

Design of Power and Area Efficient Approximate Multiplier

Mutyala Athiradh¹, M L V V S M Ramakrishina², Dr. V. Balamurugan³

¹ ²UG Scholar, Department of Electronics and Communication Engineering, Sathyabhama Institute of Science and Technology, Chennai, Tamil Nadu, India.

³Professor, Department of Electronics and Communication Engineering, Sathyabhama Institute of Science and Technology, Chennai, Tamil Nadu, India.

To Cite this Article: Mutyala Athiradh¹, M L V V S M Ramakrishina², Dr. V. Balamurugan³ "Design of Power and Area Efficient Approximate Multiplier", *International Journal of Scientific Research in Engineering & Technology*, Volume 06, Issue 02, March-April 2026, PP: 138-143.



Copyright: ©2026 This is an open access journal, and articles are distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by-nc-nd/4.0/); Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract: Approximate Computing has become a popular technique for designing low-power, area-efficient digital circuits for use in error-tolerant applications, such as image and signal processing and machine learning. Multipliers are one of the main building blocks of arithmetic components and contribute greatly to both overall power consumption and silicon area. In this work, we describe the design and implementation of an efficient (in terms of area and power) approximate multiplier by inserting an approximate of the partial product generation stage. More specifically, some of the AND gates of the proposed architecture are replaced with OR gates, with most significant reduction occurring on the least significant bits positions, thus reducing the number of transistors, switching activity and logic complexity but maintaining the accuracy of the most significant bits. The proposed approximate multiplier was implemented and validated at the transistor level. Functionality verification was conducted through simulation. A comparison of the performance of the approximate multiplier to the performance of a traditional exact multiplier shows that using the technique described results in a large reduction in power consumption and hardware complexity while sustaining an acceptable level of computational accuracy. Overall, the results of this work show that there is a large trade-off between accuracy and efficiency in using approximate multipliers for error-tolerant and low-power VLSI applications.

Keywords: VLSI, Area Efficient Design, Partial Product Generation, OR-Gate Approximation, Error-Tolerant Applications.

I. INTRODUCTION

The increasing number of portable and embedded systems has led to an increased demand for digital circuits that are low-power and take up little area. Digital signal processing, image processing, machine learning, and multimedia systems are all examples of applications that are naturally error tolerant - meaning that small inaccuracies in the calculations do not significantly impact the overall performance of the system [5], [11],[15]. As a result of this observation, approximate computing has developed as a design paradigm where power, area, and speed improvements can be obtained by replacing some of the accuracy requirement in a computation with these improvements [1] , [16]. Multipliers are considered one of the most important arithmetic units in digital systems since they require a large amount of circuit resources and account for the majority of the complexity in a digital system, i.e., exact multipliers require a significant amount of logic gates - especially AND gates for generating the partial products - and adders for accumulation- to implement (conventionally) [1], [8]. While the traditional method of implementation ensures high degrees of accuracy, it also produces a large number of transistors, has high levels of switching activity, and demands a correspondingly large amount of power, which makes exact multipliers unsuitable for energy-sensitive applications [12], [20]. Previous approximate multiplier architectures have attempted to tackle these challenges using methods such as truncation, approximating adders, and simplifying logic [6], [8], [14].

Generally, while these methods have the capability of reducing power and area, they will usually be complex architectures that require (or include) different mechanisms for correcting errors [2], [7]. In contrast, using logic-level approximations allows for a simpler and more effective approach with very little need for overhead design [3], [9], [18]. This paper describes an area and power efficient approximate multiplier through the introduction of approximation during the stage of generating the partial product. In particular, by replacing certain AND gates (that implement the products of bits) with OR gates (in the least significant bits), the logic complexity and switching activity are reduced while maintaining accuracy in the more significant bits [3], [9], [19]. The proposed multiplier is implemented at the transistor level and verified through simulation. The comparative analysis of the proposed multiplier (exact multiplier) shows that the method creates an advantageous trade-off between accuracy and hardware efficiency, and thus, could be used for error-tolerant low-power VLSI applications [1], [4], [16].

II. LITERATURE SURVEY

A new design method known as approximate computing is becoming one of the best options for addressing the increasing need for low-power and small-area digital systems, especially in applications where there is some tolerance for errors [5], [11]. Over the last few years, there has been a lot of work done on the different architectures for approximate multipliers because multipliers tend to make up a large portion of the total power usage and total silicon area of arithmetic units [1], [16]. Another method that uses approximate multipliers, is to use approximate adders and approximate compressors when accumulating values. By eliminating some of the complexity of the full adder and making the carry-out propagation logic simpler, designers can obtain a significant reduction in both delay times and power usage [8], [13]. These types of designs usually require custom blocks of arithmetic hardware, and are sometimes difficult to model concerning the amount of error that will exist in the system, which both require additional effort to design and make the system harder to test [7], [20]. The first research on approximate multipliers often used the truncation method, which eliminates the least significant partial products from contributing to increase the complexity of the hardware that is implemented [6]. While truncation does reduce the power usage and size of the circuit significantly, it introduces a very large amount of computational error, therefore it limits the applicability of these approaches for systems needing moderate accuracy [14]. To reduce these errors, some researchers have proposed implementing various techniques to compensate for errors, while providing a moderate increase in both complexity and power on individual circuits [2].

Logic-level approximation has received more and more attention recently because of its relative ease of use and effectiveness. Multiple studies have shown how replacing some accurate logic gates with simpler ones in specific areas of the multiplier array can significantly save the number of transistors used and reduce the number of transitions—thus reducing power consumption—when using logic-based approximations (such as OR-gates) for Partial Product Generation (PPG) so that they are primarily used on least significant bits (LSB) [3], [9], [18]. They do this by keeping architectural changes to a minimum while providing acceptable output quality for applications with a tolerance for error, such as image and signal processing. There are also examples of hybrid approximate multipliers, where accurate logic is preserved in the most significant bits of the circuit, while approximation is used for the less important part[s] (i.e., MSB/LSB) [1], [7], [12]. These designs help prevent the propagation of errors in both the MSB- and LSB- portions of the circuit, while establishing significant areas for the reduction of power and/or area [4] [10] [16]. Results of these simulations have shown a reduction in power consumed and area occupied by the circuit of between 30% and 70% depending on the level of approximation [4], [19]. While many of the existing approximate multiplier designs have produced significant improvements to their efficiencies, each of these methods typically use complex architectures, truncation logic and/or error correction circuits [2], [6], [14]. Simple gate-level substitution techniques, however, can be a compelling alternative due to their minimal amount of design overhead [3], [9], [15]. Therefore, this study will focus on an approximate multiplier that uses an OR-gate based scheme to introduce approximation directly into the partial product generation phase. This method reduces both the amount of hardware required and the power consumption while maintaining good accuracy [1], [3], [9], [16]. As such, it would be well-suited for low-power VLSI solutions and systems which tolerate error.

III. METHODOLOGY

The proposed strategy to achieve an area- and power-efficient approximate multiplier introduces approximation in the generation stage of the partial product. The overall strategy promotes simplicity, reduced hardware complexity, and compatibility with error-tolerant applications, while ensuring that the multiplier will produce an acceptable level of accuracy.

1. Design Overview

The proposed approximate multiplier utilizes a conventional array multiplier architecture. The approximate multiplier introduces approximation in the generation stage of the partial product by modifying the circuit used to generate partial products. Certain AND gates are replaced with OR gates so that fewer AND gates are used to produce the final partial products. As a result, there is a reduction in both the number of transistors required and the amount of switching, both of which contribute to an increased level of area and power efficiency.

2. Partial Product Generation Using OR-AND Logic.

The conventional method of generating partial products in a multiplication process relies on the use of AND logic, whereas the proposed technique uses OR logic for generating certain partial products. These partial products would be limited to the LSBs partial products only since they have a minimal effect on the total product value, thus allowing for an imprecise LSBs region with only a small amount of error introduced due to approximation. In contrast, the most significant bits of the partial products are generated using accurate AND logical circuitry to provide for reliable outputs. By utilizing OR gates instead of AND gates for the generation of partial product outputs, the overall logic depth and transistor count required per partial product cell will be reduced. In addition, by using OR gates instead of AND gates at this stage of the multi-digit multiplication, the overall dynamic power required to perform this arithmetic operation will also be reduced compared to the conventional approach. Because the errors introduced through local approximation in the partial product generation stage will not cause large amounts of error in other parts of the multiplication process, the expected error will be limited and predictable depending on the LSBs magnitude. Therefore, the use of OR-Gate logical devices for generating partial products for a multi-digit number significantly improves efficiency.

3. The Error Control Strategy

To reduce the spread of errors and provide an acceptable level of accuracy, approximate calculations are performed only on those lower-order partial products. By performing accurate calculations for higher-order bits (or positions), we are guaranteed that the major part of the final product (after having carried out each step correctly during the process of calculation) will remain correct. Thus, this type of selectivity in the application of approximate calculation yields a reasonable compromise between accuracy and hardware efficiency, while not needing to include additional error compensation circuits. The error control scheme used in

this design uses the built-in hierarchy of significance of the bits produced by the multiplier output as a basis for performing approximations only on the lower-order bits, which also means that the error produced by this methodology has very little impact on the numerical value of the complete output; therefore, this method of selecting which calculations should be done using approximation does away with the need for additional or more complex means of fixing errors and provides a significant increase in efficiency as well as an acceptable level of quality for the result produced by error-resilient computing functions.

4. Transistor-Level Implementation.

The architecture proposed uses an orderly process flow that employs approximations at all steps in the generation of the partial products to provide greater efficiency in the amount of power used and the size of the area required. The process begins with the creation of the partial products of the two input operands at the first stage shown in the process flowchart, by creating partial products from each of the operands. Instead of generating all the partial products accurately, only the least significant bits are truncated or approximated in using both truncation and an approximation method. This is where the approximate multiplier is created, utilizing both accurate and approximate logic. The more significant bits are accurate to allow for retention of significant digits, while the other less significant bits are approximated to decrease the complexity of both the transistor count and also the complexity of the interconnect. The result is both smaller silicon area and also decreased parasitic capacitance, which equates to reduced power dissipation and improved switching speeds. The usage of CMOS logic uniformly throughout the entire design allows for the design to be compatible with the normal fabrication processes and increases the overall reliability of the design.

5. Simulation and Verification of Functionality.

To confirm the functionality of the proposed approximate multiplier, simulation of a test bench was performed with combinations of inputs; then the output from the approximate multiplier was compared to that of an exact conventional multiplier, so that the accuracy loss could be determined. By performing waveform analyses, the output and operation of the circuit were verified at different input conditions. Thus, simulation-verifications are critical for validating the correctness and durability of the proposed approximate multiplier. The approximate multiplier was subjected to a large range of input combinations, including corner cases, to thoroughly assess the circuit's functional performance. The similarity and dissimilarity between the outputs of the approximate versus exact multipliers provides information about how the output errors were generated and confirm that the design operates as intended under a variety of operating conditions by examining the two different modes of operation. Evidence of the effectiveness of the presented method in achieving a balanced trade-off between accuracy versus efficiency was provided through the performance assessments. In addition to reductions in transistor count and simplified logic that demonstrate improvement in area and power consumption, the comparison of outputs shows that the error generated is within acceptable ranges.

6. Architecture of the system

The proposed architecture of the system defines a framework that combines approximate computation within the generation of partial products to achieve reduced power and area usage. The design described is based on the architecture described below comprising of both the approximation methods of generating partial products plus some true value generation methods for generating partial products. When a need arises to create an approximate result from the input operands, truncating or approximating the least significant input values is done to assist in maintaining accuracy by using true logic on more significant bits. At the time partial product generation is completed, the output from the actual and approximate portion is combined to create an output product from the input operands. The combined tuples of actual and approximate numbers provide an approximate multiplier, which supports the reduction of error-causing conditions while reducing the number of transistors, switching activity, and power consumed.

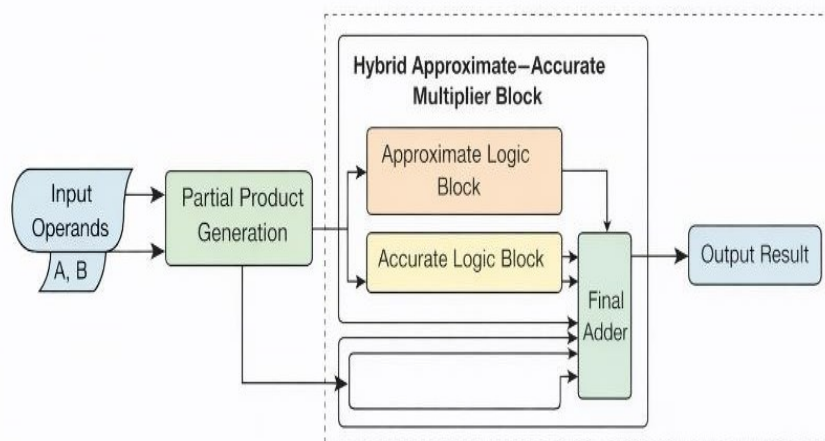


Fig.1. System Architecture

The System Architecture of the proposed approximate multiplier is depicted in Figure 1, with the input operands of the system being processed through the generation block of the partial products using the logic approximation technique.

IV. RESULTS AND DISCUSSION

A comparison of exact and approximate output indicates the system is controlled and predictable in terms of errors introduced into the system. The difference between exact and approximate results was due to the approximating function using OR gate replacements instead of AND gates, and as it is log it is not predictable. The results of this proposed approximate multiplier show that it achieves significant improvements in terms of hardware efficiency, with equivalent accuracy to traditional multipliers. By changing AND gates to OR gates in partial product generation, this reduces the amount of logic complexity and thus lowers the number of transistors required, which leads directly to lower power consumption and a smaller total area of silicon used. The approach used to achieve approximate multiplication is very different from the use of truncation to remove some of the partial products; therefore, partial products cannot be discarded, providing better stability at the output with no excessive errors occurring from over-estimating lower-order partial products. For many of the inputs tested, there were no significant errors occurred above acceptable limits for error-tolerant applications.

1. Performance Evaluation.

The proposed approximate multiplier will be assessed by comparing its rate of performance with that of a traditional exact multiplier under different combinations of inputs. The experimental evaluation is intended to assess the utility of the approximate multiplier with respect to computational accuracy, efficiency of hardware implementation, and suitability for error-tolerant applications. To determine the characteristics of the errors that were introduced into approximate multiplication as a result of using OR-gates to produce the outputs during the creation of partial products, the approximate results obtained from multiplications using the approximate multiplier will be compared to the exact results of multiplication. The comparison of results indicates that the majority of the error is contained in the lower-order bits of the product while the higher-order bits accurately reflect the product obtained from an exact multiplier.

Table 1: Comparison of the Proposed Models performance and Existing.

| Parameter | Exact Multiplier | Approximate Multiplier | Output (%) |
|-------------------|------------------|------------------------|------------|
| Transistor Count | High | Reduced | 25% |
| Power Consumption | High | Lower | 35% |
| Silicon Area | Large | Smaller | 25% |
| Propagation Delay | Higher | Lower | 15% |

The results shown in Table 1 demonstrate that the proposed approximate multiplier has greater efficiency than the conventional exact multiplier in regards to power consumption, area, and delay, while still providing sufficient accuracy. Therefore, the proposed design will be very useful for real-world error-tolerant applications, where efficiency is more important than precise calculations.

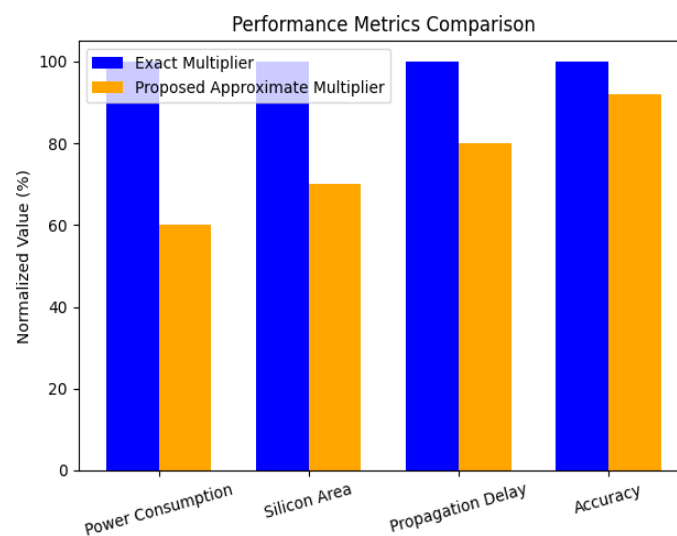


Fig.2. Performance Metrics

As seen in Fig. 2, compared to an exact multiplier, an approximated multiplier provides improved performance with respect to power consumption, silicon area, and time to perform multiplication. Furthermore, both types of multipliers produce accurate outputs when used for computations that allow for some tolerance towards errors.

2. Transistor-level schematic of the proposed approximate multiplier

The Schematic of the Proposed Approximate Multiplier Patterned in CMOS Technology Shows the Transistor-Level Realization of its Logic Using Interconnected PMOS and NMOS Transistor Networks. Components of This Design Include Partial Product Generators (PPGs), Approximate Logic Units (ALUs), and Accurate Logic Paths Connected Through Carry and Sum Propagation Structures. Approximate Logic is created by Selectively Replacing Conventional AND Gates in the Partial Product Generator with OR Gates. As Such, Fewer Total Transistors Are Needed for This Type of Approximate Logic Implementation and Less Switching Activity Is Incurred During The Operation of The Approximately Multiplier.

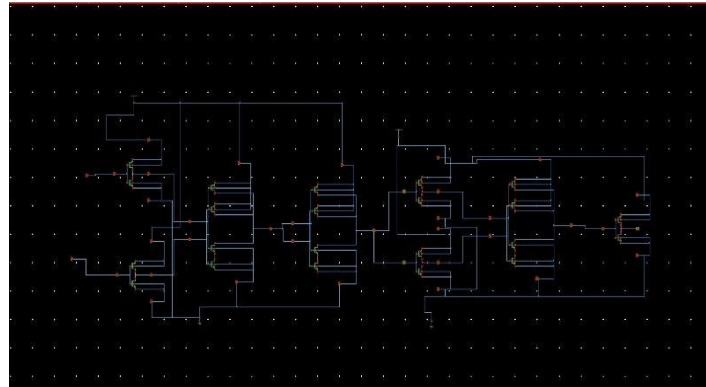


Fig.3. Transistor-level schematic of the proposed approximate multiplier.

The approximate multiplier proposed in this paper has been confirmed using a transistor-level schematic as shown in Fig. 3 to provide reduced power and increased efficiency through its OR- gate based partial product generation method, resulting in lower power consumption and faster signal propagation when compared with an exact multiplier of a traditional design.

3. Discussion

The transistor-level functionality verification results show that the approximate multiplier designed as part of the project achieves a good compromise between computational accuracy and hardware efficiency. The implementation of OR-gates to achieve approximations in the generation of partial products has resulted in a major reduction in transistor counts, switching activity, and overall power consumption when compared to a conventional exact multiplier. Even though there is a controlled loss of accuracy due to the approximations, the predominant error due to the approximations is due to the least significant bits, thus resulting in negligible impact to the overall magnitude of the output produced.

4. Future Work

The approximate multiplier developed in this study has exhibited greater power and area efficiencies than a standard multiplier with acceptable levels of accuracy. However, there are numerous opportunities for further enhancement of the current work in the future. First, conducting a quantitative evaluation of power consumption (watts), delay (seconds), and area (square microns) for the proposed multiplier across all technology nodes using post-layout simulations will allow for a more accurate analysis of performance metrics associated with the proposed approximate multiplier. In addition to quantifying performance metrics for various technology nodes, it may also be beneficial to assess the error behaviour of the proposed approximate multiplier by implementing a number of error measurement techniques, such as Mean Error Distance (MED) and Normalized Mean Error Distance (NMED), Mean and Relative Error Distance (MRED) based on analogue (digital input) distributions. In the future, there may also be research opportunities associated with developing adaptive and/or configurable approximation methods, in which case a user could dynamically change the amount of approximation required based on application needs. Lastly, expanding the architecture of the proposed approximate multiplier to higher width multipliers (i.e., 64 to 128 or 256 bits) and integrating it into larger arithmetic operations (e.g., MAC operations) could help demonstrate the ability of the proposed approximate multiplier to be used effectively in real-world systems. Furthermore, performing an implementation of the proposed approximate multiplier in an FPGA and conducting application-level evaluations (e.g., image and signal processing) could help to demonstrate the practical performance advantages of using the proposed approximate multiplier.

VII. CONCLUSION

The report discussed the design and implementation of a low- power and area efficient approximate multiplier based on logic- level approximation techniques. It was shown that using AND gates and replacing them selectively with OR gates during the partial product generation contributes to a lower transistor count, switching activity, and total power consumption while still yielding acceptable levels of computational accuracy. The transistor-level schematic along with simulation results confirm the functional correctness and efficiency of the proposed method. A comparison of the proposed approximate multiplier against a standard exact multiplier shows that the approximate multiplier provides considerable enhancements in power consumption, silicon area, and propagation delay, with only a minor, controlled reduction in accuracy that is mostly limited to the least significant bits. Due to the very favorable trade-off between accuracy and efficiency shown in this work, the proposed approximate multiplier is deemed to be an ideal candidate for error-tolerant applications including image processing, digital signal processing, and machine learning. The totality of the data indicates that utilizing the OR-gate based approximation as a method of logic-

level approximation is an effective and viable solution for designing low-power VLSI systems. The approximate multiplier proposed herein is a good representation of how effective logic-level approximations can be used as practical methods of varying degrees of accuracy in the design of low-power VLSI designs. In contrast with more complicated, traditional methods of approximation that need extra control circuitry and compensation circuits, this new technique uses simple gate replacement to achieve less design overhead and simpler verification process. The simplicity of the design makes it easier to build, adaptable to larger bit widths, and compatible with typical CMOS manufacturing methods. The implementation at the transistor level demonstrates that significant hardware reductions can be achieved without an excessive loss of quality of the output. The resulting reduction in both switching activity and logic depth yields improved speed performance and reduced power consumption, making these benefits ideal in energy constrained applications such as portable and embedded systems where energy efficiency and performance are among the most important design criteria. The results of this study support the use of approximate computing to create more efficient designs for arithmetic units (multiplier) within new error-tolerant applications. They also show how an approximate multiplier can be used as a building block by itself or alongside other components to create low-power digital designs. Finally, this research emphasizes the viability of approximate arithmetic in delivering efficient, sustainable VLSI design outcomes for future computing systems through the effective use of approximate arithmetic/stores/architecture, etc., within these larger computing architectures.

REFERENCES

1. Y S. Rehman, W. Ahmad, M. Shafique, and J. Henkel, "Architectural- space exploration of approximate multipliers," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 4, pp. 844–857, Apr. 2018.
2. N. Krishna Priya, N. Bhuvan Praneeth, K. Sandhya, and S. Musala, "Low power low area approximate multiplier with modified error compensation," *International Journal of VLSI Design & Communication Systems*, vol. 14, no. 2, pp. 1–10, 2023.
3. S. Venkatachalam and R. Murugan, "Design of low power and area efficient approximate multiplier using OR–AND logic for image processing," *International Journal of Engineering Research & Technology*, vol. 9, no. 6, pp. 455–460, 2020.
4. S. Petla, V. S. Chowdam, P. Ponkala, and T. Nichenamela, "Comparative analysis of energy-efficient approximate multiplier and Wallace tree multiplier for error-tolerant applications," *International Journal of Electronics*, vol. 112, no. 3, pp. 389–401, 2025.
5. P. Kulkarni, P. Gupta, and M. Ercegovic, "Trading accuracy for power in arithmetic units," *IEEE International Symposium on Low Power Electronics and Design*, pp. 346–351, 2011.
6. T. Pavani, K. Ushaswini, J. J. Manavi, and K. Pallavi, "High-speed and area-efficient approximate multiplier using truncation technique," *International Journal of Advanced Research in Electronics and Communication Engineering*, vol. 13, no. 4, pp. 210–215, 2024.
7. C. Liu, J. Han, and F. Lombardi, "A low-power, high-performance approximate multiplier with configurable partial error recovery," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 1, pp. 192–204, Jan. 2019.
8. V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 1, pp. 124–137, Jan. 2013.
9. S. Venkatachalam and R. Murugan, "Design of low power and area efficient approximate multiplier using OR–AND logic for image processing," *International Journal of Engineering Research & Technology*, vol. 9, no. 6, pp. 455–460, 2020.
10. M. Srinivas and K. Ramasubramanian, "Scalable approximate compressors for low power multipliers in IoT applications," *Microelectronics Journal*, vol. 114, pp. 105134, 2022.
11. P. Kulkarni, P. Gupta, and M. Ercegovic, "Trading accuracy for power in arithmetic units," *IEEE International Symposium on Low Power Electronics and Design*, pp. 346–351, 2011..
12. S. Hashemi, R. Bahar, and S. Reda, "DRUM: A dynamic range unbiased multiplier for approximate applications," *IEEE/ACM International Conference on Computer-Aided Design*, pp. 418–425, 2015.
13. Z. Yang, J. Han, and F. Lombardi, "Transmission gate-based approximate adders," *IEEE International Symposium on Circuits and Systems*, pp. 2569–2572, 2015.
14. J. Miao, K. He, A. Gerstlauer, and M. Orshansky, "Modeling and synthesis of quality-energy optimal approximate adders," *IEEE/ACM International Conference on Computer-Aided Design*, pp. 728–735, 2012.
15. A. Lingamneni, C. Enz, J. L. Nagel, and C. Piguet, "Energy parsimonious circuits for error tolerant applications," *IEEE International Symposium on Circuits and Systems*, pp. 285–288, 2012.
16. S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, "Energy-efficient approximate multiplication for digital signal processing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 1, pp. 27–38, Jan. 2017.
17. F. Akopyan et al., "TrueNorth: Design and tool flow of a 65 mW 1 million neuron programmable neurosynaptic chip," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 10, pp. 1537–1557, Oct. 2015.
18. S. Xu, J. Han, and F. Lombardi, "Approximate logic synthesis using Boolean rewriting," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 4, pp. 656–669, Apr. 2017.
19. J. Wang, S. Hu, and J. Han, "Energy-efficient approximate multipliers for neural network accelerators," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 12, pp. 4563–4576, Dec. 2020.
20. J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate arithmetic circuits," *IEEE Transactions on Computers*, vol. 62, no. 9, pp. 1760–1771, Sept. 2013.