

Implementation And Image-Quality Evaluation Of Approximate Multiplier Designs For Image Sharpening

N.K.Deepthi¹, Dr.S.Lakshmi²

¹ Dept of Electronics and Communication Engineering

² Associate prof, Dept of Electronics and Communication Engineering

^{1,2} Thirumalai Engineering College, Kanchipuram, Tamil Nadu, India.

Abstract- *Mention Multiplication is a computationally intensive operation in digital signal processing and image-processing hardware. Approximate arithmetic can reduce implementation cost when small numerical errors are tolerable, but the benefits must be assessed together with output quality. This paper documents an implementation-oriented evaluation of two approximate multiplier designs and their use in image sharpening. The supplied output contains RTL/simulation artifacts, implementation reports, and side-by-side image results labeled Exact Multiplier, Proposed Design 1, and Proposed Design 2. The reference study by Chakraborty et al. presents probability-aware approximate 4:2 compressors integrated into an 8×8 Dadda multiplier and reports a trade-off between power-delay product and image similarity [1]. The present manuscript uses that study as technical context while treating the uploaded screenshots as the available evidence for the implementation under review. The captured reports show different on-chip power summaries for the two design runs (0.222 W and 1.559 W), while the hierarchy reports show one versus two logic LUTs and six bonded I/O blocks. Because the screenshots do not unambiguously establish tool settings, device details, run-to-design mapping for every report, or quantitative image-quality scores, these values are reported descriptively rather than as a controlled comparative benchmark. The paper sets out a reproducible evaluation framework using error rate, mean error distance, structural similarity, timing, area, and power, and identifies the additional evidence required before publication claims can be finalized.*

Keywords- approximate multiplier; image sharpening; approximate computing; FPGA implementation; power analysis; structural similarity; image processing.

I. INTRODUCTION

Image-processing systems perform arithmetic repeatedly across pixels and filter windows. Multipliers are therefore important contributors to the area, latency, and energy consumption of digital signal-processing datapaths. In applications such as sharpening, smoothing, and image

multiplication, a limited numerical deviation may be visually acceptable, creating an opportunity to use approximate arithmetic. The central design problem is not simply to minimize logic; it is to balance hardware cost against numerical error and the perceptual quality of the processed image.

A multiplier typically forms partial products, reduces them through compressor stages, and adds the remaining rows. Compressor logic strongly affects the reduction network. Chakraborty et al. [1] proposed two approximate 4:2 compressor designs using input reordering and evaluated them in an 8×8 Dadda multiplier. Their results illustrate that an accuracy-oriented design and a power-delay-oriented design can have different strengths. The present work examines implementation outputs supplied for two designs and focuses on the image-sharpening use case.

The contributions of this manuscript are deliberately implementation-focused: (i) documenting the available design artifacts and implementation summaries; (ii) establishing a fair comparison protocol for exact and approximate arithmetic; (iii) describing image sharpening as the application-level test; and (iv) identifying reporting gaps that must be closed before the numerical results can support a publication-ready conclusion. The underlying approximate compressor concept is attributed to the reference work and is not claimed here as a new invention.

II. LITERATURE STUDY

It's Approximate arithmetic has been investigated to reduce the hardware cost of multipliers used in image processing, neural networks, and other error-tolerant workloads. Momeni et al. [2] analysed approximate compressors for multiplication, while Akbari et al. [3] explored dual-quality compressors that allow accuracy to be configured. Esposito et al. [4] proposed multiplier architectures based on approximate compressors, and Edavoor et al. [5] studied dual-stage 4:2 compressors. Fang et al. [6] investigated an unbiased approximate 4:2 compressor. Other

work has explored error-recovery methods [7], mixed-logic compressors [8], and approximate compressor structures in VLSI datapaths [9].

The reference paper [1] places these methods in the context of 8×8 Dadda multiplication and image processing. It reports that selective approximation in lower-significance columns can reduce hardware cost while preserving more significant bits exactly. It also evaluates image multiplication, smoothing, and sharpening using mean structural similarity. These findings motivate the evaluation criteria adopted here; however, published values from [1] are not treated as measurements of the present implementation.

For image-quality evaluation, SSIM compares local luminance, contrast, and structure between a reference image and a processed image [10]. For implementation evaluation, power and timing results are meaningful only when the target device, constraints, synthesis/implementation stage, clock assumptions, and tool settings are disclosed. Thus, image-level metrics and hardware-level metrics should be reported together.

III. DESIGN AND EVALUATION METHODOLOGY

A. Design Variants

Three output classes are relevant to the evaluation: an exact multiplier baseline, Approximate Design 1, and Approximate Design 2. The supplied document includes screenshots with the labels “Design 1”, “Design 2”, and “Image sharpening output”, plus a comparison image that places the exact and approximate results side by side. The source artifacts do not provide a complete RTL listing or a full design specification. Accordingly, the implementation is described at the variant level; exact gate-level equations and the precise mapping of each report to a variant should be inserted from the original project files before submission.

B. Image-sharpening workflow

A controlled image-sharpening experiment should begin with a fixed input image and fixed pixel representation. The image is processed once using the exact arithmetic baseline and once using each approximate design. The output images must use identical filter coefficients, border handling, clipping/saturation rules, and output scaling. A visual comparison is useful for interpretation, but quantitative metrics are needed to support a reproducible claim.

The reference study [1] gives a sharpening-kernel example and compares approximate multiplier outputs with an

exact multiplier. For the present implementation, the exact kernel, coefficient precision, image dimensions, and image source are not documented in the supplied text. These details must therefore be recovered from the source code or experiment notes instead of being inferred.

C. Evaluation Metrics

The following metrics are recommended for a complete comparison. Let y_i be the exact output and $\hat{y}_i$ the approximate output for test case i , with N total test cases.

Table 1. Evaluation Metrics

Metric	Definition / reporting practice	Interpretation
Error rate (ER)	$100 \times \text{count}(y_i \neq \hat{y}_i) / N$	Fraction of outputs that differ from exact arithmetic.
Mean error distance (MED)	$(1/N) \sum y_i - \hat{y}_i $	Average absolute numerical deviation.
Mean relative error distance (MRED)	Average of $ y_i - \hat{y}_i / y_i $ over nonzero exact outputs	Relative deviation; define treatment of zero outputs.
SSIM / MSSIM	SSIM per image, with mean across images if multiple images are used	Structural similarity of processed images.
Area	Report LUTs/CLBs or cell area, stating the target and tool stage	Hardware-resource cost.
Power	Report estimated or measured power, unit, tool stage, and activity assumptions	Energy-related implementation cost.
Delay / timing slack	Report critical-path delay or constrained WNS with clock constraints	Timing feasibility and speed.
Power-delay product	Power $\times$ delay, with consistent units and comparable runs	Combined power/timing indicator.

D. Implementation Evidence

The implementation showing a power-analysis summary, hierarchy/resource summaries, timing summaries, and an image comparison. The evidence is transcribed in Section 4. Because the screenshots do not show all project metadata and some values appear in separate reports, the values should be treated as captured tool outputs rather than a verified apples-to-apples comparison. The manuscript does not infer a technology node, FPGA model, clock frequency, or measurement methodology that is not visible in the supplied artifacts.

IV. RESULTS AND DISCUSSION

A. Captured Hardware Summaries

The captured power summaries differ substantially. That difference cannot be attributed to the approximate

multiplier alone without verifying that the same device, constraints, switching-activity assumptions, clock definition, and implementation stage were used. A power tool may report static and dynamic components separately, and an incomplete or mismatched activity file can change the estimate. Therefore, the present results should not be phrased as proof that one design consumes less power until the reports are regenerated under matched conditions.

Table 2. Captured hardware summary

Evidence in supplied output	Visible value	Reporting note
Power summary design A	Total on-chip power: 1.559 W; dynamic: 1.332 W; device static: 0.227 W	The screenshot shows a high dynamic-power share. Exact run and design mapping should be confirmed.
Power summary design B	Total on-chip power: 0.222 W; dynamic: approximately 0.001 W; device static: approximately 0.221 W	The screenshot indicates very low reported dynamic power. Confirm same device, activity file, constraints, and report stage before comparison.
Hierarchy/resource summary	One run shows 2 CLB LUTs and 2 LUTs as logic; another shows 1 CLB LUT and 1 LUT as logic; both show 6 bonded IOBs	These are small screenshot-visible counts and should be verified against full reports and top-level constraints.
Timing summary	WNS appears as 6.260 ns in the visible setup summary; one hold summary shows 1.228 ns	The screenshots also show zero total negative slack. Include clock period and full timing report to interpret these values.

B. Image-sharpening output

The supplied output presents three processed images for visual comparison. At the displayed scale, the scene structure is broadly retained across the exact and approximate outputs. However, visual inspection alone does not establish a numerical similarity value, and the source file does not provide a legible SSIM/MSSIM result tied to these images. SSIM should be computed directly from the saved full-resolution outputs, using the exact-multiplier result as the reference, and the input/output data should be archived.



Fig. 1. Supplied side-by-side image comparison labelled Exact Multiplier, Proposed Design 1, and Proposed Design 2.



Fig. 2. Additional supplied image comparison for the exact and approximate multiplier variants.

C. Interpretation

The screenshots suggest that both approximate variants were synthesized and that an image-sharpening example was produced. They do not, by themselves, establish a general performance advantage. The one- versus two-LUT difference is too small to generalize and may depend on synthesis optimization, coding style, and the chosen device. Likewise, the power values are not comparable until the reports are tied to their respective designs and run settings. A publishable evaluation should include the exact baseline and both approximate variants run through the same tool flow, with repeated or otherwise validated power estimates and explicit image-quality metrics.

D. Reproducibility Checklist

Table 3. Reproducibility Checklist

Required item	Status in supplied material	Action before submission
RTL/source files and exact design equations	Not included as readable source	Attach source files and identify design variants.
FPGA/ASIC target and tool versions	Not fully identifiable from screenshots	State device part number, tool/version, and synthesis/implementation flow.
Clock and I/O constraints	Partial timing screenshots only	Provide constraint files and clock period.
Power methodology	Power summaries visible; settings unclear	Use matched activity files or clearly state vectorless estimation and report stage.
Image/kernel specification	Image output visible; full settings absent	Record image source, dimensions, kernel coefficients, arithmetic width, rounding and clipping.
SSIM/MSSIM and error metrics	No legible numerical scores in output file	Calculate from archived outputs and include scripts/data.
Repeated runs and raw reports	Not included	Attach complete reports and document how results were extracted.

V. CONCLUSION

This paper presents an implementation-oriented evaluation of approximate multiplier designs for image sharpening, highlighting the importance of balancing hardware implementation cost with computational accuracy and image quality. The supplied implementation reports and image comparisons demonstrate the feasibility of investigating approximate arithmetic in image-processing applications. The captured hardware summaries show variations in reported power consumption and logic resource utilization across the design runs. However, these observations require further verification under identical implementation conditions before any definitive conclusions can be drawn regarding power efficiency or hardware performance. Similarly, although the processed images retain broadly similar scene structures at the displayed scale, quantitative evaluation using Structural Similarity Index Measure (SSIM), Mean Error Distance (MED), and Error Rate (ER) is necessary to establish the accuracy of the approximate designs.

Future work will focus on validating the RTL implementations, conducting controlled synthesis and implementation experiments, and evaluating all design variants using consistent device configurations, timing constraints, and power-analysis settings. Comprehensive numerical image-quality measurements and arithmetic-error analysis will also be performed to establish the relationship between hardware efficiency and output fidelity. These improvements will provide a reproducible basis for identifying suitable approximate multiplier architectures for resource-constrained image-processing systems. Overall, the study establishes an evaluation framework for assessing approximate multiplier designs and emphasizes the need for reliable hardware measurements and quantitative image-quality analysis before confirming their practical advantages.

REFERENCES

- [1] A. Chakraborty, V. P. A. Kumar, A. K. Vrudhula, J. N. K, and S. Balamurugan, "Energy Efficient Approximate Multiplier for Image Processing Applications," *Results in Engineering*, vol. 25, article 103798, 2025. <https://doi.org/10.1016/j.rineng.2024.103798>.
- [2] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," *IEEE Transactions on Computers*, vol. 64, no. 4, pp. 984–994, 2015.
- [3] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Dual-quality 4:2 compressors for utilizing in dynamic accuracy configurable multipliers," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 4, pp. 1352–1361, 2017.
- [4] D. Esposito, A. G. M. Strollo, E. Napoli, D. De Caro, and N. Petra, "Approximate multipliers based on new approximate compressors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 12, pp. 4169–4182, 2018.
- [5] P. J. Edavoor, S. Raveendran, and A. D. Rahulkar, "Approximate multiplier design using novel dual-stage 4:2 compressors," *IEEE Access*, vol. 8, pp. 48337–48351, 2020.
- [6] B. Fang, H. Liang, D. Xu, M. Yi, Y. Sheng, C. Jiang, Z. Huang, and Y. Lu, "Approximate multipliers based on a novel unbiased approximate 4-2 compressor," *Integration*, vol. 81, pp. 17–24, 2021.
- [7] M. Ha and S. Lee, "Multipliers with approximate 4–2 compressors and error recovery modules," *IEEE Embedded Systems Letters*, vol. 10, no. 1, pp. 6–9, 2018.
- [8] V. K. Odugu, "An efficient VLSI architecture of 2-D finite impulse response filter using enhanced approximate compressor circuits," *International Journal of Circuit Theory and Applications*, vol. 49, no. 11, pp. 3653–3668, 2021.
- [9] K. M. Reddy, M. H. Vasantha, Y. B. N. Kumar, and D. Dwivedi, "Design and analysis of multiplier using approximate 4-2 compressor," *AEU – International Journal of Electronics and Communications*, vol. 107, pp. 89–97, 2019.
- [10] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [11] S. Balamurugan and P. S. Mallick, "Error compensation techniques for fixed-width array multiplier design—a technical survey," *Journal of Circuits, Systems and Computers*, vol. 26, no. 3, article 1730003, 2017.
- [12] Y.-J. Chang, Y.-C. Cheng, Y.-F. Lin, S.-C. Liao, C.-H. Lai, T.-C. Wu, "Imprecise 4-2 compressor design used in image processing applications," *IET Circuits, Devices & Systems*, vol. 13, no. 6, pp. 848–856, 2019.
- [13] A. Gorantla, "Design of approximate compressors for multiplication," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 13, no. 3, pp. 1–17, 2017.
- [14] L. H. Krishna, A. Sk, J. B. Rao, S. Veeramachaneni, and N. M. Sk, "Energy efficient approximate multiplier design with lesser error rate using the probability-based approximate 4:2 compressor," *IEEE Embedded Systems Letters*, 2023.
- [15] B. Rashidi, "Efficient and low-cost approximate multipliers for image processing applications," *Integration*, vol. 94, article 102084, 2024.

- [16] B. Rashidi, “APPAs: Fast and efficient approximate parallel prefix adders and multipliers,” *The Journal of Supercomputing*, vol. 80, no. 16, pp. 24269–24296, 2024.
- [17] N. Kavand, A. Darjani, S. Rai, and A. Kumar, “Design of energy-efficient RFET-based exact and approximate 4:2 compressors and multipliers,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 9, pp. 3644–3648, 2023.
- [18] C. Yan, X. Zhao, T. Zhang, J. Ge, C. Wang, and W. Liu, “Design of high hardware efficiency approximate floating-point FFT processor,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2023.
- [19] L. Sayadi, S. Timarchi, and A. Sheikh-Akbari, “Two efficient approximate unsigned multipliers by developing new configuration for approximate 4:2 compressors,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 4, pp. 1649–1659, 2023.
- [20] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice Hall, 1989.