IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834, p- ISSN: 2278-8735.Volume 21, Issue 2, Ser. 2 (Mar – Apr 2026), PP 10-17
www.iosrjournals.org

Design And Analysis Of Energy Efficient Approximate Multipliers For Image Processing Using Verilog HDL Programming

Vaddi. Rajkumar, Alikani Vijaya Durga
IV Year Student, Department Of Electronics & Communication Engineering, Adikavi Nannaya University College Of Engineering, Adikavi Nannaya University, Rajamahendravaram, A.P-533296.
Guide & Assistant Professor, Department Of Electronics & Communication Engineering, Adikavi Nannaya University College Of Engineering, Adikavi Nannaya University, Rajamahendravaram, A.P- 533296.

Abstract
Numerous obstacles in enhancing the performance of computing systems have spurred the emergence of approximate computing. Extensive studies have been reported on approximate computing to develop high-performance, energy-efficient hardware designs tailored to error-resilient applications. In this brief, we proposed 8-bit approximate multipliers with 15 levels of accuracy using three techniques: recursive, bit-wise, and hybrid approximation using partial bit OR (PBO). Compared to the existing multipliers, investigated designs have significantly improved the area, power, delay, Power Delay Product (PDP), and Power Area Delay Product (PADP) by 41.68%, 73.16%, 35.57%, 72.65%, and 75.42% respectively on average. On resemblance with the accurate multiplier, the area, power, delay, PDP, and PADP were enhanced by 54.41%, 57.57%, 25.73%, 60.14%, and 74.33% correspondingly on average. Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) values surpassing (30 dB, 94%), (31 dB, 96%), and (26 dB, 95%) by applying them to benchmarks in image smoothing, edge detection, and image sharpening successively. Moreover, upon scrutinizing the efficacy of multipliers in hardware implementations of deep neural networks attaining the performance exceeding 95%. The obtained results confirm that suggested multipliers are well-suited for their widespread applications
Keywords: *Numerical Weather Prediction (NWP), Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), Long Short-Term Memory (LSTM), NOAA, NASA.*

Date of Submission: 11-04-2026 Date of Acceptance: 21-04-2026

I. Introduction

With the rapid advancement in image processing and artificial intelligence necessitates the integration of complex hardware in existing systems. Devices are normally characterized by limited computing capability and they are area, power, and speed-constrained [1]. To address these challenges, the emergence of inexact processing has become a prominent strategy. Due to their ability to withstand slight accuracy reductions with negligible impact on performance, approximate computing has proven highly effective in error-resilient applications such as neural networks, scientific computing, data analytics, image processing, speech processing, and multimedia applications [2], [3]. These applications require intense matrix multiplications. Therefore, it is imperative to develop hardware-efficient approximate multipliers. Harnessing error-robust capability presents a unique opportunity to achieve heightened efficiency in power, speed, and area by making calculated compromises on accuracy [4].

In recent years, improving the hardware efficiency of arithmetic circuits using approximate computing has become a key research area in digital circuits. Hence, enhancing their efficiency is crucial. Multiplication is more complex than addition. Extensive approaches have been investigated to obtain highly efficient approximate multipliers [5], [6], [7], [8], [9], [10], [11]. Approximate computing methods are divided into circuit, architectural, and software levels [3]. Circuit level includes voltage and frequency scaling [12]. Architectural level contains RISC-V ISA extension [13], [14], approximate accelerator and storage [15]. Software level mentioned Loop and Code Perforation [16]. However, the most favourable approach is imprecise numerical and logical hardware design [3], [17], [18], [19]. The latest work by Rashidi [20], where two types of multipliers were designed using a 4:2 approximate compressor and two 4-bit adders with minimizing computational cost was proposed. Multipliers are fundamental building blocks of digital systems. However, exact multipliers often impose considerable demands in terms of area, critical path, and power consumption. While extensively employed in neural networks, image processing, and various applications, their efficiency remains constrained. To address these limitations, approximate computing has gained widespread adoption and significantly improved diverse applications. In the realm of carry-overlook architecture, ignoring the carry introduces a notable increment in error. Hence, there is a pressing need for substantial modifications to existing carry- disregard methodologies.

This paper introduces a new insight: when ignoring the carry, the use of the OR gate for sum computation instead of the conventional XOR gate leads to a significant reduction in errors. Apart from this, the presented designs give exact results for any number of carry disregard bits if any operand of multipliers value is 2n (nϵW). An intriguing aspect lies in the fact that elements of filters like the Sobel, Prewitt, Laplacian, and numerous others can all be expressed as 2n . Thus, employing proposed multipliers for edge detection and image sharpening invariably yields accurate results.

II. I.Iterature Survey

"Booth encoding-based energy efficient multipliers for deep learning systems". H. Haider and S.-B. Ko.

Artificial intelligence on edge is a growing research field. In this brief, we propose a novel re-encoding scheme for reducing the size of the weights of deep neural networks (DNNs). The proposed re-encoding scheme exploits the Booth encoding scheme and the power-of-two (PO2) quantization to allow for very low energy computations during the inference of the neural networks with minimal loss in classification accuracy. We demonstrate the advantages of the proposed re-encoding scheme by computing a convolutional neural network (CNN) and a linear neural network on the proposed Extended Exact Multiplier and the proposed PO2 Multiplier. Our proposed PO2 quantization and re-encoding method reduce the model size for the CNN by 30.77% and the model size of the linear neural network by 49.86%. Furthermore, our multipliers reduce the inference energy for CNN by 50.6% and for the linear neural network by 90.1%. The PO2 Multiplier is proposed for the sensor-end computation of the linear neural network with a 77.32% reduction in the area relative to an exact Booth multiplier and it reduces the inference energy consumption of the linear neural network by 93.2% when compared to the unmodified exact multiplier. Our proposed scheme can be used to improve the energy consumption during inference for most Booth multipliers with minor modifications to the re-encoding signal arrangements. We also demonstrate that the proposed re-encoding scheme paired with the proposed multipliers outperforms all the existing designs in terms of resource utilization with a minimal impact on the inference accuracy of the neural networks.

"A retrospective and prospective view of approximate computing [point of view,"W. Liu, F. Lombardi, and M. Shulte.

Computing systems are conventionally designed to operate as accurately as possible. However, this trend faces severe technology challenges, such as power consumption, circuit reliability, and high performance. For nearly half a century, performance and power consumption of computing systems have been consistently improved by relying mostly on technology scaling. As per Dennard's scaling, the size of a transistor has been considerably shrunk and the supply voltage has been reduced over the years, such that circuits operate at higher frequencies but nearly at the same power dissipation level. However, as Dennard's scaling tends toward an end, it is difficult to further improve performance under the same power constraints. Power consumption has been a major concern, and it is now an industry-wide problem of critical importance. In addition to power, reliability deteriorates when the feature size of complementary metal-oxide-semiconductor (CMOS) technology is reduced below 7 nm, because parameter variations and faults at advanced nanoscales become difficult to control and prevent. Thus, to ensure the complete accuracy of signals, logic values, devices, and interconnects, manufacturing and verification costs will increase significantly.

I. improve the performance. Here, we propose a methodology for designing approximate N-bit array multipliers based on carry disregarding. We evaluate and analyze the proposed multipliers both experimentally and theoretically. The proposed 8-bit multipliers,
II. the modified sorting technique. This article implements 8×8 and 16×16 multipliers by employing the proposed approximate compressors in TSMC 28 nm. The experimental results indicate that our designs have about 18% delay, 43%–52% area-delay product (ADP) reduction compared to the exact multiplier, and 20%–55% ADP optimization compared to compressors with the same accuracy. This article further verifies the efficacy of the proposed compressors through image blending and matrix multiplication applications.
III. also requiring smaller errors. Case studies for error-tolerant computing applications are provided.

III. Proposed System

This paper introduces 15 approximate multipliers using three approaches: i) a recursive multiplier utilizing 8×4 multipliers, aimed at precise or less error-resilient applications, guided by Figure 2; ii) bit- wise approximation designed for highly error-resilient scenarios, mentored by Figure 1; iii) a hybrid approximation utilizing two 4:2 approximate compressors to strike a balance between hardware efficiency and accuracy.

TABLE II

COMPARING ERROR DISTANCE: XOR GATE VERSUS OR GATE FOR SUM FUNCTION WITH CARRY DISREGARD

Input-1	Input-2	XOR result	OR result	Exact result	Err_i XOR	P (Err_i) XOR	Err_i OR	P (Err_i) OR
00	00	00	00	00	0	-	0	-
00	01	01	01	01	0	-	0	-
00	10	10	10	10	0	-	0	-
00	11	11	11	11	0	-	0	-
01	00	01	01	01	0	-	0	-
01	01	00	01	10	2	9/256	1	9/256
01	10	11	11	11	0	-	0	-
01	11	10	11	100	2	3/256	1	3/256
10	00	10	10	10	0	-	0	-
10	01	11	11	11	0	-	0	-
10	10	00	10	100	4	9/256	2	9/256
10	11	01	11	101	4	3/256	2	3/256
11	00	11	11	11	0	-	0	-
11	01	10	11	100	2	3/256	1	3/256
11	10	01	11	101	4	3/256	2	3/256
11	11	00	11	110	6	1/256	3	1/256

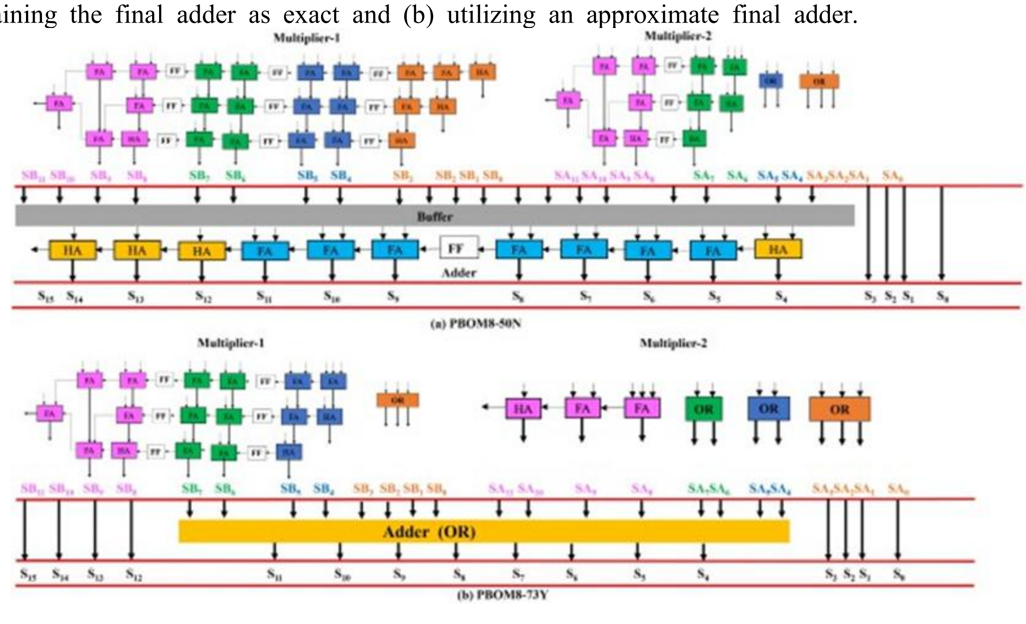
$$P_E = \sum_{i=1}^{n} P(Err_i)$$

$$E_{mean} = \sum_{i=1}^{n} P(Err_i)(Err_i)$$

Proposed Approximation for 8× 4 Multiplier

Figure 4 illustrates four types of 8 × 4 multipliers using PBO adders, named as PBO-3, PBO-5, PBO-7, and PBO-10. In all four variants, approximation begins from the second.Fig. 3. Schematic depiction of the approximate 8-bit unsigned multiplier design utilizing two exact/approximate 8 × 4 multipliers:

(a) Retaining the final adder as exact and (b) utilizing an approximate final adder.



Proposed Approximation for 8× 8 Multiplier Using 8× 4 Multiplier

Figure 3 showcases the proposed approximation schemes for 8×8 multipliers using two 8×4 multipliers and one adder. Seven approximate multipliers are introduced utilizing two different combinations of 8×4 multipliers with the exact adder, in which one design is depicted in Figure 3(a). Additionally, one approximate multiplier is designed using two approximate 8×4 multipliers and one approximate adder, as depicted in Figure 3(b). Exclusion of configurations such as PBOM8_10, PBOM8_20, PBOM8_40, PBOM8_41, PBOM8_42, etc. are done because these designs are not able to remove buffers between combinational paths. Higher configurations like PBOM8_77, PBOM8_107, and PBOM8_1010 are excluded because it is having complex designs for higher errors. In Table III, suitable combination of two distinct 8×4 multipliers and one adder is elucidated to further reduce area, power, delay, and error. While the architecture in Figure 2 has

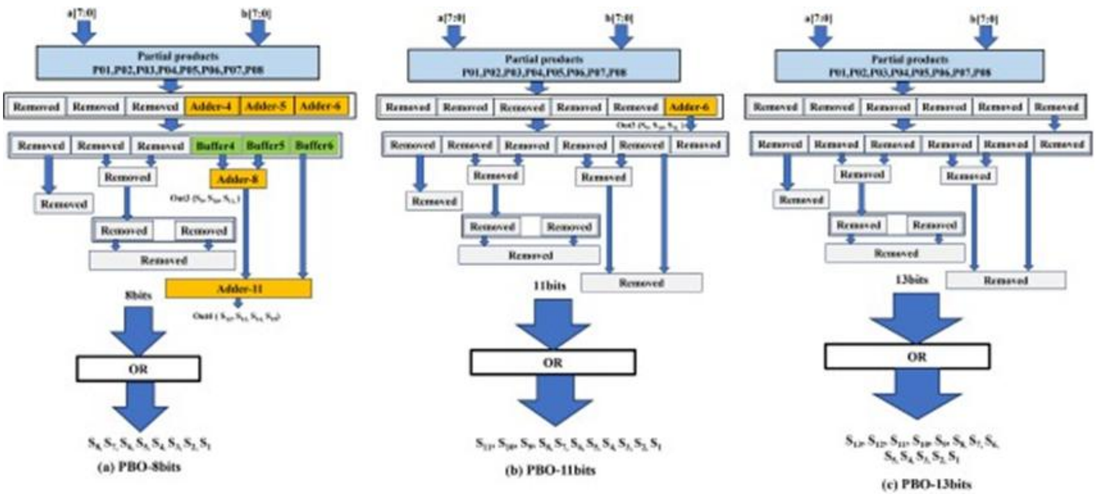


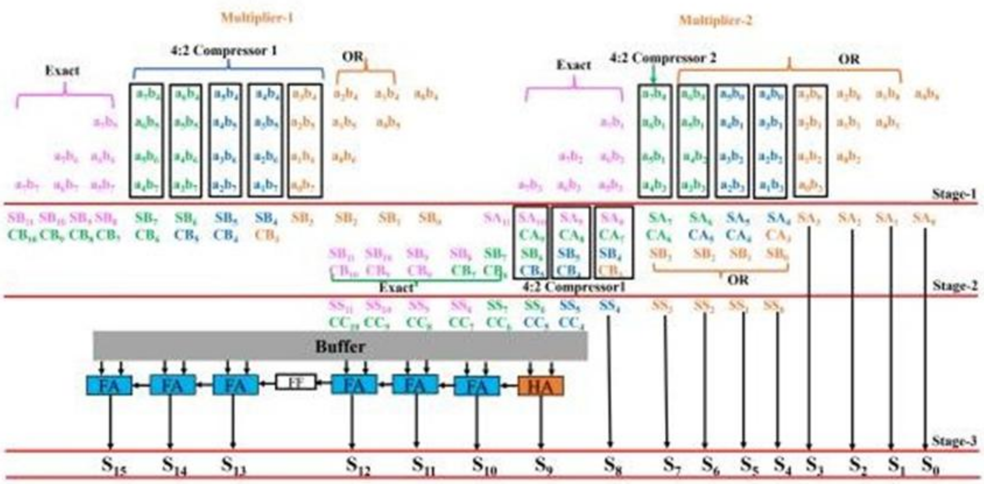
Fig. 5. The architecture of a bitwise approximation for 8-bit unsigned 8× 8 multipliers for high error resilient application.

TABLE III
PROPOSED 8 × 8 MULTIPLIER UTILIZING TWO EXACT/APPROXIMATE 8 × 4 MULTIPLIERS AND ADDER

Design PBO	Multiplier-2 Approximation	Multiplier-1 Approximation	Adder Approximation
PBOM8-30N	Column 2 to 4	Exact	Exact
PBOM8-50N	Column 2 to 6	Exact	Exact
PBOM8-33N	Column 2 to 4	Column 2 to 4	Exact
PBOM8-53N	Column 2 to 6	Column 2 to 4	Exact
PBOM8-73N	Column 2 to 8	Column 2 to 4	Exact
PBOM8-75N	Column 2 to 8	Column 2 to 6	Exact
PBOM8-105N	Column 2 to 11	Column 2 to 6	Exact
PBOM8-77N	Column 2 to 8	Column 2 to 8	Exact
PBOM8-107N	Column 2 to 11	Column 2 to 8	Exact
PBOM8-1010N	Column 2 to 11	Column 2 to 11	Exact
PBO-73Y	Column 2 to 8	Column 2 to 4	Column 5 to 11

and PBOM8_1010N. These three designs are prone to higher error due to the approximation of higher significant digits, particularly in Multiplier-1. Employing complex architectures for designing high-approximation circuits is not advisable. Therefore, the multiplier depicted in Figure 1 has been approximated for higher error-resilient applications.

Proposed Approximation for 8 × 8 Multiplier for High Error Resilient Application



Hybrid Multiplier

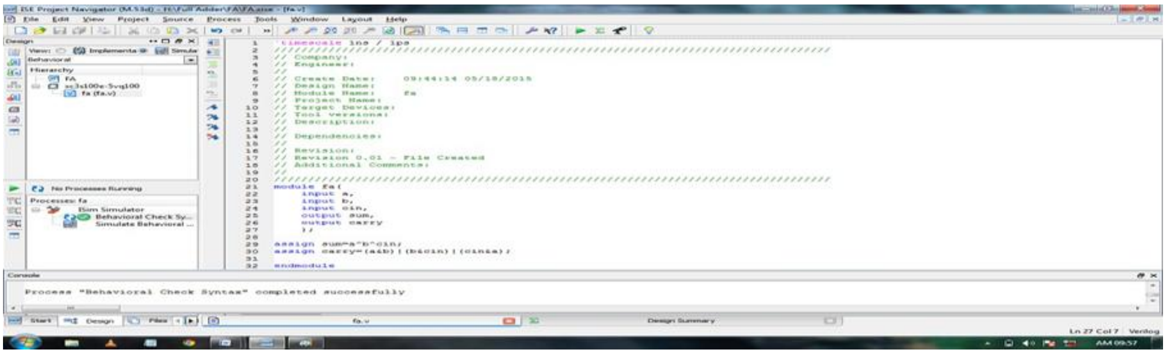
TABLE IV
ERROR DISTANCE COMPARISON OF 4-BIT INPUTS SUM FOR XOR, OR, COMPRESSOR-1, AND COMPRESSOR-2

Input	Err$_i$ XOR	P(Err$_i$) XOR	Err$_i$ OR	P(Err$_i$) OR	Err$_i$ Comp-1	P(Err$_i$) Comp-1	Err$_i$ Comp-2	P(Err$_i$) Comp-2
0000	0	-	0	-	0	-	0	-
0001	0	-	0	-	0	-	0	-
0010	0	-	0	-	0	-	0	-
0011	2	9/256	1	9/256	0	-	1	9/256
0100	0	-	0	-	0	-	0	-
0101	2	9/256	1	9/256	0	-	1	9/256
0110	2	9/256	1	9/256	0	-	1	9/256
0111	2	3/256	2	3/256	0	-	0	-
1000	0	-	0	-	0	-	0	-
1001	2	9/256	1	9/256	0	-	1	9/256
1010	2	9/256	1	9/256	0	-	1	9/256
1011	2	3/256	2	3/256	0	-	0	-
1100	2	9/256	1	9/256	0	-	1	9/256
1101	2	3/256	2	3/256	0	-	0	-
1110	2	3/256	2	3/256	0	-	0	-
1111	4	1/256	3	1/256	2	1/256	1	1/256

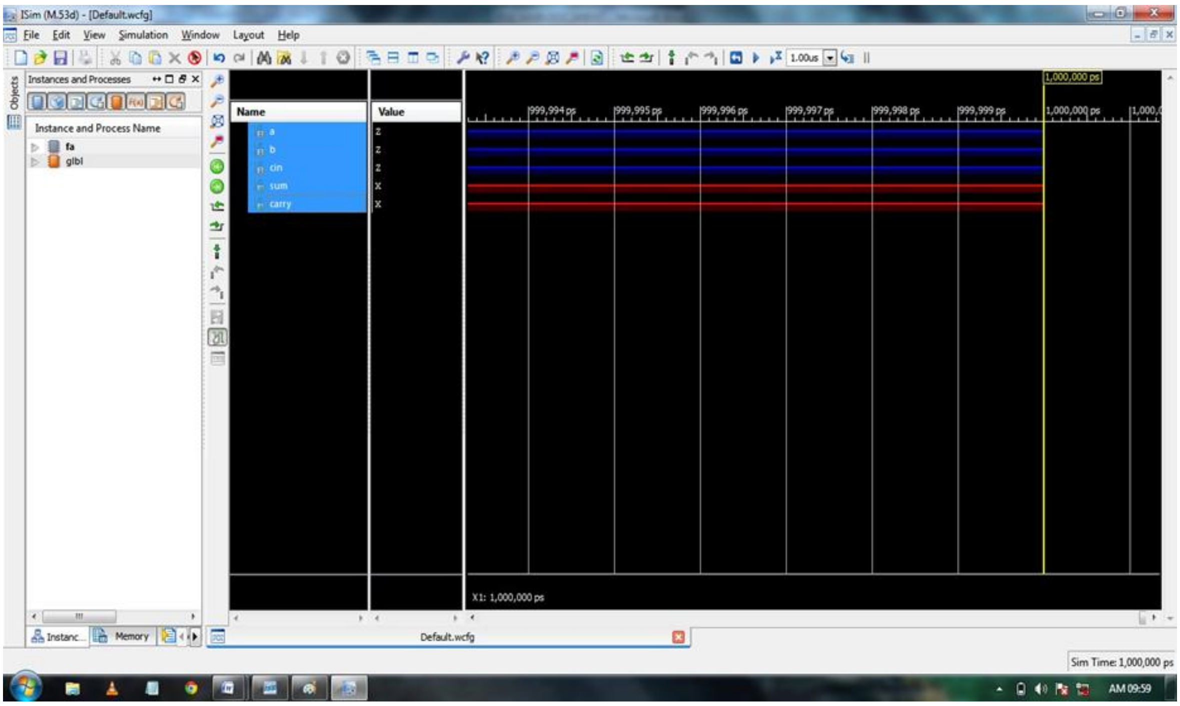
IV. Results & Discussion

Running the Simulation and Observing the Resulting Waveforms

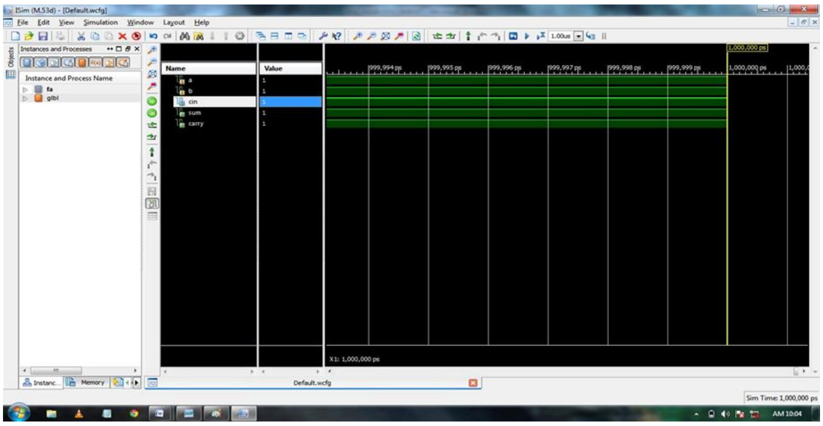
Click on "Simulation." After opening ISim Simulator, selecting the current file, and clicking on Behavioral verify Syntax, you may check for mistakes.



If everything checks out, hit the "Simulate Behavioral Model" button. A new window appears.

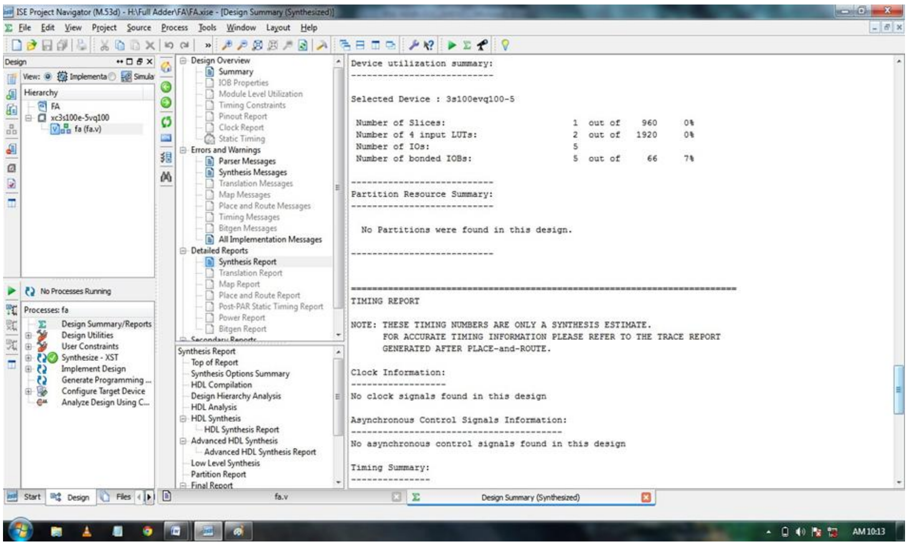


The inputs may be provided here. Forcing a constant value into an input requires a right-click on the chosen input, then entering the value and clicking OK. To see the waveforms of the input and output, click the Run option on the toolbar.

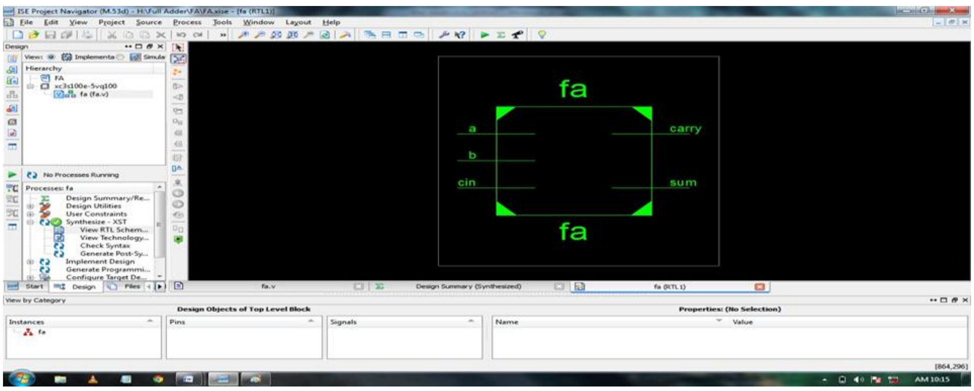


Conceptualization and Execution of the Plan

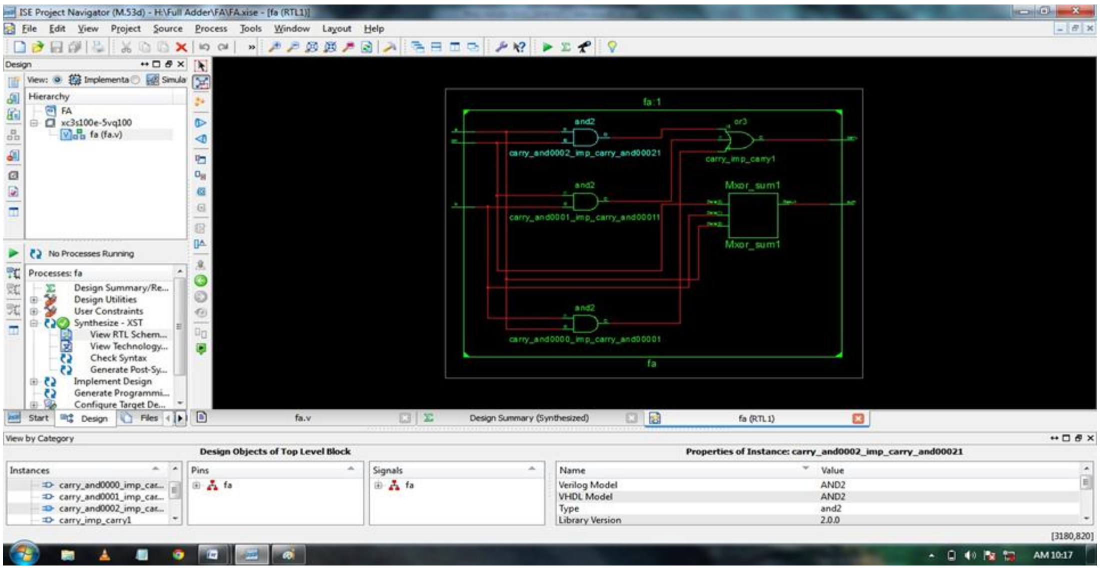
Select "Implementation." Double-click Synthesize-XST after selecting the current file. Fix any mistakes if you find them. Click on the reports and design summary when there are no mistakes View the current project's Devices Usage Report and Timing Report by opening the Synthesis Report under the Detailed Reports menu.



Look at the RTL schematic: Once you've expanded Synthesize-XST, go to the display RTL Schematic tab. Then, click 'ok'.



The internal modules may be seen by opening the window with the Top module. Tap on the topmost module.



Process of FPGA Design:

This section of the lesson will provide a brief overview of the FPGA design flow. The flowing diagram provides a condensed form of the design flow.

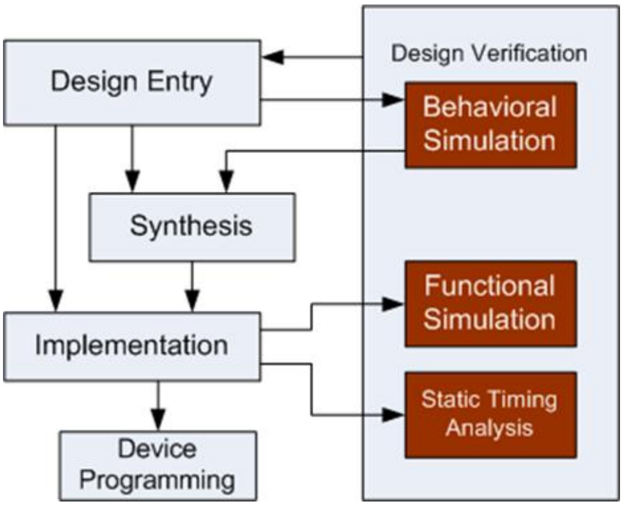


Figure 5.6.1 FPGA Design Flow

V. Conclusion

This study investigated fifteen types of approximate multipliers: eight employing recursive methods for less error resilient applications, six utilizing bit-wise approximation for high error- tolerant applications, and one applying a hybrid approach for balanced error. Compared to the accurate multiplier, significant improvements in area, power, delay, PDP, and PADP by 41.68%, 73.16%, 35.57%, 72.65%, and 75.42% on average. In the existing multipliers, enhancements in area, power, delay, PDP, and PADP are 54.41%, 57.57%, 25.73%, 60.14%, and 74.33% respectively. On contrasting the performance of existing and designed multipliers using trade-off plots, least power, PDP, and PADP were observed. Additionally, for bit-wise approximation design, it attains the smallest area. Most of the multipliers demonstrated acceptable performance in image smoothing, surpassing PSNR and SSIM values of 30dB and 94% correspondingly. Edge detection and image sharpening using 3 × 3 Sobel operator and Laplacian filter resulted in outstanding PSNR and SSIM values tending to infinity and 100% respectively. When employing the 5 × 5 Sobel operator for edge detection, all recommended multipliers yielded PSNR and SSIM values of more than 31dB and 96%. Additionally, in image sharpening using HPF, a PSNR of 27dB and SSIM of 95% were attained. Furthermore, implementing a DNN using the MNIST dataset with the proposed multipliers achieved an accuracy of more than 95%. Our future plans involve integrating approximate multiply and accumulate (MAC) units into Reduced Instruction Set Computing (RISC-V) processors for various AI/ML applications. We anticipate that this approach will significantly enhance the performance of RISC-V in tasks related to image processing and advancing AI/ML technology.

References

[1] M. H. Haider And S.-B. Ko, “Booth Encoding-Based Energy Efficient Multipliers For Deep Learning Systems,” Ieee Trans. Circuits Syst. Ii, Exp. Briefs, Vol. 70, No. 6, Pp. 2241–2245, Jun. 2023.

[2] W. Liu, F. Lombardi, And M. Shulte, “A Retrospective And Prospective View Of Approximate Computing [Point Of View,” Proc. Ieee, Vol. 108, No. 3, Pp. 394–399, Mar. 2020.

[3] N. Amirafshar, A. S. Baroughi, H. S. Shahhoseini, And N. Taherinejad, “Carry Disregard Approximate Multipliers,” Ieee Trans. Circuits Syst. I, Reg. Papers, Vol. 70, No. 12, Pp. 4840– 4853, Dec. 2023.

[4] B. K. Mohanty, “Efficient Approximate Multiplier Design Based On Hybrid Higher Radix Booth Encoding,” Ieee J. Emerg. Sel. Topics Circuits Syst., Vol. 13, No. 1, Pp. 165–174, Mar. 2023.

[5] H. Waris, C. Wang, C. Xu, And W. Liu, “Axrms: Approximate Recursive Multipliers Using High-Performance Building Blocks,” Ieee Trans. Emerg. Topics Comput., Vol. 10, No. 2, Pp. 1229–1235, Apr. 2022.

[6] L. Sayadi, S. Timarchi, And A. Sheikh-Akbari, “Two Efficient Approximate Unsigned Multipliers By Developing New Configuration For Approximate 4:2 Compressors,” Ieee Trans. Circuits Syst. I, Reg. Papers, Vol. 70, No. 4, Pp. 1649–1659, Apr. 2023.

[7] T. Kong And S. Li, “Design And Analysis Of Approximate 4–2 Compressors For High-Accuracy Multipliers,” Ieee Trans. Very Large Scale Integr. (Vlsi) Syst., Vol. 29, No. 10, Pp. 1771– 1781, Oct. 2021.

[8] 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.

[9] 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.

[10] 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. (Vlsi) Syst., Vol. 27, No. 5, Pp. 1161–1173, May 2019.

[11] K. Chen, W. Liu, J. Han, And F. Lombardi, “Profile-Based Output Error Compensation For Approximate Arithmetic Circuits,” Ieee Trans. Circuits Syst. I, Reg. Papers, Vol. 67, No. 12, Pp. 4707–4718, Dec. 2020.

[12] A. Akbar Bahoo, O. Akbari, And M. Shafique, “An Energy-Efficient Generic Accuracy Configurable Multiplier Based On Block-Level Voltage Overscaling,” Ieee Trans. Emerg. Topics Comput., Vol. 11, No. 4, Pp. 851–867, Dec. 2023.

[13] F. Guella, E. Valpreda, M. Caon, G. Masera, And M. Martina, “Marlin: A Co-Design Methodology For Approximate Reconfigurable Inference Of Neural Networks At The Edge,” Ieee Trans. Circuits Syst. I, Reg. Papers, Vol. 71, No. 5, Pp. 2105–2118, May 2024.

[14] A. S. Baroughi, S. Huemer, H. S. Shahhoseini, And N. Taherinejad, “Axe: An Approximate-Exact Multi-Processor System-On-Chip Platform,” In Proc. 25th Euromicro Conf. Digit. Syst. Design (Dsd), Aug. 2022, Pp. 60–66.

[15] S. Xu And B. C. Schafer, “Approximate Reconfigurable Hardware Accelerator: Adapting The Micro-Architecture To Dynamic Workloads,” In Proc. Ieee Int. Conf. Comput. Design (Iccd), Nov. 2017, Pp. 113–120.

[16] 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.

[17] N. Amirafshar, A. S. Baroughi, H. S. Shahhoseini, And N. Taherinejad, “An Approximate Carry Disregard Multiplier With Improved Mean Relative Error Distance And Probability Of Correctness,” In Proc. 25th Euromicro Conf. Digit. Syst. Design (Dsd), Los Alamitos, Ca, Usa, Aug. 2022, Pp. 46–52, Doi: 10.1109/Dsd57027.2022.00016.

[18] H. Waris, C. Wang, W. Liu, J. Han, And F. Lombardi, “Hybrid Partial Product-Based High- Performance Approximate Recursive Multipliers,” Ieee Trans. Emerg. Topics Comput., Vol. 10, No. 1, Pp. 507–513, Jan. 2022.

[19] K. Bhardwaj, P. S. Mane, And J. Henkel, “Power- And Area-Efficient Approximate Wallace Tree Multiplier For Error-Resilient Systems,” In Proc. 15th Int. Symp. Quality Electron. Design, Mar. 2014, Pp. 263–269.

[20] B. Rashidi, “Efficient And Low-Cost Approximate Multipliers For Image Processing Applications,” Integration, Vol. 94, Jan. 2024, Art. No. 102084. [Online]. Available: Https://Www. Sciencedirect.Com/Science/Article/Pii/S0167926023001268

[21] S. Rehman, W. El-Harouni, M. Shafique, A. Kumar, J. Henkel, And J. Henkel, “Architectural-Space Exploration Of Approximate Multipliers,” In Proc. Ieee/Acm Int. Conf. Comput.-Aided Design (Iccad), Nov. 2016, Pp. 1–8.

[22] P. Kulkarni, P. Gupta, And M. Ercegovac, “Trading Accuracy For Power With An Underdesigned Multiplier Architecture,” In Proc. 24th Int. Conf. Vlsi Design, 2011, Pp. 346– 351.

[23] E. Zacharelos, I. Nunziata, G. Saggese, A. G. M. Strollo, And E. Napoli, “Approximate Recursive Multipliers Using Low Power Building Blocks,” Ieee Trans. Emerg. Topics Comput., Vol. 10, No. 3, Pp. 1315–1330, Jul. 2022.

[24] Y. Guo, H. Sun, And S. Kimura, “Design Of Power And Area Efficient Lower-Part-Or Approximate Multiplier,” In Proc. Ieee Region Conference, 2018, Pp. 2110–2115.

[25] S. K. N. Mahammad, “Low Power, High Speed Approximate Multiplier For Error Resilient Applications,” Integration, Vol. 84, Pp. 37–46, May 2022. [Online]. Available: Https://Www. Sciencedirect.Com/Science/Article/Pii/S0167926022000013

[26] M. Ha And S. Lee, “Multipliers With Approximate 4–2 Compressors And Error Recovery Modules,” Ieee Embedded Syst. Lett., Vol. 10, No. 1, Pp. 6–9, Mar. 2018.

[27] U. A. Kumar, S. V. Bharadwaj, A. B. Pattaje, S. Nambi, And S. E. Ahmed, “Caam: Compressor-Based Adaptive Approximate Multiplier For Neural Network Applications,” Ieee Embedded Syst. Lett., Vol. 15, No. 3, Pp. 117–120, Aug. 2023.

[28] Y. Chang, Y. Cheng, Y. Lin, S. Liao, C. Lai, And T. Wu, “Imprecise 4–2 Compressor Design Used In Image Processing Applications,” Iet Circuits, Devices Syst., Vol. 13, No. 6, Pp. 848–856, Sep. 2019. [Online]. Available: Https://Ietresearch.Onlinelibrary. Wiley.Com/Doi/Abs/10.1049/Iet-Cds.2018.5403