

面向星载应用的高性能图像压缩核设计与实现

傅志宇^{1,2} 张学全¹

1(中国科学院国家空间科学中心 北京 100190)

2(中国科学院大学 北京 101408)

摘要 为了满足航天应用中图像数据高效存储与传输的需求,基于FPGA设计并实现了一种符合CCSDS 122.0-B-1标准的星载高性能图像压缩核。充分考虑算法特性与实现平台特点,设计了新颖的编码控制逻辑与数据组织结构。针对压缩效率这一关键指标,设计了段大小为256的算法实现架构,压缩效率处于同类应用前列。通过分段压缩机制,有效防止了误码扩散,并支持压缩质量连续调节与图像渐进传输。针对扫描与编码环节的性能瓶颈,提出了全并行扫描及动态优化并行编码方法,实测编码效率提高约50%。压缩核支持图像最大尺寸4096×4096 pixel,最大位深度16 bit。通过实际测试,压缩核吞吐率可达 $90.64 \times 10^6 \text{ sample} \cdot \text{s}^{-1}$,可满足大部分航天任务的图像压缩需求。

关键词 图像压缩, FPGA, CCSDS, 小波变换, 位平面编码

中图分类号 TP751.1

Design and Implementation of a High-performance Image Compression Core for Spaceborne Applications

FU Zhiyu^{1,2} ZHANG Xuequan¹

1(National Space Science Center, Chinese Academy of Sciences, Beijing 100190)

2(University of Chinese Academy of Sciences, Beijing 101408)

Abstract To address the critical need for efficient image storage and transmission in aerospace applications, this study presents a CCSDS 122.0-B-1-compliant compression core implemented on FPGA. The design incorporates innovative encoding control logic and optimized data organization through co-optimization of algorithmic features and hardware constraints. A segment-based architecture with 256-pixel blocks achieves superior compression efficiency among existing solutions, while effectively containing error propagation through segmented compression. The architecture further enables continuous quality adaptation and progressive image transmission. To resolve performance bottlenecks in scanning and encoding processes, fully parallelized scanning with adaptive parallel encoding was developed, and a 50% efficiency improvement was demonstrated in validation tests. Supporting images up to 4096×4096 pixel with 16-bit depth, the core delivers $90.64 \times 10^6 \text{ sample} \cdot \text{s}^{-1}$ throughput, meeting operational requirements for diverse space missions.

2025-02-14 收到原稿, 2025-06-04 收到修定稿

©The Author(s) 2026. This is an open access article under the CC-BY 4.0 License
(<https://creativecommons.org/licenses/by/4.0/>)

Key words Image compression, FPGA, CCSDS, Wavelet transform, Bit-plane encoding

0 引言

随着空间遥感成像技术的不断进步, 卫星图像的分辨率大幅提高, 产生的图像数据量显著增大. 由于卫星存储与传输带宽的限制, 图像压缩技术得到了广泛应用. 图像压缩能够在保证图像质量的同时大幅减少数据量, 从而减轻卫星存储与传输的压力.

目前, 中国已实施的航天任务中大量采用了图像压缩技术, 以缓解图像数据存储与传输的压力. 2007 年中国发射的嫦娥一号探测器搭载实时多光谱图像无损压缩系统, 采用了基于 AR 模型的 DPCM 算法^[1]; 2020 年, 嫦娥五号着陆器采用了 SPIHT 算法, 实现了对全景相机图像的多模实时压缩编码; 在 2020 年的天问一号火星探测任务中, 火星环绕器采用了 JPEG2000 算法来压缩中分辨率相机与高分辨率相机的图像, 而祝融号火星车则使用了首 1 游程的图像位面并行编码算法, 对多光谱相机和导航地形相机等产生的各类图像进行有效压缩^[2].

针对星载图像压缩需求, 空间数据咨询委员会 (CCSDS) 于 2005 年 11 月发布了适用于星载图像压缩的 CCSDS 122.0-B.1 推荐标准^[3], 简称 CCSDS-IDC 算法. 在现有的图像压缩算法中, SPIHT, JPEG2000 和 CCSDS-IDC 因性能出色而成为航天领域的主流方案. 相比 JPEG2000 和 SPIHT 算法, CCSDS-IDC 算法则针对航天场景进行了专门优化设计, 采用基于位平面的低复杂度编码架构, 更适用于星上有限的计算资源. CCSDS-IDC 标准的规定方案中包含了抗单粒子翻转的容错机制, 通过分段式码流结构及冗余校验设计, 确保在空间辐射环境下仍能保证数据的可靠性和完整性.

国际上 Kranitis 等^[4] 2014 年设计了 CCSDS-IDC 的 IP 核, 提供了一种可以方便复用的 CCSDS 算法解决方案. 该设计不需要任何外部存储器进行数据缓冲, 支持段大小最大为 $S=128$, 其并行与流水线融合的架构提供了高速处理能力, 基于 Virtex-5 QV FPGA 的最大运行频率为 142 MHz, 经典图像的吞吐率平均为 $29.4 \times 10^6 \text{ sample} \cdot \text{s}^{-1}$. Chandraprabha 等^[5] 2020 年使用两块 FPGA 分别实现小波变换与位

平面编码, 并使用了外部 SDRAM 作为数据缓存, 压缩系统支持 4096 pixel 宽的条带式图像输入, 但像素位深度限制为 10 bit. 该系统基于 RTAX2000 S 实现, 工作主频为 40 MHz, 能够以 $64 \text{ Mbit} \cdot \text{s}^{-1}$ 速率输入图像并压缩, 无法适用于高速应用的场景.

Sun 等^[6] 2019 年改进了 CCSDS 推荐的 Rice 算法中的 k 值选择算法, 减少了编码过程中遍历选择 k 值的耗时, 但会略微降低压缩效果, 并且未针对位平面编码部分进行优化设计, 仅保留了无损压缩的工作模式. Zhou 等^[7] 2018 年对 CCSDS 的位平面编码的扫描过程展开研究, 对比分析了全并行、全串行、两组并行及四组并行情况下所需的扫描时间与扫描资源, 提出了 4 组并行的串并结合的扫描方法, 但未基于硬件平台进行验证, 并且该方法仅是基于部分图像数据得到的经验优化, 泛用性不高. Hu 等^[8] 2011 年针对 CCSDS-IDC 算法中的位平面编码器, 基于 Cyclone II EP2 C70 FPGA 的实现展开研究, 设计了直流系数编码模块与交流系数编码模块两大模块 (交流系数编码模块包含块深度编码功能), 并对各模块采用乒乓操作和流水线设计, 提高算法的编码速度.

星载图像压缩算法实现平台主要有 CPU, GPU, FPGA, ASIC^[9]. CPU 受限于串行架构, 难以满足实时性及高吞吐需求; GPU 适合执行大量并行计算任务, 但并不适合复杂的编码逻辑控制; ASIC 虽具备高速特性, 但存在开发周期长且缺乏可重构性的缺陷. FPGA 凭借硬件并行架构, 能显著提升 CCSDS-IDC 算法的处理效率, 其动态可编程特性支持在轨算法迭代优化, 同时兼具开发成本低的优势.

针对 CCSDS-IDC 的实现, 研究使用了各种优化设计, 但仍然无法满足高速率应用需求. 为解决压缩核效率不高及吞吐率较低的问题, 本文提出了一种段大小为 256、段编码模块采用全并行扫描及动态优化并行编码架构的压缩核设计. 压缩核整体设计了小波变换的帧间流水与段编码的段间流水架构, 对各个流水环节的输入输出缓存采用乒乓操作, 设计了可以容纳 16 个群组 (gaggle) 的段大小缓存和逻辑. 在最复

杂且耗时的段编码环节, 设计了并行扫描架构和自适应位平面编码架构, 大大提升了压缩核的吞吐率. 为了支持更大尺寸图像并实现段大小为 256 的设计, 使用了 DDR 作为外部存储器, 同时针对 DDR 读写过程延时较大的问题进行了优化.

本文设计的压缩核支持最大输入图像尺寸为 4096×4096 pixel, 位深度为 8~16 bit. 基于 XC7 K325T FPGA 及 512 MB DDR3 实现, 主频 200 MHz 的情况下吞吐率可达 90.65×10^6 sample·s⁻¹. 使用测试图像集验证了压缩核设计的正确性, 并测试了性能, 为压缩核用于实际任务提供了有力支撑.

1 相关算法

CCSDS 122.0-B-1 标准提出的静止图像压缩算法是一种以离散小波变换 (Discrete Wavelet Transform, DWT) 和位平面编码 (Bit Plane Encoding, BPE) 技术为核心的压缩方法. 该算法首先对输入的图像数据进行离散小波变换, 然后执行位平面扫描及熵编码处理, 最后将单个编码结果拼接形成输出码流, 如图 1 所示. 位平面扫描与编码按照从高位平面到低位平面的顺序进行, 直至完成最低位平面或者达到压缩要求为止.

1.1 离散小波变换

离散小波变换能够对图像进行精细的多层次分解, 可将能量有效集中于少数系数上, 同时清晰区分图像中的平滑区域、细节与边缘信息, 这有助于在图像压缩中更高效地去除冗余, 保留关键视觉特征. CCSDS-IDC 标准中推荐采用三级二维离散小波变换, 该变换可通过多次一维离散小波变换实现. 一维离散小波变换有两种类型: 9/7 浮点 DWT 与 9/7 整数 DWT. 二者的计算效果相近, 浮点 DWT 的精度略高, 但由于实现平台为 FPGA, 整数 DWT 的硬件实现更简单高效, 因此更具适用性^[10].

CCSDS 122.0-B-1 标准蓝皮书中给出了一维 9/7 整数 DWT 的计算方法, 即

$$\begin{aligned} D_j &= x_{2j+1} - \left\lfloor \frac{9}{16} (x_{2j} + x_{2j+2}) - \frac{1}{16} (x_{2j-2} + \right. \\ &\quad \left. x_{2j+4}) + \frac{1}{2} \right\rfloor, \quad j = 0, \dots, N-1; \\ C_j &= x_{2j} - \left\lfloor -\frac{D_{j-1} + D_j}{2} + \frac{1}{2} \right\rfloor, \\ &\quad j = 0, \dots, N-1. \end{aligned} \quad (1)$$

对于一维像素数据 x_j ($j = 0 \sim 2N-1$), 计算结果 D_j 和 C_j 分别为高通分量与低通分量. 针对式 (1) 边界数据不足的情况, 需要采用镜像延拓方式补齐, 即对于左边界 ($j < 0$), 令 $x_j = x_{-j}$; 对于右边界 ($j > 2N-1$), 令 $x_j = x_{2(2N-1)-j}$.

对原本需要使用 R 位表示的图像进行三级二维整数离散小波变换, 变换结果需要使用 $R+4$ 位表示, 并且需要对各个子带进行加权, 加权后最多需要使用 $R+7$ 位进行表示. DWT 对图像的尺寸也有要求: 如果原始图像的宽和高不是 8 的整数倍, 则应使用边缘复制的方式填充, 即在图像右侧与底部添加最少数量的额外列和行, 将其尺寸补齐为 8 的整数倍.

三级小波变换后得到 10 个子带, LL3 是低频子带, 通常包含了图像的主要信息, 9 个高频子带 (HH1, HL1, LH1, HH2, HL2, LH2, HH3, HL3, LH3) 分别为图像在不同分辨率下的细节信息, 信息的重要性由低阶到高阶逐渐增加.

根据图像小波系数之间的相关性, 算法在小波系数的基础上定义了块 (block), 这是进行扫描和编码的基本单位, 其在小波系数中的定义如图 2 所示. 一个块包含 10 个子带的 64 个系数, 具体为 1 个直流系数 (DC) 和 63 个交流系数 (AC). AC 系数按照 3 个方向形成 3 个家族 (F_0, F_1, F_2), 对于任意家族 i 都对应一个父系数 (parent, p_i)、4 个子系数 (children, C_i) 以及 16 个孙系数 (grandchildren, G_i).

1.2 位平面编码

根据 CCSDS-IDC 标准, 位平面扫描与编码是以段 (segment) 为单位进行的. 将 S 个连续的块构成一个段 (其中 $2^4 \leq S \leq 2^{16}$), 段中包含的块越多, 压缩效果越好, 但相应的压缩复杂度也会提高, 所需硬件资

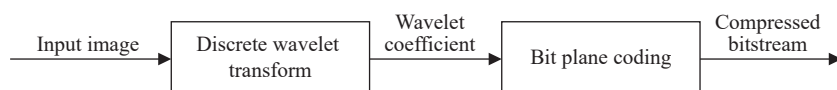


图 1 编码器总体设计

Fig. 1 General design of the encoder

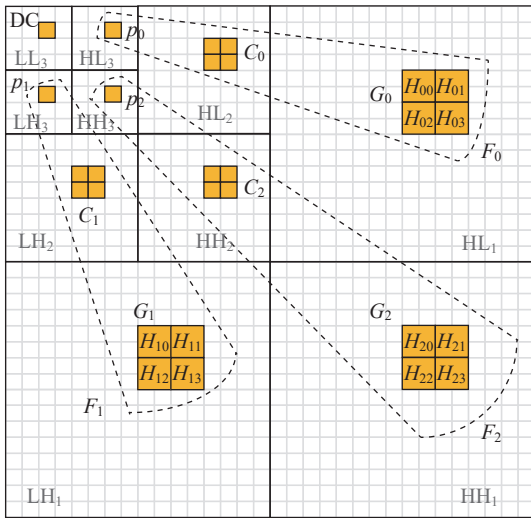


图2 块在小波系数中的定义

Fig. 2 Block definition in Wavlet coefficients

源也会增加。在熵编码时,最多16个块组成一个群组,同一群组采用同一套编码方式,一个段可包含多个群组。位平面编码主要包括段头编码、直流编码、交流位深度编码、交流系数的位平面编码四个部分。段与段之间独立进行位平面编码,这样确保空间传输中丢失的信息仅被限制在当前段内,有效避免误码扩散至其他段。

段头编码用于生成标准定义的段头信息,包含段大小、直流位深度(BitdepthDC)、交流位深度(BitdepthAC)等必要信息。直流编码与交流位深度编码以量化编码为主,直流编码分量化编码、额外 $q - \text{BitdepthAC}$ 个位平面编码及阶段0(Stage 0)编码。交流位深度编码与直流量化编码类似。交流系数的位平面编码是整个压缩的核心,整体思路是对一个段中的所有AC系数进行扫描,从位平面BitdepthAC-1开始持续向下减少,直到位平面为0,

再将扫描的字以群组为单位进行自适应熵编码,熵编码采用协议定义的映射表。

对于某一块特定定位平面的AC系数扫描来说,系数 x 在位平面 b 存在一个类型 $t_b(x)$,其值按照下式确定:

$$t_b(x) = \begin{cases} 0, & |x| < 2^b; \\ 1, & |x| = 2^b; \\ 2, & |x| > 2^b; \\ -1, & b < B(\Gamma). \end{cases} \quad (2)$$

其中, $B(\Gamma)$ 为 x 所在子带的加权系数, Γ 表示 x 所在子带。

在阶段1~阶段3中,对每个交流系数 x 的第 b 位幅值进行编码,前提是 x 在当前位平面不显著,即 $t_{b+1}(x) = 0$,编码时也将按结构分别为父系数、子系数和孙系数定义二进制字类型 $t_b[P], t_b[C_i], t_b[H_{ij}]$,按照一个集合共同编码为一串二进制字。在比较高的位平面,许多系数集合不显著,存在较多为0值的二进制字类型, t_B, t_D, t_G, t_{Hi} 分别表示所述集合是否存在显著的值。阶段4为细化阶段,对 $t_{b+1}(x) = 2$ 的系数进行编码。

2 压缩核架构设计

设计的压缩核顶层架构如图3所示。整体采用模块化设计,各功能模块间执行高效的流水线工作。设计了输入、输出接口,以输入图像帧和输出压缩后码流。使用内部存储器与外部存储器充当数据缓存,外部存储器使用DDR实现。压缩核主要包含离散小波变换模块、段预处理模块和段编码模块三个功能模块,其中各功能模块中又包含实现具体功能的子模块。

离散小波变换模块使用9/7整数离散小波变换,

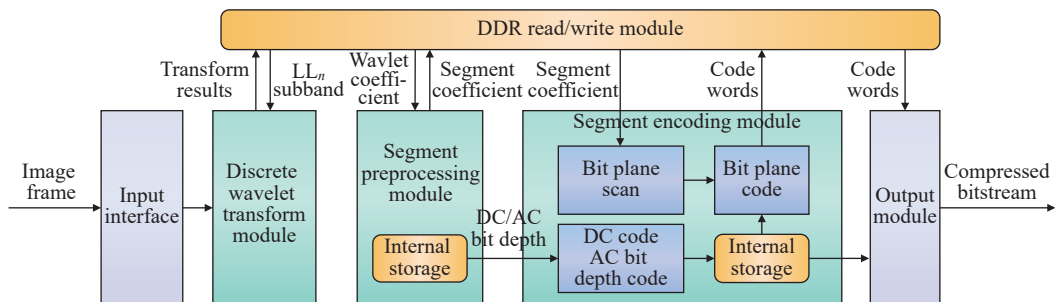


图3 压缩核顶层架构

Fig. 3 Top-level architecture of the compression core

输出整数变换结果, 支持对图像的无损压缩. 段预处理模块接收小波系数, 将其转换为段系数, 并按照式 (2) 提前计算 AC 系数在每个位平面的类型 type, 即平面类型值, 用 t_b 表示. 最后, 位平面编码模块对 DC 和 AC 系数进行熵编码, 编码结果用码字 (Code word) 表示, 部分存于内部存储, 一部分存于 DDR 模块. 码流输出模块将已经编码好的段按照优先顺序渐进式输出.

相比压缩参数 $S=16$ 时, 每个段仅包含 1 个群组, 本架构实现 CCSDS-IDC 在参数 $S=256$ 下的图像压缩, 此时每个段包含 16 个群组, 熵编码时会对每个群组内的各位平面采用自适应熵编码, 同时为了提高压缩速率, 采用群组间流水、群组内位平面并行的编码设计.

本架构还配备多组内部存储器及由 DDR 读写模块控制的外部存储器. 具体而言, 小波变换模块会在内部暂存图像行数据, 变换后的子带则存储于外部存储器中. 此外, 预处理模块将 DC 系数、DC 和 AC 的最大位深度等直流编码所需要的值保存至内部存储器中, 将 AC 系数、平面类型值等保存至外部存储器中, 由扫描模块直接取用.

为了提升压缩核的整体吞吐量, 本架构对各主要模块采用了流水线设计加乒乓操作 (见图 4). 具体而言, 小波变换模块以帧为单位变换, 变换完成的一帧存放于外部缓存, 随即可以开始下一帧变换, 而段预处理、段编码及码流输出是以段为单位进行的, 前一级完成后, 下一级有缓存空间即可开始运行. 流水线中, 压缩速率取决于周期最长的流水环节, 即段编码步骤. 因此, 本文适当放宽了其他模块对速率的要求, 并对段编码模块进行优化加速设计. 此外, 各模块内部也广泛应用了流水线与并行技术, 将在下面进行详

细说明.

2.1 离散小波变换模块

离散小波变换模块按照 1.1 节中方法, 采用折叠式三级小波变换设计, 通过复用一级小波变换以实现三级变换, 其结构如图 5 所示. 该方法通过对一级小波变换模块进行复用, 可以节省大量的资源, 由于二级变换和三级变换的数据量是一级变换的 $1/4$ 和 $1/16$, 其花费的时间也应为一级变换所用时间的对应比值, 折叠式三级小波变换消耗的时间仅是非折叠式的 1.3125 倍左右^[1]. 为了提高一级小波变换的速度, 设计了行变换与列变换之间的流水线结构, 并对变换公式进行拆分优化, 在输入部分数据后即可开始输出数据, 并使输出速率与输入速率相匹配.

一维变换的计算公式 (1) 经过变形后可写为如下形式:

$$D_j = x_{2j+1} - \left\lfloor \frac{1}{16} [8(x_{2j} + x_{2j+2}) + (x_{2j} + x_{2j+2}) - (x_{2j-2} + x_{2j+4})] \right\rfloor, \quad j = 0, \dots, N-1;$$

$$C_j = x_{2j} - \left\lfloor \frac{1 - D_{j-1} - D_j}{2} \right\rfloor, \quad j = 0, \dots, N-1. \quad (3)$$

其中量的定义及边界延拓方式与式 (1) 相同.

直接使用式 (3) 进行计算仍然过于复杂, 通过将乘法转换为加法和移位, 并将计算步骤拆解为两个步骤, 采用流水线设计, 即可实现实时变换. 具体而言, 将式 (3) 拆分为以下两步, 使用临时变量 temp_1 缓存中间数据. 每个时钟周期运算一步, 除去启动运算前需要读入数据的 5 个时钟周期, 每个时钟周期高通分量 D_j 与低通分量 C_j 交替输出, 即

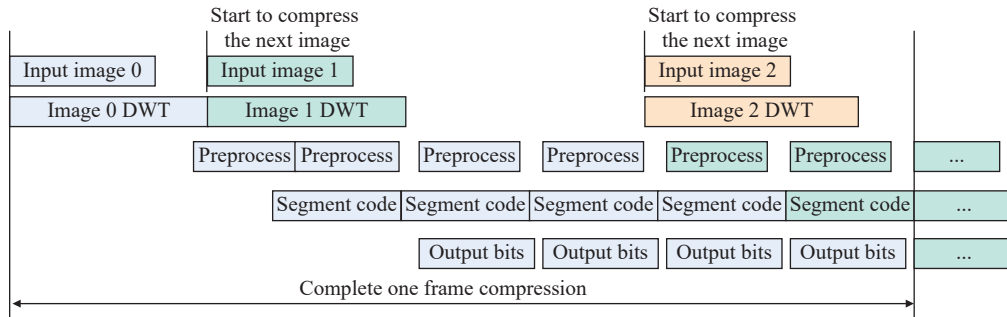


图 4 流水线设计

Fig. 4 Design of the pipeline

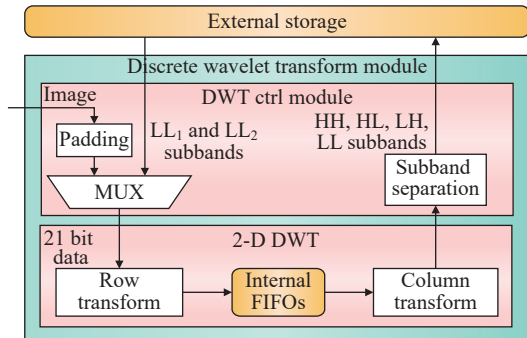


图5 离散小波变换模块架构

Fig. 5 Architecture of discrete wavelet transform module

Step1: $D_j = x_{2j+1} - (t_1 \gg 4), D_{j-1} = D_j;$

Step2: $C_j = x_{2j} - (2 - D_{j-1} - D_j) \gg 2,$
 $t_1 = (x_{2j} + x_{2j+2}) \ll$
 $3 + (x_{2j} + x_{2j+2}) - (x_{2j-2} + x_{2j+4}).$

行变换的结果将会缓存在 FIFO 中, 使用了 7 个 FIFO 作为行存储器, FIFO0 至 FIFO6 首尾相连, 供列变换模块同时读入 7 行数据. 当行变换系数存满 5 行之后, 即开始进行列变换, 行列变换的流水线结构如图 6 所示. 由于 FIFO 例化的深度限制, 支持图像像素不超过 4096×4096 pixel, 并且由于变换是以行为单位进行的, 在输入速度低于处理速度时, 仅需等待完成一行输入即可继续变换. 本文设计的小波变换架构, 对于一张尺寸为 512×512 pixel、总计 262144 pixel 的图像, 需要 265745 个周期即可完成一级变换, 启动花费的周期占比为 1.4%, 完成三级变换总共花费 350546 个周期, 约为一级变换的 1.32 倍. 考虑到本压缩核架构中小波变换并非最耗时的流水环节, 本折叠式三级变换设计占用资源较少, 因此这里采用折叠式结构设计.

为了减少外部存储器的突发次数, 增加单次突发长度, 以提高外部存储器的利用率, 设计了高效的小波系数在 DDR 中的存储地址. 具体而言, 首先为每

个子带分别分配存储地址, 接着进行更细致的划分, 将 1 级小波系数划分为 4 个行, 2 级小波系数划分为 2 个行, 3 级小波系数仅需 1 个行, 各个块按照各子带的行连续存放, 共需要 $4 \times 3 + 2 \times 3 + 4 = 22$ 个偏移地址.

2.2 段预处理模块

小波变换后的系数按子带行分别存放, 无法直接用于扫描编码, 需要将其转换为按块存储的基本编码结构. 为了提高压缩效率, 通过设定子带加权因子来决定不同子带的位平面被编码的相对顺序, 本设计中选用默认的加权值. 此外, 为了实现位平面扫描的并行化操作, 需要对每个位平面的显著性系数进行解耦, 计算每个位平面中每个块的家族系数, 即平面类型值. 具体而言, 一个块在某一位平面有如下 20 个平面类型值: MAX_AC, B, D0, D1, D2, G0, G1, G2, H00, H01, H02, H03, H10, H11, H12, H13, H20, H21, H22, H23, 分别对应图 2 中一个块在家族结构的某些像素集合. 每个值即为 0 或 1 的二进制表示, 其为 1 时, 代表对应的系数集合中在当前位平面存在显著值. 由于只需标记出系数在哪个位平面开始显著, 在计算平面类型值时不需要使用比较逻辑, 使用按位或即可.

为了减少资源占用, 提高处理效率, 段预处理模块内部拆分为组织块结构与计算平面类型值两个功能模块, 模块间采用流水操作. 首先从外部存储中读取 1 个群组的子带行, 提取 DC 系数并乘以权重后, 组织为 16 个块存放于缓存中; 再计算这 16 个块的平面类型值, 计算完成的平面类型值与 AC 系数一同缓存至外部存储中, 同时启动下一个群组的组织; 最终完成一个段的数据预处理. 预处理后的段即可进行段编码, 同时段预处理模块开始下一个段的预处理.

2.3 段编码模块

段编码模块是压缩核的核心模块, 该模块的处理速度决定了压缩核的整体吞吐率. 编码时会对一整个段进行编码, 段越大, 段编码所需的资源越多, 设计的

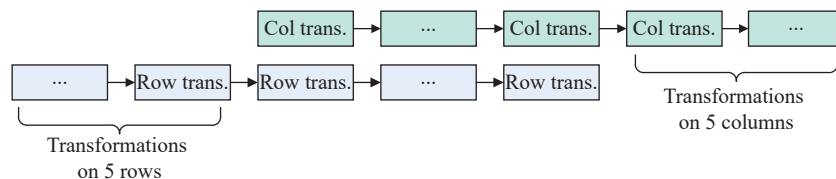


图6 行列变换流水线结构

Fig. 6 Structure of row and column transformation pipeline

复杂性也越高. 在 $S=256$ 时, 段中包含 16 个群组, 每个群组适用不同的熵编码选项, 因此需要对一个群组的所有位平面进行熵编码, 完成之后再行下一个群组的熵编码. 而协议要求段中所有块按照位平面顺序输出码流, 二者不一致, 因此需要设计额外存储和排序逻辑.

由于 DC 和 AC 位深度编码与 AC 系数的位平面扫描互不依赖, 将这两部分并行处理, AC 系数的位平面编码依赖于前两步的数据, 因此设计了直流群组模块和扫描群组模块并行, 二者共同与编码群组模块形成两级流水线的机制, 如图 7 所示. 在处理时, 这三个模块均以群组为单位进行处理, 这与熵编码的编码选项作用范围一致, 并且 1 个群组中仅包含 16 个块, 片上存储即可满足编码时所需的缓冲空间.

2.3.1 直流群组模块

直流群组模块对 DC 系数和 AC 位深度进行编码, 二者采用同样的 Rice 算法^[12], 因此通过多路复用器选择 DC 系数或者 AC 最大值, 分时复用该 Rice 编码模块完成 DC 编码和 AC 位深度编码. 同时 DC 系数的编码除了量化编码部分之外, 还存在阶段 0 和额外位平面编码 (Additional bit plane) 两个部

分, 需要在 Rice 编码的同时进行计算, 如图 8 所示. 输出的码字分别存储于对应的 FIFO 中, 使用 AX-IS 协议的 tlast 信号标识一个段的结束.

2.3.2 扫描群组模块

扫描群组模块结构如图 9 所示, 通过将 block 的各个位平面的 AC 系数与平面类型值按阶段进行扫描, 得到扫描字 (Scan words) 并存储于 BRAM 的特定地址中. 扫描共分 4 个阶段, 阶段 1~4 分别对应父系数、子系数、孙系数以及细化扫描. 扫描时, 由于阶段 4 的细化扫描在阶段 1~3 的任意阶段均可能发生, 因此在阶段 1~3 中需同时进行细化扫描, 各阶段生成的扫描字会存放于各位平面对应 BRAM 的指定地址内. 一个群组内 4 个阶段的所有扫描字将连续存储, 最终以一个空白扫描字标识结束.

对于色深达到 16 bit 的图像, 其小波变换后的系数经过加权可达 23 位, 因此需要处理的位平面最高可达 23 个. 为了尽量发挥 FPGA 的并行性优势, 使得扫描环节耗时最小, 采用面积换速度的思想设计了 23 个位平面的并行扫描架构, 并且对输入块和扫描块做了两级流水线设计, 采用两个块缓存 (Block cache) 完成乒乓操作. 并且, 由于在预处理阶段已对

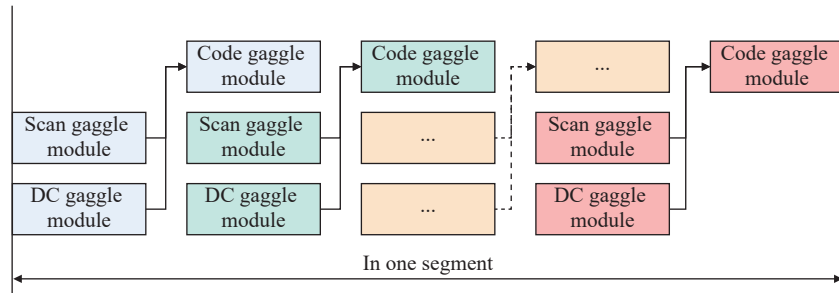


图 7 段编码模块内部流水线机制

Fig. 7 Schematic diagram of the internal pipeline of the segment encoding module

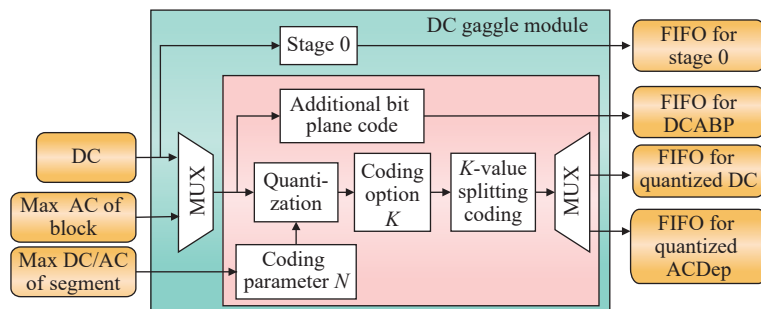


图 8 直流群组模块结构

Fig. 8 Structure of DC gaggle module structure

每个位平面的显著性系数进行解耦, 扫描时只需要每个位平面的平面类型值和 AC 系数, 不用反复计算每个位平面中每个系数值的类型 $t_b(x)$.

为实现对一个多达 256 个块的段进行编码, 本模块设计以群组为单位进行读取, 扫描时以块为单位进行扫描, 模块内部存储仅需保存 2 个块的临时信息. 同时接收上层模块的段信息输入, 实现段内中不同群组的特殊处理. 在输出的 BRAM 中可以存储 2 个群组的扫描字, 实现乒乓操作.

2.3.3 编码群组模块

编码群组模块的结构如图 10 所示, 通过对 23 个位平面的扫描字进行熵编码, 得到的码字将存储于外部存储中. 编码时, 编码核首先读入扫描字, 计算编码选项确定当前群组的最优编码, 再按照阶段 1~4 的顺序完成熵编码并生成码字. 同时, 阶段 0 的内容也会在此模块中与阶段 1~4 的结果一同输出, 存储到外部存储的对应地址内.

为了匹配扫描群组模块的速度, 并且充分发挥硬件的并行优势, 设计了动态优化并行编码方法, 其结构如图 11 所示. 传统的并行扫描需根据统计数据设计多个编码核, 并将位平面固定分配至对应编码核, 例如 Zhou 等^[7]设计的并行位平面扫描编码模块, 但

这种基于统计数据的位平面分配无法在所有情况下均达到最优效果, 且需尽可能占用全部编码核才能最大化处理速率. 根据以上思路, 本次设计的动态优化并行编码方法充分利用 A, B, C 三个编码核: 位平面指针从最高位平面开始, 检测编码核的空闲状态——如果有编码核处于空闲状态, 便将当前位平面任务分配给该编码核, 并指向下一位平面; 如果编码核处于忙碌状态, 则会等待至某一编码核空闲后再进行分配, 循环此流程, 直至所有位平面完成编码.

为进一步加快编码速率、减少逻辑资源占用, 分析原始算法, 扫描字的可变长编码需通过两步实现: 将扫描字映射为符号 (Symbol); 对符号进行熵编码. 其中, 扫描字到符号的映射表已在标准中给出, 符号的熵编码方式可根据编码选项调整, 但其结果都是确定的, 可提前计算. 因此, 本文将以上两步合并为一步进行, 提前计算了各个扫描字在不同编码选项下的编码结果以及其长度, 得到预编码表 (Precoding table), 并使用 BRAM 作为存储数据结构, 以扫描字作为地址存储预编码表.

同时, 为了支持一个段的编码, 本模块以群组为单位进行编码, 输出的码字按照位平面及阶段划分地址, 因此不同群组产生的相同阶段的码字将在外部存

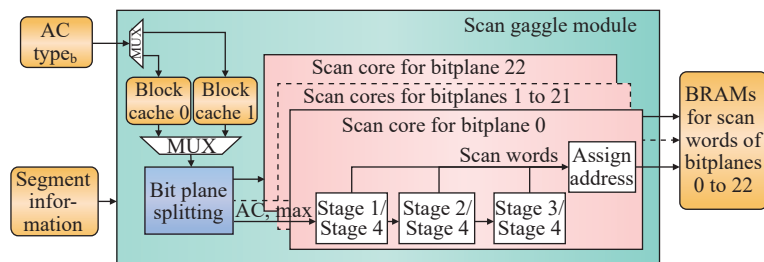


图 9 扫描群组模块结构

Fig. 9 Structure of scan gaggle module

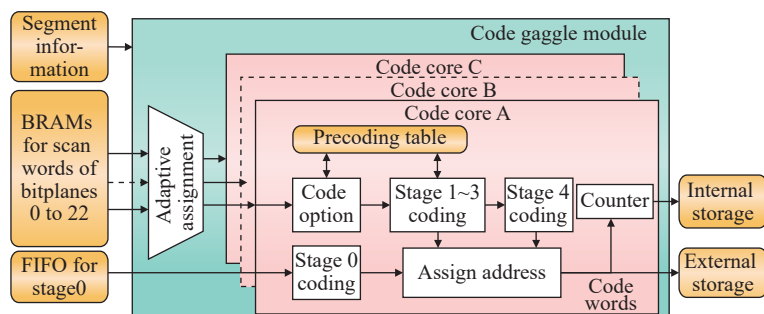


图 10 编码群组结构模块

Fig. 10 Structure of coding gaggle modules

存储器中连续存储, 以实现协议中的要求, 即段内所有块按照阶段的输出顺序排列。

3 实验结果与分析

使用 Verilog 硬件描述语言基于 Vivado 2023.2 集成开发工具完成压缩核的开发。为了验证压缩核的正确性并测试其性能, 对压缩核进行多种仿真测试及板级测试。仿真测试使用功能仿真, 编写了针对多个模块在不同场景下的 testbench 文件, 然后分析仿真波形输出。板级测试基于 XC7 K352 T FPGA 开发板进行综合实现, 并使用了卫星图像数据模拟源进行测试。卫星图像数据模拟源系统通过 LVDS 接口与待测压缩核 FPGA 连接, 使用 PC 上的上位机软件即可完成图像管理、图像加载、图像发送、图像接收等操作, 通过分析接收的压缩码流, 即可对压缩核工作状态进行分析, 如图 12 所示。实验时, 将模拟源系统与压缩板相连接, 并启动系统, 如图 13 所示。

使用的测试集包括 CCSDS 提供的真实卫星图像以及使用软件生成的渐变色和纯色图像, 这些图像位深度 8~16 bit, 使用 RAW 格式保存。

基于仿真验证及板级测试的方式分析压缩核的性能, 在衡量压缩效果时采用峰值信噪比 (Peak Signal-to-Noise Ratio, PSNR) 作为有损压缩的评价指标, 在衡量吞吐率时采用兆像素每秒为单位, 表示压缩核的吞吐率。其中 PSNR 值越高, 代表图像失真越小, 质量越好; 反之, PSNR 值越低, 图像失真越大, 质量越差。

3.1 压缩效果分析

使用 CCSDS 图像集对压缩核进行有损压缩测试, 指定不同的目标像素深度 (Bit Per Pixel, BPP),

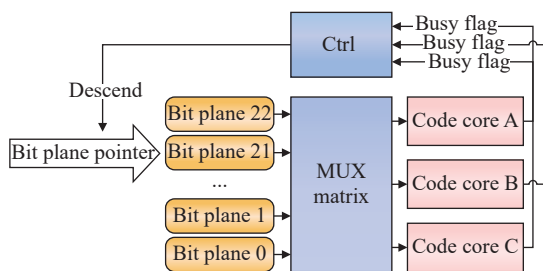


图 11 动态优化并行编码方法

Fig. 11 Method of adaptive parallel encoding

部分图像测试结果列于表 1。结果表明, 本文设计的压缩核可对图像实现高效率的压缩, 在压缩比较低时, PSNR 均可达到 40 dB 以上, 表明图像质量极好, 非常接近原始图像。对比所有测试图像, 整体压缩效率与绿皮书中吻合, 对比压缩核码流符合标准格式, 表明该压缩核设计正确, 符合 CCSDS-IDC 标准要求。

将 marseille_G6_10 b 图像在 BPP=0.25, 0.5, 1 和 2 条件下压缩后, 解压缩得到的图像截取左上角 200×200 pixel 部分, 可以看到, 当 BPP=0.25 时, 地貌不清楚, 房屋无法辨认, 随着 BPP 提高, 房屋街道变得清晰可见, 如图 14 所示。

对提出的 CCSDS-IDC ($S=256$) 算法与 CCSDS-IDC ($S=16$) 算法、SPIHT 算法和 JPEG2000 算法进行了有损压缩性能对比实验。其中 CCSDS-IDC ($S=16$) 算法通过 CCSDS 122.0 标准 C++ 软件实现, 设置段大小为 16; SPIHT 算法严格遵循 Said 等^[13]提出的原始算法, 采用自主实现的四级小波变换方案; JPEG2000 算法基于国际标准化组织提供的 Open-JPEG 2.5.0 开源库^{*}实现。如图 15 所示, 在相同 BPP 下, 本文提出的算法实现相比 CCSDS-IDC ($S=16$) 和 SPIHT 算法展现出更高的 PSNR, 表明其具有更优的压缩性能; 相比 JPEG2000 算法, 本文算法在 PSNR 相近的情况下, 复杂度显著降低, 具有更高的

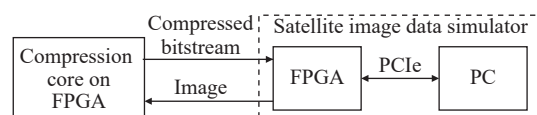


图 12 板级测试

Fig. 12 Schematic diagram of board-level testing

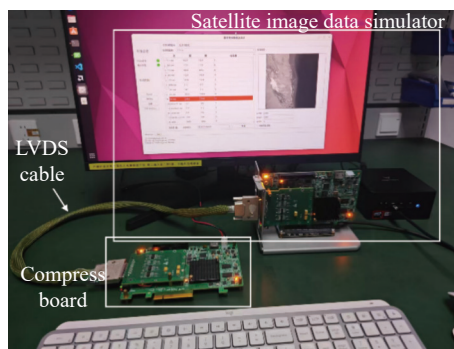


图 13 板级测试实验

Fig. 13 Board-level testing

* <https://www.openjpeg.org/>

表 1 不同目标比特率下压缩图像的 PSNR 值 (单位: dB)
Table 1 PSNR values of compressed images at different target BPP (Unit: dB)

图像名称	位深度/bit	BPP=0.25	BPP=0.50	BPP=1.00	BPP=2.00
b1	8	40.35	42.59	45.20	48.90
b2	8	40.34	42.84	45.73	49.75
lunar	8	27.64	30.68	35.07	41.10
ocean_2 kb1	10	36.03	39.42	43.89	50.71
landesV_G7_10 b	10	40.98	42.88	45.70	51.88
marseille_G6_10 b	10	28.35	31.53	35.35	41.55
solar	12	41.92	45.01	48.73	54.67
sun_spot	12	48.97	52.61	55.50	59.68
sar16 bit	16	47.64	49.76	52.88	58.50
p160_b_f	16	30.02	33.09	36.50	42.13

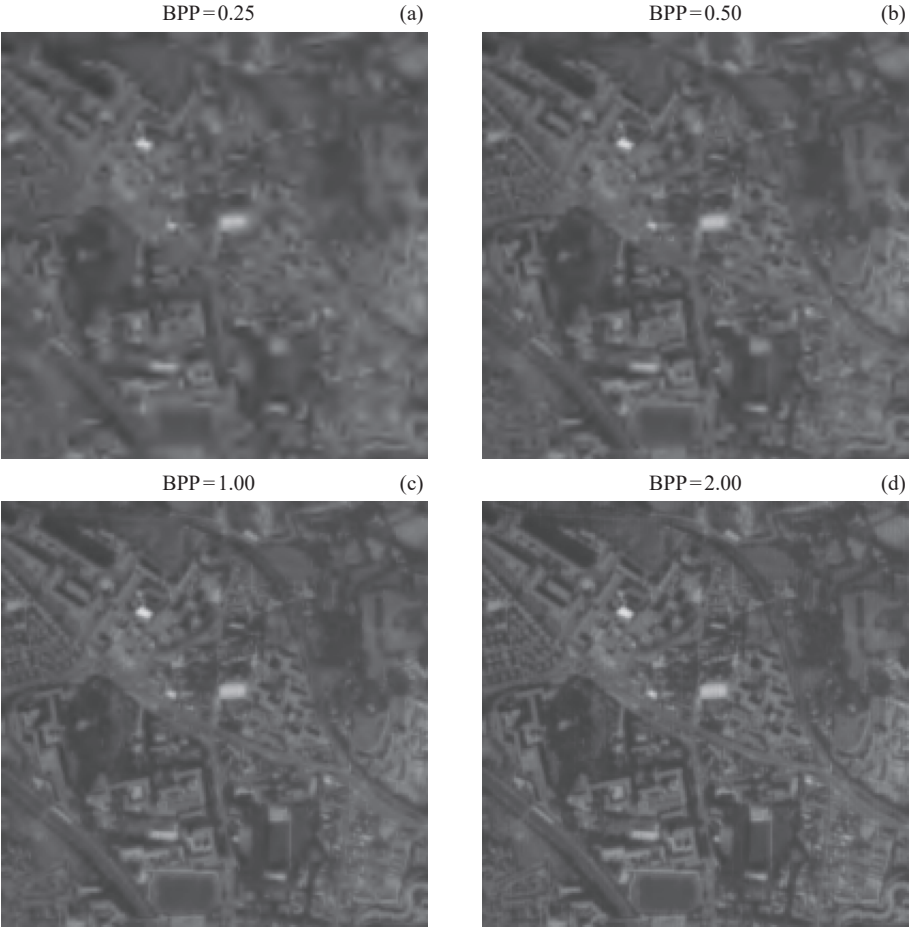


图 14 marseille_G6_10 b 在不同 BPP 条件下的压缩结果对比
Fig. 14 Comparison of compression results of marseille_G6_10 b at different BPP levels

应用价值.

3.2 吞吐率分析

本文提出的压缩核架构通过 4 级流水线设计, 减

少了压缩时间, 并针对其中耗时最长的段编码模块进行了优化设计. 通过测试 8 ~16 bit 共 12 张图像, 分别得到段编码所用周期数, 然后通过图像的像素数除

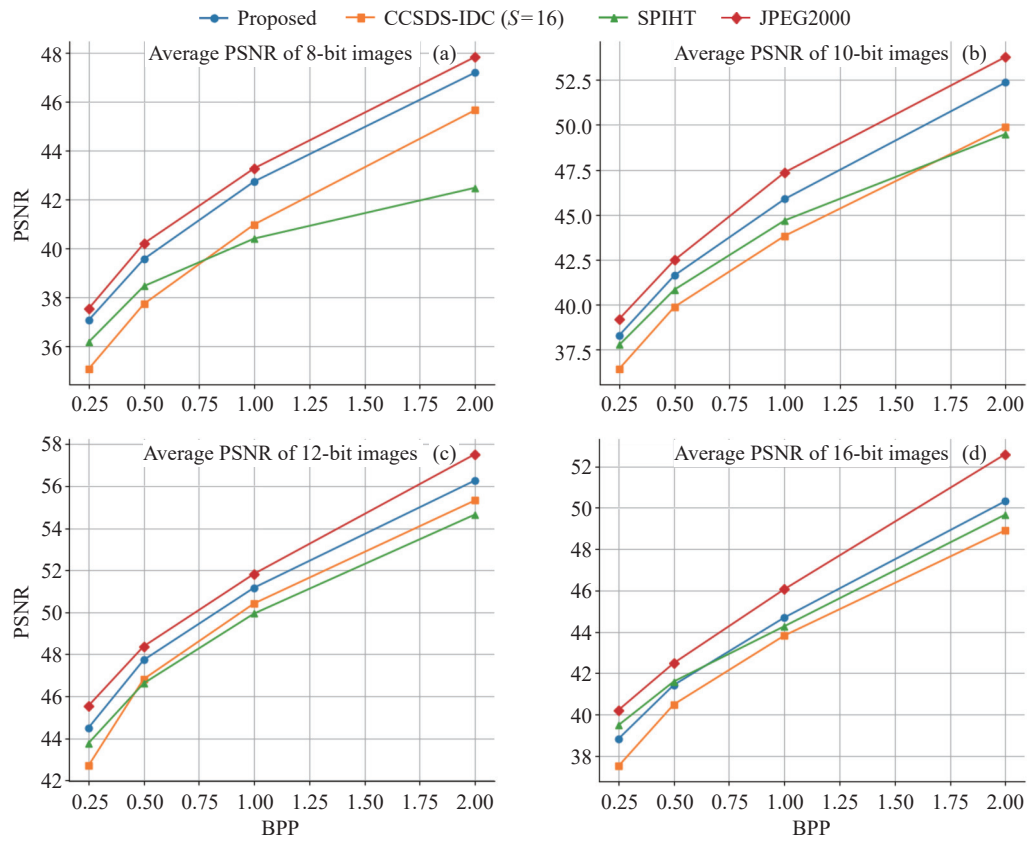


图 15 不同算法压缩效率对比

Fig. 15 Comparison of compression efficiency of different algorithms

表 2 测试图像段编码耗时及吞吐率

Table 2 Time consumption of segment encoding and throughput for test images

图像名称	位深度/bit	像素数/pixel	段编码耗时/cycle	处理速度/(sample-cycle ⁻¹)	200 MHz时的吞吐率 (×10 ⁶)/(sample·s ⁻¹)
b1	8	1048576	2399590	0.437	87.40
b2	8	1048576	2393047	0.438	87.64
marstest	8	262144	569853	0.460	92.00
lunar	8	262144	559553	0.468	93.70
ice_2 kb1	10	4194304	9220637	0.455	90.98
ice_2 kb4	10	4194304	8851466	0.474	94.77
india_2 kb1	10	4194304	8934049	0.469	93.89
india_2 kb4	10	4194304	8524784	0.492	98.40
solar	12	1048576	2375073	0.441	88.30
foc	12	524288	1202098	0.436	87.23
sar16 bit	16	262144	602975	0.435	86.95
p160_b_f	16	4194304	9700596	0.432	86.48

以周期数即可得到吞吐率, 具体结果列于表 2. 本文提出的压缩核实现了平均 $0.45 \text{ sample} \cdot \text{cycle}^{-1}$ (即 $2.21 \text{ cycle} \cdot \text{sample}^{-1}$) 的处理速度. 基于 XC7 K352 T

FPGA 实现压缩工作频率 200 MHz, 平均吞吐率为 $90.65 \times 10^6 \text{ sample} \cdot \text{s}^{-1}$.

设计的全并行扫描加动态优化并行编码方案相

比根据经验设定的并行编码方式,大幅缩短了周期数.通过对上述图像进行测试,得到本文方法优化后扫描环节和编码环节消耗的时间.测试结果显示,扫描环节耗时平均减少 30.61%,编码环节耗时平均减少 42.82%;优化前压缩核处理速率约为 $0.30 \text{ sample} \cdot \text{cycle}^{-1}$,优化后为 $0.45 \text{ sample} \cdot \text{cycle}^{-1}$,编码效率整体提升约 50%,如图 16 所示.

将测试图像的扫描环节和编码环节消耗的周期数与使用 2~4 个编码核消耗的周期数进行对比,如图 17 所示.从图中可以看到,编码环节与扫描环节的速度基本匹配,其中性能瓶颈出现在扫描环节上,如果是 2 核编码,编码环节会变为性能瓶颈;如果是 4 核编码,则其性能过剩,与扫描环节速度不匹配,因此 3 核编码是在保证性能前提下的最优选择.

吞吐率作为压缩核的一项重要指标,直接反映了压缩核在单位时间内能够处理的图像样本数量.由于

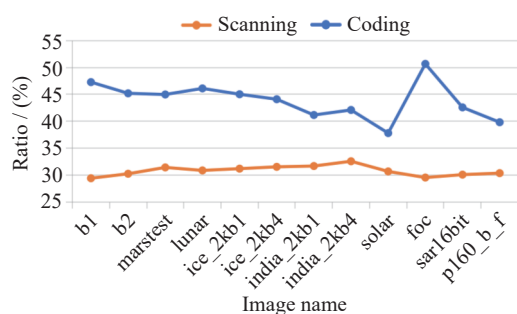


图 16 扫描与编码环节优化后时间减少百分比

Fig. 16 Time reduction percentage after optimization of the scanning and encoding processes

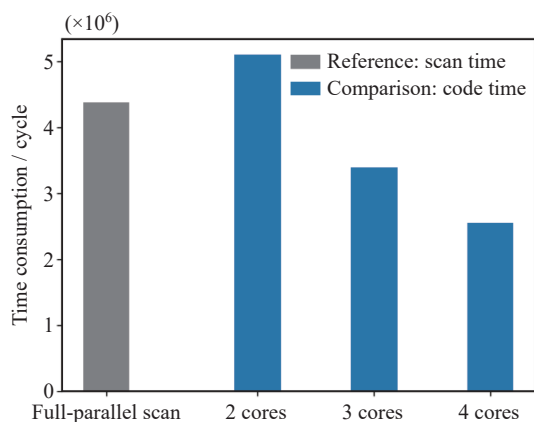


图 17 扫描环节与多核编码环节耗时

Fig. 17 Time consumption for the scanning process and multi-core encoding process

星上资源有限且对实时性要求较高,因此压缩核的吞吐率十分重要.本文提出的一种新的扫描与编码实现架构不仅实现了较高的吞吐率,还能够不依赖于统计数据而适应多种类型图像,有较高的实用价值.

4 结论

对 CCSDS 122.0-B-1 图像压缩算法进行了研究,针对 FPGA 对该算法进行优化改进,设计了段大小为 256 的压缩核架构,引入了新的加速段编码的方法.在设计中,充分利用 FPGA 的并行化优势,设计了整体的 4 级流水线架构,对段编码模块中的扫描和编码步骤进行了优化设计,采用 23 个位平面全并行扫描和 3 个编码核动态优化并行编码,大大提高了压缩核的吞吐率.运行于 XC7 K352 T FPGA 中,实现压缩工作频率为 200 MHz,吞吐率为 $90.65 \times 10^6 \text{ sample} \cdot \text{s}^{-1}$,最大支持 $4096 \times 4096 \text{ pixel}$ 的图像.使用卫星图像与生成图像,通过仿真测试与板级测试对压缩核性能进行评估,实验结果表明,本压缩核按照 CCSDS-IDC 标准实现了图像压缩,与协议中定义的段头字段、码字字段完全一致,支持协议中码率控制、渐进传输等功能,并且拥有较高的压缩吞吐率.

本文设计的基于 FPGA 的 CCSDS 122.0-B-1 图像压缩核架构不但提高了压缩效率,而且具备良好的适配性,仅需一块 FPGA 及 DDR 即可实现,国产器件也可以支持使用,为中国卫星图像的高效传输与处理提供了一种有效的解决方案.未来可继续优化压缩算法和硬件架构,探索更高效的并行处理技术和更低功耗的实现方案,以满足航天器对图像压缩技术的更高要求.

利益冲突 所有作者均声明不存在利益冲突.

参考文献

- [1] ZHANG Xuequan. Implementation of Onboard Image Compression System Using FPGA[D]. Beijing: Center for Space Science and Applied Research, Chinese Academy of Sciences, 2009 (张学全. 基于 FPGA 的星载图像压缩系统实现方法研究 [D]. 北京: 中国科学院研究生院 (空间科学与应用研究中心), 2009)
- [2] ZHU Jianbin, XU Yong, WANG Cuilian, *et al.* Image compression software design for Zhurong Mars exploration rover[J]. *Journal of Deep Space Exploration*, 2021, 8(5):

- 503-510 (朱剑冰, 徐勇, 王翠莲, 等. “祝融号”火星车图像压缩软件的设计与实现 [J]. *深空探测学报 (中英文)*, 2021, 8(5): 503-510)
- [3] CCSDS 122.0-B-1: Image Data Compression[S]. Washington D C: Consultative Committee for Space Data Systems (CCSDS), 2005
- [4] KRANITIS N, THEODOROU G, TSIGKANOS A, *et al.* A reconfigurable FPGA implementation of CCSDS 122.0-B-1 image data compression for ESA PROBA-3 coronagraph system payload[C]//On-Board Payload Data Compression Workshop. Venice: ESA & CNES, 2014
- [5] CHANDRAPRABHA K, SHIAK N A, LAKSHMI A, *et al.* An efficient FPGA implementation of DWT based image data compression system for microsatellites[C]//Proceedings of the 8th International Conference on Advanced Computing and Communication Systems (ICACCS). Coimbatore: Indian Institute of Industrial Engineering (IEEE), 2022
- [6] SUN Jianwei, ZHANG Zhongwei, ZHENG Tie, *et al.* Design of lossless compression system for CCSDS on-board data based on FPGA[J]. *Chinese Journal of Space Science*, 2019, 39(5): 694-700 (孙建伟, 张忠伟, 郑铁, 等. 基于 FPGA 的 CCSDS 星载数据无损压缩系统设计 [J]. *空间科学学报*, 2019, 39(5): 694-700)
- [7] ZHOU Wenjing, ZHANG Xuequan, AN Junshe, *et al.* Optimal implementation of bit plane encoder for CCSDS-IDC[J]. *Microelectronics & Computer*, 2017, 34(9): 32-37 (周文敬, 张学全, 安军社, 等. CCSDS-IDC 位平面编码的优化实现 [J]. *微电子学与计算机*, 2017, 34(9): 32-37)
- [8] HU Yonggang. Research on Bit-Plane Encoding of CCSDS Image Compression and Implementation on FPGA[D]. Xi'an: Xidian University, 2011 (胡永刚. CCSDS 图像压缩算法位平面编码技术研究及其 FPGA 实现 [D]. 西安: 西安电子科技大学, 2011)
- [9] SUN Nan. Design and Realization of Airborne Large Size Image Compression System[D]. Taiyuan: North University of China, 2024 (孙楠. 机载大尺寸图像压缩系统的设计与实现 [D]. 太原: 中北大学, 2024)
- [10] MACHAIRAS E, KRANITIS N. A 13.3 Gbps 9/7M discrete wavelet transform for CCSDS 122.0-B-1 image data compression on a space-grade SRAM FPGA[J]. *Electronics*, 2020, 9(8): 1234
- [11] DONG Mingyan, LEI Jie, WANG Keyan, *et al.* Highly efficient VLSI architecture for DWT with low-storage implementation[J]. *Journal of Xidian University*, 2016, 43(2): 35-40 (董明岩, 雷杰, 王柯俨, 等. 高效低存储 DWT 的 VLSI 结构设计 [J]. *西安电子科技大学学报: 自然科学版*, 2016, 43(2): 35-40)
- [12] JIANG Z C, PAN W D, SHEN H D. Universal Golomb-Rice coding parameter estimation using deep belief networks for hyperspectral image compression[J]. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2018, 11(10): 3830-3840
- [13] SAID A, PEARLMAN W A. A new, fast, and efficient image codec based on set partitioning in hierarchical trees[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 1996, 6(3): 243-250

作者简介



傅志宇 男, 2000 年 1 月出生于重庆市, 现为中国科学院国家空间科学中心硕士研究生, 专业为计算机应用技术, 主要研究方向为星载图像处理。

E-mail: fuzhiyu22@mailsucas.ac.cn



张学全 (通信作者) 男, 1979 年 1 月出生于吉林省白城市, 现为中国科学院国家空间科学中心高级工程师, 硕士生导师, 主要研究方向为空间综合电子技术与星上信息处理等。

E-mail: zxq_cas@163.com