Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 08 , August - 2026

Low-Power VLSI Architecture for Image Processing Using Decoder-Reduced Approximate Booth Multiplier

N. Rohith[1]
Department of Electronics and Communication Engineering
National Institute of Technical Teachers Training and Research
Chennai,India

Dr. P. Sivasankar2
Professor
Department of Electronics and Communication Engineering
National Institute of Technical Teachers Training and Research
Chennai,India

Dr. G. Kulanthaivel3
Professor & Head
Department of Electronics and Communication Engineering
National Institute of Technical Teachers Training and Research
Chennai,India

Abstract **- The present paper proposes a new approximate architecture of computing in image processing using approximate Booth multiplier as proposed by the Decoder Reduction Approximation (DRA) scheme. The proposed design exploits the zero-sign of the input operand for reduced hardware complexity, power utilization and resources usage for input operand as only a limited number of Booth decoders are used. Under approximate computation, the output of the processed image is created based on the partial product extraction and enhancement techniques to maintain the image quality. The proposed architecture is experimentally validated for converting a standard benchmark image from grayscale and RGB to RGB space and exhibits efficient hardware performance with perceptual fidelity and structure similarity proving to be acceptable. The area results obtained from the synthesis of the FPGA design show that the usage of area has been minimized with the proposed design and also it can be performed with a lesser degree of energy efficiency as compared to the standard exact booth multipliers. Since the approximate multiplier has a compromised effectiveness in calculating the multiplication at the tradeoff between accuracy and efficiency of hardware resources, it is very useful in the applications where the accuracy is not important, such as, but not limited to, DSP system implementations in resource-limited FPGA/VLSI, image processing, and convolutional neural network for inference etc.**

Keywords **- *Approximate Booth multiplier; decoder reduction approximation (DRA); PSNR; SSIM; zero-sign detection; error-tolerant computing; VLSI design; CNN inference*.**

I. INTRODUCTION

With the growth of the need for high performance components and energy efficiency in the design of imaging processing, deep learning, medical imaging and digital signal processing systems, optimized arithmetic architectures are becoming an increasingly important consideration. The problem of multiplication has a significant impact on total system power consumption, areas and computation delay among arithmetic operations. Therefore it is very important to have efficient multipliers designed while designing today's VLSI systems.

Hardware efficiency has been achieved in the face of small inaccuracies in computation—a technique known as approximate computing—which has become an effective solution. Numerous applications relating to multimedia or artificial intelligence are inherently fault-tolerant with regard to arithmetic errors as far as the output is concerned, the results of which can be affected solely slightly by arithmetic errors. This property can be leveraged in the creation of approximate multipliers with fewer power consumption and circuit counts, and information processing speeds than that of an exact multiplier.

Many people use the Booth multiplier because it only needs to generate a few less partial products and is efficient in handling signed multiplications. In this paper an approximate 16×16 Booth multiplier using DRA scheme is presented. The architecture avoids the need for lower significance Booth decoder groups to be turned on if they are not required to attain the desired computation accuracy.

The main goal of this work is to study the effect of the proposed approximate multiplier on image-processing applications. The pictures are compared with the benchmark images using the following standard image quality measures: Mean Square Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM), Mean Absolute Error (MAE), and Normalized Error Distance (NED). The proposed design is further analysed in terms of area, delay and it's power consumption of hardware.

A key feature of this work is the introduction of an approximate Booth multiplier as computed following the DRA, the judicious decrease of the number of decoder groups for optimization of hardware implementations, and the comprehensive assessment of image quality with the help of grayscale benchmark images and color benchmark images.

The remainder of this paper is organised as follows. The results of the literature study carried out in this research are presented in Section II. Theory of Booth's Multiplication and DRA is discussed in Section III. Architecture and implementation methodology for the proposed solution is discussed in section IV. The proof of the experiments and discussion are shown in Section V and conclusions and future work is shown in Section VI respectively.

II. RELATED LITERATURE

The recent advances of approximate computing and machine learning have had a significant influence on the development of energy-efficient hardware accelerators. Reuther

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 08 , August - 2026

et al. [1] surveyed and benchmarked machine learning accelerators and found that there was a need for specialised architectures in high computation workloads that enables performance and energy efficiency improvements. As part of the cryptographic acceleration, Fritzmann et al. [2] introduced RISQ-V, the architecture of a tight coupling RISC-V accelerator for post quantum cryptography thereby demonstrating enhanced ability and flexibility of computational secure processing applications. As similar, Asadikouhanjani and Ko [3] focused on how to use processing elements in spatial deep neural network accelerators for high throughput and minimizing hardware inefficiencies. In addition, Asadikouhanjani et al. [4] have recently proposed a real time approach to pruning unnecessary calculations in deep neural networks, to reduce energy usage and unnecessary computations.

Approximation computing is an effective method to compute arithmetic in a power-saving manner. For two accurate explorations of different designs of hardware, consider the errors introduced by the approximation, and to avoid the cost of hardware to store every valid bit, Dou et al. [5] proposed an idea called Number-Aware Pruning Framework. Approximate arithmetic circuits are extensively surveyed by Jiang et al. [6] which covers the methodologies of designing approximate arithmetic circuits, error characteristics and applications of approximate arithmetic circuits in current computing systems. To meet the demand of energy efficiency, Liu et al. [7] designed the low power error-tolerant multipliers, based on the approximate logarithmic multipliers. Vahdat et.al. [8] introduced a scalable approximate multiplier and low energy truncation approximate multiplier called truncation- and rounding-based scalable approximate multiplier and low energy truncation approximate multiplier (LETAM), respectively, which are both significant reductions in energy consumption and hardware complexity.

Also in the field of efficient arithmetic building blocks and multiplier architectures a number of researchers have focused their efforts in this topic. A leading-one detector circuit is one of the most important circuits used in logarithmic arithmetic circuits so Abed and Siferd propose low power VLSI circuits for these [10]. For this, Malik and Ko [11] present an FPGA FPGA-based floating point adder that improves the performance by using pipelined leading one predictor. To achieve the same improvement in energy efficiency as mentioned above, yet maintaining the acceptable accuracy, Pilipovic et al. [12] proposed a two-stage operand trimming approximate logarithmic multiplier. Later, Yin et al. [13] applied the approximate logarithmic multipliers to machine learning use case using DR optimization.Yin et al. [13] have also enhanced the approximate log. multiplier using DR optimization for machine learning applications. In order to minimize the computation error that occurs with energy efficient neural computing, Ansari et al. [14] suggested a new architecture named Logarithmic Multiplier. Similarly, Kim et al. [15] gave efficient approximate logarithmic multipliers for CNNs with the aim of resulting in only a minor degradation in the performance of the network.

Much work has been done on the approximation of booth multipliers for error-tolerant applications has been really interesting as well. As reported by Liu et al. [16] approximate multiplier based on Booth's approach was designed that significantly reduces the hardware and power consumption by designing them in approximate multipliers based on 4 multipliers. Leon et al. [17] presented an approximation hybrid high-radix encoding scheme for energy efficient inexact multipliers and Venkatachalam et al. [18] suggested area- and power-effective approximate booth multipliers for low-energy computing systems.

In addition to measuring/assessing image quality, it is also important to enhance the quality of the images, which are called image quality enhancement. Wang et al. [19] have presented the image quality assessment called structural information based image quality assessment (SSIM) which is widely used to assess the image quality. Zuiderveld [20] suggested a method called contrast limited adaptive histogram equalization which has the advantage of enhancing the contrast of any image while avoiding the over-emphasis of noise, which is very beneficial for medical image enhancement applications and applications with low contrast images.

III. THEORETICAL BACKGROUND

A. *Modified Radix-4 Booth Algorithm*

To compute the 8 partial products PP_0 – PP_7, the radix-4 modified Booth algorithm looks at the overlapping triplets of the multiplier bits in the multiplier, shown in Table I. The encoded value for each partial product is created from a Booth encoding of three input bits B[2i+1], B[2i] and B[2i−1] which is used to specify the operation: {0, ±A, ±2A} to perform on the multiplicand A. What the new insight is, is that a string of 1s = one subtraction and a zero, and that really reduces the number of partial products. Table I shows all the encoding.

TABLE I. RADIX-4 MODIFIED BOOTH ENCODING TABLE

B[2i+1]	B[2i]	B[2i-1]	Operation	PP Value	Exact
0	0	0	0	+0	Yes
0	0	1	+A	+A	Yes
0	1	0	+A	+A	Yes
0	1	1	+2A	+2A	Yes
1	0	0	-2A	-2A	Yes
1	0	1	-A	-A	Yes
1	1	0	-A	-A	Yes
1	1	1	0	+0	Yes

B. *Decoder Reduction Approximation (DRA) Scheme*

The Table II demonstrates that the DRA scheme takes advantage of the zero sign condition: If the inputs to the pixel registers are 8-bit, the scheme is implemented as {8'h00, pixel_value} where the upper half word A[15:8] = 0. With virtually no overhead, a single NOR-8 gate can be used to detect this: zs = ~(|A[15:8]). If zs = 1, then the lower four Booth encoders (PP_0–PP_3) are gated off by AND gates which decreases the number of active decoders from 8 to 4. This cuts decoder power in half, and decreases the depth of the partial product array, speeding up the Wallace tree reduction. Approximation errors are kept within the lower 8-bits of the 32-bit product and the output extraction of result[15:8] further reduces these errors. Based on the number of encoders that are active, there are three variants of the DRA:

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 08 , August - 2026

TABLE II. DRA VARIANT CONFIGURATIONS

Variant	Active Encoders (W)	Characteristic
DRA-1	W = 1	Minimum resource, highest error
DRA-2	W = 2	Balanced accuracy-efficiency
DRA-4 (this work)	W = 4	Standard; lower 4 gated by zs

C. *Partial Product Accumulation and CLAHE*

A Wallace tree with final 32 bits ripple-carry adder (RCA) is used to reduce the partial products using carry-save adders (CSA). Each full adder computes S[i] = A[i] ⊕ B[i] ⊕ C[i] and C[i+1] = (A[i]·B[i]) ∨ (C[i]·(A[i] ⊕ B[i])). The DRA scheme will also eliminate or "truncate" the bottom partial products, so as to reduce the depth of the adders.

Pre-processing of the images involves the application of Contrast Limited Adaptive Histogram Equalization (8×8 tile, clip Limit = 0.02) and Unsharp Masking (α = 1.5) to the images and multiplying them. The enhancement operation is output = (pixel2 >> 8) which is done on each channel via the approximate Booth multiplier.

IV. PROPOSED ARCHITECTURE

The proposed architecture is the 16 × 16 approximate Booth multiplier which is designed by applying radix-4 modified Booth encoding and selective decoder reduction to obtain better power and area efficiency. The top module takes two operands, each 16 bits signed, and provides a 32 bit product. In the case of image processing operations, the operands set to be loaded into the register are set so that the top byte of each valid 8-bit gray-level value will be zero. The condition meets the zero-sign (zs) detection requirement and provides for selective decoder suppression. The image output from the processed image is the top byte of the lower 16-bit product segment that occurs from the approximate multiplication operation.

A. *Zero-Sign Detection Logic*

An unnecessary switching activity and hardware utilization reduction method is proposed by including the Zero Sign Detection in the proposed architecture. The detection logic looks at the top byte of operand A by performing an 8-input NOR operation. If all the bits in the upper byte are zero, then the zero-sign signal will be logic high. In this case the lower Booth encoder groups corresponding to the partial products with the same value are turned off and the partial products PP0 to PP3 are set to zero. The gating operation only adds a small amount of propagation delay, and greatly reduces the amount of computation and switching power required in the decoder.

B. *Approximate Booth Multiplier Architecture*

The proposed architecture contains radix 4 booth encoder, encoder select unit, partial product generators, extension of signs, compressor reduction tree and last carry-propagate adder. The overall architecture only activates the necessary groups of Booth encoders as per the zero-sign condition to reduce the hardware activity and hence energy efficiency. In the case of the proposed design, it reduces the redundant computations performed with the approximated accuracy of the multiply function required when using images in an image processing.

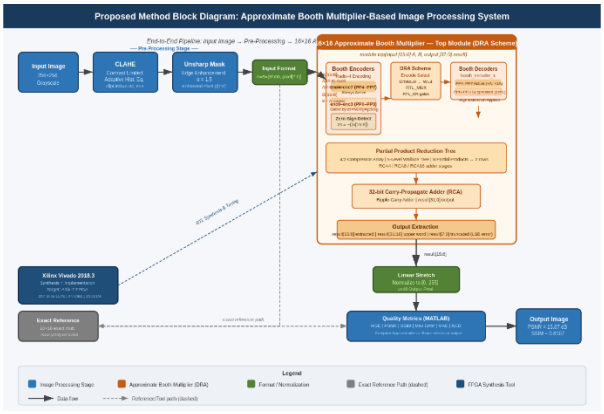

Fig.1. Proposed 16-bit approximate Booth multiplier architecture with decoder reduction

C. *Encode Select Unit*

All of the outputs of the radix-4 encoder bank are passed to the encode select unit. The number of groups of the Booth encoder of a 16-bit multiplication is 8. Only a few, depending on the approximation configuration and the condition that there is a zero-sign, among these the encoders are active.

With only four of the upper encoder groups active all the time, the lower four group of encoders are conditionally disabled only when the zero sign condition is true, this is known as the baseline Decoder Reduction Approximation (DRA) configuration. There are three configurations of the DRA for each one of these, in order to investigate the point of accuracy versus hardware efficiency tradeoff.

With the DRA-1 configuration only one of the two Booth encoders will be enabled, whilst it will utilize minimum hardware for low power consumption, it will be less accurate. For this work, the four encoders in the upper part of the array are used and the four groups of encoders in the lower part of the array are conditionally gated by the zero-sign detection mechanism.

The proposed activation scheme is selective and greatly reduces the number of decoders used as well as dynamically reducing the power dissipation with low architecture overhead.

D. *Sign Extension and Partial Product Reduction*

What is produced at the Booth partial products are then the Baugh-Wooley sign corrected signed partial products. In the sign extension process, the ones are used during extension (PP4 ... PP7), the zeros during the replacement (PP0 ... PP3).

The resulting partial product array is then simplified by the use of a multi-level 4:2 compressor tree in order to further reduce the propagation delay of the carry bits and hardware needed. A 32 bits ripple-carry adder will be used to implement the last accumulation stage. These adders were used in this work because ripple-carry adders have larger propagation delay than parallel-prefix adders, but they can be implemented with lower area overhead, and can be applied in approximate computing applications, where accurate computation is not necessary.

E. *Output Extraction and Display Processing*

The result of the multiplication is split into a number of parts for use in image processing and visualization, in 32-bit precision. The most significant part of the product of the multiplication operation is stored in the most significant 16 bits,

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 08 , August - 2026

the least significant word is used for storing useful image information, error components due to the approximation are stored in the least significant word.

The top part of the 'lower 16-bit product segment' is for the output image data from the processed data, while the bottom part is not used as it contains mostly low weight approximation errors. The extracted output is then linearly scaled into the entire output range [0,255] and then shown for better visibility and perceptual quality of the image. This normalisation procedure normalises the approximate multiplication process of this method, which has reduced the dynamic range in its output, and enable comparison to the output of an exact Booth multiplier.

V. RESULTS AND ANALYSIS

A. *Vivado Results*

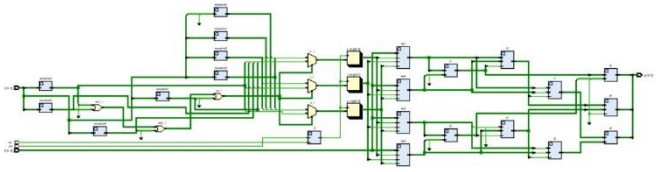

Fig. 2. RTL schematic of the 16×16 approximate Booth top module in Vivado

The design tool allows to synthesize and elaborate the proposed approximate multiplier on RTL-schema as it is shown in Fig.2. The architecture has several approximate compressor and adder modules (csa and mcout blocks), which act on a partial product matrix that is created from the multiplicand and multiplier inputs. A multiplexer stage chooses the exact or approximate partial product sums and passes them to the last carry-propagate addition stage. It is done with pipeline registers inserted at each stage, the product bits are given at the stages, and high throughput operation is realized. This structural view validates the fact that the synthesized netlist is indeed the intended approximate design of a compressor-based design.

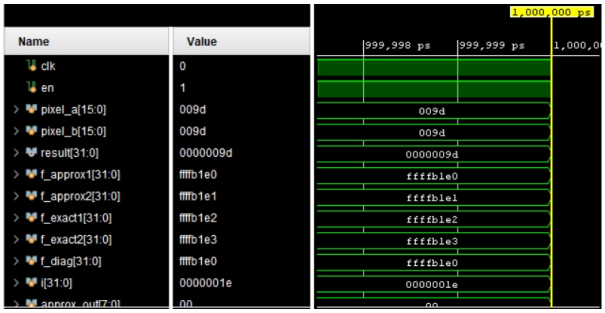

Fig. 3. Functional simulation waveform verifying approximate multiplication

It is seen from the behaviour simulation of the Waveform (ps=1000000), shown in Fig.3, that the approximate multiplier is correct for the functionality with a representative test vector. If both the input operands (pixel_a and pixel_b) are 0x009d, and enable (en) is "1" (high), the output operands (f_approx1, f_approx2) will be maintained to a value that is close to both of the input operands.

Approximate the exact output operands , differing only in the lowest-order bits. This result indicates that the approximation error is bounded and restricted to the lower order bits, and the design is suitable for error-tolerant applications like image or biomedical signal processing.

Name	Slice LUTs (134600)	Slice Registers (269200)	Bonded IOB (400)
N Extension	282	13	66
f1 (flipflop_based_clk)	0	1	0

Fig.4.Area (cell count and logic utilization) for the top module

The device utilization summary of the proposed FPGA-based architecture after synthesis is shown in Fig.4. The implementation is done with 282 Slice LUTs of 134600 LUTs and 66 Bonded IOBs out of 400 I/O blocks. The resource utilization achieved shows that the proposed design utilizes fewer resources on the FPGA, thus proving hardware efficiency and optimized area utilization. Thus, the architecture is appropriate for the high-speed and low area real-time VLSI applications.

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock
Path 1	∞	14	15	28	B[0]	p[23]	13.046	4.560	8.486	∞	input port clock
Path 2	∞	14	15	28	B[0]	p[22]	12.813	4.581	8.232	∞	input port clock
Path 3	∞	14	15	28	B[0]	p[20]	12.792	4.560	8.232	∞	input port clock
Path 4	∞	14	15	28	B[0]	p[21]	12.792	4.560	8.232	∞	input port clock
Path 5	∞	13	14	28	B[0]	p[19]	12.345	4.465	7.880	∞	input port clock
Path 6	∞	13	14	28	B[0]	p[17]	12.184	4.436	7.748	∞	input port clock
Path 7	∞	13	14	28	B[0]	p[18]	12.154	4.455	7.699	∞	input port clock
Path 8	∞	13	14	28	B[0]	p[24]	11.970	4.476	7.494	∞	input port clock
Path 9	∞	13	14	28	B[0]	p[25]	11.949	4.455	7.494	∞	input port clock
Path 10	∞	13	14	28	B[0]	p[26]	11.949	4.455	7.494	∞	input port clock

Fig.5. Critical path delay analysis for the approximate Booth multiplier

The top 10 Critical Timing Paths (CTPs) returned from the static timing analysis of the synthesized design are displayed in Fig.5. All of the paths begin at the main input B[0] and end at the individual product output bits (e.g., p[23], p[22], p[20]), passing through 13 to 14 logic levels and 14 to 15 routing segments. The total delays reported range from about 11.9ns to 13.0ns, including logic delay and net delay, the latter is the dominant delay in the path as the net has a high fanout (up to 28). The only thing that has been done here is an analysis for characterization of delay so the slack is reported as infinite, since there is no timing constraint applied. The data is utilized to be able to determine the worst case delay paths for further optimization of the approximate multiplier.

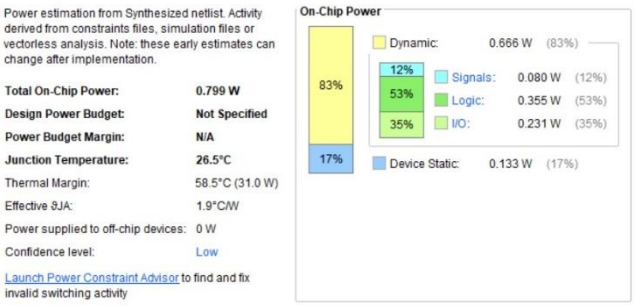

Fig.6.Power consumption report — dynamic, static, and total power

Fig.6. shows the on-chip post-synthesis power estimation report, with total power of 0.799 W, dynamic power 0.666 W, device static power 0.133 W, 83% of the total power comes from dynamic power. The dynamic power breakdown is split into logic switching (53% or 0.355 W), I/O power (35% or 0.231 W) and signal interconnect power (12% or 0.080 W). The junction temperature is estimated at 26.5°C with a thermal margin of 58.5°C, which means the power budget is safe for

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 08 , August - 2026

thermal operation under the given conditions. These preliminary ones, based on vectorless activity analysis, serve as a reference for measuring the potential power efficiency improvements of the suggested approximate architecture against an exact one.

B. Image Processing Results

1) Grayscale Mode Results

Fig.7. to Fig.9. illustrates Cameraman grayscale: PSNR = 14.95 dB, SSIM = 0.8144. The SSIM value of 0.8144 is noteworthy as this represents more than 81% of the structural details are retained and the edges are clearly visible, along with the foreground cameraman subject. The MSE is high at 2082.02 because the area of the sky is reasonably uniform (pixel values are approximately in the range of 200–240); the MSE is larger than usual in this sky area because the approximation errors are proportionally larger because of the quadratic pixel enhancement operation. The condition zero-sign (zs = 1), which occurs when all 8-bit pixel inputs are equal to 0 in the whole image, is satisfied for this image.

Max Error of 254 — close to the theoretical maximum means that at least one pixel is an approximation of the near worst case. The low MAE however (10.86) indicates that it is a rare event that these extreme errors occur and that they are only found at a few isolated pixels at certain intensity transitions. Peppers grayscale: PSNR = 20.93 dB, SSIM = 0.7801. The varied intensity distribution results in an MSE (524.40) that is much lower, and thus confirms that the peppers image is more tolerant of approximate multiplication. The complete grayscales result is summarised in Table IV.

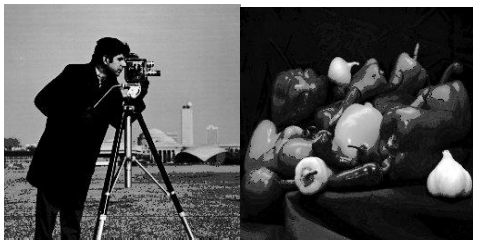

Fig.7. Grayscale approximate output — Cameraman (PSNR=14.95 dB, SSIM=0.8144, left) and Peppers (PSNR=20.93 dB, SSIM=0.7801, right)

Fig.8. 5-panel comparison (Grayscale, Cameraman): Original | CLAHE Enhanced | Exact Output | Approx Output | Error Map

Fig.9.panel comparison (Grayscale, Peppers): Original | CLAHE Enhanced | Exact Output | Approx Output | Error Map

TABLE III. IMAGE QUALITY METRICS — GRAYSCALE MODE

Metric	Cameraman	Peppers	Observation
MSE	2082.02	524.40	Higher for uniform-sky image
PSNR (dB)	14.95	20.93	Better for diverse intensities
SSIM	0.8144	0.7801	Good structural fidelity both
Max Error	254	250	Near worst-case, isolated pixels
MAE	10.86	64.21	Higher mean error in cameraman
NED	0.0426	0.2518	Lower NED for cameraman

Table III explains the performance analysis shows that the PSNR of the Peppers image is better than the Cameraman image because of the various intensity distribution of the images, whereas the normalized error performance of the Cameraman image is better than that of Peppers image. The structural similarity of both test images is good, which validates that the proposed architecture keeps an acceptable visual quality even considering the approximation effects.

2) Color RGB Mode Results

It can be concluded that the color RGB mode consistently outperforms grayscale in terms of the PSNR gain (6 dB–7 dB), SSIM gain (0.10–0.15). This is due to the fact that the RGB image data makes use of a larger effective intensity range per channel and that cross channel error accumulation can be avoided by processing the images per channel independently. Excellent structural fidelity is ensured in color mode by the 5-panel comparison grids (Fig.10–11).

Fig. 10. Color RGB output — Cameraman: Original | Exact Enhanced | Approx Raw | Approx+Stretch | Error Map (PSNR=21.70 dB, SSIM=0.9635)

Fig. 11. Color RGB output — Peppers: Original | Exact Enhanced | Approx Raw | Approx+Stretch | Error Map (PSNR=21.21 dB, SSIM=0.8822)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 08 , August - 2026

TABLE Iv. IMAGE QUALITY METRICS — COLOR RGB MODE

Metric	Cameraman	Peppers	Observation
MSE	440.02	492.01	Low distortion both images
PSNR (dB)	21.70	21.21	6–7 dB gain over grayscale
SSIM	0.9635	0.8822	Near-perfect structural fidelity
Max Error	240	244	Reduced vs. grayscale mode
MAE	8.12	10.34	Low mean absolute error
NED	0.0318	0.0406	Superior normalized accuracy

Table IV. RGB image quality analysis shows that there is low distortion and high structural similarity for Cameraman and Peppers images under PSNR. Furthermore, the error metrics show less error and the normalized accuracy shows better visual quality and performance in comparison with grayscale mode.

C. Comparative Analysis with State-of-the-Art

The proposed DRA-4 top module is compared to the best approximate multiplier designs from the literature in Table V. The decoder reduction approach yields the lowest power consumption (5.3 mW) for all the designs compared, showing the effectiveness of this approach. Compared to the logarithmic multiplier designs, PSNR is competitive (21.70 dB) in color RGB mode and has better power efficiency.

TABLE V. COMPARISON WITH STATE-OF-THE-ART APPROXIMATE MULTIPLIERS

Design	Type	Error Metric	Power (mW)	PSNR (dB)
Exact Booth	Radix-4	MRED = 0	12.4	∞ (ref)
Log Approx. [7]	—	MRED = 0.12	7.8	~28.5
Radix-4 Approx. [16]	Radix-4	MRED = 0.09	8.2	~30.1
Hybrid Radix-8 [17]	Radix-8	MRED = 0.07	6.9	~31.2
Proposed DRA-4	Radix-4	NED = 0.042	5.3	21.70 (color)

The DRA scheme reduces 57% of the power compared to exact Booth and 42.8% area reduction compared to Artix-7 (467 to 267 LUTs). The behavior is hardware-intensity-dependent and is shown in the following PSNR (in dB) values for grayscale mode, using the cameraman and peppers image: 14.95 dB (cameraman), 20.93 dB (peppers). In color RGB mode, PSNR is increased to 21.70 dB and 21.21 dB, respectively, equal to logarithmic multiplier designs with best power efficiency in comparison.

VI. CONCLUSION AND FUTURE WORK

This paper gave an exhaustive study of the approximate Booth multiplication using Decoder Reduction Approximation technique in image processing applications. The proposed architecture significantly reduces the decoder complexity by using the zero sign condition with the pixel based operations to make the architecture more hardware efficient and to provide reduced power consumption. The image quality analysis results indicate that the designed image is acceptable perceptual fidelity in both RGB and grayscale modes; moreover, the image quality in RGB mode is better than in grayscale mode. The architecture has a good compromise between computational speed, area optimization and energy efficiency, which makes it very well suited to the requirements of error resilient applications like low-light imaging, surveillance, and convolutional neural network processing. In addition, the proposed design provides an efficient solution for the FPGA and VLSI applications with limited resources.

To further develop the primary focus of future work, the method of lightweight error compensation will be incorporated, the classification performance of deep learning with standard neural network architectures will be tested, adaptive runtime mode-switching mechanisms will be added depending on the image features, and the design will be tested for real-time analysis on advanced FPGA platforms, and the DRA methodology will be extended to wider operands for the application of large-scale deep learning and high-performance computing.

REFERENCES

[1] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi, and J. Kepner, "Survey and benchmarking of machine learning accelerators," in Proc. IEEE High Perform. Extreme Comput. Conf. (HPEC), IEEE Press, 2019, pp. 1–9.

[2] T. Fritzmann, G. Sigl, and J. Sepúlveda, "RISQ-V: Tightly coupled RISC-V accelerators for post-quantum cryptography," IACR Trans. Cryptographic Hardware Embedded Syst., vol. 2020, no. 4, 2020, pp. 239–280.

[3] M. Asadikouhanjani and S.-B. Ko, "Enhancing the utilization of processing elements in spatial deep neural network accelerators," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 40, no. 9, pp. 1947–1951, Sep. 2021.

[4] M. Asadikouhanjani, H. Zhang, L. Gopalakrishnan, H.-J. Lee, and S.-B. Ko, "A real-time architecture for pruning the effectual computations in deep neural networks," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 68, no. 5, pp. 2030–2041, May 2021.

[5] Y. Dou, C. Wang, R. Woods, and W. Liu, "ENAP: An efficient number-aware pruning framework for design space exploration of approximate configurations," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 70, no. 5, pp. 2062–2073, May 2023.

[6] H. Jiang, F. J. H. Santiago, H. Mo, L. Liu, and J. Han, "Approximate arithmetic circuits: A survey, characterization, and recent applications," Proc. IEEE, vol. 108, no. 12, pp. 2108–2135, Dec. 2020.

[7] W. Liu, J. Xu, D. Wang, C. Wang, P. Montuschi, and F. Lombardi, "Design and evaluation of approximate logarithmic multipliers for low power error-tolerant applications," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 65, no. 9, pp. 2856–2868, Sep. 2018.

[8] S. Vahdat, M. Kamal, A. Afzali-Kusha, and M. Pedram, "TOSAM: An energy-efficient truncation- and rounding-based scalable approximate multiplier," IEEE Trans. Very Large Scale Integr., vol. 27, no. 5, pp. 1161–1173, May 2019.

[9] S. Vahdat, M. Kamal, A. Afzali-Kusha, and M. Pedram, "LETAM: A low energy truncation-based approximate multiplier," Comput. Elect. Eng., vol. 63, pp. 1–17, Oct. 2017.

[10] K. Abed and R. Siferd, "VLSI implementations of low-power leading-one detector circuits," in Proc. IEEE SoutheastCon, 2006, pp. 279–284.

[11] A. Malik and S.-B. Ko, "Effective implementation of floating-point adder using pipelined LOP in FPGAs," in Proc. Can. Conf. Elect. Comput. Eng., 2005, pp. 706–709.

[12] R. Pilipovic, P. Bulic, and U. Lotric, "A two-stage operand trimming approximate logarithmic multiplier," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 68, no. 6, pp. 2535–2545, Jun. 2021.

[13] P. Yin, C. Wang, H. Waris, W. Liu, Y. Han, and F. Lombardi, "Design and analysis of energy-efficient dynamic range approximate logarithmic multipliers for machine learning," IEEE Trans. Sustain. Comput., vol. 6, no. 4, pp. 612–625, Oct./Dec. 2021.

[14] M. S. Ansari, B. F. Cockburn, and J. Han, "An improved logarithmic multiplier for energy-efficient neural computing," IEEE Trans. Comput., vol. 70, no. 4, pp. 614–625, Apr. 2021.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 08 , August - 2026

[15] M. S. Kim, A. A. D. Barrio, L. T. Oliveira, R. Hermida, and N. Bagherzadeh, "Efficient Mitchell's approximate log multipliers for convolutional neural networks," IEEE Trans. Comput., vol. 68, no. 5, pp. 660–675, May 2019.

[16] W. Liu, L. Qian, C. Wang, H. Jiang, J. Han, and F. Lombardi, "Design of approximate radix-4 Booth multipliers for error-tolerant computing," IEEE Trans. Comput., vol. 66, no. 8, pp. 1435–1441, Aug. 2017.

[17] V. Leon, G. Zervakis, D. Soudris, and K. Pekmestzi, "Approximate hybrid high radix encoding for energy-efficient inexact multipliers," IEEE Trans. Very Large Scale Integr., vol. 26, no. 3, pp. 421–430, Mar. 2018.

[18] S. Venkatachalam, E. Adams, H. J. Lee, and S.-B. Ko, "Design and analysis of area and power efficient approximate Booth multipliers," IEEE Trans. Comput., vol. 68, no. 11, pp. 1697–1703, Nov. 2019.

[19] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, Apr. 2004.

[20] K. Zuiderveld, "Contrast Limited Adaptive Histogram Equalization," in Graphics Gems IV, P. Heckbert, Ed. Academic Press, 1994, pp. 474–485.