Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 6 (2023)

Power-Optimized Approximate Multiplier with Tunable Accuracy

Mohammed Naseer Baig[1], Vanga Ganesh Sai Kartheek[2], Peddibhotla Satya Kowmudi[3], Rajulapati Pradeep[4], Komma Anitha[5]

[1, 2, 3, 4, 5] *P V P Siddhartha Institute of Technology, Vijayawada, Andhra Pradesh, India*

Abstract:- Multiplication plays a crucial role in signal processing algorithms. However, designing efficient multipliers poses inherent challenges, including substantial area requirements, extended latency, and significant power consumption. Attention now turns to designing compressors with the goal of reducing power consumption, latency, and the number of steps involved in the product computation process in order to address these problems. In particular, this paper presents a novel strategy by suggesting a roughly optimized compressor design that is deliberately used to reduce the multiplier's stage count. Specifically, the paper introduces an innovative approach through the proposition of an approximate compressor design, strategically employed to decrease the number of stages in a multiplier. The presented multiplier architecture utilizes dynamic input truncation, in which lower significance bits of the input operands are removed in each multiplication operation depending on the acceptable level of accuracy loss for the target application. The emphasis on approximation allows for a reduction in computational complexity, providing a potential solution to the challenges posed by traditional multiplier designs. This proposed work is designed and analyzed using the Xilinx Vivado tool.

Keywords*: Accurate Computing, 4:2 Compressor, Approximate Multiplier, Tunable Accuracy.*

1. Introduction

Arithmetic functional units, particularly multipliers, play important roles in a wide range of applications, including digital signal processing, computer vision, multimedia processing, and artificial intelligence. However, the extensive use of multiplications in these applications leads to significant power consumption, especially challenging in mobile devices. To address this issue, numerous studies have explored methods to reduce power consumption in multiplier circuits [1], [2].

One approach involves approximating multiplication, especially in applications where a certain degree of error tolerance is acceptable, such as those related to human senses. Due to limitations in human sensory perception, accurate results may not always be necessary. Approximate multipliers, which trade off accuracy for reduced cell area, timing delay, and power consumption, have emerged as a solution.

Approximate multipliers can be broadly categorized into two types. The first type involves controlling the timing path through dynamic voltage scaling. By applying a lower voltage, the critical path's delay increases, introducing errors when violations occur. The second type focuses on modifying the functional behaviors of multipliers, often through the redesign of accurate multiplier circuits like Wallace Tree and Dadda Tree multipliers.

However, many previous approximate multipliers provide fixed output accuracy and power requirements [3], [4]. In certain applications, especially in dynamic environments like digital image processing, the ability to adjust accuracy and power consumption dynamically is crucial, despite the additional hardware cost.

- Introducing a high-accuracy approximate 4-2 compressor.
- Designing a simple error compensation circuit to further minimize error distances.

- Proposing a dynamic input truncation technique for adjusting accuracy and power consumption, particularly beneficial for applications like digital image processing with varying power requirements across different processing stages.
- Presenting a high-accuracy and reconfigurable approximate multiplier based on the proposed 4-2 compressor, error compensation circuit, and dynamic input truncation technique.

2. Background and Analysis

A. Computing with Traditional Multipliers

Traditional multipliers such as the Wallace Tree Multiplier and Dadda Tree Multiplier have long been relied upon for accurate computing as shown in Fig 1(a). These structures arrange full adders to enable simultaneous processing of partial products, resulting not just in precision but also speedy multiplication. Meticulous handling of each bit ensures minimal information loss and overall accurate final results. The strengths of these traditional multipliers lie in their attention to detail in propagating partial products, knack for reducing critical path delay thanks to parallel processing, and their well-defined architectures, they carefully consider each bit interaction.

Thus where precision is non-negotiable, as in many scientific computing applications, these reliable yet hardware-intensive structures become indispensable.

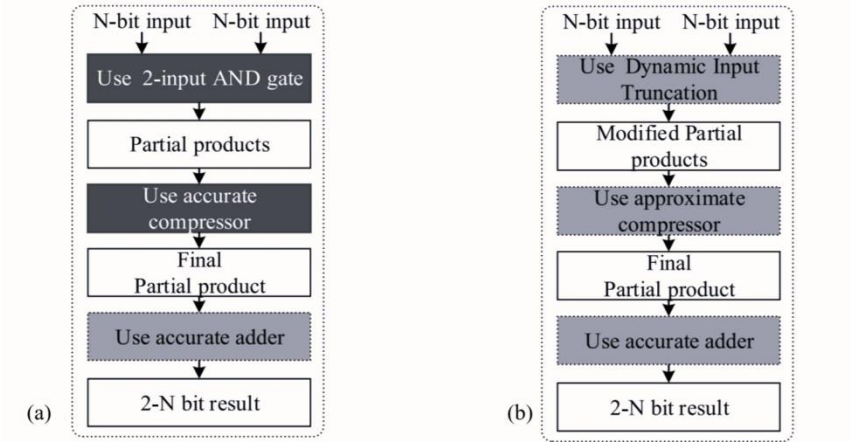

Fig 1. (a) Traditional multiplication (b) Approximate multiplication

However, the precision comes at a cost in power consump- tion. In the sections that follow, we explore the challenges tied to the energy demands of these traditional accurate multipliers. We also unveil our innovative techniques tailored to balance accuracy and efficiency for applications where some approx- imation is acceptable. By selectively reducing overhead in handling non-critical bits, our methods aim to find a sweet spot between precision, hardware complexity and energy efficiency. The approximation that is proposed can be adjustable according to the specific needs of the applications. This accuracy is achieved only for error tolerant applications.

B. Approximate Computing

In computing, multiplication has always targeted the highest level of accuracy, constructing multipliers to ensure perfect computations at any cost. For these designs to remain accurate, extremely high hardware requirements, time, and power are needed. However, in many real-world situations, adequate precision is sufficient [5].

This insight has led to a new concept called approxi- mate computing for multipliers. Deliberately allowing for some controlled errors during multiplication, opening the door to striking a balance between precision and resource efficiency. It gives designers a dial to configure variable precision levels in line with specific app needs. So for a video processing algorithm, an approximate multiplier can adapt to provide just the right fidelity for the human eye at minimal hardware cost. By relaxing the complete accuracy requirement, approximate

designs in Fig 1(b) can significantly simplify multiplier architectures. This leads to huge benefits in hardware footprint, clock speeds, power savings all while keeping errors within acceptable limits. Designers must carefully weigh these computational accuracy trade offs based on tolerance levels. And simulation tools are crucial for assessing different techniques

to guide balanced decisions.

In essence, approximate computing marks a new era for multipliers in giving applications precisely the precision they require nothing more, nothing less. It's about time-efficient computing, with the understanding that not all ones and zeros are created equal. There is flexibility in the bits where approximation designs can be creatively used.

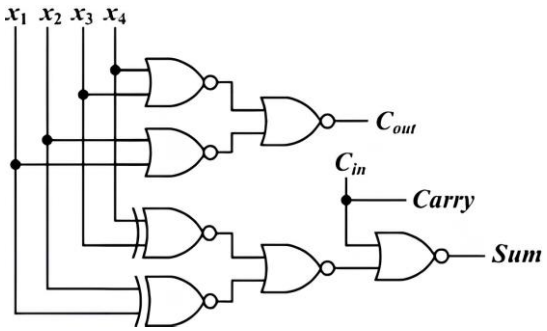

Fig 2. Accurate Compressor Gate level Design

3. Proposed High Accuracy 4-2 Compressors

A. Accurate Compressor

The carry output in equation (1), *Carry* is approximated to equal the input carry *Cin* in the Fig 2 using an approximation method. In twenty-four of the thirty-two states, this approx- imation is correct, meaning that the carry output is almost always in line with the input carry as per the TABLE I .

$$Carry = Cin \tag{1}$$

To be more precise, simplifying the total to zero stream- lines the overall design complexity while also minimizing the difference between the approximate and accurate outputs. Furthermore, the deliberate injection of faults into the sum signal helps to generate the approximate total more quickly, which lowers the overall design latency.

$$Sum = \overline{C_{in}}(\overline{x_1 \oplus x_2} + \overline{x_3 \oplus x_4}) \tag{2}$$

Moreover, differences in the value of *Cout* (Carry-out) in particular states may reduce the error distance caused by the approximate carry and sum [6]. These carry-out modifications in equations (2), (3) show the complex trade-offs between computational accuracy and design complexity in addition to making the design simpler to implement.

$$Cout = \overline{(\overline{x_1 \cdot x_2} + \overline{x_3 \cdot x_4})} \tag{3}$$

In the field of digital circuit design, intentional utilization of approximations such as setting Carry equal to *Cin* offers a customized method for striking a compromise between accuracy and power.

B. 4-2 Approximate Compressor

A 4-2 approximation compressor is a digital circuit that reduces four input data points to two. In order to balance producing a work that is sufficiently accurate with conserving computer resources, such as time and electricity, minor errors are purposefully allowed in this procedure. When handling information efficiently by the computer is more vital than being perfectly precise, this type of compressor comes in handy [7], [8]. The

Proposed Compressor's goal was to raise the approximation levels above the compressor's actual design in order to dramatically lower the error rate and enhance performance. This compressor consists of 4-inputs and 2-outputs.

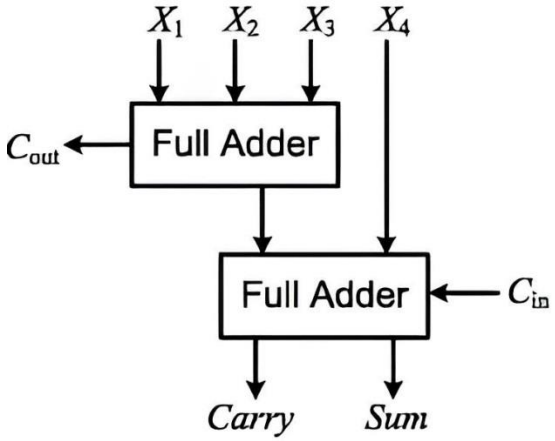

Fig 3. Accurate Compressor

Table 1. Truth Table of Accurate Compressor

cin	x_4	x_3	x_2	x_1	sum	carry	*cout*
0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0
0	0	0	1	0	1	0	0
0	0	0	1	1	0	0	1
0	0	1	0	0	1	0	0
0	0	1	0	1	0	0	1
0	0	1	1	0	0	0	1
0	0	1	1	1	1	0	1
0	1	0	0	0	1	0	0
0	1	0	0	1	0	1	0
0	1	0	1	0	0	1	0
0	1	0	1	1	1	0	1
0	1	1	0	0	0	1	0
0	1	1	0	1	1	0	1
0	1	1	1	0	1	0	1
0	1	1	1	1	0	1	1
1	0	0	0	0	1	0	0
1	0	0	0	1	0	1	0
1	0	0	1	0	0	1	0
1	0	0	1	1	1	0	1
1	0	1	0	0	0	1	0
1	0	1	0	1	1	0	1

cin	x_4	x_3	x_2	x_1	sum	carry	*cout*
1	0	1	1	0	1	0	1
1	0	1	1	1	0	1	1
1	1	0	0	0	0	1	0
1	1	0	0	1	1	1	0
1	1	0	1	0	1	1	0
1	1	0	1	1	0	1	1
1	1	1	0	0	1	1	0
1	1	1	0	1	0	1	1
1	1	1	1	0	0	1	1
1	1	1	1	1	1	1	1

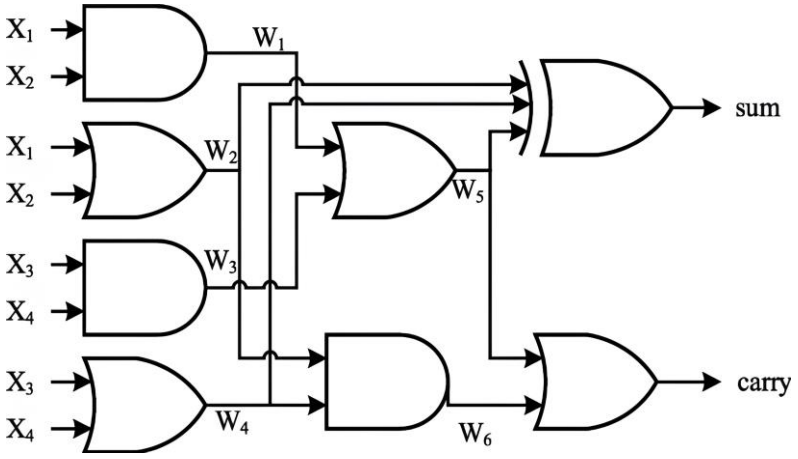

Fig 4. Approximate Compressor Gate level Design

$$Sum = (\overline{x_1 \oplus x_2} + \overline{x_3 \oplus x_4}) \quad (4)$$

$$Carry = \overline{(x_1 \cdot x_2 + x_3 \cdot x_4)} \quad (5)$$

Due to their equal weight in the computation, a key approach in this design is to estimate the input carry *Cin* as well as the *Carry*. Setting *Cin* to zero eliminates the need to explicitly account for *Carry*, making things simpler (4), (5). This intentional simplification highlights a conscious effort to achieve faster performance by lowering computing complexity and raising abstraction, in addition to producing a more effective design.

$W_1 = x_1 \cdot x_2$ (6)

$W_2 = x_1 + x_2$ (7)

$W_3 = x_3 \cdot x_4$ (8)

$W_4 = x_3 + x_4$ (9)

$W_5 = W_1 + W_3$ (10)

$W_6 = W_2 \cdot W_4$ (11)

$Carry = W_5 + W_6$ (12)

From TABLE II it is clear that an error occur only when

Table II. Truth Table of 4-2 Compressor

x_4	x_3	x_2	x_1	sum	carry	difference
0	0	0	0	1	0	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	1	0
0	1	1	1	1	1	0
1	0	0	0	1	0	0
1	0	0	1	0	1	0
1	0	1	0	0	1	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

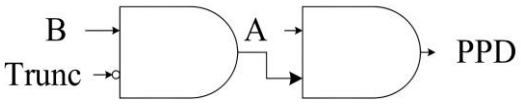

Fig 5. A modified version of the product.

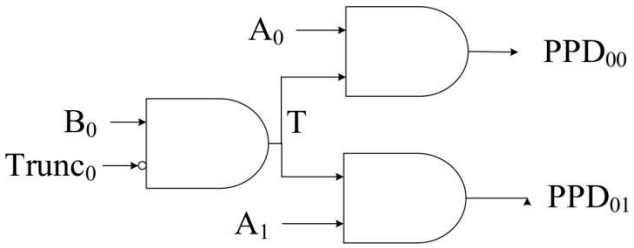

Fig 6. Gate sharing that minimizes the number of gates

all the inputs of compressor are high. This error is negligible according to the application requirements.

$$Error = W1 \text{ AND } W3 \quad (13)$$

Since W1 uses an AND gate to determine whether both X1 and X2 are 1, and W3 uses an AND gate to determine whether both X3 and X4 are 1, we simply need an additional AND gate for error detection in order to determine whether both W1 and W3 are 1 [9]. The error detection circuit (EDC) equation is displayed in equation (13). Thus, by including an additional AND gate, the error compensation circuit of the suggested 4-2 compressor can be built with ease.

4. Proposed Approximate Multiplier

A. Dynamic Input Truncation

To realize an adjustable approximate multiplier at runtime, this study puts forward a dynamic input truncation method- ology using two 2-input AND gates, as depicted in Fig 5, to generate a partial product. Equation (14) formulates the partial product, where A signifies the multiplicand and B denotes the multiplier. The Trunc signal controls whether the partial product PPD undergoes truncation. A Trunc value of 1 truncates PPD to 0. More precisely, the Trunc signals conserve power by truncating the PPDs to zeroes for select multiplications. In effect, the Trunc signals have the impact of disabling hardware units in the respective columns.

$$PPD_{ij} = (\sim Trunc \text{ AND } B_i) \text{ AND } A_j \quad (14)$$

In an 8x8 multiplier, each multiplier bit matches to 8 mul- tiplicand bits. To optimize hardware utilization, this imple- mentation shares gates using an additional AND gate. For instance, PPD_{00} equals $\sim Trunc_0 \cdot B_0 \cdot A_0$while PPD_{01} equals $\sim Trunc_0 \ B0.A_1$. Here, $\sim Trunc_0 \ B_0$ gets computed in advance to create a mask, needing only three 2-input AND gates, as shown in Fig 6. The next subsection will discuss the control logic for the *Trunc* signals in the presented approximate multiplier.

B. Multiplication

Fig 7 shows our proposed approximate multiplier. Our proposed multiplier is designed for 8 bit but it can be extended to N-bit with respect to the precise application. Previously proposed multipliers are modified to decrease the stages in partial product reduction [10]-[13].The three stages are as follows:

1) *Generating partial products:* At the outset, the algorithm makes use of the effectiveness of two-input AND gates to multiply individual bit pairs. A key component of precision control is introduced by the four-bit *Trunc* signal, which enables the selective truncation of partial products accord- ing to the required accuracy level. Using a 3-4-4-4 division scheme optimizes the trade-off between accuracy preservation and resource utilization by offering an organized method of controlling truncation. Alternative partition alternatives, like the 3-3-3-3-3 layout, give more granular control at the cost of additional hardware overhead, even though this scheme offers a well-balanced compromise.

2) *Partial Product Sorting and Compression:* Sorting and compressing partial products becomes the main focus of the second stage. Based on the importance of each area, a strategic choice was made to divide the partial products into distinct regions: accurate (columns 14th–8th) and approximate (columns 7th–0th). To guarantee accuracy in the outcomes, high-importance, exact partial products are subjected to precise 4-2 compressors. In order to effectively compress approxima- tion products with regulated error margins, custom-designed approximate 4-2 compressors and error compensation circuits are implemented simultaneously. In order to prepare the data for the final summation, this step is crucial.

3) *Final Product:* The final stage focuses on generating the end results and efficiently processing errors. Replacing addition operations with OR gates in columns 3 to 0 simplifies computation of less significant results, improving efficiency. To further enhance error management, mechanisms integrate into stage two to detect and compensate for substantial in- accuracies arising in the approximate region. Precise adders then strategically summarize the most significant columns to ensure thorough and accurate final outputs. In summary, the proposed three-stage methodology carefully optimizes the computational flow [14]. Stage one streamlines the initial mul- tiplication process through approximation. Stage two handles error detection and correction in less critical areas. Finally, stage three accurately computes the critical columns. This structured approach roles out specialized techniques in each stage to optimize resource use, preserve accuracy, and effec- tively constrain errors. The deliberate functionality partitioning across stages plays a key role in boosting the overall efficiency and reliability of the computational procedure.

5. Simulation And Results

In order to evaluate the approximation multipliers, this section first explains the experimental setup. Next, it examines the cell area, power consumption, critical path delay, and power-delay product (PDP) by comparing the different ap- proximate multipliers.

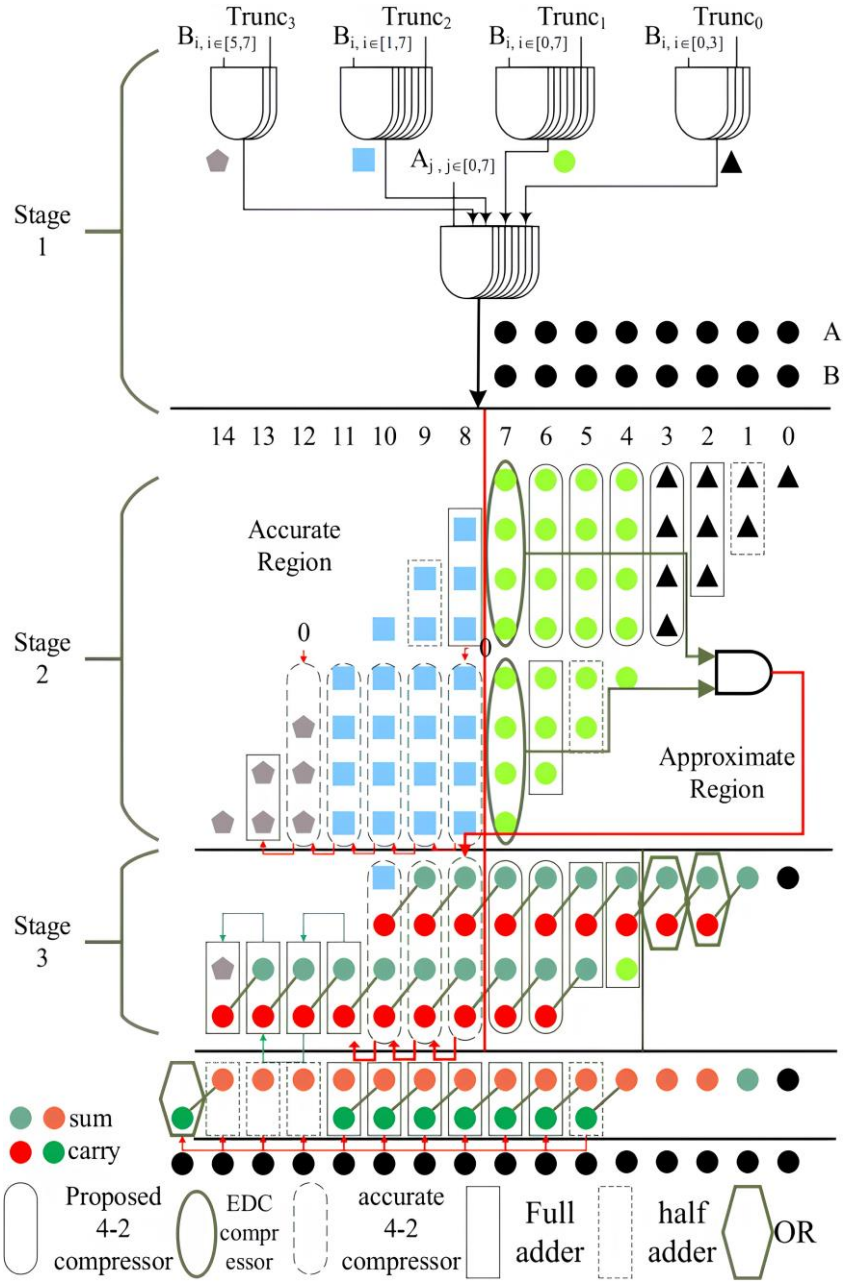

Fig 7. Approximate Multiplier

At first we choose the number of input bits for multiplication. In this paper we have implemented 8-bit multiplication of two binary numbers using Xilinx Vivado. The modified partial products are generated by dynamic input trun- cation technique and then these partial products are reduced by using our proposed approximate compressor, full adder and half adder. Finally, a carry save adder is used to generate the final product. The RTL code generates the schematic diagram of 8-bit approximate multiplier. We analyzed the performance of our proposed approximate multiplier.

Table III. Power, Area and PDP Evaluations

	Power	Area	PDP
Exact	100%	100%	100%
Yandg_adj	84%	90%	68%
Strollo_acc	92%	90%	64%
Proposed	82%	93%	60%

Previously many approximate compressors are proposed which are referred as “Yang1”, “Yang2”, “Yang3”, “Momeni”, “Akbar1”, “Ha”. All these compressors show approximately same result. There isn’t a single, optimal approximate com- pressor topology since the level of precision needed determines the optimal design. We compared the performance of our proposed approximate multiplier with previously proposed approximate multipliers. The compared results of accuracy, area and power are shown in TABLE III. Our proposed ap- proximate multiplier has shown good trade-off between power and accuracy when compared with other proposed multipliers.

6. Conclusion

This paper presents a new kind of multiplier architecture that tackles the problem of high power and long processing time in traditional designs. This approach rebuilds compressor blocks to reduce complexity and increase processing effi- ciency. The proposed strategy involves the implementation of a selectively approximate compressor, which by truncating non- critical bits during multiplication selectively depending on an acceptable accuracy loss for a given application.

This power efficient adaptable accuracy approach imple- mented by using Xilinx Vivado demonstrates promising im- provements as it balances processing demands against power limitations. The results reveal possible advantages when en- ergy efficiency is a major concern.

This paper emphasizes the importance of reconsidering approximations in multiplication with judicious use of it to optimize for tradeoffs between Power and Accuracy. This mul- tiplier design therefore offers solutions for signal processing and other computation-heavy workloads by balancing compu- tational performance versus power consumption. Furthermore, it suggests more flexible methodologies beyond the limitations of classical multipliers across various application domains.

References

[1] B. Moons and M. Verhelst, “DVAS: Dynamic voltage accuracy scal- ing for increased energy-efficiency in approximate computing,” in Proc.IEEE/ACM Int. Symp. Low Power Electron. Design (ISLPED), Jul. 2015, pp. 237–242.

[2] K. Yin Kyaw, W. Ling Goh, and K. Seng Yeo, “Low-power high- speed multiplier for error-tolerant application,” in Proc. IEEE Int. Conf. Electron Devices Solid-State Circuits (EDSSC), Dec. 2010, pp. 1–4.

[3] M. de la Guia Solaz, W. Han, and R. Conway, “A flexible low power DSP with a programmable truncated multiplier,” IEEE Trans. Circuits Syst., vol. 59, no. 11, pp. 2555–2568, Nov. 2012.

[4] R. Zendegani, M. Kamal, M. Bahadori, A. Afzali-Kusha, and M. Pedram,“RoBa multiplier: A rounding-based approximate multiplier for highspeed yet energy-efficient digital signal processing,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 2, pp. 393–401, Feb. 2017.

[5] C. S. Wallace, “A suggestion for a fast multiplier,” IEEE Trans. Electron. Comput., vol. EC-13, no. 1, pp. 14–17, Feb. 1964.

[6] A. Weinberger, “4:2 carry-save adder module,” IBM Tech. Discl. Bull., vol. 23, no. 8, pp. 3811–3814, 1981.

[7] 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.

[8] Z. Yang, J. Han, and F. Lombardi, “Approximate compressors for errorresilient multiplier design,” in Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst. (DFTS), Oct. 2015, pp. 183–186.

[9] C.-H. Lin and I.-C. Lin, “High accuracy approximate multiplier with error correction,” in Proc. IEEE 31st Int. Conf. Comput. Design (ICCD), Oct. 2013, pp. 33–38.

[10] P. J. Edavoor, S. Raveendran, and A. D. Rahulkar, “Approximate multiplier design using novel dual-stage 4:2 compressors,” IEEE Access, vol. 8, pp. 48337–48351, 2020.

[11] F. Sabetzadeh, M. H. Moaiyeri, and M. Ahmadinejad, “A majority-based imprecise multiplier for ultra-efficient approximate image multiplica- tion,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 66, no. 11, pp. 4200–4208, Nov. 2019.

[12] A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra, and G. D. Meo,“Comparison and extension of approximate 4–2 compressors for lowpower approximate multipliers,” IEEE Trans. Circuits Syst. I, Reg. Papers,vol. 67, no. 9, pp. 3021–3034, Sep. 2020.

[13] H. Xiao, H. Xu, X. Chen, Y. Wang, and Y. Han, “Fast and high-accuracy approximate MAC unit design for CNN computing,” IEEE Embedded Syst. Lett., early access, Dec. 21, 2021, doi: 10.1109/LES.2021.3137335.

[14] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, “Dual-quality 4:2 compressors for utilizing in dynamic accuracy configurable multi- pliers,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 4,pp. 1352–1361, Apr. 2017.