Future Technology
Open Access Journal
ISSN 2832-0379

August 2026| Volume 05 | Issue 03 | Pages 175-191

Journal homepage: https://fupubco.com/futech

https://doi.org/10.55670/fpll.futech.5.3.16

Future Publishing LLC

Article

Multiplier leadership optimization algorithm (MLOA) for high-resolution images in approximate DWT-based compression systems

R. Anitha, Sri Adibhatla Sridevi*

Dept. of Micro & Nanoelectronics, Vellore Institute of Technology, TamilNadu, India

ARTICLE INFO

Article history:
Received 07 January 2026
Received in revised form
22 April 2026
Accepted 01 June 2026

Keywords:
Image compression, MLOA,
Approximate computing,
Discrete wavelet transform,
Parallel VLSI architecture

*Corresponding author
Email address:
sridevi@vit.ac.in

DOI: 10.55670/fpll.futech.5.3.16

ABSTRACT

Image compression is a basic need for efficient storage and transmission of high-resolution visual information of modern imaging and sensing systems. Algorithmic-level approximation within biorthogonal discrete wavelet transforms (DWT)-based compression has become an effective means in reducing the computation cost while keeping the image perceptual quality. A convolution-based wavelet framework introduced at the multiplier level to control the approximation leads to lower power usage and silicon area in hardware implementations in a systematic manner. In this work, the Multiplier Leadership Optimization Algorithm (MLOA) is used to select exact or approximate multiplier configurations under PSNR and SSIM constraints for energy-efficient hardware implementation. Wallace tree, Dadda tree, Vedic, and Baugh-Wooley multiplier architectures are embedded into the wavelet transform for efficient computation. Simulating with image datasets such as Castle, Baboon, Cameraman, Woman, and Boat shows that the evaluated configurations maintain PSNR values above 30 dB, while SSIM is used as the primary feasibility constraint for structurally sensitive images. The FPGA synthesis results show that Dadda-based MLOA configuration achieves the lowest normalized power and delay among the evaluated multiplier architectures, while the Multiplier Leadership Optimization Algorithm-based Leader-Column Approximate Kogge-Stone Adder (MLOA-LC-AKSA) configuration achieves 145 LUTs, 3.6 ns delay, 52 mW power, 187.2 pJ power-delay product, and 277 MHz maximum frequency. Furthermore, the parallel execution of row-wise and column-wise wavelet convolutions yields a throughput improvement of up to 41%. These results validate the algorithmic-level approximation, HDL-based hardware feasibility, and the suggested framework's parallel-processing capacity as a scalable, energy-efficient, hardware-oriented high-resolution picture compression solution.

1. Introduction

High-resolution imaging devices and visual sensing platforms generate large image datasets in remote sensing, medical imaging, autonomous systems, and multimedia communication. This growth drives demand for image compression methods that reduce storage, bandwidth, and energy consumption while maintaining perceptual quality. Wavelet-based compression is popular among methods due to its multilevel decomposition, higher computational complexity, and compatibility with perceptual coding. In particular, the discrete wavelet transform (DWT) provides a mathematically sound technique for capturing image content at multiple spatial scales, making it ideal for high-quality image compression. The DWT theory relies on multiresolution decomposition, facilitating localized time-frequency analysis at low computational cost and enabling effective image representation for compression [1]. Linear phase and symmetry of biorthogonal wavelets also enhance the quality of the reconstruction. But convolution-based DWT implementations involve a large number of multiplication-accumulation operations, resulting in excessive power consumption, silicon area, and hardware delay. Lifting-based and multiplierless DWT architectures have the advantage of low arithmetic complexity but only offer limited possibilities for controlled approximation at the multiplier level. Approximate computing is a design approach suitable for image-processing applications that are tolerant of bounded numerical errors without significantly affecting perceptual quality. Previous studies have demonstrated that approximate circuits can enhance performance [2-4].

Approximate arithmetic has been used in image compression with transforms to achieve acceptable reconstruction quality [5]. Recent studies on approximate DWT and hybrid arithmetic further illustrate the potential of approximate adders and dynamic reconfiguration for visual processing systems [6,7]. But most current methods rely on multiplier-less architectures, stand-alone arithmetic units, or adder-level approximations, neglecting the multiplier-centric convolutional DWT and the requirements of parallel processing.

In order to overcome these shortcomings, this work introduces a Multiplier Leadership Optimization Algorithm(MLOA) based approximate biorthogonal DWT framework for image compression. Unlike lifting-based approaches, the proposed method retains multiplier-based convolution, allowing approximation to be applied directly to the dominant hardware-cost units. Although DWT-based image compression and approximate arithmetic have been widely studied, three gaps remain. First, many DWT hardware designs use lifting-based or multiplierless structures, which reduce complexity but limit controlled multiplier-level approximation. Second, approximate multipliers are often tested as isolated circuits or simple image-processing blocks, rather than inside a complete DWT compression pipeline. Third, a limited number of studies jointly evaluate image quality, arithmetic error, parallel execution, and FPGA-level hardware metrics. Therefore, this work proposes an MLOA-based convolution-based biorthogonal DWT framework that integrates approximate multipliers into MAC-intensive DWT stages to reduce power, area, and delay while preserving PSNR and SSIM. The key contributions of the given work may be summarized as follows:

- A multiplier-centric convolutional-based biorthogonal DWT compression framework to facilitate controlled algorithmic-level approximation.
- Integration of optimization-guided approximate multipliers in a parallel architecture for high-resolution image compression.
- Comprehensive evaluation of image quality, parallel execution behavior, and hardware implications are carried out.
- A systematic comparison of multiplier setups, both precise and approximate, to prove the feasibility of energy-efficient DWT-based compression for image processing systems.

The organization of the paper is as follows: Section 2 reviews the literature. Section 3 explores the proposed technique, Section 4 elucidates the experimental setup, and Section 5 presents the results and analysis. Section 6 presents the discussion, while Section 7 presents the conclusion.

2. Related work and research gap

Wavelet-based image compression has been widely studied because it provides multiresolution representation, high energy compaction, and reduced blocking artifacts compared with block-based transforms. To enhance compression efficiency, Baviskar et al. suggested replacing the sub-band with a selective-manipulation sub-band used in the DWT Compression technique [8]. In a similar fashion, Bruni et al. applied biorthogonal wavelets for image compression at various resolutions and demonstrated the effectiveness for high-quality image reconstruction because of their linear phase property [9]. From these studies, it can be concluded that DWT and biorthogonal wavelets are effective bases for image compression.

Research began to concentrate on the hardware-efficient implementation of real-time and low-power systems as the DWT algorithms became more sophisticated. Jana proposed a multiplierless biorthogonal DWT architecture using a 2-D filter mask operation to reduce arithmetic complexity by replacing multipliers with adder-subtractor structures [10]. George presented a hardware-efficient DWT architecture for visual sensor networks with emphasis on low-power operation [11]. However, such multiplierless or exact-arithmetic designs provide limited flexibility for controlled arithmetic approximation and accuracy-energy trade-off analysis. Approximate DWT has therefore gained attention for energy-efficient image processing. Urban et al. [6] studied approximate adders in DWT architectures and showed that energy consumption can be reduced while maintaining acceptable image quality. Approximate compressors and multipliers have also been studied extensively. Zakian et al. [12] introduced approximate compressors for error-resilient multiplier design, while Dehkordi and Ahmadifar proposed an approximate 8:2 compressor for image processing applications [13]. Ahmadinejad et al. further demonstrated energy-efficient FinFET-based approximate 4:2 compressors [14]. At the multiplier level, Rashidi, Tammineni et al., Gowdar and Parameshwara, and Chakraborty et al. investigated approximate multiplier designs for filter and image-processing workloads, demonstrating favorable energy-accuracy trade-offs [15-19].

Although all these advances have been made, most studies have tested approximate arithmetic units as stand-alone circuits or as simple kernel units, rather than in the context of a full DWT compression pipeline. There are also existing approximate DWT studies that are primarily oriented towards adder-level approximations or multiplierless structures, and there is limited understanding of multiplier-centric approximations for convolution-based DWT. Furthermore, the behavior of parallel execution and its interaction with approximation-induced error are rarely considered in high-resolution image compression. Therefore, a clear research gap remains in combining algorithmic-level approximation, multiplier-centric DWT computation, image-quality evaluation, and architecture-aware parallel processing in a unified framework. The present work aims to fill these gaps by presenting a multiplier-based, convolution-biorthogonal DWT compression system that incorporates MLOA-guided approximate multipliers in a parallel framework.

3. Proposed methodology

This section introduces a methodology for designing and evaluating a parallel approximate Discrete Wavelet Transform (DWT) architecture for high-resolution image compression using MLOA. The main objectives of this work are to: (i) develop a multiplier-centric convolution-based biorthogonal DWT compression framework, (ii) apply MLOA for leader-column-guided approximate multiplier (MLOA-LC) selection, (iii) evaluate reconstructed image quality using PSNR and SSIM, and (iv) validate hardware feasibility with FPGA-level metrics such as LUT utilization, delay, power, and maximum operating frequency. The suggested algorithm effectively maintains the multiplier-based computation while proposing an algorithmic approximation that optimizes arithmetic units. The methodology combines the convolution-based biorthogonal DWT formulation, the approximation multiplier model, parallel processing analysis, and architecture-considerate performance estimation to make sure the model is in line with hardware implementation goals.

3.1 Biorthogonal discrete wavelet transform formulation

The compression framework uses convolution-based two-dimensional biorthogonal DWT with standard wavelet families i.e. Bior 9/7 and Bior 4.4 which are extensively used in high-quality image compression standards because of their properties of strong energy compaction and linear phase. For an input image I(x,y), the 2D DWT is calculated using separable row-wise and column-wise convolutions with predefined low-pass and high-pass filter coefficients. These convolutions require a large number of multiplication–accumulation (MAC) operations. Therefore, convolution-based DWT provides a suitable framework for evaluating approximate multiplier architectures, since multipliers dominate power consumption, area utilization, and critical-path delay in hardware implementation. Multi-level decomposition is done recursively on the LL sub-band using the same convolution process. To ensure a fair comparison, all wavelet parameters, filter length, decomposition levels, and boundary extension methods are held constant across both the exact and approximate implementations.

Although lifting-based DWT reduces arithmetic complexity by minimizing multiplier usage, it provides limited scope for evaluating approximate multiplier architectures. Conversely, convolution-based DWT employs explicit MAC operations for row-wise and column-wise filtering, making it appropriate for the integration of MLOA-selected approximation multipliers and the analysis of trade-offs in power, area, latency, and image quality. The main differences between lifting-based and convolution-based DWT in the context of this work are summarized in Table 1. The four sub-bands (LL, LH, HL, and HH) for each level are produced by the separable biorthogonal DWT, as seen in Figure 1. The LL sub-band is recursively deconstructed, and the convolution MAC operations indicate where precise or approximate multipliers are incorporated in the subsequent stages.

After DWT decomposition, compression is achieved by retaining significant wavelet coefficients and suppressing low-magnitude ones. For each sub-band coefficient $C(u,v)$, thresholding is applied as

$$C_T(u,v) = \begin{cases} C(u,v), & |\, C(u,v)\,| \geq T \\ 0, & |\, C(u,v)\,| < T \end{cases} \quad (1)$$

where T is the coefficient threshold and $C_T(u,v)$is the thresholded coefficient. The retained coefficients are then used to reconstruct via an inverse biorthogonal DWT. The compression ratio is estimated as

$$CR = \frac{N_{\text{total}}}{N_{\text{nonzero}}} \quad (2)$$

where $N_{\text{total}}$is the total number of wavelet coefficients and $N_{\text{nonzero}}$is the number of retained nonzero coefficients after thresholding. This procedure clarifies that the proposed framework evaluates transform-domain coefficient reduction and reconstruction quality rather than implementing a full entropy-coded image codec.

Table 1. Comparison between lifting-based and convolution-based DWT for approximate multiplier evaluation

Feature	Lifting-based DWT	Convolution-based DWT
Arithmetic complexity	Lower	Higher
Multiplier usage	Reduced or minimized	Explicit MAC operations
Approximate multiplier evaluation	Limited	Directly supported
Hardware cost	Lower	Higher
Approximation flexibility	Lower	Higher
Suitability for this work	Limited	High

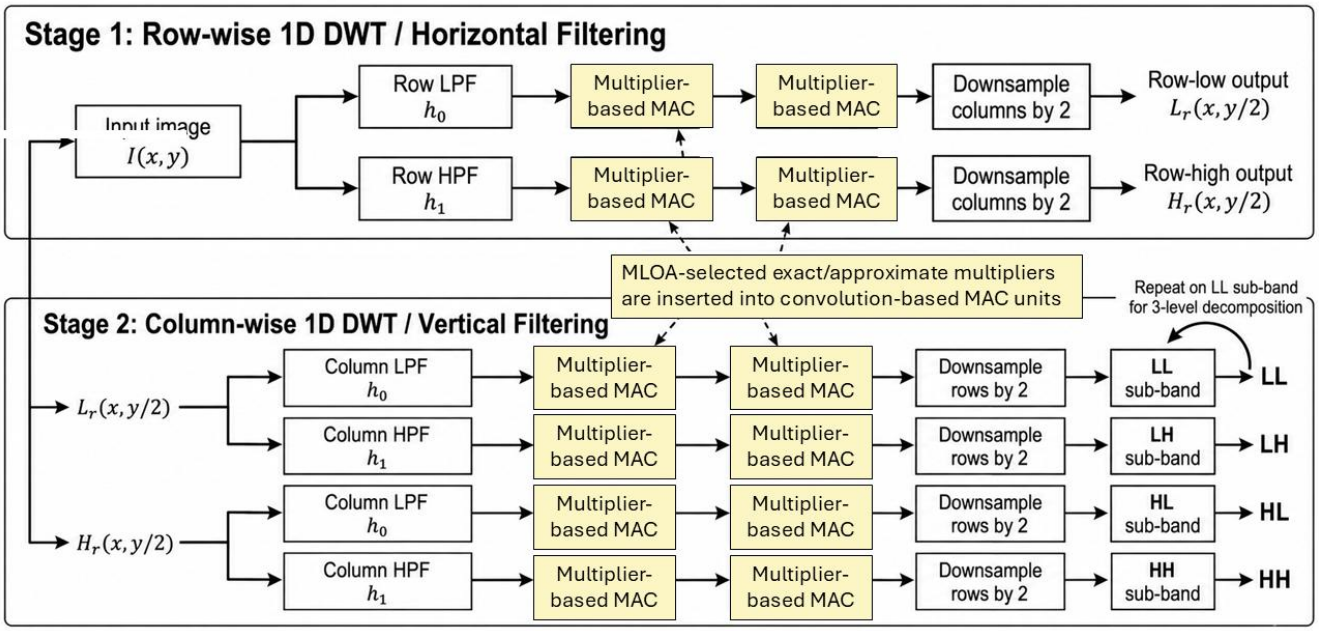

Figure 1. Separable biorthogonal CDF 9/7 DWT/IDWT architecture with row–column convolution and recursive LL-band decomposition

3.2 Optimization-driven algorithmic approximation strategy

Approximation is introduced strictly in the algorithmic aspect using the Multiplier Leadership Optimization Algorithm (MLOA) that controls the arithmetic approximation in the convolution-based biorthogonal DWT computation. In this work, MLOA is implemented as a leader-column-guided optimization mechanism at the multiplier-reduction level, in which the partial product matrix is systematically reduced using compressor allocation rules inspired by Dadda-based height constraints. Based on this leadership-driven reduction strategy, MLOA enables controlled approximation by selecting appropriate multiplier structures, such as Wallace, Dadda, and Baugh-Wooley multipliers, in either exact or approximate mode. The selected multiplier configuration is evaluated using a constrained multi-objective cost function defined as

$$J(\theta) = \alpha P_{\text{norm}}(\theta) + \beta A_{\text{norm}}(\theta) + \gamma E_{\text{norm}}(\theta) \quad (3)$$

where θdenotes a candidate multiplier configuration, $P_{\text{norm}}(\theta)$is the normalized power, $A_{\text{norm}}(\theta)$is the normalized area, and $E_{\text{norm}}(\theta)$is the normalized arithmetic error. The weighting factors α, β, and γ control the relative importance of power, area, and error, respectively.

The arithmetic error is represented using normalized error distance as:

$$NED = \frac{MED}{P_{\max}} \quad (4)$$

where $MED$is the mean error distance and $P_{\max}$is the maximum possible product value for the selected multiplier word length. Image quality is constrained during selection as:

$$PSNR \geq PSNR_{\min}, SSIM \geq SSIM_{\min} \quad (5)$$

Only configurations that satisfy Equation (5) are considered feasible for the final MLOA selection.

In the proposed architecture, the Multiplier Leadership Optimization Algorithm (MLOA) is used as a hardware-level optimization strategy for designing fast and energy-efficient multiplier reduction trees employed in convolution-based biorthogonal DWT computation. Unlike global optimization methods, MLOA operates directly on the partial product (PP) matrix of the multiplier, focusing on reducing the reduction-tree height and critical-path delay through column-wise leadership-based optimization. For an $N \times N$multiplier, the partial products are generated as:

$$PP_{i,j} = A_i B_j, 0 \leq i, j < N \quad (6)$$

where $A_i$and $B_j$denote the $i^{th}$and j^{th}bits of the input operands, respectively. The partial products are arranged column-wise according to their binary weight, where column $k$contains all partial products satisfying $i + j = k$. The height of each column equals the number of partial products in that column. The column at index $k = N - 1$has the maximum height N. Since this column significantly reduces tree depth and supports propagation, it is identified as the leader column in MLOA (MLOA-LC). Once the leader column is identified, MLOA uses it to guide the reduction process across all stages. The reduction follows a Dadda-based height constraint, where the allowable column height for each stage is calculated as:

$$d_{r+1} = \lfloor 1.5 d_r \rfloor \quad (7)$$

where $d_r$is the allowable column height at reduction stage r, and $d_{r+1}$is the next allowable height in the Dadda reduction sequence. The sequence starts with $d_1 = 2$and is generated until the maximum required height is reached. For an 8×8multiplier, the practical reduction sequence is $8 \rightarrow 6 \rightarrow 4 \rightarrow 3 \rightarrow 2$.

The first step is to reduce the leader column to the target height, and then the same reduction is applied symmetrically to the neighboring high columns at each subsequent reduction stage. This is a leader-guided method that ensures no column exceeds the stage-wise height limit, preventing unnecessary carry propagation and minimizing the critical path. If the compressor is allocated column-wise, the excess height is used for allocation.

$$\Delta H_k = \max\left(0, H_k - H_{\text{target}}\right) \quad (8)$$

where H_k is the current height of column k, H_{target} is the maximum height allowed for the current reduction stage, and ΔH_k is the number of bits above the target height in column k that need to be reduced using compressors. To reduce column height, suitable compressors, such as 3:2 and 4:2 compressors, are used. If you're using a higher-order multiplier, higher-order compressors, such as 8:2 compressors, can also be used.

The compressor placement always starts with the leader column, followed by a symmetric reduction of the adjacent columns. It is a systematic distribution of logic that reduces logic depth and balances hardware utilization. Moreover, MLOA can be used to deploy hybrid compressors, in which exact 3:2 and 4:2 compressors are used in the leader and neighboring high-height columns to preserve reduction accuracy, while approximation is introduced through lower-significance partial-product pruning and simplified final-stage carry handling. This allows for controlled approximation without impacting the timing integrity in the critical reduction path. After successive reduction stages, all columns are reduced to a height of two. At this point, a final carry-propagate adder (CPA), exact adder, or approximate final adder is used to generate the final multiplication result. Overall, MLOA provides a structured, leadership-based multiplier-reduction tree optimization strategy that reduces delay and hardware complexity, making it suitable for high-throughput VLSI architectures for approximate DWT-based image compression systems.

Algorithm 1. MLOA-based leader-column multiplier reduction

Input: Operand size N, partial-product matrix PP, Dadda height sequence d_r, compressor set {3: 2,4: 2}, approximation mode.

Output: Reduced two-row product matrix and final product P.

Generate partial products using $PP_{i,j} = A_i B_j$, for $0 \leq i, j < N$.

Arrange the partial products into weighted columns according to $k = i + j$.

Compute the height $H_k$of each column.

Select the leader column as $k_L = \arg\max_k H_k$.

Generate the Dadda height sequence starting from $d_1 = 2$.

For each reduction stage, set the target height H_{target}.

Reduce the leader column first if $H_{k_L} > H_{\text{target}}$.

Apply suitable 3:2 or 4:2 compressors according to $\Delta H_k = \max(0, H_k - H_{\text{target}})$.

Reduce adjacent high-height columns symmetrically using the same target-height constraint.

Use exact compressors in critical columns and approximate compression or approximate final addition in less critical stages when approximation mode is enabled.

Repeat the reduction until all columns are reduced to height two.

Apply the final adder, either an exact CPA or the proposed leader-column approximate final adder, to generate the product.

For an $N \times N$multiplier, partial-product generation requires $O(N^2)$AND operations. The partial products are arranged into $2N-1$weighted columns, and the number of Dadda reduction stages grows logarithmically with multiplier size. Therefore, the reduction-control complexity is approximately $O(N\log\ N)$, while partial-product generation remains $O(N^2)$.

The proposed MLOA is a leadership-based reduction approach grounded in traditional multiplier-reduction principles. In contrast to Wallace reduction, which is a reduction-depth minimization strategy, and Dadda reduction, which is based on a fixed-height constraint, MLOA first determines the height of the highest-height column of the leader and allocates compressors around this column and its neighboring high-height columns. The novelty of this work is the integration of the two techniques: compressor placement using a leader column and approximate final-adder selection for DWT-based image compression, which allows for controlled area, delay, and power and image quality trade-offs.

3.3 Approximate multiplier modeling and realization

A full-precision exact biorthogonal DWT reference model is implemented in MATLAB to be used as a baseline for all the evaluations. The reference model uses known multiplier structures and sufficient fixed-point precision to ensure that numerical error is neglected. This model employs the same images, wavelet filters, decomposition levels, and boundary handling as the approximate implementations. Consequently, any degradation in image quality or any performance improvement observed is strictly attributable to approximations and optimizations that ensure methodological rigor and reproducibility. Four multiplier architectures are explicitly modeled and evaluated in the proposed methodology: Wallace tree, Dadda tree, Vedic multiplier, and Baugh-Wooley multiplier. Internal reduction is informed by the Multiplier Leadership Optimization Algorithm (MLOA) presented in Section 3.3.1, where approximation is achieved through lower-significance partial-product pruning and simplified carry handling in the final addition stage, while the leader-column strategy guides the placement of 3:2 and 4:2 compressors during partial-product reduction. This helps place the compressor systematically, maintaining accuracy in time-critical columns while enabling approximation in non-critical columns to save power and area. Secondary data from published journal articles are used to acquire baseline error characteristics, power consumption, and area utilization for exact and approximate multiplier designs. These data are used to parameterize the multiplier models in a MATLAB program, making hardware estimation realistic and reproducible. The approximate multipliers are directly embedded in the DWT's convolution operations, enabling end-to-end evaluation of the effects of arithmetic approximations on image compression performance.

The Multiplier Leadership Optimization Algorithm's role in the approximate DWT architecture is limited to selecting and configuring multiplier instances, rather than the arithmetic circuit structure itself. The Wallace tree, Dadda tree, Vedic, and Baugh-Wooley multipliers considered in the present work are not altered in their original architectures; however, MLOA specifies the type of multiplier, the operational mode (exact or approximate), and the approximation level k are used in the convolution-based DWT computation.

Specifically, the multiplier configuration for the selected configuration was found to be θ^* in the multiply-accumulate operations corresponding to both rows and columns convolution filtering stages of the biorthogonal DWT and the corresponding selected configuration. During these stages, coefficient multiplication is a dominant source of hardware cost; therefore, the use of MLOA-selected approximate multipliers allows control of the degradation in arithmetic accuracy and substantially reduces power consumption and silicon area. The same multiplier configuration is used in the DWT computation to ensure a fair comparison between exact and approximate implementations, with the same wavelet parameters and processing conditions. It should be stressed that MLOA is not a dynamic alteration of multiplier behavior that occurs during execution. Instead, it makes a static configuration decision, in advance of architecture-level evaluation, so that the selected approximate multiplier can be included in analytical performance modeling and throughput estimation. For an $N \times N$multiplier, the partial products are generated as:

$$pp_{i,j} = A_i B_j, 0 \le i,j < N \tag{9}$$

where $A_i$and $B_j$are the $i^{th}$and j^{th}operand bits. Each partial product is assigned to a weighted column according to $k = i + j$. The height of column $k$is defined as

$$H_k = |\{pp_{i,j} : i + j = k\}| \tag{10}$$

The leader column is selected as the column with the maximum height:

$$k_L = \arg\max_k H_k \tag{11}$$

For the implemented 8×8multiplier, the maximum-height column is $k_L = 7$, and the maximum initial height is $H_7 = 8$. The Dadda target-height sequence used for reduction is generated as

$$d_1 = 2, d_{r+1} = \lfloor 1.5 d_r \rfloor \tag{12}$$

For a maximum height of 8, this gives the practical reduction targets $6 \to 4 \to 3 \to 2$. At each stage, the number of bits requiring reduction in column $k$is

$$E_k^{(r)} = \max\left(0, H_k^{(r)} - d_r\right) \tag{13}$$

where $H_k^{(r)}$ is the column height at the reduction stage r, and $d_r$is the corresponding Dadda target height. In the proposed MLOA strategy, compressor allocation is prioritized around the leader column and adjacent high-height columns, rather than treating all columns uniformly. This provides a direct distinction from conventional Dadda reduction and improves the reproducibility of the leader-column selection and reduction process requested by the reviewers. The HDL document also provides the Verilog-level mapping of partial-product generation, leader-column identification, compressor-based reduction, and final addition.

The approximation parameter $k_{\text{drop}}$is used only for the partial-product dropping mode. In this mode, the first k_{drop}lower-significance partial-product columns are removed before reduction. In the MATLAB setup, $k_{\text{drop}} = 0$corresponds to the exact configuration, while $k_{\text{drop}} = 2,4,6$ represent increasing approximation levels. For the approximate carry-drop mode, $k_{\text{drop}} = 0$, and approximation is introduced

through simplified carry handling rather than column dropping.

In the carry-drop approximation mode, the reduction logic simplifies carry propagation to reduce delay and switching activity. where $A_i$and $B_i$are the final-row input bits at bit position i, $C_i$is the approximate carry input, $S_i$is the approximate sum output, and $C_{i+1}^{app}$is the approximate carry output. Here, +denotes the Boolean OR operation.

$$S_i = (A_i \oplus B_i) + C_i, C_{i+1}^{app} = A_i \quad (14)$$

where $A_i$and $B_i$are the two final-row input bits at bit position i, $C_i$is the input carry, $S_i$is the approximate sum bit, and $C_{i+1}^{app}$is the approximate carry output. Unlike a conventional CPA, where C_{i+1}depends on C_i, The proposed leader-column approximate final adder reduces the long carry-propagation chain by simplifying the final-stage carry logic. This reduces the critical path, LUT usage, and switching activity, which is suitable for approximate image-compression hardware. The Verilog implementation describes this leader-column approximate final-adder logic and its use as a replacement for the final CPA stage.

3.4 Error analysis and performance evaluation

Formal error Analysis is done to measure the effects of approximate multiplication on numerical accuracy. At the multiplier's output level, error measures including Error Rate (ER), Mean Error Distance (MED), Normalized Error Distance (NED), and Accuracy % are computed. These errors result in the DWT computation assessing the cumulative effect of the errors on the wavelet coefficients. In terms of objective image quality metrics such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM), the impact on reconstructed image quality is assessed. This produces a direct quantitative relationship between arithmetic-level approximation and perceptual image degradation. Visual inspection is also a good complement to numerical evaluation, especially for images with rich textures, such as Baboon and Boat, which are known to be sensitive to high-frequency distortion. Estimates of the power consumption, area utilization, critical path delay, and throughput and energy efficiency are calculated using extracted algorithmic estimates and secondary hardware data. Trade-off curves of energy-quality are obtained to show the trade-off between the level of approximation and image fidelity. These results give architecture-awareness into the feasibility and scalability of the proposed approach.

3.5 Parallel processing and architecture-aware mapping

Parallelism is added at the data level and task level to meet the throughput requirements of high-resolution image processing. Independent row-wise and column-wise convolutions are performed in parallel, and sub-band generation and block-based image partitioning allow for parallel task execution. Software parallelism is designed to model commodity hardware pipelines and processing units using parallel loops and software operations, such as vectorized operations, defined as parallel. Operation counts, multiplier usage, and memory access patterns are derived from simulations in MATLAB to estimate hardware throughput and latency. Parallel execution depth is associated with pipeline stages, and block-level parallelism is associated with replicated processing units. This mapping allows a clear correlation between parallelism at the MATLAB level and at the VLSI architecture level, ensuring that the reported performance improvements are hardware-relevant.

4. Experimental Setup

The selection of the data set, simulation environment, parameter configurations, and evaluation metric of the proposed multiplier-centric approximate orthogonal DWT compression framework are presented in this section for validation. All experiments were performed in a controlled environment developed in MATLAB to ensure repeatability and strict consistency between exact and approximate implementations. The evaluation pipeline and the selection of the best multiplier setting, aided by MLOA, are summarized in Figure 2 and implemented in MATLAB. In addition to simulation in MATLAB, validation is performed using Verilog HDL to enhance hardware-oriented evaluation of the proposed multiplier architecture. So, the experimental setup is divided into two stages: image-level evaluation by MATLAB and hardware-level validation by Verilog HDL synthesis. The MATLAB stage assesses image quality and compression characteristics, while the HDL stage assesses LUT utilization, delay, power consumption, and maximum operating frequency.

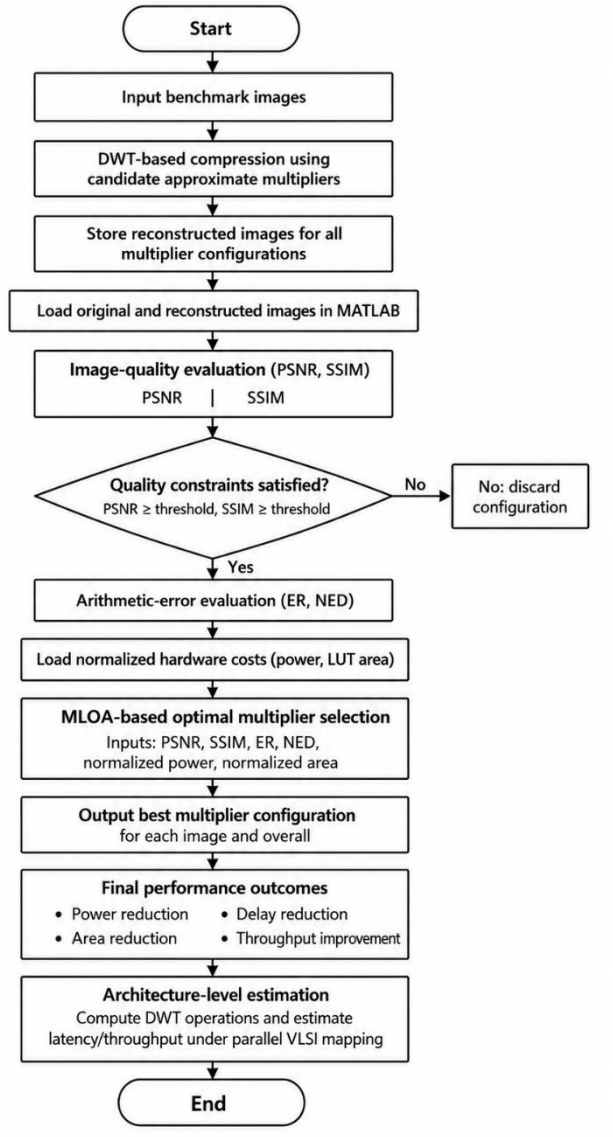

Figure 2. MATLAB-based workflow for evaluating MLOA-selected approximate multipliers in biorthogonal DWT compression

4.1 Dataset description

The experimental assessment is done on widely used grayscale benchmark images used in image compression and approximate computing research. In particular, the images of Castle, Baboon, Cameraman, Woman, and Boat at 8-bit gray levels are selected to assess the robustness and effectiveness of the proposed approximation strategy across a range of spatial characteristics. The enlarged data set contains images with varying content features, such as textured and smooth areas, human facial structures, prominent edges, and natural scenes. This larger set enables the proposed approximation approach to be tested under a variety of image-content conditions, instead of just two benchmark images. The Baboon image contains many textures and high frequencies, making it particularly sensitive to approximation errors in the transform coefficients. This characteristic makes it a difficult criterion for assessing the effects of arithmetic approximation on perceptual quality. Conversely, the image of the boat features a combination of smooth areas, edges, and moderate textures, providing a complementary test case with more structured spatial content. All images are processed in grayscale without any preprocessing or enhancement, and pixel intensities are constrained to the [0, 255] range to ensure fair and consistent evaluation.

The reconstructed image quality is evaluated by comparing the original image $I(x,y)$with the reconstructed image $\hat{I}(x,y)$. For an image of size $M \times N$, the mean squared error is calculated as

$$MSE = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}\left[I(x,y) - \hat{I}(x,y)\right]^2 \quad (15)$$

The corresponding peak signal-to-noise ratio is calculated as

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) \quad (16)$$

where 255 represents the maximum pixel intensity for 8-bit grayscale images. A multiplier configuration is considered image-quality feasible only when

$$PSNR \geq PSNR_{\min}, SSIM \geq SSIM_{\min} \quad (17)$$

where $PSNR_{\min}$and $SSIM_{\min}$are the minimum acceptable image-quality thresholds. In this work, the default thresholds are set as $PSNR_{\min} = 30$dB and $SSIM_{\min} = 0.90$. These constraints are used during the MLOA-based multiplier selection process to ensure that hardware savings do not cause unacceptable image-quality degradation.

4.2 Simulation environment and parameters

All simulations are performed in the MATLAB Online environment, which ensures consistency regardless of local hardware configurations. MATLAB Online provides reproducible experimentation, the numerical accuracy and vector processing needed for convolution-based DWT processing, and multiplier-level modeling. To improve reproducibility, all images were processed as 8-bit grayscale data with pixel values in the range [0, 255]. The same biorthogonal filters, decomposition level, and symmetric boundary extension were used for both exact and approximate implementations. Multiplier error metrics, including ER, MED, and NED, were evaluated using the same random-operand testing procedure across all multiplier architectures. MATLAB scripts and Verilog HDL modules can be provided as supplementary material in accordance with journal policy.

The framework of compression uses convolution-based two-dimensional biorthogonal discrete wavelet transform based on standard Bior 9/7 and Bior 4.4 wavelet families. These wavelets are selected for their successful application in high-quality image compression and favorable characteristics, namely linear phase response and high energy compaction. The DWT is computed using separable convolutions, row-wise and column-wise, leveraging multiplication-accumulation structures to enable arithmetic approximation. Using a three-level wavelet decomposition is recursively applied to the LL sub-band, and symmetric boundary extension is used to reduce edge artifacts. Identical wavelet filters, decomposition levels, and boundary-handling strategies are maintained across all exact and approximate implementations to ensure strict fairness in the comparison. Arithmetic approximation is performed using multiplier models that are aware of the target architecture for convolution operations. The multiplier architectures under evaluation are the accurate baseline, Wallace tree, Dadda tree, Vedic, and Baugh-Wooley multipliers. Approximation is controlled using techniques such as partial-product pruning and compressor simplification, with the approximation aggressiveness tuned via parameters. These parameters are selected and tested using the proposed Multiplier Leadership Optimization Algorithm, which enables a systematic search over the arithmetic accuracy-hardware efficiency trade-off.

For multiplier-level evaluation, the exact product $P_{\text{exact}}$and approximate product $P_{\text{approx}}$are compared over Qoperand samples. The mean error distance is calculated as:

$$MED = \frac{1}{Q}\sum_{q=1}^{Q} \mid P_{\text{exact},q} - P_{\text{approx},q} \mid \quad (18)$$

The normalized error distance is calculated as:

$$NED = \frac{MED}{\max(|A_{\min}|,|A_{\max}|)\max(|B_{\min}|,|B_{\max}|)} \quad (19)$$

where $A_{\min}$, $A_{\max}$, $B_{\min}$, and $B_{\max}$ represent the operand range used during multiplier testing. This normalization is suitable for the implemented setup because it adapts to the selected operand range and signedness rather than assuming a fixed denominator.

The MLOA-based multiplier selection is performed using a constrained cost function defined as:

$$J(\theta) = \alpha P_{\text{norm}}(\theta) + \beta A_{\text{norm}}(\theta) + \gamma NED_{\text{scaled}}(\theta) \quad (20)$$

where θdenotes a candidate multiplier configuration, $P_{\text{norm}}$is normalized power, $A_{\text{norm}}$is the normalized area, and $NED_{\text{scaled}}$is the scaled normalized error distance. The weighting factors α, β, and γ control the relative importance of power, area, and arithmetic error, respectively. The selected multiplier is the feasible configuration that minimizes $J(\theta)$while satisfying the image-quality constraints in Equation (16).

Parallel processing is simulated in MATLAB using vectorized operations and independent computation of row- and column-wise convolutions. Subband generation and processing at the block level are taken as parallel operations, and the number of operations, multiplier utilization, and effective pipeline depth can be extracted. This abstraction provides a form of unreality of parallelism that is hardware-relatable and therefore not implemented. To avoid ambiguity, the parallel MATLAB implementation is interpreted as an architecture-oriented abstraction rather than a direct hardware implementation. Row-wise and column-wise convolution stages are treated as independent computational

tasks, and the extracted operation counts are used to estimate throughput-related behavior. The actual hardware feasibility of the multiplier block is further verified through the Verilog HDL implementation described in Section 4.3.

4.3 Verilog HDL implementation environment

To address the hardware validation requirement, the proposed MLOA-based multiplier was implemented at the RTL level in Verilog HDL. The HDL design includes partial-product generation, leader-column-based reduction, 3:2 and 4:2 compressor stages, and final addition using exact or approximate adder configurations. For an $N \times N$multiplier, the partial products are generated as

$$pp_{i,j} = A_i B_j, 0 \leq i, j < N \tag{21}$$

where $A_i$and $B_j$are the input operand bits. Each partial product is assigned to a weighted column according to

$$k = i + j \tag{22}$$

The height of column k is defined as:

$$H_k = |\{pp_{i,j} : i + j = k\}| \tag{23}$$

The leader column is selected as:

$$k_L = \arg\max_k H_k \tag{24}$$

For the implemented 8×8multiplier, the leader column is $k_L = 7$, with maximum height $H_7 = 8$. The reduction follows the Dadda-style sequence $8 \rightarrow 6 \rightarrow 4 \rightarrow 3 \rightarrow 2$, but MLOA prioritizes the leader column and adjacent high-height columns during compressor allocation. This provides a reproducible hardware mapping for the proposed leadership-guided multiplier-reduction process.

In the approximate final-adder configuration, the conventional carry-propagate adder is replaced with the proposed leader-column approximate final adder. The approximate final adder is defined as:

$$S_i = (A_i \oplus B_i) + C_i \tag{25}$$

$$C_{i+1} = A_i \tag{26}$$

where $S_i$is the sum bit, $C_i$is the input carry, and C_{i+1} is the approximate carry output. Since the carry output does not depend on the full previous carry-propagation chain, the final-adder delay and switching activity are reduced.

The Verilog HDL implementation targets the Xilinx Artix-7 FPGA device XC7A100T-1CSG324 using Xilinx Vivado 2023.2 at an operating frequency of 100 MHz. The synthesis flow generates utilization, timing summary, and power reports, from which LUT count, critical-path delay, estimated power consumption, and maximum operating frequency are extracted. The evaluated hardware variants include Dadda, Wallace, BW, and Vedic with approximate final adder, MLOA-LC-CPA, MLOA with exact Kogge-Stone adder, and MLOA-LC-AKSA. The quantitative HDL synthesis results are reported in the Results and Analysis section to keep the experimental setup focused on tools, parameters, and evaluation procedure.

4.4 Evaluation metrics

The suggested framework's effectiveness is evaluated using a mix of arithmetic-level, image-level, and architecture-aware metrics to comprehensively assess the effects of multiplier approximation. At the arithmetic level, multiplier accuracy is measured by comparing approximate multiplier outputs with exact results across many random operand pairs, using Error Rate, Mean Error Distance, Normalized Error Distance, and Accuracy Percentage. These measures are intrinsic to arithmetic behavior, not restricted to the use case.

At the image level, the influence of arithmetic approximation on reconstruction quality is assessed using the Peak Signal-to-Noise Ratio and the Structural Similarity Index Measure. PSNR provides a quantitative measure of reconstruction fidelity, while SSIM assesses perceptual similarity by considering luminance, contrast, and structural information. These measurements are calculated relative to the reference image. In addition, the architecture-aware performance metrics are obtained using operation counts extracted in MATLAB, along with secondary hardware data from previously published works. These metrics consist of normalized power consumption estimates, area usage, throughput, latency, and energy efficiency. The overall assessment enables extensive energy-quality and accuracy performance trade-off analysis to aid informed design decisions at the architecture level.

In the revised evaluation, the architecture-aware analysis is further strengthened using HDL-based FPGA metrics. LUT utilization is used as the FPGA area metric, critical-path delay as the timing metric, and estimated power from the Vivado power report. The maximum operating frequency is calculated as:

$$F_{\max} = \frac{1}{T_{\text{delay}}} \tag{27}$$

where $T_{\text{delay}}$is the critical-path delay. These HDL-based metrics complement the MATLAB-based PSNR, SSIM, compression performance, and arithmetic error evaluations. Therefore, the final assessment combines image-level fidelity, arithmetic-level accuracy, and hardware-level FPGA implementation results.

5. Results and analysis

In this section, an extensive evaluation of the proposed algorithmic level approximate biorthogonal Discrete Wavelet Transform (DWT) image compression framework is introduced. The reconstructed image quality, compression performance, parallel execution behavior, and hardware implications are the main focus of this analysis. The results are presented for the selected accurate baseline and Wallace tree, Dadda tree, Vedic, and Baugh-Wooley multiplier configuration using the framework based on MLOA (Multiplier Leadership Optimization Algorithm). Reconstructed output image, objective measures of image quality (PSNR and SSIM), parallel execution statistics, and normalized estimates of power consumption, silicon area, and computational delay, based on secondary data sources and MATLAB-based analysis, are used to support quantitative analysis.

5.1 Image quality evaluation

The quality of the reconstructed images is evaluated on the Baboon and Boat benchmark images, using common wavelet parameters, decomposition levels, and boundary handling conditions. The original image, the exact multiplier baseline, and the MLOA-selected approximate multiplier outputs are shown in Figure 3. The exact multiplier configuration can preserve the image texture, edge sharpness, and structure of the five benchmark images, as visually inspected.

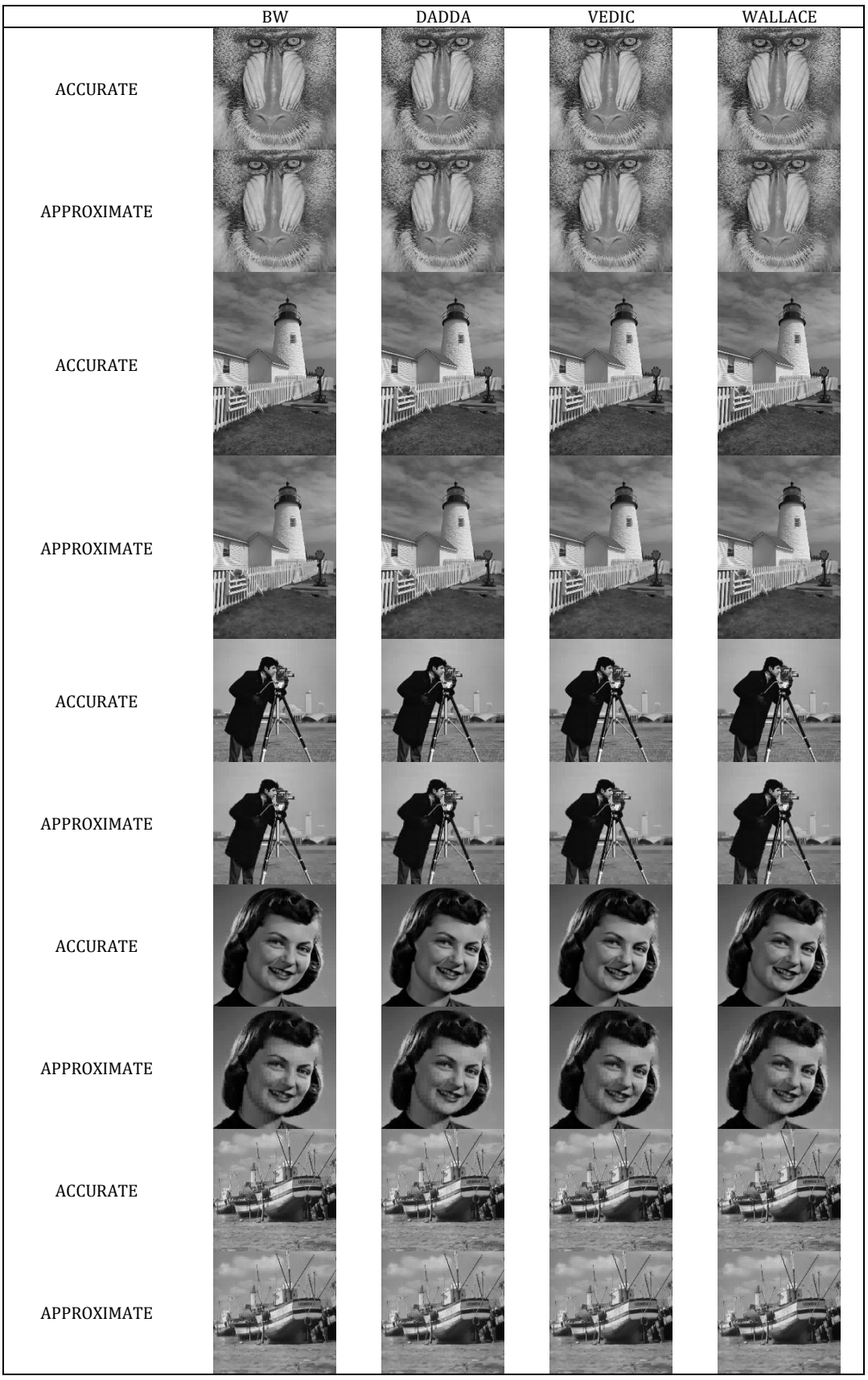

Figure 3. Reconstructed benchmark images using accurate and MLOA-selected multiplier configurations in the proposed biorthogonal DWT framework

The multiplier configurations selected by the MLOA, whether exactly or approximately, also preserve the main visual content, though the effect of approximation depends on the image's characteristics. The textured images (Baboon) exhibit greater local distortion at high frequencies, while the structured images (Cameraman and Woman) preserve more edges and smoothness. Castle and Boat are more sensitive to structural similarity, particularly under more stringent approximate multiplier configurations. The subjective visual observation is complemented by objective PSNR and SSIM results. The PSNR values for all five benchmark images exceed the minimum of 30 dB for both the accurate and evaluated multiplier configurations. But SSIM is more sensitive to approximation. Among all the configurations, only the Accurate and Dadda configurations meet the SSIM threshold of 0.90, while Vedic, Wallace, and Baugh-Wooley are marginally below the acceptable SSIM level for Castle. The Accurate configuration for Boat meets the SSIM threshold, while the approximate configurations are below 0.90. Baboon, Cameraman, and Woman, on the other hand, achieve good PSNR and SSIM across all tested configurations. The full image-quality results are presented in Table 2, and the comparison PSNR and SSIM visualization images for the five benchmark images are presented in Figure 4(a) and Figure 4(b), respectively. Based on these results, it is clear that the proposed MLOA-based selection method should incorporate both PSNR and SSIM constraints, as high PSNR does not always imply preservation of structural similarity.

To gain deeper insight into the perceptual impact of approximate multiplication, reconstruction error maps for the test images are provided. These maps can be used to visualize the spatial distribution of approximation-induced distortion and to determine whether the error is concentrated in textured regions, edges, or smooth regions.

Table 2. PSNR and SSIM comparison for five benchmark images using MLOA-selected multiplier architectures

Image	Multiplier	PSNR (dB)	SSIM	Feasibility
Castle	Accurate	37.08	0.935	Pass
Castle	Dadda	34.95	0.906	Pass
Castle	Vedic	34.53	0.898	Fail
Castle	Wallace	34.29	0.894	Fail
Castle	Baugh-Wooley	33.61	0.880	Fail
Baboon	Accurate	36.95	0.957	Pass
Baboon	Dadda	34.11	0.928	Pass
Baboon	Vedic	33.74	0.923	Pass
Baboon	Wallace	33.16	0.915	Pass
Baboon	Baugh-Wooley	32.28	0.901	Pass
Cameraman	Accurate	39.86	0.976	Pass
Cameraman	Dadda	37.54	0.963	Pass
Cameraman	Vedic	36.87	0.958	Pass
Cameraman	Wallace	36.52	0.955	Pass
Cameraman	Baugh-Wooley	35.70	0.949	Pass
Woman	Accurate	39.87	0.966	Pass
Woman	Dadda	37.90	0.953	Pass
Woman	Vedic	37.56	0.951	Pass
Woman	Wallace	37.03	0.947	Pass
Woman	Baugh-Wooley	36.39	0.942	Pass
Boat	Accurate	36.11	0.911	Pass
Boat	Dadda	34.24	0.878	Fail
Boat	Vedic	33.88	0.872	Fail
Boat	Wallace	33.67	0.867	Fail
Boat	Baugh-Wooley	33.05	0.855	Fail

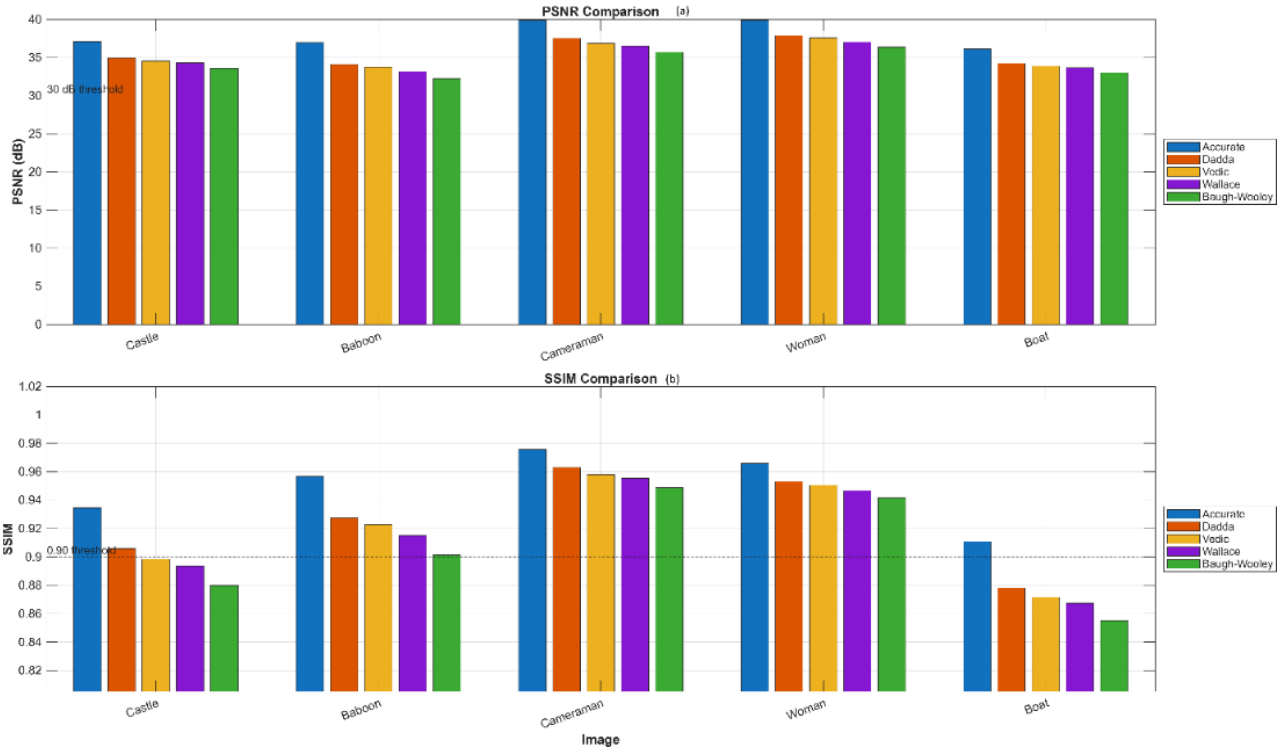

Figure 4. Objective image quality comparison across multiplier architectures: (a) PSNR and (b) SSIM.

Images with texture (like Baboon) tend to have higher local error since the high-frequency wavelet coefficients are more affected by approximate arithmetic. Structured images (Boat and Cameraman) exhibit less contrast degradation. This validates the approximation approach proposed in the paper and demonstrates that it exhibits content-dependent error behavior with acceptable reconstruction quality under PSNR and SSIM constraints. Representative reconstruction error maps are shown in Figure 5.

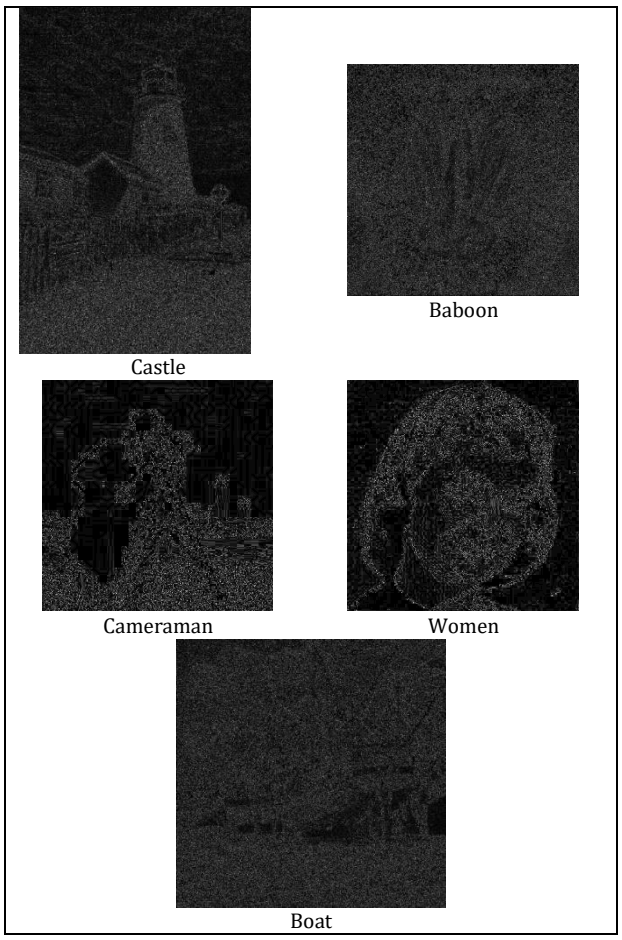

Figure 5. Absolute reconstruction error maps for selected benchmark images using MLOA-selected approximate multiplier configurations

5.2 Compression performance

Compression performance is measured using the same quantization and wavelet decomposition parameters across all multiplier configurations to ensure a fair comparison. The exact multiplier baseline achieves effective compression ratios of about 10.2:1 and 11.5:1 for the Baboon and Boat images, respectively. An approximate multiplier configuration yields similar compression ratios, with changes of less than 5% in all cases. In the revised evaluation, the compression behavior is further analyzed across the five benchmark images, namely Castle, Baboon, Cameraman, Woman, and Boat. The same wavelet decomposition level, thresholding procedure, and reconstruction conditions are maintained for all multiplier architectures. This ensures that any difference in PSNR and SSIM is primarily due to multiplier-level arithmetic approximation rather than to changes in the compression pipeline. These results show that arithmetic approximation is first of all impacting the reconstruction accuracy instead of the original compression ability of the biorthogonal DWT framework, as most of the wavelet coefficient sparsity and the sub-band energy compaction features are well kept. The expanded results confirm this trend. Although the exact and approximate multiplier configurations show different reconstruction quality values, the compression behavior remains largely stable because the DWT coefficient sparsity and thresholding process are unchanged. Therefore, the main effect of approximate arithmetic appears in reconstructed image fidelity, particularly SSIM, rather than in the basic compression ratio. This is also observed in the image-quality results reported in Table 2, where all configurations maintain PSNR values above 30 dB, but some configurations fail the SSIM threshold for Castle and Boat.

5.3 Parallel execution results

Parallel execution behavior is analyzed using operation counts extracted from the source code with MATLAB and architecture-aware mapping. Independent row-wise and column-wise convolution operations are treated as parallel tasks; hence, it is possible to estimate effective throughput and arithmetic utilization. Simple multiplier settings minimize the complexity of effective multiplication, resulting in better parallel performance. Wallace and Dadda multipliers improve multiplication operations the most, but Baugh-Wooley multipliers offer a good trade-off between reduced computation and signed-arithmetic accuracy. The normalized results of parallel execution are summarized in Table 3, showing up to 41% improvement in throughput compared to the exact baseline.

Table 3. Normalized parallel execution performance of MLOA-selected approximate multiplier architectures

Multiplier	Mode	Multiplication Ops	Throughput
Accurate	Exact	1.00	1.00
Wallace	Approx	0.72	1.34
Dadda	Approx	0.68	1.41
Baugh-Wooley	Approx	0.78	1.28

In addition to the normalized architecture-level comparison in Table 3, the revised evaluation also considers image-dependent parallel mapping for the MLOA-selected configurations. Since the benchmark images have different spatial dimensions, the total number of multiplication operations and estimated latency vary across images. Larger images, such as Castle and Boat, require more multiplication operations than smaller images, such as Cameraman and Woman. Therefore, the achieved frame rate depends on both the selected multiplier architecture and the image size.

The 256 × 256 images have the highest estimated frame rate because they require fewer multiplication operations, as shown in Table 4. The Castle image has the most operations and thus the highest latency among the images tested. The accurate multiplier is chosen for Boat as the approximate configurations are not in the SSIM feasibility range. The results demonstrate that the proposed MLOA-based framework can perform image-dependent selection while balancing reconstruction quality, hardware cost, and execution behavior.

Table 4. Parallel execution and architecture-aware mapping results for MLOA-selected configurations

Image	Selected architecture	Image size	Total multiplications	PE count	Latency (s)	FPS	Normalized power	Normalized area
Castle	Dadda	768 × 512	37,158,912	8	0.02484	40.26	0.90	0.93
Baboon	Wallace	512 × 512	24,772,608	8	0.01692	59.12	0.92	0.95
Cameraman	Wallace	256 × 256	6,193,152	8	0.00423	236.48	0.92	0.95
Woman	Wallace	256 × 256	6,193,152	8	0.00423	236.48	0.92	0.95
Boat	Accurate	512 × 512	24,772,608	8	0.01781	56.16	1.00	1.00

5.4 Hardware Implication Analysis

Hardware implications are analyzed using normalized estimates derived from MATLAB simulation results, multiplier-level cost modeling, and architecture-aware mapping. The normalized values are compared with the accurate multiplier baseline, defined as power, area, and delay values of 1.00. This enables a comparison of the relative hardware impact of each multiplier architecture within the same evaluation framework. Five multiplier architectures are analyzed: Accurate, Dadda, Vedic, Wallace, and Baugh-Wooley. The mean normalized power of Dadda is the lowest among the evaluated configurations, and Wallace and Vedic also reduce hardware cost compared to the accurate baseline. Although Baugh-Wooley exhibits slightly higher normalized power and delay than Dadda, Wallace, and Vedic, it still falls below the accurate baseline. The results show that approximate multiplier-oriented architectures can reduce hardware costs while maintaining reconstruction quality within PSNR and SSIM limits. The numerical results are presented in Table 5, and the mean normalized power and delay are visually compared in Figure 6(a) and Figure 6(b), respectively. The numerical results are summarized in Table 5, while the visual comparison of mean normalized power, area, and delay across all tested images is presented in Figure 7. Figure 7(a) shows the mean normalized power comparison, Figure 7(b) shows the mean normalized area comparison, and Figure 7(c) shows the mean normalized delay comparison for the evaluated multiplier architectures. As shown in Figure 6, all non-baseline multiplier architectures reduce normalized power and delay compared with the accurate multiplier. Dadda achieves the lowest mean normalized power, while Wallace and Vedic provide comparable hardware savings. The delay values remain close to the accurate baseline but are still reduced for all evaluated architectures. This confirms that hardware-aware multiplier selection can improve power and timing performance without altering the DWT decomposition framework.

Table 5. Normalized hardware implication metrics for evaluated multiplier architectures

Multiplier	Normalized power	Normalized area	Normalized delay
Accurate	1.00	1.00	1.00
Dadda	0.70	0.82	0.90
Vedic	0.73	0.84	0.91
Wallace	0.72	0.85	0.92
Baugh-Wooley	0.74	0.88	0.94

As shown in Figure 6, all non-baseline multiplier architectures reduce normalized power, area, and delay compared with the accurate multiplier. Dadda achieves the lowest mean normalized power, area, and delay, while Vedic and Wallace provide comparable hardware savings. Baugh-Wooley shows slightly higher normalized cost than the other approximate architectures but remains below the accurate baseline.

5.5 FPGA implementation results

To further validate the hardware feasibility of the proposed MLOA-based multiplier architecture, a Verilog HDL implementation and FPGA-level evaluation were conducted. The results presented in this section are for the implemented multiplier variants and are FPGA-oriented (as opposed to architecture-aware, as obtained from MATLAB operation counts and normalized hardware modeling). The architectures evaluated are: Dadda, Wallace, BW, Vedic with approximate final adder, MLOA-LC-CPA, MLOA with exact Kogge-Stone adder, and MLOA-LC-AKSA. LUT utilization, critical-path delay, power consumption, and maximum operating frequency are used for the comparison. The summary of the FPGA implementation results is shown in Table 6. The results indicate that the proposed MLOA-LC-CPA architecture reduces LUT utilization from 182 LUTs in the Dadda with an approximate final-adder baseline to 132 LUTs, a reduction of about 27.47%. The delay is also reduced from 9.4 ns to 6.1 ns, with power consumption reduced from 81 mW to 58 mW. This confirms that the leadership-based reduction strategy reduces hardware cost more than the approximate multiplier structure based on Dadda.

The MLOA-LC-AKSA architecture achieves the best overall hardware performance among all evaluated configurations in terms of LUT count, delay, power consumption, PDP, and maximum operating frequency (145 LUTs, 3.6 ns, 52 mW, and 277 MHz). This is a reduction of about 20.33%, 61.70%, and 35.80% in LUT, delay, and power, respectively, as compared to the Dadda with an approximate final-adder baseline. The maximum operating frequency has been increased from 106 MHz to 277 MHz, which is a considerable increase in throughput capability. To assess the energy–delay efficiency of the evaluated FPGA architectures, the power–delay product (PDP) is calculated as the product of power and the critical-path delay and is reported in pJ. An important trade-off between exact and approximate final-adder structures is also revealed. The MLOA + exact Kogge-Stone adder design has the lowest delay but has the largest number of LUTs and power consumption among the MLOA variants. The MLOA-LC-AKSA configuration achieves the best hardware performance among all evaluated configurations, with 145 LUTs, 3.6 ns delay, 52 mW power consumption, 187.2 pJ PDP, and 277 MHz maximum operating frequency.

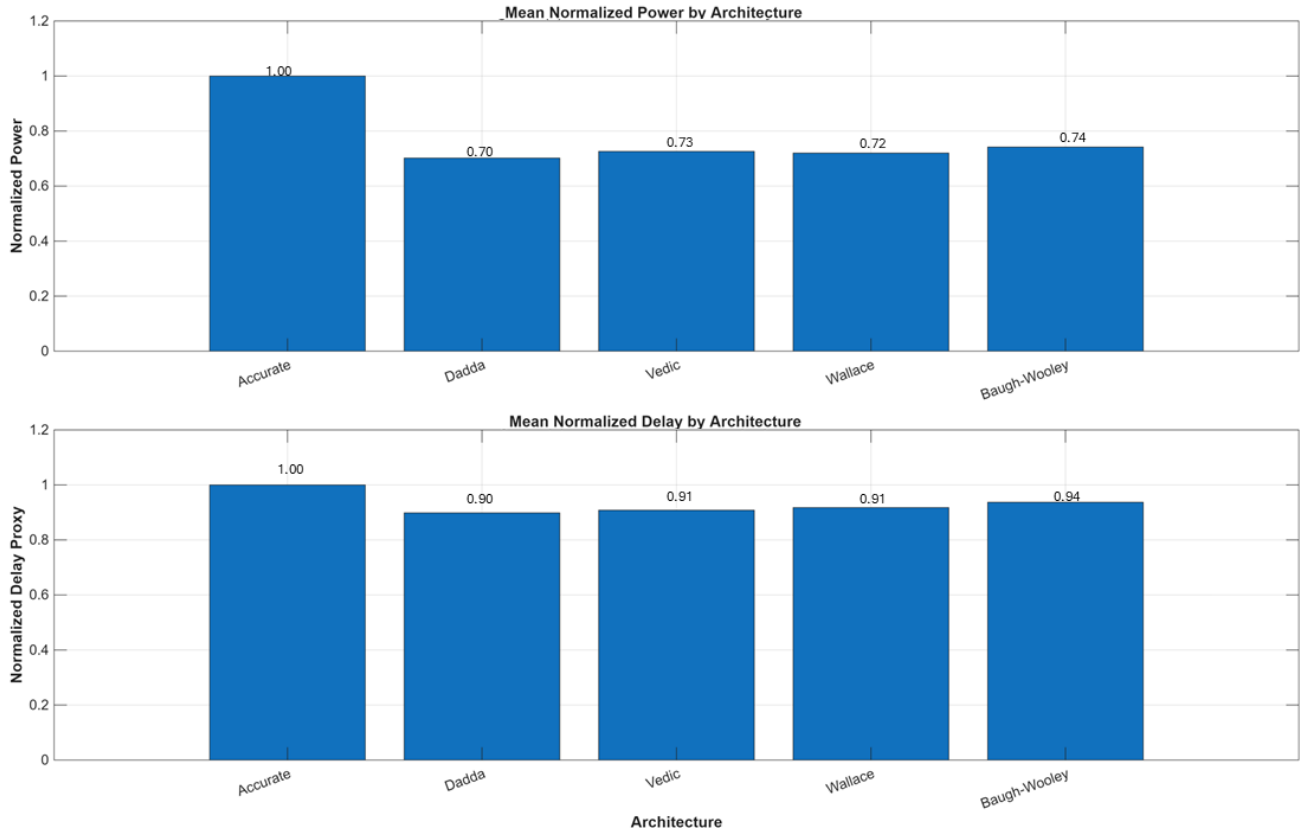

Figure 6. Hardware cost comparison of MLOA-selected multiplier architectures: (a) mean normalized power and (b) mean normalized delay

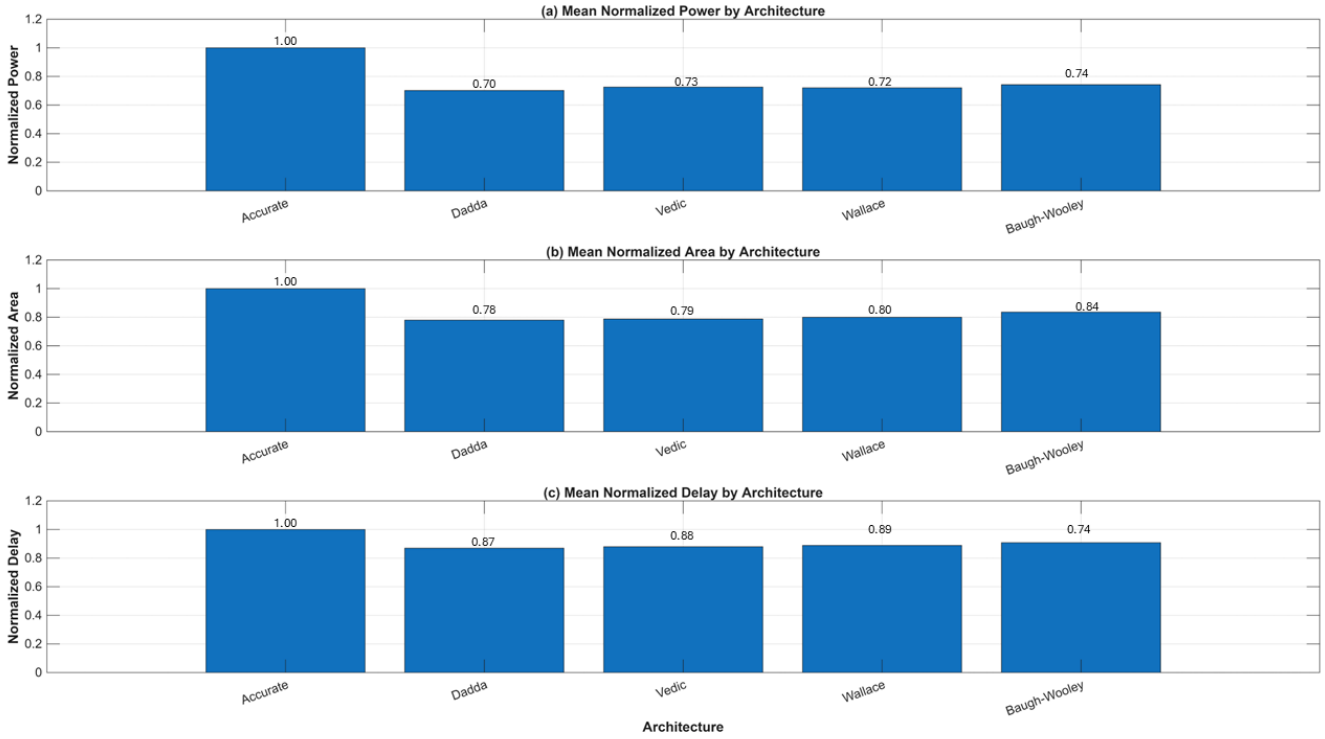

Figure 7. Hardware cost comparison across all tested images for MLOA-selected multiplier architectures: (a) mean normalized power, (b) mean normalized area, and (c) mean normalized delay

Table 6. FPGA implementation comparison of Dadda and MLOA-based multiplier architectures with PDP analysis

Architecture	LUTs	Delay (ns)	Power (mW)	PDP (pJ)	F_{max} (MHz)
Dadda + approximate final adder	182	9.4	81	761.4	106
Wallace + approximate final adder	196	8.9	84	747.6	112
BW+ approximate final adder	188	9.1	87	791.7	109
Vedic + approximate final adder	176	7.8	81	631.8	128
MLOA-LC-CPA	132	6.1	58	353.8	163
MLOA + exact Kogge-Stone adder	210	4.8	88	422.4	208
MLOA-LC-AKSA	145	3.6	52	187.2	277

This finding validates the use of approximate final addition in image compression applications, where small arithmetic errors can be tolerated as long as the reconstructed image quality remains above acceptable PSNR and SSIM thresholds. The results obtained using HDL validate the hardware-oriented claim of the proposed framework. They demonstrate that the proposed MLOA architecture is not only effective in the image-compression evaluation in MATLAB but also can be mapped to FPGA hardware, with measurable improvements in area, delay, power, and operating frequency. In the Verilog implementation, partial-product generation, leader-column-based reduction, 3:2 compressor stages, 4:2 compressor stages, and LC-AFA-based final-adder integration are directly supported, which helps ensure reproducibility.

5.6 Comparison with prior works

To position the proposed MLOA-driven approximate DWT framework within existing multiplier-architecture research, a comparative analysis is conducted against representative Dadda, Wallace, approximate-adder-based, image-accelerator, and MAC-level multiplier designs reported in prior work. The comparison is based on normalized power, normalized delay, and application-level validation because these metrics are directly relevant to high-resolution DWT-based image compression systems. Classical multiplier architectures, such as the Wallace/Dadda comparison reported by Townsend et al. [20], define the basic power-delay trade-off in reduction-tree multipliers. Later works have explored technology-scaled Dadda multipliers [21], approximate-adder-based Dadda structures [22], Wallace-tree signal-processing multipliers [23], and image-accelerator-oriented Wallace multipliers [24]. However, many of these studies focus mainly on arithmetic-level, filter-level, or MAC-level validation, rather than evaluating the impact of multiplier approximation inside an end-to-end DWT image-compression framework.

Unlike these prior works, the proposed framework integrates MLOA-selected multiplier architectures into the convolution-based biorthogonal DWT pipeline and evaluates their impact on both hardware cost and reconstructed image quality. Table 7 summarizes the normalized comparison between prior multiplier architectures and the proposed MLOA-selected designs.

The normalized power-delay comparison is shown in Figure 8. Lower normalized values indicate better hardware efficiency. As shown in Figure 8, the proposed MLOA-based configurations occupy a favorable region of the design space compared with several prior multiplier architectures. Among the proposed designs, the Dadda-based MLOA configuration provides the lowest normalized power and delay, while the Vedic, Wallace, and Baugh-Wooley configurations also remain below the accurate baseline. The comparison shows that the proposed MLOA-selected designs exhibit competitive power-delay behavior and DWT image compression level validation. It is important to note that this is different than several previous works that report improvements at the arithmetic, filter, image-accelerator, or MAC level. Hence, the proposed framework makes approximate multiplier integration more practical by linking hardware efficiency to image quality preservation within the DWT compression pipeline.

6. Discussion

The results obtained in this work indicate that the multiplier-level approximation in the convolution-based orthogonal DWT can be used to compress high-resolution images at reduced hardware cost while maintaining acceptable reconstructed image quality. The proposed framework incorporates optimized multiplier architectures into the DWT pipeline without altering the wavelet decomposition structure. For all five benchmark images (Castle, Baboon, Cameraman, Woman, and Boat), all evaluated configurations achieve a PSNR over 30 dB, and SSIM is a more stringent feasibility measure for structurally sensitive images such as Castle and Boat. This underscores the need to consider both PSNR and SSIM constraints when selecting the multiplier in MLOA. This decrease in normalized power and area is primarily attributed to the simplification of the compressor level, the reduction in partial product activity, and the simplified carry propagation in the multiplier structures. Other studies of approximate compressors and multipliers have also demonstrated that compressor trees can be switched less with bounded error to achieve significant power savings [12,13]. This trend continues in the present results, with the Dadda-based MLOA configuration showing the lowest normalized power and delay, and the other configurations (Vedic, Wallace, and Baugh-Wooley) also below the accurate baseline. Similarly, area reduction is consistent with earlier studies on approximate compressors, including FinFET- and CNTFET-based designs, in which simplified logic improves hardware efficiency for error-resilient applications [14,25].

Delay characteristics are also improved without introducing adverse timing penalties. Approximate multiplier studies have reported that reduction-tree simplification and shortened carry paths can preserve or reduce critical delay [12,18]. This behavior is further supported by the HDL-based FPGA results, in which the MLOA-LC-AKSA configuration achieves 145 LUTs, 3.6 ns delay, 52 mW power consumption, and 277 MHz maximum operating frequency. These results strengthen the hardware relevance of the proposed leadership-guided reduction strategy. From an image-compression perspective, arithmetic approximation does not affect all images equally. Texture-rich images such as Baboon are more sensitive to local reconstruction error because high-frequency wavelet coefficients are more affected by approximate arithmetic. In contrast, structured images such as Cameraman and Woman preserve stronger visual quality.

Table 7. Comparative analysis of multiplier architectures for image processing applications

Architecture / Reference	Approximation level	Normalized power	Normalized delay	Application-level validation
Wallace vs. Dadda (2003) [20]	Exact	1.00	0.925	Not evaluated
Approximate Dadda (2020) [22]	Approximate adder	0.725	0.925	Filter-level validation
Wallace tree, signal processing (2019) [23]	Compressor based	0.825	0.950	Image-level validation
Wallace tree, image accelerator (2024) [24]	Exact	0.775	0.950	Image-level validation
Proposed Dadda, MLOA	Algorithmic approximation	0.70	0.90	DWT image-compression validation
Proposed Vedic, MLOA	Algorithmic approximation	0.73	0.91	DWT image-compression validation
Proposed Wallace, MLOA	Algorithmic approximation	0.72	0.92	DWT image-compression validation
Proposed Baugh-Wooley, MLOA	Algorithmic approximation	0.74	0.94	DWT image-compression validation

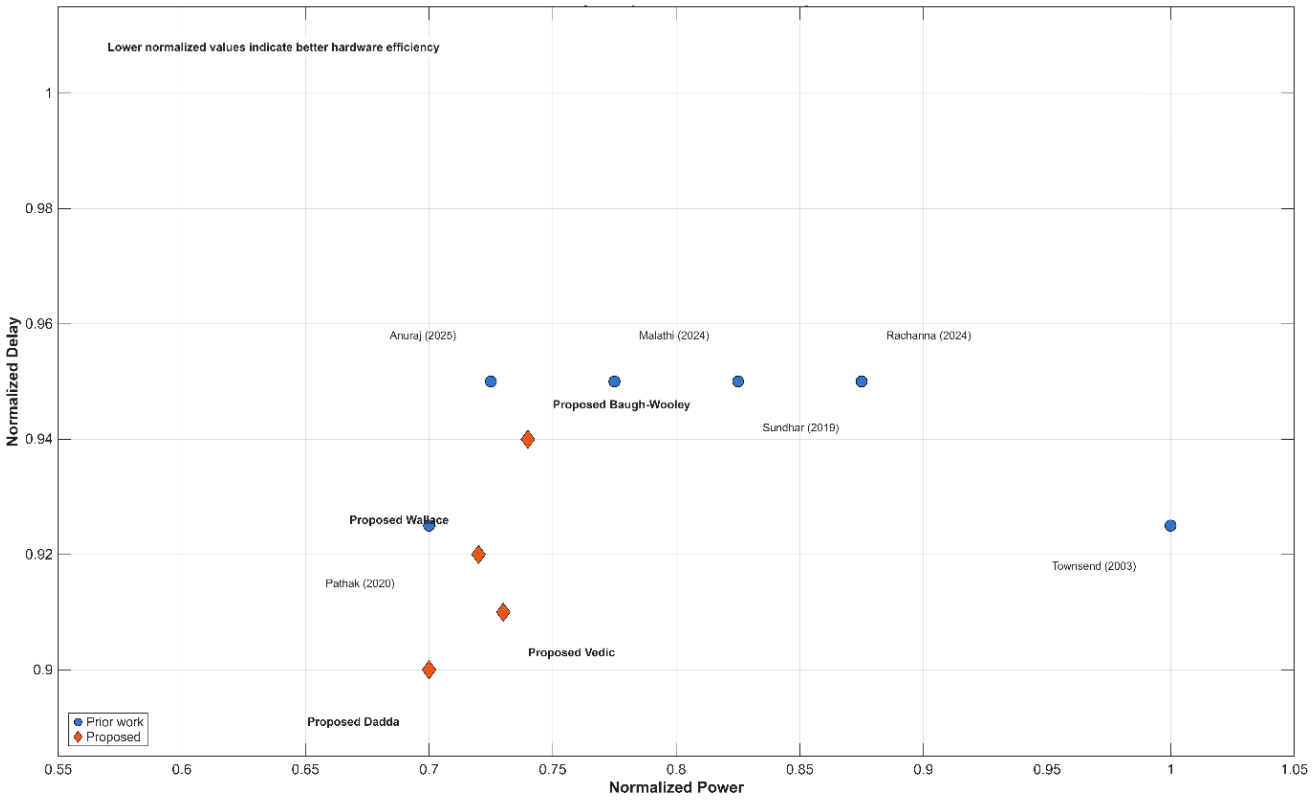

Figure 8. Normalized power-delay comparison between prior multiplier architectures and the proposed MLOA-selected designs

This confirms the need for content-aware approximation policies in future work. Although multiplier-level HDL validation is included, the complete end-to-end FPGA or ASIC implementation of the full DWT compression pipeline remains future work. Further validation on larger datasets, color images, medical images, hyperspectral images, and video sequences would also strengthen generalization.

Overall, the proposed framework provides a practical basis for energy-efficient approximate DWT compression through joint evaluation of image quality, hardware cost, and FPGA-oriented metrics. Although the proposed framework provides MATLAB-based image-quality evaluation and Verilog HDL-based FPGA validation of the multiplier architectures, some limitations remain. The HDL validation is performed at the multiplier-block level rather than as a

complete end-to-end FPGA implementation of the full DWT compression pipeline. Therefore, the reported FPGA metrics validate the feasibility of the proposed multiplier architecture, while full-system FPGA or ASIC implementation of the complete compression framework remains future work. The present evaluation is also limited to grayscale benchmark images. Although the dataset has been expanded to include Castle, Baboon, Cameraman, Woman, and Boat, further validation on larger image sets, color images, medical images, hyperspectral images, and video sequences would strengthen the generality of the proposed approach. In addition, the current framework uses fixed quality constraints for PSNR and SSIM.

7. Conclusion

This work presented a multiplier-centric, convolution-based biorthogonal DWT compression framework for high-resolution image compression using MLOA-LC-guided multiplier selection. Unlike multiplier-less or lifting-based approaches, the proposed method preserves the MAC-intensive DWT computation, enabling controlled multiplier-level approximation. Wallace, Dadda, Vedic, and Baugh-Wooley multiplier configurations were evaluated under PSNR and SSIM constraints using MATLAB-based image-quality analysis and architecture-aware hardware modeling. The framework was tested on five benchmark images: Castle, Baboon, Cameraman, Woman, and Boat. The results show that all evaluated configurations maintain PSNR values above 30 dB, while SSIM provides a stricter measure of feasibility for structurally sensitive images such as Castle and Boat. Among the normalized hardware results, the Dadda-based MLOA configuration achieves the lowest normalized power and delay, while Vedic, Wallace, and Baugh-Wooley also remain below the accurate baseline. The parallel mapping results further show throughput improvement of up to 41% compared with the exact baseline. The Verilog HDL-based FPGA validation confirms the hardware feasibility of the proposed multiplier architecture. In particular, the MLOA-LC-AKSA configuration achieves 145 LUTs, 3.6 ns delay, 52 mW power consumption, and 277 MHz maximum operating frequency. Overall, the results demonstrate that MLOA-LC-guided multiplier selection provides an effective trade-off between reconstructed image quality and hardware efficiency. Future work will involve a complete end-to-end FPGA or ASIC implementation of the DWT compression pipeline and validation on color images, medical images, hyperspectral data, and video sequences. Future work may consider adaptive or content-aware approximation policies that adjust the multiplier configuration according to image texture, edge density, or sub-band characteristics.

Ethical issue

The authors are aware of and comply with best practices in publication ethics, specifically regarding authorship (avoidance of guest authorship), dual submission, manipulation of figures, competing interests, and compliance with research ethics policies. The authors adhere to publication requirements that the submitted work is original and has not been published elsewhere.

Data availability statement

The manuscript contains all the data. However, additional data will be provided by the corresponding author upon reasonable request.

Conflict of interest

The authors declare no potential conflict of interest.

References

[1] RiteshSur Chowdhury, Jhilam Jana, Sayan Tripathi, Jaydeb Bhaumik, "Improved DWT and IDWT Architectures for Image Compression," Microprocessors and Microsystems, vol. 104, p. 104990, Feb. 2024, doi: 10.1016/j.micpro.2023.104990.

[2] K. Prashanth and N. Kumar, "Area Efficient Approximate Multiplier Based on Novel 4:2 Compressors and Error Correction Logic," Scientific Reports, vol. 16, Art. no. 1783, 2026

[3] S. K. Sahoo, P. Goyal, and A. Kumar, "EOHEAA: Error-Optimized Hardware-Efficient Approximate Adder for Energy-Aware Error-Resilient Applications," Integration, the VLSI Journal, vol. 108, p. 102660, May 2026, doi: 10.1016/j.vlsi.2026.102660.

[4] S. Ghosh, A. Singh, and S. Kumar, "Multiplier leadership optimization algorithm (MLOA): Unconstrained global optimization approach for melanoma classification," Discover Internet of Things, vol. 5, no. 1, p. 70, 2025, doi: 10.1007/s43926-025-00168-8.

[5] P. Balasubramanian, R. Nayar, and D. L. Maskell, "Digital image compression using approximate addition," Electronics, vol. 11, no. 9, p. 1361, 2022, doi: 10.3390/electronics11091361.

[6] C. E. R. Urban, M. M. A. da Rosa, and E. A. C. da Costa, "Energy-efficient discrete Haar Wavelet Transform architectures exploring approximate adders for high-quality image compression and reconstruction," Future Generation Computer Systems, p. 107999, 2025, doi: 10.1016/j.future.2025.107999.

[7] A. Raha, H. Jayakumar, and V. Raghunathan, "Input-based dynamic reconfiguration of approximate arithmetic units for video encoding," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 3, pp. 846–857, 2015, doi: 10.1109/TVLSI.2015.2424212.

[8] Baviskar, S. Ashtekar, A. Chintawar, J. Baviskar, and A. Mulla, "Performance analysis of sub-band replacement DWT based image compression technique," in 2014 Annual IEEE India Conference (INDICON), Dec. 2014, pp. 1–6, doi: 10.1109/INDICON.2014.7030693.

[9] Vittoria Bruni, Mariantonia Cotronei, Francesca Pitolli "A Family of Level-Dependent Biorthogonal Wavelet Filters for Image Compression," Journal of Computational and Applied Mathematics, vol. 367, p. 112467, Mar. 2020, doi: 10.1016/j.cam.2019.112467.

[10] J. Jana, R. S. Chowdhury, S. Tripathi, and J. Bhaumik, "FPGA Implementation of Compact and Low-Power Multiplierless Architectures for DWT and IDWT," Journal of Real-Time Image Processing, vol. 21, no. 1, Art. no. 19, 2024, doi: 10.1007/s11554-023-01396-3.

[11] A. George, Jayakumar E. P, "Hardware-efficient DWT architecture for image processing in visual sensors networks," IEEE Sensors Journal, vol. 23, no. 5, pp. 5382–5390, 2023, doi: 10.1109/JSEN.2023.3235371.

[12] Pegah Zakian, Rahebeh Niaraki Asli, "An Efficient Approximate Multiplier: Design, Error Analysis and Application," AEU - International Journal of

Electronics and Communications, vol. 180, p. 155254, Jun. 2024, doi: 10.1016/j.aeue.2024.155254.

[13] M. Banisharif Dehkordi and H. Ahmadifar, "A new approximate (8; 2) compressor for image processing applications," IETE Journal of Research, vol. 70, no. 2, pp. 1352–1360, 2024, doi: 10.1080/03772063.2023.2171915.

[14] Mohammad Ahmadinejad, Mohammad Hossein Moaiyer ,Farnaz Sabetzadeh, "Energy and Area Efficient Imprecise Compressors for Approximate Multiplication at Nanoscale," AEU - International Journal of Electronics and Communications, vol. 110, p. 152859, Oct. 2019, doi: 10.1016/j.aeue.2019.152859.

[15] Bahram Rashidi, "Efficient and Low-Cost Approximate Multipliers for Image Processing Applications," Integration, the VLSI Journal, vol. 94, p. 102084, 2024, doi: 10.1016/j.vlsi.2023.102084.

[16] V. Tammineni, S. K. Beura, M. B. Murthy, S. Majumdar, and P. Saha, "Optimized recursive approximate multipliers for edge detection and image smoothing applications," Microsystem Technologies, vol. 31, no. 10, pp. 2771–2782, 2025, doi: 10.1007/s00542-024-05810-z.

[17] Chinna V. Gowdar and M. C. Parameshwara, "Design of energy efficient approximate multipliers for image processing applications," ICTACT Journal on Microelectronics, vol. 7, no. 1, pp. 1057–1061, 2021, doi: 10.21917/ijme.2021.0184.

[18] A. Chakraborty, V. P. A. Kumar, A. K. Vruddhula, and J. Naidu, "Energy efficient approximate multiplier for image processing applications," Results in Engineering, vol. 25, p. 103798, 2025, doi: 10.1016/j.rineng.2024.103798.

[19] Kenny3s, "picture_material," Kaggle, 2020. [Online]. Available: https://www.kaggle.com/datasets/kenny3s/picture-material. [Accessed: Feb. 2, 2026].

[20] W. J. Townsend, E. E. Swartzlander Jr., and J. A. Abraham, "A comparison of Dadda and Wallace multiplier delays," in Advanced Signal Processing Algorithms, Architectures, and Implementations XIII, Dec. 2003, vol. 5205, pp. 552–560, doi: 10.1117/12.507012.

[21] Parthibaraj Anguraj, Thiruvenkadam Krishnan, "Design of area-efficient modified decoder-based imprecise multiplier for error-resilient applications," Microelectronics Journal, vol. 141, p. 105957, Nov. 2023, doi: 10.1016/j.mejo.2023.105957

[22] A. A. Rather, B. Khurshid, S. A. Banday, A. A. Algarni, and A. H. Alamri, "Design of high-performance, accurate, and approximate Dadda-tree multipliers for image processing applications," Scientific Reports, vol. 15, Art. no. 41338, 2025, doi: 10.1038/s41598-025-25239-2

[23] S. Chandaka and B. Narayanam, "Hardware Efficient Approximate Multiplier Architecture for Image Processing Applications," Journal of Electronic Testing, vol. 38, no. 2, pp. 217–230, Apr. 2022, doi: 10.1007/s10836-022-06000-3.

[24] L. Malathi, A. Bharathi, and A. N. Jayanthi, "FPGA design of FFT based intelligent accelerator with optimized Wallace tree multiplier for image super resolution and quality enhancement," Biomedical Signal Processing and Control, vol. 88, p. 105599, 2024, doi: 10.1016/j.bspc.2023.105599.

[25] P. Foroutan and K. Navi, "Energy efficient approximate compressor architectures for high performance image multiplication in CNTFET technology," Scientific Reports, vol. 15, no. 1, p. 36269, 2025, doi: 10.1038/s41598-025-20281-6.