IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Design Of High-Speed Accuracy Controllable Of 32-Bit Approximate Multiplier

Y. Sri Chakrapani
Associate Professor,
Dept. of E.C.E, Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru, India,

A. Supriya
B.Tech Student, Dept. of ECE, Seshadri Rao
Gudlavalleru Enginnering College, Gudlavalleru, India.

B. Prasad
B.Tech Student, Dept. of
ECE, Seshadri Rao
Gudlavalleru Enginnering College, Gudlavalleru, India.

B. Mahendra
B.Tech Student, Dept. of ECE, Seshadri Rao
Gudlavalleru Enginnering College, Gudlavalleru, India.

G. Aruna
B.Tech Student, Dept. of ECE, Seshadri Rao
Gudlavalleru Enginnering College, Gudlavalleru, India.

Abstract: Multiplication is a basic operation in numerous error-resilient applications, especially in digital signal processing, and image processing. Approximate multiplication offers a promising approach to balancing energy efficiency, computational performance, and accuracy. This paper presents a 32-bit accuracy-controllable approximate multiplier that leverages a Carry-Save Adder (CSA) while replacing the conventional Incomplete Adder Cell (iCAC**)** to enhance performance. The design incorporates an Approximate Tree Compressor (ATC) combined with a Carry-Maskable Adder (CMA) enabling dynamic accuracy adjustment while enhancing energy efficiency compared to Incomplete Adder-based Approximate Multipliers. The effectiveness of the design is validated through image processing applications, where minor computational errors have minimal perceptual impact. Experimental results that the show proposed multiplier significantly improves the balance between power usage as well as performance while maintaining high-quality image outputs, the proposed architecture achieves a 56% reduction in power consumption by using a carry save adder and carry maskable adder

Keywords**:** Multiplication, Approximate tree compressor, Carry save adder, Carry maskable adder, Multiplier

1. INTRODUCTION

Multiplication is a key arithmetic operation for reducing its accuracy and can significantly lower power consumption. Approximate computing offers an effective solution by balancing accuracy and energy efficiency, making it well-suited for Error-resilient applications. Multiple sectors, including image processing and recognition, can tolerate minor inaccuracies while requiring high computational power. Error-tolerant applications have accuracy requirements across different processing phases. Adapting the multiplier's

precision based on these needs can significantly reduce power consumption. Therefore, approximate multipliers should support dynamic reconfiguration to balance energy efficiency and computational accuracy. This paper presents a dynamically reconfigurable approximate multiplier design that adjusts accuracy as needed. It introduces **a** carry-maskable adder (CMA)**,** which can operate as a conventional carry-save adder (CSA) by carry propagation. Instead of a standard carry propagation adder (CPA) in the final stage, the proposed design integrates the CMA to enhance flexibility. Additionally, An Approximate Tree Compressor (ATC) is employed to minimize the depth of the partial product accumulation tree, Improving overall efficiency. This paper introduces a multiplier capable of dynamically adjusting and Optimizing the partial product reduction (PPR) process to balance power and accuracy requirements. The proposed approximate multiplier, which integrates a carry-save adder, was developed alongside a conventional multiplier and previously studied approximate multipliers based on inomplete adders. The implementation was carried out using Verilog HDL on Xilinx software.

The design was analyzed for power efficiency, showing a 47.3%–56.2% reduction in power consumption compared to traditional Wallace Tree multipliers and approximate multipliers utilizing incomplete adders. Among the tested approximate designs, the proposed approach achieved the best trade-off between power efficiency and accuracy. Its effectiveness was further validated through real-world image processing applications.

The paper is structured as follows: reviews related work, and details the proposed multiplier, including the tree compressor and Carry-Maskable Adder (CMA), and presents experimental evaluations of different multipliers, followed by their application in image processing. Finally, concludes the study.

2. LITERATURE SURVEY

Adders play a crucial role in multipliers, significantly influencing power consumption, processing speed, and overall accuracy. Numerous researchers have explored various techniques to enhance the efficiency of multipliers through approximate computing.

Mahdiani et al. [2] proposed a Lower-Part-OR Adder, which integrates OR gates for lower-order bit addition while using accurate adders for higher-order computations. This method aims to minimize power consumption while ensuring reasonable accuracy. The proposed Carry-Maskable Adder (CMA) shares similarities with this approach by leveraging OR gates for approximate addition. However, unlike the lower-part-OR adder, our CMA is dynamically reconfigurable**,** allowing real-time adjustments to the trade-off between accuracy and power consumption.

Moons et al[3]. investigated a system-level approach to balance accuracy and power efficiency by selectively disabling certain portions of combinational logic and dynamically reconfiguring pipeline registers and logic circuits to adjust pipeline stages. Their approach also incorporated voltage-accuracy scaling, where power consumption is optimized by reducing supply voltage based on accuracy demands. While this method provides dynamic accuracy control, it involves complex control mechanisms and additional pipelining stages, which increase design complexity and may introduce pipeline overhead. In contrast, the proposed multiplier utilizes a Carry-Save Adder (CSA) in the accumulation stage to achieve an efficient trade-off between power consumption and accuracy.

Liu et al. [4] introduced an approximate adder aimed at reducing carry propagation delay during partial product accumulation. Their method incorporates an error recovery vector, which selectively corrects errors caused by approximation, thereby enhancing precision, and offering some flexibility in accuracy control. However, despite this adaptability, the accuracy remains fixed at design time and cannot be dynamically modified, making it less flexible compared to the proposed multiplier.

Hashemi et al. [5] introduced a method to minimize the effective multiplier size by identifying the leading one in an operand and selecting a fixed number of subsequent bits as abridged operands. The value of these abridged operands is determined during the design phase, which sets the accuracy of the core multiplier. This method establishes a fixed trade-off between accuracy and power consumption but lacks adaptability. Since the operand width is predefined at design time, the multiplier cannot adjust accuracy based on runtime needs, making it less effective for applications requiring dynamic accuracy adjustments.

Bodapati et al. [6] presented an accuracy-scalable approximate multiplier designed to enhance both power efficiency and computational speed. Their approach leverages an approximation tree compressor, and incomplete adder cells, and carries maskable adders to selectively approximate computations, leading to reduced energy consumption and lower delay. By configuring accuracy at the architectural level, the design achieves significant improvements in power and performance. However, its accuracy control is confined to preset configurations, limiting adaptability to runtime variations. Although this method enhances efficiency over conventional multipliers, it may not be ideal for applications requiring dynamic accuracy adjustments.

Gu et al. [7] proposed a reconfigurable truncation approach to improve power efficiency while preserving computational accuracy in approximate multipliers. This method enables dynamic truncation adjustments based on processing requirements, allowing for an optimized balance between energy consumption and precision. By reducing switching activity and avoiding unnecessary computations, the design effectively lowers power usage. However, the efficiency of this technique depends on selecting appropriate truncation settings for different tasks. While it provides greater flexibility compared to fixed-approximation methods, frequent reconfigurations may introduce additional computational overhead in some applications.

In contrast, our proposed multiplier employs a Carry-Save Adder (CSA) in the accumulation stage to enhance speed and reduce power consumption without the need for additional pipeline control or voltage scaling mechanisms. By selectively disabling parts of the combinational logic in the Carry-Propagation Adder (CPA)**,** our design further reduces power usage while maintaining flexibility in accuracy control. Unlike previous methods, our approach enables real-time dynamic accuracy adjustment without the need for complex pipeline management, making it more efficient for power-constrained applications such as image processing and recognition.

Overall, while previous works provide various static and dynamic approximation techniques, they often require trade-offs between accuracy and power consumption that are fixed at design time or rely on complex system-level modifications.

Our proposed multiplier overcomes these limitations by offering a dynamically reconfigurable design that efficiently balances speed, power, and accuracy without introducing significant overhead**.**

3. PROPOSED APPROXIMATE MULTIPLIER

Approximate multiplier is generally composed of three main stages:

- Generating partial products using AND gates,
- Reducing partial products through an adder tree or Carry-Save Adder (CSA), and
- Final result calculation with a Carry-Maskable Adder (CMA).

3.1 Partial product generation using an AND gate

In binary multiplication**,** the first step is partial product generation, which is achieved using AND gates**.** Since binary numbers only have two values (0 and 1), multiplying two bits is straightforward as shown in Eq1. $P=Ai\times Bj$ ____Eq1

A (Multiplicand Bit)	B (Multiplier Bit)	Partial Product (A AND B)
0	0	0
0	1	0
1	0	0
1	1	1

Table 3.1. Truth Table for AND gate operation.

This operation is performed using an AND gate, as shown in the Table 3.1 below: For an N-bit multiplier, every bit of the multiplicand (A) is multiplied with every bit of the multiplier (B) using AND gates, forming an N×N matrix of partial products.

Sum Calculation: The sum of three inputs (a, b, and c) is computed using the XOR (^) operation. This avoids carrying propagation and speeds up computation.

Carry Calculation: The carry is generated using the majority function: (a & b) | (b & c) | (a & c), ensuring that the carry is set when at least two inputs are 1.

The Carry-Save Adder efficiently processes multiple operands by reducing carry propagation at each stage. Instead of immediate carry propagation, it generates intermediate sum and carry values, improving latency and power efficiency. This enhances overall multiplier performance, especially in high-speed and low-power applications.

3.2 Approximate Tree Compressor

The Approximate Tree Compressor (ATC) is designed to optimize the partial product reduction stage in multipliers by approximating carry propagation. In traditional multipliers, accurate carry propagation leads to high power consumption and delays due to the complex addition stages. The ATC simplifies this by approximating the carry operation, which reduces the logic complexity and overall computational delay. This approximation significantly lowers power consumption, as fewer logic gates are used and fewer carry bits need to be propagated. The reduced complexity also increases the speed of the multiplication process, making it more suitable for high-speed arithmetic units**.** The ATC is particularly useful in error-tolerant applications like image processing, where slight inaccuracies in the results do not affect the overall functionality.

By strategically trading off some accuracy, the ATC achieves a balance between power**,** speed**,** and accuracy**.** This makes it an excellent choice for energy-efficient systems, such as battery-powered devices and embedded systems, where reducing power usage is crucial. Its ability to speed up computations while consuming less power makes the ATC a valuable solution for modern applications requiring both performance and efficiency.

3.3 Carry Save Adder

The Carry-Save Adder (CSA**)** is used to perform efficient summation of three input bits without immediately propagating the carry. This helps reduce the delay by keeping carry propagation as shown in Fig 3.3.

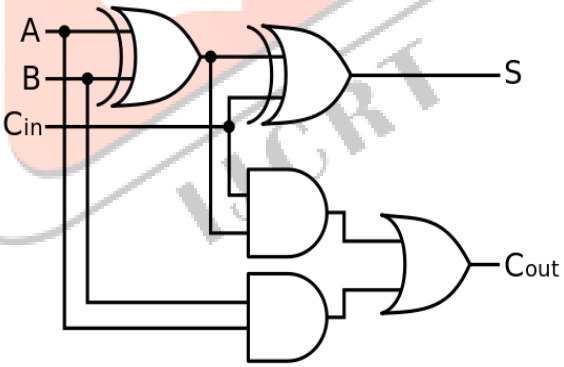

Fig 3.3. Carry save Adder

3.4 Carry Maskable Adder

The Carry-Maskable Adder (CMA) allows for dynamic control over carry propagation, enabling flexibility in balancing power and accuracy. By selectively masking carry bits when full precision is unnecessary.

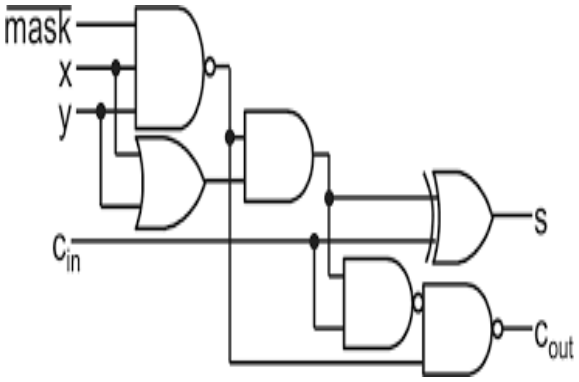

Fig 3.4 Carry maskable of full adder.

it reduces power consumption while still providing adequate accuracy for many applications as shown in Fig 3.4. This selective carry propagation makes the CMA ideal for approximate computing, where some level of imprecision is acceptable to achieve power and performance efficiency. The CMA is particularly well-suited for adaptive processing applications, where accuracy requirements can change depending on the context, such as in image processing or signal processing, where slight inaccuracies do not compromise overall functionality.

3.5 ARCHITECTURE

In Stage 1, thirty-two rows of partial products (PPs) are compressed into eleven sum rows (S1–S11) and one accuracy compensation vector (V1) using an Approximate Tree Compressor (ATC-32)**.** This reduction minimizes computational complexity by utilizing approximate addition techniques**,** including Carry-Save Adders (CSA) as shown in Fig 3.5.1.

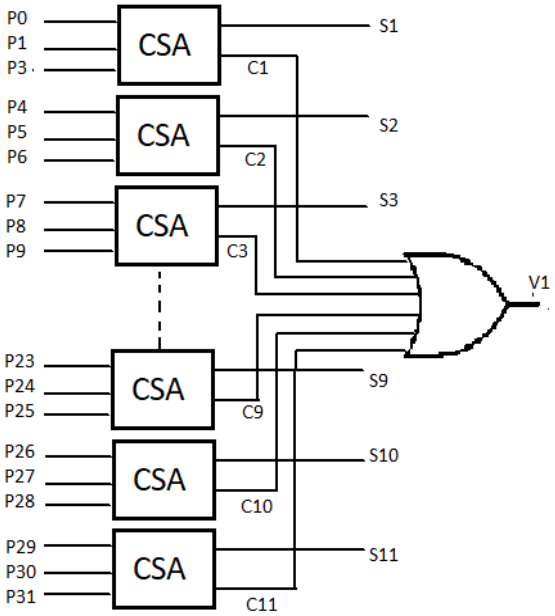

Fig 3.5.1 The first stage of compression

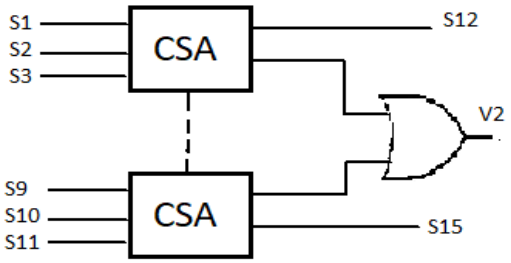

Fig 3.5.2 The second stage of compression

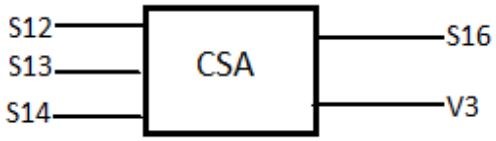

Fig 3.5.3 The third stage of compression

In Stage 2, eleven rows (S1–S11) obtained from Stage 1 are further compressed into four rows (S12, S13, S14, S15) and one accuracy compensation vector (V2**).** This reduction is achieved using an additional layer of approximate addition techniques, such as Carry-Save Adders (CSA) as shown in Fig 3.5.2.

In Stage 3, the four sum rows (S12, S13, S14, S15) obtained from Stage 2 are further compressed into a single sum row (S16) and one accuracy compensation vector (V3)**.** This final reduction is performed using approximate addition techniques, such as Carry-Save Adders (CSA) as shown in Fig 3.5.3.

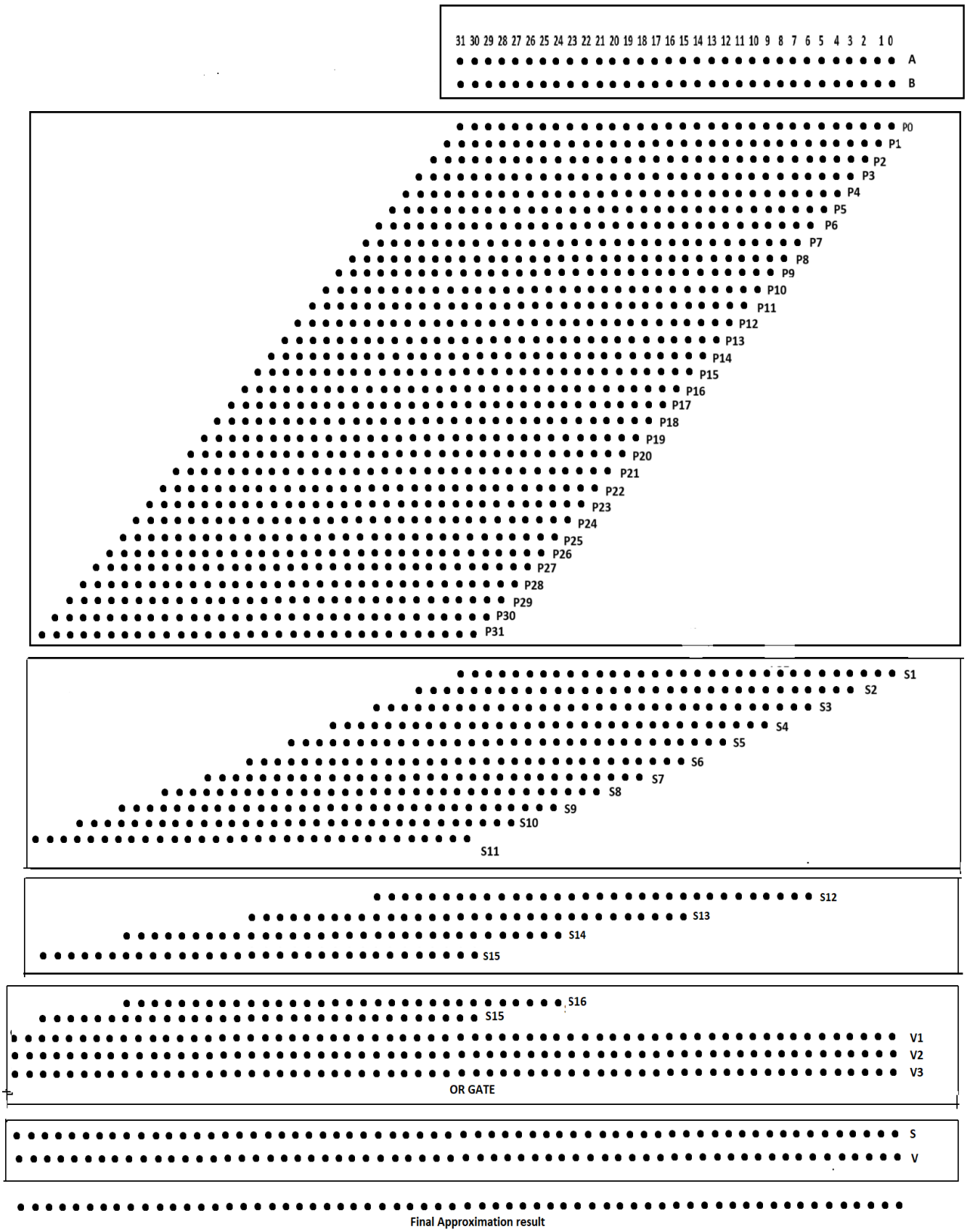

Fig 3.5 Architecture of 32-bit Approximate Multiplier

In Stage 4**,** the sum rows S15 and S16 are combined through addition to generate the final sum. At the same time, the accuracy compensation vectors V1, V2, and V3 from earlier stages are processed using an OR gate**,** producing a single accuracy compensation vector V. This approach helps consolidate approximation effects while Ensuring an efficient and balanced computation.

In the final stage, the Carry Maskable Adder (CMA) is used to efficiently produce the final sum. This adder processes the sum rows from the previous stage while selectively handling carry propagation to optimize performance. By controlling carry operations, the CMA helps balance

computational accuracy and hardware efficiency. As shown in Fig 3.5

4. EXPERIMENTAL RESULTS

4.1 Experimental Setup:

Comparing the performance of proposed multiplier with existing approaches, such as the approximate multipliers using Incomplete Adders (ACA**),** and evaluating these multipliers based on metrics like power consumption and power delay product as shown in Table 4.1.1 ,4.1.2 ,4.1.3

The comparison involves multipliers with varying bit-widths, including Eight-bit, Sixteen-bit, and Thirty-two-bit configurations. For example, your proposed multiplier features dynamically controllable accuracy, unlike other approximate multipliers. By analyzing these comparisons, you can demonstrate the effectiveness of your proposed multiplier in terms of energy efficiency and speed.

All approximate multipliers, including the approximate multiplier by using incomplete adder multiplier and proposed were implemented for 8'bit, 16'bit, and 32'bit operations using Verilog HDL. The designs were simulated using Synopsys VCS, and value change dump (VCD) files were generated to accurately analyze power consumption.

Table 4.1.1 Comparison of 8-bit Multiplier

Multiplier	Power (mW)	Energy (nJ)
Approximate multiplier by incomplete adder	13	0.1192
Approximate multiplier by carry save adder (proposed)	5	0.0467

Table 4.1.2 Comparison of 16-bit Multiplier

Multiplier	Power (mW)	Energy (nJ)
Approximate multiplier by incomplete adder	19	0.2142
Approximate multiplier by carry save adder (proposed)	7	0.1008

Table 4.1.3 Comparison of 32-bit Multiplier

Multiplier	Power (mW)	Energy (nJ)
Approximate multiplier by incomplete adder	22	0.2579
Approximate multiplier by carry save adder (proposed)	10	0.1348

4.2 Accuracy Results:

Error Distance (ED) and Mean Error Distance (MED) serve as key metrics for evaluating the accuracy For approximate arithmetic circuits, ED in multipliers is defined as the absolute difference between the exact and approximate products, while MED is the average ED computed over multiple outputs by using Eq2 ,Eq3.

To further refine accuracy evaluation, Mean Relative Error Distance (MRED) and Normalized Mean Error Distance (NMED) were also taken into account. The Relative Error Distance (RED) is calculated by dividing the Error Distance (ED) by the exact product. The MRED is then computed as the average of the RED values, similar to the MED. Meanwhile, NMED is derived by normalizing the MED concerning the maximum possible output of the exact multiplier. Another crucial metric is the error percentage (ER), which determines the proportion of incorrect results across all possible input combinations. The proposed multiplier is analyzed using NMED, MRED, and ER, as these metrics provide valuable insights into its accuracy and overall performance.

Table 4.2 provides a comparative analysis of accuracy, showing that the precision of the proposed multiplier varies based on its configuration. While its most accurate version has higher NMED and MRED values compared to some other designs, it offers improved controllability. This is particularly advantageous when compared to approximate multipliers that rely on incomplete adders. One of the key strengths of the proposed multiplier is its dynamic controllability**,** which distinguishes it from traditional approximate multipliers. Unlike existing designs, it allows for flexibility in adjusting accuracy to meet different application requirements.

$$MSE=\frac{1}{(M\times N)}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[I(i,j)-K(i,j)]^2$$

_____Eq2. M, N = Dimensions (height and width,I(i, j) represents the pixel value at coordinates (i, j) in the original picture., K(i, j) denotes the pixel value at coordinates (i, j) in the compressed or processed image. MAXI=Maximum possible pixel value , MSE=Mean Squared Error

$$PSNR=10\log_{10}(MAXI^2/MSE)$$

$$PSNR=20\log_{10}(MAXI/\sqrt{MSE})$$

_____Eq3.

Table 4.2 Accuracy Comparison 8-BIT Approximate Multiplier

	Existing			Proposed		
Image No	NMED (%)	MRED (%)	ER (%)	NMED (%)	MRED (%)	ER (%)
1.	2.95	8.69	91.41	2.85	8.60	90.23
2.	1.86	6.08	84.77	1.25	4.36	76.17
3.	3.18	7.30	84.38	2.15	5.36	76.17
4.	2.00	2.93	86.72	1.48	2.30	80.47
5.	7.63	17.15	98.44	7.82	17.19	96.48

6.	3.21	5.89	89.44	2.70	5.06	89.06
7.	2.16	4.95	89.06	1.41	3.41	80.47

16-BIT Approximate Multiplier

	Existing			Proposed		
Image No	NMED (%)	MRED (%)	ER (%)	NMED (%)	MRED (%)	ER (%)
1.	2.98	8.95	94.92	2.85	8.56	91.02
2.	2.08	7.33	89.84	1.25	4.23	78.91
3.	3.05	5.53	92.97	2.71	5.07	88.28
4.	3.12	7.47	85.94	2.12	5.36	73.05
5.	1.96	2.93	87.89	1.45	2.24	73.83
6.	7.81	17.0	97.66	7.81	17.13	97.27
7.	2.11	5.14	88.67	1.37	3.31	76.56

32-BIT Approximate Multiplier

4.3 Power and power delay product Results:

The power consumption of various multipliers is compared, with the x-axis representing the names of the multipliers and the y-axis representing power in mW as shown in Fig:4.3.1

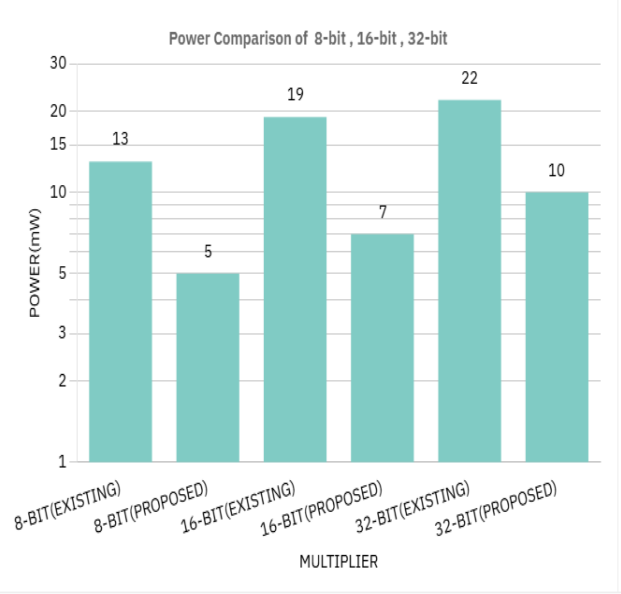

Fig 4.3.1 Power consumption results relative to the multiplier

The power delay product means the Energy of various multipliers are compared, with the x-axis representing the multipliers' names and the y-axis being Energy in nJ as shown in figure 4.3.2.

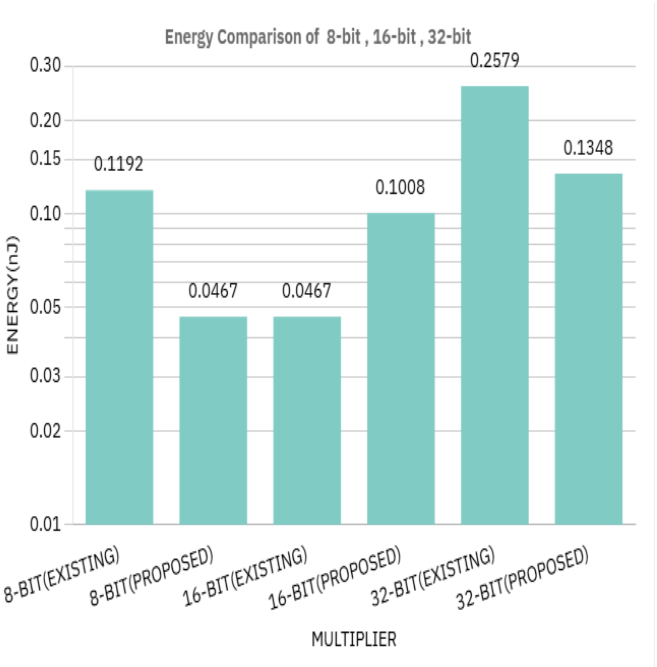

Fig 4.3.2 Power delay product relative to the multiplier

The total power consumption considered in this evaluation includes dynamic components. The proposed multiplier demonstrates efficiency Regarding power consumption and power-delay product, the accuracy-controllable multiplier achieves the lowest power usage compared to the existing multiplier.

	Existing			Proposed		
Image No	NMED (%)	MRED (%)	ER (%)	NMED (%)	MRED (%)	ER (%)
1.	3.01	8.99	90.63	2.89	8.69	92.58
2.	2.07	6.94	90.63	1.24	4.20	76.95
3.	2.91	6.56	85.94	2.20	5.53	74.22
4.	2.07	3.10	85.55	1.49	2.30	81.64
5.	7.86	17.39	97.27	7.80	17.22	96.88
6.	3.17	5.98	90.63	2.72	5.07	89.84
7.	2.02	4.74	85.94	1.31	3.29	78.91

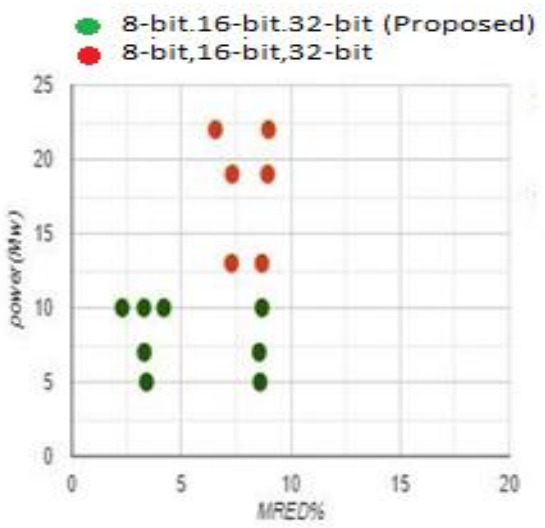

Fig 4.3.3 Power Consumption results relative to the MRED

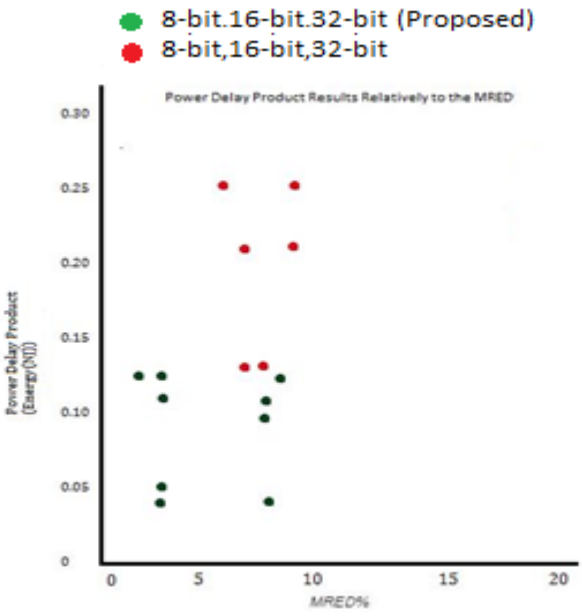

Fig 4.3.4 Power Delay Product Results Relatively to the MRED

The total power consumption considered in this evaluation includes both dynamic and static components. The proposed multiplier demonstrates efficiency. Regarding power consumption and power delay products,the accuracy-controllable multiplier achieves the lowest power usage across all accuracy levels (MRED).

Image No.	Description
1.	Home
2.	Bird
3.	Tajmahal
4.	Ship
5.	Traffic
6.	Butterfly
7.	Girl

Table-4.3.5 INUPUT IMAGES

An image-processing experiment was conducted using a widely used image compression algorithm for assessing approximate multipliers. Five grayscale images, each with a resolution of 512×512, were selected from online sources. The details of these images are presented in Table 5. Only the multiplication operations were performed approximately, while addition, Subtraction, and division retained accuracy. The quality of the processed image was evaluated using the Peak signal-to-noise ratio (PSNR). It is mathematically expressed as the mean squared error (MSE).

	APPROXIMATE MULTIPLIER					
	Incomplete Adder (Existing)			Carry Save Adder (Proposed)		
Image No.	8'BIT	16'BIT	32'BIT	8'BIT	16'BIT	32'BIT
1.	39.0700	39.0072	39.1692	52.8860	52.8941	52.8834
2.	39.1243	39.1182	39.0515	52.9298	52.9234	52.9581
3.	39.0757	39.0789	39.1215	52.8799	52.8969	52.9077
4.	39.1231	39.0921	39.0982	52.9130	52.9708	52.9170
5.	39.0577	39.0488	39.1314	52.9198	52.9162	52.8706
6.	39.0084	39.0565	39.1119	52.9053	52.8694	52.8939
7.	39.0682	39.0990	39.0447	52.8955	52.9206	52.9021

Table 4.3.6 PSNR Results of the Approximate Multiplier, In dB

The PSNR values of the approximate multipliers, expressed in dB, indicate picture quality, with higher values signifying better results. As observed, the PSNR values vary across various images in each table column**.** This variation highlights the importance of adaptive flexibility in scenarios where different quality levels are required. Moreover, the suggested accuracy-configurable multiplier exhibited a wide range of PSNR values, with its peak values matching those of other approximate multipliers.

5. CONCLUSION

This paper introduces a precision-adjustable approximate multiplier that offers reduced power consumption compared to conventional designs. The dynamic controllability of the multiplier is achieved through the proposed Carry Maskable Adder (CMA). The assessment was performed at both the circuit and application levels. Experimental findings validate that the proposed multiplier achieves substantial power savings while utilizing considerably less circuit area compared to the conventional Wallace tree multiplier. Additionally, it demonstrates substantial improvements in both power efficiency and energy

consumption when compared to previously studied approximate multipliers. Finally, the capability of the proposed multiplier to adjust accuracy dynamically was validated through application-level testing and the proposed architecture achieves a 56% reduction in power consumption by using a carry save adder and carry maskable adder.

Future work includes enhancing the efficiency of the Carry Maskable Adder (CMA) and Carry-Save Adder (CSA) to further optimize power consumption. Extending the design to higher bit-width architectures, such as 64-bit and 128-bit, will allow for a comprehensive evaluation of scalability. Exploring advanced approximation techniques can improve the balance between accuracy and energy efficiency

7. ACKNOWLEDGMENT

We want to express our gratitude to Mr. Y. Sri Chakrapani, Associate Professor in the Department of Electronics and Communication Engineering at Seshadri Rao Gudlavalleru Engineering College, India, for his valuable Guidance and support. His expertise in VLSI and embedded systems has been instrumental in shaping the direction of this research.

REFERENCES

[1] Yang, T., Ukezono, T., & Sato, T. (2018). "A low-power high-speed accuracy-controllable approximate multiplier design," Proceedings of the 23rd Asia and South Pacific Design Automation Conference (ASP-DAC), 244-249.

[2] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, "Bio-Inspired imprecise computational blocks for efficient VLSI implementation of Soft-Computing applications," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 57, no. 4, pp. 850-862, Apr. 2010.

[3] B. Moons, M. Verhelst, "DVAS: Dynamic Voltage Accuracy Scaling for increased energy-efficiency in approximate computing," IEEE/ACM International

Symposium on Low Power Electronics and Design (ISLPED), Jul. 2015.

[4] C. Liu, J. Han, and F. Lombardi, “A Low-Power, High-Performance approximate multiplier with configurable partial error recovery,” Design, Automation & Test in Europe Conference & Exhibition (DATE), Mar. 2014.

[5] S. Hashemi, R. I. Bahar, and S. Reda, “DRUM: A Dynamic Range Unbiased Multiplier for approximate applications,” IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 418 425, Nov. 2015.

[6] Bodapati, J., Sudhakar, O., & Raju, A. G. V. K., "An Improved Design of Low-Power High-Speed Accuracy Scalable Approximate Multiplier," Journal of VLSI Circuits and Systems, vol. 11, no. 3, pp. 45-56, 2022.

[7] Gu, F. Y., Lin, C., & Lin, J. W., "A Low-Power and High-Accuracy Approximate Multiplier with Reconfigurable Truncation," IEEE Access, vol. 10, pp. 12345-12356, 2022.

[8] Zacharelos, E., Nunziata, I., & Saggese, G., "Approximate Recursive Multipliers Using Low Power Building Blocks," IEEE Transactions on Emerging Topics in Computing, vol. 11, no. 2, pp. 321-334, 2022.

[9] Guo, Y., Sun, H., & Kimura, S., "Small-Area and Low-Power FPGA-Based Multipliers Using Approximate Elementary Modules," in Proceedings of the 25th Asia and South Pacific Design Automation Conference (ASP-DAC), 2020, pp. 345-350

[10] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, “Design and analysis of approximate compressors for multiplication,” IEEE Transactions on Computers, vol. 64, no. 4, pp. 984-994, Apr. 2015.

[11] K. C. Bickerstaff, E. E. Swartzlander, and M. J. Schulte, “Analysis of column compression multipliers,” 15th IEEE Symposium on Computer Arithmetic, pp. 33-39, Jun. 2001.