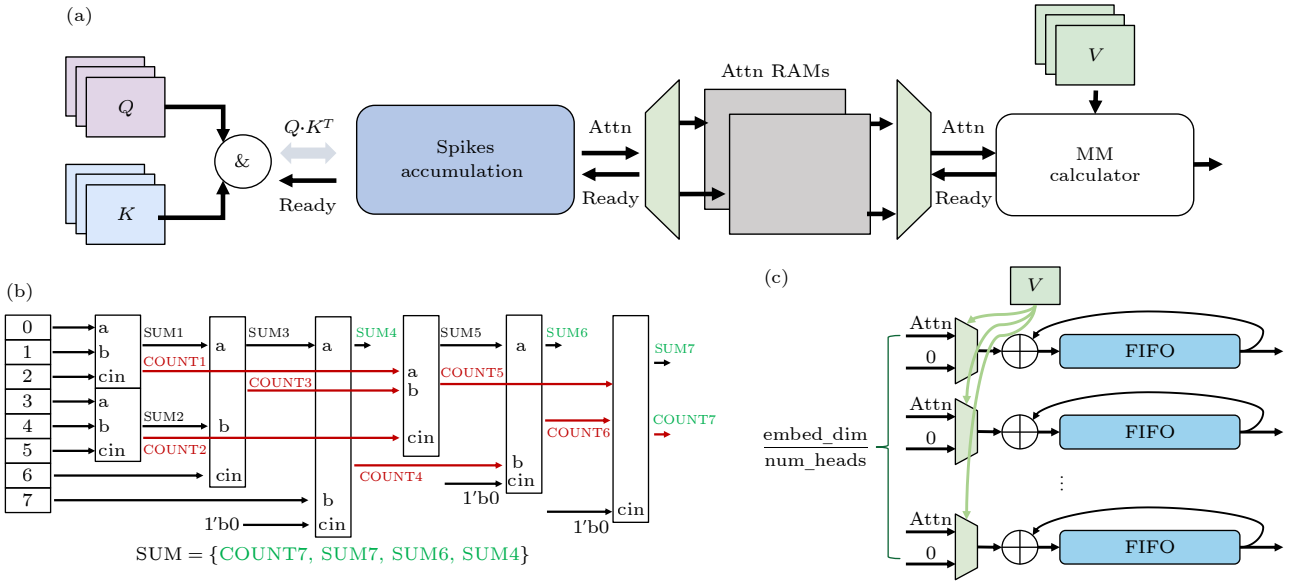


图 12 QKV 矩阵连乘计算流水线处理 (a) 脉冲 QKV 计算流水线; (b) 脉冲加速器实现细节; (c) 矩阵乘法计算器实现细节

Fig. 12. QKV matrix multiplication pipeline processing: (a) Spiking QKV computation pipeline; (b) implementation details of spikes accumulation component; (c) implementation details of matrix multiplication calculator component.

表 3 SNN 加速器资源使用情况对比结果

Table 3. Comparison with other SNN accelerators.

	ISCAS[35]	TCAD[36]	APACE[37]	Ours
网络结构	FC	CNN	ViT	Spikformer
数据集	MNIST	MNIST	CIFAR10	CIFAR10
平台	Kintex Ultra.	Zynq7000	Zynq Ultra.	Zynq Ultra.
频率(MHz)	140	200	200	200
LUTs/FFs	$4.162\times10^5/9.50\times10^4$	$4.59\times10^4/2.05\times10^4$	$4.532\times10^5/9.41\times10^4$	$1.439\times10^5/1.791\times10^5$
BRAM(36 k)	216	262	784	233.5
GSOP/s	179.0	22.6	307.2	336.0
GSOP/W	21.49	19.3	25.6	46.8

如表 4 所示, 在部署 Spikformer-1-384 模型时, GPU, CPU 和 FPGA 三种平台展现出截然不同的性能特性. 其中, GPU 的功耗测试采用官方提供的显卡管理工具 nvidia-smi, 在网络推理过程中抓取板载功耗值, 计算平均值得到; CPU 的功耗测试误差较大, 本文给出大致的范围; FPGA 的功耗为板载整体, 包含 PL 端的功耗 2.407 W, PS 端功耗 2.825 W, DDR 部分以及外部电路.

表 4 Spikformer-1-384 在各平台性能表现

Table 4. Spikformer-1-384 performance on various platforms.

平台	GPU	CPU	FPGA
型号	RTX4060	i9-13900	xczu7ev
工艺/nm	5	10	16
功耗/W	18	25—45	7.181
频率/GHz	2.2—5.4	1.5—2.4	0.200
精度	FP32	FP32	INT8
准确率/%	94.57	94.57	94.35
时延/ms	13.03	73.05	53.00
FPS	76.75	13.69	18.90
FPS/W	4.26	0.30—0.55	2.63

从推理延迟与吞吐率来看, GPU 平台虽然推理速度快, 但其端到端功耗达到 18 W, 能效受限. CPU 平台推理延迟较大, 且功耗高达 25—45 W, 能效最低 (0.30—0.55 FPS/W), 难以满足边缘应用需求. 相比之下, FPGA 平台在性能与能效之间取得了较优平衡: 在仅 200 MHz 运行频率下, 实现了 53.00 ms 的端到端推理延迟和 18.90 FPS 的吞吐率, 明显优于 CPU, 同时端到端功耗仅为 7.181 W, 对应能效达到 2.63 FPS/W, 较 CPU 提升 5—9 倍. 在精度方面, FPGA 采用 INT8 量化后在 CIFAR-10 上获得 94.35% 的准确率, 相比 GPU/CPU 的 FP32 实现仅下降 0.22%, 验证了量化与 BN 融合策略的有效性.

表 5 对比了 GPU, CPU 与 FPGA 在端到端推理及关键算子核上的时延表现. 对于输入尺寸为 $3\times32\times32$ 的端到端推理, 随着时间步 T 由 4 增至 16, 三平台时延均近似线性增长; GPU 始终保持最低时延, 而 CPU 增长最为显著. FPGA 在各 T 下均优于 CPU: 在 $T=4/8/16$ 时端到端时延分别降低约 27%, 24% 和 9%, 体现了所提架构对多时间步推理的可扩展性.

表 5 不同场景各平台时延对比

Table 5. Latency comparison across different platforms in different scenarios.

场景	平台	延迟/ms	FPS
$T=4$ Input_size = (3, 32, 32)	GPU	13.03	76.75
	CPU	73.05	13.69
	FPGA	53.00	18.87
$T=8$ Input_size = (3, 32, 32)	GPU	21.67	46.15
	CPU	132.04	7.57
	FPGA	100.70	9.93
$T=16$ Input_size = (3, 32, 32)	GPU	37.30	26.81
	CPU	216.39	4.62
	FPGA	196.93	5.08
Block(Attn+MLP) Input_size = (4, 1, 64, 384)	GPU	7.89	—
	CPU	26.58	—
	FPGA	**4.64**	—

进一步在算子层面, 对输入张量 4×1×64×384 的 Attn+MLP 计算, FPGA 仅需 4.64 ms, 较 GPU 和 CPU 分别加速 1.70×和 5.73×, 表明所设计的注意力与 MLP 硬件实现通过算子融合与数据流优化有效降低访存开销, 是 FPGA 获得系统级优势的重要原因.

5 结 论

本文针对 Spiking Transformer 在边缘部署中面临的能效与时延瓶颈, 提出并实现了一种基于 FPGA 的 Spikformer 硬件加速方案, 实现了算法与硬件架构的协同优化. 算法层面, 通过卷积与 BN 融合及量化感知训练将模型权重量化至 8 位定点, 在显著降低存储与计算复杂度的同时将精度损失控制在较小范围内. 硬件层面, 构建了面向脉冲数据流的可配置加速器体系, 支持多时间步并行的 LIF 神经元计算及注意力算子融合, 从而提升并行度并降低访存开销. 实验结果表明, 在 Xilinx Zynq UltraScale+ MPSoC 平台上, Spikformer-1-384 在 CIFAR-10 上实现约 53 ms 的端到端推理延迟和 7.181 W 的系统功耗, 对应能效为 2.63 FPS/W. 结果验证了所提出方案在保持精度基本不变的前提下, 显著提升了能效与实时性, 为脉冲 Transformer 在资源受限边缘设备上的高效部署提供了可行参考. 此外, 针对更大规模脉冲 Transformer 模型的部署需求, 本文架构在设计上保留了可扩展性空间, 但当模型规模进一步增大时, 片上计算阵列并行度与片外存储带宽可能成为系统性能的主要约束因素. 未来可通过多 FPGA 协同扩展、层级化数据复用与片上缓存优化策略, 以及更低比特宽度量化等方法进一步缓解资源与带宽压力. 同时, 结合脉冲神经网络的事件驱动特性, 引入稀疏计算与脉冲驱动调度机制, 有望进一步减少无效访存与冗余计算, 从而持续提升整体能效与系统吞吐率.

参考文献

[1] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L, Polosukhin I 2017 *Adv. Neural Inf. Process. Syst.* **30** 5998

[2] Devlin J, Chang M W, Lee K, Toutanova K 2019 *Proc. NAACL-HLT* **2019** 4171

[3] DeepSeek-AI 2024 arXiv: 2401.02954 [cs.CL]

[4] Sun Y, Wang S H, Li Y K, Feng S K, Chen X Y, Zhang H, Tian X, Zhu D X, Tian H, Wu H 2019 arXiv: 1904.09223 [cs.CL]

[5] Bai J Z, Bai S, Chu Y F, Cui Z Y, Dang K, Deng X D, Fan Y, Ge W B, Han Y, Huang F, et al. 2023 arXiv: 2309.16609 [cs.CL]

[6] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N 2021 arXiv: 2010.11929 [cs.CV]

[7] Chen G H, Yao J C, Zhu H F, Zhi T, Wang J, Xue J J, Chen L, Tao T, Tao Z K 2025 *Acta Phys. Sin.* **74** 190203 (in Chinese) [陈冠桦, 姚俊驰, 朱惠芳, 智婷, 汪金, 薛俊俊, 陈琳, 陶涛, 陶志阔 2025 物理学报 **74** 190203]

[8] OpenAI 2024 https://openai.com/index/video-generation-models-as-world-simulators/

[9] Li Z, Wang W, Li H, Xie E, Sima C, Lu T, Qiao Y, Dai J 2025 *IEEE Trans. Pattern Anal. Mach. Intell.* **47** 2020

[10] Ahn D, Brohan A, Brownlee J, Chebotar Y, Cortes O, David B, Finn C, Gopalakrishnan K, Hausman K, Herzog A, Ho D, Ichter B, Irpan A, Julian R, Kalashnikov D, Kuang Y, Lee K H, Levine S, Lu Y, Pastor P, Rao K, Sermanet P, Singh J, Xu C 2023 arXiv: 2307.15818 [cs.RO]

[11] Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B 2021 *Proc. IEEE/CVF Int. Conf. Comput. Vis.* **2021** 10012

[12] Xu C, Hao H Y, Wang Y, Ma Y H, Yan Q F, Chen B, Ma S D, Wang X G, Zhao Y T 2023 *J. Image Graph.* **28** 2927 (in Chinese) [许聪, 郝华颖, 王阳, 马煜辉, 阎岐峰, 陈浜, 马韶东, 王效贵, 赵一天 2023 中国图象图形学报 **28** 2927]

[13] Roy K, Jaiswal A, Panda P 2019 *Nature* **575** 607

[14] Wu C C, Zhou P J, Wang J J, Li G, Hu S G, Yu Q, Liu Y 2022 *Acta Phys. Sin.* **71** 148401 (in Chinese) [武长春, 周莆钧, 王俊杰, 李国, 胡绍刚, 于奇, 刘洋 2022 物理学报 **71** 148401]

[15] Gallego G, Delbruck T, Orchard G, Bartolozzi C, Taba B, Censi A, Leutenegger S, Davison A J, Conradt J, Daniilidis K, Scaramuzza D 2022 *IEEE Trans. Pattern Anal. Mach. Intell.* **44** 154

[16] Zhou Z, Zhu Y, He C, Wang Y, Yan S, Tian Y, Yuan L 2022 arXiv: 2209.15425 [cs.NE]

[17] *Learning Multiple Layers of Features from Tiny Images*, Krizhevsky A 2009 https://www.cs.toronto.edu/~kriz/conv-cifar10-aug2010.pdf

[18] Deng J, Dong W, Socher R, Li L J, Li K, Li F F 2009 *IEEE Conference on Computer Vision and Pattern Recognition* Miami, FL, USA, June 20–25, 2009 pp248–255

[19] Fang W, Yu Z, Chen Y, Masquelier T, Huang T, Tian Y 2021 *Adv. Neural Inf. Process. Syst.* **34** 21056

[20] Meng Q Y, Xiao M Q, Yan S, Wang Y S, Lin Z C, Luo Z Q 2022 arXiv: 2205.00459 [cs.CV]

[21] Yu S H, Yi M J, Wu Z, Shen F R, Zhao J 2025 *J. Softw.* **36** 1758 (in Chinese) [俞诗航, 易梦军, 吴洲, 申富饶, 赵健 2025 软件学报 **36** 1758]

[22] Ye S, Liang X, Yin S, Wei S 2023 *IEEE Trans. Circuits Syst. I* **70** 412

[23] Sun M, Li Z, Lu A, Ma H, Yuan G, Xie Y, Tang H, Li Y, Leeser M, Wang Z, Lin X, Fang Z 2022 *Proc. 59th ACM/IEEE Des. Autom. Conf.* **2022** 1394

[24] Wang T, Gong L, Wang C, Yang Y, Gao Y, Zhou X, Chen H 2022 *IEEE Trans. Comput. -Aided Des. Integr. Circuits Syst.* **41** 4088

[25] Qi X, Li X, Lou Y, Li Y, Wang G, Tang K T, Zhao J 2024 *IEEE J. Solid-State Circuits* **59** 3366

[26] Li J D, Shen G B, Zhao D C, Zhang Q, Zeng Y 2023 arXiv: 2301.01905 [cs.NE]

[27] Hu Y, Tang H, Pan G 2023 *IEEE Trans. Neural Netw. Learn. Syst.* **34** 5200

[28] Pfeiffer M, Pfeil T 2018 *Front. Neurosci.* **12** 774

[29] Hou Y, Xiang S Y, Zou T, Huang Z Q, Shi S X, Guo X X, Zhang Y H, Zheng L, Hao Y 2025 *Acta Phys. Sin.* **74** 148701 (in Chinese) [侯悦, 项水英, 邹涛, 黄志权, 石尚轩, 郭星星, 张雅慧, 郑凌, 郝跃 2025 物理学报 **74** 148701]

[30] Deng L, Wu Y, Hu Y, Liang L, Li G, Hu X, Ding Y, Li P, Xie Y 2023 *IEEE Trans. Neural Netw. Learn. Syst.* **34** 2791

[31] Wu Y, Deng L, Li G, Zhu J, Xie Y, Shi L 2019 *Proc. AAAI Conf. Artif. Intell.* **33** 1311

[32] Han S, Mao H, Dally W J 2016 arXiv: 1510.00149 [cs.CV]
[33] Krizhevsky A, Sutskever I, Hinton G E 2012 *Communications of the ACM* **60** 84
[34] Chen Y H, Krishna T, Emer J S, Sze V 2016 *Proc. Int. Symp. Comput. Archit.* **2016** 367
[35] Kuang Y S, Cui X X, Zou C L, Zhong Y, Dai Z H, Wang Z L 2022 *IEEE International Symposium on Circuits and Systems (ISCAS)* Austin, TX, USA, May 27–June 01, 2022 p3468
[36] Chen Q Y, Gao C, Fang X Y, Luan H T 2022 *IEEE Trans. Comput. Aided Des.* **41** 5732
[37] Li Z, Mao W, Zhang S, Dong Q, Wang Z 2024 *Proc. IEEE Asia-Pac. Conf. Appl. Electromagn.* **2024** 250

An FPGA-based high-energy-efficiency hardware accelerator for spiking transformer*

ZOU Tao [1)] XIANG Shuiying [1)2)†] LU Xiaofeng [1)] HUANG Zhiquan [1)]
HOU Yue [1)] GUO Xingxing [1)] ZHANG Yahui [1)] ZHENG Ling [3)]
PAN Weitao [1)‡] HAO Yue [2)]

1) (*State Key Laboratory of Integrated Service Networks*, *Xidian University*, *Xi'an* 710071, *China*)

2) (*State Key Discipline Laboratory of Wide Bandgap Semiconductor Technology*, *Xidian University*, *Xi'an* 710071, *China*)

3) (*School of Communication and Information Engineering*, *Xi'an University of Posts and Telecommunications*, *Xi'an* 710121, *China*)

(Received 16 January 2026; revised manuscript received 6 February 2026)

Abstract

Spiking neural networks (SNNs) feature event-driven processing, sparse activation, and low-bit data representation, and therefore exhibit strong potential for energy-efficient intelligent computing, especially for edge-side deployment. Recently proposed spiking Transformer models combine temporal spike dynamics with global attention mechanisms, but their practical deployment efficiency is still constrained by conventional computing architectures due to mismatched dataflow patterns, intensive memory access, and insufficient support for temporal parallelism. To address the latency and energy-efficiency bottlenecks in spiking Transformer inference, this work presents an algorithm–hardware co-designed FPGA accelerator targeting the Spikformer model. At the algorithm level, a deployment-oriented lightweight optimization strategy is adopted by fusing convolution and batch normalization (BN) layers and applying quantization-aware training (QAT). The model parameters are quantized to INT8 while preserving spike-driven characteristics, reducing storage and computation complexity. For the Spikformer-1-384 network, the parameter size is compressed from 15.92 MB to 3.98 MB with accuracy degradation controlled within 1%. At the hardware level, a configurable accelerator architecture tailored for spiking dataflow is designed on a field-programmable gate array (FPGA), consisting of spike encoding, spiking convolution, and self-attention-MLP compute engines with modular organization.

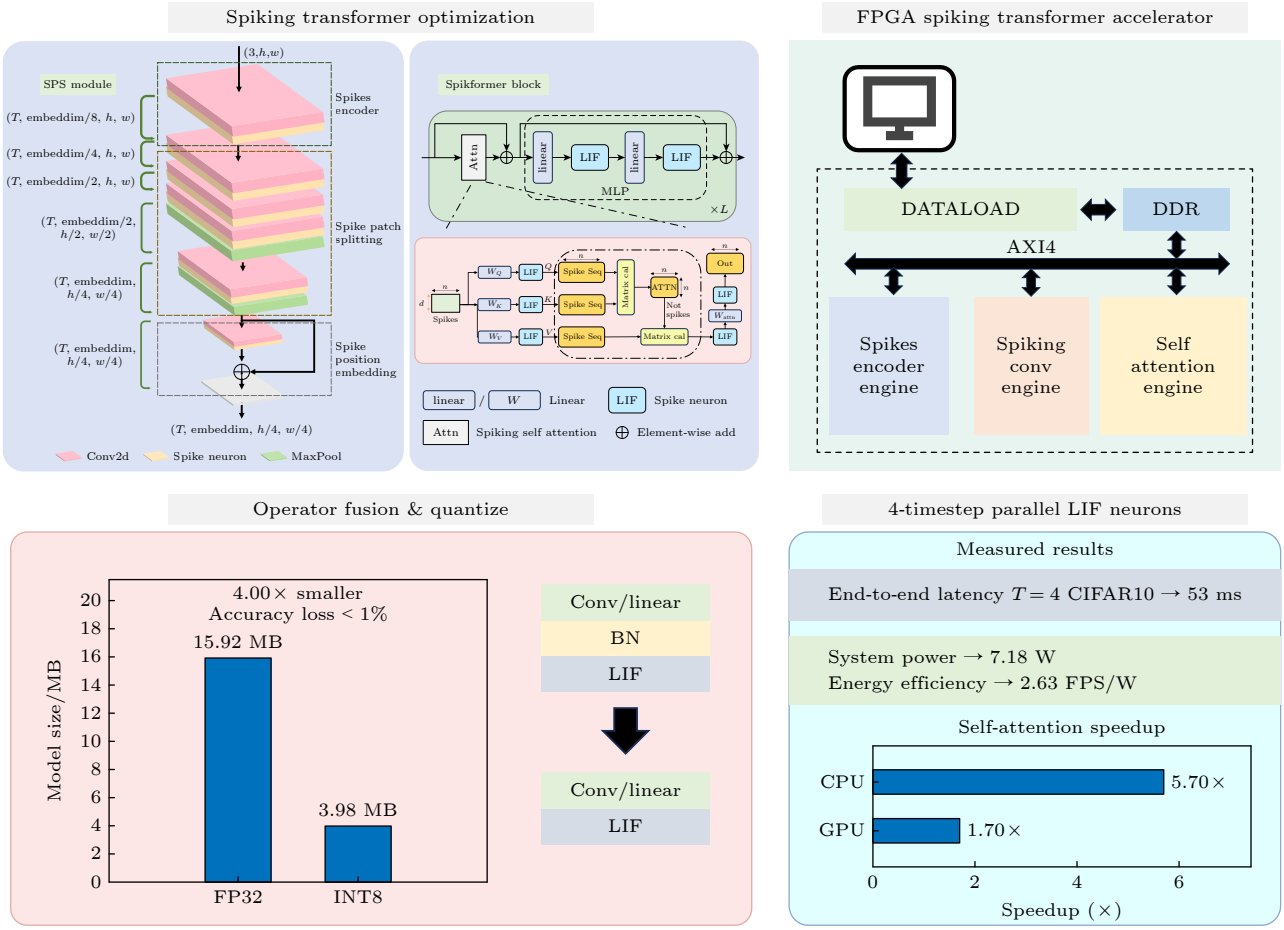
Multi-timestep parallel leaky integrate-and-fire (LIF) neuron processing is supported to exploit temporal parallelism, and operator fusion is applied to attention and feed-forward blocks to reduce intermediate off-chip memory traffic. In the attention path, spike-based matrix operations are implemented using bitwise logic and spike-count accumulation instead of conventional multipliers, significantly lowering DSP usage and improving compute density. A hierarchical memory and dataflow scheme combining DDR burst transfer, on-chip BRAM buffering, and ping-pong scheduling is further employed to enhance bandwidth utilization and pipeline

* Project supported by the National Key Research and Development Program of China (Grant No. 2021YFB2801900), the National Natural Science Foundation of China (Grant No. 62535015), and the Fundamental Research Funds for the Central Universities (Grant No. QTZX23041).

† Corresponding author. E-mail: syxiang@xidian.edu.cn

‡ Corresponding author. E-mail: wtpan@mail.xidian.edu.cn

continuity. The accelerator is implemented on a Xilinx Zynq UltraScale+ MPSoC platform and evaluated with the CIFAR-10 dataset. With four timesteps, the system achieves an end-to-end inference latency of 53 ms and a throughput of 18.9 FPS. The measured total power consumption is 7.181 W, corresponding to an energy efficiency of 2.63 FPS/W. For the attention and MLP block with input size (4, 1, 64, 384), the proposed design achieves 1.70× and 5.73× speedup over GPU and CPU implementations, respectively. The results demonstrate that the proposed co-optimized architecture provides an effective, scalable, and hardware-friendly solution for high-efficiency deployment of spiking Transformer models on resource-constrained edge platforms. The open-source link for this project is: https://github.com/tooddler/FPGA_SpikingTransformer.



Keywords: spiking neural network, field-programmable gate array, transformer, energy-efficient accelerator

DOI: 10.7498/aps.75.20260085 **CSTR:** 32037.14.aps.75.20260085

物理学报 Acta Physica Sinica
Chinese Physical Society
Institute of Physics, CAS

基于FPGA的脉冲Transformer硬件高能效加速器实现

邹涛 项水英 卢小峰 黄志权 侯悦 郭星星 张雅慧 郑凌 潘伟涛 郝跃

An FPGA-based high-energy-efficiency hardware accelerator for spiking transformer

ZOU Tao XIANG Shuiying LU Xiaofeng HUANG Zhiquan HOU Yue GUO Xingxing ZHANG Yahui ZHENG Ling PAN Weitao HAO Yue

引用信息 Citation: Acta Physica Sinica, 75, 100005 (2026) DOI: 10.7498/aps.75.20260085
CSTR: 32037.14.aps.75.20260085
在线阅读 View online: https://doi.org/10.7498/aps.75.20260085
当期内容 View table of contents: http://wulixb.iphy.ac.cn
