Applied and Computational Mathematics
2024, Vol. 13, No. 5, pp. 178-185
https://doi.org/10.11648/j.acm.20241305.16

Research Article

Approximate Multiplication and Division Calculation for DSP Applications

Dina Mohamed Ellaithy*

Microelectronics Department, Electronics Research Institute (ERI), Cairo, Egypt

Abstract

Digital signal processing (DSP) is a promising alternative to accomplish a wide variety of signal processing operations. Multiplications and divisions have been urgently proposed to perform different DSP operations in real-time applications. However, the design and implementation of multiplication and division calculations have some limitations such as strict timing, bounded power consumption, and higher accuracy. The exact execution of these complex operations consumes great hardware resources and power consumption. Approximate computation for the main and complex arithmetic functions is a demand solution to decrease power, area, and delay. In this paper, low-power and high-accuracy approximate multiplication and division processes have been implemented using a 90 nm CMOS process, 1.0 V supply voltage standard cell library. The approximate multiplication and division algorithm depends on enhanced logarithmic converters. The logarithm-based arithmetic exhibits a good performance with decreasing the used hardware resources and runs at a higher speed. The proposed scheme achieves less hardware with minimal power consumption. The proposed approximate structure demonstrates up to a 62% saving in power with a rise in the accuracy level as compared with the prior approximate works. At the same time, the proposed multiplier and divider can carry out the multiplication and division processes in 1.8 ns, respectively.

Keywords

Approximation Algorithm, ASIC, Division Process, Energy-Efficient Processing, Logarithm Arithmetic, Low Power

1. Introduction

With the rise of digital signal processing (DSP) in various applications such as digital image and video processing, data compression, image compression, and biomedical engineering, the execution of different processes have the spot of the integrated circuit designers to meet the real-time requirements [1, 2]. In DSP a wide variety of signal processing operations are performed. Examples of the frequently used functions in the carrying out of DSP operations are: multiplication function (ML), division function (DV), square function (SQ), square root function (SQRT), root mean squares (RMS) and total harmonic distortion (THD). The exact execution of these operations consume large amount of hardware. As well, the power dissipation and delay increases significantly to can handle the extraordinarily numbers of additions and subtractions that takes place for the implementations of these complex functions. As a consequence, the overall performance of the DSP operations will be affected. The exact achievement of the commonly used functions may not be the appropriate solution for real-time DSP applications that require fast processing time with minimum consumed power for battery- powered operated devices. In the

*Corresponding author: dina_elessy@eri.sci.eg (Dina Mohamed Ellaithy)

Received: 12 August 2024; **Accepted:** 2 September 2024; **Published:** 20 September 2024

conventional worst case design, all possible partial products are generated and propagated. To decrease the hardware complexity of the exact computations, the approximate arithmetic computations scheme is established. It depends on releases the long channel of generating and shifting of all partial products. Instead of implementing all the required logic gates to compute certain function, cutting down some partial products helping to relax the hardware complexity. Figure 1 illustrates the partial products of two-bit multiplication operation; Figure 1(a), the exact partial products, Figure 1(b), the approximation partial products. While Figure 2 shows the exact implementation of two-bit multiplication process and the approximation implementation for the same process.

The common approximation technique depends on eliminating some less significant bits to decrease the hardware cost. For two-bit multiplication process showing in Figures 1 and 2, the exclude of the first generated partial product has led to elimination of an AND gate from the hardware resources. Another example for 8-bit approximate multiplication process is presented in Figure 3. To decrease the hardware complication and relax the limitations in power and time, some of the least significant groups of partial products have to not be established and another some of generated partial products may be discarded. As shown in Figure 3, eight-bit approximate multiplication process has been established with ten unformed partial products and eleven discarded generated partial products. Also, the division process is probably one of the most challenging function of the basic arithmetic operations. It is extremely complex and consumes much more power and time. To decrease the cost of hardware for the implementation of division function, a truncation procedure is taken place. By using this approach, the processing time and, thus, the overall time needed to complete division are significantly decreased. Consequently, some level of uncertainty is exhibited in the output which degrades the system performance. However, Because DSP applications are error resilient, processing and calculations can occur with some errors without compromising overall performance. [3, 4]. Consequently, the approximation arithmetic calculations approach may be used to decrease the overhead in terms of power, area, and latency by relaxing the establishment of partial products. Algorithms for approximate calculations are effective on hardware and provide good accurate results [5-13]. In order to attain the demand for low-cost, low-power implementation of complex functions with acceptable level of accuracy, a logarithm based computations is proposed in this paper.

For the multiplication function approximation implementation, in previous work, in order to decrease the load of partial product generation and the complexity of the hardware, approximate multipliers cut the least significant groups of the partial product arrangement [5, 13]. This model, which trades precision for a large energy savings, is known as the array-based approximation arithmetic computation model for multipliers. Another promising alternatives to reach high energy efficiency is the approximate logarithmic number system arithmetic [8].

In order to reduce the produced partial products, approximate logarithmic multiplication scheme has been utilized. This logarithmic-based multipliers relax the production and addition of the partial products by replacing these power and time consuming procedures by a simple addition and logarithm and antilogarithm converters. Moreover, the logarithmic techniques [14-26] exploits the simple hardware implementation of the piecewise linear logarithm and antilogarithm converters.

Else, Approximate logarithmic multiplication has been suggested as a crucial approach for very low power, perhaps replacing earlier approximate multiplication methods. By lowering area, latency, and power consumption at the expense of approximation error, all of these approximation strategies contribute to the end outcome.

In this paper, the proposed multiplication and division hardware design is demonstrated in section II. Section III reports the VLSI hardware implementation, synthesis results, and comparison with prior works. At the end, the conclusion of this paper is demonstrated in section IV.

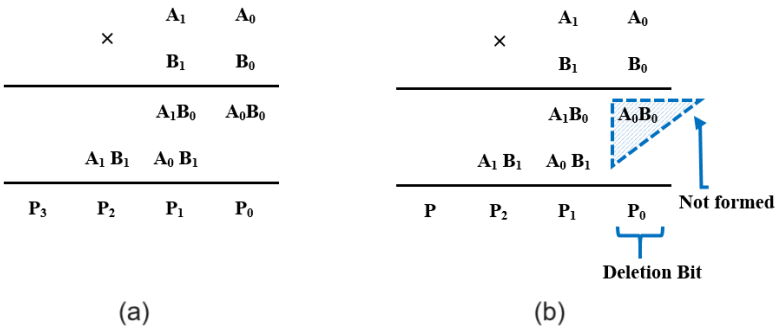

Figure 1. *Partial products of two-bit multiplication process; (a) the exact partial products, (b) the approximation partial products.*

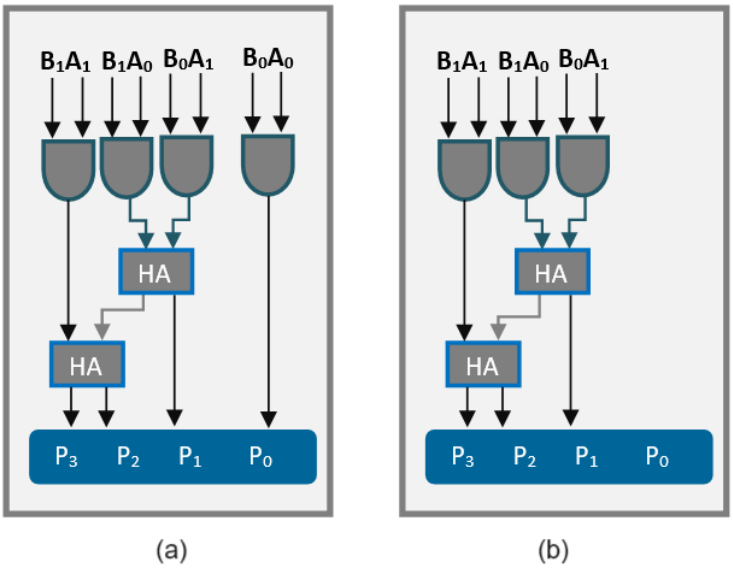

Figure 2. *The Hardware Implementation of (a) 2-bit exact multiplication operation, (b) 2-bit approximate multiplication operation.*

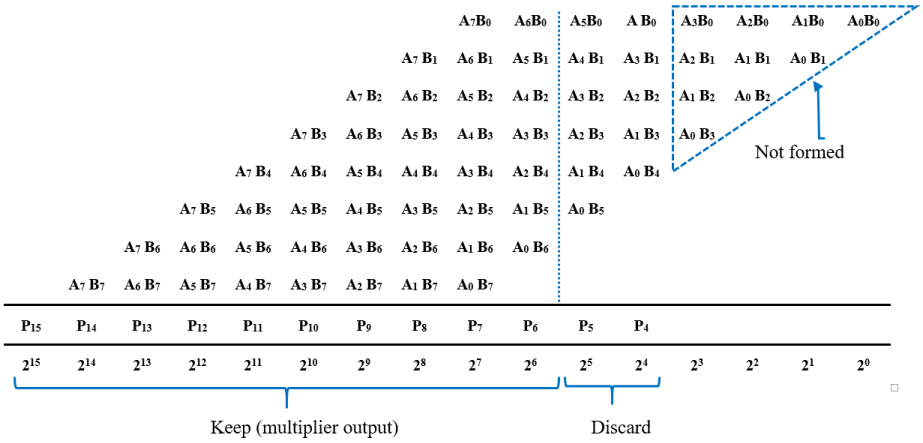

Figure 3. *Partial products of 8-bit approximate multiplication process.*

2. Proposed Multiplication & Division Hardware Design

Logarithmic arithmetic computations [14, 24, 25] have been proposed to decrease the process strength. They make use of the fact that the complex functions are converted to less complex process when applying the logarithm algorithm. By applying the logarithm approximation algorithm on the multiplication and division process as follow:

$$M = A \times B$$
$$log_2 M = log_2 A + log_2 B \quad (1)$$
$$\therefore M = 2^{[(log_2 A)+(log_2 B)]}$$

$$D = A \div B$$
$$log_2 D = log_2 A - log_2 B \quad (2)$$
$$\therefore D = 2^{[(log_2 A)-(log_2 B)]}$$

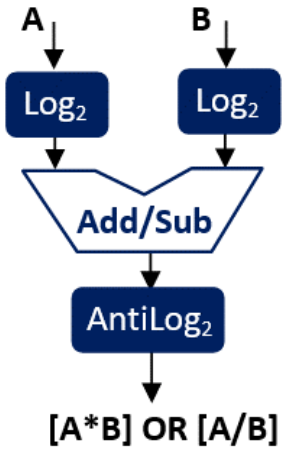

Figure 4. *The block diagram of the multiplication and division process in logarithm algorithm.*

The multiplication process is transformed to addition process as states by Equation (1). These algorithm is based on the logarithmic converters and antilogarithmic converters. As illustrated in Figure 4, the conversion from multiplication process to addition process starts by converting the multiplier input A and the multiplicand input B to their logarithm value $\log_2 A$ and $\log_2 B$, respectively. The second step is attained by applying the addition operation on the two logarithmic values, $\log_2 A$ and $\log_2 B$. In order to complete the multiplication process, the antilogarithm converter is operated to transform the results of the addition from logarithmic value to binary value. The same procedures have been applied for the division process except replacing the addition operation by subtracting operation. The logarithmic approximation algorithm depends considerably on the transformations from binary to logarithm or the transformations from logarithm to binary. Broadly, the implementation of logarithm and antilogarithm converters based on the piecewise linear approximation [14-24] and can be divided into two mainly categories; uniform segmentation approach and nonuniform segmentation approach. Several previous piecewise linear approximation logarithm and antilogarithm converters have been proposed to decrease the hardware complexity and minimize the consumed power. In these approximation approaches, the conversion coefficients are selected to be power of two for straightforward hardware implementation. Accordingly, the implementation of the logarithm and antilogarithm converters includes only shift and add processes. However, due to these approximations, a reduction in accuracy is attained. Some of the previous work proposes a large divisions to increase the accuracy level, however, the hardware cost increase significantly and consume more power.

The employed logarithmic and antilogarithmic converters in this work is the uniform eight-region piecewise linear approximation approach to achieve an agreeable level of accuracy with reduction in hardware implementation and power consumption [14]. The proposed converters save power by up to 5% for the logarithmic converter and up to 37% in antilogarithmic converter as compared with previous work with approaching same degree of accuracy. The hardware architecture of the multiplication and division operations are presented in Figure 5. The logarithmic and antilogarithmic converters hardware includes just shifting and adding operations because of the conversion coefficients are a combination of power of two. The leading one detector (LOD) is the core of the logarithmic converters to distinguish the location of the most significant one bit [26]. The remaining of the architecture is the shifting and adding operations to complete the logarithmic conversion. Similarly the architecture of the antilogarithmic converters except that it doesn't include LOD which makes it much simpler [16-18].

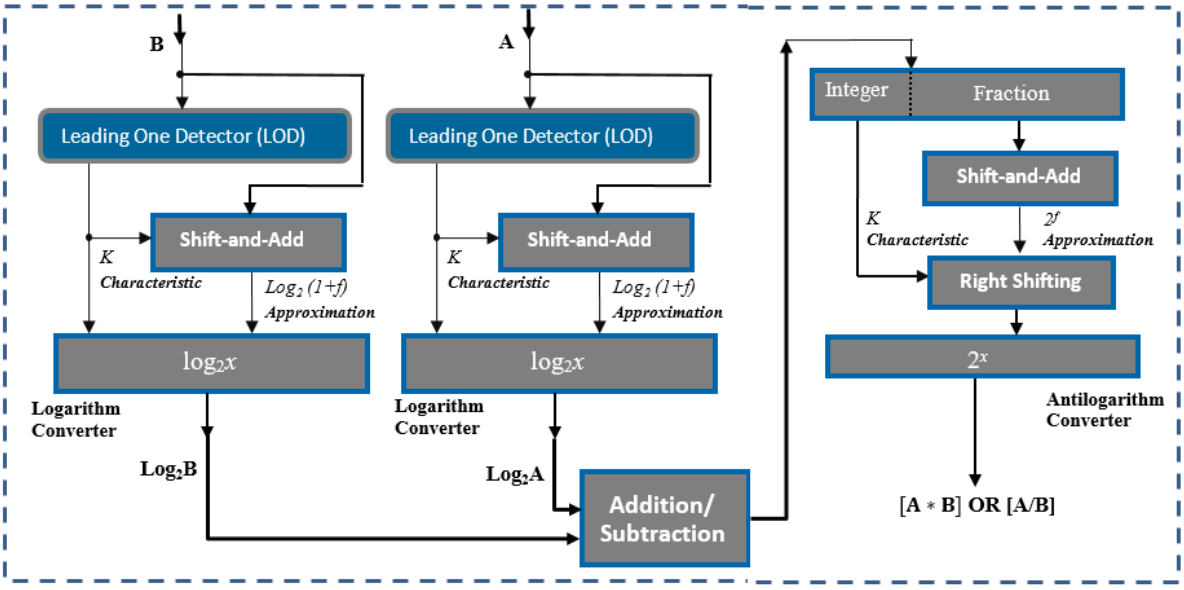

Figure 5. *Proposed structure of the multiplication and division operations.*

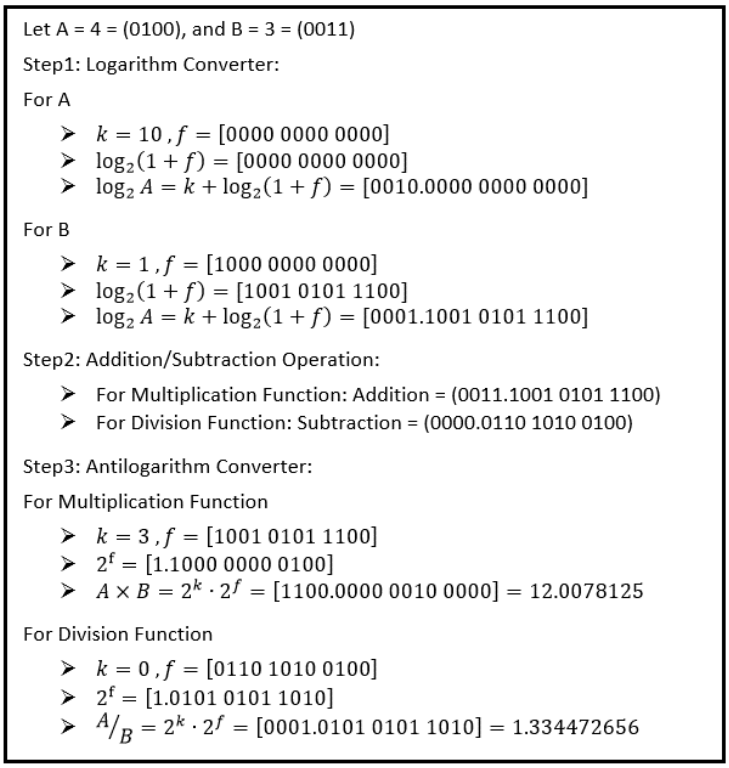

Figure 6. *Example of the proposed algorithm for Multiplication and Division Operation.*

As presented in Figure 6, an example by numbers are given to illustrate the proposed logarithmic approximation algorithm for multiplication and division computations. An input operand A equals 4 decimal $(0100)_2$ and an input operand B equals 3 decimal $(0011)_2$. The first step is attained by using the logarithmic converters. The transformation from binary domain to logarithmic domain starting by determine the location of the most significant one (k) by employing the LOD in the logarithmic converter structure. For the input operands A and B, the leading one position (k) is 10 and 01, respectively. After the decision of the leading one position, the remaining bits are considered the fraction bits. In this paper, a low power, high accuracy piecewise linear eight-segment uniform approximation logarithmic and antilogarithmic converters [14] are employed to perform the required conversion from logarithmic domain to the binary domain or vice versa. The logarithm fraction bits are determined according to the approximated coefficients in the appointed segment. After completing the conversion from binary domain to logarithmic domain by determine the k bits which represents the integer bits and the fraction bits, the step two is initiated. An addition or subtraction of the two logarithmic values of the two input operand A and B are achieved for the multiplication or division operations, respectively. In the third step, the antilogarithmic converter is employed to transform the results from the addition or subtraction from logarithmic domain to binary domain. This conversion is simpler than the logarithmic conversion. The integer bits represent the shifting (k) of the determined fraction bits. For example, the shifting (k) equals 3 and 0 for the results of the addition and subtraction, respectively. The fraction bits are determined corresponding to the approximation coefficients in the appointed division of the antilogarithmic converter [14]. After this conversion, the multiplication or division of the two input operands A and B are accomplished by a certain level of accuracy as presented in Figure 6. The implementation results of the proposed approximate logarithm based multiplication and division operations are presented in the next section.

3. Hardware Implementation and Synthesis Results

This section demonstrates the VLSI hardware implementation results of the proposed multiplication and division calculation logarithm-based compared with prior works. The proposed approximate scheme is implemented in structure level Very High-Speed Integrated Circuit Hardware Description Language, then synthesized using 90-nm CMOS technology, 1.0 V supply voltage standard cell library. Comparison with the accurate multiplication process, prior work, and the proposed algorithm is listed in Table 1.

Table 1. *Comparison performance between the proposed multiplication and division implementation and previous work.*

Approach	Process (nm)	Number of Bits	Power (mW)	Area (μm^2)	Delay (ns)	Error Metrics			
						MAE	RED	MRED	MSE
Accurate Multiplication	90	16	2.85	6130	4. 59	0	0	0	0
[5] 2015	90	16	1.2	3755	2.99	--	--	--	0.15
[6] 2020	28	8	0.288	157	0.242	3.21	--	0.251	0.175
[7] 2018	28	8	0.046	111.14	0.62	0.93	1.11	--	--
[8] 2015	28	8	0.051	255.3	1.64	0.28	--	0.0296	0.0068
[9] 2014	40	16	1.98	700	0.65	1	2.7	--	--
[10] 2014	28	8	0.055	71.2	0.189	0.067	0.758	0.075	0.01
[11] 2011	28	16	0.662	727	0.318	0.69	--	0.009	0.003
[12] 2011	180	16	0.188	3839	6.382	0.02	--	0.93	0.0013
[13] 2011	90	16	1.196	3755	2.99	--	--	--	0.154
This Work	90	16	0.738	5710	1.8	0.02	2	0.05	0.002

In Table 1, technology, the number of bits, power consumption, area, delay, and error metrics are demonstrated in the second–tenth column.

To determine the accuracy performance of the proposed work and the prior works, several error performance parameters are used. Among them are the maximum absolute error (MAE), the relative error distance (RED), the mean relative error distance (MRED), and the mean squared error (MSE). The definition of the error performance parameters are defined as:

$$\text{MAE} = \max[\text{Exact square value} - \text{approximated square value}] \quad (3)$$

$$\text{RED} = \frac{\text{Exact square value} - \text{approximated square value}}{\text{Exact square value}} \quad (4)$$

$$\text{MRED} = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{\text{approximated value} - \text{true value}}{\text{true value}}\right] \quad (5)$$

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{\text{approximated value} - \text{true value}}{\text{true value}}\right]^2 \quad (6)$$

N represents the total number of potential input combinations. The mean error of the proposed scheme is 0.05. Our proposed approximate scheme based on logarithm algorithm has been compared against different approximate schemes for multiplication operation. The proposed approximate work exhibits good accuracy level that can be accepted for DSP applications. Compared with an exact multiplier, enhancements are obtained in power, and speed at least 74%, and 60%, respectively.

The implementation results of the proposed algorithm accomplishes power saving at least 62% as compared to approximate multiplication scheme [9]. Moreover, at least 70% saving in delay is accomplished with greater accuracy level as compared with [12]. For the proposed logarithmic-based multiplication operation, the power is reduced by up to 38% as compared with the prior work [5] and [13] in 90 nm process, and runs 1.66 higher speed. The hardware complexity for approximate multiplication and division operations is reduced significantly by applying the logarithm-based algorithm. With sufficient accuracy, the proposed algorithm uses less power and lowers hardware costs. Comparing the proposed scheme to prior approximation schemes, the hardware complexity and power constraints are relaxed. When the number of output bits increases and the cost of the hardware rises, it becomes worthwhile.

4. Conclusions

In this paper, an approximate low power multiplication and division operations are achieved. The logarithm-based scheme is used to decrease the hardware complexity with sufficient accuracy level. The multiplication and division operations are implemented by utilizing efficient piecewise linear logarithm and antilogarithm converters which leads to less hardware complexity and high power saving. Comparing the proposed algorithm with the accurate scheme, the proposed logarithm-based scheme accomplishes at least 74% saving in power and operates 2.55 higher speed.

Abbreviations

DSP	Digital Signal Processing
ML	Multiplication Function
DV	Division Function
SQ	Square Function
SQRT	Square Root Function
RMS	Root Mean Squares
THD	Total Harmonic Distortion
LOD	Leading One Detector
MAE	Maximum Absolute Error
RED	Relative Error Distance
MRED	Mean Relative Error Distance
MSE	Mean Squared Error

Author Contributions

Dina Mohamed Ellaithy is the sole author. The author read and approved the final manuscript.

Funding

This work is supported by Electronics Research Institute (ERI), Cairo, Egypt.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Avramović, Aleksej, Zdenka Babić, Dušan Raič, Drago Strle, and Patricio Bulić. "An approximate logarithmic squaring circuit with error compensation for DSP applications." Microelectronics Journal. 2014, 45(3), 263-271. https://doi.org/10.1016/j.mejo.2014.01.005

[2] Ansari, Mohammad Saeed, Bruce F. Cockburn, and Jie Han. "Low-power approximate logarithmic squaring circuit design for DSP applications." IEEE Transactions on Emerging Topics in Computing. 2020, 10(1), 500-506. https://doi.org/10.1109/TETC.2020.2989699

[3] Loukrakpam, Merin, Ch Lison Singh, and Madhuchhanda Choudhury. "Energy-efficient approximate squaring hardware for error-resilient digital systems." In 2018 IEEE Electron Devices Kolkata Conference (EDKCON). 2018, pp. 202-206. IEEE. https://doi.org/10.1109/EDKCON.2018.8770453

[4] Reddy, K. Manikantta, Moodabettu Harishchandra Vasantha, YB Nithin Kumar, and Devesh Dwivedi. "Design of approximate booth squarer for error-tolerant computing." IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2020, 28(5), 1230-1241. https://doi.org/10.1109/TVLSI.2020.2976131

[5] Shao, Botang, and Peng Li. "Array-based approximate arithmetic computing: A general model and applications to multiplier and squarer design." IEEE Transactions on Circuits and Systems I: Regular Papers. 2015, 62(4), 1081-1090. https://doi.org/10.1109/TCSI.2015.2388839

[6] A.G. Strollo, De Caro, D., Napoli, E., Petra, N. and Di Meo, G., "Low-Power Approximate Multiplier with Error Recovery using a New Approximate 4-2 Compressor," *In IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020, pp. 1-4, IEEE. https://doi.org/10.1109/ISCAS45731.2020.9180767

[7] M.S. Ansari, Mrazek, V., Cockburn, B.F., Sekanina, L., Vasicek, Z. and Han, J., "Improving the Accuracy and Hardware Efficiency of Neural Networks using Approximate Multipliers," IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2019, 28(2), pp. 317-328. https://doi.org/10.1109/TVLSI.2019.2940943

[8] Mohammad Saeed Ansari, Bruce F. Cockburn, and Jie Han, "An Improved Logarithmic Multiplier for Energy-Efficient Neural Computing," *IEEE Transactions on Computers*. 2020, 70(4), pp. 614-625. https://doi.org/10.1109/TC.2020.2992113

[9] D., Strollo Esposito, A.G.M., Napoli, E., De Caro, D. and Petra, N., "Approximate Multipliers Based on New Approximate Compressors," IEEE Transactions on Circuits and Systems I: Regular Papers. 2018, 65(12), pp. 4169-4182. https://doi.org/10.1109/TCSI.2018.2839266

[10] Strollo, Antonio Giuseppe Maria, Ettore Napoli, Davide De Caro, Nicola Petra, Gerardo Saggese, and Gennaro Di Meo. "Approximate multipliers using static segmentation: Error analysis and improvements." IEEE Transactions on Circuits and Systems I: Regular Papers. 2022, 69(6), 2449-2462. https://doi.org/10.1109/TCSI.2022.3152921

[11] A. G. M. Strollo, Napoli, E., De Caro, D., Petra, N. and Di Meo, G., "Comparison and Extension of Approximate 4-2 Compressors for Low-Power Approximate Multipliers," IEEE Transactions on Circuits and Systems I: Regular Papers. 2020, 67(9), pp. 3021-3034. https://doi.org/10.1109/TCSI.2020.2988353

[12] S. Nambi, Kumar, U. A., Radhakrishnan, K., Venkatesan, M. and Ahmed, S. E., "DeBAM: Decoder-Based Approximate Multiplier for Low Power Applications," IEEE Embedded Systems Letters. 2020, 13(4), pp. 174-177. https://doi.org/10.1109/LES.2020.3045165

[13] Shao, Botang, and Peng Li. "A model for array-based approximate arithmetic computing with application to multiplier and squarer design." In Proceedings of the 2014 international symposium on Low power electronics and design, 2014, pp. 9-14. https://doi.org/10.1145/2627369.2627617

[14] D. M. Ellaithy, "Approximate Square and Square Root Calculation without Multiplication or Division for DSP Applications," Journal of Advanced Research in Applied Sciences and Engineering Technology. 2024, 48(2), pp. 121-135. https://doi.org/10.37934/araset.48.2.121135

[15] Mitchell, John N. "Computer multiplication and division using binary logarithms." IRE Transactions on Electronic Computers 4(1962): 512-517. https://doi.org/10.1109/TEC.1962.5219391

[16] Dina M. Ellaithy, Magdy A. El-Moursy, Ghada Hamdy, Amal Zaki, and Abdelhalim Zekry. "Efficient piecewise non-uniform approximation logarithmic and antilogarithmic converters." In 2017 Intl Conf on Advanced Control Circuits Systems (ACCS) Systems & 2017 Intl Conf on New Paradigms in Electronics & Information Technology (PEIT). 2017, pp. 149-152. IEEE. https://doi.org/10.1109/ACCS-PEIT.2017.8303034

[17] Chao-Tsung Kuo, and Tso-Bing Juang. "Design of fast logarithmic converters with high accuracy for digital camera application." Microsystem Technologies. 2018, 24. 9-17. https://doi.org/10.1007/s00542-016-3105-y

[18] Loukrakpam, Merin, and Madhuchhanda Choudhury. "Error-aware design procedure to implement hardware-efficient antilogarithmic converters." Circuits, Systems, and Signal Processing. 2019, 38(9), 4266-4279. https://doi.org/10.1007/s00034-019-01062-9

[19] Dina M. Ellaithy, Magdy A. El-Moursy, Ghada Hamdy, Amal Zaki, and Abdelhalim Zekry. "Accurate piecewise uniform approximation logarithmic/antilogarithmic converters for GPU applications." In 2017 29th International Conference on Microelectronics (ICM). 2017, pp. 1-4. IEEE. https://doi.org/10.1109/ICM.2017.8268821

[20] Lyu, Fei, Zhelong Mao, Jin Zhang, Yu Wang, and Yuanyong Luo. "PWL-based architecture for the logarithmic computation of floating-point numbers." IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2021, 29(7), 1470-1474. https://doi.org/10.1109/TVLSI.2021.3081572

[21] Chao-Tsung Kuo, and Tso-Bing Juang. "Area-efficient and highly accurate antilogarithmic converters with multiple regions of constant compensation schemes." Microsystem Technologies. 2018, 24, 219-225. https://doi.org/10.1007/s00542-016-3238-z

[22] Loukrakpam, Merin, and Madhuchhanda Choudhury. "Error-aware design procedure to implement hardware-efficient logarithmic circuits." IEEE Transactions on Circuits and Systems II: Express Briefs. 2020, 67(5), 851-855. https://doi.org/10.1109/TCSII.2020.2979937

[23] Xiong, Botao, Yuanfeng Sui, Zhi Jia, Sicun Li, and Yuchun Chang. "Utilize the shift - and - add architecture to approximate multiple nonlinear functions." International Journal of Circuit Theory and Applications. 2021, 49(7), 2290-2297. https://doi.org/10.1002/cta.2994

[24] Dong, Hongxi, Manzhen Wang, Yuanyong Luo, Muhan Zheng, Mengyu An, Yajun Ha, and Hongbing Pan. "PLAC: Piecewise linear approximation computation for all nonlinear unary functions." IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2020, 28(9), 2014-2027. https://doi.org/10.1109/TVLSI.2020.3004602

[25] Ellaithy, Dina M., Magdy A. El-Moursy, Ghada H. Ibrahim, Amal Zaki, and Abdelhalim Zekry. "Double logarithmic arithmetic technique for GPU." In 2017 12th International Conference on Computer Engineering and Systems (ICCES). 2017, pp. 373-376. IEEE. https://doi.org/10.1109/ICCES.2017.8275335

[26] Abed, Khalid H., and Raymond E. Siferd. "CMOS VLSI implementation of a low-power logarithmic converter." IEEE Transactions on Computers. 2003, 52(11), 1421-1433. https://doi.org/10.1109/TC.2003.1244940

Biography

Dina Mohamed Ellaithy received the Ph. D. degree from the electronics and communications department - Ain Shams University, Egypt, in 2018. She is presently a researcher with the Microelectronics Department, Electronics Research Institute (ERI), Cairo, Egypt. Her current research interests include low-power integrated circuit design, low-power arithmetic, approximate arithmetic, logarithmic arithmetic, digital ASIC circuit design, phase locked loop, RF circuit design, power amplifiers, and wireless communication transceivers.

Research Field

Dina Mohamed Ellaithy: low-power arithmetic, approximate arithmetic, logarithmic arithmetic, digital ASIC circuit design, digital integrated circuit design.