星载卷积神经网络 FPGA 加速器的软硬协同容错机制与优化

刘　晖[1,2]，周　海[1*]，卞春江[1]，成晓蕾[1,2]，王鹏飞[1]，刘一腾[1]

（1. 中国科学院国家空间科学中心 复杂航天系统电子信息技术重点实验室, 北京 100190 ; 2. 中国科学院大学 计算机科学与技术学院, 北京 100049）

摘　要：针对星载 CNN 推理面临的 FPGA 资源、存储带宽和功耗受限，以及辐射软错误易导致推理异常的问题，本文提出一种面向 FPGA 的分级加速架构，以兼顾推理性能与运行可靠性。在性能优化方面，根据网络层级敏感度采用 INT8/FP16 混合精度计算，并结合解耦访问/执行（DAE）调度机制，提高计算资源利用率和访存与计算并行度。在可靠性增强方面，针对数据通路和控制通路故障敏感性不同的特点，分别设计算法层异构双分支结构和硬件层反馈型三模冗余（TMR）控制逻辑加固机制，用于抑制中间特征扰动传播并降低关键控制状态异常风险。实验结果表明，在 XC7VX690T 平台部署 VGG-16 网络时，所设计加速器吞吐率达到 226.4 GOPS，推理时延为 136.52 ms，底层计算阵列平均有效执行效率达到 81.5%；在故障注入条件下，系统仍保持较好的分类准确率。该方法可为资源受限场景下星载边缘智能推理的高效可靠实现提供参考。

关键词：星载计算；卷积神经网络；FPGA 加速器；混合精度量化；软硬件协同容错

中图分类号：V557

Software–Hardware Co-Designed Fault-Tolerance and Optimization for a Spaceborne CNN FPGA Accelerator

LIU Hui[1,2], ZHOU Hai[1], BIAN Chunjiang[1], CHENG Xiaolei[1,2], WANG Pengfei[1], LIU Yiteng[1]*

(1. Key Laboratory of Electronics and Information Technology for Complex Space Systems, National Space Science Center, Chinese Academy of Sciences, Beijing 100190 , China ;2.University of Chinese Academy of Sciences , Beijing 100049 , China)

Abstract: To address the constraints of limited FPGA resources, storage bandwidth, and power consumption, as well as the inference instability caused by radiation-induced soft errors in spaceborne CNN inference, this paper proposes a hierarchical FPGA accelerator architecture to balance inference performance and operational reliability. For performance optimization, INT8/FP16 mixed-precision computation is configured according to layer-wise sensitivity, and a decoupled access/execute (DAE) scheduling mechanism is adopted to improve computational resource utilization and the overlap between memory access and computation. For reliability enhancement, considering the different fault sensitivities of the data path and the control path, an asymmetric dual-branch structure at the algorithm level and a feedback-based triple modular redundancy (TMR) mechanism at the hardware level are designed to suppress the propagation of intermediate feature disturbances and reduce the risk of abnormal critical control states. Experimental results show that, when deploying VGG-16 on the XC7VX690T platform, the designed accelerator achieves a throughput of 226.4 GOPS, an inference latency of 136.52 ms, and an average effective execution efficiency of 81.5% for the computing array. Under fault injection conditions, the system still maintains relatively high classification accuracy. The proposed method provides a reference implementation path for efficient and reliable spaceborne edge intelligence inference under resource-constrained conditions.

Keywords: on-board computing; convolutional neural networks; FPGA accelerator; mixed-precision quantization; software–hardware co-designed fault tolerance

0　引言

随着对地观测任务对空间分辨率和重访频次要求的不断提高，卫星平台遥感载荷产生的数据规模持续增长。受限于星地链路带宽、卫星过境窗口以及任务处理时延，传统的原始数据全量下传并由地面完成解译的处理模式，已难以满足高时效任务需求。为此，遥感数据处理正由地面后处理逐步向在轨智能处理转变，即在卫星平台端直接完成数据筛选与目标判别，以提升下传数

* **基金项目：**民用航天技术预先研究项目（D030312）
投稿日期：2026-03-09
E-mail: zhouhai@nssc.ac.cn

最新录用

据的有效性和任务响应能力[1-2]。在众多星载计算平台中，现场可编程门阵列（Field-Programmable Gate Array，FPGA）因兼具较高吞吐率、较低功耗和硬件可重构性，已成为卷积神经网络（Convolutional Neural Networks，CNN）星载推理加速的重要实现平台。

面向星载资源受限场景，现有研究主要从硬件架构优化和数值量化两个方面提升计算性能。在硬件架构方面，流式架构通过静态映射构建超长流水线，以提高推理吞吐率。Li 等提出了一种面向大规模卷积神经网络的 FPGA 高性能加速器，通过端到端流水化实现获得了 565.94 GOP/s 的吞吐率[3]。然而，此类架构在面对层间计算复杂度差异显著的深层网络时，容易因流水级间负载不均导致硬件单元闲置，难以在严苛星载约束下保持较高的算力利用效率。相较之下，单引擎架构通过时域复用提高了结构灵活性。Zhang 等提出的 Caffeine 框架将卷积计算映射为通用矩阵乘法并部署于脉动阵列，从而提升了带宽利用率[4]；Gokhale 等则通过指令调度进一步优化了计算能效[5]。在数值量化方面，训练后量化（Post-Training Quantization，PTQ）和混合精度技术已成为重要研究方向。Wang 等利用强化学习搜索最优位宽配置[6]，Yao 等通过 Hessian 矩阵评估层敏感度，以平衡模型精度与计算性能[7]。然而，现有混合精度方法大多聚焦于数值误差与任务精度之间的权衡，较少关注空间辐射环境下数值冗余削减可能带来的系统可靠性风险。

除计算性能外，抗辐射可靠性同样是星载环境下不可忽视的关键约束。近年来，已有研究开始针对星载神经网络加速器开展专门的容错设计，以应对空间辐射引发的软错误对推理过程的影响[8-9]。但现有方案普遍存在容错代价与防护收益不匹配的问题。传统全系统三模冗余（Triple Modular Redundancy，TMR）虽然能够提升系统可靠性，但同时会带来显著的功耗、时延和硬件资源开销，其综合代价可达基线方案的 4.5 倍[10]，难以满足星载平台对尺寸、重量和功耗的严格约束。另一方面，基于梯度敏感度的算法级加固方法虽具有较低实现成本，但通常默认底层调度逻辑始终正确。当强辐射环境导致控制通路发生单粒子翻转（Single Event Upset，SEU）时，算法层容错机制可能因控制失效而整体失效[11-12]。因此，当前星载推理加速器设计面临的关键问题在于：仅强调计算性能提升，往往忽视控制通路的可靠性保障；而单纯增强可靠性，又容易过度依赖系统级冗余，从而引入过高的资源代价。

针对上述问题，本文围绕星载 CNN 推理中的性能—可靠性协同优化问题，构建面向受限 FPGA 资源的软硬件协同加速框架。本文主要贡献如下：

（1）提出性能—可靠性协同优化框架，将混合精度计算、DAE 调度、算法层数据通路容错和硬件层控制通路加固统一到高效可靠推理目标下，明确各模块在性能提升与可靠性增强中的作用边界。

（2）构建分级容错机制，算法层通过异构双分支、置信度评估和动态融合抑制中间特征扰动，硬件层通过反馈型 TMR 加固关键控制状态机，降低控制通路单点故障风险。

（3）设计并实现寄存器传输级（Register Transfer Level，RTL）全定制 INT8/FP16 混合精度 FPGA 加速架构，通过混合精度 MAC、片上缓存与 DAE 调度提升资源利用和数据供给效率，并在 VGG-16 上完成性能与故障注入验证。

1 系统总体框架与分级容错机制

1.1 系统级协同框架

系统级协同框架如图 1 所示。该框架由离线准备与板级部署两个阶段组成。离线阶段主要执行算法级鲁棒性训练与 PTQ，利用层级敏感度分析指导网络在不同层间进行自适应精度配置。

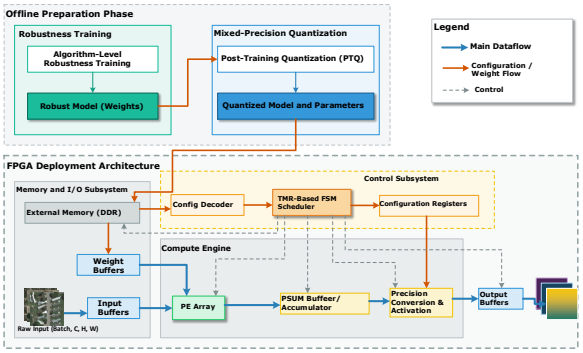


图 1　加速器系统级协同框架

Fig.1 System-level synergistic framework of the accelerator

在板级部署阶段，加速器依托 FPGA 实现全系统及计算引擎子系统协同运行。内存子系统通过 Xilinx DDR 控制器的原生接口（Native Port）对接片外存储器，并利用片上权重与输入缓存实现数据的高效数据重排与搬运；控制子系统以具备状态反馈的 TMR 状态机为核心，实现对各执行阶段的时序控制与错误修复；计算引擎子系统则

利用并行化混合精度乘加阵列完成密集的卷积推理运算。该架构通过区分数据通路与控制通路的可靠性需求，实现算法层软容错与硬件层硬加固的协同提升。同时基于 RTL 实现全硬件确定性调度，从而消除传统软件协议栈引发的时序波动。

1.2 软硬件协同分级容错策略

星载 CNN 加速器中，数据通路与控制通路对单粒子翻转（SEU）的敏感性存在明显差异。数据通路中的瞬态故障通常表现为激活值或中间结果的局部数值偏移，其影响可受到低位量化、饱和/截断运算以及激活函数非线性映射的共同抑制；同时，此类故障多在单次读写后即被后续正确数据覆盖，通常不易产生持续性的误差累积。相较之下，由有限状态机（FSM）等时序控制逻辑构成的控制通路对 SEU 更为敏感，单比特翻转即可导致控制状态异常或时序失配，且该类错误难以通过算法层机制加以屏蔽，易进一步引发系统停滞或功能失效。因此，本文采用面向数据通路的算法层软容错与面向控制通路的硬件层硬容错相结合的分级防护策略，以在受限星载环境下协同提升系统可靠性与实现效率。

1.2.1 算法层容错设计

算法层容错设计通过构建具有内生抑制能力的异构双分支特征提取架构实现。主干分支在执行常规语义提取的同时，利用卷积层后的批归一化层（batch normalization, BN）约束特征统计分布，以抑制 SEU 诱发的数值偏移向深层网络传播与累积。其映射逻辑描述为：

$$F_{\text{main}} = \phi_{\text{main}}(X;\theta_{main})$$

式中，X为输入样本张量，ϕ_{main}为主干特征提取算子，θ_{main}为对应的参数集合。

为规避共模故障，系统并行构建轻量化备份分支，采用大卷积核卷积与深层下采样策略拓展感受野，形成与主干异构的特征提取路径：

$$F_{\text{backup}} = \phi_{\text{backup}}(X;\theta_{backup})$$

式中，ϕ_{backup}为主干特征提取算子，θ_{backup}为对应的参数集合。

该非对称结构确保当主干分支受干扰导致特征畸变时，备份分支仍能维持稳定的语义表示。

系统利用轻量级错误检测器对特征状态进行实时评估，以量化主干分支的不可靠程度。在双分支特征经全局池化对齐维度后，该模块通过线性投影主干特征F_{main}捕获异常模式h_{det}，进而经非线性映射生成归一化的不可靠度评分α：

$$h_{\text{det}} = \text{RELU}(F_{\text{main}}W_{\text{det1}} + b_{\text{det1}})$$

$$\alpha = \sigma(h_{\text{det}}W_{\text{det2}} + b_{\text{det2}})$$

式中，$W_{det\,1}$、$W_{det\,1}$、$b_{det\,1}$、$b_{det\,2}$为可学习的投影参数，$RELU$为激活运算,σ为 Sigmoid 激活函数。

利用评分α作为门控系数，对主备分支特征进行动态加权组合，定义自适应融合特征F_{fused}为：

$$F_{fused} = (1-\alpha) \odot F_{\text{main}} + \alpha \odot F_{\text{backup}}$$

式中，$\odot$为逐元素相乘。

为保障备份分支在接管任务时具有一致的判别效能，训练阶段引入特征一致性约束。整体损失函数L由分类损失与一致性损失组成：

$$L = L_{\text{cls}}(y,\hat{y}) + \lambda * MSE(F_{main},F_{backup})$$

式中，L_{cls}为交叉熵分类损失，y为真实标签，$\hat{y}$为模型预测输出，用于保障模型的判别性能；MSE项约束主备分支在语义空间中保持一致；λ为平衡容错能力与结构差异性的调节系数。

1.2.2 硬件容错设计

控制逻辑缺乏 CNN 推理网络天然的数值冗余，单比特翻转即可引发状态跳转异常、地址控制错误或执行时序失配，因此是系统可靠性设计中的关键保护对象。本文针对控制器、有限状态机 FSM 以及地址/使能控制信号等关键控制路径实施硬件级 TMR 加固，以降低控制通路单点故障对推理流程的影响。

（1）控制通路可靠性建模

为明确模型适用范围，本文可靠性建模主要面向控制器、FSM 状态寄存器、地址控制信号和使能控制信号等关键控制节点，用于分析单帧 CNN 推理任务中控制通路状态异常的相对失效趋势，而非对整个 FPGA 器件、配置存储单元或数据通路进行绝对失效率预测。模型主要考虑单粒子翻转（Single Event Upset，SEU）引起的单点瞬态故障，并近似认为单帧推理时间窗口内故障到达较为稀疏、不同冗余分支之间单点失效相互独立。该假设未显式覆盖强空间相关多位翻转（Multiple Bit Upset，MBU）、共模扰动以及共享时钟、复位或电源异常等情况。

设单次传输事务中控制、地址与响应信号的综合等效失效率为 P_{err}，根据失效独立性近似原则[13]，其发生概率可估算为：

$$P_{err} = p_{ctrl} + p_{adr} + p_{rsp} = 1.0 \times 10^{-8} + 6.4 \times 10^{-9} + 2.0 \times 10^{-9} = 1.86 \times 10^{-8}$$

式中，p_{ctrl}、p_{adr}、p_{rsp} 分别代表单事务中控制、地址与响应信号的等效失效率。若未对控制逻辑采取硬件保护措施，则在给定等效故障率参数下，

一次推理任务中基于泊松分布模型的成功执行概率可近似表示为：

$$P_{success} \approx e^{-N_{crit} \cdot P_{err}} e^{-10^7 \times 1.86 \times 10^{-8}} \approx 83\%$$

式中，N_{crit}为单帧任务的关键控制周期数。上述结果表明，在本文设定的等效故障率参数下，未加固控制通路可能存在较高的状态异常和任务执行失效风险，因此有必要对关键控制路径实施硬件层加固。

需要说明的是，本文未开展器件级辐照测试，式中失效率参数主要用于相对趋势分析，不作为特定轨道环境下的绝对物理失效率。其中，单点等效翻转概率 p可由器件级 SEU 截面、轨道粒子通量、屏蔽条件和关键状态位数量进行标定，单帧任务平均故障到达率 λ可结合 p、关键控制周期数 N_{crit}及 RTL 运行周期统计进行估算；在本文实验中，p 和 λ 主要作为故障注入强度的等效表征，用于比较控制通路加固前后的相对失效趋势。

（2）反馈型三模冗余架构

如图 2 所示，本文对控制通路采用状态回传型 TMR 架构，各分支均配置独立的次态逻辑与状态寄存器。不同于传统单向表决结构，该设计通过闭环纠错链路实时重置受损位，实现了硬件级单周期自修复能力。

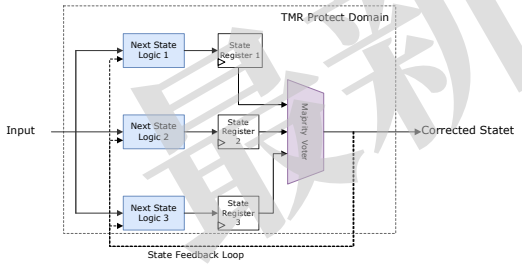


图 2　基于 TMR 的状态机（FSM）硬件架构

Fig.2　TMR-Based FSM Hardware Architecture

设 TMR 状态机的输出分别为S_i, S_j, S_k，多数表决器依据分支一致性执行分级响应逻辑：若三路全匹配（$S_i = S_j = S_k$），判定系统运行正常，输出标准状态值；若出现两路匹配（如$S_i = S_j \neq S_k$），判定发生 SEU，此时表决器输出占优状态S_i，并通过纠错链路在下一时钟周期将正确状态强制回注至受损分支寄存器，实现硬件级单周期自修复；若三路均不互相匹配（$S_i \neq S_j$，$S_i \neq S_k$，$S_j \neq S_k$），则判定发生 MBU 或共模故障，系统将触发报错信号并强制执行安全复位，以阻断错误控制状态继续向下游传播。

（3）容错效能评估

评估加固架构的有效性时，可将 TMR 容错过程抽象为不可恢复控制故障的检测与恢复问题。设单模块在单位周期内的等效失效概率为 p，状态机逻辑状态空间规模为 M，定义漏检率（False Negative Rate，FNR）为系统输出非预期非法状态而表决逻辑未预警的概率。在独立单点失效假设下，漏检主要发生在至少两个冗余分支同时失效且状态恰好一致的情况下：

$$P_{FNR} = \binom{3}{2} p^2(1-p)\frac{1}{M-1} + p^3 \frac{1}{(M-1)^2} \approx \frac{3p^2}{M}$$

由式可知，在状态空间较大且单分支等效失效概率较低时，TMR 表决结构可降低非法状态漏检风险。该结论依赖于分支失效近似独立、表决逻辑无共模扰动等前提；实际工程中，布局布线延迟、表决器毛刺以及公共时钟或复位扰动仍可能影响误检和漏检行为。

实施 TMR 加固后，系统不可恢复故障概率可采用经典三模失效模型近似描述。若单个冗余分支在单位周期内的等效失效概率为 p，则单位周期内 TMR 结构发生不可恢复故障的概率为：

$$P_{TMR} = 3p^2 - 2p^3$$

进一步地，单次推理任务中的等效失效概率可写为：

$$P_{fail} \approx 1 - e^{-N_{crit}(3p^2 - 2p^3)}$$

在本文设定的等效故障率参数下，模型估计结果表明，反馈型 TMR 能够降低控制通路单点失效导致的不可恢复风险。该分析主要用于说明硬件层 TMR 对关键控制路径的风险抑制趋势，而不作为真实轨道辐射环境下的绝对可靠性预测。

2　星载高效加速器架构的硬件实现

2.1　加速器系统架构设计

星载边缘计算任务对实时响应与确定性时序具有严格要求，加速器设计需尽量减少软件层参与带来的控制延迟。传统 CPU 软核协同模式依赖指令解析与中断响应，执行过程易受软件协议栈影响而产生时序波动，若在该架构中进一步引入三模冗余，则会极大增加设计复杂度。为此，本文采用全硬件控制架构，由专用控制逻辑直接驱动计算与访存模块运行，从底层消除软件调度开销。

在物理实现层面，系统基于 Verilog 进行 RTL 级全定制设计，避免了高级综合（high-level

最新录用

synthesis, HLS）在算子映射过程中易引入的逻辑冗余、设计可控性不足以及整体功耗—性能—面积（power, performance, area, PPA）优化受限等问题。该方案实现了对块存储器（block RAM, BRAM）与专用乘加单元（digital signal processor, DSP）的精细化映射与调度，从而提高资源利用率。

加速器系统架构如图 3 所示。系统采用解耦访问/执行（Decoupled Access/Execute，DAE）范式，将访存与卷积计算解耦，实现数据搬运与计算并行。整体架构由内存与 I/O、控制以及计算引擎三部分组成，并通过片上缓冲结构构建流水化数据通路。

其中，内存与 I/O 子系统负责片外 DDR 数据加载与结果回写，并利用多级缓存完成权重与特征图的数据重排与带宽适配，以降低访存延迟。控制子系统基于带纠错的 TMR 状态机，通过生成精确的时序使能信号与地址控制信号，实现对数据预取与调度、缓冲管理及阵列计算的全流程调度。计算引擎子系统由并行处理单元阵列（Processing Element Array, PE Array）构成，执行卷积乘加运算，卷积结果进一步经后处理流水线完成偏置叠加、激活、池化与结果打包。

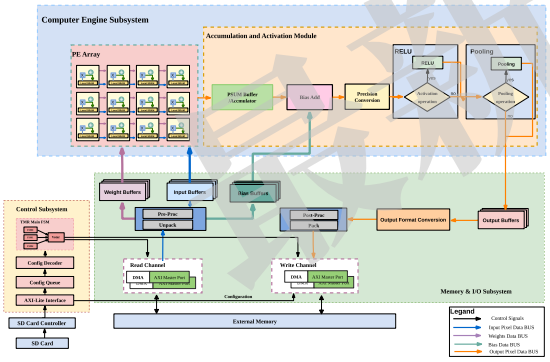


图 3 CNN 加速器系统总体架构

Fig.3 Overall architecture of the CNN accelerator

2.2 混合精度并行计算单元设计

2.2.1 混合精度配置策略

星载遥感图像具有复杂背景结构和较大的像素动态范围，不同网络层对量化扰动的敏感程度存在差异。为兼顾计算效率与识别精度，本文基于层级敏感度分析构建混合精度配置策略，对各层分配差异化计算精度。该策略对权重与激活值采用仿射均匀量化模型，其正向量化与反向重建过程分别表示为：

$$q = \mathrm{clip}\left(\left\lfloor \frac{x}{s} \right\rceil + z,\ q_{\min}, q_{\max}\right)$$

$$\hat{x} = s \cdot (q - z)$$

式中，x为全精度浮点输入，q为量化后的定点数值，$\hat{x}$为反量化后的重建浮点数值。s为尺度因子（scale），z为零点（zero-point），$\lfloor \cdot \rceil$表示四舍五入，clip(·)将量化值截断到整数动态范围$\lfloor q_{\min}, q_{\max} \rfloor$。

为定量评估不同网络层对量化扰动的容忍度，引入层级敏感度指标Δ_i度量：

$$\Delta_i = A_{fp} - A_i$$

式中，A_{fp}为全精度基线性能，A_i为仅对第i层实施量化后的系统性能。基于敏感度Δ_i与预设阈值τ，系统通过自适应决策机制确立各层级的精度配置策略p_i：

$$p_i = \begin{cases} FP16, \Delta_i > \tau \\ INT8, \Delta_i < \tau \end{cases}$$

如图 4 所示，对数值敏感的输入层和输出层采用 FP16 计算以保持特征表征能力，而数值鲁棒性较高的中间卷积层则采用 INT8 量化。量化参数在离线训练阶段确定，并在部署时写入配置寄存器，从而避免推理阶段进行动态量化参数更新。各层在执行前加载相应精度配置，控制底层乘累加（multiply–accumulate, MAC）单元在 INT8 与 FP16 模式之间进行层级粒度的计算模式切换。

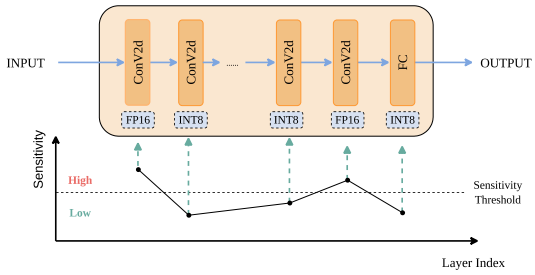


图 4 基于层级敏感度的混合精度配置示意图

Fig.4 Sensitivity-guided mixed-precision assignment

2.2.2 混合精度 MAC 实现

支持 INT8 与 FP16 的混合精度 MAC 单元作为卷积链路的核心计算基元，通过将异构精度的计算需求统一映射至同一物理结构，可减少硬件重复构建带来的面积开销。其微架构如图 5 所示，该单元向量化组织输入激活张量 X 与权重矩阵 W，并配置 40 bit 位宽的部分和缓冲(Partial Sum Buffer, PSUM Buffer)以吸纳累加过程的动态范围溢出。

录用稿件

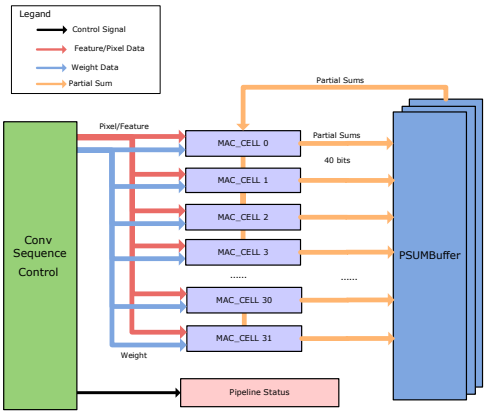


图 5 混合精度 MAC 架构

Fig. 5 Mixed-Precision MAC Architecture

该运算单元遵循非对称流水线执行原则：INT8 模式支持单周期乘加运算，而 FP16 模式则采用两周期流水线拆分策略。通过将浮点路径中的指数处理、对齐与规格化等高时延后处理开销分摊至两个时钟节拍，可有效降低关键路径压力。同时对 FP16 运算语义 FP16（IEEE standard 754）进行裁剪，通过省略 NaN 与 Inf 等特殊数值处理并采用截断策略，从而减少规格化移位网络规模。

在底层物理映射方面，通过状态机分时控制实现对 Xilinx DSP48E2 资源的复用。FP16 与 INT8 运算共享同一 16×16 乘法器及 5-bit CLA 加法器，并在 FP16 第二周期动态切换 DSP 内部级联路径。表 1 给出了不同实现方案的资源与性能对比。本文方案仅占用 1 个 DSP，相比异构双通路设计节省 2 个 DSP 并减少约 200 个 LUT，且 FP16 模式的有效计算效率优于 Xilinx 官方 IP。

表 1 不同 MAC 单元实现方案性能对比

Tab.1 Performance comparison of different MAC unit implementation schemes

指标	本文方案	异构双通路[14]	Xilinx FP16 IP
支持精度	INT8/FP16	INT8/FP16	FP16
计算周期（INT8/FP16）	1/2	1/1	3
DSP 占用	1	3	1
逻辑资源（LUT/FF）	258/92	458/-	170/77

2.3 硬件调度逻辑与并行访存优化

硬件调度器通过任务分解实现访存与计算的并行执行，其物理微架构如图 6 所示。每个网络层开始时，配置控制器读取层级参数并将执行描述写入配置队列（Config Queue）。执行平面随后依据队列信息发起权重与激活数据的片外 DDR 访问请求，数据经输入缓冲与行缓存重组后送入并行处理单元阵列（Processing Element Array，PE Array）执行卷积计算。计算产生的部分和在后处理流水线中完成累加、激活与量化操作，最终通过输出缓冲写回片外存储器。

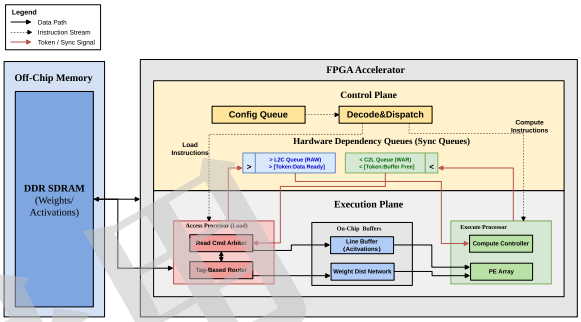


图 6 硬件调度器架构

Fig. 6 Hardware scheduler architecture

为提高片外访存带宽利用率，访问处理器内部构建了基于标签分发（Tag-based Dispatching）的共享读通道架构。系统通过读命令仲裁器对权重、激活及偏置数据的访存请求进行时分复用调度，返回的乱序数据流由控制器依据数据包头部的标签字段分发到各缓存。权重数据（Tag=WGT）经分发网络广播写入处理单元（Processing Element，PE）内部的本地存储器（PE Local RAM）；激活数据（Tag=ACT）则注入行缓冲区（Line Buffer），并通过流水线预填充建立满足卷积核高度的滑动窗口上下文，从而使计算阵列在启动后能够进入稳定的满载执行状态。

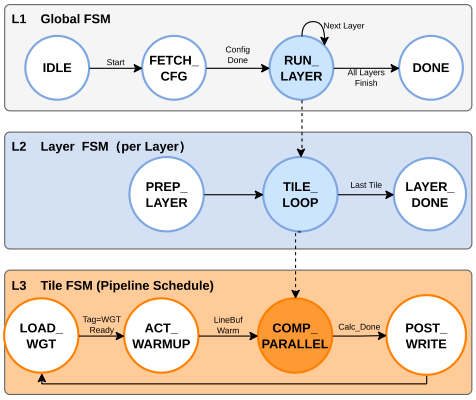


图 7 分层块级调度状态机

Fig. 7 Hierarchical FSM for tile scheduling

在完成数据访问组织与标签分发后，为保证执行过程的确定性调度，调度器内部构建了分层 FSM 体系，其结构如图 7 所示。调度逻辑自顶向下划分为全局调度层、层级配置层和块级执行层。其中，全局状态机负责网络层级的整体调度与阶段切换；层级状态机完成单层计算任务的参数配置与循环控制；底层块级状态机（Tile FSM）则协调加载（Load）、计算（Compute）与写回（Store）阶段的流水节拍。

图 7 所示 Tile FSM 的具体执行流程如算法 1 所示。单次分块计算周期内，状态机在发起权重与激活预取后，并行启动数据流调度（Data Drain task）与计算控制流：前者持续从同步 FIFO 中提取激活数据以避免队头阻塞（Head-of-Line Blocking），后者驱动 PE 阵列执行卷积乘加运算。部分和在后处理流水线中依次完成累加、偏置叠加、ReLU 激活与量化、反量化转换，最终由写控制信号生成器通过握手协议控制结果持续突发写入片外存储器。

算法 1　基于标签分发的硬件分块调度

Alg.1　Tag-dispatch based tile scheduling

输入：CMD_Stream（指令流），DDR_Bus（总线数据）

输出：PE_Array（计算阵列控制），WB_Stream（写回数据流）

```
switch State do
  case FETCH_CFG:
    Load_Layer_Params(DDR_Bus);
  case LOAD_WGT:
    Issue_DDR_Read(Tag=WGT, Addr=W_Base,
          Len=Oc_Size);
    while FIFO.Top().Tag == WGT do
      PE_Local_RAM <= FIFO.Pop();
    end while
  case LOAD_ACT_WARMUP:
  //激活层预取
    Issue_DDR_Read(Tag=ACT, Addr=A_Base,
          Len=In_Size);
    while LineBuffer.Valid_Rows < Kh do
      LineBuffer.Push(FIFO.Pop(Tag=ACT));
    end while
  case COMPUTE_PARALLEL:
    // 并行执行数据抽排与阵列计算
    while More_Acts_In_DDR do
      LineBuffer.Push(FIFO.Pop());
    end while
    for Window in Tile do
      Accumulate(Reduce_Lanes(
        PE_Array_Calc(Window)));
    end for
  case POSTPROC_WRITE:
    for Result in Accumulator do
      Stream_To_DDR(Pack(Quant(ReLU(
        Add_Bias(Result)))));
    end for
    if not Tile_Done then
      State = LOAD_WGT;
    end if
end switch
```

3　实验结果与分析

3.1　实验设置

实验选用 NWPU-RESISC45 遥感数据集对所提出架构进行验证，训练集与测试集按 8:2 比例划分，并将输入图像统一调整为 224×224像素。本文选取 VGG-16 作为基础网络，主要考虑其结构规整、卷积层占比较高，便于开展混合精度映射、硬件流水调度和故障注入分析；同时，VGG 系列网络在 FPGA CNN 加速器研究中使用较多，便于与已有工作进行横向参考。

整体实验流程包括离线训练与板级验证两个阶段。离线阶段基于 PyTorch 2.1.0 完成算法层容错训练与 PTQ 参数提取；板级验证部署于 Xilinx Virtex-7 XC7VX690T FPGA 平台，并利用 Vivado 2024.2 完成综合与实现。实验指标包括总体分类准确率（OA）、推理吞吐率（GOPS）以及故障注入条件下的分类性能变化。其中，OA 和 GOPS 分别定义为：

$$OA = \frac{N_{correct}}{N_{total}} \times 100\%$$

$$GOPS = \frac{OP_{total}}{T}$$

式中，$N_{correct}$为正确分类样本数，N_{total}为测试样本总数，OP_{total}为单次推理总运算量，T为推理时间。

3.2　容错机制有效性与消融分析

为评估分级容错机制在软错误扰动下的作用，并区分算法层容错与硬件层 TMR 的贡献，

最新录用

本文在推理阶段开展故障注入与消融实验。实验保持数据集、网络主体、混合精度配置、DAE 调度框架、FPGA 平台及测试样本一致，仅改变算法层异构双分支和控制通路 TMR 两个因素。故障注入分为数据通路和控制通路两类：前者针对中间特征图、卷积输出和激活值进行随机位翻转，低强度注入为 1、2、5、10 个 bit flips，高强度注入为 5、10、20、50 个 bit flips；后者针对 FSM 状态寄存器、控制寄存器及地址/使能信号进行翻转注入，并以任务未发生异常终止、状态失配或超时的比例统计任务完成率。本文故障注入属于加速模拟验证，不等同于真实空间辐照实验。

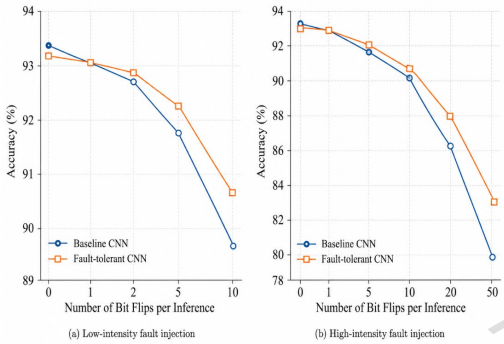


图 8 不同强度故障注入条件下模型分类准确率变化

Fig. 8 Classification accuracy under different fault injection intensities

图 8 给出了不同强度数据通路故障注入条件下模型分类准确率的变化趋势。图中 Baseline CNN 对应未启用算法层容错的基线模型，Fault-tolerant CNN 对应启用异构双分支和动态融合机制的算法层容错模型。由图可见，在低强度故障注入时，两种模型的分类准确率差异较小，说明少量瞬态扰动对模型输出影响有限；随着 bit flips 数量增加，Baseline CNN 的准确率下降更加明显，而 Fault-tolerant CNN 的退化趋势相对平缓。该结果表明，算法层容错机制能够在一定程度上抑制数据通路中间特征扰动的传播。

为区分算法层容错与硬件层 TMR 的具体贡献，本文设计了 4 组消融配置，结果如表 2 所示。其中，数据通路准确率保持率定义为高强度故障注入后准确率与无故障分类准确率之比；控制通路任务完成率表示控制故障注入条件下推理任务未发生异常终止、状态失配或超时的比例；混合故障综合保持率定义为：

$$R_{\mathrm{mix}} = \frac{R_{\mathrm{data}} \times R_{\mathrm{ctrl}}}{100}$$

式中，R_{data} 为数据通路准确率保持率，R_{ctrl} 为控制通路任务完成率。

表 2 软硬件容错消融实验配置与结果

Tab. 2 Ablation configurations and results of software–hardware fault tolerance

项目	C0 Baseline	C1 仅算法容错	C2 仅硬件容错	C3 完整方案
算法层容错	×	√	×	√
控制通路 TMR	×	×	√	√
验证目的	无容错基准	数据通路扰动抑制	控制通路故障防护	混合故障互补覆盖
无故障分类准确率/%	93.3	93.2	93.3	93.2
高强度数据通路故障后准确率/%	79.8	82.6	80.0	82.9
数据通路准确率保持率/%	85.5	88.6	85.7	88.9
控制通路任务完成率/%	83.4	83.7	95.1	95.4
混合故障综合保持率/%	71.3	74.2	81.5	84.8

由表 2 可见，仅启用算法层容错时，C1 的高强度数据通路故障后准确率由 C0 的 79.8% 提升至 82.6%，对应准确率保持率由 85.5% 提升至 88.6%，而控制通路任务完成率变化较小，说明算法层双分支结构主要改善数据通路扰动下的分类稳定性。仅启用控制通路 TMR 时，C2 的任务完成率由 83.4% 提升至 95.1%，但数据通路故障后准确率基本不变，说明硬件加固主要作用于状态跳转、地址控制和执行时序等关键控制路径。

完整方案 C3 在数据通路准确率保持率和控制通路任务完成率上均保持较高水平，综合保持指标为 84.8%。与 C1 和 C2 相比，C3 的优势并非体现为单一指标的大幅提升，而是同时覆盖数据通路扰动和控制通路故障两类失效路径。该结果说明，算法层容错与硬件层 TMR 在故障防护对象上具有互补性，可共同形成分级防护效果。

3.3 硬件性能评估与对比

图 9 给出了加速器 RTL 时序仿真波形。系统启动后，控制逻辑能够稳定驱动数据预取、并

录用稿

行乘加与后处理流水线，各模块调度节拍连续一致，未出现总线死锁或计算空转现象。综合与布局布线结果表明，架构最高工作频率可达 168 MHz；在实际部署的 150 MHz 时钟约束下，单次端到端推理任务消耗 20 478 000 个时钟周期，对应推理时延为 136.52 ms，系统吞吐率达到 226.4 GOPS，底层计算阵列平均有效执行效率为 81.5%。



图 9 加速器 RTL 时序仿真波形

Fig. 9 RTL timing simulation waveform of the accelerator

表 3 给出了 FPGA 资源使用情况。系统 BRAM 占用率为 82.11%，主要源于片上缓存、行缓存和乒乓缓冲结构的使用，以减少重复 DDR 访问并提高访存—计算重叠程度。LUT、FF 和 DSP 利用率分别为 33.6%、25.6% 和 28.8%，整体处于可实现范围内，说明混合精度 MAC 复用和控制通路 TMR 加固未引入过高逻辑资源开销。

表 3 FPGA 资源使用情况

Tab.3 FPGA resource utilization

资源类型	已用数量	可用数目	利用率
LUT	145,626	433,200	33.6%
FF	221,993	866,400	25.6%
DSP	1,038	3,600	28.8%
BRAM	1,207	1,470	82.11%

表 4 给出了 VGG-16 关键层执行性能。Conv2_2、Conv3_3 和 Conv4_3 等主要卷积层 DSP 利用率分别达到 91.2%、89.8% 和 88.5%，说明在当前分块配置下，DAE 调度和片上缓存能够较好地维持计算阵列稳态供数。相比之下，Conv1_1 的 DSP 利用率为 72.4%，可能受到首层行缓存预热、输入复用不足和边界开销影响；Classifier 层 DSP 利用率为 38.6%，主要与计算量较小、参数复用率较低以及调度开销占比升高有关。

表 4 VGG-16 关键层性能统计

Tab.4 Performance statistics of VGG-16 key layers

网络层	输出形状	计算量/GOP	时延/ms	DSP 利用率	吞吐量/GOPS
Conv1_1	224×224×64	0.17	0.58	72.4%	293.1
Conv2_2	112×112×128	3.70	8.92	91.2%	414.8
Conv3_3	56×56×256	7.40	18.15	89.8%	407.7
Conv4_3	28×28×512	7.40	18.42	88.5%	401.7
Conv5_3	14×14×512	2.36	6.20	83.5%	380.6
Classifier	1×1×45	0.01	0.12	38.6%	83.3
Total	-	30.9	136.52	81.5%	226.4

为进一步分析性能来源，本文从访存需求与访存—计算重叠角度进行说明。由于实验平台未配置专用 DDR 性能计数器，本文不报告板级实测 DDR 带宽利用率，而结合层级数据规模和表 4 中 DSP 利用率进行趋势分析。对于第 l 个卷积层，其理论 DDR 访问量可近似表示为：

$$B_l = B_{in,l} + B_{w,l} + B_{out,l} + B_{bias,l}$$

式中，$B_{in,l}$、$B_{w,l}$、$B_{out,l}$ 和 $B_{bias,l}$ 分别表示输入特征图、权重、输出特征图和偏置参数的访存量。实际访存压力还受到特征复用、权重复用及边界重叠访问影响。本文采用 DAE 调度将数据加载、卷积计算和结果写回分离，并在稳态阶段并行推进 Load、Compute 和 Store 过程，从而在片上缓存持续供数时隐藏部分访存延迟。表 3 中较高的 BRAM 占用率体现了以片上存储换带宽的设计取向，该策略适用于当前 VGG-16 网络和 XC7VX690T 平台下的性能—资源折中，但在更大模型或多任务场景下仍需进一步优化存储调度。

为评估本文架构在同类 FPGA CNN 加速器中的性能定位，表 5 选取具有代表性的 VGG/VGG 类 FPGA 实现方案进行对比。所选工作分别覆盖 8 bit 定点计算、8/16 bit 混合精度计算和相近频率条件下的 VGG 主干网络加速方案，可用于从数据精度、吞吐率和资源效率等方面进行参考。其中 DSP 使用效率定义为：

录用稿件，非最终出版稿

$$\eta_{\mathrm{DSP}}=\frac{P_{\mathrm{GOPS}}}{N_{\mathrm{DSP}}}$$

式中，P_{GOPS} 为系统吞吐率，N_{DSP} 为 DSP 使用数量。该指标用于衡量单位 DSP 资源所获得的计算吞吐能力。

表 5 架构性能与资源效率对比

Tab. 5 Performance and resource efficiency comparison of different architectures

评估指标	文献[15]	文献[16]	文献[17]	本文
硬件平台	XC7K325T	Ultra96-v2	ZC706 / xc7z045	XC7VX690T
网络模型	VGG	VGG	VGG-based	VGG-16
数据精度	8 bit	8/16 bit	16 bit	8/16 bit
工作频率/MHz	200	200	150	150
DSP 使用量	—	—	891	1,038
吞吐量/GOPS	819.2	175.30	167.5	226.4
DSP 使用效率/(GOPS/DSP)	—	—	0.188	0.218
容错机制	—	—	—	软硬协同

由表 5 可见，文献[15]的 8 bit 定点方案具有较高吞吐率，体现了低比特计算在峰值性能方面的优势，但未考虑混合精度表达能力和容错需求。本文相较于文献[16]的混合精度方案取得了更高的端到端吞吐率；相较于公开 DSP 使用量的文献[17]，DSP 使用效率由 0.188 GOPS/DSP 提升至 0.218 GOPS/DSP，表明混合精度 MAC 复用、片上缓存和 DAE 调度能够提高计算资源利用率。同时，本文进一步引入算法层数据通路容错和硬件层控制通路 TMR 加固，在故障注入条件下增强了推理稳定性。因此，本文方案不仅具有较好的吞吐性能和资源效率，也更符合星载资源受限与可靠性约束并存的应用需求。

4 结论

本文面向星载 CNN 推理中 FPGA 资源、存储带宽与运行可靠性约束并存的问题，提出一种性能—可靠性协同优化的 FPGA 加速器设计方法。该方法依据数据通路与控制通路的故障敏感性差异进行分级设计：在性能优化方面，采用层级敏感度驱动的 INT8/FP16 混合精度计算，并结合 DAE 调度、片上缓存和标签分发机制改善访存—计算并行性；在可靠性增强方面，算法层通过异构双分支和动态融合抑制中间特征扰动，硬件层通过反馈型 TMR 对关键控制逻辑进行选择性加固。

实验结果表明，在 XC7VX690T FPGA 平台部署 VGG-16 网络时，加速器在 150 MHz 下实现 226.4 GOPS 的端到端吞吐率，单帧推理时延为 136.52 ms，底层计算阵列平均有效执行效率为 81.5%。消融实验表明，算法层容错主要改善数据通路故障下的准确率保持情况，控制通路 TMR 主要提高控制故障下的任务完成率，完整方案能够同时覆盖两类故障路径。该架构可为在轨遥感图像初筛、灾害快速识别和星载边缘智能载荷等任务提供参考。本文结果基于 VGG-16 和随机位翻转故障注入，尚不能等同于真实轨道辐照实验结论。

参考文献（References）

[1]YIN Z S, WU C H, GUO C B, *et al.* A comprehensive survey of orbital edge computing: systems, applications, and algorithms[J]. *Chinese Journal of Aeronautics*, 2025, **38**(7): 103316

[2]MYSTKOWSKA G, MONOPOLI M, NANNIPIERI P, *et al*. Hardware platforms enabling edge AI for space applications: a critical review[J]. *IEEE Access*, 2025, **13**: 143939-143956

[3]LI H M, FAN X T, JIAO L, *et al*. A high performance FPGA-based accelerator for large-scale convolutional neural networks[C]//Proceedings of 2016 26th International Conference on Field Programmable Logic and Applications (FPL). Lausanne: IEEE, 2016: 1-9

[4]ZHANG C, FANG Z M, ZHOU P P, *et al*. Caffeine: towards uniformed representation and acceleration for deep convolutional neural networks[C]//Proceedings of the 35th International Conference on Computer-Aided Design. Austin: Association for Computing Machinery, 2016: 1-8

[5]GOKHALE V, ZAIDY A, CHANG A X M, *et al*. Snowflake: a model agnostic accelerator for deep convolutional neural networks[OL]. arXiv preprint arXiv: 1708. 02579, 2017

[6]WANG K, LIU Z J, LIN Y J, *et al*. HAQ: hardware-aware automated quantization with mixed precision[C]//Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern

录用稿件，非最终出版稿

Recognition. Long Beach: IEEE, 2019: 8612-8620
[7] **LI Z F, NI B B, YANG X K, et al. Residual quantization for low bit-width neural networks[J]. IEEE Transactions on Multimedia, 2021, 25: 214-227**
[8]SHAO Y Z, WANG J Y, HAN X D, *et al.* Research on spaceborne neural network accelerator and its fault tolerance design[J]. *Remote Sensing*, 2024, **17**(1): 69
[9]RECH P. Artificial neural networks for space and safety-critical applications: reliability issues and potential solutions[J]. *IEEE Transactions on Nuclear Science*, 2024, **71**(4): 377-404
[10]KIM C, LEE D, NA J. A comparison of reliability and resource utilization of radiation fault tolerance mechanisms in spaceborne electronic systems[J]. *Aerospace*, 2025, **12**(2): 152
[11]GAVA J, HANNEMAN A, ABICH G, *et al*. A lightweight mitigation technique for resource-constrained devices executing DNN inference models under neutron radiation[J]. *IEEE Transactions on Nuclear Science*, 2023, **70**(8): 1625-1633
[12] **HUANG H T, XUE X H, LIU C, et al. Statistical modeling of soft error influence on neural networks[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2023, 42(11): 4152-4163**
[13]GAO Z, GAO S H, YAO Y, *et al*. Systematic reliability evaluation of FPGA implemented CNN accelerators[J]. *IEEE Transactions on Device and Materials Reliability*, 2023, **23**(1): 116-126
[14]JUNAID M, ALIEV H, PARK S, *et al.* Hybrid precision floating-point (HPFP) selection to optimize hardware-constrained accelerator for CNN training[J]. *Sensors*, 2024, **24**(7): 2145. DOI: 10.3390/s24072145
[15]LI Shasha, LI Xiayu, LIU Shanshan, *et al*. Design and implementation of an FPGA-based general accelerator for convolutional neural networks[J]. *Journal of Fudan University(Natural Science)*, 2022, **61**(1): 69-76, 84 (李沙沙, 李夏禹, 刘珊珊, 等. 一种基于 FPGA 的通用卷积神经网络加速器的设计与实现[J]. 复旦学报(自然科学版), 2022, **61**(1): 69-76, 84. DOI: 10.15943/j.cnki.fdxb-jns.2022.01.003)
[16]CHEN Yi, LIU Bosheng, XU Yongqi, *et al*. FPGA accelerator design for hybrid precision frequency domain convolutional neural network[J]. *Computer Engineering*, 2023, **49**(12): 1-9 (陈逸, 刘博生, 徐永祺, 等. 混合精度频域卷积神经网络 FPGA 加速器设计[J]. 计算机工程, 2023, **49**(12): 1-9. DOI: 10.19678/j.issn.1000-3428.0066701)
[17]ZENG Chenglong, LIU Qiang. Design of High performance convolutional neural network accelerator for embedded FPGA[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2019, **31**(9): 1645-1652 (曾成龙, 刘强. 面向嵌入式 FPGA 的高性能卷积神经网络加速器设计[J]. 计算机辅助设计与图形学学报, 2019, **31**(9): 1645-1652)

作者简介：



刘 晖（第一作者） 男, 2001 年出生于安徽省安庆市, 现为中国科学院国家空间科学中心学生, 研究方向为智能星上信息处理、星上目标检测系统研究等.
E-mail: liuhui23@mails.ucas.ac.cn



周 海（通信作者） 男, 1987 年出生于安徽省蚌埠市, 现为中国科学院国家空间科学中心智能室主任, 博士生导师, 研究员, 研究方向为目标探测与识别、遥感图像处理等.
E-mail: zhouhai@nssc.ac.cn