

XMULT: An Energy-Efficient Design of Approximate Multiplier

Ahsan Rafiq*, Maksim Jenihhin

Department of Computer Systems, Tallinn University of Technology, Tallinn, Estonia

*ahsan.rafiq@taltech.ee

Abstract—Multipliers are the basic building blocks in all computing systems. It is an indispensable strategy to design multipliers to achieve promising trade-offs for energy efficiency and error metrics. For this purpose, approximate computing effectively provides high hardware energy efficiency and accuracy in various edge computing applications. In this paper, we propose a new approximate unsigned multiplier that aims to have high energy efficiency while maintaining the low error. In the proposed design, the first four partial product rows are compressed using the proposed lightweight circuitry. Besides this, OR gates based approximate compression is further employed to gain more promising results in terms of energy efficiency. The Proposed approximate multiplier design, along with other state-of-the-art designs, are synthesized in the FreePDK 45nm Nangate technology. The proposed design shows over 34.23% improvement in power delay product as compared to the state-of-the-art design.

Index Terms—High-performance circuits and Systems, Approximate Computing, circuit-level approximation, Edge AI

I. INTRODUCTION

Approximate computing is an effective means of achieving high energy efficiency by sacrificing a certain degree of acceptable accuracy in various applications, i.e., image processing, machine learning, deep learning, etc [1]. These are error-tolerant applications that use approximate multipliers instead of exact ones because these applications do not require fully accurate outputs. In this way, approximation helps in improving energy efficiency by reducing design complexity [2]. In a multiplication process, partial products are generated, compressed, and then added to get the final output products. The use of an approximation technique helps in these stages, mostly in the first two stages, to lower the design complexity [3]. It means that when a reduced circuit is used instead of an accurate one, the hardware performance parameters of delay, power, area, and power delay products are improved.

Various techniques in the literature focus on approximations at different levels in computing hardware [1]. Logarithmic approximation [4], the use of inexact compressors [5]-[7], and approximate radix encoding [8] are the major approximation techniques in the inexact or approximate multipliers to improve energy efficiency. In [9], a novel design of approximate multipliers were proposed using approximate adders. These approximate adders were designed by doing circuit level optimizations. In the case of the EvoApprox library [10], a series of multipliers are proposed by cartesian genetic

programming. Most of the designs inside that library are horizontally truncated, and some have zero masking in the least significant bits (LSBs) from the input side. In [11], two approximate multipliers are proposed to simplify the generation and reduction of partial products. They proposed circuitry based on the idea that two partial product rows can be simplified in the generation and compression stages. In [12], a decoder based approximate multiplier is proposed for low-power applications. In these mentioned designs, the authors carefully approximated the designs by allowing acceptable tradeoffs between efficiency and degree of inaccuracy. In the case of the proposed design, it also deals with an 8x8 unsigned multiplier. The main contributions in the proposed designs are the following:

- A lightweight circuitry is proposed for partial products' compression, which takes into account the idea that two adjacent partial product rows are identical diagonally. It means that two partial product terms in two adjacent partial product rows are diagonally identical.
- Alongside the lightweight circuitry for the adjacent partial products, OR gate-based compression is used to compress the partial products for the final addition stage. The important thing is that this compression only uses OR gates with carry propagation, which means that the carry bits are discarded.

The paper is organized in the following manner. Section II elaborates the proposed design methodology along with an example. Section III discusses the results of the proposed design with those of the other ones in the literature. Finally, Section IV concludes the work.

II. PROPOSED APPROXIMATE DESIGN SCHEME

For an $N \times N$ multiplication, partial products are generated by multiplying each operand bit X by each bit of operand Y using AND logic gates. N rows and $2N-1$ columns of partial product terms are generated in this case. This paper considers 8x8 multiplication, which generates eight rows and fifteen columns of partial product terms, as shown in Fig. 1. The proposed design of an 8x8 approximate multiplier consists of three regions. They are the following: (1) truncation region, (2) approximate region, and (3) exact region. The truncated and approximate regions are represented column-wise by the terms t and m , respectively. Let $N = 8$ denote the middle column, t consists of columns $N - 7$ to $N - 5$, while m is the column range from N to $N + 4$. In addition, the

term K and K' represent the two adjacent number of rows, i.e., the first and the second, and the third and the fourth rows. The proposed design considers the first four rows for the lightweight approximation.

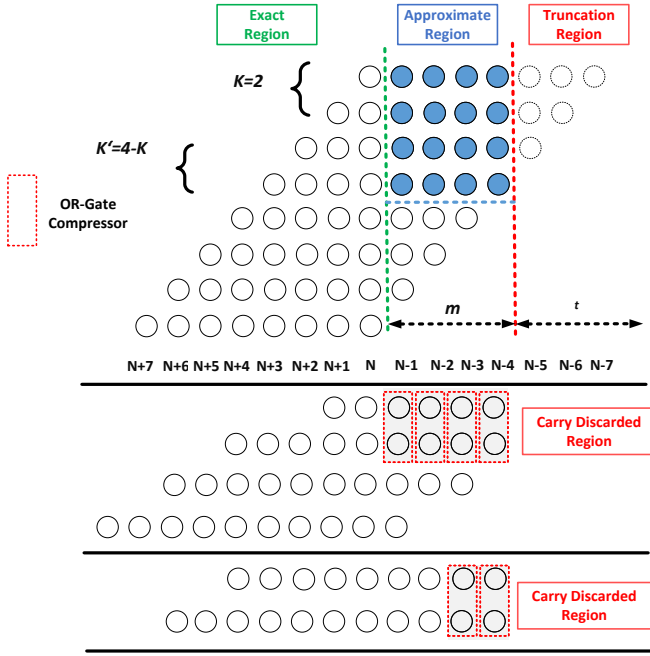


Fig. 1: Proposed Design of an 8x8 Approximate Multiplier

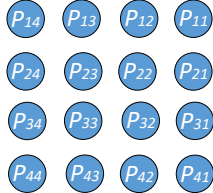


Fig. 2: A 4x4 matrix of first four partial products considered for the approximation

A. Lightweight Approximation of Adjacent Rows

The lightweight approximation of the rows is performed in the m region, which is a 4x4 partial product matrix as shown in Fig. 2. Let p_{ij} be the row and column number of the partial product matrix in the approximate region m , where i and j are $1 \leq i \leq 4$ and $1 \leq j \leq 4$, respectively, as shown in Fig. 2. This approximation is based on the idea that partial product terms are repeated diagonally. It means that terms P_{11} , P_{12} and P_{13} are identical to P_{22} , P_{23} and P_{24} . Similarly, P_{31} , P_{32} and P_{33} are identical to P_{42} , P_{43} and P_{44} . Based on this approach, two identical circuits are formed to compress this matrix of partial products. As shown in Fig. 3 and Fig. 4, both of these circuits are the same. They compress the adjacent partial products. The terms x_0 , x_1 , x_2 , and x_3 are the corresponding enabling bits of the operand X .

In the case of the exact region, it is compressed using the ripple carry adders. Partial product rows five and six, and seven and eight are compressed using ripple carry adders, respectively, as shown in Fig. 1.

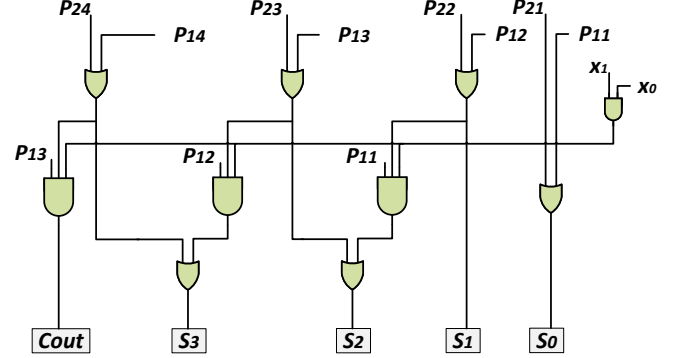


Fig. 3: Lightweight approximate circuit to compress the terms in the first and the second partial products

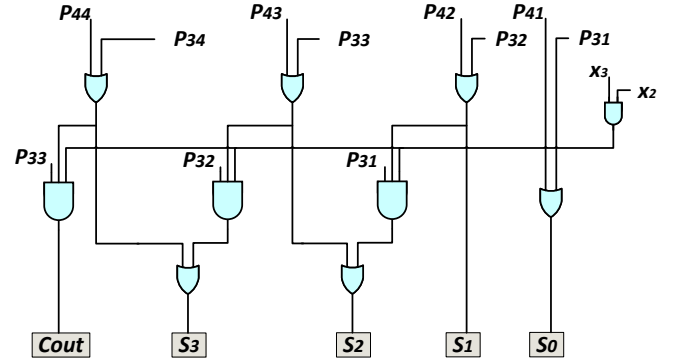


Fig. 4: Lightweight approximate circuit to compress the terms in the third and the fourth partial products

B. OR-Gate-Based Compression by Discarding Carry

The resultant bits of 4x4 approximate matrix are further compressed by merely using OR logic gates as shown in Fig. 1. In this way, a large amount of footprints and power are saved by replacing the conventional half adders and full adders with just OR logic gate based compressor. It means that the carry bits from the terms of this 4x4 matrix are discarded. Finally, the last two rows of partial product are added using ripple carry adder to get the final output products.

C. Approximate Multiplication Example

An example of proposed approximate multiplication is mentioned out of a total 65536 cases for 8x8 multiplication in Fig. 5. In this example, there are two operands. The first one X is 159, and the second one Y is 215. They are both multiplied using the proposed approximation-based multiplication. The exact result of the multiplication is 34185, while the proposed approximate one provides 34232. It means that the approximate design depicts a -47 difference from the exact values in this particular example.

$$\begin{array}{r}
\text{X}=159=(10011111)_2 \quad \text{Y}=215=(11010111)_2 \\
\begin{array}{r}
215 \\
\times 159 \\
\hline
\text{Exact} = 34185
\end{array}
\end{array}$$

$$\begin{array}{r}
\begin{array}{c}
K=2 \left\{ \begin{array}{l} 11010111 \\ 11010111 \\ 11010111 \end{array} \right. \\
K'=4-K \left\{ \begin{array}{l} 11010111 \\ 11010111 \\ 00000000 \\ 00000000 \\ 11010111 \end{array} \right. \\
c_0 \quad 101111 \\
1101001111 \\
0011010111 \\
01110101110 \\
\hline
\text{Approximate} = 34232 = (1000010110111000)_2
\end{array}
\end{array}$$

Difference = Exact-Approximate = -47

Fig. 5: Example of multiplication using Proposed Design

III. RESULTS AND DISCUSSION

A. Error Analysis

TABLE I: Error Metric Analysis of 8x8 Approximate Multipliers

Designs	MRED (10^{-2})	MAE
1JJQ [10]	0	0
OCLA [9]	-	34.71
HOAANED [9]	-	11.49
DeBAM [12]	2.37	333.37
Improv-MUL [11]	0.43	26.38
XMUL(Proposed)	0.93	38

To extract the mean absolute error (MAE) and mean relative error (MRED) [13] from the proposed and the other state-of-the-art designs, error metric analysis consisting of MAE and MRED are given in Table I. Observing the error, it can be seen from these tables that the proposed designs offer a smaller values for MAE and MRED which are 38 and 0.93×10^{-2} . This is the comparable error with the other designs. The main thing is to find out the hardware performance parameters to find the best possible designs out of these by using the Pareto graphs.

B. Hardware Performance Evaluation

TABLE II: Hardware Performance Analysis of 8x8 Approximate Unsigned Multipliers

Designs	Delay (ns)	Power (μ W)	PDP (fJ)	Area (μm^2)
1JJQ [10]	1.182	34.82	41.157	316
OCLA [9]	0.855	42.95	36.722	278
HOAANED [9]	0.91	47.58	43.29	315
DeBAM [12]	0.930	28.91	26.88	222.64
Imp-MUL [11]	0.92	30.80	28.33	265.73
XMUL(Proposed)	1.02	18.27	18.63	239.93

To evaluate the hardware performance of the proposed design with those in the state-of-the-art literature, the designs are described in Verilog HDL using 45nm PDK. The designs in [9] already used 45nm PDK. Power estimation is obtained through saif file switching activity by applying 65536 input combination at 500MHz frequency. The nominal voltages were 1.1V using the typical corner. The result of the proposed

synthesis are given in Table II. It can be seen that the proposed design improves the power delay product significantly than the ones that are now in the literature. The proposed design is shown to be provided a 34.23% improvement in the power delay product as compared to the state-of-the art design [11], which has the lower MRED and MAE (Table I) out of all the given designs.

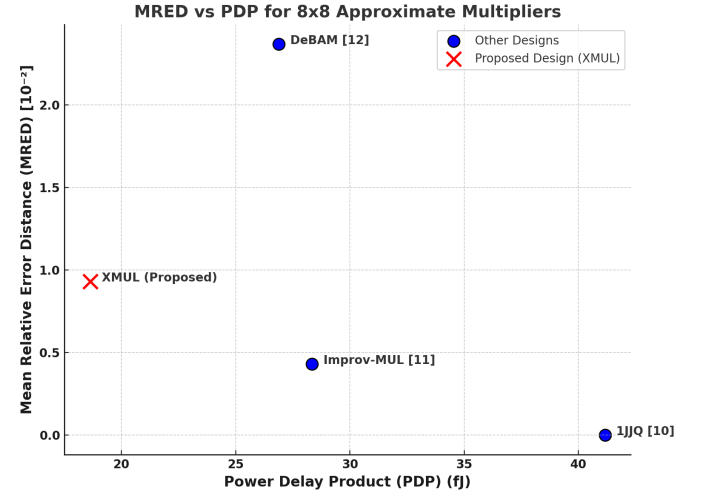


Fig. 6: Hardware performance vs Error Comparison (MRED) in 8x8 Unsigned Multipliers

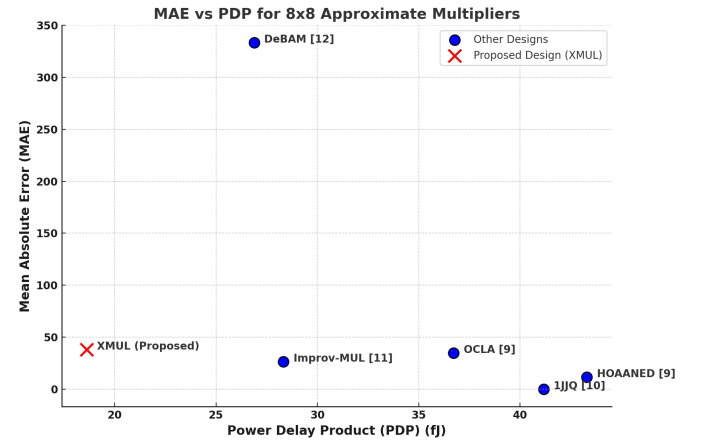


Fig. 7: Hardware performance vs Error (MAE) in 8x8 Unsigned Multipliers

IV. CONCLUSION

This paper presented the approximate design of an 8x8 multiplier. The proposed design used lightweight approximation to reduce partial product terms. Moreover, the partial products are compressed using the OR logic gates instead of conventional blocks of half and full adders. Experimental results (Table I and Table II) demonstrate that the proposed design achieves high energy efficiency while maintaining an affordable error. In addition, figures 6 and 7 illustrate the comparison of

hardware performance parameters (PDP) and errors (MAE and MRED). Both figures indicate that the proposed design achieves a superior balance between hardware performance and error metrics compared to competitive designs.

ACKNOWLEDGMENT

This work was supported in part by the Estonian Research Council grant PUT PRG1467 "CRASHLESS" and by EU Grant Project 101160182 "TAICHIP".

REFERENCES

- [1] V. Leon et al. "Approximate computing survey, Part I: terminology and software & hardware approximation techniques." *ACM Computing Surveys*, 1-36, 2025.
- [2] W. Liu et al., "Approximate Computing: From Circuits to Applications [Scanning the Issue]," in *Proceedings of the IEEE*, vol. 108, no. 12, pp. 2103-2107, Dec. 2020.
- [3] S. Mittal, "A survey of techniques for approximate computing," *ACM Computing Surveys (CSUR)*, 48(4), 1-33, 2016.
- [4] R. Makimoto, T. Imagawa and H. Ochi, "Approximate Logarithmic Multipliers Using Half Compensation with Two Line Segments," 2023 IEEE 36th International System-on-Chip Conference (SOCC), Santa Clara, CA, USA, 2023.
- [5] L. Sayadi et al., "Two Efficient Approximate Unsigned Multipliers by Developing New Configuration for Approximate 4:2 Compressors," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 4, pp. 1649-1659, April 2023.
- [6] R. Duan et al., "A hardware-efficient approximate multiplier combining inexact same-weight N: 2 compressors and remapping logic with error recovery," In 2023 IEEE 36th International System-on-Chip Conference (SOCC), pp. 1-6, 2023.
- [7] Y. Guo et al., "Low-cost approximate multiplier design using probability-driven inexact compressors," In 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), pp. 291-294, 2018.
- [8] Vasileios Leon, Georgios Zervakis, Dimitrios Soudris, and Kiamal Pekmestzi, "Approximate Hybrid High Radix Encoding for Energy-Efficient Inexact Multipliers, " *IEEE Trans. on Very Large Scale Integration (VLSI) Systems* 26, 421-430, 2018.
- [9] Y. Manneppalli et al., "Novel approximate multiplier designs for edge detection application." *GLSVLSI*, pp. 371-377. 2021.
- [10] V. Mrazek, Z. Vasicek, and L. Sekanina, "EvoApproxLib: Extended Library of Approximate Arithmetic Circuits," in *Workshop on Open-Source EDA Technology*, in *WOSET'19*. 2019
- [11] Alamuri, P., Kumar, U. A., Vannuru, V., & Ahmed, S. E., "Improved approximate multiplier architecture for image processing and neural network applications," *Microprocessors and Microsystems*, 101, 2023.
- [12] S. Nambi, U. A. Kumar, K. Radhakrishnan, M. Venkatesan and S. E. Ahmed, "DeBAM: Decoder-Based Approximate Multiplier for Low Power Applications," in *IEEE Embedded Systems Letters*, vol. 13, no. 4, pp. 174-177, Dec. 2021.
- [13] C. Liu et al., "An Analytical Framework for Evaluating the Error Characteristics of Approximate Adders," in *IEEE Trans. on Comp.*, vol. 64, no. 5, pp. 1268-1281, 1 May 2015.