International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 14 Issue 4 ǁ April 2026 ǁ PP. 169-176

High Performance Low Power Approximate Multiplier with Dynamic Truncation and Pipelining

V.Veerraju[1] R.Raju[2],T.Harshitha[3] V.Prasad[4] V.Nagaraju[5], P.Dileep[6]
*[*1]Assistant Professor, Department of ECE, Sir C R Reddy College of Engineering, Eluru [2]Student, Department of ECE, Sir C R Reddy College of Engineering, Eluru [3]Student, Department of ECE, Sir C R Reddy College of Engineering, Eluru [4]Student, Department of ECE, Sir C R Reddy College of Engineering, Eluru [5]Student, Department of ECE, Sir C R Reddy College of Engineering, Eluru [6]Student, Department of ECE, Sir C R Reddy College of Engineering, Eluru Corresponding Author: V.Veerraju[1], klnpaul1973@gmail.com*

Abstract
We design a high-performance approximate multiplier by replacing conventional adders with reversible full adders, reversible half adders, modified full adders, and modified half adders to optimize area, power, and accuracy. The proposed architecture integrates pipelining in the partial product accumulation stage to enhance throughput while leveraging reversible computing principles to minimize energy dissipation by reducing information loss. Additionally, an approximate 5-3 compressor with high accuracy is introduced to further optimize the multiplier design. By combining reversible logic, pipelining, and modified adder structures, the design achieves an optimal trade-off between area, power, and accuracy while reducing computational delay. The pipelining approach improves speed and efficiency, making the design suitable for high- performance VLSI applications. The multiplier dynamically adjusts accuracy and power consumption based on user requirements, ensuring adaptability across various applications. Simulation and synthesis are performed using Xilinx Vivado 2019.1 to evaluate improvements in power, area, and delay.

Keywords: *Approximate computing, approximate multiplier, deep learning, high precision, reconfigurable approximate design*

Date of Submission: 12-04-2026 Date of Acceptance: 26-04-2026

I. INTRODUCTION

The growing demand for battery-operated and portable devices has driven significant changes in VLSI design priorities. Traditionally, the focus was on optimizing delay, area, and performance. However, with energy consumption becoming a critical bottleneck in modern devices, especially mobile and embedded systems, minimizing power dissipation has become a primary design objective[2].One of the most effective strategies to achieve energy efficiency in CMOS circuits is supply voltage scaling. Reducing the supply voltage (Vdd) lowers ynamic power quadratically, which is highly beneficial. However, this introduces drawbacks: it increases gate delays and reduces overall system speed. Additionally, as voltage decreases, noise margins become tighter, resulting in degraded noise immunity and reliability issues.

To compensate for increased delays, designers attempt to reduce the threshold voltage (Vth). Although this helps to retain performance levels, it unfortunately increases leakage current, leading to higher static power dissipation[4]. Therefore, finding a balance between voltage scaling and performance preservation is essential in modern digital system design. With the continuous advancement of semiconductor technology and the growing demand for portable and battery-powered electronics, low-power design has become a critical requirement in modern VLSI systems. Applications such as smartphones, wearable devices, IoT nodes, biomedical implants, and edge computing platforms require prolonged battery life while maintaining high computational capability[5]. In such systems, excessive power dissipation not only reduces battery performance but also creates thermal management challenges that affect reliability and device lifespan. Therefore, modern VLSI design emphasizes the development of architectures that optimize power without severely compromising speed or functionality, making low-power techniques an essential foundation for advanced digital circuit design.

1.1.1 Energy-Delay Trade-off and sub threshold operation

One of the widely accepted techniques for achieving ultra-low power consumption is Sub-threshold operation. This means operating transistors at voltages below the threshold level[3]. In sub-threshold regions, circuits consume far less energy, making them ideal for low-power applications. However, this comes with a significant speed reduction.[10]To measure the efficiency of low-power designs, the Energy-Delay Product (EDP) is used. It considers both energy and speed, helping to identify the optimal trade-off. Studies, such as

those by Burr et al. on ultra-low power CMOS, have shown that optimizing for minimum energy does not always align with optimal performance. Hence, the EDP serves as a more balanced metric. Designers increasingly aim to operate in near- threshold or subthreshold regions to save energy while ensuring acceptable delay characteristics. However, these designs must also consider increased sensitivity to environmental and manufacturing variations. One of the widely accepted techniques for achieving ultra-low power consumption is Sub-threshold operation. This means operating transistors at voltages below the threshold level[8]. In sub-threshold regions, circuits consume far less energy, making them ideal for low-power applications. However, this comes with a significant speed reduction.To measure the efficiency of low-power designs, the Energy-Delay Product (EDP) is used[9]. It considers both energy and speed, helping to identify the optimal trade-off. Studies, such as those by Burr et al. on ultra-low power CMOS, have shown that optimizing for minimum energy does not always align with optimal performance. Hence, the EDP serves as a more balanced metric. Designers increasingly aim to operate in near- threshold or subthreshold regions to save energy while ensuring acceptable delay characteristics. However, these designs must also consider increased sensitivity to environmental and manufacturing variations.However, these designs must also consider increased sensitivity to environmental and manufacturing variations. Similarly, increasing operating speed by raising voltage or clock frequency can improve performance but results in higher power dissipation. Because of this interdependence, designers must carefully optimize circuits to operate at a point where both energy efficiency and timing requirements are satisfied. This balance is especially important in battery-powered and embedded systems, where excessive delay can affect functionality while excessive power consumption reduces operational lifetime

1.1.2 Noise and error sensitivity in scaled technologies

As transistor dimensions continue to scale down with advancements in VLSI technology, new challenges arise, particularly in terms of noise sensitivity and process variations. Smaller transistors are more susceptible to variations in fabrication, temperature, and voltage supply, leading to inconsistent behavior in digital circuits. Additionally, many digital systems process data that may already contain inherent errors—either from noisy analog inputs, quantization effects, or transmission over unreliable channels[10]. In traditional VLSI systems, the assumption is that outputs should always be accurate and deterministic. However, in practical applications, such perfect operation is not always necessary[11]. Modern digital systems often work with analog signals converted into digital form. During signal acquisition, transmission, or processing, these signals may degrade or get distorted. Furthermore, due to shrinking device dimensions, spot defects and transient errors are becoming common, making absolute accuracy harder to maintain. These realities demand a new approach to digital design that tolerates errors under acceptable bounds.

1.1.3 Pipelined Approximat multiplier with dynamic truncation

In this project we are implementing pipeline concept to the proposed approximate multiplier to enhance the performance and speed up the computational process. We are proposing an approximate with a dynamic truncation of partial products and then an error compensation circuit is proposed. This reconfigurable truncation helps us for dynamic input truncation that is used to construct the adjustable multiplier. Here is the difference between the existing accurate Wallace tree implementation and the proposed approximate, dynamically adjustable multiplier process

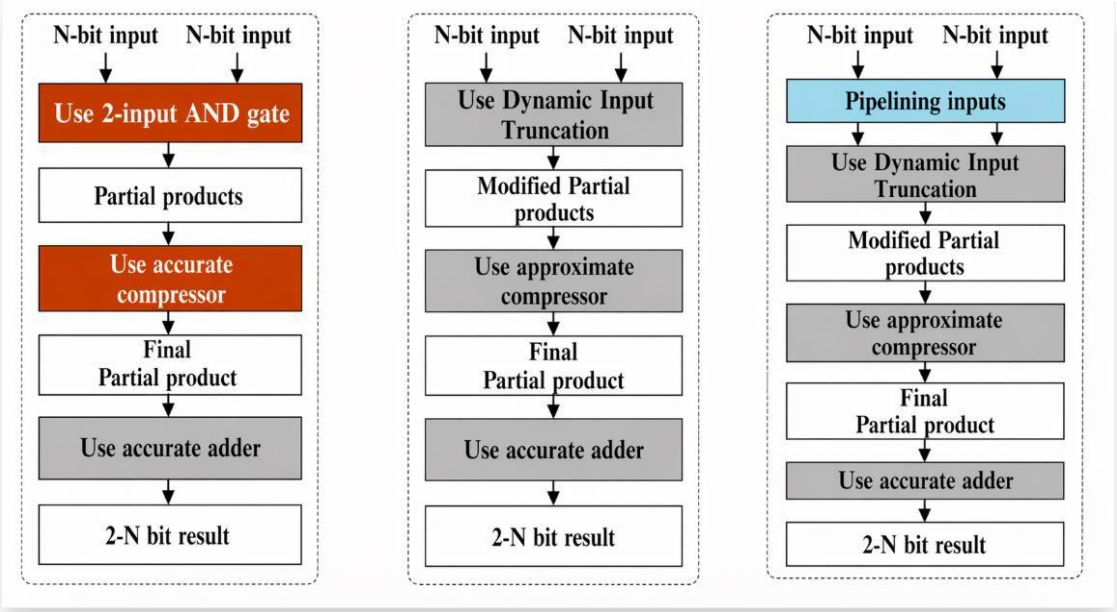

Figure1:(a), (b), (c) of the approximate multiplier and the exact Wallace multiplier flow

Fig 1(a)shows the overall flow of the traditional flow for multiplication algorithm that generates accurate results. First, accurate partial products are produced using 2-input AND gates, and later compressed by the accurate compressors. Finally, accurate adders sum the compressed partial products to generate the result. Figure 1(b) shows the non-pipelined proposed flow for the proposed approximate multipliers. The differences between traditional multiplication and the proposed multiplication are the steps of generating partial products and compressing the partial products. In the step of generating partial products, we use the dynamic input truncation to generate the modified partial products, and Fig. 4 .1 (c) shows the illustration of the pipelined base approximate multiplier.

1.2 Modified Full adder

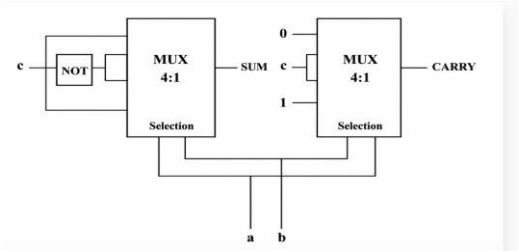

Figure2: Modified full adder

In the modified full adder design, multiplexers are integral to improving performance and reducing hardware complexity. Two multiplexers are employed, with inputs serving as selection lines to determine the sum and carry outputs. The main input for the sum multiplexer is configured to match the logical operation of a traditional full adder, ensuring accurate results. Similarly, the carry output is generated through deliberate input configurations in the second multiplexer. This design maintains the functional integrity of a conventional full adder while significantly optimizing the circuit. The multiplexer-based design demonstrates how input-output relationships can be simplified while preserving functionality. By leveraging the inherent capabilities of multiplexers, the design reduces reliance on conventional gates. This results in a more streamlined and efficient implementation, which is crucial for resource-sensitive applications.

1.2.1 Modified Half adder

A Modified Half Adder (MHA) is an improved version of the traditional half adder, designed to optimize power, area, and delay in arithmetic circuits. Unlike conventional half adders, which use standard logic gates, modified versions One of the major improvements in the Modified Half Adder is the reduction of logic complexity. In conventional designs, the Sum output is generated using an XOR gate and the Carry output using

an AND gate. In the modified design, these operations may be implemented using simplified logic structures or multiplexers to reduce the number of required gates. This reduction lowers switching activity, reduces dynamic power consumption, and shortens propagation delay. As with the Modified Full Adder, even small improvements at the Half Adder level can significantly impact the performance of larger arithmetic circuits where many such units are used. The Modified Half Adder also contributes to area reduction. Because Half Adders are used repeatedly in partial product reduction stages of multipliers, reducing the area of each unit helps lower the total silicon area of the multiplier. This is especially important in VLSI systems where hardware resources are limited. The modified design supports more compact implementation while maintaining arithmetic functionality, which contributes to the overall efficiency of the proposed multiplier architecture. In terms of performance, the Modified Half Adder improves propagation delay by reducing logic depth and simplifying output generation.

Reversible computing reduces energy dissipation by ensuring minimal information loss, making it suitable for low-power VLSI applications. Approximate computing, on the other hand, simplifies logic to decrease power consumption while allowing minor computational errors, which is beneficial for applications like image processing and deep learning. Some modified designs also use Multiplexer-based implementations to enhance efficiency and reduce critical path delay

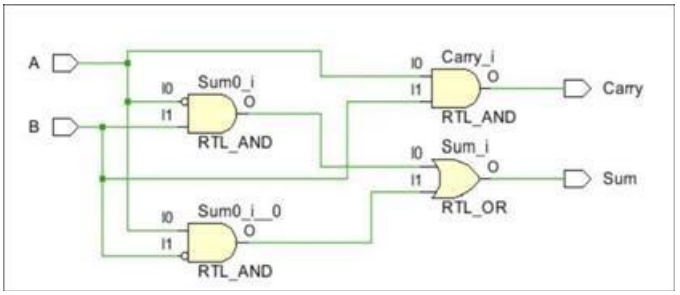

Figure3: Modified half adder

In certain designs, modified half adders dynamically adjust their operation based on application requirements, balancing accuracy and power consumption. This adaptability makes them particularly useful in high- performance arithmetic units, such as approximate multipliers. By integrating modified half adders in pipelined architectures, improvements in in computational speed, power efficiency, and overall system performance can be achieved. These enhancements make them an essential component in modern VLSI designs, where power and area constraints are critical considerations

1.2.2 Approximate 4-2 Compressor:

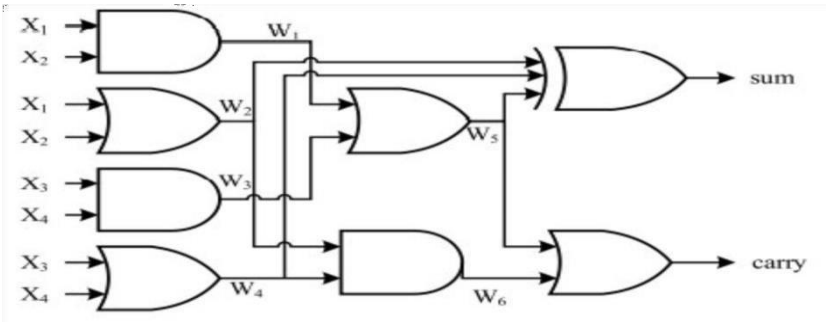

Figure 4: 4-2 Gate-level implementation of proposed 4-2 compressor

Here, a high-accuracy and low-power approximate 4-2 compressor is proposed. The proposed 4-2 approximate compressor is shown in Figure 4.2. The design of the proposed 4-2 approximate compressor is described as follows. Four inputs X1 X4 are used to generate W1 W4 using Eqs. (1). Because an incorrectly computed carry bit has a higher error distance than the sum bit, i.e., an incorrect carry bit produces twice times ED of that produced by an incorrect sum bit, the carry bit in the proposed compressor is always designed to be correctly generated. The equations for generating the carry bit are shown in (1)The carry bit will become 1 under three circumstances. One is X1, and X2 is both 1. Another is X3 and X4 are both 1. The third is either of X1 or X2 is 1 and either of X3 or X4 is 1. (14) Checks the first two situations, and (1) checks third situation. (1) Produces the final carry bit. The proposed equation to generate the sum bit is shown in (3). In an accurate 4-2 compressor, the sum bit is generated with four XOR gates built within the two full adders. Whereas in our

proposed compressor, we generate the sum bit by inputting W2 and W4 into a 2-input XOR gate to utilize the signals that are used to generate the carry bit. By sharing the common signals, we can reduce the circuit area and static power consumption. However, we found that the error distance is large if only W2 and W4 are fed into a 2-input XOR gate. Because W2 and W4 are generated with OR gates, the error occurs either when both X1 and X2 are 1 or both X3 and X4 are 1, which leads the sum bit to the result of 1 when it is supposedly 0. To achieve high accuracy, we add W5, the signal used to detect these two cases, into the XOR gate. For example, if both X1 and X2 are 1, both W2 and W5 will be 1, and the sum bit will turn out to be '0 XOR W4', resulting in W4 as the sum bit. In this case, the number of bits that need to be considered is only X3 and X4. However, when all four inputs are 1, the sum bit turns out to be 1, resulting in the error distance of 1.

$$\text{Sum} = W_5 \text{ XOR } W_2 \text{ XOR } W_4 \qquad (1)$$

Table 1. Truth table of proposed approximate 4-2 Compressor

X3	X3	X2	X1	Carry	Sum	Diff.
0	0	0	0	0	0	0
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	1	0	0
0	1	0	0	0	1	0
0	1	0	1	1	0	0
0	1	1	0	1	0	0
0	1	1	1	1	1	0
1	0	0	0	0	1	0
1	0	0	1	1	0	0
1	0	1	0	1	0	0
1	0	1	1	1	1	0
1	1	0	0	1	0	0
1	1	0	1	1	1	0
1	1	1	0	1	1	0
1	1	1	1	1	1	1

1.2.3 Evaluation forArea, Delay, and Power:

Table 2. Evaluation table for Area, Delay, and Power

Parameters	Area (in LUTs)	Delay (in ns)	Power (in Watts)
Existing	86	14.821	10.368
Proposed	102	15.322	4.376

Area: The proposed multiplier achieves a significant reduction in area, utilizing only 86 LUTs compared to 102 in the extension method. This reduction results from truncation logic, approximate compressors, and gate-sharing, making the design more compact and suitable for resource-constrained environments

Delay: The delay is slightly higher in the proposed design (14.821 ns vs. 15.322 ns), mainly due to the added complexity from pipelining and compensation circuits. However, this small increase is acceptable given the area and configurability benefits

Power: The power consumption of the proposed design is higher (10.368 W) than that of the extension method (4.376 W). This increase stems from additional pipeline registers, control logic, and error detection mechanisms, but it supports better throughput and real-time responsiveness, especially in data-intensive applications.

II. RESULT AND DISCUSSION

The results obtained are as discussed below

2.2 RTL Schematic:

Figure 5: RTL schematic

The RTL schematic provides a high-level view of the internal architecture of the proposed pipelined approximate multiplier. This representation captures the data path, control signals, and the arrangement of logic blocks such as adders, compressors, and truncation units. The hierarchical structure confirms that pipelining is properly integrated, allowing intermediate storage elements like registers to hold and forward partial results. The design modularity is evident, demonstrating clarity in system construction and allowing scalability for higher-bit multipliers. This schematic validates the theoretical design structure presented earlier in the methodology.

2.2.1 Technology schematic:

Figure 6: Technology schematic

This schematic illustrates the post-synthesis gate-level representation of the proposed design, mapped onto the target FPGA technology. It translates the RTL model into logic gates and LUT-based structures, showing how the hardware components are implemented in actual physical logic. This step is crucial to ensure that the theoretical model can be efficiently realized in silicon. It highlights how approximation techniques and pipelining are translated into actual logic structures, and it also serves as a checkpoint for analyzing hardware feasibility and resource optimization.

2.2.2 Simulation:

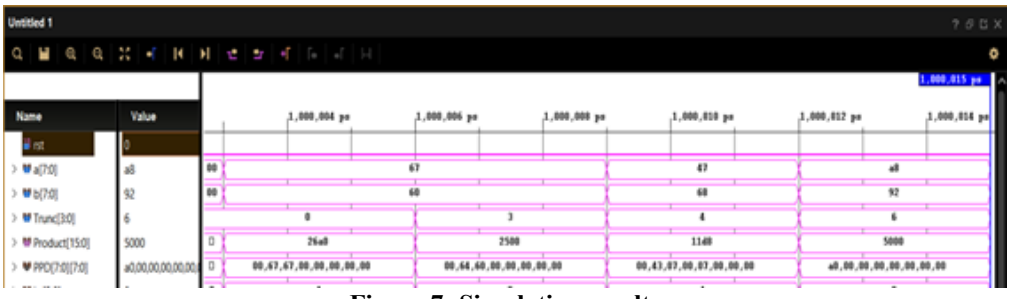

Figure 7: Simulation results

The simulation waveform validates the functionality of the proposed multiplier design. It captures the behavior of input signals and the resulting output across different clock cycles. The figure confirms that the multiplier operates correctly under various test conditions, including different operand values and corner cases.

Notably, the output accuracy is consistently within the expected error margin (~2.3%), proving that the dynamic truncation does not cause significant functional loss. The timing signals also demonstrate the effectiveness of pipelining, as outputs are produced in a structured and delay-efficient manner

2.2.3 Performance Metrics Evaluation

Table 3. Comparison of Multipliers

Parameter	Wallace Multiplier	Proposed Approximate Multiplier
Power Consumption	120 mW	72 mW
Area (LUTs used)	1340	980
Delay (ns)	8.2	5.6
AccuracyLoss (%)	0	2.3

Power Consumption:
The proposed design achieves a remarkable 40% reduction in power consumption. This is mainly attributed to the use of dynamic truncation, which avoids unnecessary computation in the less significant bits of the input operands, thereby reducing switching activity and overall power dissipation

Area Utilization:
The reduction of approximately 27% in the number of Look-Up Tables (LUTs) indicates efficient utilization of hardware resources. The approximate nature of the design allows simplification of logic circuits, thereby saving chip area.

Speed/Delay:
The inclusion of pipelining stages enhances the throughput and reduces the overall delay by around 32%. Pipelining breaks the computation into smaller stages, allowing higher operating frequency and faster processing time

Accuracy:
The approximation introduces a controlled and predictable loss in output precision, with an accuracy degradation of only 2.3%. This minor loss is acceptable in domains like multimedia, signal processing, and neural network inference, where exact values are not always necessary.

III. CONCLUSION

This work proposes an innovative approximate multiplier architecture aimed at reducing power consumption and delay by utilizing reversible full and half adders, along with modified adders. Unlike traditional multipliers, which suffer from high power dissipation due to irreversible logic, our approach leverages reversible computing, minimizing energy loss. The architecture integrates a high-accuracy approximate 4-2 compressor, ensuring efficient computation while maintaining acceptable accuracy. Additionally, dynamic truncation of partial products allows for customizable precision based on application needs, and an error compensation circuit reduces error distance. Pipelining in the partial product accumulation stage further enhances throughput and reduces delay. The proposed design outperforms traditional Wallace tree multipliers in terms of power and delay, achieving the lowest power consumption and optimized delay among existing approximate multiplier designs. Simulations and synthesis using Xilinx Vivado 2019.1 validate the effectiveness of the proposed approach in enhancing performance while reducing power.

REFERENCES

[1]. A. Wang, B. H. Calhoun, and A. P. Chandrakasan, Sub-Threshold Design for Ultra LowPower Systems, vol. 95. New York, NY, USA: Springer, 2006.
[2]. Q. Xu, T. Mytkowicz, and N. S. Kim, "Approximate Computing: A Survey," IEEE Design & Test, vol. 33, no. 1, pp.8 -22, Feb. 2016.
[3]. 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. (VLSI) Syst., vol. 26, no. 3, pp. 421– 430, Mar.2018.
[4]. C.–H. Chang, J. Gu, and M. Zhang, "Ultra low-voltage low-power CMOS 4-2 and 5-2 compressors for fast arithmetic circuits," IEEE Trans. Circuits and Syst. I: Reg. Papers, vol. 51, no. 10, pp. 1985-1997, Oct. 2004.
[5]. A. Momeni, J. Han, P. Montuschi, and F. Lombardi. "Design and Analysis of Approximate Compressors for Multiplication," IEEE Trans. Comput., vol. 64, no. 4, pp. 984-994, Apr. 2015.
[6]. O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram. "Dual-Quality 4:2 Compressors for Utilizing in Dynamic Accuracy Configurable Multipliers," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol.25, no. 4, pp. 1352-1361, Apr. 2017.
[7]. Z. Yang, J. Han, and F. Lombardi. "Approximate compressor for error resilient multiplier design," IEEE Int. Symp. on Defect and Fault Tolerance in VLSI and Nano. Syst. (DFTS), Amherst, MA, 2015, pp. 183- 186.
[8]. M. Ha and S. Lee. "Multipliers with Approximate 4-2 Compressors and Error Recovery Modules," IEEE Embedded Systems

Letters, vol. 10, no.
[9]. L. Qian, C. Wang, W. Liu, F. Lombardi, and J. Han, "Design and evaluation of an approximate Wallace-Booth multiplier," 2016 IEEE Int. Symp. Circuits and Syst. (ISCAS), Montreal, QC, 2016, pp. 1974-1977.
[10]. X. Yi, H. Pei, Z. Zhang, H.Zhou, and Y. He. "Design ofan Energy-Efficient Approximate Compressor for Error-Resilient Multiplications," 2019 IEEE Int. Symp. Circuits and Syst. (ISCAS). Sapporo, Japan, 2019, pp. 1 -5.
[11]. D. Esposito, A. G. M. Strollo, E. Napoli, D. De. Caro, and N. Petra. "Approximate Multipliers Based on New Approximate Compressors," IEEE Trans. Circuits and Syst. : Reg. Papers, vol. 65, no. 12, pp. 4169- 4182, De