

64-BIT LOWPOWER AND HIGH SPEED APPROXIMATE MULTIPLIER USING HIGHER ORDER COMPRESSORS FOR MEASUREMENT SYSTEM

Mr.V.Murali Krishna¹, U.Chaitanya lakshmi², S. Jessy³, S.Rohith Sai⁴, V.Hasanthi⁵

¹ Assistant Professor, Dept. of Electronics and Communication Engineering Department, Seshadri Rao Gudlavalleru Engineering College, Andhra Pradesh, India

^{2,3,4,5} Student, Department of Electronics and Communication Engineering, Seshadri Rao Gudlavalleru Engineering college, Andhra Pradesh, India

Abstract - In modern digital systems, especially in image processing and digital signal processing (DSP) applications, achieving high performance with low power consumption is crucial. Approximate computing has emerged as a promising paradigm to optimize power, area, and delay by allowing controlled inaccuracies in computations where exact results are not critical. This paper proposes the design and implementation of a 64-bit approximate multiplier using high-order compressors such as 8:2, 4:2, and 3:2 to reduce the number of partial product reduction stages. The use of higher-order compressors significantly lowers the critical path delay and power consumption by minimizing the number of intermediate additions required in the reduction tree. The proposed 64-bit multiplier is evaluated and compared against conventional exact multipliers in terms of delay, power, and accuracy metrics. Simulation results demonstrate a substantial improvement in delay and power efficiency, making the design well-suited for image processing tasks and DSP operations where approximate results are acceptable.

Keywords: Approximate Computing, Approximate Multiplier, High-Order Compressors, Partial Product Reduction, Low-Power VLSI Design.

1. INTRODUCTION

Approximate computing is an emerging concept in digital design that relaxes the requirement of exact computation to achieve improvements in power consumption, speed, and hardware efficiency. This approach is especially useful for embedded and mobile systems that operate under strict energy and performance constraints. Many modern applications such as multimedia processing, image processing, machine learning, and data mining can tolerate small errors, meaning that a perfectly accurate result is not always necessary. Multipliers are fundamental components in microprocessors, digital signal processors, and embedded systems where they are used in operations such as filtering, convolution, and neural network computations. However, multipliers are complex circuits and consume significant power and hardware resource because of this, the design of Approximate multipliers has become an important research area to improve system performance and reduce energy consumption. In the proposed multiplier design, exact compressors are used in specific parts of the circuit to maintain computational accuracy. While approximate computing reduces power consumption and delay by allowing small errors, using exact compressors in critical sections of the multiplier ensures that the overall error does not significantly affect the final output. In particular, exact compressors are typically used in the most significant (MSB) region of the partial product reduction stage, where errors would have a larger impact on the final result. By combining approximate techniques with exact compressors, the design achieves a balance between performance and accuracy. The approximate compressors help reduce delay, power consumption, and hardware complexity, while the exact compressors preserve the reliability of important computations. This hybrid approach improves the efficiency of the multiplier while still maintaining acceptable output quality for applications such as digital signal processing and machine learning. In our project, the partial product reduction stage is implemented using 8:2, 4:2, and 3:2 compressors. These compressors help combine multiple input bits into fewer output bits, thereby reducing the number of reduction stages and improving the overall speed of the multiplier. The use of these compressors creates a balanced reduction tree that decreases propagation delay and improves power efficiency while maintaining acceptable accuracy, making the design suitable for applications such as digital signal processing and artificial intelligence systems.

2. LITERATURE REVIEW

Narayanamoorthy, S., Moghaddam, H. A., Liu, Z., Park, T., & Kim, N. S. (2015). Energy- Efficient Approximate Multiplication for Digital Signal Processing and Classification Applications. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 23(6), 1180–1184. The need to support various digital signal processing (DSP) and classification applications on energy-constrained devices has steadily grown. Such applications often extensively perform matrix multiplications using fixed-point arithmetic while exhibiting tolerance for some computational errors. Hence, improving the energy efficiency of multiplications is critical. In this brief, we propose multiplier architectures that can tradeoff computational accuracy with energy consumption at design time. Compared with a precise multiplier, the proposed multiplier can consume less energy/op with average computational error of $\approx 1\%$. Finally, it is demonstrated that such a small computational error does not notably impact the quality of DSP and the accuracy of classification applications. Summary: In this paper, Power consumption and area can further be reduced. Zervakis

G., Xydis, S., Tsoumanis, K., Soudris, D., & Pekmestzi, K. (2015). Hybrid approximate multiplier architectures for improved power-accuracy trade-offs. 2015 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED). Approximate computing forms a promising design alternative for inherently error resilient applications, trading accuracy for power savings. In this paper, we exploit multi-level approximation, i.e. at the algorithmic, the logic and the circuit level, to design low power approximate arithmetic architectures for hardware multipliers. Motivated from the limited power savings that approximation techniques can achieve in isolation, we explore hybrid methods that apply simultaneously more than one techniques from different layers. We introduce the concept of perforation for approximate arithmetic circuit design and we explore the newly defined design space of hybrid designs showing that it leads to lower power consumption at every examined error range. To address the increased complexity of the target design space, we introduce an heuristic optimization technique and the corresponding design framework that automatically generates hybrid low-power approximate multipliers requiring a small number of design evaluations, i.e. synthesis, simulation power and timing analysis. Through extensive experimentation, we show that the proposed techniques converge towards optimal solutions and deliver approximate designs that are always more efficient with respect to state-of-art approaches. Power savings of 11% are reported for small error bounds and more than 30% in case of more relaxed error constraints. Summary: In this paper, power consumption can further be reduced.

A. Momeni, J. Han, "Design and Analysis of Approximate Compressors for Multiplication" *IEEE Transactions on Computers*. Inexact (or approximate) computing is an attractive paradigm for digital processing at nanometric scales. Inexact computing is particularly interesting for computer arithmetic designs. This paper deals with the analysis and design of two new approximate 4-2 compressors for utilization in a multiplier. These designs rely on different features of compression, such that imprecision in computation (as measured by the error rate and the so-called normalized error distance) can meet with respect to circuit-based figures of merit of a design (number of transistors, delay and power consumption). Four different schemes for utilizing the proposed approximate compressors are proposed and analyzed for a Dadda multiplier. Extensive simulation results are provided and an application of the approximate multipliers to image processing is presented. The results show that the proposed designs accomplish significant reductions in power dissipation, delay and transistor count compared to an exact design; moreover, two of the proposed multiplier designs provide excellent capabilities for image multiplication with respect to average normalized error distance and peak signal-to-noise ratio. Summary: In this paper, high speed is achieved, but transistor count is more

3. PROPOSED WORK

The proposed method focuses on the design and implementation of a 64-bit approximate multiplier and 64 bit Exact multiplier that utilizes a combination of high-order compressors — specifically 8:2, 4:2, and 3:2 compressors — to improve speed and power efficiency at the cost of a small reduction in accuracy. This design is particularly suited for error-tolerant applications such as image processing and digital signal processing (DSP), where a trade-off between accuracy and hardware efficiency is acceptable.

1. Motivation Traditional multiplier architectures for higher-bit computations involve numerous addition stages to compress the partial products, resulting in increased critical path delay, power consumption, and silicon area. The use of approximate computing, combined with high-order compressors, offers a solution by simplifying the hardware and

reducing the number of intermediate stages.

2. Compressor Hierarchy and Selection The core idea is to reduce the number of partial product reduction stages by using high-order compressors capable of handling more input bits per compression unit. The proposed hierarchy includes:

- **8:2 Compressor:** Takes 8 input bits and produces 2 outputs (sum and carry), significantly reducing the depth of the reduction tree. It is used in early reduction stages where more partial products are available.
- **4:2 Compressor:** Used in the middle stages to further compress the already reduced partial products.
- **3:2 Compressor (Full Adder):** Utilized in the final reduction stages to prepare inputs for the final adder. These compressors are designed using approximate logic principles, where some carry outputs or sum bits are simplified to reduce logic complexity, resulting in lower power and faster propagation delay

3. Partial Product Generation The partial products are generated using the AND gate array method, where each bit of the multiplicand is ANDed with each bit of the multiplier, generating a total of $64 \times 64 = 4096$ bits. These are arranged in a matrix form for reduction.

4. Partial Product Reduction using Compressors Instead of using traditional Carry Save Adders or just 3:2 compressors, the partial products are fed

through the following stages:

Stage 1: 8:2 compressors are used to aggressively reduce partial products across rows.

Stage 2: The outputs of 8:2 compressors are further compressed using 4:2 compressors.

Stage 3: Remaining bits are reduced using 3:2 compressors to align the result into two rows (sum and carry). This optimized multi-stage reduction strategy minimizes the number of clock cycles and critical path delay.

5. Final Addition Stage After reduction, the final two rows are added using a fast approximate adder, such as a Carry Lookahead Adder (CLA) or an Approximate Carry Select Adder (ACSA), to obtain the final product.

1.16-bit multiplier with exact 8:2 compressor The proposed exact XOR-MUX 8:2 compressor is used to build the 16x16 multiplier as shown in below figure.

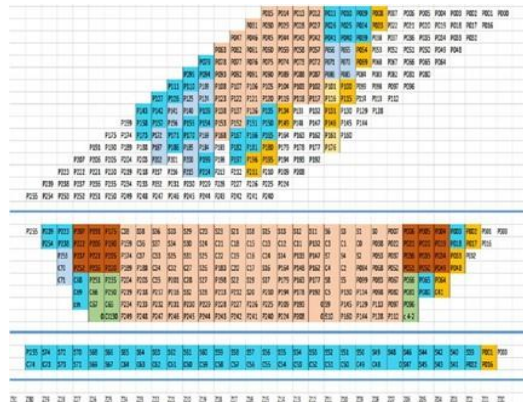


Figure 1: 16x16 multiplier with 8:2 compressor

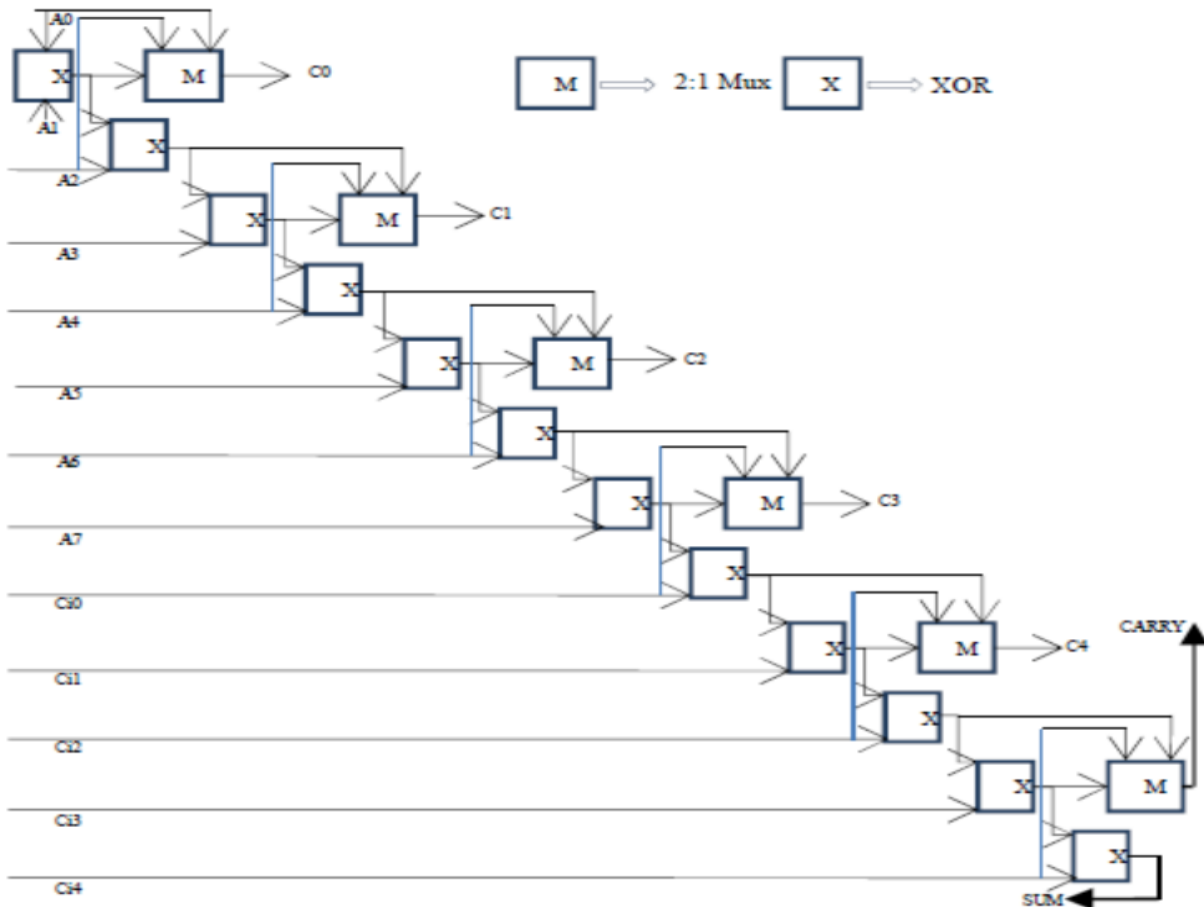


Figure 2 :8:2 XOR-MUX based compressor

Presented in this this in a lower-order compressor with tolerable error and used in the construction of higher-order compressors , so the erroneous in the higher- order compressor is not calculated accurately. This work overcomes the above problem by creating the architecture comparing all the inputs to all the outputs for the accurate calculation of error that can be created by approximating in any part of the circuit. adder and lower order compressors The compressors are used in the multiplier for the tree reduction stage usually made up of full-adder. The full-adder is named as 3:2 compressors or counter is usually used for the construction for any higher-order compressor. One full-adder will be constructed with two XOR's and one MUX is proposed so as reduce the area and power without any change in the truth table.

The multiplier with 8:2 compressor has only three reduction stages is less than the 4:2 compressor has the reduction stage of four, thereby it is efficient to use higher-order compressor if the multiplier width is increased. In Fig.1 the color code is used to identify different components: pink – 8:2 compressors, red – 4:2 compressors, thick and light blue – full-adders

2.16x16 multiplier with approximate 8:2 XOR mux based compressor With the approximation finder method, many inputs are matched with outputs for 75% of input combination. In our proposed approximate 8:2 compressor cin4 is bypassed to carry, so as to reduced area, power, delay without affecting the image quality. The approximate computing is the major concern in reducing the power, area, and delay. Novel approximation technique is presented in this paper. The proposed 8:2 compressors consist of 13 inputs and having $2^{13} = 8192$ input combinations. The circuit consists of 7

output $c_{out0} - c_{out4}$, sum, carry. The previous work in approximation is done in a lower-order compressor with tolerable error and used in the construction of higher-order compressors, so the erroneous in the higher-order compressor is not calculated accurately. This work overcomes the above problem by creating the architecture comparing all the inputs to all the outputs for the accurate calculation of error that can be created by approximating in any part of the circuit. The flow chart shown which demonstrates the correlation of every input to every output. The design of several 15:4, 9:4, 8:2 compressors are proposed respectively. All these designs are developed using full-adder and lower order compressors. The approximation feasibility overall this design is very low. The Proposed 8:2 compressor is designed with the straight forward approach with the parallel stream of input to output through XOR – MUX architecture as demonstrated. The compressors are used in the multiplier for the tree reduction stage usually made up of full-adder. The full-adder is named as 3:2 compressors or counter is usually used for the construction for any higher-order compressor. One full-adder will be constructed with two XOR's and one MUX is proposed so as reduce the area and power without any change in the truth table.

3. The figure illustrates the compressor-tree based partial product reduction structure used in the proposed multiplier architecture. It demonstrates how partial products are grouped column-wise and reduced using Exact 4:2 compressors, along with full adders and half adders, across successive stages. Although a smaller structure is shown for clarity, the same reduction principle is applied in the proposed 16×16 multiplier design. The multi-stage compression process reduces the number of partial product rows to two final operands, which are then summed using a carry-propagate adder to obtain the final product. This approach minimizes the critical path delay while ensuring exact multiplication results.

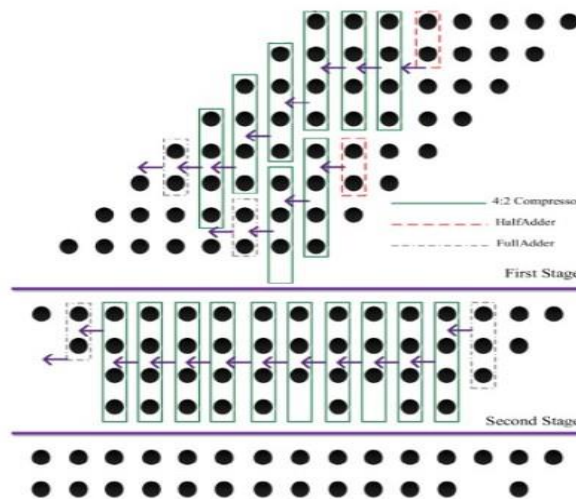


Figure3:8x8 multiplier with exact 4:2 compressor

4.Results:

The obtained RTL schematics confirm the proper structural realization of the compressor-tree based reduction network and final carry-propagate addition stage. demonstrating reduced logic complexity and improved hardware efficiency. This reduction is achieved by selectively employing approximate compressor units in less significant bit positions while preserving exact computation in the more significant bit regions, thereby maintaining acceptable output accuracy. Furthermore, the compressor-based reduction approach shortens the critical path by minimizing the number of sequential addition stages, which contributes to higher operating speed. Overall, the experimental results indicate that the proposed multiplier architecture provides a favorable trade-off between accuracy, area, and performance, making it well suited for error-tolerant applications such as image processing, signal processing, and multimedia computing.

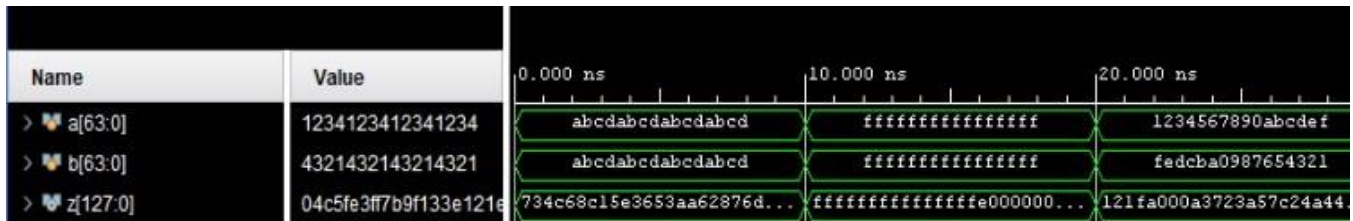


Figure 4 : Simulation Waveform for 64 bit Exact Multiplier using higher order compressors

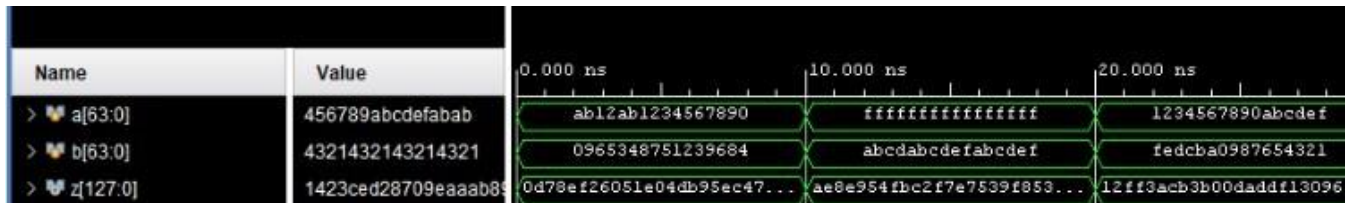


Figure 5: Simulation Waveform for 64 bit Approximate Multiplier using higher order compressor

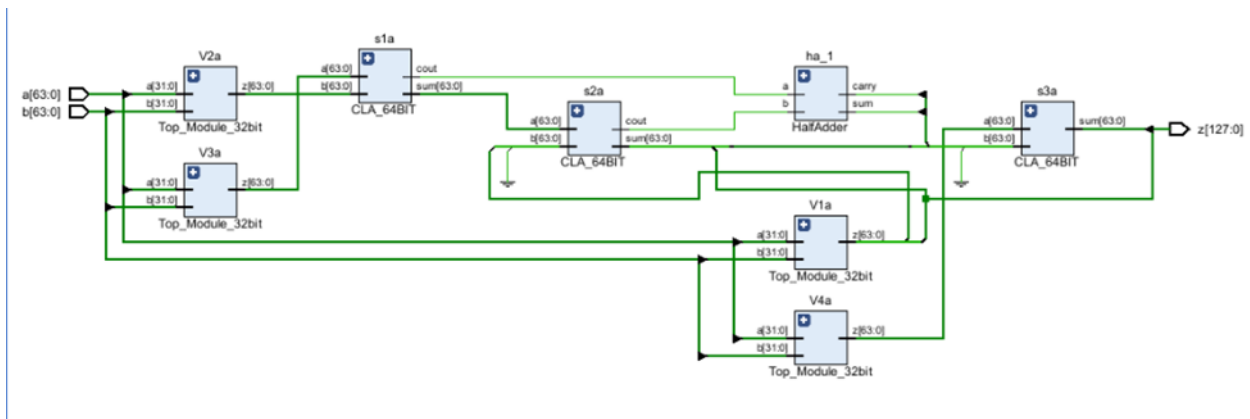


Figure 6: RTL Schematic for 64 bit Exact Multiplier using higher order compressor

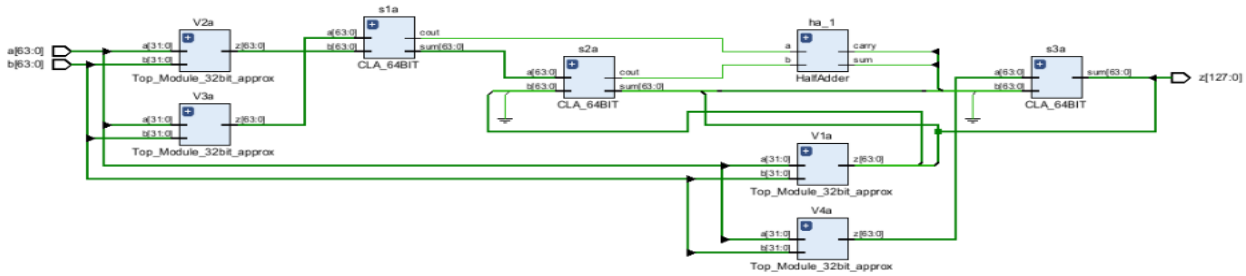


Figure 7: RTL Schematic for 64 bit Approximate Multiplier using higher order compressors

Table 1. Delay Analysis of Exact vs Approximate Multiplier Architectures

Multiplier Type	Total Delay (ns)	Logic Delay (ns)	Net Delay (ns)
Exact Multiplier	28.54 ns	7.44 ns	21.09 ns
Approximate Multiplier	28.36 ns	7.31 ns	21.05 ns

6. CONCLUSIONS

The proposed 64-bit multiplier design, implemented using 32-bit submodules and optimized with compressors/adders, demonstrates efficient architecture for high-speed arithmetic operations. By replacing conventional carry look-ahead adders with compressor-based structures, the design achieves fast partial product accumulation, reduced propagation delay, and improved Performance scalability. The modular approach using 32-bit multipliers allows easier synthesis, verification, and hardware realization on FPGA or ASIC platforms. This architecture is highly suitable for applications in digital signal processing, image processing, and cryptographic systems where both speed and accuracy are critical.

6. REFERENCES

- [1] Gorantla, A. and Deepa, P., 2019. Design of Approximate Subtractors and Dividers for Error Tolerant Image Processing Applications. Journal of Electronic Testing, pp.17.
- [2] Kim, Y., Zhang, Y. and Li, P., 2014. Energy efficient approximate arithmetic for error resilient neuromorphic computing. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 23(11), pp.2733-2737
- [3] Zhou, Y., Lin, J., Wang, J. and Wang, Z., 2018, October. Approximate Comparator: Design and Analysis. In 2018 IEEE International Workshop on Signal Processing Systems (SiPS) (pp. 1-5). IEEE.
- [4] Monajati, M., Fakhraie, S.M. and Kabir, E., 2015. Approximate arithmetic for low-power image median filtering. Circuits, Systems, and Signal Processing, 34(10), pp.3191-3219

-
- [5] Chang, C.H., Gu, J. and Zhang, M., 2004. Ultra low-voltage low-power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 51(10), pp.1985-1997.
- [6] Taheri, M., Arasteh, A., Mohammadyan, S., Panahi, A. and Navi, K., 2020. A novel majority based imprecise 4: 2 compressor with respect to the current and future VLSI industry. *Microprocessors and Microsystems*, 73, p.102962.
- [7] Moaiyeri, M.H., Sabetzadeh, F. and Angizi, S., 2018. An efficient majority-based compressor for approximate computing in the nano era. *Microsystem Technologies*, 24(3), pp.1589-1601.
- [8] Gorantla, A., 2017. Design of approximate compressors for multiplication. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 13(3), pp.117.
- [9] Marimuthu, R., Rezinold, Y.E. and Mallick, P.S., 2016. Design and analysis of multiplier using approximate 15-4 compressor. *IEEE Access*, 5, pp.1027-1036.
- [10] Guo, Y., Sun, H., Guo, L. and Kimura, S., 2018, October. Low-cost approximate multiplier design using probability-driven inexact compressors. In *2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)* (pp. 291-294). IEEE.
- [11] Marimuthu, R., Bansal, D., Balamurugan, S. and Mallick, P.S., 2013. Design of 8-4 and 9-4 Compressors For high Speed Multiplication. *American Journal of Applied Sciences*, 10(8), p.893.